



<b>Title</b>	<b>Visibility-based coverage of mobile sensors in non-convex domains</b>
<b>Author(s)</b>	<b>Lu, L; Choi, YK; Wang, WP</b>
<b>Citation</b>	<b>The 8th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD2011), Qingdao, China, 28-30 June 2011. In Proceedings of the 8th ISVD, 2011, p. 105-111</b>
<b>Issued Date</b>	<b>2011</b>
<b>URL</b>	<b><a href="http://hdl.handle.net/10722/140005">http://hdl.handle.net/10722/140005</a></b>
<b>Rights</b>	<b>International Symposium on Voronoi Diagrams in Science and Engineering. Copyright © IEEE.</b>

## Visibility-based Coverage of Mobile Sensors in Non-convex Domains

Lin Lu<sup>\*†</sup>, Yi-King Choi<sup>†</sup> and Wenping Wang<sup>†</sup>

<sup>\*</sup>*School of Computer Science and Technology, Shandong University, China*

<sup>†</sup>*Dept. of Computer Science, The University of Hong Kong*

<sup>‡</sup>*Email: lulin.linda@gmail.com, {ykchoi, wenping}@cs.hku.hk*

**Abstract**—The area coverage problem of mobile sensor networks has attracted much attention recently, as mobile sensors find many important applications in remote and hostile environments. However, the deployment of mobile sensors in a non-convex domain is nontrivial due to the more general shape of the domain and the attenuation of sensing capabilities caused by the boundary walls or obstacles. We consider the problem of exploration and coverage by mobile sensors in an unknown non-convex domain. We propose the definition of “visibility-based Voronoi diagram” and extend the continuous-time Lloyd’s method, which only works for convex domains, to deploy the mobile sensors in the unknown environments in a distributed manner. Our simulations show the effectiveness of the proposed algorithms.

**Keywords**—mobile sensor networks; sensor deployment; sensor coverage; distributed control

### I. INTRODUCTION

A mobile sensor network is composed of a distributed collection of nodes, each of which has sensing, computation, communication and locomotion capabilities [13]. It is a kind of wireless sensor networks (WSN), which consists of mobile sensors rather than static sensors. A mobile sensor node can be seen as a sensor equipped with a locomotive platform, and thus it can move around after initial deployment. Benefiting from this feature, mobile sensor networks have been widely used in remote and hostile environments where sensors cannot be manually deployed or air-dropped.

The coverage problem is a fundamental problem in WSNs and has been attracting a great deal of research attention these years. Coverage is a measure of the quality of service (QoS) of the sensing function [15] which indicates how well the sensors observe the sensing field. Compared with static nodes, the coverage of a mobile sensor network is more challenging as it depends on both the initial network configurations and the mobility behavior of the sensors. Note that in robotics the definition of the coverage problem may be different; it aims to command the robots to move over all points in the target environment [7] and is not the interests of this work.

Cardei and Wu [6] classified the coverage problem into three types based on different applications: area coverage, point coverage and barrier coverage, among which the area coverage problem receives most attention. The objective of

area coverage is to ensure that every point in the sensing field is monitored by the sensing ranges, or to maximize the sensor detection probability of random events in the environment. We shall use the word “coverage” to refer to area coverage which is the major focus in this work. Regarding their implementations, coverage algorithms can generally be categorized into centralized and distributed methods. For centralized methods, there is usually a centralized controller to issue control laws to drive the movements of the sensors. It has a prerequisite that the sensing field is known a priori. While in distributed methods, a sensor determines its motion depending on its neighborhood information. The entire environment could be unknown, which is a very common setting for a real-life application.

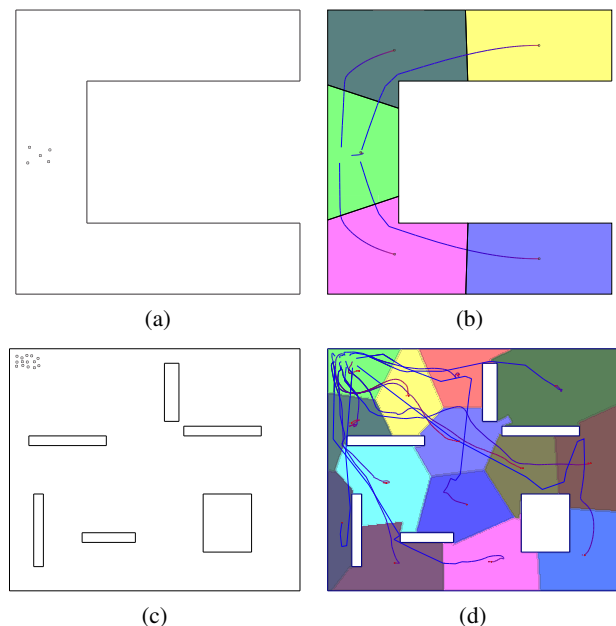


Figure 1. Coverage for mobile sensors in non-convex domains. (a) and (c) the initial configurations of 5 and 14 sensors in a U-shaped domain and a rectangular domain with 6 obstacles, respectively; (b) and (d) the coverage result of the deployment of the sensors. The trajectories are shown in connected lines.

In this paper, we shall deal with the problem of exploration and coverage problem by mobile sensors in an

unknown non-convex domain. We assume that the following sensor properties:

- (P1) each sensor is equipped with unlimited range omnidirectional communication devices so that it can exchange information with other sensors;
- (P2) each sensor is equipped with omnidirectional monitoring devices, such as a camera, whose sensing range is unlimited but however attenuates to zero at an obstacle or a boundary. In effect, a sensor can cover (or monitor) points along its lines of sights without any obstruction.

There is also a cost associated with each sensor for monitoring the region that it is responsible to. Our aim is to deploy the mobile sensors to explore the unknown domain as much as possible, and at the same time minimizing the total monitoring costs. See Figure 1 for an illustration. The main contribution of this work is the introduction of the “visibility-based Voronoi diagram” which takes into account the visible regions of the sensors, thereby solving the above mobile sensors coverage problem in a non-convex domain.

The remainder of this paper is organized as follows. We first review the previous work in Section II. The problem formulation of the mobile sensor coverage problem in both convex and non-convex domains is then given in Section III. In Section IV, we describe the main algorithms of deploying the mobile sensors. Simulations and discussions are presented in Section V. Finally, we conclude the paper in Section VI.

## II. PREVIOUS WORK

We classify the review of the work on mobile sensor coverage based on whether the domain under considerations is convex or not.

### A. Convex environments

Most work in the literature on mobile sensor coverage deals with the convex domains [10], [9], [21], [20], [19], [1]. Here, we focus only on the work by Cortes *et al.* [10] which proposed a distributed method based on Voronoi partition and Lloyd’s method, from which many subsequent research studies including this work follow upon. In [10], the sensors are assumed to move according to first order dynamics and it is proved that the locations of the sensors converge to a centroidal Voronoi configuration which is also a local optimal coverage with respect to an objective function related to the sensing performance. Their algorithm, assuming that the sensing field is a simple convex polygon, cannot be directly adapted to a non-convex environment with obstacles for the following two reasons. Firstly, the sensing capability may be attenuated due to obstacles, which renders the Euclidean distance based sensing model invalid. Secondly, the centroid of the Voronoi region serving as the target position of a sensor is not guaranteed to be reachable, e.g., when the centroid lies outside the domain.

### B. Non-convex environments

Caicedo-Nunez and Zefran [5], [4] transformed the non-convex domain to a convex one through a diffeomorphism, and solved the problem by applying the convex domain methods. They assume that the domain is either simply-connected or convex with obstacles inside it. However, the algorithm may get stuck at the so-called “orthogonality point” even though the target centroid is reachable, as remarked in [4]. Howard *et al.* [13] presented an approach based on potential field, in which the sensors are treated as virtual particles subjected to virtual forces. The virtual forces repel the nodes from each other and from obstacles, so that the sensors may spread out to maximize coverage area. Renzaglia and Martinelli [18] extended this method by adding an attractive potential from the centroid of the sensor’s Voronoi region. Pimenta *et al.* [17] replaced the Euclidean distance with geodesic distance and computed the generalized gradient to produce the control law. However, the geodesic Voronoi diagram is expensive to compute and this method will cause the sensors to get trapped in a saddle point or be driven into an obstacle [3]. Breitenmoser *et al.* [3] formulated the coverage problem in non-convex environments with the same objective function as used by Cortes *et al.* [10], with the constraints that each sensor must stay inside the feasible area. Their algorithm combines Lloyd’s method and path planning algorithms for computing the movement of the sensors. The sensing attenuation caused by the obstacles or boundaries is not considered, meaning that the sensors can “see” through obstacles. Li and Cassandras [14] defined an objective function based on a probabilistic model representing a sensor’s detecting ability and proposed a gradient-based motion control scheme to maximize the objective function. Zhong and Cassandras [22] extended their work to the mission space with obstacles, and the gradient is computed via discretization of the mission space. This method does not guarantee the optimal coverage at convergence and it may happen that two sensors remain very close to each other.

All the above methods assume that the environment is known, and some allow also the existence of unknown obstacles (e.g., [18]). Ganguli *et al.* [12], on the other hand, studied the art gallery problem on an unknown non-convex domain (without holes), and solved it via a visibility-based distributed deployment algorithm, so that every point in the environment is “visible” by at least one guard with an unlimited sensing range. This problem setting is similar to ours, except that the monitoring cost is not considered in [12]. Bhattacharya *et al.* [2] computed the geodesic Voronoi tessellation in feedback control laws using the discrete representation of the unknown non-convex environment and extended the continuous-time Lloyd’s method with entropy-based metrics to deploy the sensors. However, the attenuation of sensing capabilities due to occlusion is not

taken into account.

### III. PROBLEM FORMULATION

Given  $n$  mobile sensors  $S = \{\mathbf{s}_i\}_{i=1}^n$  in a 2D polygonal domain  $\Omega \subset \mathbb{R}^2$  whose boundary is a simple polygon, that is, without self-intersection, and  $m$  obstacles  $\{O_i\}_{i=1}^m \subset \Omega$ , each of which is also a simple polygon, we define  $I$  as the *feasible domain*, a subspace of  $\Omega$ , that is,  $I = \Omega \setminus (O_1 \cup \dots \cup O_m)$ . We also assume that the sensors have the two properties (P1) and (P2) as given in Section I. While (P1) allows for communication between “neighbouring” sensors and therefore renders a distributed control possible, (P2) implies that we must take into consideration a visible region for each sensor in a non-convex environment which will shortly be discussed in Section III-B. We first briefly introduce the Voronoi-based formulation of the coverage problem for convex domains.

#### A. Convex environments

Suppose that  $\Omega$  is convex and  $m = 0$ , then we have  $I = \Omega$ . The coverage problem is defined as the optimization of the objective function [10]:

$$\mathcal{H}(S, \mathcal{W}) = \sum_{i=1}^n \int_{W_i} f(\|\mathbf{q} - \mathbf{s}_i\|) \phi(\mathbf{q}) \, d\mathbf{q}, \quad (1)$$

where  $\|\cdot\|$  denotes the Euclidean distance function,  $\phi(\cdot)$  a distribution density function defined in  $I$ ,  $d\mathbf{q}$  the differential area element of  $\Omega$ , and  $W_i$  the so-called “dominance region” over which sensor  $\mathbf{s}_i$  is responsible for measurements and  $\mathcal{W} = \{W_i\}_{i=1}^n$ . Also,  $\cup_{i=1}^n W_i = \Omega$  and the interior of  $W_i$ 's is disjoint. We have  $f(\cdot)$  as a function of distance for a quantitative assessment of the sensing performance, e.g., a larger distance from a point to a sensor induces a higher cost. We use the common setting  $f(\|\mathbf{q} - \mathbf{s}_i\|) = \|\mathbf{q} - \mathbf{s}_i\|^2$  as in [10].

The function  $\mathcal{H}$  is to be minimized with respect to both the locations of the sensors  $S$  and the assignment of the dominance regions  $\mathcal{W}$ . We can easily see that the optimal partition of  $I$  by fixing the locations of the sensors is the Voronoi partition. Hence, Equation (1) can be rewritten as

$$\mathcal{H}_{\mathcal{V}}(S) = \sum_{i=1}^n \int_{V_i} \|\mathbf{q} - \mathbf{s}_i\|^2 \phi(\mathbf{q}) \, d\mathbf{q}, \quad (2)$$

where  $V_i$  stands for the Voronoi region of  $\mathbf{s}_i$  in  $I$  which is defined as

$$V_i = \{\mathbf{q} \in I \mid \|\mathbf{q} - \mathbf{s}_i\| \leq \|\mathbf{q} - \mathbf{s}_j\|, \forall j \neq i\},$$

and  $\mathcal{V} = \cup_{i=1}^n V_i$ . It is clear that  $\mathcal{H}_{\mathcal{V}}(S)$  in Equation (2) is essentially the centroidal Voronoi tessellation (CVT) energy function [11], and therefore a minimizer of  $\mathcal{H}_{\mathcal{V}}(S)$  corresponds to a CVT in which each sensor location coincides with the centroid of its Voronoi region.

#### B. Non-convex environments with obstacles

We now consider the case where  $\Omega$  is non-convex and the number of obstacles is  $m > 0$ . Sensor property (P2) follows that each sensor  $\mathbf{s}_i$  is associated with a *visibility polygon*, denoted by  $\text{Vis}(\mathbf{s}_i) \subseteq I$ , which is the set of points in  $I$  that are visible from  $\mathbf{s}_i$ , or equivalently,

$$\text{Vis}(\mathbf{s}_i) = \{\mathbf{p} \in I \mid \overline{\mathbf{s}_i \mathbf{p}} \subset I\},$$

where  $\overline{\mathbf{s}_i \mathbf{p}}$  represents a line segment connecting  $\mathbf{s}_i$  and a point  $\mathbf{p}$ .

We then define the objective function of the sensor coverage problem in non-convex domains as

$$\bar{\mathcal{H}}(S, \mathcal{W}) = \sum_{i=1}^n \int_{W_i} g(\mathbf{q}, \mathbf{s}_i) \phi(\mathbf{q}) \, d\mathbf{q}, \quad (3)$$

where

$$g(\mathbf{q}, \mathbf{s}_i) = \begin{cases} \|\mathbf{q} - \mathbf{s}_i\|^2 & \text{if } \mathbf{q} \in \text{Vis}(\mathbf{s}_i), \\ \infty & \text{otherwise.} \end{cases} \quad (4)$$

It can be seen that the objective function in Equation (3) is essentially the same as Equation (1) for convex environments. On the other hand, in a non-convex environment, with the consideration of the sensor visible regions, the optimal partitioning by the dominance regions  $\mathcal{W}$  with respect to Equation (3) is no longer a CVT.

### IV. DISTRIBUTED MOBILE SENSOR COVERAGE ALGORITHMS

In this section, we will describe our algorithms for distributed coverage. We will first present a basic coverage strategy, discuss its properties and then further devise an improved strategy.

#### A. Visibility-based Voronoi diagram

Intuitively, in order to achieve the optimal coverage in terms of the objective function given by (3), a point  $\mathbf{x} \in I$  should be assigned to be covered by the closest sensor to which  $\mathbf{x}$  is visible. Hence, we introduce the notion of *visibility-based Voronoi diagram* as follows:

**Definition 1.** The visible Voronoi region of  $\mathbf{s}_i$ , denoted by  $\bar{V}_i$ , is defined as:

$$\bar{V}_i = \{\mathbf{q} \in \text{Vis}(\mathbf{s}_i) \mid \|\mathbf{q} - \mathbf{s}_i\| \leq \|\mathbf{q} - \mathbf{s}_j\|, \forall \mathbf{s}_j, \mathbf{q} \in \text{Vis}(\mathbf{s}_j)\}.$$

It is clear that the interior of all  $\bar{V}_i$ 's is disjoint.

**Definition 2.** The visibility-based Voronoi diagram (VVD)  $\bar{\mathcal{V}}$  of the sensors  $S$  is the set of visible Voronoi regions of all sensors, that is,  $\bar{\mathcal{V}} = \{\bar{V}_i\}_{i=1}^n$ .

We remark here that the given number of sensors may not be enough to cover all points in  $I$ . We therefore define the *visible area* of  $I$  by a set of sensors  $S$ , denoted by  $\text{Vis}(S)$ , as the set of points that are visible to at least one sensor in  $S$ . Clearly, the visible area is the union of all visibility

polygons, that is,  $\text{Vis}(S) = \cup_{i=1}^n \{\text{Vis}(s_i)\}$ . Also, we have  $\text{Vis}(S) \subseteq I$ . It follows from Definition 1 that  $\bar{V}_i \subseteq \text{Vis}(s_i)$ , and since every point in  $\text{Vis}(S)$  must belong to some  $\bar{V}_i$  in  $\bar{\mathcal{V}}$ , the VVD  $\bar{\mathcal{V}}$  is a partition of the visible area  $\text{Vis}(S)$ . See Figure 2 for two examples.

It is also worth noting that by Definition 1, a visible Voronoi region of a certain sensor  $s_i$  can be disconnected (Figure 3a). However, it is generally more desirable for the monitoring zone of a sensor to be of a single connected component in practice. Hence, we consider a variant of Definition 1 in which the visible Voronoi region of  $s_i$  is confined to be the connected component of  $\bar{V}_i$  which contains  $s_i$ . We shall refer to this restricted definition of visible Voronoi region in the sequel. Note that under this new definition,  $\bar{\mathcal{V}}$  is no longer a partition of  $\text{Vis}(S)$  and  $\cup_{i=1}^n \{\bar{V}_i\} \subseteq \text{Vis}(S)$  (Figure 3b).

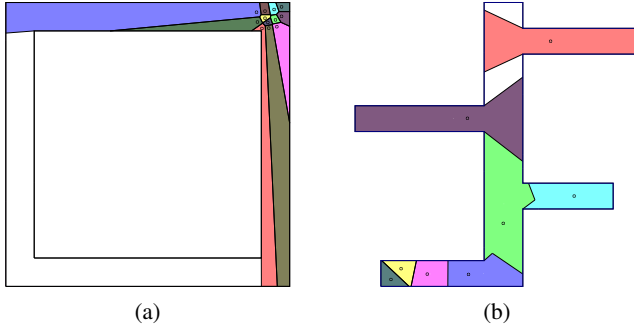


Figure 2. Two examples of the visibility-based Voronoi diagrams with given sensors (black dots). The colored region is the visible Voronoi region of its corresponding sensor.

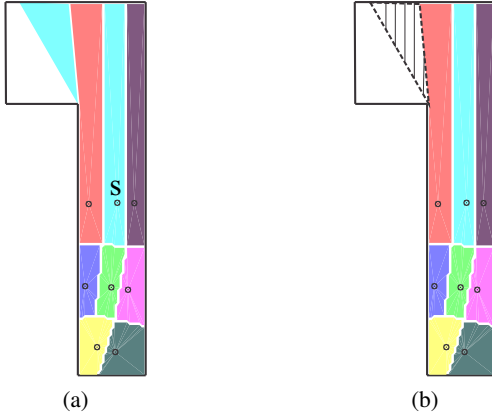


Figure 3. (a) The VVD by Definition 1 allows disconnected visibility-based Voronoi region, e.g., that of sensor  $s$  colored in cyan. (b) The VVD with the restriction that each visibility-based Voronoi region must be connected. The shaded triangle is the area which is part of the visible area but does not belong to any of the visibility-based Voronoi region.

In a distributed network setting, the VVD can be determined using a similar idea as in [2] for computing a geodesic Voronoi tessellation, but without the need of performing any Dijkstra's search in our case. Specifically, the feasible

domain  $I$  is first uniformly discretized into a set of points  $\mathcal{P}$ , and each sensor will compute its distance to every point in  $\mathcal{P}$  that falls within its visibility polygon. Assuming unlimited communication among the sensors, a point  $\mathbf{p}$  in  $\mathcal{P}$  will then be assigned to the sensor with a shortest distance to  $\mathbf{p}$ , and the VVD is thus determined.

### B. Distributed coverage strategies

The basic VVD-based distributed coverage strategy is given as follows:

---

#### Algorithm SC1.0:

INPUT: An initial configuration of  $n$  sensors  $S = \{s_i\}_{i=1}^n$  in an unknown 2D domain  $I$ .

OUTPUT: A visibility-based coverage of  $S$  in  $I$ .

STEPS:

- 1) Compute the visibility-based Voronoi diagram  $\bar{\mathcal{V}} = \{\bar{V}_i\}_{i=1}^n$  for all sensors.
  - 2) For each  $\bar{V}_i$ , compute its centroid  $\mathbf{c}_i$ . If  $\mathbf{c}_i \notin \bar{V}_i$ , project  $\mathbf{c}_i$  to  $\bar{V}_i$  by replacing  $\mathbf{c}_i$  with its closest point in  $\bar{V}_i$ , that is,  $\mathbf{c}_i \leftarrow \arg \min_{\mathbf{x} \in \bar{V}_i} \|\mathbf{x} - \mathbf{c}_i\|$ .
  - 3) Move each sensor  $s_i$  by the first-order dynamic behavior  $\dot{\mathbf{s}}_i = -\lambda(\mathbf{s}_i - \mathbf{c}_i)$ , where  $\lambda$  is a pre-defined value in  $(0, 1]$ .
  - 4) If no sensor is moved in Step 3, return  $S$ ; otherwise goto Step 1.
- 

Similar to the continuous-time Lloyd's method used for convex domains [10], our algorithm drives each sensor towards the centroid of its visibility Voronoi region. The constant  $\lambda$  serves as the velocity of the sensor movement. It is also possible that each sensor moves at a different speed  $\lambda_i$ .

The centroid  $\mathbf{c}_i$  of  $\bar{V}_i$  can be computed by

$$\mathbf{c}_i = \frac{\int_{\mathbf{q} \in \bar{V}_i} \phi(\mathbf{q}) \mathbf{q} d\mathbf{q}}{\int_{\mathbf{q} \in \bar{V}_i} \phi(\mathbf{q}) d\mathbf{q}}.$$

In Step 2 of the algorithm, when a centroid is found not to fall within the corresponding visible Voronoi region, it will be projected to its closest point in the region. This is to make sure that a sensor always moves within its visible Voronoi region. We note here that the projection we used is not exclusive, and one may use other schemes for this purpose. The proof of the convergence of algorithm SC1.0 remains to be established. Nevertheless, when given enough number of sensors, the algorithm terminates in all our experiments with each sensor  $s_i$  coincides with the centroid or projected centroid of its visible Voronoi region  $\bar{V}_i$ .

Algorithm SC1.0 focuses mainly on achieving a low cost coverage by having the sensors be moved to the centroid of its visibility Voronoi region in each iteration. However, this may limit the ability of the sensor in terms of exploring the unknown domain and hence we aim to improve this basic

algorithm in the following. We first classify the boundary edge of a visible Voronoi region  $\bar{V}_i$  into the following three types:

- (A) a bisector between  $s_i$  and any of its neighboring sensors,
- (B) an edge lying on the boundary of the feasible domain  $I$ , or
- (C) an edge of  $\text{Vis}(s_i)$  which is not lying on the boundary of  $I$ .

It is apparent that the existence of a Type C edge in a VVD indicates an unexplored region in  $I$ . In other words, if the edges of all  $\bar{V}_i$ 's are of Type A or Type B, we are done and the entire feasible domain is covered, that is,  $\cup_{i=1}^n \{\bar{V}_i\} = I$ . Moreover, if  $\bar{\mathcal{V}}$  is the same as the Voronoi diagram of  $S$  in  $I$  at the same time, the resulting configuration is a minimizer of  $\bar{\mathcal{H}}(S, W)$ , as is shown in the example in Figure 4(a). Otherwise, unexplored region still exists. Figure 4(b) shows an example in which each sensor coincides with the centroid of its visible Voronoi region, but yet this configuration is far from optimal because there is still unexplored region which can be detected by the existence of a Type C edge in the visible Voronoi region of  $s_1$ . We also call those sensors with a Type C edge the *interfacing sensors*.

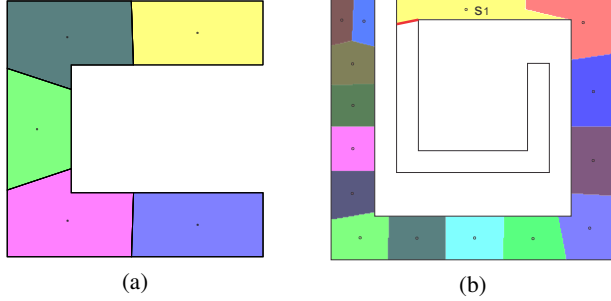


Figure 4. Examples of the stopping configurations of algorithm SC1.0. (Note: We use constant density function in all of the examples for simplicity.)

Algorithm SC2.0 is an improved version of SC1.0 which takes into account the possible unexplored region in the unknown domain:

---

**Algorithm SC2.0:**

INPUT: An initial configuration of  $n$  sensors  $S = \{s_i\}_{i=1}^n$  in an unknown 2D domain  $I$ .

OUTPUT: A visibility-based coverage of  $S$  in  $I$ .

STEPS:

- 1) Compute the visibility-based Voronoi diagram  $\bar{\mathcal{V}} = \{\bar{V}_i\}_{i=1}^n$  for all sensors.
- 2) For each  $\bar{V}_i$ , compute its centroid  $c_i$ . If  $c_i \notin \bar{V}_i$ , project  $c_i$  to  $\bar{V}_i$  by replacing  $c_i$  with its closest point in  $\bar{V}_i$ , that is,  $c_i \leftarrow \arg \min_{x \in \bar{V}_i} \|x - c_i\|$ .
- 3) For each interfacing sensor  $s_i$  with Type C edges  $\mathcal{E}$ , pick an edge  $e \in \mathcal{E}$  and compute the shortest path

$\tau$  from  $c_i$  to the midpoint of  $e$ , and then set  $c_i$  to be the midpoint on  $\tau$ .

- 4) Move each sensor  $s_i$  by the first-order dynamic behavior  $\dot{s}_i = -\lambda(s_i - c_i)$ , where  $\lambda$  is a pre-defined value in  $(0, 1]$ .
- 5) If no sensor is moved in Step 4, go to Step 6; otherwise goto Step 1.
- 6) If there is no interfacing sensors, return  $S$ ; otherwise, for each interfacing sensor  $s_i$  with Type C edges  $\mathcal{E}$ , pick an edge  $e_i \in \mathcal{E}$ , move  $s_i$  to the midpoint of  $e_i$ , then goto Step 1.

---

As one can see, the main differences between SC2.0 and SC1.0 are in steps 3 and 6 for dealing with interfacing sensors. There are still some algorithmic details regarding SC2.0:

- Several criteria are viable to pick an edge when the visible Voronoi region of an interfacing sensor has more than one Type C edge, such as (a) the longest one, (b) the farthest one, or (c) the one that leads to the smoothest path according to the previous movement of the sensor. As smooth trajectories save the energy of a sensor, we use criteria (c) in our simulations.
- In Step 3, the shortest path  $\tau$  from the centroid  $c_i$  to the midpoint  $q$  of a Type C edge  $e$  so as to ensure that the subsequently computed target position is always reachable by the corresponding sensor. One can take any point  $p$  along  $\tau$  to serve as the target position for a sensor, but a more greedy selection of  $p$  towards  $q$  may introduce new uncovered area. We opt to use the midpoint on  $\tau$  which works well in our simulation.
- Similarly in Step 6, when there is still unexplored region but the sensors cannot be driven further, the interfacing sensors are forced to move directly to the midpoint of a Type C edge in order to reveal more unexplored region. This, however, may also risk the possibility of causing new uncovered area in other explored regions at the same time.

*C. Smoothness constraints*

We may introduce the smoothness constraints to the deployment process, so as to keep all sensors to move on a rather smooth path and avoid making sharp turns. To achieve this, we store the footprints for each sensor. Let  $v_{\text{next}}$  be the vector from  $s_i$  to  $c_i$  and  $v_{\text{prev}}$  be the vector from  $s_i$ 's last footprint to  $s_i$ . If the angle between  $v_{\text{next}}$  and  $v_{\text{prev}}$  is smaller than a user specified threshold, say 90 degrees, we consider  $c_i$  satisfying the constrains and  $s_i$  will be advanced as usual, or otherwise  $s_i$  is kept unchanged for the current iteration.

V. EXPERIMENTAL RESULTS

We simulate the application of our visibility-based coverage algorithm (SC2.0) for deploying mobile sensors with

a computer implementation to demonstrate its effectiveness. The visibility polygon for each sensor  $s_i$  as required in Step 1 of is computed using the VisiLibity library [16].

We use the clustering algorithm that is similar to the distortion-minimizing flooding method proposed in [8] to compute the VVD in our simulation. We discretize the domain into uniformly distributed points, each of which is attached with a label to indicate whether it is selected or not. All points are set *unselected* at the beginning. We start by labeling each sensor  $s_i$  as *selected* and inserting its 4-neighbor points  $\mathbf{q}_j$  in a global priority queue, with a priority equal to their respective distance  $g(\mathbf{q}_j, s_i)$  given in Equation (4), and an additional tag  $i$  indicating which sensor they are being tested against. For each point  $\mathbf{q}_j$  with tag  $i$  being popped out from the queue, if it is unselected, we assign it to the sensor  $s_i$ , label it *selected* and push its unselected neighbors that lie in  $\text{Vis}(s_i)$  into the queue with the tag  $i$ . The main modification compared with the flooding method in [8] is that an unselected neighbor  $\mathbf{q}$  with a tag  $s_i$  is pushed into the priority queue only if  $\mathbf{q} \in \text{Vis}(s_i)$ . More details can be found in [8]. This flooding algorithm is simple and fast, and it maintains the connectivity of each region. Hence, each visibility-based Voronoi region thus computed is connected which is required by our algorithm.

Figure 5 illustrates the sensor deployment process using our algorithm. Three more examples are shown in Figures 6, 7 and 8 (See also the accompanying videos.) From the experimental results, we find that Algorithm SC1.0 works well for simple environments, such as the examples shown in Figures 1(a), 6 and 7. Algorithm SC2.0, on the other hand, converges faster due to the actions of the interfacing sensors, and therefore SC2.0 is more suitable for complex environments, such as the examples in Figures 1(b), 5 and 8.

## VI. CONCLUSIONS

In this paper, we have proposed two distributed algorithms SC1.0 and SC2.0 to deploy a given set of mobile sensors in an unknown non-convex environment. By introducing the notion of visibility-based Voronoi diagram, we can achieve an optimal visibility-based coverage of the unknown environment with respect to an objective function. Our experimental results show that given enough number of sensors, our algorithms achieves a good configuration of the sensors.

The theoretical convergence of our algorithm is yet to be proved. A thorough study of the configuration of the local minimizers of  $\tilde{\mathcal{H}}(S, \mathcal{W})$  in Equation (3) is desired. We would also like to try different assumptions of our problem setting (e.g., sensing ability) and extend our algorithm to deal with other related problems in sensor coverage, such as:

- Sensor networks with various sensing ranges.
- $K$ -coverage sensor networks.
- Sensor networks in a 3D environment.

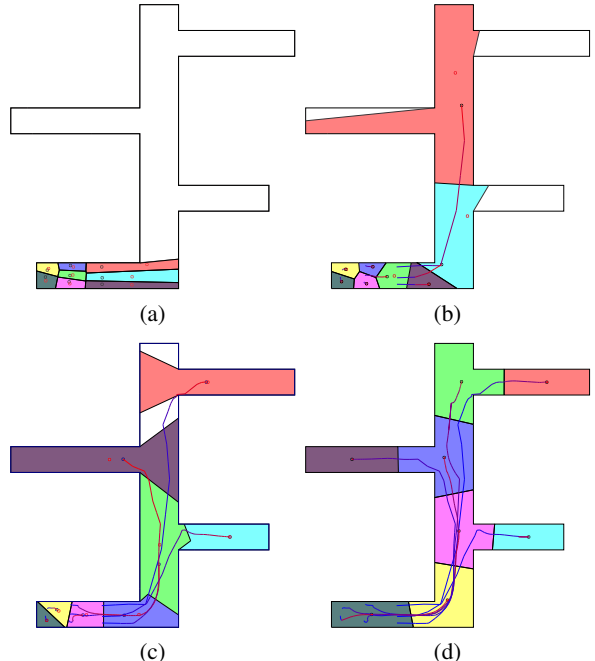


Figure 5. The deployment of eight sensors in a non-convex domain. (a): the initial position of the sensors; (b) and (c) show two states of the deployment process, where the black and red dots represent the sensor and its target position, respectively; (d) the stable result of the coverage, with the trajectory of each sensor's movement. The colored regions show the visibility-based Voronoi diagram. We use 90 degrees as the smoothness constraint and  $\lambda = 0.6$ .

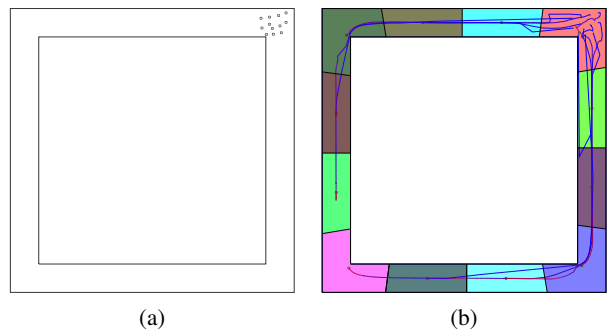


Figure 6. Example of deploying 12 sensors in a square domain with an obstacle inside. (a) the initial configuration; (b) the coverage result with trajectories.  $\lambda = 1.0$ .

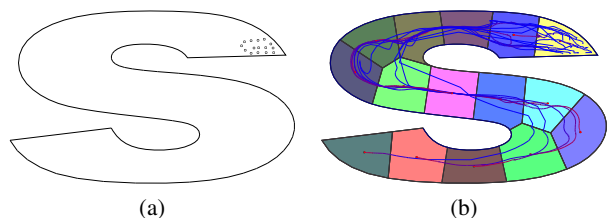


Figure 7. Example of deploying 15 sensors in an S-shaped domain. (a) the initial configuration; (b) the coverage result with trajectories.  $\lambda = 1.0$ .

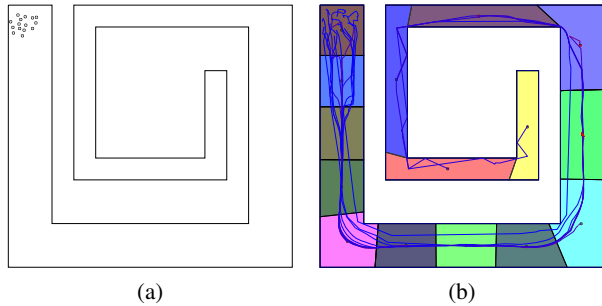


Figure 8. Example of deploying 15 sensors in a maze-like domain. (a) the initial configuration; (b) the coverage result with trajectories.  $\lambda = 1.0$ .

#### ACKNOWLEDGMENT

This work was partially supported by the China national natural science foundation 60703028.

#### REFERENCES

- [1] M. Argany, M.A. Mostafavi, and F. Karimipour. Voronoi-based approaches for geosensor networks coverage determination and optimisation: A survey. In *Proceedings of the 2010 International Symposium on Voronoi Diagrams in Science and Engineering*, pages 115–123, Jun. 2010.
- [2] Subhrajit Bhattacharya, Nathan Michael, and Vijay Kumar. Distributed coverage and exploration in unknown non-convex environments. In *Proceedings of DARS 2010*, Switzerland, November 2010.
- [3] Andreas Breitenmoser, Mac Schwager, Jean-Claude Metzger, Roland Siegwart, and Daniela Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4989, 2010.
- [4] C.H. Caicedo-Nunez and M. Zefran. A coverage algorithm for a class of non-convex regions. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4244–4249, 2008.
- [5] C.H. Caicedo-Nunez and M. Zefran. Performing coverage on nonconvex domains. In *Control Applications, 2008. CCA 2008. IEEE International Conference on*, pages 1019–1024, 3-5 2008.
- [6] M. Cardei and J. Wu. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, chapter Coverage in Wireless Sensor Networks. CRC Press, United States of America, 2005.
- [7] Howie Choset. Coverage for robotics — a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001. 10.1023/A:1016639210559.
- [8] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23:905–914, August 2004.
- [9] Sesh Commuri and Mohamed K. Watfa. Coverage strategies in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2:333–353, 2006.
- [10] J. Cortés, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, April 2004.
- [11] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [12] A. Ganguli, J. Cortes, and F. Bullo. Distributed coverage of nonconvex environments. In V. Saligrama, editor, *Networked Sensing Information and Control (Proceedings of the NSF Workshop on Future Directions in Systems Research for Networked Sensing, May 2006, Boston, MA)*, pages 289–305. 2007.
- [13] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, pages 299–308, 2002.
- [14] Wei Li and C.G. Cassandras. Distributed cooperative coverage control of sensor networks. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 2542–2547, 2005.
- [15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1380–1387, 2001.
- [16] K. J. Obermeyer and Contributors. The VisiLiberty library. <http://www.VisiLiberty.org>, 2008. R-1.
- [17] L. Pimenta, V. Kumar, R.C. Mesquita, and G. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3947–3952, Dec. 2008.
- [18] Alessandro Renzaglia and Agostino Martinelli. Distributed coverage control for a multi-robot team in a non-convex environment. In *IEEE IROS09 3rd Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2009.
- [19] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *Int. J. Rob. Res.*, 28:357–375, March 2009.
- [20] Bang Wang, Hock Beng Lim, and Di Ma. A survey of movement strategies for improving network coverage in wireless sensor networks. *Computer Communications*, 32(13-14):1427–1436, 2009.
- [21] Guiling Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.
- [22] M. Zhong and C. G. Cassandras. Distributed coverage control in sensor network environments with polygonal obstacles. In *IFAC World Congress*, pages 4162–4167, Seoul, Korea, July 2008.