Nonparametric Belief Propagation for Self-Calibration in Sensor Networks

Alexander T. Ihler
MIT / LIDS
77 Massachusetts Ave
Cambridge MA 02139

ihler@mit.edu

John W. Fisher III
MIT / CSAIL
77 Massachusetts Ave
Cambridge MA 02139

fisher@csail.mit.edu

Randolph L. Moses
Ohio State University
2015 Neil Ave
Columbus OH 43210

moses.2@osu.edu

ABSTRACT

Automatic self-calibration of ad-hoc sensor networks is a critical need for their use in military or civilian applications. In general, self-calibration involves the combination of absolute location information (e.g. GPS) with relative calibration information (e.g. time delay or received signal strength between sensors) over regions of the network. Furthermore, it is generally desirable to distribute the computational burden across the network and minimize the amount of inter-sensor communication. We demonstrate that the information used for sensor calibration is fundamentally local with regard to the network topology and use this observation to reformulate the problem within a graphical model framework. We then demonstrate the utility of *nonparametric belief propagation* (NBP), a recent generalization of particle filtering, for both estimating sensor locations and representing location uncertainties. NBP has the advantage that it is easily implemented in a distributed fashion, admits a wide variety of statistical models, and can represent multi-modal uncertainty. We illustrate the performance of NBP on several example networks while comparing to a previously published nonlinear least squares method.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Statistical Computing, Nonparametric Statistics; G.4 [**Mathematical Software**]: Algorithm Design and Analysis; J.2 [**Physical Sciences and Engineering**]: Engineering

General Terms

Algorithms

Keywords

sensor network, calibration, localization, nonparametric, belief propagation, NBP

1. INTRODUCTION

Improvements in sensing technology and wireless communications are rapidly increasing the importance of sensor networks for a wide variety of application domains [12, 8]. Collaborative networks are created by deploying a large number of low-cost, self-powered sensor nodes of varying modalities (e.g. acoustic, seismic, magnetic, imaging, etc). Sensor localization, i.e. obtaining estimates of each sensor's position as well as accurately representing the uncertainty of that estimate, is a critical step for effective application of large sensor networks. Manual calibration of each sensor may be impractical or even impossible, while equipping every sensor with a GPS receiver or equivalent technology may be cost prohibitive. Consequently, methods of *self-calibration* which can exploit relative information (e.g. obtained from received signal strength or time delay between sensors) and a limited amount of global reference information as might be available to a small subset of sensors are desirable. In the wireless sensor network context, self-calibration is further complicated by the need to minimize inter-sensor communication in order to preserve energy resources.

We present a self-calibration method in which each sensor has available noisy distance measurements to neighboring sensors. In the special case that the noise on distance observations is well modeled by a Gaussian distribution, self-calibration may be formulated as a nonlinear least-squares optimization problem. In [15] it was shown that a relative calibration solution which approached the Cramer-Rao bound could be obtained using an iterative, centralized optimization approach.

In contrast, we reformulate the process of self-localization as an inference problem on a graphical model. This allows us to apply nonparametric belief propagation (NBP, [21]), a variant of the popular belief propagation (BP) algorithm [18], to obtain an approximate solution. The NBP approach provides several advantages:

- It exploits the local nature of the problem; a given sensor's estimate of location depends primarily on information about nearby sensors.

- It naturally allows for a distributed estimation procedure.

- It is more general in that it is not restricted to Gaussian measurement models.

- It produces both an estimate of sensor locations and a representation of the location uncertainties.

The last is notable for random sensor deployments where multi-modal uncertainty in sensor locations is a frequent occurrence. Furthermore, estimation of uncertainty (whether multi-modal or not) provides guidance for expending additional resources in order to obtain more refined solutions.

2. SELF-LOCALIZATION OF SENSOR NETWORKS

We restrict our attention to cases in which individual sensors obtain noisy distance measurements of a (usually nearby) subset of the other sensors in the network. This includes, for example, scenarios in which each sensor includes a transceiver and distance is estimated by received signal strength or time delay of arrival between sensor locations. Although this formulation is slightly less general than that presented in [14], it is straightforward to extend our methodology to allow for the inclusion of direction-of-arrival information and/or scenarios in which sources are not co-located with a cooperating sensor.

Specifically, let us assume that we have N sensors scattered in a planar region. Denote the two-dimensional location of sensor t by x_t. Two sensors t and u obtain a noisy measurement d_{tu} of the distance between them with some probability $P_o(x_t, x_u)$:

$$d_{tu} = \|x_t - x_u\| + \nu_{tu} \qquad \nu_{tu} \sim p_\nu \qquad (2.1)$$

We use the binary random variable o_{tu} to indicate whether this observation is available, i.e. $o_{tu} = 1$ if d_{tu} is observed, and $o_{tu} = 0$ otherwise. Finally, each sensor t has a (potentially uninformative) prior distribution, denoted $p_t(x_t)$.

In general, finding the maximum likelihood (ML) sensor locations x_t given a set of observations $\{d_{tu}\}$ is a complex nonlinear optimization problem. If the uncertainties above are Gaussian (i.e. the distributions $p_\nu = \mathrm{N}(0, \sigma_\nu^2)$, $p_t(x_t) = \mathrm{N}(\hat{x}_t, \sigma_x^2)$) and P_o is assumed constant, ML estimation of the x_t's reduces to a nonlinear least-squares optimization [15]. In the case that we observe distance measurements between *all* pairs of sensors (i.e. $P_o(\cdot) \equiv 1$), this also corresponds to a well studied distortion criterion ("STRESS") in multidimensional scaling problems [23]. However, for large-scale sensor networks, it is reasonable to assume that only a subset of pairwise distances will be available, primarily between sensors which are in the same region. One potential model assumes that the probability of detecting nearby sensors falls off exponentially with some power of the distance:

$$P_o(x_t, x_u) = \exp\left(-\frac{\|x_t - x_u\|^\rho}{R_1^\rho}\right) \qquad (2.2)$$

The intuition is that detection probability is directly related to received power. Both quadratic $\rho = 2$ (e.g. [15]) and quartic $\rho = 4$ (for fading channels [19]) power laws have been discussed in the literature.

We also draw a distinction between solving for a *relative* sensor geometry versus estimating the sensor locations with respect to some absolute frame of reference. It was shown in [15] that in some cases these two problems can be equivalent, essentially when the influence of prior information $\prod_t p_t(x_t)$ is weak or nonexistent. Given only the relative measurements $\{d_{tu}\}$, the sensor locations x_t may only be solved up to an unknown rotation, translation, and negation (mirror image) of the entire network. We avoid ambiguities in the relative calibration case by assuming known conditions for three sensors (denoted s_1, s_2, s_3):

1. *Translation*: s_1 has known location (taken to be the origin: $x_1 = [0; 0]$)
2. *Rotation*: s_2 is in a known direction from s_1 ($x_2 = [0; a]$ for some $a > 0$)
3. *Negation*: s_3 has known sign ($x_3 = [b; c]$ for some b, c with $b > 0$).

When our goal is absolute calibration, we assume that the prior distributions $p_t(x_t)$ contain sufficient information to resolve this ambiguity.

A number of methods have already been proposed to estimate sensor locations when only a subset of the pairwise distances are measured. For example, one may approximate each unobserved distance by the length of the shortest path along observed distances between them, then apply classical multidimensional scaling; a similar technique has become popular for low-dimensional embedding problems [22]. Alternatively, iterative least-squares methods have also been observed to yield good performance [15]. Yet another possibility is to minimize rank using heuristics while preserving the fidelity of the observed distances [5].

However, the methods above are typically formulated as centralized, joint optimizations. In many cases, we would prefer a solution whose computation is easily distributed throughout the network. Decentralized estimation is particularly useful when the number of sensors in our network is large but some (possibly small) fraction of them have absolute location information (e.g. from GPS receivers). In such scenarios, collecting and disseminating information across the entire network can become more expensive than the local communications required by a distributed method; see, for example, the hop-count based algorithm of [17].

Perhaps more importantly, the methods above do not provide an estimate of the remaining uncertainty about each sensor location. As we show in the next sections, non-Gaussian uncertainty is a common occurrence in sensor localization problems. In consequence, the Cramer-Rao bound may give an overly optimistic (and thus less useful) characterization of the actual sensor location uncertainty, particularly for multi-modal distributions. Estimating which, if any, sensor positions are unreliable is an important task when parts of the network are under-determined. Furthermore, simulations in Section 4 suggest that under-determined networks of sensors may be surprisingly common.

In this paper we propose an approximate solution making use of a recent sample-based message-passing estimation technique called *nonparametric belief propagation* (NBP). Prior to describing an NBP-based approach to sensor localization, we attempt to characterize some of the uncertainties which occur in self-calibration of sensor networks. We then analyze an idealized version of the calibration problem for randomly deployed sensors showing how frequently a unique solution of sensor locations exists. In Section 5 we re-formulate the self-calibration problem as a graphical model, and present a solution based on the NBP algorithm in Section 5.2. We conclude with several empirical examples demonstrating the ability of NBP to solve difficult distributed localization problems.

3. UNCERTAINTY IN SENSOR LOCATION

The sensor localization problem as described in the previous section involves the optimization of a complex nonlinear likelihood function. However, it is often desirable to also have some measure of confidence in the estimated locations. Even for Gaussian noise ν on measured distance, the nonlinear relationship of inter-sensor distances to sensor positions results in highly non-Gaussian uncertainty.

For sufficiently small networks it is possible to use Monte Carlo techniques such as Gibbs sampling [7] to obtain samples from the joint distribution of the sensor locations. In Figure 1(a), we show an example network with five sensors. Calibration is performed relative to measurements from the three sensors marked by circles. A line is shown connecting each pair of sensors which obtain a distance measurement. Contour plots of the marginal distributions for the two remaining sensors are given in Figure 1(b); these sensors do not have sufficient information to be well-localized, and in particular have highly non-Gaussian, multi-modal uncertainty (suggesting the utility of a nonparametric representation). Although we defer
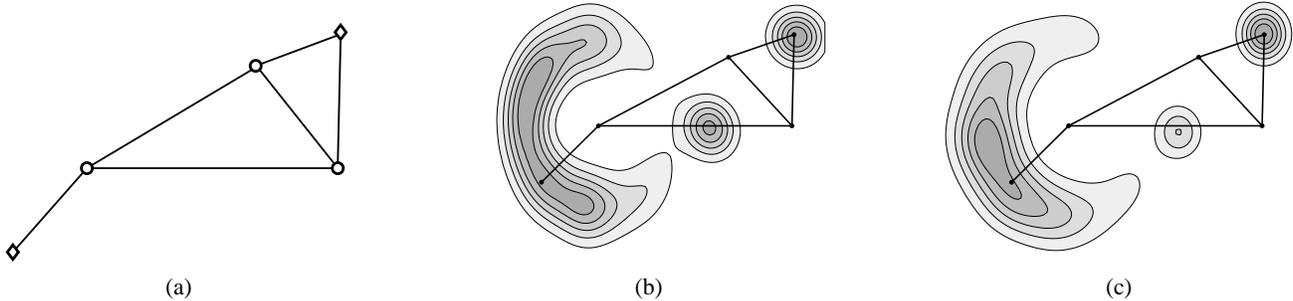
Figure 1: Example sensor network. (a) Sensor locations are indicated by symbols and distance measurements by connecting lines. Calibration is performed relative to the three sensors drawn as circles. **(b)** Marginal uncertainties are shown for the two remaining sensors (one bimodal, the other crescent-shaped) indicating that their estimated positions may not be reliable. **(c)** Estimate of the same marginal distributions using NBP.

the details of NBP to Section 5.2, for purposes of comparison an estimate of the same marginal uncertainties performed using NBP is displayed in Figure 1(c). In the next section, we investigate how often we may expect non-unique solutions for randomly deployed sensor networks. In particular, we find that non-uniqueness of at least part of the network is surprisingly likely unless the number of observed distance measurements is high.

4. UNIQUENESS

To address the problem of how often we may expect a network to have uniquely determined locations, we examine an idealized situation in which a solution is more readily quantified. Let us take N sensors which are distributed at random (in a spatially uniform manner) within a planar circle of radius R_0, and let

$$P_o(x_t, x_u) = \begin{cases} 1 & \text{for } \|x_t - x_u\| \leq R_1 \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

so that sensors t and u obtain a measurement of their distance d_{tu} if and only if $d_{tu} \leq R_1$. We assume that no prior location information is available to any sensor ($\forall t, p_t(x_t)$ is uninformative) and that the uncertainty ν_{tu} present in each measurement d_{tu} is negligibly small. An example of sensors distributed in this manner is given in Figure 2(a).

As discussed, without prior knowledge of the absolute location of sensors in the network this problem can only be solved up to an unknown rotation, translation, and negation. Therefore, we assume a minimal set of known values; in the negligible-noise case (assuming these sensors are mutually co-observing) this is equivalent to assuming known locations for three sensors: $x_1 = [0; 0], x_2 = [0; d_{12}], x_3 = [b; c]$, where

$$b = \sqrt{d_{13}^2 - c^2} \qquad c = \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2 d_{12}}$$

4.1 A sufficient condition for uniqueness

We now derive a sufficient condition for all sensors to be localizable (have a uniquely determined location). Some subtleties arise if any sensors are perfectly co-linear, however, under our model this occurs with probability zero and we proceed to describe conditions which are sufficient for uniqueness with probability one. This same sufficient condition (called a *trilateration graph*) has also recently been investigated by [4].

Let S be the set of nodes which are localizable (with probability one), and let "$\sim$" denote the equivalence relation of observing an

inter-sensor distance. It is straightforward to show that

$$s_A, s_B, s_C \in S \quad \text{and} \quad s_D \sim s_A, \quad s_D \sim s_B, \quad s_D \sim s_C$$
$$\Rightarrow \quad s_D \in S \quad (4.2)$$

We then define S recursively as the minimal set which satisfies (4.2) with $\{s_1, s_2, s_3\} \subset S$; all sensor locations are uniquely determined if $|S| = N$. In practice we may evaluate this condition by initializing $S = \{s_1, s_2, s_3\}$ and iteratively adding to S all nodes with at least three neighbors in S. This condition also has the nice property that it is computable using only the distance measurements.

While this condition is sufficient to uniquely determine all sensor locations, it is not necessary. A useful source of information, not used in (4.2), arises from the *lack* of distance measurement between two sensors. Specifically, the lack of measurement d_{tu} (so that $o_{tu} = 0$) implies $\|x_t - x_u\| > R_1$; thus, to draw a parallel to (4.2), two sensors $s_A \sim s_D, s_B \sim s_D$ and a third $s_C \not\sim s_D$ may localize s_D, or may not (depending on the locations of the sensors involved). An example of each case is shown in Figure 2(b). This yields an alternative sufficient condition to (4.2), which we also investigate.

4.2 Probability of uniqueness

The existence of a unique solution to our idealized problem may now be addressed, in terms of how often a graph generated in the manner described satisfies either of the given sufficient conditions as a function of the parameters N and $\frac{R_1}{R_0}$. We use Monte Carlo trials to investigate the frequency with which the conditions are true. In doing so, we note a number of interesting observations – first, that almost all information useful for localizing sensor s_t is in a local neighborhood around x_t; and second, that in order to have high probability that a random network is uniquely determined, we require a surprisingly high average connectivity (significantly greater than the minimum of 3).

The probability of a random graph having a unique solution as a function of $\frac{R_1}{R_0}$ is shown in Figure 3 for several values of N. The solid lines indicate the probability when *all* sensors contribute information (i.e. we also utilize information between sensors which do not obtain a distance measurement), while the dashed lines illustrate the comparative loss in performance when only information from co-observing sensors is used. Both follow the same trend in N, and appear to agree with the asymptotic behavior predicted by [4].

Notably, most of the information for computing a sensor position is local to the sensor. We quantify the notion of locality by the
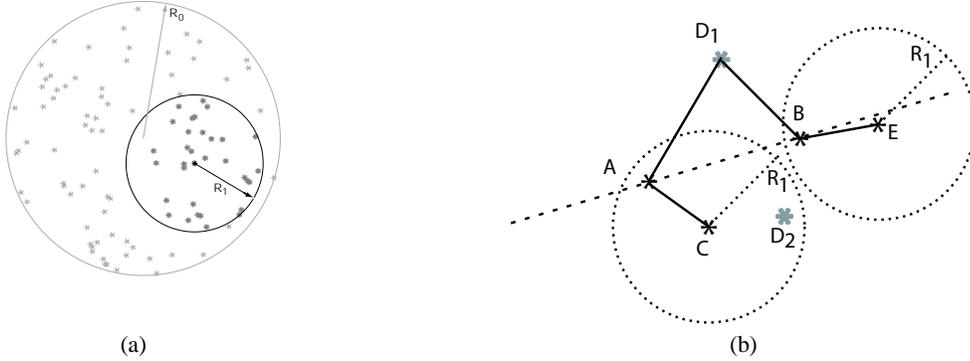
Figure 2: (a) N sensors distributed uniformly within radius R_0, each sensor seeing its neighbors within radius R_1. (b) The two potential locations of sensor D (denoted D_1 and D_2) given distance measurements from sensors A and B is resolved by the *lack* of observation at sensor C, while sensor E is uninformative.
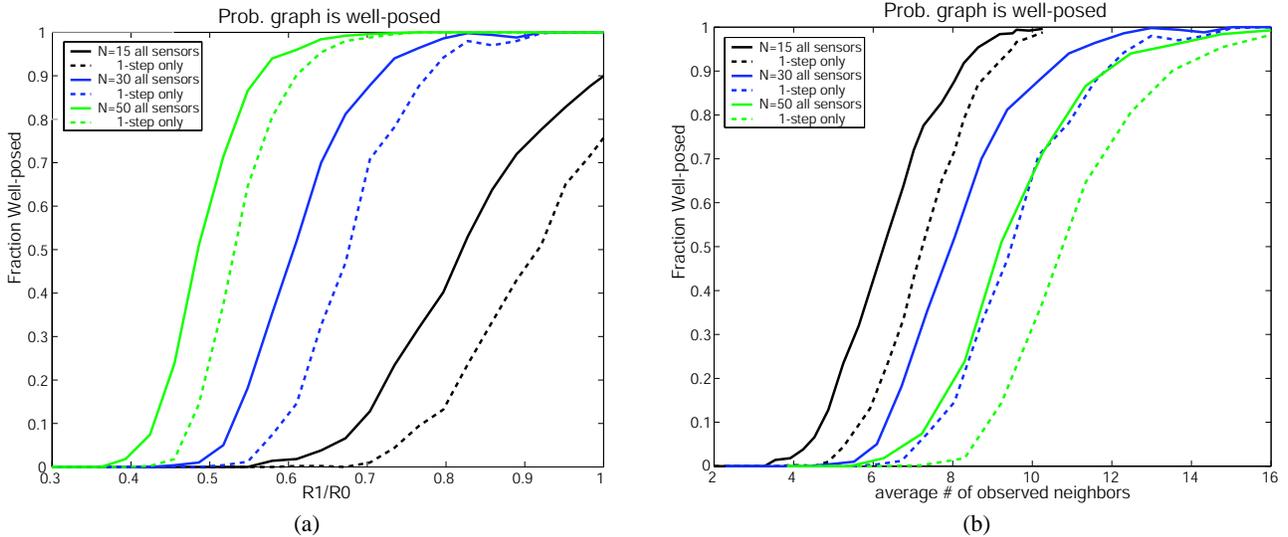

Figure 3: Probability of satisfying the uniqueness condition for various N, as a function of (a) R_1/R_0; (b) Expected number of observed neighbors given R_1/R_0 and N. Solid lines use information from all sensors (equivalent to 2-step neighbors); the dashed lines use only the 1-step neighbor constraints.

following: define the "1-step" neighbors of sensor t as those sensors u which observe a distance d_{tu} from t, the "2-step" neighbors as those for which we observe d_{tv} and d_{vu} for some node v but not d_{tu}, and so forth. We see that a substantial portion (though not all) of the information is already captured by the 1-step neighbors; using more distant sensors reduced the radius required to achieve a given probability of uniqueness by about 10%. Furthermore, in 500 Monte Carlo trials at each setting of N and $\frac{R_1}{R_0}$, every network which was uniquely determined was also uniquely determined using only the 1- and 2-step neighbors. This locality of information is an important part of creating a distributed algorithm for sensor localization.

It is also interesting to note the relationship between how frequently we obtain a unique solution and the average number of neighboring sensors which observe a distance. Clearly a minimal value is three (or two, with the possibility that a sensor which does not observe its distance may assist); but we find that the average is quite high (10+ for even relatively small networks). This is also predicted by theoretical results of [4], and is indicative of the fact

that the minimum connectivity is the driving factor in uniqueness. The implication of this statement is that in practical networks, there may be a number of under-determined sensors, and suggests that having an estimate of the uncertainty associated with each sensor position may be of great importance.

Nonparametric methods provide an appealing means to characterize highly non-Gaussian uncertainties. For example, particle filters [3, 1] are an increasingly popular technique for inference in nonlinear, non-Gaussian time series. For the sensor calibration problem, we turn to a recent generalization of particle filtering, called NBP. This requires that we first describe sensor self-calibration in the framework of graphical models; we then discuss how NBP may be applied to estimate the sensor locations.

5. GRAPHICAL MODELS FOR SELF-CALIBRATION

The results of Section 4 indicate that the information present for self-localization in sensor networks is primarily limited to a small
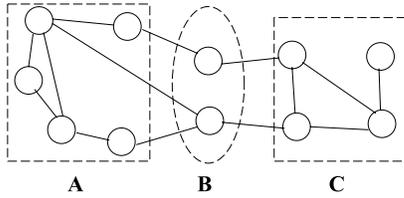
Figure 4: Graph separation and conditional independence of variables: all paths between the sets A and C pass through B, implying $p(x_A, x_C | x_B) = p(x_A | x_B) p(x_C | x_B)$.

locale; to be precise, that a sensor position x_t is (nearly) independent of the rest of the network given the position of nearby sensors. This type of conditional independence relationship is exactly the information exploited by graphical models (also sometimes referred to as Markov random fields) [13]. An undirected graphical model consists of a set of vertices $V = \{v_t\}$ and a collection of edges $e_{tu} \in E$. Two vertices v_t, v_u are *connected* if there exists an edge $e_{tu} \in E$ between them, and a subset $A \subset V$ is *fully connected* if all pairs of vertices $v_t, v_u \in A$ are connected.

Each vertex v_t is also associated with a random variable x_t, and the edges of the graph are used to indicate conditional independence relationships through *graph separation*. Specifically, if every path from a set $A \subset V$ to another $C \subset V$ passes through a set $B \subset V$ (see Figure 4), then the sets of random variables $x_A = \{x_a : v_a \in A\}$ and $x_C = \{x_c : v_c \in C\}$ are independent given $x_B = \{x_b : v_b \in B\}$. This relationship may also be written in terms of the joint distribution: $p(x_A, x_B, x_C) = p(x_B) p(x_A | x_B) p(x_C | x_B)$.

The Hammersley-Clifford theorem [2] quantifies the relationship between a graph and the joint distribution of its random variables x_t, in terms of *potential functions* ψ which are defined solely on the cliques (the fully connected subsets of V), which we denote by Q:

$$p(x_1, \ldots, x_N) = \prod_{\text{cliques } Q} \psi_Q(\{x_i : i \in Q\}) \qquad (5.1)$$

Again taking x_t to be the location of sensor t, we may relate Equation (5.1) to the self-calibration problem by examining the form of the joint distribution between locations $\{x_t\}$ and observations $\{o_{tu}\}, \{d_{tu}\}$. This joint distribution is given by

$$p(x_1, \ldots, x_N, \{o_{tu}\}, \{d_{tu}\}) =$$
$$\prod_{(t,u)} p(o_{tu} | x_t, x_u) \prod_{(t,u):o_{tu}=1} p(d_{tu} | x_t, x_u) \prod_t p_t(x_t) \quad (5.2)$$

since by definition, both the (binary) variable o_{tu} and (if $o_{tu} = 1$) the observed distance d_{tu} depend only on the sensor locations x_t, x_u, and all other sensor information is captured by the prior information $p_t(x_t)$.

From (5.2) we can immediately define potential functions which satisfy (5.1). Notably, this only requires functions defined over single nodes and pairs of nodes. Take

$$\psi_t(x_t) = p_t(x_t) \qquad (5.3)$$

to be the single-node potential at each node v_t, and

$$\psi_{tu}(x_t, x_u) = \begin{cases} P_o(x_t, x_u) \, p_\nu(d_{tu} - \|x_t - x_u\|) & \text{if } o_{tu} = 1 \\ 1 - P_o(x_t, x_u) & \text{otherwise} \end{cases} \qquad (5.4)$$

to be the pairwise potential between nodes v_t and v_u. It then follows that the joint posterior likelihood of the x_t is given by

$$p(x_1, \ldots, x_N | \{o_{tu}\}, \{d_{tu}\}) \propto \prod_t \psi_t(x_t) \prod_{t,u} \psi_{tu}(x_t, x_u) \qquad (5.5)$$

Notice that for non-constant P_o *every* sensor t has some information about the location of each sensor u (i.e. there is *some* information contained in the fact that two sensors do not observe a distance between them). By keeping only a subset of edges deemed sufficiently informative, a *local* approximation, we can perform inference in a distributed manner. Specifically, we consider two possible graphs (stemming from the notions of distance and neighbors in Section 4). Let the "1-step" graph be the graph in which we join only "1-step" neighbors, i.e. $e_{tu} \in E$ if and only if we observe a distance d_{tu}; note that this is exact if P_o is a constant. We create the "2-step" graph by also adding the "2-step" neighbors ($e_{tu} \in E$ if we observe d_{tv} and d_{vu} for some sensor v, but not d_{tu}). The former type of edges we refer to as *observed*, and call the latter *unobserved* edges.

5.1 Belief Propagation

Having defined a graphical model which encapsulates the calibration information present in a sensor network, we now turn to the task of estimating the sensor locations. Inference between variables defined on a graphical model is a problem which has received considerable attention. Although exact inference in general graphs can be NP-hard, approximate inference algorithms such as loopy belief propagation (BP) [18, 16] produce excellent empirical results in many cases. BP can be formulated as an iterative, local message passing algorithm, in which each node v_t computes its "belief" about its associated variable x_t, communicates this belief to and receives messages from its neighbors, then updates its belief and repeats. In the wireless localization context, such algorithms are particularly apropos.

The following discussion focuses on the belief propagation (or sum-product) algorithm, whose purpose is to estimate the posterior marginal distributions $p(x_t | \{o_{ij}\}, \{d_{ij}\})$ of each variable x_t. Note that ideally, we would like the to find the joint MAP (maximum *a posteriori*) configuration of sensor locations. While there exists an algorithm (called the *max-product* or belief revision algorithm [18]) for estimating the MAP configuration of a discrete-valued graphical model, this technique has yet to be generalized to continuous-valued graphical models. However, determining a likely configuration with the maximum likelihood location of each marginal estimated via BP is a common practice [6]. In fact, investigation of the performance of both max- and sum-product algorithms in iterative decoding schemes have shown that the latter may even be preferable in some situations [24]. Thus, we apply BP to estimate each sensor's posterior marginal, and use the maximum of that marginal and associated uncertainty to characterize sensor placements.

One appealing consequence of using a message-passing inference method and assigning each vertex of the graph to a sensor in the network is that computation is naturally distributed. Each node (sensor) performs computations using information communicated from its neighbors, and disseminates the results. This process is repeated until some convergence criterion is met, after which each sensor is left with an estimate of its location. If there is a sufficient amount of information nearby to meet the convergence criterion (for example when a number of sensors scattered through the network have strong prior distributions on their absolute location) this can require relatively few inter-sensor communications.

The computations performed at each iteration of BP are rela-

tively simple. In integral form, each node v_t computes its belief about x_t (a normalized estimate of the posterior likelihood of x_t) at iteration n by taking a product of its local potential ψ_t (if any) with the messages from its neighbors, denoted Γ_t:

$$\hat{p}^n(x_t) = \alpha \psi_t(x_t) \prod_{u \in \Gamma_t} m_{ut}^n(x_t) \tag{5.6}$$

Here α denotes an arbitrary constant of proportionality, usually chosen to normalize $\hat{p}^n$, i.e. $\int \hat{p}^n(x_t) dx_t = 1$. The messages m_{tu} from the node v_t to v_u are computed in a similar fashion:

$$m_{tu}^n(x_u) = \alpha \int_{x_t} \psi_{tu}(x_t, x_u)\psi_t(x_t) \prod_{v \in \Gamma_t \setminus u} m_{vt}^{n-1}(x_t)\, dx_t$$
$$= \alpha \int_{x_t} \psi_{tu}(x_t, x_u) \frac{\hat{p}^{n-1}(x_t)}{m_{ut}^{n-1}(x_t)}\, dx_t \tag{5.7}$$

(with α again chosen to normalize m_{tu}). Each of these equations is easily computed for discrete or Gaussian likelihood functions; however for more general likelihood functions (such as those occurring in sensor localization) exact computation is intractable. We thus approximate the computations using a recent Monte Carlo method called *nonparametric belief propagation* (NBP), discussed in the next section.

5.2 Nonparametric Belief Propagation

Neither discrete nor Gaussian BP is well-suited for the sensor self-localization problem, since even the two-dimensional space in which the x_t reside is too large to accommodate an efficient discretized estimate (for M bin locations per dimension, calculating each message requires $\mathcal{O}(M^4)$ operations), and the presence of nonlinear relationships and potentially highly non-Gaussian uncertainties makes Gaussian BP undesirable as well. Development of a version of BP making use of particle-based representations, called nonparametric belief propagation (NBP) [21], enables the application of BP to inference in sensor networks.

In NBP, each message is represented using either a sample-based density estimate (as a mixture of Gaussians) or as an analytic function. Both types are necessary for the sensor localization problem. Messages along observed edges are represented by samples, while messages along unobserved edges must be represented as analytic functions since often $1 - P_o(x_t, x_u)$ is not normalizable (see e.g. Equation 2.2) and thus is poorly approximated by any finite set of samples. The belief and message update equations (5.6-5.7) are performed using stochastic approximations, in two stages: first, drawing samples from the estimated marginal $\hat{p}(x_t)$, then using these samples to approximate each outgoing message m_{tu}. We discuss each of these steps in turn, and summarize the procedure in Algorithm 1.

Given M samples $\{x_t^{(i)}\}$ from the marginal estimate $\hat{p}_t^n(x_t)$ obtained at iteration n, computing a Gaussian mixture estimate of the outgoing message from t to u is relatively simple. We first consider the case of observed edges. Given a measurement of the distance d_{tu}, each sample $x_t^{(i)}$ is moved in a random direction by d_{tu} plus noise:

$$m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})]$$
$$\text{where} \quad \theta^{(i)} \sim U[0, 2\pi],\ \nu^{(i)} \sim p_\nu \tag{5.8}$$

The samples are then weighted by $\frac{P_o(m_{tu}^{(i)})}{m_{ut}(x_t^{(i)})}$ (see Eq. (5.7)), and a single covariance Σ_{tu} is assigned to all samples. There are a number of possible techniques for choosing the covariance Σ_{tu}; one

simple method is the *rule of thumb* estimate [20], given by computing the (weighted) covariance of the samples (denoted $\text{Covar}[m_{tu}^{(i)}]$) and dividing by $M^{\frac{1}{6}}$.

Here we have used a suggestion of [11], in which a re-weighted marginal distribution $\hat{p}_t^n(x_t)$ is used as an estimate of the product of messages (see Equation (5.7)). In addition to the advantages discussed in [11], this has two desirable qualities. First, it can be computed more efficiently (requiring one product of $|\Gamma_t|$ messages rather than $|\Gamma_t|$ products of $|\Gamma_t| - 1$ messages). Second, this procedure has a hidden communication benefit – all messages from t to its neighbors Γ_t may be communicated *simultaneously*. This is because the message from t to each $u \in \Gamma_t$ is a function of the marginal $\hat{p}_t^{n-1}$, the previous iteration's message from u to t, and the compatibility ψ_{tu} (which depends only on the observed distance between t and u). Since the latter two quantities are also known at node u, t may simply communicate its estimated marginal $\hat{p}_t^n$ to all its neighbors, and allow u to deduce the rest.

As stated previously, messages along unobserved edges (pairs t, u for which d_{tu} is not observed) are represented using an analytic function. Using the probability of detection P_o and samples from the marginal at x_t, an estimate of the outgoing message to u is given by

$$m_{tu}(x_u) = 1 - \sum_i w_t^{(i)} P_o(x_u - x_t^{(i)}) \tag{5.9}$$

Estimation of the marginal $\hat{p}^n = \psi_t \prod m_{ut}$ is potentially more difficult. Since it is the product of several Gaussian mixtures, computing $\hat{p}^n$ exactly is exponential in the number of incoming messages. However, efficient methods of drawing samples from the product of several Gaussian mixture densities has been previously investigated in [9]; in this work we primarily use a technique called *mixture importance sampling*. Denote the set of neighbors of t having *observed* edges to t by Γ_t^o. In order to draw M samples, we create a collection of $k \cdot M$ weighted samples (where $k \geq 1$ is a parameter of the sampling algorithm) by drawing $\frac{kM}{|\Gamma_t^o|}$ samples from each message $m_{ut}, u \in \Gamma_t^o$ and assigning each sample a weight equal to the product of the other messages $\prod_{v \in \Gamma_t \setminus u} m_{vt}$. We then draw M values from this collection with probability proportional to their weight (with replacement), yielding samples drawn from the product of all incoming messages.

Changing the form of the noise distribution p_ν is straightforward so long as sampling remains tractable. One convenient consequence is the ability to incorporate a broad outlier process. Specifically, we can exploit the Gaussian mixture form of NBP's messages by augmenting each message by a single, high-variance Gaussian to approximate an outlier process in the uncertainty about d_{tu}. This representation (similar to a technique proposed by [10]) requires fewer samples to adequately represent the message, and thus is also more computationally efficient.

6. EMPIRICAL CALIBRATION EXAMPLES

We show a number of example sensor networks to demonstrate NBP's utility. All the networks in this section have been created in a manner similar to those of Section 4; N sensors are placed at random with spatially uniform probability, and each sensor observes its distance from another sensor (corrupted by Gaussian noise with variance σ_ν^2) with probability given by (2.2) where $\rho = 2$. We first investigate the *relative* calibration problem, in which the sensors are given no absolute location information. We then show the potential improvement when several sensors which are distributed randomly within the network have absolute location estimates.

Compute outgoing messages: Given M weighted samples $\{w_t^{(i)}, x_t^{(i)}\}$ from $\hat{p}^n(x_t)$, construct an approximation to $m_{tu}^n(x_u)$ for each neighbor $u \in \Gamma_t$:

- If $o_{tu} = 1$ (we observe inter-sensor distance d_{tu}), approximate with a Gaussian mixture:

  - Draw random values for $\theta^{(i)} \sim U[0, 2\pi)$ and $\nu^{(i)} \sim p_\nu$

  - Means: $m_{tu}^{(i)} = x_t^{(i)} + (d_{tu} + \nu^{(i)})[\sin(\theta^{(i)}); \cos(\theta^{(i)})]$

  - Weights: $w_{tu}^{(i)} = \frac{P_o(m_{tu}^{(i)}) \, w_t^{(i)}}{m_{ut}^{n-1}(x_t^{(i)})}$

  - Variance: e.g. $\Sigma_{tu} = M^{-\frac{1}{6}} \cdot \text{Covar}[m_{tu}^{(i)}]$

- Otherwise, use the analytic function:

  - $m_{tu}(x_u) = 1 - \sum_i w_t^{(i)} P_o(x_u - x_t^{(i)})$

Compute local marginals: Given several Gaussian mixture messages $m_{ut}^n = \{m_{ut}^{(i)}, w_{ut}^{(i)}, \Sigma_{ut}\}, u \in \Gamma_t^o$, compute samples from $\hat{p}^{n+1}(x_t)$:

- For each observed neighbor $u \in \Gamma_t^o$,

  - Draw $\frac{kM}{|\Gamma_t^o|}$ samples $\{x_t^{(i)}\}$ from each message m_{ut}^n

  - Weight by $w_{ut}^{(i)} = \prod_{v \in \Gamma_t \setminus u} m_{vt}^n(x_t^{(i)})$

- From these kM locations, re-sample by weight (with replacement) M times

Algorithm 1: Using NBP to compute messages and marginals for sensor localization.

The first example (Figure 5(a)) shows a small graph ($N = 10$). One sensor (the lowest) has significant multi-modal location uncertainty due to the fact that it observes only two distance measurements. The joint MAP configuration is shown in Figure 5(c) while the "1-step" NBP estimate is shown in Figure 5(d). Comparison of the error residuals would indicate that NBP has significantly larger error on the sensor in question. However, this is mitigated by the fact that NBP has a representation of the marginal uncertainty (shown in Figure 5(e)) which accurately captures the bi-modality of the sensor location, and which could be used to determine that the location estimate is questionable. Additionally, exact MAP uses more information than "1-step" NBP. We approximate this information by including some of the unobserved edges ("2-step" NBP). The result is shown in Figure 5(f); the resulting error residuals are now comparable to the exact MAP estimate.

While the previous example illustrates some important details of the NBP approach, our primary interest is in automatic calibration of moderate- to large-scale sensor networks with sparse connectivity. We examine a graph of a network with 100 sensors, shown in Figure 6. For problems of this size, computing the true MAP locations is considerably more difficult. The iterative nonlinear minimization of [15] converges slowly and is highly dependent on initialization. As a benchmark to show the best possible performance, an idealized estimate in which we initialize using the *true* sensor locations is shown in Figure 6(c). In practice, we cannot expect to perform this well; starting from a more realistic value (initialization given by classical MDS [22]) finds the alternate local minimum shown in Figure 6(d). The "1-step" and "2-step" NBP solutions after 12 iterations (approximately 600 messages total) are shown in Figures 6(e) and (f). Errors due to multi-modal uncertainty similar to those discussed previously arise for a small number of sensors in the "1-step" case. Examination of the "2-step" solution shows that the errors compare favorably to the estimate with an idealized initialization. The errors also appear to be less correlated than in the nonlinear least squares approach. Recall that the NBP solution

is attained via a distributed algorithm, while the nonlinear least-squares approach is a centralized algorithm.

Additionally, we expect the performance of NBP to improve and achieve faster convergence (requiring fewer communications), when there is absolute calibration information scattered throughout the network. We simulate this case using the same 100-node network, but now providing 6 additional sensors (chosen at random) strong prior information (in the form of a Gaussian prior $p_t(x_t)$) about their location. The resulting solution, shown in Figures 6(g-h), is significantly better (average error is halved) and requires fewer iterations (8 iterations, or about 500 messages transmitted), as expected. Note, however, that for these experiments we have made no effort to optimize or reduce NBP's communications cost; such optimization is one subject of ongoing research.

7. DISCUSSION

We have empirically demonstrated that multi-modal uncertainty is a common occurrence in the sensor localization problem. Surprisingly, a relatively high degree of connectivity is required in order to obtain a unique solution in the zero-noise case; this is only more difficult with noisy measurements. Additionally, we showed that calibration information is dominated by local relationships, characterized by observed inter-sensor distances ("1-step" information) and a few unobserved distances ("2-step" information).

We proposed a novel approach to the sensor self-calibration problem, applying a graphical model framework and using a nonparametric message-passing algorithm to solve the ensuing inference problem. The methodology has a number of advantages. First, it is easily distributed (exploiting *local* computation and communications between nearby sensors), potentially reducing the amount of communications required. Second, it computes and makes use of estimates of the uncertainty, which may subsequently be used to determine the reliability of each sensor's location estimate. The estimates easily accommodate complex, multi-modal uncertainty. Third, it is straightforward to incorporate additional sources of information, such as a model for the probability of obtaining a distance measurement between sensor pairs. Lastly, in contrast to other methods, it is easily extensible to non-Gaussian noise models, potentially including outlier processes. In empirical simulations, NBP's performance is comparable to the centralized MAP estimate, while additionally representing the inherent uncertainties. Application of NBP to large sensor networks may be particularly advantageous when absolute location information is available to a small number of sensors distributed throughout the network.

There remain many open directions for continued research. For example, BP estimates each sensor's marginal distribution, rather than a joint MAP configuration. An alternative inference algorithm (e.g. max-product) might improve performance if adapted to high-dimensional non-Gaussian problems. Also, alternative graphical model representations may bear investigating; it may be possible to retain fewer edges, or improve BP by clustering nodes (grouping tightly connected variables, performing optimal inference within these groups, and passing messages between groups). Finally, it may be possible to increase computational efficiency by improving how particles are chosen, and to reduce the required communications via a more judicious representation of each message. Given its promising initial performance and many possible avenues of improvement, NBP appears to provide a useful tool for estimating unknown sensor locations in large *ad-hoc* networks.

8. ADDITIONAL AUTHORS

Additional authors: Alan S. Willsky (Massachusetts Institute of Technology, Cambridge MA 02139. Email: willsky@mit.edu).

(a) "1-step" graph   (b) "2-step" graph   (c) MAP estimate

(d) "1-step" NBP estimates   (e) "1-step" NBP marginal   (f) "2-step" NBP estimates
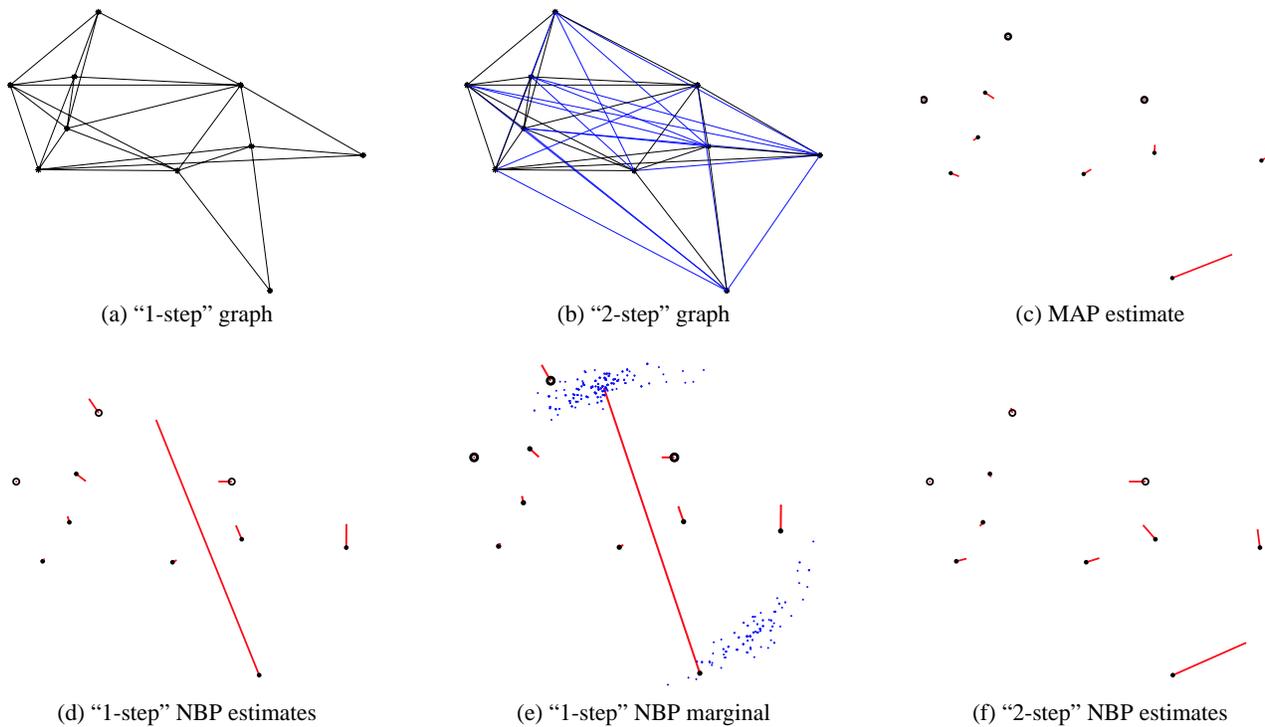
Figure 5: (a) A small (10-sensor) graph and observed pairwise distances; (b) the same network with "2-step" unobserved relationships also shown. Calibration is performed relative to the sensors drawn as open circles. (c) A centralized estimate of the MAP solution shows generally similar errors (lines) to (d), NBP's approximate (marginal maximum) solution. However, NBP's estimate of uncertainty (e) for the poorly-resolved sensor displays a clear bi-modality. Adding "2-step" potentials (f) results in a reduction of the spurious mode and an improved estimate of location.

9. REFERENCES

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[2] P. Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems*, pages 19–32. Oxford University Press, Oxford, 1990.

[3] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[4] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur. Rigidity, complexity, and randomization in network localization. Technical Report TR1257, Yale University, 2003.

[5] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings, American Control Conference*, 2003.

[6] B. Frey, R Koetter, G. Forney, F. Kschischang, R. McEliece, and D. Spielman (Eds.). Special issue on codes and graphs and iterative algorithms. *IEEE Transactions on Information Theory*, 47(2), February 2001.

[7] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[8] H. Gharavi and S. Kumar (Eds.). Special issue on sensor networks and applications. *Proceedings of the IEEE*, 91(8), August 2003.

[9] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky. Efficient multiscale sampling from products of Gaussian mixtures. In *Neural Information Processing Systems 17*, 2003.

[10] M. Isard. PAMPAS: Real–valued graphical models for computer vision. In *IEEE Computer Vision and Pattern Recognition*, 2003.

[11] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Uncertainty in Artificial Intelligence 15*, pages 324–333, 1999.

[12] S. Kumar, F. Zhao, and D. Shepherd (Eds.). Special issue on collaborative information processing. *IEEE Signal Processing Magazine*, 19(2), March 2002.

[13] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[14] R. Moses, D. Krishnamurthy, and R. Patterson. Self-localization for wireless networks. *Eurasip Journal on Applied Signal Processing*, 2003.

[15] R. Moses and R. Patterson. Self-calibration of sensor networks. In *SPIE vol. 4743: Unattended Ground Sensor Technologies and Applications IV*, 2002.

[16] K. Murphy, Y. Weiss, and M. Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence 15*, pages 467–475, July 1999.

[17] D. Niculescu and B. Nath. Ad-hoc positioning system. In *Proceedings, IEEE GlobeCom*, November 2001.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

[19] G. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[20] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.

[21] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *IEEE Computer Vision and Pattern Recognition*, 2003.

[22] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), December 2000.

[23] M. W. Trosset. The formulation and solution of multidimensional scaling problems. Technical Report TR93-55, Rice University, 1993.

[24] Y. Weiss. Belief propagation and revision in networks with loops. Technical Report 1616, MIT AI Lab, 1997.
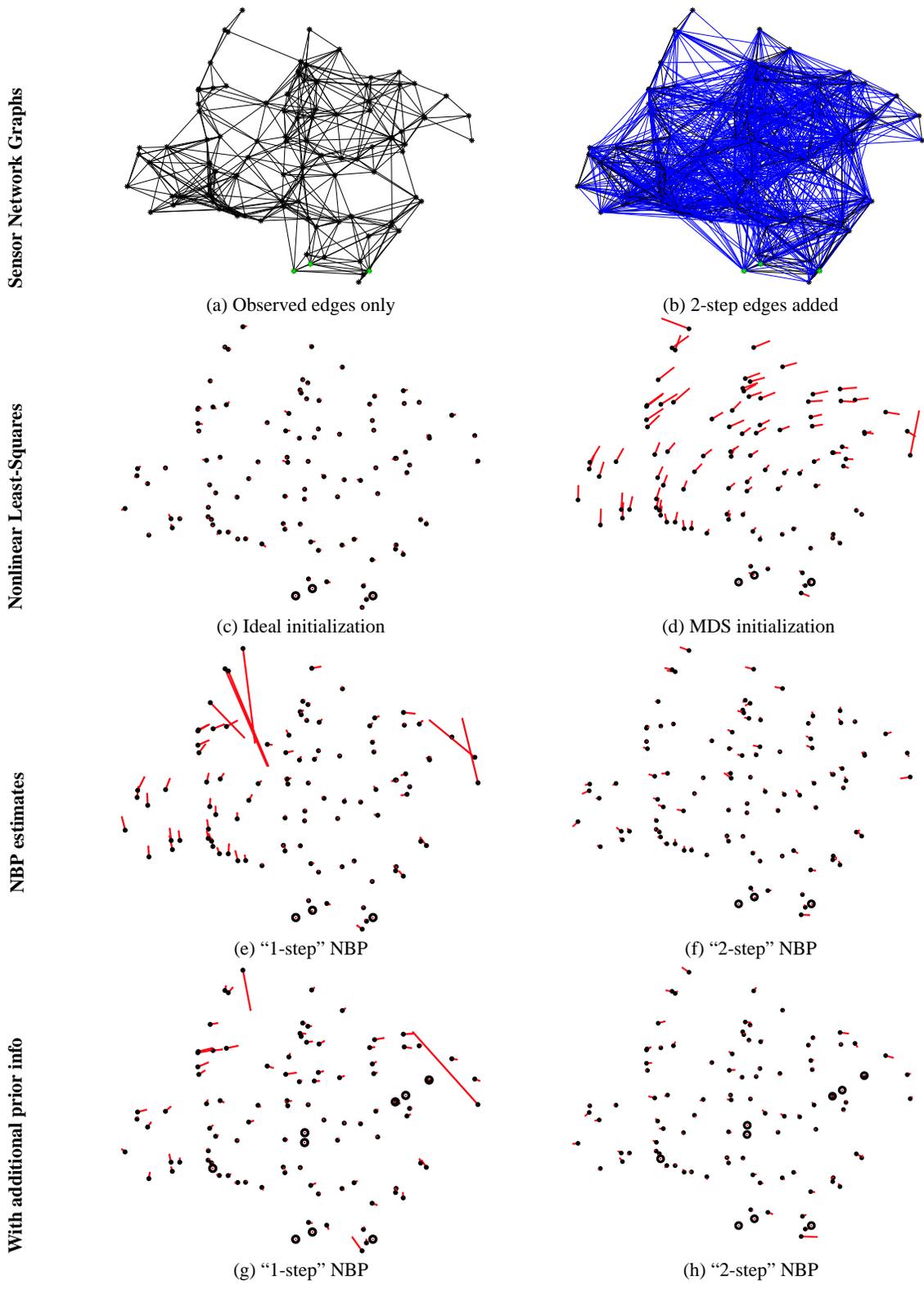
Figure 6: Large (100-node) example sensor network. (a-b) 1- and 2-step edges. Even in a centralized solution we can at best hope for (c) the local minimum closest to the true locations; a more realistic initialization (d) yields higher errors. NBP (e-f),(g-h) provides similar or better estimates, along with uncertainty, in an easily distributed computation. Calibration in (c-f) is performed relative to the three sensors shown as open circles; (g-h) improve performance and convergence rate by adding extra prior information scattered throughout the network (shown as additional circles).