2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Conservative Edge Sparsification for Graph SLAM Node Removal

Nicholas Carlevaris-Bianco and Ryan M. Eustice

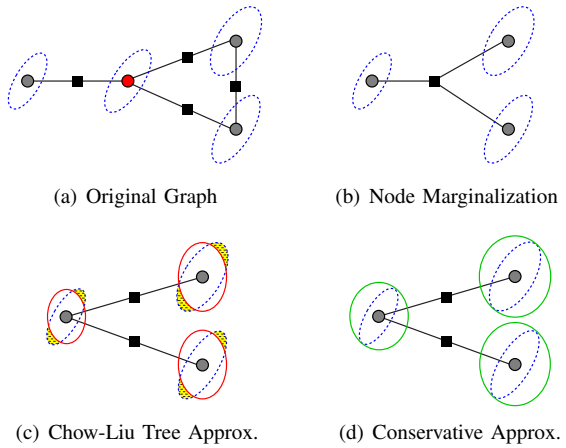
***Abstract*—This paper reports on optimization-based methods for producing a sparse, conservative approximation of the dense potentials induced by node marginalization in simultaneous localization and mapping (SLAM) factor graphs. The proposed methods start with a sparse, but overconfident, Chow-Liu tree approximation of the marginalization potential and then use optimization-based methods to adjust the approximation so that it is conservative subject to minimizing the Kullback-Leibler divergence (KLD) from the true marginalization potential. Results are presented over multiple real-world SLAM graphs and show that the proposed methods enforce a conservative approximation, while achieving low KLD from the true marginalization potential.**

I. Introduction

Graph-based simultaneous localization and mapping (SLAM) [1–7] has been used to successfully solve many challenging SLAM problems in robotics. In graph SLAM, the problem of finding the optimal configuration of historic robot poses (and optionally the location of landmarks), is associated with a Markov random field or factor graph. In the factor graph representation, variables are represented by nodes, and measurements between nodes by factors. Under the assumption of Gaussian measurement noise the graph represents a least squares optimization problem. The computational complexity of this problem is dictated by the density of connectivity within the graph, and by the number of nodes and factors it contains. Therefore, the computational complexity of the graph's optimization problem can be reduced by removing nodes (variable marginalization) and by removing edges from the Markov random field (sparsification).

Methods that directly sparsify the graph connectivity in an information filtering framework include [8–10]. In Thrun et al. [8], weak links between nodes are removed to enforce sparsity. Unfortunately, this removal method causes the resulting estimate to be overconfident (i.e., inconsistent) [11]. In Walter et al. [9], odometry links are removed in order to enforce sparsity in feature-based SLAM.

Methods that remove nodes from the graph include [12–16]. True node marginalization induces dense connectivity between nodes in the elimination clique. In Folkesson and Christensen [12], and in the dense-exact version of generic linear constraints (GLCs) presented by Carlevaris-Bianco and Eustice [14], a linearized factor exactly reproducing the effect of marginalization at the given linearization point is introduced over the elimination clique. Unfortunately, when removing many nodes, the dense connectivity induced by true marginalization quickly compounds, causing a loss of sparsity and greatly increasing the computational complexity of the graph optimization problem. This quickly outweighs the computational benefits of node removal, making these methods impractical for many applications.



(a) Original Graph (b) Node Marginalization

(c) Chow-Liu Tree Approx. (d) Conservative Approx.

Fig. 1: Method overview. Starting with the original factor graph (a), the red node is marginalized. This induces a densely connected factor over the marginalization clique (b). The true uncertainty ellipses (dashed blue lines) are not affected by marginalization. The dense marginalization potential is then approximated using a sparse Chow-Liu tree (c). The uncertainty ellipses after the Chow-Liu tree approximation (red lines) are overconfident (note the yellow regions that are no longer probabilistically plausible). The sparse Chow-Liu tree approximation is adjusted so that it is conservative (d), modifying the uncertainty ellipses (green lines).

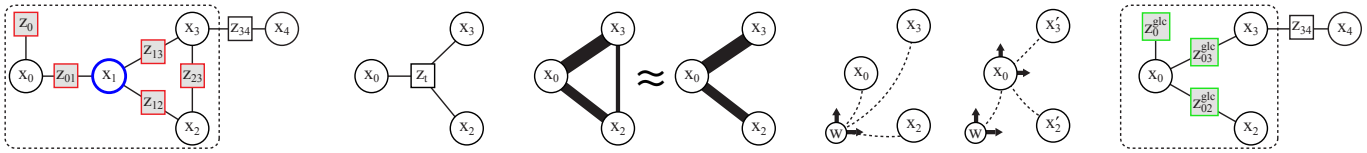
Kretzschmar and Stachniss [13] propose using a Chow-Liu tree (CLT) [17] approximation, calculated over the conditional distribution of the elimination clique, to guide sparse measurement composition in producing new non-linear factors over the elimination clique. This heuristic approximates true marginalization while maintaining sparsity. However, because measurement composition is used to compute the new factors, the true CLT approximation is not computed. Additionally, information may be double counted during measurement composition, producing an inconsistent estimate [14].

The sparse approximate version of GLC proposed in [14] addresses these issues using linear factors to accurately implement the CLT approximation without double counting measurement information. This method was successfully applied to control the computational complexity in long-term multi-session SLAM [15].

*This work was supported in part by the National Science Foundation under award IIS-0746455, the Office of Naval Research under award N00014-12-1-0092, and Ford Motor Company via the Ford-UM Alliance under award N015392.

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) Target Info. (c) Chow-Liu Tree (d) Root Shift (e) Final Graph

Fig. 2: Sample factor graph where node $\mathbf{x}_1$ is to be removed (a). 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 in (a)) are those included in calculating the target information over the marginalization clique $\mathbf{X}_t = [\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_3]$ (b). The original distribution associated with the target information, $p(\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{Z}_m)$, is approximated using the Chow-Liu maximum mutual information spanning tree as $p(\mathbf{x}_0|\mathbf{Z}_m)p(\mathbf{x}_2|\mathbf{x}_0, \mathbf{Z}_m)p(\mathbf{x}_3|\mathbf{x}_0, \mathbf{Z}_m)$ (c). The pairwise potentials are reparameterized with respect to $\mathbf{x}_0$ to avoid linearization in the world-frame (d). New GLC factors are computed and inserted into the graph replacing $\mathbf{Z}_m$ (highlighted in green in (e)). Note that node removal only affects the nodes and factors within the Markov blanket of $\mathbf{x}_1$ (dashed line). **The methods proposed in this paper modify the CLT step (c) in order to ensure that the approximation is conservative.**

The CLT produces an approximation with the lowest Kullback-Leibler divergence (KLD) among all tree structures. Unfortunately, achieving minimum KLD often requires that the CLT approximation be slightly overconfident with respect to the true distribution, as illustrated in Fig. 1(c). In most SLAM applications, conservative estimates are strongly preferred to overconfident estimates. During map building, overconfidence can adversely affect data association, causing the system to miss valid loop closures. Additionally, when using the resulting map, overconfident estimates can lead to unsafe path planning and obstacle avoidance [18]. Here, as in [10] and [16], we define a conservative estimate as one where the covariance of the sparsified distribution is greater than or equal to that of the true distribution, i.e., $\tilde{\Sigma} \geq \Sigma$.

Recently, Vial et al. [10] and Huang et al. [16] have proposed methods that explicitly ensure conservative approximations during graph sparsification. In Vial et al. [10], an optimization-based method is proposed that, given a desired sparsity pattern, minimizes the KLD between the sparsified distribution and the true distribution while ensuring that the sparsified distribution is conservative. This method performs favorably in comparison with [8] and [9], however, as the authors of [10] acknowledge, the computational cost of the optimization grows quickly with the size of the matrix being sparsified. To avoid this they propose a problem reduction that allows their method to be applied to a subset of the graph's variables. The problem reduction still involves the expensive inversion of the block of the information matrix associated with the entire graph beyond the subproblem's Markov blanket, which will also be intractable for large graphs.

The method proposed by Huang et al. [16] performs node marginalization by inducing a densely connected linear factor as in [12] and dense-exact GLC [14]. To perform edge sparsification, the authors formulate an optimization problem that seeks to minimize the KLD of the approximation while requiring a conservative estimate and encouraging sparsity through $\mathcal{L}_1$ regularization. This optimization problem is then applied to the linearized information matrix associated with the entire graph, which limits its applicability to relatively small problems, and prevents relinearization after sparsification. Using $\mathcal{L}_1$ regularization to promote sparsity is appealing because it does not require the sparsity pattern to be specified—instead, it automatically removes the least important edges. However, because the sparsity pattern produced is arbitrary, it is unclear how the resulting information matrix might be decomposed into a sparse set of factors, which is important if one wishes to exploit existing graph SLAM solvers such as iSAM [7, 19].

In this paper we explore optimization-based methods for conservative sparsification of the dense cliques induced by node marginalization. The proposed methods are integrated within the GLC framework proposed in [14] and are designed to maintain the advantages of sparse-approximate GLC. Our proposed methods address some of the aforementioned shortcomings of the methods proposed in [10] and [16].

- Like [10] and [16] our proposed methods ensure that the sparse approximation remains conservative while providing a low KLD from the true distribution.
- Our methods produce a new set of factors using only the current factors as input, and do not require the full linearized information matrix as input as in [10] and [16].
- The computational complexity of our method is dependent only upon the size of the elimination clique, and not on the size of the graph beyond the clique. We do not require a large matrix inversion to formulate the subproblem as in [10], nor do we operate over the entire graph as in [16].

The remainder of this paper is outlined as follows: In Section II we briefly review the GLC node removal framework within which our proposed methods are implemented. The proposed conservative sparsification techniques are described in Section III. In Section IV the proposed methods are evaluated over a variety of real-world SLAM datasets. Finally, a discussion and concluding remarks are provided in Section V and Section VI.

II. Generic Linear Constraint Node Removal

The conservative sparsification methods proposed in this paper are developed within the GLC node removal framework. Therefore, we first provide an overview of GLC node removal. For a full discussion and derivation we refer the reader to [14]. Here we focus on the CLT-based sparse-approximate version of GLC, as the goal of the proposed methods are to produce a sparse approximation with low KLD, with the

additional constraint that the approximation is conservative. Sparse-approximate GLC node removal, illustrated in Fig. 2, is performed as follows:

1) The potential induced by marginalization over the elimination clique is computed.
2) The potential is approximated using a Chow-Liu tree.
3) The variables in the CLT potentials are reparameterized as relative transforms.
4) The CLT potentials are implemented as linear factors, referred to as generic linear constraints, replacing the original factors over the elimination clique in the graph.

The methods proposed in this paper modify the CLT step (step (2), Fig. 2(c)) in order to ensure that the approximation is conservative.

III. Method

The first step in the GLC node removal process is to identify the potential induced in the graph by marginalization. This potential is characterized by its information matrix, which we refer to as 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 $p(\mathbf{X}_m|\mathbf{Z}_m) \sim \mathcal{N}^{-1}(\boldsymbol{\eta}_m, \Lambda_m)$. From Λ_m we can then compute the desired dense target information, Λ_t, by marginalizing out the elimination node using the standard Schur-complement form [5]. Because marginalization only affects the elimination clique, a factor, or set of factors, which induces the potential characterized by Λ_t, will induce the same potential as true node marginalization at the given linearization point.

It is 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 $p(\mathbf{X}_m|\mathbf{Z}_m)$. As a result, Λ_t, and distributions derived from Λ_t, may be low rank.

As in Vial et al. [10] and Huang et al. [16], we wish to minimize the KLD of the sparse approximation while producing a consistent estimate. When the distribution means are equal (i.e., $\boldsymbol{\eta}_t = \Lambda_t \boldsymbol{\mu}_t$ and $\tilde{\boldsymbol{\eta}}_t = \tilde{\Lambda}_t \boldsymbol{\mu}_t$), the KLD between the marginalization-induced factor characterized by Λ_t and its approximation characterized by $\tilde{\Lambda}_t$ is given by

$$\begin{aligned} &\mathcal{D}_{KL}\left(\mathcal{N}^{-1}(\boldsymbol{\eta}_t, \Lambda_t) \| \mathcal{N}^{-1}(\tilde{\boldsymbol{\eta}}_t, \tilde{\Lambda}_t)\right) \\ &= \frac{1}{2}\left(\operatorname{tr}(\tilde{\Lambda}_t \Lambda_t^{-1}) + \ln \frac{|\Lambda_t|}{|\tilde{\Lambda}_t|} - \dim(\boldsymbol{\eta}_t)\right). \end{aligned} \tag{1}$$

Noting that Λ_t and the state dimension are constant, the KLD optimization objective with respect to $\tilde{\Lambda}_t$ can be written as

$$f_{KL}(\tilde{\Lambda}_t) = \operatorname{tr}(\tilde{\Lambda}_t \Lambda_t^{-1}) - \ln |\tilde{\Lambda}_t|. \tag{2}$$

However, as previously mentioned, Λ_t will, in general, be low rank, making the KLD ill defined. In [10], a full rank subproblem is defined, but its implementation requires inverting the information matrix associated with the rest of the graph beyond the subproblem's Markov blanket. In [16], optimization is performed over the full information matrix, which will always be full rank for well-posed SLAM graphs.

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

$$\Lambda_t = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_q \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_q \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_q^\top \end{bmatrix} = \mathrm{U}\mathrm{D}\mathrm{U}^\top, \tag{3}$$

where U is a $p \times q$ orthogonal matrix, D is a $q \times q$ matrix, p is the dimension of Λ_t, and $q = \operatorname{rank}(\Lambda_t)$. Noting that the KLD is invariant under parameter transformations, we rewrite the KLD objective as

$$f_{KL}(\tilde{\Lambda}_t) = \operatorname{tr}(\mathrm{U}^\top \tilde{\Lambda}_t \mathrm{U}\mathrm{D}^{-1}) - \ln |\mathrm{U}^\top \tilde{\Lambda}_t \mathrm{U}|. \tag{4}$$

When Λ_t is full rank

$$\begin{aligned} f_{KL}(\tilde{\Lambda}_t) &= \operatorname{tr}(\tilde{\Lambda}_t \mathrm{U}\mathrm{D}^{-1}\mathrm{U}^\top) - \ln |\mathrm{U}^\top||\tilde{\Lambda}_t||\mathrm{U}| \\ &= \operatorname{tr}(\tilde{\Lambda}_t \Lambda_t^{-1}) - \ln |\tilde{\Lambda}_t|, \end{aligned}$$

which is exactly equivalent to (2). When Λ_t is low rank (4) computes the KLD over the subspace where Λ_t is well defined. This allows us to work with the low-rank target information and limit the extent of the optimization problem to the elimination clique. Intuitively, this parameter transformation can be thought of as using the pseudo-inverse [20] of Λ_t to compute the KLD. However, it is important that the transformation be applied to $\tilde{\Lambda}_t$ so that, during optimization, $\ln |\mathrm{U}^\top \tilde{\Lambda}_t \mathrm{U}|$ is evaluated instead of $\ln |\tilde{\Lambda}_t|$, which will be undefined because the optimal $\tilde{\Lambda}_t$ will also be low-rank.

A. Chow-Liu Tree Approximation

The original version of sparse-approximate GLC approximated the marginalization-induced factor using a Chow-Liu tree. The CLT produces the minimum KLD among all trees by computing the pairwise mutual information between all nodes, and building the maximum mutual information spanning tree. The CLT can be expressed as

$$\mathcal{N}^{-1}(\boldsymbol{\eta}_t, \Lambda_t) \approx \mathcal{N}^{-1}(\tilde{\boldsymbol{\eta}}_t, \tilde{\Lambda}_{\mathrm{CLT}}) = \prod_i p(\mathbf{x}_i|\mathbf{x}_{\mathrm{p}(i)}), \tag{5}$$

where $\mathbf{x}_{\mathrm{p}(i)}$ is the parent of $\mathbf{x}_i$, and for the root of the CLT $p(\mathbf{x}_0|\mathbf{x}_{\mathrm{p}(0)}) = p(\mathbf{x}_0)$. The information added to the graph by the CLT approximation can then be written as

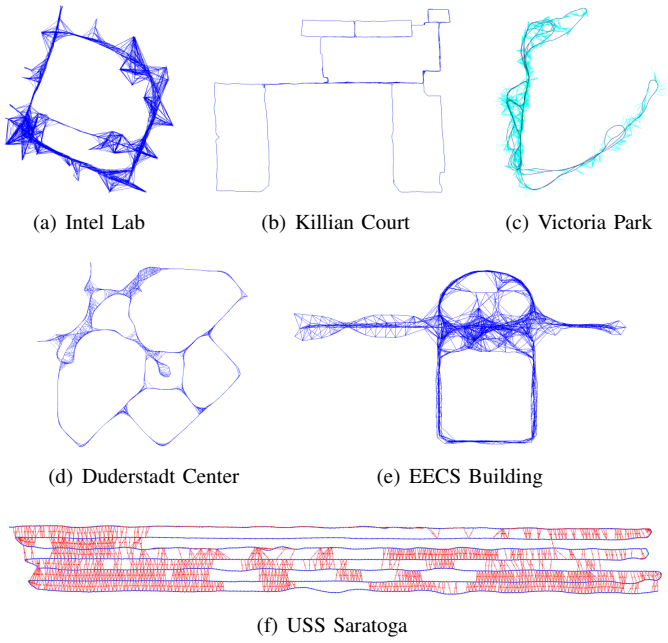
$$\tilde{\Lambda}_{\mathrm{CLT}} = \sum_i \Psi_i, \tag{6}$$

where each Ψ_i is the information associated with one of the unary or binary factors in the tree, padded with zeros so that the appropriate dimensions are achieved.

The methods proposed in this paper all start with the CLT approximation and then use optimization methods to "adjust" the approximation to ensure that it is conservative. Intuitively, this can be thought of as numerically growing the uncertainty of the CLT so that it is conservative, while minimizing the additional KLD from the true distribution. Each method will produce, by construction, an approximation with the same sparsity pattern as the CLT.

TABLE I: Experimental Datasets

Dataset	Node Types	Factor Types	# Nodes	# Factors
Intel Lab	3-DOF pose	3-DOF odometry, 3-DOF laser scan-matching	910	4,454
Killian Court	3-DOF pose	3-DOF odometry, 3-DOF laser scan-matching	1,941	2,191
Victoria Park	3-DOF pose, 2-DOF Landmark	3-DOF odometry, 2-DOF landmark observation	7,120	10,609
Duderstadt Center	6-DOF pose	6-DOF odometry, 6-DOF laser scan-matching	552	1,774
EECS Building	6-DOF pose	6-DOF odometry, 6-DOF laser scan-matching	611	2,134
USS Saratoga	6-DOF pose	6-DOF odometry, 5-DOF monocular-vision, 1-DOF depth	1,513	5,433 %



(a) Intel Lab (b) Killian Court (c) Victoria Park

(d) Duderstadt Center (e) EECS Building

(f) USS Saratoga

Fig. 3: Graphs used in evaluation. Blue links represent full-state (3-DOF or 6-DOF) relative-pose constraints from odometry and laser scan-matching. Red links represent 5-DOF relative-pose constraints modulo-scale from monocular vision. Cyan links represent landmark observation factors.

B. Covariance Intersection

As discussed in §I, the CLT approximation is often over-confident in practice. This is due to the fact that the tree structure is not, in general, capable of capturing the full correlation structure of the original distribution. The covariance intersection algorithm, proposed in [21], can be used to consistently merge measurements with unknown correlation and can be used to weight the CLT factors so that their sum is conservative. Clearly, we should be able to do better than covariance intersection because the true correlation in the original distribution, Λ_t, is known. Covariance intersection, however, does provide an easy-to-compute lower bound on the approximation performance to which we can compare additional methods. Additionally, it provides a strictly-feasible starting point for more complex optimization problems. The approximate target information produced by covariance intersection is defined as a convex combination of the CLT factors,

$$\tilde{\Lambda}_{\mathrm{CI}}(\mathbf{w}) = \sum_i w_i \Psi_i, \tag{7}$$

where each w_i scales the information added by each factor. The optimal weights can then be found by solving the convex semidefinite program,

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & f_{KL}(\tilde{\Lambda}_{\mathrm{CI}}(\mathbf{w})) \\ \text{subject to} \quad & \sum_i w_i = 1. \end{aligned} \tag{8}$$

C. Weighted Factors

Because the true distribution is known, we can relax covariance intersection's requirement that weights sum to one. Instead we constrain the weights to be between zero and one, and add the conservative constraint proposed in [10] and [16]. We refer to this formulation as "weighted factors," and its approximate target information is defined as

$$\tilde{\Lambda}_{\mathrm{WF}}(\mathbf{w}) = \sum_i w_i \Psi_i. \tag{9}$$

The optimal weights can then be found by solving

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & f_{KL}(\tilde{\Lambda}_{\mathrm{WF}}(\mathbf{w})) \\ \text{subject to} \quad & 0 \leq w_i \leq 1, \ \forall i \\ & \Lambda_t \geq \tilde{\Lambda}_{\mathrm{WF}}(\mathbf{w}), \end{aligned} \tag{10}$$

which is again a convex semidefinite program.

D. Weighted Eigenvalues

Instead of weighting each factor by a single value, finer grained control can be achieved by weighting each factor along its principal axes independently. We refer to this formulation as "weighted eigenvalues." Each factor has an eigen-decomposition given by

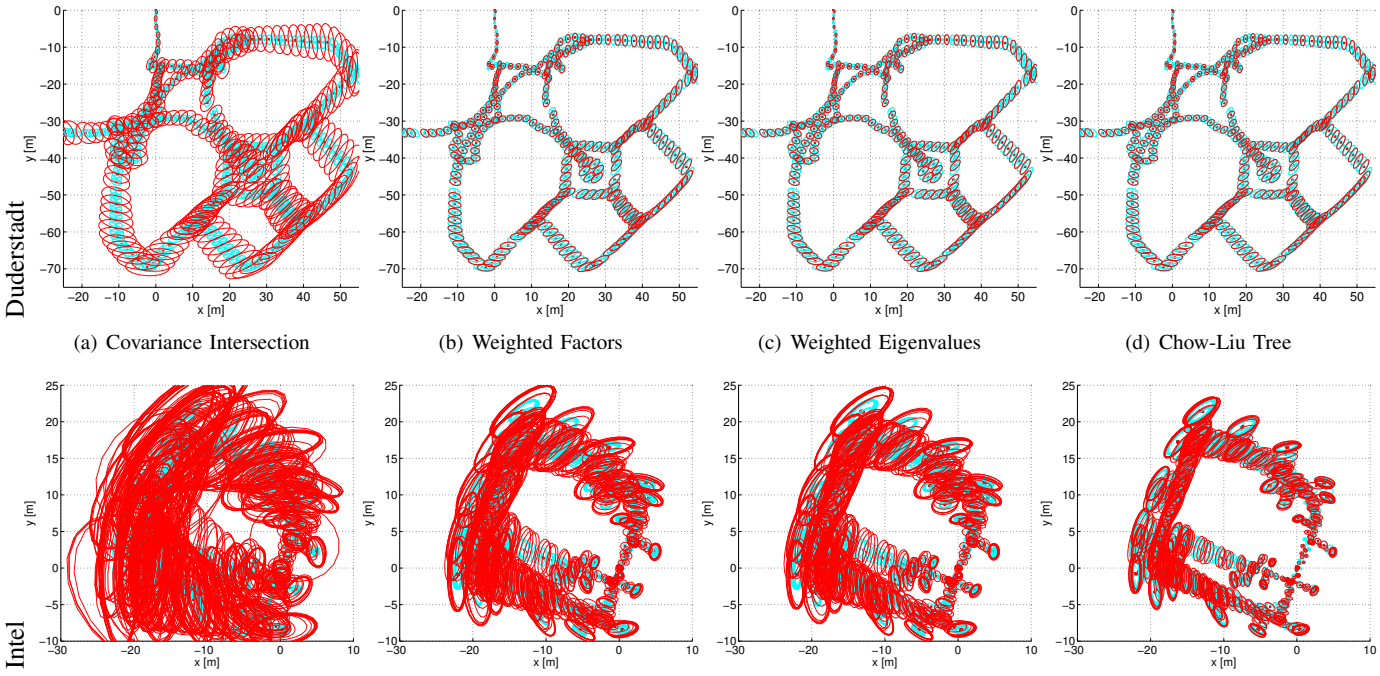
$$\Psi_i = \begin{bmatrix} \mathbf{u}_1^i & \cdots & \mathbf{u}_{q_i}^i \end{bmatrix} \begin{bmatrix} \lambda_1^i & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_{q_i}^i \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{i\top} \\ \vdots \\ \mathbf{u}_{q_i}^{i\top} \end{bmatrix}. \tag{11}$$

Using the eigen-decomposition of each factor we can write the approximate target information as

$$\begin{aligned} \tilde{\Lambda}_{\mathrm{WEV}}(\mathbf{w}) &= \sum_i \sum_{j=1}^{q_i} w_j^i \lambda_j^i \mathbf{u}_j^i \mathbf{u}_j^{i\top} \\ &= \sum_k w_k \lambda_k \mathbf{u}_k \mathbf{u}_k^\top, \end{aligned} \tag{12}$$

with the optimal weights found by solving

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & f_{KL}(\tilde{\Lambda}_{\mathrm{WEV}}(\mathbf{w})) \\ \text{subject to} \quad & 0 \leq w_k \leq 1, \ \forall k \\ & \Lambda_t \geq \tilde{\Lambda}_{\mathrm{WEV}}(\mathbf{w}). \end{aligned} \tag{13}$$



(a) Covariance Intersection (b) Weighted Factors (c) Weighted Eigenvalues (d) Chow-Liu Tree

(e) Covariance Intersection (f) Weighted Factors (g) Weighted Eigenvalues (h) Chow-Liu Tree

Fig. 4: Sample 3-σ uncertainty ellipses for the Duderstadt graph with 50.0% node removal (top row) and the Intel graph with 33.3% node removal (bottom row). True marginal ellipses are shown in cyan, while the marginal ellipses from the approximate distribution are shown in red. For the Duderstadt graph both weighted factors (b) and weighted eigenvalues (c) produce distributions very similar to the CLT (d) while remaining conservative. For the Intel graph both weighted factors (f) and weighted eigenvalues (g) produce similar distributions that are noticeably more conservative than the CLT (h).

Note that the number of optimization variables has increased in comparison to the covariance intersection and weighted factors formulations. As will be demonstrated in §IV, this results in a significant increase in the computational cost.

E. Implementation Considerations

Each of the proposed semidefinite programs are convex and can be efficiently solved using interior point methods [22–24]. Interior point methods require that a strictly-feasible starting point be found before optimization, i.e., an initial approximation, where $0 < w_i < 1, \ \forall i$ and $\Lambda_t > \tilde{\Lambda}_t(\mathbf{w})$. Covariance intersection with uniform weights provides an easy-to-compute strictly-feasible starting point, and is used in all experiments.

For low rank target information, the conservative constraint, $\Lambda_t - \tilde{\Lambda}_t(\mathbf{w})$, is semidefinite and will have at least one zero eigenvalue, and therefore, no strictly-feasible starting point exists. Additionally, it prevents the evaluation of the optimization problem's gradient and Hessian. Instead, the conservative constraint is implemented as

$$\Lambda_t + \epsilon \mathrm{I} \geq \tilde{\Lambda}_t(\mathbf{w}),$$

so that a strictly-feasible starting point exists and the gradient and Hessian can be evaluated. Our experimental evaluation indicates that the actual value of ϵ has very little effect on the results. All experiments are performed with $\epsilon = 0.1$, though values $10^{-5} \leq \epsilon \leq 1$ produced nearly equivalent results.

IV. Experimental Results

To evaluate the proposed methods, we test their performance on a variety of SLAM graphs (summarized in Fig. 3 and Table I), including:

- Two standard 3-degree of freedom (DOF) pose-graphs, *Intel Lab* and *Killian Court*.
- The *Victoria Park* 3-DOF graph with both poses and landmarks.
- Two 6-DOF pose-graphs built using data from a Segway ground robot equipped with a Velodyne HDL-32E laser scanner as the primary sensing modality, *Duderstadt Center* and *EECS Building*.
- A 6-DOF graph produced by a Hovering Autonomous Underwater Vehicle (HAUV) performing monocular SLAM for autonomous ship hull inspection [25], *USS Saratoga*.

The proposed algorithms were implemented using iSAM [7, 19, 26] as the underlying optimization engine. For each graph, the original full graph is first optimized using iSAM. Then the different node removal algorithms are each used to remove a set 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 from the full graph about the optimization point and performing Schur complement marginalization. This provides a ground-truth distribution that we can directly compare our conservative

TABLE II: Experimental KLD Results

		Covariance Intersection		Weighted Factors		Weighted Eigenvalues		Chow-Liu Tree
Dataset	% Removed	KLD	CLT Ratio	KLD	CLT Ratio	KLD	CLT Ratio	KLD
Intel Lab	33.3%	45,954.10	175.85×	4,191.73	16.04×	3,439.63	13.16×	261.33
Killian Court	66.7%	984.30	20.58×	194.85	4.07×	186.40	3.90×	47.82
Victoria Park	75.0%	3,062.49	15.85×	835.38	4.32×	599.00	3.10×	193.24
Duderstadt Center	50.0%	14,552.30	3,038.06×	60.82	12.70×	33.54	7.00×	4.79
EECS Building	25.0%	1,671.48	357.15×	32.31	6.90×	18.49	3.95×	4.68
USS Saratoga	33.3%	11,290.80	13,441.43×	12.13	14.44×	4.63	5.51×	0.84

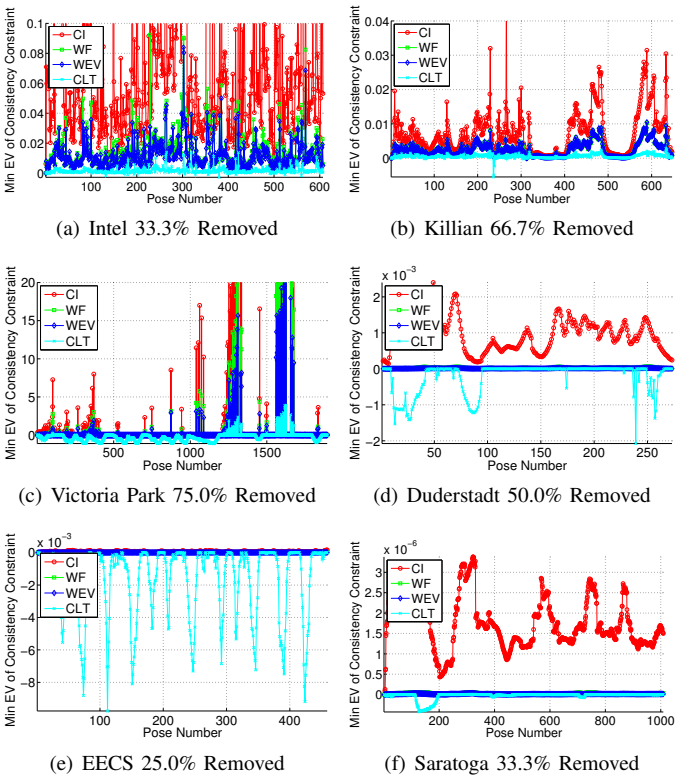
distribution against. In order to provide a benchmark, the CLT approximation as proposed in [14] is also evaluated.

The results for each method, in terms of KLD, are shown in Table II. The "CLT ratio" columns provide a direct comparison with the CLT, which is not guaranteed to be conservative, but serves as a baseline as it is the minimum KLD distribution among all spanning trees. As one would expect, covariance intersection produces a very high KLD because it is excessively conservative. The weighted factors formulation improves the KLD significantly with respect to covariance intersection, while the weighted eigenvectors formulation improves the KLD further still.

For the Duderstadt, EECS, and Saratoga graphs the subjective difference in the quality of the estimates is very small, with weighted factors, weighted eigenvalues, and the CLT producing visually indistinguishable results, see Fig. 4 top row. For the Intel dataset, and to a lesser extent the Killian and Victoria datasets, there appears to be more room for improvement, with a noticeable difference between the weighted eigenvalues result and the CLT for the Intel graph, see Fig. 4 bottom row.

To evaluate the "conservativeness" of the proposed methods, we plot the minimum eigenvalue of the covariance-form consistency constraint for each node marginal, i.e. $\min \lambda(\tilde{\Sigma}_{ii} - \Sigma_{ii})$, depicted in Fig. 5. Values below zero indicate overconfidence, with only the CLT producing overconfident results. Covariance intersection, weighted factors, and weighted eigenvalues all produce conservative estimates, with each producing a slightly tighter estimate (closer to zero) than the previous.

Finally, we consider the computational cost of the proposed methods. A plot showing the node removal time as a function of the number of variables in the elimination clique is shown in Fig. 6. The average node removal times for covariance intersection, weighted factors, weighted eigenvalues and the CLT were 7, 32, 448, and 5 milliseconds, respectively. Even though covariance intersection and weighted factors were both solving optimization problems over the same number of variables, weighted factors is more expensive. This is due to the fact that the equally-weighted covariance intersection solution, used as the initial point for all optimizations, was often very close to the optimal covariance intersection solution and therefore, covariance intersection converged quickly. Weighted eigenvalues solves a larger optimization problem and therefore is substantially slower. Still, the weighted eigenvalues formulation will often have significantly fewer variables than [10] (which optimizes *all non-zero entries* in the upper triangle of the information matrix) and [16] (which optimizes *every entry*



(a) Intel 33.3% Removed (b) Killian 66.7% Removed

(c) Victoria Park 75.0% Removed (d) Duderstadt 50.0% Removed

(e) EECS 25.0% Removed (f) Saratoga 33.3% Removed

Fig. 5: Minimum eigenvalue of the consistency constraint for each pose marginal (i.e., $\min \lambda(\tilde{\Sigma}_{ii} - \Sigma_{ii})$). Covariance intersection, weighted factors, and weighted eigenvalues all produce conservative estimates (values greater than zero), with each producing a slightly tighter estimate (closer to zero) than the previous.

in the upper triangle of the information matrix) for a given information matrix size.

We note that the computational cost of the proposed methods increases quickly with the size of the node removal cliques. However, as experimentally shown in [15], sparse approximate node removal maintains small cliques in real-world SLAM graphs even when removing a very high percentage of nodes. This, in turn, results in essentially constant node removal time regardless of the the number of nodes removed and size of the graph beyond the elimination clique.

V. Discussion

The proposed algorithms are presented in the context of sparsifying the dense cliques produced by node marginalization. However, these techniques could be applied to portions of the graph to perform sparsification, without removing nodes.

All of the presented methods start with the Chow-Liu tree

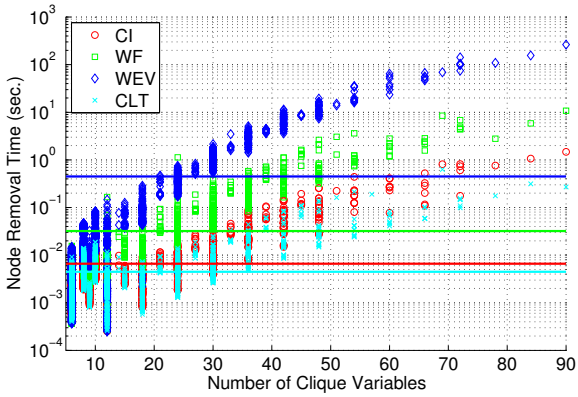


Fig. 6: Node removal processing time as a function of the number of variables in the elimination clique. Average node removal times (solid lines) for covariance intersection, weighted factors, weighted eigenvalues and the CLT were 7, 32, 448, and 5 milliseconds, respectively.

as their basis. We believe that this is a reasonable starting point as it is the minimum KLD spanning tree. However, there is no guarantee that, after using the optimization based methods to ensure a conservative estimate, the CLT's sparsity pattern remains optimal. Furthermore, other non-tree sparsity patterns may be of interest. This is one strongly appealing aspect of the $\mathcal{L}_1$ regularization method proposed in [16] in that it automatically selects the sparsity pattern.

Finally, there is a trade off between the complexity of the optimization problem and the accuracy of the approximation it is able to achieve. Based upon our experimental results, we feel that the weighted factors formulation provides the best trade off between KLD and computation time, as weighted eigenvalues only performs marginally better while being substantially more computationally expensive.

VI. Conclusions

This paper explored several optimization-based methods that can be used to produce a sparse, conservative approximation of potentials induced by node marginalization in SLAM graphs. Starting with the sparse, but overconfident, CLT approximation of the marginalization potential, these methods adjust the CLT approximation so that it is conservative, while minimizing the KLD from the true marginalization potential. Using results from multiple real-world SLAM graphs, we have shown that the algorithm enforces a conservative approximation, while achieving low KLD.

References

[1] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, Apr. 1997.

[2] 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.

[3] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.

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

[5] 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.

[6] 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.

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

[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.*, San Francisco, CA, USA, Sep. 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.*, Edmonton, Alberta, Canada, Aug. 2005, pp. 3281–3288.

[12] J. Folkesson and H. Christensen, "Graphical SLAM—a self-correcting map," in *Proc. IEEE Int. Conf. Robot. and Automation*, New Orleans, LA, April 2004, pp. 383–390.

[13] 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.

[14] N. Carlevaris-Bianco and R. M. Eustice, "Generic factor-based node marginalization and edge sparsification for pose-graph SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, Karlsruhe, Germany, May 2013, pp. 5728–5735.

[15] ——, "Long-term simultaneous localization and mapping with generic linear constraint node removal," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Nov. 2013, pp. 1034–1041.

[16] G. Huang, M. Kaess, and J. J. Leonard, "Consistent sparsification for graph optimization," in *Proc. European Conf. Mobile Robotics*, Barcelona, Spain, Sep. 2013.

[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] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, MA: MIT Press, Sep. 2005.

[19] M. Kaess, H. Johannsson, D. Rosen, N. Carlevaris-Bianco, and J. Leonard, "Open source implementation of iSAM," http://people.csail.mit.edu/kaess/isam, 2010.

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

[21] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, Albuquerque, NM, USA, Jun. 1997, pp. 2369–2373.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[23] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[24] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. on Matrix Analysis and Applicat.*, vol. 19, no. 2, pp. 499–533, 1998.

[25] 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.

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