Opportunistic Processing and Query of Motion Trajectories in Wireless Sensor Networks

Dengpan Zhou*   Jie Gao*

* Department of Computer Science, Stony Brook University. {dpzhou, jgao}@cs.sunysb.edu

Abstract—**We study the problem of in-network processing and queries of trajectories of moving targets in a sensor network. The main idea is to exploit the spatial coherence of target trajectories for opportunistic information dissemination with no or small extra communication cost, as well as for efficient probabilistic queries searching for a given target signature in a real-time manner. Sensors near a moving target are waken up to record information about this target and take the communication opportunities to exchange their knowledge with preceding and descending sensor nodes along the trajectory. Thus a moving target's information is naturally detected, recorded, and disseminated along its trajectory, as well as the motion trajectories that enter the sensor field afterwards.**

We analyzed and through simulations tested the dissemination cost and query success rate for randomly generated data sets. Trajectories of reasonable length can be discovered by probabilistic in-network queries with high probability. Compared with the scheme without opportunistic dissemination, the in-network processing of trajectories, with modest cost on dissemination, allows substantially reduced query cost and delay.

I. INTRODUCTION

The development of wireless sensor networks has enabled the possibility of large-scale and long-term environmental monitoring. One driving application for such technologies is the tracking of targets with embedded sensor nodes in an unobtrusive manner, with the targets being rare animals (e.g., Petrel birds on great duck island [18], [30]) in habitat monitoring, or vehicles/humans in security applications. There has been a lot of prior work on tracking moving targets by distributed sensors (see [10], [14], [27], [31] and references therein). A few tracking systems have also been deployed and evaluated in real testbeds [2], [11], [26].

A closely related problem for target tracking is the design of the interface with which the target trajectories are accessed by the users. In habitat monitoring, the most adopted approach has the sensors that detect a target record the detection event in the data logger or report it to a base station, where the target trajectories are assembled from the individual detections for post-experiment analysis. The deployment of such a monitoring network is often one-time experiment with the goal of collecting a sufficient amount of scientific data.

In this paper we are mainly motivated by the deployment of sensor nodes for long-term monitoring in a civil environment with the goal of creating an intelligent environment to support real-time sensing, situation understanding and knowledge extraction. The detected data needs to be processed in a timely manner, and accessed by users residing in the same physical space, sometimes with stringent delay requirements.
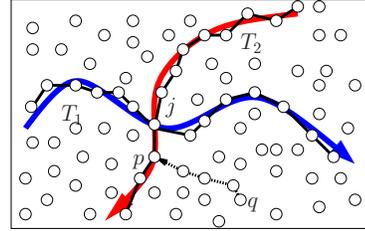

Fig. 1. Opportunistic information dissemination: sensors waken up by a moving target will record this detected event and help to disseminate information about other trajectories they have learned so far to the descending sensor nodes. In this case, target T_2 enters the sensor field after target T_1. The nodes in the trajectory T_2 after the junction node j will learn the information of both T_1 and T_2. A query message from q that visits one such node (p) is able to discover T_1 as well.

The approach of reporting detection events to a base station in this case has not fully explored the potential of collaborative information processing enabled by a large number of networked sensor nodes. Further, storing all the data at a central server has the problem that the central server represents a single point of failure, and the communication cost for trajectory reporting for each individual detection can be high when the number of moving targets is large.

Our approach. In this paper we explore the potential of an orthogonal approach and employ a low-cost processing and storage scheme for target trajectories in the network, as well as an in-network query algorithm to retrieve the detected trajectories. As a target moves, nearby sensors are triggered to record the signal signature (e.g., visual or acoustic signatures) and the time stamp when the target was detected. Users live in the same physical space and may inject queries to the network searching for the existence of a particular target and its recent moving trails:

- *Existence query:* was there a red-color truck moving in the field in the past half an hour?
- *Trajectory report query:* Report the trajectory of the red-color truck if it ever appeared in the past half an hour.

The problem we study in this paper involves how to store and process the moving trajectories and how to answer queries efficiently with low delay for these moving targets.

We exploit two naturally available opportunities to help with efficient in-network trajectory query: (i) the continuity of the motion trajectory suggests that the detection of one sensor on the trajectory allows us to locally follow the trajectory and discover the entire path; (ii) the hand-over of nearby sensors that detect the same target provides communication opportunities to exchange their respective knowledge of previously

detected targets. Thus, as a motion trajectory for a given target 'ages', i.e., as more motion trajectories are detected afterwards, this target trajectory becomes easier to find as its information is possibly widely disseminated along the trajectories that newly develop. See the example in Figure 1. The trajectory T_2 arrives later than trajectory T_1. As a sensor near T_2 is waken up anyway to record the signal signature of T_2, it also communicates with the nearby precedent and descent sensors for trajectory hand-over, and exchanges the signal signatures of target trajectories each has learned so far. In particular, after the junction node j located near both trajectories is triggered by T_2, the information j knows (in this case both T_1 and T_2) will be carried along the nodes near trajectory T_2 afterwards[1]. We call this *opportunistic information dissemination.*

From a sensor node's point of view, its task on information storage and processing is extremely simple. When it is waken up by a target in its neighborhood it will be activated to record the information of this particular target, as well as the information carried over by the precedent sensor node along this trajectory. As time goes by the information a target knows will accumulate. As trajectories that are too old are becoming increasingly not interesting in many applications, we can also associate an expiration time for each target trajectory. Trajectories that are too old will be removed from the storage space of a node and a node only keep the K newest trajectories sorted by their time stamps.

To query for a target with a given acoustic signature, we again exploit the spatial and temporal coherence of the motion trajectories. The query algorithm is probabilistic and initiates query messages, each traveling along a random direction from the query node (similar to rumor routing [4] and implemented by greedy geographical routing, e.g., in [15], [20]), until one of them has discovered a node with information about the given target trajectory, or when the number of query trials exceeds a threshold, the trajectory in question is either not existing, too short or too new. In general, the number of query messages issued depends on the length of the trajectory to be searched (a trajectory that cuts corners is unlikely to be discovered by a random query path), and its age (defined as the number of trajectories that enter the sensor field afterwards; as more trajectories develop, the chance to be discovered increases).

To summarize, both the target detection and target query are locally processed in the network and do not assume a central server. Unlike the approach of gathering trajectory knowledge at a central server in which the communication cost increases with the number of targets, here more target trajectories will provide more opportunities and reduce the cost for trajectory queries. This makes the in-network processing and query scheme attractive in the scenarios with 'heavy traffic' and those when real-time access of the trajectories are desirable.

In this paper we analyzed the probability that a random query discovers a given trajectory, with respect to its length

and age (under opportunistic information dissemination). We also evaluated the performance of the system, in terms of the query success rate and communication cost, as well as storage requirement. The results show the effectiveness of opportunistic information dissemination. A trajectory with a reasonable length/appearance can be discovered on average with only a couple of queries as long as there have been a few trajectories that helped to disseminate it.

Related work. Data storage and retrieval in a distributed sensor network is a major architectural component and has attracted a lot of interest in recent years. Existing data discovery approaches can be classified as data-centric routing [1], [12], [13], [17] and data-centric storage [22]. In data-centric routing, limited preprocessing is undertaken and the query actively searches in the network for relevant data. Data is aggregated along the routing paths back to the user to reduce communication cost [5], [21], [25]. In data-centric storage, the discovered data is preprocessed and stored in the distributed sensors. Partial aggregates can also be evaluated to facilitate complex multi-dimensional queries [7]–[9], [24].

Our scheme belongs to the data-centric storage category. The major difference in this paper is to exploit the spatial structure in the data itself – for both efficiently querying the trajectory and for helping to disseminate knowledge around without incurring additional communication costs.

The query algorithm is in spirit similar with rumor routing [4], and double rulings schemes [6], [16], [23], [29]. In double rulings scheme, a data source (i.e., the sensor node with detected information) will take some specially designed routing paths to disseminate the data availability information. A query from a node will take another special routing path searching for the available data indices. In our setting information about the data source is naturally carried by the target itself and thus the dissemination stage is simple and implicit. The rigorous analysis of how information is disseminated in the network will also apply to rumor routing and supplement the experimental study in [4].

Remotely related to the work here, the idea of exploiting existing communication opportunities also appears in other settings such as opportunistic routing to improve network throughput [3].

In the remaining of the paper we first present the analytical results in calculating the query success probability under opportunistic information dissemination. The analysis is complemented by simulations in section III.

II. TRAJECTORY QUERIES

We have a sensor field in which the sensors are used to track moving targets. Each target possibly enters and exits anywhere in the field. A sensor near the motion trajectory is waken up, say, by a high acoustic signal in its vicinity. These sensors then record a signal signature specifying the characteristics of the target detected. Naturally all the sensors near or on the trajectories detect the target and store the target signature. By movement continuity, the sensors are laid out along a curve connecting the entrance and exit of the target in the sensor

---

[1]When the trajectory T_2 exits the region of interest, we may also disseminate along the trajectory T_2 backward the information accumulated all time along to aid trajectory query with only a multiplicative factor 2 on the total communication cost.

field. We explore schemes on how these trajectories are stored and pre-processed and how a query node can efficiently extract a target trajectory given its signature.

A. Probabilistic queries

We first analyze in-network query without opportunistic information dissemination. When there is no preprocessing, only the sensor nodes on the trajectory have knowledge about the target. Thus the problem is to figure out one of these sensor nodes (for existence query), and possibly follow the trajectory (for trajectory report query). We explore a probabilistic query scheme in which the query node sends a query message along a random line, hoping that this message will hit one node on the trajectory of interest. In particular, for a line ℓ through the query point q, two query messages are sent out traveling in opposite directions, one along each ray from q. A query message will return when it hits a node on the trajectory or when it hits the sensor network boundary, in which case the query returns and does not discover the trajectory.

The query node can send out multiple such random queries to increase the chance that one of them discovers the data. The probability that a random query finds out the given trajectory, as well as the average number of query messages needed to eventually discover it with a sufficiently high probability, depend greatly on the specific shape of the trajectory. If a trajectory cuts the corner of the sensor field, i.e., the exit node is very close to the entrance node on the sensor field boundary, it is more difficult to discover it.

In the following analysis we start with a simple case when the target moves along a straight line and always enters and exits the sensor field at some boundary nodes, i.e., we assume no target appears or disappears inside the sensor field. Then we move on to the case of an arbitrary trajectory.

1) Query on straight-line trajectories: We analyze the cost of a query initiated at a node distributed uniformly randomly in the sensor field. This scenario captures the *average-case* query cost for all possible positions.

A query message travels along a random direction. Since the analysis is all probabilistic, we will need to define rigorously what we mean by a 'random line' [28]. Here we will analyze three possible ways to generate a random query. In the first definition, a query travels along a line connecting the query node q with a random point w on the sensor field boundary, with w uniformly distributed along the boundary. We call this model the *random boundary point model*. In the second definition, a query travels along a line with angle θ counter-clockwise from the positive x-axis, where θ is taken uniformly randomly from 0 to 2π. This model is called the *random angle model*. In the third definition, a query travels along a random chord, defined as the chord connecting two random boundary nodes. This is called the *random chord model*.

Of course, different ways to define a random line will lead to different probability measures but we will show below that they differ by only a constant factor. In our algorithm we implemented the random angle model as it only uses local knowledge. When the sensor nodes know their locations, or

have a local compass, the query can be routed with only local knowledge, in the same way as compass routing [15]. In the analytical result for the opportunistic information dissemination (subsection II-B) we use the random chord model as that makes the analysis easier.

In what follows we assume the sensors are deployed uniformly randomly in a disk. We define the *effective length* of a trajectory by the distance between the entrance and exit nodes along the sensor field boundary. Without loss of generality we can assume the total length of the sensor field boundary is 1. We assume the length of the trajectory between 0 and $\frac{1}{2}$, due to symmetry. These assumptions will be relaxed later for a generic probabilistic analysis in section II-A2.

Now we will analyze the probability that a random line intersects with the trajectory and the total query cost in order to guarantee that the trajectory is discovered with probability at least $1-\varepsilon$, by using the above two different ways of generating a random query.

a) Random boundary point model.: Given a motion trajectory of length ℓ, suppose that the query starts from a given query node q. A random query is selected to follow the line through q and a random point on the sensor network boundary. We now bound the probability that this query finds the desired trajectory.

Lemma 2.1. *The probability of a random query (in the random boundary point model) intersecting the given motion trajectory with effective length ℓ is $p_1(\ell) = 2\ell - 2\ell^2 + (\ell - 1/2)\frac{\sin 2\pi\ell}{\pi}$.*
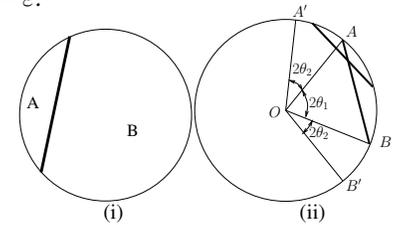
Proof: As Figure 2 (i) shows, if the query point locates in area A, then the probability that it generates a query that hits the given trajectory is $1 - \ell$, otherwise the probability is ℓ. The probability that the query point locates in area A is $(\pi r^2 \cdot \frac{2\pi\ell}{2\pi} - \frac{1}{2} \cdot 2r\sin\pi\ell \cdot r\cos\pi\ell)/(\pi r^2) = \ell - \frac{\sin 2\pi\ell}{2\pi}$. So the probability that a random query point hits the given trajectory is $(1 - \ell)(\ell - \frac{\sin 2\pi\ell}{2\pi}) + \ell(1 - (\ell - \frac{\sin 2\pi\ell}{2\pi})) = 2\ell - 2\ell^2 + \frac{\ell\sin 2\pi\ell}{\pi} - \frac{\sin 2\pi\ell}{2\pi}$. ∎

Theorem 2.2. *Given a trajectory with length ℓ and a number $\varepsilon \in (0, 1)$, with $K_1(\ell, \varepsilon) \geq \frac{1}{\ell}\ln 1/\varepsilon$ query lines under the random boundary point model, the probability to hit the given trajectory is greater than $1 - \varepsilon$.*

Proof: The probability that all the k random query lines do not hit the given trajectory is $(1 - p_1(\ell))^k$, where $p_1(\ell)$ is as defined in Lemma 2.1. So the probability that at least one random query lines hit that trajectory is $1 - (1 - p_1(\ell))^k$. We need

Fig. 2. (i) When the query point locates in A or B, it has different probability to send a query intersecting the given trajectory. (ii) Two chords have an intersection.

$1 - (1 - p_1(\ell))^k \geq 1 - \varepsilon$, that is $(1 - p_1(\ell))^k \leq \varepsilon$, i.e., $(1 - (2\ell - 2\ell^2 + \frac{\ell\sin 2\pi\ell}{\pi} - \frac{\sin 2\pi\ell}{2\pi}))^k \leq \varepsilon$. As $\ell \leq \frac{1}{2}$, $\frac{\ell\sin 2\pi\ell}{\pi} - \frac{\sin 2\pi\ell}{2\pi} = \frac{\sin 2\pi\ell}{\pi}(\ell - \frac{1}{2}) \leq 2\ell(\ell - \frac{1}{2})$.

We want $\varepsilon \geq (1 - 2\ell + 2\ell^2 - 2\ell(\ell - 1/2))^k = (1 - \ell)^k$. So $k \geq \frac{\ln 1/\varepsilon}{\ln 1/(1-\ell)}$, as $\ln\frac{1}{1-\ell} > \ell$, then $\frac{\ln 1/\varepsilon}{\ln 1/(1-\ell)} \leq \frac{1}{\ell}\ln\frac{1}{\varepsilon}$.

Then $K_1(\ell, \varepsilon) \geq \frac{1}{\ell} \ln \frac{1}{\varepsilon}$ is enough to meet the probability requirement. ∎

b) Random angle model.: In the second model, the query line is chosen as the line through a given query point with a random angle relative to the positive x-axis. Again, we consider the expected cost with respect to a random query point. We start with a technical lemma.

Recall that the *central angle* of a chord AB with two endpoints A, B on the circle boundary is the angle $\angle AOB$ where O is the center of the circle. The *inscribed angle* is exactly half of the central angle and is in between 0 and $\pi/2$. The following lemma is independent of the definition of a random line.

Lemma 2.3. *The probability that two random chords with inscribed angles θ_1 ($0 \leq \theta_1 \leq \frac{\pi}{2}$) and θ_2 ($0 \leq \theta_2 \leq \frac{\pi}{2}$) intersect within the circle is Prob$\{$intersection$|\theta_1, \theta_2\} = \frac{2}{\pi} \min(\theta_1, \theta_2)$. Specifically, when one chord (say the one with half angle θ_1) is given, then the probability is Prob$\{$intersection$|\theta_1\} = \frac{2}{\pi} \int_0^{\theta_1} \theta_2 f(\theta_2) d\theta_2 + \frac{2}{\pi} \int_{\theta_1}^{\pi/2} \theta_1 f(\theta_2) d\theta_2$, where $f(\theta)$ is the probability density function of the random variable θ.*

Proof: Without loss of generality, suppose $\theta_1 > \theta_2$. Once chord 1, say AB, has been placed, chord 2 will intersect chord 1 if and only if one endpoint falls inside the arc AB and another endpoint falls inside AA' or BB' (see Figure 2 (ii)). The probability for this to happen is $\frac{2\theta_2}{\pi}$. Thus we have obtained Prob[intersection$|\theta_1, \theta_2] = \frac{2}{\pi} \min(\theta_1, \theta_2)$. When chord 1 is given, the probability that a second random chord intersects chord 1 is Prob$\{$intersection$|\theta_1\} = \frac{2}{\pi} \int_0^{\pi/2} \min(\theta_1, \theta_2) f(\theta_2) d\theta_2 = \frac{2}{\pi} \int_0^{\theta_1} \theta_2 f(\theta_2) d\theta_2 + \frac{2}{\pi} \int_{\theta_1}^{\pi/2} \theta_1 f(\theta_2) d\theta_2$, where $f(\theta)$ depends on the probabilistic model we use to define a 'random chord'. ∎

Now we will consider the case that a random chord is generated by dropping a point at random in the circle and drawing a chord through that point in a randomly chosen direction.

Lemma 2.4. *Assume that the random point falls at a distance r from the center of the circle, and the random chord is selected by taking a randomly chosen direction through that point. Now the chord intercepts an arc of inscribed angle θ. Under this model, the density function $f(\theta|r) = \frac{2r \sin \theta}{\pi \sqrt{r^2 - \cos^2 \theta}}$, and $f(\theta) = \frac{4}{\pi} \sin^2 \theta$, where $0 \leq \theta \leq \frac{\pi}{2}$.*

Proof: Refer to [28][Page 139]. ∎

From Lemma 2.3 and Lemma 2.4, we can get the following theorem. The proofs are omitted.

Lemma 2.5. *For a given trajectory with length ℓ, and a query point randomly placed inside the disk, generate a random query by choosing a direction randomly, then the probability that the query will hit the trajectory is $p_2(\ell) = 2(\pi\theta_1 - \theta_1^2 + \sin^2 \theta_1)/\pi^2$, where $\theta_1 = \pi\ell$.*

Theorem 2.6. *Given a trajectory with length ℓ and ε ($0 < \varepsilon < 1$), with $K_2(\ell, \varepsilon) \geq \frac{1}{\ell} \ln 1/\varepsilon$ query lines under the random*

angle model, the probability to hit the given trajectory is greater than $1 - \varepsilon$.

c) Random chord model.: In the random chord model, the query is taken as a random chord with its two endpoints chosen uniformly random on the network boundary.

Lemma 2.7. *Given a trajectory with effective length ℓ, a random query under the random chord model will discover it with probability $p_3(\ell) = 2\ell - 2\ell^2$.*

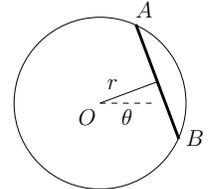*Proof:* $p_3(\ell) = 1 - (\ell^2 + (1 - \ell)^2) = 2\ell - 2\ell^2$. ∎

Theorem 2.8. *Given a trajectory with length ℓ and a number $\varepsilon \in (0, 1)$, with $K_3(\ell, \varepsilon) \geq \frac{1}{\ell} \ln 1/\varepsilon$ query lines under the random chord model, the probability to hit the given trajectory is great than $1 - \varepsilon$.*

The proof is similar to Theorem 2.6 and omitted.

2) Query on trajectories of arbitrary shape: In general, a motion trajectory can have any shape. And a target may start (appear) or stop (disappear) anywhere in the sensor field. We can use the same random query to search for such trajectories. The probability that a random query message discovers a target trajectory also depends on its shape and length.

To analyze the cost of such random queries, we will make use of standard geometric probability. A query starts from the query node and follows a random line with angle θ counterclockwise from the positive x axis (i.e., the random angle model). We define the length of a trajectory as the perimeter of the convex hull of the trajectory. In a degenerate case when the trajectory is a straight line segment, the length is defined as twice its Euclidean length.

Specifically, the trajectory T has length ℓ. A random line G is represented by two parameters r and θ, where r is the distance from the origin to the line, and θ is the angle formed by the normal to the line and the x-axis. Both r and θ are uniformly randomly chosen among their respective ranges. As we will see later, that this definition of a random query line also gives a similar bound on the query success probability. See Figure 3.

Fig. 3. A random line with distance r from the origin and angle θ between its normal and the positive x-axis.

By geometric probability the collection of lines that intersect a curve is represented precisely by the length of the curve [28].

Lemma 2.9 ([28]). *The measure of the set of straight lines that intersect a curve C is its length ℓ. That is $\int_{G \cap T \neq \emptyset} dG = \int_{G \cap T \neq \emptyset} dr \wedge d\theta = \int_0^{2\pi} r d\theta = \ell$.*

Now suppose T_1 is a chord of Euclidean length ℓ_1 inside a region $\mathcal{R}$. Then the chord is a degenerate rectangle with perimeter $2\ell_1$. When $\mathcal{R}$ is a disk with radius R, then $\ell_1 = 2R \sin \pi\ell$, where ℓ is the effective length of the chord.

Lemma 2.10 ([28]). *The probability that a random line intersecting $\mathcal{R}$ with effective length ℓ intersects with a trajectory T_1*

of length ℓ_1, is given by $\frac{\int_{G \cap T_1} dG}{\int_{G \cap \mathcal{R}} dG} = \frac{2\ell_1}{\ell}$. *Especially, when $\mathcal{R}$ is a disk, the probability is $\frac{2 \sin \pi \ell}{\pi}$.*

Now we can extend the theorem to the general case.

Theorem 2.11. *Given a convex boundary field with perimeter L, a trajectory with effective length ℓ and a constant ε ($0 < \varepsilon < 1$), the number of random query lines to hit the given trajectory with probability great than $1 - \varepsilon$ is $\frac{L}{2\ell} \ln \frac{1}{\varepsilon}$.*

The proof is similar to Theorem 2.6.

Despite various models of random queries, the analysis arrives at very similar results. This will be useful in the next section as we analyze the scheme of using motion trajectories themselves to disseminate information.

B. Opportunistic information propagation

In many tracking applications the targets being tracked come in the monitored field at different points in time. We can make use of the temporal diversity and perform opportunistic information propagation. To simplify the analysis, we assume that the targets come in the field one by one. When a new target moves in the field, some of the sensors nodes on its trajectory may have already had stored signatures of other targets that have passed by. Thus a sensor node waken up by a new target should take the opportunity to propagate the information it has recorded so far. Naturally as a target enters the field, the node along the trajectory will hand over the knowledge it has learned to the next node on the trajectory. After a target exits the monitored field, a backward propagation may also be conducted such that all the nodes on the trajectory share the union of the signal signatures of them. In the analysis for simplicity we assume the backward propagation. In our simulations we do not conduct backward propagation. By using opportunistic information propagation, a target signature is known not only to nodes along its trajectory but also possibly carried to other nodes by the targets that come in afterwards. The longer a signature exists in the system, the wider it gets propagated in the sensor field. We thus give each motion trajectory an *age* which is the number of targets that come in after itself, but before the query is initiated.

Suppose there are two trajectories A and B that show up in the monitored field sequentially. They have effective lengths ℓ_A and ℓ_B respectively. Say a node sends a random query along the line C. Then there are two possibilities for the query to discover the trajectory A, either because C intersects with A directly or because C intersects with B which intersected with A and helped to propagate information about A. Here is an interesting tradeoff — the older a trajectory is, the easier to query for it; yet a detection long time ago was not going to be interesting from many real-time applications' point of view. The number of trajectories kept at each node is also bounded by the storage requirement. In the following analysis we will evaluate the minimum 'lifetime' of a trajectory, that is, how old a trajectory needs to be, in order to keep the query success rate high. Thus a sensor node would only record the newest

k trajectories it has learned, where k is taken as the lifetime above or as application requires.

We remark that as our queries are also probabilistic, we can also take the communication opportunities provided by queries issued in the network to further spread information around. In this case a query is simply regarded as a 'fake' target trajectory and can be treated the same as the other real ones.

1) Technical lemmas: We start by some technical lemmas to be used later. We assume that each trajectory has a random entrance point and a random exit of the sensor field. It can be modeled as the chord connecting two endpoints selected uniformly randomly on the sensor field boundary. Recall that the effective length of a trajectory is defined with respect to its entrance and exit points on the sensor network boundary, which partition the sensor network boundary into two segments, one is longer than the other. The effective length is taken as the normalized length of the shorter segment and is always smaller than $1/2$.

Corollary 2.12. *Two random trajectories intersect with each other with probability $2/3$.*

Proof: We will integrate over $p_3(\ell)$ for $0 \le \ell \le 1/2$ and obtain $2 \int_{\ell=0}^{1/2} p_3(\ell) d\ell = 2/3$. ∎

Lemma 2.13. *Given n random chords, the probability that they form a connected network is at least $1 - O(1/n)$.*

Proof: Each random chord connects two random points on the network boundary. Denote by the random endpoints as x_i, y_i. And the collection of these endpoints form a cyclic permutation of $2n$ points on the network boundary. If the trajectories do not form a connected network, one can partition the set of endpoints into 2 groups with $2k$ and $2n - 2k$ endpoints each, consecutively distributed along the sensor network boundary, $1 \le k \le n - 1$, such that no chord chooses its endpoints in different groups. Now we calculate the probability for this to happen:

$$
\begin{aligned}
& \text{Prob}\{n \text{ random trajectories are not connected}\} \\
\le \quad & f(n) = \sum_{k=1}^{n-1} \binom{n}{k} / \binom{2n}{2k} \\
= \quad & \frac{1}{2n-1} \cdot \left[\sum_{k=1}^{n-2} \binom{n-1}{k} / \binom{2n-2}{2k} \cdot (2n - 2k - 1) + 1 \right] \\
\le \quad & \frac{1}{2n-1} \cdot \left[f(n-1) \cdot n + 1 \right].
\end{aligned}
$$

The last inequality uses the symmetry of the series $\binom{n}{k} / \binom{2n}{2k}$ in terms of k and sums up the first element with the last element, the 2nd element with the second last element and so on. We can check that if $f(n-1) \le \frac{c}{2n-3}$, for $c \ge 5$ and $n \ge 4$,

$$
\begin{aligned}
f(n) & \le \frac{1}{2n-1} \cdot [f(n-1) \cdot n + 1] \\
& \le \frac{1}{2n-1} \cdot \left[\frac{cn}{2n-3} + 1 \right] \le \frac{c}{2n-1}.
\end{aligned}
$$

Thus with probability at least $1 - O(1/n)$ the random trajectories are connected. ∎

Corollary 2.14. *Given n random chords, the probability that at least $n - h$ of them form a connected network is at least $1 - O(1/n^h)$.*

Proof: We calculate the probability $q(n, h)$ as the probability that the largest connected component has size at most

$n - h$. With a similar argument as the previous lemma, if the largest connected component has size $\leq n - h$, we can partition the endpoints into 2 groups with $2k$ and $2n-2k$ with $h \leq k \leq n - h$ such that each chord chooses its endpoints within one group only. We have

$$
\begin{aligned}
q(n,h) \quad &= \text{Prob}\{\text{\# largest connected component } \leq n - h\} \\
&\leq f(n,h) = \sum_{k=h}^{n-h} \binom{n}{k}/\binom{2n}{2k} \\
&= f(n-1,h) \cdot \frac{n}{2n-1} + \binom{n}{h}/\binom{2n}{2h}.
\end{aligned}
$$

For large n and a small constant h, we have $\binom{n}{h} \approx \frac{n^h}{h!}$, $\binom{2n}{2h} \approx \frac{n^{2h}}{(2h)!}$. One can then verify that $f(n,h) = O(1/n^h)$. Thus the probability that at least there is a connected network of $n - h$ chords is $1 - O(1/n^h)$. ∎

Given a set of n connected random trajectories, define the *separation length* of this set as the maximum distance between the adjacent endpoints on the network boundary, normalized by the total length of the perimeter of the sensor network.

Lemma 2.15. *Given n connected random trajectories, the separation length is in expectation $\frac{1}{2n}$ and $\frac{c \ln n}{n}$ with probability at least $1 - O(1/n)$, for a constant c.*

Proof: The analysis follows from the standard random sampling techniques [19]. A short remark is that the endpoints are randomly selected on the network boundary. We can ignore the fact that the trajectories are connected because this only determines the combinatorics of how the endpoints are paired up to n chords. ∎

Lemma 2.16. *For n random chords that form a connected trajectory network E, any random chord intersects the union of these chords with probability at least $1 - O(\ln^2 n/n)$.*

Proof: Given that the random chords form a connected network, the endpoints of these chords partition the network boundary into $2n$ segments ℓ_i, $1 \leq i \leq 2n$. $\sum_{i=1}^{2n} \ell_i = 1$. Therefore, the probability that a random chord does not intersect this trajectory network is $1 - \sum_{i=1}^{2n} \ell_i^2$. With probability $1 - O(1/n)$, $\ell_i \leq \frac{c \ln n}{n}$ for all i (denoted by event A). Therefore the probability that a random chord will intersect the trajectory network is

$$
\begin{aligned}
p_4(n) &= \text{Prob}\{\text{a random chord intersects } E\} \\
&\geq \text{Prob}\{\text{a random chord intersects } E|A\}(1 - O(1/n)) \\
&= (1 - \sum_{i=1}^{2n} \ell_i^2) \cdot (1 - O(1/n)) \\
&\geq 1 - O(\ln^2 n/n).
\end{aligned}
$$
∎

2) Opportunistic information dissemination: We will now analyze how the trajectories that come after a particular trajectory T can help to disseminate information about T. Notice that now the sequence of the trajectories showing up matters in how information gets propagated in the network. In particular, we define a *dissemination network* of n trajectories such that the i-th trajectory comes after the $(i-1)$-th trajectory and intersects with one of the previous trajectories. We now analyze the probability that a dissemination network of n trajectories is formed and how it helps to disseminate the

information about a random trajectory T such that a random query will be able to retrieve T with probability $1 - \varepsilon$, for $0 < \varepsilon < 1$. In this subsection we use the random chord model for a random query.

Consider f trajectories entering the field one by one after a trajectory T. We examine how large f should be in order to form a dissemination network with n trajectories to help disseminate information about T. We will divide the process into epochs such that after epoch 1 we have another trajectory T_1 that intersects T. The union of the trajectories T_1 and T is the dissemination network N_1 after epoch 1. Similarly, after epoch i we have a trajectory T_i that intersects with the current trajectory network N_{i-1} and the trajectory network N_i is updated to include T_i as well. Denote by f_i the number of trajectories in epoch i and take $f = \sum_{i=1}^{n} f_i$ as the total number of trajectories. Now we calculate the expected number of f_i. Recall that a trajectory will intersect a trajectory network of size $i-1$ with probability at least $1 - c\ln^2 i/i$, for a constant c, by Lemma 2.16. Thus the expected number of trajectories in epoch i would be at most $1/(1 - c\ln^2 i/i)$. Therefore the expected total number of trajectories to obtain a network of size n is

$$
E(f) = \sum_{i=1}^{n} E(f_i) \leq \sum_{i=1}^{n} 1/(1 - c\ln^2 i/i) = n + o(n).
$$

We also calculate the variance of f. As f_i's are independent of each other and each f_i has a geometric distribution of probability at least $1 - c\ln^2 i/i$, then $V(f_i) \leq \frac{c\ln^2 i/i}{(1-c\ln^2 i/i)^2}$.

$$
\begin{aligned}
V(f) &= \sum_{i=1}^{n} V(f_i) \leq \sum_{i=1}^{n} (c\ln^2 i/i)/(1 - 2c\ln^2 i/i) \\
&= c\ln^3 n + o(\ln^3 n).
\end{aligned}
$$

Now we can calculate the probability that f is greater than $(1 + \delta)n$, by Chebyshev inequality:

$$
\text{Prob}\{f \geq n + \delta n\} \leq 1/(\delta n/V(f))^2 = c^2 \ln^6 n/(\delta^2 n^2).
$$

Now we can summarize the main result for opportunistic information dissemination, by combining the analysis above and Lemma 2.16.

Theorem 2.17. *Assuming the trajectories are random, a trajectory T with age a can be discovered by a random query with probability $1 - O(\ln^2 a/a)$.*

Proof: By the analysis above, the trajectories after T will form a dissemination network of $a/(1 + \delta)$ trajectories with probability at least $1 - c^2 \ln^6 a(1+\delta)^2/(\delta^2 a^2)$, for a constant $\delta < 1$. Further, the probability that a query will find out information about T from the dissemination network is at least $1 - O(\ln^2 a/a)$, from Lemma 2.16. Therefore the probability that a random trajectory with age a will be discovered with probability $1 - O(\ln^2 a/a)$. ∎

Corollary 2.18. *Assuming the trajectories are random, the number of queries to discover a random trajectory with age a with probability $1 - \varepsilon$ is $\Omega(\frac{\ln(1/\varepsilon)}{\ln a})$.*

Proof: The probability that k random queries do not discover the trajectory T with age a is $(O(\ln^2 a/a))^k \leq \varepsilon$. Thus $k \geq \Omega(\ln(1/\varepsilon)/\ln a)$. ∎

During the opportunistic dissemination process, each node has to store all the trajectory information that the trajectory carries. So the storage space will increase with time. Here we will give an analysis on how long should the information of a trajectory be kept for the purpose of opportunistic dissemination. That is the expiration time for a given trajectory.

For a trajectory with length ℓ, the probability that another random trajectory will intersect with it is $p = 2\ell(1-\ell)$. Suppose its age is a. We want to calculate the proper value of age a, so that the trajectory with length ℓ can be discovered in constant times of queries after age a. The expected number of queries to discover a trajectory with length ℓ after age a is

$$
\begin{aligned}
N(\ell, a) & = \quad \text{Prob}\{\exists \text{ one trajectory that hits } T\} \cdot c_1 + \\
& \qquad \text{Prob}\{\nexists \text{ trajectory that hits } T\} \cdot 1/p \\
& = \quad [1 - (1-p)^a] \cdot c_1 + (1-p)^a \cdot 1/p,
\end{aligned}
$$

for a constant c_1. We want $N(\ell, a)$, that is, $(1-p)^a \cdot 1/p$ to be a constant. That means $(1-p)^a = c \cdot p$, for a constant c. Now we have $a = \frac{\ln 1/(cp)}{\ln 1/(1-p)} \leq \frac{\ln 1/\ell}{2\ell(1-\ell)}$. So for small ℓ, we need $a \approx \frac{c' \ln 1/\ell}{\ell}$, where c' is a constant.

From the above analysis, a trajectory does not need to be kept forever for the purpose of opportunistic dissemination. Each trajectory with (normalized) length ℓ only needs to be kept for an age of $\frac{c' \ln 1/\ell}{\ell}$. In this way, the storage requirement at each node will be reduced significantly, yet still have the trajectories to be discovered with a small number of queries.

III. Simulations

Our previous analysis focuses on the asymptotic bound of the number of queries for a given target. The simulation results below give an idea of the number of queries needed in a practical setting, as well as the tradeoff of storage requirement versus the query cost. We evaluate the opportunistic dissemination and in-network query scheme by simulations on random trajectories, and compare it with the scheme without opportunistic dissemination (i.e., only the nodes near the target trajectory record information about it). In particular, we evaluate the following three important measures:

- The number of queries issued to search for a given trajectory with respect to its length and age.
- The total communication cost comparison between the opportunistic dissemination scheme and the scheme without opportunistic dissemination. Note that the cost includes both the preprocessing cost and the query cost.
- The storage requirement at each node.

In the simulation, we generate a network with sensor nodes uniformly randomly distributed in a circular field with radius 25. We use a unit disk graph model with a communication range of 1. The average degree of the sensor network is about 7. The trajectories are randomly generated within the sensor field with the entrance and exit points taken randomly on the sensor field boundary. The nodes within distance 1 from the
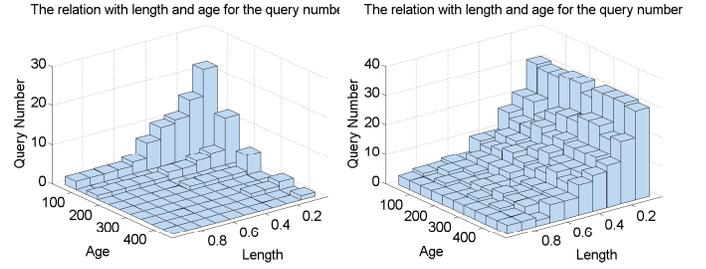
Fig. 4. Left: The query number for a trajectory with different lengths and different ages with opportunistic dissemination. Right: The query number for a trajectory with different lengths and different ages without opportunistic dissemination.

trajectory are waken up to record and disseminate information about the trajectory. In all our following simulations, we generate 500 random trajectories before starting the query.

Queries are initiated at nodes randomly chosen inside the sensor field. For a particular query, the query messages follows a randomly chosen direction until either it visits some sensor node that contains information about the desired trajectory, in which case the information is returned back to the query node; or it hits the sensor field boundary, in which case the query is not successful and another query is generated. Routing of a query is done by greedy geographical routing.

In the simulation we focus on trajectory existence query that answers whether or not a trajectory with a given ID was present or not. With trajectory report queries, we need to record not simply an ID but also the detailed trajectory information.

Query number v.s. age and length. We examine the relationship of the number of queries to find a trajectory with its age and length. We randomly choose a query point for each trajectory and repeat this process for 100 times. We divide the age and length into small ranges and take the average query number for all the trajectories with the (age, length) combination falling into the same bucket. Here the trajectory length is measured by its Euclidean length, normalized by the network size. The age of a trajectory is the number of trajectories detected after it but before the query is issued. In this experiment we assume each node keeps every trajectory it detects or exchanges from its neighbors during opportunistic dissemination.

From Figure 4, in opportunistic dissemination scheme, the query number will be reduced greatly with the increasing of age or length. When either the age or length is large enough, the query number is close to 1. Without opportunistic dissemination, the query number just relates to the length, and can be much larger than that of the trajectory with similar length in opportunistic dissemination scheme.

Required storage size. In this simulation, we will evaluate the affect of the storage size on the query number. Here, storage size means how many trajectory each node will keep. There is a tradeoff between the preprocess cost and the query cost, which is greatly affected by the storage size.

We have different ways to choose which trajectories will be kept depending on different requirement. For example, if we
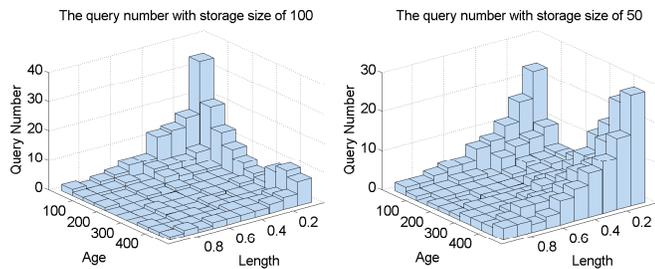
Fig. 5. Left: The query number for a trajectory with different lengths and different ages with opportunistic when the storage size is 100. Right: The query number for a trajectory with different lengths and different ages with opportunistic when the storage size is 50.

choose to keep the most frequently queried trajectory, it will help to distribute the popular trajectories, which may dominate the total communication cost and will help to reduce the total communication cost.

In our simulation, we simply choose the most recent k trajectories, where k is the storage size. But each node will always keep the trajectories that it detects.

From Figure 5, and the left Figure 4, we can see the affect of the storage size on the query number. With an increasing storage size, the query number for most trajectories will be reduced. When the storage size decreases, old trajectories will be removed from some nodes, so cost for the old enough trajectories will be increased greatly.

The best storage size chosen in the scheme will depend on users' requirement. In our simulation, 100 (or around 100) seems to be a reasonable storage size for queries of the past 500 trajectories. As we are typically interested in querying recent trajectories, so a small storage size will work out fine. If we hope to query old trajectories, then we need larger storage size. The best storage size will vary significantly as the topologies and density of sensor nodes change. In practice one can use an adaptive algorithm to gradually reduce the storage size while still meet the delay requirement. This remains as future work.

Communication cost. We compare the communication cost in the situations with and without opportunistic dissemination. If we just consider the query cost, obviously it is better to adopt the opportunistic dissemination scheme. However, we have to pay additional cost for opportunistic dissemination, which is called the preprocessing cost. There is no preprocessing cost without the opportunistic dissemination, but the query cost is higher. So we will evaluate the the overall communication cost including both the preprocessing cost and the query cost for a certain query frequency.

We use the summed packet size to evaluate the communication cost. We adopt 36 bytes as the overhead for packet header from TinyOS. We assume 1 byte cost for each trajectory ID stored at the sensor node during the information exchange. For convenience, we consider the communication cost as the packet size divided by 36. Under this assumption, each packet adds 1 to the communication cost without opportunistic dissemination. With opportunistic dissemination, the packet may contain k prior trajectories and we divide the cost equally

among the k trajectories. This is the additional cost that each trajectory has to pay to have its information disseminated. In the query stage, each packet contributes 1 to the cost for the given query trajectory. So the total communication cost is the sum of the preprocess cost and the total query cost (query frequency times the cost per query). From previous section, we know that storage size of 100 works well for our simulation. So in our following simulation, we set the storage size to be 100.

Figure 6 shows the results. The oldest trajectories have high communication cost because they were disseminated with a lot of preprocessing cost but were not queried on time before such information is flushed out by new trajectories. For the trajectories with middle-ranged ages, the communication cost is much smaller in opportunistic scheme than that without opportunistic dissemination. Intuitively, the benefit of dissemination (i.e., lower communication cost) shows up with increasing query frequency. When the query frequency is higher than 200, the total communication cost is always smaller for opportunistic scheme. If we just consider the most recent 200 trajectories as in the middle Figure 6, after 100 queries, the average communication cost has already been less than that without opportunistic dissemination. While without opportunistic dissemination, the average cost will keep the same no matter how many queries are issued for a given trajectory.

IV. CONCLUSION

We have presented in this paper the exploitation of the spatial and temporal coherence of motion trajectories in the domain of in-network storage and query. As sensors have scarce resources, the use of existing opportunities in the detection itself in managing and query for the detection data, is expected to be useful in a more general domain of data management in wireless sensor networks.

In this paper, we mainly address the discovery of existence of a certain target. In real life some queries are aggregate queries. For example, how many trucks have passed in the past hour. The same framework can be used to answer aggregate queries in a relatively naive way – use a random query to return all the trajectories detected and compute the aggregate. Advanced mechanism for answering aggregate queries of the motion data remains as future work. Our current algorithm and analysis is based on the assumption that the trajectories start and end randomly. We leave it as future work to extend the analysis to the setting when the entry and exit points are clustered in a small region.

REFERENCES

[1] W. Adjie-Winoto, E. Schwartz, and H. Balakrishnan. The design and implementation of an intentional naming system. In *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 186–201, December 1999.

[2] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus. Tracking a moving object with a binary sensor network. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 150–161, 2003.

[3] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(1):69–74, 2004.
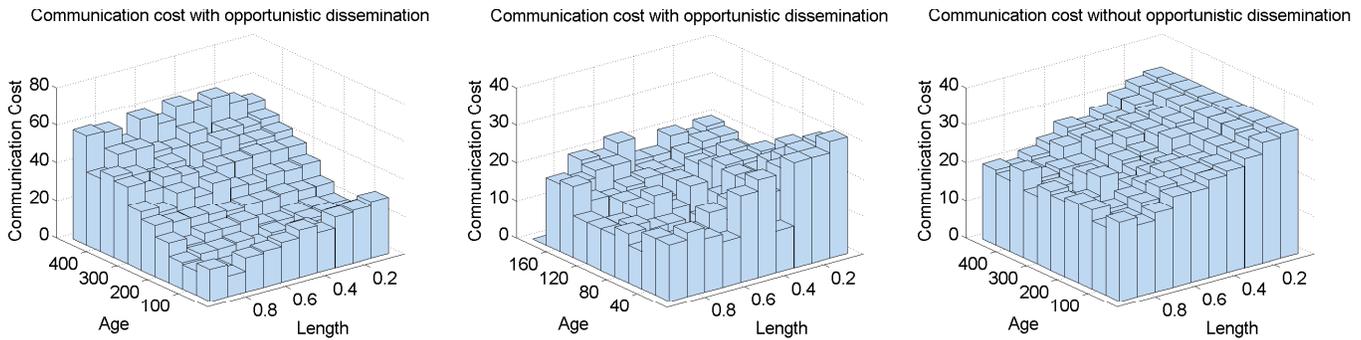
Fig. 6. Left: The average communication cost by querying each trajectory 100 times under opportunistic dissemination. Middle: The average communication cost by querying each trajectory(with age less than 200) 100 times under opportunistic dissemination. Right: The average communication cost by querying each trajectory 100 times with no opportunistic dissemination.

[4] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proc. of the 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 22–31, September 2002.

[5] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, pages 449–460, Washington, DC, USA, 2004. IEEE Computer Society.

[6] Q. Fang, J. Gao, and L. J. Guibas. Landmark-based information storage and retrieval in sensor networks. In *The 25th Conference of the IEEE Communication Society (INFOCOM'06)*, pages 1–12, April 2006.

[7] D. Ganesan, D. Estrin, and J. Heidemann. DIMENSIONS: Why do we need a new data handling architecture for sensor networks. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 143–148, 2002.

[8] J. Gao, L. Guibas, J. Hershberger, and L. Zhang. Fractionally cascaded information in a sensor network. In *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pages 311–319, April 2004.

[9] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. DIFS: A distributed index for features in sensor networks. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 163–173, Anchorage, Alaska, May 2003.

[10] L. J. Guibas. Sensing, tracking and reasoning with relations. *IEEE Signal Processing Magazine*, 19(2):73–85, March 2002.

[11] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2(1):1–38, 2006.

[12] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proceedings of the Symposium on Operating Systems Principles*, pages 146–159, October 2001.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *ACM Conf. on Mobile Computing and Networking (MobiCom)*, pages 56–67, 2000.

[14] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham. On target tracking with binary proximity sensors. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 301–308, 2005.

[15] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. 11 th Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.

[16] X. Liu, Q. Huang, and Y. Zhang. Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 122–133. ACM Press, 2004.

[17] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[18] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM Press.

[19] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[20] B. Nath and D. Niculescu. Routing on a curve. *SIGCOMM Comput. Commun. Rev.*, 33(1):155–160, 2003.

[21] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 250–262, New York, NY, USA, 2004. ACM Press.

[22] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage in sensornets. In *Proc. 1st ACM Workshop on Wireless Sensor Networks ands Applications*, pages 78–87, 2002.

[23] R. Sarkar, X. Zhu, and J. Gao. Double rulings for information brokerage in sensor networks. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 286–297, September 2006.

[24] R. Sarkar, X. Zhu, and J. Gao. Hierarchical spatial gossip for multi-resolution representations in sensor networks. In *Proc. of the International Conference on Information Processing in Sensor Networks (IPSN'07)*, pages 420–429, April 2007.

[25] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: New aggregation techniques for sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239–249, New York, NY, USA, 2004. ACM Press.

[26] N. Shrivastava, R. M. U. Madhow, and S. Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 251–264, 2006.

[27] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley. Tracking multiple targets using binary proximity sensors. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 529–538, New York, NY, USA, 2007. ACM Press.

[28] H. Solomon. *Geometric Probability*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial Mathematics, 1987.

[29] I. Stojmenovic. A routing strategy and quorum based location update scheme for ad hoc wireless networks. Technical Report TR-99-09, SITE, University of Ottawa, September, 1999.

[30] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, January 2004.

[31] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, 2002.