Guarding Scenes against Invasive Hypercubes*

Mark de Berg† Haggai David‡ Matthew J. Katz‡ Mark Overmars†
A. Frank van der Stappen† Jules Vleugels†

Abstract

In recent years *realistic input models* for geometric algorithms have been studied. The most important models introduced are *fatness*, *low density*, *unclutteredness*, and *small simple-cover complexity*. These models form a strict hierarchy. Unfortunately, small simple-cover complexity is often too general to enable efficient algorithms. In this paper we introduce a new model based on *guarding sets*. Informally, a guarding set for a collection of objects is a set of points that approximates the distribution of the objects. Any axis-parallel hyper-cube that contains no guards in its interior may intersect at most a constant number of objects. We show that guardable scenes fit in between unclutteredness and small simple-cover complexity. They do enable efficient algorithms, for example a linear size binary space partition. We study properties of guardable scenes and give heuristic algorithms to compute small guarding sets.

1 Introduction

In recent years *realistic input models* for geometric algorithms have been studied extensively (see for example [2, 10, 11]). The idea is to define restrictions on the allowed sets of geometric objects such that more efficient algorithms can be devised to deal with them. Clearly, the goal is to make the restrictions as mild as possible, while keeping the algorithm running times as low as possible. The most important models introduced are *fatness*, *low density*, *unclutteredness*, and *small simple-cover complexity*. De Berg et al. [6] showed that these models form a strict hierarchy in the sense that fatness implies low density, which in turn implies unclutteredness, which implies small simple-cover complexity, and that the reverse implications are false. So small simple-cover complexity is the most general model. Unfortunately for many geometric problems it is unclear how to use this property to derive fast algorithms. For example, de Berg [2] presents an algorithm for computing a linear-size binary space partition (BSP) for uncluttered d-dimensional scenes. He also describes a linear-size data structure, based on the underlying BSP tree, supporting logarithmic-time point location queries. For sets with small simple-cover complexity (for dimensions larger than 2) no method for computing linear sized BSP's is known. Also in a recent paper [5] we bound the complexity of the free space for

*Preliminary versions of parts of this paper appeared in [3, 4].

†Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands. E-mail: {markdb,markov,frankst,jules}@cs.uu.nl . Supported by the Dutch Organization for Scientific Research (N.W.O.) and by the ESPRIT IV LTR Project No. 21957 (CGAL).

‡Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. E-mail: {davidcha,matya}@cs.bgu.ac.il . Supported by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities.

a bounded-reach robot with f degrees of freedom among n obstacles. For a 3-dimensional workspace, low density implies a linear bound on the complexity. For uncluttered scenes this bound becomes $\Theta(n^{2f/3} + n)$, while for scenes with simple-cover complexity, the bound can be as bad as for arbitrary scenes ($\Theta(n^f)$).

So it seems that simple cover complexity is often too general to obtain complexity reduction. In this paper we define a new realistic input model, using *guarding sets*. Informally, a guarding set for a collection of objects is a set of points that approximates the distribution of the objects. Any axis-parallel hyper-cube that contains no guards in its interior may intersect at most a constant number of objects. For example, suppose we have a set of n disjoint discs in the plane. Now consider the set of $5n$ points obtained by choosing for each of the discs its center point and its topmost, bottommost, leftmost, and rightmost points. It is easy to verify that any axis-parallel square that does not contain a guard in its interior can intersect at most 4 discs. (We assume here that the discs and squares are open.)

A scene is called *guardable* if it has a linear size guarding set. We show that there exist scenes that are guardable but that are not uncluttered. We then study the relation between being guardable and having small simple-cover complexity. We prove that in the plane these two properties are equivalent. However, in higher dimensions, having small simple-cover complexity is more general than being guardable, that is, a guardable scene has small simple-cover complexity, but the opposite statement is not necessarily true. Hence, in the hierarchy of realistic input models guardable scenes lie between unclutteredness and small simple-cover complexity.

We show that guardable scenes have linear size binary space partition trees. In another paper [5] we show that for 3-dimensional guardable scenes the complexity of the free space for a bounded-reach robot with f degrees of freedom is $\Theta(n^{2f/3} + n)$, like for uncluttered scenes. Hence, guardable scenes do allow for efficient geometric algorithms.

The paper is organised as follows. In Section 2 we formally define the notion of a guarding set and prove a number of properties of such sets. In particular we prove that any fat range that does not contain guards intersects only a constant number of objects. Also we discuss an interesting connection between guarding sets and the well-known concept of ε-nets. In Section 3 we study where guardable scenes fit into the hierarchy of realistic input models. In Section 4 we study the computation of small guarding sets. To use de Berg's linear size BSP-tree we need to have such a set. (It is not enough to know that such a set exists.) We describe three heuristic algorithms for computing a (hopefully) small guarding set and evaluate them both theoretically and experimentally. Finally, in Section 5 we draw some conclusions and indicate directions for further research.

2 Guarding sets

Before we define guarding sets, we need to fix some terminology. Whenever we talk about squares, cubes, rectangles, and so on, we implicitly assume that they are axis-aligned. Furthermore, all squares and other geometric objects we consider are (relatively) open; in particular, if we talk about a point lying in a square, we mean that the point lies in the interior of the square. Furthermore, all objects we consider are assumed to have constant complexity: they are connected compact subsets of $\mathbb{R}^d$, bounded by a constant number of algebraic surface patches of constant maximum degree. The volume of an object o is denoted by $\mathrm{vol}(o)$.

A guarding set for a collection of objects is, loosely speaking, a set of points that approximates the distribution of the objects. More precisely, guarding sets are defined as follows.

Definition 2.1 *Let $\mathcal{O}$ be set of objects in $\mathbb{R}^d$, let $\mathcal{R}$ be a family of subsets of $\mathbb{R}^d$ called ranges, and let κ be a positive integer. A set $\mathcal{G}$ of points is called a* κ-guarding set for $\mathcal{O}$ against $\mathcal{R}$, *if any range from $\mathcal{R}$ not containing a point from $\mathcal{G}$ intersects at most κ objects from $\mathcal{O}$.*

We often call the points in $\mathcal{G}$ *guards*, and refer to ranges not containing guards as *empty ranges*.

Let us look at an example. Suppose the set $\mathcal{O}$ is a set of n disjoint discs in the plane, and that the family of ranges is the family of all (axis-aligned) squares. For a disc D, define $\mathcal{G}_D$ to be the set of the following five points: the center of D plus the topmost, bottommost, leftmost, and rightmost point of D. When a square σ intersects a disc D, and σ does not contain a point from $\mathcal{G}_D$, then D contains a vertex of σ. So σ can intersect at most 4 discs. Hence, the set $\mathcal{G} := \{\mathcal{G}_D : D \in \mathcal{O}\}$ is a 4-guarding set of size $5n$ for $\mathcal{O}$ against the family of squares.

Another example. Again $\mathcal{O}$ is a set of n disjoint discs in the plane, but this time $\mathcal{R}$ is the family of all lines. Then it can happen that no finite κ-guarding set exists for $\kappa < n$: if, for instance, the centers of the discs are collinear, then there are infinitely many lines stabbing all the discs and no finite guarding set can capture all these lines.

Fortunately, when we later study the relation between guarding sets and some of the realistic input models proposed in the literature, we shall see that in many settings there exist linear-size κ-guarding sets against hypercubes, where κ is a constant. But first we prove some basic properties of guarding sets, and discuss the relation between guarding sets and a well-known concept from the literature, namely ε-nets.

2.1 Basic properties of guarding sets

From the second example above it seems wise to restrict ourselves to fat ranges, since long and skinny ranges such as lines are likely not to admit finite-size guarding sets. We shall mainly consider one type of fat ranges, namely hypercubes. This is justified by Theorem 2.7 below, which states that the same number of guards (asymptotically) is needed against hypercubes as against fat ranges. We use the following definition of fatness. (There are several different definitions of fatness which are all more or less equivalent—at least for convex objects. The definition below is similar to the one used in van der Stappen's thesis [11].)

Definition 2.2 *Let $\mathcal{P} \subseteq \mathbb{R}^d$ be an object and let β be a constant with $0 < \beta \leqslant 1$. Define $U(\mathcal{P})$ as the set of all balls centered inside $\mathcal{P}$ whose boundary intersects $\mathcal{P}$. We say that the object $\mathcal{P}$ is* β-fat *if for all balls $b \in U(\mathcal{P})$:*

$$\text{vol}(\mathcal{P} \cap b) \geqslant \beta \cdot \text{vol}(b).$$

The fatness *of $\mathcal{P}$ is defined as the maximal β for which $\mathcal{P}$ is β-fat.*

We sometimes omit the fatness parameter β, and call an object *fat* if it is β-fat for some not-too-small constant β.

We shall prove that, from an asymptotic point of view, it doesn't matter whether one wants to guard against hypercubes only, or against arbitary fat convex ranges: the same number of guards (asymptotically) suffices.

Construction of the guarding set. Let $\mathcal{O}$ be a set of n objects in $\mathbb{R}^d$. Let $\mathcal{G}_S$ be a κ-guarding set of size m for $\mathcal{O}$ against hypercubes. We want to construct a guarding set for $\mathcal{O}$ against β-fat convex ranges. We need the following result.

Lemma 2.3 (de Berg [2]) *Let $\mathcal{P}$ be a set of points in $\mathbb{R}^d$, and let $\sigma(\mathcal{P})$ be a hypercube containing all points from $\mathcal{P}$. Then there exists a covering of $\sigma(\mathcal{P})$ by $O(2^d|\mathcal{P}|)$ hypercubes without points from $\mathcal{P}$ in their interior, such that no point in $\mathbb{R}^d$ is contained in more than 2^d hypercubes from the covering.*

The guarding set $\mathcal{G}_F$ for $\mathcal{O}$ against fat ranges is now constructed as follows. Let $\sigma(\mathcal{G}_S)$ be a hypercube containing all guards from $\mathcal{G}_S$. We cover $\sigma(\mathcal{G}_S)$ by a collection $\mathcal{S}$ of hypercubes not containing guards from $\mathcal{G}_S$, according to Lemma 2.3. For each hypercube $\sigma \in \mathcal{S}$, we define a collection $\mathcal{G}_\sigma$ of guards, as follows. Define $v(\sigma)$, the *vicinity* of σ, to be the hypercube obtained by scaling σ with a factor 5 with respect to its center. For a facet f of σ, let f' denote the corresponding facet of $v(\sigma)$, and let f^* denote the translated copy of f' such that f^* still lies inside $v(\sigma)$ and is midway between f and f'—see Figure 1. Note that f^* is a

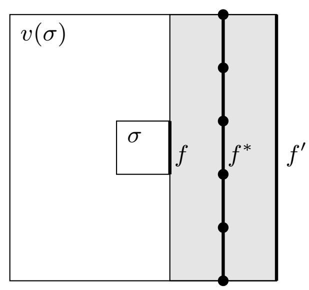

Figure 1: Illustration of the definition of the collection $\mathcal{G}_f$ of guards defined for a facet f.

$(d-1)$-dimensional hypercube. Subdivide f^* into a regular grid of M^{d-1} cells; here M is a parameter we will choose later, depending on the fatness β of the ranges we want to guard against. The guarding set we define for f is the set $\mathcal{G}_f$ of $(M+1)^{d-1}$ grid points on f^*, and the set $\mathcal{G}_\sigma$ is the union of the $2d$ sets $\mathcal{G}_f$ over all facets f of σ. Figure 1 illustrates this. Finally, the complete guarding set $\mathcal{G}_F$ consists of the union of the sets $\mathcal{G}_\sigma$ for all $\sigma \in \mathcal{S}$. Since the number of hypercubes in $\mathcal{S}$ is $O(2^d m)$, we get the following lemma.

Lemma 2.4 *The guarding set $\mathcal{G}_F$ consists of $O(dM^{d-1}2^d m)$ points.*

The analysis. Next we prove that the set $\mathcal{G}_F$ constructed above is a guarding set against fat ranges. We start with the following technical lemma.

Lemma 2.5 *Let U be a hypercube in $\mathbb{R}^d$, and assume U is subdivided into a regular grid of M^d cells for some positive integer M. Let R be a convex set contained in U and not containing any of the grid points. Then $\mathrm{vol}(R) \leqslant d \cdot \mathrm{vol}(U)/M$.*

Proof: We prove the lemma for unit hypercubes, from which the general result readily follows. The proof is by induction on d. We write $\mathrm{vol}_k(\cdot)$ to denote the k-volume of a k-dimensional set in $\mathbb{R}^d$ (that is, its volume when considered as a subset of its affine hull).

For $d = 1$, the lemma is obvious: in this case U is a unit segment subdivided into M subsegments of length $1/M$ and R must be contained in one of them, so its length is at most $1/M$.

Now let $d > 1$. Define $h(t)$ to be the hyperplane orthogonal to the x_1-axis at distance t from the origin, that is, $h(t) := (x_1 = t)$. Define $\alpha : [0 : 1] \to \mathbb{R}$ to be the function giving the $(d-1)$-volume of the cross-section of R and $h(t)$, that is, $\alpha(t) := \mathrm{vol}_{d-1}(R \cap h(t))$. Because R is convex, the function α is unimodal. (Avis et al. [1] prove this for convex polytopes using the Brunn-Minkowski theorem, but the proof holds verbatim for arbitrary convex sets.) Let t^* be a value of t for which $\alpha(t)$ achieves its maximum. Let i be the integer such that $(i-1)/M \leqslant t^* \leqslant i/M$. The cross-section of $h((i-1)/M)$ with U is a $(d-1)$-dimensional hypercube subdivided into a regular grid of M^{d-1} cells. Hence, by the induction hypothesis we have $\alpha((i-1)/M) \leqslant (d-1)/M$. Together with the unimodality of $\alpha(t)$ this implies that

$$\alpha(t) \leqslant \alpha((i-1)/M) \leqslant (d-1)/M \qquad \text{for all } t \leqslant (i-1)/M.$$

Similarly, we have

$$\alpha(t) \leqslant \alpha(i/M) \leqslant (d-1)/M \qquad \text{for all } t \geqslant i/M.$$

Finally, the volume of the part of R between $h((i-1)/M)$ and $h(i/M)$ is obviously bounded by $1/M$, which is the volume of the part of U between these hyperplanes. It follows that

$$\begin{aligned}
\mathrm{vol}_d(R) &= \int_0^1 \alpha(t)dt \\
&= \int_0^{(i-1)/M} \alpha(t)dt \;+\; \int_{(i-1)/M}^{i/M} \alpha(t)dt \;+\; \int_{i/M}^{1} \alpha(t)dt \\
&\leqslant (i-1)/M \cdot (d-1)/M \;+\; 1/M \;+\; (1-i/M) \cdot (d-1)/M \\
&\leqslant d/M.
\end{aligned}$$

□

Next we show that a range intersecting a hypercube σ must either be contained in the vicinity of σ, or it must be skinny. In the next lemma, ω_d is a constant such that the volume of a hypershere of radius r in $\mathbb{R}^d$ is $\omega_d r^d$.

Lemma 2.6 *Let σ be a hypercube, let $v(\sigma)$ be its vicinity, and let $\mathcal{G}_\sigma$ be the set of guards defined for σ as explained above. Let R be a convex β-fat range intersecting σ and not fully contained in $v(\sigma)$. Then*

$$\beta \leqslant \frac{5^{d-1}(d-1)(2^{d+1}-2)(3/2)^d}{\omega_d M}.$$

Proof: Let R be a convex β-fat range intersecting σ and not fully contained in $v(\sigma)$. Then R must intersect some facet f' of $v(\sigma)$. Let f be the facet of σ corresponding to f', and let $h(f)$ be the hyperplane containing f. Define B to be the part of $v(\sigma)$ lying to the same side of $h(f)$ as f'. Figure 2 illustrates these definitions for the planar case. Our strategy is to bound $\mathrm{vol}(B \cap R)$, and use that to derive a lower bound on β.

Assume without loss of generality that σ is the unit hypercube. As before, we let f^* denote the $(d-1)$-dimensional hypercube situated midway between f and f'. Recall that

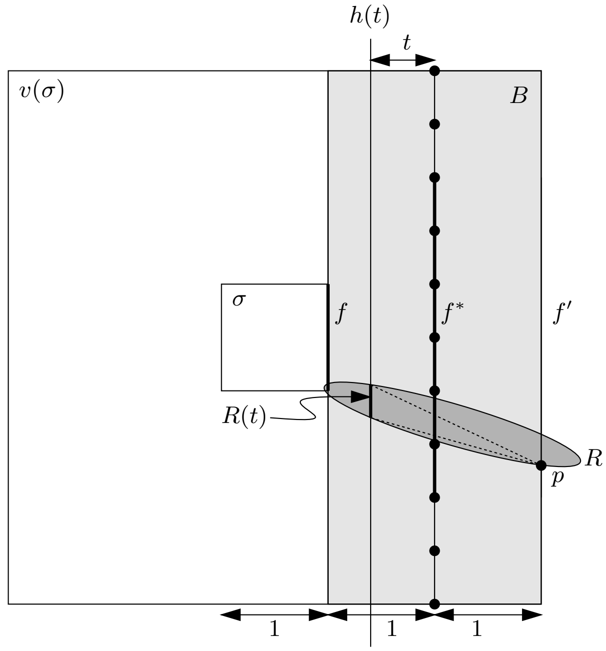

Figure 2: Illustration for the proof of Lemma 2.6.

the guards we defined for f are the vertices of a regular grid on f^* with M^{d-1} cells. Let p be an arbitrary point in $R \cap f'$. For $0 \leqslant t \leqslant 1$, define $h(t)$ to be the hyperplane parallel to f^* at distance t from f^* and between f and f^*. Define $R(t) := R \cap h(t)$. By convexity of R, the projection of $R(t)$ onto f^* with p as center must be contained in R. But f^* is a $(d-1)$-dimensional hypercube of edge length 5 on which we placed a grid of M^{d-1} guards. Hence, by Lemma 2.5 we have $\text{vol}_{d-1}(R \cap f^*) \leqslant A$, where $A := 5^{d-1}(d-1)/M$. Since $R(t)$ is a scaled copy of its projection onto f^*, with scaling factor $1+t$, this implies that

$$\text{vol}_{d-1}(R(t)) \leqslant A(1+t)^{d-1}.$$

We can apply the same arguments to bound the volume of the part of $R \cap v(\sigma)$ lying between f^* and f'. We conclude that

$$\text{vol}(B \cap R) \leqslant 2\int_0^1 A(1+t)^{d-1} dt \leqslant A(2^{d+1}-2).$$

Next we argue that there is a hypersphere $b \subset B$ of radius $2/3$ whose center lies in R. Indeed, any segment connecting f' to σ intersects f^* in a point at distance at least $2/3$ from the boundary of B. Since R intersects f and σ, this implies that there is a hypershere b with the stated properties. Since R is not fully contained in b, we have

$$\beta \leqslant \text{vol}(b \cap R)/\text{vol}(b) \leqslant A(2^{d+1}-2)/(\omega_d(2/3)^d).$$

Plugging in the value $A = 5^{d-1}(d-1)/M$, we get the desired result. □

We can now prove the main result of this section. To construct a guarding set $\mathcal{G}_F$ against

β-fat ranges, we proceed as described above. The value for M that we use in the construction is

$$M = 1 + \left\lceil \frac{5^{d-1}(d-1)(2^{d+1}-2)(3/2)^d}{\omega_d \beta} \right\rceil .$$

According to the previous lemma, this means that any β-fat convex range intersecting a hypercube $\sigma \in \mathcal{S}$ must be contained in the vicinity of σ.

Theorem 2.7 *Let $\mathcal{O}$ be set of objects in $\mathbb{R}^d$, where d is a constant, and suppose there is a κ-guarding set of size m for $\mathcal{O}$ against hypercubes. Let β be a constant with $0 < \beta \leqslant 1$. Then there is a set of cm points that is a κ'-guarding set against convex β-fat ranges, for constants c (depending on β and d) and κ' (depending on κ and d).*

Proof: The fact that the guarding set $\mathcal{G}_F$ consists of cm points for a constant c depending on β and d follows immediately from Lemma 2.4 and the choice of M. We now prove that $\mathcal{G}_F$ is indeed a guarding set against convex β-fat ranges.

Let R be a convex β-fat range. Consider the set $\mathcal{S}$ of hypercubes used in the construction of the guarding set $\mathcal{G}_F$. Let $\sigma \in \mathcal{S}$ be the smallest hypercube intersecting R. Assume without loss of generality that σ has unit size. We know that the vicinity $v(\sigma)$ contains R. Let T be the hypercube obtained by scaling σ by a factor of 7 with respect to its center, or, in other words, by growing $v(\sigma)$ by the side length of σ in all axis-parallel directions—see Figure 3. Notice that $\text{vol}(T) = 7^d$. Now let $\sigma' \in \mathcal{S}$ be another hypercube intersecting R. By

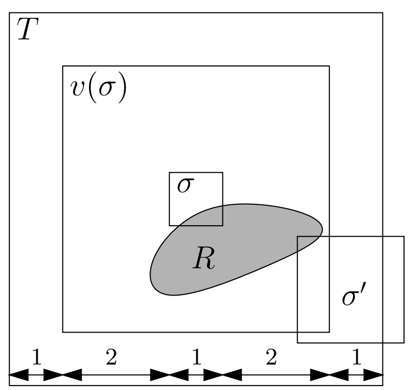

Figure 3: Illustration for the proof of Theorem 2.7.

assumption, $\text{vol}(\sigma') \geqslant 1$. As R is contained in $v(\sigma)$, the hypercube σ' must intersect $v(\sigma)$, which implies that it shares at least a unit volume with T. Since no point is, by construction of $\mathcal{S}$, covered by more than 2^d hypercubes, the sum of the volumes of the intersections of T and the hypercubes $\sigma' \in \mathcal{S}$ cannot exceed $2^d \cdot \text{vol}(T) = 14^d$. Since the hypercubes in $\mathcal{S}$ intersect at most κ objects, the number of objects intersecting $v(\sigma)$ is bounded by $\kappa' := 14^d \kappa$. Clearly this is an upperbound on the number of objects intersecting R as well. $\square$

Theorem 2.7 implies that from an asymptotic point of view, it does not matter whether we study guarding against hypercubes or against any other convex shape such as balls. Hence, from now on we restrict our attention to hypercubes.

The next theorem deals with the connection between the distribution of objects and the distribution of guards. We show that the number of guards in a hypercube is at least linear in the number of objects intersecting the hypercube, assuming κ is a constant. The reverse of the theorem is not necessarily true: a hypercube that is intersected by only a few objects may contain many guards.

Theorem 2.8 *Let $\mathcal{G}$ be a κ-guarding set against hypercubes for a set $\mathcal{O}$ of objects. Any hypercube with exactly g guards in its interior is intersected by $O(\kappa g)$ objects.*

Proof: Let g be the number of guards in some hypercube σ. According to Lemma 2.3 there is a partitioning of σ into $O(g)$ empty hypercubes and L-shapes. Each hypercube is intersected by at most κ objects and each L-shape by at most $(2^d - 1)\kappa$ objects (because it can be covered by $2^d - 1$ empty hypercubes). Hence, σ is intersected by $O(\kappa g)$ objects. $\square$

Although the theorem is not surprising and easy to prove, it has an interesting consequence which we describe in the next subsection.

In another paper [5] we establish another, more surprising, relation between the distribution of the objects and the distribution of the guards. Again we look at hypercubes, but this time we only look at objects that are at least as large as the hypercube, and not only consider the guards inside the hypercube but also the ones close to it (i.e., in its vicinity). We show that the number of relatively large objects intersecting a hypercube σ cannot be more than (roughly) $O(g^{1-1/d})$, where g is the number of guards in its vicinity, assuming κ is a constant.

In general, we are especially interested in scenes for which there exists a κ-guarding set of size cn, where κ and c are small constants. We say that such scenes are *guardable*. Guardable scenes of polyhedral objects admit a linear-size BSP, and a linear-size data structure for logarithmic-time point location; this follows directly from the results by de Berg [2]. Furthermore, the worst-case free-space complexity of a (bounded-reach) robot in a guardable scene is considerably smaller than the worst-case free-space complexity in unrestricted scenes [5].

2.2 Relation with ε-nets

In a geometric setting, one can define *ε-nets* as follows [8, 9]. A subset N of a given set $\mathcal{O}$ of n objects in $\mathbb{R}^d$ is called an ε-net with respect to a family $\mathcal{R}$ of ranges, if any 'empty' range, i.e., any range not intersecting an object from N, intersects at most εn objects from $\mathcal{O}$. (The actual definition of ε-nets is more general than the definition we just gave: it is purely combinatorial.)

If we take $\kappa = \varepsilon n$ then the notions of κ-guarding set and ε-net become very similar. The major difference is that a guarding set consists of points, whereas an ε-net is a subset of the given set $\mathcal{O}$ (or, in the case of a weak ε-net, an arbitrary set of objects from this class). This means that sometimes a finite guarding set does not exist, whereas an ε-net always exists. On the other hand, the fact that guarding sets consist of points makes further processing easier from an algorithmic point of view.

For the case where the scene consists of fat objects, we can formulate an ε-net-type theorem. Let $\mathcal{O}$ be a set of n objects, and suppose we have a κ-guarding set $\mathcal{G}$ of size $m = O(n)$ against hypercubes for $\mathcal{O}$. Now suppose we wish to have a κ'-guarding set for $\mathcal{O}$ for some $\kappa' > \kappa$, say for $\kappa' = \delta n$. Then we can obtain such a guarding set by taking an ε-net N for $\mathcal{G}$ with respect to the family of (open) hypercubes, for a suitable $\varepsilon = \Theta(\delta/\kappa)$: a hypercube not containing a point from N contains less than $\varepsilon m = O(\varepsilon n)$ guards from $\mathcal{G}$, so by Theorem 2.8 it intersects $O(\kappa\varepsilon n) = O(\delta n)$ objects. Because the underlying range space has finite VC-dimension, there is such an ε-net of size $O((\kappa/\delta)\log(\kappa/\delta))$ [9]. Combining this with the fact that any collection of disjoint fat objects has a linear-size guarding set for some constant κ—this set is simply the set of bounding box vertices of the objects, see also Section 3—we obtain the following ε-net-type result:

Theorem 2.9 *Let $\mathcal{O}$ be a set of n disjoint fat objects in $\mathbb{R}^d$. There exists a guarding set of $O((1/\delta)\log(1/\delta))$ points, such that any hypercube intersecting more than δn objects contains at least one guard.*

3 Guarding sets and realistic input models

In this section we study the connection between guarding sets and two of the recently proposed *realistic input models*: unclutteredness and simple-cover complexity. But first we formally define these two models.

Unclutteredness was introduced by de Berg [2]. The model is defined as follows.

Definition 3.1 *Let $\mathcal{O}$ be a set of objects in $\mathbb{R}^d$. We call $\mathcal{O}$ a κ-*cluttered *scene if any hypercube whose interior does not contain a vertex of one of the bounding boxes of the objects in $\mathcal{O}$ is intersected by at most κ objects in $\mathcal{O}$. The* clutter factor *of a scene is the smallest κ for which it is κ-cluttered.*

We sometimes call a κ-cluttered scene for which κ is a small constant *uncluttered*.

The following definition of simple-cover complexity is a slight adaptation of the definition of Mitchell et al. [10], as proposed by de Berg et al. [6]. Given a scene $\mathcal{O}$, we call a ball *δ-simple* if it intersects at most δ objects in $\mathcal{O}$.

Definition 3.2 *Let $\mathcal{O}$ be a set of objects in $\mathbb{R}^d$, and let $\delta > 0$ be a parameter. A* δ-simple cover *for $\mathcal{O}$ is a collection of δ-simple balls whose union covers the bounding box of $\mathcal{O}$. We say that $\mathcal{O}$ has* (s, δ)-simple-cover complexity *if there is a δ-simple cover for $\mathcal{O}$ of cardinality sn.*

We will say that a scene has *small simple-cover complexity* if there are small constants s and δ such that it has (s, δ)-simple-cover complexity.

It has been shown [6] that a scene consisting of fat objects is both uncluttered and has small simple-cover complexity. It has also been shown that small simple-cover complexity is more general than unclutteredness: an uncluttered scene has small simple-cover complexity, but the opposite is not always true.

We now investigate the relation between guardable scenes on the one hand, and uncluttered scenes and scenes with small simple-cover complexity of the other hand. We start with the relation with uncluttered scenes. It turns out that uncluttered scenes are always guardable, but that the reverse is (even in the plane) not true.

Theorem 3.3

(i) Let $\mathcal{O}$ be a set of n objects in $\mathbb{R}^d$. If $\mathcal{O}$ is κ-cluttered, then it has a κ-guarding set of size $2^d n$ against hypercubes.

(ii) For any $n > 4$, there is a planar scene of n objects that admits a 1*-guarding set of size* $6n$*, but that is not κ-cluttered for any $\kappa < \lfloor\sqrt{n}\rfloor$.*

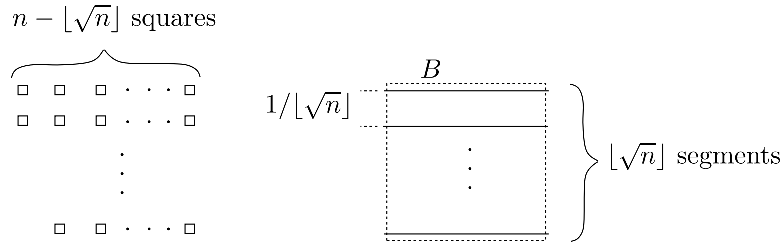

Figure 4: A scene that is guardable but not uncluttered.

Proof: Part (i) of the theorem immediately follows from the definitions: the $2^d n$ vertices of the bounding boxes of the objects in a κ-cluttered scene form a κ-guarding set against hypercubes.

Next we prove part (ii). Consider Figure 4. In this figure we have a collection of $\lfloor\sqrt{n}\rfloor$ horizontal line segments, each of unit length and at distance $1/\lfloor\sqrt{n}\rfloor$ from each other. The scene is completed by adding $n - \lfloor\sqrt{n}\rfloor$ tiny squares in a grid-like pattern so that the distance between the squares is larger then their side length. This scene is not κ-cluttered for any $\kappa < \lfloor\sqrt{n}\rfloor$, since the square B in the figure is empty (of bounding-box vertices) and is intersected by $\lfloor\sqrt{n}\rfloor$ objects. However, there is a 1-guarding set of size $5\lfloor\sqrt{n}\rfloor^2 + \lfloor\sqrt{n}\rfloor \leqslant 6n$ for this scene. It is obtained by placing $\sqrt{n}$ equally-spaced guards between each pair of consecutive line segments, plus a guard on each vertex of a square and of a line segment. $\square$

Next we investigate the relation between having small simple-cover complexity and being guardable. We prove that in the plane a scene has small simple-cover complexity if and only if it is guardable. But in higher dimensions, having small simple-cover complexity is more general than being guardable. That is, if a scene is guardable, then it also has small simple-cover complexity, but there exists scenes with small simple-cover complexity that are not guardable.

Theorem 3.4

(i) Let $\mathcal{O}$ be a set of n objects in $\mathbb{R}^d$. If $\mathcal{O}$ has a κ-guarding set of size m against hypercubes, then it has (s, δ)-simple-cover complexity for $\delta = O(\kappa)$ and $s = O(m/(2^d n))$.

(ii) If a 2-dimensional scene $\mathcal{O}$ has (s, δ)-simple-cover complexity, then it has an 8δ-guarding set against squares of size $O(sn)$.

(iii) There are 3-dimensional scenes of $(s, 3)$-simple-cover complexity for a constant s such that any κ-guarding set has size $\Omega(n^2/\kappa^2)$.

Proof:

(i) The proof of this part is identical to the proof due to de Berg et al. [6] that a κ-cluttered scene has $(s, O(\kappa))$-simple-cover complexity, for some constant s, except that we need to replace the collection of bounding-box vertices of the objects by a κ-guarding set $\mathcal{G}$ of size m. As a consequence of the replacement, the size of the simple cover increases by a factor of $m/(2^d n)$.

(ii) Consider a (s, δ)-simple cover $\mathcal{B}$ for $\mathcal{O}$. The $s \cdot n$ discs in $\mathcal{B}$ are δ-simple, that is, each of them intersects at most δ objects in $\mathcal{O}$, and together they cover the bounding box of $\mathcal{O}$. Let $\overline{G}$ be the set of $5|\mathcal{B}|$ points obtained by taking for each disc in $\mathcal{B}$ its center and its topmost, bottommost, leftmost, and rightmost point. We add to $\overline{G}$ all the locally x-extreme and locally y-extreme points on each object in $\mathcal{O}$. (If an object has a vertical or horizontal edge that is locally extreme, we add an arbitrary point on that edge to $\overline{G}$.) Because objects in $\mathcal{O}$ have constant complexity we add $O(n)$ points in this manner. We will show that $\overline{G}$ is a 8δ-guarding set against squares.

Consider a square σ not containing a guard from $\overline{G}$. Let n_σ denote the number of objects from $\mathcal{O}$ intersecting σ. Because of the locally extreme points added to $\overline{G}$ for each object, we know that an object intersecting σ must intersect at least two edges of σ. Hence, there is an edge e intersected by at least $n_\sigma/2$ objects. If the number of objects intersecting e is less than $2\delta + 2$ then $n_\sigma < 4\delta + 4$ and we are done, so assume this is not the case. Assume without loss of generality that e is the left edge of σ. The edge e must be covered by a subset of the discs in $\mathcal{B}$. (Since the discs are open, we must assume here that e does not contain a guard, because such a guard would not be covered. If e does contain guards, we should replace e by a segment e' obtained by moving e slightly to the right.) Observe that any disc $D \in \mathcal{B}$ intersecting σ must contain a vertex of σ, otherwise one of the five guards added for D would lie inside σ.

We define a collection of *obstacle points* on e, as follows. Consider for each object its highest point of intersection with e. Let $p_1, \ldots, p_{\delta+1}$ be the $\delta + 1$ highest intersection points, where $p_{\delta+1}$ is the lowest among them. Of the remaining objects, consider the lowest point of intersection with e, and let $q_1, \ldots, q_{\delta+1}$ be the $\delta + 1$ lowest intersection points, where $q_{\delta+1}$ is the highest among them—see Fig. 5. (We assume for simplicity that the highest intersection points are distinct, and that the lowest intersection points are distinct; it is straightforward to adapt the proof to the general case.) Observe that the intersection of e with any remaining object—any object not defining one of the points p_i or q_i—lies between $p_{\delta+1}$ and $q_{\delta+1}$.

Of all the discs from $\mathcal{B}$ containing the top left vertex of σ, let D_1 be the one containing the largest portion of e. Note that D_1 can cover only the obstacle points $p_1, \ldots, p_\delta$. Similarly, the disc D_2 containing the largest portion of e among all discs containing the bottom left vertex of σ can cover only the obstacle points $q_1, \ldots, q_\delta$. Let D_3 be a disc from $\mathcal{B}$ covering $p_{\delta+1}$ and let D_4 be a disc from $\mathcal{B}$ covering $q_{\delta+1}$. Since D_3 does not contain the top or bottom left vertex of σ, it must contain the top or bottom right vertex. In fact, since the center of D_3 does not lie in σ we know that D_3 must contain the entire right edge of σ. The same holds for D_4. Hence, the segment connecting $p_{\delta+1}$ to the lower right corner of σ is contained in D_3, and the segment connecting $q_{\delta+1}$ to the upper right corner of σ is contained in D_4. But this implies that any object with a point on e between $p_{\delta+1}$ and $q_{\delta+1}$ must intersect either D_3 or D_4 (or both). We can conclude that any object intersecting e intersects one of the four discs $D_1, \ldots, D_4$. Hence, there can be at most 4δ such objects, implying that the total number of objects intersecting σ is at most 8δ.

(iii) We now give an example of a 3-dimensional scene with small simple-cover complexity that is not guardable. Consider n unit circles parallel to the yz-plane, and whose centers lie on the x-axis (see Figure 6a). We fix the distance between any two consecutive circle centers to be $1/n$. Let A denote the cylinder induced by the n circles, i.e., the cylinder of radius $1/2$ whose axis is defined by the centers of the leftmost and rightmost circles. Let

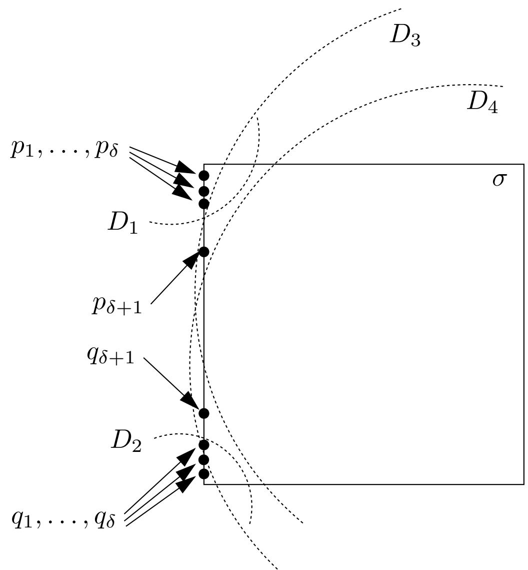

Figure 5: Illustration for the proof of Theorem 3.4(ii).

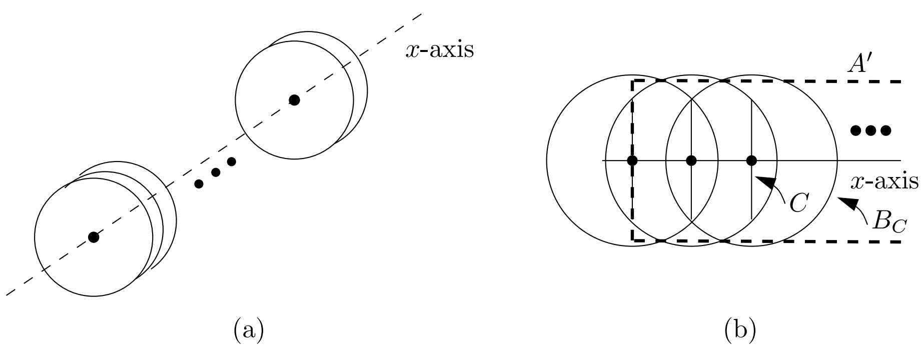

Figure 6: (a) The n circles drawn as discs. (b) The projection on the xz-plane of the circles C, the balls B_C, and the cylinder A'.

κ be a constant. We first show that any κ-guarding set for this scene against hypercubes consists of $\Omega(n^2/\kappa^2)$ guards, and then show that this scene can be covered by a linear number of balls, each of which intersects at most 3 circles.

Consider a batch of $\kappa + 1$ consecutive circles. We can place $\Omega(n/\kappa)$ disjoint cubes, each of side length κ/n, such that each of them intersects all $\kappa + 1$ circles of the batch. Thus, any guarding set for our scene must have a guard in each of these cubes. Since we can form $n/(\kappa + 1)$ such (pairwise disjoint) batches, we immediately obtain the $\Omega(n^2/\kappa^2)$ lower bound.

We now show how to cover the bounding box of the scene with a linear number of balls, each of which intersects at most 3 circles. For each circle C in the scene, we place a ball B_C of radius slightly greater than $1/2$ around C's center. The radius r of B_C is chosen such that the boundary of B_C contains the two (or only one if C is one of the extreme

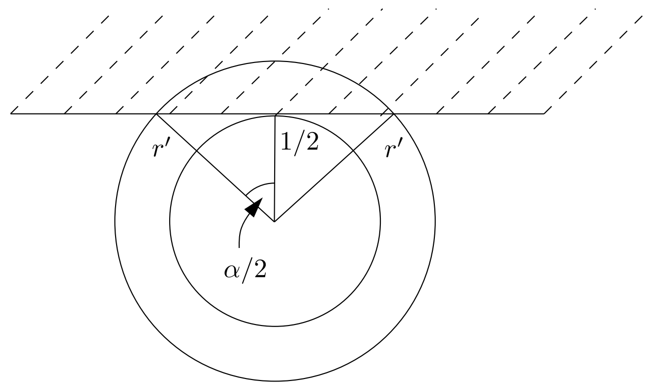

Figure 7: A half plane tangent to the inner disc and not containing it.

circles) adjacent circles of C; it is $\sqrt{1/4+1/n^2}$. Clearly B_C intersects exactly 3 circles. The union of the balls B_C contains a cylinder A', with $A' \supseteq A$, of radius $r' = \sqrt{r^2 - 1/(4n^2)} = \sqrt{1/4 + 3/(4n^2)}$, whose axis is the line segment defined by the center points of the leftmost and rightmost circles (see Figure 6b). We still need to take care of the region of the bounding box of the scene that is not covered by A'. We do this by placing a sufficient number of half spaces (i.e., balls of infinite radius) that are nearly tangent to the cylinder A and do not contain A. The difference between r', the radius of A', and $1/2$, the radius of A, allows us to do this with $O(n)$ half spaces. To see this it suffices to verify that given two concentric discs of radii $1/2$ and r' in the plane, one can cover the entire plane with $O(n)$ half planes that are tangent to the boundary of the inner disc and do not contain it. Consider a line that is tangent to the boundary of the inner disc (see Figure 7). The part of this line that is contained in the outer disc rests on a central angle α such that $\sin(\alpha/2) = \sqrt{3}/\sqrt{n^2+3}$. Thus, assuming n is large, α is approximately equal to $2\sqrt{3}/\sqrt{n^2+3} = \Omega(1/n)$, and therefore only $O(n)$ half planes are needed.

□

Remark: The definition of simple-cover complexity uses δ-simple balls to cover the space. We could also define simple-cover complexity in terms of δ-simple hypercubes. It is possible to show that Theorem 3.4 holds for this definition of simple-cover complexity as well. This implies that, at least in the plane, the definitions of simple-cover complexity using discs and squares, respectively, are equivalent up to constant factors.

4 Computing guarding sets

De Berg [2] presents an algorithm for computing a linear-size binary space partition (BSP) for uncluttered d-dimensional scenes. He also describes a linear-size data structure, based on the underlying BSP tree, supporting logarithmic-time point location queries. It is easy to see that both his BSP algorithm and data structure apply, without any change in the bounds, to guardable scenes. Thus d-dimensional guardable scenes admit a linear-size BSP and a linear-size point location structure. However, in order to apply them, one must have a linear-size κ-guarding set, for some constant κ. For some collections of objects such guarding sets follow immediately from the shape of the objects (for example for discs), but in general it is unclear how to compute such sets.

In this section, we study some heuristic algorithms for computing small κ-guarding sets for guardable scenes, where κ is a small constant. We will describe and test the algorithms for planar sets, but they can easily be generalized to higher-dimensional spaces. Let $\mathcal{O}$ be a set of n objects in the plane, and assume that $\mathcal{O}$ has a finite κ-guarding set against axis-parallel squares. We describe three algorithms for computing a κ-guarding set $\mathcal{G}$, for $\mathcal{O}$. These algorithms might fail for some particularly difficult scenes, where, e.g., a specific point must be present in any κ-guarding set for the scene (see below). We evaluate the algorithms according to their generality, and according to the sizes of the guarding sets that they produce, which is determined experimentally. We are less interested at this point in the efficient implementation of the algorithms.

4.1 Algorithms

Let s_0 be a smallest bounding square of the input scene. All three algorithms construct a quad tree $\mathcal{T}$, through which a guarding set is computed. Each node of $\mathcal{T}$ represents a square that is contained in s_0, where the root represents s_0 itself. The collection of squares associated with the leaves of $\mathcal{T}$ forms a subdivision of s_0 into squares.

Algorithm $\mathcal{A}_1$ constructs the tree in the standard way, except that the stopping criterion is adapted to our purpose. Initially $\mathcal{T}$ consists of a single (root) node representing the bounding square s_0, and $\mathcal{G}$ is empty. Now, for a node v representing a square s_v, we check whether s_v is intersected by more than $\lfloor \kappa/2 \rfloor$ objects of $\mathcal{O}$. If the answer is negative, then we do not expand v; it becomes a leaf of $\mathcal{T}$, and we add the four corners of s_v to the guarding set $\mathcal{G}$. If the answer is positive, then we continue expanding the tree by creating four new nodes corresponding to the four quadrants of s_v and attaching them as the children of v.

Claim 4.1 *The set $\mathcal{G}$, computed by algorithm $\mathcal{A}_1$, is a κ-guarding set for $\mathcal{O}$ against squares.*

Proof: Let c be a square and assume that $c \cap \mathcal{G} = \emptyset$. The square c cannot contain any corner of a square s associated with a leaf of $\mathcal{T}$ (since the corners belong to $\mathcal{G}$). Therefore, c can be covered by at most two squares associated with leaves of $\mathcal{T}$. Since each of these at most two squares is intersected by at most $\lfloor \kappa/2 \rfloor$ objects of $\mathcal{O}$, we conclude that c is intersected by at most κ objects of $\mathcal{O}$. (Note that if c is not fully contained in s_0, then $c \cap s_0$ is covered by a single square associated with a leaf of $\mathcal{T}$, so, in this case, c is intersected by at most $\lfloor \kappa/2 \rfloor$ objects of $\mathcal{O}$.) $\square$

Algorithm $\mathcal{A}_2$ differs from $\mathcal{A}_1$ in (i) the stopping criterion and (ii) the rule by which points are added to $\mathcal{G}$. We stop expanding a node v associated with a square s_v, if s_v is intersected by at most $\lfloor \kappa/6 \rfloor$ objects of $\mathcal{O}$. The guarding points are the corners of the squares associated with the *internal* nodes of $\mathcal{T}$ (rather than the corners of the squares associated with the leaves, as in $\mathcal{A}_1$).

Claim 4.2 *The set $\mathcal{G}$, computed by $\mathcal{A}_2$, is a κ-guarding set for $\mathcal{O}$ against squares.*

Proof: Let c be a square and assume that $c \cap \mathcal{G} = \emptyset$. If s_v is a square associated with a leaf v of $\mathcal{T}$, then it is impossible that s_v is fully contained in (the interior of) c (because

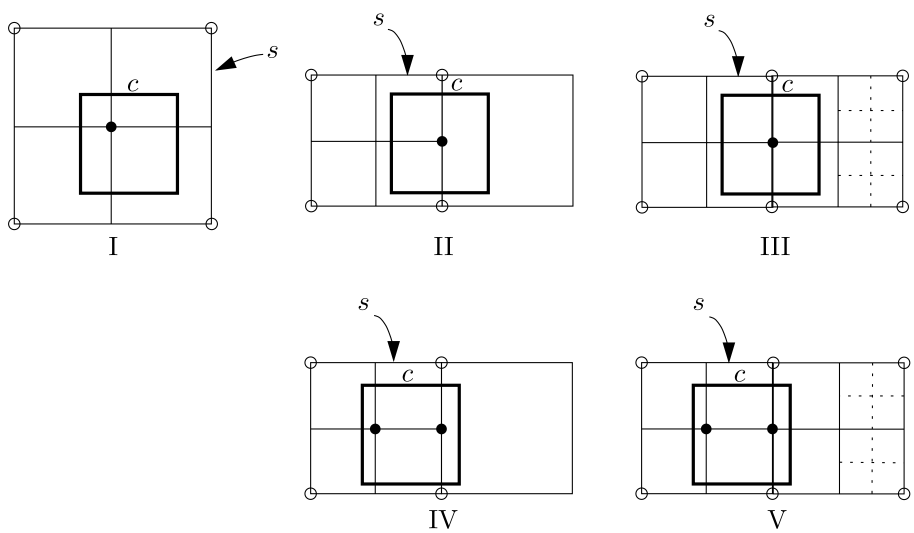

Figure 8: All possible ways in which c can be covered by the subdivision of s_0.

the corners of s_p, the square associated with the parent of v, are in $\mathcal{G}$, and s_v shares a corner with s_p). Moreover, it is easy to verify that the number of vertices of the subdivision of s_0 (formed by the squares associated with the leaves of $\mathcal{T}$) that lie in the interior of c cannot exceed 2, and that the number of squares of the subdivision that intersect c is at most 6. (Figure 8 shows all possible ways in which c can be covered by the squares of the subdivision, where in cases II and IV the large square may be replaced by a larger square containing it.) Each of these at most 6 squares is associated with a leaf of $\mathcal{T}$, and is therefore intersected by at most $\lfloor \kappa/6 \rfloor$ objects of $\mathcal{O}$. Hence c is intersected by at most κ objects of $\mathcal{O}$. □

Algorithm $\mathcal{A}_3$ is completely different; it is based on the notion of *neighborhood*.

Definition 4.3 *Let u be a leaf in the* bottommost *level of a quadtree $\mathcal{T}$. A leaf v of $\mathcal{T}$ is a* neighbor *of u, if either an edge of s_u is contained in an edge of s_v, where s_u (resp., s_v) is the square associated with u (resp., v), or s_v shares a vertex with s_u. The* neighborhood *of u is the set of all its neighbors (see Figure 9).*

Note that v can be a neighbor of u while u is not a neighbor of v. Neighbors of a leaf u must have size at least the size of u. It is easy to verify that the number of neighbors of a leaf u is at most 8.

In the third algorithm, $\mathcal{A}_3$, the quad tree is not constructed in the standard way; it is constructed level by level as follows. Initially, $\mathcal{T} = \mathcal{T}_0$ consists of a single node representing the bounding square s_0, and $\mathcal{G}$ is empty. Let $\mathcal{T}_i$, $i \geq 0$, be the tree that is obtained after constructing levels 0 to i. We describe how to construct level $i+1$. For each node u in level i (i.e., in the bottommost level of $\mathcal{T}_i$), we check whether the union of s_u, the square associated with u, and the squares associated with the nodes in the neighborhood of u (referring to $\mathcal{T}_i$) is intersected by more than κ objects of $\mathcal{O}$. If so, we add the four corners of s_u to $\mathcal{G}$, and expand u by creating four new nodes corresponding to the four quadrants of s_u and attaching them as the children of u. After applying this check to all nodes in level i (in arbitrary order),

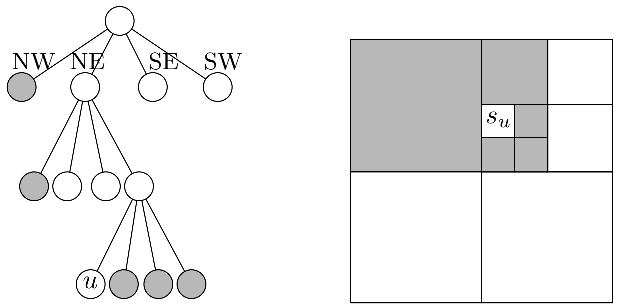

Figure 9: The neighborhood of the leaf u associated with square s_u is shown in grey.

we obtain the tree $\mathcal{T}_{i+1}$. We stop when a level is reached for which none of the nodes needs to be expanded.

Claim 4.4 *The set $\mathcal{G}$, computed by $\mathcal{A}_3$, is a κ-guarding set for $\mathcal{O}$ against squares.*

Proof: Let c be a square and assume that $c \cap \mathcal{G} = \emptyset$. We know (see proof of Claim 4.2) that, since the guarding points are the corners of the squares associated with the internal nodes of $\mathcal{T}$, c can be covered by at most six squares associated with leaves of $\mathcal{T}$. Refer to Figure 8. Let w be the leaf corresponding to the square s, and assume w is in level i. Then, in $\mathcal{T}_i$, the squares associated with the leaves in the neighborhood of w include the at most 5 other squares that intersect c. Since w was not expanded when constructing level $i+1$, the union of the squares that intersect c is intersected by at most κ objects of $\mathcal{O}$, and therefore c is intersected by at most κ objects. □

Notice that scenes in which many object share the same boundary point might be problematic. If more than κ objects share a point, this point must be chosen as a guard. Otherwise we could place a small enough square around such a point that is both empty and is intersected by more than κ objects. Unfortunately, the algorithms described above normally fail to find this point. We call such scenes *degenerate scenes*.

Algorithm $\mathcal{A}_3$ can more easily deal with degenerate scenes than the other algorithms. This becomes more evident when dealing with 3-dimensional scenes. The stopping criterion of $\mathcal{A}_1$ and $\mathcal{A}_2$ deteriorates, when moving to 3-space. For example, in the 3-dimensional version of $\mathcal{A}_1$, we stop expanding a node if the cube that is associated with it is intersected by at most $\lfloor \kappa/4 \rfloor$ (rather than $\lfloor \kappa/2 \rfloor$) objects of $\mathcal{O}$. This implies that if the underlying scene contains a special point that is common to more than $\lfloor \kappa/4 \rfloor$ objects, the algorithm will surely fail to produce a κ-guarding set. On the other hand, the stopping criterion of $\mathcal{A}_3$ does not change.

It is not possible to claim that one of the three algorithms is always better than the others. In [3] we show that for each of the algorithms there exists a scene for which it is better than the other two, in the sense that it produces a smaller guarding set. In the next subsection we evaluate the algorithms experimentally. Here the advantage of algorithm $\mathcal{A}_3$ will become evident.

4.2 Experimental evaluation

We have implemented the three algorithms described in the preceding section, using the CGAL software library of geometric data structures and algorithms (see `http://www.cgal.org/`), and have performed various experiments in order to learn about their suitability in practice.

Our goal has been to evaluate the algorithms, according to the sizes of the guarding sets that they produce. Recall that the size of the guarding set is closely related to the size of the BSP and data structures that are subsequently constructed for the input scene, assuming de Berg's algorithms [2] are being used. We have applied our algorithms to uncluttered scenes with clutter factor κ, and checked (i) whether the κ-guarding sets that are obtained tend to be smaller than $4n$ (recall that, by definition, the set consisting of all $4n$ bounding-box vertices is a κ-guarding set for such scenes), and (ii) assuming the answer is positive, what is the smallest value, $\kappa_0 \leqslant \kappa$, for which the scene still has a κ_0-guarding set.

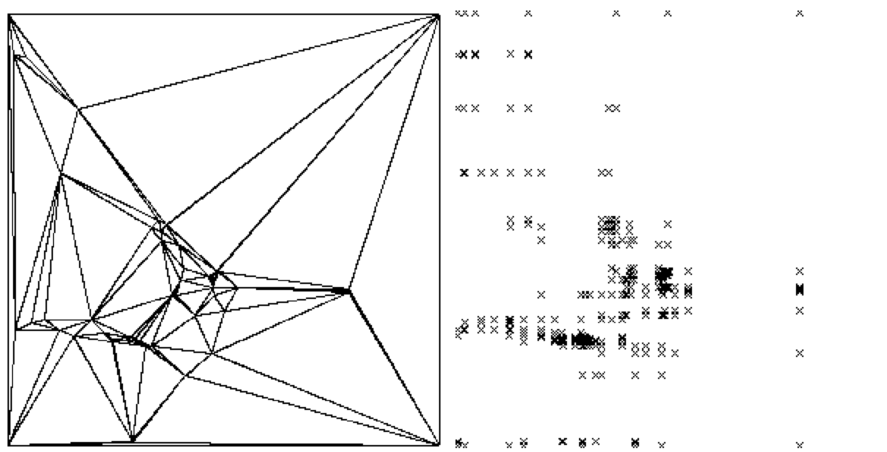

Figure 10: Left: Death Valley (first scene). Right: The 25-guarding set consisting of all 808 bounding-box vertices.

As test scenes we used projected polyhedral terrains. Polyhedral terrains are often used to represent pieces of the earth's surface in Geographic Information Systems. Most of the polyhedral terrain algorithms work with the terrain's projection on the xy-plane. Thus our terrain test scenes were generated from DEM files of certain areas in Canada and the U.S. as follows. A DEM file specifies the elevation of a set of sample points in the underlying area, where the sample points form a regular grid. Using the so-called VIP method [7] the m most important points were extracted for various values of m. The terrain test scene was then generated by computing the Delaunay triangulation of the extracted sample points.

The results of some of our tests are presented in the table below. The left table corresponds to two "Death Valley" scenes, consisting of 202 and 402 triangles, respectively. The clutter factors of these scenes are 25 and 29, respectively. The right table corresponds to two "San Bernardino" scenes, again consisting of 202 and 402 triangles, respectively. The clutter factors of these scenes are 16 and 18, respectively. An 'x' entry in the column of algorithm $\mathcal{A}_i$ means that for the appropriate value of κ, $\mathcal{A}_i$ failed to produce a guarding set, before some halting condition was fulfilled (indicating that $\mathcal{A}_i$ would probably never terminate without the halting condition).

Death Valley						San Bernardino					
n	clutter factor	κ	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$	n	clutter factor	κ	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$
202	25	72	62	147	56	202	16	60	49	119	40
		25	307	x	232			20	278	x	172
		22	x	x	291			16	x	x	271
		18	x	x	391			15	x	x	271
		16	x	x	433			13	x	x	364
		12	x	x	685			11	x	x	529
		11	x	x	x			10	x	x	677
402	29	66	100	284	105	402	18	66	75	198	74
		29	427	x	307			22	462	x	286
		22	610	x	439			18	x	x	399
		20	x	x	485			14	x	x	636

For each of the test scenes, we applied the three algorithms for various values of κ, beginning with rather high values and ending around the smallest value κ_0 for which one of the algorithms still succeeds in producing a guarding set. As expected (see discussion at the end of the previous subsection), the first two algorithms begin to fail once the values are less than $2\kappa_0$ and $6\kappa_0$, respectively.

Consider for example the first Death Valley scene (see Figure 10(left)). Since the number of objects in this scene is 202 and the clutter factor is 25, we can obtain a 25-guarding set of size at most $4n = 808$, by taking all bounding-box vertices (see Figure 10(right)). However, both $\mathcal{A}_1$ and $\mathcal{A}_3$ produce much smaller 25-guarding sets. For this scene, the value κ_0 is about 12, and algorithm $\mathcal{A}_3$ produces a 12-guarding set of size 685, which is still less than $4n$. Since κ_0 is about 12, it is not surprising that algorithm $\mathcal{A}_1$ fails for values below 24, and that algorithm $\mathcal{A}_2$ fails for even higher values. Figure 11 show the guarding set computed by algorithm $\mathcal{A}_3$ for κ-values 25 and 16.

In general, the sizes of the guarding sets produced by $\mathcal{A}_2$ are much larger than the sizes of those produced by $\mathcal{A}_1$ and $\mathcal{A}_3$, and the sets produced by $\mathcal{A}_3$ are usually smaller than those produced by $\mathcal{A}_1$. In conclusion, algorithm $\mathcal{A}_3$ seems to perform rather well in practice. This result was verified with a number of other test scenes, including scenes of randomly generated triangles.

5 Concluding remarks

After introducing the notion of a guarding set for a set of objects $\mathcal{O}$, we proceeded in three main directions. We first proved some basic properties concerning guarding sets, and discussed the connection between guarding sets and ε-nets. Next, we studied the relation between guardable scenes and two realistic input models proposed in the literature, namely, unclutteredness and small simple-cover complexity. Although every uncluttered scene is guardable, the opposite statement was found false; a scene that is guardable but not uncluttered was constructed. Nevertheless, de Berg's linear-size BSP construction for d-dimensional uncluttered scenes applies also to d-dimensional guardable scenes (assuming the guarding set has already been computed). In the last part of the paper, we proposed three heuristic algorithms

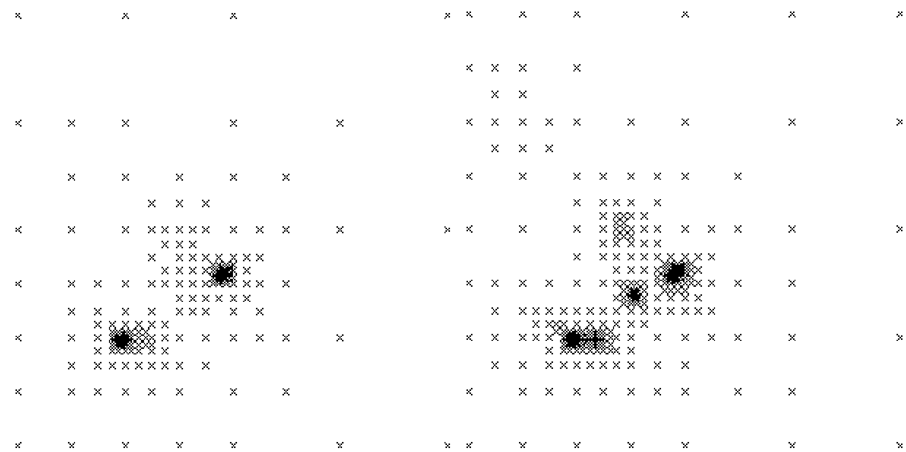

Figure 11: Left: A 25-guarding set of 232 points computed by $\mathcal{A}_3$. Right: A 16-guarding set of 433 points computed by $\mathcal{A}_3$.

for computing a small κ-guarding set for a set of objects, and evaluated them experimentally. In particular the third algorithm produces small guarding sets for small values of κ.

There are a number of interesting directions for further research. First of all, the notion of a guarding set seems to be interesting to study further. Secondly, it would be interesting to see what other geometric problems can be solved efficiently for guardable scenes. Finally, there is the problem of computing small guarding sets. It is likely that the problem of computing a minimal size guarding set is NP-complete (although no proof is known yet), but efficient approximation algorithms might exist.

References

[1] D. Avis, P. Bose, T.C. Shermer, J. Snoeyink, G. Toussaint, and B. Zhu. On the sectional area of convex polytopes. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages C-11–C-12, 1996.

[2] M. de Berg. Linear size binary space partitions for uncluttered scenes. To appear in *Algorithmica*.

[3] M. de Berg, H. David, M. Katz, M. H. Overmars, A. F. van der Stappen, and J. Vleugels. Guarding sets against invasive hypercubes. In *Proc. 2nd Workshop on Algorithm Engineering*, pages 110–120, 1998.

[4] M. de Berg, M. J. Katz, M. H. Overmars, A. F. van der Stappen, and J. Vleugels. Models and motion planning. In *Proc. 6th Scandinavian Workshop on Algorithm Theory, LNCS 1432*, pages 83–94, 1998.

[5] M. de Berg, M. J. Katz, M. H. Overmars, A. F. van der Stappen, and J. Vleugels. *Models and motion planning.* Technical Report UU-CS-2000-41, Utrecht University, 2000.

[6] M. de Berg, M. J. Katz, A. F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 294–303, 1997.

[7] Z. Chen and J. A. Guevara. System selection of very important points (VIP) from digital terrain model for constructing triangular irregular networks. In *Proc. 8th Internat. Sympos. Comput.-Assist. Cartog. (Auto-Carto)*, pages 50–56, 1988.

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

[9] J. Matoušek. Epsilon-nets and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 69–89. Springer-Verlag, 1993.

[10] J. S. B. Mitchell, D. M. Mount, and S. Suri. Query-sensitive ray shooting. *Internat. J. Comput. Geom. Appl.*, 7(4):317–347, August 1997.

[11] A. F. van der Stappen. *Motion Planning amidst Fat Obstacles.* Ph.D. dissertation, Dept. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, 1994.