Topological Mapping through Distributed, Passive Sensors

Dimitri Marinakis, Gregory Dudek

Centre for Intelligent Machines, McGill University

3480 University St, Montreal, Quebec, Canada H3A 2A7

{dmarinak,dudek}@cim.mcgill.ca

Abstract

In this paper we address the problem of inferring the *topology*, or inter-node navigability, of a sensor network given non-discriminating observations of activity in the environment. By exploiting motion present in the environment, our approach is able to recover a probabilistic model of the sensor network connectivity graph and the underlying traffic trends. We employ a reasoning system made up of a stochastic Expectation Maximization algorithm and a higher level search strategy employing the principle of Occam's Razor to look for the simplest solution explaining the data. The technique is assessed through numerical simulations and experiments conducted on a real sensor network.

1 Introduction

In this paper, we are interested in recovering a topological representation of a sensor network embedded region that identifies physical inter-sensor connectivity from the point of view of an agent navigating the environment (figure 1); as opposed to a description of the network's wireless communication connectivity (the conventional use of the word topology in wireless networks).

We assume that we have no prior knowledge of the relative locations of the sensors and that we have only unlabelled observations of activity in the environment (*i.e.* we make the pessimistic assumption that the objects being observed cannot be distinguished from one another). We must use observational data returned from our sensors to understand the motion of agents present in the environment. By inferring underlying patterns in their motions we can then recover the relationships between the sensors of our network.

Our approach employs a reasoning system that is built on a fundamental topology inference algorithm that takes the sensor observations and environmental assumptions as inputs and returns the network parameters. This algorithm is formulated using Monte Carlo Expectation Maximization (MCEM), but it depends on fixed values for certain numerical parameters that represent *a priori* knowledge regarding traffic patterns in the environment. The reasoning system searches over the input parameter space of the fundamental algorithm
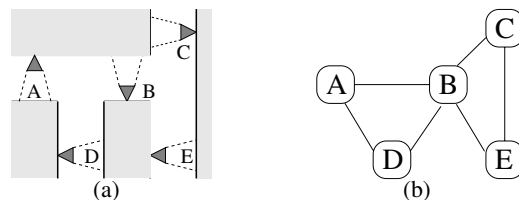

Figure 1: Example of a sensor network layout (a) and corresponding topology (b).

to find a global solution that optimizes a more abstract objective function based on the principle of Occam's Razor. [1]

The final output of our approach is a probabilistic model of the sensor network connectivity graph and the underlying traffic trends. The simplest application of this work would allow a set of sensors to be "dropped" into an environment and to automatically learn the topology of their layout. This autonomous calibration ability can be considered a step towards the larger goal of self-configuring intelligent systems. The information could be used in the self-calibration of a monitoring application or for route-planning purposes in a hybrid system employing a mobile robot. Additionally, applications that log data for offline analysis should be able to benefit from our technique. For example, a vehicle monitoring network distributed about a city could help make decisions about road improvements which might best alleviate congestion.

The topology of a sensor network, as we define it, must take into account the spatial constraints of the environment as they determine the inter-node connectivity parameters. The topological mapping problem has been well explored in mobile robotics [Shatkay and Kaelbling, 1997] [Choset and Nagatani, 2001] [Remolina and Kuipers, 2004] [Ranganathan and Dellaert, 2005]. Most sensor network related investigations, however, have been more recent [Makris *et al.*, 2004] [Marinakis and Dudek, 2006]; the outcome is generally a graph where vertices represent embedded sensors in the region and edges indicate navigability. By combining this topological description with any additional metric information obtainable from the surrounding environment, further in-

---

[1]Occam's Razor is the principle enunciated by William of Occam that the simplest explanation is the best.

formation regarding obstructions and motion corridors could be inferred. For example, two spatially proximal nodes that were not topologically adjacent would suggest a barrier of some sort.

While much of the research conducted on sensor networks is based on developing distributed and efficient algorithms appropriate for networks of low-powered sensor platforms, recently there has been a shift towards more complex approaches incorporating advanced probabilistic techniques and graphical models [Ihler *et al.*, 2005] [Paskin *et al.*, 2005]. The traditional sensor network assumption of homogenous systems of impoverished nodes is making way for tiered architectures that incorporate network components of some computational sophistication [Dantu and Sukhatme, 2006]. Note that a hierarchical arrangement based on computational power holds true for several real world sensor networks, especially in data collection systems [Wang *et al.*, 2003].

One problem in sensor networks that occasionally requires above average computational effort is the processing of distributed and information-poor observations. For example, in [Songhwai Oh and Sastry, 2006], event detections alone were used for the tracking of multiple targets using Markov Chain Monte Carlo (MCMC). Similarly, [Pasula *et al.*, 1999] approached a traffic monitoring problem using limited sensor data observations through a stochastic sampling technique.

A related problem domain that generally employs a complex probabilistic framework and computationally intensive techniques is the simultaneous localization and mapping (SLAM) problem in mobile robotics. Recently hybrid robot / sensor network systems have been employed to address SLAM issues. Examples include [Rekleitis *et al.*, 2005] in their use of an extended Kalman filter, and [Djugash *et al.*, 2006] who incorporate inter-sensor range data from a deployed sensor network in their approach.

In remainder of this paper we describe a computationally heavy but powerful approach for constructing a topological representation of a network embedded region based on distributed observations collected from passive, information-poor sensors. Our approach builds on some recent work [Marinakis *et al.*, 2005] [Marinakis and Dudek, 2006] that has shown the validity of an MCEM-based algorithm for sensor network topology inference. This technique uses only detection events from the deployed sensors and is based on reconstructing plausible agents trajectories. However, the algorithm requires significant prior knowledge regarding local traffic patterns that limit its general applicability. In the approach presented here, we incorporate the fundamental algorithm developed in this previous work into a higher level reasoning system that is able to remove much of its reliance on prior assumptions.

2 Problem Description

We describe the problem of topology inference in terms of the inference of a weighted directed graph which captures the spatial relationships between the positions of the sensors' nodes. The motion of multiple agents moving asynchronously through a sensor network embedded region can be modeled as a semi-Markov process. The network of sensors is described as a directed graph $G = (V, E)$, where the vertices $V = v_i$ represent the locations where sensors are deployed, and the edges $E = e_{i,j}$ represent the connectivity between them; an edge $e_{i,j}$ denotes a path from the position of sensor v_i to the position of sensor v_j. The motion of each of the N agents in this graph can be described in terms of their transition probability across each of the edges $A_n = \{a_{ij}\}$, as well as a temporal distribution indicating the duration of each transition D_n. The observations $O = \{o_t\}$ are a list of events detected at arbitrary times from the various vertices of the graph, which indicate the likely presence of one of the N agents at that position at that time. In other words, each observation is identifiably generated by one of the sensors.

The goal of our work is to estimate the parameters describing this semi-Markov process based on a number of assumptions. We assume that the behavior of the agents can be approximated as being homogeneous; *i.e.* the motion of all agents are described by the same A and D. In addition, we must make some assumptions about the distribution of the inter-vertex transition times. Generally, we make the assumption that the delays are normally distributed and bounded within a fixed range. We will show later, however, that we relax this assumption in some situations.

Given the observations O and the vertices V, the problem is to estimate the network connectivity parameters A and D, subsequently referred to as θ.

3 Fundamental Topology Inference Algorithm

The fundamental topology inference algorithm used by our technique infers the connectivity of a sensor network given non-discriminating observations. It assumes knowledge of the number of agents in the environment and additionally, requires some prior information regarding motion patterns in the system. The inference algorithm is based on the statistical technique of Expectation Maximization (EM). It constructs plausible trajectories of agent motions based on current estimates of connectivity parameters (E Step), and then updates the parameters to maximum likelihood estimates based on the sampled trajectories (M Step). In this section, we will briefly outline the methodology behind this technique; full details can be found in [Marinakis and Dudek, 2006].

The algorithm simultaneously converges toward both the correct observation data correspondences and the correct network parameters by iterating over the following two steps:

1. *The E-Step:* which calculates the expected log likelihood of the complete data given the current parameter guess:

$$Q\big(\theta, \theta^{(i-1)}\big) = E\Big[\log p(O, Z|\theta)|O, \theta^{(i-1)}\Big]$$

where O is the vector of binary observations collected by each sensor, and Z represents a hidden variable that determines the data correspondence between the observations and agents moving throughout the system.

2. *The M-Step:* which then updates our current parameter guess with a value that maximizes the expected log likelihood:

$$\theta^{(i)} = \underset{\theta}{\operatorname{argmax}}\, Q\big(\theta, \theta^{(i-1)}\big)$$

The E-Step is calculated by approximating $Q(\theta, \theta^{(i-1)})$ with M samples of an ownership vector $L^{(m)} = \{l_i^m\}$ which uniquely assigns the agent i to the observation o_i in sample m:

$$\theta^{(i)} = \underset{\theta}{\arg\max} \left[\frac{1}{M} \sum_{m=1}^{M} \log p(L^{(m)}, O | \theta) \right]$$

where $L^{(m)}$ is drawn using the previously estimated $\theta^{(i-1)}$ according to a MCMC sampling technique. At every iteration, the M samples of the ownership vector L are used to re-estimate the connectivity parameter θ (the M-Step). The algorithm continues to iterate over the E-Step and the M-Step until subsequent iterations result in very small changes to θ.

Markov Chain Monte Carlo sampling is used to assign each of the observations to one of the agents. Given some guess of the connectivity parameter θ and the current state in the Markov Chain specified by the current observation assignment L, proposals are generated by reassigning a randomly selected observation to a new agent selected uniformly at random. This new data association L' is then accepted or rejected based on an acceptance probability which is determined by the Metropolis algorithm.

The technique uses an inter-vertex delay model that allows for the possibility of agent transitions to and from sources and sinks in the environment. In addition to maintaining a vertex that represents each sensor in the network, the algorithm assumes an additional vertex that represents the greater environment outside the monitored region: a *source/sink node*. A mixture model is employed during the E-Step of the iterative EM process in which potential changes to agent trajectories are evaluated. An inter-vertex delay time is assumed to arise from either a Gaussian distribution or from a uniform distribution of fixed likelihood.

The data assigned to the Gaussian distribution are assumed to be generated by "through-traffic" and are used to during the M-Step to update our belief of the inter-node delay times and transition likelihoods. However, the data fit to the uniform distribution are used only for updating the belief of transitions to and from the source/sink node for the associated vertices.

The portion of data fit to each component of the mixture model is controlled through a a tunable parameter, called Source Sink Log Likelihood (SSLLH), that determines the threshold probability necessary for the delay data to be incorporated into parameter updates. The probability for an inter-vertex delay is first calculated given the current belief of the (Gaussian) delay distribution. If this probability is lower than the SSLLH then this motion is interpreted as a transition made *via* the source/sink node and the transition is not used to update the network parameters associated with the origin and destination vertices. The value assigned to the SSLLH parameter determines how easily the algorithm discards outliers and, hence, provides a compromise between robustness to observational noise and a tendency to discard useful data.

4 Automatic Parameter Selection

Our reasoning system treats the fundamental topology inference algorithm described in the previous section as a 'black box' and attempts to search over its input parameter space to

find reasonable solutions. We construct a heuristic evaluation function that quantitatively assesses a potential solution based on the principle of Occam's Razor. The topology inference algorithm takes the following inputs: the observations O; the assumed number of agents in the environment N; and the $SSLLH$ parameter. The outputs of the algorithm are the network parameters θ and the *ratio* of data R_{data} incorporated into the parameter updates:

$$(\theta, R_{data}) \leftarrow alg(O, N, SSLLH)$$

Different input values result in different environmental assumptions and, hence, produce different outputs.

We have created a metric that attempts to assess the validity of a solution by making the assumption that a good solution both explains the majority of the data and is as *simple* as possible. This principle, known as Occam's razor, states, "if presented with a choice between indifferent alternatives, then one ought to select the simplest one." The concept is a common theme in computer science and underlies a number of approaches in AI; *e.g.* hypothesis selection in decision trees and Bayesian classifiers.

Our simplicity metric incorporates a measure of the simplicity of the transition matrix and the amount of data explained by the solution. We measure the simplicity of a transition matrix by rewarding it in *inverse proportion* to how close it is to a uniform belief of transition probabilities:

$$A_{simp} = \sum_{a_i \in A} (a_i)^{\beta}$$

where β determines the degree of the reward. We measure the utility of a given data use ratio by constructing an adjusted data ratio that attempts to reflect our belief in the solution as a function of the data used. The adjusted data ratio should incorporate the fact that some small portion of discarded data is actually optimal, but that our belief tails off rapidly as the discarded portion grows:

$$R_{data} = \frac{|\text{\# Explained Observations}|}{|\text{\# Total Observations}|}$$

$$R_{adj} = \exp^{-\frac{1}{\tau}(R_{data}-\gamma)^2}$$

where γ and τ describe the shape of the belief curve (figure 2). The final simplicity metric incorporates a weighted combination of A_{simp} and R_{adj}:

$$Q_{simp} = (A_{simp})^{\kappa} * (R_{adj})^{\lambda}$$

where κ and λ reflect the relative weights assigned to the two portions.

With the construction of the simplicity metric Q_{simp}, we have shifted our dependence from specific *a priori* assumptions that must be made on a case to case basis. Instead, we depend on more general assumptions regarding the attributes of a believable solution for this problem domain.

5 Simulation Results

In this section, we attempt to validate our general approach for selecting nearly optimal input parameters for the fundamental topology inference algorithm by using attributes of
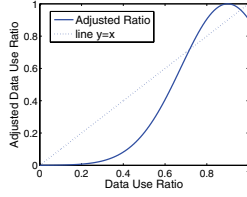
Figure 2: Example relationship between R_{data} and R_{adj} with $\gamma = 0.9$ and $\tau = 0.1$.

β	γ	τ	κ	λ
2	0.9	0.2	2	1

Table 1: Table of values used to shape the simplicity quotient Q_{simp}.

the solution it produces. We select parameters defining the Q_{simp} metric based both on domain knowledge and experimental methods (Table 1).

In order to justify these parameter values and to assess the effectiveness of this approach, we conducted a number of simulations in which we varied the input parameters and looked for a correlation between the performance of the algorithm and the simplicity metric.

Experiments were conducted by simulating agent traffic through an environment represented as a planar graph. The simulation tool takes as input the number of agents in the system and a weighted graph where the edge weights are proportional to mean transit times between the nodes. The output is a list of observations generated by randomly walking the agents through the environment. Two types of noise were modeled in order reflect observations collected from realistic traffic patterns. First, a 'white' noise was generated by removing a percentage of correct observations and replacing them with randomly generated spurious observations. Second, a more systematic noise was generated by taking a percentage of inter-vertex transitions and increasing the Gaussian distributed delay time between them by an additional delay value selected uniformly at random. The range of this additional delay time was selected to be from 0 to 20 times the average normal delay time.

For each experiment, the results were obtained by calculating the squared error between the true A and inferred A' transition matrix:

$$Err_A = \sum_{a_{ij} \in A, a'_{ij} \in A'} (a_{ij} - a'_{ij})^2$$

Input parameters that resulted in good algorithm performance also resulted in solutions that generated high Q_{simp} quotient values (figure 3). When the error in the inferred transition matrix was plotted against the value obtained for the simplicity quotient Q_{simp} for a number of simulations, there was evidence of a definite correspondence (figure 4). The effect appeared robust to moderate levels of observational noise and different sizes of graphs. This result gives support for our adoption of Occam's Razor as a mechanism for selecting input parameters.
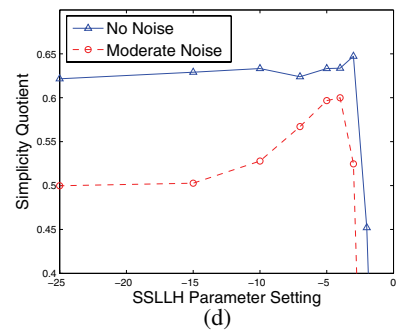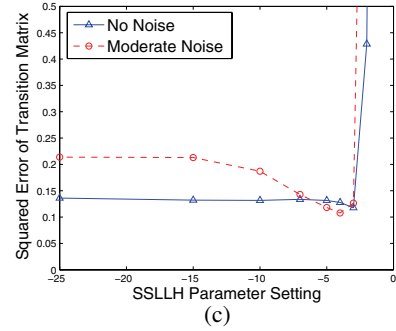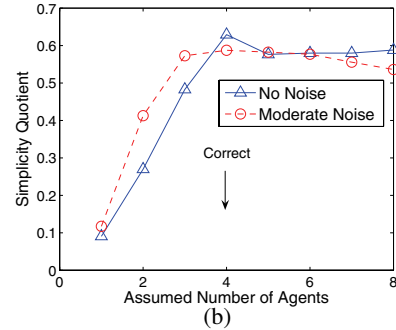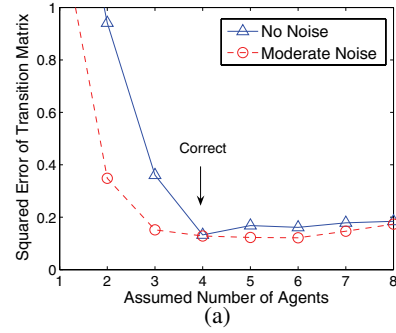

Figure 3: The effect of varying assumed input parameters on performance and the simplicity quotient. Results are averaged over 20 graphs using 4 simulated agents on 12 node, 48 edge graphs with 4000 observations. Simulations labeled 'Moderate Noise' had 5 *per cent* of both white noise and systematic noise added to the observations. For charts a) and b) SSLLH was set to -5.0 while for charts c) and d) the assumed number of agents was set to 4.
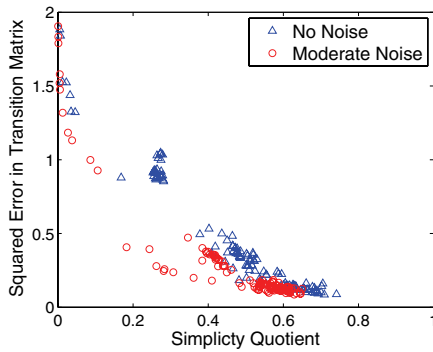
Figure 4: The mean error in the inferred transition matrix elements plotted against Q_{simp} for data obtained from the simulator with 4 true agents from 4 random graphs of 12 nodes, 48 edges and 4000 observations. Input parameters to the algorithm were varied: assumed number of agents from 2 to 7; and SSLLH from -2 to -7. Trials labeled 'Moderate Noise' contained 5 *per cent* of both white noise and systematic noise.

The accuracy of the solution we obtain depends heavily on the assumed number of agents in the environment. The lowest error was consistently observed when the assumed number of agents was set to the correct value, and generally, the closer to the correct value this parameter was set, the better the results. Over-estimating the assumed number of agents had less impact on accuracy than under-estimation.

A correctly tuned SSLLH parameter was also important to the accuracy of the final solution. As the input value for this parameter was increased, there appeared to be a "phase transition" in the accuracy of the results. Past a certain threshold, the error suddenly increased dramatically. Interestingly, the best results for both the inferred mean delay times (not presented here) and transition likelihoods seems to be obtained just before this sudden degradation in performance.

While, the shaping of the Q_{simp} metric is ongoing work, the current parameter values are adequate to demonstrate the correlation between the correctness and simplicity of the inferred transition matrix. In our experimental work, described in the next section, we took advantage of this correlation to select appropriate input parameters since the 'correct' values were unknown.

6 Experimental Results

In order to test our technique under real-world conditions, we setup an experiment using a real sensor network of nine nodes and analyzed the results using our approach. The sensor nodes were built up of photocell-based sensors running on low-powered commercial devices and vision-based sensors running on single board computers. Both types of sensors were programmed to act as simple motion detectors sending event messages to a central server, which logged the origin and time of the activity.

The experiment was conducted in the hallways of one wing of an office building (figure 5) and the data were collected during a six and a half hour period from 10:00am to 4:30 pm on a weekday. In total, approximately 4700 time-stamped events were collected.

To determine appropriate input parameters for our inference algorithm we conducted an exhaustive search over the range of $N = 2, .., 6$ and $SSLLH = -7, .., -3$. We then chose the output values that maximized our Q_{simp} metric. (We used the same shaping parameters for the Q_{simp} metric that were verified through simulations.) The maximizing arguments were: $N = 5$ and $SSLLH = -5$. Therefore, we selected the solution generated by these parameter values as our inferred network.

Except for a few small differences, the network parameters inferred by our topology inference algorithm closely corresponded to the ground truth topology. Figure 6 compares the analytically determined and inferred topological maps. Disregarding reflexive links, the difference between the inferred and 'ground truth' results amounted to a Hamming error of 2. The two significant errors are: an extra edge found between sensors A and B; and a missing one-way edge from sensor D to I. Additionally, the connections to the inferred source/sink node occur primarily for boundary nodes (figure 6(c)) and are therefore consistent with an analytical assessment of the traffic patterns. Since traffic commonly enters and exits the monitored region *via* one of the boundary nodes, the inference algorithm should commonly employ the source/sink node in order bring the agent back into the system.

7 Conclusion

In this paper we presented a method for inferring the topology of a sensor network given non-discriminating observations of activity in the monitored region. Our technique recovers the network connectivity information opportunistically through the exploitation of existing motion. Our work improves considerably on earlier related efforts for topology inference which require prior knowledge regarding motion in the environment. It is worth noting that our final technique recovers a much more complete description of network connectivity than just a topological map of the environment. Additionally, we learn information regarding inter-node delay distributions, inter-node transition likelihoods, and other statistics regarding motion patterns in the system.

References

[Choset and Nagatani, 2001] Howie Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125 – 137, April 2001.

[Dantu and Sukhatme, 2006] Karthik Dantu and Gaurav S. Sukhatme. Rethinking data-fusion based services in sensor networks. In *The Third IEEE Workshop on Embedded Networked Sensors*, 2006.

[Djugash *et al.*, 2006] Joseph Djugash, George Kantor, Sanjiv Singh, and Wei Zhang. Range-only slam for robots operating cooperatively with sensor networks. In *Proc. of the International Conference on Robotics and Automation*, May 2006.
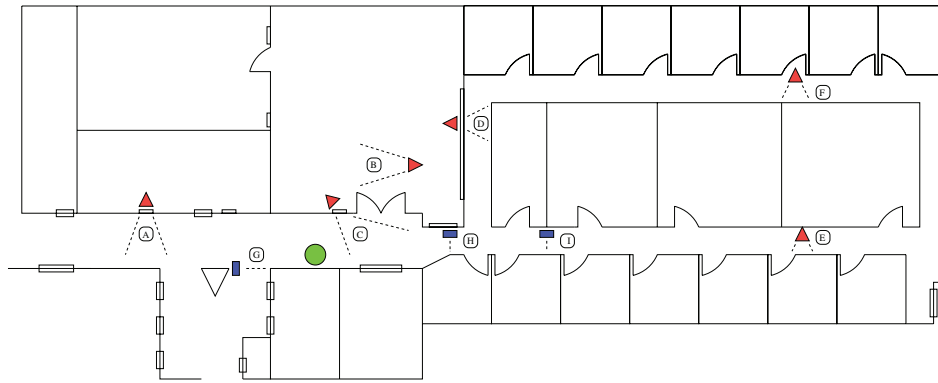
Figure 5: The layout of the nine senor (heterogeneous) network used for the experiment. Labeled triangles represent vision-based sensor positions (A-F) and labeled rectangles represent low-powered photo-based sensors (G-I). The circle represents the location of the central server.
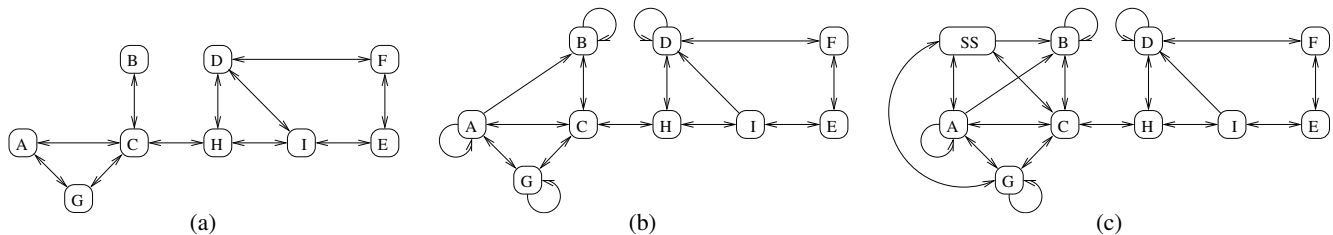

Figure 6: Topological maps of the environment that were: a) analytically determined by a human based on the actual network layout; b) inferred by the algorithm; c) inferred by the algorithm including the source/sink node.

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

[Makris *et al.*, 2004] D. Makris, T.J. Ellis, and J. Black. Bridging the gaps between cameras. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2004*, Washington DC, June 2004.

[Marinakis and Dudek, 2006] D. Marinakis and G. Dudek. A practical algorithm for network topology inference. In *IEEE Intl. Conf. on Robotics and Automation*, Orlando, Florida, May 2006.

[Marinakis *et al.*, 2005] D. Marinakis, G. Dudek, and D. Fleet. Learning sensor network topology through monte carlo expectation maximization. In *IEEE Intl. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005.

[Paskin *et al.*, 2005] Mark A. Paskin, Carlos E. Guestrin, and Jim McFadden. A robust architecture for inference in sensor networks. In *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks 2005 (IPSN-05)*, 2005.

[Pasula *et al.*, 1999] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *IJCAI-99*, Stockholm, 1999.

[Ranganathan and Dellaert, 2005] Ananth Ranganathan and Frank Dellaert. Data driven MCMC for appearance-based topological mapping. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[Rekleitis *et al.*, 2005] I. Rekleitis, D. Meger, and G. Dudek. Simultaneous planning localization, and mapping in a camera sensor network. *Robotics and Autonomous Systems (RAS) Journal, special issue on Planning and Uncertainty in Robotics*, 2005.

[Remolina and Kuipers, 2004] Emilio Remolina and Benjamin Kuipers. Towards a general theory of topological maps. *Artif. Intell.*, 152(1):47–104, 2004.

[Shatkay and Kaelbling, 1997] Hagit Shatkay and Leslie Pack Kaelbling. Learning topological maps with weak local odometric information. In *IJCAI97*, pages 920–929, San Mateo, CA, 1997.

[Songhwai Oh and Sastry, 2006] Michael Manzo Songhwai Oh, Phoebus Chen and Shankar Sastry. Instrumenting wireless sensor networks for real-time surveillance. In *Proc. of the International Conference on Robotics and Automation*, May 2006.

[Wang *et al.*, 2003] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao. Target classification and localization in a habitat monitoring application. In *In Proc. of the IEEE ICASSP*, 2003.