2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Decentralized Mapping of Robot-Aided Sensor Networks

Joseph Djugash, Sanjiv Singh, and Benjamin Grocholsky

Abstract— **A key problem in the deployment of sensor networks is that of determining the location of each sensor such that subsequent data gathered can be registered. We would also like the network to provide localization for mobile entities, allowing them to navigate and explore the environment. In this paper, we present a robust decentralized algorithm for mapping the nodes in a sparsely connected sensor network using range-only measurements and odometry from a mobile robot. Our approach utilizes an Extended Kalman Filter (EKF) in polar space allowing us to model the nonlinearities within the range-only measurements using Gaussian distributions. We also extend this unimodal centralized EKF to a multi-modal decentralized framework enabling us to accurately model the ambiguities in range-based position estimation. Each node within the network estimates its position along with its neighbor's position and uses a message-passing algorithm to propagate its belief to its neighbors. Thus, the global network localization problem is solved in pieces, by each node independently estimating its local network, greatly reducing the computation done by each node. We demonstrate the effectiveness of our approach using simulated and real-world experiments with little to no prior information about the node locations.**

I. Introduction

The recent growth in wireless communication and sensing technologies have emphasized the importance and applicability of sensor networks for a wide variety of application domains. Within this growing field, the need for collaborative networks that consist of large numbers of low-cost, low-power, and self-maintaining sensor nodes is rapidly increasing. While these networks and nodes vary in their modalities, they all share the common need of self-localization. Sensor localization, the task of obtaining the estimate 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 to almost all subsequent tasks (such as target tracking, surveillance, control, planning and coordination). Manually surveying the node locations might not always be practical or even possible. Additionally, the use of GPS or other equivalent technologies is restrictive to outdoor environments. Consequently, methods of self-localization that can exploit relative position information between the nodes and estimate the location and uncertainty of the network, with respect to a global or local reference frame, are desirable. However, the in context of WSNs, the process of localization is further complicated by the need to minimize the inter-node communication and the computation performed at each node to minimize power consumption.

Here we present a localization algorithm in which each node has available noisy distance measurements to several of its neighboring nodes. Our goal is to recover the pose of the nodes given inter-node range measurements. However, in many real world scenarios only sparse measurement connectivity is available. In such scenarios, achieving an unique globally (or even locally) accurate localization solution is impossible. We can improve network localization with the addition of a few mobile nodes that can maintain an approximate estimate of their motion via odometry. This information can be coupled with the existing inter-node distances to produce a solution to an otherwise under-constrained network.

In this paper we present a localization method that incorporates information from mobile and stationary nodes. Our method requires little or no prior information about the location of any of the nodes in the network. In addition, we present extensions that allow this problem to be solved in a decentralized fashion – each node can create a map of the other nodes with minimal communication to its neighbors. The mapping problem is then solved in pieces, by each node, independently, though the use of a message passing algorithm that propagates every node's local belief to its neighbors. Additionally, we present a control scheme, which can be used to control mobile nodes such that the network uncertainty estimate is minimized. In our system these mobile nodes are mobile robots with odometry, thus in the context of this paper the terms "mobile node" and "robot" will be used interchangeably. Our approach is able to efficiently estimate the locations of the nodes even in a sparsely connected network where other existing methods of self-location fail. Furthermore, while our approach is designed for the task of mapping the node locations, it can be easily applied to solve the simpler target tracking/localization problem given the locations of a few stationary nodes.

II. Related Work

In sensor networks the problem of mapping the locations of all the nodes in the network is also known as self-calibration or self-localization. Most sensor networks are capable of measuring relative bearing, range or in some cases both range and bearing between nodes within the environment. Of particular interest to us are those that use range to localize the network. For instance, the RADAR system, developed by Bahl and Padmanabhan, utilizes signal strength of packets in the commonly available 802.11b wireless networks for localization of network devices [1]. However, signal strength measurements are often erratic and can be affected by slight changes in the environment. Alternately, the Cricket system uses fixed ultrasound emitters and embedded

This work is supported by the National Science Foundation under Grant No. IIS-0426945

J. Djugash, S. Singh, and B. Grocholsky are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {josephad, ssingh, grocholsky}@ri.cmu.edu

receivers in the target object to localize the target [2], [3]. Our system utilizes the Parrot sensors nodes, [4]. These sensors use radio frequency (RF) signals to communicate data and ultrasonic measurements to range between each other.

Some of the early work in localizing a sensor network with range-only information relied on solving a least-squares optimization problem. Methods such as Multi-dimensional Scaling (MDS) provide a good solution if the network is fully connected [5]. For a less connected network with sufficient connections to provide "rigidity" to the network, it is still possible to determine the map of the network. Moore et al. introduced the idea of the *robust quadrilaterals* as a way to avoid ambiguities in the solution [6]. In practice, however, rigidity is not easy to achieve and a high degree of connectivity between nodes of the network. However, we present a probabilistic approach that can model the uncertainty of the estimate and provide solutions even when network rigidity is not available.

Recently Ihler et al. present an approach to the sensor self-calibration problem with a non-parametric message-passing algorithm in a graphical model framework to solve the ensuing inference problem [7]. While their approach lends itself to a distributed implementation and maintains an estimate of the uncertainty, it represents the state, (x, y), with samples from the true distribution, which could lead to divergence in the absence of sufficient samples. Their approach assumes a negative information model to reduce the uncertainty and improve estimation. However, in most real-world applications, the lack of a measurement could be due to a variety of reasons, making it difficult to accurately model negative information. In contrast, our approach scales well to large uncertainty distributions without the need to use negative information.

While most research in sensor networks has focused on static nodes (network localization), work in SLAM has focused on the incorporation of motion from mobile robots into the estimation of static and mobile nodes. In [8], T.L. Song examined the observability of the system in the presence of motion and limited connectivity, where it was revealed that a nonzero jerk/acceleration motion is necessary for accurate localization with range-only sensors. Previously we have presented a method based on an EKF that jointly estimates location of the static and mobile nodes but this method is prone to the problems of linearization and multi-modality [9]. This means that poor initialization, large measurement error and sparse data all hurt performance. Furthermore, these methods scale poorly to larger networks due to their centralized processing formulation. In this article we present a decentralized formulation of our approach that scales well to large networks.

Stump et al., in [10], present a set-valued approach to estimation that overcomes limitations due to nonlinearities. While their method provides an accurate estimate of the true distribution by processing measurements in a high dimensional space, the complexity involved with projecting the filter's state into the xy-space. This makes it difficult to incorporate even a simple motion model into the filter. Another parametric approach that also works in a high dimensional space, called the ROP parametrization (or Polar parametrization), is presented by Funiak et al., in [11]. The authors model their state using polar coordinates, making the back projection into the Cartesian xy-space a much simpler task. Our approach presented in this paper borrows from this work and extends it to better deal with the non-linearities and multi-modalities encountered within the range-only measurement domain. In addition, we present a decentralized implementation that reduces the computational and memory requirements placed on each individual node in the network.

III. Polar Parameterization in an EKF

We model the network localization problem as a linear dynamical system. At each time step, t, the state of node i is represented by $X_{i,t} = [c_i^x, c_i^y, r_i, \theta_i]^T$. Each node's estimate is represented in a polar coordinates (similar to [11]), where (c_i^x, c_i^y) are the center of the polar coordinate frame and (r_i, θ_i) are the corresponding range and angle values. The use of this parameterization derives motivation from the polar coordinate system, where annuli, crescents and other ring-like shapes can be easily modeled. In addition four variable polar parameterization, for each mobile node (robot) within the system, an addition term that represents the current heading of the node, ϕ_i, is maintained. Thus, the complete state vector at time t is represented as:

$$X_t = [X_{1,t}, \phi_1, ..., X_{M,t}, \phi_M, X_{M+1,t}, X_{M+2,t}, ..., X_{N,t}]^T.$$

where M is the number of mobile nodes and N is the total number of nodes. At each time step, we get some set of motion and range observations, u_t and z_t respectively. The belief state at time t is defined as $p(X_t|z_{1:t}, u_{1:t})$. Our filtering algorithm iteratively computes the belief state at time $t+1$ using the previous belief state at time t. Specifically, in our implementation the belief state is represented by a mean vector μ_t and a covariance matrix Σ_t, and it is computed using an Extended Kalman Filter (EKF).

A. Motion Model

Motion of a given node can be modeled as Brownian motion ([11]) or when odometry is available, the dynamics of the mobile node can be described as:

$$X_{i,t} = X_{i,t-1} + \begin{bmatrix} \Delta D_{i,t}\cos(\phi_{i,t-1}) \\ \Delta D_{i,t}\sin(\phi_{i,t-1}) \\ 0 \\ 0 \end{bmatrix} + \nu_t. \qquad (1)$$

$$\phi_{i,t} = \phi_{i,t-1} + \Delta T_{i,t} + \nu_t^{\phi}.$$

where ν_t and ν_t^{ϕ} are zero-mean Gaussian noise vectors, $\Delta D_{i,t}$ is the odometric distance traveled, and $\Delta T_{i,t}$ is the orientation change. For every new control input vector, $u_{i,t} = [\Delta D_{i,t}, \Delta T_{i,t}]^T$, that is received, the belief state can be updated by computing:

$$p(X_{t+1}|z_{1:t}, u_{1:t+1}) = \int p(X_t|z_{1:t}, u_{1:t}) \cdot p(X_{t+1}|X_t, u_{t+1})dX_t. \qquad (2)$$

$$p(X_{t+1}|X_t, u_t) = \prod_{i=1}^{M} p(X_{i,t+1}|X_{i,t}, u_{i,t}). \qquad (3)$$

In Eq.2, the first term of the r.h.s is the posterior estimate from the previous iteration and the second term is the motion model for any mobile nodes, which can be acquired from Eq.3. Assuming that the motion of each mobile node is independent of the other nodes' motion, we can decompose the likelihood as depicted in Eq.3. In implementation, the motion observations are treated as control inputs and standard EKF prediction is performed.

B. Measurement Model

When two nodes, i and j, are within a given range and sensor F.O.V. to each other, a range observation is generated which is represented by, $z_t^{i,j}$. This observation depends on the position of the two nodes i and j:

$$z_t^{i,j} = \widehat{z}_t^{i,j}(X_{i,t}, X_{j,t}) + \delta.$$

$$\widehat{z}_t^{i,j} = \sqrt{(m_{i,t}^x - m_{j,t}^x)^2 + (m_{i,t}^y - m_{j,t}^y)^2}. \qquad (4)$$

$$m_{k,t}^x = c_{k,t}^x + r_{k,t} \cdot \cos(\theta_{k,t}).$$

$$m_{k,t}^y = c_{k,t}^y + r_{k,t} \cdot \sin(\theta_{k,t}).$$

where δ is zero-mean Gaussian noise and $(m_{k,t}^x, m_{k,t}^y)$ is the projection of the estimate for node k from the polar parameterization into Cartesian xy-space. The belief state is then conditioned on the observations of the current time step by computing:

$$p(X_{t+1}|z_{1:t+1}, u_{1:t+1}) = \eta p(X_{t+1}|z_{1:t}, u_{1:t+1}) \cdot p(z_{t+1}|X_{t+1}). \qquad (5)$$

$$p(z_{t+1}|X_{t+1}) = \prod_k p(z_{t+1}^k|X_{i \in g(z_{t+1}^k), t+1}). \qquad (6)$$

where η is the normalization constant. The second term in the r.h.s of Eq.5 is the likelihood of the current observations. Eq.6 shows how this likelihood can be decomposed under the assumption that observations are independent given the locations of the nodes that made the observation. Note that each observation depends only upon the locations of the nodes in the set $g(z_{t+1}^k)$, which is the set of nodes that made the observation, and not the joint state vector. The range observations are augmented into the belief state by multiplying into the belief state a likelihood for each observation.

Upon the first observation of a particular node, the true distribution of the node is best represented as an annulus, see Figure 1(a). While an annulus is extremely non-Gaussian and difficult to model within the Cartesian xy-space, using the polar parameterization it is possible to approximate the annulus by an elongated Gaussian in polar coordinates ($r\theta$-space). This Gaussian approximation is given an arbitrary mean in θ (within the range $[0, 2\pi)$) with a large variance term, such that the probability along the θ dimension is near uniform, see Figure 1(b). Figure 1(c) shows the Gaussian ellipse (blue ellipse) overlaid on top of the true distribution (green shaded rectangle) in polar coordinates. By using this polar parameterization, a simple ellipse in polar coordinates transforms into an nonlinear annulus when projected into the xy-space. It must also be noted that the elongated ellipse in the polar coordinate extends past the range of the true distribution. This extended tail of the Gaussian ellipse, when projected into the xy-space appears curled up within the estimated annulus, as can be seen in Figure 1(b).

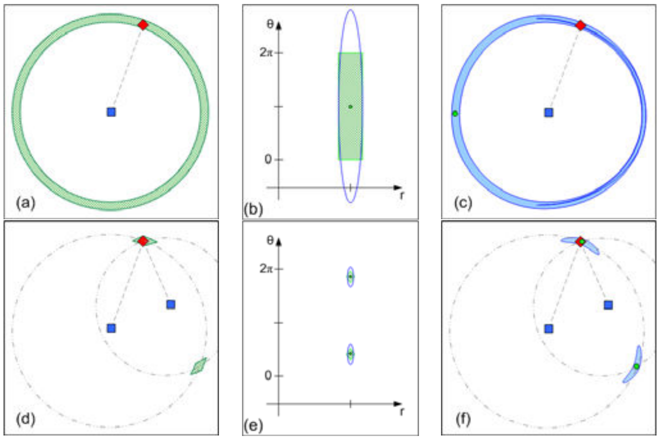

Fig. 1. Blue squares represent *observing* nodes, whose location is known. Red diamonds represent the true location of the *observed* node, whose position is being estimated and green circles represent the mean(s) for each mode of the estimated node. Green shaded regions (in (a),(b),(d),(e)) represent the true uncertainty distribution and blue ellipses (in (b),(c),(e),(f)) represent the estimated uncertainty distribution. The dashed gray lines and circles (in (c),(f)) represent the observed range measurements. (a) The true distribution (an annulus) of a single range observation. (b) The true distribution of an annulus (shaded rectangle), shown in polar coordinates, along with the unimodal Gaussian approximation (ellipse) of the true distribution. (c) The projection of the unimodal Gaussian ellipse from polar coordinates, shown in (b), to Cartesian xy-space. Note that the elongated Gaussian in polar coordinates, when project into the xy-space maintains a tail (curled up within itself), which helps make the distribution uniform along θ (in the range $[0, 2\pi)$). (d) The true (dual mode) distribution of two unique range observations. (e) The true multi-modal distribution (shaded region) and its multi-modal Gaussian approximation (ellipse), shown in its native polar coordinates. (f) The projection of the multi-modal Gaussian approximation, shown in (e), into the xy-space.

C. Extending to a Multi-Modal Representation

Thus far, we have assumed an unimodal Gaussian model, capable of approximating the non-linearities within single range observations. We have also presented a probabilistic filtering method that is well suited for an EKF-based network localization system. While this approach deals with non-linearities of an annulus, it fails to adequately deal with the multi-modal distribution of the system ((Figure 1(d))). Without a proper model to account for these naturally occurring ambiguities, the filter could pick the incorrect mode causing it to diverge when begin to disagree with the estimate. In order to properly deal with multi-modal distributions, we extend our approach to a multi-hypothesis representation, [12]. While particle filters are typically used to achieve multi-modal distributions, our approach uses the multi-hypothesis representation of the EKF to achieve similar performance.

Given an annulus-like prior distribution, a new range observation that intersects the annulus at two distinct locations leads to a multi-modal distribution with two distinct modes (peaks/local maxima in the distribution). We model these dual modes using separate filters/hypotheses for each mode.

To elaborate, whenever an annulus is split into separate modes, we simply duplicate the filter and adjust the mean of each filter to represent the two distinct intersection points. Then, by performing a measurement update using the new mean, we are able to appropriately update the covariance terms within the filter. The simple case of splitting a single annulus into two separate modes given a new range observation is shown in Figure 1(d) and (e). Figure 1(f) shows the Gaussian ellipses (blue ellipses) for the dual-modes overlaid on top of the true distribution (green shaded rectangles) in polar coordinates. The mean of the two modes can be determined easily using triangulation, given the location of the two observing nodes, as described in [13]. While this illustration demonstrates the case where each observing node has a single mode, in the case where each observer has Ω_1 and Ω_2 number of modes respectively, the number of modes the observed node can have is at most $2 \cdot \Omega_1 \cdot \Omega_2$. If a given node has more than two observers, then the number of modes the observed node can grow by the product of the number of modes for each of its observers. As might be expected, in an under-constrained system, when only sparse connectivity exists between nodes, the combinatorial of each node's modes will explode quickly.

While this solution doesn't scale well to the addition more nodes with sparse connectivity, intelligently deciding when to add new hypotheses and delete duplicate or unlikely ones could help limit the excessive growth of hypothesis count. In our implementation, at each iteration when the belief state is updated, we remove any duplicate hypotheses. A hypothesis is considered duplicate, when it has a mean and covariance similar to another hypothesis. This can be checked using a distribution comparison metric such as the Kullback-Leibler distance (KL-distance). In addition, at the end of each update, we check the (normalized) likelihood of each hypothesis, given all the measurements, and retain hypotheses with likelihoods above a certain threshold (relative to the likelihoods of all existing hypotheses), thus greatly reducing the number of hypotheses.

D. Decentralization

The model we have developed so far does not scale well to large networks for two reasons. First, as we had mentioned above, the representation of the multiple hypotheses grows exponentially as additional nodes are added without resolving existing ambiguities. Second, the computation requirement also grows as more and more nodes are added, making it difficult for a real-time implementation of the method on low-end processors typically available on sensor network nodes. Here we propose a very simple scheme for distributing the estimation algorithm in a decentralized manner. Deriving motivation from the non-parametric belief propagation work by Ihler et al., [7], we propose that each node be able to share a message to its immediate neighbors (nodes that have connectivity to this node). In these messages, each node shares the part of its belief state that encodes information about its own estimate that is novel to each of its neighbors. For example, a message from node i to node j at time t is of the form:

$$m_t^{i,j} = p(X_{i,t}|z_{1:t}, u_{1:t}, m_{t-1}^{k,i}); k \in \Gamma_i^o \backslash j \tag{7}$$

where, $\Gamma_i^o \backslash j$ is the set of observed neighbors to node i excluding node j. Upon receiving each such message, the node updates its estimate based on the message prior to performing any measurement updates. This allows for proper flow of information along the network.

The key difference in our approach compared to that of Ihler et al., [7], is that each node shares its own estimate rather than its marginalized estimate of its neighbor's position. The particular benefit Ihler et al. gain in not sharing each node's own estimate is that each node does not have to maintain an estimate of its neighbors, which is particularly difficult within the particle filter representation they utilize. However, since we use an EKF the cost of maintaining your neighbor's estimate is considerably low. Furthermore, by maintaining all neighbor's estimate within each node's state vector, the information encoded within cross-correlation terms of the covariance matrix in the EKF is not lost (providing an improved estimate of the true distribution). The inherent distributed nature of this message-passing algorithm, lends itself to a decentralized implementation where the problem of global network localization is solved independently, in small parts, by each individual node.

Another change we enforce in our decentralized implementation, is to limit the state vector of each node to *only* maintain an estimate of its immediate neighbors. This implies that node i does not maintain an estimate of node j if there is no connectivity (equivalent to observing a measurement) between node i and node j. By enforcing this constraint, we not only gain a benefit in computation cost (due to the reduced state vector size) but each node now also maintains considerably fewer hypotheses since the combinatorial of the modes of each node in its state vector will not grow too large, especially within a sparsely connected network. A particular drawback of this approach is that in the presence of loops (as is the case in most sensor network applications) some information can indeed be counted twice, making it less likely to accurately recover the exact state representation as the centralized approach. However, in our experiments, this approach has shown excellent empirical performance and deserves mention.

IV. Reducing Uncertainty with Motion

In this section, we describe a motion planning algorithm for the mobile nodes that seeks to minimize the uncertainty of both static and mobile nodes. This approach is similar to that described by Sim and Roy, [14], except that instead of using an SEIF, we utilized the EKF described above. We also think of the problem of reducing uncertainty as the problem of gathering data efficiently to produce a precise belief state, which in turn is about selecting new measurements that are maximally informative. Since the EKF is a generative Bayesian estimator, we select new data such that it is maximally informative about the belief state, $\xi = p(X_t|z_{1:t}, u_{1:t})$. Decision theory tells us that the gain in information between any two distributions is the relative change in entropy [15]. We therefore choose exploration strategies that maximally reduce the entropy of the belief state.

The entropy of a Gaussian distribution can be computed directly from its covariance matrix as below:

$$H(\xi) = \int_{\Xi} \xi \cdot \log \xi \propto \log \det(\Sigma). \tag{8}$$

The maximally informative trajectory must therefore have the smallest covariance matrix, Σ. Thus the gain in information from time t to $t+1$ is:

$$\Delta H = \log \det(\Sigma_{t+1}) - \log \det(\Sigma_t) \tag{9}$$

If we find the shortest trajectory that minimizes this quantity, we should converge to the most accurate estimate the fastest. If we apply this control scheme while restricting ourselves to the class of trajectories that avoid recently visited regions, a gradual exploration strategy that also minimizes the uncertainty of the belief state is achieved. This is implemented within a 1-step look ahead planner which chooses the best next step given a set of simulated measurements, computed based on the most likely hypothesis within our multi-hypothesis filter. While the most likely hypothesis changes over time based on the measurements (until the node estimates converge), by picking a path that best fits a specific hypothesis, we are able to avoid the saddle points (and other such inflection points) within the planning space cause by the multi-modal distribution.

V. Results

We present a number of example sensor networks, one simulation and one real-world experiment to demonstrate the scalability and utility of our methods. In addition, we present an analysis of our method comparing the performance of our decentralized implementation with its centralized counterpart.

A. Large-Scale Simulation

In this example there are 49 static nodes and 1 mobile node carried by a robot. The nodes are sparsely connected and clustered into smaller groups, making it difficult to achieve unimodal localization results without motion. We corrupt both the odometric and range measurements using Gaussian noise approximating the real sensors we use in the real-world example below. Figure 3 (Row 1) shows the final localization result achieved by our method when the mobile node moves within the environment using the control law presented in Section IV. Numerical results of this experiment are presented in Table I.

B. Real-World Experiment

1) Setup and Hardware: We carried out our real-world experiment[1] on a Pioneer 1 robot from ActivMedia. It was equipped with a Parrot node ([4]) placed on top of the robot at about 1 ft. above the floor (See Fig. 2). The robot was driven around within a large indoor office area with partial clutter. Ground Truth of the robot's position was estimated using a SICK laser scanner and the Adaptive Monte Carlo Localization (AMCL) algorithm within the Player/Stage ([16]) code repository. In addition to the node that was placed on the robot, 13 other nodes were placed around the environment on top of stands, 1 ft. above the floor. The nodes are once again sparsely connected, making it difficult to achieve unimodal localization results without motion. The locations of these nodes were accurately surveyed to allow proper evaluation of the accuracy of our mapping results. It should also be noted that in addition to the control law presented in Section IV, the robot was also running a low level obstacle avoidance scheme that avoided collisions while attempting its best to keep to the planned trajectories. This was necessary because the robot was not given a map of the environment *a priori* and therefore could not plan around physical obstacles that blocked its path.

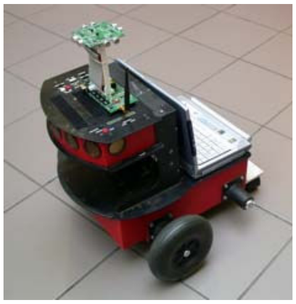

Fig. 2. Robot used for our experiments shown along with a range sensing node mounted on top. The robot has wheel encoders that measure distance traveled and estimate heading. Neither estimate is good but the heading estimate is particularly poor.

2) Results: Figure 3 (Row 2) shows the final localization result achieved by our method when the mobile node moves within the environment using the control law discussed above. A particular challenge with using real hardware is the slow rate of range measurements. Since the hardware doesn't support instantaneous range observations from several nodes at once, special considerations must be made to ensure that sufficient constraints exist to resolve ambiguities. To do this, we collect measurements over a period of 1 second and process them together, in order to retain correlations within the sequential observations.

C. Analysis

Table I presents some numerical results that compare our decentralized algorithm with its centralized implementation. Note that the centralized results are run offline using the logs recorded during the online run of our decentralized implementation. These results reveal that while the decentralized approach does not fully converge to the centralized approach, it still achieves reasonable accuracy in our experiments. In particular, while the final mapped locations of the nodes are similar, the estimated path of the robot in centralized approach is a little more accurate.

[1] An extended movie of the real-world experiment is available at http://www.frc.ri.cmu.edu/projects/emergencyresponse/dsgICRA08ex.mp4

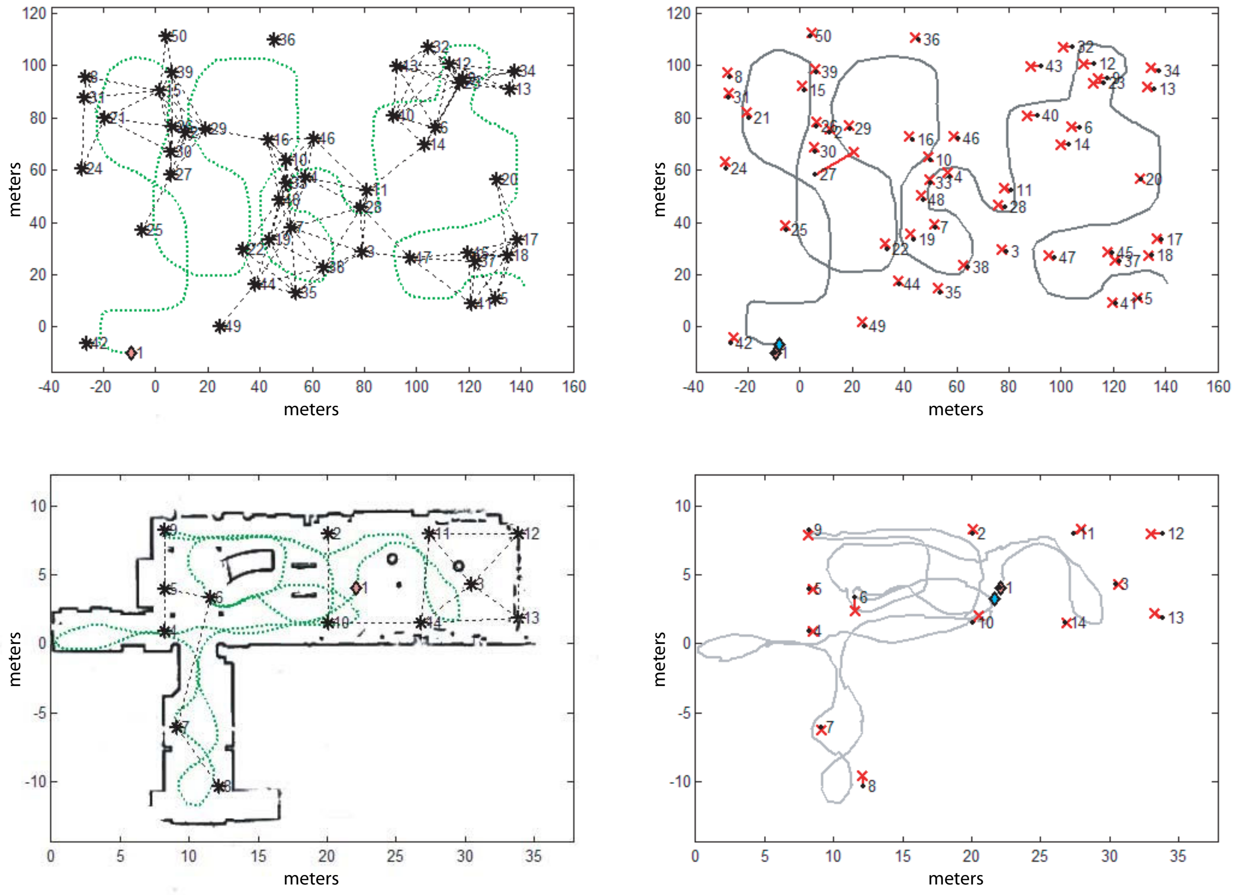

Fig. 3. Top row shows results from a simulation experiment (50 nodes) while the bottom row shows results from a real world experiment in an office building (14 nodes). Note that in both these experiments, the nodes are placed such that some nodes are isolated, making it impossible to accurately localize the full network without the presence of a mobile node. The map of the environment for the real-world example is overlaid on top of the true node locations and path. *Left Column*: Shows the true locations of the nodes ($*$), all the inter-node measurements received (dashed black line) and the true path the robot (ID #1) took (dotted green line). *Right Column*: Shows the error lines connecting true and estimated positions of the nodes along with the estimated path of the robot (solid gray line). The red cross marks ($\times$) the estimated location of a node and the error lines (solid red) connect the estimates to the true location of the nodes (black dots). Our approach achieves a path error of $1.1927m$ and mapping error of $1.5161m$ for the simulation experiment and $0.6198m$ and $0.4467m$ respectively for the real-world experiment.

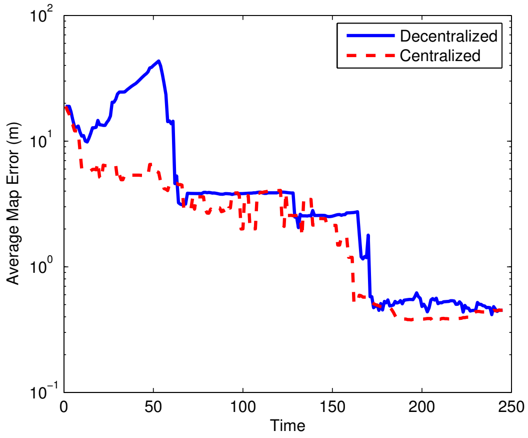

Fig. 4. Mean position error of all the nodes in the real-world experiment for both the centralized (dashed red line) and decentralized (solid blue line) implementations. In the decentralized approach, the estimates of the isolated nodes drift at the start in the absence of sufficient measurements, increasing its mean error.

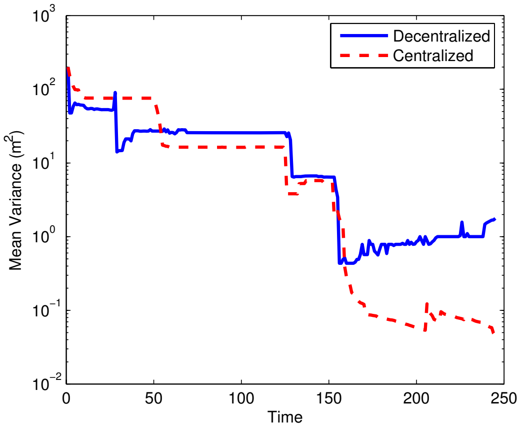

Fig. 5. Mean uncertainty (i.e. mean variance) in the position of the nodes for both centralized and decentralized implementations corresponding to the real-world experiment. The centralized method has to deal with more ambiguities at the start when few measurements are available but eventually produces a better result.

TABLE I

ROBOT PATH AND NODE MAPPING ERRORS (IN METERS) FOR BOTH SIMULATED AND REAL-WORLD EXPERIMENTS WITH CENTRALIZED AND DECENTRALIZED IMPLEMENTATIONS.

Method	Simulation		Real-World	
	Path Err.	Map Err.	Path Err.	Map Err.
Centralized	0.99m	1.48m	0.55m	0.45m
Decentralized	1.19m	1.52m	0.62m	0.45m

For the real-world data set, Figure 4 shows the mean mapping error of the nodes over time as the mobile node produces more data. Figure 5 shows mean uncertainty in the position estimate of the nodes over time. We see that with the decentralized implementation, initially the mean uncertainty can be low but error in the solution is high. This is because the estimates of the isolated nodes, maintained independently by each node, drift at the start in the absence of sufficient measurements. It is only after the mobile node travels within range of the isolated nodes can their estimates be fixed within the joint coordinate frame (without which their estimates remain free-floating).

In contrast, the centralized method has to deal with more ambiguities (multi-modal distributions) at the start when few measurements are available. And since the method tries to jointly estimate the positions of all the nodes within the same coordinate frame, the estimates of the isolated nodes do not drift. It should be noted here that without the presence of the mobile node, regardless of the method used, achieving an accurate estimate of the full network is impossible due to the lack of rigidity within the network.

VI. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated an efficient decentralized algorithm for localizing the nodes in a sparsely connected sensor network using range-only measurements and motion from a mobile robot. Our parameterized EKF-based approach requires little to no prior information about the node locations. We presented an extension of the EKF that handles multi-modalities within range-only mapping measurements. In addition, we formulated a decentralized framework for our algorithm which is well suited for a real-world implementation, reducing the computation and memory requirements on each nodes. Finally, we demonstrated the performance of our method using both simulated and real-world experiments and comparing the performance of our decentralized method with its centralized counterpart.

There remain many open directions for continued research. For example, the multi-modal extension presented here includes an empirical pruning step to remove unlikely and similar hypotheses. By changing the parameterization of the states, one can hope to encode the multi-modalities within the estimate into a single state representation, similar in principles to what was proposed by Stump et al., [10]. Also, alternate control strategies for reducing self and network estimate uncertainties can be explored. While the current approach provides a reasonable solution for sparsely cluttered environments, it still has potential to get stuck in local minima. In particular if the uncertainties of the nodes are symmetric around the mobile node, the controller will drive the robot on a sub-optimal path which might cause the uncertainties to grow. In addition, improvements to the algorithm that lend themselves to decentralized coordination of multiple mobile robots could prove to reduce the time it takes to accurately estimate the entire network, while providing additional capabilities for subsequent tasks that require decoupled cooperation of the mobile robots.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge Geoffrey Hollinger and Bradley Hamner for their insightful comments and assistance with the experiments.

REFERENCES

[1] P. Bahl and V. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *In Proc. of the IEEE Infocom 2000*, Tel Aviv, Israel, March 2000.

[2] N. Priyantha, A. Chakraborty, and H. Balakrishman, "The cricket location support system," in *In Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*, Boston, MA, August 2000.

[3] A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha, "Tracking Moving Devices with the Cricket Location System," in *2nd International Conference on Mobile Systems, Applications and Services (Mobisys 2004)*, Boston, MA, June 2004.

[4] W. Zhang, J. Djugash, and S. Singh, "Parrots: A range measuring sensor network," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-06-05, March 2007.

[5] I. Borg and P. Groenen, "Modern multidimensional scaling: theory and applications." New York: Springer, 1997.

[6] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *in SenSys'04:Proc 2nd international conference on Embedded networked sensor systems*. New York: ACM Press, 2004, pp. 50–61.

[7] 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.

[8] T. L. Song, "Observability of target tracking with range-only measurements," *Oceanic Engineering, IEEE Journal of*, vol. 24, no. 3, pp. 383–387, Jul 1999.

[9] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only slam for robots operating cooperatively with sensor networks," in *IEEE Int'l Conf. on Robotics and Automation (ICRA '06)*, 2006.

[10] E. Stump, B. Grocholsky, and V. Kumar, "Extensive representations and algorithms for nonlinear filtering and estimation," in *The Seventh International Workshop on the Algorithmic Foundations of Robotics*, July 2006.

[11] S. Funiak, C. E. Guestrin, R. Sukthankar, and M. Paskin, "Distributed localization of networked cameras," in *Fifth International Conference on Information Processing in Sensor Networks (IPSN'06)*, April 2006, pp. 34 – 42.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[13] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, M. J. Carey and D. A. Schneider, Eds., San Jose, California, 22–25 1995, pp. 163–174.

[14] R. Sim and N. Roy, "Active Exploration Planning for SLAM using Extended Information Filters," 2004.

[15] D. J. C. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, California Institute of Technology, 1991.

[16] B. Gerkey, R. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, June 30 - July 3, 2003*, 2003, pp. 317–323. [Online]. Available: http://playerstage.sourceforge.net/