A Distributed Message Passing Algorithm for Sensor Localization

Max Welling and Joseph J. Lim

Bren School of Information and Computer Science
UC Irvine, CA 92697-3425, USA
{welling, josephjl}@ics.uci.edu

Abstract. We propose a fully distributed message passing algorithm based on expectation propagation for the purpose of sensor localization. Sensors perform noisy measurements of their mutual distances and their relative angles. These measurements form the basis for an iterative, local (i.e. distributed) algorithm to compute the sensor's locations including uncertainties for these estimates. This approach offers a distributed, computationally efficient and flexible framework for information fusion in sensor networks.

1 Introduction

Sensor localization and information fusion have become an active area of research in recent years. It has become clear that distributed approaches will become increasingly important to handle the computational challenges in large scale networks. Many such approaches have been proposed in the past (e.g. [5, 3, 4] and many more), but very few can handle uncertainty in a distributed computational setting. The most well-known approach (not presented as a distributed algorithm) is based on the Fisher information [3], but this method can be shown to grossly under-estimate uncertainty for high noise levels. The state-of-the-art in this field is presumably "non-parametric belief propagation" that propagates sample sets over the edges the graph and can as such handle non-gaussian distributions [1]. The price being paid for this flexibility is that the algorithm is computationally intensive, a factor that is not unimportant in applications where energy is on a tight budget.

In this paper we propose a method which trades computation with accuracy. The proposed method maintains and communicates Gaussian estimates (which can be represented by 5 real numbers) instead of sample sets. The price being paid is obviously that it cannot represent multi-modal distributions. The underlying algorithm is expectation propagation (EP), a form of belief propagation that has recently been introduced in the context of Bayesian modeling [2]. We have adapted that framework to the problem of sensor localization and resolved some numerical issues in the process.

2 A MRF model for Sensor Localization

Our general approach is based on constructing a Markov random field model (MRF) for the joint distribution over all sensor locations $\{\mathbf{x}_i,\ i = 1..N\}$ with $\mathbf{x} \in \mathbb{R}^2$. The

general expression for a MRF with pairwise interactions is given by,

$$P(\mathbf{x}_1, .., \mathbf{x}_N) \propto \prod_{ij \in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i \in \mathcal{V}} \psi_i(\mathbf{x}_i) \tag{1}$$

where $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ denotes an interaction potential between sensors i and j and $\psi_i(\mathbf{x}_i)$ a node potential which could for instance encode an absolute measurement of sensor position.

The problem that we need to solve consist of two parts: 1) finding suitable expressions for the potential functions and 2) solving for the marginal distribution $p(\mathbf{x}_i),\ \forall i$. We'll first deal with the first question.

Interactions between sensors consist of two types in this paper: noisy distance measurements between sensors and noisy angle measurements between sensors. The noisy distance measurement is modeled using a gamma distribution for the *squared* distances. The reason we use squared distances is computational. More precisely we use,

$$\psi_{ij}^{\text{dist}}(\mathbf{x}_i, \mathbf{x}_j) \propto r_{ij}^{2(\alpha_{ij}-1)} \exp[-\beta_{ij} r_{ij}^2] \tag{2}$$

where $r_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$ and α_{ij} and β_{ij} are free parameters. The interaction between angles is based on the Von-Mises distribution,

$$\psi_{ij}^{\text{angle}}(\mathbf{x}_i, \mathbf{x}_j) \propto \exp[\kappa \cos(\theta_{ij} - \rho_{ij})] \tag{3}$$

where $\theta_{ij} = \arctan_2(\mathbf{x}_i - \mathbf{x}_j)$ denotes the angle[1] between the sensors, measured relative to absolute north (this requires the sensors to carry a compass) . The constant κ controls the variance of this distribution while ρ_{ij} denotes its mean.

In the following we will assume that the variance of these distributions is determined offline, i.e. the distribution over measurement errors is assumed known. On the other hand, a single measurement will determine the value for the mode of the distribution. Given a particular observation $r_{ij} = d_{ij}, \theta_{ij} = \phi_{ij}$ on the edge ij and given a variance v on squared distance measurements and ν on relative angle measurements we derive the following values for the parameters,

$$\beta_{ij} = (d_{ij}^2 + \sqrt{d_{ij}^4 + 4v})/(2v) \tag{4}$$

$$\alpha_{ij} = 1 + \beta_{ij} d_{ij}^2 \tag{5}$$

$$\rho_{ij} = \phi_{ij} \tag{6}$$

$$\kappa = F(\nu) \tag{7}$$

where $F(\nu)$ is obtained by inverting the relation $\nu = 1 - \frac{I_1(\kappa)^2}{I_0(\kappa)^2}$ with I_j a Bessel function of order j. As mentioned, these parameter settings will place the mode of the distributions at the observed values and sets the variance equal to the given values.

In this paper we will not use any node potentials. This implies that we don't have access to any absolute location information and that the localization can only be successful up to a global translation of the coordinate frame. Note that we do have access

[1] We use the "four quadrant inverse tangent" $\arctan_2(\cdot)$ so that $\mathbb{R}$ is mapped to $[-\pi, \pi]$.

to absolute north and that the absolute orientation can indeed be recovered. In this paper we will assume that the translational degrees of freedom will be determined by a separate procedure. To align results with ground truth we will perform a global transformation which minimizes root mean squared error (RMSE). By adding node potentials we could in fact have avoided this separate phase.

The final MRF is now defined as the product of all the interaction potentials, i.e.

$$P(\mathbf{x}_1, .., \mathbf{x}_N) \propto \prod_{ij \in \mathcal{E}} \psi_{ij}^{\text{dist}}(\mathbf{x}_i, \mathbf{x}_j) \psi_{ij}^{\text{angle}}(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

We will be interested in computing the marginal distribution,

$$P(\mathbf{x}_i) = \int d\mathbf{x}_1 .. d\mathbf{x}_{i-1} d\mathbf{x}_{i+1} .. d\mathbf{x}_N P(\mathbf{x}_1, .., \mathbf{x}_N) \tag{9}$$

This problem is intractable, but we will employ an effective approximation technique to find Gaussian projections of these marginals.

3 Localization by Expectation Propagation

The computation of the marginal distributions $P(\mathbf{x}_i)$ is intractable because $\mathbf{x}$ is continuous and the interaction potentials are nonlinear. Expectation propagation offers an approximate solution. It sends messages over the edges of the communication graph (corresponding to pairs of sensors that can exchange information). Each message is loosely interpreted as an estimate by sensor i of the distribution over locations for sensor j as approximated by a normal distribution. This estimate is based on similar estimates from sensor i'th neighbors but excluding sensor j in that estimate (this is done to avoid evidence coming from j to directly flow back to j). For more details we refer to [2].

Applied to our problem we maintain the following quantities: 1) Gaussian estimates of the marginal distributions over sensor locations $b_i(\mathbf{x}_i)$ and 2) Gaussian messages $M_{ij}(\mathbf{x}_j)$. Messages and marginals relate in a simple way,

$$b_i(\mathbf{x}_i) \propto \prod_{j \in \mathcal{N}_i} M_{ji}(\mathbf{x}_i) \tag{10}$$

The algorithm iteratively updates messages and marginals as we show in the algorithm box below. Note that the computation involves a "projection operator" $\mathcal{P}$, which converts a general distribution in the plane to the closest normal distribution in KL-divergence by matching the first and second moments. The updates also involve a difficult integration over $\mathbf{x}_j$ which should be done numerically in general. The constant $\gamma \in [0, 1)$ is a damping factor and should be increased if the system has difficulty converging. We note that we have used $\gamma = 0.7$ in all our experiments which was sufficient for the algorithm to consistently converge.

In applying the general form of the EP algorithm to sensor localization one runs into some numerical issues. Firstly, the integral has to be performed in a four dimensional space which may be computationally taxing. Fortunately, one can analytically

EP Algorithm

1. Initialize:
1A. $M_{j\to i}(\mathbf{x}_i)$ as random Gaussian distributions
1B. $b_i(\mathbf{x}_i) \propto \prod_{j\in\mathcal{N}_i} M_{j\to i}(\mathbf{x}_i)$
2. Repeat until convergence: update messages $(j \to i)$.
2A. $P_{ij}(\mathbf{x}_i) = \mathcal{P}\int d\mathbf{x}_j\ \psi_{ij}(\mathbf{x}_i,\mathbf{x}_j)\frac{b_i(\mathbf{x}_i)b_j(\mathbf{x}_j)}{M_{i\to j}(\mathbf{x}_j)M_{j\to i}(\mathbf{x}_i)}$
2B. $b_{i,\text{new}}(\mathbf{x}_i) = P_{ij}(\mathbf{x}_i)^{1-\gamma} b_i(\mathbf{x}_i)^{\gamma}$
2C. $M_{j\to i,\text{new}}(\mathbf{x}_i) = \left(\frac{P_{ij}(\mathbf{x}_i)}{b_i(\mathbf{x}_i)}\right)^{1-\gamma} M_{j\to i}(\mathbf{x}_i)$

integrate out two dimensions by observing that all interaction potentials only depend on the relative coordinates $\mathbf{x}_- = \mathbf{x}_i - \mathbf{x}_j$ and not on the coordinates $\mathbf{x}_+ = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$. The algorithm detailed below first transforms to these new coordinates, performs the integral over $\mathbf{x}_+$ analytically while the two- dimensional integral over $\mathbf{x}_-$ is performed numerically (e.g. by importance sampling). Finally, the the results are converted back to the space $\mathbf{x}_i, \mathbf{x}_j$ and the messages are updated.

Numerical instabilities can also occur because we need to perform an integral over a product of local densities and then renormalize. If two of these terms peak at different locations and decay fast, their product is very small (e.g. smaller than machine precision) everywhere. The result is that we cannot reliably compute the normalization constant. We have circumvented this problem by merging the exponential term of the gamma distribution in eqn.2 with the normal distributions for the messages and marginals (see step 2A. of the "EP Algorithm" box). We then sample from the resulting Gaussian distribution and evaluate the remaining terms at the sampled values in order to approximate the integral. The final algorithm is provided in the algorithm box below.

4 Experimental Results

In the following experiments we will study the influence of various parameters on localization accuracy. Accuracy is quantified using the root mean squared error (RMSE),

$$RMSE = \sqrt{\frac{2}{N(N-1)}\sum_{i<j} ||\mathbf{y}_i - \mathbf{y}_j||^2} \tag{11}$$

Before computing the error we first align the estimates by computing the optimal rotation and translation as follows: first translate both the true sensors locations and the predicted sensor locations to the origin: $\mathbf{x}_i \to \mathbf{x}_i - \boldsymbol{\mu}$. Call $\mathbf{x}$ the $2 \times N$ matrix of true centered locations and $\mathbf{y}$ the predicted locations. Next decompose $\mathbf{x}\mathbf{y}^T$ using an SVD: $USV^T = \mathbf{x}\mathbf{y}^T$ and transform $\mathbf{y} \to \mathbf{y}' = \mathbf{y}UV^T$.

The error in the uncertainty ellipse was assessed using the following measure,

$$E_C = \frac{1}{N}\sum_i \frac{|\text{Trace}(C_i^{\text{est}}) - \text{Trace}(C_i^{\text{true}})|}{\text{Trace}(C_i^{\text{true}})} \tag{12}$$

SLEEP Algorithm

1. Input: $\gamma,\ \alpha_{ij}, \beta_{ij}, \kappa, \rho_{ij}\ \forall ij,$ Output: $\Sigma_{i,\text{new}}, \mu_{i,\text{new}}\ \forall i$
2. Initialize: $V_{i\to j}, \boldsymbol{\nu}_{i\to j}\ \forall (i\to j)$

2A. $\Sigma_i^{-1} = \sum_{j\in\mathcal{N}(i)} V_{j\to i}^{-1}\ \forall i$

2B. $\boldsymbol{\mu}_i = \Sigma_i(\sum_{j\in\mathcal{N}(i)} V_{j\to i}^{-1}\boldsymbol{\nu}_{j\to i})\ \forall i$

3. Repeat until convergence:
choose edge ij according to schedule and update.

3A. Remove messages $i\to j$ and $j\to i$:

3Aa. $\Sigma_{i\setminus j}^{-1} = \Sigma_i^{-1} - V_{ji}$

3Ab. $\Sigma_{j\setminus i}^{-1} = \Sigma_j^{-1} - V_{ij}$

3Ac. $\boldsymbol{\mu}_{i\setminus j} = \Sigma_{i\setminus j}(\Sigma_i^{-1}\boldsymbol{\mu}_i - V_{ji}^{-1}\boldsymbol{\nu}_{ji})$

3Ad. $\boldsymbol{\mu}_{j\setminus i} = \Sigma_{j\setminus i}(\Sigma_j^{-1}\boldsymbol{\mu}_j - V_{ij}^{-1}\boldsymbol{\nu}_{ij})$

3B. Transform to relative coordinates:

3Ba. $\Sigma_+ = \Sigma_{i\setminus j} + \Sigma_{j\setminus i}$

3Bb. $\Sigma_- = \Sigma_{i\setminus j} - \Sigma_{j\setminus i}$

3Bc. $\boldsymbol{\mu}_+ = \frac{1}{2}(\boldsymbol{\mu}_{i\setminus j} + \boldsymbol{\mu}_{j\setminus i})$

3Bd. $\boldsymbol{\mu}_- = \boldsymbol{\mu}_{i\setminus j} - \boldsymbol{\mu}_{j\setminus i}$

3C. Absorb Gamma distribution into Gaussian:

3Ca. $C = (I + 2\beta\Sigma_+)^{-1}$

3Cb. $K = C\Sigma_+$

3Cc. $\mathbf{s} = C\boldsymbol{\mu}_-$

3D. Compute the following moments:

3Da. $Z = \int d\mathbf{x}_-\ \exp\left[\kappa\cos\left(\arctan_2(\mathbf{x}_-) - \rho_{ij}\right)\right]\ ||\mathbf{x}_-||^{2\alpha-2}\mathcal{N}_{\mathbf{x}_-}[\mathbf{s}, K]$

3Db. $\mathbb{E}[\mathbf{x}_-] = \int d\mathbf{x}_-\ \exp\left[\kappa\cos\left(\arctan_2(\mathbf{x}_-) - \rho_{ij}\right)\right]\ ||\mathbf{x}_-||^{2\alpha-2}\,\mathbf{x}_-\mathcal{N}_{\mathbf{x}_-}[\mathbf{s}, K]/Z$

3Dc. $\mathbb{E}[\mathbf{x}_-\mathbf{x}_-^T] = \int d\mathbf{x}_-\ \exp\left[\kappa\cos\left(\arctan_2(\mathbf{x}_-) - \rho_{ij}\right)\right]\ ||\mathbf{x}_-||^{2\alpha-2}\,\mathbf{x}_-\mathbf{x}_-^T\mathcal{N}_{\mathbf{x}_-}[\mathbf{s}, K]/Z$

3E. Compute moments for $\mathbf{x}_+$ variables:

3Ea. $\Lambda = \frac{1}{2}\Sigma_-\Sigma_+^{-1}, \quad \boldsymbol{\delta} = \boldsymbol{\mu}_+ - \Lambda\boldsymbol{\mu}_-$

3Eb. $\mathbb{E}[\mathbf{x}_+] = \boldsymbol{\delta} + \Lambda\mathbb{E}[\mathbf{x}_-]$

3Ec. $\mathbb{E}[\mathbf{x}_+\mathbf{x}_+^T] = \frac{1}{4}(\Sigma_+ - \Sigma_-\Sigma_+^{-1}\Sigma_-) + \boldsymbol{\delta}\boldsymbol{\delta}^T + \boldsymbol{\delta}\mathbb{E}[\mathbf{x}_-]^T\Lambda^T + \Lambda\mathbb{E}[\mathbf{x}_-]\boldsymbol{\delta}^T + \Lambda\mathbb{E}[\mathbf{x}_-\mathbf{x}_-^T]\Lambda^T$

3Ed. $\mathbb{E}[\mathbf{x}_+\mathbf{x}_-^T] = \boldsymbol{\delta}\mathbb{E}[\mathbf{x}_-]^T + \Lambda\mathbb{E}[\mathbf{x}_-\mathbf{x}_-^T]$

3Ee. $\mathbb{E}[\mathbf{x}_-\mathbf{x}_+^T] = \mathbb{E}[\mathbf{x}_+\mathbf{x}_-^T]^T$

3F. Transform back to sensor coordinates:

3Fa. $\mathbb{E}[\mathbf{x}_i] = \mathbb{E}[\mathbf{x}_+] + \frac{1}{2}\mathbb{E}[\mathbf{x}_-]$

3Fb. $\mathbb{E}[\mathbf{x}_j] = \mathbb{E}[\mathbf{x}_+] - \frac{1}{2}\mathbb{E}[\mathbf{x}_-]$

3Fc. $\mathbb{E}[\mathbf{x}_i\mathbf{x}_i^T] = \mathbb{E}[\mathbf{x}_+\mathbf{x}_+^T] + \frac{1}{4}\mathbb{E}[\mathbf{x}_-\mathbf{x}_-^T] + \frac{1}{2}(\mathbb{E}(\mathbf{x}_+\mathbf{x}_-^T] + \mathbb{E}(\mathbf{x}_-\mathbf{x}_+^T])$

3Fd. $\mathbb{E}[\mathbf{x}_j\mathbf{x}_j^T] = \mathbb{E}[\mathbf{x}_+\mathbf{x}_+^T] + \frac{1}{4}\mathbb{E}[\mathbf{x}_-\mathbf{x}_-^T] - \frac{1}{2}(\mathbb{E}(\mathbf{x}_+\mathbf{x}_-^T] + \mathbb{E}(\mathbf{x}_-\mathbf{x}_+^T])$

3G. Compute candidate sensor locations and covariances:

3Ga. $\boldsymbol{\mu}_{i,\text{cand}} = \mathbb{E}[\mathbf{x}_i]$

3Gb. $\boldsymbol{\mu}_{j,\text{cand}} = \mathbb{E}[\mathbf{x}_j]$

3Gc. $\Sigma_{i,\text{cand}} = \mathbb{E}[\mathbf{x}_i\mathbf{x}_i^T] - \boldsymbol{\mu}_{i,\text{cand}}\boldsymbol{\mu}_{i,\text{cand}}^T$

3Gd. $\Sigma_{j,\text{cand}} = \mathbb{E}[\mathbf{x}_j\mathbf{x}_j^T] - \boldsymbol{\mu}_{j,\text{cand}}\boldsymbol{\mu}_{j,\text{cand}}^T$

3H. Compute new sensor locations and covariances:

3Ha. $\Sigma_{i,\text{new}}^{-1} = (1-\gamma)\Sigma_{i,\text{cand}}^{-1} + \gamma\Sigma_i^{-1}$

3Hb. $\Sigma_{j,\text{new}}^{-1} = (1-\gamma)\Sigma_{j,\text{cand}}^{-1} + \gamma\Sigma_j^{-1}$

3Hc. $\boldsymbol{\mu}_{i,\text{new}} = \Sigma_{i,\text{new}}((1-\gamma)\Sigma_{i,\text{cand}}^{-1}\boldsymbol{\mu}_{i,\text{cand}} + \gamma\Sigma_i^{-1}\boldsymbol{\mu}_i)$

3Hd. $\boldsymbol{\mu}_{j,\text{new}} = \Sigma_{j,\text{new}}((1-\gamma)\Sigma_{j,\text{cand}}^{-1}\boldsymbol{\mu}_{j,\text{cand}} + \gamma\Sigma_j^{-1}\boldsymbol{\mu}_j)$

3I. Compute new messages:

3Ia. $V_{ji,\text{new}}^{-1} = V_{ji}^{-1} + \Sigma_{i,\text{new}}^{-1} - \Sigma_i^{-1}$

3Ib. $V_{ij,\text{new}}^{-1} = V_{ij}^{-1} + \Sigma_{j,\text{new}}^{-1} - \Sigma_j^{-1}$

3Ic. $\nu_{ji,\text{new}} = V_{ji,\text{new}}(V_{ji}^{-1}\boldsymbol{\nu}_{ji} + \Sigma_{i,\text{new}}^{-1}\boldsymbol{\mu}_{i,\text{new}} - \Sigma_i^{-1}\boldsymbol{\mu}_i)$

3Id. $\nu_{ij,\text{new}} = V_{ij,\text{new}}(V_{ij}^{-1}\boldsymbol{\nu}_{ij} + \Sigma_{j,\text{new}}^{-1}\boldsymbol{\mu}_{j,\text{new}} - \Sigma_j^{-1}\boldsymbol{\mu}_j)$

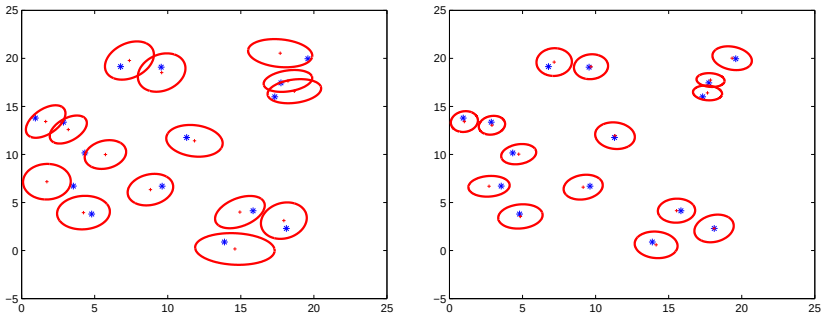

Fig. 1. Left: Localization results with $v = 0.2$ and infinite communicating range but ignoring angle information. Right: same as left but now including angle information with $\kappa = 3$.

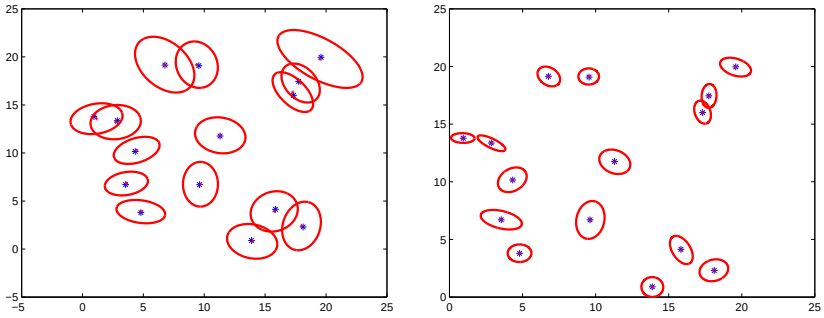

Fig. 2. Left: Ground truth Localization with $v = 0.2$ and infinite communicating range but ignoring angle information. Right: same as left but now including angle information with $\kappa = 3$.

where C_i is the covariance of sensor i. Note that we divide by Trace(C_i^{true}) to divide out the scale of the variance so that smaller variances do not automatically imply smaller error (i.e. we try to quantify the error in the size and shape of the ellipse).

In our experiments we first placed sensors randomly on a square. Then, we perturbed the true distances into noisy distances by sampling them from a log-normal distribution centered at the true distance and with variance v. We used a log-normal distribution because it was shown in [4] to provide a good fit to actual distance measurements. A similar procedure was followed to obtain noisy measurements for the angles which we were sampled from a Von-Mises distribution with a certain value for κ.

We estimated ground truth for the sensor location uncertainty by drawing many samples according to the procedure above and for each of them minimizing the log-

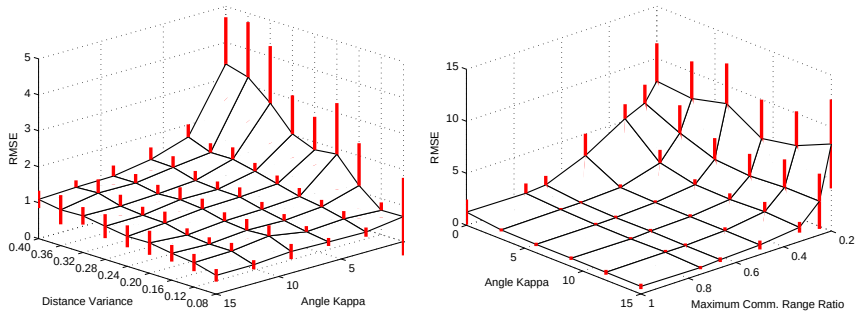

Fig. 3. Left: RMSE as a function of variance in squared distance measurements (v) and κ. All sensors communicate. κ is inversely related to the variance in the angle measurement, i.e. larger κ implies smaller variance. Right: RMSE as a function of communication range and κ ($v = 0.2$). A communication range of 1 is equal to the diagonal spanning the box in which the sensors are placed. In both plot there are $N = 15$ sensors.

probability of the sensor locations using a standard optimization package[2]. Note that this procedure can not be used in a non-simulated environment because we typically only have access to a single sample of a distance measurement. For each marker in a plot we ran the algorithm 30 times and computed average performance measures and their standard deviations shown as error-bars.

In the following examples we vary over the settings of a number variables. The standard settings are given by: $v = 0.2$, $N = 15$ and infinite communication range.

In figure 1 we show two runs of our algorithm, one without angle information (left) and one with angle information (right). It can clearly be observed that the angle information is helpful in disambiguating sensors which are nearby. Figure 2 shows the corresponding ground truth uncertainty regions.

In figure 3 (left) we show RMSE as a function of the variance in distance measurements and as a function of κ which is inversely related to variance in angle measurements. One can observe that even small values of κ around $\kappa = 2$ (corresponding to a std. of approximately $\pi/4$ in angle) can be very helpful in improving the accuracy of the location estimates.

In figure 3 (right) we plot RMSE as a function of the communication range (the fraction of the maximal (diagonal) distance in our 20 by 20 window) and as a function of κ. Again, angle information can help building good maps and is especially helpful when the communication range is small.

Figure 4 (left) shows RMSE for varying κ and N (the number of sensors). As expected, more sensors are helpful and bigger κ is helpful. For values of $\kappa > 3$ we see diminishing returns.

[2] http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize.old/minimize.m

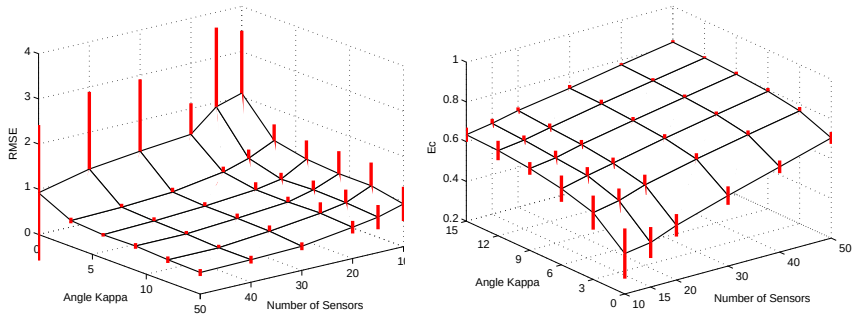

Fig. 4. Left: RMSE as a function of the number of sensors N and κ ($v = 0.2$ and $N = 15$). Right: E_C as a function of the number of sensors N and κ ($v = 0.2$ and $N = 15$).

Finally, we studied the error in the uncertainty itself. Figures 4 (right) and 5 (left and right) show the dependence of E_C w.r.t. κ, v, N and communication range. While decreasing the variance in the distance measurements improves the estimates for uncertainty, the trend for the variance in angle measurements in reversed: less variance makes the estimation harder. We suspect that decreasing variance will improve E_C if the location estimates themselves are bad because in that case estimating uncertainty is essentially impossible. However, if the sensor map is in the right ballpark then there may be another effect which offsets this. Decreasing the variance implies that the sensors are more tightly coupled and that loopy inference by EP deteriorates as evidence over-counting becomes a problem. This assessment is supported by the fact that more sensors (which all communicate) also has a detrimental effect on E_C presumably because it creates more tight cycles in the communication graph.

We have also compared our algorithm to a current state-of-the-art algorithm, namely the "anchor-free localization" (AFL) algorithm discussed in [5]. Since this algorithm was only defined for distance measurements we ignored angle information as well. This is a distributed algorithm based on quadratic penalties for deviations between the noisy observed distances and distances measured in the sensor map. However, unlike SLEEP, AFL has a preprocessing phase where it unfolds the map approximately, avoiding local minima. We have enhanced AFL with a computation of the inverse Fisher information in order to compute uncertainty in localization [3]. In all our experiments we have found that our algorithm works significantly better than AFL both in terms of RMSE as well as E_C. These results will be published elsewhere in full detail and are left out due to space constraints.

5 Discussion

We presented a flexible distributed computational paradigm for sensor localization. Various kinds of information are easily integrated by adding potential functions and adapt-

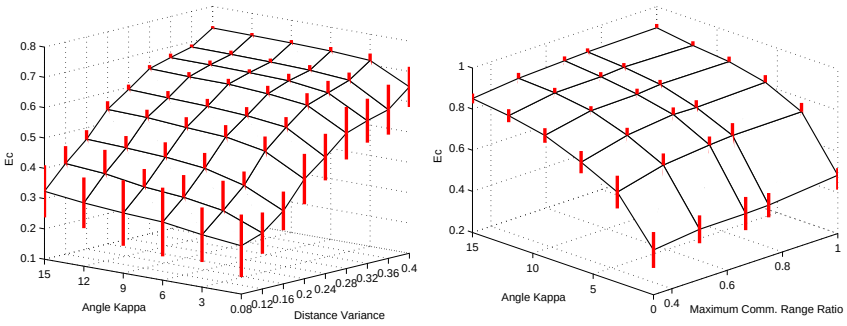

Fig. 5. E_C as a function of variance in squared distance measurements (v) and κ. All sensors communicate. Right: E_C as a function of communication range and κ ($v = 0.2$). In both plot there are $N = 15$ sensors.

ing EP to handle inference. In this paper we have presented an explicit algorithm that incorporates noisy distance and angle information.

The experiments have shown that the algorithm is able to competitively estimate both location and uncertainty in location. We have also shown that approximate inference in graphs with many tight loops begins to break down in terms of estimating uncertainty (but not in terms of the location estimates themselves). This suggests interesting avenues for future research. For instance, one could try to develop distributed implementations for generalized EP algorithms.

There are a few other issues that need further scrutiny. We are currently using importance sampling to do the required numerical integration but this may not be optimal in terms of computational efficiency. We plan to apply quadrature integration to improve upon this. Also, we plan to extend the current model with more sources of information. For instance one may want to simultaneously infer location and, say, the concentration of a chemical in the atmosphere. Extensions to dynamic models where robots move around in their environment are also under investigation.

Acknowledgements

We thanks Tom Minka for important suggestions and Alex Ihler for interesting discussions.

References

1. A.T. Ihler, J. W. Fisher III, R.L. Moses, and A.S. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. In *Information Processing in Sensor Networks*, 2004.
2. T. Minka. Expectation propagation for approximate Bayesian inference. In *UAI*, pages 362–369, 2001.

3. Randolph L. Moses, Dushyanth Krishnamurthy, and Robert Patterson. A self-localization method for wireless sensor networks. *EURASIP Journal on Applied Signal Processing*, 4:348–358, 2003.
4. Rong Peng and Mihail L. Sichitiu. Robust, probabilistic, constraint-based localization for wireless sensor networks. pages 541–550, 2005.
5. Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Anchor-free distributed localization in sensor networks. Technical Report TR-892, MIT LCS, April 2003.