This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Terrain-Constrained Mobile Sensor Networks

Shu Zhou
The University of New Mexico
Albuquerque, USA

Min-You Wu
Shanghai Jiao Tong University
Shanghai, China

Wei Shu
The University of New Mexico
Albuquerque, USA

***Abstract*— [1] To turn an ideal mobile sensor network simulation model into reality, terrain is one of the inevitable factors we have to consider. This work is the first attempt to address this issue, by studying the behavior of mobile sensors under constraints of terrain. Specifically, we study the power consumption and the loss ratio of mobile sensors when they are moving on different terrains. These two metrics are utilized to select the best destinations, the best paths to the destinations, and match these destinations with mobile sensors.**

I. INTRODUCTION

Along with the rapidly developing technologies on sensing and communication, wireless sensor networks are going to be ubiquitous very soon. Researchers have proposed enormous models and algorithms to represent different aspects of the sensor networks, such as network traffic, power consumption, data mining, etc.

In the physical layer, sensor placement models study location of sensors and its impacts on the network. Currently, researches are focused on power consumption, transmission, network coverage, and exposure [1], [2], [7], [8], [10], [11], [13], [18]. Several models [4], [6], [12], [15], [16], [17] were proposed in recent years, using mobile sensors to improve the network performance. Such wireless sensor network with mobility is called *mobile sensor network* (MSN). Majority of current works of MSN [4], [6], [16], [17] assume all sensors have such mobility, while few others [12], [15] assume that only small fraction of sensors can move, since mobile sensors are much more expensive than traditional stationary sensors.

To actually implement these proposed MSN models, the effects of terrain have to be considered. In most cases, sensors are deployed in a realistic environment. In this work, we focus on terrain's impact on MSN, i.e., the mobile sensors. Specifically:

- Mobile sensors moving along different types of terrain, or crossing different elevation will consume different amount of energy.
- The terrain model defines the loss ratio of mobile sensors, i.e., the *degree of risk*. It varies by different terrain types or elevation.
- If mobile sensors take power consumption and loss ratio into account, they might reconsider their destination selection and path planning.

A new terrain model is required to formulate these constraints.

[1]Min-You Wu is currently on leave from The University of New Mexico, USA

The above discussion motivates our framework of terrain-aware mobile sensor networks. It is based on the partial-mobile sensor network model, focusing on building up the terrain model and studying its effects. We utilize this new model to define a new strategy of mobile sensor movement, which consumes minimum energy, takes minimum risk, and yet gets the maximum performance improvement.

The remainder of this paper is organized as follows. We define the problem in section II. Section III presents our framework and algorithms. Simulation and case studies are presented in section IV. Section V lists related works. Finally, we conclude this paper in section VI.

II. PROBLEM DEFINITION

From the system point of view, there are four main roles in an MSN: events, sensors, observers, and the environment. Sensors detect events, and exchange data with each other through wireless communication; the detection results are transmitted to observers, or stored in some media for later retrieval. All these procedures are performed in the environment, and are constrained by the environment.

Environment has many elements, such as *terrain*, *weather*, *temperature*, etc. In this work, we study the behavior of MSN under the constraints of terrain.

To formulate the terrain, we need to consider both of its geometrical and geographical aspects. Geometrically, the 2D projection of a terrain is an Euclidean plan. Every point i in this plan has a coordination (x_i, y_i). An edge between neighbor (including straight neighbors and diagonal neighbors) points x and y is defined as the straight line connecting them together, denoted by e_{xy}. A path, denoted by p_{ba}, is defined as a set of edges connected one by one from point b to point a. The distance between two points b and a is denoted by $L(b, a)$.

Geographically, a terrain's characteristic can be described through *terrain type*, *elevation*, *obstacles*, *surface drainage*, etc. The two most important factors among them are terrain type and elevation, since they are key factors to constrain the movement of mobile sensors. In this paper, the effects are represented by *cost* and *risk*. When a mobile sensor is moving, the cost represents the energy it consumes, and the risk represents the dangerousness of its moving path.

In this work, we use *detection probability* as the metric to represent performance of an MSN. Compared with commonly used *coverage*, it can measure the detection more precisely, especially in the areas overlapped by multiple sensors.

Define the distance from a sensor s_i to a point a as $d(i,a)$, and the sensing ability of s_i at a as $s(i,a)$. $s(i,a)$ generally diminishes as $d(i,a)$ increases. The specific function of $s(i,a)$ depends on the type of s_i. A Gaussian function can model most generic sensors.

$$s(i,a) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d(i,a)^2}{2\sigma^2}} \tag{1}$$

where σ is a manufacture-depend parameter. Without loss of generality, we use it as the detection function of sensors in our simulation.

Based on the definition of $s(i,a)$, we define the overall detection probability at position a as:

$$S(a) = 1 - \prod_i (1 - s(i,a)) \tag{2}$$

For the entire surveillance area A, the overall detection probability, denoted by S, is defined as:

$$S = \frac{1}{||A||} \sum_{a \in A} S(a) \tag{3}$$

The ultimate goal of the whole MSN is to maximize the detection probability. So the duty of individual mobile sensors is to move to the mostly needed places, getting the maximum detection probability improvement. Thus, the question for mobile sensors is: *where are the destinations?*

In our work, the selection of destinations is based on the potential improvement of detection, cost and risk. Cost and risk are path-related metrics, so we use a path planning algorithm to first find the optimal path with minimum cost and risk. Then a matching algorithm will match mobile sensors with their destinations.

Therefore, we define the problem as:

Given a randomly distributed mobile sensor network MSN, with N_s stationary sensors and N_m mobile sensors, deployed in some terrain,

1) *find the best destinations for mobile sensors, subject to maximize the detection probability.*
2) *find the best paths for mobile sensors, subject to minimize the power consumption and loss ratio.*

III. SYSTEM FRAMEWORK

This framework describes the behavior of an MSN under the constraints of terrain. As discussed in section II, the three important metrics related to a particular mobile sensor's movement are detection improvement, cost and risk. The following subsections will discuss in detail how to define these metrics, and how to use them in path planning and matching.

A. Detection Improvement

Generally, sensor networks are randomly deployed. The locations of sensors are not well planned, thus not optimized. Holes and weak areas of detection are inevitable. These areas are destination candidates of mobile sensors. When a sensor is placed at some position a, it not only improves the detection probability at a, but also neighboring positions. The overall improvement is defined as:

$$I(a) = S^{+a} - S \tag{4}$$

where S and S^{+a} denote the overall detection probability before and after a sensor been deployed at a, respectively.

We use the Gaussian function to simulate the detection ability of sensors. To compute $I(a)$ with reasonable accuracy, only local information about the sensors around position a is good enough.

Consider a mobile sensor s_i moving from point b to a, through path p_{ba}. Denote the risk of this path as $R(b,a)$, then by definition, $1 - R(b,a)$ is the probability of *not losing* mobile sensor s_i. Thus we have the the *expected detection improvement* as:

$$I^*(b,a) = I(a)(1 - R(b,a)) \tag{5}$$

We define the *expected improvement-cost ratio* caused by mobile sensor s_i moving through path p_{ba} as:

$$H_i(b,a) = \frac{I^*(b,a)}{C(b,a)} \tag{6}$$

where $C(b,a)$ denotes the cost of s_i moving through p_{ba}.

B. Cost and Risk

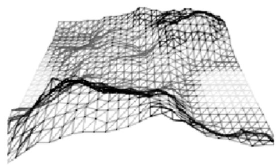
Cost and risk are related to terrain type and elevation. Terrain type defines the type of the ground, and the major vegetation. Since different materials have different coefficient of friction, terrain type has a great effect on the power consumption of mobile sensor movement. Certain types of terrain is also dangerous for mobile sensors, such as water, swamp, sands, etc. Same as terrain type, elevation is also an issue of both power consumption and safety.

The first step of calculating cost and risk is preprocessing the map to compress type and elevation information. Normally the original terrain data is huge, for example, a typical 1:250,000 DEM terrain data [14] is at least several mega bytes in size. It is impractical to store or process such bulk information in a resource-limited sensor. Figure 1(a) is a 3D look of the original terrain data, which defines water, swamp, forest, grass land, dirt, hill, and the elevation. Figure 1(b) and (c) are the abstracted type and elevation data, respectively.

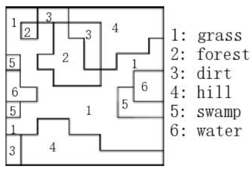
Suppose a mobile sensor s_i is moving along an edge e_{xy}. By the very simple physical model, the force of s_i's engine, denoted by F, should balance all other forces in the opposite direction.

$$F = \begin{cases} m(\mu cos(|\theta|) + sin(|\theta|)) & \text{uphill} \\ m(\mu cos(|\theta|) - sin(|\theta|)) & \text{downhill} \\ m(sin(|\theta|) - \mu cos(|\theta|)) & \text{downhill, brake} \end{cases} \tag{7}$$

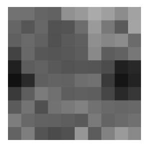
Equation (7) formulates three basic cases of mobile sensor movement, which are moving uphill, moving downhill with engine on, and moving downhill with brake, respectively. m denotes the weight of this mobile sensor, μ denotes the current coefficient of friction and θ denotes the angle of this edge.



(a) Original Terrain



(b) Terrain Type



(c) Terrain Elevation

Fig. 1. Terrain Processing

By the definition of *work*, the cost on this edge, denoted by $c(x, y)$, is:

$$c(x, y) = F \cdot L(x, y) \tag{8}$$

The total cost of a path p_{ba} is:

$$C(b, a) = \sum_{e_{xy} \in p_{ba}} c(x, y) \tag{9}$$

From Equation (7), we can also calculate the boundaries of angle θ in those three cases. The first threshold θ_{t1} is the maximum gradient that a mobile sensor can climb up, due to the limitation of its engine. When a mobile sensor is going downhill, if the gradient is smaller than threshold θ_{t2}, it still needs power from the engine; otherwise, the brake is needed to control the speed. θ_{t3} is the third threshold. If the gradient larger than θ_{t3}, the mobile sensor will be out of control.

$$\begin{aligned}
\theta_{t1} &= sin^{-1}\frac{F + \sqrt{F^2 - (1+\mu^2)(F^2 - m^2\mu^2)}}{m(1+\mu^2)} && (10)\\
\theta_{t2} &= tg^{-1}\mu && (11)\\
\theta_{t3} &= cos^{-1}\frac{\sqrt{F^2\mu^2 - (1+\mu^2)(F^2 - m^2)} - F\mu}{m(1+\mu^2)} && (12)
\end{aligned}$$

Risk is categorized into two classes, elevation-based and type-based. Elevation-based risk comes from the gradient, which may challenge the engine. For an edge e_{xy} with gradient θ, the elevation-based risk is simulated by an exponential function:

$$r_e(x, y) = e^{1 - \frac{\theta_t}{|\theta|}} \tag{13}$$

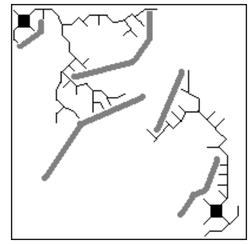
Where θ_t represents the threshold of the angle. It is calculated from Equations (10) or (12), for uphill or downhill edges, respectively.

Type-based risks include water, swamp, sands, etc. We use $r_t(y)$ to denote the type-based risk of edge e_{xy}, which is the risk of terrain type at point y. The total risk of this edge is:

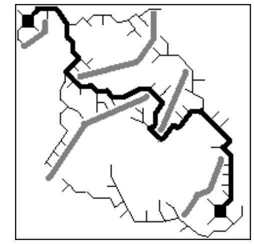
$$r(x, y) = 1 - (1 - r_e(x, y))(1 - r_t(y)) \tag{14}$$

The overall risk of $p_{ba}$is:

$$R(b, a) = 1 - \prod_{e_{xy} \in p_{ba}} (1 - r(x, y)) \tag{15}$$



(a) Trees growing



(b) Complete a path

Fig. 2. Rapidly-exploring Random Trees

C. Path Planning

In this work, we use a path planning algorithm to find the optimal path from a mobile sensor's original location to its destination.

RRT-connect [9] is a simple yet efficient algorithm for solving single-query path planning problems. It works by incrementally building two Rapidly-exploring Random Trees (RRTs) rooted at both the source and the destination. These two trees each explores space around them and also advances toward each other through the use of a greedy heuristic. Figure 2 demonstrates how this algorithm works.

RRT-connect algorithm is not limited to compute length of paths. It works on any objective functions of edges, such as cost, weight, etc. In this work, we want to maximize the expected improvement-cost ratio $H_i(b, a)$. Based on its definition in Equation (6), $I(a)$ is a constant, so the factor $w = \frac{C(b,a)}{1 - R(b,a)}$ is the objective function needs to be minimized.

Paths computed from RRT-connect algorithm have been observed to be relatively short. But they may suffer from loops or self-intersections. So we modify this algorithm by keeping a cache in each intermediate node, to record tree-related information. The cache has five fields:

1) **w**: the objective function we defined above.
2) **f**: true, if this node has been added to a tree, otherwise false.
3) **p**: its parent node, if this node belongs to some tree. It is used to back track the final path.
4) **c**: cost of the path between this node and its root.
5) **r**: risk of the path between this node and its root.

By using this cache, we can not only prevent loops and dead ends, but also save intermediate results for next computation.

Figure 3 describes this algorithm in detail, where a tree rooted at node r is denoted by T_r.

D. Matching

We use the stable marriage algorithm [5] to match mobile sensors with all the destination candidates.

Each mobile sensor first selects its destination candidates as those positions that has big detection probability improvement, and also within the maximum possible range. This range is calculated by assuming that the path between the mobile sensor and its destination is a straight line in plan dirt road. After getting the set of candidates, this mobile sensor will use the modified RRT-connect algorithm to compute the total

expand(tree T_r, Destination point g)
  For every point k of tree T_r
    For every neighbor x of k that is not in T_r
      insert x into set Q_n
      $cost = k.c + c(k, x)$
      $risk = 1 - (1 - k.r)(1 - r(k, x))$
      $w = \frac{cost}{1 - risk}$
      If $w < x.w$
        $x.w = w$, $x.c = cost$, $x.r = risk$, $x.p = k$
  For every node x in Q_n
    estimated total: $\widehat{w} = x.w \cdot \frac{L(r,g)}{L(r,x)}$
  select the node $\widehat{x}$ with minimum $\widehat{w}$
  $\widehat{x}.f = yes$
End function

RRT(Source point g_1, Destination point g_2)
  $T_s = \{g_1\}$, $g_1.w = 0$, $g_1.p = NULL$, $g_1.f = yes$
  $T_d = \{g_2\}$, $g_2.w = 0$, $g_2.p = NULL$, $g_2.f = yes$
  While $T_s \cap T_d = \Phi$
    expand(T_s, g_2)
    expand(T_d, g_1)
End function

Fig. 3. Modified RRT-Connect Algorithm

cost and risk of each potential path. Positions that are not reachable due to the power limitation are eliminated from the candidate set. Finally, the traditional stable marriage algorithm is called to match between mobile sensors and their candidates. The expected improvement-cost ratio $H_i(b, a)$ is used as the *preference* in this matching. Figure 4 shows the details of this algorithm.

For every mobile sensor s_i originally at position b
  insert s_i into set Q_m
  For every position a within s_i's maximum range
    If $I^*(b, a) > I_t$ (I_t is a predefined threshold)
      add a into s_i's destination candidate set Q_i
  For every a in Q_i
    $RRT(b, a)$
    delete a from Q_i, if it is not reachable
destination set $Q_d = \cup Q_i$
call stable marriage to match between Q_m and Q_d

Fig. 4. Match Mobile Sensors and Destinations

After this step, each mobile sensor s_i will have its final destination a. We call this *a match between* s_i *and* a.

The stable marriage is not necessarily an optimal solution, but can at least guarantee that there is no better result if two pairs exchange their matches.

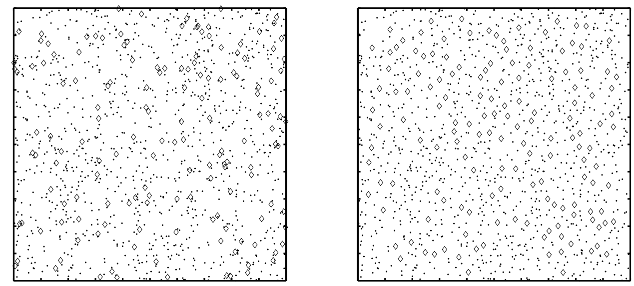
IV. SIMULATION AND CASE STUDY

We build up a testbed that simulates real surveillance scenarios. The MSN is deployed in a 1 km^2 ground. For both stationary and mobile sensors, the detection ability is designed to follow Gaussian function in Equation (1), where $3\sigma = 20m$.

We first randomly throw 1000 stationary sensors into this area, then another 150 mobile sensors. These sensors are uniformly distributed. For each mobile sensor, the weight is 200g, the total power supply is 5000 joule, and the output power is 0.3 watt. These parameters are selected based on the currently available sensor and battery products.

TABLE I
RISKS OF DIFFERENT TERRAIN TYPES

Terrain Type	swamp	grass	water	dirt	forest	hill
risk/grid	0.05	0.004	1	0.002	0.015	1



(a) Before mobile sensors move (b) After mobile sensors move

Fig. 5. Sensor Distribution

The terrain is generated by terraform [3], a small program that can generate random terrain maps. The risks for different type of terrain are listed in Table I. Risk ranges from 0 to 1 with 1 as the highest risk (100% loss probability). The threshold for destination selection (I_t) is 0.02%.

Figure 5 demonstrates our framework, by comparing the sensor distribution before and after mobile sensor movement. In these two plots, gray dots represent stationary sensors, and black diamonds represent mobile sensors. We can see that these mobile sensors are at the positions that are mostly needed, i.e., the open areas. After throwing in about 15% more sensors, there are hardly any empty spaces left.

Initially, the overall detection probability of 1000 stationary sensors is 80.77%. We first randomly deploy 150 mobile sensors into this network. Before they move, the overall detection probability is increased to 88.74%. The overall detection probability becomes 92.27% after mobile sensors moving to their destinations. It changes to 93.08%, when terrain constraints are ignored. Figure 6 plots the overall detection probabilities for different numbers of mobile sensors. The performance with terrain constraints does not increase as fast as that without the constraints, but still has remarkable advantage over the performance of pure stationary sensors.

Table II further compares the average risk, cost and $\frac{1-R}{C}$ of 150 mobile sensors, after they select their destinations and plan the corresponding optimal paths. Here R denotes the average risk, and C denotes the average cost. The data set are for cases with and without terrain constraints. It shows great differences between these two cases. This again emphasizes the notable effects of terrain, that are not neglectable.

V. RELATED WORKS

Early researches of sensor placement and deployment are based on stationary sensors. S. Meguerdichian *et al.*[11], [10]

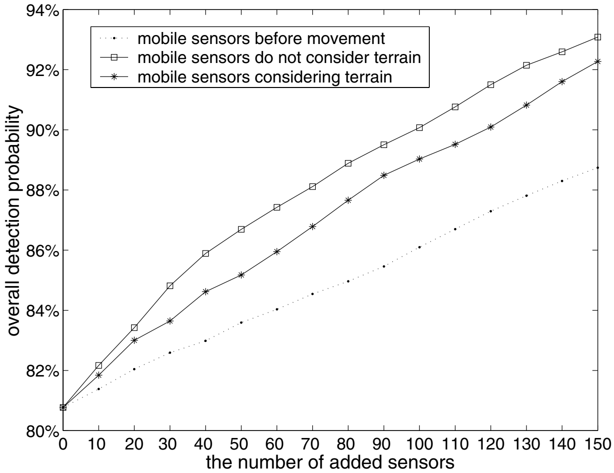


Fig. 6. Overall Detection Probabilities

TABLE II
TERRAIN'S EFFECT ON COST AND RISK

	R	C	$\frac{1-R}{C}$
without constraints of terrain	0	571	0.00175
with constraints of terrain	0.3387	946	0.00070

studied exposure. S. Shakkottai *et al.*[13] studied coverage and connectivity. K. Chakrabarty *et al.*[1] analyzed grid coverage by using a distributed algorithm. S. Dhillon *et al.*[2] worked on imprecise detection coverage. C. Huang *et al.*[7], [8] discussed coverage problem in both 2D and 3D scenarios. Y. Zou *et al.*[18] studied uncertainty-aware coverage.

Later, researchers started to work on mobile sensor networks. At first, they assume all sensors are mobile. Y. Zou *et al.*[17] proposed an algorithm using virtual forces to move sensors. A. Howard *et al.*[6] and G. Wang *et al.*[16] moved sensors for maximum coverage. D. Goldenberg *et al.*[4] used mobile sensors to optimize network transmission. Recently, partial mobile sensor networks have also been studied. S. Zhou *et al.*[12] studied connectivity of a partial mobile sensor network. G. Wang *et al.*[15] used a mixture of mobile and stationary sensors to improve overall coverage. However, none of these works considered the constraints of terrain.

VI. CONCLUSION AND FUTURE WORKS

In this work, we propose a terrain-aware framework of mobile sensor networks. It studies the behavior of the MSN under the constraints of terrain, and presents methods and algorithms to get the best detection improvement. Simulations and case studies show that this model can provide more accurate solutions, save energy, and reduce the risk of losing mobile sensors. It is the first attempt to study the real mobile sensor networks under the constraints of terrain.

Based on this work, we are continue studying the impact of terrain on events. Terrain affects not only the mobile sensors, but also the events that need to be detected. Applying these effects into our framework can yield even better results. Other open questions may include studying other factors of terrain that may have impact on MSN, designing better algorithms to do path planning and matching, etc.

Acknowledgments

This research was supported partially by Natural Science Foundation of China grant # 60442004.

REFERENCES

[1] K. Chakrabarty, S. Sitharama, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans on Computers*, 51(12):1448–1453, December 2002.
[2] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *International Conference on Information Fusion (FUSION 2002)*, pages 1581–1587, 2002.
[3] R. Gasch and D. Bartold. Terraform. In *http://terraform.sourceforge.net*.
[4] D. K. Goldenberg, J. Lin, and A. S. Morse. Towards mobility as a network control primitive. In *The Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '04)*, pages 163–174. ACM, May 2004.
[5] D. Gusfield and R. Irving. *The Stable Marriage Problem: Structure and Algorithms*. The MIT Press, 1989.
[6] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem. In *the 6th International Conference on Distributed Autonomous Robotic Systems (DARS 02)*, pages 299–308, June 2002.
[7] C. Huang and Y. Tseng. The coverage problem in a wireless sensor network. In *Wireless Sensor Networks and Applications Workshop*, pages 115–121. ACM, September 2003.
[8] C. Huang, Y. Tseng, and L. Lo. The coverage problem in three-dimensional wireless sensor networks. In *IEEE GlobeCom 2004*, pages 3182–3186. IEEE, November 2004.
[9] J. Kuffner and S. LaValle. RRT-Connect algorithm with preliminary analysis. In *IEEE International Conference on Robotics and Automation*. IEEE, April 2000.
[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *InfoCom*. IEEE, April 2001.
[11] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Mobile Computing and Networking*. ACM, July 2001.
[12] S. Zhou, M-Y. Wu and W. Shu. Finding optimal placements for mobile sensors: Wireless sensor network topology adjustment. In *IEEE 6th CAS Symposium on Emerging Technologies*. IEEE, May 2004.
[13] S. Shakkottai, R. Srikant, and N. B. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. *CSL Technical Report*, June 2002.
[14] U.S. Geological Survey. Digital Elevation Model data. In *http://www.usgs.gov*.
[15] G. Wang, G. Cao, and T. L. Porta. A bidding protocol for deploying mobile sensors. In *11th IEEE International Conference on Network Protocols*. IEEE, November 2003.
[16] G. Wang, G. Cao, and T. L. Porta. Movement-assisted sensor deployment. In *IEEE InfoCom 2004*. IEEE, March 2004.
[17] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *InfoCom*. IEEE, April 2003.
[18] Y. Zou and K. Chakrabarty. Uncertainty-aware sensor deployment algorithms for surveillance applications. In *IEEE GlobeCom*. IEEE, December 2003.