A Markov Chain Monte Carlo Approach to Closing the Loop in SLAM

Michael Kaess* and Frank Dellaert
College of Computing
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332, USA
{kaess,dellaert}@cc.gatech.edu

***Abstract*— The problem of simultaneous localization and mapping has received much attention over the last years. Especially large scale environments, where the robot trajectory loops back on itself, are a challenge. In this paper we introduce a new solution to this problem of closing the loop. Our algorithm is EM-based, but differs from previous work. The key is a probability distribution over partitions of feature tracks that is determined in the E-step, based on the current estimate of the motion. This virtual structure is then used in the M-step to obtain a better estimate for the motion. We demonstrate the success of our algorithm in experiments on real laser data.**

***Index Terms*— loop closing, SLAM, localization, mapping**

I. Introduction

In this paper we present a novel method for "closing the loop" when building large scale environment models from long sequences of sensor data. When a sequence is taken such that the robot trajectory turns back on itself, a difficult problem arises of identifying which measurements correspond to previously seen features of the environment. An important insight is that this is essentially a model selection problem: we are interested in estimating structure and motion parameters from measurement data, but it is unclear how many structure elements there are.

The problem is important because accumulated measurement error poses an inherent limitation on the accuracy by which environment models can be recovered. Identifying which measurements have already been seen is essential in order to obtain accurate, globally correct models. Any incremental method that tracks features and instantiates new tracks in the standard way will end up representing the same environment features multiple times.

Many different algorithmic approaches to loop closing have been proposed in the literature. It is common to use some method for detecting loop closings, and then spreading the error along the already created path. One such method for loop detection is described in [9], where sets of new measurements are globally correlated with the map, which itself is generated incrementally by locally registering new data. This approach is therefore called "Local Registration and Global Correlation". Other authors assume that the point of loop closing is already known [18], and focus on how to correct the map. In both cases, an extended Kalman filter is used to correct the map backwards in time.

A different approach is taken in [20] and [1], which both use the the Expectation-Maximization (EM) algorithm [14] to iteratively generate a better map. Thrun et al. [20] work with a hybrid map representation, where the same algorithm is first applied to the topological map to correct large scale errors in the odometry, followed by finer adjustments of the metric map. The E-step uses the current estimate of the map to obtain a probability distribution over the robot trajectory. A Kalman smoother is used for revising past beliefs. In the M-step the most likely map is then calculated based on the result from the E-step. In contrast, the approach in [2] employs a metric representation and remembers a history of hypotheses over robot trajectories that were generated by a hybrid localization method in the E-step, and selects the most likely one in the E-step before applying the Kalman smoother.

An alternative approach to correcting the map is taken in [10], which uses the Rao-Blackwellized particle filter (RBPF) [17] in combination with scan matching. The RBPF was first successfully applied to robotics localization in [16] where it is known as FastSLAM, and in which each particle in a filter represents a complete robot trajectory hypothesis. In theory the RBPF can represent the probability distribution over all possible trajectories and is therefore capable of closing the loop. However, since the dimensionality of the trajectory grows over time, the number of particles would have to be increased exponentially to avoid eliminating the potential for loop closing. Haehnel et al. [10] therefore combine multiple scans by scan matching to correct the odometry, and then uses only a sparse set of pre-corrected poses in FastSLAM to extend the spatial range over which multiple hypotheses can be represented.

The method we propose also uses the EM algorithm to find the most likely robot trajectory and close the loops. However, in contrast to other work like [20] the E-step is used to create a probability distribution over partitions

*This work was partially carried out when the author was at Wolfram Burgard's Autonomous Intelligent Systems lab in Freiburg, Germany of feature tracks, while the M-step maximizes the robot trajectory based on the virtual structure obtained in the E-step. This is related to our previous work in [4], [5], [6], in which the correspondence problem in structure from motion is addressed. As a side effect, at convergence, we also obtain a distribution over all possible *environment models*, associated with the maximum a posteriori (MAP) trajectory. This distribution ranges over models of different complexity, and is useful in its own right to perform Bayesian inference about the structure. If needed the distribution can also be used to output a single most likely structure model, e.g. to visualize the environment.

II. The Loop Closing Problem

We are interested in reconstructing a mobile robot's trajectory through an unknown environment, based on measurements from onboard range sensors. A model of the environment is typically also of interest, or can even be the main goal of an application. Indeed, motion and model are dual in some sense: if we know one, it is generally much easier to determine the other. If the motion is known, a model can be created easily by triangulation, and if the model is known, it can be used to localize the robot.

A. Problem Statement

We want to determine the robot's trajectory together with the corresponding 2D structure of the environment. The robot's trajectory, or its *motion*, $M = \{m_i\}_{i=1}^n$ specifies its pose m_i for each step i. Since for 2D mapping the mobile robot operates on a planar surface, the pose has three degrees of freedom that are described by a 2D translation (x, y) and a rotation θ. The environment's *structure* $X = \{x_j\}_{j=1}^N$ is a set of N 2D structure points $x_j \in \mathbb{R}^2$.

The input data is a set $U = \{u_k\}_{k=1}^K$ of K *feature tracks* u_k, where each track u_k corresponds to one or more observations of a single 2D structure point x_j. Note that several tracks can refer to the same structure point and that the overall number N of structure points is unknown. Assignments of tracks to structure points can be formalized by a *partition* $J = \{j_k\}_{k=1}^K$ of the feature tracks U into non-empty subsets, that assigns each track u_k to a structure point $x_{j_k}, 1 \leq j_k \leq N$. This partition is usually unknown, and is the key to solving the loop closing problem.

B. Known partition

In the case that the partition J is known, it is fairly straightforward to determine the structure X and motion M. The maximum a-posteriori estimate of structure and motion based on the known tracks and partition is defined by

$$X^*, M^* = \underset{X,M}{\operatorname{argmax}} \ \log L(X, M; U, J) + \log P(M) \quad (1)$$

containing the likelihood $L(X, M; U, J)$ and a prior $P(M)$ on the motion, which can be based on odometry if available. The likelihood $L(X, M; U, J)$ is proportional to the conditional density $P(U, J|X, M)$ of the tracks U and partition J given structure X and motion M and can be factored based on the independence of the measurements under known motion. Defining $N(J)$ as the number of structure points predicted by the partition J, the set of tracks U under this partition can be written as a union

$$U = \bigcup_{j=1}^{N(J)} U_j \quad (2)$$

of sets U_j of tracks, where all tracks in such a subset U_j correspond to the same structure point x. The independence of the measurements allows rewriting the log-likelihood $L(X, M; U, J)$ from (1) as a sum of terms for each subset U_j:

$$\log L(X, M; U, J) \propto \sum_{U_j \in U} \log P(U_j|x_j, M)$$

Again based on the independence argument, each likelihood term over a subset U_j can be split into terms corresponding to its components $u \in U_j$. The log-likelihood of a track u itself is given by the sum of log-likelihoods of its individual observations $o_{ik} \in u$, where $o \in \mathbb{R}^2$ describes the observation in terms of a range measurement and i indicates in which step the observation took place:

$$\log P(U_j|x_j, M) = \sum_{u \in U_j} \sum_{o_{ik} \in u} \log P(o_{ik}|x_j, m_i) \quad (3)$$

We use a generative model to define the log-likelihood of an individual observation. The generative model consists of an observation function h that geometrically predicts the observation o_{ik} based on the robot's pose m_i in step i, the j_kth structure point location x_{j_k} and some additive noise n:

$$o_{ik} = h(m_i, x_{j_k}) + n \quad (4)$$

For a range sensor, the observation function h is a 2D rigid transformation:

$$h(m_i, x_j) = R_i (x_j - t_i)$$

where the rotation R_i and translation t_i are the components of the i-th robot pose $m_i = (R_i, t_i)$.

If we assume the noise n in the generative model (4) to be i.i.d. zero-mean Gaussian noise with standard deviation σ, then the log-likelihood of an observation o of the structure point x taken at pose m can be written as the negative, squared reprojection error:

$$\log P(o|x, m) = -\frac{1}{2\sigma^2} \|o - h(m, x)\|^2$$

C. Unknown partition

We derive how to solve the loop closing problem when the partitions are unknown, which is typically the case. We are interested in the posterior $P(X, M|U)$ over structure X and motion M given the feature tracks U. By summing over the discrete space of possible partitions, we can marginalize over the unknown partition J_X, where the index indicates the dependence on the dimensionality of the chosen structure X:

$$P(X, M|U) = \sum_{J_X} P(X, M, J_X|U)$$

The expectation maximization (EM) algorithm cannot be applied here, since the dimensionality of the structure X is not known. Even if sampling over the huge combined space of structure and motion might be feasible, the evaluation of the sum over all partitions is not, as the number of possible partitions for a specific number of feature tracks K is given by the Bell number

$$B_K = \sum_{k=1}^{K} S(K, k) \tag{5}$$

with

$$S(K, k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i \left(\begin{array}{c} k \\ i \end{array} \right) (k-i)^K \tag{6}$$

the Stirling number of the second kind. The Bell number grows hyper-exponentially in K with $B_1 = 1$, $B_5 = 52$, $B_{10} = 115975$ and $B_{50} \approx 1.9 \cdot 10^{47}$ and therefore prevents any enumeration for non-trivial problems.

In order to avoid the unknown dimensionality problem, we can integrate out the structure X, resulting in the posterior over the motion given the feature tracks. Indeed, once we have the motion of the robot, it is straightforward to solve for the structure. The posterior $P(M|U)$ over the motion M is obtained by marginalizing over both, the structure X_J and the partition J, where now the dimensionality of the structure depends on the specific partition J as indicated by the index:

$$P(M|U) = \sum_J \int_{X_J} P(M, J, X_J|U) \tag{7}$$

Again a sum over partitions makes direct evaluation impossible. However, this time EM is applicable, because we know the dimensionality of the motion M and can typically get a good initial estimate, either from the odometry of the robot, or from incremental scan matching or just by a heuristic.

III. An EM-Based Solution

We apply the EM algorithm to the loop closing problem based on the posterior $P(M|U)$ over the motion M from equation (7). Finding the motion M^* that maximizes this posterior $P(M|U)$ is equivalent to maximizing the joint distribution $P(M, U)$, as is maximizing the log of that distribution:

$$\begin{aligned} M^* &= \underset{M}{\text{argmax}} \;\; P(M|U) \\ &= \underset{M}{\text{argmax}} \;\; P(M, U) \\ &= \underset{M}{\text{argmax}} \;\; \log P(M, U) \end{aligned} \tag{8}$$

As in equation (7), the partition can be included into (8) by marginalization:

$$\begin{aligned} M^* &= \underset{M}{\text{argmax}} \;\; \log P(M, U) \\ &= \underset{M}{\text{argmax}} \;\; \log \sum_{J \in \mathcal{J}} P(M, J, U) \end{aligned} \tag{9}$$

A direct evaluation is not possible because of the combinatorial size of the sum, instead we apply the EM algorithm [14]. EM is an iterative algorithm that alternates between an expectation (E) and a maximization (M) step. A derivation of the algorithm and deeper insights in terms of a lower bound formulation is provided in [15]. Our notation is based on [3].

The EM algorithm applied to the loop closing problem iteratively improves an initial estimate of the motion M^0. In iteration t, the *E-step* determines the expected log-likelihood $Q^t(M)$ over the motion M,

$$Q^t(M) \triangleq \langle \log P(U, J|M) \rangle$$

where the expectation $\langle . \rangle$ is taken with respect to the distribution $f^t(J)$ over the partition J given the tracks U and the current motion estimate M^t:

$$f^t(J) \triangleq P(J|U, M^t) \tag{10}$$

The *M-step* then optimizes the expected log-posterior $Q^t(M) + \log P(M)$ with respect to the free motion variable M to obtain a better motion estimate M^{t+1}:

$$M^{t+1} \triangleq \underset{M}{\text{argmax}} \;\; \left[Q^t(M) + \log P(M) \right] \tag{11}$$

The distribution $f^t(J)$ over partitions cannot be evaluated directly because of the hyper-exponential number of partitions for K feature tracks that is given by the Bell number B_K as defined in (5). Instead we sample from this distribution in the E-step, which is known as Monte Carlo EM.

IV. Monte Carlo EM

We show how to apply Monte Carlo EM to the problem of loop closing. In Monte Carlo EM, or short MCEM [19], the distribution $f^t(J)$ from (10) over the hidden variable, in our case the partitions J, that is needed in the E-step, is replaced by a Monte Carlo approximation. We obtain this approximation by using Markov chain Monte Carlo or MCMC sampling.

MCMC methods produce samples from a target distribution $\pi(J)$ by simulating a Markov chain with the same equilibrium distribution. The algorithm starts from a random initial state $J^{(0)}$ and proposes probabilistically generated successor states, which is equivalent to running a Markov chain. It is important to note that this method produces samples of an arbitrary target function $\pi(J)$ while only requiring function evaluations at specific states.

A. E-step

For the expectation step we have to determine the function over partitions to sample from, starting from equation (10). Applying Bayes Law to this equation yields two terms, a prior $P(J|M^t)$ on the partition, and the likelihood $P(U|J, M^t)$ of the partition:

$$\begin{aligned} f^t(J) &= P(J|U, M^t) \\ &\propto P(U|J, M^t)P(J|M^t) \end{aligned}$$

The prior $P(J|M^t)$ on the partition depends on the current motion estimate M^t. We can therefore use information about the distance between points generated by tracks to determine their prior probability of belonging to the same set in the partition.

Using the independence of the measurements given fixed motion, in this case the current motion estimate M^t, the likelihood can be factored into sets U_j of tracks as defined in (2), that correspond to unique structure points:

$$P(U|J, M^t) = \prod_{j=1}^{N(J)} P(U_j|M^t)$$

In order to further evaluate this expression in terms of single measurements, the unknown location of the structure point x_j described by the set U_J has to be included by marginalization. By applying the chain rule, we obtain a prior on the structure point x_j as well as a likelihood $P(U_j|x_j, M^t)$ of a set of tracks that was defined in (3):

$$\begin{aligned} P(U_j|M^t) &= \int_{x_j} P(x_j, U_j|M^t) \\ &= \int_{x_j} kP(x_j|M^t)P(U_j|x_j, M^t) \end{aligned} \tag{12}$$

To evaluate equation (12) we have to solve for the integral over the point locations x_j. If we optimize each x_j given the motion M^t and all tracks $u \in U_j$ that apply to it and assume that the resulting posterior

$$P(x_j|U_j, M^t) = kP(x_j|M^t)P(U_j|x_j, M^t)$$

is approximated well by a Gaussian centered at $x_j^*(U_j|M^t)$, the optimized value of x_j, with covariance matrix $\Sigma_j(U_j|M^t)$, modulo an unknown constant k, then the integral from equation (12) can be approximated via Laplace's method by the following distribution

$$P(U_j|M^t) \approx P(x_j^*|M^t)\sqrt{|2\pi\Sigma_j^*|}P(U_j|x_j^*, M^t)$$

which yields the required density $f^t(J)$ over partitions J from equation (10):

$$f^t(J) \approx P(J|M^t) \prod_{j=1}^{N(J)} P(x_j^*|M^t)\sqrt{|2\pi\Sigma_j^*|}P(U_j|x_j^*, M^t) \tag{13}$$

Integrating out the continuous point x in the context of sampling is known as *Rao-Blackwellization* [12], as the variance in the sample is reduced by treating the continuous part of the state analytically.

B. M-step

In the maximization step, a better estimate M^{t+1} for the motion M is found using equation (11) to maximize the expected log-posterior $Q^t(M) + \log P(M)$ based on the approximated distribution $f^t(J)$ over the partitions J obtained in the E-step:

$$M^{t+1} = \operatorname*{argmax}_M \left[Q^t(M) + \log P(M)\right]$$

One of the resulting terms is the log prior $\log P(M)$ on the motion, that can for example be based on robot odometry or a motion estimate based on an incremental scan matching approach. The other term is the expected log-likelihood, that can be approximated by a sum over the samples $J^{(r)}, 1 \leq r \leq R$ that were obtained in the E-step, and finally rewritten in terms of a histogram $C^t(J)$ over the partitions:

$$\begin{aligned} Q^t(M) &= \langle \log P(U, J|M) \rangle_{P(J|U,M^t)} \\ &= \sum_J P(J|U, M^t) \log P(U, J|M) \\ &\approx \frac{1}{R} \sum_r \log P(U, J^{(r)}|M) \\ &= \sum_J C^t(J) \log P(U, J|M) \end{aligned}$$

A key realization to implementing this efficiently is that indexing the anonymous structure points x by the set of tracks U_x that determine them allows us to share the results of the optimization between many different partitions that share the set U_x. Please note that this is possible because the observations, and therefore also the tracks and their disjoint combinations as given by partitions, are independent given the motion. We define $C^t(U_x)$ to be the number of samples containing the partition U_x divided by the number of samples R:

$$C^t(U_x) = \frac{|\{J|U_x \in J\}|}{R}$$

This expression can be interpreted in terms of virtual structure. Each set of tracks corresponds to a virtual structure point, and the frequency of its occurrence in the sampling determines the weight it gets assigned. The M-step is therefore performed by an optimization with the structure replaced by the virtual structure, and the components weighted according to the histogram $C^t(U_x)$.

V. Reversible Jump MCMC Sampling

We use the trans-dimensional MCMC algorithm from [8] for Monte Carlo estimation in the E-step. We start with a random initial state $J^{(0)}$ and iterate the following steps:

1) Propose a move type $m \in M$ with probability $b_m(J^{(r)})$, where $J^{(r)}$is the current state.

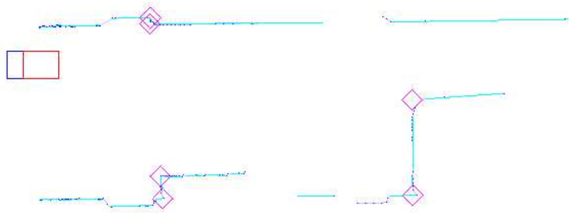

Fig. 1. The result of feature detection is shown together with the raw data. The raw laser range data is shown as black dots with the red/blue robot indicating the center of the range sensor. Incrementally fitted line segments are drawn as cyan lines, while diamonds represent corners detected as intersections of neighboring line segments.

2) Generate a random sample u from the move-specific proposal density g_m. The move type m and random sample u determine how to move from the current state $J^{(r)}$ to the proposed state J'.
3) Calculate the corresponding reverse move (m', u').
4) Compute the acceptance ratio

$$a = \frac{\pi(J')}{\pi(J^{(r)})} \frac{b_{m'}(J')}{b_m(J^{(r)})} \frac{g_{m'}(u')}{g_m(u)} \left| \frac{\partial(J', u')}{\partial(J^{(r)}, u)} \right| \quad (14)$$

where the Jacobian factor corrects for the change in variables (see below).
5) Accept $J^{(r+1)} \leftarrow J'$ with probability $\min(a, 1)$, otherwise $J^{(r+1)} \leftarrow J^{(r)}$.

The generated sequence of states $\{J^{(r)}\}$ will be a sample from $\pi(J)$ if the sampler is run sufficiently long, and one discards the samples in the initial "burn-in" period of the sampler to avoid dependence on the chosen start state.

One possible set of move types for loop closing consists of "Merge" and "Split" for combining two sets of tracks and separating them again. Care has to be taken to ensure that the move from $(J^{(r)}, u)$ to (J', u') is reversible and therefore a diffeomorphism. One requirement is that the dimensions on both sides have to match. Note that in our case the Jacobian of the diffeomorphism $\left|\frac{\partial(J',u')}{\partial(J^{(r)},u)}\right|$ is always 1 because we integrate out the continuous part of the space.

VI. Experiments and Results

We have applied the algorithm to corner features extracted from laser range data. The corners are obtained by a modified version of the incremental line fitting algorithm [7] by extracting intersections between neighboring line segments. An example of a processed laser scan is shown in Figure 1, with detected corners indicated as diamonds.

As every optimization technique, EM can easily get stuck in local minima. To avoid this problem, we have applied an annealing scheme as is commonly used in the context of EM. Starting from a multiple of the real measurement sigma, the value is decreased linearly in each iteration.

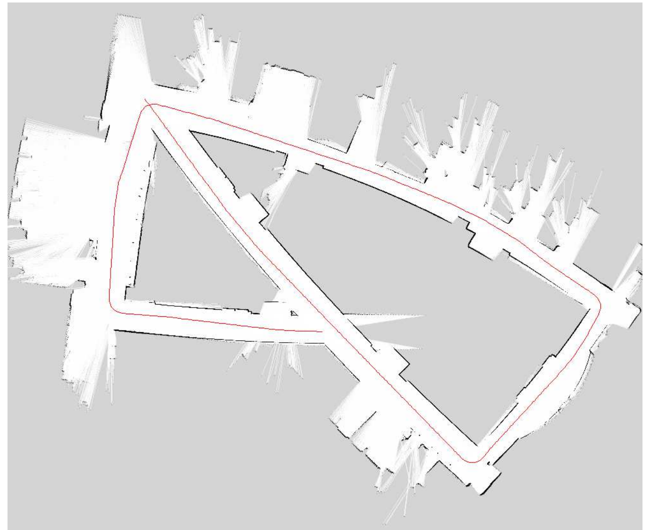

Fig. 2. Evidence grid plot of the raw laser data of our sequence covering parts of the 3rd floor of the Technology Square Research Building (TSRB).

Fig. 3. Evidence grid plot after loop closing (TSRB). Better local alignment of scans can be achieved by other methods, but is not done here to show the result based on sparse features only. The trajectory covers approximately 40 meters by 15 meters.

We have applied our algorithm to two sets of laser data. The first was recorded in our research facility, while the second data set is part of the Intel Oregon sequence that was obtained from the Robotics Data Set Repository [11]. Thanks go to Maxim Batalin for providing this data. Evidence grid plots of the original data sets are shown in Figure 2 and Figure 4. After applying our algorithm, we obtain the corrected maps as shown in Figure 3 and Figure 5. Please note that with a sparse set of features, as is used here, not all frames are perfectly aligned. However, the loop is closed, and local alignment can be improved by other methods [13].

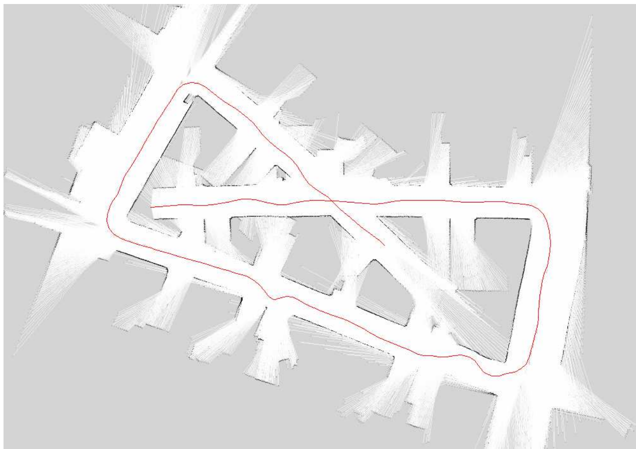

Fig. 4. Evidence grid plot of the raw laser data of parts of the Intel Oregon sequence.

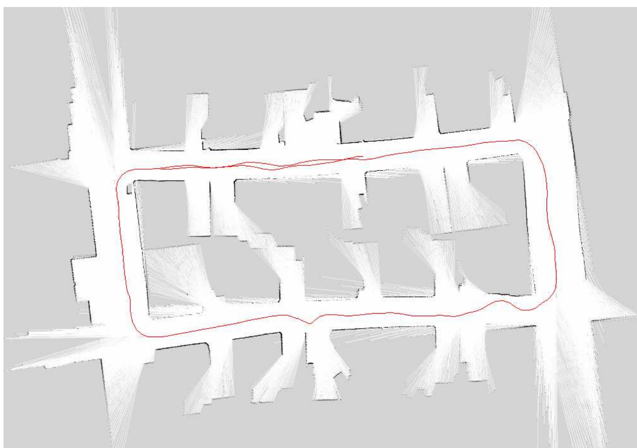

Fig. 5. Evidence grid plot after loop closing (Intel Oregon). Better local alignment of scans can be achieved by other methods, but is not done here to show the result based on sparse features only.

VII. Conclusion

In this paper we have presented a novel approach to solving the loop closing problem in range-based SLAM. Within an MCEM framework, we iterate between estimating virtual structure and improving the motion estimate. Since the number of structure points is not known, we are faced with a model selection problem, which is addressed by using a reversible jump MCMC sampler to estimate the virtual structure. It is important to note that our algorithm never commits to a specific combination of tracks or even a specific number of structure points, but yields a probability distribution over all possible maps.

The approach has been implemented and successfully applied to real data. We are investigating the extension of our loop closing algorithm to the domain of visual SLAM, for which this feature based method seems especially suitable. In this more complex domain the algorithm would especially benefit from a more intelligent proposal function. Future work also includes the extension of the algorithm to multi-robot mapping.

Acknowledgments

We would like to thank Wolfram Burgard for his generous support, and Dirk Haehnel for valuable discussions.

References

[1] H. Baltzakis and P. Trahanias. An iterative approach for building feature maps in cyclic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 576–581, 2002.

[2] H. Baltzakis and P. Trahanias. Closing multiple loops while mapping features in cyclic environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 717–722, 2003.

[3] F. Dellaert. The expectation maximization algorithm. Technical Report GIT-GVU-02-20, College of Computing, Georgia Institute of Technology, February 2002.

[4] F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun. Structure from motion without correspondence. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2000.

[5] F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun. Feature correspondence: A Markov chain Monte Carlo approach. In *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, 2001.

[6] F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine learning*, 50(1-2):45–71, January - February 2003. Special issue on Markov chain Monte Carlo methods.

[7] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2002.

[8] P.J. Green. Trans-dimensional Markov chain Monte Carlo. Chapter in Highly Structured Stochastic Systems, 2003.

[9] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, 1999.

[10] D. Haehnel, W. Burgard, D. Fox, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[11] A. Howard and N. Roy. The robotics data set repository (Radish), 2003.

[12] M. Kaess, R. Zboinski, and F. Dellaert. MCMC-based multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points. In *Eur. Conf. on Computer Vision (ECCV)*, volume 3023 of *Lecture Notes in Computer Science*, pages 329–341, Prague, Czech Republic, 2004. Springer.

[13] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *AR*, pages 333–349, April 1997.

[14] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley & Sons, 1997.

[15] T.P. Minka. Expectation-Maximization as lower bound maximization. Tutorial published on the web at http://www-white.media.mit.edu/ tpminka/papers/em.html, November 1998.

[16] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI Nat. Conf. on Artificial Intelligence*, 2002.

[17] K. Murphy and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, January 2001.

[18] S. Se, D. Lowe, and J. Little. Vision-based mapping with backward correction. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 153–158, 2002.

[19] M.A. Tanner. *Tools for Statistical Inference*. Springer Verlag, New York, 1996. Third Edition.

[20] S. Thrun, S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *AAAI*, pages 989–995, 1998.