



**HAL**  
open science

# A Simple Method for the Deployment of Wireless Sensors to Ensure Full Coverage of an Irregular Area with Obstacles

Ines Khoufi, Pascale Minet, Anis Laouiti, Erwan Livolant

► **To cite this version:**

Ines Khoufi, Pascale Minet, Anis Laouiti, Erwan Livolant. A Simple Method for the Deployment of Wireless Sensors to Ensure Full Coverage of an Irregular Area with Obstacles. MSWIM: International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Sep 2014, Montreal, Canada. 10.1145/2641798.2641805 . hal-01088700

**HAL Id: hal-01088700**

**<https://inria.hal.science/hal-01088700>**

Submitted on 28 Nov 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Simple Method for the Deployment of Wireless Sensors to Ensure Full Coverage of an Irregular Area with Obstacles\*

Ines Khoufi  
Inria Rocquencourt  
78153 Le Chesnay Cedex  
France  
ines.khoufi@inria.fr

Pascale Minet  
Inria Rocquencourt  
78153 Le Chesnay Cedex  
France  
pascale.minet@inria.fr

Anis Laouiti  
TELECOM SudParis  
CNRS Samovar UMR 5157  
91011 Evry Cedex, France  
anis.laouiti@telecom-  
sudparis.eu

Erwan Livolant  
Inria Rocquencourt  
78153 Le Chesnay Cedex  
France  
erwan.livolant@inria.fr

## ABSTRACT

In this paper, we focus on the deployment of wireless sensor nodes in an arbitrary realistic area with an irregular shape, and with the presence of obstacles that may be opaque. Moreover, we propose a simple projection-based method that tends to minimize the number of sensor nodes needed to fully cover such an area. This method starts with the optimal uniform deployment based on the triangular tessellation encompassing the whole area. Then, it projects some external sensor nodes on the border to ensure full coverage and connectivity. We show that this method outperforms the contour-based one using various types of irregular areas.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

## General Terms

Algorithms, Design, Performance

## Keywords

Wireless Sensor Network; Deployment algorithm; Full coverage; Area with irregular shape; Opaque obstacle; Hidden zone

\*This work has been partly funded by the Cluster Connexion.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*MSWiM '14*, September 21–26, 2014, Montreal, QC, Canada.  
Copyright 2014 ACM 978-1-4503-3030-5/14/09 ...\$15.00.  
<http://dx.doi.org/10.1145/2641798.2641805>.

## 1. MOTIVATION

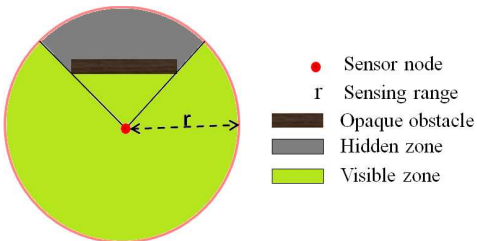
We are witnessing the deployment of many wireless sensor networks in various application domains such as pollution detection in the environment, intruder detection at home, preventive maintenance in industrial process, monitoring of a temporary industrial worksite, damage assessment after a disaster, etc. Many of these applications require the full coverage of the area considered. With the full coverage of the area, any event occurring in this area is detected by at least one sensor node. In addition, the connectivity ensures that this event is reported to the sink in charge of analyzing the data gathered from the sensors and acting according to these data.

Depending on the application targeted, and for optimality reasons, we may require a uniform deployment to fully cover such an area with a minimum number of wireless sensors. Although optimal uniform wireless sensor network deployment is a key factor for minimizing the overall cost of a wireless sensor network, it may not be accomplished in many cases. In fact the shape of the area to cover, the possible presence of obstacles within this area as well as their properties (shape, opaque or transparent to sensing range, etc) may prevent the deployment of a uniform wireless sensor network. These constraints must be considered when designing a deployment algorithm in order to reach an acceptable deployment.

Examples of applications where wireless sensor networks have proved their efficiency include monitoring of a temporary industrial worksite, damage assessment after a disaster, precision farming, intruder detection in a warehouse, health monitoring of a building. In all these cases, 2D coverage is sufficient to meet the applications requirements. Furthermore, the area to cover has an irregular shape with many edges and is not necessarily convex and may include several obstacles.

In the literature, many studies assume that this area is rectangular and adopt the classical deployment which is based on the triangular lattice that has been proved optimal [1]. In real life, things are more complex. Moreover, few papers take obstacles into account. Those that do as-

sume that obstacles are constituted by a juxtaposition of rectangles that seems an unrealistic assumption. In real deployments, the shape of obstacles may be irregular. Like in [2], we distinguish two types of obstacles: the transparent ones like ponds in outdoor environment, or tables in an indoor site that only prevent the location of sensor nodes inside them; whereas the opaque obstacles like walls or trees prevent the sensing by causing the existence of hidden zones behind them: such zones may remain uncovered, as illustrated in Figure 1. Authors in [3] propose a solution to provide an  $(1 - \epsilon)$  coverage of an area of arbitrary shape with obstacles. This solution selects some landmarks in the area and covers them. However, the boundary and the obstacles are assumed to be transparent.



**Figure 1: Hidden zone caused by an opaque obstacle.**

Opaque obstacles are much more complex to handle than transparent ones and require the deployment of additional sensors to eliminate coverage holes. That is why in this paper, we focus on the deployment of wireless sensor nodes in an irregular area with obstacles that may be opaque and propose a projection-based method that tends to minimize the number of sensor nodes needed to fully cover this area. This number is smaller than the one given by the contour-based methods [4] and [5]. Our method is characterized by its simplicity. In contrast to Delaunay triangulation [6], coverage holes are intrinsically known by our method.

The method presented here computes the deployment that is given as an input to a mobile robot. This robot is in charge of placing the sensor nodes (unable to move) at the right location in the area to fully cover. The goal is to provide a deployment of a wireless sensor network assisted by a mobile robot. An additional step can be done after the deployment to collect measures on site. These measures are used as a feedback to adapt the real deployment on the given area. Notice however that in our solution all sensor nodes are needed to provide full coverage, unlike [7] where some redundant sensor nodes can be switched off.

The remaining of this paper is organized as follows. We define the problem in Section 2. In Section 3, we give a brief state of the art, recalling known results about full coverage complexity and optimal deployment. We distinguish two approaches to deal with irregular borders and obstacles: contour-based and Delaunay-Triangulation-based. We establish general bounds on the number of sensor nodes independently from the method used. We then propose our method to fully cover an area with irregular transparent borders but no obstacles in Section 5. In Section 6, we show how to take obstacles into account. In Section 7, we generalize to borders and obstacles that can be opaque. In Sections 5 to 7, we present the main principles with an illustrative example, provide a bound dependent of our method and compare the

performances of our contribution with those of a contour-based method. Finally, we conclude in Section 8.

## 2. PROBLEM STATEMENT

In this paper, we consider wireless sensors that must be deployed to fully cover a given 2D-area of irregular shape with the presence of several obstacles.

### 2.1 Goal

Our goal is to minimize the number of sensors needed to achieve the full coverage of the area given, denoted  $\mathcal{A}$ , while meeting the assumptions listed in Subsection 2.2. The full coverage of  $\mathcal{A}$  means that any event occurring in  $\mathcal{A}$  is detected by at least one sensor node. The deployment of wireless sensor nodes is computed by a single entity that takes as inputs the vertices of the polygon defining  $\mathcal{A}$  the area to cover as well as for each obstacle, the vertices of its polygon.

### 2.2 Models

For that purpose, we adopt the following models:

- The wireless sensors are assumed to have the same sensing range denoted  $r$  and the same radio range  $R$ . The sensing model adopted is the simplest one: the disk of radius  $r$ . Similarly, the radio transmission model is also disk-based: any wireless node located in the disk of radius  $R$  centered at the sensor considered has a symmetric radio link with it. For the sake of simplicity, we also assume that the condition  $R \geq \sqrt{3}r$  is met. This condition guarantees that any deployment of wireless sensor nodes ensuring full coverage also ensures full connectivity.

- $\mathcal{A}$ , the area to fully cover, is considered as a polygon which may be not convex. This polygon is defined by its edges. These edges constitute the borders of the area. We distinguish two types of borders:

- Transparent border: a transparent border does not prevent the sensing of sensor nodes. The only constraint added by transparent borders is that no sensor must be outside the area to cover.
- Opaque border: an opaque border prevents the sensing of nodes located behind the border: a sensor node  $s$  can cover a point  $u$  within its sensing disk if and only if  $u$  is in the line of sight of  $s$  (see Figure 1). Hence, such a border modifies the *is covered* relation, which is of prime importance in the problem we want to solve.

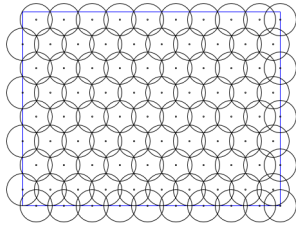
- The area considered usually has obstacles. No sensor node must be located within an obstacle. Like for borders, we distinguish two types of obstacles: transparent and opaque. An obstacle is defined by the edges of its polygon that may be of irregular shape and not convex. Let  $\mathcal{O}$  denote the set of obstacles.

## 3. STATE OF THE ART

Coverage is a basic issue in wireless sensor networks. The reader can refer to [8] for a survey on the various problems related to coverage. The pioneering work on the full coverage of an infinite 2D area was published by Kershner [1]. He proved that the triangular tessellation achieves a full coverage with an asymptotic minimum number of sensors. In this tessellation, each sensor node at the center of an hexagon has

six neighbors at a distance of  $r\sqrt{3}$  that occupy the vertices of this hexagon as depicted in Figure 2 and Figure 4. The rectangular area is covered by several lines of sensor nodes. We notice that the first sensor of an odd line is located at a distance  $r\sqrt{3}/2$  from the left border, whereas the first sensor of an even line is located at the left border. Furthermore, the first line starts at a distance  $r/2$  from the top border, and the interline is equal to  $3r/2$ .

This result has been applied in many network deployments: radio cellular networks as well as wireless sensor networks. It has been extended in [9] to find the optimal tessellation (e.g. rhombus, square, hexagon, triangle) for different conditions binding the radio range  $R$  and the sensing range  $r$ . Authors of [2] show how to take obstacles into account. They also show that the problem of full coverage of a 2D-area with opaque obstacles is NP-hard.



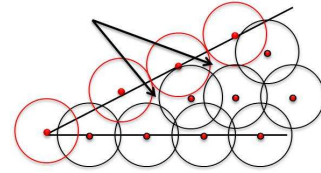
**Figure 2: Optimal deployment based on a triangular tessellation.**

To cope with this problem, the large majority of approaches encountered in the literature adopt the optimal deployment based on the triangular tessellation as a starting point. Then, sensors nodes located within an obstacle or outside the border of the area to fully cover are eliminated. This elimination usually causes coverage holes. The existing approaches differ in the way they heal the coverage holes. We distinguish the following two approaches:

- The contour-based approaches like [4, 5]: they deploy sensor nodes at a constant distance along the border of the area and along the border of each obstacle in order to heal coverage hole occurring on these contours. The distance between two successive sensor nodes deployed successively on a given contour is computed from the sensing range. Such approaches are simple but may require a high number of sensors in case of many irregular borders as shown in [6]. In contrast to our projection-based method, the coverage holes that are not adjacent to the area border or the obstacle border are not detected as shown in Figure 3.

- The Delaunay-triangulation-based approaches like [6]: they use the Delaunay triangulation to detect coverage holes and then place sensor nodes at some vertices of the triangles defined using a vertex coloring technique. However the complexity may be high, due to the presence of two modules: (a) the determination of coverage holes followed by (b) the computation of sensor locations that may be greedy in computation resources. In contrast to this approach, our method determines the sensor location without searching coverage holes. To reduce the number of sensors, our method eliminates redundant sensor nodes.

That is why we propose a simple solution requiring less computational complexity. We present in Sections 5, 6 and 7 three problems of increasing complexity. We solve these problems using our simple and efficient approach. Before, we



**Figure 3: Coverage hole that is undetected by a contour-based method.**

establish general bounds on the number of sensors required to fully cover the area.

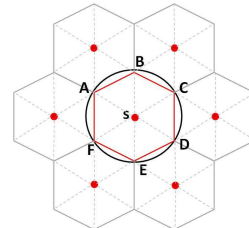
## 4. GENERAL BOUNDS ON THE NUMBER OF SENSOR NODES

In this section, we establish upper and lower bounds on the number of sensor nodes required, independently from the method used to fully cover the given area. These bounds are derived from the optimal deployment based on a triangular tessellation. We will see in the next sections whether these bounds can be improved taking into account the method used to provide the full coverage of  $\mathcal{A}$ .

### 4.1 Lower bound

We first establish a lower bound on the number of sensor nodes required to fully cover the area considered. This bound is deduced from the fraction between the surface to cover and the surface covered by a sensor node. For that purpose, we focus on Figure 4 depicting the optimal deployment based on a triangular tessellation. We notice that in this optimal deployment, we can make the following approximation: any sensor node  $s$  that is not adjacent to a border can be considered as the only sensor node covering the hexagon  $ABCDEF$  of edge  $r$  centered at itself and depicted in red in this figure. We can compute the surface of the triangle  $ABS$ . It is equal to  $r^2\sqrt{3}/4$ . The surface of the hexagon, denoted  $S_{\mathcal{H}}$ , is equal to the surface of the six triangles composing it. We have  $S_{\mathcal{H}} = 3\sqrt{3}r^2/2$ . With this result, we can give a lower bound on the number of sensor nodes needed to fully cover the area  $\mathcal{A}$  without obstacles. Let  $Min_N$  be this bound. Let  $S_{\mathcal{O}}$  be the cumulative surface of the obstacles. The surface to cover is then equal to  $S_{\mathcal{A}} - S_{\mathcal{O}}$ . We get  $Min_N = \lceil \frac{S_{\mathcal{A}} - S_{\mathcal{O}}}{S_{\mathcal{H}}} \rceil$ . Finally, we have:

$$Min_N = \lceil \frac{2(S_{\mathcal{A}} - S_{\mathcal{O}})}{3\sqrt{3}r^2} \rceil. \quad (1)$$



**Figure 4: Hexagon covered by the sensor node  $s$ .**

## 4.2 Upper bound

We can now establish an upper bound  $Max_N$  on the number of sensor nodes needed to fully cover  $\mathcal{A}$  in the presence of transparent obstacles. This bound is based on:

- $N_{In}$  the number of sensor nodes within  $\mathcal{A}$  but outside the obstacles  $\mathcal{O}$ . This number can be obtained by counting the number of sensor nodes meeting this condition in the optimal deployment based on a triangular tessellation.
- $N_{Add}$  the number of additional sensor nodes to fully cover  $\mathcal{A}$  added after the elimination of sensor nodes that are either outside  $\mathcal{A}$  or within an obstacle. To compute this number, we consider the contour of the area and the obstacles, segment by segment.

To fully cover a segment of length  $L$ , we use the principle of the optimal deployment in a rectangular area given in Section 3 and adapted to a segment as shown in Figure 5. Hence, we need  $N_b$  sensor nodes, with:

$$N_b = \lfloor \frac{L - \frac{r\sqrt{3}}{2}}{r\sqrt{3}} \rfloor + 1 + \delta_b \quad (2)$$

$$\text{with } \delta_b = \begin{cases} 1 & \text{if } L - r\sqrt{3} - \lfloor \frac{L - \frac{r\sqrt{3}}{2}}{r\sqrt{3}} \rfloor r\sqrt{3} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$\delta_b$  is equal to 1 if the distance between the projection of the last sensor on the border and the extremity of the border is higher than  $\frac{r\sqrt{3}}{2}$ .

We now apply this method to all edges of  $\mathcal{A}$  and the obstacles  $\mathcal{O}$ . We then get  $N_{Add} \leq \sum_{b \in \text{edge}(\mathcal{A} \cup \mathcal{O})} N_b$ . Since  $Max_N = N_{In} + N_{Add}$ , we get:

$$Max_N = N_{In} + \sum_{b \in \text{edge}(\mathcal{A} \cup \mathcal{O})} N_b. \quad (3)$$

We now show that in case of transparent borders and obstacles, this number  $Max_N$  is sufficient to eliminate all coverage holes resulting from the elimination of sensor nodes outside  $\mathcal{A}$  or within an obstacle. Let us consider any edge of  $\mathcal{A}$  such that a sensor node  $s$  outside the area (or inside an obstacle) at a distance  $\varepsilon > 0$  from this edge has been eliminated. This elimination can create a coverage hole in the stripe of width  $r$  parallel to this edge. If now we consider the first line of the optimal deployment that starts at a distance  $r/2$  from the border, we notice that this line fully covers the stripe of width  $r$  parallel to this border, as illustrated in Figure 5. Hence, we propose to deploy additional sensor nodes on a line parallel to the border, at a distance  $r/2$  from the border. Notice that if such a sensor is outside  $\mathcal{A}$ , it is put at the first location inside  $\mathcal{A}$  according to a projection along the border.

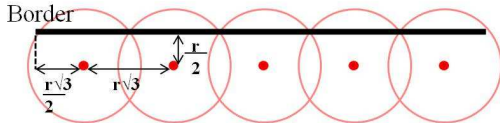


Figure 5: Optimal deployment along a border.

We can apply the same principle to the obstacles, considering the sensor nodes within the obstacles that have been eliminated.

## 5. OPTIMIZED DEPLOYMENT IN AN IRREGULAR AREA

In this section, we propose a deployment algorithm to cope with the irregular shape of an area.

### 5.1 Problem definition

In this first coverage problem, we consider any irregular 2-D area and assume that there is no obstacle and the border of the area is transparent.

### 5.2 Principle

Our method proceeds as follows:

1. We start with the optimal deployment in the rectangle circumscribing the given area  $\mathcal{A}$ : see Figure 7(a).
2. Sensor nodes that are outside  $\mathcal{A}$  are eliminated, which may cause coverage holes: see Figure 7(b).
3. For each sensor node  $s$  located outside the area at a distance strictly less than  $r$  from a border, we check whether the border segment initially covered by  $s$  is still covered by other sensor nodes within  $\mathcal{A}$ , even if  $s$  is eliminated. Otherwise  $s$  is orthogonally projected on the border: see Figure 7(c). Due to this projection technique, illustrated in Figure 6, we can guarantee that the zone initially covered by the eliminated sensor node  $s$ , still covered after the projection of  $s$ .
4. Finally, to optimize the number of sensor nodes needed, we check if some of them are providing redundant coverage, which can be eliminated in that case. They can be eliminated if and only if the intersection of  $\mathcal{A}$  and the zone they covered is fully covered by other sensor nodes that are kept (see Figure 7(d)).

Notice that the projection of a sensor node is not always on the border considered as shown in Figure 6(b). In this case, the position of the projected node is shifted in the middle of the border segment covered by this node in order to heal coverage holes.

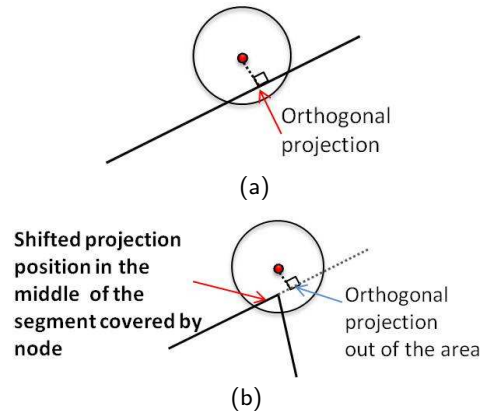


Figure 6: Projection technique.

### 5.3 Upper bound on the number of sensors required

We now establish  $OurMax_N$ , an upper bound on the number of sensors needed by our method to fully cover an

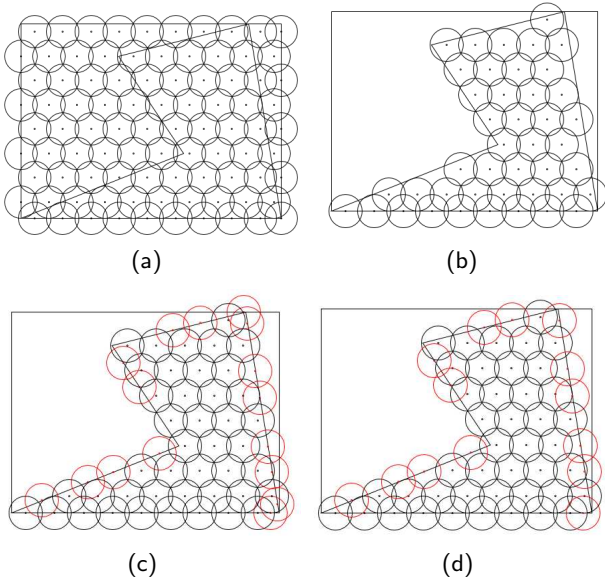


Figure 7: Principles of our method.

area with irregular shape but with transparent borders. This bound does not take into account the elimination of redundant sensor nodes done in step 4. Let  $Out_r$  denote the set of sensor nodes outside  $\mathcal{A}$  at a distance less than  $r$  from a border and  $N_{Out_r}$  denote the cardinality of  $Out_r$ . We recall that  $N_{In}$  denotes the number of sensor nodes within  $\mathcal{A}$ . Since the sensor nodes that are added come from the projections of nodes in  $Out_r$  and these projections are done only on edges whose distance is less than  $r$ , we then have our upper bound:

$$OurMax_N = N_{In} + \sum_{P \in Out_r} \sum_{e \in edge(\mathcal{A})} 1_{distance(P,e) < r} \quad (4)$$

with  $1_{distance(P,e) < r} = 1$  if  $distance(P,e) < r$  and 0 otherwise.

## 5.4 Comparative evaluation

In this subsection, we compare the performances of our method (projection-based) with the contour-based method on the boot configuration depicted in Figure 7(d). This boot configuration has a circumscribing rectangle of size  $20r \times 18r$ . In Sections 5 and 7 we study more complex configurations.

The contour-based method chosen applies the method explained in Subsection 4.2 to each edge of  $\mathcal{A}$ . Hence, it needs  $N_b$  sensor nodes to fully cover an edge of length  $L$  (see Equation 2).

We make vary the sensing range (i.e.  $r, r/2, r/4$ ), whereas the dimensions of the area are kept constant. We study the impact on the number of sensor nodes required to fully cover the area. As expected, when the sensing range decreases from  $r$  to  $r/2$ , the number of sensor nodes increases from 57 to 184. The lower bound  $Min_N$  suggests that this number should be multiplied by 4. We observe a multiplication by 3.23. Respectively, when the sensing range decreases from  $r$  to  $r/4$ , the number of sensor nodes increases from 57 to 652. Similarly, the lower bound  $Min_N$  suggests that this number should be multiplied by 16. We observe a multiplication by 11.46. This can be explained by the irregularity of the

border. The contour-based method adds 35 nodes on the border whereas our method adds only 14 nodes. Hence, the contour-based method leads to a total number of sensor nodes of 78 which is higher than this of our method (i.e. 57 nodes).

We observe that the upper bound  $OurMax_N$  is very close to the number of sensor nodes needed by our method: we obtain an upper bound of 60, 188 and 656 whereas the exact number is 57, 184 and 652. Meanwhile, we notice that the gap between the real number of sensor nodes and the number given by the contour-based method drastically increases when the sensing range decreases. It reaches 21, 39 and 73 sensor nodes when the sensing range is equal to  $r, r/2$  and  $r/4$ . This would increase the deployment cost by a factor of 36% for a sensing range  $r$  for instance.

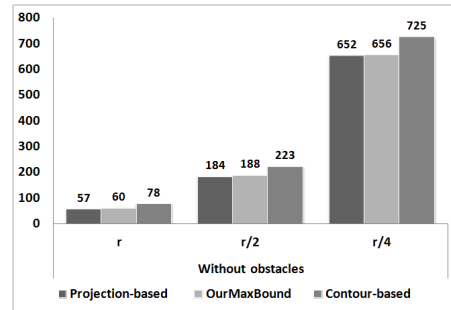


Figure 8: Impact of the sensing range on the number of sensor nodes (Boot configuration without obstacles).

## 6. OPTIMIZED DEPLOYMENT IN AN IRREGULAR AREA WITH OBSTACLES

Because of the presence of obstacles, sensor nodes of the optimal deployment that are located within an obstacle must be eliminated. They can cause coverage holes that must be healed.

### 6.1 Problem definition

In this second coverage problem, we consider any irregular 2-D area that includes obstacles and assume that both the obstacles and the border of the area are transparent.

### 6.2 Principle

We proceed as previously except that:

1. We start with the optimal deployment in the rectangle circumscribing the given area  $\mathcal{A}$ : see Figure 13(a).
2. Sensor nodes that are outside  $\mathcal{A}$  or inside the obstacles  $\mathcal{O}$  are eliminated, which may cause coverage holes: see Figure 13(b).
3. For each sensor node  $s$  outside the area at a distance strictly less than  $r$  from a border of the area, we check that the border segment initially covered by  $s$  is still covered by sensor nodes within  $\mathcal{A}$ , even if  $s$  is eliminated. Otherwise  $s$  is orthogonally projected on the border. We proceed similarly with any sensor node  $s$  inside an obstacle at a distance strictly less than  $r$  from a border of the obstacle: see Figure 13(c).
4. As a last step, redundant sensor nodes are eliminated.

### 6.3 Upper bound on the number of sensors required

We now extend our previous bound on the maximum number of sensor nodes needed by our method in the presence of obstacles. To deal with obstacles, our method projects nodes within an obstacle at a distance less than  $r$  from an edge of the obstacle. That is why, we add a third term to account for obstacles as follows:

$$OurMax_N = N_{In} + \sum_{P \in Out} \sum_{e \in edge(A)} 1_{distance(P,e) < r} + \sum_{P \in InObstr} \sum_{e \in edge(O)} 1_{distance(P,e) < r} \quad (5)$$

where  $InObstr$  denotes the set of sensor nodes within an obstacle at a distance strictly less than  $r$  from a border of an obstacle and  $1_{distance(P,e) < r} = 1$  if  $distance(P,e) < r$  and 0 otherwise.

### 6.4 Comparative evaluation

We consider different configurations to compare our method with the contour-based method described in Section 4. The configurations are various:

- The boot configuration with obstacles, (see Figure 9(a)) with the circumscribing rectangle of size  $20r \times 18r$ . This configuration is the simplest one we study.
- The star configuration. This configuration is representative of a complex shape of area with many salient angles. Its circumscribing rectangle is of size  $24r \times 28r$ . See Figure 9(b).
- The warehouse configuration, see Figure 9(c), with the circumscribing rectangle of size  $28r \times 18r$ . This configuration is representative of an indoor area with several rooms and many obstacles.

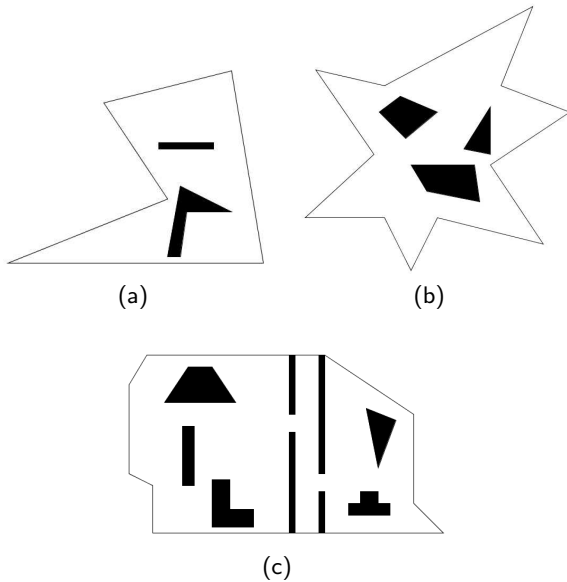


Figure 9: Configurations studied (Boot, Star, Warehouse).

In the boot configuration, our method needs only 59 sensor nodes, 2 among them are used to cope with obstacles (see Figure 10). The contour-based method needs to deploy 93 sensor nodes. Among these sensor nodes, 17 are due to the presence of obstacles. The overestimation of the contour-based method contributes to increase the deployment cost by a factor of 58%. Furthermore, we notice that this overestimation increases with the number of vertices of the area to cover and the obstacles to avoid.

When we vary the sensing range from  $r$  to  $r/2$  and  $r/4$ , the number of sensor nodes increases drastically as without obstacles. Our method outperforms the contour-based method. For instance, with  $r/4$  the contour-based method requires 737 sensor nodes instead of 645 for our method. This would require the deployment of 92 sensor nodes that could be spared. In addition, the upper bound  $OurMax_N$  is still very accurate, even in the presence of obstacles. It indicates 62, 197 and 652 instead of the real values 59, 191 and 645 provided by our method.

We obtain similar results with the star configuration: the benefit of our method reaches 40% as depicted in Figure 11. With the more complex configuration of the warehouse, the benefit increases up to 76% as depicted in Figure 12. This can be explained by the fact that this configuration includes several walls inside the area and many obstacles.

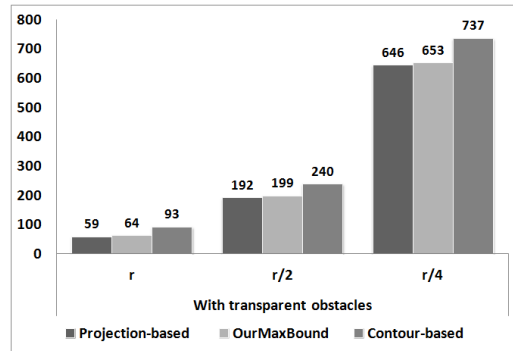


Figure 10: Impact of the sensing range on the number of sensor nodes (Boot configuration).

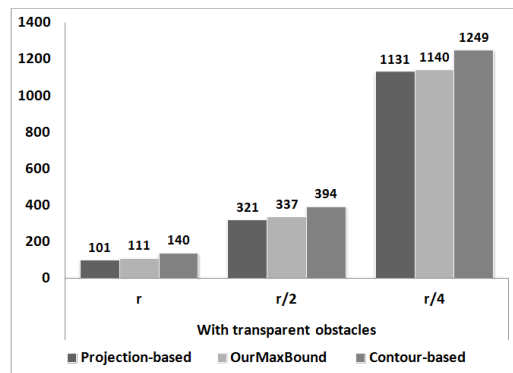


Figure 11: Impact of the sensing range on the number of sensor nodes (Star configuration).

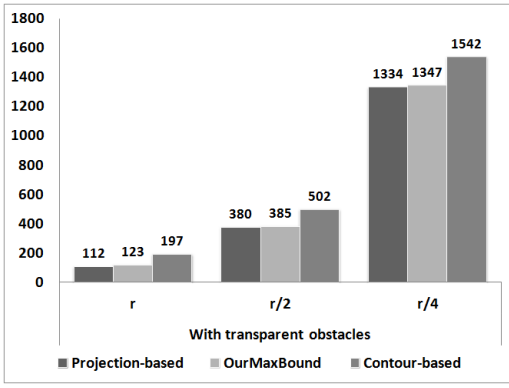


Figure 12: Impact of the sensing range on the number of sensor nodes (Warehouse Configuration).

## 7. HIDDEN ZONE

In this section, we propose a deployment algorithm to cope with the hidden zone due to opaque obstacles.

### 7.1 Problem definition

In this third coverage problem, we consider any irregular 2-D area that includes obstacles and assume that some obstacles and/or some borders of the area are opaque. This may result in hidden zones (see Figure 1). Sensor nodes must be added to cope with them.

### 7.2 Principle

In the presence of opaque borders or opaque obstacles, our method checks whether a hidden zone (see step 4 hereafter) exists. If so, sensor nodes are added. More precisely, the method proceeds according to the following steps:

1. We start with the optimal deployment in the rectangle circumscribing the given area  $\mathcal{A}$ : see Figure 13(a).
2. Sensor nodes that are outside  $\mathcal{A}$  or inside the obstacles  $\mathcal{O}$  are eliminated, which may cause coverage holes: see Figure 13(b).
3. For each sensor node  $s$  outside the area at a distance strictly less than  $r$  from a border of the area, we check that the border segment initially covered by  $s$  is still covered by sensor nodes within  $\mathcal{A}$ , even if  $s$  is eliminated. Otherwise  $s$  is orthogonally projected on the border. We proceed similarly with any sensor node  $s$  inside an obstacle at a distance strictly less than  $r$  from a border of the obstacle: see Figure 13(c).
4. For each sensor node  $s$  remaining after step 2, we check whether it is the only sensor node covering a zone in  $\mathcal{A} \setminus \mathcal{O}$  that becomes hidden because of the opacity of a border or an obstacle. If so, a new sensor node is added as the projection of  $s$  in the zone it should cover (see Figure 13(d)).
5. Finally, redundant sensor nodes are eliminated.

### 7.3 Upper bound on the number of sensors required

We now extend our previous bound on the maximum number of sensor nodes needed by our method in the presence of

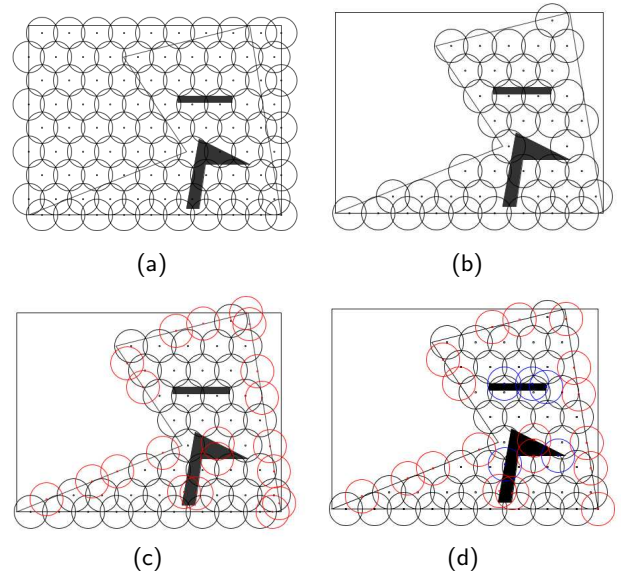


Figure 13: Principles of our method.

opaque obstacles or opaque borders. We add a fourth term to deal with opaque borders and opaque obstacles.

$$\begin{aligned}
 OurMax_N = N_{In} + \sum_{P \in Out_r} \sum_{e \in edge(\mathcal{A})} 1_{distance(P,e) < r} + \\
 \sum_{P \in InObstr} \sum_{e \in edge(\mathcal{O})} 1_{distance(P,e) < r} + \\
 \sum_{P \in In} \sum_{e \in Opaque_{edge}(\mathcal{A} \cup \mathcal{O})} 1_{distance(P,e) < r}, \quad (6)
 \end{aligned}$$

where  $In$  denotes the set of sensor nodes that remain after the elimination of step 2 and  $1_{distance(P,e) < r} = 1$  if  $distance(P,e) < r$  and 0 otherwise.

### 7.4 Comparative evaluation

We consider again the configurations defined in the previous section, but now the obstacles are opaque.

The contour-based method needs to deploy 93 sensor nodes in the boot configuration, 140 sensor nodes in the star configuration and 197 in the warehouse configuration. Notice that the contour-based method does not distinguish between opaque and transparent obstacles.

In the boot configuration, our method needs only 64 sensor nodes, 5 among them are used just to avoid hidden zones. These sensor nodes are depicted in blue in Figure 13(d). Our method outperforms the contour-based method by 48%.

In the star configuration, our method needs only 105 sensor nodes, 4 among them are used to avoid hidden zones. Our method saves 33% of the deployment cost compared to the contour-based method.

In the warehouse configuration, our method needs 134 sensor nodes, 22 among them are added to avoid hidden zones. Our method saves 47% of the deployment cost compared to the contour-based method.

When we vary the sensing range from  $r$  to  $r/2$  and  $r/4$ , we still observe that our method outperforms the contour-based method as depicted in Figures 14, 15 and 16. The bound



$OurMax_N$  always provides a very good approximation of the number of sensors required by our method.

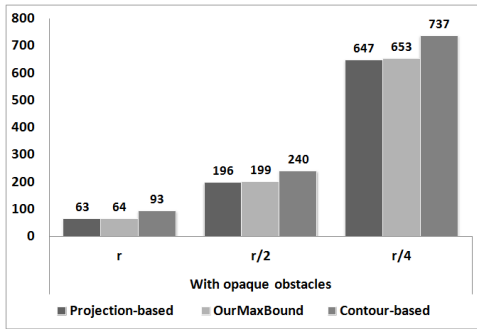


Figure 14: Impact of the sensing range on the number of sensor nodes (Boot configuration).

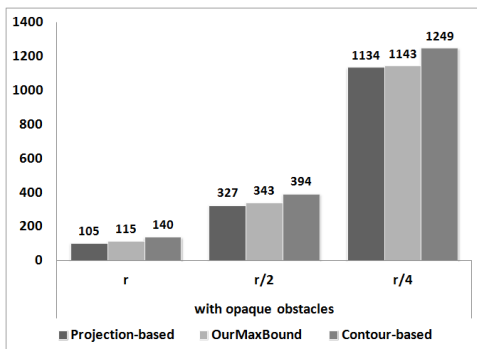


Figure 15: Impact of the sensing range on the number of sensor nodes (Star configuration).

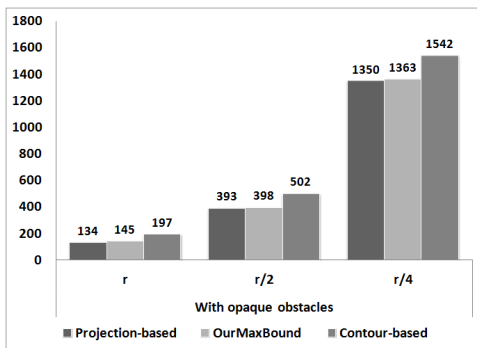


Figure 16: Impact of the sensing range on the number of sensor nodes (Warehouse configuration).

The comparative evaluation reported in Sections 5, 6 and 7 has the merit to quantitatively evaluate the impact of the complexity of the area (i.e. with/without obstacles, opaque/transparent borders, opaque/transparent obstacles) on the number of sensor nodes needed for a full coverage. The bound we computed  $OurMax_N$  is very accurate, whatever the configuration. Our method is always better than the contour-based method. Furthermore, we noticed the strong impact of border edges and obstacle edges whose length is

smaller than  $r\sqrt{3}/2$  on the number of edges required by a contour-based method.

## 8. CONCLUSION

With the tremendous progress in technology miniaturization and the high reduction of costs, many monitoring applications are supported by wireless sensor networks. The quality of this monitoring strongly depends on data gathering which assumes the full coverage of the area to monitor. For cost reason, minimizing the number of sensor nodes deployed is sought. In addition, there are many constraints related to the area to monitor: an irregular shape that may be not convex, existence of transparent obstacles, existence of opaque borders and obstacles. To ensure the coverage of such an area constitutes a challenging task that we tackle in this paper. We evaluate the impact of these three factors and propose a method whose complexity gradually increases with the complexity of the coverage problem. The performances of this method are compared with the contour-based method: using our projection-based method, simulation results show that at least 30% of the deployment cost are saved. Our method is simpler than other method based on computational geometry like Delaunay triangulation. Furthermore, we establish bounds on the number of sensors required. The deployment computed by our method can be provided to a mobile robot in charge of placing the static wireless sensor nodes at their optimized location.

## 9. REFERENCES

- [1] R. Kershner, *The number of circles covering a set*, American Journal of Mathematics, vol. 61, No. 3. pp. 665-671, Jul. 1939.
- [2] M. Shyam, K. Anurag, *Obstacle constrained total area coverage in wireless sensor networks*, arxiv-web3.library.cornell.edu/pdf/1001.4753v1, January 2010.
- [3] H. Tan, X. Hao, Y. Wang, F. C.M. Lau and Y. Lv *An Approximate Approach for Area Coverage in Wireless Sensor Networks*, Procedia Computer Science 19 (2013): 240-247.
- [4] C.Y. Chang, C.T. Chang, Y.C. Chen, *Obstacle-resistant deployment algorithms for wireless sensor networks*, IEEE Trans. on Vehicular Technology, Vol.58, N6, July 2009.
- [5] Y.C. Wang, C.C. Hu, Y.C. Tseng, *Efficient placement and dispatch of sensors in a wireless sensor network deployment*, IEEE Trans. on Mobile Computing, Vol.7 N2, February 2008.
- [6] H. Tan, Y. Wang, X. Lao, Q-S Hua, F. Lau, *Arbitrary obstacles constrained full coverage in wireless sensor networks*, WASA 2010, Beijing, China, September 2010.
- [7] A. Fotouhi, and M. Razzazi *Redundancy and coverage detection in wireless sensor networks in the presence of obstacles*, MIPRO, 2011 Proceedings of the 34th International Convention. IEEE, 2011.
- [8] B. Wang, *Coverage problems in sensor networks: a survey*, ACM Computing Surveys, vol43, N4, article 32, 2011.
- [9] X. Bai, S. Kumar, D. Xuan, Z. Yun, T.H. Lai, *Deploying wireless sensors to achieve both coverage and connectivity*, MobiHoc 2006, Florence, Italy, 2006.