On Target Tracking with Binary Proximity Sensors

Wooyoung Kim
University of California, Santa Cruz
1156 High St., Santa Cruz, CA 95064
wooyoung@soe.ucsc.edu

Kirill Mechitov, Jeung-Yoon Choi and Soo Ham
University of Illinois at Urbana-Champaign
201 N. Goodwin Ave., Urbana, IL 61801
{mechitov|choijy|sooham}@uiuc.edu

***Abstract*— We consider the use of binary proximity sensors for tracking targets. Such sensors provide only 1-bit information regarding a target's presence or absence in their vicinity, albeit with less than 100% reliability. A novel tracking method employing such binary sensors is proposed and its performance in different deployment scenarios evaluated. For a given target, the method utilizes the sensor outputs to estimate individual positions in the path of the target in the near past and finds the line which best fits the path points. This line is then used to estimate the target's current position. A performance study has been conducted through comprehensive simulation using parameters collected from a prototype deployment consisting of wireless micro-sensors with binary acoustic detectors.**

I. Introduction

Target tracking aims to detect the presence of an object and determine its path in an area of interest. This problem has been extensively studied in the field of signal processing. Continuing miniaturization of computing and (wireless) communication circuitry as well as sensor devices has morphed mass production of intelligent wireless micro-sensors at a low cost from a question of "if" to that of "when." As a result, we have witnessed a lot of research interests being drawn to target tracking as a promising application in the emerging field of wireless sensor networks (WSN) [1], [2], [3], [4].

Tracking targets with geographically dispersed, cooperating sensors is attractive for several reasons. First, it can be more robust; sensors deployed close to targets would result in more reliable signal readings. Also, it can be more cost effective. A multitude of cheap sensors may track multiple targets simultaneously without human operators in the loop. However, the necessity for cost control and ensuing miniaturization limits the sensitivity of individual sensors and consequently the quality of sensor readings. The challenge is to design a tracking method which ingeniously reconciles the two defining characteristics, abundance in quantity and inferiority in quality, to realize the desired robustness.

We study the use of wireless micro-sensors called *binary proximity sensors* in target tracking. These sensors provide only 1-bit information regarding a target's presence or absence in their vicinity. More specifically, we develop a tracking method in a two-tiered environment which consists of a self-configurable network of such sensors producing detection information and a computationally more capable node processing the information to estimate target positions. Examples of such two-tiered environments are many. A pursuer may catch an evader more quickly if it can estimate the latter's trajectory in real or nigh real time. When a restricted area is broken into, the intruders' path may be reconstructed with a postmortem analysis.

Taken individually, outputs from binary sensors contain little information. Existing tracking methods for binary sensors inadvertently subject their tracking quality to the reliability of the individual sensors by estimating the target's position at a given time using only the limited number of sensor outputs available at the time. Instead, the tracking method we develop co-opts past sensor outputs in addition to the current ones to improve the tracking resolution. The essence of the method lies in the fact that the trajectory of a target during a sufficiently small interval can often be approximated well by a straight line segment. A dynamic set of path points is maintained, which grows by the next path point which is estimated from the latest sensor outputs and shrinks if the current interval turns out to be too large. For the current set of path points, a best fitting line is sought and the current target position is computed from the line. This indirection has the positive effect of smoothing out errors inherent in outputs from simple binary sensors. In this regard, the method resembles the *moving averages* in stock price analysis which emphasize the direction of a trend and smooth out price and volume fluctuations.

After briefly reviewing research results related to target tracking with WSNs in outdoor environments, we present path-based target tracking with binary sensors in Section III, by first developing the necessary models and then describing its algorithm and heuristic weighting schemes to compute path points. Evaluation results of the method, which we have compiled from an actual prototype deployment as well as an extensive simulation study, follow in Section IV. We conclude the paper in Section V with a summary of the contributions.

II. Related Work

A string of research on tracking targets using WSNs originated from the DARPA SensIT project. Among the noteworthy results are a tracking framework for WSNs using geographical information [1], [2], [5] and information utility-based target tracking [3], [4]. The former takes a traditional approach to target tracking: it dynamically divides the region of interest based on the target's velocity and tracks multiple targets simultaneously by classifying them [1] and associating each with a particular track [2]. The latter approach attempts to select the next sensor node that most likely results in "the greatest benefit at the lowest cost" for estimation. Relatively sparse networks consisting of capable multi-modal sensors are employed in these solutions, in stark contrast to our solution which assumes a dense network of simple binary proximity sensors.

There are only a handful of research results on target tracking for binary sensor networks [6], [7]. Most closely related to our work among them is the method studied in [6]. It employs a network of sensors that report whether an object is moving toward or away from them and applies a particle filtering based algorithm using geometric properties unique to the system. Reliable sensor outputs and rather long sensing ranges are assumed for the simulation-only evaluation, suggesting that the solution may be less resilient when deployed in real environments where sensor measurements are likely noisy and sensors have a limited detection range; per contra, we study the use of less reliable binary proximity sensors with a limited sensing range. A solution to track the edge of a shadow using binary sensors that detect changes in light intensity is proposed in [7].

0-7803-9202-7/05/$20.00 (C) 2005 IEEE

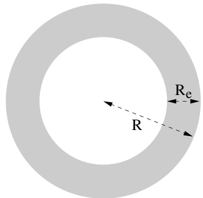


Fig. 1. A probabilistic sensor model. The chance of detection in the gray annulus decreases with distance from the sensor.

III. Tracking with Binary Sensors

In this section we develop a model of probabilistic binary sensors, an approximation model for a target's trajectory in a short time period, and a path-based tracking method based on the models, in turn. The essence of the path-based tracking lies in the way it correlates position estimates distributed in both space and time.

A. Modeling Binary-Sensor WSNs

We consider a network of N wireless binary sensors, each of which occupies a single point in a 2-D plane. Viewing sensor nodes as binary proximity sensors is attractive for target tracking in WSNs because signal detection in micro sensors is probabilistic at best, and their readings are dependent on the environment and not always reliable. Nodes may be placed randomly following a uniform distribution or they may be placed to form an equal-spaced grid. Each node contains a sensor, a radio transceiver, and computation logic and memory for local detection and classification [8]. Unlike sensors considered in traditional tracking approaches, binary sensors provide only one bit of data indicating presence or absence of a target in the sensing range. They are incapable of producing any other information, such as direction of arrival or distance to the source.

The two parameters that characterize the behavior of a sensor network are detection range and node density. We assume uniform node density and homogeneous detection ranges. The detection range of a sensor is a limit beyond which the sensor cannot distinguish a target's signal from ambient noise. Most research in this area assumes a uniform disc model for simplicity. However, detection ranges often vary widely among sensors depending on the sensitivity; fortunately much of individual variations can be compensated with careful calibration. A more critical factor is that detection ranges depend on the environmental conditions at the time of detection, such as the relative orientation of the object and the sensor. These factors make target detection near the boundary of the sensing range much less predictable.

B. Modeling Binary Proximity Sensors

The above observations give rise to a sensor model where a sensor detects presence of a target probabilistically near the boundary of its sensing range (Figure 1). The detection probability can be given as a function of the distance to the source. With the model, a sensor with a nominal sensing range R can always detect a target's presence if it is within $R - R_e$ range from the sensor. No signal from beyond distance R is ever detected. And, the detection probability drops off continuously as the distance increases between $R - R_e$ and R. We have empirically obtained this probability distribution from a prototype deployment to use in the simulation study described in Section IV-B.

In general, a sensor's detection range R varies depending on the type of a target and the signal emission strength at the time of detection. For example, a sensor may not differentiate between a far away target emitting a strong signal and a nearby target with a weak signal. Even for targets of an identical type, assuming the same R across sensors may be an optimistic proposition because sensing range is also dependent on the sensitivity and calibration of the detector. For example, a sensor may not detect a target, no matter how close it is, if it cannot distinguish the signal from ambient noise. Nonetheless, this sensing model is general enough to accommodate any sensing modality; a sensor can be analyzed using the model as long as it is capable of differentiating signals from ambient noise and has a relatively uniform (i.e., isotropic) detection range.

C. Modeling the Target Trajectory

Objects can move arbitrarily, possibly changing speed and direction at any time. Precise modeling of such arbitrary paths can prove cumbersome and unnecessarily complex for target tracking in binary-sensor WSNs. In lieu of costly precise path modeling, we adopt a piecewise linear approximation.

We allocate a small moving window to the past measurements and use a straight line segment to represent an object's trajectory in that window. Since a target's movement is governed by the laws of physics and the length of the window is determined by the network's sampling rate, a straight line segment can approximate a target's path fairly well for a short period of time, assuming that the sampling rate is reasonably high and that the target's movement is not completely erratic.

The extent to which the target's actual path diverges from its straight line approximation in a given window depends on several factors, including the target's speed and turning radius. For vehicles traveling along a highway, the difference can be very small. For a person walking along a curvy path with tight turns, the divergence may not be negligible. In either case, accuracy improves as we increase the resolution of the WSN, either by increasing the node density or by reducing the sensing range.

In addition, a target is assumed to move at a constant speed inside the window. In general, the constant velocity assumption is not a must-have requirement for object tracking. However, we need this in order to compute the target position with the path estimation. Also, it is useful for developing a heuristic weighting scheme for path point estimation discussed later in the paper.

D. Tracking with Path Estimation

A straightforward way for position estimation using binary sensors is the centroid method; it estimates a target's position at a given time as the average of the detecting sensors' positions. It is attractive for its simplicity, but it produces estimates of subpar quality (Figure 8). Instead of relying on direct methods which estimate a target's position from the detecting sensors' locations, we have adopted an indirect, two-step method. The basis of the approach remains the same as the centroid method; each detecting sensor approximates the target's position with its own location. However, the way the two-step method correlates sensor data makes it differ from the centroid method.

The method computes the weighted average of the detecting sensors' locations and uses it as an estimate of a point in the target's path. Then, it finds a line which best fits the point and other points from the recent past. (Recall that a target is assumed to have a straight line trajectory in the short term.) Finally, using the velocity estimate and the line equation, the target's position at the time is estimated. As evident from the evaluation results given in Section IV-B, indirection through path estimation tends to smooth out errors in the position estimates. The method is summarized in Procedure 1.

Procedure 1 Path-based target tracking

1: Each sensor node records detection durations.
2: Triples (time, node location, duration) from the sensor nodes are aggregated at the tracking node.
{Then, the tracking node computes}
3: Weights for the detecting nodes and then the current path point.
4: A line which best fits the path points within the given window.
5: The target's position using the velocity and the line estimate.

A number of variations may emerge from this method skeleton depending on the weighting scheme. In fact, the performance of the path-based tracking is closely related to the chosen weighting scheme. The simplest scheme is to weight sensors uniformly, which effectively puts a target at the center of mass of the detecting sensors. However, it fails to take advantage of additional information such as the sensor's closeness to the target. We develop dynamic, history-based weighting schemes with a trade-off between computational cost and accuracy. The schemes are based on a simple relation between a target's proximity to a sensor and the sensor's detection duration.

E. Distance-based Weighting

We seek to weight sensors in a manner that assigns a higher value to sensors close to the target. Consider a target moving in a sensor's range of detection (Fig. 2 (a)). Observe that a target generally stays longer in the sensing range of a sensor it passes closer than one farther [1]. Thus, a quantity related to the duration for which the sensor has tracked the target would make a good candidate for such weights.

The detection duration gives an estimate of the distance the target has traveled inside the sensing range of the sensor. For simplicity, assume a sensor's detection range is a perfect circle, *i.e.*, $R_e = 0$. For a target moving at a constant speed v, the distance $d = vt$ when the detection duration is t. We can relate the distance traveled within the sensor range to the current distance between the sensor and the object in several ways.

1) Naive or Pessimistic: This method assumes the target is on the boundary of the detecting sensor's sensing range at the time of detection and is about to move out of the range. Thus, it uses $1/R$ as the weight. The use of the successive refinement described in Section III-F makes it differ from the "uniform" weighting method.

2) Expected Distance: Since a target can enter a sensor's sensing range from any point at any angle, it can be anywhere in the annulus defined by the outer circle of radius R and the inner circle of radius d. Without loss of generality, assume that the target enters the range at (R, 0) (Figure 2 (a)). Then, given d, the relative direction angle α is bounded by α_{max} from above, which is $\cos^{-1}(d/2R)$ (Figure 2 (b)). The equation of the arc is given as

$$r = \sqrt{R^2 - 2Rd\cos\alpha + d^2},\ 0 \leq d \leq 2R,\ |\alpha| \leq \cos^{-1}\frac{d}{2R}$$

$$\alpha = \cos^{-1}(\frac{R^2 + d^2 - r^2}{2Rd})$$

The cumulative distribution function $F(r) = P(X \leq r)$ of random variable X indicating the probability that the distance to the target is less than or equal to r can be found by integrating the probability that the target enters the range with a certain direction angle over the region where α corresponds to the distance less than or equal to r.

[1]If the sensing range is circular (or isotropic).

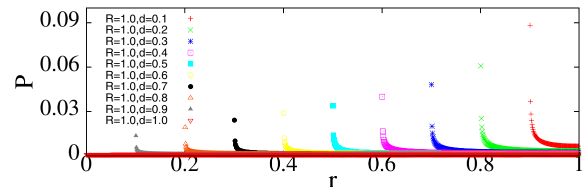


Fig. 3. Plots of $P(x \leq r < x + 0.001), x = (R - d) + 0.001i, i = 0, 1, ...$ for different d's.

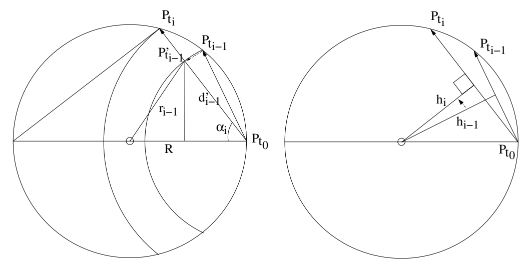


Fig. 4. Successive refinement

$$\begin{aligned} F(r) &= P(X \leq r) = \int_{-\alpha}^{\alpha} p(t)dt = 2\alpha\frac{1}{2\alpha_{max}} \\ &= \cos^{-1}\frac{R^2 + d^2 - r^2}{2Rd} / \cos^{-1}\frac{d}{2R} \end{aligned}$$

Figure 3 shows the plots of probability distribution of $P(x \leq r < x + 0.001)$. Notice the presence of an impulse at the minimum of r. From the relation between d and $F(r)$ we compute the expected distance to the target as

$$r_e = \int_{r_{\min}}^{r_{\max}} r \cdot F_d(r)dr, r_{\min} = R - d, r_{\max} = R. \quad (1)$$

and use $\frac{1}{r_e}$ as the weight.

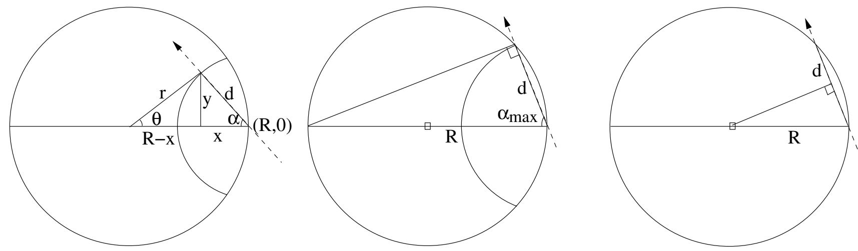
3) Distance to the Path: With the assumption that the current detection is the last (thus, *pessimistic*), we compute the distance to the path $h = \sqrt{R^2 - (d_i/2)^2}$ and use $\frac{1}{h}$ as the weight (Figure 2 (c)).

F. Successive Refinement

Given a target's straight line trajectory, the closer it passes to a sensor, the longer it stays in that sensor's detection range. The exact duration can only be known after the target moves beyond the sensor's range. This is the basis of the pessimistic assumption that the current detection is the last one in computing the weights. Consequently, duration measurements from an active sensor are most likely underestimates; another detection by the sensor rectifies the target has passed the sensor closer than estimated. Recall that we estimate points on the target's path and find a line equation to compute the current target position. The essential idea of successive refinement is to re-compute the past weights using the latest duration information to estimate the path points more precisely.

Pessimistic/Expected Distance Let d_i be the length of secant $\overline{P_{t_0}P_{t_i}}$ in Figure 4 (a). Since $\cos\alpha_i = \frac{d_i}{2R}$, for $0 < k < i$, $r'^2_k = R^2 + d_k^2 - 2Rd_k\cos\alpha_i = R^2 + d_k^2 - d_k d_i$. For the Pessimistic scheme, the new weight at t_k is $\frac{1}{r'_k}$. For the Expected Distance scheme, use $r^{max} = r'_k$ in Eq. 1 to compute the expected distance at P'_k.

Distance to the Path The distance to the path at P_i is $h_i^2 = R^2 - (\frac{d_i}{2})^2$ and it is also the distance to the path at P'_k, $0 < k < i$ (Figure 4 (b)).



(a) target position after traveling d with entrance angle α (b) maximum entrance angle α_{max} given distance d (c) distance to the path

Fig. 2. After traveling distance d within sensor range, the object will be located on the arc with radius d. The distance r from the object to the sensor ranges from 0 to R, depending on angle α.

IV. Evaluation

We have evaluated the path-based method and compared its performance under different conditions via an extensive simulation study. To make evaluation results more convincing and useful, we have taken a realism-based approach where relevant parameters are collected from actual devices and fed into subsequent simulation runs. First, we have implemented a proof-of-concept acoustic tracking prototype using MICA-2 motes and experimentally validated its performance. Microphones and tone detectors in standard MICA sensor boards are used as acoustic sensors. Although some acoustic sensors are capable of providing such information as signal strength and direction of arrival of the sound, we assume that the tone detectors do not provide such information, treating them as binary sensors. Then, we have instrumented the simulated sensors with the parameters we have collected from the prototype and compared the tracking performance under different configurations and conditions. For the simulation study, we assume that the nodes in a network are localized *a priori*, the clocks of the nodes are periodically synchronized, and an appropriate MAC protocol and a data aggregation scheme are in place, since issues such as routing and networking are not a primary focus of this study. Line fitting needed in path estimation is implemented using the GNU Scientific Library (GSL) [9].

A. Target Tracking with Binary Acoustic Sensors

To verify the viability of path-based object tracking and to collect relevant parameters to model sensors in the subsequent simulation study, we have implemented the tracking method on a small-scale network of MICA-2 motes [10] and conducted an experiment using acoustic sensors. We chose to use acoustic sensors because sound signals have a uniform signal attenuation model even in the presence of obstacles, and are omni-directional, especially in a low frequency range. An MICA-2 mote has limited network bandwidth (38.4Kbps maximum raw data rate) and data memory (only 4KB is available for both the application and the OS). Each MICA-2 mote is fitted with an MTS310 sensor board, of which only the tone detector is used. The tone detector on the MICA sensor board detects a specific range of sound frequencies and provides binary output indicating the presence or absence of a signal. The sensitivity of the microphone is rather poor; a low-power sound source must be fairly close to the sensor, otherwise the microphone cannot distinguish the sound from ambient noise. Even with a nearby source, the chance of detection by the device is probabilistic at best.

The network consists of 25 motes, laid out in a regular 5×5 grid (Figure 5). Nodes are placed 0.4 m (1 u) apart, and the gain on the microphone is adjusted to provide a reliable detection range of R = 0.6 m (1.5 u). To simulate a localization service, each sensor is given its physical coordinates *a priori*. A radio-controlled car equipped with a mote plays the role of a target. It moves through the network at the approximately constant speed of 0.2 m/sec which amounts to 0.5 u/sec. It should be noted that the boost power of each microphone as well as the speed of the object were significantly reduced in the laboratory experiment. With proper calibration, the same hardware can cover a wider area and track faster moving objects.

The results of this experiment [2] are shown in Figure 6. No successive refinement is employed in this set of experiments. No ambient noise is artificially introduced, but echoes are abundant. The distance-to-the-path scheme was used due to the lack of hardware

Fig. 5. Acoustic tracking experiment with a 5×5 grid layout.

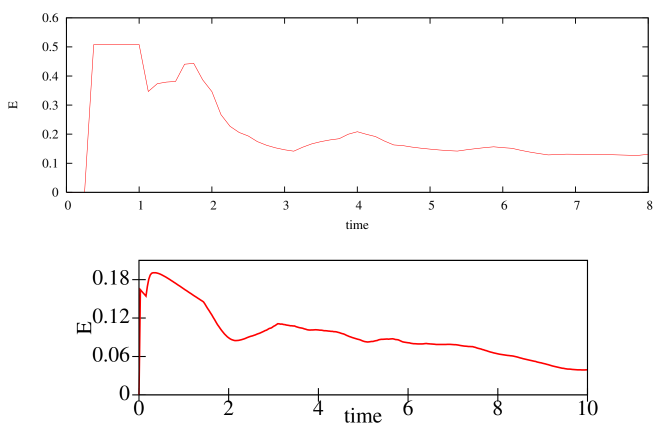


Fig. 6. Tracking results with MICA-2 acoustic sensors (top) and the corresponding simulation (bottom). The following simulation parameters are used: grid separation distance 0.4 m = 1.0 unit, sensing range R = 1.5 unit, R_e = 0.2R, and target velocity v=0.5 unit/sec.

[2]Throughout the paper, tracking error E represents the distance between an actual target location and the corresponding estimated location measured in a respective basis unit.

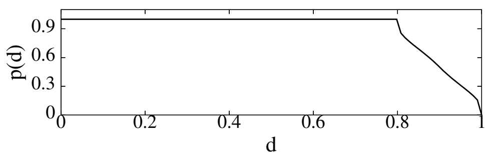


Fig. 7. Plot of detection probabilities for d when R = 1.0 u and R_e = 0.1R. d is the distance to a target.

support for floating point operations in MICA-2 motes. We observe the presence of significant errors at the initial period, which is attributed to the lack of reliable detection and velocity data; as more detections are recorded, errors quickly settle at a lower level.

Also shown in Figure 6 are the results from a simulation run with the sensor model proposed in Section III-B and with the parameters collected from the experiment (*e.g.*, R_e = 0.2R). The two results show comparable behaviors, although the magnitude of the errors is markedly higher in the actual experiment than in the simulation. This discrepancy can be attributed in part to the quality of the sensors. In the simulation, we assume ideal sensors which always detect a target within range $R - R_e$, and with diminishing probability beyond that range. Actual sensors are likely to have somewhat anisotropic detection ranges. Furthermore, echoes could have caused false detections. Tone detectors in MICA-2 are extremely unreliable; they sometimes miss presence of a sound signal emitting from a nearby source.

B. Evaluation with Simulation

Given the encouraging experimental results, we have conducted an extensive simulation study and evaluated the performance of the path-based tracking method with the three weighting schemes.

1) Simulation Setup: A 900-sensor network with a single mobile object is simulated. Unless otherwise noted, results are obtained using 30x30 grid placement with the origin at the center. The target travels from the origin along the direction of a vector (20,11) at the speed of 1.2 u/sec. The successive refinement is employed in all the simulation results reported in the section.

From the 25-node prototype experiment, we have observed that R_e = 0.2R (refer to Section III-B). Also, we note that detection rates drop stiffly around the edges in the outer gray ring in Figure 1 and decrease gradually in between. To incorporate these observations into the model, we define the detection probability at distance d, $P(d)$, to be:

$$\begin{array}{rll} 1 & \text{if} & d < R - R_e \\ 1 - 0.5e^{-\mathrm{erf}^{-1}(-f(d))} & \text{if} & R - R_e \le d \le R - R_e/2 \\ 0.5e^{-\mathrm{erf}^{-1}(f(d))} & \text{if} & R - R_e/2 < d \le R \\ 0 & \text{if} & R < d \end{array}$$

where $\mathrm{erf}^{-1}(z)$ is the inverse of the "error function" encountered in integrating the normal distribution and $f(d) = \frac{2}{R_e}d + (1 - \frac{2R}{R_e})$. Figure 7 plots the detection probabilities as a function of distance to the target d when R = 1.0u and R_e = 0.2R. All the simulation experiments performed have used a window of maximum 10 seconds for path estimation.

2) Baseline Performance: Figure 8 shows position estimation errors over time for the centroid method which computes the average of detecting sensors' locations as a target location. The plot labeled "indirect" has been obtained by first estimating path points using the centroid method, and then estimating the path using these points. Figure 9 has the theoretical best performance for the indirect tracking methods based on path estimation. The dist2target plot shows the estimation errors when the actual distance to the target at the time of detection is used as the weight. The dist2path plot is obtained when the actual distance to the path of the target is used as the weight at the detection time. Both have large initial errors but quickly converge close to the actual positions. The results suggest the validity of the weighting schemes proposed above.

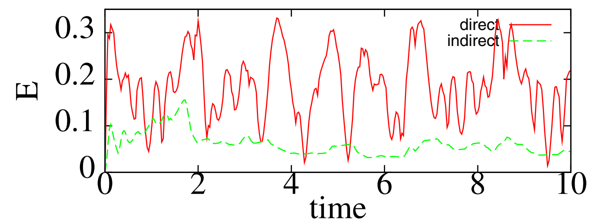


Fig. 8. Performance of centroid methods.

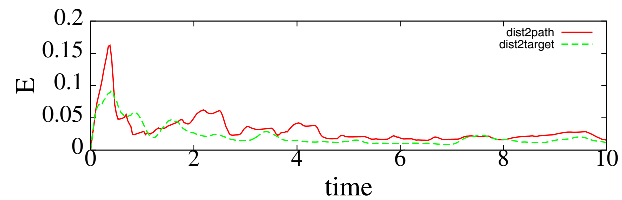


Fig. 9. The best possible performance of the distance-to-target and distance-to-path weighting schemes.

3) Performance of Different Weighting Schemes: First we study performance of the three weighting schemes in the path-based target tracking using the grid node placement noted in Section IV-B.1. Figure 10 plots changes in magnitudes of estimation errors over time for the three schemes. We observe that the magnitudes decrease in all three weighting schemes as more data become available. These results show that the path-based tracking performs quite well. The Expected Distance schemes achieves the best results among the three, but it is also most computationally expensive to run. Distance-to-the-path takes a moderate amount of computation to yield good results. We attribute the good performance to the facts that indirection through path estimation using more accurate past estimates smooths out errors in individual path points near the target's current position and compensates for scarcity of information in binary readings.

4) Effects of Sensor Placement: Next, we investigate the effect of sensor placement on the path-based tracking. This is significant because comparable performance of a random deployment would pave way for economic deployment methods such as air-dropping. We conduct the simulation with three different deployments, with increasing regularity. To our disappointment, increased irregularity turns out to be quite detrimental to estimation acccuracy (Figure 11).

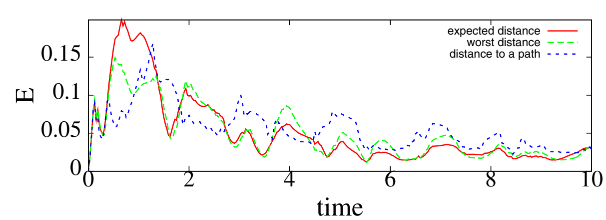
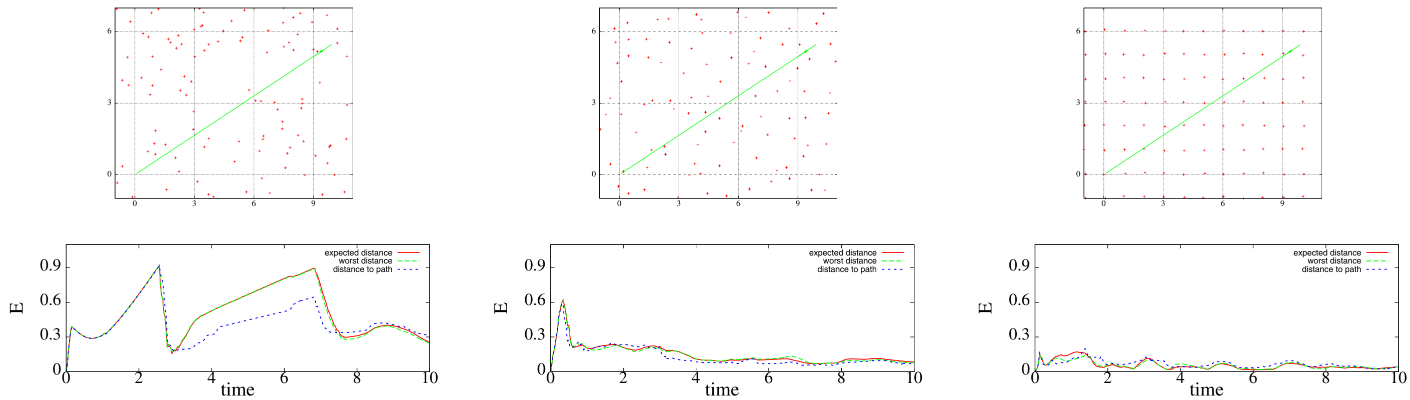


Fig. 10. Position estimation errors of different weighting schemes. With 30x30 grid placement. Grid separation $1.0u$, $N = 900$, $R = 1.0u$, $R_e = 0.2R$. Average of 100 simulation runs.



(a) random placement

(b) grid placement with random displacement up to 0.1R

(c) grid placement with random displacement up to 0.01R

Fig. 11. Three different placements and the respective tracking performance. Density = 1.0, N = 900, R = 1.0 u, R_e = 0.2R. Average of 100 simulation runs.

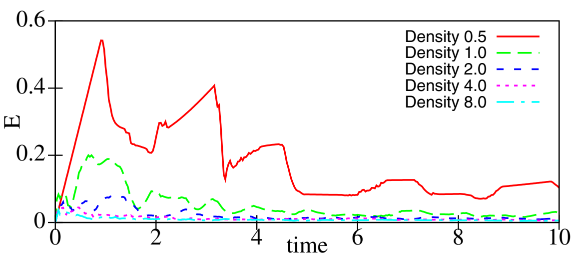


Fig. 12. A higher sensor density results in better tracking accuracy. Notice the diminishing return on increasing density; the improvement becomes marginal for densities beyond 2.0.

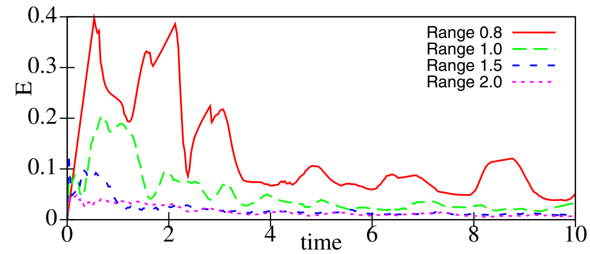


Fig. 13. A larger sensing range also results in better accuracy.

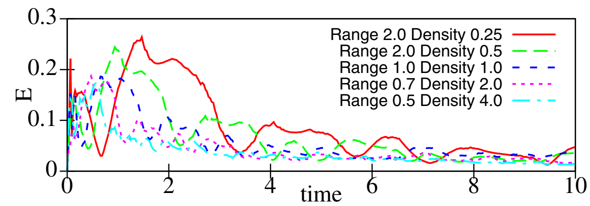


Fig. 14. Node density versus sensing range.

The placement strategy we use creates deployment instances that, while uniform as a whole, have significant non-uniformities in density at the local level – some pockets have more sensors than others. Even small displacement from grid positions results in significant [3] errors when the sensing radius is small. This is due to the use of weighted average in computing path points, because a weighted average tends to move toward a dense area and away from sparse regions. Regardless of the network size, with a small sensing range (or with a low density), only a small number of sensors detect the target at any one time, and thus even a slight local imbalance in density can result in skewed estimates. This suggests that networks with longer detection ranges or higher densities will be more resilient to local variations in sensor density; as more sensors are involved in detection, each sensor's share of contribution gets smaller, making the estimate robust against non-uniform distribution of density at the local level.

5) Effects of Node Density and Sensing Range: Intuitively, we expect a sensor network with a higher node density and a larger sensing range would yield better tracking results. The intersection of the sensors' detection areas which defines the area of uncertainty about the target's position shrinks, as either the density or the range increases. A series of simulation runs has confirmed that it is the case for binary proximity sensors (Figure 12 & 13).

Both node density and sensing range share a trait that the tracking quality improves with their values, but only with a diminishing return.

[3]In the order of average separation distance between the sensors.

To find an answer to the question of "which has a higher return on investment," we have conducted a simulation study while keeping $R_i^2 d_i$ constant (R_i: sensing range, d_i: density). The results (see Figure 14) indicate that the combination of a higher density and a smaller range is more effective than that of a lower density and a larger range. This finding is particularly encouraging since, with a smaller range, a target's trajectory can be more closely approximated to a straight line, snapping the target's position to the detecting sensor' coordinates becomes more accurate, and the computed weight is more reliable. Meanwhile, a higher density ensures more sensors detect the target at any time, guarding against errors due to reduced coverage. Also, increased density means estimates of path points become more resistant to density variation at the local level when non-uniform placement is used.

C. Adaptive Path-based Target Tracking

The path-based tracking is very effective for tracking targets with a straight line or near-linear trajectory. The results may well be

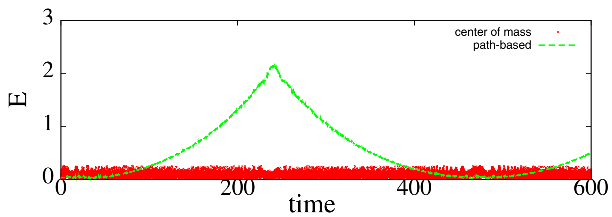


Fig. 15. Estimation errors for a target moving along a circle of curvature κ=1/300. Baseline path-based tracking vs. centroid.

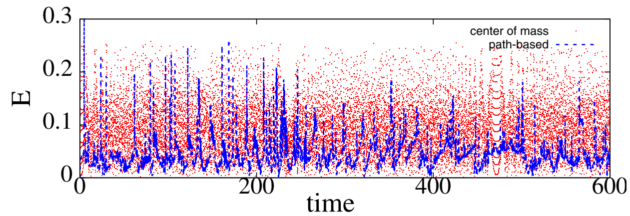


Fig. 16. Estimation errors for the same target as in Figure 15. The adaptive version vs. the centroid method.

comparable for targets moving along a line with a small curvature. But, it is hardly conceivable that this would work well for targets with a trajectory of a high curvature (see Figure 15). To determine if more robust path-based tracking is feasible, we have extended the baseline method so that the window size for past detections can be dynamically adjusted (Procudure 2). Two important parameters are when to forget (or when to stop forgetting) these points (*i.e.*, is_good(p)) and how quickly to forget (*i.e.*, $\Delta\omega$). For the experiments, we used a version of is_good(p) implemented using overlapping ranges of detecting sensors and $\Delta\omega$ = 1. Note that $\Delta\omega$ can also be adjusted dynamically depending on the target's movement characteristics. The centroid method is used as the "fallback" method. Figure 16 compares the adaptive version and the centroid method for the same target in Figure 15. The adaptive version fares a little better, and its results can be further improved by employing a more elaborated fallback method. Figure 17 shows the performance of the adaptive version for targets with different trajectories and different velocities. The first (a) is for a fast moving target (*i.e.*, 72 km/h) which changes its direction steadily and smoothly. Note that as the curvature of the target's trajectory increases, the performance of the adaptive version degenerates to that of the fallback method used (in this case, the centroid method). The second (b) is for a slow moving target which changes its direction occasionally but abruptly. We see that the adaptive version loses track of the target briefly but recovers quickly when the target changes its direction. We believe the results demonstrate viability of path-based tracking, given that sensors only provide binary proximity information that is not always reliable.

Procedure 2 Adaptive path-based target tracking

```
1:  – 2: same as Procedure 1. {Let ω be the current window.}
3:  ω := (ω < ω_min)?ω_min:((ω ≥ ω_max)?ω_max : ω)
4:  loop
5:      Find a line that best fits the path points within w.
6:      Use the line to estimate the target's position p.
7:      if (is_good(p)) then ω = ω + 1; break; end if
8:      if (ω ≤ 0) then use a fall-back for p; break; end if
9:      ω := ω − Δω
10: end loop
```

D. Discussion

Errors in path-based tracking with binary proximity sensors come from multiple sources. Some are determined at the deployment time – e.g., placement and localization errors. Others are more dynamic; for example, sampling signals at discrete intervals makes the first and last detections of a sensor occur away from the exact boundary of the sensing range. These errors can be managed with some additional cost. On the other hand, some errors are intrinsic to the sensors considered in this study. For example, the exact distance from a sensor to the target cannot be determined, and the distance must be estimated from noisy information. Also, detecting the presence of a target in its vicinity is probabilistic. We have not attempted to take into consideration all the sources of the errors in the simulation study. Instead, we have made reasonable assumptions at various levels where appropriate. Although they limit the applicability of the results to more general settings, such assumptions help simplify the deployment scenarios under consideration.

The most critical limitation of the simulation study stems from the constant velocity assumption. Currently, work is under way to incorporate variable velocity estimation into the simulator. In addition, three extensions with increasing complexity are plausible. The first immediate extension is to adapt the method for n-bit proximity sensors which provide not only presence/absence of an object but also an approximate distance to the object. Second, the path-based tracking can be augmented with selective activation to conserve energy and lengthen the mean time between failures. Lastly, the method can be extended to track multiple targets simultaneously, provided that sensors can distinguish individual targets from one another. Simple binary proximity sensors cannot tell how many targets are within their sensing range if the targets are geographically close.

V. Conclusion

This paper proposes a tracking method for use in networks of binary proximity sensors. The method finds a straight line which approximates the path of a target during a short period of time, and uses the line to estimate the target's current position. Two design parameters of the method, sensing range and node density, have been evaluated through a combination of a prototypical implementation with acoustic sensors and an extensive simulation study. The results suggest that the path-based tracking is effective for binary sensor networks with a small sensing range and a high node density. Such networks are envisioned to be the next generation of wireless sensor networks.

VI. Acknowledgments

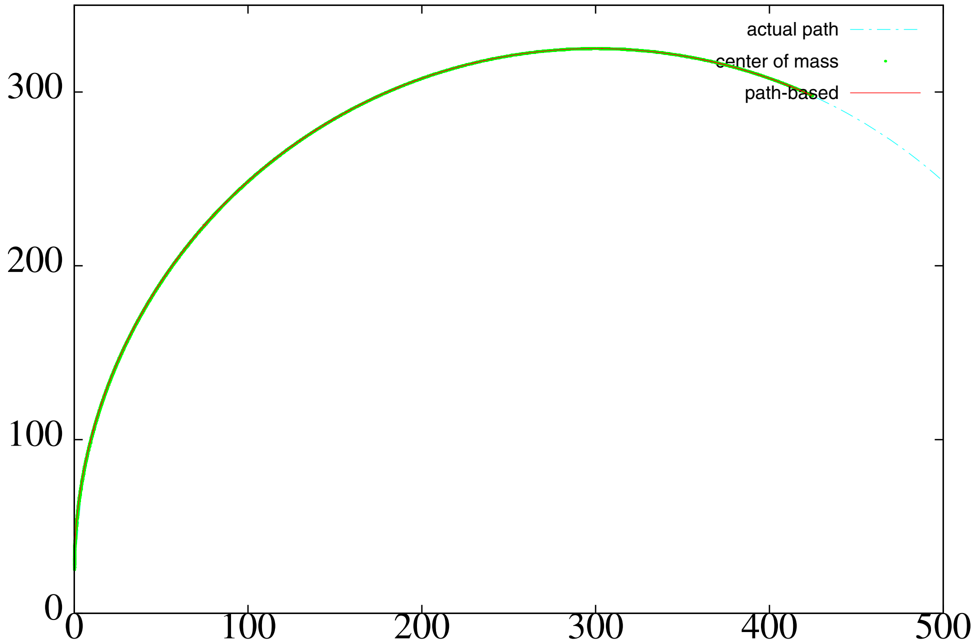
We thank Daniela Rus at MIT and the anonymous reviewers of the paper for their invaluable comments and advice. The work was supported in part by the Defense Advanced Research Projects Agency under Award and No F30602-00-2-0586 and F33615-01-C-1097.

References

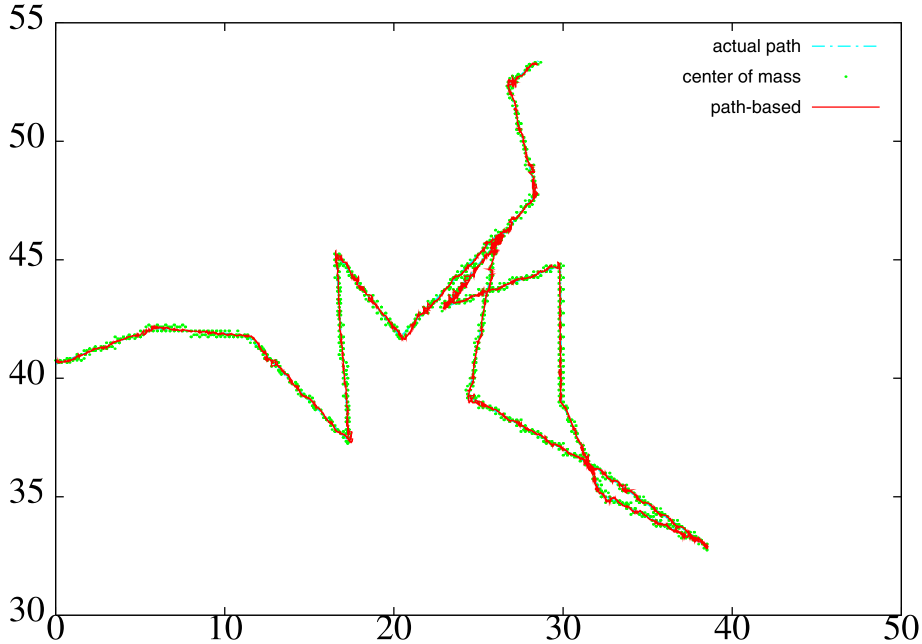
[1] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, Classification and Tracking of Targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, March 2002.

[2] R. R. Brooks, C. Griffin, and D. S. Friedlander, "Self-Organized Distributed Sensor Network Entity Tracking," *The International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 207–219, 2002.

[3] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Target Tracking," *IEEE Signal Processing Magazine*, vol. 19, no. 2, March 2002.



(a) An object moving along a circle with curvature κ=1/300. R: 0.5 u. D: 0.5 u. E: 0.2R. F: 32 Hz. V: 1.0 u/sec. T: 600 sec.



(b) An object which moves along a straight line but randomly changes its direction. R: 0.5 u. D: 0.5 u. E: 0.2R. F: 32 Hz. V: 0.1 u/sec. T: 1000 sec.

Fig. 17. Tracking results with the adaptive path-based tracking method. 1 u = 20 m. R: sensing range. D: distance between two adjacent sensors. E: R_e in Fig. 1. F: sampling frequency. V: velocity. T: simulation time.

[4] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," in *Proc. of IPSN'03*, 2003.

[5] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks," in *Proceedings of the IEEE*, 2003.

[6] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a Moving Object with a Binary Sensor Network," in *Proc. of SenSys'03*, 2003.

[7] J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks," in *Proc. of WSNA'02*, 2002.

[8] A. Arora et al., "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification and Tracking," Ohio State University, Tech. Rep. OSU-CISRC-12/03-TR71, 2003.

[9] http://www.gnu.org/software/gsl/.

[10] Crossbow, Inc., http://www.xbow.com/.