

Contour Estimation Using Collaborating Mobile Sensors

Sumana Srinivasan
Kanwal Rekhi School of Information Technology
Indian Institute of Technology Bombay
Powai, Mumbai - 400076
sumana@it.iitb.ac.in

Krithi Ramamritham
Kanwal Rekhi School of Information Technology
Indian Institute of Technology Bombay
Powai, Mumbai - 400076
krithi@it.iitb.ac.in

ABSTRACT

A mobile wireless sensor network may be deployed to detect and track a large-scale physical phenomenon such as a pollutant spill in a lake. It may be called upon to provide a description of a contour characterized by an isoline of a specific concentration value. In this paper, we examine the problem of tracing a contour of a particular concentration within a bounded region of varying pollutant concentration using a network of mobile sensors. Since controlled movement of sensors within a given region is known to improve the overall quality of measurements by reducing sensing uncertainty, we explore various ways of guiding a set of mobile sensors optimally so as to surround and trace the contour. We formulate the contour estimation problem as a nonlinear multi-extremal optimization problem. We use accuracy and latency as performance metrics and show that in majority of the cases our proposed strategy based on collaboration of sensors delivers the best performance.

Categories and Subject Descriptors

C.2.1 [Computer - Communication Networks]: Network Architecture and Design - Wireless Communication; C.3 [Special-purpose and Application Based System]: Real-time and Embedded Systems; J.2 [Physical Sciences and Engineering]: Earth and Atmospheric Sciences

General Terms

Algorithms, Design, Performance, Measurement

Keywords

Mobile Wireless Sensor Networks, Contour Level Set Estimation, Performance Metrics, Latency, Accuracy

1. INTRODUCTION

Wireless sensor technology has emphasized the importance of *in-situ* [1] measurements that could potentially re-

duce the error in measurement and increase our understanding of large scale physical phenomena like contaminant flow [2]. They are ideal for deployment in adverse settings such as explosion plumes, oil slicks etc.

Consider a pollutant spill in a water body. One of the first tasks in spill response is to contain the slick and prevent it from spreading. In order to do that, there is a need to estimate the spatial extents of hazardous areas in the spill as well as track the movement of the spill. A common query for a wireless sensor network deployed in such a scenario is to estimate the spatial extent of contour of a particular concentration (a contour of concentration T can be visualized as a boundary that separates regions with concentration higher and lower than T). For example,

- Select all the spatial coordinates for the contour of concentration 120 units, with 90% confidence.

Tracking contours also helps in determining the rapidity of flow of contaminants and this results in providing an early warning for sensitive areas located in the vicinity of the spill.

One way to architect a wireless sensor network to accomplish this task is to mount sensors on permanent moorings (static network) in the region where the spill has occurred and have each sensor measure the concentration of the pollutant at the location where it is mounted. An energy efficient algorithm would then choose appropriate nodes in the network to estimate the contour with measurements made at the chosen nodes. On the other hand, one can also make use of mobile sensors, e.g. sensors mounted over rover buoys to move, sample and measure at different locations in the region. Use of mobile sensors

1. Improves sampling resolution — They can access those areas in the region which are unreachable for a static sensor network.
2. Eases deployment — Sensors can be dropped off in the region where measurements are taken and the sensors “intelligently” move and sample the region.
3. Increases adaptability to spatio-temporal dynamics of the phenomenon — Redeploying a static sensor network to do the same might be prohibitively expensive.

However, the main limitations of mobile in-situ sensors are

1. Higher latency in estimating the contour — The sensors take time to arrive at the contour and then trace the contour.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIWANS'06, September 26, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-471-5/06/0009 ...\$5.00.

2. Additional noise due to odometry errors in addition to actual sensing errors.
3. Higher energy consumption due to mobility.

An energy efficient algorithm for mobile sensors therefore must optimize accuracy, latency and movement while computing the contour. In this paper, we propose and evaluate different movement strategies for a mobile wireless sensor network which is assigned the task of estimating a contour of a specified concentration in a region of varying pollutant concentration.

In Section 2 we describe the problem and our approach to the solution. In Section 3 we provide the details of the ingredients that go into our solution. In Section 4 we discuss mobility model, assumptions and the movement strategies used in our solution and follow it up with the description of our simulation set up in Section 5. We discuss the results in Section 6 and related work in Section 7. We conclude and give directions to our future work in Section 8.

2. OVERVIEW OF THE PROBLEM AND OUR APPROACH

Let R denote a bounded geographical region where the pollutant spill has occurred. Let $T \pm \epsilon$ denote the concentration of the target contour. In the region, let N mobile wireless sensors, each capable of measuring concentration at their current location and in a small neighborhood around their current location, be deployed. The task is to locate points on the contour such that the error in estimation as well as the time taken to estimate is minimized. Initially the sensors may find themselves anywhere in the region both inside as well as outside the contour.

There are several approaches to solve the above mentioned problem. In [12], the authors propose a mobile sensor to scan through the region to get an estimate of the boundary within a given error bound. However, instead of using a single mobile node, can multiple nodes be used? Can knowledge about the field characteristics and information from other nodes be used to arrive and trace the contour with a lower latency and a higher accuracy than a scan? In this paper, we propose an alternative scheme where multiple sensors collaborate to arrive at the contour. Sensors start from their initial positions and begin to move in a direction such that they approach the contour as well as distribute themselves around it. We define this phase of sensor movement as the *Converge Phase*. After it arrives at the contour, each sensor moves along the contour locating the points on the contour during its traversal. This phase is defined as the *Coverage Phase*.

Once the sensors are deployed as shown in Figure 1 they communicate their initial location information to each other and they embed themselves in a ring i.e., one of the sensors, say, S_1 is chosen as a lead sensor and the one closest to it (S_2) is chosen as its anti-clockwise neighbor. Further, S_2 picks the sensor (S_1 not inclusive) closest to it (S_3) and assigns it as its anti-clockwise neighbor and so on. The entire region R is divided into N sections centered at a known anchor point inside the contour (in our simulation we use centroid of the contour to be the anchor point¹). If θ_1 is

¹The anchor point or the centroid is approximated to be very near to the source of spill.

the current angle² with respect to the centroid as the origin for sensor S_1 , then the target angle of the i^{th} sensor S_i in the ring is given by $\theta_1 + \frac{2\pi(i-1)}{N}$. The target angle for each sensor is its angle of approach towards the contour. Another approach to ensure the spread of the sensors is the distance based approach where, each sensor moves as far away as possible from its neighboring sensors. The advantage of using the angular approach for achieving spread as opposed to using the distance based approach is that, the angular approach requires communication between the sensors only at the beginning when the angles are being computed while the distance based approach requires the sensors to communicate their locations at every step. In Figure 1, the sensors' initial positions are marked $\{S_1, \dots, S_5\}$. At the end of Converge phase, the corresponding positions are $\{C_1, \dots, C_5\}$.

In order to minimize the error in contour estimation and the overall time taken to perform the estimation, each sensor needs to move in a direction such that it lies on the contour at the appropriate target angle at the end of the converge phase. One approach is to favor that direction which minimizes the difference between the concentration at the current location and the target contour concentration T , and also minimizes the difference between the angle with respect to the centroid at the current location and the target angle. This is modeled as benefit of movement and it is associated with every possible direction of movement allowed per iterative step. The benefit of movement is zero or minimum when the sensor locates itself on the contour at the correct angle of approach. We derive the benefit mathematically in the next section.

In coverage phase, the sensors that converged on to the contour trace the contour in a non-overlapping fashion. One possible way is for the sensors to move towards their respective neighbors in a preset direction (e.g., anti-clockwise). Consider an arrangement of five sensors (S_1, \dots, S_5) at the end of converge phase as shown in the second figure in Figure 1. The sensors embed themselves in a ring, i.e., each sensor discovers its anti-clockwise neighboring sensor's positions (just like in the beginning of the converge phase described earlier). The sensors then move along the contour towards their neighbors recording the locations of all the points on the contour *en route*. For that, each sensor needs to know its neighbor's location also known as *target point* for the sensor. The angle with respect to the centroid of the target point is the target angle of approach for each sensor in the coverage phase. As shown in the third diagram in Figure 1, it may not always be correct to terminate movement as soon as the sensor approaches its target angle (point P is not the target point but is a point on the contour at the target angle) but terminate the movement only when the sensor arrives at its target point along the curve.

3. INGREDIENTS OF THE SOLUTION

In this section we describe the cost model for mobility and performance metrics. We begin by making a few simplifying assumptions.

²If (x_c, y_c) represents the location of the centroid, then the current angle of the i^{th} sensor at position, (x_i, y_i) is computed as $\theta_i(x_i, y_i) = \tan^{-1} \frac{y_i - y_c}{x_i - x_c}$

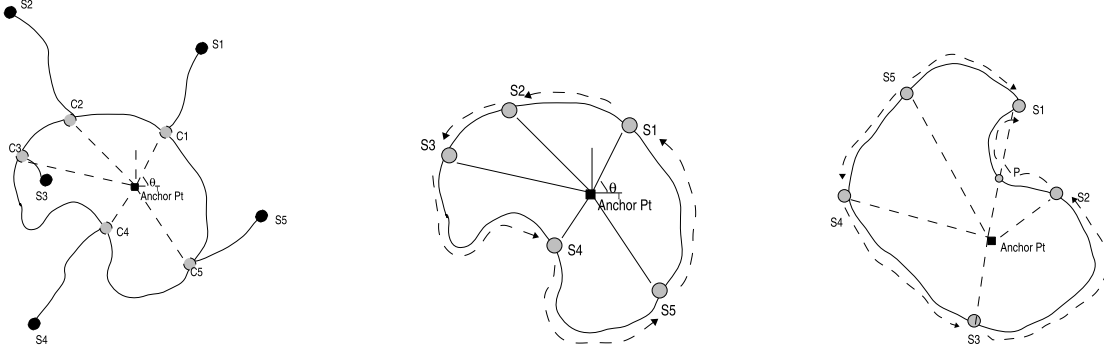


Figure 1: Movement in Converge and Coverage Phases

3.1 Assumptions

1. Region R is discretized and is represented as a two dimensional square grid with side length l and unit grid granularity.
2. Contour is a well defined (not diffused) closed curve and approximated by the grid points.
3. Contour exists in the region and its concentration value T is known to the sensors. Typically, the threshold of hazardous concentration values are known and the sensors can be deployed to search for the contour corresponding to this threshold. We consider all points in a band $T \pm \epsilon$ to be the contour.
4. An interior point in the contour referred to as *anchor point* is known to the sensors.
5. There are no sensing or odometry errors. In reality, sensors have sensing as well as odometry errors. We assume the sensor reading to be the expected value of several readings taken in the same location. We ignore odometry errors in this paper.
6. All sensors can communicate with each other (maximum transmission range per sensor, $r_{trans} = \sqrt{2}l$).
7. The sensors are aware of their current location.
8. A sensor can sense concentrations at its current location and also at its neighboring grid locations. We assume zero cost for exploring the neighbor locations.
9. The distance travelled by the sensor per iteration is one unit on the grid (the cost of moving to the diagonal neighbors or the immediate neighbors is assumed to be the same for the sake of simplicity).
10. There is enough energy available for each sensor to move n_{iters} steps each in the converge and coverage phases of movement.

3.2 The Cost Model

The benefit of moving to a particular neighboring location is modeled as cost at that location. The cost is defined per grid position per sensor. The cost c_i for the i^{th} sensor has two components namely, an *attractor* component that is responsible for attracting the sensor towards the contour and a *spread* component that enables the sensor to approach its target angle. Hence the cost $c_i(x_i, y_i)$ is a function of

- Difference between the concentration of the pollutant at the current position (x_i, y_i) and concentration at the contour T .
- Difference between the current angle (angle with respect to the centroid at the current position) and the target angle.

The *attractor* component a_{cost} is given by

$$a_{cost}(x_i, y_i) = (1 - \frac{f(x_i, y_i)}{T})^2 \quad (1)$$

where

- $f(x_i, y_i)$ — pollutant concentration at a given position (x_i, y_i)
- T — pollutant concentration at the contour.

Note, $a_{cost}(x_i, y_i) = 0$ when $f(x_i, y_i) = T$.

The *spread* component s_{cost} is derived as follows. Let,

- $\theta_i(x_i, y_i)$ — angle with respect to the centroid for the sensor at its current position (x_i, y_i) .
- θ_t — the target angle of approach for the i^{th} sensor (all angles are measured in radians).

If,

$$\theta_d(x_i, y_i) = \theta_i(x_i, y_i) - \theta_t \quad (2)$$

then s_{cost} ³ is given by

$$s_{cost}(x_i, y_i) = (\frac{\theta_d(x_i, y_i)}{2\pi})^2 \quad (3)$$

Note, $s_{cost}(x_i, y_i) = 0$ when $\theta_i(x_i, y_i) = \theta_t$ at the given point (x_i, y_i) .

The total cost $c_i(x_i, y_i)$ at a given position for the i^{th} sensor is a weighted combination of $a_{cost}(x_i, y_i)$ in Equation (1) and $s_{cost}(x_i, y_i)$ in Equation (3). For the i^{th} sensor, the cost at any given grid point (x_i, y_i) is given by:

$$c_i(x_i, y_i) = \alpha * \underbrace{a_{cost}(x_i, y_i)}_{attractor} + (1 - \alpha) * \underbrace{s_{cost}(x_i, y_i)}_{spread} \quad (4)$$

where, $(0 \leq \alpha \leq 1)$ is the *biasing factor*, $(x_{min} \leq x_i \leq x_{max})$ and $(y_{min} \leq y_i \leq y_{max})$ where $[(x_{min}, y_{min}), (x_{max}, y_{max})]$ are bounds of region R . Note that, $c_i(x_i, y_i) = 0$ when

³We divide by 2π in order to normalize.

$a_{cost}(x_i, y_i) = 0$ and $s_{cost}(x_i, y_i) = 0$. In practice, there may be more than one zero cost point per sensor if we consider $T \pm \epsilon$ instead of only T as points on the contour.

Given the cost model, the next task is to determine how a given sensor should move in each iterative step to arrive at a minimum cost (≈ 0) point. Let us examine the amount of information the sensor has at every step to make an informed decision about its next direction of movement.

- The sensor can measure (by moving and sampling) the concentration values at the grid points in its neighborhood ($f(x_i \pm 1, y_i \pm 1)$) within the bounds of the region R .
- The sensor knows *a priori* the bounds of region R , $\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$, target concentration T , target angle of approach θ_t and the coordinates of the anchor point (x_c, y_c) .

The sensor does not have any knowledge as to whether it is inside or outside the contour. The sensor needs to make a decision as to which one of its neighboring points to choose as a next step in its path towards a point of zero cost. The choice of the neighboring point is dependent upon the mobility strategy. There are points where the cost evaluates to be the smallest amongst all their neighbors but without being the smallest possible cost in the entire grid. Algorithms for mobility should deal with this situation, else the sensor will fail to converge on to a zero cost point.

Next we discuss the metrics for evaluating mobility strategies.

3.3 Metrics

The performance of any mobility strategy depends on the accuracy of estimation and the time taken for estimation. The accuracy is a measure of how well the sensors estimated the contour when compared to the actual contour. One way to measure this is to define the accuracy in terms of the difference between the actual contour and the estimated contour. This difference is the actual error in estimation. Approximating the contour with its bounding polygon is concise in representation and also sufficient for applications where contours are estimated for bounding pollutants as opposed to exactly matching the contour. We define the metrics used in our simulation as follows.

- **Relative Contour Error (RCE)** : is defined to be the relative difference in the area between the polygon formed with the points of the actual contour and the polygon formed with the points on the estimated contour. The formula is as shown below. Let,
 A_{act} — area of the polygon of actual contour
 A_{est} — area of the polygon of estimated contour

$$RCE = \frac{|A_{est} - A_{act}|}{A_{act}} \quad (5)$$

- **Latency (L)**: is defined as the maximum number of steps on the grid taken by the sensors to estimate the contour. Since the energy consumed is directly proportional to the distance travelled, latency is a measure of maximum energy consumed by a sensor due to movement. If,
 t_i — Number of steps taken by the i^{th} sensor

M — Number of sensors converged at the end of the converge phase ($M \leq N$) then,

$$L = \text{argmax}_i(t_i) \quad (6)$$

where, $i = \{1, 2, \dots, M\}$

In the next section, we present three different mobility strategies for movement of the sensor.

4. MOBILITY STRATEGIES

Consider the case where the cost at every point in the region is known to the sensor. Let us assume that these cost values are not changing dynamically. Then the problem at hand reduces to finding the shortest path from the starting point to a zero cost point on the contour. When the entire field is unknown to the sensor and only the field at neighboring locations are known to the problem solver (each sensor in our case), then the problem takes an online or distributed form.

We present three different mobility strategies to address this online problem.

4.1 Basic Steps

We will begin by outlining the common steps first and then describe the three different strategies.

- **Input** — Number of sensors (N), location of anchor point (x_c, y_c) , target contour concentration T , α (biasing factor – in the converge phase both a_{cost} and s_{cost} components are biased equally) and bounds of region R ($\{(x_{min}, y_{min}), (x_{max}, y_{max})\}$).
- **Output** — Points on the contour $(x_1, y_1), \dots, (x_k, y_k)$.

Steps 1–5 outlines the steps involved in the converge phase.

- **Step 1** — Deploy the sensor nodes in the region.
- **Step 2** — Sensors embed themselves in a ring.
- **Step 3** — For each sensor,
 1. **Step 3a** — Compute c_i at all of its neighboring grid points.
 2. **Step 3b** — Move to a neighboring point depending on the movement strategy as described in Sections 4.2, 4.3 and 4.4..
 3. **Step 3c** — If the current location is indeed the zero cost point, terminate movement else go back to Step 3a.
- **Step 4** — If all sensors have terminated or if number of iterations equal the maximum allowed (n_{iters}), terminate all sensors.
- **Step 5** — This step denotes the end of converge phase. All those sensors who failed to converge onto the contour are eliminated from the next phase.

Steps 6 – 9 outlines the steps involved in the coverage phase.

- **Step 6** — One of the converged sensors is assigned to be the lead sensor (chosen at random). At the end of this phase, all other sensors send their estimates of the points on the contour to the lead sensor. The converged sensors discover their anti-clockwise neighbors and assign their target angles to be that of their neighboring sensor's angle with respect to the centroid. In addition, the neighboring sensor's location (*target point*) is also noted to determine the termination condition.
- **Step 7** — α (the biasing factor for the cost components) is reset to be highly biased towards a_{cost} so that the sensors do not stray away from the contour (In Equation 3, c_i is high for all those points that do not lie on the contour).
- **Step 8** For each sensor,
 1. **Step 8a** — Compute c_i at all of its neighboring grid points.
 2. **Step 8b** — Move to a neighboring point depending on the movement strategy. Record the point visited.
 3. **Step 8c** — If the current location is indeed the zero cost point, and the distance between the current location and the target point is zero, then terminate movement else go back to Step 8a.
- **Step 9** — The converged sensors send all their estimated points to the lead sensor and the result is output.

4.2 Greedy Algorithm

In the Greedy Algorithm (GA) approach, the sensor moves to the neighbor with least cost. If this point has been visited by the sensor before, then the sensor is trapped at a local minimum⁴ and terminates its movement. If the sensor is neither trapped at local minimum nor has arrived at the edge after a maximum number of iterations (n_{iters}), then the sensor terminates. Thus for each move,

- **Step 3b.1** Compute least cost point amongst all the neighbors.
- **Step 3b.2** If the new position has not been visited before then move to the new position else terminate the movement and got to Step 4.

4.3 Simulated Annealing

In Simulated Annealing (SA) approach [4], the sensor chooses a neighboring point at random. If the cost at the new point is lower than the current cost, it moves to its new position. Otherwise it moves to the new position (bad move) with a certain probability, P_{bad} .

$$P_{bad} = e^{-\frac{\delta c}{T}} \quad (7)$$

If $P_{bad} \geq r \in [0, 1]$ then the bad move is made. Temperature is reduced depending on the cooling schedule chosen (linearly or geometrically with number of iterations) to achieve convergence. The sensor is terminated when it arrives at the contour or a predetermined number of maximum

⁴A sensor is said to be trapped in local minimum if it visits a point more than once.

iterations (n_{iters}) is executed. In this approach, the sensors are guaranteed to move out of local minima due to randomization. The following steps form the SA strategy.

- **Step 3b.1** Pick a neighboring point at random and compute the cost at this neighboring point.
- **Step 3b.2** If the cost at this neighboring point is smaller than the cost at the current location, move to the new position else move to the new position with probability, as computed as described in Equation 7.

4.4 Collaborative Algorithm based on Minimizing Centroid Distance

In the two strategies, Greedy Algorithm and Simulated Annealing discussed above, the sensors choose the direction of movement based on local information alone. They do not communicate. In Simulated Annealing technique the sensors do not get trapped in local minima since they always jump out with a certain probability. However, this technique has a disadvantage that it makes quite a lot of bad moves initially. Given that the sensors are capable of communication, the question one can ask is: can the sensors collaborate with each other to correct their course of movement when trapped in local minima? In this context, we propose a strategy called MCD, that attempts to provide course correction for the sensor trapped in local minimum by computing the centroids of the convex hulls formed by all the other sensor locations and neighboring points (if there are eight neighboring points, the centroid of the convex hull formed by each of these points with all other sensor locations is computed). The algorithm then chooses that neighboring point that minimizes the distance between the centroid of its hull (known as *hull centroid*) and the actual centroid point known to the sensor (prior information). This strategy is inspired by the technique of computing the maximum overlap between convex polygons under translation as described in [5] in the field of geometric algorithms. Part A in Figure 2 depicts the course correction during the converge phase when a particular sensor is trapped in a local minimum and the trajectory of the hull centroid. The hull formed by the sensors at each of these time steps and the respective path of the hull centroid are depicted.

Given the knowledge of the anchor point, it is logical to question why any given sensor should use gradient information (characterized by field cost) to approach the contour instead of moving directly towards the anchor point. Consider the case where the sensor is nearer to a point on the contour but far away from the point of intersection of the shortest path to the anchor point and the contour as shown in part B in Figure 2. Then, the gradient information allows the sensor to converge on to the contour more rapidly than approaching the anchor point. Hence there is a need to use a hybrid approach like MCD, where, the sensor uses gradient information to move until it hits a local minimum and uses anchor point to get out of the local minimum.

The following steps describe the MCD strategy.

- **Step 3b.1** Compute least cost position amongst all the neighbors.
- **Step 3b.2** If the least cost position has not been visited before then move to the new position and go to step 3c

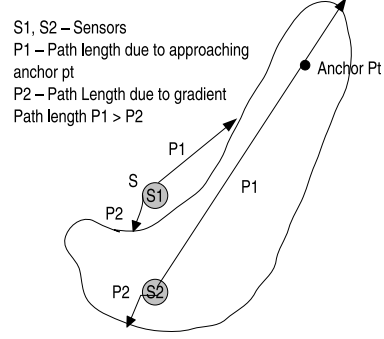
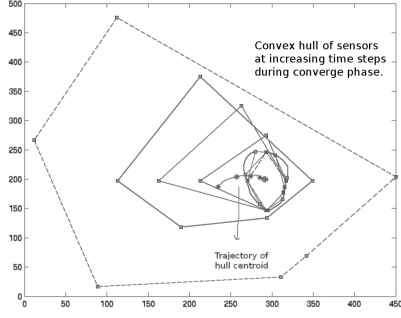


Figure 2: A: Course Correction during Converge Phase, B: Gradient Vs. Anchor Pt path comparison

- **Step 3b.3** Else, for each neighboring point (x_r, y_r) ,
 1. Compute the convex hull of all other sensor locations and (x_r, y_r) and determine the centroid of this hull.
 2. Compute the distance between the hull centroid and the actual centroid (x_c, y_c)
- **Step 3b.4** Move to the neighboring point with minimum distance between the hull centroid and the actual centroid.

In the next section, we describe our simulation set up and measure the performance of each of the mobility strategies described above.

5. SIMULATION

In our simulation, we assume homogeneous sensors (sensors with similar characteristics). In this paper we focus on the effect of varying number of sensors, type of deployment and type of contour on *RCE* and latency for the three different movement strategies and for a given sensor field.

For obtaining *R*, we used a sensor field of varying pollutant concentration generated by a pollutant flow modeling tool, *WQMAP^{TM5}*. The sensor field was generated by running the simulation for 120 time steps. The output generated by *WQMAPTM* was imported into a GIS tool, GRAM++⁶ and rasterized to generate a two dimensional pollutant concentration field of dimensions (500, 500). Interpolation and triangulation were performed to generate the grid and the contours. Figure 3 depicts the pollutant concentration field and the concentration isolines as generated by *WQMAP* and GRAM++.

The biasing factor α was chosen to be 0.5 in the converge phase so as to encourage the sensors to move towards the contour as well as the target angle. We experimented by varying α from 0 to 1 and observed that *RCE* was not significantly sensitive to α (in spite of $\alpha = 0$, MCD exhibited a low *RCE* indicating that once the sensors reached their target angles, choosing the direction that minimized the distance between the hull and the actual centroids resulted in the sensors converging onto the contour). However, latency showed

some variation with α and at $\alpha = 0.5$, both the SA and the MCD algorithm displayed a low latency value. The plot that depicts the sensitivity of the performance parameters to α has been omitted due to lack of space. In the coverage phase α was chosen to be 0.9995 so as to discourage the sensors to stray away from the contour while they approached the target angle at the target point. The sensors terminated when the cost c_i was less than a pre-specified threshold (10^{-8}). We chose two contours as representatives of convex and non-convex contours, two types of deployment scenarios, random, where the sensors were uniformly distributed in the region and regular, where sensors were deployed in a circle with varying radii. The maximum number of iterations per phase was $n_{iters} = 5000$ per phase per simulation (a simulation includes: deployment of sensors, movement until all the sensors terminate or a pre-determined number of iterations have been executed in each phase). For the random deployment case in all the three strategies and for all types of deployment in the SA technique, ($n_{sim} = 1000$) simulations were performed to obtain the average behavior. In the MCD case, we used the Graham Scan algorithm [6] whose time complexity is of the order $\Theta(n \log n)$ to compute the convex hull.

6. RESULTS

We studied the variations of *RCE* and latency with different strategies for movement, type of deployment, type of contour and varying number of sensors. For each case, we varied one and fixed all other parameters.

6.1 Variation of Performance with Mobility Strategies

Figure 4 shows the variation of *RCE* and latency for different movement strategies for a given configuration. For this experiment, the number of sensors was chosen to be $N = 10$, the type of deployment was random and the contour type was convex. The MCD algorithm had the least *RCE* and latency for the given configuration. SA performed the worst since the sensors did not converge for many random deployment configurations in the simulations and whenever they did, the latency was high.

6.2 Variation of Performance with Deployment

Next, we measured the variation of *RCE* with different deployments. The configuration comprised of $N = 10$ sensors and the contour type was convex. The graph in Figure

⁵Applied Science Associates Inc., <http://www.appsci.com/>

⁶GRAM++: Full runtime GIS supporting Vector And Raster map data creation, query, analysis and layout application, <http://www.csrie.iitb.ac.in/gram++/>

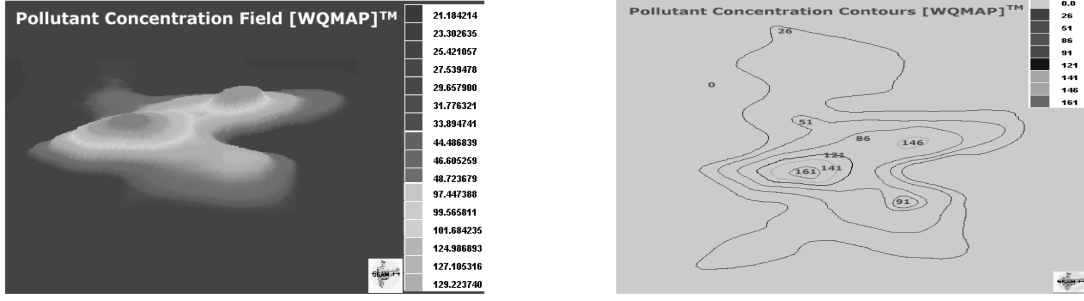


Figure 3: Pollutant Concentration Field and Contours

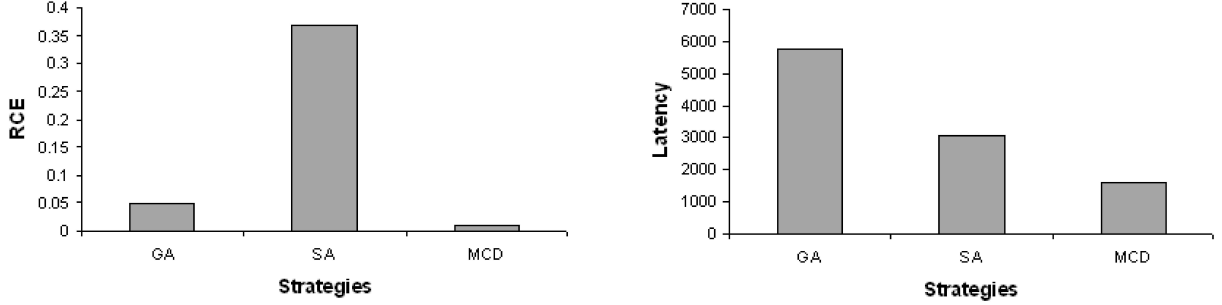


Figure 4: Performance Vs. Mobility Strategies for $N = 10$, Random Deployment and Convex Contour

5 indicates that the lowest RCE for all deployments is obtained by the MCD algorithm. The Greedy Algorithm has the worst RCE when the sensors were deployed far away from the contour. The reason for that is, many sensors got trapped in local minima and could not converge onto the contour resulting in a large RCE. The Simulated Annealing algorithm displayed large RCE for random and far deployments indicating that the sensors did not converge onto the contour in these cases. In summary, we see that RCE is low for the MCD algorithm for all the deployments. That is, MCD algorithm did not display sensitivity to the type of deployment in this configuration.

Figure 5 indicates the latency was the least for the regular deployment when the sensors were deployed inside the contour for the MCD algorithm. Therefore for low latency requirement, using MCD algorithm as the movement strategy with regular deployment is preferable. In summary, the MCD algorithm had least value for RCE and latency compared to the other strategies and was found to be sensitive to deployment.

6.3 Variation of Performance with Number of Sensors

Next, we studied the variation in performance for random deployment and convex contour and the results are as shown in Figure 6. For the MCD algorithm, except for the case where $N = 1$, the RCE values were uniformly small as N varied. This indicates that the MCD algorithm is not sensitive to varying number of sensors for RCE in the given configuration. Greedy Algorithm and Simulated Annealing strategies exhibited high RCE values for lower number of sensors and RCE decreased as number of sensors were increased. This indicates that the MCD algorithm achieves high accuracy of estimation with a lower number of sensors.

The graph for latency variation in Figure 6 indicates that SA displayed lowest latency values for higher values of N . We notice that RCE and latency for the MCD algorithm is indeed small even with a small number of sensors and does not vary much with increase in number of sensors. This implies that a high accuracy can be attained with a smaller number of sensors for MCD. However, as the number of sensors increases, RCE and latency decrease sharply for SA and the latency for MCD shows a slight increase as shown in Figure 8. This behavior can be better explained by counting the number of sensors that actually converged in each strategy at the end of converge phase. Our simulation indicated that this number was the highest for MCD. Since latency is the maximum path length of converged sensors, it tends to be high for MCD algorithm since on an average more sensors converged on to the contour. However, when large number of sensors are deployed, it may be sufficient for only a fraction of the sensors to arrive at the contour and trace it. We experimented by terminating the movement of all sensors when half the deployed sensors converged at the contour and measured the RCE and latency. In this case, the MCD algorithm had a significantly lower latency and RCE and performed better than the SA algorithm. The plot for these observations has not been provided due to lack of space.

6.4 Variation of Performance with Different Contours

Figure 7 depicts the variation of the metrics with different contours for random deployment and $N = 10$ sensors. The MCD algorithm has the overall lowest RCE and latency for various contours. This indicates that the MCD algorithm is not sensitive to the shape of the contour.

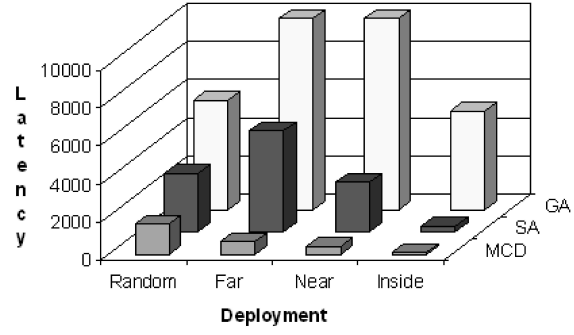
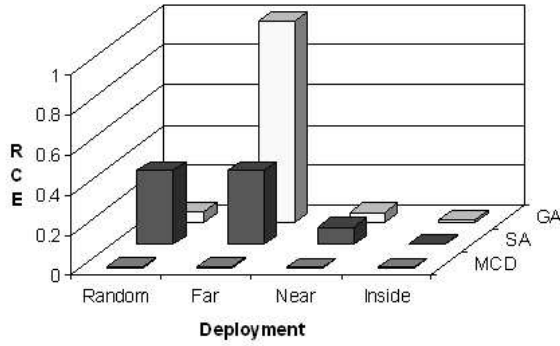


Figure 5: Performance Vs. Deployment for N=10 and Convex Contour

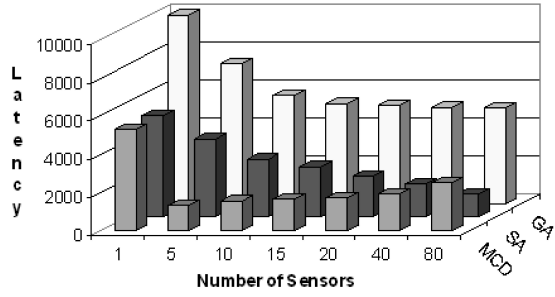
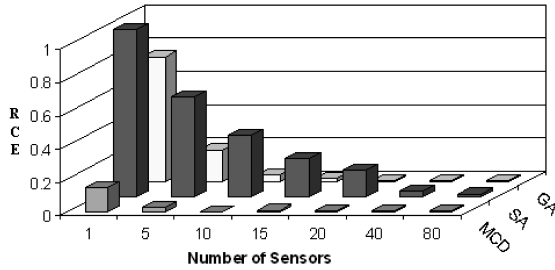


Figure 6: Performance Vs. Number of Sensors for Random Deployment and Convex Contour

6.5 Summary

Table 1 summarizes mobility strategy recommendation for various combinations of parameters considered in this paper. High accuracy requirements imply low RCE. The MCD algorithm has the lowest RCE for all deployments and contour types. Low energy available for mobility translates to low latency requirements. MCD algorithm displays overall low latency values for regular and random deployment. In environments where regular deployment is possible, using MCD algorithm as a movement strategy would reduce latency irrespective of the type of contour, with smaller number of sensors. However, in those cases where only random deployment is possible, restricting the number of sensors that need to converge before the coverage phase begins reduces the latency and MCD algorithm works out to be a better strategy. We also measured the mean square deviation (SPMSE - Shortest Path Mean Square Error) of the path length in converge phase (number of steps) of those sensors that converged onto the contour (over n_{sim} simulations) from the geometric shortest path length (we measured the distance from the initial starting point of the sensor to the point where the sensor landed at the end of converge phase on the contour). We found that the MCD exhibited a smaller deviation when compared to SA as shown in Figure 8. For most cases, the deviation was found to be between 20 - 40%.

7. RELATED WORK

Boundary detection and estimation using a network of static sensors has been studied extensively in the recent past [3]. The authors in [7] derive a theoretical bound on the number of sensors needed in a lattice network of static sen-

sors to achieve a certain accuracy. In [8] the authors explore the use of mobile sensors to improve the quality of measurement by ensuring that there are enough sensors in a pre-specified critical region. The task is to push more sensors into the critical region in the shortest possible time. However in our scenario, the position of the points on the contour is not known to the sensors. The work in [9] and [10] is close in spirit to ours where the authors use a static sensor network to guide a swarm of mobile sensors. In [9], the mobility is modeled based on group mobility vector (direction of the swarm as a “whole”) and individual mobility vector (generated using random wave-point model but within the group boundary). In this paper, we follow a different approach where in the sensor movement direction is not random but determined by the gradient of concentration in its neighborhood. The sensors do not “know” the destination *a priori*. In [10] the authors propose “RoboMotes” for contour tracking in a scenario where a single mobile node collaborates with a static sensor network for contour detection. However, in our simulation, we use multiple mobile nodes as well as the local gradient information to optimally move towards the contour. In [11], the authors discuss isobar estimation using in-network aggregation in a static sensor network whereas our work focuses on a mobile sensor network.

8. CONCLUSION AND FUTURE WORK

In this paper we considered the problem of estimating a contour of a given pollutant concentration in a region of varying pollutant concentration using controlled mobility of sensors. We defined performance metrics for the mobile sen-

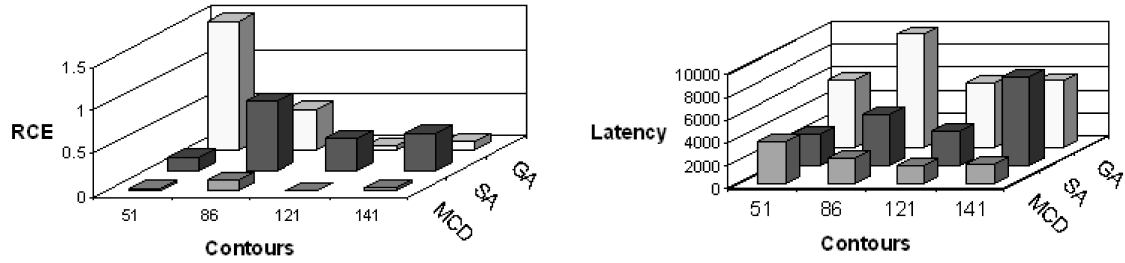


Figure 7: Performance Vs. Contours for $N = 10$ and Random Deployment

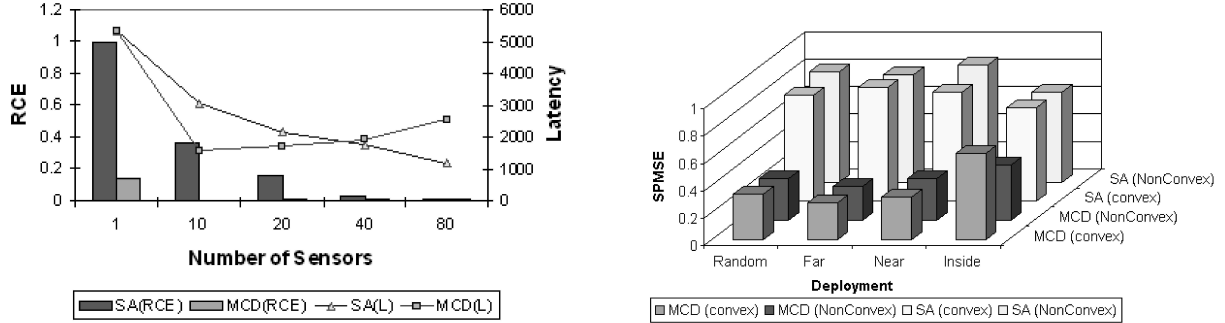


Figure 8: RCE/Latency Vs. Number of Sensors and Deviation From the Shortest Path

sor network and compared the strategies for different combinations of parameters. For the various configurations we considered, the MCD algorithm exhibited the least sensitivity to parameters and had better accuracy and latency values compared to the other strategies.

As part of our ongoing work, we intend to relax the assumption of the knowledge of the interior point and use estimates of an interior point to arrive at the contour. We also intend to introduce communication cost to our mobility model. Finally, we plan to validate our simulations with an experimental test-bed in the near future.

9. ACKNOWLEDGEMENTS

We would like to thank Parmesh Ramanathan for his invaluable support during the course of this work and our colleagues who gave us critical feedback. We would also like to thank GRAM++ group in IIT Bombay for their help in sensor field data generation.

10. REFERENCES

- [1] P.M.Teillet et. al., "A Framework For In-Situ Sensor Measurement Assimilation in Remote Sensing Applications", In Proceedings of the 22nd Canadian Symposium on Remote Sensing, Sainte-Foy, Quebec, pages 111-118, 2001.
- [2] Frank R. Burden, "In-Situ measurements of physico-chemical water quality parameters", McGraw Hill Online Library, DOI 10.1036/0071449663.CH0310.1036/0071449663.CH03
- [3] Krishna Kant Chintalapudi and Ramesh Govindan, "Localized Edge Detection in a Sensor Field", Proc. IEEE Workshop on Sensor Network Protocols and Applications (SNPA), 2003.
- [4] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", Science, Vol 220, Number 4598, pages 671-680, 1983.
- [5] de Berg, Devillers, van Kreveld, Schwarzkopf and Teillaud, "Computing the Maximum Overlap of Two Convex Polygons Under Translations", In 7th International Symposium on Algorithms and Computation, 1996
- [6] R.L.Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, vol. 1, pages 73-82, 1972.
- [7] Robert Nowak and Urbashi Mitra, "Boundary Estimation in Sensor Networks: Theory and Methods", Workshop on Information Processing in Sensor Networks, 2003.
- [8] G. Trajcevski, P. Scheuermann and H. Bronnimann, "Mission Critical Management of Mobile Sensors(or, How to Guide a Flock of Sensors)", DMSN 2004
- [9] M. Gerla and K.Xu, "Multimedia Streaming in large-scale sensor networks with mobile swarms", SIGMOD Record, 32(4), 2003.
- [10] K. Dantu et. al., "Robomote: Enabling mobility in sensor networks," In IEEE/ACM Fourth International Conference on Information Processing in Sensor Networks (IPSN-SPOTS), pages 404-409, Apr 2005.
- [11] J. Hellerstein, W. Hong, S. Madden and K. Stanek, "Beyond Average: Towards Sophisticated Sensing with Queries", Workshop on Information Processing in Sensor Networks (IPSN), 2003.
- [12] A. Singh, R. Nowak and P. Ramanathan, "Active Learning for Adaptive Mobile Sensing Networks", Proceedings of Information Processing in Sensor Networks (IPSN), 2006

Deployment	Number of Sensors	Contour Type	RCE	Latency	Combined
Regular ($d = 200$)	Low ($N = 10$)	Convex ($T = 121$)	MCD	MCD	MCD
Regular	Low	Non-convex ($T = 51$)	MCD	MCD	MCD
Random	Low	Convex	MCD	MCD	MCD
Random	Low	Non-convex	MCD	SA	MCD
Regular	High ($N = 80$)	Convex	MCD	MCD	MCD
Regular	High	Non-convex	MCD	MCD	MCD
Random	High	Convex	MCD	MCD	MCD
Random	High	Non-convex	MCD	SA	SA

Table 1: Suggested movement strategy for different combination of parameters