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. As usual, 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 to a server or a peer. However, the key assumptions considered in this work are: 1. each sensor is capable of moving (where the motion may be remotely controlled); and 2. the spatial range for which the individual sensor's reading is guaranteed to be of a desired quality is limited. 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 there 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 (see [21] for a survey) and ranges to many aspects of modeling, efficient storage and retrieval, and processing of a novel types of spatio-temporal queries in the field commonly known as 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, eventually, transmitting it within a limited range. They may also have some limited processing power and can be mobile and deployed in a certain geographical area. Networks of sensors have already been deployed in the real world [13] and very active research efforts are being undertaken both in industry and academia [14]. 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 presented in [17] and [18].

Although mobility in sensor networks has been addressed in the context of communication protocols for ad-hoc and peer-to-peer networks (e.g., [16, 22]), 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. The motivational scenario is the one of disaster management in a homeland security setting and can be described as follows. Assume that a set $\mathcal{S}$ of *mobile* sensors is de-

*Research partially supported by NSF grant IIS-0325144

Proceedings of the First Workshop on Data Management for Sensor Networks (DMSN 2004), Toronto, Canada, August 30th, 2004.
`http://db.cs.pitt.edu/dmsn04/`

[1] An recent collection of results is presented in [15]

ployed in a large geographic area in order to monitor particular value(s) of interest, e.g., the temperature and the concentration of toxic materials in the air. In case a certain subset $\mathcal{S}_k$ of the sensors, co-located in a given region, report readings which exceed a given (pre-defined) tolerance threshold, we would like to ensure the quality of readings of the sensors' data for the *critical region*. In order to do so, we would like to ensure that there are enough many sensors inside that particular critical region and our goal is to optimize the guidance of a subset of m sensors from $\mathcal{S} \setminus \mathcal{S}_k$ towards the interior of the critical geographic region.

Throughout this work, we investigate few variations of the problem of optimizing the guidance of the set of sensors towards the critical region, in order of their increasing difficulty. 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. Section 2 formally introduces the terminology used. In Section 3 we introduce the concept of *critical times* with respect to the guidance of mobile sensors and we present three variants of the problem of reachability with respect to the critical region. These variants are used as a basis for the problems addressed in Section 4, where we have more realistic requirements of the placement of the sensors within some optimal time-frame and we also consider the spatial limit on the validity of the data read by a particular sensor. Section 5 presents yet another variation of the problem of optimizing the guidance of the set of mobile sensors with respect to a critical geographical region which, in a sense, is the "opposite" of the problems presented in Sections 3 and 4. Section 6 gives a brief overview of the relevant literature and in Section 7 we present concluding remarks and we outline some areas for the future research work.

2 Preliminaries

In this section we formally introduce the terminology used in the rest of the paper.

We assume that we are given a set of distributed mobile sensors $\mathcal{S} = \{s_1, s_2, \ldots s_k\}$, where each s_i is represented as an ordered 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. Typically, the sensors motion plan for the future (future trajectory) or the past completed motion (past trajectory) can be represented as a polyline in 3D space: (x_{i1}, y_{i1}, t_{i1}), ..., (x_{in}, y_{in}, t_{in}) [24], however, without loss of generality, we omit this modeling aspect from the paper.

Each sensor s_i periodically reports the reading val_i of the value of the physical phenomenon that it is observing. Let C_v to denote a value which is a tolerance-threshold for the monitored physical value.

Definition 2.1 *A sensor s_j is called* hot *if it reads a value $val_j > C_v$.*

Assuming that, at a certain time instance, a subset of the set of the sensors have read values greater then C_v, we have the following definition:

Definition 2.2 *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 $\mathcal{S}_k$ (also denoted as $C_R(\mathcal{S}_k)$) is defined as the convex hull of the set of 2D points $\{(x_{j1}, y_{j1}),\ (x_{j2}, y_{j2}),\ \ldots,\ (x_{jk}, y_{jk})\}$.*

Defining the critical region as a convex hull of the locations of the hot sensors is justified by several "natural" properties (c.f. [20]):

- The convex hull is the convex polygon with the smallest area which encloses a set of (planar) points.
- The convex hull is the convex polygon with the smallest perimeter. which encloses a set of (planar) points.
- The concept of the convex hull and the algorithms for its computation are very well studied topics in the field of Computational Geometry.

The concepts introduced in Definition 2.1 and 2.2 are illustrated in Figure 1. The hot sensors are indicated by the dark disks and the white disks indicate the sensors which not hot. As mentioned in Section 1, for the purpose of ensuring certain *quality* of the readings of the sensor-data, we would like to ensure that there are "enough many" (application specific) sensors in the critical region and we are focusing on minimizing the time that it takes to deploy them. In the rest of the paper, we present the algorithms which handle different variants (due to different constraints/requirements) of the problem of optimal deployment of mobile sensors in a given critical region.

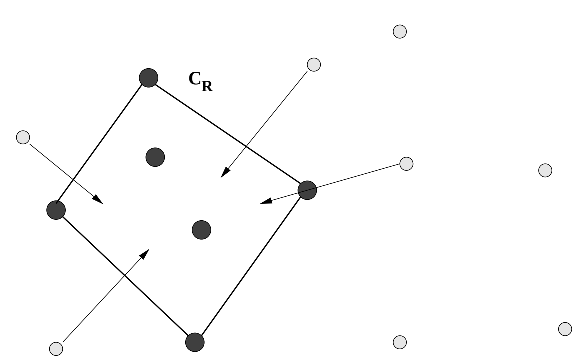

Figure 1: Guiding the sensors towards the critical area

3 Critical Times

Now we proceed with a few variations of the first category of problems of optimizing the deployment of sensors in a given critical region. In this section we address issues related to arrival of mobile sensors in the interior of the critical region which, as it turns out, are important for the settings of the problem(s) that we consider in Section 4.

3.1 Minimizing the Arrival Time

The simplest variation of the problem is the one which assumes that in order to ensure the desired quality of the sensor data (i.e., the desired coverage of the critical region) it suffices to have m sensors in the interior (or, on the boundary) of C_R. For this case, 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 a scenario where we have six hot sensors and, in order to ensure the desired coverage of C_R we need a total of ten sensors. Thus, we need to bring four more sensors inside C_R, in a quickest possible manner. The problem can be formally stated as follows:

Problem 3.1 Minimal Arrival Times (MAT):
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 $C_R(\mathcal{S}_k)$.*

Let $k = \mid \mathcal{S}_k \mid$ and observe that if $m \leq k$ we have already satisfied the quality requirements. Let t_{ai} denote the minimal time-value for which the sensor $s_i \in \mathcal{S}$ can reach the boundary of C_R, which we will call its *critical arrival time – cat_i*. Obviously, if $(x_i, y_i) \in C_R$, then $cat_i = 0$. 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 hot sensors (the ones 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 with k edges[2] can be achieved in $O(\log k)$ [20]. 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 reduces to a linear ($O(n)$) time.

3.2 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 for the purpose of ensuring the quality of the readings is different. Formally:

Problem 3.2 Minimal Furthest-Point Reachability (MFR):
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 each of the m sensors has reached the furthest point (with respect to its current location) in the interior of $C_R(\mathcal{S}_k)$.*

Let t_{fi} denote the minimum time-value 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 the *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 more desirable (and more complicated) setting of optimizing the critical time.

3.3 Critical Covering Time (cct)

The formal statement of this problem is specified as follows:

Problem 3.3 Minimal Interior Reachability (MIR):
Given: *An integer m; a set of sensors $\mathcal{S}$; and a subset $\mathcal{S}_k \subseteq \mathcal{S}$;*
Goal: *Minimize the time cct for which it can be guaranteed that there exists a subset $\mathcal{S}_m \subseteq \mathcal{S}$ of m sensors that can be brought in the interior of $C_R(\mathcal{S}_k)$ in such a manner that any point inside $C_R(\mathcal{S}_k)$ can be reached by some sensor in $\mathcal{S}_m$ in time $\leq cct$.*

Obviously, the goal of the **MIR** problem is to minimize the time-value for a given m – the number of sensors which ensures the quality of the readings in a given critical region. However, one may very naturally be interested in the dual optimization problem, which can be formulated as:

[2] The convex hull of the set of k points is a polygon which may have up to k edges/vertices [2, 20].

Problem 3.4 Minimal Number of Sensors (MNS):
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 [10]. Even this particular instance (disk-covering in 2D) is NP-complete, although it can be approximated within a constant factor [5], 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 **MIR** 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 $O(n)$ [3, 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 **MIR**). The exact value can be determined in $O(n\text{polylog}n)$ time by designing a parallel version of the decision algorithm and using parametric searching.

4 Spatial Limits on the Validity of Readings and Sensors Placement

Based on the results presented in Section 3, in this section we present a more stringent set of requirements, which are more realistic for practical purposes. The key assumption is that the readings of each sensor are *valid* only within a *limited* area, which is represented as a *disk* with radius r centered at a sensor's location-point. Building up on the results in the previous section, first we will discuss a variation in which we assume that there are enough sensors to cover the critical region C_R and, subsequently, we address the more realistic setting of limited number of available sensors.

4.1 Full Coverage of C_R

Instead of having m sensors inside C_R, our goal now is to minimize the time for which C_R can be 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 follows:

Problem 4.1 Minimal Full Coverage Time (MFC):
Given: *A set of sensors* $\mathcal{S}$ *and a region* C_R*;*
Goal: *Determine the minimal time (denote it mrt – minimal routing time), 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} and (recall that) we do not even have the limit for m set in advance. 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:

Problem 4.2 Time-Limited Full Coverage (TFC): Given: *A set of sensors* $\mathcal{S}$*; a region* C_R*; time-value (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 3 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 critical coverage time – *cct* (c.f. Section 3).

This is illustrated in Figure 2, where for simplicity we have assumed that the only hot sensors are on the vertices of C_R (we do not indicate their coverage area). White disks indicate the initial location and the area in which the readings of a particular sensor are valid and dashed circles indicate the boundaries that a particular sensor can reach for a valid reading within time t.

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

[3]Observe that some overlap will be inevitable, 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. [9]).

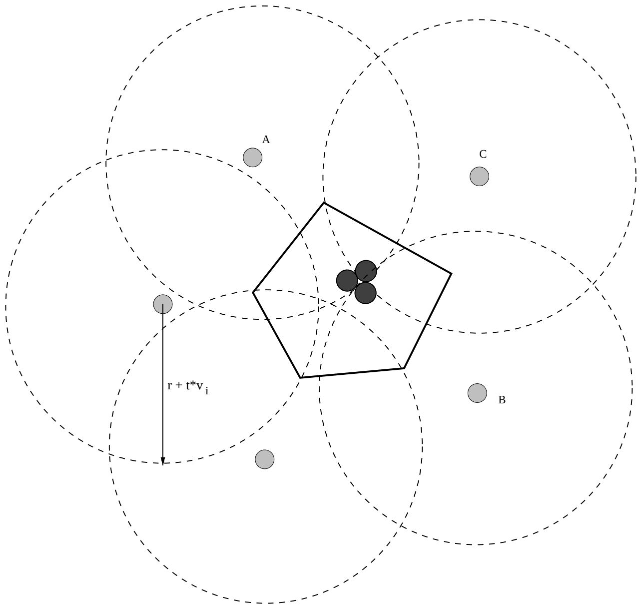

Figure 2: Spatial Coverage of the Critical Region

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 needs to be reachable by another sensor. In such cases, 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 justifies 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 [19] 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*.

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

The last variant of the problem of covering the critical region corresponds to the realistic settings of having limited resources available. The initial assumption for this section was that there are *enough* sensors available for the 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 so that C_R is fully covered with m (partially overlapping) disks. However, similar to the scenarios considered in Section 3, we now assume that we have a limit on m – the number of sensors that can be deployed inside C_R, each with some valid area of its readings. In this case, 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 4.3 Fair Coverage Problem (FC):
Given: *A critical region C_R and 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;*
Plus, 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 [19] 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 covered by moving one of the sensors by the smallest amount possible (i.e., in minimal time). Again, we need $O(mn)$ time to carry out the solution.

[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.

5 "Potpourri": Escape From the Critical Region

Now we briefly turn our attention to the "inverse-image" of the problems considered in the previous two sections. Namely, instead of optimizing the deployment of a sufficient number of sensors inside the critical region, we analyze the case when one would actually want to ensure that the sensors from the interior of the C_R (both hot and non-hot ones) are guided outside C_R as soon as possible. Such setting is of interest for scenarios like, for example, when the values read by the hot sensors could indicate that there may be a fire in a certain geographic area.

Again, we assume that the readings of each sensor are valid within a disk-like area with radius r, centered at the sensor's location-point. In order to ensure some quality of the monitoring of the values along the critical region's boundary, we would like to guide each individual sensor at distance r from the boundary of C_R. Formally, the problem that we address in this section can be stated as follows:

Problem 5.1 Minimal Escape Time (ET):
Given: *A critical region C_r and a subset of sensors $\mathcal{S}_{CR} \subseteq \mathcal{S}$ such that each $s_a \in \mathcal{S}_{CR}$ has its location-point inside (or, on the boundary of) C_R.*
Goal: *Minimize the time that it takes to move all the sensors from $\mathcal{S}_{CR}$ at distance r from C_R.*

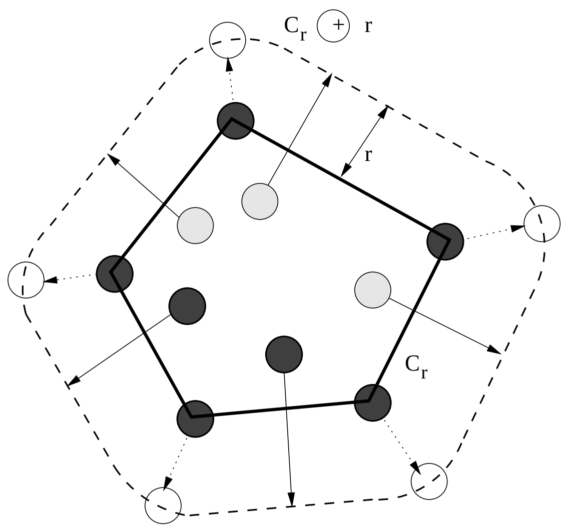

Figure 3: Escaping from the critical region

The illustration is provided in Figure 3. Solid disks indicate the initial locations of the (hot and non-hot) sensors inside C_R. Since the quality of the readings of each sensor is guaranteed within a disk of radius r, for safety, we want to guide the sensors at distance r from C_R's boundary. For that, we first need to determine, what is commonly called, the *Minkowski Sum* of the region C_R with a disk with radius r. We will use $C_R \oplus r$ to denote the operation of Minkowski Sum and informally[5] it can be described as the region obtained when the disk with radius r is "swept" along the boundary of C_r. The empty disks in Figure 3 indicate the final locations of the sensors which were initially the vertices of C_R.

Since the construction of the convex hull (C_R) is assumed to yield a polygon with $O(k)$ edges/vertices (in time linear in k), the construction of the Minkowski Sum of a convex polygon (C_R) with a disk with radius r can be done in $O(k)$ (c.f. [2]). Similarly to the discussion in Section 3 (**MAT** problem), for each sensor's location-point in C_R, we can find the closest point on the boundary of $C_R \oplus r$ in $O(\log k)$. Assuming that (worst case) initially all the sensors from $\mathcal{S}$ were in C_R, the time-complexity of the algorithm for solving the **ET** problem is $O(n \log k)$.

In case certain quality of the sensor's readings needs to be ensured by placing a given number of sensors on the boundary of $C_R \oplus r$, we can apply some of the variations of the guidance problems that we considered in Section 3 and 4, respectively.

6 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 [15]. 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 [22]. Some Computational Geometry techniques (dual space transformation) have been employed for efficient tracking of mobile sensors, which enable efficient communication and power management [16]. 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 [11] and [23]. In [11] the authors consider

[5] Formally, the Minkowski Sum of two sets of points P_1 and P_2 can be specified as $P_1 \oplus P_2 = \{p_1 + p_2 \mid p_1 \in P_1, p_2 \in P_2\}$, where the summation is of *vector* p_1 with *vector* p_2 (c.f. [2]).

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, in a sense, complementary to the one in [11] – we address the problem of ensuring that there are enough many sensors brought in a given critical region. On the other hand, [23] 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 the purpose of ensuring different quality criteria.

7 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 focusing on obtaining comparative experimental results for our heuristics.

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. [11]). 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 to 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 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, one may observe that we did not consider the issues of the limited communication range and the limited mobility (e.g., a road-map in an urban environment), which are parts of our ongoing work.

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 can, in turn, pose challenges 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 [7, 24]. The problem of imprecision of sensor data has also been tackled in [8]. 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 explicitly, 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 [6] 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 [4]. 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 total 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 challenges of our results is the efficient management of mobile critical regions (e.g., the fire is spreading dynamically). The MOD researchers have already addressed the issue of algebraic modeling of moving polygons (c.f. [12]) and the CG researchers have already addressed the issue of incrementally computing the convex hull of a set of moving points with known motion plans (c.f. [1]). Another extension of our work is how to manage the *mobile swarms* (c.f. [11]) and *mobile sensors* in the context of quality of reading and processing of sensor-generated data, which can readily be categorized as a "mobile version" of the clustering problem.

References

[1] P. Agarwal, L. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. In *WADS*, 1997.

[2] P. K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of minkowski sums. *Computational Geometry*, 21(1-2), 2002.

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

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

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

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

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

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

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

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

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

[12] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. Lorentzos, E. Nardeli, M. Schneider, and J. R. R. Viqueira. Spatio-temporal models and languages: An approach based on data types. In *Spatio-Temporal Databases: the Chorochronos Approach*. 2003.

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

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

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

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

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

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

[19] J. O'Rourke. http://cs.smith.edu/ orourke/TOPP/P6.html.

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

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

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

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

[24] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems (TODS)*, 2004. (to appear in 29(3)).