Simultaneous Localization, Calibration, and Tracking in an ad Hoc Sensor Network

Christopher Taylor
taylorc@alum.mit.edu

Ali Rahimi
ali@csail.mit.edu

Jonathan Bachrach
jrb@csail.mit.edu

Howard Shrobe
hes@mit.edu

Anthony Grue
tonyg@mit.edu

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

We introduce Simultaneous Localization and Tracking, called SLAT, the problem of tracking a target in a sensor network while simultaneously localizing and calibrating the nodes of the network. Our proposed solution, LaSLAT, is a Bayesian filter that provides on-line probabilistic estimates of sensor locations and target tracks. It does not require globally accessible beacon signals or accurate ranging between the nodes. Real hardware experiments are presented for 2D and 3D, indoor and outdoor, and ultrasound and audible ranging-hardware-based deployments. Results demonstrate rapid convergence and high positioning accuracy.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Distributed applications*; G.3 [**Probability and Statistics**]: Statistical computing

General Terms

Algorithms

Keywords

Localization, calibration, tracking, wireless sensor networks, statistical machine learning, position estimation

1. INTRODUCTION

Many sensor network applications require that the sensor nodes be calibrated and localized. Automatic localization of nodes is challenging due to the impracticality of precise node placement, the unavailability of GPS (e.g. due to geography or cost), and the fact that range information between nodes is often unreliable. Furthermore, spatially varying environmental factors require that some of the sensor parameters be calibrated in the field. Previous work [1–3] has shown that the information provided by mutual sightings of a target (or *mobile*) is sufficient to localize the nodes; this is an attractive solution for tracking applications, since it requires no additional hardware on the nodes other than that required for the actual tracking task. However, the methods used so far require that the position of the mobile be known at all times.

In this paper, we show how to track the position of an unconstrained target while localizing and calibrating the sensor nodes. We call this problem Simultaneous Localization and Tracking (SLAT) and note that it is related to the Simultaneous Localization and Mapping problem in robotics. Like the more recent work in this area [4–9] we employ a Bayesian filter that uses range measurements to the target to update a joint probability distribution over the positions of the nodes, the trajectory of the target, and the calibration parameters of the network. The Bayesian filtering framework is a non-linear, non-Gaussian generalization of the Kalman Filter. To avoid some of the representational and computational complexity of general Bayesian filtering, we use Laplace's method to approximate this distribution with a Gaussian after incorporating each batch of measurements. Accordingly, we call our algorithm LaSLAT.

LaSLAT has several desirable features, many of which are a direct consequence of using a Bayesian filtering framework. Measurement noise is automatically averaged out as more measurements become available, improving localization and tracking accuracy in the high-traffic areas – precisely the areas of interest for a tracking application. The filtering framework incorporates measurements in small batches, providing online estimates of all locations, calibration parameters, and their uncertainties. This speeds up the convergence of the algorithm and reduces the communication impact on the network. Targets may move arbitrarily through the environment, with no constraint on their trajectory or velocity, and multiple targets may be simultaneously tracked. Ancillary localization information such as position estimates from GPS, beacons, or radio-based ranging is easily incorporated into this framework. Our algorithm is fast and permits a distributed implementation because it operates on

IPSN'06, April 19–21, 2006, Nashville, Tennessee, USA.

sparse inverse covariance matrices rather than dense covariance matrices. When the user does not specify a coordinate system, LaSLAT recovers locations in a coordinate system that is correct up to a translation, rotation, and possible reflection.

We demonstrate these features through a series of indoor/outdoor and 2D/3D experiments using two different ranging techniques. Using ultrasound ranging we are able to accurately localize a dense network of 27 nodes to within two centimeters, with the bulk of the error being removed within only a minute of tracking. The nodes are wireless Crickets [10] capable of measuring their distance to a moving beacon using a combination of ultrasound and radio pulses. In a larger and sparser network using inexpensive audible ranging, we localize nodes to within eight centimeters. In both cases, a measurement bias parameter is accurately calibrated for all nodes. We also present results from an experiment in three dimensional localization and tracking. Our algorithm accurately localizes a network of 40 crickets by observing a mobile for about 80 seconds. This mobile moves in an unconstrained 3d path that includes changes in speed, loops, and twists. Finally, we demonstrate LaSLAT in a 27 m x 32 m outdoor deployment of 40 motes with 4.6 m spacing using inexpensive audible ranging. Despite crude 40 cm ranging resolution and only one measurement event every 3 seconds, we are able to achieve 47 cm average error in only 2 minutes.

2. RELATED WORK

The most common sensor network localization algorithms rely on range or connectivity measurements between pairs of nodes [11–16]. When such measurements are available, these methods can supplement LaSLAT by providing a prior or an initial estimate for the location of the sensors (Section 3.3).

Various authors have used mobiles to localize sensor networks [1–3, 17], but these methods assume the location of the mobile is known. One exception is [2], which builds a constraint structure as measurements become available. Compared to [2], we employ an extensible statistical model that allows more realistic measurement models. Our method is most similar to [17], which used an Extended Kalman Filter (EKF) to track an underwater vehicle while localizing sonar beacons capable of measuring their range to the vehicle. We replace the EKF's linearized measurement model with one based on Laplace's method. This provides faster convergence and greater estimation accuracy. We also calibrate various parameters of the sensor nodes, and show that the computation can distribute over the sensor nodes in a straightforward way.

Our solution to SLAT adopts various important refinements to the original Extended Kalman Filter (EKF) formulation of SLAM [9]. LaSLAT processes measurements in small batches and discards variables that are no longer needed, as demonstrated by McLauchlan [18]. Following [6], LaSLAT operates on inverse covariances of Gaussians rather than on covariances to speed up updates and facilitate distributed computation.

3. LASLAT

As a mobile moves through the network, it periodically emits *events* which allow some of the sensors to measure their distances to the mobile. LaSLAT updates a distribution over the mobiles' position, the sensor locations, and other sensor parameters. The resulting posterior distribution is then propagated forward in time using a dynamics model to make it a suitable prior for incorporating the next batch of measurements. When incorporating each batch of measurements, the posterior distribution is approximated with a Gaussian using Laplace's method [19]. Because the Gaussian is a parametric distribution, the storage required to represent it at each time step is bounded, regardless of the amount of observed data. The Gaussian approximation also simplifies propagation with the dynamics model and the incorporation of the next batch of measurements. To speed up convergence and reduce communication overhead, we process measurements in batches that span several events.

Let $\mathbf{e}^t = \{\mathbf{e}_1^t, \mathbf{e}_2^t, \ldots\}$ denote the collection of mobile positions in the tth batch, where $\mathbf{e}_j^t$ denotes the location of the mobile generating the jth event in the batch.

Let $\mathbf{s} = \{\mathbf{s}_i\}$ denote the set of unknown sensor parameters, where for each sensor i, $\mathbf{s}_i = \begin{bmatrix}\mathbf{s}_i^x & \mathbf{s}_i^\theta\end{bmatrix}$ consists of the sensor's position $\mathbf{s}_i^x$, and $\mathbf{s}_i^\theta$ denotes its other calibration parameters.

For each batch, $\mathbf{e}^t$ and $\mathbf{s}$ are the unknowns that must be estimated. We aggregate these into a single variable $\mathbf{x}^t = \begin{bmatrix}\mathbf{e}^t & \mathbf{s}\end{bmatrix}$ for notational simplicity.

Let $\mathbf{y}^t = \{y_{ij}^t\}$ denote the collection of all range measurements in batch t, with the scalar y_{ij}^t denoting the range measurement between sensor i and the jth event in batch t,

After the measurements $\mathbf{y}^t$ in a batch have been acquired, the goal of the algorithm is to compute a distribution over $\mathbf{x}^t$ given all measurements so far:

$$p(\mathbf{x}^t|\mathbf{y}^1, \ldots, \mathbf{y}^t). \tag{1}$$

LaSLAT provides an efficient way to recursively update this distribution as each measurement batch $\mathbf{y}^t$ becomes available. Because LaSLAT represents this distribution approximately with a parametric Gaussian distribution, the measurements $\mathbf{y}^t$ may be discarded after updating the distribution.

To do this, we take advantage of the independence assumptions made in other Bayesian filters. The posterior distribution can be rewritten in terms of a measurement model and the posterior obtained from incorporating the last batch of measurements:

$$\begin{aligned} p(\mathbf{x}^t|\mathbf{y}^1, \cdots, \mathbf{y}^t) &\propto p(\mathbf{x}^t, \mathbf{y}^t|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1}) && (2)\\ &= p(\mathbf{y}^t|\mathbf{x}^t, \mathbf{y}^1, \cdots, \mathbf{y}^{t-1}) p(\mathbf{x}^t|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1}) \\ &= p(\mathbf{y}^t|\mathbf{x}^t) p(\mathbf{x}^t|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1}). && (3)\end{aligned}$$

The proportionality follows because $p(x|y) \propto p(x, y)$ as a function of x. The first equality follows because $p(x, y) = p(y|x)p(x)$. The final equality follows because the new batch of measurements depends only on sensor and target parameters, and not on old measurements.

The distribution $p(\mathbf{y}^t|\mathbf{x}^t)$ is a measurement model: given a particular configuration of sensors and event locations, it provides a distribution over the measurements that might be observed (Section 3.1).

The distribution $p(\mathbf{x}^t|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1})$ summarizes all information collected about $\mathbf{x}^t$ prior to the current batch of measurements, and can be computed from $p(\mathbf{x}^{t-1}|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1})$, the posterior obtained from incorporating the last batch of measurements (Section 3.2).

In most cases, $p(\mathbf{x}^t|\mathbf{y}^1, \cdots, \mathbf{y}^{t-1})$ is not a conjugate prior

1. Observe a new batch of measurements $\mathbf{y}^t$.
2. Using Newton-Raphson (Section 3.1), compute the curvature at the mode of $p(\mathbf{y}^t|\mathbf{x}^t)p(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^{t-1})$ and use it to construct the approximate posterior $q(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)$.
3. Compute the prediction $p(\mathbf{x}^{t+1}|\mathbf{y}^1,\cdots,\mathbf{y}^t)$ from $q(\mathbf{x}^t|\mathbf{y}^t,\cdots,\mathbf{y}^t)$ (Section 3.2).
4. Using this prediction as the new prior, return to step 1 to process batch $t+1$.

Table 1: One iteration of LaSLAT incorporates batch t and prepares to incorporate batch $t+1$.

[19] for $p(\mathbf{y}^t|\mathbf{x}^t)$, and (3) becomes difficult to compute. We cope with this by computing a Gaussian approximation $q(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)$ to $p(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)$ using Laplace's method (Section 3.1). Table 1 summarizes the steps of LaSLAT.

Other approximate Bayesian filters such as the Extended Kalman Filter (EKF) and particle filters could also be used in place of our Laplacian method. The EKF linearizes the measurement model, forcing it, and the posterior to be linear-Gaussian conjugate pairs. This linearization amounts to performing only one step of the nonlinear optimization procedure that underlies the Laplace approximation. We show in section 4 that this simplification can diminish accuracy and the convergence rate of the overall algorithm. Non-parametric methods such as particle filters represent distributions as weighted instances of the state vector $\mathbf{x}$. Updating these samples requires evaluating the likelihood model for each sample. Since many samples are typically needed to represent high-dimensional distributions, these evaluations can be computationally prohibitive.

3.1 Incorporating Measurements with the Measurement Model

The distribution $p(\mathbf{y}^t|\mathbf{x}^t)$ probabilistically models the process that generates a batch of measurements given a setting of the sensor parameters and mobile positions. In this paper, we assume that each measurement is a corrupted version of the true distance between the mobile and the sensor reporting the measurement:

$$y_{ij}^t = \|\mathbf{s}_i^x - \mathbf{e}_j^t\| + s_i^\theta + \omega_{ij}^t. \tag{4}$$

Here, $\|\cdot\|$ is the Euclidean norm, so the first term is the true distance between the mobile and the sensor. The scalar s_i^θ is a bias parameter to be estimated. It models an unknown offset due to process variations in the ranging hardware of each sensor. The scalar random variable ω_{ij}^t is zero-mean with known variance σ^2, and models nondeterministic noise in the measurement hardware, and other factors not accounted for in this model.

Equation (4) defines $p(y_{ij}^t|\mathbf{s}_i, e_j^t)$ as a univariate Gaussian with mean $\|\mathbf{s}_i^x - \mathbf{e}_j^t\| + s_i^\theta$ and variance σ^2. More sophisticated measurement models may also be used. For example, using a heavy-tailed distribution such as the student-t in place of the Gaussian would provide automatic attenuation of outlying measurements (such as those caused by echoes). Other calibration parameters could also be included in the measurement model.

Since each measurement y_{ij}^t depends only on the sensor that took the measurement, and the location $\mathbf{e}_j^t$ of the mobile that generated the event, the measurement model for a batch is:

$$p(\mathbf{y}^t|\mathbf{x}^t) = \prod_{i,j} p(y_{ij}^t|\mathbf{s}_i, \mathbf{e}_j^t), \tag{5}$$

where the product is over the sensors and the events that they perceived in batch t.

To fit an approximate Gaussian distribution $q(x)$ to a distribution $p(x)$, Laplace's method first finds the mode x^* of $p(x)$, then computes the curvature of the negative log posterior at x^*.

$$\Omega = -\frac{\partial^2}{\partial x^2}\log p(x)\Big|_{x=x^*}. \tag{6}$$

The mean and inverse covariance of $q(x)$ are then set to x^* and Ω respectively. Notice that when p is Gaussian, the resulting approximation q is exactly p. For other distributions, the Gaussian q locally matches the curvature of p about its mode.

We use Newton-Raphson to find the mode $\mathbf{x}^{t*}$, and the curvature (see the Appendix):

$$\begin{aligned}
\mathbf{x}^{t*} &= \arg\max_{\mathbf{x}^t} p(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t) && (7)\\
&= \arg\min_{\mathbf{x}^t} -\log p(\mathbf{y}^t|\mathbf{x}^t)p(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^{t-1}) && (8)\\
&= \arg\min_{\mathbf{s},\mathbf{e}^t} \frac{1}{2}(\mathbf{s}-\mu_0)^T\Omega_0(\mathbf{s}-\mu_0) \\
&\qquad + \frac{1}{2\sigma^2}\sum_{i,j}(\|\mathbf{s}_i^x - \mathbf{e}_j^t\| + s_i^\theta - y_{ij}^t)^2, && (9)
\end{aligned}$$

where μ_0 and Ω_0 are the means and inverse covariance of $p(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^{t-1})$, respectively.

Following Laplace's method, the mean μ of $q(\mathbf{x}^t|\mathbf{y}^t,\cdots,\mathbf{y}^t)$ is set to $\mathbf{x}^{t*}$ and its inverse covariance $\mathbf{\Omega}$ is set to the Hessian at this mode. Representing q using its inverse covariance allows us to avoid computing the inverse of the Hessian after adding each measurement, which significantly simplifies computation and facilitates a distributed implementation of our algorithm.

3.2 Dynamics Model

In this paper, we assume mobiles can move arbitrarily and that sensors do not move over time. To propagate the posterior $q(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)$ forward in time, we replace the estimate of the target's trajectory in batch t with a prediction for the target's path during batch $t+1$. Since mobiles can move arbitrarily, the prediction of its location in the next batch becomes a nearly uniform distribution. Since the sensor parameters do not change over time, their estimates remain the same in the prediction. Therefore, the prediction step is:

$$p(\mathbf{x}^{t+1}|\mathbf{y}^1,\cdots,\mathbf{y}^t) = u(\mathbf{e}^{t+1})q(\mathbf{s}|\mathbf{y}^1,\cdots,\mathbf{y}^t) \tag{10}$$

$$q(\mathbf{s}|\mathbf{y}^1,\cdots,\mathbf{y}^t) = \int_{\mathbf{e}^t} q(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)\, d\mathbf{e}^t. \tag{11}$$

The distribution $q(\mathbf{s}|\mathbf{y}^1,\cdots,\mathbf{y}^t)$ over sensor parameters is obtained by marginalizing over the mobile's trajectory during batch t.

These steps can be carried out numerically by operating on the mean and inverse covariance of $q(\mathbf{x}^t|\mathbf{y}^1,\cdots,\mathbf{y}^t)$.

First, partition these according to $\mathbf{s}$ and $\mathbf{e}^t$:

$$\mu = \begin{bmatrix} \mu_{\mathbf{s}} \\ \mu_{\mathbf{e}} \end{bmatrix} \qquad \Omega = \begin{bmatrix} \Omega_{\mathbf{s}} & \Omega_{\mathbf{se}} \\ \Omega_{\mathbf{es}} & \Omega_{\mathbf{e}} \end{bmatrix}. \tag{12}$$

Marginalizing out $\mathbf{e}^t$ produces a distribution $q(\mathbf{s}|\mathbf{y}^1, \cdots, \mathbf{y}^t)$ with mean $\mu_{\mathbf{s}}$ and with inverse covariance $\Omega_{\mathbf{s}} - \Omega_{\mathbf{se}}\Omega_{\mathbf{e}}^{-1}\Omega_{\mathbf{es}}$.

To augment $q(\mathbf{s}|\mathbf{y}^1, \cdots, \mathbf{y}^t)$ with the uniform distribution $u(\mathbf{e}^{t+1})$, pad the former's means and inverse covariance with zeros, to denote an infinite marginal covariance for $\mathbf{e}^{t+1}$:

$$\mu_0 = \begin{bmatrix} \mu_{\mathbf{s}} \\ \mathbf{0} \end{bmatrix} \qquad \Omega_0 = \begin{bmatrix} \Omega_{\mathbf{s}} - \Omega_{\mathbf{se}}\Omega_{\mathbf{e}}^{-1}\Omega_{\mathbf{es}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{13}$$

These are the desired mean and inverse covariance of $p(\mathbf{x}^{t+1}|\mathbf{y}^1, \cdots, \mathbf{y}^t)$.

If we have an *a priori* guess about $\mathbf{e}^{t+1}$, then the mean can be augmented with a non-zero value instead, and the $\mathbf{0}$ diagonal block of the inverse covariance can be replaced with an inverse covariance that reflects the uncertainty in this prior.

3.3 Prior Information and Initialization

Prior information about the sensor parameters might be available when the sensors are placed in roughly known positions, or when another less accurate source of localization is available. Calibration in the factory might supply additional prior information.

If such information is available it can be supplied as the prior for incorporating the first batch of measurements. We set the covariance of this prior to $\sigma_0\mathbf{I}$, with σ_0 a large scalar to allow measurements to override the positions prescribed by the prior while providing a sensible default when few measurements are available. The mode of this prior is also used as the initial iterate for the Newton-Raphson iterations. To obtain the initial iterate for events, we use the average estimated location of the three sensors with the smallest range measurements to the event.

In our experiments, we utilize the radio connectivity of the sensors to obtain prior localization information. The initialization step described by Priyantha et al. [14] provides rough position estimates to serve as a prior before any measurements are introduced. This prior takes the form $p(\mathbf{s}^x) \propto \exp\left[-\frac{1}{2\sigma_0^2}\sum_i \|\mathbf{s}_i^x - x_i^0\|^2\right]$, where x_i^0 is the position predicted by the initialization step of the ith sensor and σ_0 is a large variance.

Because the sensor nodes are nearly identical, we know *a priori* that the variation between their calibration parameters is small. These small variations are due mainly to environmental effects, so sensors that are close together tend to have similar calibration values. We encode this information in a prior of the form $p(\mathbf{s}^\theta) \propto \exp\left[-\frac{1}{2}\sum_{i\sim j}(s_i^\theta - s_j^\theta)^2\right]$, where the summation is over sensors that are in close proximity to each other.

4. RESULTS

Our first set of experiments use the Cricket ranging system [10]. Sensor Crickets were placed on the floor, and a mobile Cricket was attached to a moving target. The mobile Cricket emitted an event (a radio and ultra-sound pulse) every second. The difference in arrival time of these two pulses to a sensor is proportional to the distance between the sensor and target. Crickets computed ranges from these arrival times. No range measurements between sensor Crickets were collected. The measurements were transmitted to a desktop machine, which processed them in batches using LaSLAT. The ultra-sound sensor on a Cricket occupies a 1 cm x 2 cm area on the circuit board, so it is difficult to estimate the ground truth location of a Cricket beyond those dimensions.

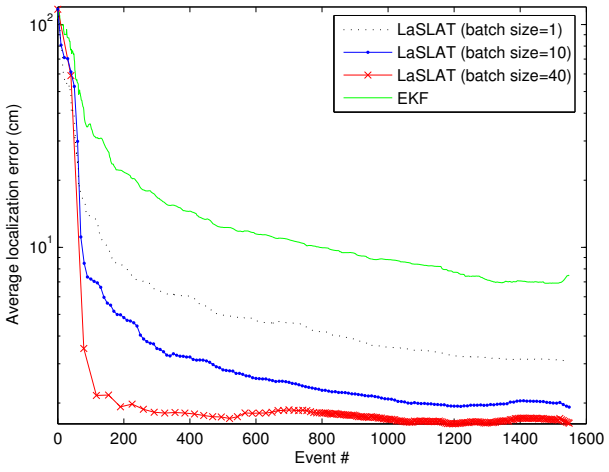

Figure 1: Localization error as a function of the number of events observed for EKF and various batch sizes for LaSLAT. LaSLAT converges more quickly and attains a lower steady-state error than the EKF. Furthermore, larger batch sizes improve the convergence rate and the steady-state error of LaSLAT. Note that in the case of LaSLAT with batch size of 40, the estimates converge after only 200 events, corresponding to 66 seconds of real time.

Our first experiment involved a network with 27 Cricket sensor nodes deployed in a 7 m x 7 m room. In this experiment, we manually pushed a target through the network for about 25 minutes, generating around 1500 events. On average, each event was heard by about 10 sensors.

In this experiment, a measurement bias of about 23 cm was computed for each sensor node. This experiment used batches of 10 measurements and produced a final localization error of 1.9 cm.

To demonstrate the benefits of the Laplace approximation, we compared LaSLAT with the Extended Kalman Filter (EKF). The EKF can be obtained in the limit when LaSLAT performs only one Newton-Raphson iteration for each measurement batch. Figure 1 shows average localization errors as a function of the number of observed events, computed using the EKF and LaSLAT with various batch sizes. The EKF performs best with no batching (batch size = 1). LaSLAT converges faster and also exhibits lower steady-state localization error. As batch sizes are increased, so does the rate of convergence of LaSLAT. Batching also improves the final localization error. LaSLAT, with batch sizes of 1, 10 and 40, produced final localization errors of 3 cm, 1.9 cm, and 1.6 cm respectively. On average, LaSLAT took 3 Newton-Raphson iterations to incorporate each batch. The EKF's final localization error was 7.5 cm, which is outside the error tolerance for the ground truth.

Figure 2 shows localization results on a larger network (49 sensors) deployed over a larger area (10 m x 17 m). With

Figure 2: LaSLAT localization result on a sparser sensor network with 49 nodes in a 10m by 17m environment. Crosses indicate the recovered sensor locations, projected onto the image. The average localization error was 7.5 cm.

about one sensor per three square meters, this network is somewhat less dense than the previous one, which had a sensor per two square meters. As a result, on average only 5 sensors heard each event, and the localization error was about 7.5 cm. The algorithm also determined measurement biases of about 20 cm for all nodes. For all batch sizes, the EKF produced an average localization error of about 80 cm, showing that the improvement due to Laplace's method can be very significant.

4.1 3D SLAT

LaSLAT can localize and track sensors and targets in a three dimensional environment. In our experiments, 40 crickets were placed on the floor and walls of a 4 m x 6 m room furnished with typical lab furniture: tables, chairs, printers, and a refrigerator. The mobile was equipped with additional ultrasound transmitters so that it could broadcast in all directions. This mobile was carried by hand through the room and moved in a natural path, including changes in speed, loops, and twists.

Due to ultrasound echoes and ambient ultrasound from devices in the building, a more sophisticated measurement model was employed. In this measurement model, ranges were modeled as draws from a mixture of ranges emitted from the standard Gaussian model of equation (4) and outliers emitted from a uniform distribution. The mode finding operation (9) was then carried out using Expectation Maximization (EM).

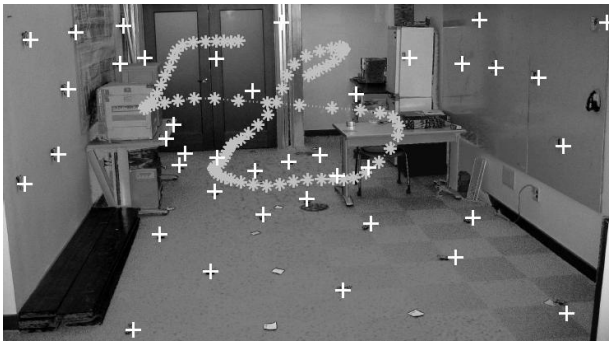

Figure 3: LaSLAT results plotted on a picture on a picture of the network. Plus signs indicate the estimated 3D positions of the sensors. A small portion of the target trajectory (about 80 events) is plotted as asterisks connected by a dotted line. LaSLAT localized sensors to within 7 cm.

In the room shown in Figure 3, LaSLAT localized sensors to within 7 cm while successfully tracking the path of the target in 3d. Much of this error is accounted for by the difficulty of measuring ground truth in this environment.

The best 3d results were obtained using a relatively large batch size of 250 events. Smaller batch sizes caused LaSLAT to converge slowly.

4.2 Extending LaSLAT to Outdoor Acoustic Ranging

It is also possible to use LaSLAT to localize and track sensors and targets using audible ranging. We conducted our experiments using the eXtreme Scaling Motes [21].

The mobile periodically emits a short radio message immediately followed by a loud audible chirp. The difference in time of reception allows sensor nodes to estimate their distance to the mobile. The algorithm for localizing chirps in time operates in the frequency domain to improve accuracy, and was devised by MITRE. This system is less precise than the Cricket's ultrasound ranging system, but is substantially cheaper, since it depends only on a piezo speaker on the mobile and inexpensive microphones on the passive sensors.

In this experimental setup, 42 motes were arranged in a 6 x 7 grid at 4.6 m spacing yielding a 27 m x 32 m area. The motes were placed on 10 cm x 10 cm x 10 cm wooden blocks placed on a field of roughly 15 cm-tall grass. Ground truth was obtained using a tape measure. The beacon was carried by a person, and emitted a ranging event every three seconds.

Figure 4 shows the results plotted over ground truth and Figure 5 shows the convergence plot. The average localization error over three runs at 2 minutes is 47cm and at 3 minutes is 43cm. This shows that the algorithm is capable of running with larger spacing, running with cruder and cheaper ranging hardware, running outdoors, and running in real time.

The ranging resolution of the acoustic ranging system is limited to 40 cm increments. Its accuracy could be further improved with more sophisticated signal processing. This in turn would increase the accuracy of the LaSLAT system.

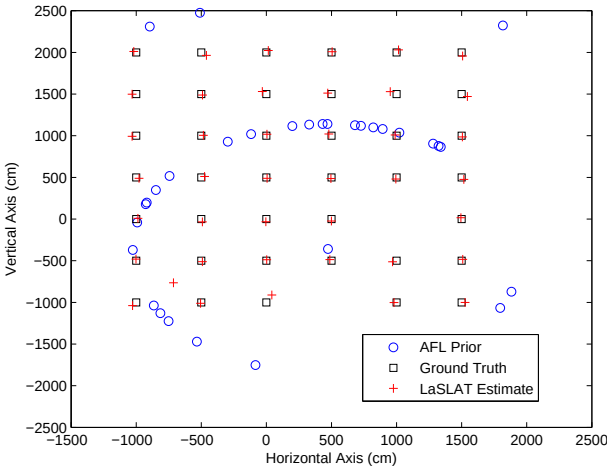

Figure 4: The positioning results for one LaSLAT run on our outdoor deployment involving 40 nodes placed in a 6 x 7 grid with 4.6 m spacing. Here we show the ground truth as squares, the AFL prior as circles, and the final positions as pluses. The final average error for this run is 32 cm. Two sensors in this experiment were defective and failed to report any measurements at all. They are omitted from these results.

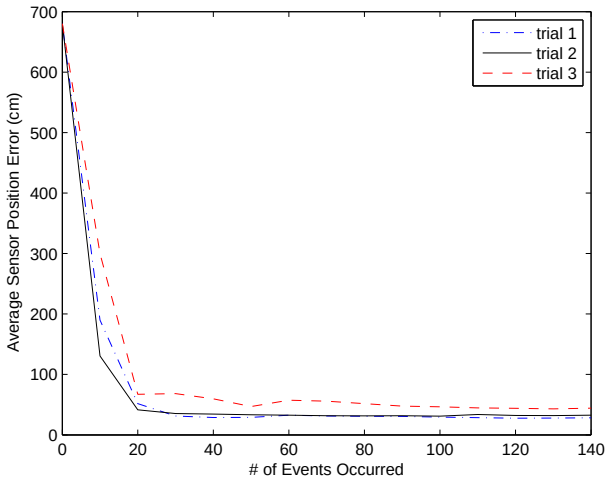

Figure 5: Error over time for three different runs in our outdoor experiment. The batch size is 10 events and the average error at 40 events (with one event every 3 seconds) is 47cm.

5. CENTRALIZED VERSUS DISTRIBUTED

We have implemented both centralized and distributed versions of LaSLAT [23]. In the centralized implementation, the network transports all range observations to a central computer, which performs LaSLAT computations and optionally returns position estimates to the network. In a distributed implementation, the LaSLAT computations are performed in-network using local iterative methods. Only nodes that have witnessed common events need communicate with one another, leading to a natural parallel distributed algorithm. Although this style of in-network processing is the norm in sensor network systems, there are actually interesting trade-offs between the two approach that we briefly outline.

Consider a small network in a controlled environment such as a building. In this case, the central computer can be positioned within a single radio hop of all or nearly all sensors and therefore, each node that senses the target need only transmit one message. In a distributed implementation each node would have to transmit several messages, one for each cycle of the iterative solution methods. In this case, the best performance is obtained by transferring all the measurements to the central computer, reducing the radio bandwidth, memory, and processing requirements on the sensors and decreasing the hardware cost of the network. As the network grows the number of messages transmitted by each node in an distributed iterative method stays roughly constant, however, multiple hops are required to transmit data to the central computer, and the energy and bandwidth cost of centralization increases, particularly for nodes near the central computer. In this case, distributing LaSLAT may save power by keeping computation and communication local.

However, as sensor networks grow even larger there are compelling reasons for building a hierarchical system in which an ensemble of nodes surrounds a more powerful base station with longer-range communications capabilities; these base stations form a second tier of the network, etc. This architecture has the advantage that the individual sensors can be simple, requiring little more than a radio, an ultrasound receiver, and a tone detector. Such a limited sensor is likely to be very cheap and the cost savings can be used to pay for the more powerful second tier nodes.

6. CONCLUSION AND FUTURE WORK

We have demonstrated the benefits of applying Bayesian to the problem of simultaneous localization, tracking, and calibration. We showed experimentally that we can track and localize sensors to within one or two centimeters using ultrasound ranging and to within 43cm in a large scale outdoor deployment with 4.6m spacing and using less accurate acoustic ranging.

The Bayesian framework provides many other advantages that we hope to demonstrate in the future. Different types of measurements such as bearings could be included by suitably modifying the measurement model. By modeling dynamics on the position of sensors, LaSLAT may be able to localize sensors that move over time.

7. ACKNOWLEDGMENTS

We are grateful to Michel Goraczko for providing the cricket hardware and David Moore for providing data sets for some of our early experiments. This research is partly supported by DARPA under contract number F33615-01-C-1896, whose program coordinator, Vijay Raghavan, provided helpful discussions. We are thankful to UIUC for showing us how to use powered piezo speakers. Kenneth Parker, who was Lead System Architect of the NEST Program at the time, designed MITRE's ranging algorithm and implemented the FFT, Brian Flanagan implemented the ranger, and Bryan George assisted with integration and debugging.

8. APPENDIX: FINDING A MODE

The update (9) can be rewritten in the form of non-linear least squares. Letting $f_{ij}(\mathbf{x}) = \|\mathbf{s}_i^x - \mathbf{e}_j^t\| + s_i^\theta$, and defining $\mathbf{f}(\mathbf{x})$ as a column vector consisting of all $f_{ij}(\mathbf{x})$, we can recast (9) as:

$$\arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{f}(\mathbf{x}) - \mathbf{y}^t\|^2 + \frac{1}{2}(\mathbf{x}-\mu_0)^T\Omega_0(\mathbf{x}-\mu_0). \quad (14)$$

Each iteration of Newton-Raphson maps an iterate $\mathbf{x}^{(t)}$ to the next iterate $\mathbf{x}^{(t+1)}$ by approximating (14) by linearizing $\mathbf{f}$ about $\mathbf{x}^{(t)}$, and optimizing over $\mathbf{x}$:

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\left\|\nabla\mathbf{f}^{(t)}\mathbf{x} - \mathbf{b}\right\|^2 + \frac{1}{2}(\mathbf{x}-\mu_0)^T\Omega_0(\mathbf{x}-\mu_0), \quad (15)$$

where $\nabla\mathbf{f}^{(t)}$ is the gradient of $\mathbf{f}$ with respect to $\mathbf{x}$ at $\mathbf{x}^{(t)}$, and $\mathbf{b} = \nabla\mathbf{f}^{(t)}\mathbf{x}^{(t)} - \mathbf{f}\big(\mathbf{x}^{(t)}\big) - \mathbf{y}^t$.

This is a linear least squares problem in terms of $\mathbf{x}$. Its solution can be found by setting the derivative with respect to $\mathbf{x}$ to zero to obtain a linear problem that can be solved by matrix inversion:

$$\big[\Omega_0 + \frac{1}{\sigma^2}\nabla\mathbf{f}^{(t)\top}\nabla\mathbf{f}^{(t)}\big]\mathbf{x} = \Omega_0\mu_0 + \frac{1}{\sigma^2}\nabla\mathbf{f}^{(t)\top}\mathbf{b}. \quad (16)$$

Furthermore, differentiating (15) one more time results in $\Omega = \Omega_0 + \frac{1}{\sigma^2}\nabla\mathbf{f}^{(t)\top}\nabla\mathbf{f}^{(t)}$. Since (15) is an approximation to the negative log posterior (9), Ω serves as an approximation to its Hessian at $\mathbf{x}^{(t)}$.

Because the true distance f_{ij} depends only on sensor i and event location j, each row of $\nabla\mathbf{f}^{(t)}$ is made up of zeros, except at locations corresponding to the ith sensor and the jth event. Thus $\nabla\mathbf{f}^{(t)\top}\nabla\mathbf{f}^{(t)}$ has local connectivity. If Ω_0 has local connectivity, then the updated inverse covariance matrix Ω also has local connectivity. Therefore incorporating a batch of measurements preserves local connectivity.

9. REFERENCES

[1] P. Pathirana, N. Bulusu, S. Jha, and A. Savkin, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, Jul/Aug 2005.

[2] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *Information Processing In Sensor Networks (IPSN)*, 2004.

[3] V. Cevher and J.H. McClellan, "Sensor array calibration via tracking with the extended kalman filter," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, vol. 5, pp. 2817–2820.

[4] J. J. Leonard and P. M. Newman, "Consistent, convergent, and constant-time SLAM," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003.

[5] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun, "Using EM to learn 3D models of indoor environments with mobile robots," in *IEEE International Conference on Machine Learning (ICML)*, 2001.

[6] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," Submitted for journal publication, April 2003.

[7] A.J. Davison and D.W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[8] N. Ayache and O. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Tran. Robot. Automat.*, vol. 5, no. 6, pp. 804–819, 1989.

[9] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Uncertainity in Artificial Intelligence*, 1988.

[10] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang, "Lessons from developing and deploying the cricket indoor location system," Tech. Rep., MIT Computer Science and AI Lab, `http://nms.lcs.mit.edu/projects/cricket/#papers`, 2003.

[11] Shang, Ruml, Zhang, and Fromherz, "Localization from mere connectivity," in *MobiHoc*, 2003.

[12] X. Ji and H. Zha, "Sensor positioning in wireless ad hoc networks using multidimensional scaling," in *Infocom*, 2004.

[13] L. Doherty, L. El Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks," in *Proceedings of Infocom 2001*, April 2001.

[14] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," 2003.

[15] David Moore, John Leonard, Daniela Rus, and Seth Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of ACM Sensys-04*, Nov 2004.

[16] A. Ihler, J. Fisher, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *Information Processing in Sensor Networks (IPSN)*, 2004.

[17] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," in *Proceedings of Autonomous Underwater Vehicles*, 2004.

[18] P. F. McLauchlan, "A batch/recursive algorithm for 3D scene reconstruction," *Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 738–743, 2000.

[19] A. Gelman, J. Carlin, H. Stern, and D. Rubin., *Bayesian Data Analysis*, Chapman & Hall/CRC, 1995.

[20] G. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1989.

[21] Prabal Dutta, "On random event detection in wireless sensor networks," M.S. thesis, Ohio State University, Aug. 2004.

[22] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha, "Resilient localization for sensor networks in outdoor environments," Eng Tech Report 2449, UIUC, 2004.

[23] C. Taylor, "Simultaneous localization and tracking in wireless ad-hoc sensor networks," M.Eng. thesis, MIT CSAIL, May 2005.