Approximate Covariance Estimation in Graphical Approaches to SLAM

Gian Diego Tipaldi Giorgio Grisetti Wolfram Burgard

***Abstract*— Smoothing and optimization approaches are an effective means for solving the simultaneous localization and mapping (SLAM) problem. Most of the existing techniques focus mainly on determining the most likely map and leave open how to efficiently compute the marginal covariances. These marginal covariances, however, are essential for solving the data association problem. In this paper we present a novel algorithm for computing an approximation of the marginal. In experiments we demonstrate that our approach outperforms two commonly used techniques, namely loopy belief propagation and belief propagation on a spanning tree. Compared to these approaches, our algorithm yields better estimates while preserving the same time complexity.**

I. Introduction

In the past, different approaches have been proposed for solving the simultaneous localization and mapping (SLAM) problem. One popular class of algorithms are the probabilistic techniques, which deal with the uncertainties in the problem by estimating a probability distribution over the possible problem solutions. Whereas these techniques yield robust solutions, the computational effort required for tracking this distribution can prevent their application to large problem instances.

A second class of algorithms for solving the SLAM problem are the so-called maximum likelihood (ML) approaches. Instead of maintaining a posterior, the goal of these approaches is to calculate the maximum likelihood map based on the observations of the robot and its motions. In ML algorithms, the problem instances are typically described by a graph, whose nodes represent either robot poses or landmark locations. An edge between two nodes represents a relative measurement of them. Finding a maximum likelihood solution to this problem means to determine the assignment of poses to the nodes of the graph which provides the best explanation of the measurements. Traditional ML approaches assume the data associations as given and focus mainly on estimating the position of the nodes, not their uncertainty. However, finding potential data associations requires to estimate the marginal probability distribution over the nodes locations. Still, once the ML configuration of the nodes is known, the marginals can be computed by inverting the (sparse) information matrix of the system. Unfortunately, real world problem instances are often described by graphs having thousands of nodes. Inverting matrices of this size can prevent us from applying ML approaches in real time.

Gian Diego Tipaldi is member of Univ. "La Sapienza", Dipartimento di Informatica e Sistemistica, I00181 Roma, Italy.

Giorgio Grisetti and Wolfram Burgard are members of University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany

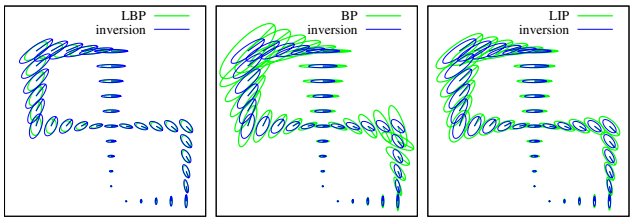

Fig. 1. Marginal covariances computed using the matrix inversion (exact), loopy belief propagation (LBP), belief propagation on a spanning tree (BP) and our approach (LIP). Whereas the exact covariances are depicted in black/dark blue, the approximations are depicted in gray/green. Compared to LBP our approach is more conservative, while it provides tighter estimated than BP.

In this paper we describe a novel algorithm for computing these marginal covariances. Our approach models the SLAM graph as a Gaussian Markov Random Field and performs inference on this representation. A similar idea has originally been proposed by Ranganathan *et al.* [12], who utilized belief propagation as inference algorithm. Our approach outperforms the technique introduced by Ranganathan *et al.* [12] while keeping the same time complexity. Additionally, our algorithm is able to obtain estimates which are closer to the exact ones and generally more conservative. This is important when solving data association problems since over-confident covariances may result in losing valid associations. Figure 1 provides a motivating example and illustrates the result of our approach also in comparison to popular alternative solutions executed on a simple graph.

This paper is organized as follows. In Section II, we discuss related work. Whereas Section III describes how to model the SLAM problem as a Gaussian Markov Random Field, Section IV introduces the approximate inference algorithms based on loopy graphs. We then present our algorithm called loopy intersection propagation (LIP) in Section V. Finally, Section VI contains experimental results illustrating the advantages of our approach.

II. Related Work

The work described in this paper belongs to the family of ML algorithms. One of the first approaches of this type has been proposed by Lu and Milios [8]. Later, Howard *et al.* [6] used Gauss-Seidel relaxation to localize the robot and build a map. Duckett *et al.* [1] proposed Gauss-Seidel relaxation to minimize the error in the network of constraints. Their approach has been subsequently extended by Frese *et al.* [4] by the introduction of the multi-level relaxation (MLR) framework, which applies relaxation on different resolutions.

Olson *et al.* [10] addressed the problem by using gradient descent on a network described in a form which allows for efficient analytical updates. Graphical SLAM [3] builds a

graphical model of the smoothing problem. It optimizes the graph by defining an energy function for each node and then minimizing this energy.

Whereas these methods mainly focus on estimating the most likely configuration of the map, they leave open how to estimate the uncertainty of the solution. To the best of our knowledge, the only approach which computes both the ML configuration of the nodes ant their marginal distribution has been proposed by Ranganathan *et al.* [12]. They model the smoothing problem as a Gaussian Markov random field (GMRF) and use loopy belief propagation on this model. A complete characterization of GMRFs can be found in the work by Weiss and Freeman [13] and by Malioutov *et al.* [9].

III. SLAM as a Gaussian Markov Random Field

In this section we describe how the SLAM problem can be expressed in the Gaussian random Markov field (GMRF) framework. We first describe the GMRF framework. Then we model the SLAM problem in its delayed-state formulation and present the mapping between them.

A general Markov random field (MRF) is a model for describing the joint probability distribution over set of random variables. More formally, a MRF represents the conditional dependence between variables using an undirected graph $G = (V, E)$. Each vertex $v \in V$ in the graph represents a random variable. An edge $\{u, v\} \in E$ between two nodes represents a dependency between the random variables u and v. The edges and the vertices of the graph are labeled with a set of potential functions ϕ defined over a subset of V.

GMRFs are a particular case of MRF suitable for describing multivariate Gaussian distributions. In this case, we just have singleton potential functions (describing a prior belief) and pairwise potential functions (describing the relationship between two different variables). The full joint distribution can be written as

$$p(x) = \frac{1}{Z} \prod_{i=1}^{n} \phi_i(x_i) \prod_{j=i+1}^{n} \phi_{i,j}(x_i, x_j). \tag{1}$$

Here $\phi_i(x_i)$ represents the prior belief about the variable x_i and $\phi_{i,j}(x_i, x_j)$ represents the stochastic constraint between the variables x_i and x_j.

An alternative formulation of the SLAM problem is to use a delayed-state representation rather than a feature based one [2]. Delayed-state representations do not explicitly model features in the environment. Instead, the state vector is composed only by a sequence of poses. In this representation, raw data are registered to provide virtual observations of pose displacement. These virtual observations arise, for instance, by matching pairwise laser range data or camera images.

When using this representation, the information matrix of the corresponding multivariate Gaussian is exactly sparse, as it has been pointed out by Eustice *et al.* [2]. Providing a topological order over the poses ($i < j$), each pose displacement can be expressed by the following non linear stochastic function

$$\mathbf{x}_j = \mathbf{x}_i \oplus \boldsymbol{\delta}_{ij} + \boldsymbol{\omega}. \tag{2}$$

Here $\boldsymbol{\delta}_{ij}$ is the virtual observation made from the pose i about the pose j, $\oplus$ is the standard motion composition operator and $\boldsymbol{\omega}$ is a zero-mean Gaussian variable with covariance matrix $\mathbf{R}$, representing the uncertainty of the measurement. In the following, we consider the linearized version, being $\mathbf{F}_{ij}$ the Jacobian of the $\oplus$ function and $\tilde{\mathbf{x}}_{ij} = \mathbf{x}_i \oplus \boldsymbol{\delta}_{ij} - \mathbf{x}_j$ the error.

Translating a SLAM problem formulated according to the delayed-state framework into a GMRF is quite straightforward. Since the structure of the graph between the two models is preserved, all we need is to define the nature of the potential functions, such that the resulting joint probability distribution is unchanged.

If we consider the canonical parameterization of the Gaussian, we can express each pairwise potential as

$$\phi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = exp\left\{ \mathbf{c} + \boldsymbol{\eta}_{ij}^T \mathbf{x}_{ij} - \frac{1}{2} \mathbf{x}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{x}_{ij} \right\}, \tag{3}$$

where

$$\mathbf{x}_{ij} \triangleq \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} \tag{4}$$

$$\boldsymbol{\eta}_{ij} \triangleq \begin{bmatrix} \boldsymbol{\eta}_{ij}^i \\ \boldsymbol{\eta}_{ij}^j \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{ij}^T \\ -\mathbf{I} \end{bmatrix} \mathbf{R}^{-1} \tilde{\mathbf{x}}_{ij} \tag{5}$$

$$\boldsymbol{\Omega}_{ij} \triangleq \begin{bmatrix} \boldsymbol{\Omega}_{ij}^{[ii]} & \boldsymbol{\Omega}_{ij}^{[ij]} \\ \boldsymbol{\Omega}_{ij}^{[ji]} & \boldsymbol{\Omega}_{ij}^{[jj]} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{ij}^T \\ -\mathbf{I} \end{bmatrix} \mathbf{R}^{-1} \begin{bmatrix} \mathbf{F}_{ij}^T \\ -\mathbf{I} \end{bmatrix}^T \tag{6}$$

Here, $\boldsymbol{\Omega}$ and $\boldsymbol{\eta}$ are respectively the information matrix and the information vector of a measurement between two nodes. The singleton potentials are generally set to the unity except for the first pose, which is fixed at the origin. In this work we are only interested in the covariance computation. Therefore, we will simplify the model by considering its "translated" version: a zero mean Gaussian with the same covariance matrix.

IV. Belief Propagation on Loopy Graphs

The goal of belief propagation is to compute the marginal distribution over a graphical model by means of local message passing. This algorithm has been introduced by Pearl [11] for inference on Bayesian Network. If the graphical model does not contain loops, this local passing scheme is guaranteed to give the exact solution for the marginals. Loopy belief propagation is an approximated algorithm, which uses the same equation of belief Propagation, but in a graph with cycles. As discussed by Weiss and Freeman [13] and Malioutov *et al.* [9], loopy belief propagation on general graphs computes correct marginal means and generally incorrect covariances.

Belief propagation works by iteratively computing local messages and beliefs for every node in the graph, starting with constant messages. The propagation of the messages is repeated until a fixed point is reached. It can be shown that in the context of trees only two iterations are needed: from leafs to root and vice versa. Using the superscript (t) for denoting the current iteration, the belief parameters (denoted

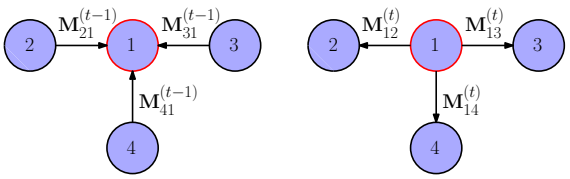

Fig. 2. Message flow of BP on a simple graph at time t for node 1. In the left image node 1 computes its marginal from the messages sent by its neighbors. In the right image, node 1 calculates the new messages and send them . The messages are computed according to (9) and (10).

by $\mathbf{m}^{(t)}$ and $\mathbf{M}^{(t)}$) are given by

$$\mathbf{m}_i^{(t)} = \boldsymbol{\eta}_i + \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^{(t-1)} \quad (7)$$

$$\mathbf{M}_i^{(t)} = \boldsymbol{\Omega}_i + \sum_{j \in \mathcal{N}_i} \mathbf{M}_{ji}^{(t-1)}, \quad (8)$$

where $\boldsymbol{\eta}_i$ and $\boldsymbol{\Omega}_i$ are the parameters of the prior belief (the singleton potential functions), $\mathcal{N}_i$ is the neighboring set of node i, and the messages from node i to node j are defined as

$$\mathbf{m}_{ij}^{(t)} = \boldsymbol{\eta}_{ij}^{j} - \boldsymbol{\Omega}_{ij}^{[ji]} \left(\boldsymbol{\Omega}_{ij}^{[ii]} + \mathbf{M}_i^{(t)} - \mathbf{M}_{ji}^{(t-1)} \right)^{-1} \cdot \left(\boldsymbol{\eta}_{ij}^{i} + \mathbf{m}_i^{(t)} - \mathbf{m}_{ji}^{(t-1)} \right) \quad (9)$$

$$\mathbf{M}_{ij}^{(t)} = \boldsymbol{\Omega}_{ij}^{[jj]} - \boldsymbol{\Omega}_{ij}^{[ji]} \cdot \left(\boldsymbol{\Omega}_{ij}^{[ii]} + \mathbf{M}_i^{(t)} - \mathbf{M}_{ji}^{(t-1)} \right)^{-1} \boldsymbol{\Omega}_{ij}^{[ij]} \quad (10)$$

using the definition in (5) and (6). See Figure 2 for an example of message flow.

When considering the simplified version (zero-mean), all the messages and the beliefs for computing the marginal mean are also zero. This allows us to focus our attention only on the covariance messages. In the remainder of this section, we analyze two different approaches introduced in the SLAM context by Ranganathan *et al.* [12]. The first approach is loopy belief propagation over the full graph, while the second is standard belief propagation over its minimum spanning tree.

A. Loopy Belief Propagation

When loopy belief propagation is applied, the marginal covariances can be either overconfident or conservative. However, Weiss and Freeman [13] showed that they are always overconfident for GMRFs with pairwise cliques, which is the case of SLAM. These estimates often result in a poor approximation of the true marginals, which cannot be used for data association, since this results in valid associations being rejected.

A more information theoretic analysis can be derived by considering how the marginal beliefs of every node are computed. With respect to (8), the marginal belief are computed by summing up all the incoming messages from the neighboring nodes. However, this integration is correct only if the two observations are independent, which is not the case of graphs with loops.

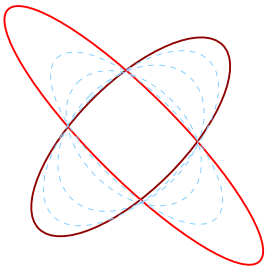

Fig. 4. The shape of the CI update. The thick outer ellipses represent the covariances of $\mathbf{A}$ and $\mathbf{B}$. The dashed ellipses represent resulting covariances of $\mathbf{C}$ by using different values of ω.

B. Belief Propagation over a Spanning Tree

A different approach is to approximate the full graph model by its spanning tree. Since the tree is obtained by eliminating edges (and therefore constraints) from the GMRF and inference on the tree is exact, it is possible to obtain conservative estimate of the true marginal covariances. However, the result of this approximation strongly depends on the property of the tree used. The best results are obtained when using a minimal spanning tree. Moreover, inference on the spanning tree do not consider the loopy structure at all, resulting in too conservative estimates.

Figure 1 show the results obtained by applying the aforementioned algorithms to a simple graph. The graph simulates a robot performing two loops in a indoor environment. As can be seen, LBP produces overconfident estimates while the covariances obtained by using BP on a minimum spanning tree are overly conservative. This is evident in the upper left part of the graph.

V. Loopy Intersection Propagation

In this section, we introduce our approach for computing the marginal covariances on a loopy graph. Before describing our algorithm, we introduce an information fusion framework for dealing with unknown correlations between different estimates. This framework has been introduced by Uhlman and Julier under the name of Covariance Intersection [7].

Covariance Intersection is a fusion rule for combining two different estimates when the correlations between them are unknown. Suppose we have two consistent estimates $\langle \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 \rangle$ and $\langle \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2 \rangle$ for the same Gaussian random variable, expressed in terms of mean and covariance matrix. Furthermore suppose the cross correlation between the two covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_1$ to be unknown.

Covariance Intersection combines the two estimates, in order to obtain a new one $\left\langle \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}} \right\rangle$, according to the following equations

$$\hat{\boldsymbol{\Sigma}} = (\omega \boldsymbol{\Sigma}_1^{-1} + (1-\omega)\boldsymbol{\Sigma}_2^{-1})^{-1} \quad (11)$$

$$\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\Sigma}}(\omega \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + (1-\omega)\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2). \quad (12)$$

If $\omega \in [0, 1]$, the resulting estimate has been proved to be consistent. Moreover, it can be shown that the approach is optimal in the case in which the cross correlations are unknown [7]. Let us consider the plot in Figure 4. This figure shows the covariance ellipses for the two estimates

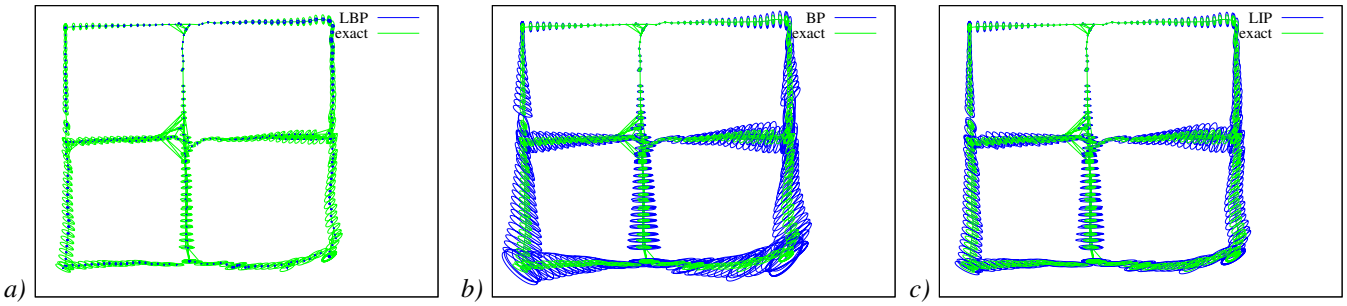

Fig. 3. Comparison of the marginal covariances on the ACES dataset. The exact covariances, computed by inverting the full matrix, are depicted in blue/dark gray, the approximated ones are in green/light gray. The results were obtained using *a)* Loopy Belief Propagation, *b)* Belief Propagation on a Minimum Spanning Tree, and *c)*Loopy Intersection Propagation. Whereas LBP is overconfident and BP on a spanning tree is overly conservative, our approach (LIP) provides an results which are in general closer to the exact estimate.

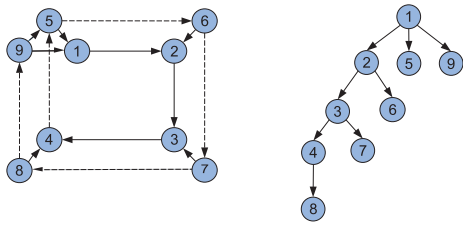

Fig. 5. An example loopy graph (left) and its spanning tree (right). The off-tree edges are the dashed edges in the graph.

(thick outer ellipses) and the resulting update with different values for ω (dashed inner ellipses). The optimal estimate, lies within the intersection region of the estimate ellipses. The CI ellipses always circumscribe this region, resulting in a covariance *bigger* (in a matrix sense) than the exact one.

A. Fusing Loops with Spanning Trees

A natural solution to this problem is to combine the conservativeness of inference on a spanning tree with the information coming from the off-tree edges. In this section, we will describe how this can be achieved using the Covariance Intersection framework. Figure 5 shows an example graph. A spanning tree of this graph is depicted on the right part of the image. The off-tree edges (the ones which are not present in the spanning tree) are depicted as dashed edges in the graph.

Our goal is to obtain a tree approximation from the original graph. Let $\mathbf{\Omega}$ be this joint Information Matrix and $\hat{\mathbf{\Omega}}$ its tree approximation. For any tree-structured Information Matrix there exist a symmetric matrix $\mathbf{K}$ such that

$$\hat{\mathbf{\Omega}} = \mathbf{\Omega} - \mathbf{K}. \tag{13}$$

The matrix $\mathbf{K}$ acts to remove edges from the graph, therefore it is referred to as *cutting matrix*. We will focus our attention on a restricted class of cutting matrices, called *regular* cutting matrices[1].

In order to derive a tree approximation of the GMRF, we have to analyze the structure of the corresponding cutting matrix. For the sake of simplicity, we will restrict the analysis on cutting a single edge, being the extension to multiple edges straightforward.

When using the naive spanning tree approximation, we can express the matrix cutting an edge between the node i and the node j as

$$\mathbf{K}_{ij} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{ij}^{[ii]} & \mathbf{0} & \mathbf{\Omega}_{ij}^{[ij]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{ij}^{[ji]} & \mathbf{0} & \mathbf{\Omega}_{ij}^{[jj]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{14}$$

Cutting this edge results in subtracting the information of an edge from the overall information matrix of the system.

To consider the off-tree information while keeping the regularity condition of the cutting matrix, we are forced to modify its block diagonal entries. Let $\mathbf{P}_{ij}^{[i]}$ and $\mathbf{P}_{ij}^{[j]}$ be the information about node i and j arising from the off-tree edge. The modified cutting matrix will be of the following form

$$\mathbf{K}_{ij} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{ij}^{[ii]} - \mathbf{P}_{ij}^{[i]} & \mathbf{0} & \mathbf{\Omega}_{ij}^{[ij]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{ij}^{[ji]} & \mathbf{0} & \mathbf{\Omega}_{ij}^{[jj]} - \mathbf{P}_{ij}^{[j]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{15}$$

Note that subtracting information from the cutting matrix results in adding this information to the tree approximation $\hat{\mathbf{\Omega}}$ according to Eq. (13).

In the a graphical model perspective, those two pieces of information, $\mathbf{P}_{ij}^{[i]}$ and $\mathbf{P}_{ij}^{[j]}$, can be seen as a prior knowledge about the nodes, acting as singleton potential functions. In other words, we approximate the GMRF with its spanning tree by considering the off-edge information as *prior* knowledge in this approximation.

B. Algorithm

Our approach transforms the graph into a tree augmented with prior information. First it computes a spanning tree on the GMRF. Second, it performs BP on the computed tree, to obtain open loop estimates for the information matrices $\{\mathbf{M}_i\}$ of all nodes. Third, for each edge $\langle i, j \rangle$ in the graph which does not appear in the tree it computes the priors $\mathbf{P}_{ij}^{[i]}$ and $\mathbf{P}_{ij}^{[j]}$ for the nodes i and j. These priors aim to recover part of the information which has been lost when

[1]For a regular cutting matrix $\mathbf{K}$ corresponding to an embedded tree, all off-diagonal entries not corresponding to cut edges must be zero. The block diagonal entries for nodes from which no edge is cut must be zero. The off-diagonal entries corresponding to cut edges must be equal to the original matrix ones.

removing the edge. $\mathbf{P}_{ij}^{[i]}$ and $\mathbf{P}_{ij}^{[j]}$ are computed considering the mutual information introduced by the cut edge based on the estimates $\mathbf{M}_i$ and $\mathbf{M}_j$ computed by the first application of BP. Let $\mathbf{E}_{ij}^{[i]}$ and $\mathbf{E}_{ij}^{[j]}$ be these estimates. They can be computed as follows.

$$\begin{aligned}\mathbf{E}_{ij}^{[i]} &= \mathbf{\Omega}_{ij}^{[ii]} - \mathbf{\Omega}_{ij}^{[ij]}(\mathbf{M}_j + \mathbf{\Omega}_{ij}^{[jj]})^{-1}\mathbf{\Omega}_{ij}^{[ji]} \\ \mathbf{E}_{ij}^{[j]} &= \mathbf{\Omega}_{ij}^{[jj]} - \mathbf{\Omega}_{ij}^{[ji]}(\mathbf{M}_i + \mathbf{\Omega}_{ij}^{[ii]})^{-1}\mathbf{\Omega}_{ij}^{[ij]}. \end{aligned} \quad (16)$$

Intuitively, $\mathbf{E}_{ij}^{[i]}$ is obtained by propagating the edge information from the BP estimate $\mathbf{M}_j$ of the node j along the cut edge. $\mathbf{E}_{ij}^{[j]}$ is computed in a symmetric way.

For each node in a cut edge, we have therefore two estimates: $(\mathbf{E}_{ij}^{[i]}, \mathbf{M}_i)$ and $(\mathbf{E}_{ij}^{[j]}, \mathbf{M}_j)$. We can compute improved estimates by applying covariance intersection, for each off-tree edge, as:

$$\begin{aligned}\hat{\mathbf{M}}_i &= \omega_i \mathbf{M}_i + (1-\omega_i)\mathbf{E}_{ij}^{[i]} \\ \hat{\mathbf{M}}_j &= \omega_j \mathbf{M}_j + (1-\omega_j)\mathbf{E}_{ij}^{[j]}. \end{aligned} \quad (17)$$

In our implementation, we choose ω_i and ω_j so that the determinants of $\hat{\mathbf{M}}_i$ and $\hat{\mathbf{M}}_j$ are minimal. In other words, we select the smaller covariance which can be obtained by CI.

We compute the *priors* $\mathbf{P}_{ij}^{[k]}$ as the difference between the improved estimate and the BP one as

$$\mathbf{P}_{ij}^{[k]} = \hat{\mathbf{M}}_k - \mathbf{M}_k. \quad (18)$$

Here the $\mathbf{P}_{ij}^{[k]}$ represent the desired priors coming from the suppressed edge $\langle i, j \rangle$.

The final step consists in performing a final inference using BP on the spanning tree in which we injected these terms. It is worth noticing, that our algorithm has the best performance on the incremental spanning tree defined by Grisetti *et al.* [5] because the off-edge information is used immediately. Moreover, this tree is as easily maintained as any minimum one.

Note that the prior for a node i is computed by considering the contribution of all the cutted edges connected to node i. Whereas suppressing a single edge leads to a conservative estimate, suppressing multiple edges may lead to overconfident estimates. As shown in Section VI-B, the level of overconfidence increases with the connectivity of the network.

VI. Experiments

In this section we evaluate the performance of our algorithm with respect to standard belief propagation and loopy belief propagation. All algorithms have a complexity linear in the number of edges of the graph. Therefore, we are only interested in measuring the quality of the approximation. Given a node, we want to compare the approximate covariance estimate $\hat{\mathbf{\Sigma}}$ with the exact one $\mathbf{\Sigma}$. This can be done by considering the norm of the matrix difference

$$\|\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_F, \quad (19)$$

where $\|\cdot\|_F$ is the Frobenius norm.

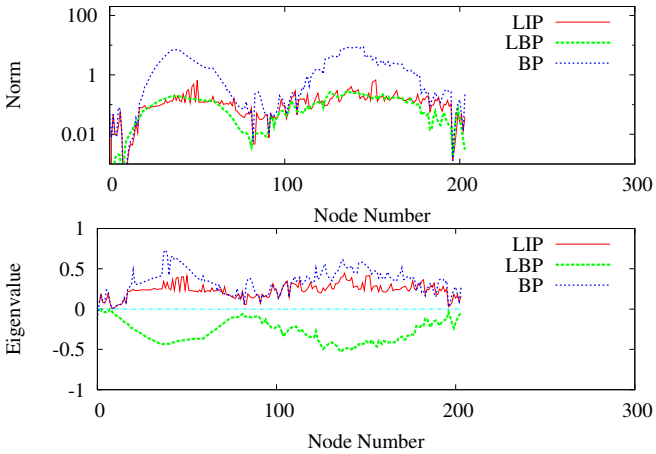

Fig. 7. Analysis of the approximation on the Intel dataset. The upper plot shows the approximation error and the lower one shows the conservativeness analysis of the different nodes. Note that LBP is overconfident, BP on a spanning tree is overly conservative, and our approach (LIP) provides an intermediate result, being in general closer to the exact estimate.

Furthermore, given a conservative and an overconfident estimate, we prefer the conservative one, since it allows to better deal with data association. An estimate is conservative if it is *bigger* than the exact one, thus being $\hat{\mathbf{\Sigma}} - \mathbf{\Sigma} \geq 0$. Measuring the conservativeness means to measure how far the matrix $\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}$ is to be positive definite. This can be done by considering the value of the smallest eigenvalue: it is negative in the case of overconfidence and positive in the case of conservativeness.

We performed experiments on real world datasets and on simulated ones. Furthermore, using simulated data we compared the performances of our algorithm with respect to LPB and BP on a spanning tree on randomly generated networks of different sizes, to collect statistics.

A. Real World Data

We analyzed the behavior of our algorithm on graphs extracted from two standard datasets: the Intel Lab of Seattle (Figure 6), and the ACES building of the University of Austin, Texas (Figure 3). For each node of the network we measured the distance between the estimate and the exact value of the covariance, according to (19). In all the cases our approach provided a better estimate than belief propagation on the spanning tree. Furthermore it provided more conservative estimates than loopy belief propagation. Quantitative results on the Intel dataset are depicted in Figure 7.

B. Statistical Experiments

We performed statistical experiments on simulated networks of different sizes. The networks were randomly generated by simulating a random walk in the $\mathcal{SO}(3)$ space. Loops were simulated by considering the Euclidean distance between nodes. We then compared our approach with LBP and BP on a spanning tree. We used networks with 500, $1,000$, $3,000$, and $5,000$ nodes. The plots in Figure 8 and 9 show the average values and the 95% confidence intervals.

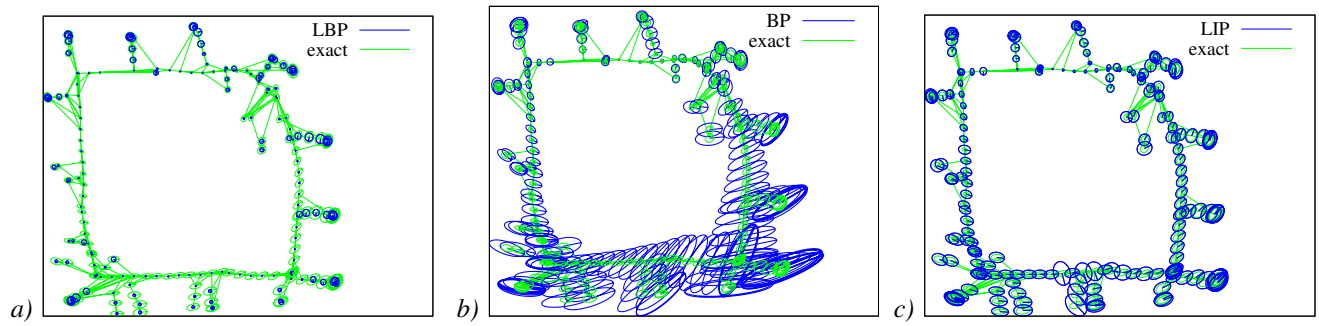

Fig. 6. Marginal covariances comparison on the Intel Lab dataset: the exact covariances, computed by inverting the full matrix, are depicted in red (solid line), the approximated ones are in green (dashed line). Results obtained by *a)* Loopy Belief Propagation. *b)* Belief Propagation on a Minimum Spanning Tree. *c)*Loopy Intersection Propagation.

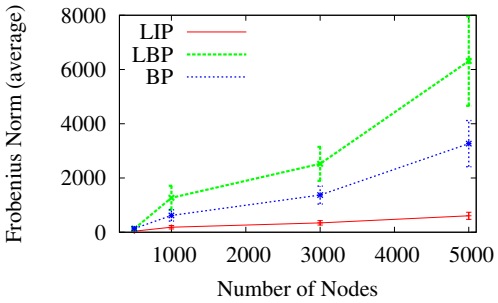

Fig. 8. Average and standard deviation of the Froebius norm of the node estimates of a randomly generated network. This measures the quality of the approximation of the three approaches as a function of the size of the network.

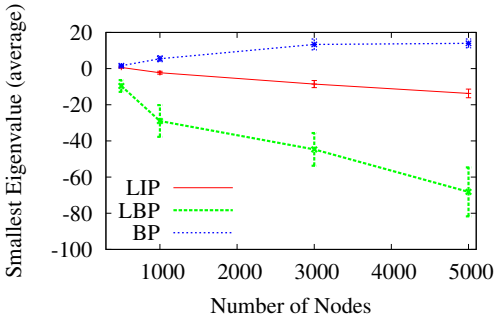

Fig. 9. Average and standard deviation of the minimum eigenvalue of the difference matrix of nodes estimates of a randomly generated network. This measures the conservativeness of the estimate.

Figure 8 shows the approximation error of the three approaches, computed according to (19). As can be seen, our approach scales better than LBP and BP. This is due to the increasing number of loops occurring in the network as its size grows: Whereas BP does not use loop information, our approach does consider it in a better way than LBP.

As for the overconfidence, Figure 9 shows the evolution of the minimum eigenvalue of the error matrix. BP always produces conservative estimates, while LBP produces overly overconfident ones. Our approach lies in the middle with a small level of overconfidence.

VII. Conclusions and Future Work

In this paper we presented a novel algorithm for marginal covariance computation in Graph-SLAM. Our approach has been validated by an extensive set of experiments. In general, the estimated covariances are conservative and when they are overconfident, their overconfidence level is close to 0. Furthermore, our approach provides estimates closer to the exact ones with respect to other techniques of the same family like loopy belief propagation or belief propagation on a spanning tree. In the future, we aim to develop an incremental version and to integrate a data association algorithm for a dynamic construction of the graph.

Acknowledgment

This work has partly been supported by the EC under contract number FP6-IST-34120, Action Line: 2.5.2.: Micro/Nano Based Subsystems.

References

[1] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Journal of Autonomous Robots*, 12(3):287 – 300, 2002.

[2] R. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2428–2435, Barcelona, Spain, 2005.

[3] J. Folkesson and H. Christensen. Graphical slam - a self-correcting map. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, New Orleans, LA, USA, 2004.

[4] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12, 2005.

[5] G. Grisetti, C Stachniss, S. Grzonka, and W. Burgard. Tree parameterization for efficiently computing maximum likelihood maps using stochastic gradient descent. In *Proc. of Robotics: Science and Systems (RSS)*, Atalanta, USA, 2007. To appear.

[6] A. Howard, M.J. Matarić, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1055–1060, 2001.

[7] S. Julier and J. Uhlmann. A nondivergent estimation algorithm in the presence of unknown correlations. In *Proc. of the American Conf. on Control (ACC)*, 1997.

[8] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Journal of Autonomous Robots*, 4:333–349, 1997.

[9] Dmitry Malioutov, Jason Johnson, and Alan Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7, October 2006.

[10] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, Orlando, FL, USA, 2006.

[11] J. Pearl and S. Russel. Handbook of brain theory and neural networks. Technical report, UCLA Cognitive Systems Laboratory,, 2000.

[12] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy SAM. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Hyderbad, India, 2007.

[13] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.