Efficient Sensor Placement for Surveillance Problems*

Pankaj K. Agarwal Esther Ezra Shashidhara Ganjugunte

Department of Computer Science, Duke University

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 guards. Given a parameter $\varepsilon > 0$, let $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \varsigma$. We prove that one can compute a set L of $O(\lambda \log^2 \lambda)$ *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 guards, 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. 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. For a set $X \subseteq P$, define $V(X) = \bigcup_{x \in X} V(x)$ and $\mathcal{V}(X) = \{V(x) \mid x \in X\}$. We define a function $\tau : P \to \mathbb{N}$, 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

*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.

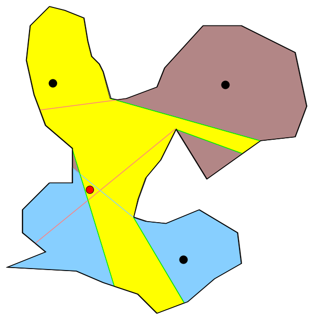

Figure 1: A set of landmarks and their respective visibility polygons.

p, we refer to the problem as the *uniform coverage* problem.

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 [19] that the communication is in infrared signals, (ii) in an urban environment, wireless transmissions are stronger inside streets (due to the reflections from walls) while lost of energy occurs in 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 [7, 13, 14, 17, 20, 21, 23, 25]. It is beyond the scope of this paper to mention all the related work, so we briefly mention a few of them. 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 [25]. The problem is known to be NP-hard and efficient approximation algorithms are known. 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. [14]. These methods are particularly useful for monitoring spatial phenomena.

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 [11], 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 [6, 9] and the references therein. Gonz*á*lez-Banos and Latombe [13] 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 approached have been widely used in the context of path planning. Efrat *et al.* [10] 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.

Our results. We follow a widely used approach for sensor coverage, i.e., sampling a finite a 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. 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. For a given parameter $\varepsilon > 0$, set $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \varsigma$. We prove that one can choose a set L of $m = O(\lambda \log^2 \lambda)$ 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 $O(nm^2 h\varsigma \log(mn) \log m)$ expected time. The expected running time of our algorithm can be improved to $O(nm^2\varsigma \log(mn))$. 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. 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.

For simplicity and due to lack of space, we assume r, the sensing radius to be ∞ for most of the paper, and then briefly sketch in Section 4 how to extend our results to finite sensing radius. Section 2 proves the sampling theorem (under the assumption $r = \infty$), and Section 3 describes the greedy algorithms. In Section 4, we briefly mention various extensions of our algorithm. 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. For simplicity, we assume the cover radius to be infinity. We begin by reviewing a few concepts from random sampling and statistical learning theory [22]. Let X be a (possibly infinite) set of objects, let μ be a measure function on X, and let $\mathcal{R}$ be a (possibly infinite) family of subsets of X. The pair $\Sigma = (X, \mathcal{R})$ is called a *range space*. A finite subset $Y \subseteq X$ *shatters* 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 VC-dim(Σ), is defined to be the largest size of a set that shatters 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. The VC-dimension of $\mathbb{V}$ is known to be $O(1 + \log \varsigma)$ [24].

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 $\mu(r) \geq \varepsilon\mu(X)$. A seminal result by Haussler and Welzl [16] shows that if 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 $1 - \delta$. We conclude this discussion by mentioning a useful property of VC-dimension: If $\Sigma_1 = (X, \mathcal{R}_1), \ldots, \Sigma_k = (X, \mathcal{R}_k)$ are k range spaces and VC-dim$(\Sigma_i) \leq d$ for all $i \leq k$, then the VC-dimension of the range space $\Sigma = (X, \mathcal{R})$, where $\mathcal{R} = \{\bigcup_{i=1}^k r_i \mid r_1 \in \mathcal{R}_1, \ldots, r_k \in R_k\}$, is $O(kd)$. We now state and prove the main result of this section.

Theorem 2.1 (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. Set $\lambda := \lambda(h, \varepsilon) = (h/\varepsilon) \log \varsigma$. Let $L \subset P$ be a random subset of $c_1 \lambda \ln^2 \lambda$ 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 $(1 - \varepsilon)$ fraction of the area of P with probability at least $1 - 1/\lambda$.*

Proof: Set $k = c_2 h \ln \lambda$. Let $\bar{\mathcal{V}}^k = \left\{ P \setminus \bigcup_{i=1}^k V(x_i) \mid x_1, \ldots, x_k \in P \right\}$, that is, each range in $\bar{\mathcal{V}}^k$ is the complement of the union of (at most) k coverage regions. Set $\bar{\mathbb{V}}^k = (P, \bar{\mathcal{V}}^k)$. Since VC-dim$(\mathbb{V}) = O(\log \varsigma)$, the above discussion implies that VC-dim$(\bar{\mathbb{V}}^k) = O(k \log \varsigma)$. Therefore if we choose a random subset $L \subset P$ of size $O((k/\varepsilon) \log(\varsigma) \log((k \log \varsigma)/\varepsilon))$, then L is an ε-net of $\bar{\mathbb{V}}^k$ with probability at least $1 - \varepsilon/(k \log \varsigma) \geq 1 - 1/\lambda$. Let S be a set satisfying the assumptions in the lemma, that is, S is a set of at most k guards 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. □

The Sampling Theorem suggests the following simple approach to compute a placement for P. Given a set L of landmarks, the procedure GREEDY-COVER(P, L), described in the next section, computes a cover of L of size $O(\tilde{h}\log|L|)$, where $\tilde{h} \leq h$ is the number of sensors needed to cover L.

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

Figure 2: A landmark based algorithm for covering P.

Remark: We note that the actual number of guards that the algorithm computes is an approximation to the optimal number of guards 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 GREEDY-COVER(P, L) that computes a placement of sensors to cover a set $L \subset P$ of m landmarks. 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., [18].

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 $O(m_i^2 n\varsigma)$ vertices, edges, and faces, and that $\mathcal{A}(L_i)$ can be computed in time $O(m_i^2 n\varsigma\log n)$. 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 $O(m^2 n\varsigma|S|\log n)$.

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 structures; see [1]. The asymptotic running time now becomes $O(m^2 n\varsigma\log^2 n)$, 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 $O(\tilde{h}\ln|L|)$ [1]. 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 [2, 4], 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 jth 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 $O(n(m_i/\Delta)^2\varsigma \log n \log^2 m)$.

This procedure computes an overly 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 visibility polygons (that is, the vertices and edges of the overlay that are not contained in the interior of any polygon) is significantly smaller than the complexity of the entire overlay. Specifically, Gawali *et al.* [12] showed that, if the boundary of P is connected, the overall number of vertices and edges that appear along the boundary of the union of the polygons in $V(L)$ is only $O(mn + m^2)$. In that 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 $\Omega(m^2 n\varsigma)$ in the worst case; we provide the lower bound construction in the full paper.

4 Extensions

We briefly describe the following extensions and omit the details because of lack of space.

Finite coverage radius. The algorithms presented above assume that a sensor can cover points that are arbitrarily far if they are visible from the sensor. In order to extend the Sampling Theorem to the case of finite coverage radius, we generalize the range-space $\mathbb{V}^k$ in Section 2 to support intersections of visibility regions and discs, and show that the VC-dimension increases by a multiplication factor of three. Therefore the asymptotic size of landmarks remains the same. Next, we extend the analysis of the greedy algorithm to this case by showing that the size of overlay of coverage regions only increases by a constant factor with respect to that of the original problem, so the running time of the algorithms remains the same.

The case of cameras. Another natural extension of the coverage region is the case in which each sensor can cover an area that is contained in a *cone* of directions (e.g., camera). Our analysis can easily be extended to this case as well by modifying the range space appropriately and bounding the complexity of the overlay of coverage regions.

Weighted coverage. We are now given a weight function $w : P \to \mathbb{R}^+$. 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 $\bar{\mathbb{V}}_t^k = \{\bar{V}_t(X) \mid X \subset P \text{ is a set of } k \text{ points}\}$ and $\bar{\mathbb{V}}_t^k = (P, \bar{V}_t^k)$. We prove VC-dim$(\bar{\mathbb{V}}_t^k) = O(k^t t^2 \log(\varsigma) \log(kt \log \varsigma))$, 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.

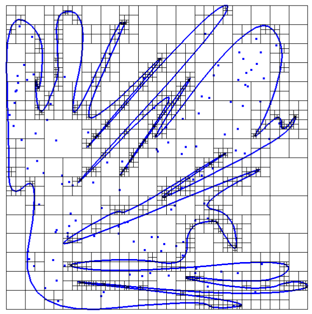

Figure 3: A quadtree representing a polygon. The dots are the landmarks.

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 case.

Approximated coverage region. Recall that the algorithms compute the overlay of the coverage region of the landmarks. Computing the exact overlay is, however, inefficient in practice, and might result in robustness issues, see [15] 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*.

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 we are left with at most a single polygon vertex inside each cell; see, e.g., [8] for further details about quadtrees. 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. This approximated representation improves as we refine the decomposition. In fact, in our implementation, we use heuristics to further refine a quadtree cell, so as to get a relatively good approximation for P. Having the pixels at hand, we represent each of the visibility polygons 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, we thus fix a threshold parameter $0 < \alpha < 1$, and report that s sees C if it sees at least α of its area.

The quadtree implementation, albeit yielding an approximated representation, is useful in the sense that it eliminates the need to construct the arrangement of the polygons, and thus results in good performance in practice, as indicated by our experimental results.

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 [5] for generating random simple polygons, and for the manually produced polygons, we use the inputs of Amit *et al.* [3]. Out data sets are listed, as follows:

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

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

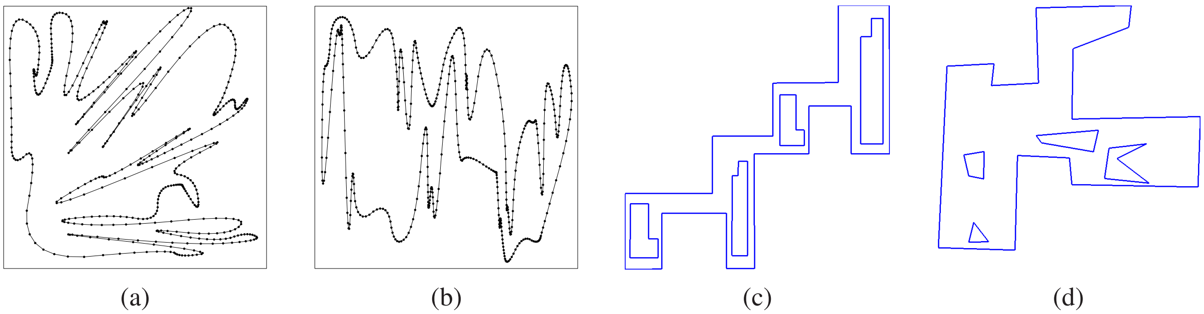

Figure 4: The data sets. (a) Bouncy. (b) x-monotone. (c) Orthogonal. (d) General.

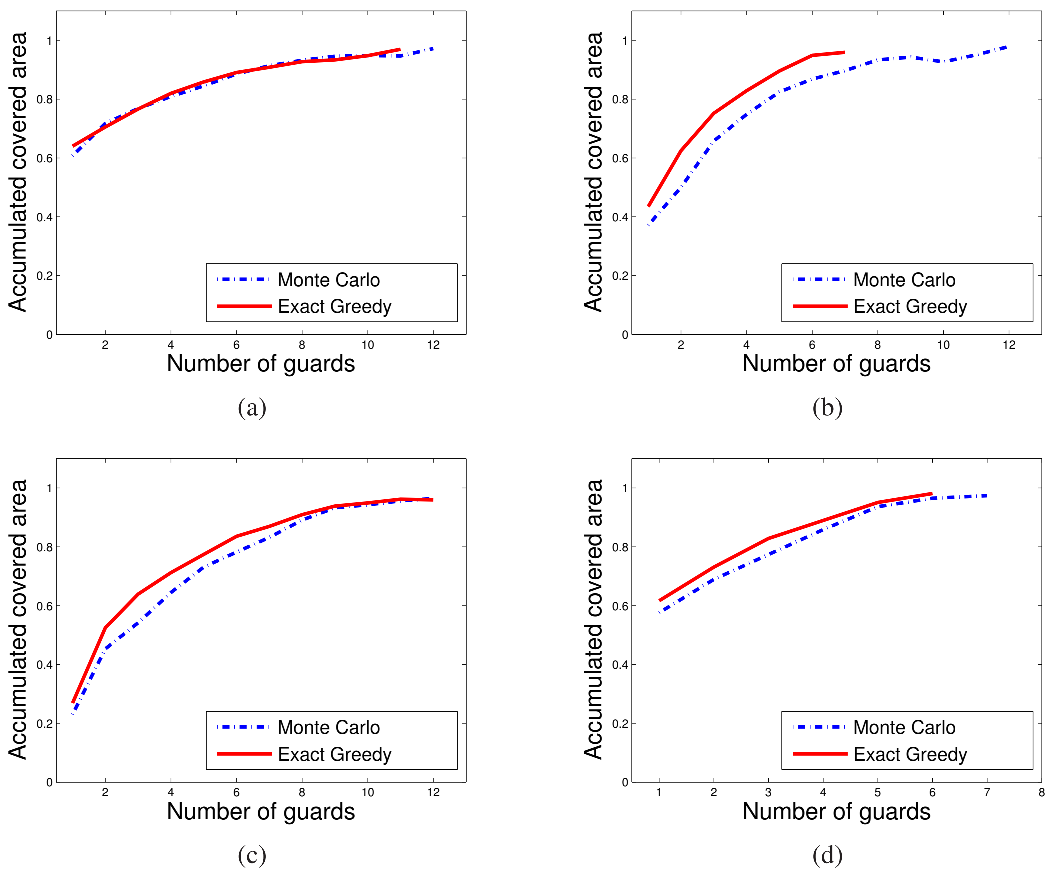

Figure 5: The (average) accumulated covered area (y-axis) at each iteration (x-axis) for the Exact Greedy and the Monte Carlo algorithms, for each of the data sets.

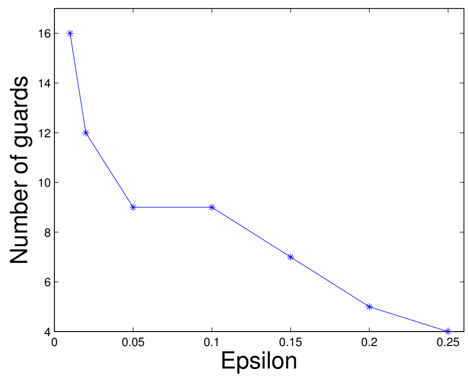

Figure 6: The number of guards as a function of ε for the bouncy polygon.

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. (iv) The convergence ratio of the algorithm; at each iteration i we measure the ratio between the overall area just covered (over all iterations $1, \ldots, i$) 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 $\varepsilon = 0.05$.

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 between two and eight times smaller with respect to that of the Exact Greedy algorithm.

The number of landmarks in each of our experiments is small, as expected. Specifically, the expected number of landmarks increases almost linearly as a function of $1/\varepsilon$. For $\varepsilon = 0.05$, the expected numbers of landmarks, chosen by the Monte-Carlo algorithm, are 75 for the bouncy polygon, 59 for the x-monotone polygon, 56 for the orthogonal polygon, and 33 for the general one. These numbers do not depend on the size n of the polygons, but only in the (optimal) size of the cover and ε.

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 Figure 5) 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 twice 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, and in particular, for the orthogonal one, since it is difficult to find a single guard that sees most of the polygon, and thus any algorithm that aims to cover this polygon has to use a relatively large number of guards; see once again Figure 4(c).

In addition to the results reported above, we have tested the dependency of the number of guards in ε. As one may expect, the number of guards should decrease as we enlarge ε. In Figure 6 we present these results for the bouncy polygon under the Monte Carlo algorithm, as indicated by our experiments, the number of guards 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.

Acknowledgments. The authors wish to thank Alon Efrat for very useful comments on this paper. The authors also wish to thank Sariel Har-Peled for useful discussions, and for Eli Packer and Martin Held for the input polygons used in Section 5.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2000.

[9] A. Efrat and S. Har-Peled. Guarding galleries and terrains. In *2nd IFIP Internat. Conf. Theo. Comp. Sci*, 181–192, 2002.

[10] 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.

[11] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001

[12] 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.

[13] 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.

[14] 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.

[15] D. Halperin. Robust Geometric Computing in Motion. *International Journal of Robotics Research*, 21(3):219–232, 2002.

[16] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[17] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[18] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[19] 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.

[20] 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.

[21] J. O'Rourke. *Art gallery theorems and algorithms.* The Internation Series of Monographs in Computer Science. Oxford University Press, New York, NY, 1987.

[22] J. Pach and P. K. Agarwal. *Combinatorial geometry*. Wiley-Interscience, San Diego, 1995

[23] J. Urrutia. Art Gallery and illumination problems. In *Handbook of Computational Geometry*, (J. Sack and J. Urrutia, eds), Elsevier, Amsterdam, pages 973–1027, 2000.

[24] P. Valtr. Guarding galleries where no point sees a small area. Israel J. Math, 104:1–16, 1998.

[25] F. Zhao and L. J. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, San Francisco, 2004.