*Sensors* **2011**, *11*, 9904-9927; doi:10.3390/s111009904

OPEN ACCESS

*sensors*

**ISSN 1424-8220**

www.mdpi.com/journal/sensors

*Article*

# Coverage Assessment and Target Tracking in 3D Domains

**Noureddine Boudriga** [1,⋆], **Mohamed Hamdi** [1] **and Sitharama Iyengar** [2]

[1] Communication Networks and Security Research Lab, University of Carthage, Ariana, 2083, Tunisia; E-Mail: hamdi.mm@gmail.com

[2] Robotics Research Laboratory, Louisiana State University, Baton-Rouge, LA 70802, USA; E-Mail: iyengar@csc.lsu.edu

⋆ Author to whom correspondence should be addressed; E-Mail: noure.boudriga2@gmail.com; Tel.: +21-671-857-000 ext. 1016.

*Received: 20 August 2011; in revised form: 15 September 2011 / Accepted: 15 September 2011 / Published: 20 October 2011*

---

**Abstract:** Recent advances in integrated electronic devices motivated the use of Wireless Sensor Networks (WSNs) in many applications including domain surveillance and mobile target tracking, where a number of sensors are scattered within a sensitive region to detect the presence of intruders and forward related events to some analysis center(s). Obviously, sensor deployment should guarantee an optimal event detection rate and should reduce coverage holes. Most of the coverage control approaches proposed in the literature deal with two-dimensional zones and do not develop strategies to handle coverage in three-dimensional domains, which is becoming a requirement for many applications including water monitoring, indoor surveillance, and projectile tracking. This paper proposes efficient techniques to detect coverage holes in a 3D domain using a finite set of sensors, repair the holes, and track hostile targets. To this end, we use the concepts of Voronoi tessellation, Vietoris complex, and retract by deformation. We show in particular that, through a set of iterative transformations of the Vietoris complex corresponding to the deployed sensors, the number of coverage holes can be computed with a low complexity. Mobility strategies are also proposed to repair holes by moving appropriately sensors towards the uncovered zones. The tracking objective is to set a non-uniform WSN coverage within the monitored domain to allow detecting the target(s) by the set of sensors. We show, in particular, how the proposed algorithms adapt to cope with obstacles. Simulation experiments are carried out to analyze the efficiency of the proposed models. To our knowledge, repairing and tracking is addressed for the first time in 3D spaces with different sensor coverage schemes.

**Keywords:** wireless sensor networks; coverage holes; 3D Voronoi diagrams; Vietoris-Rips complex

## 1. Introduction

One among the main WSN issues that should be addressed while dealing with target tracking and monitoring applications, in 3D environments with obstacles, is area coverage. This is because a sensor can detect the occurrence of events or the presence of hostile targets only if they are within its sensing range. Coverage reflects how well a zone is monitored or a system is tracked by sensors. Therefore, the WSN detection performance depends on how well the wireless sensors observe the physical space under control.

Several metrics have been provided in the literature to measure the quality of coverage. Among these metrics, one can mention the following: (a) the number of coverage holes; (b) the proportion of uncovered area with respect to the area under monitoring; and (c) the so called Average Linear Uncovered Length (ALUL), which has been developed in 2D zones to estimate the average distance a mobile target can traverse before being detected by one sensor [1]. The ALUL can be used to assess the detection efficiency of the WSN in more general spaces. However, the major shortcoming of this approach is its heavy computational load making it non-conforming with the severe processing and energy limitations characterizing WSNs.

Obstacles in monitored 3D domains may complicate seriously the role of the monitoring sensors, increase their power consumption, and limit the coverage efficiency of the process providing coverage control [2,3]. Procedures set up to implement coverage control and target tracking efficiency should be optimal. They should take into consideration the geographic nature of the monitored area and cope with the number and the shape of obstacles.

This paper proposes a coverage assessment approach amenable to implement advanced target tracking functionalities. First, it provides a technique based on the concept of retraction by deformation applied to a special space, called the Rips complex, associated with the deployment of a set of sensors to develop a low complexity algorithm for locating coverage holes. Second, it constructs a collaborative mechanism to repair coverage holes, assuming that the sensors have mobility capabilities. Third, the paper builds on higher-order Voronoi diagrams to define an efficient scheme to coordinate tracking activities of single and multiple targets. To the best of our knowledge, this is the first time where retraction by deformation and higher-order Voronoi tessellations are used for hole assessment and target tracking in 3D domains with obstacles using sensors. The major contributions of this paper are as follows:

- The definition proposed to distributively reduce the Rips complex associated to the sensors is general, in the sense that it applies to a large variety of sensor, detection techniques, monitored domains, and obstacles.

- The proposed cooperative coverage repairing approach considerably reduces the uncovered areas and provides efficient handling of obstacles with respect to existing methods. The detection and localization of holes is done with low complexity.

- We show that the higher-order Voronoi tessellations we utilize are useful for performing multiple tasks including activity scheduling and coordination. In addition, we show that local coverage information, when gathered using the Voronoi diagram, can be used to implement coverage preserving mobility models.

The remaining part of this paper is organized as follows: Section 2 describes the state of the art of coverage control in various areas in general and in 3D spaces in particular. Section 3 surveys the definition of the mathematical objects needed for coverage and tracking control, the Vietoris complex and the Voronoi diagram and discusses the retraction by deformation. Section 4 discusses different schemes based on the Vietoris complex to detect and count the coverage holes in 3D domains, locate these holes, and repair them. It also defines a special procedure to reduce the complexity of the Vietoris complexes without modifying their topological properties. Section 5 sets up models for coverage assessment, sensor mobility, and target tracking. Section 6 analyzes the complexity of the algorithms constructed in this paper and sets some extensions of our results to more general types of sensors. Section 7 develops simulation experiments to evaluate the performance of a monitoring system implementing our techniques. Section 8 concludes this paper.

## 2. Related Work

Studies on coverage, holes, and boundary detection have been addressed using three main categories of techniques: geometric methods, statistical/probabilistic methods, and topological methods.

Studies using probabilistic approaches usually make assumptions on the probability distribution of the sensor deployment. Fekete *et al.* [4] assume uniformly randomly distributed sensors inside a geometric region for their boundary detection algorithm. Their approach hinges on the idea that the boundary nodes would have lower average degrees than that of the "interior" nodes and statistically provide a degree threshold to differentiate interior and boundary nodes. Kuo *et al.* [5] propose an error model for location estimation using probabilistic coverage, while Ren *et al.* [6] presents an analytical model based on probabilistic coverage to track moving objects in a densely covered sensor field. Most of probabilistic approaches have focused on the detection and tracking of objects in a sensor field. They did not address other related issues such as location of the holes, number of such holes and repairing.

A number of literature has addressed the static or "blanket" coverage. Dynamic or "sweeping" coverage [7] has been also a common and challenging task with applications ranging from security to housekeeping. Two primary approaches to static coverage problems in the literature. The first uses computational geometry tools applied to exact node coordinates. Such approaches are very rigid with regards to inputs: one, for example, must know exact node coordinates and must know the geometry of the domain to determine the Delaunay complex. To alleviate the former requirement, many authors have turned to probabilistic tools. For example, in [8], the author assumes a randomly and uniformly distributed collection of nodes in a domain with a fixed geometry and proves expected area coverage. Other approaches give probabilistic or percolation results about coverage for randomly distributed nodes. The drawback of these methods is the fact that uniform distribution of nodes may not be always realistic.

More recently, the robotics community has explored how networked sensors and robots can interact and augment each other: (see e.g., [9] for more details). There are several new approaches to networks

without localization that come from research works in ad hoc wireless networks that are not unrelated to coverage questions. One example is the routing algorithm of [10], which generally works in practice but is a heuristic method involving heat-flow relaxation. This work investigates the issues of maintaining coverage and connectivity by keeping minimum number of sensor nodes to operate in the active mode. The authors show that if the radio range is at least twice the sensing range, then complete coverage implies connectivity. A decentralized and localized density control algorithm, called OGDC, is devised to control and maintain coverage and connectivity. However, their approach requires knowledge of node location. The authors claim that this requirement can be relaxed to that each node knows its relative location to its neighbors.On the other hand, Hsin and Liu [11] give methods for localizing an entire network if localization of a certain portion is known. They address the problem of target tracking in face of partial sensing coverage by considering the effect of different random and coordinated scheduling schemes. In their coordinated-coverage algorithm, a sensor might decide to sleep for some time after acknowledgments from its neighbor(s) that must be active. These decisions are not synchronized as individual sensors could negotiate with sponsors independently.

Since coverage verification is inherently a geometric problem, many research done in this area are based on computational geometry, and more precisely on the Voronoi Tessellation (and its dual, Delauney Triangulation). Motivated from the early success of the application of geometric techniques to cope with coverage problems (*Art Gallery Problem*), researchers have applied these techniques to ad-hoc distributed sensor networks ([12–15]).

The most important drawback of these approaches is that they are too computationally expensive to be implemented in real-time contexts. Another severe limitation is the impact of localization uncertainty on the performance of these approaches. These claims are well-documented in ([16]). In fact, to detect coverage holes, the locations of the sensors must be exactly known. Obviously, this cannot be always provided, especially when the sensing nodes are mobile. Moreover, equipping sensors with localization devices may considerably increase the deployment cost of the WSN and reduce its resources. In the following paragraphs, we summarize the methodologies, problems addressed and results of some of the recent, notable studies in the area of detection and coverage in wireless sensor networks.

Meguerdichian *et al.* [13] study the problem of computing a path along which a target is least or most likely to be detected. They provide an optimal polynomial time algorithm that uses graph theoretic and computational geometric (Voronoi diagram) methods. They address the issues of maximal breach path, maximal support path and provide best and worst case coverage using computational geometry. Delaunay triangulation was used to find the best-coverage path. In addition, deployment heuristics are provided to improve coverage. Since computational geometric methods require location information, the authors implement a location procedure prior to their coverage scheme. This procedure requires that a few of the deployed nodes (called beacons) must know their locations in advance (either from GPS or pre-deployment). Li *et al.* [14] uses local Delauney triangulation, relative neighborhood graph and the Gabriel graph to find the path with the best-case coverage.

Huang *et al.* [15] study the problem of $k$-coverage. They propose solutions to the $k$-UC and $k$-NC (Unit Disks and Non-Unit Disks) coverage problems which are modeled as decision problems whose goal is to determine if each location of a target sensing area is sufficiently covered. They present a polynomial-time algorithm with a geometric approach that runs in $O(nd \log d)$ time.

Ghrist *et al.* [17] use topological methods to detect insufficient sensor coverage and holes. In their seminal work on using homological concepts for addressing hole detection and coverage, their algorithm detects holes with no knowledge of their location. Although the approaches by Ghrist *et al.* have many desirable properties, the assumption of a static network and the centralized scheme are not suitable for dynamic networks.

## 3. Mathematics for Coverage and Tracking

The objective of this section is to provide a mathematical model for accurately gauging the coverage degree of a monitored domain in the 3D space $\mathbf{R}^3$ and repairing the coverage holes. This model uses the Vietoris Complex [6,8].

The following assumptions will be used in the next subsections: Let $\mathbf{M}$ be a bounded domain (or manifold) in $\mathbf{R}^3$ with non-empty boundary $\partial M$. The boundary is assumed to be an orientable topological surface (*i.e.*, a closed surface homeomorphic to some number of spheres and some number of connected sum of $g$ tori, for $g \geq 1$, [18]). Let $\delta : \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}^+$ denoting the Euclidean distance. We denote by $S$ a set of sensors deployed in $\mathbf{R}^3$ to monitor $M$, and by $|S|$ the number of these sensors. We will designate indifferently by $p \in S$ the sensor in $S$ and its location $(x_p, y_p, z_p)$ in $\mathbf{R}^3$. Let us notice, finally, that the sensors can be deployed inside $\mathbf{M}$ or outside it.

### *3.1. Voronoi Diagrams for Spherical-Detection Sensors*

Let us assume that the sensors in $S$ have identical covered area represented by a ball with radius $\rho$. For every pair $p, q \in S$, we denote by $B(p,q)$ the plane, in $\mathbf{R}^3$ perpendicular to segment $[p,q]$ and passing by its middle point and by $H(p,q)$ the half space of $\mathbf{R}^3$ containing the $p$ and delimited by $B(p,q)$. Thus, $B(p,q)$ and $H(p,q)$ are expressed as follows:

$$B(p,q) = \left\{x \in \mathbf{R}^3 / \delta(p,x) = \delta(q,x)\right\}. \tag{1}$$

$$H(p,q) = \left\{x \in \mathbf{R}^3 / \delta(p,x) \leq \delta(q,x)\right\}. \tag{2}$$

We also denote by $H_M(p,q)$ and $B_M(p,q)$ the intersection of $H(p,q)$ and $B(p,q)$ with $\mathbf{M}$, respectively.

The Voronoi cell generated by $p \in S$ is nothing but the common area to the $(|S|-1)$ closed half spaces containing $p$ involving the other sensors. Therefore, the Voronoi cell generated by $p$ is expressed by:

$$V_S(p) = \bigcap_{q \in S \setminus p} H(p,q) \tag{3}$$

The Voronoi cell of a sensor is convex and contractible. The common boundary of two Voronoi cells $V_S(p) \cap V_S(q)$ is included in $H(p,q)$. It can be a plane, a half plane, an edge, a point, or an empty set. The Voronoi diagram associated to the set $S$ of sensors deployed to monitor $\mathbf{M}$ is the unique subdivision defined in $\mathbf{R}^3$ by the Voronoi cells associated to all sensors. Thus, every cell of the subdivision contains

the nearest neighbors defined in $S$ for a sensor $p$. The Voronoi diagram of $S$ is the set of point belonging to the all the Voronoi cell. Hence we have:

$$V^D(S) = \bigcup_{p \in S} V_S(p). \tag{4}$$

In particular, the Voronoi diagram $V^D(S)$ has no vertices and no edges when the sensors are located at collinear points. In that case, the faces of the Voronoi diagram are parallel planes. In addition, one can notice that when $p \in S$ lies on the boundary of the convex hull of $S$, then the Voronoi cell of $p$ is unbounded in $\mathbf{R}^3$.

Since in this paper, we are rather interested in partitioning a domain $\mathbf{M}$ into cells according to $k$-nearest neighbors in $S$, for a given integer $1 \leq k \leq n-1$, we turn now to the definition of the High-order Voronoi diagrams, as they are useful concepts to define these sets and support target tracking. An order $k$ Voronoi diagram is defined as follows: Let $T \subset S$ containing $k$ sensors, the $T$-generated cell is defined by

$$V(T) = \{x \in \mathbf{R}^3 | \forall p \in T, \forall q \in S - T, \delta(x,p) \leq \delta(x,q)\}. \tag{5}$$

The order $k$ Voronoi diagram is given by:

$$V_k^D(S) = \bigcup_{T \subset S, |T|=k} V(T). \tag{6}$$

One can easily see that the order 1 Voronoi diagram $V_1^D(S)$ is just $V^D S$, that $V(T)$ can be empty, and that $V^D(S)$ induces a partition on the domain M into bounded components.

### *3.2. Vietoris-Rips Complexes*

We consider a set of points $S = \{v_1, .., v_n\}$ corresponding to the locations of a set of sensor nodes in a 3D space. For brevity, $(v_i)_{1 \leq i \leq n}$ will be simply referring indifferently to as sensor nodes and points. We suppose that each sensor is capable of covering a disk of radius $r_c$ and communicate with the other sensors within a distance $r_b \leq \sqrt{3} r_c$. The total region covered by the sensor network can be represented by:

$$\Gamma(S) = \bigcup_{v_i \in S} \Gamma_{v_i, r_c} \tag{7}$$

where $\Gamma_{v_i,r_c} = \{x \in \mathbb{R}^3 : \|x - v_i\| \leq r_c\}$.

A $k$-simplex (or a simplex of dimension $k$) $\sigma$ is an unordered set $\sigma = \{v_0, v_1, .., v_k\} \subseteq S$, where $v_i \neq v_j$ and $\delta(v_1, v_J)$, for all $i \neq j$. A face of the $k$-simplex $\sigma$ is a $(k-1)$-simplex formed by $k$ elements (or vertices) of $\sigma$. Clearly, any $k$-simplex has exactly $k+1$ faces. The collection of all k-simplices of S is called the abstract associated with $\Gamma(S)$. In fact, an abstract simplicial complex $X$ is a finite collection of simplices which is closed with respect to the inclusion of faces; meaning that, if $\sigma \in X$, then all faces of $\sigma$ are also in $X$. It is noteworthy that a simplicial complex is a generalization of a graph; that is, the connectivity graph is nothing but the set of 1-simplices of the simplicial complex associated to a set $V$ of points in the 3D space.

Now let us discuss the definition of the Vietoris-Rips complex. This complex captures the features related to connectivity and coverage of WSNs.

**Definition 3.1.** *(Vietoris-rips complex) Let $S$ be a set of points in a 3D space and a given radius $\epsilon$. The Vietoris-rips complex of $S$, denoted by $R_\epsilon(S)$, is the simplicial complex whose $k$-simplices correspond to unordered $(k+1)$-tuples of points in $S$ which are pairwise within Euclidean distance $\epsilon$ of each other.*

A subset of $k+1$ points in $S$ determine a $k$-simplex of for the Vietoris-rips complex if, and only if, each of these points lies within the intersection of the balls of radius $\epsilon$ centered at the other $k$ points.

The reader, however, may wonder whether such topological structure can be computed in practice by tiny motes equipped with radio devices and limited storage capabilities. To answer this question, we propose a simple mechanism allowing a fully distributed construction of the Vietoris-rips complex. Through a 3-step broadcast of connectivity information, each sensor node can be aware of what simplices it belongs to, and what other simplices its neighbors belong to. To this end, we assume that every sensor node has a unique identifier (typically a layer-2 address) and has enough space to maintain a table of identifiers. The protocol performs as follows:

1. Initialization: Every sensor $v_i$ broadcasts its identity to its neighbors. Upon receipt of the message, each sensors builds the list, denoted by $\Sigma_0^i$, of 0-simplices formed by its neighbors.

2. Edge construction: Sensor $v_i$ appends its identity to the vertices in $\Sigma_0^i$ to construct the list, say $\Sigma_1^i$, of all 1-simplices it belongs to. It also determines the number $n_i$ of its neighbors. Then it informs its neighbors about the 1-simplices it built.

3. Simplicial iteration: On receiving the information from its neighbors, sensor $v_i$ starts building the lists $\Sigma_j^i$, $2 \leq j \leq n_i$, by simply adding appropriately the structures it has received to the ones it has already constructed.

An informal explanation of the construction algorithm is as follows. Simplices of higher dimension are constructed iteratively. In the first iteration, the 2-simplices are constructed by applying the following rule:

$$< v_i, v_j > \in \Sigma_1^i, < v_i, v_k > \in \Sigma_1^i, < v_j, v_k > \in \Sigma_1^j \longrightarrow < v_i, v_j, v_k > \in \Sigma_2^i$$

for every $i, j$, and $k$, provided that $i \neq j, i \neq k, j \neq k$. The rules used for the following iterations are similar.

### *3.3. Homotopy and Retraction*

Let $X$ and $Y$ be two topological spaces and $f, g : X \to Y$ be two maps (or continuous functions). We say that $f$ and $g$ are homotopic if there is a map $F : [0,1] \times [0,1] \longrightarrow X$ such that

$$F(x,0) = f(x), F(x,1) = g(x), \forall x, y \in X$$

Let $x_0 \in X$ be a given basepoint of $X$. A loop based on $x_0$ is a map $\alpha : [0,1] \longrightarrow X$, such that $x_0 = \alpha(0) = \alpha(1)$. An equivalence relation on the set of all loops based at $x_0$ can be defined by stating that loops $\alpha_1$ and $\alpha_2$ are equivalent if they are homotopic with respect to $x_0$; meaning that there exists a homotopy $F$ between $\alpha_1$ and $\alpha_2$ such that

$$F(0,t) = F(1,t) = x_0, \forall t \in [0,1].$$

We denote the equivalence class of a loop $\alpha : [0,1] \longrightarrow X$ based at $x_0$ by $[\alpha]$ and call it the based homotopy class of the loop $\alpha$. The set of equivalence classes of loops based at $x_0$ is denoted by $\pi_1(X, x_0)$ and is called the fundamental group. It can be equipped with a multiplication defined by $[\alpha_1] \star [\alpha_2] = [\alpha_1.\alpha_2]$, for all loops $[\alpha_1]$ and $[\alpha_2]$ based at $x_0$, where $\alpha_1.\alpha_2$ is the loop obtained by attaching $\alpha_1$ to $\alpha_2$. A second group of homotopy, denoted by $\pi_2(X, x_0)$ can be defined as the set of homotopy equivalence classes of applications $\beta : [0,1]^2 \longrightarrow X$, based at $x_0$. It is an Abelian group, [19].

On the other hand, a map $f : X \longrightarrow Y$ is called a homotopy equivalence if there is a map $g : Y \longrightarrow X$ such that $f \circ g$ is homotopic to the identity function in $X$ and $g \circ f$ is homotopy to the identity function $Y$. Thus, one can say that two spaces are homotopy equivalent if they have "the same shape".

A deformation retraction of a space $X$ onto a subspace $A \subseteq X$ is a map $f : X \times [0,1] \longrightarrow X$ such that:

$$f(x,0) = x, f(x,1) \in A, f(a,t) = a, \ \forall x \in X, a \in A, 0 \leq t \leq 1.$$

In other words, The subset $A$ is a retraction by deformation of the space $X$ if, starting from the original space $X$ at time 0, we can continuously deform $X$ until it becomes the subspace $A$ at time 1 and deformation is performed without ever moving the subspace $A$ in the process. It is obvious that, if $A$ is a retraction by deformation of $X$, then $X$ and $A$ are homotopically equivalent.

Finally, let $K$ be complex, a retraction filtration of $K$ is a nested finite sequence of subcomplexes $K_i$,

$$K_0 \subseteq K_1 \subseteq ... \subseteq K_n = K.$$

such that, for all $k \geq 0$, $K_k$ is a retract by deformation of $K_{k+1}$. Thus, it can be shown easily that, $K_0$ and $K_n$ have the same type of homotopy and the same homotopy group.

Let $T = \{p_1, .., p_k\}$ be a simplex and A $T_1 = \{p_2, .., p_k\}$ be one of its faces. Then $A = T - (T_1 - \partial T_1)$ be the part of the boundary of $T$ that is not internal to $T_1$. Then $A$ is a deformation retract of $T$.

Let $R(S)$ be the Rips complex associated with $S$, repeating the process of retraction of simplexes that are on the boundary of $R(S)$, with faces external to $R(S)$, would lead to a filtration of $R(S)$, say $K_k, 0 \leq k \leq n$, such that, for all $k \geq 0$, $K_k$ is a retract by deformation of $K_{k+1}$ and $K_{k+1}$ is obtained from $K_k$ by adding one simplex, external to $K_k$ and belonging to $R(S)$. The object $K_0$ has no simplex with external face that is retractible.

## 4. Coverage Hole Management of Spherical Sensors

In this section, we propose a novel distributed technique to count the coverage holes of WSN using the retraction theory of spaces. In particular, we show that the Vietoris-rips complex associated with the WSN can be reduced to a simpler space that is tightly related to the number of holes.

In the following, let $D \subseteq \mathbb{R}^3$ be a compact domain in the 3D space $\mathbb{R}^3$ and $\partial D$ be its boundary. We consider that $D$ contains no obstacles. We also consider that a collection $S = \{v_1, .., v_n\}$ is deployed over domain $D$ and that the sensors are equipped with local communication and sensing capabilities. In fact, each sensor is capable of communicating directly with other sensors in its proximity (within a given distance $r_b$) and has a limited sensing range $\epsilon$.

### *4.1. Reducing the Vietoris-Rips Complex*

We assume, in this subsection, a complete absence of localization capabilities and metric information, in the sense that the sensors in the network can determine neither distance nor direction. Under these assumptions, we are interested in designing distributed algorithms for coverage assessment and hole detection.

To this end, we need first to introduce a special procedure, called *Retract*, that reduces the size of the Vietoris-rips complex while keeping its type of homotopy. Repeating this procedure several times will eliminate all the 3-cells of the Vietoris-rips complex.

Let $R_\epsilon(S)$ be the Vietoris-rips complex. Let $\{v_0, ..., v_3\}$ be a 3-simplex in $R_\epsilon(S)$ such that one of the 2-cell $\{v_0, v_1, v_2\}$ does not belong to another 3-simplex in $R_\epsilon(S)$. If such a situation does not exist, then one can easily deduce that $R_\epsilon(S)$ has no 3-cells. Let $X_1$ and $A_1$ be the set of points $x \in R_\epsilon(S)$ belonging to simplex $\{v_0, v_1, v_2\}$ and the subset of $X_1$ generated by the other two faces, respectively. Then its is easy to construct a map $h_1 : X_1 \times [0, 1] \longrightarrow X_1$ such that:

$$h_1(x, 0) = x, h_1(x, 1) \in A_1, h_1(a, t) = a, \ \forall x \in X, a \in A_1, 0 \leq t \leq 1.$$

Map $h_1$ can be easily extended to a map

$$Retract : R_\epsilon(S) \times [0, 1] \longrightarrow R_\epsilon(S)$$

such that:

$$Retract(x, 0) = x, Retract(x, 1) \in A, Retract(a, t) = a, \ \forall x \in X, a \in A, 0 \leq t \leq 1.$$

where $A$ is $R_\epsilon(S) - X_1) \cup A_1$.

Repeating the map *Retract* several times will lead to eliminating all the 3-simplices in $R_\epsilon(S)$. The map *Retract* can also be reapplied several times to delete all 2-simplices and 1-simplices that a free face. The resulting space, say $R_\epsilon^{red}(S)$.

**Proposition 4.1.** *Let $S$ be a set of sensors. If $R_\epsilon(S)$ is path-connected, then $R_\epsilon^{red}(S)$ satisfies the following properties:*

1. *$R_\epsilon^{red}(S)$ is homotopy equivalent to $R_\epsilon(S)$*
2. *the number of holes delimited by $R_\epsilon^{red}(S)$ is equal to the number of holes of the vietoris space $R_\epsilon(S)$*

*Proof.* Applying the map *Retract* several times helps creating a retraction filtration of $R_\epsilon(S)$ such that:

$$R_\epsilon^{red}(S) = K_0 \subseteq K_1 \subseteq ... \subseteq K_n = R_\epsilon(S).$$

where $n$ is number of 3-simplices in $R_\epsilon(S)$. Since, for every $i$, $K_i$ is homotopy equivalent to $K_{i+1}$, we can deduce that $R_\epsilon^{red}(S)$ is homotopy equivalent to $R_\epsilon(S)$.

The second statement of the theorem can be deduced from the following features:

- a holes is a path connected component that is surrounded by the delimiting space ($R_\epsilon^{red}(S)$ and $R_\epsilon(S)$).
- Retracting a 3-simplex in $R_\epsilon(S)$ may enlarge a hole but does not eliminate it.
- The retraction process does not create holes since it operates on the simplices that have free faces.

### *4.2. Counting and Locating Coverage Holes*

To count and locate holes, we set up a 3-step algorithm. In the first step, we construct the external boundary of $R_\epsilon(S)$. This is the subset of $S$ containing all the nodes occurring on free faces and facing the boundary $\partial D$ of the domain. In the second step, we define an algorithm that detects holes by progressively transforming the external boundary by retracting all its external simplices. In the third step, the following process is repeated: one external 2 simplex is deflated, the Retract map is applied several times to reduce appearing simplices with free faces, and the external boundary is updated. The number of iterations of this process gives the number of coverage holes.

#### 4.2.1. Constructing the Boundary of $R_\epsilon(S)$

Let us assume that the boundary $\partial D$ of the domain $\mathbf{D}$ under monitoring can be seen (or detected) by the sensors in $S$ and that the nodes in $S$ broadcast periodically their unique ID numbers. The construction is based on the three following actions:

- Every sensor node detecting a boundary component of $\mathbf{D}$ or finding itself on an external facet sends this information to its neighbors.
- The information related to boundary detection, when received by sensors should be put together to form the external boundary of $R_\epsilon(S)$, by simply allowing every sensor node to know which neighbor is on the external boundary.
- The nodes broadcast information related the external boundary of $R_\epsilon(S)$ so that every node on the boundary can have a precise picture of the boundary.

#### 4.2.2. Counting Coverage Holes

Counting the coverage holes can be set up by an algorithm that repeats iteratively the following major procedures:

- Boundary retraction: Let $C_n$ be a n-simplex on the boundary of $R_\epsilon(S)$ and $C_{n-1}$ be one of its external faces, then $C_n$ can be retracted using the procedure Retract and the boundary is updated by adding a new node (the one in $C_n - C_{n-1}$), if $n \geq 2$, or by deleting the node occurring in $C_{n-1}$, if $n = 1$.
- Boundary deflation: When all the simplices on the boundary of have been retracted, a pre-selected node in $S$ (in charge of the counter) selects one of the nodes of the new external boundary, withdraws it from the boundary, and increments the counter.

#### 4.2.3. Locating Coverage Holes

It is worth noticing that, when a deflation of a 2-simplex on the boundary $R_\epsilon(S)$ is applied after retraction is complete, a hole is reduced from the coverage zone. This because the selected node, for deflation, is observing the hole, since it is one of the nearest nodes surrounding the reduced hole. Thus, this node can start the construction of the boundary of the reduced hole by determining the list of the nodes surrounding immediately the hole.

One can conclude, therefore, that any time a deflation is operated, a hole can be located by simply constructing its boundary using the nearest nodes to that hole.

### *4.3. Repairing Coverage Holes*

Let us here assume that the 3D domain $\mathbf{D}$ under monitoring has no obstacles and let us denote by $\chi$ ($\chi = 4\pi\epsilon^3/3$) the volume of the area covered by a sensor and by $Vol(D)$ the volume of $\mathbf{D}$. One can state that the number $|S|$ of sensors in $S$ should be higher than the number $N_0 = Vol(D)/\chi$ to be able to guarantee full coverage of $\mathbf{D}$, at least after hole detection and coverage optimization. Therefore, we will assume in the sequel that this condition is satisfied. Finally, we assume that the sensors are able to move and detect the external boundary of $\mathbf{D}$, when they are close to it, like in the above subsection.

Repairing holes aims at extending the coverage by eliminating the holes, or at least by shrinking considerably their size. An algorithm can be defined to this purpose. It can be built based on the following general rules:

- A node detecting the external boundary $\partial M$ should keep seeing the boundary when it moves.
- A node on the external boundary of $R_\epsilon(S)$ should move towards the uncovered area, when it does not see the boundary.
- When two neighbor nodes on the external boundary of $R_\epsilon(S)$ are separated by a distance higher than a predefined threshold, say $\theta_1$, and one of them is not seeing the boundary of $\mathbf{D}$, then the sensor unable to see the boundary asks its successor (*i.e.*, a neighbor involved in the retraction of the simplex containing this sensor) to move towards the external boundary.
- A node seeing the boundary should inform its neighbors so that they can move accordingly.
- When the distance between a sensor $s$ and its neighbors on the boundary of a hole is lower than a predefined value, say $\theta_2$, then $s$ should move in the opposite direction of the hole, while the other sensors should move towards the hole so that when they see each other, $s$ can withdraw itself from the minimal surface after informing its neighbors.
- A node on the external boundary, finding itself unable to move informs, its successor to move towards its direction.

## 5. Target Tracking in 3D Domains

In this section, we use 3D Voronoi diagrams to optimize sensor coverage and target tracking performance. We first propose a strategy to measure the uncovered zones of the monitored region.

Then, we develop two mobility models that provide target tracking using order k Voronoi diagrams and optimize the coverage ratio of a zone using Voronoi cells. Finally, we extend these models to multiple target tracking. We assume in this section that the sensors have spherical coverage. The vector-guided case can be addressed using similar techniques.

### *5.1. Measuring Uncovered Areas*

Assume that a location $x$ within the surveillance area is not covered by any sensor. Let $\mathcal{L}(x,\theta)$ define the Linear Uncovered Length (LUL) at location $x$ with direction $\theta$. This is the undetected path length of a target traveling from location $x$ with direction $\theta = (\theta_1, \theta_2)$, for $0 \leq \theta_1 \leq 2\pi, -\pi/2 \leq \theta_2 \leq \pi/2)$.

The Average Linear Uncovered Length (ALUL), denoted by $ALUL(x)$, introduced in [20,21], for the 2D space, gives an approximation of the average distance that can be made by a target, moving in 3D space, before being detected by the sensor network. The Average Linear Uncovered Length (ALUL) function can be defined by the following formula:

$$ALUL(x) = \begin{cases} 0, & \text{if } x \text{ is covered.} \\ \frac{1}{(2\pi)^2} \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \mathcal{L}(x,\theta_1,\theta_2) d\theta_1 d\theta_2, & \text{otherwise.} \end{cases}$$

More generally, when $A$ is a subregion of the 3D domain under supervision, the Average Linear Uncovered Length related to $A$, $ALUL(A)$, that a target can travel within $A$ without been detected by a sensor is given by the expression:

$$\textit{ALUL}(A) \equiv \frac{\int_{x \in A} ALUL(x) dx}{\parallel A \parallel}, \tag{8}$$

where $\parallel A \parallel$ is the volume of $A$.

the ALUL metric was developed to deal with a static deployment, which is not the case of our study. When a mobility model is implemented, the topology of the WSN is no longer static. To overcome this, we extend this notion so as to support sensor node mobility. The ALUL should also vary according to time and should use a function, denoted by the $\mathcal{L}(x,\theta,t)$, that defines the Linear Uncovered Length at location $x$ with direction $\theta$, at time $t$. Based on this reasoning, we define the metric $ALUL_m(x,t)$ representing the ALUL in a location $x$ at time $t$ and given by:

Due to sensor node mobility, the ALUL, over time, in a point $x$ will be expressed by:

$$ALUL_m(x) = \int_0^{\infty} ALUL_m(x,t) dt. \tag{9}$$

Finally, $ALUL_m(A)$ can be computed by Equation (8) by replacing $ALUL(x)$ by $ALUL_m(x)$.

From the performance evaluation perspective, two important points should be highlighted:

- $ALUL_m(A,t)$ gives information about the coverage-preserving capabilities of the mobility model. It can be used to state whether the steady state is rapidly reached, and whether the mobility model affect the detection performance of the sensor network.

- $ALUL_m(A)$ provides information about the long-term behavior of the mobility model. It can be used to evaluate the impact of mobility on the possibility for a target to be undetected within the monitored region.

### *5.2. Mobility Models for Target Tracking*

In this section, we show how the Voronoi cells can be used to implement target tracking using a sensor mobility model. In fact, we define two mobility models:

- The first model is called k-mobility model. Sensor nodes in this model move toward the regions where the hostile target is supposed to be and collaborate to keep the target controlled by k sensors all the time, To this end, the order k Voronoi diagrams are used and maintained all the time.
- The second model is called simplified model. It relies on estimating the uncovered zones within the Voronoi cells, using the ALUL metrics and moving sensor nodes toward the "uncovered zones".

While the first model is triggered by the occurrence of targets, the second model aims at adapting the covered area so that the targets can be detected with higher probabilities. Obviously, the k-mobility model is more energy-consuming than the second since it encompasses the prediction of the target position and requires tracking using k sensor nodes. Therefore, we suppose that the second model can be used when energy resources become scarce. The performance of both models will be assessed in Section 7. Moreover, one can notice that the prediction function we are using is tightly related to the coverage of the zones where the targets are expected and that the mobility models assume that nearest sensor nodes can move to these zones while reducing the coverage of other zones where targets are not expected. In fact, the greater is the number of target detection signals, the better is the prediction precision to command sensor movements.

#### 5.2.1. The k-Mobility Model

In the following, we distinguish two cases: (a) a target crossing a $k-$covered area and (b) a target crossing non $k-$covered zone.

#### 5.2.2. For a Target Crossing a $k-$covered Zone

The mobility algorithm is triggered upon the detection of a target presence. Every detecting sensor sends its detection signal to the relevant intermediate sensor (called IS). The latter collects all detection signals, verifies their integrity, deduces the current zone where the target might be, estimates the positions of the target in the next of time slot, and commands k sensors to move to monitor the new zone to ensure tracking continuity.

Typically, the selected zone of target presence is taken among other zones (when more then k sensors detect the target presence). These zones are ordered according to the probability of presence of the target. The zone selected is the one presenting the highest probability among those which are $k-$covered.

The mobility algorithm is defined through five steps:

1. Assume that $k'$ sensors detect the target ($k' > k$). The $k'$ sensors $s_i$, $1 \leq i \leq k'$, send their detection data $d_i$ to an intermediate node under the form:

$$d_i = (r_{t,i}, \theta_{t,i}, \tau_{t,i}, s_i)$$

where $r_{t,i} = \delta(x_i, z_{t,i})$ is the Euclidean distance separating $s_i$ from the position $z_{t,i}$ of the target as seen by $s_i$, $\theta_{t,i} = (\alpha_{t,i}, \beta_{t,i})$ is the direction of the vector $\overrightarrow{z_{t,i} - x_i}$, and $\tau_{t,i}$ is the detection instant.

2. In the case where detection signals are sent to different intermediate nodes; the intermediate coordinate to gather all signals (or at least $k$ of them) at a unique node IS, which verifies first the authentication of the messages.

3. IS constructs:

   - The zone of target presence $Z_{t,i}^{tau}$ for each sensor based the errors made for the values reported. This zone is delimited by the following eight points:

     $$(r_{t,i} \pm \Delta r, (\alpha_{t,i} \pm \Delta\alpha, \beta_{t,i} \pm \Delta\beta)$$

     as defined by the estimated detection errors.

   - The most likely target presence zone $Z^{\tau}(t)$. Several strategies can be used for this including selecting the largest intersection of k zones of the form $Z_{t,i}^{tau}$. It can also be the largest union of k zones. Let $T$ be the set of $k$ sensors involved in the definition of $Z^{\tau}(t)$.

   Then, IS computes the order k Voronoi cell $V^S(T)$. Obviously, it contains $Z^{\tau}(t)$.

4. IS estimates the zone $Z^{\tau,+}(t)$, where target $z_t$ is likely to be in the next time slot. Several strategies can be used for this estimation including extrapolation of older positions or some information related to target direction and speed. It also estimates the most likely new position of $z_t$.

5. IS selects $k$ sensors based on a specific criteria and order them to move towards $Z^{\tau,+}(t)$ to increase its coverage. If no criteria is used, then the order goes to the sensors in $T$. A criteria can simply to reduce sensor movement.

When a criteria is applied for the selection of k sensors to cover the new position, some of the selected sensors (say $k''$ sensors) may belong to $T$ and the other (say $k-k''$) have to be added among the neighbors of $T$. This situation is addressed in the following subsection.

5.2.3. For a Target Crossing a Non $k-$covered Zone

In this case, only $k'$ ($k' \leq k$) detection signals are received by the intermediate sensor IS, which should proceeds at the construction of the probable current zone of presence of the target the way the preceding algorithms does. Then it starts the selection of the remaining $(k - k')$ required signals. Then, it orders the movement of the k sensor provide $k$ monitoring to the target. For this purpose, IS executes the following steps:

1. IS computes the most likely zone of target presence let $z_t$ using the $k'$ reports from $k'$ sensors denoted by $s_1, ..., s_{k'}$.

2. For each $i \leq k'$, IS selects the nearest $k$ sensors to $s_i$. It computes the related $k-$Voronoi cell $V_i^{(k)}$ and deduces the intersection $z_t \cap V_i^{(k)}$

3. For each $i \leq k'$, IS gets the number of sensors $k_i$", $0 \leq k_i" < k$, that have sent detection signals to IS.

4. IS classifies the $k-$ Voronoi cells according to the value of $k_i$". The greater $k_i$" is, the most important is the probability of presence of the target in $V_i^{(k)}$. A small value of $k_j$" induces that the target is going in or out the cell $V_i^{(k)}$.

5. IS selects the nearest $k$ sensors involved in $\partial V_i^{(k)}$, where $k_i" = max_{j \leq k'} k_j$, and guides the $(k - k")$ added sensors (among the nearest sensors to $s_i$) to move towards $\partial V_i^{(k)}$. For that, it sends them a mobility instruction including the probability of presence of the target. A mobility instruction is defined by the 3-tuple.

$$(r_i, \alpha_i, \pi_i)$$

where $r_i \geq \delta(s_i, p)$ such that $\forall q \in \partial V_i^{(k)}, \delta(s_i, p) \geq \delta(s_i, q)$. and $\alpha_i = argmax\widehat{xs_iy}$ where $x, y \in v_i$ and $v_i$ is the set of the vertices of the boundary $\partial V_i$, $\pi_i = k"/k$ is the probability of presence of the target in $\delta V_i^{(k)}$.

To enhance coverage while keeping more mobility freedom, we implement a group mobility model in which ground sensors move in groups in order to preserve the $k-$coverage. To this purpose, for each mobility step, the sensors define randomly groups of $k$ members for each, the latter are not required to be the nearest neighbors. Each group chooses randomly a head which chooses the first mobility step. The remaining members of the group take into account this choice to determine their next mobility step. By this way, every sensor's mobility will depend on the integrating group. Furthermore, a sensor may move from one group to another in each mobility step. This model enables the definition of overlapping $k-$Voronoi groups which increases the guarantee of having a $k-$coverage.

#### 5.2.4. Simplified Mobility Model

We propose hereafter a mobility model which is based on the use of simple Voronoi diagram to identify and reduce coverage holes.

This model can serve to implement a mobility strategy where a sensor node looks for one or more neighbors that are at least $2\rho$-distant from it. If such nodes exist, the sensor node moves toward the most distant neighbor, denoted by $n_f$, with a distance $\frac{\delta(s_i, n_f) - 2\rho}{2}$.

The following result extends this strategy to the case where the monitored region is required to be $k$-covered using the simplified algorithm. It uses a set, denoted by $X(s_i, V(S))$, which defined the set of intersection points expressed as follows:

$$X(s_i, V^D(S)) = \mathfrak{D}\left(V^D(S \setminus \{s_i\})\right) \bigcap \Gamma(s_i, R_{s_i}), \tag{10}$$

where $\mathfrak{D}$, for a region $R \subseteq \mathbf{R}^3$, denotes the boundary of $R$.

For the sake of parsimony, we do not provide proofs for these corollaries in this paper.

**Lemma 5.1.** *For $s_i$ in $S$, if $|N(s_i, V^D(S))| < k$, where $|.|$ denotes set cardinality, then $V^D(s_i)$ is not $k$-covered. For $s_i$ in $S$, if $|X(s_i, V^D(S))| < k$, then $V^D(s_i)$ is not $k$-covered.*

This lemma shows how simple Voronoi diagrams can be used to detect the coverage holes based on the distance between the sensor node and the edges of its Voronoi cell. It is based on the concept that the Voronoi tessellation is a partition of the points belonging to the monitored area according to their proximity to the sensor nodes. In other terms, if a point is not detected by the sensor node located at the generator of the Voronoi cell it belongs to, it cannot be detected by any other sensor node. If a sensor detects that the distance to one among the edges of its Voronoi edges is more than its coverage range, it has to move towards this edge to cover the corresponding hole. The uncovered can therefore be gradually reduced using this distributed strategy. However, a sensor node can detect that more than one of its Voronoi neighbors do not fulfill the condition of the lemma, it will therefore move towards the most distant neighbor.

The major advantage of this strategy, with respect to the advanced strategy, is that it relies on simple Voronoi diagrams to deal with $k$-coverage while the advanced model proposed in the previous subsection is based on order $k$ Voronoi tessellations which are more complex to build.

A more accurate comparison between the two models will be carried out in the simulation section.

### *5.3. Multi-Target Tracking*

The two tracking models presented in the above can be extended to the tracking of multiple targets. To describe the extension let us assume, for the sake of clarity, only two targets are detected by sensors in $S$. Let $z_t$ and $z_{t'}$ be the reported positions.

The extension of the simplified model considers two cases:

- Only one node has detected the presence of the two targets: In that case, the sensor keeps monitoring one of the targets and invites the nearest neighbor to the second target to monitor the second and provides it with relevant information it collects.
- More than one node have detected the targets: In that case,two sensor among those that have detected the targets are selected to keep monitoring the targets independently.

On the other hand, the k-mobility model extends in following way: if d nodes detect the targets, these sensors are divided into two subsets, each in charge of monitoring one target, then the subsets are extended so that any of them contains k sensors.

## 6. Complexity Analysis of Coverage Management and Tracking

### *6.1. Complexity*

In this section, we analyze the complexity of the different algorithms we have developed in the previous sections for the detect and locate holes or to repair coverage holes. Our approach to estimate the complexity can be based on the following metrics:

- The number of messages exchanged between the sensors during the execution of the algorithm.
- The number of additions and deletions of simplices to the Vietoris complex.
- The number of sensor movements made during the execution of an algorithm.

Some other operations can be added for a more accurate estimation of complexity. These metrics may include, for example, the number of storing operations made at the node level to update the related data structures. The messages exchanged during the execution of an algorithm can be of different types. In particular, they can be sent to a neighbor to tell it to change its status from internal (to the Vietoris complex) to external (*i.e.*, on the boundary of Vietoris complex). They also can be used to construct the initial boundary of Rips complex, or used to reduce the external boundary. They also can be sent after the retraction or the deflation of a simplex, or they are sent by a leader node the command a coordinated movement of sensors.

For the sake of clarity, we will focus on the complexity on the detection and counting of coverage hole. In this case, let $n$ be the number of sensors in $S$, and $e$ be the number of 1-simplices, $f$ the number of 2-simplices, and $t$ the number 3-simplices in RIPS complex of $S$). Let also $p$ the number of vertices at the initial boundary of the Rips complex.

The number of messages sent during the execution of the algorithm should be lower or equal to the number of messages exchanged if all the polyhedra (external and internal) have been retracted first and that after deflation all the facets have been retracted. In that case, one can state that the number $N_1$ of messages sent is given by:

$$N = p + (f - p) + (e - (f - p)) = p + e \leq |S| + e,$$

where $p$ is the number of external vertices. This result can be deduced from the preceding and the fact that $p \leq |S|$.

Now let us assume, without loss of generality, that the deployment of sensors (initial and current) guarantees that every node in the Rips complex of $S$ has at most $v$ neighbors ($v$ is a fixed value coping with the volume of the area to monitor and the radius of coverage). Then, one can conclude that $e$ is smaller than $v \times |S|$, we deduce that

$$N \leq n + v \times |S| \leq (v + 1)n.$$

Let us notice that the assumption is not mandatory and a direct proof can be given. This shows that the algorithm to detect and count the coverage holes has a linear complexity.

### *6.2. Extending Results*

The results presented in the previous sections can be extended in two dimensions: the type of the sensors and the occurrence of obstacles in the domain under monitoring.

**Coping with obstacles.** The algorithms developed in the preceding sections can be adapted to the occurrence of obstacles. Obstacles in monitored 3D areas may complicate seriously the role of monitoring sensors, increase their power consumption, and limit the coverage efficiency. Two particular objects have to be modified in our algorithms. First, the coverage holes that have to be counted should not contain obstacles. one can assume for this that the sensors are able to recognize an obstacle). Second, the mobility model used to increase coverage or to provide tracking should consider moving the sensor vertically as an alternative.

**Coping with semi spherical.** The algorithms developed in the preceding sections can be extended to semi-spherical sensors (sensor having a semi spherical covered area). It is worth noticing, at this

point, that this type is sufficiently general to represent various sensors-base applications. In particular, the model can be used to represent fire and smoking-based sensors or camera-based sensors. To cope with semi-spherical sensors, one can notice that the concepts of Vietoris-rips and Voronoi diagram can be extended, so that coverage holes can be handles in a similar way. However, when repairing a hole, the mobility model of the sensor should include rotating a sensor to increase the coverage of a specific area by the sensor.

It is worth noticing that the additions made to the developed algorithms do not modify significantly the complexity of the algorithms. In particular, the complexity of the hole counter remains linear (as shown in the simulation discussed in the following section).

## 7. Experimental Results

In this section, we carry out a set of experiments to prove the efficiency of the proposed techniques. We first address the coverage hole problem by evaluating the performance of the higher-order Voronoi-based strategy for coverage optimization. To this purpose, we define a metric representing the ratio of uncovered area with respect to the total area of the monitored region. Second, We assess the target tracking approach by estimating the maximum linear distance that can be made by a hostile target without being detected. Finally, we evaluate the complexity of our coverage control and mobility techniques. We use the number of transmitted messages as a main criterion to estimate this complexity since data transmission consumes much more power than computational steps in WSNs.

### *7.1. Coverage Control and Hole Reduction*

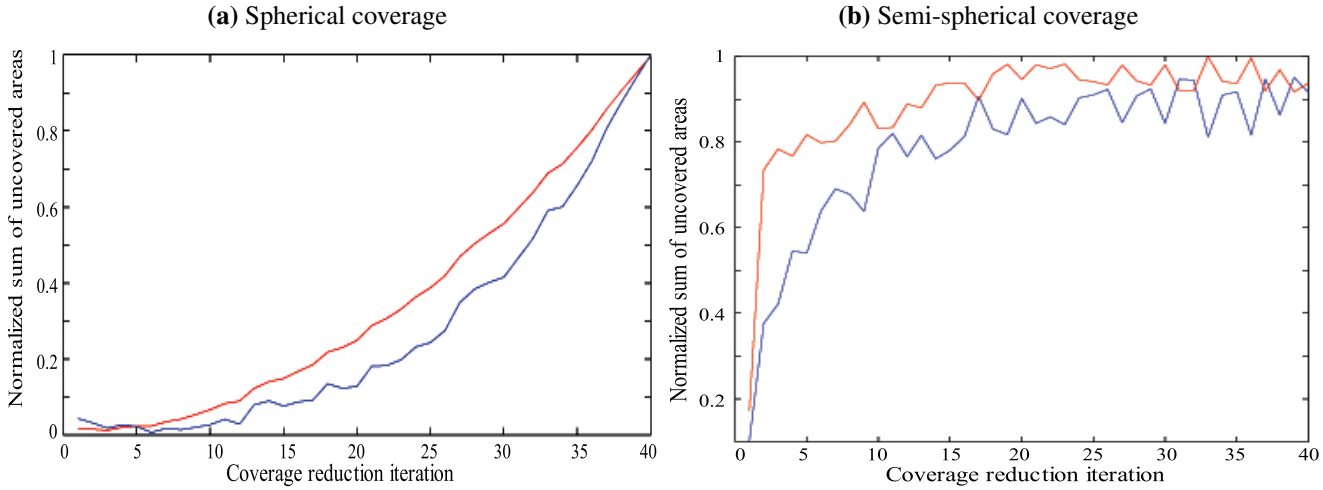
The first experiment aims at evaluating the hole reduction strategy based on three-dimensional Voronoi tessellations with spherical coverage. We define the following metric to evaluate the performance of hole reduction.

$$\mu = \frac{\mathrm{Sum\_of\_hole\_volumes}}{\mathrm{Total\_volume\_of\_the\_monitored\_area}}. \tag{11}$$

Figure 1 shows the evolution of $\mu$ according to the number of iterations of the coverage hole reduction algorithm. We compared the Voronoi-based hole reduction strategy with the Homotopy-based strategy proposed in Section 4. It can be noticed that the increase in terms of normalized uncovered proportion is about 15% when the number of iteration is low. In addition, when the number of iteration exceeds 30, both approaches perform well since the normalized sum of uncovered areas becomes higher than 90%.

A similar experiment is conducted for vector guided sensors, assuming that at every step of the iteration, the mobility is provided along with an orientation of the vector to achieve better coverage. Figure 1 shows the evolution of $\mu$ according to the number of iterations of the coverage hole reduction algorithm and compares it to the Voronoi-based hole reduction strategy with the Homotopy-based strategy. One can conclude that while the homotopy-based approach is less complex, since linear for detection and localization, the Voronoi-based method reaches better results. In addition, a comparison between the results obtained for spherical sensors and semi spherical sensors shows the following:

**Figure 1.** Evolution of the uncovered area proportion according to time.




- The approach performs better with spherical sensors for the first iterations. Indeed, the normalized uncovered proportion reaches 70%, with spherical sensors, after 10 iterations, while it stays under 10% for semi spherical sensors.
- The approach performs the same for both types of sensors after 30 iterations.

This can be explained by the fact that the density of sensors is the same for both types and, therefore, it takes more mobility for semi spherical sensors the coverage holes.

### *7.2. Mobility Modeling*

The Average Linear Uncovered Length (ALUL), denoted by $\mathcal{L}(x,\theta)$, gives an approximation of the average distance that can be made by a target, moving in 3D space, before being detected by the sensor network. The metric $ALUL_m(x,t)$ representing the ALUL in a location $x$ at time $t$ is given by:

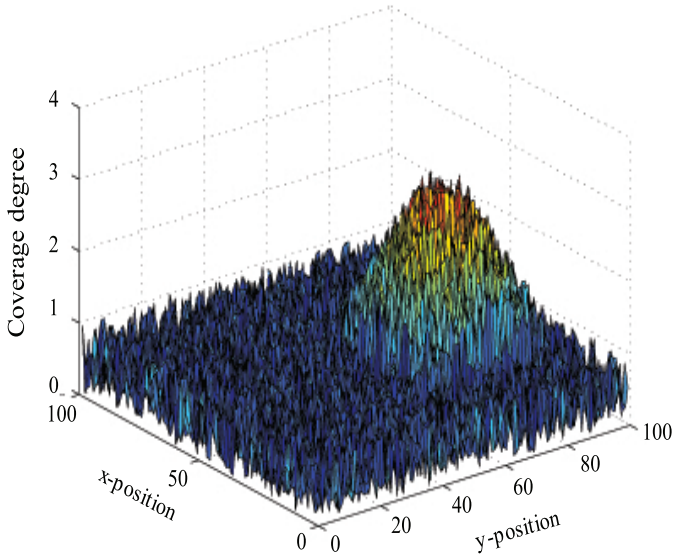
$$ALUL_m(x,t) = \begin{cases} 0, & \text{if } x \text{ is covered by a sensor.} \\ \frac{1}{(2\pi)^2}\int_{-\pi/2}^{\pi/2}\int_{0}^{2\pi}\mathcal{L}(x,\theta_1,\theta_2,t)d\theta_1 d\theta_2, & \text{otherwise.} \end{cases}$$

From the performance evaluation perspective, $ALUL_m(A)$ provides information on the coverage-preserving capabilities of the mobility model and the long-term behavior of the mobility model.

In order to visually illustrate the performance of coverage reduction models, we use the local node density distribution that gives the number of sensors that cover every point of the monitored region. Figure 2 shows that, in the simple context where one target is moving within a 100 m$^2$-size monitored zone, the coverage degree considerably varies according proximity to the mobile target. In fact, after 5 mobility steps, the local sensor density is less than 1 in regions that are far from the target location (which is (70,30)) and reaches 2.7 in points that are close to this target.

More interestingly, Figure 3 addresses the case where two targets are present within the region of interest. We notice that the sensors are initially uniformly distributed. The density then increases for the three following iterations in the regions where the targets are. In fact, this proves that our tracking scheme is precise enough to distinguish between the two different targets.

**Figure 2.** Local node density distribution after 5 mobility iterations.




To confirm these results, we used the $\text{ALUL}_m$ metric to evaluate the evolution of the uncovered area with respect to time. In fact, this allows to know whether uncovered regions are created due to the density increase in the zones that are close to the target. We compared our scheme to four known mobility models, which are: the random walk model, the random waypoint model, the random direction model, and the Gauss-Markov model. The results of this comparison are depicted in Figure 4.

We notice that the proposed mobility models, denoted by Advanced Voronoi-based Mobility Model (AVBMM) and Distributed Voronoi-based Mobility Model (DVBMM), clearly outperform the existing models. They also return a better performance than the Density-Preserving Mobility Model. This is because the latter model, despite its ability to guarantee a nearly uniform node density within the monitored area, does not take into account the presence of hostile targets in the zone of interest.

### *7.3. Complexity Evaluation*

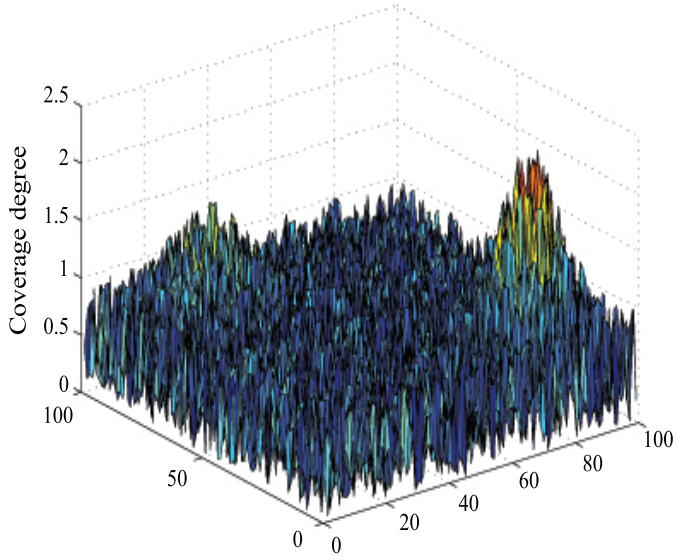
In this subsection, we evaluate the communication overhead resulting from the proposed retract-based coverage control approach. To this end, we only consider the complexity of the detection and localization steps in our algorithm and do not address the complexity of the repair step, since the repair step complexity is mainly dependent on the first deployment. However, one can easily deduce that if the deployment guarantees that holes size do not exceed a threshold, then the linear complexity can be verified.

We considered that the dimensions of the monitored zone are 10 m $\times$ 10 m $\times$ 3 m. We varied the number of nodes deployed within this zone and we measured the number of messages required to setup our coverage control protocol. We first supposed that all sensor nodes have a spherical coverage of range 0.5 m. Figure 5 depicts the number of messages for densities ranging from 0.5 sensors per $\text{m}^2$ to 5 sensors per $m^2$. The major remark is that this number is nearly linear with respect to the number of sensors per area unit.

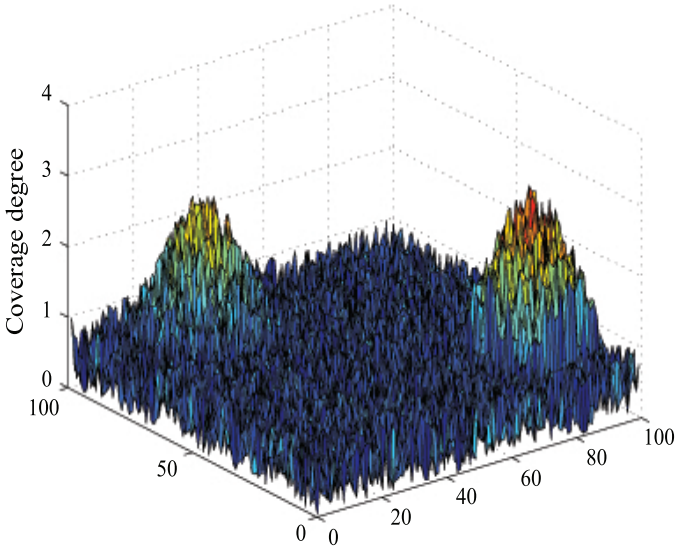
Moreover, we considered the case where sensors have semi-spherical coverage (with the same range). Figure 5 shows that the communication overhead is also linear in this situation but with a smoother slope.

**Figure 3.** Illustration of 3 mobility iterations for a context where two targets are considered.

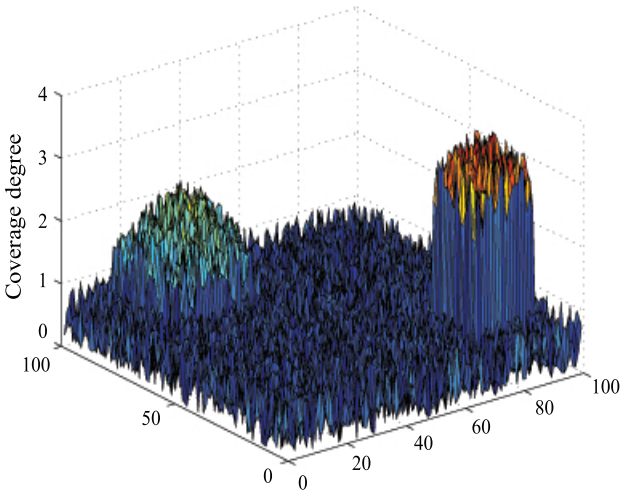
**(a)** Iteration 2




**(b)** Iteration 3




**(c)** Iteration 4




One can deduce the following statements from the aforementioned figures:

- The number of exchanged messages is independent of the density. It is close to 4 for the spherical sensors and 10 for semi spherical sensors. This fact may appear strange; however, one can notice that when a deployment is performed, the detection and location will only search for holes surrounded by the Vietoris space. The latter is reduced when the density is low.

- The number of messages exchanged by the semi spherical sensors for detection and localization is 2.5 times higher than the number observed for spherical. Two reasons can be mentioned for this. First, the area covered by a semi spherical sensors is half the area covered by spherical sensors. Second, the guiding vectors is randomly oriented.

**Figure 4.** Evolution of the $\text{ALUL}_m$ metric according to time.

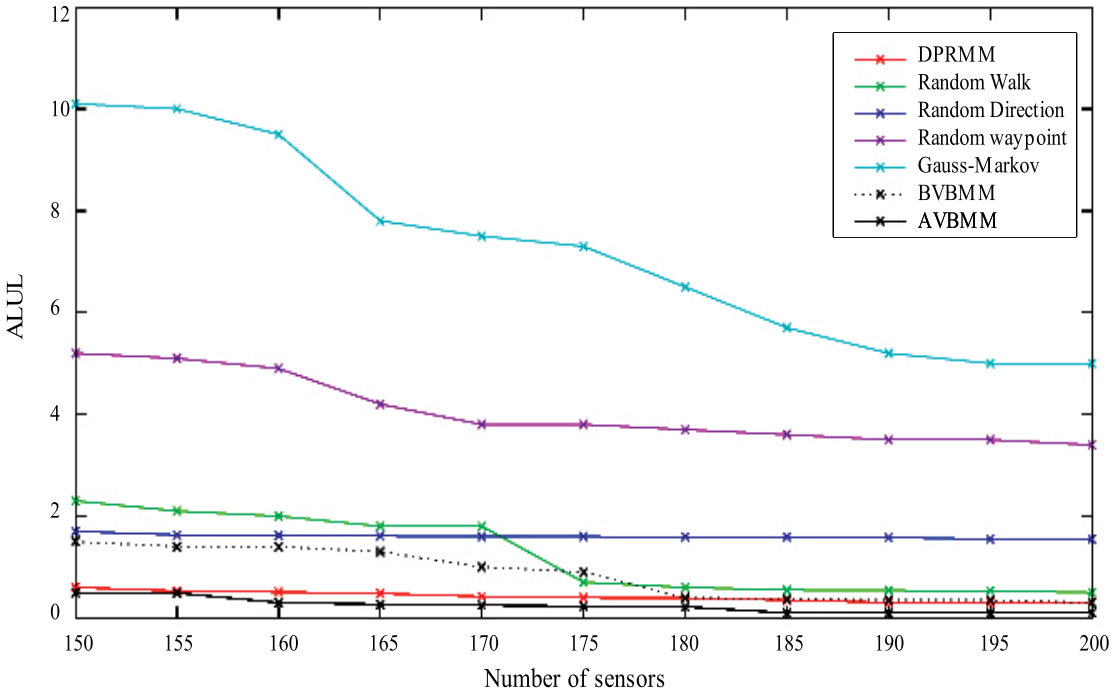
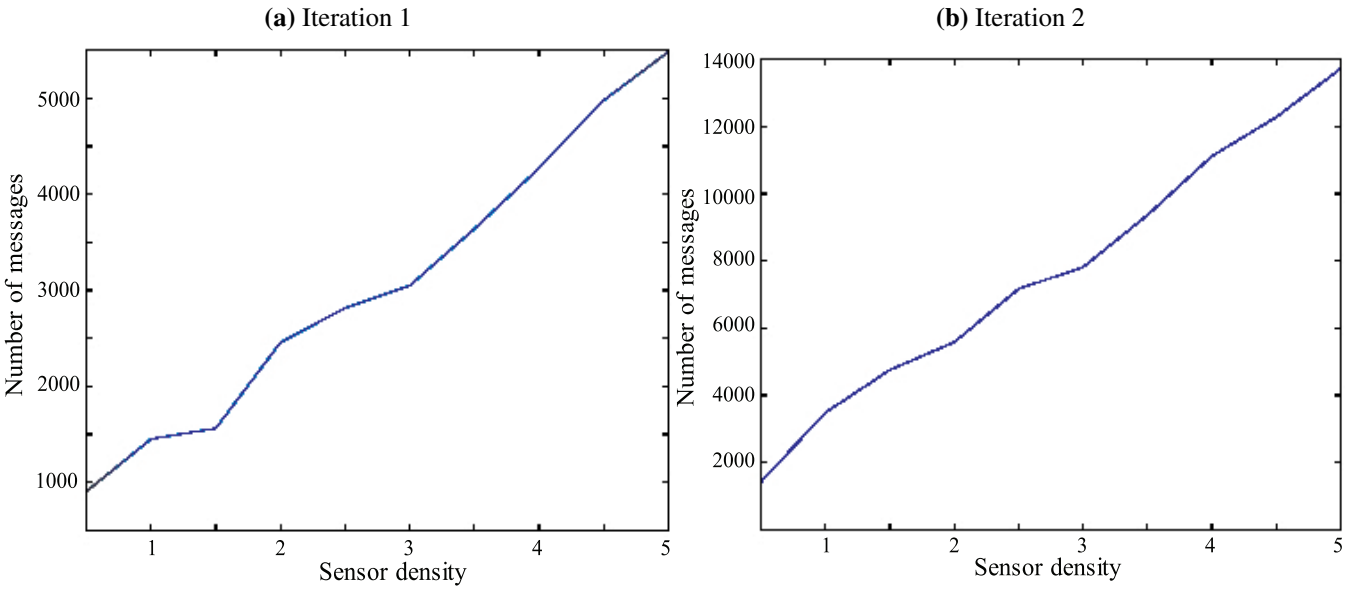



**Figure 5.** Illustration of complexity.




## 8. Conclusion

This paper developed a low complexity approach to detect and localize sensing holes in 3D spaces. It also constructed efficient algorithms to repair holes and track (multiple targets). Our approach has built on two concepts, the Vietoris complex and the Voronoi diagram, and demonstrated that the technique called retraction by deformation achieves low complexity algorithms for the detection of coverage holes in WSNs.

Our approach can be easily extended to more general sensors, for which the Vietoris complex and the Voronoi diagram can be defined. Such sensors can be called conical sensors or vector guided sensors and can represent camera sensors.

## References


1. Gui, C.; Mohapatra, P. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (Mobicom 04)*, Philadelphia, PA, USA, 26 September–01 October 2004; pp. 129-143.
2. Liu, B.; Towsley, D. A study of the coverage of large-scale sensor networks. In *Proceedings of 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, FL, USA, 25–27 October 2004; pp. 475-483.
3. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization in distributed sensor networks. *ACM Trans. Embed. Comput. Syst.* **2004**, *3*, 61-91.
4. Fekete, S.P.; Kroller, A.; Pfisterer, D.; Fischer, S.; Buschmann C. Neighborhood-based topology recognition in sensor networks. In *Proceedings of ALGOSENSORS*, Turku, Finland, July 2004; pp. 123-136.
5. Kuo, S.-P.; Tseng, Y.-C.; Wu, F.-J.; Lin, C.-Y. A probabilistic signal-strength-based evaluation methodology for sensor network deployment. In *Proceedings of 19th International Conference on Advanced Information Networking and Applications, AINA 2005*, Taipei, Taiwan, 28–30 March 2005; Volume 1, pp. 319-324.
6. Ren, S.; Li, Q.; Wang, H.; Chen, X.; Zhang, X. A study on object tracking quality under probabilistic coverage in sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.* **2005**, *9*, 73-76.
7. Muhammad, A.; Jadbabaie, A. Dynamic coverage verification in mobile sensor networks via switched higher order Laplacians. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
8. Tahbaz-Salehi, A.; Jadbabaie, A. Distributed coverage verification in sensor networks without location information. In *Proceedings of 47th IEEE Conference on Decision and Control, CDC 2008*, Cancun, Mexico, 9–11 December 2008; pp. 4170-4176.
9. Corke, P.; Peterson, R.; Rus, D. Localization and navigation assisted by networked cooperating sensors and robots. *Int. J. Rob. Res.* **2005**, *24*, 771-786.
10. Zhang, H.; Hou, J. Maintaining sensing coverage and connectivity in large sensor networks. *Wirel. Ad Hoc Sensor Netw.* **2005**, *1*, 89-123.
11. Hsin, C.-F.; Liu, M. Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms. In *Proceedings of The 3rd International Symposium on Information Processing in Sensor Networks, IPSN'04*, New York, NY, USA, April 2004; pp. 433-442.
12. Abdelkader, M.; Hamdi, M.; Boudriga, N. Using higher-order Voronoi tessellations for WSN-based target tracking. In *Proceedings of the IASTED International Symposium, Distributed Sensor Networks, DSN'08*, Calgary, AB, Canada, 21–24 September 2008; pp. 430-435.

13. Meguerdichian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of INFOCOM 2001 Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, AK, USA, 22–26 April 2001; Volume 3, pp. 1380-1387.
14. Li, X.-Y.; Wan, P.-J.; Frieder, O. Coverage in wireless *ad hoc* sensor networks. *Comput. IEEE Trans.* **2003**, *52*, 753-763.
15. Huang, C.-F.; Tseng, Y.-C. The coverage problem in a wireless sensor network. In *Proceedings of The 2nd ACM International Conference on Wireless Sensor Networks and Applications, WSNA'03*, San Diego, CA, USA, 19 September 2003; pp. 115-121.
16. Koskinen, H. On the coverage of a random sensor network in a bounded domain. In *Proceedings of the 16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp, Belgium, 31 August–02 September 2004; pp. 11-18.
17. Ghrist, R.; Muhammad, A. Coverage and hole-detection in sensor networks via homology. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN'05*, Los Angeles, CA, USA, 25–27 April 2005; pp. 254-260.
18. Gramain, A. *Topology of Surfaces*; BCS Associates: Moscow, ID, USA, 1984.
19. Hatcher, A. *Algebraic Topology*; Cambridge University Press: Cambridge, UK, 2001.
20. De Silva, V.; Ghrist, R. Coverage in sensor networks via persistent homology. *Alg. Geom. Topology* **2007**, *7*, 339-358.
21. De Silva, V.; Ghrist, R. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *Int. J. Rob. Res.* **2006** *25*, 1205-1222.