1-1-2009

Self organization of sensor networks for energy-efficient border coverage

Mohamed K. Watfa
University of Wollongong in Dubai, mwatfa@uow.edu.au

Sesh Commuri
University of Oklahoma, Scommuri@ou.edu

Recommended Citation

Watfa, Mohamed K. and Commuri, Sesh: Self organization of sensor networks for energy-efficient border coverage 2009, 57-71.
https://ro.uow.edu.au/dubaipapers/22

Self Organization of Sensor Networks for Energy-Efficient Border Coverage

Mohamed K. Watfa and Sesh Commuri

Abstract: **Networking together hundreds or thousands of cheap sensor nodes allows users to accurately monitor a remote environment by intelligently combining the data from the individual nodes. As sensor nodes are typically battery operated, it is important to efficiently use the limited energy of the nodes to extend the lifetime of the wireless sensor network (WSN). One of the fundamental issues in WSNs is the coverage problem. In this paper, the border coverage problem in WSNs is rigorously analyzed. Most existing results related to the coverage problem in wireless sensor networks focused on planar networks; however, three dimensional (3D) modeling of the sensor network would reflect more accurately real-life situations. Unlike previous works in this area, we provide distributed algorithms that allow the selection and activation of an optimal border cover for both 2D and 3D regions of interest. We also provide self-healing algorithms as an optimization to our border coverage algorithms which allow the sensor network to adaptively reconfigure and repair itself in order to improve its own performance. Border coverage is crucial for optimizing sensor placement for intrusion detection and a number of other practical applications.**

Index Terms: **Ad-hoc and sensor networks, border coverage, energy savings, intrusion detection, optimal deployment, self organization, surveillance, three dimensional (3D) modeling.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been under development for many years and are about to gain widespread use as technology improves, prices drop, and new applications are developed. Smart disposable micro-sensor nodes can be deployed on the ground, in the air, under water, on bodies, in vehicles, and inside buildings. Sensor networks are playing an important role in bridging the gap between the physical world and the virtual information world [1], [2]. Unsupervised intrusion detection, which involves detecting and identifying the encroachment of a monitored region by an object, is one of the applications of wireless sensor networks. Algorithms for wireless sensor networks must have low communication overhead, rely as much as possible on local information, adapt to failures and changes in network conditions, and produce results in a timely fashion. Given the requirements to minimize the power, it is desirable to select the bare essential number of sensor nodes dedicated for the task while all other sensor nodes should preferably be in the hibernation or off state. Even though target tracking has been widely studied for sensor networks with large nodes and distributed tracking algorithms are available [3]–[5], intrusion detection in ad hoc networks with micro sensor nodes poses different challenges due to communication, processing and energy constraints. The impact of the number of nodes on the capacity of multi-hop wireless networks was analyzed for deployments in two dimensions (2Ds) [6] and 3Ds [7]. Under a protocol model of non-interference, if n nodes, each with a transmission rate of W bits/second, are randomly distributed in a disc of area A square meters (m^2), then the throughput obtained by each node for transmission to a randomly chosen sink is given by $\Theta(W/\sqrt{n \log n})$ [8]. Similarly, in the 3D deployment of wireless nodes, the throughput achieved when n nodes are located in a sphere of volume V is given by $\Theta(W/(n \log^2 n)^{1/3})$ [9]. Since the number of active nodes depends on the type of sensing required and the region of interest, the overall communication and energy efficiency of the WSN can be significantly improved by optimizing the number of nodes while guaranteeing the necessary quality of service.

Border surveillance is one of the major applications of sensor networks. The border represents the physical extent of the region to be monitored and depending on the application, it is required to sense the intrusion into the monitored region or exit from the monitored region of the object being monitored. In a typical deployment of sensor nodes, sensor nodes are distributed across the entire region of interest and it is necessary to determine a minimal set of sensor nodes that can adequately monitor the border. Thus, it is necessary to find a scalable and energy efficient solution to the border coverage problem. Such a solution would extend the scalability of wireless sensor networks and enable the monitoring of the largest international borders [10]. In [11], the authors developed theoretical foundations for laying barriers of wireless sensors. By their very nature, the deployments for barrier coverage are expected to be in very long thin belts (a region bounded by two parallel curves) as opposed to in regular structures such as squares, disks, cubes, or spheres. Further, since their only goal is to detect intruders before they have crossed the barrier as opposed to detecting them also after they have crossed a region by finding the boundary of coverage holes as well, the results from [11] can not be directly useful for us. Also, in [11], the authors did not provide any distributed algorithms to select the sensor nodes that lie on the border of a given region of interest after deployment. They derived critical conditions for weak barrier coverage, using which one can compute the minimum number of sensors needed to provide weak k-barrier coverage with high probability in a given belt region. Also, unlike our work in this paper, they did not analyze the 3D border coverage problems. Another related work is [12] where

Manuscript received February 16, 2007; approved for publication by Seong-Lyun Kim, Division III Editor, February 10, 2008.

This work was supported by the University Research Board (URB) at the American University of Beirut in supporting this work through grant # DDF-11135-888138.

M. K. Watfa is with the Department of Computer Science in the American University of Beirut, Beirut, Lebanon, email: mohamed.watfa11@gmail.com.

S. Commuri is with the School of Electrical and Computer Engineering at the University of Oklahoma, Norman, OK, USA, email: scommuri@ou.edu.

1229-2370/09/$10.00 © 2009 KICS

the authors consider eliminating redundant nodes to establish the boundary of the coverage region. Unlike our work in this paper, the authors did not consider the coverage of the boundary and their work is in 2D.

The full coverage problem, which verifies if every point in the region of interest is covered by at least one active sensor, has been studied in a variety of contexts. Our previous work [13]–[15] focused on the full coverage problem in 2D and 3D regions and provided algorithms to locate redundant sensor nodes in the region and deactivate them using simple geometric techniques. There, it was shown that eliminating coverage redundant sensor nodes increases the overall lifetime of the WSN. In [8], Gupta and Das design and analyze algorithms for self organization of a sensor network to reduce energy consumption. In particular, they develop the notion of a connected sensor cover and design a centralized approximation algorithm that constructs a topology involving a near optimal connected sensor cover. The works in [9] and [16] consider a large population of sensor nodes, deployed randomly for area monitoring. The goal is to achieve an energy-efficient design that maintains area coverage. As the number of sensor nodes deployed is greater than the optimum required to perform the monitoring task, the solution proposed is to divide the sensor nodes into disjoint sets, such that every set can individually perform the area monitoring tasks. Shakkottai *et al.* in [17] consider an unreliable sensor grid-network and derive necessary and sufficient conditions for the coverage of the region and connectivity of the network in terms of the transmission radius, sensing radius, and failure rate of the sensor nodes. In [18], Lieska *et al.* formulate coverage problems to address the quality of service (surveillance) provided by a sensor network. In particular, they address the problem of finding maximal paths of lowest and highest observabilities in a sensor network. The coverage concept with regard to the robot systems was introduced by Gage [19]. He defined three types of coverage: Blanket coverage, barrier coverage, and sweep coverage. In blanket coverage, the goal is to achieve a static arrangement of sensor nodes that maximizes the total detection area. In barrier coverage the goal is to achieve a static arrangement of nodes that minimizes the probability of undetected penetration through the barrier, whereas the sweep coverage is more or less equivalent to a moving barrier. One might wonder why we can not simply apply full coverage algorithms developed in our previous research work [13]–[15] and in many others [9], [16]–[18] since border coverage could be analyzed as applying full coverage algorithms on a line. That would result in unnecessary computations and also we propose further extensions to our algorithms to not only find the sensor nodes covering a border of a given region of interest but to also find the sensor nodes on the boundary of coverage and therefore the nodes on the boundary of the coverage holes.

In this paper, the problem of determining the minimum number of sensor nodes for covering the boundaries of a target region is addressed. Unlike the full coverage problem, here the primary interest is in the detection of movement of an object across the boundary. A recent work [20] considers the determination of holes in the coverage area of a sensor. In this work, the authors introduced a new technique for detecting holes in coverage by means of homology, an algebraic topological invariant. In [21], Carbunar *et al.* study the problem of detecting and eliminating redundancy in a sensor network with a view to improving energy efficiency, while preserving the network's coverage. They also examine the coverage boundary detection by reducing it to the computation of Voronoi diagrams. Most existing results related to the coverage problem in wireless sensor networks focused on planar networks [9], [16], [17]; however, 3D modeling of the sensor network would reflect more accurately real-life situations. Some applications of our results would be:

1) Disaster recovery: In the case of a natural disaster (flood, hurricane, and fire) may lead to the sensor nodes being on different planes and thus 3D coverage techniques need to be developed. Identifying the sensor nodes on the boundary of the holes help indicate the extent of the disaster. Sensor networks also provide the ability to gather reliable and accurate information from a range of sources, enabling early warnings and rapid coordination of responses to potential threats.
2) Topographical properties: Random dense sensor deployment on irregular terrains like mountains and hills leaves 3D coverage holes that indicate the topographical properties of the terrain. Understanding the topography of an area enables the understanding of watershed boundaries, drainage characteristics, water movement, impacts on water quality, and soil conservation.
3) Space exploration [22], [23]: Wireless sensor networks will play an important role in planetary explorations. A rover functioning as a base station collects measurements and relays aggregated results to an orbiter.
4) Undersea monitoring [24]–[26]: Sensor deployment underwater will enable real-time monitoring of selected ocean areas, remote configuration and interaction with onshore human operators. This can be obtained by connecting underwater instruments by means of wireless links based on acoustic communication. Under water acoustic sensor networks (UW-ASN) consist of a variable number of sensor nodes and vehicles that are deployed to perform collaborative monitoring tasks over a given area.

We refer to [27] and [28] for an extensive list of possible applications of both 2D and 3D sensor networks. As for practical ways of deployment of a 3D sensor network, we refer to [23] where a lot of deployment strategies are proposed for 3D sensor networks (e.g., the Mars mission). Unlike any other related works in this field, we first provide optimal 2D and 3D techniques for the deployment of a sensor network for border coverage of a given region. The auxiliary problem of selecting the minimal subset of previously deployed active sensor nodes for border coverage is then addressed. The energy efficiency of a WSN is studied in the context of the deployment of sensor nodes and the border coverage obtained. The minimum number of sensor nodes required for border coverage is used to specify a "measure of optimality" that can serve as a metric for the energy efficiency of a WSN. The border coverage algorithm developed is used to determine the savings that can be obtained by deactivating a subset of nodes while still maintaining the coverage of the region. The proposed techniques are analyzed mathematically and the algorithms are demonstrated through numerical examples.

The rest of the paper is organized as follows. The border coverage problem is formulated in Section II. A deterministic sensor deployment to guarantee border coverage of a region is provided in Section III. In Section IV, distributed algorithms for selecting a border cover regions are studied. In Section V, an optimization of our border coverage algorithm is introduced where the border cover heals itself in order to improve its performance. Numerical simulation results that validate the proposed algorithms are presented in Section VI. Possible extensions to our developed algorithms and conclusions are summarized in Section VII.

II. PROBLEM FORMULATION

An emerging application area for sensor networks is intelligent surveillance and intrusion detection. Sensor nodes are randomly distributed in an area to be monitored. The ultimate goal is to detect an intruder target and alert the sensor nodes which are close to the predicted path of the target. However, minimizing the power consumed should be the most important design goal. The lifetime of the sensor network can be significantly extended by optimizing the energy consumption of each sensor node. In this section, the notions of sensing region and border coverage are first defined. The border coverage and optimization problems are then formulated.

Let O_i be the output of a sensor S_i that is capable of sensing a phenomenon P. Let the sensing radius of the sensor node S_i be R_i. It is assumed that each sensor node is aware of its own location, the location of the boundaries of the region to be monitored and the location of its neighbors. This assumption is not too stringent and it can be satisfied by communications between adjacent nodes in the network on startup.

Definition II.1: The phenomenon P located at $y \in R^3$ is said to be **detected** by sensor S_i located at $X_i \in R^3$ if and only if there exists a constant threshold δ such that

$$O_i(y) \begin{cases} \geq \delta & \text{if the phenomenon } P \text{ is present,} \\ = \epsilon & \text{otherwise.} \end{cases}$$

The quantity 'δ' in the definition above is the signal threshold and is specific to the type of sensor used. The sensing region of sensor S_i located at $X_i(x_i, y_i, z_i)$ is the collection of all points where the phenomenon P can be detected by the sensor S_i, i.e., $A_i = \{y \in R^3 | P \text{ is detectable by } S_i\}$. In this paper, without loss of generality, we restrict the **sensing region** of S_i to be a closed ball centered at $X_i \in R^3$. We also assume that all the deployed sensor nodes have equal sensing radii R_3. As noted in Section VII, our developed algorithms can be modified to handle the case where the sensor nodes have unequal sensing radii.

Definition II.2: Let $Y = \{y \in R^3 | O_i(y) > \delta\}$. The **sensing region** of sensor S_i located at $X_i \in R^3$ is defined as $A_i = \{y \in Y | \parallel y - X_i \parallel \leq R_3\}$, where $\parallel \cdot \parallel$ is the Euclidean distance between y and X_i.

In the case of 2D, the sensing region is assumed to be a disk of radius R_3. The sensing boundary (circle) of sensor S_i, in this case, is denoted by Cir_i. We assume that any two nodes S_i and S_j can directly communicate with each other if their Euclidean distance is less that the communication range R_c. Although a network can be rendered useless if it loses its connectivity, we characterize the system lifetime by just observing the resulting border coverage. Zhang and Hou [29] showed that if the communication range is at least twice the sensing range, then complete coverage of a convex area implies connectivity among the nodes. Assuming the communication range is twice the sensing range ($R_c \geq 2R_s$), the theorems in [29] could be easily extended to handle border coverage of the region as well.

Definition II.3: An **intruder** is any object that is subject to detection by the sensor network as it crosses the border.

A reasonable assumption is made that no intruder is aware of the location of the deployed sensor nodes. The following gives precise definitions to what a border of a region is.

Definition II.4: Let **R** be a subset of the (2D or 3D) space. The point 'p' is said to be **near R** if every neighborhood of 'p' contains a point from **R**, i.e.,

$$\forall \epsilon > 0, \exists x \in \text{Ball(p}, \epsilon) \text{ and } \text{x} \in \mathbf{R}.$$

In the definition above, $\text{Ball(p}, \epsilon)$ means the set of all points whose Euclidean distance from p is less than ϵ.

Definition II.5: The set of all points in **R** and near **R** is called the **closure** of **R** and is denoted by $cl(\mathbf{R})$, i.e.,

$$cl(\mathbf{R}) = (\mathbf{R}) \cup \{\text{All points near } \mathbf{R}\}.$$

Definition II.6: The **border** of a region **R** denoted by $B(\mathbf{R})$ is defined as the set of all points that are common to **R** and its complement, i.e., $B(\mathbf{R}) = cl(\mathbf{R}) \cap cl(\overline{\mathbf{R}})$ where $\overline{\mathbf{R}}$ is the complement of the region **R**, i.e., all the points that do not belong to **R**.

According to Definitions II.3–II.6, a region is said to be border covered if and only if an intruder is always detected as it crosses the border of the region. A sensor is called a border sensor if its sensing region intersects the border of the region of interest.

Definition II.7: A set of sensor nodes C_{Border} is said to be a **border cover** of a region **R** if every point on the border of **R** belongs to the sensing region of at least one sensor in C_{Border}, i.e., $\forall p \in B(\mathbf{R}), p \in S_i$ for some $S_i \in C_{\text{Border}}$.

Definition II.8: A set of sensor nodes $C_{\text{Border,Reduced}}$ is said to be a **reduced border cover** of a region **R** if $\forall p \in B(\mathbf{R})$, $p \in S_i$ for some $S_i \in C_{\text{Border,Reduced}}$ and no proper subset of $C_{\text{Border,Reduced}}$ is a border cover of **R**, i.e., $C_{\text{Border,Reduced}} - S_l$, for any $S_l \in C_{\text{Border,Reduced}}$, is not a border cover of **R**.

Definition II.9: A sensor node is called a **redundant sensor node** if its sensing region is completely covered by its neighboring sensor nodes. Deactivating a redundant sensor does not affect the overall full coverage of the region of interest.

Definition II.10: A sensor node is called a **redundant border sensor** node if the portion of the border covered by it is completely covered by its neighboring sensor node.

Definition II.11: A sensor node is called a **non-border sensor** node if its sensing region does not intersect the boundary of the region of interest.

From Definitions II.10 and II.11, it can be seen that the deactivation of a border redundant sensor or a non-border sensor does not affect the overall border coverage of the region of interest (Fig. 1(b)). Using Definitions II.1–II.11, the border coverage problem is analyzed in this paper by dividing it into the following two sub problems:

1) Optimal deployment for border coverage: Find the minimum number of sensor nodes and their placements for border coverage of a given region 2D/3D **R**.

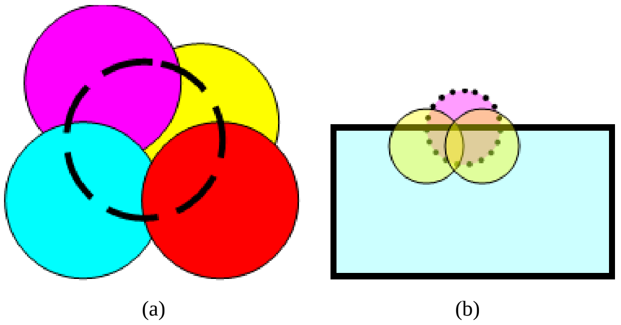

(a) (b)

Fig. 1. Example of (a) a redundant sensor (dashed circle) and (b) a border redundant sensor (dashed circle).

2) Optimal selection for border coverage: Given a dense deployment of sensor nodes in a region **R**, find a minimum subset of active nodes that guarantee border coverage of **R**.

The discussion in the following sections assumes that the region to be monitored is large in comparison to the sensing region of an individual sensor node and that the location of all sensor nodes is known. All through the paper we use the following notations:

- **R**: Region of interest,
- S: Set of sensors in the region,
- R_S: Sensing radius of each sensor,
- A_i: Sensing region of sensor S_i,
- Cir$_i$: Boundary of the 2D sensing region of sensor S_i,
- C_{full}: Set of sensors fully covering the region,
- C_{Border}: Set of sensors border covering the region.

III. OPTIMAL DEPLOYMENT STRATEGY FOR BORDER COVERAGE

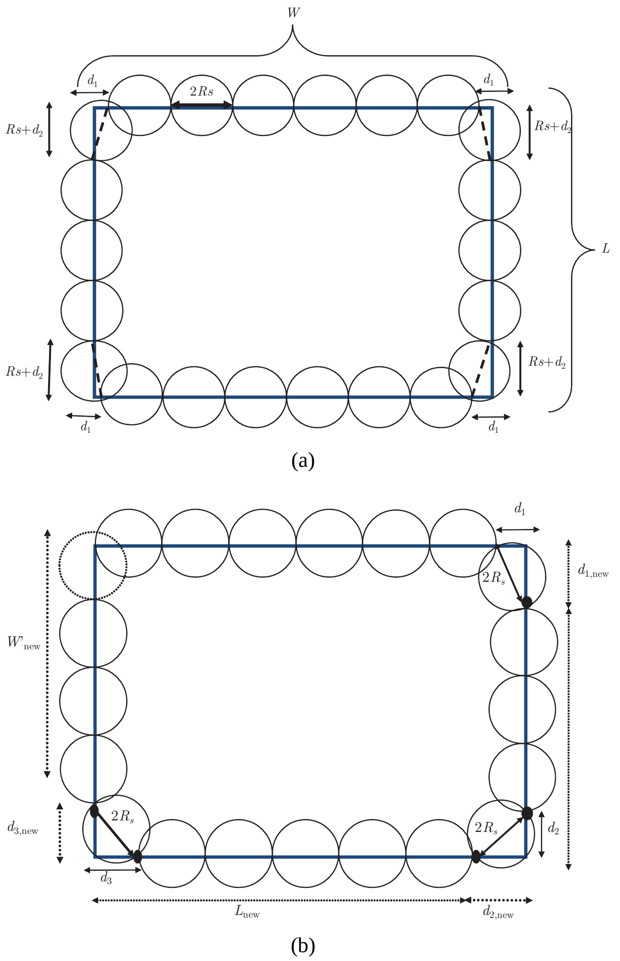
A critical issue in WSNs is the deployment and organization of the sensor network. Although many scenarios assume random deployment, such a deployment is not optimum and therefore a lot of energy is wasted due to multiple active nodes in a given region. When flexibility in deployment exists, it is advantageous to find an optimum border deployment of the sensor nodes so that border coverage can be achieved using a minimum number of nodes. In this section, theorems for optimal deployment of the sensor nodes are developed. These theorems provide lower bounds on the number of nodes needed to border cover both 2D and 3D regions of interest.

A. Optimal 2D Deployment for Border Coverage

In the 2D deployment problem, the minimum number of sensor nodes modeled as disks and their locations for border coverage of a given rectangular region **R** are to be determined. While the region to be border covered is assumed to be a rectangular region, the algorithms could be extended to border cover any arbitrary shape of a region with minor modifications.

Theorem III-A.1: Consider a rectangular region **R** of length 'L' and width 'W.' The lower bound on the number of sensor nodes needed to achieve border coverage of **R** is:

$$\left[\frac{L}{2R_S}\right]+\left[\frac{W_{\text{new}}}{2R_S}\right]+\left[\frac{L_{\text{new}}}{2R_S}\right]+\left[\frac{W'_{\text{new}}}{2R_S}\right]$$



(a)

(b)

Fig. 2. (a) The optimal deployment of sensor nodes modeled as disks in 2D to border cover a rectangular region. The special case illustration of the best possible way to minimize the number of sensor nodes covering the border is shown and (b) the deployment strategy is explained and it is shown how to take advantage of a row sensor covering part of the column resulting in extra savings.

where

$$W_{\text{new}} = W - \sqrt{4R_S^2 - L \mod 2R_S},$$
$$L_{\text{new}} = L - \sqrt{4R_S^2 - W_{\text{new}} \mod 2R_S},$$
$$W'_{\text{new}} = W - \sqrt{4R_S^2 - L_{\text{new}} \mod 2R_S}.$$

Proof: The optimal way to deploy the sensor nodes to achieve border coverage of the region is to deploy the sensor nodes across the perimeter of the entire region such that any 2 adjacent sensor nodes that are on the same row or column are tangent to each other. $[L/2R_S]$ is the least number of sensor nodes to cover a line of length 'L.' For a rectangular region of length 'L' and width 'W,' the perimeter can be optimally covered by $2([L/2R_S]+[W/2R_S])$ sensor nodes. However such a cover will have overlapping sensing coverage at the vertices of the rectangle. The number of sensor nodes does not exactly cover each edge then the last sensor would partly cover the adjacent edge, so a better way would be to select the next position of the center such that its sensing circle intersects the last circle in its boundary intersection. □

Special case (Fig. 2(a)): Let $2d_1$ and $2d_2$ be the remainders of L divided by $2R_s$ and W divided by $2R_s$, respectively, i.e., $2d_1 = L \pmod{2R_S}$ and $2d_2 = W \pmod{2R_S}$. The optimal way of deployment will be achieved if the corner sensor node can completely cover the leftover d_1 on the length of the rectangle and $R_S + d_2$ on the width of the rectangle on each corresponding corner. By doing so, we would cover using one sensor node more than its diameter based on the third side triangle rule which states that the sum of two side of a triangle should be greater than the third side, i.e., $d_1 + (R_S + d_2) > 2R_S$. So, the optimal deployment is established if one of the following two conditions holds:

$$\text{Condition (1): } d_1^2 + (R_S + d_2)^2 = 4R_S^2,$$
$$\text{Condition (2): } d_2^2 + (R_S + d_1)^2 = 4R_S^2.$$

In this case, we only need $2([L/2R_S] + [W/2R_S] + 1)$ nodes to cover the border of the rectangle.

General case (Fig. 2(b)): If we do not have one of the two conditions stated above, then we use the following deployment strategy based on the above special case observation. We first cover L by $[L/2R_S]$ nodes packing them side by side. We then construct a right triangle on the corner whose hypotenuse is of length $2R_S$ and its sides are $2d_1 = L \pmod{2R_S}$ and $d_{1,\text{new}} = \sqrt{rR_S^2 - d_1}$. We deploy a sensor node on that corner covering d_1 from L and $d_{1,\text{new}}$ from W. We now take $W_{\text{new}} = W - d_{1,\text{new}}$ and perform the same deployment procedure. We will guarantee maximum border coverage on all the corners until we get to the last corner where we need one extra sensor node, i.e., the total number of sensor nodes needed to border cover a rectangular region is:

$$\left[\frac{L}{2R_S}\right] + \left[\frac{W_{\text{new}}}{2R_S}\right] + \left[\frac{L_{\text{new}}}{2R_S}\right] + \left[\frac{W'_{\text{new}}}{2R_S}\right]$$

where

$$W_{\text{new}} = W - \sqrt{4R_S^2 - L \bmod 2R_S},$$
$$L_{\text{new}} = L - \sqrt{4R_S^2 - W_{\text{new}} \bmod 2R_S},$$
$$W'_{\text{new}} = W - \sqrt{4R_S^2 - L_{\text{new}} \bmod 2R_S}.$$

If we assumed that the region to be monitored is large in comparison to the sensing region of an individual sensor node, then a very simple approximation on the number of nodes to cover a rectangular region would simply be $2(\lceil L/2R_S \rceil + \lceil W/2R_S \rceil)$.

B. Optimal 3D Deployment for Border Coverage

The 3D optimal sensor deployment for border coverage is far more complex that the 2D case. It is addressed by determining the minimum number of sensor nodes required to cover the surface of a cubical region of interest. Since the coverage region of a sensor node is modeled as a closed ball, the border coverage problem requires the determination of all the points on the surface of the cube that are covered by the sensor nodes. To address this issue, the intersection of the sensing regions and a boundary plane is first defined. This definition will then be used to determine the least number of sensor nodes required for border coverage.

Definition III-B.1: A **great circle** on a sphere is the intersection of that sphere with a plane passing through the center of the sphere.

Definition III-B.2: The thickness 'θ' of a sensor cover is defined as the average number of sensor nodes that cover a point in the space. Let θ be a volume of one sensing region per volume of the fundamental region, i.e., $((\sum_{i=1}^{n} V_i/n)/(V_{\text{total}}/n)) = nV/V_{\text{total}}$, where '$n$' is the number of active sensor nodes; V_i is the volume of sensing region of sensor S_i; V_{total} is the total volume of the sensed region R_i.

Definition III-B.3: The covering radius R_{cover} of spheres centered at $X_1, X_2, \cdots, X_n$ is the minimum sensing radius that will cover the region **R**.

Lemma III-B.1: The centers of all the optimal deployed spheres to guarantee border coverage of a cube must lie on a face of the cube.

Proof: It is clear that each sensor covers a maximum area when the coverage region lies in the plane passing through the center of the sphere representing the sensing region. Thus, minimizing the number of sensor required to cover the surfaces of the region to be monitored is equivalent to maximizing the coverage area of each sensor. This is possible only when the centers of all the sensor nodes lie on the surface of the region to be monitored. □

Theorem III-B.1: The optimal deployment locations of sensor nodes to border cover a 3D cubical region is the locations of the spheres whose centers form a lattice of spacing $\Lambda = 1.7322R_S$ on each face of the cube.

Proof: In a 2D space, the optimal covering of a region with circles is obtained when the centers of the circles lie at the vertices of a hexagonal lattice. If the distance between adjacent vertices in this case is one unit, then the entire region can be covered by copies of a disc whose covering radius is $R_{\text{cover}} = 1/\sqrt{3} = 0.5773$ (minimum sensing radius that completely covers a region). Such a lattice is also periodic and completely reduced. Moreover, the thickness of the cover is $\theta = 2\pi/3\sqrt{3} = 1.2092$ (average number of sensor nodes that cover a point in the region). Thus, the deployment is optimal if the spacing between the centers of adjacent discs equal $\Lambda = R_S/0.5773$. So, the optimal deployment to cover the border of a 3D cubical region is by placing the sphere at centers of the lattice on each face of the cube with spacing $\Lambda = 1.7322R_S$. □

Theorem III-B.2: Consider a cubical region **R** of side 'a' ('a' is sufficiently large in comparison to R_S). An approximation on the lower bound on the number of sensor nodes of sensing radius R_S to achieve border coverage of **R** is:

$$N_{\min} = \frac{2\sqrt{3}a}{3R_S^2}\left(2a - \left(1 - \frac{\sqrt{3}}{2}\right)R_S\right).$$

Proof: Our approach is based on the problem of covering a rectangle by circles which has been studied by Kershner [30] and Verblunsky [31] where the least number of circles of unit radius which can cover a rectangle was determined. However, the placement pattern of the circles to fully cover the rectangle was not identified. In [40], the authors suggested a placement

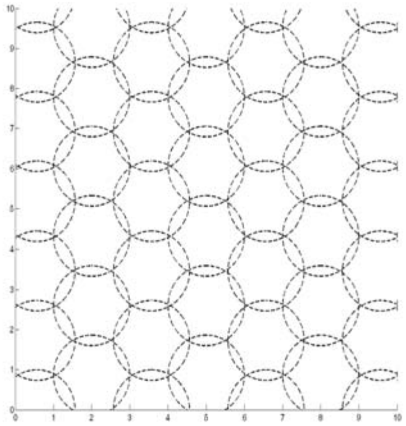


Fig. 3. Covering a rectangle using minimum number of circles.

pattern to cover a rectangle with overlapping circles and they proved that their solution is optimal when compared to Keshner's results [30]. □

Definition III-B.4: A pattern of R_s-*strip* is composed of a string of R_s disks placed along a vertical line such that the distance between the centers of any two adjacent nodes is $\sqrt{3}R_S$.

The deployment strategy to fully cover the rectangle is achieved by placing m columns of R_s-*strips* oriented parallel to the y-axis with the distance between the centers of any two adjacent R_s-*strips* is $1.5R_s$ as shown in Fig. 3.

In a global Cartesian coordinate where the origin is at the left bottom of the rectangle, we place mR_s-*strips* parallel to the y-axis with n disks in each strip to completely cover the rectangle. The center of the kth row $(1 \leq k \leq n)$ and lth column $1 \leq l \leq m$ disk is at $[x_c^k, y_c^k]$:

$$[x_c^k, y_c^k] = \begin{cases} 0.5 + (l-1)\dfrac{3}{2}R_S, (k-1)\sqrt{3}R_S, \\ \qquad\qquad \text{if } l \text{ is an odd integer} \\ 0.5 + (l-1)\dfrac{3}{2}R_S, \dfrac{\sqrt{3}}{2}R_S + (k-1)\sqrt{3}R_S, \\ \qquad\qquad \text{if } l \text{ is an even integer.} \end{cases}$$

The minimum number of sensor nodes (of sensing radius R_s) needed to cover a rectangular region **R** of length 'L' and width 'W' as was derived in [40] (and proven to be optimal) is $N = (2\sqrt{3}LW)/(9R_S^2)$. Our goal however is to border cover a 3D cube using sensor nodes that are modeled as 3D balls. This problem can be defined as fully covering each face of the cube with circles. However, some spheres might interest more than one face of the cube resulting in more border coverage. Let the first row of the cube in the $x-y$ plane cut the $x-z$ plane in disks of equations:

$$\begin{cases} (x - (1/2 + (l-1)\dfrac{3}{2}R_S))^2 + y^2 = R_S^2, \qquad \text{if } l \text{ is odd,} \\ (x - (1/2 + (l-1)\dfrac{3}{2}R_S))^2 + (y - \dfrac{\sqrt{3}}{2}R_S)^2 = R_S^2, \\ \qquad\qquad \text{if } l \text{ is even.} \end{cases}$$

For the square in the $x-z$ plane, we start covering that face from a line above the x-axis at $d = (1 - (\sqrt{3}/2))R_S$. We get a rectangle of sides a and $a_1 = a - d$ and perform the covering pattern mention earlier to completely cover one face of the cube. Doing the same for all the other faces of the cube, we therefore need to cover 3 squares of side a each and 3 rectangles of sides a and $a_1 = a - d$ each. So the total number of sensor nodes needed to cover a cubical region of side a is:

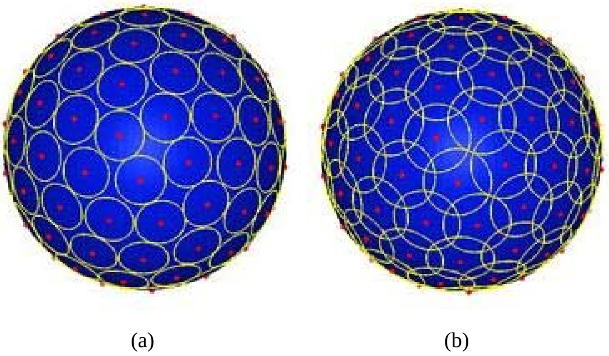

(a) (b)

Fig. 4. The densest packing of the great circles on the surface of (a) a spherical region and (b) a circle covering of the surface of the sphere.

$$N = \frac{2\sqrt{3}a}{3R_S^2}(a + a_1) = \frac{2\sqrt{3}a}{3R_S^2}\left(2a - \left(1 - \frac{\sqrt{3}}{2}\right)R_S\right).$$

If the region of interest was considered to be a spherical region (instead of a cubical region) then all the centers of the optimal deployed spheres to border cover the spherical region lie on the surface of the sphere and the optimal deployment locations of sensors modeled as spheres to border cover a 3D spherical region **R** is the locations of the centers of circles which form a dense covering on the surface of a sphere **R** (Fig. 4).

Theorem III-B.3: The lower bound on the number of sensor nodes to border cover a spherical region **R** of radius $\Re$ is $n = [1.76\Re/(1 - \cos R_S)]$ where the sensing radius R_S is in [rad].

Proof: Since we are concerned with the lower bound of sensor nodes needed to cover the boundary of the sphere, we will treat this problem as a packing problem. The minimum number of cycles needed to pack the boundary of a sphere will be derived. The area of a circle is calculated as the area of a spherical cap. On the unit sphere, the area S of a circle having the radius r [rad] is: $S = 2\pi \int_0^r \sin r_1 dr_1 = 2\pi(1 - \cos r)$. Therefore, if there are non-overlapping n circles having a radius R_S on the unit sphere, the packing density D is: $D = nS/4\pi\Re = n/2\Re(1 - \cos R_S)$. Since our goal is to minimize the number of sensors needed to border cover the spherical region, that is equivalent to finding the densest packing of the great circles covering the surface of the spherical region (although a tighter bound can be achieved). Many approaches to find the circle configuration and resulting packing density were performed to the circle packing problems. A near optimal packing density was achieved (0.88) [34], [36], [37]. Knowing the optimal packing density, we can find the minimum number of sensor nodes needed to border cover the spherical region, $n = 2D/(1 - \cos R_S)$. Since we are concerned with the lower bound, assuming $D = 0.88$ then $n = \lceil 1.76\Re/(1 - \cos R_S) \rceil$. □

In the next section, given a random deployment of sensor nodes in a region of interest, optimal selection strategies are designed to self organize the network for border coverage.

IV. DISTRIBUTED ALGORITHMS FOR BORDER COVER SELECTION

The results in Section III enable the optimum placement of sensor nodes for border coverage of a given region (2D and 3D regions). In practice, however, given an existing distribution of sensor nodes, it is often necessary to minimize the number of nodes that remain active while still achieving border coverage of the entire region. In this section, an algorithm is developed where the nodes make local decisions on whether to sleep or join the set of active nodes. The 2D and 3D cases for selecting an optimum border cover of a given region are studied. A measure of optimality is also proposed to compare the performance of the border coverage of a given sensor network with the optimum coverage obtained in Section III. The border coverage algorithm presented in this section has the following key features:

1) It is a decentralized algorithm that depends only on the local states of the sensing neighbors.
2) It provides guaranteed degrees of border coverage.
3) It handles the case where the nodes have unequal sensing radii.
4) Is is computationally simple resulting in minimum energy usage.

A. A 2D Distributed Border Cover Selection Algorithm

In order to solve the border coverage problem for a 2D region of interest, it is assumed that the region to be monitored is a rectangle specified by its vertices V_1, V_2, V_3, and V_4. It is also assumed that all the sensor nodes are aware of the location of the vertices, i.e., the sensor nodes are aware of the extent of coverage that is required. The border coverage algorithm can be applied to any shape of boundary but the region of interest is assumed to be a rectangular region for the sake of ease of presentation. The algorithm depends on the fact that individual sensor nodes can verify if they have overlapping border coverage with their neighbors. If the border covered by a sensor node is covered by other sensor nodes in the neighborhood, then deactivating this sensor does not affect the overall border coverage. In this section, we first derive conditions that indicate overlapping border cover for a given sensor. We start by giving some definitions and assumptions that will aid us in developing an algorithm to select a border cover. Let $B(\mathbf{R})$ represent the boundary of the region $\mathbf{R}$ to be covered. Then, $B(\mathbf{R})$ can be represented as

$$B(\mathbf{R}) = \bigcup_{i,j=1}^{4} B_{i,j}$$

where $B_{i,j}$ is the segment connecting vertices V_i and V_j. Without loss of generality, suppose the boundary edges are ordered as as shown in Fig. 5.

Definition IV-A.1: An intersection segment is the portion of the boundary covered by the sensing region of a sensor node and is represented by the closed interval $[x, y]$ such that: $[x, y]$ is an intersection segment $\Longrightarrow \forall z \in [x, y]$, $\exists i \in 1, \cdots, n$ such that $z \in A_i$ and $x, y \in \text{Cir}_i$.

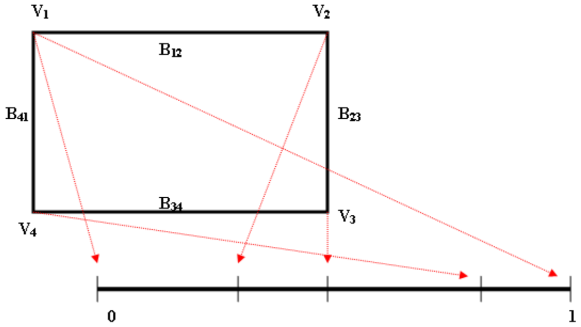


Fig. 5. The ordering based on the mapping function.

A segment $\text{Seg}_i = [x_i, y_i]$ is represented by its start point x_i and end point y_i. Since our algorithms depend on the concept of ordering, we define a mapping: $\varphi : B \longrightarrow [0, 1]$ based on the distance metric from the nearest origin, i.e., $\forall x \in B_{ij}, \varphi(x) = \{d(x, V_i) + d(V_1, V_i)\}/|P|$ where $|P|$ is the total length of the perimeter of the rectangular boundary and $d(V_i, V_j)$ is the distance along the boundary of the region, i.e., for example in Fig. 4, $d(V_1, V_3) = |B_{12}| + |B_{23}|$. A special case should be taken for the sensing region covering the origin vertex v_1, where the resulting intersection segment is divided into 2 sub-segments each of which is mapped separately.

Definition IV-A.2: We call $\text{Seg}_j = [x_j, y_j]$ the **successor** of $\text{Seg}_i = [x_i, y_i]$ denoted by $\text{Seg}_i \succ \text{Seg}_j$ if the following conditions are satisfied:

▷ $\text{Seg}_j \cap \text{Seg}_i = \text{Seg}_{ji} \neq \emptyset$,
▷ there is no other starting point in Seg_{ji},
▷ $\forall p \in \text{Seg}_{ji}, p \neq x_k$ for some $k \neq i, j$.

Theorem IV-A.1: Consider the set of segments: $\mathbf{S} = \{\text{Seg}_1, \text{Seg}_2, \cdots, \text{Seg}_m\}$ where $\text{Seg}_i = [x_i, y_i] \subset [0, 1]$. Assume that no two segments are contained in each other, i.e., $\forall \text{Seg}_i, \text{Seg}_j \in S, \text{Seg}_i \not\subset \text{Seg}_j, i \neq j$. A segment $\text{Seg}(= [x, y])$ is covered by $\bigcup_{i=1}^{m} \text{Seg}_i$ if and only if the following hold:

(a) There exist integers $1 \leq a, b, \cdots, k \leq m$ such that $x \in \text{Seg}_a$ and $y \in \text{Seg}_b$,
(b) $\text{Seg}_a \succ \text{Seg}_b \succ \cdots \succ \text{Seg}_k$.

Proof: $\text{Seg}_a \succ \text{Seg}_b \Longrightarrow \text{Seg}_a \cup \text{Seg}_b = [x_a, y_b]$. Therefore, $\bigcup_{i=a}^{k} \text{Seg}_i = [x_a, y_b]$. Further from (a), $x_a < x$, and $y_k > y$. $[x, y] \subset [x_a, y_k] = \bigcup_{i=a}^{k} \text{Seg}_i \subset \bigcup_{i=1}^{m} \text{Seg}_i$. On the other hand, suppose that the segment $[x, y]$ is covered by the segments $\text{Seg}_1, \cdots, \text{Seg}_m$. Since the segment is covered, there exists some segment Seg_a such that $x \in \text{Seg}_a$. Similarly, there exists at least one segment 'k' such that $y \in [x_k, y_k]$. Thus, condition (a) is easily satisfied. Now, if $y_a > y$, then $[x, y] \subset \text{Seg}_a$ and condition (b) is trivially satisfied. Otherwise, there exists a segment Seg_b such that $\text{Seg}_a \succ \text{Seg}_b$. If this was false, then it means that $y_a \geq y_k$ for $1 \leq k \leq m$. This would then imply that there exist points in the interval $(y_a, y]$ that are not covered, thereby contradicting the assumption that the segment $[x, y]$ is covered. If $y_b > y$, then condition (b) is proved. Otherwise, repeating the process, we obtain integers $a, \cdots, k$ such that $\text{Seg}_a \succ \text{Seg}_b \succ \cdots \succ \text{Seg}_k$ and $y_k > y$. □

Therefore, conditions (a) and (b) together imply that the segment $[x, y]$ is covered by the collection of segments $\mathrm{Seg}_1, \mathrm{Seg}_2, \cdots, \mathrm{Seg}_m$.

Theorem IV-A.1 indicates that a sensor node is completely border redundant if each segment in the partitioning of its intersection segment by its neighbors' intersection segments has a successor and the end points are also covered. Therefore, to check if a sensor S_0 is a border redundant sensor and therefore could be deactivated without affecting the overall border coverage, one has to first find all the adjacent sensor nodes that lie on the border of the region of interest. For each sensor, find the resulting intersection segment (or segments) with the boundary lines and check if S_0's portion of border coverage is completely covered by its neighboring sensor nodes. That can be done using Theorem IV-A.1. An algorithm is presented that illustrates the steps in this process.

2D distributed border coverage algorithm

```
For each node $S_i$, form the set of neighbors, $N(i)$. Do the following:
Step 1: Find the intersection segments $\mathrm{Seg}_i$.
    Find '$\mathrm{Seg}_i$' the intersection segment of $S_i$ with the boundary of the region of interest and map it to [0, 1].
    If $\mathrm{Seg}_i = \emptyset$ or a point then $S_i$ is declared as a non border node.
    Else go to step 2.
Step 2: Non containment property.
    Let $\overline{\mathrm{Seg}}_i$ be the set of segments covering $\mathrm{Seg}_i$ and is initially set to $\emptyset$.
    For every pair of nodes $S_j, S_k$ in $N(i)$
        ▷ Find the common intersection segments $\mathrm{Seg}_j = [x_j, y_j]$ and $\mathrm{Seg}_k = [x_k, y_k]$, respectively.
        ▷ If the end points appear in increasing order as $x_j, x_k, y_k, y_j$, i.e., $\mathrm{Seg}_k \subseteq \mathrm{Seg}_j$ and can be ignored.
        ▷ Update $\overline{\mathrm{Seg}}_i$ to include $\mathrm{Seg}_j$, i.e., $\overline{\mathrm{Seg}}_i = \overline{\mathrm{Seg}}_i \cup \{\mathrm{Seg}_j\}$.
Step 3: Check for endpoints coverage.
    Check that, $\exists \mathrm{Seg}_f = (x_f, y_f)$ and $\mathrm{Seg}_l = (x_l, y_l)$ in $\overline{\mathrm{Seg}_i}$ for $x_f \leq x_i \leq y_f$ and $x_l \leq y_i \leq y_l$.
    If true go to step 4.
Step 4: Check for successor.
    Check that, for each element $\mathrm{Seg}_m = (x_m, y_m)$ in $\overline{\mathrm{Seg}}_i - \mathrm{Seg}_l$, $\exists \mathrm{Seg}_n = (x_n, y_n) | \mathrm{Seg}_m \succ \mathrm{Seg}_n$ and $m \neq n$.
```

If this condition is satisfied, the boundary intersecting segment Seg_i of the given sensor is completely covered and S_i is declared as a border redundant sensor and can be deactivated without affecting the overall border coverage. The algorithm guarantees that every point on the boundary of the target region is covered by at least one sensor. The optimal set of sensor nodes is also selected. The computational complexity of the redundancy selection algorithm developed in this section depends on $N = (\max_{i=1}^{n} |N(i)|)$, the maximum number of nodes in the neighbor set of any sensor in the network and n, the total number of sensor nodes in the network. The computational complexity of the border redundancy checking algorithm is $O(N^2)$. Since we have 'n' sensor nodes to be checked, then the complexity is $O(nN^2)$. For large networks, the number of neighbors of any sensor is small compared to the size the network ($N \ll n$) so the computational complexity of the algorithm for such large networks is of order 'n' ($O(n)$) where n is the total number of sensor nodes in the network.

B. A 3D Distributed Border Cover Selection Algorithm

The 3D optimal sensor border coverage problem is far more complex than the 2D case. We approach it from a different angle and try to transform it to optimal complete coverage of the sensor nodes in a 2D plane. We start by proving some theorems and then provide a 3D distributed algorithm.

Lemma IV-B.1: The problem of 3D border coverage of a cube by sensor nodes modeled as 3D balls is equivalent to the problem of complete coverage of a 2D plane by sensor nodes modeled as circles.

Proof: According to the definition of border coverage, each point on the border should be covered by at least one sensor. The border $B(\mathbf{R})$ of the cubical region $\mathbf{R}$ is represented by 6 faces (2D planes). First, if each face of the cube $B_a \in B(\mathbf{R})$, $a = 1, \cdots, 6$ is completely covered by a set of circles $\mathrm{Cir}_a = \{\mathrm{Cir}_{a1}, \cdots, \mathrm{Cir}_{an}\}$ and if D_i is the disc bounded by the circle Cir_i then $\forall p \in B_a, p \in D_{ai}$ for some $\mathrm{Cir}_{ai} \in \mathrm{Cir}_a$. The 3D border coverage is now transformed to finding the spheres whose border intersections are these circles $\forall p \in B(\mathbf{R}),\ p \in A_i$, for some $\mathrm{Cir}_i = A_i \cap B(\mathbf{R})$. Now, if we have a set of sensor nodes that border cover a 3D cubical region, taking the intersection of the spherical sensing regions of the sensor nodes with each face of the cube will result in the formation of circles which completely cover the 2D plane. So, the 3D border coverage problem is transformed to the 2D full coverage problem. □

Let the intersection of any boundary plane B and sphere S_i be circle Cir_i, i.e., $\mathrm{Cir}_i = B \cap A_i$. The interior of the circle Cir_i is said to be the disc bounded by the circle Cir_i, i.e., $D_i = \textit{interior}(\mathrm{Cir}_i)$.

Definition IV-B.1: A circle Cir_i is completely covered if the disc bounded by the circle is completely covered, i.e.,

$$\forall p \in D_i, p \in \bigcup_{j=1}^{n} A_j.$$

Definition IV-B.2: A sensor S_i is a **border-redundant** sensor if $\mathrm{Cir}_i = B \cap A_i$ is completely covered by neighboring spheres.

In [33], the authors proved that a convex region A is Ks-covered by a set of sensors S if 1) there exist in region A intersection points between sensors or between sensors and A's boundary; 2) all intersection points between any sensors are at least Ks-covered; and 3) all intersections points between any sensor and A's boundary are at least Ks covered. In this paper, we are interested in border redundant sensor nodes rather that redundant sensor nodes so their theorems can not be used. Next, we provide a theorem that aids us to determine which sensor nodes are border redundant.

Theorem IV-B.1: A sensor S_0 is **border-redundant** if all the intersection points $\mathrm{Cir}_i \cap \mathrm{Cir}_j \in D_0,\ \forall i, j = 1, \cdots, n$ are covered by one or more adjacent spheres.

Proof: Consider an uncovered point 'p' in D_0. Since some parts of D_0 are covered by adjacent sensor nodes, these spheres are going to partition D_0 into regions bounded by arcs from the boundary of Cir_0 and/or arcs from circles $\mathrm{Cir}'_k s,\ k = 1, \cdots, n$.

Suppose 'p' belongs to a region R_x in D_0. Since 'p' is not covered, it is easy to see that R_x has to be bounded only by the exterior arcs of the circles. Also, the entire boundary of R_x, including the intersection points of the arcs, must have the same coverage status as 'p,' i.e., all the intersection points on the boundary of R_x in D_0 must not be covered. This contradicts the assumption that all the intersection points $\text{Cir}_i \cap \text{Cir}_j \in D_0, \forall i, j = 1, \cdots, n$ are covered. Therefore, if all the intersection points $\text{Cir}_i \cap \text{Cir}_j \in D_0$ are covered by one or more adjacent spheres, then D_0 is covered. Consequently, Cir_0 is covered. □

Theorem IV-B.1 indicates that a sensor node S_0 is border redundant if all the intersection points $\text{Cir}_i \cap \text{Cir}_j \in D_0$ are covered by some sensor $S_l, l \neq i, j = 1, \cdots, n$. Therefore, to check if S_0 is border redundant; one has to first find all the circles obtained by the intersection of $S_0 \cap B_m,\ m = 1, \cdots, 6$. For each Cir_k, find all the intersection points that lie within D_k. If all these intersection points are covered, then the circles Cir_k are covered. Then, by the Theorem IV-B.1, S_0 is border redundant and can be deactivated without affecting the overall cubical border coverage.

```
3D distributed border coverage algorithm
For each node $S_i$, form the set of neighbors $N(i)$.
Step 1: Find the intersection circles $\text{Cir}_i$.
    Find the intersection circle $\text{Cir}_i$ resulting from the intersection of $S_i$'s sensing region with the boundary of the region of interest. Note: $S_i$ might intersect 2 or 3 boundary planes of the region in semi circles. The same procedure will still apply.
Step 2: For every pair of nodes $S_k, S_l$ in $N(i)$.
  ▷ Find the intersection circle $\text{Cir}_{k,m} = A_k \cap B_m$ and $\text{Cir}_{l,m} = A_l \cap B_m$ (where $B_m$ is the boundary plane being tested).
  ▷ Find the intersection points $\text{Cir}_{k,m} \cap \text{Cir}_{l,m}$ in $\text{Cir}_i$.
  ▷ If the intersection points are all covered, i.e., $\text{Cir}_{k,m} \cap \text{Cir}_{l,m} \in A_n$, for some $S_n \in N(i)$, where $n \neq i, k, l$, then deactivate $S_i$ since it is a border redundant node.
```

The algorithm guarantees that every point on the boundary of the target region is covered by at least one sensor. The minimal set of sensor nodes is also selected. The computational complexity of the algorithm developed in this section is $O(nN^3)$ where n is the total number of sensor nodes in the network and $N = (\max_{i=1}^{n} |N(i)|)$. The key to both 2D and 3D border coverage algorithms is that they are performed in distributed manner. The 3D distributed coverage algorithm requires that each sensor node knows the information about locations of all sensing neighbors.

All nodes are assumed to know their own location using a localization algorithm or using the onboard GPS devise. The algorithm maintains a table of known sensing neighbors based on the beacons (HELLO messages) that it receives from its communication neighbors. Assuming that $R_c > 2R_s$, the sensor nodes need to include only their locations in the HELLO messages. When a network is deployed, all nodes are initially active. Redundant nodes will switch to the inactive mode until no more nodes can be turned off without causing coverage holes in the region. The distributed algorithm consists of two steps. First, each node advertises its position and listens to HELLO messages from other nodes to obtain neighboring nodes' position information. Secondly, each node runs the border coverage algorithm (2D or 3D) discussed earlier and decides whether to deactivate or not. The details of these two steps are introduced as follows. To obtain neighbor node information, a simple approach is that each node broadcasts a HELLO message, which contains node ID and its current location, at the beginning of each round. Note: If nodes have different sensing ranges (due to depletion of power), the message should also include the current sensing range of the transmitter as well. After finishing the collection of neighbor information, each node evaluates its eligibility for turning off by running the 3D coverage algorithm. However, if all nodes make decisions simultaneously, blind points may appear. To avoid such a problem, each node announces to its neighbors that it is currently running the coverage algorithm. If the node is redundant and is eligible for turning off without affecting the overall coverage, it will broadcast a GOODBYE message to its neighboring nodes. Neighboring nodes receiving a GOODBYE message will delete the sender's information from their neighbor lists.

While the results of Section IV make possible the selection of a subset of sensor nodes in a WSN to border cover a region, the result is a reduced border cover but not necessarily an optimum cover for the region. Further, since the algorithm does not produce a unique result, it is advantageous to have a performance measure for comparing two different collections of sensor nodes that border cover a region.

Definition IV-B.3: The measure of optimality of a border cover of a WSN is the ratio of the number of active border sensor nodes in the network to the minimum number of sensor nodes that can border cover the same region.

The results in Section III found the locations of sensor nodes to achieve optimum deployment for border coverage a region $\mathbf{R}$. Therefore, given the region to be monitored for border coverage, one could easily find the number of sensor nodes required and their location for border coverage. However, if the sensor nodes are already deployed and a subset of these sensor nodes selected to keep active, then the *measure of optimality* is a measure of excess energy spent in monitoring the region as compared to an optimum deployment of the sensor nodes. A network with a lower *'measure of optimality'* would result in lesser expenditure of energy in monitoring the region.

V. ALGORITHM OPTIMIZATIONS

In this section, we provide an optimization to our border coverage selection algorithm in order to improve the border coverage lifetime of the region. Fault-tolerance is the ability of a system to deliver a desired level of functionality in the presence of faults. Fault-tolerance is crucial for many systems and is becoming vitally important for computing- and communication-based systems as they become intimately connected to the world around them, using sensor nodes and actuators to monitor and shape their physical surroundings. Sensor networks introduce new challenges for fault-tolerance. Sensor networks are inherently fault-prone due to the shared wireless communication medium: Message losses and corruptions (due to fading, collision, and hidden-node effect) are the norm rather than the exception. Moreover, node failures (due to crash and energy exhaustion) are common. Thus, sensor nodes can lose synchrony and their programs can reach arbitrary states. Since on-site maintenance is not feasible, sensor network applications should be self-healing. Another challenge for fault-tolerance is the energy-constraint of the sensor nodes. The algorithms presented in Section IV allow us to select a minimum subset of the sensor nodes already deployed that will guarantee border coverage of a given region. However, the emergence of border coverage holes in the target area is unavoidable due to the following reasons:

1) *Sensor failures*: Nodes are subject to failures due to depleted batteries or, more generally, due to environmental influences.
2) *Position changing*: A lot of environmental factors (wind or storms) may change the sensor node's positions over time and possibly resulting in some border coverage holes in the network.
3 *Presence of obstruction*: Some obstacles in the region of interest might impair the nodes sensing/communication functionality and thus result in some border coverage holes.

A proactive method of utilizing the total energy is to assign tasks for each sensor nodes so that a hole is never formed. Though this solution might give optimal solutions, it is impractical in real-time applications. In this section, we provide a reactive and practical approach to minimizing border coverage holes as they (or before) are formed. We provide this *self-healing* algorithm as an enhancement to our border coverage algorithm developed in the previous section. We call it "self-heal" as the actuation is not governed by a user command or application but initiated by the WSN to salvage its own performance.

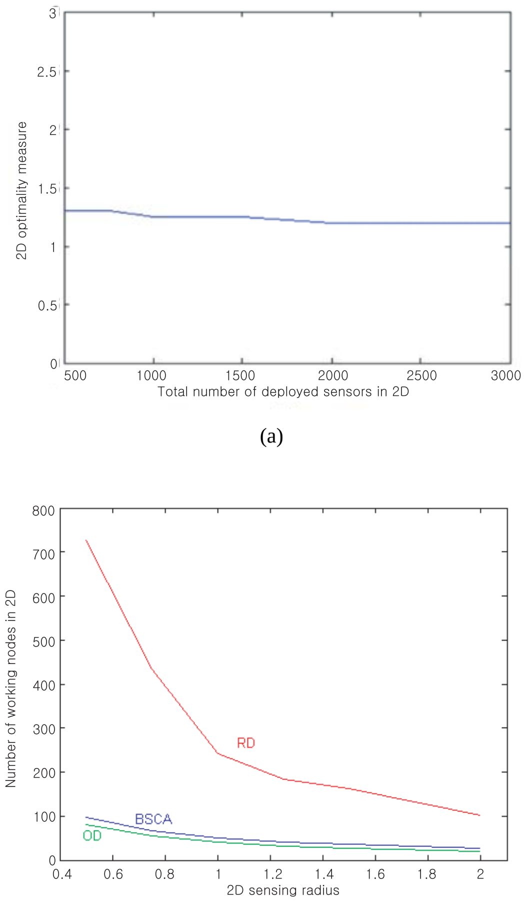
We assume that the nodes know their initial energy content and can keep track of their energy expenditure and therefore can predict their own death. Sensor nodes are randomly deployed in a region of interest to be border covered. Every node acquires information about its location and communication neighbors. The border coverage selection algorithm developed in Section IV is run in a distributed fashion on each node and an optimal border cover of the region of interest is selected. However, while running the border coverage selection algorithm, each sensor node keeps track of the border neighbors, i.e., the neighboring nodes who are also border sensor nodes. If a node is about to run out of energy (before the energy level goes below a specific threshold), it runs the selection algorithm on its border neighbors to select an optimal set of sensor nodes to be its substitute border cover, i.e., to cover its border intersection in case of its death. It then broadcasts a HELP message in order to activate the sleeping nodes that will minimize the border coverage hole. Sensor nodes can be also misplaced or destroyed accidentally or deliberately. Since each node knows its own location and whether it is a part of the border cover set or not, upon realizing the malfunctioning of its sensor, a node broadcasts a HELP message in order to cover the border coverage hole. This simple extension of our border coverage algorithm results in better energy utilization and extends the border coverage lifetime of the region.

VI. SIMULATION AND NUMERICAL RESULTS

The theoretical developments in Sections II–V are validated through numerical examples in this section. The case of random deployment of sensor nodes is studied and compared to the optimum deployment for border coverage. Both 2D and 3D cases are considered and the number of sensor nodes required for border coverage is studied. The number of sensor nodes required to cover a 2D region of size 10 units by 10 units (or a 3D region of size $10 \times 10 \times 10$) is considered. Random deployment, optimal deployment and optimal selection of the nodes for border coverage are studied for different values of the sensing radius. The optimization to our border algorithm is also tested and resulting border coverage lifetime of the network is analyzed.

To test for border coverage we divide the region of interest in a 2D or 3D grid and develop a centralized algorithm which tests for border coverage by generating an occupancy grid and checking if the first and last row and the first and last column in this grid are covered. If all the cells in the first and last row and the first and last column are occupied, then the entire region is border covered. The region to be covered is divided into squares of side equaling half the sensing radius of each sensor nodes. Since the region to be covered is divided into a grid with cell size equal to $R_s/2$, any cell in this grid is completely covered only if its center is within a distance of $R_s/2$ from the sensor. Since we are only concerned with the border coverage, at most 12 cells need be checked to verify the border coverage of a sensor and a maximum of $12n$ cells need to be checked for the border coverage region of 'n' sensor nodes. Thus, for each sensor in the network, the covered cells, and thereby the entries in the occupancy matrix, are determined. The cells in the occupancy matrix corresponding to the covered cells are then indexed by '1.' A zero entry in the occupancy matrix indicates an uncovered cell in the region. The number of uncovered cells and their locations can then be used to determine the size and locations of the uncovered border regions. Further, since the cell entries in the occupancy matrix are indexed, an entry 'k' indicates that the cell is covered by 'k' sensor nodes. Thus, the smallest entry in the occupancy matrix gives the minimum thickness of the border cover. The grid generated is the smallest grid entirely covering the region $\mathbf{R}$. If $\mathbf{R}$ is smaller than the grid, then only relevant cubes in the grid can be chosen for testing coverage. This is done by assigning 'X,' i.e., do not care, to cells that are outside the desired region of coverage. Comparing this algorithm to the traditional sequential testing for coverage algorithm, it can be seen that this algorithm requires $O(n)$ steps while the traditional sequential search algorithms requires $O(n^3)$ steps to verify if a region is border covered. The algorithm presented is simple and easy to implement. It not only helps identify the extent of border coverage but also identifies the size and location of the holes in the border coverage.

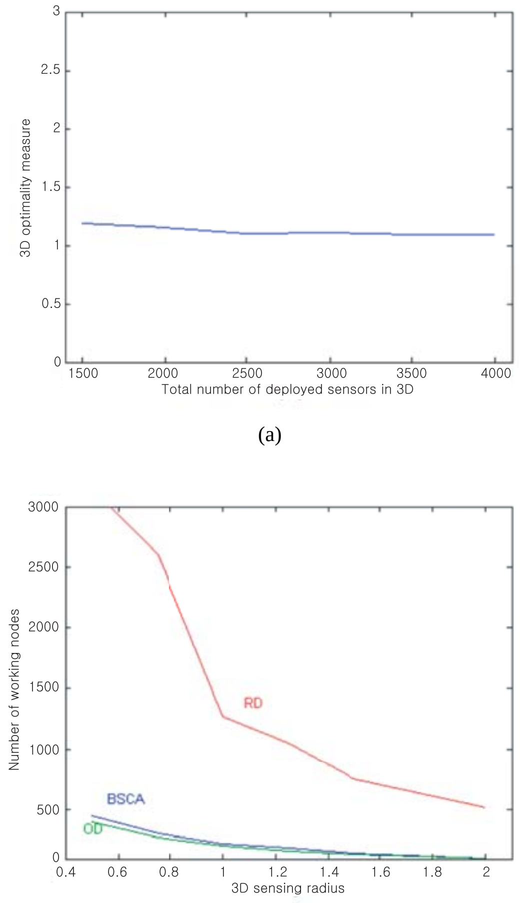
In the first experiment, the optimum 2D coverage algorithm is used to find the optimum border cover of region 10×10 units when sensor nodes are randomly deployed. The nodes have a sensing radius of 1 unit and initially different numbers of nodes are randomly deployed in this region using a uniform distribution. It can be seen that the average *optimality measure* of our border selection algorithm is 1.228 and the nodes that were active in the optimum border cover resulted in average savings of 98.4% (when the number of deployed nodes 500, 1000, 1500, 2000, 2500, and 3000) (Fig. 6(a)). In Fig. 6(b), the required number of sensor nodes with different radii using random de-



(a)

(b)

Fig. 6. (a) The optimality measure of the border selection algorithm for different number of deployed nodes in 2D and (b) comparison between random deployment (RD), optimal deployment (OD), and border selection cover algorithm (BSCA) in 2D.



(a)

(b)

Fig. 7. (a) The optimality measure of the border selection algorithm for different number of deployed nodes in 3D and (b) comparison between random deployment (RD), optimal deployment (OD), and border selection cover algorithm (BSCA) in 3D.

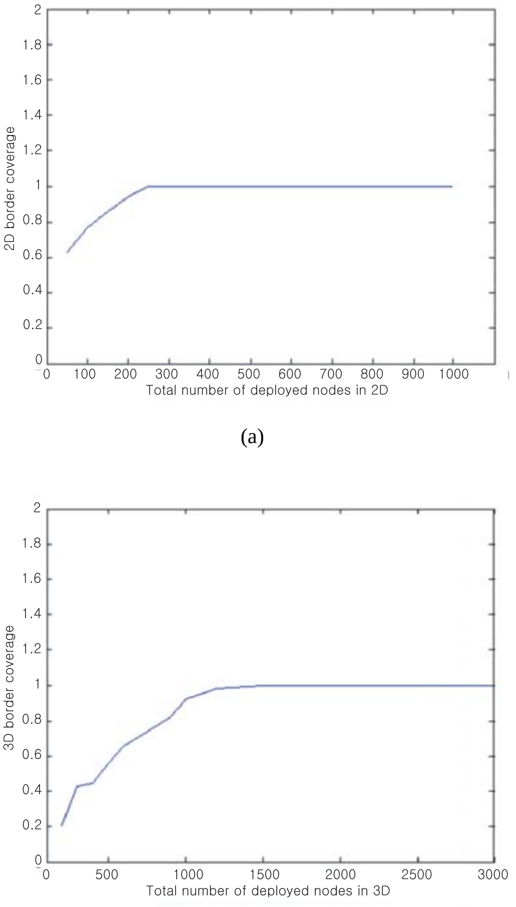
ployment, optimal 2D border deployment and 2D border selection algorithm are compared.

In the second experiment, the same comparison (Fig. 7(a)) is done for the 3D case and the resulting average optimality measure is 1.123 and when the 1500, 2000, 2500, 3000, 3500, and 4000 nodes were randomly deployed. The border selection algorithm resulted in average savings of 93.71%.

In Fig. 7(b) the required number of sensor nodes with different radii using random deployment, optimal 3D border deployment and 3D border selection algorithm are compared.

In the third experiment, we evaluate the border coverage percentage of the region when the sensor nodes are randomly deployed and the border coverage selection algorithm is applied. As we vary the number of deployed nodes, we evaluate the border coverage of the region using the border cover obtained (Figs. 8(a) and 8(b)). It is noticed that after a specific threshold value for 2D and 3D cases, the border coverage percentage is always one. The reason is that random deployment of the sensor nodes does not guarantee border coverage of the region below that threshold.

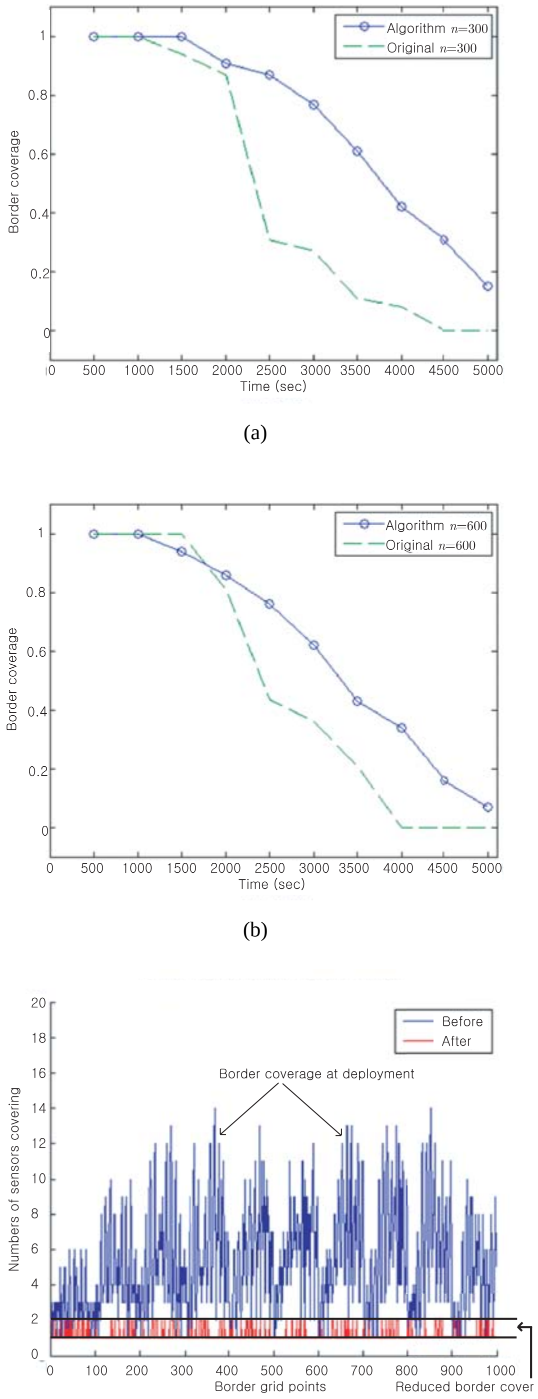
In the fourth experiment, we evaluate the system lifetime. The metrics used in evaluating system lifetime is the border *coverage lifetime*. The overall border coverage lifetime is the continuous operational time of the system before the border coverage drops below its specified threshold (for example 0.9). In Figs. 9(a) and 9(b), the system lifetime is evaluated assuming that each sensor node has a limited energy supply (300 J) and when it runs out of energy it is deactivated. The node deployment densities are 300 and 600, respectively. We started with 300 nodes deployed since that is the minimum number of nodes that guarantee border coverage of the region using random deployment. With each density, the nodes are randomly distributed in a 10 × 10 region network field and each of them starts with an initial energy of 300 J. The power consumption of transmit (Tx), receive (Rx), idle and sleeping modes are 1400 mW, 1000 mW, 830 mW, and 130 mW, respectively. As time passes, sensor nodes will be deactivated due to lack of energy and will leave some coverage holes in the border of the region. If 300 sensor nodes were deployed, after approximately 1600 seconds, the border coverage percentage using the original network will drop below 0.9. However, using the border selection algorithm it needs about 2300 seconds to drop below the threshold. If we increase the number of deployed nodes to 600, the cost for calculating the border cover will increase and thus after approx-



(a)

(b)

Fig. 8. Varying the number of deployed nodes will result in different border coverage percentage for both (a) 2D and (b) 3D regions of interest.



(a)

(b)

(c)

Fig. 9. The coverage lifetime of the network with different number of deployed sensor nodes (a) $n = 30$, (b) $n = 500$, and (c) the degree of border coverage before and after running the border selection algorithm.

imately 1690 seconds the border coverage percentage will go below 0.9. In both experiments, the border coverage lifetime of the network using our border selection algorithm is much better than that using the original network. In Fig. 9(c), we divide the border of the region into 1000 grid points and test how many sensor nodes cover each grid point before and after running the algorithm. When the number of deployed nodes is 600, before starting the algorithm the degree of border coverage is much higher than after running the algorithm which implies that the random deployment is not optimum and therefore a lot of energy is wasted due to multiple active nodes in a given border region. After running the algorithm, most of the redundant border nodes are deactivated resulting in anh energy efficient deployment of the nodes.

In the fifth and last experiment, we do the same comparison that was done in the fourth experiment however with the optimizations mentioned in Section V added. We notice that the system lifetime (border coverage lifetime) is much better than the case if we had started with the original set of deployed nodes. The strength of our developed algorithm is that it allows the sensor network to adaptively reconfigure and repair itself in order to improve its own performance. In Fig. 10, as we increase the number of deployed nodes (from 300 to 600 nodes), the *self-healing* border coverage algorithm performs better since activating a substitute set will result in better percentage of border coverage and therefore the border coverage lifetime of the network is increased. In Fig. 11, an example of the active nodes before and after running the algorithm is presented. 2000 nodes were deployed, and after running the border selection algorithm,

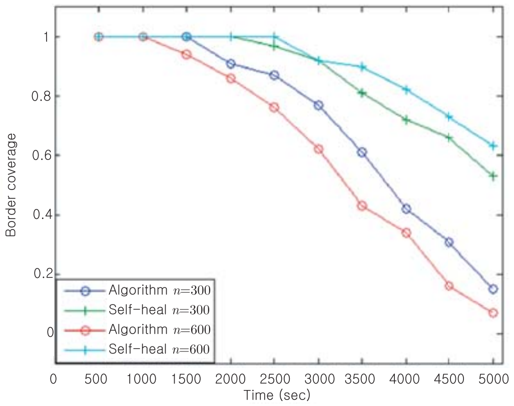
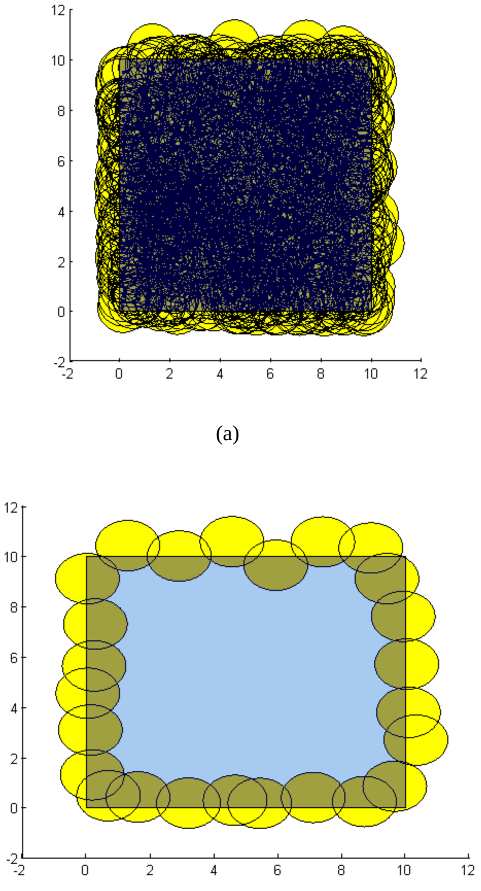


Fig. 10. The coverage lifetime of the network with different number of deployed sensor nodes when using self-healing enhancement of the algorithm.



(a)

(b)

Fig. 11. An example of the border selection algorithm. Active nodes, before and after running the algorithm are shown.

1974 nodes were deactivated resulting in savings of 98.7%.

Minimizing the number of sensor nodes active to border cover a region of interest will result in minimizing the energy consumed by the whole sensor network and thus increasing the lifetime of the network as demonstrated in the simulation results.

VII. CONCLUSIONS AND POSSIBLE EXTENSIONS

In this paper, the border coverage problem in wireless sensor networks was formulated and analyzed. Algorithms were proposed to compute the minimum number of sensor nodes required for border coverage of a given region. Unlike most of the work done in this area we studied both the 2D and 3D cases. A measure of optimality was also proposed that compares a given border deployment of WSN with optimum deployment. This metric is shown to be indicative of the energy efficiency of the WSN and serves as a useful means to select between two different deployments of a WSN. We also provided *self-healing* algorithms as an optimization to our border coverage algorithms which allow the sensor network to adaptively reconfigure and repair itself in order to improve its own performance. Part of our future work is to use the algorithms developed in this paper for tracking applications.

Our algorithms presented can be easily extended to handle different shapes of region to be monitored. If we have region of an irregular shape, we can always use polygon approximation and simplification techniques [32] to find the polygon that bounds the region of interest. In addition to that, the sensing radius of each sensor node need not be equal and our distributed algorithm could be applied to sensor networks with different sensing radii. Fig. 12 illustrates the polygon approximation and the unequal sensing radii properties of some of the sensor nodes. In this paper, we also assumed that the sensing region of each sensor nodes is a disk (2D) or a closed ball (3D). This simplified model is clearly not applicable to all types of sensing measurements, i.e., measuring spatial distribution of local quantities like temperature where the sensing range or disc makes little sense. This model is still overly simplistic for sensing acoustic signals where the signal strength attenuates with distance like electromagnetic waves. Our algorithms could be easily extended to handle a general model based on certain signal-to-noise ratio thresholds, with proper data fusion mechanism to reduce the variance of measurement. The goal would be to find a valid approximation of the intersection segment (or circle) with the boundary (Fig. 13). Part of our future work is to evaluate our algorithms based on realistic general sensing models and to design hybrid border coverage protocols capable of delivering accurate spatio-temporal profile of different kinds of sensing measurements. Some other extensions to our deployment algorithms include deploying the sensor nodes at multiple levels of border coverage (Fig. 14).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the valuable assistance of Miss Diana Audi, MS for her assistance in proving some of the theorems stated in this paper and her insight towards solving the problem in general.

REFERENCES

[1] B. Warneke and K. Pister, “MEMS for distributed wireless sensor networks,” in *Proc. the 9th Int. Conf. Electron., Circuits and Syst.*, Sept. 2002, pp. 291–294.

[2] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for self-

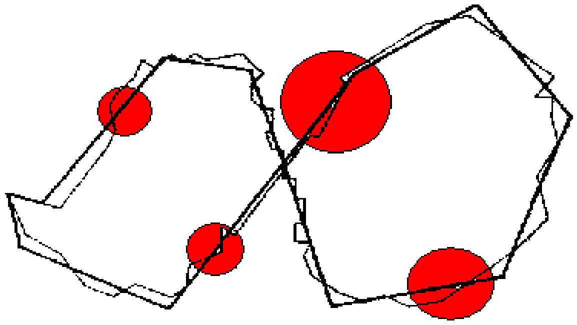


Fig. 12. An example of a polygon approximation and simplification. The thin lines represent the original polygon which has 70 segments. The thick lines are the simplified polygon. This polygon has only 11 segments. Sensor nodes with different sensing radii are also displayed.

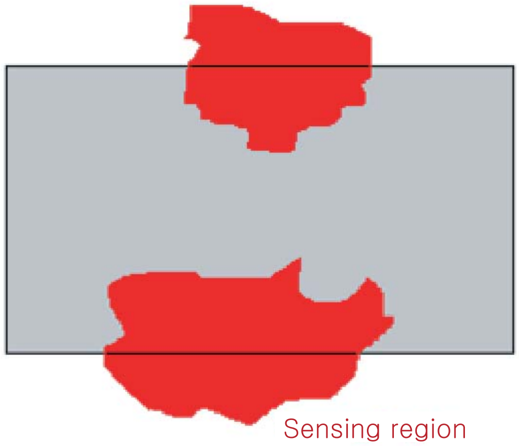
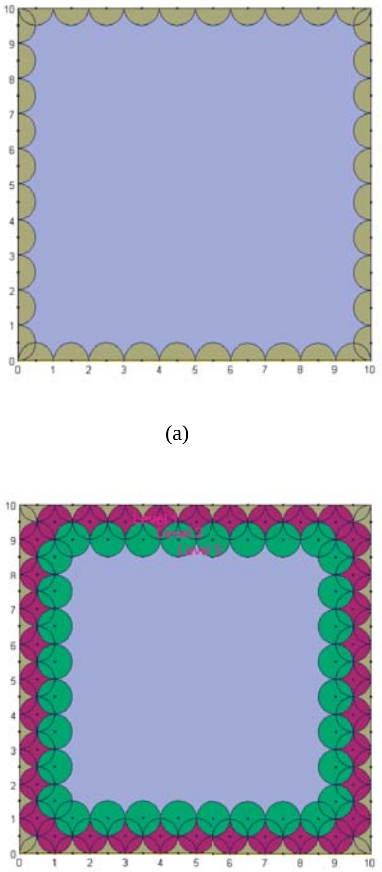


Fig. 13. The sensing region of each sensor node is not uniform. Our Border Coverage algorithm can still work as long as we can find the segment of intersection with the border.



(a)

(b)

Fig. 14. (a) Optimal deployment of sensor nodes in 2D to cover the border of a rectangular region and (b) optimal deployment of sensor nodes with multiple levels.

organization of a wireless sensor network," *IEEE Pers. Commun. Mag.*, pp. 16–27, Oct. 2000.

[3] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High-Performance Computing Appl.*, vol. 16, no. 3, pp. 90–110, Aug. 2002.

[4] J. Aslem, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proc. the 1st Int. Conf. Embedded Networked Sensor Syst.*, Nov. 2003, pp. 150–161.

[5] Q. Fang, F. Zhao, and L. Guibas, "Lightweight sensing and communication protocols for target enumeration and aggregation," in *Proc. the 4th ACM MobiHoc*, June 2003, pp. 165–176.

[6] P. Gupta and P. Kumar, "The capacity of wireless sensor networks," *IEEE Trans. Inf. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[7] P. Gupta and P. Kumar, "Internet in the sky: The capacity of three dimensional wireless networks," *J. Commun. Inf. Syst.*, vol. 1, pp. 33–49, Jan. 2001.

[8] H. Gupta, S. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proc. MobiHoc*, June 2003, pp. 189–200.

[9] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *J. Interconnection Netw.*, vol. 3, pp. 213–229, Mar. 2000.

[10] Extreme scale Wireless Sensor Networking. Tech. Rep., 2004. [Online]. Available: http://www.cse.ohio-state.edu/exscal

[11] S. Kumar, T. Lai, and A. Arora. "Barrier coverage with wireless sensors," in *Proc. the 11th Annual Int. Conf. Mobile Computing and Netw.*, June 2005, pp. 284–298.

[12] B. Cbunar, A. Grama, J. Vitek, and O. bunar, "Redundancy and coverage detection in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp 94–128, Feb. 2006.

[13] M. Watfa and S. Commuri, "Coverage strategies in 3D wireless sensor networks," *Int. J. Distributed Sensor Netw.*, vol. 2, no. 4, pp. 333–353, Oct. 2006.

[14] M. Watfa and S. Commuri, "An energy efficient and self-healing 3-dimensional sensor cover," *Int. J. Ad Hoc and Ubiquitous Computing*, vol. 2, pp. 121–138, Jan. 2007.

[15] M. Watfa and S. Commuri, "The three dimensional coverage problem," *J. Netw.*, vol. 1, no. 4, pp. 10–20, Aug. 2006.

[16] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE ICC*, June 2001, pp. 472–476.

[17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1380–1387.

[18] K. Lieska, E. Laitinen, and J. Lahteenmaki, "Radio coverage optimization with genetic algorithms," in *Proc. IEEE PIMRC*, Sept. 1998, pp.318–322.

[19] D. Gage, "Command control for many-robot systems," in *Proc. the 19th Annual AUVS Technical Symp.*, June 2002, pp. 22–24.

[20] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Proc. IPSN*, Apr. 2005, pp.254–260.

[21] B. Carbunar, A. Grama, and J. Vitek. "Distributed and dynamic voronoi overlays for coverage detection and distributed hash tables in ad-hoc networks," in *Proc. the 10th Int. Conf. Parallel and Distrib. Syst.*, July 2004, pp. 549–559.

[22] X. Hong, M. Gerla, R. Bagrodia, P. Estabrook, T. Kwon, and G. Pei, "The mars sensor network: Efficient, power aware communications," in *Proc. IEEE MILCOM*, Oct. 2001, pp. 112–117.

[23] NASA Sensor Webs Project. [Online]. Available: http://sensorwebs.jpl.nasa.gov/

[24] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor

networks: Research challenges," *J. Ad Hoc Netw.*, vol. 3, pp. 257–279, May 2005.

[25] D. Pompili and T. Melodia, "Three-dimensional routing in underwater acoustic sensor networks," in *Proc. the 2nd ACM Int. Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Netw.*, Oct. 2005, pp. 214–221.

[26] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, "Shallow water acoustic networks," *IEEE Commun. Mag.*, vol. 39, pp. 114–119, June 2001.

[27] Wireless Integrated Sensor Networks. [Online]. Available: http://www.janet.ucla.edu/WINS

[28] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Netw.*, vol. 38, pp. 393–422, Aug. 2002.

[29] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Int. J. Wireless Ad Hoc and Sensor Netw.*, vol. 1, no. 2, pp. 89–124, Sept. 2005.

[30] R. Kershner, "The number of circles covering a set," *American J. Mathematics*, vol. 61, pp. 665–671, 1993.

[31] S. Verblunsky, "On the least number of unit circles which can cover a square," *J. London Math. Society*, vol. 24, pp. 164–170, 1994.

[32] Y. Kurozumi and W. A. Davis, "Polygonal approximation by the minimax method," *Computer Graphics and Image Process.*, vol. 19, pp. 248–264, 1982.

[33] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. the First ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2003, pp. 28–39.

[34] D. Kotwwitz "The densest packing of equal circles on a sphere," *Acta. Cryst. Sect. A*, pp. 158–165, 1991.

[35] Y. Guo and Z. Qu, "Coverage control for a mobile robot patrolling a dynamic and uncertain environment," in *Proc. the fifth World Congress on Intelligent Control and Automation*, June 2004, pp. 4899–4903.

[36] C. Zong and J. Talbot, *Sphere Packings.* New York: Springer-Verlag, 1999.

[37] H. S. M. Coxeter "The problem of packing a number of equal nonoverlapping circles on a sphere," *Trans. New York Acad. Science.* pp. 320–331, 1962.

[38] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proc. USENIX Tech. Annual Conf.*, 2002, pp. 317–328.

[39] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *Cluster Comput.* vol. 5, no. 2, pp. 157–167, 2002.

[40] Y. Go and Z. Qu, "Coverage control for a mobile robot patrolling a dynamic and uncertain environment," in *Proc. the 5th World Congress on Intelligent Control and Automation*, Hangzhou, China, June 15–19, 2004.

Mohamed K. Watfa is currently an Assistant Professor at the American University of Beirut (AUB). He received his Ph.D. from the School of Electrical and Computer Engineering at the University of Oklahoma in Norman, OK. He obtained his B.S. in Computer Science from American University of Beirut in 2002 and his Masters in Engineering Science from the University of Toledo, OH in 2003. His research interests include wireless sensor networks, wireless networking, resource management, energy issues, tracking, routing, and performance measures. He was the youngest doctor to ever graduate from his university at the age of 23. He has more than 25 top journal and conference papers in the areas of wireless sensor networks. He is also the author of a number of books and book chapters in the area of wireless networking. He is a Member of the ACM and IEEE.

Sesh Commuri is an Associate Professor in the School of Electrical and Computer Engineering at the University of Oklahoma in Norman. He received his Masters in Electrical Engineering from the Indian Institute of Technology, Kanpur, India in May 1989 and Ph.D. in Electrical Engineering from the University of Texas at Arlington in May 2006. Prior to 2002, he helped several positions in the industry in the development of embedded controllers and WCDMA cellular phones. His research interests include wireless sensor networks, intelligent systems, real-time systems, mechatronics, embedded controls, and robotics.