Mission-Critical Management of Mobile Sensors (or, How to Guide a Flock of Sensors)

Goce Trajcevski, Peter Scheuermann*
Department of ECE
Northwestern University
Evanston, Il
{goce,peters}@ece.norethwestern.edu

Herve Brönnimann
Department of CIS
Polytechnic University
Brooklyn, NY
{hbr}@poly.edu

Abstract

This work addresses the problem of optimizing the deployment of sensors in order to ensure the quality of the readings of the value of interest in a given (critical) geographic region. We assume that each sensor is capable of reading a particular physical phenomenon (e.g., concentration of toxic materials in the air) and transmitting it, say, to a server and is also capable of moving, where the motion of each sensor may be remotely controlled. In scenarios like disaster management and homeland security, in case some of the sensors dispersed in a larger geographic area report a value higher than a certain threshold, one may want to ensure a quality of the readings for the affected region. This, in turn, implies that one may want to ensure that there are enough sensors in the affected region and, consequently, guide a subset of the rest of the sensors towards the affected region. In this paper we explore variants of the problem of optimizing the guidance of the mobile sensors towards the affected geographic region and we present algorithms for their solutions.

1 Introduction and Motivation

The management of the transient *(location,time)* information for a large amount of mobile users has recently spurred a lot of scientific research. It began with the investigation of the trade-offs in updating the information vs. minimizing the look-up time of a particular user's location [18] and ranges to many challenging aspects of modeling, efficient storage and retrieval, and processing of a novel types of spatio-temporal queries in the field called Moving Objects Databases[1] (MOD).

On the other hand, a challenging research field which recently emerged is the management of a *sensor-generated* data. Sensors are low-cost devices which are capable of measuring a value of a particular physical phenomenon and transmitting it within a limited range and, they may also have some limited processing power. These sensors can be

mobile and may be deployed in a certain geographical area that is determined by t heir speed. Networks of sensors have already been deployed in the real world [10] and very active research efforts are being undertaken both in industry and academia [3]. Various aspects of interest for managing the sensor-generated data have been investigated (e.g., battery-life management, communication management of ad-hoc networks, stream-like management of the sensor-generated data, etc.) and a recent collection reporting the status of different research works is given in [12] and [13].

Although mobility in sensor networks has been addressed in the context of communication protocols for ad-hoc and peer-to-peer networks (e.g., [11, 21]), we believe that the mobility dimension plays an additional important and unexplored role in the overall topic of the sensor data management, which is the basic motivation for our research. This particular work is based on the fact that the spatial range for which the quality of the readings that a sensor can guarantee is limited and we tackle the problem of how to deploy a *sufficient* number of sensors in a given region. In particular, we consider the setting of disaster management in a homeland security where a set $\mathcal{S}$ of *mobile* sensors is deployed in a large geographic area in order to monitor particular value(s) of interest, e.g., the temperature or the concentration of toxic materials in the air. In case a certain subset $\mathcal{S}_k$ of the sensors, co-located in

*Research partially supported by NSF grant IIS-0325144

[1] An excellent collection of up-to-date results is presented in [14]

a given region, become *hot*, in a sense that their readings exceed a given (pre-defined) tolerance threshold, we would like to ensure that we optimize the guidance of a subset of m sensors from $\mathcal{S} \setminus \mathcal{S}_k$ in order to ensure the quality of the readings in the critical geographic region. In some way, our work can be viewed as a step towards adding spatio-temporal context awareness in managing sensor data.

The rest of this paper is organized as follows. In Section 2 we introduce the formal terminology and we address the issue of *critical times* with respect to the guidance of mobile sensors. In Section 3 we address the more stringent requirement of the *placement* of the sensors within some optimal time-frame. Section 4 gives a brief overview of the relevant literature and in Section 5 we present concluding remarks and we outline some areas for future work.

2 Critical Times

In this section we formally introduce the terminology and part of the problem statements.

We assume that we are given a set of distributed mobile sensors $\mathcal{S} = \{s_1, s_2, \ldots s_k\}$, where each sensors is represented as a pair $s_i = ((x_i, y_i), v_i)$. (x_i, y_i) denotes the location of the sensor s_i and v_i denotes its speed[2]. Each sensor s_i periodically reports the reading val_i of the value of the physical phenomenon that it is observing and we use the term *critical value* – C_v to denote the threshold for each val_i. Assuming that, at a certain time instance, some of the sensors have read a value greater then C_v, we define

Critical Region: Given a subset $\mathcal{S}_k \subseteq \mathcal{S}$ of sensors $\mathcal{S}_k = \{s_{j1}, s_{j2}, \ldots, s_{jk}\}$ such that $(\forall i)(val_{ji} \geq C_v)$, the *critical region* C_R of $\mathcal{S}_k$ ($C_R(\mathcal{S}_k)$) is defined as the convex hull of the set of 2D points $\{(x_{j1}, y_{j1}), (x_{j2}, y_{j2}), \ldots, (x_{jk}, y_{jk})\}$.

Our goal is to ensure that the deployment of the set of sensors is optimized, subject to the requirements which will ensure some *quality* of the readings inside C_R.

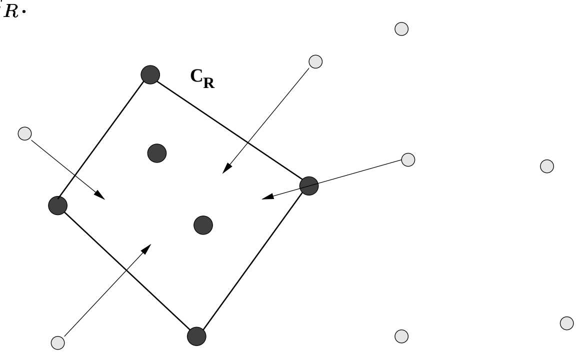

Figure 1: Guiding the sensors towards the critical area

[2] We omit the details of modeling of the sensors motion plan (future trajectory) or past completed motion (past trajectory) [20].

The concepts are illustrated in Figure 1, where the hot sensors are indicated by dark circles and white circles indicate the sensors which are not hot, some of which should be guided towards C_R.

Now we proceed with the different variations of the first category of the problem of optimizing the deployment of the sensors in the critical area.

2.1 Minimizing the Arrival Time

This is the simplest variation of the problem and we assume that in order to ensure the desired accuracy of the readings (i.e., the desired coverage of the critical region) it suffices to have m sensors in the interior of C_R. We would like to ensure that the *time* it takes for a desired number of sensors to *arrive* inside C_R is *minimized.* Figure 1 illustrates the case where we need six sensor inside C_R and we already have two hot ones, which implies we need to bring four more in a fastest possible manner.

The problem can be formally stated as follows:

Given: An integer m; a set of sensors $\mathcal{S}$; and a subset $\mathcal{S}_k \subseteq \mathcal{S}$;

Goal: Minimize the time for which it can be guaranteed that there are m sensors in the interior of $C_R(\mathcal{S}_k)$.

Let t_{ai} denote the minimal time for which the sensor $s_i \in \mathcal{S}$ can reach the boundary of C_R, call it its *critical arrival time* – cat_i. Obviously, if $(x_i, y_i) \in C_R$, then $cat_i = 0$. If we let k_1 denote the number of edges/vertices of the $C_R(\mathcal{S}_k)$ then $m = k - k_1$. Observe that the minimal time for a particular sensor $s_i \in \mathcal{S}$, which is outside C_R, to reach the boundaries of C_R, is actually equivalent to the time it takes for a circle centered at (x_i, y_i) and with radius $v_i \cdot t_i$ to intersect C_R.

Clearly, after constructing the convex hull for the location-points of the sensors in $\mathcal{S}_k$, one only needs to determine the set of m closest points to $C_R(\mathcal{S}_k)$ and get their arrival times $cat_{a1} \leq cat_{a2} \leq \ldots \leq cat_{am}$. In order to achieve our goal, we need at least $cat = cat_{am}$ time units. The asymptotic complexity of this approach is bounded by $O(n\log k)$, since the determination of the minimal distance from a point to a given convex region can be achieved in $O(\log k)$ [16]. Observe that *cat* is the lower bound on the time that we need to ensure that there are m sensors *anywhere* inside $C_R(\mathcal{S}_k)$.

Let us point out that in case the critical region $C_R(\mathcal{S}_k)$ is defined as a circle, the diameter of which is the diameter of the set of location-points of the sensor in $\mathcal{S}_k$, the complexity of calculating *cat* becomes linear ($O(n)$).

2.2 Minimizing the Furthest-Point Reachability Time

The next variation of the problem of ensuring the quality of the sensors' readings in the critical region considers the upper bound on the time it takes to bring the desired number of sensors (m)inside the critical region. Once again, we have a problem of selecting a subset of ($\mathcal{S} \setminus \mathcal{S}_k$) of size m, except now the selection criterion is different.

Let t_{fi} denote the minimum time for which the sensor $s_i \in \mathcal{S}$ can reach the furthest point in C_R with respect to its location (x_i, y_i). We will call it its *critical furthest-point time* – cft_i. In a manner similar to the calculation of *cat* time, in $O(n \log k)$ we can obtain the set of m sensors such that $t_{f1} \leq t_{f2} \leq \ldots \leq t_{fm}$. If we set $cft = t_{fm}$ then cft is, in a sense, an upper bound on the time it takes to ensure that there are m sensors *anywhere* inside $C_R(\mathcal{S}_k)$.

Now we proceed with the most desirable (and most complicated) setting of optimizing the critical time.

2.3 Critical Covering Time (cct)

The formal statement of this problem is:
Given: An integer m; a set of sensors $\mathcal{S}$; and a subset $\mathcal{S}_k \subseteq \mathcal{S}$;
Goal: Minimize the time *cct* such that there exists a subset $\mathcal{S}_m \subseteq \mathcal{S}$ of m sensors such that *any point inside* $C_R(\mathcal{S}_k)$ can be reached by some sensor in $\mathcal{S}_m$ in time $\leq$ *cct*.

The dual problem can be formulated as:
Given: A time-value *cct*; a set of sensors $\mathcal{S}$; and a subset $\mathcal{S}_k \subseteq \mathcal{S}$;
Goal: Minimize m, such that a subset $\mathcal{S}_m \subseteq \mathcal{S}$ with m elements exists, for which any point within $C_R(\mathcal{S}_k)$ can be reached by some sensor in $\mathcal{S}_m$ in time $\leq$ *cct*.

However, this is an instance of the set-cover problem which is NP-complete [7]. Even this instance (disk-covering in 2D) is NP-complete, although it can be approximated within a constant factor [2] as opposed to logarithmic at best for the general set cover. Thus, the best solution one can hope for is a heuristic solution. One possible approach is to relax the limit of m and ask how *all* the sensors can achieve the desired covering of C_R (i.e., set $m = n$). In this case, the decision problem **DP1** amounts to constructing the union of all the n disks and checking if it covers C_R. This can be done in $O(n \log^2 n)$ time, as the union of disks (even of different radii) has complexity $O(n)$ [1, Ex 3.6]. Let us point out that by applying binary searching, one can determine the $\mathcal{S}_m$ and *cct* up to any accuracy (using **DP1**). (The exact value can be determined in $O(n$polylog$n)$ time by designing a parallel version of the decision algorithm and using parametric searching.)

3 Spatial Limits on the Validity of Readings and Sensors Placement

Now we present another set of requirements for the quality of the sensors' readings, which pertains to the spatial coverage of the critical region C_R and we explore the time-boundaries for these variations of the problem. The key assumption is that the readings of each sensor are *valid* only within a limited geographic region which is a *disk* with radius r centered at a sensor's location-point. First, we present a general setting where we assume we have enough sensors and then we address the problem of limited resources.

3.1 Full Coverage of C_R

As a first variation, instead of simply having m sensors inside C_R, we require that C_R is entirely covered by disks of radius r centered at the sensors location-points. We assume that we have sufficient number of disks to ensure the coverage of C_R. The problem can now be stated as
Given: set of sensors $\mathcal{S}$; region C_R;
Goal: determine the minimal time (denote it minimal routing time *mrt*) such that a subset $\mathcal{S}_m \subseteq \mathcal{S}$ exists which can be moved inside C_R in such a manner that every point in C_R is at distance $\leq r$ from the location-point of some sensor $s_{mi} \in \mathcal{S}_m$.

Observe that the problem has some implicit requirements – we need to determine the *trajectory* of each mobile sensor s_{mi}. Moreover, we don't even have the boundary for m. If we let $A(C_R)$ denote the area of the critical region and ε denote the maximal percentage of overlap between two disks that a user allows[3], then we can have a reasonable *lower bound* on m calculated as $A(C_R)/(\pi \cdot (1-\varepsilon))$. Clearly, the more sensors we have available, the smaller value of *mrt* we can obtain. The formulation of the corresponding dual-like problem can be specified as
Given: set of sensors $\mathcal{S}$; region C_R; time limit *mrt*;
Goal: Determine the minimal m such that m sensors can be placed inside C_R in such a manner that C_R can be covered by disks of radius r centered at the sensors location-points.

Obviously, the techniques presented in Section 2 cannot be directly applied in these settings. However, they can still give us some useful bounds. Let (x_i, y_i) denote the "current" location of the i-th sensor (which is, before it is routed towards C_R). Then *mrt* must be large enough such that the union of the disks centered at each (x_i, y_i), with respective radii $r + v_i \cdot mrt$, covers the entire region C_R. This is equivalent to the requirement that *any* point of C_R (interior + boundary) can be reached by *at least* one sensor. Thus, a reasonable lower bound for *mrt* is the *cct* (c.f. Section 2).

This is illustrated in Figure 2, where for simplicity we have assumed that the only hot sensors are on the vertices of C_R and we don't indicate their coverage area. White disks indicate the initial location and the area in which the readings of a particular sensor are

[3] Observe that some overlap will be enivitable, e.g., even if we are to cover a unit disk D with disks of radius $\rho < 1$, we have that the limit (as $\rho \to 0$) of the ratio of the area of the disk D and the sum of the areas of all the covering disks is $\frac{3 \cdot \sqrt{3}}{2\pi}$ (c.f. [6]).

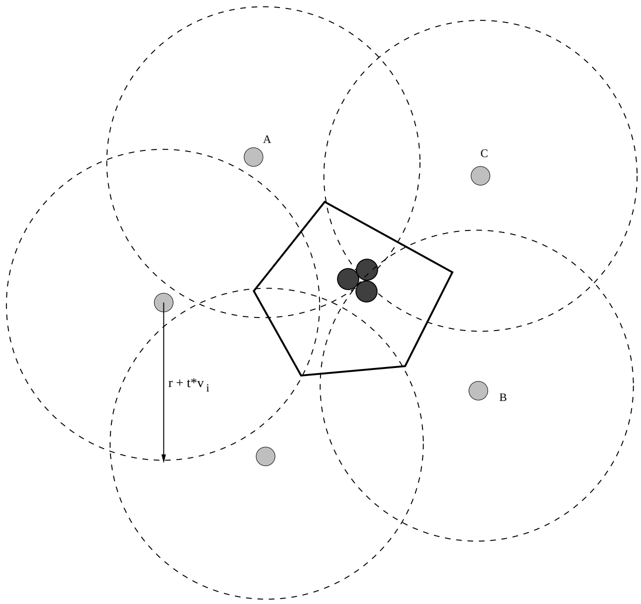

Figure 2: Spatial Coverage of the Critical Region

valid and dashed circles indicate the boundaries that a particular sensor can reach for a valid reading within time t and, again, we don't show it for the vertices of C_R.

We propose two heuristic solutions. The first one, which is trying to cater the worst case, can be explained as follows. At the time $t = cct$, at which *all* points of C_R can be reached by some sensor, pick a point that was reached (covered) last – this point takes *longest* time, but will *have to* be covered. Remove the corresponding covering disk(s) and repeat the procedure to the leftover of C_R. This is illustrated by the disks A, B and C in Figure 2, where the dark disks indicate their final positions and are to be removed from the cover. If the sensors are well distributed, we expect that we can cover all the other points within time *cct*. This may not be the case however, since removing the corresponding disk increases the *cct* of the remaining C_R. This can happen if some point previously reached before *cct* by the removed sensor now has to be reached by another sensor and if there is not close-enough sensor, the new *cct* will increase. This heuristic is also not guaranteed to give a optimal number of covering sensors, since it starts from the center and works the covering towards the boundary of C_R. In fact, we expect it to yield twice the optimal number of sensors. Note, however, that it will not be worse than four times the optimal number of sensors, since the centers of the disks are chosen outside the union of the already chosen final disk positions (a standard packing/covering argument implies that the halved circles with the same center are disjoint, and an area-based argument proves the claim).

Our second heuristic tries to address precisely the problem of minimizing m. Essentially, the above solution is suboptimal because it only looks at sensors locally, and one by one. Trying to look at the entire picture, we can decide a priori the final location of the sensors by computing a minimal covering in the shape, say, of a honeycomb, or using an incremental algorithm to add the disks one by one. Note that standard arguments, similar to the one above, can be used to imply that the number of disks in such a packing is within a small constant ϕ from the optimal number. Next, we compute a Euclidean minimum matching [15] between the sensors and the final positions, which tells us which sensors go where. As a last step, in order to "refine" the solution, we may even want to apply a local perturbation scheme for the purpose of optimizing the critical covering time *(cct)*, while retaining the covering property[4]. All of the above can be carried out in $O(mn)$ time. As a last observation, let us point out that using more sensors than the number m found by the cover, we can expect to lower the value of *mrt*.

3.2 "Fair" Coverage of C_R with Limited Number of Sensors

We finish this section with yet another variant of the problem. So far we assumed that there are *enough* resources available for a coverage of C_R. In other words, depending on the value of the valid coverage area r of an individual sensor's readings, we assumed that the value of m is large enough. However, another realistic assumption is that we have a limit on m, somewhat similar to the settings in Section 2. In such cases, the question becomes *how* to select the locations for each of the m sensors inside C_R so that we can guarantee that, whenever needed, any point within C_R can be reached by one of the m sensors within "reasonable time". Again, we will have a subsequent step to handle, which is, which of the n sensors in $\mathcal{S}$ should be the ones to be placed in the chosen m locations inside C_R. More formally, now we have to solve

Problem FC:
Given: a critical region C_R; an integer m;
Goal: determine the locations of a set of m points $\mathcal{P} = \{ p_1, p_2, ..., p_m\}$ in the interior of C_R such that the *time* for which every point on inside or on the boundary of C_R can be reached by a sensor located at some p_i is minimized;
together with its "next stage" of selecting *which* m sensors should be guided in each p_i so that the *mrt* is minimized too (Euclidean minimum matching [15] again).

To handle **FC** we obtain a fair distribution by finding a value r' ($r' \geq r$) such that C_R can be covered with at most m disks of radius r'. Once we have determined the locations of the centers of the m disks, placing a sensor in each center ensures that even C_R is not entirely covered, any point not covered can be

[4] Observe that in this solution the number m of sensors needed to cover is essentially dictated by the value of r – the radius of validity of sensors' readings.

covered by moving one of the sensors by the smallest amount possible (i.e., in minimal time). Again, we need $O(mn)$ to carry out the solution.

4 Related Literature

MOD researchers have addressed many aspects of interest for management of spatio-temporal data. Largest efforts were made in the area of indexing a collection of moving objects for a purpose of efficient query processing, however, MOD-related problems turned out to have many challenging aspects: modeling/representation based on different ontologies and algebraic types; linguistic aspects; novel query types and their processing algorithms [14]. In this work, we addressed a novel aspect of a "semantic-based" management of moving objects were the semantics of the problem was motivated by the settings of sensor data management.

Mobility aspects in a data-motivated settings have been addressed from perspective of ad-hoc and P2P networks. However, most of the works are targeted towards organizing structures which would ensure a dissemination of information (communication) and effectiveness of routing [21]. Some Computational Geometry techniques (dual space transformation) have been employed for efficient tracking of the mobile sensors which enables efficient communication and power management [11]. Our work is, in a sense, orthogonal to the existing results because we focused on the guidance of a set of mobile sensors for the purpose of quality assurance of the data read by those sensors in a given geographic region.

Two works which are close in spirit to ours are presented in [8] and [17]. In [8] the authors consider some spatio-temporal correlation with the quality of the data read. They introduce the notion of *swarms*, which are nodes with higher processing capabilities than the regular sensor nodes and address the problem of efficient guidance of the swarms towards the location(s) of a *hot* static sensor(s). Our work is complementary to the ones in [8] – we address the problem of ensuring that there are enough many sensors brought in a given critical region. On the other hand, [17] considers the problem of limited transmission range and arrangements of the nodes in ad-hoc network which will ensure probabilistic bound on connectedness. However, we consider the aspect of limited range of the sensor readings for different quality criteria.

5 Concluding Remarks and Future Work

We have addressed the problem of the efficiency of ensuring some quality of data-readings by a set of mobile sensors in a given critical geographic region . We presented different variations of the problem and derived algorithms for their solution. Currently we are focused on obtaining some comparative experimental results about our heuristics and we are looking for an efficient solution(s) to the (variants of) the "opposite" problem – given a set of *hot* sensor, what is the optimal way of guiding all of the sensor in the critical region *outside of it*, subject to some quality requirements, e.g., there must be at least m sensors remaining on the approximate hull of the critical region.

The work that we presented here is part of a larger research effort that we are currently undertaking in the area of context-aware MOD. Our MOD database consists of information about mobile users, e.g.,their motion plan, preferences, etc, information about static objects of interest, as well as the information collected by the various sensors. This database is maintained in a distributed fashion, with the current sensor data being kept by various sensor nodes and the historical sensor data being accumulated at some sensor servers which can be mobile themselves, in a similar spirit to the concept of *swarms* (c.f. [8]). Our system handles continuous queries and notifications which need to be re-evaluated when there are some changes in the motion plans of the users or in the environmental context. The sensor data falls into the environmental context dimension and this data is used in order detect which objects in the users' database need to be notified of the changes in the environment or which objects' trajectories need to be modified accordingly. For example, some unusually high temperatures and low winds detected by the sensors are used to detect a fire. The system will then check if there are any outstanding requests for user notifications that need to be triggered. In this case, only the users who have requested to be notified of a fire within a certain geographic area are notified. We observe here that the individual readings of the sensor nodes need to be aggregated at the coordinating sensor servers, so that some intelligent reasoning can be performed there, such as the fact that a fire has been detected. Thus, the sensor data can be viewed as consisting of a number of a number of data cubes, each having at minimum the time and location dimensions.

The "correlation" we considered in this paper was between the (critical) geographic region determined by the set of sensors which simply report a value past certain threshold and the number of sensors in that region. However, we envision a lot of interesting topics in the field of sensor data management which can benefit from the extensions of some of the existing works in the database research and pose challenge for database researchers:

• Uncertainty – The problem of imprecision of the values in the MOD with respect to the real-world values of the entities represented has been addressed both in the context of modeling and processing nearest-neighbor and range queries [19, 20]. The problem of imprecision of sensor data has also been tackled in [5]. What are the consequences when the uncertainties in both context dimensions *(location,time)* and *data* values are brought together? What are the queries that can be posed and how can they be processed?

• Data reduction – Although not explicit, the problem of data reduction can be viewed as a "flip-side of the coin" of uncertainty management. Reducing the size of the data set with deterministic bounds on the query error has been addressed independently in the MOD settings [4] and the stream-like database settings where the number of passes over the data should be minimized and yet the sample retained should exhibit a bound on the query-errors [9]. What is the impact of the difference of the context dimensions (semantics of the data read vs. location and time of the sensor) on the algorithms which could reduce the size of the data kept in a database?

• Computational Geometry Techniques – In this work we have already utilized some results from CG literature. Is there are room for more collaborative results between the database and the CG researchers in the context of sensor data management? We believe so – to a large extent. In particular, one of the immediate extensions of our work is how to manage the *mobile swarms* (c.f. [8]) and *mobile sensors* in the context of quality of reading and processing of sensor-generated data can readily be categorized as a "mobile version" of the clustering problem.

References

[1] J.D. Boissonat and M. Yvinec. *Algorithmic Geometry.* Cambridge University Press, 1998.

[2] H. Brönnimann and M.T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete and Computational Geometry*, 14, 1995.

[3] J. Hill, R. Sczeczyk, A. Woo, S. Hollar and D.C.K. Pister. System architecture directions for networked sensors. In *ASPLOS*, 2000.

[4] H. Cao, O. Wolfson and G. Trajcevski. Spatio-temporal data reduction with deterministic error bounds. In *MobiHoc*, 2003.

[5] R. Cheng and S. Prabhakar. Managing uncertainty in sensor databases. *SIGMOD Record*, 32(4), 2003.

[6] S.R. Finch *Circular Coverage Constants.* Cambridge University Press, 2003.

[7] M.R. Garey and D.S. Johnson. *Computer and Intractability: a Guide to Theory of NP-Completeness.* W.H.Freeman, 1979.

[8] M. Gerla and K. Xu. Multimedia streaming in large-scale sensor networks with mobile swarms. *SIGMOD Record*, 32(4), 2003.

[9] M. Dash H. Brönnimann, B. Chen and Peter Scheuermann. Efficient data reduction with ease. In *SIGKDD*, 2003.

[10] W. Hong J. Hellerstein and S. Madden. The sensore spectrum: Technology, trends and requirements. *SIGMOD Record*, 32(4), 2003.

[11] L. Guibas J. Liu, P. Cheung and F. Zhao. A dual-space approach to tracking and sensor management in wireless sensor networks. In *WSNA*, 2002.

[12] L. Liu, editor. *SIGMOD Record*, volume 32. ACM Press, 2003.

[13] L. Liu, editor. *SIGMOD Record*, volume 33. ACM Press, 2004.

[14] M. Koubarakis, T. Sellis et al. (editors). *Spatio-Temporal Databases: The CHOROCHRONOS Approach.* Springer (LNCS 2520), 2003.

[15] http://cs.smith.edu/ orourke/TOPP/P6.html.

[16] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 2000.

[17] D.M. Blough P. Santi and F. Vainstein. A probabilistic analysis for the range assignment problem in ad hoc networks. In *MibiHOC*, 2001.

[18] E. Pitoura and G. Samaras. Locating objects in mobile computing. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(4), 2001.

[19] D.V. Khalashnikov R. Cheng and S. Prabhakar. Querying imprecise data in moving objects environments. *IEEE-TKDE*, 16(7), 2004. (to appear).

[20] G. Trajcevski, O. Wolfson, K. Hinrichs and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems (TODS)*, 2004. (to appear).

[21] Z.J. Haas, V. Ramasubramanian and E.G. Sirer. Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *MobiHOC*, 2003.