36th Asilomar Conf. Signals, Systems & Computers, 2002

DISTRIBUTED DETECTION AND TRACKING IN SENSOR NETWORKS

Donal McErlean, Shrikanth Narayanan
{mcerlean, shri}@sipi.usc.edu
Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, California 90089

ABSTRACT

This work develops a framework for using a wireless ad-hoc network of distributed sensors for detection and tracking of moving objects. Possible applications might be vehicular traffic monitoring in highways or in battlefields, detecting and tracking people movement in secure areas, clots flowing through the bloodstream or even nuclear devices moving around cities. The central idea is to utilize decentralized processing within the network to build a better picture of what is going on (locally), before reporting the results(centrally). This notion, which relies on collaboration among sensors, reduces false alarms and avoids sending all the sensor measurements back to a centralized processing unit. Simple assumptions are made about the structure of the network and the capabilities of the individual nodes. Uncertainty is introduced in the form of uncertain node positions, uncertain position estimates, false events and an unknown number of objects.

Introduction

The emerging technology of wireless sensor networks provides many exciting and interesting applications. Such networks can provide an immense raw sensing capability in many different modalities. The huge difficulty in harnessing these networks lies in trying to process all the sensed data in a meaningful and power-efficient manner [1].

An important task for a sensor network is to be able to detect, track and classify objects. As objects move around the sensor field, they affect the observations at nearby nodes. The key to collaboration across nodes is to work out if and how the observations at different nodes are related, and then use these related observations to form more accurate estimates for the objects existence, track and type.

The problem of object detection and classification has been explored in [3] on an individual node basis. There is very little research on distributed detection and tracking within networks of wireless sensors. Object tracking is a topic that has been studied and developed extensively but primarily in the domain of active and passive radar. Graphical modeling techniques such as Kalman filtering and HMMs ([5], [6]) have been employed very successfully in this domain. Complex multiple hypothesis testing techniques are incorporated into their frameworks that rigorously evaluate every possible origin of the measurements received. However, they assume that all the measurements are available for processing at a centralized node. Before continuing, we will outline some key design criteria for any proposed detection or tracking algorithms in the domain of wireless sensor networks:

▪**1. Decentralized processing -** While it is easier to consider and design algorithms in an architecture where the sensor outputs are communicated back to a central processing unit, this is generally not feasible. When dealing with a network of un-tethered nodes, a finite amount of energy is a factor that must be taken into consideration. Communication is the primary energy consumer, particularly when one considers that radio signal power in ground based sensor networks drops off at r^4 due to the short antenna heights [2]. The key is to process the sensor outputs as much as possible within the network, so as to avoid communicating large amounts of information over large distances.

▪**2. Processing sensed data at the nodes** - There are many levels in which the sensed data can be shared and processed among nodes- e.g. signal level, feature level and decision level. At each of these levels, the information content is reduced, but this in turn reduces the required amount of data to be communicated between nodes. In short, processing is cheap and communication is expensive.

▪**3. Dealing with uncertainty -** Typically, the exact positions of the nodes might not be known, which will affect any sort of position estimation algorithms. The nodes are typically very low-cost, low-power throwaway devices that might be prone to noise, increasing the chance of false measurements.

▪**4. Incoherent signal processing** - Accurately synchronizing such simple cheap devices could be difficult. This combined with non-overlapping sensing ranges in sparse networks limits the use of traditional coherent array processing algorithms [9].

▪**4. Generic algorithms for different modalities -** Nodes might be equipped to record signals from many different modalities. These might include acoustic, optical, IR, temperature, radioactive or seismic modalities. (The sensing nodes in the SensIT [7] project recorded signals in three modalities). Devising a generic algorithm that can be applied to the modality available is essential.

Taking these points into account, a possible scenario is proposed, along with a framework for distributed detection and tracking. This is followed with some computer simulations implementing the algorithms and an analysis of the performance and energy savings.

1.1 Scenario

The sensor network consists of a sparse set of sensing nodes of one modality that have a finite sensing radius due to signal attenuation. Objects and other phenomena may pass through the sensor field. The objects affect the signals received by the nodes.

Figure 2 below displays the signal received by three types of sensor at a node in the SensIT experiment [7] as three vehicles pass by in succession. Each of the three modalities displays very different properties. The seismic modality provides the largest sensing range, whereas the IR modality has a limited one. The seismic and acoustic modalities have a greater range of amplitude in their peaks, which could be used to estimate the distances of the various objects from the node, whereas in the IR case, the response is more binary in nature. Additionally, the acoustic and seismic signals might also allow an estimate of velocity based on the rate of change of frequency and amplitude, but the IR sensor does not allow this.

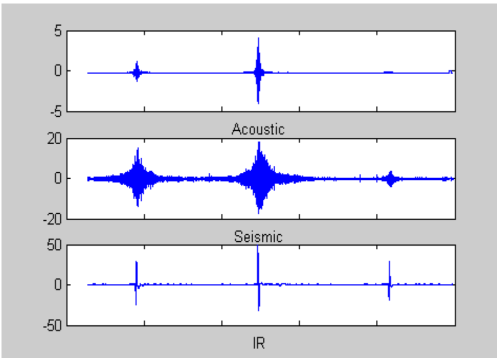


Figure 1. Three vehicles passing a sensor node in the SensIT 00 experiment, and the corresponding response in each modality.

The one property that can be extracted from all three is the time that the discrete 'event' occurs. The resulting event implicitly provides an estimate for the position of the object, based on the position of the node. The event could be detected using a simple peak detector with a suitable energy threshold, as described in [3]. This type of detector assumes little about the modality or characteristics of the source, although it can be prone to both false and missed detections.

As objects move through the sensor field, they produce a set of events. While the SensIT experiment was conducted under very controlled circumstances, in a real application many detected events unrelated to moving objects might occur. These false events might be due to background phenomena, other types of objects not of interest or perhaps faulty sensors. Figure 2 shows the detections two objects cause as they move through a fairly dense network along with some unrelated events.

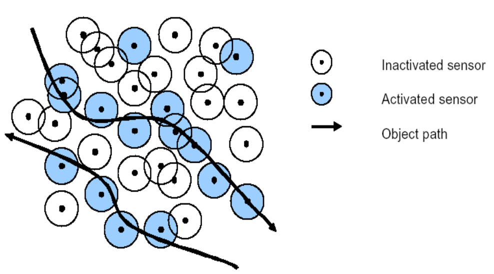


Figure 2. Objects traversing a network and resulting detections (ellipses define sensing radius).

In such a scenario, a stand alone detector at an individual sensor is not that useful. Given an event, it is not possible to infer that an object caused it. Rather than report back each event to a distant central node, it would be logical to share the evidence among neighboring nodes and then report any useful results with the central node. If several correlated events are associated together, a more accurate hypothesis as to whether the object is present can be formed, along with an increasingly accurate estimate of its position and velocity. This idea implicitly performs distributed tracking and can be extended to that application. In addition, it has been shown in [4] that several associated classification experiments can improve classification of the object.

1.2 Definition of the Network and the Objects

In this section, simple models for the sensor network and objects are defined. A system level approach is taken, assuming much about the capabilities of the network in terms of organization and communication. Efficient protocols for these are being developed elsewhere [1].

1.2 .1 Network Topology

A detailed study of network layouts can be found in [8]. The specifications and topology of our network is defined by the following:

- *Dimension* - (X, Y)
- *Number of nodes* - M
- *Actual node positions* -Ai, $\forall$ i=1:M. These are the actual physical location of the nodes. The nodes are normally distributed around grid points with a variance of $\sigma_{pos.}$
- *Known node positions* - Ki, $\forall$ i=1:M. The known node positions are each normally distributed around the actual positions with a variance of $\sigma_{k.}$

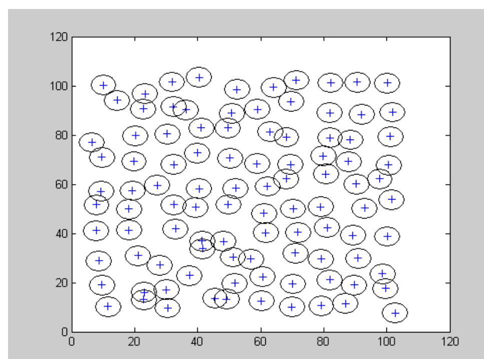


Figure 3. Sample network with X=100m,Y=100m, M=100, σ_{pos}=2, and S=4m.

1.2.2 Sensor Nodes

The nodes check for an object at every discrete detection interval ζ. It is assumed that an object triggers the detector and causes an event when it crosses into the sensing radius **S** and is at its nearest position to the node. Nodes can broadcast to all other nodes within a certain radius **R.**

1.2.3 False Event Statistics

A model is needed for the false events at the sensor nodes. These could be caused by interference from objects not of interest, background phenomena, and faulty sensors. A simple model incorporating many of these phenomena would be a discrete Bernoulli i.i.d. random process at each node with probability of occurrence **p** at each time interval ζ of the simulation.

1.2.4 Object Movements

Simple linear trajectories for the objects are assumed. It is assumed the objects travel at an initial velocity taken from a uniform distribution centered on zero and extending to some maximum $\mathbf{V_{max}}$. Each object has a 4-element state vector $\mathbf{x}$ consisting of position and velocity with evolves at each time interval ζ according to the following model:

$$x(t_{k+1}) = \Theta x(t_k) + \Gamma w(t_k) \qquad (1)$$

$w(t_k)$ is a white noise sequence with zero mean and covariance Q. T is the time between two successive intervals.

$$\Theta = \begin{vmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad Q = \begin{vmatrix} qT & 0 \\ 0 & qT \end{vmatrix}$$

2.1 Event Associations and Object State Estimation

There are many algorithms that have been developed for tracking over time using a continuous set of noisy measurements. The classic reference [6] applies the standard Kalman filtering algorithm to the tracking of objects with a linear model. The algorithm receives continual measurements at fixed time intervals from the sensor (in radar tracking for example). In the domain, measurements are only available when the objects pass the nodes, and more importantly the measurements themselves are only available at the nodes. Despite the asynchronous measurement intervals, the Kalman filtering approach is very appealing due to its sequential convergence properties and the 'prediction step' that provides a means for associating new measurements.

As an object passes through the network, it will produce a set of events $\mathbf{E=[e_1, e_2,\ldots,e_n]}$. Each event $\mathbf{e_i}$ provides a time $\mathbf{t_i}$ and the node position $\mathbf{z(t_i)}$. The position is related to the underlying object state at the time of the event by means of the following equation:

$$z(t_i) = Hx(t_i) + v(t_i) \qquad (2)$$

$v(t_i)$ is a white noise sequence with zero mean and covariance R

$$H = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix} \qquad R = \begin{vmatrix} r & 0 \\ 0 & r \end{vmatrix}$$

Here the vector $\mathbf{z(t_i)}$ only measures position, not velocity, and is given by the known position of the node. The additive noise vector $\mathbf{v(t_i)}$ represents the uncertainty both in the known node position and in the position of the object within the sensing radius of the node. The variance caused by the non-zero sensing radius can be modeled by $\mathbf{S/3}$ (assumption), and the variance of the node position by $\boldsymbol{\sigma_k}$. Hence we get:

$$r = (s/3) + \sigma_k \qquad (3)$$

Assuming all the events are caused by the same underlying object, the conditional probability distribution of the state variables is a multivariate normal distribution given by the Kalman filter [10]. The mean $\bar{x}$ and variance $\bar{P}$ evolve between detections according to the following "time-update" equations:

$$\begin{aligned} \bar{x}(t_{k+1}) &= \Theta \hat{x}(t_k) \\ \bar{P}(t_{k+1}) &= \Theta \hat{P}(t_k)\Theta^T + \Gamma Q \Gamma^T \end{aligned} \qquad (4)$$

At every new event, the conditional mean $\hat{x}$ and covariance $\hat{P}$ are given by the following "measurement update" equations:

$$\begin{aligned} \hat{x}(t_{k+1}) &= \bar{x}(t_k) + K\left[z(t_k) - H\bar{x}(t_k)\right] \\ \hat{P}(t_{k+1}) &= \bar{P} - \bar{P}H^T\left(H\bar{P}H^T + R\right)^{-1} H\bar{P} \\ K &= \hat{P}H^T R^{-1} \end{aligned} \qquad (5)$$

As a data association criterion, the conditional probability distribution of the state variables provided by (5) may be used to validate any new event. The new event is associated if it falls within a certain n-sigma of the time-updated state distributions. The measurements can only be viewed at sporadic time intervals, depending on when the object passes a node. This means $\mathbf{\Theta}$ and $\mathbf{Q}$ must be scaled according to the time interval between two sequential events. While $\mathbf{R}$ gives the initial position covariance, the initial velocity distribution should be dependant upon the range of velocities of the object to be detected. It is assumed the variance is given by a third of the maximum velocity of the objects to be tracked, $\mathbf{V_{max}/3}$, and the mean is zero.

2.2 Experiments

Given a set of events generated by an object moving through the network, the objects position and velocity at each event can be estimated using this algorithm. Figures 4a and 4b display the performance of the above algorithm as a single object moves through a sensor field. In this example, the object originates at position (5,5), and traverses the network at an unknown velocity taken from a uniform distribution bounded by V_{max}. The state noise is zero for the simulations.

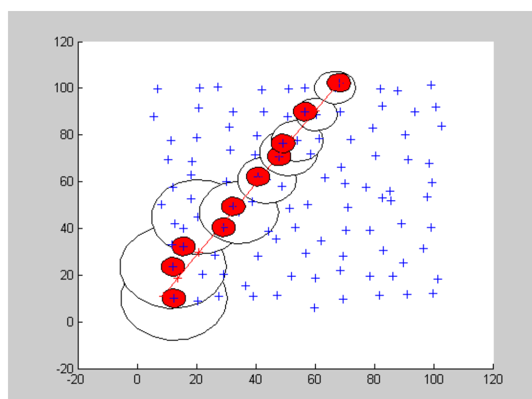


Figure 4a. The line represents the true object track. The darkened nodes (and their sensing range) represent nodes at which events have occurred. The large circles denote the 3-sigma prediction gates at the time of the next event in the sequence. (X=100m, Y=100m,σ_{pos}=2, σ_k=2, S=4,M=150, ζ=0.1s, V_{max}=10m/s)

To view the performance of the algorithm over time, 50 such simulations were run for randomly generated networks and objects, and the results can be shown in Figure 4b. While the position estimate is initially quite small due to the relatively small sensing radii, the velocity estimate takes several transitions to converge.

The performance of the estimator will vary considerably for different network setups, but an analysis of this is beyond the scope of this paper. It may also be noted from figure 4a that a 3-sigma gate successfully associates the subsequent events in the sequence.

While not optimal by any means, this algorithm provides an increasingly accurate state estimate as more events are associated, and the gating property allows the associations to become increasingly selective as time progresses.

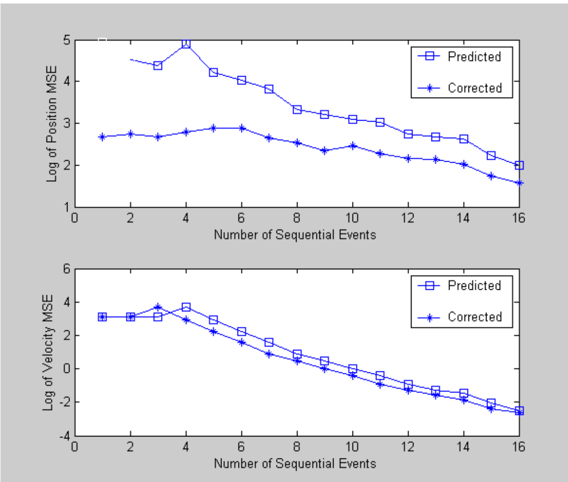


Figure 4b. The MSE error in estimated position and velocity versus number of sequential events.

3.1 Hypothesis Propagation through the Network

In the previous tracking experiment, it was assumed that the set of events were all caused by the same object and could be processed by a centralized algorithm. In a real application, there will not be *a priori* knowledge of how many objects there are or whether the events are the result of other phenomena. What is needed is an algorithm for making sense of the events in a distributed and sequential manner. It is assumed no centralized processing node is available, but that there is some distant reporting node where the final results need to be reported to.

By initiating a sequence for estimating after an event, the node is starting a hypothesis that the event was caused by an object moving in the sensor field. This hypothesis is sent to surrounding nodes where it is stored. If an event occurs at these nodes that is associated with a stored hypothesis, this hypothesis can be updated and broadcast again. Thus, the hypotheses and their estimates are propagated through the nodes in the network. They are stored at the nodes, and are updated as events occur. Old hypotheses expire, and redundant hypotheses are pruned.

For detection purposes, a confirmed target is reported after a hypothesis has reached some detection criteria (a suitable number of transitions **c** will be used here). For longer-term tracking applications of a confirmed object, the track can be reported at regular time intervals, but not necessarily at every transition. The inherent power saving in such distributed algorithms is obvious - many local broadcasts but relatively few global broadcasts. While distributed tracking/classification is possible using this idea, for brevity distributed detection (or track initiation) will be examined for the rest of this paper.

3.2 Distributed Detection

A distributed detection algorithm is proposed that operates as follows. Whenever an event occurs at a node in the network, it carries out the following algorithm:

1. Check if the event occurs within the prediction gates of any previous 'time-updated' hypotheses received from the surrounding nodes. If so, 'measurement update' these hypotheses using (5).
2. Check if the convergence criterion is met. If so, report the confirmed object to the central node. A distributed tracking algorithm takes over at this point.
3. Initiate a new hypothesis starting at this event
4. Broadcast the hypotheses to the surrounding nodes

In the proposed algorithm, the processing that occurs at each node in the sequence when an event occurs is outlined in Figure 5 below.

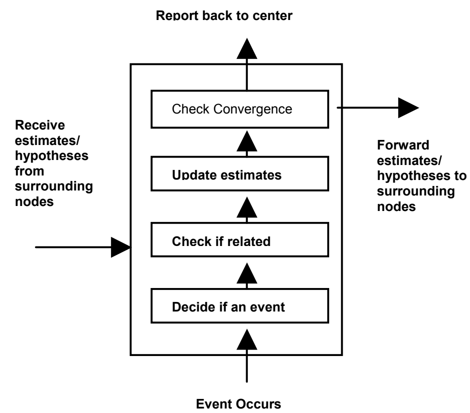


Figure 5. The processing at each node when events occur.

Figure 6 below displays how the hypotheses propagate through a network where four detections have occurred. The detections are labeled in the order in which they occurred. All the hypotheses generated at each node by this sequence of detections are displayed, without any pruning.

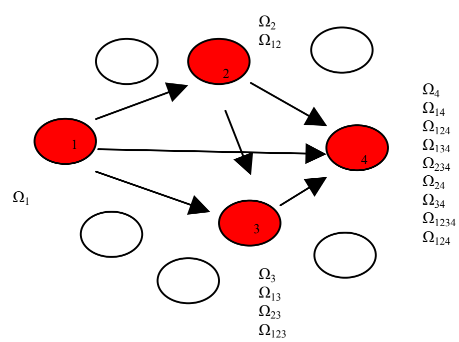


Figure 6. A sequence of 4 detections and the hypotheses generated at each activated sensor.

Given a set of **D** events occurring over a certain time period, there are exactly $\mathbf{2^D\text{-}1}$ possible ways of associating the detections together into causal sequences. Clearly this is an enormous search-space that increases exponentially with the number of events. A single object on a restricted trajectory cannot cause the majority of these sequences, whereas others are redundant (caused by the same object). Processing, broadcasting and storing excess hypotheses is costly. The gating region governed by **ή**-sigma discards many of these hypotheses, as does the fact that each node can only broadcast hypotheses to others nodes within a distance **R**. Some additional criteria for eliminating these excess hypotheses need to be developed. Two additional steps are added to the algorithm to prune the non-promising hypotheses from being propagated when an event occurs at a node:

1. Hypotheses are not updated if their last update was more than **τ** seconds previous. The expired hypotheses are discarded.
2. For updated hypotheses that have similar state estimates (i.e. Euclidean distance between them < **δ**), only the hypothesis with the lowest variance is broadcast. The redundant hypotheses are then discarded.

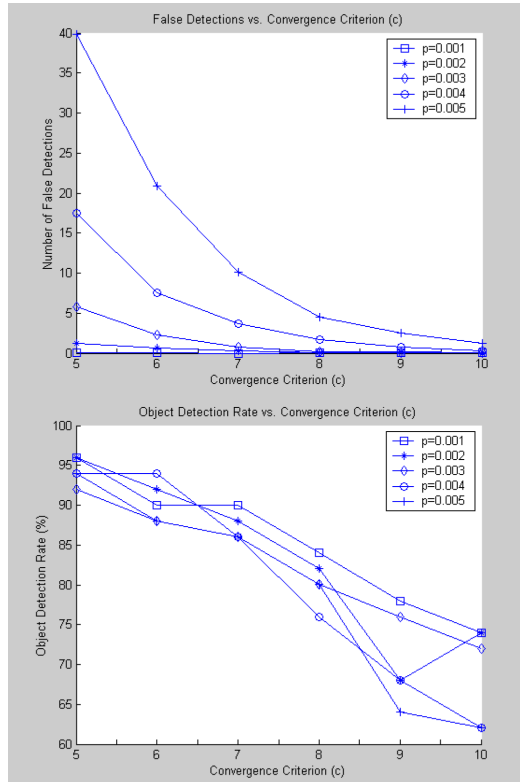
In applications where events are caused by many different phenomena, this algorithm will allow the network to detect moving objects in an energy efficient, accurate and decentralized manner. Extending the convergence criterion will delay the detection, but will provide a more accurate decision.

3.3 Experiments

In the simulated environment, over a certain time period **T**, one object traverses the network and generates a sequence of events. In addition, the independent false event processes at each node generate unrelated events. The nodes in the network execute the algorithm described above and report back any confirmed detections. A true detection occurs if the node, which it originates from, contains the object within its sensing radius. Any additional detection reported is considered a false detection.

The network and object parameters are the same as those in the previous simulations. The duration is set **T**=10 minutes. The object starts its transition at unknown position [5,5] during this time. The broadcast radius is **R**=30m, hypotheses expire after **τ** =20s, the similarity distance is **δ** =5, and the gating region is governed by **ή**=4. 50 simulations were executed and the results were averaged. Figure 7a below displays the level false detections per simulation, as we vary the level of unrelated events with **p** and the convergence criterion with **c**. Figure 7b displays the object detection rate for the same range of parameters.

It is noted that the number of false detections reported decreases exponentially as the convergence criterion **c** is increased. In addition the number of false detections reported increases exponentially with **p**. The object detection rate does not appear to be affected greatly by **p**, but decreases linearly with **c.** This is because the pruning procedure causes some true hypotheses to be eliminated, and this becomes more likely as the required length of the sequences increases (it is noted the average object created a sequence of 14 events as it crossed the network). A detailed examination of how the pruning parameters (**R**, **τ**, **δ, ή**) affect the results in beyond the scope of this paper.



Figures 7a , 7b: Performance with respect to convergence criterion.

4 Conclusions and Future Work

In this paper, a framework for distributed tracking and detection in sparse sensor networks is presented. Discrete detected events occurring in the network are associated together on a local basis using a decentralized hypothesis propagation algorithm. Confirmed objects may be reported once a suitable confidence criteria has been reached. Only a single short-range broadcast is required when an event occurs. Long-range broadcasts only occur after a confirmed detection occurs. The power savings over a centralized detection algorithm become significant when the number of events is much greater than the number of confirmed detections.

In future work, a more rigorous probabilistic framework will be developed, incorporating known object and false event densities. The decision process at each node will be explicitly modeled and will take into account soft decisions. Fully-fledged distributed continuous tracking of targets in such networks will be explored. It is also hoped to implement some of the proposed algorithms in a working sensor network.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks", in *Proc. ICASSP'2001*, Salt Lake City, UT, 2001

[2] G. Pottie and W. Kaiser, "Wireless sensor networks", Communications of the ACM, vol. 43, pp. 51-58, May 2000

[3] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, Classification, and Tracking of Targets", *IEEE Signal Processing Magazine,* vol. 19, pp. 17-29, March 2002

[4] C. Meesookho, S. Narayanan, “"Collaborative Classification Applications in Sensor Networks"”, *IEEE Sensor Array and Multi-Channel Signal Processing Workshop,* Rosslyn, VA, August 2002.
[5] F. Martinerie, “"Data Fusion and Tracking Using HMMs in a Distributed Sensor Network",” *IEEE Transactions on Aerospace and Electronics System,* vol. 33, no. 1, January 1997.
[6] D. Reid, “An Algorithm for Tracking Multiple Targets”, *IEEE Transactions on Automatic Control*, vol. 24, no. 1, December 1979.
[7] SensIT Project Database, http://dstl.bbn.com.
[8] L. Doherty, “Algorithms for Position and Data Recovery in Wireless Sensor Networks”, UC Berkeley EECS Masters Report, May 2000.
[9] K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli, “Blind beamforming on a randomly distributed sensor array system”, *IEEE JSAC*, vol. 16, no. 8, Oct. 1998.
[10] R. E. Kalman, “A new approach to linear filtering and prediction problems”, *J. Basic Eng*., vol. 82-D, pp. 35-45, 1960.