Loopy SAM

Ananth Ranganathan, Michael Kaess, and Frank Dellaert
Center for Robotics and Intelligent Machines, College of Computing
Georgia Institute of Technology
{ananth,kaess,dellaert}@cc.gatech.edu

Abstract

Smoothing approaches to the Simultaneous Localization and Mapping (SLAM) problem in robotics are superior to the more common filtering approaches in being exact, better equipped to deal with non-linearities, and computing the entire robot trajectory. However, while filtering algorithms that perform map updates in constant time exist, no analogous smoothing method is available. We aim to rectify this situation by presenting a smoothing-based solution to SLAM using Loopy Belief Propagation (LBP) that can perform the trajectory and map updates in constant time except when a loop is closed in the environment. The SLAM problem is represented as a Gaussian Markov Random Field (GMRF) over which LBP is performed. We prove that LBP, in this case, is equivalent to Gauss-Seidel relaxation of a linear system. The inability to compute marginal covariances efficiently in a smoothing algorithm has previously been a stumbling block to their widespread use. LBP enables the efficient recovery of the marginal covariances, albeit approximately, of landmarks and poses. While the final covariances are overconfident, the ones obtained from a spanning tree of the GMRF are conservative, making them useful for data association. Experiments in simulation and using real data are presented.

1 Introduction

The problem of Simultaneous Localization and Mapping (SLAM) is a core competency of robotics which has attracted a large amount of attention. The classical solution to the SLAM problem uses an Extended Kalman Filter (EKF) [Smith and Cheeseman, 1987] that maintains the joint distribution on the current robot pose and the landmarks in the map in the form of a Gaussian distribution. Since updating the covariance matrix of this distribution requires $O(n^2)$ time, the EKF algorithm does not scale to large maps. Recent work on the SLAM problem has focussed on improving the time complexity of the solutions. One commonly used technique is to update the information form of the Gaussian distribution. Though the information matrix becomes dense through repeated marginalization of poses, recently introduced techniques such as the Sparse Extended Information Filter (SEIF) [Thrun *et al.*, 2004] and the Thin Junction Tree Filter (TJTF) [Paskin, 2003] present approximations to keep the information matrix sparse. The sparsity of the representation allows for a constant time update of the filter.

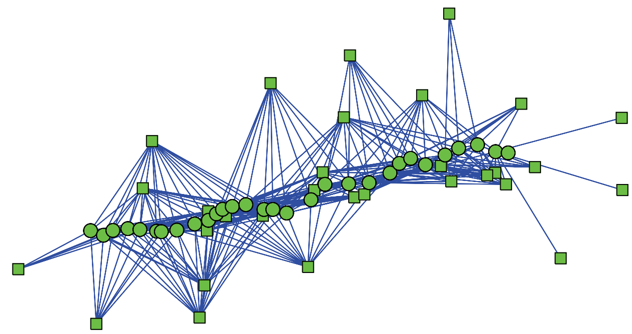

Figure 1: The Gaussian Markov Random Field (GMRF) corresponding to an illustrative Smoothing and Mapping (SAM) problem with 30 poses in the robot trajectory. Poses in the graph are shown as circles and landmarks as squares.

In contrast to the above filtering approaches which only compute the current robot pose with the map, a smoothing approach recovers the *complete* robot trajectory and the map. The smoothing information matrix naturally stays sparse over time without the need for any approximations. Further, smoothing can incorporate new information about past robot poses as opposed to filtering approaches that cannot update a pose after it has been marginalized out. Even though the number of variables increases continuously for smoothing, in many real-world scenarios, especially those involving exploration, this larger matrix still contains far less entries than in filter-based methods [Dellaert, 2005] [Eustice *et al.*, 2005]. Filters are more efficient then smoothing only when a limited number of structure points is observed repeatedly. However, even in this case the inability to correct previous poses may lead to poor results.

The crucial limitation of all the above approaches based on the information form of the Gaussian is that recovering an estimate of the map and robot pose involves a matrix inversion, and hence, is computationally infeasible for large systems. Indeed, in the case of the filtering approaches such as the SEIF, obtaining even the mean involves a matrix inversion.

The solution is obtained in these cases through an iterative process that solves a linear system wherein only a constant number of iterations is performed at each step so as to maintain the runtime bounds of the algorithm. This approximation is based on the assumption that the SLAM solution evolves only slowly over time so that a relinearization of the problem at each time step is not required. The iterative method, however, still does not yield the covariance, which is needed for maximum-likelihood data association and also as an estimate of the uncertainty in the landmark locations and pose.

This paper deals with the use of Loopy Belief Propagation (LBP) [Weiss and Freeman, 1999] to obtain an approximate solution to the SLAM problem, including the marginal covariances on the map and pose. We consider the case wherein the map consists of a set of features, and build on the Smoothing and Mapping (SAM) approach as introduced in [Dellaert, 2005], which updates the joint distribution on the robot trajectory and the map using the square root information form of the Gaussian distribution. Our approach is based on the fact that the SAM problem can be treated from the graphical model viewpoint as a Gaussian Markov Random Field (GMRF) [Winkler, 1995] on which inference can be performed using LBP.

We provide a constant time update algorithm, which we call the *Wildfire algorithm*, for estimating the map and robot trajectory, including the marginal covariances. While a naive implementation of LBP requires $O(n)$ time to update the marginal distributions, where n is the number of nodes in the graph, the Wildfire algorithm discards negligibly small messages between nodes. Subsequently, only those nodes which have either received or sent significant messages get updated. We show that the number of nodes involved in a significant message exchange is $O(1)$ unless a significant event (such as loop closing) occurs when the complete graph may get updated, increasing the time bound to $O(n)$.

Relinearization of the GMRF graph, which is required since the SAM problem is non-linear in general, is performed in constant time using a technique similar to the Wildfire algorithm in that only nodes that have changed are relinearized. Moreover, it need only be done periodically when the difference between the current estimate and the linearization point exceeds a specified threshold. As before, relinearization may take linear time after a significant update of the GMRF graph.

We prove that LBP on a GMRF is equivalent to Gauss-Seidel relaxation and provides the exact MAP estimates of the nodes. The covariances are, however, overconfident due to the approximate nature of the inference. As a conservative approximation, the covariances obtained by restricting inference to a spanning tree of the GMRF graph can be used to perform data association.

In terms of related work, closest are the relaxation based approaches given in [Duckett *et al.*, 2000; Frese and Duckett, 2003]. However, these are unable to compute the marginal covariances on the poses and landmarks. Graphical SLAM [Folkesson and Christensen, 2004] is another method that builds a graphical model of the smoothing problem. Our approach shares many common features with Graphical SLAM, including the ability to modify data association on the fly, and increase efficiency by eliminating variables judiciously. In addition, our algorithm is incremental and largely runs in $O(1)$ time, which are advantages over Graphical SLAM.

2 Smoothing and Mapping

We begin by reviewing the SAM framework as given in [Dellaert, 2005]. In the smoothing framework, our aim is to recover the maximum a posteriori (MAP) estimate for the entire trajectory $X \triangleq \{x_i : i \in 0 \dots M\}$ and landmark locations (the map) $L \triangleq \{l_j : j \in 1 \dots N\}$, given the landmark measurements $Z \triangleq \{z_k : k \in 1 \dots K\}$ and odometry $U \triangleq \{u_i\}$. This can be computed by maximizing the joint posterior on the trajectory and landmark locations $P(X, L|Z)$

$$\Theta^* \triangleq \operatorname*{argmax}_{\Theta} \; P(X, L|Z, U) \tag{1}$$

where $\Theta \triangleq \{X, L\}$ is the set of unknown variables, i.e the robot trajectory and landmark locations. The posterior can be factorized using Bayes law as

$$\begin{aligned} P(X, L|Z, U) &\propto P(Z, U|X, L)P(X, L) \\ &\propto P(x_0)\prod_{i=1}^{M} P(x_i|x_{i-1}, u_i) \prod_{k=1}^{K} P(z_k|x_{i_k}, l_{j_k}) \end{aligned} \tag{2}$$

Here $P(x_i|x_{i-1}, u_i)$ is the motion model, parameterized by the odometry u_i, and $P(z_k|x_{i_k}, l_{j_k})$ is the landmark measurement model, assuming that the correspondences (i_k, j_k) are known. The prior on the initial pose $P(x_0)$ is taken to be a constant, usually the origin, while the priors on the remaining poses and landmark locations are assumed to be uniform.

We assume Gaussian motion and measurement models, as is standard in the SLAM literature. The motion model is given as $x_i = f_i(x_{i-1}, u_i) + w_i$, where w_i is zero-mean Gaussian noise with covariance matrix Q_i. Similarly, the measurement equation is given as $z_k = h_k(x_{i_k}, l_{j_k}) + v_k$ where v_k is normally distributed zero-mean measurement noise with covariance R_k.

In practice, only linearized versions of the motion and measurement models are considered, for example for use in the Extended Kalman Filter (EKF) approach to SLAM [Smith *et al.*, 1990], with multiple iterations being used to obtain convergence to the minimum. In the following, we will assume that either a good linearization point is available or that we are working on one iteration of a non-linear optimization method. It can be shown [Dellaert, 2005] that this results in a sparse linear least squares problem in $\delta\Theta$

$$\delta\Theta^* = \operatorname*{argmin}_{\delta\Theta} \left\{ \sum_{i=1}^{M} \|F_i^{i-1}\delta x_{i-1} + G_i^i \delta x_i - a_i\|_{Q_i}^2 + \sum_{k=1}^{K} \|H_k^{i_k}\delta x_{i_k} + J_k^{j_k}\delta l_{j_k} - c_k\|_{R_k}^2 \right\} \tag{3}$$

where F_i^{i-1} is the sparse Jacobian of $f_i(.)$ at the linearization point x_{i-1}^0 and $a_i \triangleq x_i^0 - f_i(x_{i-1}^0, u_i)$ is the odometry prediction error. Analogously, $H_k^{i_k}$ and $J_k^{j_k}$ are respectively the sparse Jacobians of $h_k(.)$ with respect to a change in x_{i_k} and l_{j_k}, evaluated at the linearization point $(x^0_{i_k}, l^0_{j_k})$, and $c_k \triangleq z_k - h_k(x^0_{i_k}, l^0_{j_k})$ is the measurement prediction error. $||.||_\Sigma$ is the Mahalanobis distance with respect to the covariance matrix Σ and we use the matrix $G^i_i = -I_{d\times d}$, d being the dimension of x_i, to avoid treating δx_i specially.

The solution to the least squares problem (3) is given by the linear system $A\delta\Theta - b = 0$ where $A = J^T J$ is the Hessian matrix, J being the system Jacobian matrix obtained by assembling the individual measurement Jacobians, and $b = J^T e$ where e is the vector of all the measurement errors.

SAM as a Markov Random Field

We now show that inference on the posterior (2) can be performed by placing the SAM problem in a Markov Random Field (MRF) framework. The graph of an MRF is undirected and its adjacency structure indicates which variables are linked by a common factor (a measurement or constraint).

A pairwise MRF is described by a factored probability density given as

$$P(X, L) \propto \prod_i \phi(y_i) \prod_{\{i,j\}} \psi(y_i, y_j) \tag{4}$$

where the second product is over the pairwise cliques $\{i, j\}$, counted once. The posterior (2) can be modeled as a pairwise MRF where the singleton cliques correspond to marginal prior distributions or singleton measurements (such as GPS) on the unknown variables and the edge cliques correspond to the motion and measurement likelihoods respectively.

For the linear case considered in the previous section, these equations translate to

$$\begin{aligned} -\ln\phi(y_i) &\propto \frac{1}{2}\|x_i - \mu_i\|^2_{P_i} \\ -\ln\psi(x_{i-1}, x_i) &\propto \frac{1}{2}\|F^{i-1}_i \delta x_{i-1} + G^i_i \delta x_i - a_i\|^2_{Q_i} \\ -\ln\psi(x_{i_k}, l_{j_k}) &\propto \frac{1}{2}\|H^{i_k}_k \delta x_{i_k} + J^{j_k}_k \delta l_{j_k} - c_k\|^2_{R_k} \end{aligned} \tag{5}$$

where $y_i \in \{X, L\}$. This gives us a Gaussian Markov Random Field (GMRF) corresponding to the linearized SAM problem. Note that in the context of the SAM problem, the singleton clique factors are most often set to unity except for the first pose, which is clamped to the origin.

3 Loopy Belief Propagation for SAM

We use Loopy Belief propagation (LBP) on the SAM GMRF to solve the linear system (3) and obtain not only the MAP values of the unknown variables but also the covariances. For ease of computation, we use the information form of the Gaussian to specify the single and pairwise clique potentials in the GMRF given by (5), wherein these are expressed using the *information vector* η and the *information matrix* Λ, and we define the information form as

$$\mathcal{N}^{-1}(x; \eta, \Lambda) \triangleq \exp\left\{C + \eta^T x - \frac{1}{2}x^T \Lambda x\right\} \tag{6}$$

where C is an additive constant. If Λ is full-rank, equation (6) corresponds to a Gaussian density as

$$\mathcal{N}(x; \Lambda^{-1}\eta, \Lambda^{-1}) = \mathcal{N}(x; \mu, P) \tag{7}$$

where K is a constant of proportionality.

Using this formulation, we can write the edge potentials from (5) as $\psi(x_i, l_j) \propto \mathcal{N}^{-1}(y_{ij}; \eta_{ij}, \Lambda_{ij})$, where y_{ij} is the vector $\begin{bmatrix} x_i & l_j \end{bmatrix}^T$, and we have dropped the index k from the corresponding pair $\{i_k, j_k\}$ for convenience. The distribution parameters are then given as

$$\eta_{ij} \triangleq \begin{bmatrix} \eta^i_{ij} \\ \eta^j_{ij} \end{bmatrix} = \begin{bmatrix} H^i_k \\ J^j_k \end{bmatrix} R^{-1}_k c_k \tag{8}$$

$$\Lambda_{ij} \triangleq \begin{bmatrix} \Lambda^{ii}_{ij} & \Lambda^{ij}_{ij} \\ \Lambda^{ji}_{ij} & \Lambda^{jj}_{ij} \end{bmatrix} = \begin{bmatrix} H^i_k \\ J^j_k \end{bmatrix} R^{-1}_k \begin{bmatrix} H^i_k \\ J^j_k \end{bmatrix}^T \tag{9}$$

and the potentials on the edges linking two poses are defined analogously.

The GMRF potentials on the singleton cliques are similarly defined as $\phi(y_i) = \mathcal{N}^{-1}(y_i; \eta_i, \Lambda_i)$ where $y_i \in \{X, L\}$, and $\eta_i = \Lambda_i \mu_i$, $\Lambda_i = P^{-1}_i$ follows from (5) and (7). The potentials $\phi(y_i)$ represent the beliefs on the unknowns and if only a uniform prior exists on an MRF node y_i, both η_i and Λ_i can be set to zero. Note that setting Λ_i to zero is equivalent to having a Gaussian prior with infinite covariance.

Belief Propagation

The goal of belief propagation in our case is to find the marginal probability, or the *belief*, $P(y_i)$ of a node $y_i \in \Theta = \{X, L\}$ in the SAM graph. A complete characterization of the belief propagation algorithm for GMRFs is beyond the scope of this paper, but we provide the high-level equations and data flow below. Details can be found in [Weiss and Freeman, 1999].

The beliefs are computed using a message passing algorithm wherein each node passes messages to its neighbors and in turn uses its incoming messages to compute its belief. While belief propagation is guaranteed to compute the correct beliefs only if the graphical model does not contain loops, it has been proven that it finds the correct MAP solution for Gaussian graphical models even in the presence of loops, i.e the means of the beliefs are correct while the covariances are incorrect in general [Weiss and Freeman, 1999].

LBP works by repeatedly computing the messages and beliefs for every node in the graph starting with constant messages [Weiss and Freeman, 1999], either in synchronous or asynchronous fashion. This sweep over the graph is iterated until the beliefs converge. Denoting the current iteration by a discrete time superscript t, the belief parameters (the information vector and matrix) for node y_i are given as

$$m^t_i = \eta_i + \sum_{j\in\mathcal{N}_i} m^{t-1}_{ji} \tag{10}$$

$$M^t_i = \Lambda_i + \sum_{j\in\mathcal{N}_i} M^{t-1}_{ji} \tag{11}$$

and the parameters of the message $\mathcal{M}_{ij}(y_j)$ from node y_i to node y_j as

$$m^t_{ij} = \eta^j_{ij} - \Lambda^{ji}_{ij}\left(\Lambda^{ii}_{ij} + M^t_i - M^{t-1}_{ji}\right)^{-1} \left(\eta^i_{ij} + m^t_i - m^{t-1}_{ji}\right) \tag{12}$$

$$M^t_{ij} = \Lambda^{jj}_{ij} - \Lambda^{ji}_{ij}\left(\Lambda^{ii}_{ij} + M^t_i - M^{t-1}_{ji}\right)^{-1}\Lambda^{ij}_{ij} \tag{13}$$

where use has been made of the definitions in (8) and (9).

Algorithm 1 The Wildfire algorithm for LBP

1. Push the most recently added nodes and their neighbors in the SAM graph $G(V, E)$ onto the queue Q
2. Do until Q is empty
   (a) Pop node y from Q and compute messages $\mathcal{M}_{yk}^{t}$ for all $k \in \mathcal{N}_y$ using (12-13)
   (b) If $\mathcal{M}_{yk}^{t} - \mathcal{M}_{yk}^{t-1} > \epsilon$, push k onto Q
   (c) Update the belief $\mathcal{B}(y)$ using (10-11)

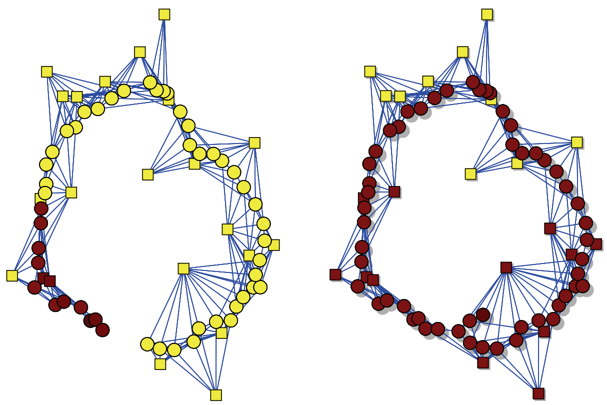

Figure 2: The Wildfire algorithm only updates nodes with significant message deviations, starting from the most recently added node. In normal operation (left) only the frontier of the graph is updated. However, during loop closings (right) most of the graph is updated. Updated poses are shown in brown (dark) while the unchanged nodes are in yellow (light). The robot is moving counter-clockwise starting at the bottom.

4 Constant Time Loopy SAM

Each iteration of LBP as defined above is $O(n)$ where n is the number of nodes in the graph. This is too slow to be useful, especially since the trajectory is included in the node count. We can, however, improve this bound to $O(1)$ for incremental operation, i.e. when new nodes or edges are added to the GMRF.

The key observation to getting a constant time algorithm is that at each step only a few of the nodes get updated, i.e. experience an actual change in their values, and these nodes are the ones that have been recently added to the graph. If we commence the belief updation sweep in every iteration of LBP from the most recently added node, the new messages are initially seen to deviate from their previous values by large amounts. However, these deviations become smaller as we move further back in the graph. Once the message deviations become negligible (smaller than some pre-specified significance threshold), the beliefs no longer change and the remaining nodes need not be updated.

The Wildfire algorithm, so called because the message deviations from the previous iteration spread like wildfire from their point of origin and gradually become negligible, is illustrated in Figure 2. The algorithm implementation features a queue onto which nodes with significant message deviations on incoming edges are pushed. A summary of the algorithm is given in Algorithm 1.

The Wildfire algorithm has $O(1)$ running time since in most situations the number of edges with large message deviations is $O(1)$. However, this is not true when special events such as loop closures occur as shown in Figure 2. In such instances, the complete graph is updated resulting in $O(n)$ complexity. Note that no special clause is required to handle these instances - the algorithm automatically updates all the nodes in the graph as the message deviations do not die down.

Relinearization of the SAM graph, which involves computing the clique potentials at new linearization points, can also be performed using the Wildfire algorithm, the only difference being that the deviations of the node beliefs from their respective linearization points are now used instead of the message deviations. In this case, however, two variants are possible. One technique is to delay relinearization until the belief deviations are greater than a certain threshold for some given proportion of nodes in the graph. As this proportion approaches unity, relinearization becomes $O(n)$ but is performed only very rarely. A second strategy is to perform relinearization at regular time intervals so that it remains $O(1)$. The choice between these strategies depends on the type of application, the environment, and size of the SAM graph.

We now have all the components to describe the Loopy SAM algorithm. At each step, a new pose node, new measurement links and possibly new landmark nodes, are added to the SAM GMRF. A fixed number of LBP iterations are then performed using the Wildfire algorithm to update the beliefs on the nodes. Periodically, relinearization is performed, again using a variant of the Wildfire algorithm. The means of the pose and landmark nodes in the SAM GMRF, obtained from the belief information vector and matrix at each node, correspond to the MAP solutions for the trajectory and the map respectively.

4.1 Marginal Covariances

We now show how to recover the approximate marginal covariances of the nodes from the LBP beliefs. An important use of the covariances is to bound the search area for possible correspondences in data association. This is usually done using the projection of the combined pose and landmark uncertainty into the measurement space, the computation of which requires knowledge of the marginal covariances of the poses and landmarks.

The inverse of the belief information matrix at each node in the SAM GMRF gives the marginal covariance. While in general the belief obtained from LBP can be overconfident or underconfident, it has been proven to be always overconfident for Gaussian MRFs with pairwise cliques [Weiss and Freeman, 1999]. Overconfident covariances cannot be used for data association since this results in many valid correspondences being rejected. A first conservative approximation would be to use the inverse of the diagonal blocks of the system Hessian matrix as the marginal covariances. However, this results in overly conservative estimates.

A better approximation can be obtained by restricting message passing to a spanning tree of the SAM GMRF. Since the spanning tree is obtained by deleting edges from the GMRF and inference on it is exact, we get a conservative estimate of the true marginal covariances that is also better than our first

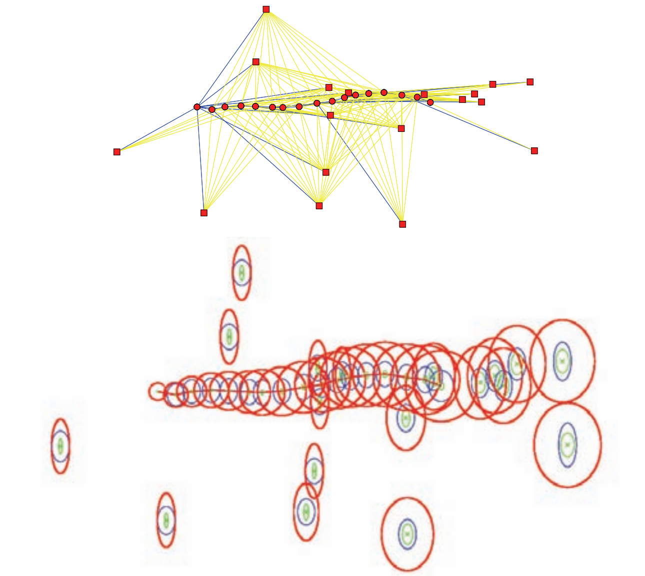

Figure 3: (top) A spanning tree of the GMRF graph, with edges colored blue (dark), for a linear simulation. (bottom) Covariances for the same simulation - ground truth is shown in blue (thin dark), overconfident LBP covariances are shown in green (light), and conservative covariance estimates obtained by restricting BP to the spanning tree are shown in red (thick dark).

order approximation. Maintenance of the spanning tree and message passing on it require extra work. However, the spanning tree can be extended at each step in $O(1)$ so that, with the use of the Wildfire algorithm, the overall scheme still remains constant time. Figure 3 illustrates the nature of this approximation.

5 Loopy SAM and Gauss-Seidel Relaxation

We prove in this section that computing the MAP solution using LBP for a GMRF is equivalent to a modified form of Gauss-Seidel relaxation, which is commonly used to solve linear systems. Relaxation has previously been used in the context of SLAM [Duckett *et al.*, 2000] but is not capable of recovering the marginal covariances, a short-coming that LBP overcomes. The equivalence of the MAP portion of LBP and relaxation is an interesting result that connects our technique with a well-understood method from linear algebra.

Relaxation solves the linear system $A\delta\Theta - b = 0$, encountered in Section 2, by iterating over a set of equations, each of which updates the value of exactly one unknown using the current estimate of all the others

$$A_{ii}y_i = b_i - \sum_{j\neq i} A_{ij}y_j \tag{14}$$

where the sum is over all the variables except y_i, and $y_i \in \Theta = \{X, L\}$ as before.

We begin our proof by noting that the message from y_j to y_i given by (12) can be re-written as

$$m^t_{ji} = J^T_{ki}e_k - A_{ij}y^{-i}_j \tag{15}$$

where y^{-i}_j is the estimate of y_j without considering the node y_i, k is the index of the measurement linking y_i and y_j in the

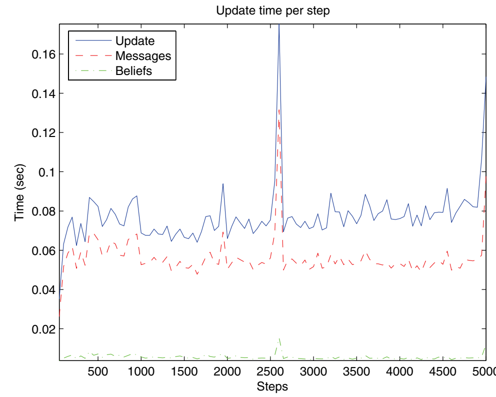

Figure 4: Runtime for a simulation consisting of 5000 steps. The times for the message and belief computation, and the total update step are shown. All times were obtained on a 1.8GHz Linux machine using an implementation of the algorithm in Ocaml. Note the two spikes (one at the very end) corresponding to loop closures.

system Jacobian J, and e is the corresponding measurement error. This result is true since we have

$$m^t_j - m^{t-1}_{ij} = \eta_j + \sum_{k\in\mathcal{N}_j\setminus i} m^{t-1}_{kj} = m^{-i}_j$$

where m^{-i}_j is the estimate of m_j without considering node y_j, and similarly $M^t_j - M^{t-1}_{ij} = M^{-i}_j$. Since $M^{-i}_j y^{-i}_j = m^{-i}_j$ holds by the definition of the information form of the Gaussian, substituting these identities and the definitions of η and Λ from (8-9) into the message equation (12) gives us the result (15).

Using (15) in the belief equation (10), we get

$$\begin{aligned} m^t_i &= \eta_i + \sum_{j\in\mathcal{N}_i} J^T_{k_j i}e_{k_j} - \sum_{j\in\mathcal{N}_i} A_{ij}y^{-i}_j \\ &= \eta_i + b_i - \sum_{j\in\mathcal{N}_i} A_{ij}y^{-i}_j \end{aligned} \tag{16}$$

which closely corresponds to the relaxation equation (14). Hence, the MAP computation in LBP is simply a modified version of Gauss-Seidel relaxation.

6 Results

Linear Gaussian Simulations

We first present the results of applying our approach in a controlled simulation devoid of non-linearities. The simulation consisted of an environment with randomly placed, uniformly distributed landmarks. The robot followed a large 8-shaped figure in this environment and the run included two loop closures at the "waist" of the 8-figure.

Figure 4 gives the results for a linear simulation consisting of 5000 steps, i.e. a trajectory of length 5000. The total number of nodes in the SAM GMRF was 6742. It can be verified that the algorithm is $O(1)$, and the two spikes in the graph corresponding to the loop closures are also clearly visible. The spikes occur since the LBP update makes a sweep through the whole graph in these instances.

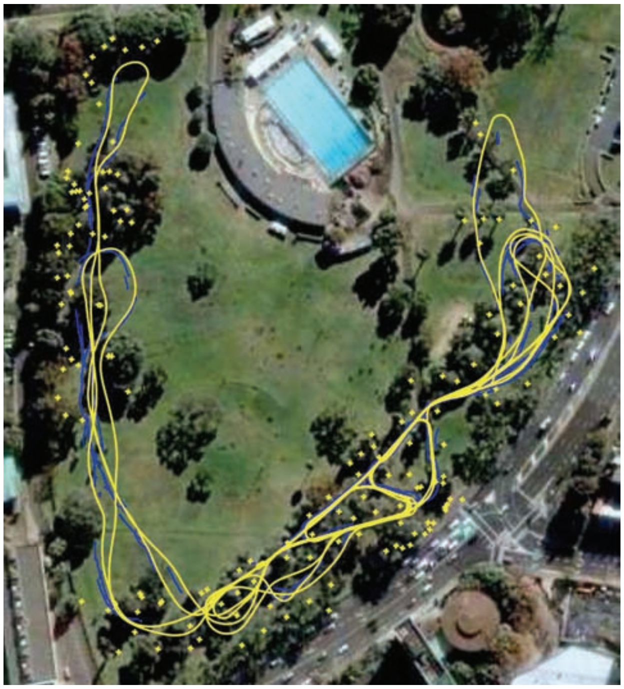

Figure 5: Trajectory and map computed from the Victoria Park dataset overlaid on satellite imagery of the park.

6.1 Victoria Park Dataset

We applied our algorithm to the Sydney Victoria Park dataset (available at `http://www.acfr.usyd.edu.au/homepages/academic/enebot`), a popular test dataset in the SLAM community. The trajectory consists of 6968 poses along an approximately 4 kilometer long trajectory. Landmarks were obtained by running a simple tree-detection algorithm on the laser range data, yielding 158 tree-like features and a total of 3631 measurements.

Figure 5 shows the resulting computed trajectory and map, where the landmarks are shown using crosses, overlaid on an image of the park. The GPS trajectory, which was not used in the estimation, is also shown for reference. This demonstrates the applicability of our technique to real-world data.

7 Conclusion

We presented an algorithm to perform smoothing-based SLAM using Loopy Belief Propagation (LBP). We provide a flavor of LBP that makes map and trajectory updates constant time except when closing loops in the environment. In this case, LBP is equivalent to a modified version of Gauss-Seidel relaxation, a result that we proved here.

While our technique has advantages over existing state-of-the-art in that we can recover the marginal covariances in $O(1)$ time, these are a heuristic approximation to the true covariances. It is our experience that the LBP covariances are of good quality when the robot is exploring new areas in the environment but become highly overconfident when it revisits previously visited areas. In general, the more loopy the GMRF, the more overconfident the covariances from LBP. This observation can be intuitively explained by appealing to the fact that the covariances become incorrect due to multiple use of the same evidence when it is propagated around a loop in the graph.

It is future work to define rigorous bounds on the covariance obtained from LBP, and also investigate avenues to recover the exact covariance using related techniques, an example being [Sudderth *et al.*, 2004].

Acknowledgements

The authors were partly supported during this work by the National Science Foundation award IIS - 0448111 "CAREER: Markov Chain Monte Carlo Methods for Large Scale Correspondence Problems in Computer Vision and Robotics"

References

[Dellaert, 2005] F. Dellaert. Square Root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, 2005.

[Duckett *et al.*, 2000] T. Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, 2000.

[Eustice *et al.*, 2005] R. Eustice, H. Singh, and J. Leonard. Exactly Sparse Delayed-State Filter. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005.

[Folkesson and Christensen, 2004] J. Folkesson and H. I. Christensen. Graphical SLAM - a self-correcting map. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 383 – 390, 2004.

[Frese and Duckett, 2003] U. Frese and T. Duckett. A multigrid approach for accelerating relaxation-based SLAM. In *Proc. of the IJCAI-03 on Reasoning with Uncertainty in Robotics*, 2003.

[Paskin, 2003] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.

[Smith and Cheeseman, 1987] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Intl. J. of Robotics Research*, 5(4):56–68, 1987.

[Smith *et al.*, 1990] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in Robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[Sudderth *et al.*, 2004] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Embedded trees: Estimation of Gaussian Processes on graphs with cycles. *IEEE Transactions on Signal Processing*, 52(11):3136–3150, 2004.

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

[Weiss and Freeman, 1999] Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *Advances in Neural Information Processing Systems (NIPS)*, pages 673–679, 1999.

[Winkler, 1995] G. Winkler. *Image analysis, random fields and dynamic Monte Carlo methods*. Springer Verlag, 1995.