Surveillance with wireless sensor networks in obstruction: Breach paths as watershed contours

Ertan Onur, Cem Ersoy, Hakan Deliç, Lale Akarun

Wireless and Mobile Communications Group, EEMCS, Delft University of Technology, P.O. Box 5031, 2628 CD Delft, The Netherlands

NETLAB, Department of Computer Engineering

WCL, Department of Electrical and Electronics Engineering

nLab, Department of Computer Engineering, Bogaziçi University, Bebek 34342, Istanbul, Turkey

Abstract

For surveillance applications of wireless sensor networks, analysis of sensing coverage and quality of sensing is crucial. For rough terrains where obstacles block the sensing capability, region-based approaches must be employed to determine the sensing quality. In this paper, we present a method to determine the breach paths and the deployment quality defined as the minimum of the maximum detection probabilities on the breach paths in the presence of obstacles. We propose the utilization of watershed segmentation on the iso-sensing map that reveals the equally-sensed regions of the field-of-interest in a surveillance application. Probabilistic sensor models are utilized to produce the iso-sensing map considering the sensing coverage degree and reliability level as the design criteria. The watershed segmentation algorithm is applied on the iso-sensing map to identify the possible breach paths. An algorithm is proposed to convert the watershed segmentation to an auxiliary graph which is then employed to determine the deployment quality measure (DQM). The effects of the sensor count and coverage degree on the DQM are analyzed.

1. Introduction

Suppose that a rough terrain is to be monitored to detect unauthorized entries as shown in Fig. 1a. This task can be risky for humans. Instead of deploying a wired surveillance system, a wireless sensor network is an easy alternative where the sensors can be dropped by an aircraft. As soon as the sensors configure themselves, they can start sensing the environment and communicate the target presence decisions to a sink.

Measures must be defined to analyze the sensing quality that signifies how well a surveillance wireless sensor network (SWSN) covers a region and senses the phenomena of interest such as an intrusion. The sensing quality depends on the characteristics of the sensor, target and environment, as well as the number and the deployment scheme of the sensors. The variety of sensor technologies makes the coverage analysis difficult because the underlying signal processing and detection procedures depend on the physics of the devices. If the target detection probability is well-defined, a quality of sensing measure can be established. The sensing quality can be defined through how well the breach path is covered. Finding breach paths is referred to as the weakest breach path problem [1], or the best coverage problem [2]. Megerian et al. define the worst-and the best-case coverage of a WSN for homogeneous sensors [3]. The target wants to avoid the sensors. Thus, passing far from the sensor is good from the target’s viewpoint. The worst case coverage (maximal breach path) is defined with the quality of sensing measure as the closest distance to the sensors while the target crosses the field [4]. A similar problem derived from the worst case coverage definition is to find the farthest distance to the sensors when the target wants to...
stay as close as possible to the sensors. This problem is referred to as the best case coverage (maximal support path) problem. Such worst case measures can be employed in applications with high security requirements such as the surveillance of a mission-critical place. Clouqueur et al. consider the unauthorized traversal problem where the likelihood of detecting the target (referred as path exposure) is studied [5]. The field is modelled as a grid in [5] and the grid size is equal to the product of the target speed and the sensor sampling period. If the sampling periods of sensors are not synchronized, this approach weakens the accuracy. Moreover, signal characteristics and environmental conditions are not taken into account in [3,5].

The main research question addressed in this paper is: How can the sensing quality be measured in the presence of obstacles in a SWSN? We propose a deployment (sensing) quality measure in terms of the detection probability of a target crossing the field on breach paths. This deployment quality measure (DQM) can be used to design the network to provide a required sensing quality and to answer typical questions such as how many sensors or how the sensors should be deployed. If the field of interest is known a priori, and deterministic deployment is possible, the designer may utilize the near optimal sensor placement algorithm proposed by Lin and Chiu [6] to provide a complete sensing coverage. For random deployment, it is concluded that the optimal sensor placement problem is NP-complete [6]. Younis and Akkaya present an extensive survey on the issues for node placement in WSNs in [7]. Therefore, it is practically important to have an accurate DQM because the main functionality of a SWSN is to secure a field-of-interest.

In the next section, we discuss how to find the breach paths using well-known morphological segmentation methods when there are obstacles in the field. How to determine the weakest breach path is beyond the scope of this paper. After formulating the problem, the deployment quality measure is introduced in Section 5. Then, simulation results are discussed and the conclusion is drawn in Section 7.

2. Breach paths

Traditionally, Voronoi decomposition is utilized to reveal breach paths based on exposure definitions [8,3,9,10]. Voronoi decomposition of a discrete set of sensors distributed in the Euclidean space is the partition of the plane where each sensor is associated with a region. All of the points in the region of any sensor are closer to that sensor than any other. The borders between the regions, which are equidistant to multiple sensors, are the breach paths. Given a set $S$ of discrete number of generating points (sensors) in the Euclidean space (surveillance field), for any point $(x, y)$ in this space, there is one point from $S$, say $s_i$, to which $(x, y)$ is closer than any other point in $S$ except the equally distant ones. Hence, the set of closest points to $s_i \in S$ produces a convex polytope referred to as the Voronoi cell. The set of Voronoi cells tessellates the whole Euclidean space [11].

The major difference of our work from the others is the presence of obstacles. When there are obstacles in the field model, some parts of the field cannot be monitored because sensors may not detect the phenomenon due to the lack of line-of-sight. If there is an obstacle between the sensor and the target, considering only the sensor-to-target distance misleads the calculations. The simple Voronoi decomposition approach considers just the positions of the sensors (in a way just the Euclidean distance), and therefore falls short of finding the breach paths when there are obstacles in the field. Obstacles may also block the trajectory of the target. A simple scenario is depicted in

![Fig. 1. How Voronoi segmentation fails and watershed segmentation avoids the obstacles.](image-url)
Fig. 1b, where the breach path found through Voronoi segmentation passes over the obstruction, which may be impossible geographically.

A generalized weighted Voronoi decomposition approach can be developed to address the line-of-sight problem caused by obstacles. For example, a region growing algorithm can be designed where the generating points of the Voronoi tessellation are the positions of the sensors and the obstacles. Each generating point is marked with an explicit tag. All points covered by the obstacles are tagged. Then, at each stage of the algorithm, the generating points offer their tags to neighboring grid points with a predetermined weight (speed) until all grids are tagged. The region growing weight is the number of the hops to the grid to which the tag is offered. If the region growing speeds of the obstacles and sensors are set to zero and one, respectively, then, the produced contours do not cross the obstacles and pass far from the sensors. Determining the weights is not easy when heterogeneous sensors are used or the geographic characteristics are not uniform throughout the field. Delaunay triangulation, which is the straight-line dual of the Voronoi diagram [12], is used in [13] to determine locations in the field for redeployment, where deployment quality is defined as the coverage percentage of the field. An adaptive sensor deployment algorithm that uses Delaunay triangulation and considers the boundaries of the obstacles as virtual sensors in order to avoid them is presented in [14]. Such an approach increases the number of generating points and may not work in a 3-D field model.

When there are many obstacles or when the topology of the field is complex, another approach that works using the topology instead of individual sensor and obstacle positions is required. This hints at using the watershed segmentation that considers the topology as a discretized image. Because Voronoi decomposition considers only the sensor (and obstacle) positions, it is not possible to fuse the decisions of the sensors and subsequently determine the breach paths. Morphological segmentation methods can be utilized instead of the generalized weighted Voronoi decomposition approach. In this paper, we employ a well-known morphological segmentation method, namely the watershed segmentation [15] to determine the breach paths in the presence of geographic or man-made obstacles as shown in Fig. 1c. The details of the watershed segmentation are presented in Section 4. Meyer proved that if appropriate distance measures are used, then morphological segmentation produces Voronoi tessellations [16]. When there are no obstacles in the field, the geodesic distance reduces to the Euclidean distance and the geodesic Voronoi decomposition becomes the simple Voronoi decomposition and fails to solve the line-of-sight problem.

When the field-of-interest is modeled as a grid, the detection probabilities (or exposure levels) can be calculated for each grid point depending on the positions of the sensors. Adding the detection probability as the third dimension to a 2-D field model, a surface referred as the iso-sensing map is obtained as shown in Fig. 2. The iso-sensing map represents the equally-sensed areas. In the next section, we explain how the watershed segmentation algorithm is applied on the iso-sensing map to identify the possible breach paths.

3. Iso-sensing map definition

In this section, we define how the iso-sensing map is produced, and present how the watershed algorithm is used to determine the breach paths.

3.1. Field and obstacle models

The field is a long narrow strip modeled by an $N \times M$ grid as seen in Fig. 2, where $N > M$. Throughout the paper, we use the term grid points to address the dimensionless intersection points of equally space horizontal and vertical lines. For a grid point $(x, y)$, the $4$-connected neighborhood is defined as the set of grid points $G_4(x, y) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\}$ and the $8$-connected neighborhood is the set of grid points $G_8(x, y) = G_4(x, y) \cup \{(x + 1, y + 1), (x - 1, y - 1), (x - 1, y + 1), (x + 1, y - 1)\}$. We use the $8$-connected neighborhood definition in the field model. The two grid points $(x_1, y_1)$ and $(x_2, y_2)$ are assumed to be connected if $(x_2, y_2) \in G_8(x_1, y_1)$.

We tag one side of the field as insecure and the opposite side as secure as shown in Fig. 1a. The secure and insecure sides represent the destination and the origin of intruder traversals, respectively. We do not address any specific points in either side. The definition of the secure and insecure sides is an engineering decision to be given by the network designer based on the environmental properties such as the topology and the application requirements.

To be faithful to the real world, obstacles that disable sensing and physical traversals are incorporated in the field model. The objective of the target is to go through the field-of-interest while simultaneously avoiding the obstacles. A sample field model can be seen in Fig. 2. In this example, the field is modeled by $160 \times 40$ grids where each grid is $1 \times 1$ m. In this work, sensors are deployed randomly and their locations are assumed to be known as in [10,18,17]. The sensor and the obstacle positions are not restrained to the grid points and can be floating numbers.

In the simulations, we consider obstacles as isotropic discs where the position and the radii are uniform random.
variables. The coordinates of the obstacle centers and the perimeters of the obstacles do not need to align with the grid points. The obstacles block both the line-of-sight of sensors and the breach paths. Depending on the operational characteristics of the sensors, the effect of obstacles on the sensing performance may change. For example, if radar operates using radio signals, as the frequency increases, the penetration capability of signals through the obstacle reduces significantly. For infrared and sonar, the effect may be different. The ability to sense through obstacles depends on the size, shape, motion and temporal change of the nature of the obstacle. For the sake of analytical simplicity, we assume that sensing through an obstacle is impossible. Instead of trying to model obstacle effects for all kinds of sensors, in this paper, we present a generic model. For a detailed discussion on the effects of obstacles the reader may refer to [18].

The probability of detecting a target on each grid point can be calculated if there is line-of-sight between the target and the sensor (e.g., no obstacles exist in between). To incorporate the obstacles in the model, we define \( o_{ijk} = 1 \) if there is line-of-sight between the target at grid point \((i,j)\) and the sensor \(k\), and \( o_{ijk} = 0 \) otherwise. If obstacles are not modeled, then \( o_{ijk} = 1, \forall i,j,k \). Depending on the type of the sensor, obstacles may not disable sensing functionality completely. For such situations, the designer may assume \( 0 < o_{ijk} < 1 \) as a sensing degradation factor. In this paper, we consider the Boolean approach as Kansal et al. presented in [19], where they analyze the effect of obstacles on the sensing coverage.

A similar obstruction approach is presented in [20].

### 3.2. Detector models

Binary detector is the most common model in WSN research [21]. The sensing coverage of a sensor is modeled as an isotropic disc with radius \( d_t \) [22]. If the target trajectory intersects any disc, it is assumed to be detected. As experimented by Cao et al. in [23], the sensing capability of the passive infrared sensors (PIR) is not isotropic and the sensing ranges for any direction can be modeled with normal distribution. However, for PIR sensors, if the sensor-to-target distance exceeds the sensing range by \( 3-7 \), the detection performance deteriorates sharply. This result shows that PIR sensors can be approximated by the binary detection model. Isotropic binary detection assumption is commonly adopted for its analytical simplicity, and it may be acceptable when line-of-sight is ensured, e.g., for indoor deployment.

The micropower impulse radar (MIR) [24,25] can be employed outdoors. Current commercial MIR devices have a sensing range around 15 m. The signal path-loss occurs with propagation exponent \( \eta \). During an observation epoch, the breach decision is made based on \( I \) data samples, where the sensor-to-target distance is assumed to remain the same. The detection probability of a target at grid point \((i,j)\) by sensor \(k\) in additive white Gaussian noise with zero-mean and variance \( \sigma^2 \) is

\[
p_{ijk} = 1 - \Phi \left( \Phi^{-1}(1 - \alpha) - \sqrt{\frac{2}{\pi}} d_{ijk}^{-\eta} \right),
\]

where the parameters \( d_1^2, d_2^2, \beta \) and \( \beta \) are adjusted according to the physical properties of the sensor. Both detectors exhibit an exponential behavior, and Elfes’s model can accommodate the Neyman–Pearson detector through proper parameter matching [21].

In this paper, instead of the common binary detection with static [29–31] and adjustable sensing ranges [32,33], we use the Neyman–Pearson and Elfes’s detector in the simulations. The latter models present a particularly suitable formulation for radar sensing, which is pivotal in the surveillance of very large areas. In the following section, we define different approaches to produce the iso-sensing map in obstruction. Since the iso-sensing map definition does not depend on any particular detector model, the readers may consider using other probabilistic sensor models to produce the iso-sensing map.

### 3.3. Iso-sensing map

In this section, we present how the iso-sensing map is produced. We assume that a set of sensors is deployed in the 2-D field and there are obstacles in the field. We assume that obstacles block target traversals, as well as all detections and communications. The detector models, presented in the previous section, define the sensing quality based on the sensor-to-target distance. Using such detector models, the target detection probability for any point in the field can be calculated using the iso-sensing map definitions presented in this section. This calculation produces the iso-sensing map which resembles a topographic map where the detection probability can be deemed as the altitude; i.e., the third dimension. To decrease the computational complexity of the calculation, the field is modeled as a grid and the detection probabilities are calculated only for grid points. The iso-sensing map can also be considered as an image where the detection probabilities are mapped to the intensities of the pixels. In Section 4, we present these issues in detail.

In the iso-sensing map definitions, the decisions of a subset of sensors are fused to produce the breach decision which may or may not be the case in an active network. Operationally, all nodes are in sensing mode unless any sleep scheduling for the sensing circuitry is implemented. Therefore, the following iso-sensing map definitions and the calculations can be deemed as an off-line process.
For grid point \((i, j)\), define the set of sensors in decreasing order of detection probabilities as \(S_j = \{s_1, s_2, \ldots, s_l, \ldots, s_R\}\) where \(s_l = 1, \ldots, R\), is the identity of the sensor with \(p_{\text{pt}}^{(i,j)} \geq p_{\text{pt}}^{(i,j)} \geq \cdots \geq p_{\text{pt}}^{(i,j)},\) and \(R\) is the total number of sensors deployed in the field. Throughout the lifetime of the WSN, sensor nodes do not communicate to sort out their detection probabilities for any grid point; they just function to sense and communicate the phenomenon.

### 3.3.1. K-degrees of iso-sensing (KIS)

In this type of iso-sensing map definition, \(K < R\) of the closest sensors (first \(K\) sensors in \(S_j\)) act on the decision for a grid point. Then, the detection probability of a target on grid point \((i, j)\) is

\[
p_{\text{pt}}^{\text{KIS}} = 1 - \prod_{l=1}^{K} (1 - p_{\text{pt}}^{(i,j)} o_{ijl}).
\]

Kumar et al. propose algorithms in [34] to determine whether a deployment provides \(K\) degrees of coverage or not. This type of iso-sensing surface definition is useful when the nodes sleep from time to time for energy conservation. Designing a sleep scheduling algorithm according to the sensing performance is beyond the scope of this paper. But, for example, Lui et al. propose a scheduling algorithm without accurate location information subject to sensing coverage and connectivity requirements [35]. For a given coverage degree, they propose a lower bound on the required number of sensors to provide a coverage intensity level. Coverage intensity is defined as the ratio of active time to the total time where the points in the field is covered with at least one active sensor. For other scheduling algorithms considering sensing coverage and connectivity, the reader may refer to [36,37] and a detailed survey of coverage and connectivity issues can be found in [38].

Other application level motivations for more coverage degrees can be based on accuracy or fault tolerance. For applications such as battlefield surveillance, increasing the degree of sensing coverage, improves the sensing accuracy. In other words, failure of an individual sensor must not block network operation. Hence, fault tolerance can be provided with higher coverage degrees. Another motivation is that when the position of the target is to be estimated, measurements from multiple sensors are required. For triangular positioning algorithms, the degree of coverage may be adjusted accordingly.

### 3.3.2. Reliable K-degrees iso-sensing (KRIS)

In KIS, large values of \(K\) increase the variance of sensor decisions. For a more reliable design, \(S_j\) can be bounded with a reliability factor such that \(r_{ijl} = 1\), if \(p_{\text{pt}}^{(i,j)} > p_l\), and \(r_{ijl} = 0\), otherwise, giving rise to the Reliable K-degrees iso-sensing (KRIS) where the detection probability becomes

\[
p_{\text{pt}}^{\text{KRIS}} = 1 - \prod_{l=1}^{K} (1 - p_{\text{pt}}^{(i,j)} o_{ijl} r_{ijl}).
\]

The threshold probability \(p_l \in (0.5, 1)\) represents the confidence level of the sensor and it is an engineering parameter which depends on the application requirements. Large values of it indicate better security since the points in the field with smaller target detection probabilities than the threshold are regarded as sensing holes with zero detection probabilities. Hence, \(p_l\) sets the requirement for the full coverage of the field with a minimum security level. Although the \(p_l\) does not impact the computational complexity, larger values of it impose greater sensor demand.

From another viewpoint, the reliability threshold can be mapped to sensing distance. That is, sensor decisions are deemed sufficiently reliable to be incorporated in the calculations only at those \(d_{ijl}\) distances where \(p_{\text{pt}}^{(i,j)} > p_l\). Depending on the application and the false alarm requirement, typically \(p_l \geq 0.9\). Ideally, \(p_l < p_{\text{pt}}^{(i,j)}\); otherwise, the degree of the coverage is smaller than \(K\). Moreover, in an active network the sensor does not decide if it is reliable or not; it just decides if there is a target according to its functional design.

### 3.3.3. Reliable subset iso-sensing (RIS)

Since only those sensor decisions that are sufficiently reliable are incorporated in KRIS, to alleviate the problem of determining an appropriate value of \(K\), the designer may prefer to set \(K = R\) so that

\[
p_{\text{pt}}^{\text{RIS}} = 1 - \prod_{k=1}^{R} (1 - p_{ijl} o_{ijl} r_{ijl}),
\]

which is referred to as the reliable subset iso-sensing (RIS). Uncorrelated sensor decisions are implicitly assumed here. However, multiple sensors may observe a common volume, and if a sensor detects a target, it is highly probable that there is another sensor seeing the same target. With RIS, the total effect of the sensor correlations is bounded because low probability values are truncated. Hence, the variance is not affected.

### 3.3.4. Most-dominant-sensor iso-sensing (MDIS)

In dense deployment, assuming the same environmental conditions, a target can be detected by several sensors, implying correlations among sensor decisions. Based on this phenomenon, the designer may assume that the sensor with the largest detection probability dominates the grid point. A similar approach is considered in [8] where observability by the closest sensor is dominating. In this case,

\[
p_{\text{pt}}^{\text{MDIS}} = \max_{1 \leq k \leq R} (p_{ijl} o_{ijl}).
\]

This model is equivalent to KIS with \(K = 1\). A stricter approach would be to assume that \(p_{\text{pt}}^{\text{MDIS}} = 0\) if the detection probability of the closest sensor is less than some threshold. Clearly, KIS and RIS definitions necessitate redundant deployment and they function in a similar way in terms of detection performance. More sensors must be put in the field to decrease target-to-sensor distances and increase the coverage degree. Consequently, while designing a network according to KIS, KRIS or RIS provides greater reliability compared to one designed following the MDIS formulation, it is also more costly because of the denser deployment requirement. When the SWSN designs based on these models are sorted in terms of decreasing reliability level, it follows the KRIS, RIS, KIS and MDIS order.
Calculating the detection probabilities for each grid point produces the iso-sensing map (see Fig. 2). After generating the 2-D image (Fig. 3), the watershed algorithm can be applied to determine the breach paths. In image processing, the watershed algorithm is applied to a gradient image to find edges in the original image, or the highest peaks in the gradient image [15]. Therefore, our iso-sensing map may be interpreted as an inverted gradient image. In the following section, we point to an analogy between an iso-sensing map and a topographical relief and present how the breach paths can be revealed through the watershed algorithm.

4. Watershed segmentation

In image processing, the gradient images which correspond to the topographic reliefs are often denoted with gray-scale pictures. The gray tones in the image depict the elevations in the region. Image segmentation is a process to discriminate the objects in an image from the background. A common approach is to find disjoint regions that are homogeneous with respect to some property. Watershed segmentation is a region-based approach and the idea behind this algorithm comes from nature. The watershed transform is a region-based segmentation approach.

Watershed segmentation can be described analogously to water swamping from the minimal plateaus of a 3-D topographic surface, where the third dimension is the altitude. Minimal plateau is a set of points with an altitude from which it is impossible to reach a lower altitude without having to climb. As the water rises in the catchment basins, the floods will merge. Dams are built to separate the floods. The catchment basin associated with a minimal plateau is the set of points such that a water drop that falls on one of these points flows until it reaches the minimal plateau. After the immersion, only the dams that separate the floods are visible. That is, the topographic surface is divided into regions separated by the dams referred as watersheds or the dividing lines that separate the catchment basins. The labelling process of the revealed regions is referred to as the watershed transform [39]. This approach is generally referred to as immersion analogy. The watershed segmentation algorithm by simulated immersion is presented by Vincent and Soille in [15].

When the miss probabilities are quantized to gray-scale color intensities, the iso-sensing map can be considered as a 2-D image on which the watershed segmentation can be easily applied. After deployment, the iso-sensing map of the SWSN can be computed. The iso-sensing map depicts hills and valleys where the altitude is mapped to the miss probability. The lowest points of the surface are the sensor positions. Therefore, analogously, water starts flooding from the sensor positions. The contours (dams) produced by the watershed segmentation correspond to possible breach paths. The watershed segmentation of the sample iso-sensing map is depicted in Fig. 4. The right and left sides are marked as boundary regions. These regions are not included in the analysis since they are not completely covered.

When simulating the immersion process, there are two approaches: (1) the basins are found, then watersheds are formed by taking a set complement; (2) the image is completely partitioned into basins, and then, by boundary detection, the watersheds are discovered. There are several sequential and parallel algorithms for watershed transformation [39]. They can be divided into two classes: those based on the recursive algorithm by Vincent and Soille [15], and others based on distance functions by Meyer [40]. In this paper, we use the Vincent–Soille algorithm, which has two main steps: sorting and flooding. After sort-
ing the gradient values of each pixel, the algorithm starts working with the pixels with the lowest gradient values and assigns a unique label to each minimum and its corresponding basin using a breach-first technique. If a pixel is adjacent to more than one basin it is marked as a watershed edge. The time complexity of the algorithm is linearly proportional to the number of pixels.

For a grid, the intensity of grid point \( p \) is denoted by \( I(p) \) that takes discrete values in \([0, N]\). The path \( P \) between grid point \( p \) and \( q \) is described with an \( l \)-tuple, \((p, x_1, x_2, \ldots, x_{l-2}, q)\) where \( p \in G_0(x_1), x_i \in G_0(x_{i-1}), x_{i-2} \in G_0(q), i = 1, 2, \ldots, l - 2, \) and \( l \) is the length of the path \( P \). A minimal plateau of \( l \) around point \( p \) denoted with \( M(l) \) is a connected plateau of grids from which it is not possible to reach a point with lower altitude (intensity) without having to climb. Then, the minimal plateau is defined with a set of points
\[
\forall q \in M(l) \text{ such that } I(q) \geq \text{slant } I(p) \quad \text{and} \quad \forall P = (p, x_1, \ldots, x_{l-2}, q) \text{ such that } I(x_i) \geq \text{slant } I(p) \quad \text{where} \quad i = 1, 2, \ldots, l-2.
\]
Then, the catchment basin \( C(M) \) associated with the minimal plateau \( M \) is the set of points around \( p \) with higher intensities that corresponds to the altitude to where a falling water drop flows until it reaches \( M \). The dividing lines that separate the catchment basins are the watershed contours [15].

In watershed segmentation, oversegmentation occurs when tiny and insignificant regional minima form their own catchment basins. One solution is to modify the image to remove the minima that are too shallow. This is a common problem in image processing when the gradient field is not smooth. In our problem, the iso-sensing surface is quite smooth. Because the sensors are identical and their positions are random, tiny and insignificant regional minima cannot occur and oversegmentation is not probable. However, for other iso-sensing definitions, oversegmentation may increase the algorithmic time complexity.

Undersegmentation occurs when a segment contains grid points from two regions that belong to two or more different sensors. Undersegmentation of the iso-sensing map implicitly suggests that there are other contours with smaller detection probabilities than the undersegmented. Therefore, some of the contours are not produced and those are not the ones we are after in this work. Therefore, undersegmentation, by its nature, does not impact the result and the performance of the model.

5. Deployment quality measure

The most secure path for a target follows the grid points that are the most distant from the sensors in the field. From the WSN’s point of view, this path is the weakest breach path. Thus, the maximum detection probability on the weakest breach path provides a measure to analyze the quality of deployment [1]. Watershed segmentation produces several contours in which the weakest breach path resides.

The watershed contours are the points in the iso-sensing map with the lowest detection probabilities, which is in favor of the target. Many combinations of watershed contours exist that connect the insecure side to the secure side. Among the alternative paths, the one with the least maximum detection probability measures the quality of deployment. To determine the DQM, one with the least maximum detection probability measures the quality of deployment. To determine the DQM, please cite this article in press as: E. Onur et al., Surveillance with wireless sensor networks in obstruction: Breach paths as watershed contours, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.09.006

![Fig. 5. Demonstration of Algorithm 1 to produce the auxiliary graph based on the example shown in Fig. 1a.](image-url)
an auxiliary graph is constructed using the labelled watersheds edges (as can be seen in Fig. 5). The objective of the labelling process is to assign a weight which shows the level of breach security. The watershed contours are denoted with nodes in the auxiliary graph and the nodes are assigned weights as the maximum detection probability of any grid point on the same watershed contour as shown in Fig. 5b.

Watershed segmentation algorithm produces a labelled image where all the watershed contours have the same label. To discriminate between the individual watershed contours and to label them, we propose Algorithms 1 and 2. Suppose that each grid point \((i, j)\) is marked with \(n(i, j) = 1\) if \((i, j)\) belongs to a watershed edge, and \(n(i, j) = 0\), otherwise. The degree of the grid point \((i, j)\) can be defined as \(d(i, j) = n(i+1, j) + n(i−1, j) + n(i, j+1) + n(i, j−1)\). The grid points which connect two or more watershed edges are referred to as the connection points and for a connection point \((i, j)\), \(n(i, j) > 1\) and \(d(i, j) > 2\) as shown in Fig. 5a. With a minor modification of the watershed algorithm, the connection points can be obtained easily.

Denoting the state of all of the grid points \(s(i, j) = \text{unknown}\), for each connection point \((i, j)\), if the state of an adjacent node \(s(i, m) = \text{unknown}\) and \(n(i, m) = 1\), mark the adjacent node \((i, m)\) with a unique label and set \(s(i, m) = \text{known}\) and record the maximum detection probability for this labelled edge and continue to apply the same algorithm to adjacent grid points of \((i, m)\) until it is another connection point or field borders are crossed.

Each labelled edge (watershed contour) is represented as a node; e.g., as in Fig. 5a. The nodes are connected in the auxiliary graph if the respective edges originate from the same connection point. A sample auxiliary graph for the iso-sensing surface in Fig. 2 is shown in Fig. 6a. The connection matrix of the auxiliary graph is represented with \(C\), and the vector \(W\) denotes the weights of the nodes that is the maximum detection probability of the respective edge in the iso-sensing map. The labels of each grid point are denoted with the vector \(L\). The start node \(s\) and the destination node \(d\) denote the insecure and secure sides, respectively. If any edge crosses the boundary regions on either side of the field, remove the nodes from the auxiliary graph. If any edge touches the insecure side, connect the representing node to the destination node, and if the edge touches the insecure side, connect the representing node to the start node. This algorithm constructs the auxiliary graph. Algorithms 1 and 2 have linear time complexities proportional to the number of points on the watershed contours.

In Fig. 6a, nodes 2 and 11 are disconnected from the graph since their corresponding edges cross the boundary regions. Nodes 1, 5, 8 and 10 are connected to the start node \(s\) since they cross through the border of the insecure side and nodes 12, 13 and 14 are connected to the destination node since they cross the border of the secure side.

The auxiliary graph shows the possible breach paths which a target may prefer. For example, in Fig. 6a among others, \((s, 1, 3, 14, d)\), \((s, 1, 3, 4, 14, d)\) or \((s, 8, 7, 6, 13, d)\) are some breach paths. The path with a set of nodes which has the minimum weights is the best from the view point of the target.

### Algorithm 1. ConstructAuxiliaryGraph

1: if \(\forall (i, j) \in \text{boundary}\) then
2: \(state(i, j) = \text{boundary}\)
3: else
4: \(state(i, j) = \text{unknown}\)
5: end if
6: \(c = 0\) /*labels and the nodes of the auxiliary graph*/
7: for all \((i, j)\) where \(n(i, j) = 1 \) and \(d(i, j) > 2\) and \(state(i, j) = \text{boundary}\) do
8: Disconnect \((i, j)\) from adjacent nodes
9: \(A = 0\) /*Create an empty label set*/
10: for all \((i, m)\) in \(G_c(i, d)\) do
11: if \(n(i, m) = 1\) then
12: if \(state(i, m) = \text{unknown}\) then
13: \(c = c + 1\) /*generate new edge label*/
14: end if
15: end if
16: if Return Value = true then
17: \(A = A + (c, m)\) /*Add the new label to label set*/
18: end if
19: end if
20: end for
21: if Return Value = true then
22: end if
23: end for
24: end for
25: end for

On the auxiliary graph, instead of determining the weakest breach path, a simpler approach, described in Algorithm 3, can be employed to determine the bottleneck.
edge through which the weakest breach path may pass. In this algorithm, the weights of the nodes are sorted, and the node with the largest weight is removed from the graph. If the start and destination nodes are disconnected, the weakest breach path must pass through this edge if there are no other edges with the same weight. If the graph is not discon-

Fig. 6. Trace of Algorithm 3. This is the auxiliary graph of the watershed contours in Fig. 4 for the iso-sensing surface in Fig. 3 produced using Algorithm 1.
nected, the same algorithm is applied on the residual graph until a disconnected residual graph is obtained. The weight of the final node that is removed produces the DQM. The weakest breach path cannot be determined with this approach because it may follow another edge with the same weight, whose removal does not disconnect the start and destination nodes.

In Fig. 6, the trace of Algorithm 3 is depicted for the auxiliary graph in Fig. 6a. The DQM algorithm removes the nodes with the largest weight from the auxiliary graph to obtain a residual graph where the start and destination nodes are disconnected. Hence, the algorithm builds a heap using the weights of the nodes, extracts the index of the node with the largest weight and removes the node, as well as the incoming and outgoing edges and runs the depth first search algorithm to determine if the residual graph is disconnected. In the first step (Fig. 6a), the node with the largest weight is node 4 and it is removed and the disconnected residual graph shown in Fig. 6b is obtained. In the second step, node 6 is removed because it is the node with the largest weight in the residual graph and the connected residual graph in Fig. 6c is obtained. Continuing like this, in the sixth step, node 7 is with the largest weight in the residual graph so that it is removed, and the residual graph becomes disconnected now. Consequently, the algorithm stops and declares that the DQM is 0.84 because it is the weight of the last removed node.

Algorithm 3: DeploymentQualityMeasure

\begin{algorithm}
\begin{algorithmic}
\State \text{buildHeap } H \text{ using } W_c$
\While{\text{not } H.\text{empty}()}
\State \text{h } = \text{H.extractMax()}$
\State \text{Remove node } h \text{ from auxiliary graph}$
\If{\text{s and d are disconnected}}$
\State \text{Return } h.W_c \text{ \text{"Depth first search can be used"}}$
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}

Table 1: Effect of the false alarm rate on the DQM when 90 sensors are deployed for MDIS.

<table>
<thead>
<tr>
<th>False alarm rate</th>
<th>DQM</th>
<th>Variance</th>
<th>95% Confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.050</td>
<td>0.93</td>
<td>0.020</td>
<td>(0.90–0.96)</td>
</tr>
<tr>
<td>0.075</td>
<td>0.96</td>
<td>0.015</td>
<td>(0.94–0.98)</td>
</tr>
<tr>
<td>0.100</td>
<td>0.97</td>
<td>0.012</td>
<td>(0.96–0.99)</td>
</tr>
<tr>
<td>0.125</td>
<td>0.98</td>
<td>0.010</td>
<td>(0.97–0.99)</td>
</tr>
</tbody>
</table>

fewer than 20 sensors, some parts of the field can be assumed to be monitored with zero probability since the associated detection probabilities are lower than the reliability threshold. Indeed, the DQM is zero for $\alpha = 0.05$ for 20 sensors.

When RIS is bounded with the coverage degree parameter $K$, the results do not change significantly as can be seen in Fig. 10. Because, to increase the detection probability at each grid point, dense deployment is required. Considering the spatial distribution of the sensors, to attain at least the threshold probability level, the required number of sensors provide a coverage degree greater than three. In other words, deploying more sensors to increase the detection probability at each grid point automatically increases the coverage degree and vice versa. For RIS and KRIS, when the required DQM value is achieved, it means that the deployment is saturated, and additional sensors will not improve performance. Thus, for RIS and KRIS, it is more appropriate to consider saturated and unsaturated deployment scenarios instead of sparse, moderate and dense deployments.
Fig. 12. Effect of the sensor count on the detection ratio of a target following the watershed contours for several target speeds (g/s denotes the velocity in terms of grids per second).

For some applications such as battlefield surveillance, higher accuracy may be required. For those applications, KIS with $K > 1$ can be employed. The effect of coverage degree on the deployment quality measure is analyzed in Fig. 11 where Elffe’s detectors (see Eq. (2)) are utilized. The sensing range and the decay of the detectors are arranged according to parameters $d^1_L = 10$, $d^2_L = 30$, $\alpha = 0.1$ and $\beta = 0.9$. The same DQM level can be provided with fewer sensors for larger coverage degree requirement.

The iso-sensing map is determined according to MDIS after reaching the insecure side (see Fig. 1a). From the target’s standpoint, following the watershed contours for several target speeds (g/s denotes the velocity of the target in terms of grids per second). As the target speed increases, the sensor coverage probability increases. For example, to attain a DQM of 0.9 for KIS with $K = 1$, $K = 2$ and $K = 3$, the required number of sensors are 100, 80 and 60, respectively.

The simulation is repeated 2000 times per deployment and the results are the averages of 100 distinct deployments. Three circular obstacles are placed where the radii are uniformly randomly chosen between 8 m and 15 m.

The detection ratio of targets with several constant velocities (denoted in grids per second, g/s) for varying sensor counts is depicted in Fig. 12. If the target remains in the field for a longer time period, then its detection probability will be larger. For example, when nine sensors are deployed, for a target with a velocity of 5 g/s, the detection ratio is around 0.68, whereas for a velocity of 1 g/s, the same ratio rises to approximately 0.95. Consequently, the target should pass far from the sensors rapidly.

7. Conclusion

In this paper, we propose the utilization of the watershed segmentation algorithm to find the possible breach paths in a surveillance field with obstacles. In order to apply the watershed segmentation, the iso-sensing map is defined and a recursive algorithm is designed to find the deployment quality measure defined as the maximum detection probability on the weakest breach path. The simulations indicate the impact of the false alarm rate and the sensor count on the deployment quality measure.

The wireless sensor network application designer may merely use MDIS (KIS with $K = 1$) if the required security level is not extremely high. KIS and KRIS are beneficial when large number of sensors are to be deployed and a sleep scheduling algorithm is to be implemented to prolong the lifetime of the network. As a future work, we plan to generalize the obstacle models by utilizing a 3-D topography and to analyze the effect of grid size on the sensitivity of the deployment quality.

Acknowledgement

This work is supported by the State Planning Organization of Turkey under the Grant Number 03K120250, and by TUBITAK under the Grant Number 106E082. Parts of this work are presented in preliminary form in the IEEE International Conference on Communications, Istanbul, Turkey, June 2006.

References


Ertan Delic received the B.S. degree in Computer Engineering from Ege University, Izmir, Turkey in 1997, and the M.S. and Ph.D. degrees in Computer Engineering from Bogazici University, Istanbul, Turkey in 2001 and 2007, respectively. He is a BALTIC graduate. After the B.S. degree, he worked for LMS Durability Technologies GmbH, Kaiserslautern, Germany and Global Bilgi, Istanbul, Turkey. During the Ph.D degree, he worked as a R&D project manager at Argela Technologies, Istanbul. Presently, he is a postdoctoral research at Technical University of Delft, Netherlands. His research interests are in the area of telecommunications, personal networks, wireless and sensor networks. He is a member of IEEE.

Cem Ersoy received his B.S. and M.S. degrees in Electrical Engineering from Bogaziçi University, Istanbul, in 1984 and 1986, respectively. He worked as an R&D engineer in NETAS A.S. between 1984 and 1986. He received his Ph.D. in Electrical Engineering from Polytechnic University, Brooklyn, New York in 1992. Currently, he is a professor in the Computer Engineering Department of Bogaziçi University. His research interests include performance evaluation and topological design of communication networks, wireless networks and mobile applications. He is a senior member of IEEE.

Hakan Delic received the B.S. degree (with honors) in Electrical and Electronics Engineering from Bogaziçi University, Istanbul, Turkey, in 1988, and the M.S. and the Ph.D. degrees in Electrical Engineering from the University of Virginia, Charlottesville, in 1990 and 1992, respectively. He was a Research Associate with the University of Virginia Health Sciences Center from 1992 to 1994. In September 1994, he joined the University of Louisiana at Lafayette, where he was on the Faculty of the Department of Electrical and Computer Engineering until February 1996. He was a Visiting Associate Professor in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, during the 2001–2002 academic year. He is currently a professor of Electrical and Electronics Engineering at Bogaziçi University. His research interests are in the area of telecommunications, personal networks, wireless and sensor networks. He is a member of IEEE.
interests lie in the areas of communications and signal processing, with current focus on ultra-wideband communications, iterative decoding, robust systems, and sensor networks. He frequently serves as a consultant to the telecommunications industry. He is a senior member of IEEE.

Lale Akarun received the B.S. and M.S. degrees in Electrical Engineering from Bogaziçi University, Istanbul, in 1984 and 1986, respectively. She obtained her Ph.D. from Polytechnic University, New York in 1992. Since 1993, she has been working as a faculty member at Bogaziçi University. She became a professor of Computer Engineering in 2001. Her research areas are face recognition, modeling and animation and human activity and gesture analysis. She has worked on the organization committees of IEEE NSIP99, EUSIPCO 2005, and eNTER-FACE2007. She is a senior member of the IEEE.