Hybrid Inference for Sensor Network Localization using a Mobile Robot

Dimitri Marinakis,
CIM, McGill University
dmarinak@cim.mcgill.ca

David Meger,
University of British Columbia
dpmeger@cs.ubc.ca

Ioannis Rekleitis,
Canadian Space Agency
yiannis@cim.mcgill.ca

Gregory Dudek
CIM, McGill University
dudek@cim.mcgill.ca

Abstract

In this paper, we consider a hybrid solution to the sensor network position inference problem, which combines a real-time filtering system with information from a more expensive, global inference procedure to improve accuracy and prevent divergence. Many online solutions for this problem make use of simplifying assumptions, such as Gaussian noise models and linear system behaviour and also adopt a filtering strategy which may not use available information optimally. These assumptions allow near real-time inference, while also limiting accuracy and introducing the potential for ill-conditioning and divergence. We consider augmenting a particular real-time estimation method, the extended Kalman filter (EKF), with a more complex, but more highly accurate, inference technique based on Markov Chain Monte Carlo (MCMC) methodology. Conventional MCMC techniques applied to this problem can entail significant and time consuming computation to achieve convergence. To address this, we propose an intelligent bootstrapping process and the use of parallel, communicative chains of different temperatures, commonly referred to as parallel tempering. The combined approach is shown to provide substantial improvement in a realistic simulated mapping environment and when applied to a complex physical system involving a robotic platform moving in an office environment instrumented with a camera sensor network.

Introduction

We propose a technique whereby the individual sensing nodes that make up a distributed sensor network can determine their respective positions and configuration in 2D space without having any *a priori* knowledge of where they are. We address this instance of the *sensor network localization* problem by augmenting the stationary nodes of the network with a mobile robot, as illustrated in figure 1. As we discuss below, several authors have considered the maximum likelihood estimation of a network configuration in this context, but this does not account for either the certainty of the estimates or a strategy for how to improve it. A small number of authors have also considered methods for estimating the probability distribution over node locations, but at substantial computational cost. This paper exploits the synthesis of maximum likelihood estimation and stochastic
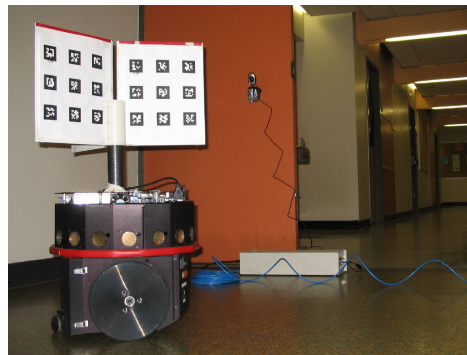
Figure 1: The target hardware platform for our inference technique: a mobile robot is observed by one of the component stationary cameras in a sensor network.

estimation of the probability distribution to attempt to capture some of the advantages of each approach.

Networks of sensors are rapidly becoming ubiquitous as our society deploys emplaced motion sensors, emplaced cameras, mobile robots, and even cell phones which can sense their environment. This increasing prevalence has motivated researchers to consider the potential of a *smart house* in which actuators and sensors respond to the behaviour of the resident in order to help them with their daily activities. While the physical deployment of such sensors is relatively trivial, the configuration of such a system has remained a challenge. Localization information for each device is required in order to carry out the assigned tasks, but obtaining such information by manual intervention is time consuming and error-prone. Obtaining this localization information directly from the sensor data, so that the configuration process is accurate and automatic would help realize the practical deployment of smart houses as a consumer technology.

A mobile robot's motion within the network can facilitate localization by explicitly transferring positional information between disparate sensor locations. By maintaining an ongoing estimate of the robot's location, the positions of any sensor it interacts with can be estimated (and updated) as it moves through the network. This is accomplished through probabilistic modeling of the robot's position and motion, the ability of the network nodes to sense the robot when it enters their neighborhood, and estimation of the displace-

ment between the robot and the nodes it encounters.

When designing our localization system, we seek not only a maximum likelihood estimate (MLE) for the robot's position, but also a probability density function (PDF). This information enables later decisions to be made conditional on the confidence of the estimates, and facilitates adaptive exploration processes and higher level task planning. Obtaining a PDF may entail more computational cost than obtaining the MLE, and so has often been overlooked by previous authors. In our scenario, the localization problem can be thought of as an initial map building phase, which will be followed by a much longer period of deployed use of the network. For this reason, the additional computation required to obtain an accurate distribution is well justified.

The remainder of this paper will detail a method which attempts to achieve the conflicting goals of near real-time position estimation, and consistent, convergent inference of the underlying distributions. To accomplish this, we will propose a combination of techniques. The computationally efficient, but error prone extended Kalman filter will be used in an online fashion to process the incoming sensory data. A more computationally expensive, but highly accurate, inference technique based on Markov Chain Monte Carlo will be used periodically to increase accuracy, prevent divergence, and diagnose faults. The significant and time consuming computation required by standard MCMC inference to achieve convergence is addressed by our combined method in two ways: 1) bootstrapping the chain with a high likelihood state obtained from the EKF; and, 2) the use of parallel, communicative chains of different temperatures, commonly referred to as parallel tempering. Finally, The combined method will be evaluated on simulated results and a physical localization scenario.

Background

The sensor network localization problem has received considerable attention recently. For our purposes, previous approaches can be grouped into those which operate on a static network with some form of node to node communication, and those which operate in the context of hybrid static and mobile nodes. When all nodes in the network are static, information such as radio signals, or overlapping images of the environment can be employed to obtain internode distances after careful inference. Recently, both (Ihler *et al.* 2005) and (Marinakis & Dudek 2006) have applied sampling-based techniques to estimate the posterior distribution of sensor nodes. In the case of a single mobile robot within a static network, the robot's motion can facilitate localization. (Meger, Rekleitis, & Dudek 2006; Rekleitis, Meger, & Dudek 2006) considered this scenario, but employed a relatively simple online method for estimating sensor positions. We extend this work by combining online and batch inference, in order to achieve higher accuracy.

In many ways, the issue we are considering is similar to the simultaneous localization and mapping (SLAM) problem in traditional mobile robot research, and this research area has inspired several solution techniques which we draw upon. The extended Kalman filter, as was pioneered by

(Smith, Self, & Cheeseman 1990), is a common approach to SLAM. EKF-based SLAM solutions share many of the same limitations that will be discussed below, so a family of hybrid sampling-based approaches referred to as FastSLAM (Montemerlo & Thrun 2003) have been proposed. Furthermore, several authors have recently noted that the filtering approach, which maintains only the most recent pose of the mobile agent, is prone to errors, and have instead estimated the entire set of previous poses. For example, (Dellaert & Kaess 2006) apply many of the familiar Kalman filter assumptions in the context of smoothing rather than filtering. Several methods which employ hybrid online and offline, global correction methods for the SLAM problem are perhaps the most similar to our own. Scan-Matching for Alignment (Lu & Milios 1995) and its later practical implementation, known as Local Registration and Global Correlation (Gutmann & Konolige 1999) are two examples of such approaches. These methods are similar in concept to our approach, but do not return a full representation of the underlying distribution and furthermore are targeted at robotic mapping, whereas our work targets the more sparse observations obtained when mapping a sensor network.

Although our network localization problem can be viewed in the context of SLAM, there are some practical differences. First, our landmarks are actually network components and can be considered uniquely identifiable, so there is no correspondence issue. Second, we can assume that in our system, the robot will operate for the most part within the confines of sensor-network deployed region and will ultimately visit the local area of each network component many times. Given such behaviour, it will be desirable to know not only the most likely location for each sensor, but with what confidence or certainty such a location is known. Finally, in the scenario we consider, our mobile robot is only able to detect or be detected by the deployed sensors and is assumed not to have sensing capabilities that could be used to identify additional landmarks in the environment and use these for localization.

Problem Definition

Sensor network localization based on a moving agent involves inferring the positions of each sensor node m_i, which is part of the map of the sensors $m^n = [m_1 m_2 ... m_n]$. These postions can only be measured relative to the position of the above mentioned mobile robot at a given time, $s^t = [s_1 s_2 ... s_t]$, and so both quantities must be estimated simultaneously. The measurements available are the position of a sensor relative to the robot at time t, denoted z_t and the position of the robot at time t relative to its postion at time $t-1$, denoted u_t.

This problem can be modelled as probabilistic inference of the map and the robot poses conditioned on the observations, as represented by the directed graphical model shown in figure 2. The posterior distribution, $p(m^n, s^t | z^t, u^t)$, can be factored into the product of many local conditional distributions, by exploiting the conditional independencies in the network as is common practice for probabilistic graphical models.
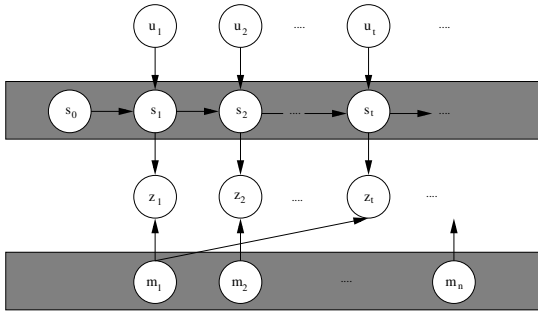
Figure 2: The quantities of interest in the Sensor Network localization problem can be modelled as a Bayesian Network in order to exploit conditional independencies. The robot poses and map (highlighted in grey) must be inferred, given observed data.

For the sensor network localization problem, there are two classes of local conditional likelihoods: $p(z_t|s_t, m)$, which is referred to as the measurement model; and $p(s_t|u_t, s_{t-1})$ which is known as the motion, or odometry, model. These distributions for a particular physical instantiation can be determined empirically or determined by physical modeling; however, there are several challenging properties common to each. The local distributions for this problem are over the continuous valued, potentially high dimensional variables representing position and orientation. Exact inference in this context remains an unsolved problem, so we must turn to simplifying approximations or sophisticated computational techniques to perform inference.

Extended Kalman Filter Localization

The dynamic structure of the problem and conditional independencies inherent in the graphical structure facilitate the formulation commonly referred to as a Bayes Filter (note that s in the following equation represents the previous state):

$$p(m^n, s^t|z^t, u^t) \propto \qquad (1)$$
$$p(z_t|s_t, m) \int p(s_t|u_t, s)p(s^{t-1}, m^n|z^{t-1}, u^{t-1})ds$$

The recursive formulation given in equation (1) provides a formulation for accumulating information from one time-step to the next. This process remains intractable for general distributions due to the need to integrate over all possible robot postions s_t. Kalman developed a computationally efficient, closed-form solution, suitable for online estimation assuming a linear system with normally distributed posterior and noise models, which is commonly referred to as the Kalman filter. The Kalman filter is unique among linear filters in that it is proveably optimally efficient in a least-squares sense under the assumptions given above. That is, the estimates minimize the expected value of squared error and the estimated covariance is the best possible measure for the error in the estimate. Unfortunately, real physical systems are rarely linear and Gaussian, so the assumptions of the Kalman filter do not hold in practice. In this case, the true physical models may be approximated as linear (for example, using a truncated Taylor series expansion) and the closest possible Gaussian can be fit to the true error model. These steps produce a non-linear filter, known as the extended Kalman filter, which provides no guarantees of optimality, but often performs well in practice.

The implications of the approximations made by the EKF are evident in its application to our sensor network localization problem. The linearization of the motion and measurement models is performed at the current mean estimate, not at the unobservable, underlying states of the world. This process introduces non-negligible errors, and significant loss of accuracy. Also, the additional unmodeled errors from linearization are not accounted for in the computation of estimated covariance, which bias the EKF towards overconfidence. That is, the estimated covariance is an underestimate of the true error, which can lead to hazardous situations in a practical scenario. Finally, the approximation of the input noise distributions as Gaussian may be inaccurate, and limits the ability to model general posteriors.

While such limitations paint a bleak picture for the EKF as an estimator, it is often able to perform well for short periods of time on realistic problems. The accumulation of errors may only become a problem for quite long term operation. The following section will present the second, complementary technique which comprises our hybrid approach.

MCMC Localization

In this section we will describe a technique that attempts to directly sample the posterior distribution, $p(m^n, s^t|z^t, u^t)$, for the pose of every node of our sensor network. Conceptually, we form a graph $< V, E >$, where V is the set of vertices and E the set of connecting edges. The vertices of this graph are the robot positions over time s^t, and the sensor locations m^n. The edges, or constraints, are the odometry measurements u^t connecting consecutive robot positions over time and the measured relative positions z^t between the robot poses and network components. Using a model characterizing the error in the measurements, we can calculate the density of any particular configuration $x = (m^n, s^t)$ of the graph through the application of Bayes law and the assumption of a uniform prior over all potential configurations:

$$p(x|z^t, u^t) = \frac{p(z^t, u^t|x)p(x)}{p(z^t, u^t)}$$
$$p(x|z^t, u^t) \propto p(z^t, u^t|x)$$

We then define our target distribution π to be the relative likelihood of a given configuration x:

$$\pi(x) = p(z^t, u^t|x) \qquad (2)$$

Given this ability to calculate the density of our target distribution at a specific point, we can employ the well known Metropolis-Hastings (M-H) algorithm to generate representative samples. In our application of the M-H algorithm to sensor network localization, we use a proposal function $Q(x)$, which we will define below, that generates a new state x' given the current state x. The proposal x' is then either accepted or rejected with probability α, where α is calculated as:

$$\alpha = \min\left(1, \frac{\pi(x')Q(x)}{\pi(x)Q(x')}\right) \quad (3)$$

We selected the following proposal function, $Q(x)$: given the current state in the chain x, a symmetric transition is proposed to a new state x' by altering a feature (*either* the orientation, θ or spatial position, (x, y)) of a single vertex in the graph. The vertex and feature are selected at random and some amount of normally distributed noise is added to the original value. The new state x' is then accepted or rejected based on the relative density of x and x' as calculated with equations 2 and 3.

Additionally, to speed convergence and mixing of the chain, we used the technique of parallel tempering. Parallel tempering (Geyer 1991) is a MCMC variant in which multiple replica-exchanging chains of different *temperatures* are simulated in parallel. The temperature of a chain can be thought of as specifying the relative 'smoothness' of its target distribution. Usually, a chain C_k of temperature t, will use the density: $\pi_k = (\pi)^{\frac{1}{t}}$. While the lowest temperature chain attempts to sample from the target distribution, π, the higher temperature chains sample potentially easier to characterize versions of the original target distribution.

During a simulation, after a number of within chain proposals, two consecutive chains C_i and C_{i+1} are selected randomly and their current configurations X_i and X_{i+1} are exchanged (or not) according to the M-H acceptance ratio:

$$\alpha = \min\left(1, \frac{\pi_i(X_{i+1})\pi_{i+1}(X_i)}{\pi_i(X_i)\pi_{i+1}(X_{i+1})}\right) \quad (4)$$

In this work we employed 10 chains geometrically distributed in temperature between $t = 1$, the cold chain, and $t = 10$, the hottest chain. We varied the standard deviation of the normally distributed noise used for proposal generations from a factor of 1 cm for the coldest chain to 12 cm for the hottest chain. These values were determined by cross-validation and found to provide satisfactory performance for the experimental work presented below. After an initial burn-in stage, samples are drawn periodically from the cold chain and are used to form a particle representation of the PDF for the pose of the network.

Parallel tempering achieves good performance by allowing high temperature chains to make fast, less-restrained exploration of the underlying probability landscape. Promising realizations discovered by these hot chains are fed down to colder chains, and ultimately to the principle chain. The result is faster mixing than in the single chain variant, and hence more complex target distributions require less computational effort to characterize through sampling. The technique of parallel tempering has become wide spread in fields such as physics and biology. Tempering has some conceptual similarities to existing techniques in the related field of SLAM that exploit relaxed versions of the final problem in the inference process; *e.g.* (Frese, Larsson, & Duckett 2005). Here we consider the application of tempering to sensor network self-localization.

Hybrid Sensor Network Self-Localization

With the EKF, we have available a real time localization technique that can be used when there has not been the opportunity to run the MCMC localization technique. This can be useful for the initial exploration, and also when quick adjustments are required such as when a new network component is added, or an existing component fails. Additionally, the filter can be employed for everyday navigation tasks once the relative position of the network components has been determined to satisfaction. In the self-localization task, we envision the MCMC localization being run, periodically, after some amount of exploration.

The two localization methods can complement each other. The EKF can be used to bootstrap the MCMC process. Initializing the chain with a high likelihood sample reduces the burn in time and speeds convergence, allowing the chain to produce more accurate results more quickly. Additionally, the results obtained by the MCMC could be used to adjust both the means and confidences of the EKF. In the next sections we will examine some of these issues with experiments conducted in simulations and with real hardware.

Results from Simulations

Both the EKF and our hybrid technique were performed on data from a realistic simulator. Our results demonstrate that: 1) the hybrid method is capable of improving upon the pose estimate provided by the EKF, when given sufficient data; and 2) the method more faithfully represents the underlying distribution. Figure 3 shows the typical error progression during the exploration process for a network of six sensors when the MCMC was initialized with the EKF calculated values at various intervals. When there are few constraints, the MCMC approach is unable to greatly improve upon the mean EKF estimate. However, when the robot has visited enough network components, the MCMC technique is capable of finding a higher likelihood global configuration which is usually more accurate than the EKF mean. When the mean squared error of sample sets drawn from the original EKF are compared to the corrected MCMC samples, it can be seen that the global inference technique considerably tightens the distribution around high likelihood values.

A second observation verified through simulations is that the MCMC technique generally recovers a considerably different distribution than the one suggested by the EKF. In general the distributions diverge as more data is collected. Figure 3(c) displays the generalized ($95th$ quantile) Hausdorff distance was used with a L2-norm kernel. This metric estimates the distance between distributions by drawing a single test set of samples from the EKF, and using the mean distance between this set and 10 additional sets of drawn EKF samples to normalize comparisons with the EKF test set to MCMC obtained samples. The Hausdorff distance increases as the robot's path length increases. This occurs because the estimated distribution becomes more complex as more constraints are added, and the single Gaussian used by the EKF is an increasingly poorer approximation. This demonstrates the danger of relying on linearized representations of distributions, even if the mean is accurately deter-
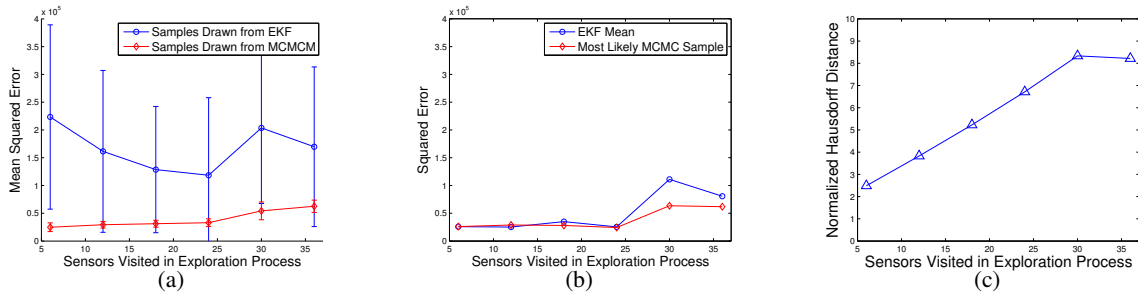
Figure 3: Results obtained from a simulated, six sensor environment. The MCMC technique was run after every 6 sensor visits (steps) of the mobile robot. Vertical axis reports the squared error in sensor and final robot locations from the ground truth positions. (a) The average error of a sample drawn from the respective distributions. (b) The total squared error of the respective ML estimated positions. (c) Normalize Hausdorff distance between the distributions suggested by the EKF and MCMC.

mined. In the next section we will present results demonstrating our hybrid localization technique on a real sensor network.

Experimental Results

Our hybrid approach has been applied to mapping data gathered using real robot and sensor network hardware. The target sensor network is located in an office environment, and consists of seven networked cameras. The robot travelled through a rectangular loop and a triangular loop connected by a long straight hallway with length approximately 50 m as shown in figure 4(a). A Nomadics Scout robot mounted with a target with six recognizable patterns, such as the one shown in figure 1 was used to perform a calibration procedure and obtain position measurements, using a method previously described in (Rekleitis, Meger, & Dudek 2006). Due to the size of the environment, and lack of line-of-sight between camera positions, ground truth data could not be collected for this experiment, but there are several measures which can be used for qualitative assessment of estimation accuracy on this data. First, care was taken to return the robot to within a few centimetres of its initial position at the end of the run, which implies the first and last robot positions should agree very closely in any accurate estimate. Also, estimated camera location accuracy can be estimated visually, by comparing to the camera locations recorded on figure 4(a).

The odometry and camera position measurements gathered during these experiments were used as input for our hybrid estimation procedure. Figure 4 demonstrates the performance in terms of final camera position estimates as well as the ability of the parallel tempering method to estimate the distribution efficiently. The final particle clouds produced by the hybrid method are shown in figure 4(b). The final robot positions can be observed to lie within a meter from the initial position, which is a strong indicator of map accuracy, as the path length is over 360 m in total. The inference in this experiment took several hours on a 2.2GHz P4 with 1GB of RAM (implementation was in unoptimized matlab).

Figure 4(c) provides a comparison of the various estimation techniques presented in this paper with respect to the

likelihood of the data given each final model. The EKF mean produces a relatively poor estimate, with the MCMC-based approach achieving a much higher likelihood. All of the chains of different temperatures used in parallel tempering are shown, with the lowest temperature chain achieving the highest likelihood, as expected. The parallel tempering approach was able to achieve a higher likelihood state after many fewer proposals than the standard M-H algorithm which was implemented for comparison purposes.

Additionally, an informal variance analysis suggests that the cold chain of the parallel tempering variant mixes much faster than the standard single chain approach and exhibits variance characteristics typical of a system at or near convergence (figure 5).

Conclusion

This paper has examined a novel combination of online and batch state estimation techniques for the sensor network localization problem. The extended Kalman filter has been shown to provide fast, short-term results which facilitate operation-time robot navigation. Parallel tempering MCMC has been shown as an effective global optimization technique for this problem, providing highly accurate correction over long periods. The combination of the two methods is beneficial not only for practical reasons, but also due to the fact that each method is able to aid the other in computation. The EKF estimate gives a reliable starting point for the MCMC refinement, and speeds convergence significantly. Once a reliable estimate has been obtained, the mean and covariance EKF can be corrected, allowing it to remain consisted over longer periods. This component of the system has been valided informally, and we leave the formal validation of the long term behaviour for future work.

Several aspects of the sensor network localization problem remain challenging, and worthy of continued study. Distribution of computation is a central issue for many network applications, and has not been considered in this work since the presence of a central, high powered computational device in our hardware setup did not require this capability. Also, we have not formally addressed steady state convergence, or when enough samples have been drawn to accu-
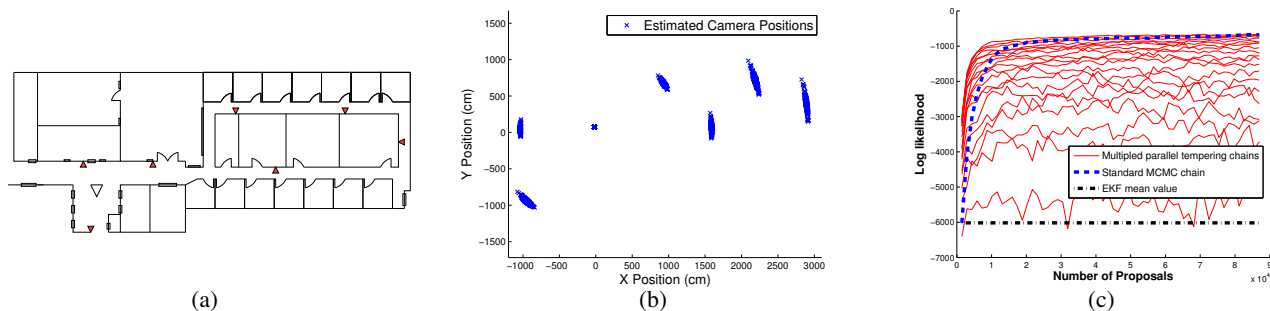
Figure 4: Results from the hybrid approach on estimating the map shown in (a). (b) The final distributions of camera and robot positions estimated by parallel tempering. (c) The likelihoods of each of our approaches (not all of data shown for purposes of clarity).
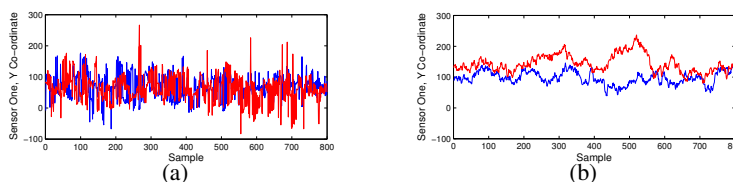

Figure 5: Overlaid plots of samples drawn from two post burn-in proposal windows for the cold chain of the parallel tempering MCMC variant (a) and for the single chain variant (b).

rately represent the target distribution. Finally, the frequency at which MCMC inference is required to keep the EKF consistent is an interesting question. Measures such as magnitude of covariance and likelihood of each incoming measurement under the estimated model are among promising approaches which indicate the need for a correction to be applied.

The added accuracy and robustness produced by our hybrid method makes this algorithm and excellent candidate for application on the type of error-prone hardware that is so common in real robot platforms. In fact, given the tendency of common approaches for state estimation to diverge under realistic error conditions, a method for diagnosis and correction of catastrophic errors, such as ours, is almost essential in a real system.

References

Dellaert, F., and Kaess, M. 2006. Square Root SAM: Simultaneous location and mapping via square root information smoothing. *International Journal of Robotics Research*. Special issue on RSS 2006.

Frese, U.; Larsson, P.; and Duckett, T. 2005. A multi-level relaxation algorithm for simultaneous localisation and mapping. *IEEE Trans. Robot.* 21(2):196–207.

Geyer, C. J. 1991. Markov chain monte carlo maximum likelihood. In *Computing Science and Statistics: Proc. of the 23rd Symposium on the Interface*, 156–163.

Gutmann, J.-S., and Konolige, K. 1999. Incremental mapping of large cyclic environments. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA'99),*.

Ihler, A. T.; Fisher III, J. W.; Moses, R. L.; and Willsky, A. S. 2005. Nonparametric belief propagation for self-calibration in sensor networks. *IEEE Journal of Selected Areas in Communication*.

Lu, F., and Milios, E. 1995. Optimal global pose estimation for consistent sensor data registration. In *International Conference in Robotics and Automation*, volume 1, 93–100. IEEE.

Marinakis, D., and Dudek, G. 2006. Probabilistic self-localization for sensor networks. In *AAAI National Conference on Artificial Intelligence*.

Meger, D.; Rekleitis, I.; and Dudek, G. 2006. Autonomous mobile robot mapping of a camera sensor network. In *The 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 155–164.

Montemerlo, M., and Thrun, S. 2003. Simultaneous localization and mapping with unknown data association using fastslam. In *IEEE International Conference on Robotics and Automation*, volume 2, 1985 – 1991.

Rekleitis, I.; Meger, D.; and Dudek, G. 2006. Simultaneous planning, localization, and mapping in a camera sensor network. *Robotics and Autonomous Systems* 54(11):921–932.

Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. In Cox, I., and Wilfong, G. T., eds., *Autonomous Robot Vehicles*. Springer-Verlag. 167–193.