

Efficient Sensor Placement for Surveillance Problems^{*}

Pankaj K. Agarwal, Esther Ezra, and Shashidhara Ganjugunte

Dept. of Comp. Sci., Duke University, Durham, NC 27708-0129

Abstract. We study the problem of covering a two-dimensional spatial region P , cluttered with occluders, by sensors. A sensor placed at a location p covers a point x in P if x lies within sensing radius r from p and x is visible from p , i.e., the segment px does not intersect any occluder. The goal is to compute a placement of the minimum number of sensors that cover P . We propose a *landmark-based* approach for covering P . Suppose P has ς holes, and it can be covered by h sensors. Given a small parameter $\varepsilon > 0$, let $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \varsigma$. We prove that one can compute a set L of $O(\lambda \log \lambda \log (1/\varepsilon))$ landmarks so that if a set S of sensors covers L , then S covers at least $(1 - \varepsilon)$ -fraction of P . It is surprising that so few landmarks are needed, and that the number does not depend on the number of vertices in P . We then present efficient randomized algorithms, based on the greedy approach, that, with high probability, compute $O(\tilde{h} \log \lambda)$ sensor locations to cover L ; here $\tilde{h} \leq h$ is the number sensors needed to cover L . We propose various extensions of our approach, including: (i) a weight function over P is given and S should cover at least $(1 - \varepsilon)$ of the weighted area of P , and (ii) each point of P is covered by at least t sensors, for a given parameter $t \geq 1$.

1 Introduction

With the advances in sensing and communication technologies, surveillance, security, and reconnaissance in an urban environment using a limited number of sensing devices distributed over the environment and connected by a network is becoming increasingly feasible. A key challenge in this context is to determine the placement of sensors that provides high coverage at low cost, resilience to sensor failures, and tradeoffs between various resources. In this paper we present a *landmark based* approach for sensor placement in a two-dimensional spatial region, containing occluders (a common model for an urban environment in the context of the sensor-placement problem). We choose a small set of landmarks, show that it suffices to place sensors that cover these landmarks in order to guarantee a coverage of most of the given domain, and propose simple, efficient algorithms (based on greedy approach) for placing sensors to cover these landmarks.

Our model. We model the 2D spatial region as a polygonal region P , which may contain holes (occluders). Let n be the number of vertices in P and ς the number of holes in P . A point $p \in P$ is *visible* from another point q if the segment pq lies inside P .

^{*} Work on this paper was supported by NSF under grants CNS-05-40347, CFF-06-35000, and DEB-04-25465, by ARO grants W911NF-04-1-0278 and W911NF-07-1-0376, and by an NIH grant 1P50-GM-08183-01.

Let $r \geq 0$ denote the *sensing radius*. We assume that a point x is covered by a sensor located at p , if $\|xp\| \leq r$ and p is visible from x . For a point $x \in \mathbb{R}^2$, we define $V(x) = \{q \in P \mid \|qx\| \leq r, pq \subset P\}$ to be the *coverage (surveillance)* region of the sensor located at x (this region is a portion of the so called “visibility polygon” of x). For a set $X \subseteq P$, define $V(X) = \bigcup_{x \in X} V(x)$ and $\mathcal{V}(X) = \{V(x) \mid x \in X\}$. When r is finite, we say that the model has a *finite sensing radius*, otherwise, the model has an *infinite sensing radius*. We define a function $\tau : P \rightarrow \mathbb{N}$, such that $\tau(p)$ is the *coverage requirement* of p , i.e., p should be covered by at least $\tau(p)$ sensors. A placement $S \subset P$ is a finite set of locations where sensors are placed. For a point $p \in P$, define $\chi(p, S) = |\{s \in S \mid p \in V(s)\}|$, i.e., the number of sensors that cover p . We say that S *covers* P if $\chi(p, S) \geq \tau(p)$, for every $p \in P$. (In Section 4, we mention a few extensions of our model.) The *coverage problem* asks for computing the smallest set S that covers P . If $\tau(p) = 1$, for all p , we refer to the problem as the *uniform coverage problem*.

In our model we assume that the polygon is given, as well as the occluder locations (otherwise, a hole detection algorithm can be used [4]), but the size of the smallest set S that covers P is unknown to us. The algorithm that we present approximates that measure up to a multiplicative factor of 2.

We note that it is commonly assumed that *connectivity* between the sensors mostly happens when they have a line of sight between them, this is because: (i) There are models, mostly in the case of swarms of robots [30] where the communication is through infrared signals, (ii) in an urban environment, wireless transmissions are stronger along a street (due to the reflections from walls) while loss of energy occurs at turning corners.

Related work. The sensor coverage problem has been studied extensively in several research communities, including sensor networks, machine learning, computational geometry, robotics, and computer vision [11, 17, 18, 28, 36, 40]. It is beyond the scope of this paper to mention all the related work, so we briefly mention a few of them.

Coverage problems are typically cast as an instance of the discrete set-cover problem, which given a finite set S and a set of subsets of S seeks to find a minimum number of subsets whose union is S . This is known to be NP-hard and hard to approximate. However, a standard greedy algorithm [29] gives $O(\log |S|)$ -approximation, where S is the number of sensors. This algorithm also generalizes to set multi-cover problem with a $O(\log |S|)$ -approximation [33]. However, these algorithms do not generalize to the continuous set-cover problem.

The early work in the sensor-network community on sensor coverage focused on covering an open environment (no occluders), so the problem reduced to covering P by a set of disks [40] (see also [25]). The problem is known to be NP-hard and efficient approximation algorithms are known. Tian *et al.* [35] presented algorithms to select the smallest number of sensors out of a given densely deployed sensors to provide coverage. However, they do not address how to place sensors at the initial setting. In a related work [9] provided sensor placement algorithms for covering discrete grid points. A variant of the coverage problem, which involves communication connectivity constraints among sensors, has been studied in [8, 24, 38].

In an urban environment, one has to consider line-of-sight constraints and other interference while placing sensors. The so-called *art gallery problem* and its variants,

widely studied in computational geometry and robotics, ask for covering P by a set of sensors so that every point in P is visible from at least one sensor. The art-gallery problem is known to be APX-hard [15], i.e., constant-factor approximation cannot be computed in polynomial time unless $P = NP$. If there is no restriction on the placements of sensors, no approximation algorithm is known. However, if the placement needs to be chosen from a finite set of candidate locations (e.g. uniform grid), polynomial-time log-approximation algorithms are known; see [10, 13] and the references therein. González-Banos and Latombe [17] proposed to choose a set of random positions as candidate locations for sensors. They proved that the randomized approach leads to a good approximation algorithms under some assumptions on P ; similar approaches have been widely used in the context of path planning. Efrat *et al.* [14] developed approximation algorithms for the non-uniform coverage for $\tau(p) = 2, 3$ and with additional constraints on coverage, under the assumption that the sensor locations are chosen from a set of grid points.

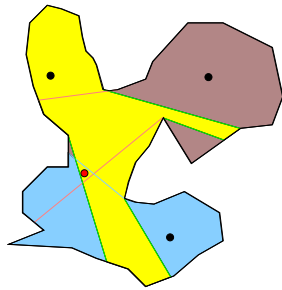


Fig. 1. A set of landmarks and their respective coverage regions.

The problem of selecting a set of sensors to provide k -coverage (and in some cases also to form a connected network), from a prior dense deployment has been studied for specific settings such as grid deployment, random uniform deployment or for specific types of sensing regions, see for example [20, 22, 23, 27, 39, 41]. Abrams *et al.* [1] study a variant, namely, computing k -disjoint covers from a set of sensors already deployed so as to facilitate scheduling to save energy.

Meguerdichian *et al.* [31] study variants of the coverage problem, namely maximal breach (resp. support) path problem in which given a deployment of sensors and a set of source and destination locations the goal is to find a path between a source and a destination so that for any point on the path the distance to the closest sensor is maximized (resp. minimized). In a related problem, studied at [32], a deployment of the sensors is given and the goal is to find a path which is the least covered; this is an instance of the so called *exposure problem*. These solutions assume a deployment of sensors and try to characterize the locations for additional sensors so as to minimize breach. However, our approach is more pro-active, and all the key areas of a domain to be monitored can be given suitable weights, and our algorithm returns a set of location for the sensors which ensures that the key areas are monitored.

Because of measurement errors, the dependence of the coverage quality on the distance, and correlation between the observations of nearby sensors, statistical models have been developed for sensor coverage and entropy based methods that try to maximize the information acquired through the sensors; see e.g. [18]. These methods are particularly useful for monitoring spatial phenomena.

Our results. We follow a widely used approach for sensor coverage, i.e., sampling a finite set of points and then using a greedy algorithm to compute the location of sensors, but with a twist: instead of sampling the set of candidate placements for sensors, we sample landmarks that should be covered. This twist has the advantage that unlike

previous studies, we can obtain a provable bound on the performance of the algorithm without any assumption on P . There are two main contributions of the paper. First, we prove that the number of landmarks needed is surprisingly small—it is independent of the number of vertices in P . Second we describe simple, efficient algorithms for implementing the greedy approach—a straightforward implementation of the greedy approach is expensive and not scalable.

Suppose h is the number of sensors needed for uniform coverage of P (as noted above, this value is unknown to us). For a given parameter $\varepsilon > 0$, set $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \zeta$. We prove that one can choose a set L of $m = O(\lambda \log \lambda \log(1/\varepsilon))$ landmarks so that if S uniformly covers L , then S uniformly covers at least $(1 - \varepsilon)$ fraction of the area of P . We refer to this result as the sampling theorem.

Next we describe algorithms for computing $O(\tilde{h} \log \lambda)$ sensors, with high probability, that cover L , where \tilde{h} is the minimum number of sensors needed to cover L ; obviously $\tilde{h} \leq h$, but it can be much smaller. We show that a straightforward implementation takes $O(nm^2 h \zeta \log(mn) \log m)$ expected time. The expected running time of our algorithm can be improved to $O(nm^2 \zeta \log(mn))$. We then present a Monte Carlo algorithm, which, albeit having the same running time as the simpler greedy algorithm in the worst case, is faster in practice, as evident from our experimental results. If P is a simple region, i.e., its boundary is connected, the expected running time of the Monte Carlo reduces to $O((mn + m^2)h \log^2 m \log(mn))$.

Our overall approach is quite general and can be extended in many ways. For instance, it can incorporate other definitions of the coverage region of a sensor, such as when a sensor (e.g. camera) can cover the area only in a *cone* of directions; in this case the model is called the *camera model*. We can extend our sampling theorem to a weight setting: We are given a weight function $w : P \rightarrow \mathbb{R}^+$ and we wish to ensure that at least $(1 - \varepsilon)$ fraction of the weighted area of P is covered. This extension is useful because we have hot spots in many applications and we wish to ensure that they are covered. Finally, we extend our sampling theorem to non-uniform coverage—the number of landmarks increases in this case.

In Section 2 we describe our landmark based approach, and prove the sampling theorem under either the finite sensing radius or the camera model. Section 3 describes the greedy algorithms. In Section 4, we briefly mention various extensions of our algorithm, including the weighted case and non-uniform coverage. In Section 5, we present experimental results with software implementation of the Monte Carlo algorithm, showing that our approach is useful in practice.

2 Landmark Based Algorithm for Uniform Coverage

In this section we describe a landmark based algorithm for the uniform sensor-coverage problem under either the finite sensing radius or the camera model. We begin by reviewing a few concepts from random sampling and statistical learning theory [34]. Let X be a (possibly infinite) set of objects also referred to as the “space”, let μ be a measure function on X , and let \mathcal{R} be a (possibly infinite) family of subsets of X , those subsets are called the “ranges”. The pair $\Sigma = (X, \mathcal{R})$ is called a *range space*. A finite subset $Y \subseteq X$ is *shattered* by Σ if $\{r \cap Y \mid r \in \mathcal{R}\} = 2^Y$, i.e., the restriction of \mathcal{R} to Y can realize all subsets of Y . The *VC-dimension* of Σ , denoted by $\text{VC-dim}(\Sigma)$, is defined to be the largest size of a subset of X shattered by Σ ; the VC-dimension is infinite if

the maximum does not exist. Many natural range spaces have finite VC-dimension. In our context, let $\mathcal{V} = \{V(x) \mid x \in P\}$, and let $\mathbb{V} = (P, \mathcal{V})$ be the range space. That is, the space is the polygon P and the ranges are all the coverage regions of the sensors located in P . When the sensing radius is infinite, the VC-dimension of \mathbb{V} is known to be $O(1 + \log \varsigma)$ [37].

For a given $\varepsilon > 0$, a subset $N \subseteq X$ is called an ε -net of Σ if $r \cap N \neq \emptyset$ for all $r \in \mathcal{R}$ such that $\mu(r) \geq \varepsilon \mu(X)$. In other words, an ε -net is a hitting set for all the “heavy” ranges. A seminal result by Haussler and Welzl [21] shows that if $\text{VC-dim}(\Sigma) = d$, then a random subset $N \subseteq X$ of size $O((d/\varepsilon) \log(d/\varepsilon\delta))$ is an ε -net of Σ with probability at least $1 - \delta$. This bound has later been improved to $O(d/\varepsilon \log(1/\varepsilon))$ by Komlós, Pach, and Woeginger [26], who showed that a random sample of that size is an ε -net with constant probability¹.

We now mention a useful property of VC-dimension: If $\Sigma_1 = (X, \mathcal{R}_1), \dots, \Sigma_k = (X, \mathcal{R}_k)$ are k range spaces and $\text{VC-dim}(\Sigma_i) \leq d$ for all $i \leq k$, then the VC-dimension of the range space $\Sigma = (X, \mathcal{R})$, where $\mathcal{R} = \{\bigcup_{i=1}^k r_i \mid r_1 \in \mathcal{R}_1, \dots, r_k \in \mathcal{R}_k\}$, is $O(kd)$. Returning to our model, when the sensing radius r is finite, or each sensor can cover an area that is contained in a cone of directions, we define a new range space, so that the space remains P , and the ranges are modified as follows. In the first case, each range is the intersection of a coverage region $V(x)$ (of an unbounded radius), $x \in P$, and a disc centered at x and having radius r . In the camera case, each range is the intersection of $V(x)$ and a cone whose apex is in x (and whose opening angle corresponds to the camera “vision range”). It is well known that the range space $\mathbb{V}_D = (P, D)$, where D is the set of all disks of radius r with centers at P , has a constant VC-dimension (see, e.g., [34]). Using the properties stated above, it follows that the VC-dimension of the range space corresponding to the finite sensing radius model is $O(1 + \log \varsigma)$. For the camera model, it can be shown in an analogous manner that the VC-dimension of the corresponding range space is $O(1 + \log \varsigma)$ as well, by observing that the VC-dimension of the range space $\mathbb{V}_C = (P, C)$ is bounded by a constant, where C is the set of all cones whose apex lies in P .

We now state and prove the main result of this section.

Theorem 1 (Sampling Theorem). *Let P be a polygonal region in \mathbb{R}^2 with ς holes, and let $\varepsilon > 0$ be a parameter. Suppose P can be covered by h sensors, each of which with a finite sensing radius r . Set $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \varsigma$. Let $L \subset P$ be a random subset of $c_1 \lambda \ln \lambda \ln(1/\varepsilon)$ points (landmarks) in P , and let S be a set of at most $c_2 h \ln \lambda$ sensors that cover L , where $c_1 \geq 1$, $c_2 \geq 1$ are sufficiently large constants. Then S covers at least $(1 - \varepsilon)$ fraction of the area of P with constant probability. The same asymptotic bounds hold for the camera model.*

Proof. We prove the theorem for the finite sensing radius model; the proof for the camera model proceeds almost verbatim.

Set $k = c_2 h \ln \lambda$. Let $\bar{\mathcal{V}}^k = \left\{ P \setminus \bigcup_{i=1}^k V(x_i) \mid x_1, \dots, x_k \in P \right\}$, that is, each range in $\bar{\mathcal{V}}^k$ is the complement of the union of (at most) k coverage regions. Set $\bar{\mathbb{V}}^k =$

¹ In fact, the actual bound stated in [26] is slightly tighter, and the authors prove the existence of an ε -net of that size. We use the more conservative bound $O(d/\varepsilon \log(1/\varepsilon))$, since the analysis in [26] shows that a random sample of that size yields an ε -net with constant probability.

(P, \bar{V}^k) . Since $\text{VC-dim}(\mathbb{V}) = O(\log \varsigma)$, the above discussion implies that $\text{VC-dim}(\bar{V}^k) = O(k \log \varsigma)$. Therefore if we choose a random subset $L \subset P$ of size $O((k/\varepsilon) \log(\varsigma) \log(1/\varepsilon))$, then L is an ε -net of \bar{V}^k with constant probability. Let S be a set satisfying the assumptions in the lemma, that is, S is a set of at most k sensors that cover L , i.e., $L \subset V(S)$. Then, clearly, $L \cap (P \setminus V(S)) = \emptyset$. By the definition of ε -net, $\mu(P \setminus V(S)) \leq \varepsilon \mu(P)$. Hence, S covers at least $(1 - \varepsilon)$ fraction of the area of P . This completes the proof of the theorem. \square

The Sampling Theorem suggests the following simple approach, which we summarize by the procedure `PLACESENSORS` (P, ε) , to compute a placement for P with an error threshold ε . Given a set L of landmarks, the procedure `GREEDYCOVER` (P, L) , described in the next section, computes a cover for L of size $O(\tilde{h} \log |L|)$ (under either of the two above models), where $\tilde{h} \leq h$ is the number of sensors needed to cover L . Since h (or \tilde{h}) is unknown to us, we apply an exponential search in order to approximate that value up to a factor of 2 (see below at the procedure `PLACESENSORS` (P, ε)).

```

PLACESENSORS  $(P, \varepsilon)$ 
INPUT: Polygonal region  $P$ , error threshold  $\varepsilon$ 
OUTPUT: A placement of sensors covering
         $\geq (1 - \varepsilon)\mu(P)$ 
1.  $i := 1$ 
2. repeat
3.    $h := 2^i, d := c_2(1 + \log \varsigma),$ 
       $k = c_2 h \ln(dh/\varepsilon), m := (c_1 dk/\varepsilon) \ln(1/\varepsilon)$ 
4.    $L := m$  random points in  $P$ 
5.    $S := \text{GREEDYCOVER}(P, L)$  //  $L \subset V(S)$ 
6.    $i := i + 1$ 
7. until  $(|S| \leq k) \wedge (\mu(V(S)) \geq (1 - \varepsilon)\mu(P))$ 

```

Fig. 2. A landmark based algorithm for covering P .

Remark: We note that the actual number of sensor placements that the algorithm computes is an approximation to the optimal number of sensors that cover L (and not necessarily the entire polygon P), which may be much *smaller* than h .

3 The Greedy Algorithm

In this section we describe the procedure `GREEDYCOVER` (P, L) that computes a placement of sensors to cover a set $L \subset P$ of m landmarks under each of the two models defined above. We first describe a simple algorithm, which is a standard greedy algorithm, and then discuss how to expedite the greedy algorithm. For a point $x \in P$ and a finite subset $N \subset P$, we define the *depth* of x with respect to N , denoted by $\Delta(x, N)$, to be $|V(x) \cap N|$, the number of points in N that lie in the coverage region of x . Set $\Delta(N) = \max_{x \in P} \Delta(x, N)$.

Simple algorithm. The simple procedure computes the placement S incrementally. In the beginning of the i -th step we have a subset L_i of m_i sensors and a partial cover S ; S

covers $L \setminus L_i$. Initially, $L_i = L$ and $S = \emptyset$. At the i -th step we place a sensor at a location z_i that covers the maximum number of landmarks in L_i , i.e., $\Delta(z_i, L_i) = \Delta(L_i)$. We add z_i to S , compute $L'_i = V(z_i) \cap L_i$, and set $L_{i+1} = L_i \setminus L'_i$. The procedure stops when $L'_i = \emptyset$. A standard analysis for the greedy set-cover algorithm shows that the algorithm computes $\tilde{h} \ln |L|$ locations, where \tilde{h} is the number of sensors needed to cover L , see, e.g., [29].

The only nontrivial step in the above algorithm is the computation of z_i . Note that $\Delta(x, L_i) = |\{p \in L_i \mid x \in V(p)\}|$, i.e., the number of coverage regions in $\mathcal{V}(L_i)$ that contain x . We compute the *overlay* $\mathcal{A}(L_i)$ of the coverage regions in $\mathcal{V}(L_i)$, which is a planar map; see Figure 1. It is known that $\mathcal{A}(L_i)$ has $O(m_i^2 n \zeta)$ vertices, edges, and faces, and that $\mathcal{A}(L_i)$ can be computed in time $O(m_i^2 n \zeta \log n)$. Next, we compute the depth of every vertex in $\mathcal{A}(L_i)$, with respect to L_i , within the same time bound. We then set z_i to be the deepest vertex in $\mathcal{A}(L_i)$. The overall running time of the GREEDYCOVER procedure is $O(m^2 n \zeta |S| \log n)$.

We note that it is unnecessary to compute $\mathcal{A}(L_i)$ in each step. Instead we can compute $\mathcal{A}(L)$ at the beginning and maintain $\mathcal{A}(L_i)$ and the depth of each of its vertices using a dynamic data structure; see [2]. The asymptotic running time now becomes $O(m^2 n \zeta \log^2 n)$, which is faster if $|S| \geq \log n$.

A Monte Carlo algorithm. We observe that it suffices to choose a point x at the i -th step such that $\Delta(x, L_i) \geq \Delta(L_i)/2$; it can be shown that the number of iterations then is still $O(\tilde{h} \ln |L|)$ [2]. Using this observation, we can expedite the algorithm if $\Delta(L_i)$ is large, as follows: We choose a random subset $K_i \subset L_i$ of smaller size (which is inversely proportional to $\Delta(L_i)$) and set z_i to be a deepest point with respect to K_i . If the size of K_i is chosen appropriately, then $\Delta(z_i, L_i) \geq \Delta(L_i)/2$, with high probability. More precisely, set $\Delta = \Delta(L_i)$, and let $q \geq \Delta/4$ be an integer. We choose a random subset $K_i \subseteq L_i$ by choosing each point of L_i with probability $\rho = c_1 \ln(m_i)/q$, where $c_1 > 0$ is a constant. Using the arguments in [3, 6], we can show that with probability at least $1 - 1/m_i$ the following two conditions hold: (i) $\Delta \geq q$ implies $\Delta(K_i) \geq 3q\rho/2 = (3c_1/2) \ln m_i$, and (ii) $\Delta \leq q$ implies $\Delta(K_i) \leq 5q\rho/4 = (5c_1/4) \ln m_i$. An exponential search procedure that sets $q = m_i/2^j$ at the j th step, chooses a subset K_i as described above, computes a deepest point x_i in $\mathcal{A}(K_i)$, and uses the above two conditions to determine whether return x_i or half the value of q , can compute a point z_i of depth at least $\Delta(L_i)/2$, with probability at least $1 - 1/m_i$. The expected running time of this procedure is $O(n(m_i/\Delta)^2 \zeta \log n \log^2 m)$.

This procedure computes an overlay of m_i/Δ coverage regions compared with that of m_i regions in the previous procedure. Note that $\Delta(L)$ may be small, in which case the running time of this procedure is the same as that of the previous one. It is, however, faster in practice, as shown in Section 5.

A crucial property that this approach relies on is the fact that the complexity of the boundary of the *union* of the coverage regions (that is, the vertices and edges of the overlay that are not contained in the interior of any regions) is significantly smaller than the complexity of the entire overlay. Specifically, Gawali *et al.* [16] showed that, if the boundary of P is connected and the sensing radius r is infinite, the overall number of vertices and edges that appear along the boundary of the union of the regions in $V(L)$ is only $O(mn + m^2)$. In the full version of this paper we prove similar asymptotic bound

for the case where r is finite, or when the coverage region of each sensor is bounded by a cone of directions. In this case, the running time of the overall GREEDYCOVER procedure becomes $O((m/\Delta)n + (m/\Delta)^2 \log m)|S| \log^2 n)$, where $\Delta = \Delta(L)$. Furthermore, by using a combination of the previous and the Monte Carlo procedures, the running time can be further improved for this special case. This, however, does not necessarily hold for a polygon with holes, in which case the union complexity is $\Omega(m^2 n \zeta)$ in the worst case; we provide the lower bound construction in the full paper.

4 Extensions

We briefly describe the following extensions and omit any further details in this version.

Weighted coverage. We are now given a weight function $w : P \rightarrow \mathbb{R}^+$. The *weighted area* of P is defined to be $\mu_w(P) = \int_{x \in P} w(x) dx$, and is normalized to be 1. The weighted area of a coverage region $V(l)$ for a location $l \in P$, $\mu_w(V(l))$, is $\int_{x \in V(l)} w(x) dx$. The goal is to find a placement S of sensors such that $\mu(V(S)) \geq (1 - \varepsilon)\mu(P)$. This extension is useful in practice as it enables us to highlight “hot-spots” in the environment. Since the random sampling theory is applicable to the weighted setting, the above algorithm and the analysis extends to this case verbatim.

Multiple coverage. In order to be resilient to sensor failures, we now require that each point $p \in P$ to be covered by at least t distinct sensors, for a given parameter $t \geq 1$. The definition of the range-space becomes more complex: For a set $X \subset P$ of k points, we now define $\bar{V}_t(X) \subseteq V(X)$ to be the set of points whose depth is less than t . Set $\bar{V}_t^k = \{\bar{V}_t(X) \mid X \subset P \text{ is a set of } k \text{ points}\}$ and $\bar{V}_t^k = (P, \bar{V}_t^k)$. We prove $\text{VC-dim}(\bar{V}_t^k) = O(k^t t^2 \log(\zeta) \log(kt \log \zeta))$, which in turn leads to a bound on the number of landmarks. Next, we extend the greedy algorithm to find a placement of sensors so that each landmark is covered by at least t sensors.

5 Experiments

In this section we present experimental results showing that our approach is also useful in practice. We have implemented both simple and Monte Carlo algorithms for the uniform-coverage problem under the finite sensing radius model.

Implementation. Recall that the algorithms compute the overlay of the coverage region of the landmarks. Computing the exact overlay is, however, complicated, inefficient in practice, and suffers from robustness issues, see [19] for a detailed review about the various difficulties that arise in practice. We thus resort to computing the coverage regions and their overlays approximately using *quadtrees* (see, e.g., [12]).

Given the input polygon P , we recursively partition the plane into four quadrants, each of which contains a portion of the boundary of P (in fact, this portion contains at most half of the vertices of P). The recursion bottoms out when the following three conditions are satisfied: We encounter to a cell that (i) contains at most a single polygon vertex, (ii) meets at most two edges of the polygon, (iii) has an area that is sufficiently small with respect to the polygon area. The cells that are generated at the bottom of the recurrence induce an approximated partition of the polygon into axis-parallel rectangles, which we also refer to as *pixels*; see Figure 3(a). This approximated representation improves as we refine the decomposition. Having the pixels at hand, we represent each

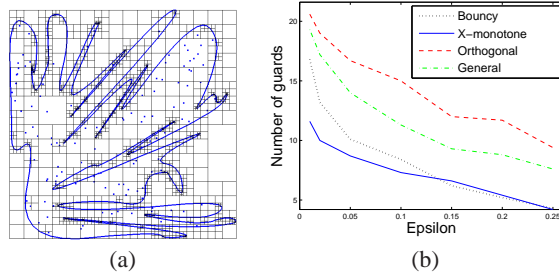


Fig. 3. (a) A quadtree representing a polygon. The dots are the landmarks. (b) The (averaged) number of sensors as a function of ε for each of the four data sets.

of the coverage regions induced by the landmarks as the corresponding subset of these pixels. In fact, this representation is performed implicitly by computing, for each pixel C , the set of the landmarks that are *visible* to C , and then setting the *depth* of C to be the number of these landmarks. Then, when we apply the greedy algorithm, we update this data accordingly. When the algorithm terminates, the depth of each pixel C becomes 0. Technically, the notion of visibility of a landmark s and a pixel C is not well-defined. In our implementation, we report that s sees C if it sees at least one point on its boundary.

The algorithm presented in Section 2 can be simplified, so that one need not choose constants c_1 , c_2 , and the number of landmarks m is estimated directly using an exponential search. This variant of the algorithm is presented in Figure 4. Our experimental results indicate that this implementation has good performance in practice—see below.

```

PLACESENSORS( $P, \varepsilon$ )
INPUT: Polygonal region  $P$ , error threshold  $\varepsilon$ 
OUTPUT: A placement of sensors covering
         $\geq (1 - \varepsilon)\mu(P)$ 
1.  $m := 16$ 
2. repeat
3.    $L := m$  random points in  $P$ 
4.    $S := \text{GREEDYCOVER}(P, L)$  //  $L \subset V(S)$ 
5.    $m := 2m$ 
6. until  $\mu(V(S)) \geq (1 - \varepsilon)\mu(P)$ 

```

Fig. 4. The implementation for the landmark based algorithm.

Input sets. We next describe our data sets. Our input consists of both random polygons and manually produced polygons, which are interesting as well as challenging in the context of our problem. For the random polygons, we use a software developed by Auer and Held [7] for generating random simple polygons, and for the manually produced polygons, we use the inputs of Amit *et al.* [5]. Our data sets are listed, as follows:

- **Bouncy** (Figure 5(a)): A random bouncy polygon with 512 vertices.

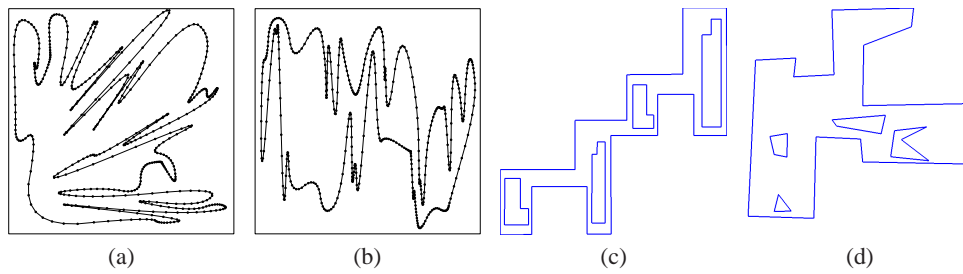


Fig. 5. The data sets. (a) Bouncy. (b) x -monotone. (c) Orthogonal. (d) General.

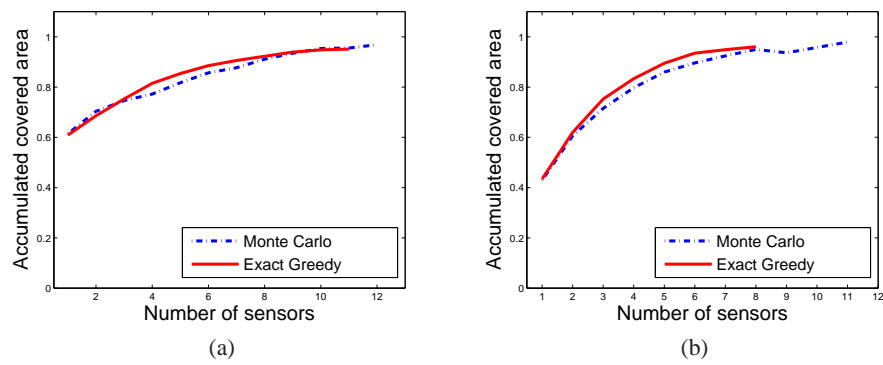


Fig. 6. The (averaged) accumulated covered area (y -axis) at each iteration (x -axis) for the Exact Greedy and the Monte Carlo algorithms, for the bouncy and x -monotone data sets.

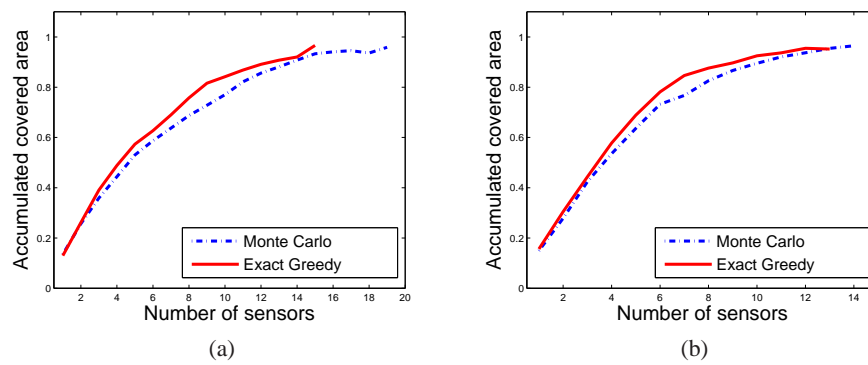


Fig. 7. The (averaged) accumulated covered area (y -axis) at each iteration (x -axis) for the Exact Greedy and the Monte Carlo algorithms, for the orthogonal and general data sets.

- ***x*-monotone** (Figure 5(b)): A random *x*-monotone polygon with 512 vertices.
- **Orthogonal** (Figure 5(c)): An orthogonal polygon with 20 vertices and 4 holes, each of which has (additional) 6 vertices.
- **General** (Figure 5(d)): A general polygon with 16 vertices and 4 holes with (additional) overall 16 vertices.

The orthogonal and general data sets represent urban environment, where the goal in experimenting with the first two data sets (bouncy and *x*-monotone) is to test the worst-case behavior of the algorithm, where the number of vertices is relatively large, and the coverage regions have long and skinny spikes—a property that follows from the structure of these polygons.

Each of these polygons is bounded by a square with a side-length of 600 units and thus the area of each of these polygons is bounded from above by $36 \cdot 10^4$. For the orthogonal and general data sets we choose a radius of length 100, where for the bouncy and *x*-monotone data sets we choose that length to be 10^4 .

Results. We present experimental results, applying both the simple greedy algorithm presented in Section 3 (which we also refer to as the *Exact Greedy algorithm*) and the Monte Carlo algorithm on each of our data sets. We measure various parameters, each of which is a yardstick for measuring and comparing the performance of the algorithms. Given a fixed $\varepsilon > 0$, these parameters are listed, as follows: (i) Number of *visibility tests* that both algorithms perform. In each of these tests we need to report whether a landmark l sees a pixel C ; the overwhelming majority of the running time is determined by the overall number of calls to this procedure, as, in our experiments, the preprocessing time for constructing the quadtree is negligible compared with the overall running time of the various visibility tests. (ii) Number of landmarks chosen by the algorithm. (iii) Number of iterations required in order to obtain a coverage of $(1 - \varepsilon)$ of the polygon; as stated in Section 3, at each iteration we choose a single sensor placement, and thus the above measure corresponds to the overall number of sensor placements chosen by the algorithm. (iv) The convergence ratio of the algorithm; at each iteration i we measure the ratio between the overall area just covered (over all iterations $1, \dots, i$) and the entire polygon area.

For each specific input set, we ran each algorithm ten times, and the results reported below are the averages over each of the above specified parameters. In all our experiments, we set $\varepsilon = 0.05$.

data set	Exact Greedy		Monte Carlo	
parameter	m ,	τ	m ,	τ
Bouncy	499,	848162	83,	288905
<i>x</i> -monotone	243,	450259	108,	202714
Orthogonal	256,	6505115	56,	95969
General	76,	147326	44,	117926

Table 1. The (averaged) number of landmarks m and visibility tests τ for the Exact Greedy and the Monte Carlo algorithms, for each of the data sets.

In all our experiments the Monte Carlo algorithm performs better than the Exact Greedy algorithm. Specifically, the expected number of visibility tests performed by the Monte Carlo algorithm is smaller (and sometimes significantly smaller) than that of the Exact Greedy algorithm; see Table 1.

The number of landmarks in each of our experiments is small, as expected. Moreover, due to our implementation, the number of landmarks m is estimated directly by an exponential search (Figure 4) and not according to the actual number of guards and ε (Figure 2), which assigns a relatively large initial value for m that corresponds to the worst-case bound on the ε -net size. Our experiments indicate that the number of landmarks is significantly smaller under the Monte Carlo algorithm. This improvement is achieved due to the property that in the Monte Carlo algorithm we perform a *second sampling step* over the set of the landmarks, in order to locate a point that covers a largest number of landmarks (approximately). In this case, it is likely that the resulting sensor placements have a better deployment than that achieved by the Exact Greedy algorithm; see once again Table 1.

The number of iterations of the Monte Carlo algorithm cannot be smaller than that of the Exact Greedy algorithm by definition (see Section 3). Nevertheless, our experiments show (see Figures 6, 7) that these values are roughly similar for all our data sets. Moreover, in all our experiments, the number of iterations of the Monte Carlo algorithm is less than 1.5 of that of the Exact Greedy algorithm. The convergence ratio for (both of) the random polygons is rapid, as indicated by our experiments, at the very first iterations, both algorithms manage to cover most of the polygon. This does not necessarily hold for the polygons with holes (Figure 7), since it is difficult to find a single sensor that covers most of the polygon (according to our findings, this is also the case when the sensing radius is significantly larger), and thus any algorithm that aims to cover these polygon has to use a relatively large number of sensors; see once again Figure 5(c)–(d).

In addition to the results reported above, we have tested the dependency of the number of sensor placements in ε . As one may expect, the number of sensors should decrease as we enlarge ε . In Figure 3(b) we present these results for each of the data sets under the Monte Carlo algorithm. As indicated by our experiments, the number of sensors significantly decreases as we loose the error-parameter ε .

6 Concluding Remarks

In this paper we presented a landmark based approach to place a set of sensors. The main contribution of the paper is that a very small number of landmarks are sufficient to guide the placement of sensors. We are currently addressing several important issues that are not included in this paper. Our algorithm does not ensure that the sensors can communicate with each other. We are currently developing an algorithm to add a few *relay* nodes so that the sensors can communicate with each other. We are also extending our algorithms for covering 3D environments. The coverage region can be quite complex under our model, therefore the complexity of the overlay of coverage regions can be quite large, which in turns makes the algorithm exorbitantly expensive. We plan to circumvent this problem by defining the notion of *robust (fuzzy)* coverage. Intuitively, we define the boundary of the coverage region in our current model as fuzzy and we allow it to be simplified.

References

1. Z. Abrams, A. Goel, S. Plotkin. Set k -cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proc. 3rd Intl. Symp. on Inform. Proc. Sensor Networks*, pp 424–432, 2004.
2. P. K. Agarwal, D. Z. Chen, S. K. Ganjugunte, E. Misiołek, and M. Sharir. Stabbing convex polygons with a segment or a polygon. In *Proc. 16th European Sympos. Algorithms*, 2008, to appear.
3. P. K. Agarwal, T. Hagerup, R. Ray, M. Sharir, M. H. M. Smid, and E. Welzl. Translating a planar object to maximize point containment. In *Proc. 10th European Sympos. Algorithms*, pp 42–53, 2002.
4. N. Ahmed, S. S. Kanhere, S. Jha. The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mobile Comput. and Commun. Review*, 9(2):4–18, 2005.
5. Y. Amit, J. S. B. Mitchell, and E. packer. Locating guards for visibility coverage of polygons. In *Proc. Workshop Alg. Eng. Exp.*, pp 120–134, 2007.
6. B. Aronov and S. Har-Peled. On approximating the depth and related problems. In *Proc. 16th Annu. ACM-SIAM Symp. on Disc. Alg.*, pp 886–894, 2005.
7. T. Auer and M. Held. Heuristics for the Generation of Random Polygons. In *Proc. 8th Canad. Conf. Comput. Geom.*, pp. 38–44, Carleton University Press, 1996.
8. X. Bai, S. Kumar, Z. Yun, D. Xuan, and T.H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *Proc. ACM MobiHoc*, pp. 131–142, 2006
9. K. Chakrabarty, S.S Iyengar, H. Qi, E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Computers*, 51(12), 1448–1453, 2002.
10. O. Cheong, A. Efrat, and S. Har-Peled. Finding a guard that sees most and a shop that sells most. *Discrete Comput. Geom.*, 37(4):545–563, 2007.
11. S. Dhillon and K. Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proc. IEEE Wireless Communications Network. Conf.*, pp. 1609–1614, 2003.
12. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2000.
13. A. Efrat and S. Har-Peled. Guarding galleries and terrains. In *2nd IFIP Internat. Conf. Theo. Comp. Sci.*, 181–192, 2002.
14. A. Efrat and S. Har-Peled and J. S. B. Mitchell. Approximation algorithms for two optimal location problems in sensor networks. In *2nd Internat. Conf. on Broadband Networks*, 1:714–723, 2005.
15. S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001
16. L. Gewali, A. Meng, J. S. B. Mitchell, and S. Ntafos. Path planning in $0/1/\infty$ weighted regions with applications. *ORSA J. Computing*, 2:253-272, 1990.
17. H. H. González-Ba nos and J. C. Latombe. A randomized art-gallery algorithm for sensor placement. In *Proc. 16th ACM Symp. Comput. Geom.*, pp. 232–240, 2000.
18. C. Guestrin, A. Krause, and A. P. Singh. Near-optimal sensor placements in Gaussian processes. In *Proc. 22th Int. Conf. Machine Learn.*, 265–272, 2005.
19. D. Halperin. Robust Geometric Computing in Motion. *International Journal of Robotics Research*, 21(3):219–232, 2002.
20. M. Hefeeda and M. Bagheri. Randomized k -coverage Algorithms for Dense Sensor Networks. In *Proc. of 26th IEEE Intl. Conf. on Comp. Commn.*, 2376–2380, 2007.
21. D. Haussler and E. Welzl. ϵ -nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

22. C.F. Huang, Y.C. Tseng. The coverage problem in a wireless sensor network. In *Proc. ACM Workshop Wireless Sensor Networks Appl. (WSNA)*, pp. 115–121, 2003.
23. C.F. Huang, Y.C. Tseng, L.C. Lo. The coverage problem in three-dimensional wireless sensor networks. In *Proc. IEEE GLOBECOM*, pp. 3182–3186, 2004.
24. R. Iyengar, K. Kar, S. Banerjee. Low-coordination topologies for redundancy in sensor networks. In *Proc. 6th ACM international sympos. Mobile ad hoc networking comput. (MobiHoc)*, pp. 332–342, 2005.
25. R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61(3):665–671, 1939.
26. J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for epsilon nets, *Discrete Comput. Geom.*, 7:163–173, 1992.
27. S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *Proc. of 10th Annu. Intl. Conf. Mobile Comput. and Networking*, pp 144–158, 2004.
28. J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
29. L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.
30. J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, B. Schmidt. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *Proc. AAAI Spring Sympos.*, 2006.
31. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, pp. 1380–1387, 2001.
32. S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Proc. 7th Annu. Intl. Conf. on Mobile Computing and Networking*, pp. 139–150, 2001.
33. S. Rajagopalan and V. V. Vazirani. Primal-Dual RNC approximation algorithms for set cover and covering integer programs. *SIAM J. Comput.*, 28(2):525–540, 1999.
34. J. Pach and P. K. Agarwal. *Combinatorial geometry*. Wiley-Interscience, San Diego, 1995
35. D. Tian, D. and N.D Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. ACM Workshop Wireless Sens. Nets. Appl. (WSNA)*, pp. 32–41, 2002.
36. J. Urrutia. Art Gallery and illumination problems. In *Handbook of Computational Geometry*, (J. Sack and J. Urrutia, eds), Elsevier, Amsterdam, pages 973–1027, 2000.
37. P. Valtr. Guarding galleries where no point sees a small area. *Israel J. Math*, 104:1–16, 1998.
38. Y.C. Wang, C.C Hu, Y.C. Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *Proc. IEEE Wireless Internet Conference (WICON)*, pp. 114–121, 2005.
39. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st Intl. Conf. on Embedded Networked Sensor Systems*, pp. 28–39, 2003.
40. F. Zhao and L. J. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, San Francisco, 2004.
41. Z. Zhou, S. Das, and H. Gupta. Connected K-coverage problem in sensor networks. In *Proceedings of the 13th Intl. Conf. on Comp. Commun. and Networks*, pp 373–378, 2004.