ROBUST TARGET TRACKING WITH UNRELIABLE BINARY PROXIMITY SENSORS

B. F. La Scala, M. R. Morelande, C. O. Savage

Melbourne Systems Laboratory, Dept. of Electrical Engineering,
University of Melbourne, Australia.

ABSTRACT

This paper describes an algorithm for target tracking in a distributed network of binary proximity sensors that is robust to sensor failures. Tracking with such sensors is a difficult problem as they only transmit a single binary digit regarding the presence of a target. The operational status of these sensors may also not be accurately known. This paper describes a Gaussian mixture-based tracking algorithm that addresses these two issues.

1. INTRODUCTION

Recent advances in low-power microsensors and wireless network technology have led to an increase in the use of wireless sensor networks for a variety of applications. Such networks are typically made up of relatively cheap sensors, often termed *motes*, with limited processing resources and battery life [1]. The limited on-board resources prevent significant computations being performed at the sensor level. Instead, these sensors transmit information over wireless communication channels that have limited bandwidth to a central processing node, which combines the data to perform tasks such as detection, target tracking and classification. This paper examines target tracking using binary proximity sensors, which transmit only a single bit of information regarding the presence of a target within their sensing range.

The simplest approach to tracking with binary proximity sensors is to record the location of each mote which reported a detection and fit a straight line through these data points. The works of [2, 3] are based on this approach. However, these methods do not make use of all the available information. The absence of a detection in the predicted target location can also provide information on the track prediction accuracy. The works of [4, 5, 6] are based on particle filters [7] and are therefore able to use information from all sensors to update the target track. This particle filtering based approach is capable of accurate tracking but can be computationally expensive, particularly when there are multiple targets present. To alleviate some of this computational burden [8, 9] propose a distributed particle filter based method. However, this requires significant processing capabilities at each sensor node.

All the methods discussed above assume all the motes in the surveillance region are functioning. These trackers therefore assume the lack of a target detection means that the target is not near the sensor. However, as these sensors are battery powered an alternative explanation is that the mote is no longer operational. To track robustly with such sensors it is necessary for the central processor to know which motes in the surveillance region are functioning. As the wireless networking protocols employed by such sensor networks are ad hoc, such status information is not necessarily available at the central processor as common routing protocols for such networks do not require this [10].

In this paper, we propose an algorithm using Gaussian mixtures that simultaneously tracks a target through a field of binary proximity sensors and also estimates which sensors are operational. This algorithm is robust to sensor failures and is less computationally complex than one based on particle filters. In addition, the central processor can use the estimates of the probability a mote is operational to schedule targeted queries to individual motes, rather than requiring status updates from all motes. The performance of this tracker when tracking a single target is examined using simulations.

2. GAUSSIAN MIXTURE TRACKER

2.1. Target Motion Model

The target is assumed to evolve according to

$$\mathbf{x}_k|\mathbf{x}_{k-1} \sim \mathcal{N}(\mathbf{F}_k\mathbf{x}_{k-1}, \mathbf{Q}_k) \tag{1}$$

for $k = 1, 2, \dots$ where $\mathcal{N}(\mu, \Sigma)$ is a Gaussian distribution with mean μ and covariance matrix Σ. The target state vector includes the position of the target in addition to other variables. The initial state is assumed to be given as $\mathbf{x}_0 \sim \mathcal{N}(\mu_0, \mathbf{P}_0)$ where μ_0 and $\mathbf{P}_0$ are known.

2.2. Sensor Measurement Model

Let there be m sensors in the surveillance region with the position of the jth sensor denoted as $\mathbf{z}_j$, which is known. At each sampling instant $t_1, t_2, \dots$ one or more sensors transmits a packet to the central processing node indicating that they have detected a target within their sensor range. Sensors that have detected nothing do not transmit as the power constraints and bandwidth limits in such networks make transmissions from each node in a sensor network at each sampling instant infeasible [10]. It is assumed that some mechanism, such as that discussed in [11], has been used to provide time synchronisation.

Let $\mathbf{y}_k = (y_{k,1}, y_{k,2}, \dots, y_{k,m})'$, $y_{k,j} \in \{0, 1\}$ denote the collection of sensor returns at time t_k where $y_{k,j} = 1$ indicates a transmission was received from the jth sensor and a zero indicates nothing was received. It is assumed that the sensors operate independently, which leads to a measurement likelihood of the form

$$p(\mathbf{y}_k|\mathbf{x}_k) = \prod_{j=1}^{m} p(y_{k,j}|\mathbf{x}_k) \tag{2}$$

There are two possible causes for the lack of a detection by a mote. One is that the mote is operational but the signal has been missed by the sensor. The other is that the mote itself has failed. While it is possible for motes to transmit status information to the

This work was supported in part by the Defense Advanced Research Projects Agency of the US Department of Defense and was monitored by the Office of Naval Research under Contract No. N00014-04-C-0437.

central node, this will only happen periodically and not necessarily very frequently. It is possible that between status updates one or more motes may have become inoperative due to battery exhaustion or other hardware failures.[1]

We assume that during the lifetime of a particular target trajectory motes do not fail. That is, the mote scenario is fixed but some of the motes may be non-operational prior to the target entering the surveillance region. Motes that transmit a detection packet during the track lifetime are clearly alive, but one or more of the other motes may not be.

Let $\mathcal{S}_k$ be the set of all possible mote configurations at time t_k and $\mathbf{s}_j \in \mathcal{S}_k$ be such that $\mathbf{s}_j = \left[s_{j,i}\right]_{i=1,\ldots,m}$ where $s_{j,i} = 1$ if mote i is alive and zero otherwise. Note that the size of the set of all possible mote configurations prior to tracking is $|\mathcal{S}_0| = 2^m$ but this can be reduced in practice if it is assumed that at most some fixed number of motes $d_{\max} < m$ are dead. This is a reasonable assumptions as mote failures will be relatively rare and a significant number of motes are required to be alive for the network to remain connected.

It seems reasonable to assume that the probability a particular sensor, when operational, will detect the target will be a decreasing function of the range from the target to the sensor therefore a Gaussian model has been used for convenience. This leads to the following measurement likelihood for a single mote

$$p(y_{k,i}=1|\mathbf{x}_k, s_{j,i}) = \begin{cases} 0 & s_{j,i}=0 \\ P_D G(\mathbf{z}_i;\mathbf{H}\mathbf{x}_k,\Sigma) & s_{j,i}=1 \end{cases} \tag{3}$$

$$p(y_{k,i}=0|\mathbf{x}_k, s_{j,i}) = \begin{cases} 1 & s_{j,i}=0 \\ 1-P_D G(\mathbf{z}_i;\mathbf{H}\mathbf{x}_k,\Sigma) & s_{j,i}=1 \end{cases} \tag{4}$$

where $G(\mathbf{z};\mu,\Sigma) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mu)'\Sigma^{-1}(\mathbf{z}-\mu)\right)$ and $\mathbf{H}$ selects the position elements from the target state vector. The positive definite matrix Σ defines the "footprint" of the sensor while $P_D \in [0,1]$ determines the ability of the sensor to detect the target. These parameters can vary between sensors but are assumed to be the same here for notational convenience. Therefore

$$\begin{aligned} p(y_{k,i}|\mathbf{x}_k, s_{j,i}) = {} & \left[s_{j,i}P_D G(\mathbf{z}_i;\mathbf{H}\mathbf{x}_k,\Sigma)\right]^{y_{k,i}} \\ & \times \left[1-s_{j,i}P_D G(\mathbf{z}_i;\mathbf{H}\mathbf{x}_k,\Sigma)\right]^{1-y_{k,i}} \end{aligned} \tag{5}$$

Denote by $\mathcal{D}_k$ the set of motes that transmitted a detection at time t_k and $\mathcal{U}_{k,j}$ the set of motes that did not detect the target but which are assumed to be alive under scenario $\mathbf{s}_j$. With these definitions and (5), the likelihood (2) can be written as

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{x}_k,\mathbf{s}_j) &= \prod_{i\in\mathcal{D}_k}\left[P_D G(\mathbf{z}_i;\mathbf{H}_k\mathbf{x}_k,\Sigma)\right] \\ &\quad \times \prod_{i\in\mathcal{U}_{k,j}}\left[1-P_D G(\mathbf{z}_i;\mathbf{H}_k\mathbf{x}_k,\Sigma)\right] \\ &= \sum_{i=0}^{d_{k,j}}(-1)^i \sum_{r=1}^{d_{k,j}(i)} \prod_{l\in\mathcal{D}_k\cup C_i^{\mathcal{U}_{k,j}}(r)} P_D G(\mathbf{z}_l;\mathbf{H}_k\mathbf{x}_k,\Sigma) \end{aligned} \tag{6}$$

where $d_{k,j} = |\mathcal{U}_{k,j}|$, $d_{k,j}(i) = \binom{d_{k,j}}{i}$ and $C_i^{\mathcal{U}_{k,j}}(r)$, $i = 0,\ldots,|\mathcal{U}_{k,j}|$, $r = 1,\ldots,d_{k,j}(i)$ is the rth collection of i integers from the set $\mathcal{U}_{k,j}$.

[1]Note, intermittent failures to receive a detection packet from a sensor at the central processor may also be due to packet loss in the network. This failure mode is not considered here as it is beyond the scope of this paper. Works such as [12], and references therein, have investigated the effect of random packet loss on estimation performance for more sophisticated sensors.

Let $\mathbf{z}_A, A \subseteq \{1,\ldots,m\}$ denote the concatenation of sensor locations with indices in the set A, then the likelihood can be written as

$$p(\mathbf{y}_k|\mathbf{x}_k,\mathbf{s}_j) = \sum_{i=0}^{d_{k,j}}(-1)^i \sum_{r=1}^{d_{k,j}(i)} P_D^{a_k+i} G\left(\mathbf{z}_{\mathcal{M}_{k,j}^{i,r}};\mathbf{H}_{a_k+i}\mathbf{x}_k,\Sigma_{a_k+i}\right) \tag{7}$$

where $a_k = |\mathcal{D}_k|$, $\mathcal{M}_{k,j}^{i,r} = \mathcal{D}_k \cup C_i^{\mathcal{U}_{k,j}}(r)$, $\mathbf{H}_j = \mathbf{1}_j \otimes \mathbf{H}$ with $\mathbf{1}_j$ denoting a column vector of j ones, $\Sigma_j = \mathbf{I}_j \otimes \Sigma$ and $\otimes$ is the Kronecker product.

2.3. Tracking Algorithm

The goal of the tracking problem is to recursively compute the posterior density of the target state and mote reliability given the measurement history, i.e. $p(\mathbf{x}_k,\mathbf{s}_j|\mathbf{y}_{1:k})$ for each $\mathbf{s}_j \in \mathcal{S}_k$ where $\mathbf{y}_{1:k}$ represents the measurements from time t_1 to time t_k. We will first derive the exact computation of the posterior at time t_k and then show how it can be approximated by a single Gaussian using a Probabilistic Data Association (PDA) approach [13]. Note, we do not assume that the actual distributions will necessarily be Gaussian, instead we believe this approach may provide an acceptable tradeoff between system assumptions and computational complexity. More detailed evaluations, including field trials using Berkeley/Crossbow motes [14, 15], are planned to confirm this.

The exact computation of the posterior density for a fixed $\mathbf{s}_j$ is given by

$$p(\mathbf{x}_k,\mathbf{s}_j|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k,\mathbf{s}_j)\,p(\mathbf{x}_k,\mathbf{s}_j|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{8}$$

Assume that the posterior density at time t_{k-1} is given by

$$p(\mathbf{x}_{k-1},\mathbf{s}_j|\mathbf{y}_{1:k-1}) = w_{k-1}^j \mathcal{N}(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1}^j,\mathbf{P}_{k-1|k-1}^j) \tag{9}$$

for each $\mathbf{s}_j \in \mathcal{S}_{k-1}$ where the weights, w_{k-1}^j, are such that $w_{k-1}^j \in [0,1]$ and $\sum_j w_{k-1}^j = 1$. It is then straightforward to show that the prior at time t_k is given by

$$\begin{aligned} p(\mathbf{x}_k,\mathbf{s}_j|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\,p(\mathbf{x}_{k-1},\mathbf{s}_j|\mathbf{y}_{1:k-1})\,d\mathbf{x}_{k-1} && (10) \\ &= w_{k-1}^j \mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1}^j,\mathbf{P}_{k|k-1}^j) && (11) \end{aligned}$$

where

$$\hat{\mathbf{x}}_{k|k-1}^j = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1}^j \tag{12}$$

$$\mathbf{P}_{k|k-1}^j = \mathbf{F}_k\mathbf{P}_{k-1|k-1}^j\mathbf{F}_k' + \mathbf{Q}_k \tag{13}$$

Substituting (7) and (11) into (8) yields, after some manipulations

$$p(\mathbf{x}_k,\mathbf{s}_j|\mathbf{y}_{1:k}) = \delta_k^{-1}\sum_{i=0}^{d_{k,j}}\sum_{r=1}^{d_{k,j}(i)} \alpha_k^{i,r,j}\mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k}^{i,r,j},\mathbf{P}_{k|k}^{i,r,j}) \tag{14}$$

where

$$\begin{aligned} \alpha_k^{i,r,j} &= w_{k-1}^j(-1)^i P_D^{a_k+i}\sqrt{\frac{|2\pi\Sigma_{a_k+i}|}{|2\pi\mathbf{S}_k^{i,j}|}} \\ &\quad \times G\left(\mathbf{z}_{\mathcal{M}_{k,j}^{i,r}};\mathbf{H}_{a_k+i}\hat{\mathbf{x}}_{k|k-1}^j,\mathbf{S}_k^{i,j}\right) \end{aligned} \tag{15}$$

$$\delta_k = \sum_j\sum_i\sum_r \alpha_k^{i,r,j} \tag{16}$$

$$\hat{\mathbf{x}}_{k|k}^{i,r,j} = \hat{\mathbf{x}}_{k|k-1}^j + \mathbf{K}_k^{i,j}\left[\mathbf{z}_{\mathcal{M}_{k,j}^{i,r}} - \mathbf{H}_{a_k+i}\hat{\mathbf{x}}_{k|k-1}^j\right] \tag{17}$$

$$\mathbf{P}_{k|k}^{i,r,j} = \left[\mathbf{I} - \mathbf{K}_k^{i,j}\mathbf{H}_{a_k+i}\right]\mathbf{P}_{k|k-1}^j \tag{18}$$

with

$$\mathbf{S}_k^{i,j} = \mathbf{H}_{a_k+i}\mathbf{P}_{k|k-1}^j\mathbf{H}'_{a_k+i} + \Sigma_{a_k+i} \tag{19}$$

$$\mathbf{K}_k^{i,j} = \mathbf{P}_{k|k-1}^j\mathbf{H}'_{a_k+i}\left(\mathbf{S}_k^{i,j}\right)^{-1} \tag{20}$$

Therefore, if at time t_{k-1} the posterior is a single Gaussian as in (9), then the posterior at time t_k is composed of a mixture of $O(|\mathcal{S}_k|2^{m-a_k})$ Gaussians, which is a function of the number of motes that did not transmit a detection and the number of possible mote scenarios. It should be noted that in the above derivation non-detections are considered from all motes. In practice, it is only necessary to consider non-detections from motes whose sensing regions overlap those of the motes that declared a detection. This significantly reduces the number of components in the mixture. Even so, the number of Gaussians increases exponentially with time, so exact computation of the posterior density is infeasible.

Note that (14) is not a Gaussian mixture in the usual sense since the weights are not constrained to the interval $[0,1]$. This prevents the use of many of the more sophisticated mixture reduction algorithms such as those described in [16]. These algorithms require the pairwise merging of suitable components. However, if the weights can be negative then there is a possibility that such merging will result in a component with a negative definite covariance matrix. This problem is avoided if all components are merged into one such that the overall mean and covariance matrix are preserved. Therefore, following the PDA approach, we approximate the posterior density for a fixed $\mathbf{s}_j$ at time t_k by a single Gaussian

$$p(\mathbf{x}_k, \mathbf{s}_j|\mathbf{y}_{1:k}) \approx w_k^j \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^j, \mathbf{P}_{k|k}^j) \tag{21}$$

where

$$\hat{\mathbf{x}}_{k|k}^j = \sum_{i=0}^{b_k}\sum_{r=1}^{d_{k,j}(i)} \gamma_k^{i,r,j}\hat{\mathbf{x}}_{k|k}^{i,r,j} \tag{22}$$

$$\mathbf{P}_{k|k}^j = \sum_{i=0}^{b_k}\sum_{r=1}^{d_{k,j}(i)} \gamma_k^{i,r,j}\left[\mathbf{P}_{k,k}^{i,r,j} + (\hat{\mathbf{x}}_{k|k}^j - \hat{\mathbf{x}}_{k|k}^{i,r,j})(\hat{\mathbf{x}}_{k|k}^j - \hat{\mathbf{x}}_{k|k}^{i,r,j})'\right] \tag{23}$$

$$\gamma_k^{i,r,j} = \alpha_k^{i,r,j} \Big/ \sum_i\sum_r \alpha_k^{i,r,j} \tag{24}$$

The updated weights are given by

$$w_k^j = \frac{\sum_i \sum_r \alpha_k^{i,r,j}}{\sum_j \sum_i \sum_r \alpha_k^{i,r,j}} \tag{25}$$

The weights, w_k^j, approximate $\mathsf{P}(\mathbf{s}_j|\mathbf{y}_{1:k})$, i.e. the posterior probability that mote scenario $\mathbf{s}_j$ is correct. From (25) the posterior probability mote i is operational is given by

$$\mathsf{P}(\text{mote } i \text{ alive}|\mathbf{y}_{1:k}) = \sum_{\substack{\mathbf{s}_j \in \mathcal{S}_k \\ s_{j,i}=1}} w_k^j \tag{26}$$

Finally, the output of the tracker at time t_k is then

$$\hat{\mathbf{x}}_{k|k} = \sum_{s_j \in \mathcal{S}_k} w_k^j \hat{\mathbf{x}}_{k|k}^j \tag{27}$$

$$\mathbf{P}_{k|k} = \sum_{s_j \in \mathcal{S}_k} w_k^j \tilde{\mathbf{P}}_{k|k}^j \tag{28}$$

where

$$\tilde{\mathbf{P}}_{k|k}^j = \mathbf{P}_{k|k}^j + \left(\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^j\right)\left(\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}^j\right)' \tag{29}$$

3. SIMULATIONS

To show the effect of estimating mote reliability on target tracking performance simulations were carried out of single target moving through a fixed field of 33 motes where up to $d_{\max} = 2$ may be dead. In this example only one mote was actually dead. This simple scenario has been used as the effect of the mote failure can be clearly distinguished. A more realistic scenario would, of course, have a larger number of sensor nodes and more inoperative motes.

Three variants of the tracking algorithm of Section 2 were used. The first variant is called the *bPDA* tracker as it believes all motes are alive. For this tracker, the only valid mote scenario is $\mathcal{S}_0 = \{s_{j,i} = 1, i = 1, \ldots, m\}$. The second variant is called the *ePDA* tracker as it estimates which motes are alive using the algorithm as described in Section 2.3. The final variant is called the *kPDA* tracker as this tracker knows which motes are alive, i.e. there is only one possible mote scenario but it is the correct one. In each case, 1000 Monte Carlo simulations were run.

The target trajectory was a straight line in two dimensions perturbed by random noise with the state at time t_k being given by $\mathbf{x}_k \in \mathbb{R}^4$, containing the position and velocity in each direction and $\mathbf{F}_k$ and $\mathbf{Q}_k$ given by

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix} \tag{30}$$

$$\mathbf{Q}_k = \mathbf{I}_2 \otimes q^2 \begin{bmatrix} \frac{\Delta t_k^3}{3} & \frac{\Delta t_k^2}{2} \\ \frac{\Delta t_k^2}{2} & \Delta t_k \end{bmatrix} \tag{31}$$

where $\Delta t_k = t_k - t_{k-1}$ and $\mathbf{I}_m$ is the $m \times m$ identity matrix.

Figure 1 shows the mote field with the true target track as the solid line and the estimated tracks from the *ePDA* and *bPDA* trackers for a single Monte Carlo run. Motes that are operational are shown by circles and the defunct motes by asterisks.

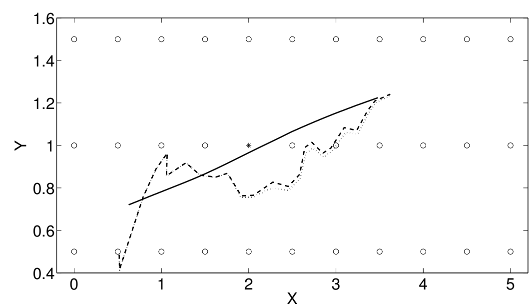

Fig. 1. Output of a single Monte Carlo run. Operational motes are shown by 'o' and defunct motes by '*'. The true target path is shown by the solid line, the *ePDA* track by the dashed line and the *bPDA* track by the dotted line.

Figure 2 shows the RMS errors in the position estimates in the y coordinate. The performance of each tracker in the position estimates in the x coordinate and the velocity estimates are similar and have been omitted due to space limitations. The target passes through the sensor region of the defunct mote, mote 14, during scans 7 to 14. It can be seen from Figure 2 that the estimates of the position in the y coordinate are biased at the times when the tracking algorithm does not know that the mote is inoperative. The performance of the *kPDA* tracker demonstrates the best results that could be obtained by this style of tracking algorithm.

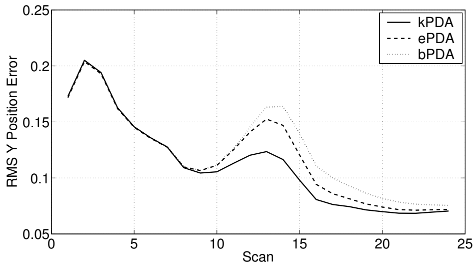

Fig. 2. RMS position errors in the *y* coordinate.

While both the *ePDA* and *bPDA* tracker exhibit bias, the *ePDA* performance is superior to that of the *bPDA*. It would be expected that the distinction between algorithms would be more pronounced when a larger number of motes are inoperative. However, even the performance shown here for the *ePDA* algorithm suggests that it is able to reliably estimate sensor health from the available data. This is illustrated in Figure 3 which shows the probability mote 14 is operational over time, as computed by the *ePDA* tracker using (26), which shows a significant drop when the target enters its sensing range. The probabilities for all other motes remained over 98% at all times.

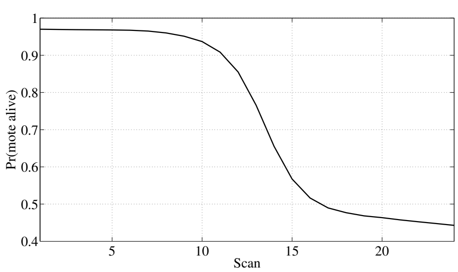

Fig. 3. Probability mote 14 is operational.

4. CONCLUSIONS

When tracking with binary proximity sensors, including information about which motes did not detect the target can improve tracking performance. However, to do this robustly requires knowing which motes are alive and which have failed as such sensors are not necessarily reliable. In this paper we have described a tracking algorithm which is robust to sensor failures as it simultaneously estimates the probability a mote is operational as well as the track state. The approach described here is being extended to incorporate more general sensor measurement models. Using this approach as a basis for scheduling mote status checks is also being investigated.

5. REFERENCES

[1] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, pp. 436–463, 2005.

[2] K. Mechitov, S. Sundresh, Y. Kwon, and G. Agha, "Co-operative tracking with binary-detection sensor networks," Tech. Rep. UIUCDCS-R-2003-2379, University of Illinois at Urbana-Champaign, September 2003.

[3] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *4th International Conference on Information Processing in Sensor Networks*, Los Angeles, CA, USA, April 2005.

[4] J. Aslam, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, November 2003.

[5] P. M. Djurić, M. Vemula, and M. F. Bugallo, "Signal processing by particle filtering for binary sensor networks," in *11th IEEE Digital Signal Processing Workshop*, Taos, NM, USA, August 2004.

[6] T. J. Stevens and D. Morrell, "Minimization of sensor usage for target tracking in a network of irregularly spaced sensors," in *IEEE Workshop on Statistical Signal Processing*, St Louis, MO, USA, September 2003.

[7] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.

[8] X. Sheng and Y.-H. Hu, "Distributed particle filters for wireless sensor network target tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA, USA, March 2005.

[9] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *4th International Conference on Information Processing in Sensor Networks*, Los Angeles, CA, USA, April 2005.

[10] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Trans. on Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[12] V. Gupta, B. Hassibi, and R. M. Murray, "On sensor fusion in the presence of packet-dropping communication channels," in *44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 3547–3552.

[13] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with Probabilistic Data Association," *Automatica*, vol. 11, pp. 451–460, 1975.

[14] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for "smart dust"," in *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, 1999, pp. 271–278.

[15] Crossbow Technology Inc., <http://www.xbow.com>.

[16] D. J. Salmond, "Mixture reduction algorithms in target tracking," in *IEE Colloquium on State Estimation in Aerospace and Tracking Algorithms*, December 1989.