

Balanced Aspect Ratio Trees: Combining the Advantages of k -d Trees and Octrees¹

Christian A. Duncan²

Department of Computer Science, University of Miami, Coral Gables, Florida

E-mail: duncan@cs.miami.edu

Michael T. Goodrich³

Center for Geometric Computing, The Johns Hopkins University, Baltimore, Maryland

E-mail: goodrich@jhu.edu

and

Stephen Kobourov²

Department of Computer Science, University of Arizona, Tucson, Arizona

E-mail: kobourov@cs.arizona.edu

Received June 28, 1999

Given a set S of n points on \mathbb{R}^d , we show, for fixed d , how to construct in $O(n \log n)$ time a data structure we call the balanced aspect ratio (BAR) tree. A BAR tree is a binary space partition tree on S that has $O(\log n)$ depth in which every region is convex and “fat” (that is, has a bounded aspect ratio). While previous hierarchical data structures such as k -d trees, quadtrees, octrees, fair-split trees, and balanced box decompositions can guarantee some of these properties, we know of no previous data structure that combines all of these properties simultaneously. The BAR tree data structure has numerous applications ranging from geometric searching problems in fixed dimensional space to the visualization of graphs and three-dimensional worlds. © 2001 Academic Press

¹ This paper is a version of [20] which extends the original work in [19].

² This research partially supported by ARO under grant DAAH04-96-1-0013 while the author was at the Johns Hopkins University.

³ This research partially supported by NSF under Grant CCR-9625289 and ARO under Grant DAAH04-96-1-0013.

1. INTRODUCTION

Geometric searching of multidimensional spaces is a fundamental operation in many fields, including astronomy, geographic information systems (GIS), computer graphics, information retrieval, pattern recognition, natural language processing, and statistics. Typical searches include nearest-neighbor searches, farthest-neighbor searches, and range queries (which are intersection queries for geometric shapes). In this paper we study efficient data structures for performing such queries in moderate-dimensional spaces, that is, in spaces where the dimensionality, d , of the space can be viewed as a constant compared to the number, n , of multidimensional points in that space.⁴

1.1. *Previous Related Work*

Data structures for performing multidimensional geometric searching in moderate-dimensional spaces have been well studied in computational geometry and spatial databases [5, 7, 10, 11, 14, 18, 23, 28, 31]. There are, in fact, hosts of available data structures specializing in many kinds of geometric queries. For excellent surveys of many of these data structures see the papers by Agarwal and Erickson [1] and Matoušek [29]. Many previous data structures for performing geometric searching for a multidimensional point set S are instances of a general class of structures known as binary space partition (BSP) trees [24] (see also [38]). Each node in a BSP tree T represents both a convex region in space and all of the objects of S , in our case points, lying inside this region. Each leaf node in T represents a region with a constant number of objects, points, of S inside it. Every other node in T has an associated hyperplane cut partitioning the region into two subregions, each corresponding to a child node. The root of T is associated with a bounding hyperbox containing S . One of the main advantages of BSP trees is that they allow for simple multidimensional searching, with a typical comparison for a node v in T simply involving a sidedness test against the hyperplane cut associated with v . BSP trees are often used to solve problems in computer graphics [16, 34, 35, 39], such as global illumination, shadow generation, ray casting, and visibility. In these cases, the set S often contains multidimensional objects. The objects stored may be points, lines, planes, or solids. They are also used in information retrieval for finding nearest neighbors and farthest neighbors, as well as performing range queries.

The performance bounds of a BSP tree T for answering such queries for a point set S are directly related to the depth of T and the “fatness” [2, 3,

⁴ We will view d as a constant relative to n throughout this paper.

17, 21, 30, 32] (that is, the boundedness of the aspect ratios) of the regions associated with T 's nodes. One class of BSP trees, k -d trees [8, 9, 25, 33], uses axis-orthogonal cutting hyperplanes that are placed to divide the set of points associated with a node roughly in half.⁵ Such trees have excellent depth properties since their depth is $O(\log n)$. Unfortunately, since the points in S can be arbitrarily distributed, the regions associated with points in a k -d tree can have arbitrarily large aspect ratios. This unbounded aspect ratio property of k -d trees partly accounts for why there are few simple theoretical results for geometric queries better than the $O(n^{1-1/d})$ worst-case running time for orthogonal range queries, even for approximate versions.

Another well-studied class of search structures include quadtrees [22, 36, 38] and octrees [4, 15, 26, 37] and is based on the alternate approach of using axis-orthogonal hyperplanes to divide region volumes equally. These structures produce space partitioning trees with regions having good aspect ratios, but their depths can be quite large, which again results in poor worst-case search times.

These poor worst-case performances of k -d trees and octrees have motivated some researchers to work beyond the BSP tree framework in search of alternate structures with good depth and aspect-ratio bounds. In particular, the balanced box-decomposition (BBD) tree structure of Arya *et al.* [5, 6], which is based on the fair-split tree of Callahan and Kosaraju [12, 13], provides a space partitioning tree that has $O(\log n)$ depth while also achieving bounded aspect ratio for regions associated with nodes. Arya *et al.* show that this structure can be used, for example, to perform approximate nearest-neighbor searching and range searching in $O(\log n + k)$ time, where k is the size of the output. One difference in this approach is that it partitions space using nonhyperplanar cuts with "holes," which sacrifices the convexity property for the regions associated with nodes. This makes the BBD tree inappropriate for several applications in computer graphics and graph drawing, where convexity of the partitioned regions is desirable (e.g., see [19]).

1.2. Our Results

In this paper, we introduce a multidimensional space partition tree, which we call the *balanced aspect ratio (BAR) tree*, that is defined for any set S of n points in \mathbb{R}^d . We build on previous work on the two-dimensional version [19] and expand on the original conference version of this paper [20]. The BAR tree data structure is conceptually quite simple, for it follows the traditional format of the BSP tree. Moreover, BAR trees

⁵ Throughout this paper, we make the assumption that the points are in general position.

simultaneously achieve the desired properties of having $O(\log n)$ depth and ranges, associated with nodes, that are convex and “fat.”

In the following sections, we give a general framework for BAR trees, some geometric searching applications that are supported by BAR trees, and finally an efficient method for constructing a BAR tree. For a point set S , our construction method uses hyperplane cuts that are either orthogonal to a coordinate axis or form 45° angles with the coordinate axes, the latter of which we call *corner cuts*. This method can be viewed as an extension of the traditional k -d tree, since our BAR tree is identical to the k -d tree defined on S as long as the k -d tree maintains a balanced aspect ratio. In fact, corner cuts are used only when an axis-parallel cut would produce a region that is too “skinny.”

2. THE GENERAL BAR TREE FRAMEWORK

In this section, we develop a general framework for constructing BAR trees. We begin by defining what we mean by bounded aspect ratio.

DEFINITION 2.1. A convex region R in \mathbb{R}^d has aspect ratio $\text{asp}(R) = O_R/I_R$ with respect to some underlying metric, where O_R is the radius of the smallest circumscribed hypersphere in \mathbb{R}^d and I_R is the radius of the largest inscribed d -hypersphere. R has *balanced aspect ratio* with maximum aspect ratio $\alpha \geq 1$, if $\text{asp}(R) \leq \alpha$. We call such a region an α -*balanced region*. A collection of regions, \mathcal{R} , has balanced aspect ratio with balancing factor α if each region $R \in \mathcal{R}$ is an α -balanced region.

Typically, we use one of the standard L_p metrics to define aspect ratios, as the aspect ratio defined for one L_p metric is within a polynomial factor of d from the aspect ratio for another L_p metric. In keeping with the current convention [3, 21, 30], we use the terms *fat* and *skinny* to refer to regions which have respectively balanced and unbalanced aspect ratios. This definition of aspect ratio implies that the ratio between the width and the diameter of a *fat* region is bounded from below by a function depending only on α and d . As in the BBD tree [6], the fact that all regions in the BAR tree are α -balanced helps prove worst-case bounds on some geometric approximation problems.

DEFINITION 2.2. A *canonical (cut) set*, $\mathcal{E} = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_\gamma\}$, is a collection of γ , not necessarily independent, vectors that span \mathbb{R}^d (thus, $\gamma \geq d$). A *canonical cut* is any hyperplane, H , in \mathbb{R}^d with a normal in \mathcal{E} . A *canonical region* is a convex polyhedron in \mathbb{R}^d with every facet having a normal in \mathcal{E} .

DEFINITION 2.3. Any two canonical cuts H and H' that are normal to the same vector in \mathcal{E} , i.e., parallel to each other, are called *opposing canonical cuts*. For any bounded region R , define the *canonical bounding cuts with respect to a direction* $\vec{v}_i \in \mathcal{E}$ to be the two unique opposing canonical cuts normal to \vec{v}_i and tangent to R . Intuitively, R is sandwiched between the two opposing cuts. For simplicity, when referring to the canonical cuts of a region R , we *always* mean the canonical bounding cuts.

The canonical set used to define a partition tree can vary from method to method. For example, the standard k -d tree algorithm uses a canonical set composed of all axis-parallel directions. For notation, we often refer to a canonical cut by its normal vector, $\vec{v}_i \in \mathcal{E}$. For instance, in the k -d tree model we would represent a cut orthogonal to the y -axis by $(0, 1, 0, \dots, 0)$. Let $|R|$ represent the number of points from a given data set S contained in the region R , i.e., its size in terms of points rather than volume.

DEFINITION 2.4. An α -balanced canonical region, R , is *one-cuttable* with reduction factor β , where $1/2 \leq \beta < 1$, if there is a cut $s_1 \in \mathcal{E}$, called a *one-cut*, dividing R into two subregions R_1 and R_2 such that

1. R_1 and R_2 are α -balanced canonical regions.
2. $|R_1| \leq \beta|R|$ and $|R_2| \leq \beta|R|$.

DEFINITION 2.5. An α -balanced canonical region, R is *k -cuttable* with reduction factor β , for $k > 1$, if there is a cut $s_k \in \mathcal{E}$, called a *k -cut*, dividing R into two subregions R_1 and R_2 such that

1. R_1 and R_2 are α -balanced canonical regions,
2. $|R_2| \leq \beta|R|$,
3. Either $|R_1| \leq \beta|R|$ or R_1 is $(k - 1)$ -cuttable with reduction factor β .

In other words, the sequence of cuts, s_k, s_{k-1}, \dots, s_1 , results in $k + 1$ balanced canonical regions each containing no more than βn points. If β is understood, we simply say R is k -cuttable.

We can now describe the basic steps to construct the BAR tree; see Algorithm (1) (Fig. 1).

THEOREM 2.1. *Suppose we are given a canonical set \mathcal{E} and values α and β . If every possible α -balanced canonical region is k -cuttable, then, for any set S , Algorithm 1 constructs a BAR tree with maximum aspect ratio α , depth $O(k \log_{1/\beta} n)$, and size $O(kn)$.*

Proof. We start with an initial bounding α -balanced canonical region on S . If the set size is larger than a constant value, the construction

```

CREATE-BAR-TREE( $u, S_u, R_u, \alpha, \beta$ )
  //  $u$  is the current node (initially the root node)
  //  $S_u$  is the current point set (initially  $S$ )
  //  $R_u$  is the  $\alpha$ -balanced region containing  $S_u$ 
  // (initially a bounding hypercube of  $S$ )
  (Optional) node  $u$  can contain any of the following:
    region  $R_u$ 
    a sample point  $p \in S_u$ 
    the size of the set,  $|S_u|$ 
  if  $|S_u| \leq \text{leafSize}$  then
    (leaf) node  $u$  stores the set  $S_u$ 
    return
  find  $s_i$ , an  $i$ -cut for  $R_u$ , for smallest value of  $i$ 
  (internal) node  $u$  stores  $s_i$ 
  create two child nodes of  $u$ ,  $v$  and  $w$ 
  partition  $S_u$  into  $S_v$  and  $S_w$  by the cut  $s_i$ 
  partition  $R_u$  into  $R_v$  and  $R_w$  by the cut  $s_i$ 
  call CREATE-BAR-TREE( $v, S_v, R_v, \alpha, \beta$ )
  call CREATE-BAR-TREE( $w, S_w, R_w, \alpha, \beta$ )

```

FIG. 1. Algorithm describing construction of the BAR tree.

algorithm finds an i -cut for the smallest value of i . The method to find the appropriate i -cut is described in Section 5. Since the regions are all k -cuttable, after at most k iterations we perform a one-cut. This process divides any region R_u into a sequence of at most $k + 1$ α -balanced subregions each containing fewer than $\beta|R_u|$ of the points. The process, down any path of subregions, can be repeated for no more than $O(\log_{1/\beta} n)$ times, resulting in the stated tree depth bound.

Each node may optionally store several values. The choice is based on performance to storage concerns. Since the canonical regions are constructed from a constant, γ , set of cutting directions, there are at most 2γ facets forming any canonical region R_u . Therefore, even if all values are stored, the total space used by the constructed BAR tree is $O(kn)$. ■

The main challenge in creating a specific instance of a BAR tree is in defining a canonical set \mathcal{C} such that every possible α -balanced canonical region is k -cuttable with reduction factor β for reasonable choices of α , β , and k . But, before we do this, let us motivate our exploration of BAR trees by describing a few of their applications.

3. BAR TREE APPLICATIONS

Suppose we are given a point set S of n points in \mathbb{R}^d . After constructing a BAR tree T on S , we are able to perform some useful geometric queries. For any query point $q \in \mathbb{R}^d$, we are able to report efficiently both the approximate-nearest and approximate-farthest neighbors of q in S . If we are also given a radius r , we are able to return efficiently all points within a distance r from q plus possibly any points that are approximately near r , which is a form of approximate range searching. Interestingly, each of these queries works under any of the Minkowski L_p metrics using the same tree T . We more formally describe some of these applications shortly.

Arya *et al.* [5, 6] propose a technique to solve the approximate nearest-neighbor and approximate range query problems by constructing a balanced box-decomposition tree. Similar to our BAR tree, these trees maintain an α -balanced aspect ratio, but only by introducing nonconvex hole cuts. Their arguments and techniques for solving these query problems, however, are easily transferable to our data structure.

DEFINITION 3.1. For a set S of points in \mathbb{R}^d , a query point $q \in \mathbb{R}^d$, and $\epsilon > 0$, a point $p \in S$ is a $(1 + \epsilon)$ -nearest neighbor of q if $\delta(p, q) \leq (1 + \epsilon)\delta(p^*, q)$, where p^* is the true nearest neighbor to q .

In other words, such a p is within a constant error factor of the true nearest neighbor. This definition can also be extended for a sequence of $(1 + \epsilon)$ -nearest neighbors. Rather than adapt all theorems presented by Arya *et al.*, we instead prove another useful query operation, applicable to both their methods and ours, and establish an important packing feature for BAR trees.

DEFINITION 3.2. For a set S of points in \mathbb{R}^d , a query point $q \in \mathbb{R}^d$, and $\epsilon > 0$, a point $p \in S$ is a $(1 - \epsilon)$ -farthest neighbor of q if $\delta(p, q) \geq \delta(p^*, q) - \epsilon D$, where p^* is the true farthest neighbor and D is the diameter of the point set.

Notice that we are using an absolute error bound rather than the standard relative error, $\delta(p, q) \geq (1 - \epsilon)\delta(p^*, q)$, because the absolute bound is tighter in every case. Imagine a point set that is tightly contained in the unit sphere and a query point that is extremely far from this sphere, say $100/\epsilon$ units. Now, any point returned would be a $(1 - \epsilon)$ -farthest neighbor of S using the standard relative error bound. In our definition, a query point is the better of the absolute and relative distances, within a constant factor of ϵ . Since we are searching for the farthest neighbor, any query point must be at least half the diameter, D , of the point set away

from one of the points in the set, $\delta(p^*, q) \geq D/2$. Letting $\epsilon' = 2\epsilon$, we see that

$$\begin{aligned} \delta(p, q) &\geq \delta(p^*, q) - \epsilon D \\ &\geq \delta(p^*, q) - \epsilon 2\delta(p^*, q) \\ &= (1 - 2\epsilon)\delta(p^*, q) \\ &= (1 - \epsilon')\delta(p^*, q). \end{aligned}$$

Hence, using an absolute error bound ϵ in the approximate farthest-neighbor query always gives a point whose distance is at least as far as the distance allowed using a relative error bound of 2ϵ .

In fact, one can extend this notion and our arguments to compensate for this problem in nearest-neighbor queries as well, i.e., when the query point is relatively far away from the entire data set. We strengthen Definition 3.1 to state that a point $p \in S$ is a $(1 + \epsilon)$ -nearest neighbor of q if $\delta(p, q) \leq \min\{(1 + \epsilon)\delta(p^*, q), \delta(p^*, q) + \epsilon D\}$. Recall our previous example with a query point q that is $100/\epsilon$ units from S . In the original definition, any point returned would be a $(1 + \epsilon)$ -nearest neighbor. Using the stricter definition, the approximate nearest neighbor must also lie within ϵ times the diameter of the point set away from p^* .

We now discuss the farthest neighbor approximation algorithm using a BAR tree. Given our query point q , we begin by finding a leaf node that is the farthest away from q . Here, a region's maximum distance from a point is considered, implying that, in theory, the node containing the point q might still also be the node that is the farthest from q . We next enumerate, via a priority queue, all leaf nodes in decreasing order of distance, i.e., farthest leaf nodes first. For every leaf node, we compute the distances between q and that node's data points and maintain the current farthest visited point p . When the distance between q and the current farthest node is less than $\delta(p, q) + \epsilon D$, we can terminate the search, as $\delta(p^*, q) < \delta(p, q) + \epsilon D$.

A priority queue can be maintained in such a way that the running time is $O(\log n)$ times the number of leaf nodes visited. The key to the algorithm's success is that the number of leaf nodes visited can be limited by using a packing argument. Let us, therefore, describe both the priority queue technique and the packing argument needed to limit the number of leaf nodes visited.

3.1. Farthest-Neighbor Search Algorithm

First we focus on the searching technique; see Algorithm 2 (Fig. 2). To avoid confusion between points in space and in the data set, we call a point


```

APPROXIMATE-FARTHEST-NEIGHBOR( $q, \epsilon$ )
   $Q \leftarrow \text{root}(T)$ 
   $p \leftarrow q$ 
  do  $u \leftarrow Q.\text{extractMax}()$ 
    if  $\Delta(u, q) \leq \delta(p, q) + \epsilon D$ 
      return  $p$ 
    while  $u$  is not a leaf
       $u_1 \leftarrow \text{leftChild}(u)$ 
       $u_2 \leftarrow \text{rightChild}(u)$ 
      for all  $p'$  in  $\text{dataSet}(u)$ 
        if  $\delta(p', q) > \delta(p, q)$ 
           $p \leftarrow p'$ 
      if  $\Delta(u_1, q) \geq \Delta(u_2, q)$ 
         $Q.\text{insert}(\Delta(u_2, q), u_2)$ 
         $u \leftarrow u_1$ 
      else
         $Q.\text{insert}(\Delta(u_1, q), u_1)$ 
         $u \leftarrow u_2$ 
    end while
  //  $u$  is now a leaf
  for all  $p'$  in  $\text{dataSet}(u)$ 
    if  $\delta(p', q) > \delta(p, q)$ 
       $p \leftarrow p'$ 
  repeat

```

FIG. 2. The basic algorithm to perform farthest-neighbor priority searching.

p a data point if $p \in S$ and a point r a real point if $r \in \mathbb{R}^d$. For any node u and its associated region R , let $\Delta(u, q) = \Delta(R, q) = \max_{r \in R} \delta(r, q)$; i.e., the distance between q and a node u is the distance between q and the farthest real point from q in the region R .

Initially, a priority queue Q starts with the root node of T . Let p be the current farthest neighbor, initially set to q . At every stage, extract from Q the node, u , that is the farthest away from q . If $\Delta(u, q) \leq \delta(p, q) + \epsilon D$, we return p as the $(1 + \epsilon)$ -approximate farthest neighbor. If u is a leaf node, let $p' \in S$ be the node's associated data point, if any. If $\delta(p', q) > \delta(p, q)$, let $p \leftarrow p'$. Remove u from consideration, and continue with the next node in Q . If u is not a leaf, let u_1 and u_2 be u 's children. Since $u = u_1 \cup u_2$, one of the two nodes must contain the real point that was farthest away from q . Without loss of generality, let this be u_1 and insert u_2 into the queue. Use u_1 as the next "extracted" node from Q and continue.

LEMMA 3.1. *Given a BAR tree T with $O(\log n)$ depth, $O(n)$ nodes, and a constant number of canonical vectors, for any query point q , a $(1 - \epsilon)$ -farthest neighbor to q can be found in $O(l \log n)$ time, where l is the number of leaf nodes visited in our algorithm.*

Proof. First, we can see the correctness of our algorithm by looking at the leaf node u^* containing p^* . If u^* has been visited, our algorithm would set $p \leftarrow p^*$ and return the correct solution upon termination. If u^* has not yet been visited, implying $p \neq p^*$, let u be the last node visited before termination. Thus, it follows that:

$$\begin{aligned} \delta(p^*, q) &\leq \Delta(u^*, q) \\ &\leq \Delta(u, q) \\ &\leq \delta(p, q) + \epsilon D. \end{aligned}$$

Since our algorithm only performs extract operations after reaching a leaf node, which has depth $O(\log n)$, we execute $O(\log n)$ queue inserts per leaf node visited and one extract per leaf node. If we use a Fibonacci heap, insertions take $O(1)$ amortized time and extractions take $O(\log n)$ time, since the queue has size $O(n)$. Since there are a constant number of cut directions, every canonical region has constant complexity. Deciding which node is farther takes constant time. Thus, if l is the number of leaf nodes visited, the algorithm terminates in $O(l \log n)$ steps. ■

3.2. Packing Constraint

As in the nearest-neighbor algorithm of Arya *et al.*, if we can limit the number of leaf nodes that we need to visit, we can bound the running time. This is where a packing constraint comes in.

DEFINITION 3.3. For any annulus A with center p and radii r_1 and r_2 , a region R pierces A if and only if there exists two real points $q_1, q_2 \in R$ such that $\delta(p, q_1) \leq r_1 \leq r_2 \leq \delta(p, q_2)$.

In other words, the region R lies on both sides of the two concentric hyperspheres defining A .

LEMMA 3.2 (Packing lemma). *Given a BAR tree with maximum aspect ratio α for a set S of data points in \mathbb{R}^d and two size parameters $r, r' > 0$, using any Minkowski metric, L_p , there are $O((\alpha\sqrt{d})^d (r/r')^{d-1})$ leaf nodes which pierce any annulus with radii $r + r'$ and r .*

Proof. Let l be a leaf node in the tree with associated region R that pierces the annulus, A . This means that the outer radius $O_R \geq r'/2$. Let B be any L_1 hyperspace of radius r' whose center lies inside R . Since R is convex and fat, from [32] we can see that $V_{B \cap R} \geq V_B / \alpha^d$. Since R pierces

A , we know that there exists a point $p \in R$ such that the hypersphere B centered at p with radius r' is completely contained in A . Let $R' = R \cap A$. Since $B \cap A = B$, $B \cap R = B \cap A \cap R \subset R'$. Therefore, $V_{R'} \geq V_{R \cap B} \geq V_B / \alpha^d \geq (r' / (\alpha\sqrt{d}))^d$.

Let us calculate the volume of any annulus, A , with radii $r + r'$ and r . Since we are dealing with many possible metrics, let us concentrate on the L_∞ metric which produces the largest volume annulus. The volume in this metric is equal to the difference between the volume of the outer and inner boxes of length $2(r + r')$ and $2r$, respectively. Thus, in the L_∞ metric, $V_A = (2r + 2r')^d - (2r)^d = (2r)^d((1 + r'/r)^d - 1)$. In any of the other L_p metrics, the volume of the annulus is less than this amount.

Since the regions associated with the leaf nodes do not overlap, the number of leaf nodes, L , piercing A is smaller than the ratio of the two volumes, i.e., $L \leq V_A / V_{R'} \leq (2\alpha\sqrt{d})^d((r/r' + 1)^d - (r/r')^d) = O((\alpha\sqrt{d})^d(r/r')^{d-1})$. ■

Our only other concern, then, is that some leaf nodes might not contain any points, as we never made this stipulation in our general framework. Ignoring these empty nodes altogether averts this problem entirely. At any nonleaf node, if one of the two children is empty, the algorithm simply proceeds to the other child, without even inserting it onto the queue, as if the split never happened. Since the regions are all k -cuttable, there are at most $k - 1$ regions in sequence that can be empty.

THEOREM 3.1. *Suppose we are given a BAR tree T with depth $O(\log_{1/\beta} n)$, a balancing factor α , and a constant number of canonical vectors on a point set S with diameter D and n data points. For any query point q , a $(1 - \epsilon)$ -farthest neighbor to q can be found in $O((\alpha\sqrt{d})^d(1/\epsilon)^{d-1} \log_{1/\beta} n)$ time.*

Proof. Except for the last node, any visited leaf node, u' , with associated point p' , has $\Delta(u', q) > \delta(p, q) + \epsilon D$. Also, by the fact that p was the farthest point found we know that $\delta(p', q) \leq \delta(p, q)$. Thus, every leaf node visited by the algorithm must completely pierce the annulus A of radii $\delta(p, q)$ and $\delta(p, q) + \epsilon D$ centered at q . Now, we can categorize the query into one of two groups. If $\delta(p, q) \leq 2D$, we know from the packing lemma (3.2) that the number of leaf nodes visited is $L = O((\alpha\sqrt{d})^d(\delta(p, q)/(\epsilon D))^{d-1}) = O((\alpha\sqrt{d})^d(1/\epsilon)^{d-1})$. Otherwise, $\delta(p, q) > 2D$, but since the diameter of S is D , all the points lie inside a ball, B_S , of radius D . Since every leaf node must also lie at least partly inside B_S , the number of leaf nodes visited is actually the number of leaf nodes piercing $A' = A \cap B_S$. Using the same volume packing argument with the more constrained volume, we see that this case also yields $L = O((\alpha\sqrt{d})^d(D/(\epsilon D))^{d-1}) = O((\alpha\sqrt{d})^d(1/\epsilon)^{d-1})$.

Applying Lemma 3.1, we get a running time of

$$O\left(\log_{1/\beta} n(\alpha\sqrt{d})^d (1/\epsilon)^{d-1}\right),$$

and the stated bound follows. ■

4. CORNER-CUT BAR TREES

We now show a specific instance of a BAR tree. Ideally, when constructing a BAR tree we would prefer to use only axis-orthogonal cuts, but this is not possible. For example, if the majority of the points are concentrated at a particular corner of a hyperbox region, any axis-orthogonal cut is either too close to the opposing face or does not partition the points in the region well, resulting in large tree depth. Figure 9a illustrates this problem. Our solution is to introduce corner cuts that yield enough freedom of direction to construct a BAR tree.

DEFINITION 4.1. A *corner cut* in \mathbb{R}^d is a hyperplane whose normal vector is of the form (I_0, I_1, \dots, I_d) where $I_i \in \{1, -1\}$.

Note that there are 2^d possible corner cuts corresponding to all combinations of ± 1 . In the plane, such a cut forms a 45° angle with both the x - and the y -axes (see Fig. 3). The advantage of using these corner cuts becomes clear in Section 5. Let us now define a canonical cut set that is sufficient for constructing an efficient BAR tree.

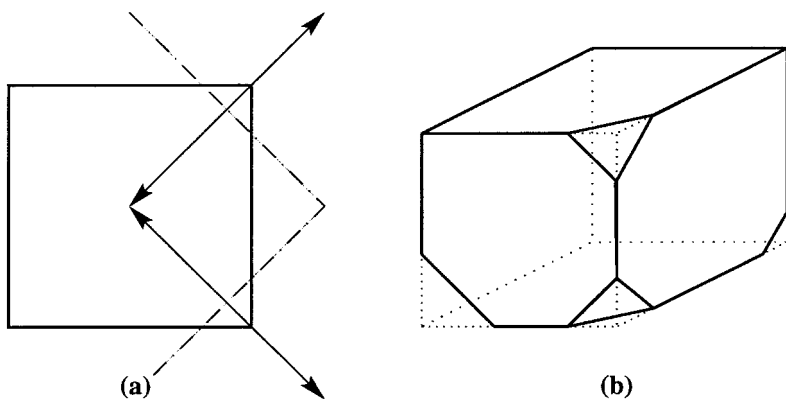


FIG. 3. (a) Corner cuts in the plane, (b) corner cuts in 3- d .

4.1. Corner Cut Canonical Set

Define the *corner cut canonical set*, \mathcal{E} , to be the set of all cuts which are either axis-orthogonal or corner cuts, that is, all cuts whose normal is of the form $(0, 0, \dots, 1, \dots, 0)$ or (I_0, I_1, \dots, I_d) where $I_i \in \{1, -1\}$. Let \mathcal{E}' be the set of axis-orthogonal cuts and \mathcal{E}'' be the set of corner cuts ($\mathcal{E} = \mathcal{E}' \cup \mathcal{E}''$).

For the remainder of this section, all references to the canonical cut set \mathcal{E} refer to the above defined corner cut canonical set. For a canonical region R consider a direction vector $\vec{v}_i \in \mathcal{E}$. Let $w_i(R)$ be the distance between any two bounding canonical cuts of R normal to \vec{v}_i . For simplicity, let us normalize the distance between two opposing planes by using the Euclidean metric, L_2 . For a region R , this means that for the two bounding canonical corner cuts b and c , with normal \vec{v}_i , whose equations are of the form $b: \vec{v}_i \vec{x} = a_1$ and $c: \vec{v}_i \vec{x} = a_2$, we have $w_i(R) = |a_1 - a_2| / \sqrt{d}$.

DEFINITION 4.2. For any canonical region R , we define several terms for the various sides of the region.

- The *maximum width of R* is $\max(R) = \max_{i \in \mathcal{E}}(w_i(R))$.
- The *maximum bounding box width of R* is

$$\max'(R) = \max_{i \in \mathcal{E}'}(w_i(R)).$$

• Similarly, we define the *minimum width of R* , $\min(R)$, and the *minimum bounding box width of R* , $\min'(R)$.

• A canonical region R has *canonical aspect ratio*, $\text{casp}(R) = \max(R) / \min(R)$.

In other words, the canonical aspect ratio is the ratio of the longest to smallest widths among the $2^{d-1} + d$ face pairs. Observe that because the bounding box is from a subset of the canonical cuts, namely all axis-orthogonal directions, we have $\max'(R) \leq \max(R) \leq \max'(R)\sqrt{d}$ and $\min'(R) \geq \min(R)$.

Since it is easier to use the canonical aspect ratio rather than the general aspect ratio for this section, let us briefly discuss why $\text{asp}(R) = \Theta(\text{casp}(R))$. Notice that $O_R \leq \max'(R)\sqrt{d}/2 \leq \max(R)\sqrt{d}/2 = O(\max(R))$. Also, notice that $O_R \geq \max(R)/2 = \Omega(\max(R))$. Therefore, the outer radius and the maximum width are equivalent, i.e., $O_R = \Theta(\max(R))$. Now let B be the largest ball inscribed in R with radius I_R . Notice that $I_R \leq \min(R)/2 = O(\min(R))$. Now, B either touches two parallel hyperplanes from R or is bounded by a tetrahedron \mathcal{T} formed by $d + 1$ different hyperplanes from R . If the former case is true, then

$I_R = \min(R)/2 = \Theta(\min(R))$. Otherwise, B is the largest inscribed ball in \mathcal{S} , and so $I_R = I_{\mathcal{S}}$. Let $\delta = \min(\mathcal{S})$ be the smallest distance between one plane and its opposing point, i.e., the smallest width in each of the $d + 1$ directions. Notice that $\delta = \min(\mathcal{S}) \geq \min(R)$. The regular tetrahedron τ whose height is δ must fit inside \mathcal{S} . Since d is assumed to be constant, we know that $I_{\tau} = \Theta(\delta)$. Therefore $I_R = I_{\mathcal{S}} \geq I_{\tau} = \Theta(\delta) = \Theta(\min(R))$, and, consequently, $I_R = \Theta(\min(R))$. Solving for the aspect ratio of R , we see that $\text{asp}(R) = O_R/I_R = \Theta(\max(R))/\Theta(\min(R)) = \Theta(\text{casp}(R))$.

As a result, for the remainder of this section, we call a canonical region R α -balanced if $\text{casp}(R) \leq \alpha$ for some $\alpha \geq 1$.

Even with the introduction of the corner cuts, it is not possible to guarantee that there exists a single cut that divides the region into two subregions with equal number of points and good aspect ratios. We do, however, prove that any balanced canonical region R is two-cuttable for sufficiently large values of α and β . Before we describe a simple linear time algorithm to determine the sequence of cuts, we give several important definitions and properties of the corner cut canonical set.

4.2. Corner Cut Set Properties

In the process of making a corner cut c' (parallel to a facet c) in a region R , it is possible to create a subregion, R_1 , such that some other bounding canonical cut $x \in R$ is no longer a bounding canonical cut in R_1 . In other words, x is no longer tangential to the new subregion R_1 . If this is the case, we simply use the bounding canonical cut $x' \in R_1$ which is parallel to x (see Fig. 4).

LEMMA 4.1. *Suppose we are given a convex polyhedral region R and two parallel hyperplanes l and r intersecting R (see Fig. 5a). We have three (possibly empty) subregions of R lying to the left, middle, and right of the two*

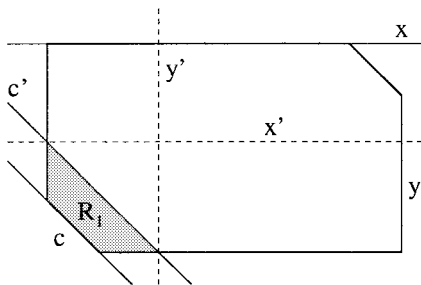


FIG. 4. Observe that x and y are not tangential to region R_1 but x' and y' are.

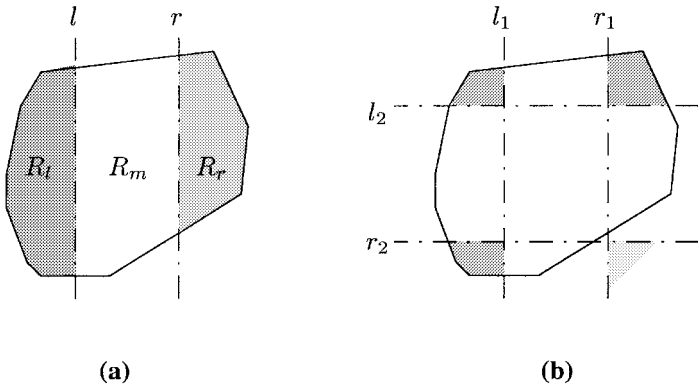


FIG. 5. (a) Two cuts l and r which divide a region R into subregions R_l , R_m , and R_r . (b) Pairs of parallel cuts (l_1, r_1) and (l_2, r_2) divide region R into several subregions. The shaded areas are the intersections of the outer regions. Note, one intersection is the emptyset $(r_1 \cap r_2)$.

hyperplanes, respectively, R_l , R_m , and R_r . For any $1 \geq \beta \geq 1/2$, one of the following is true:

- there exists a hyperplane m parallel to l and r intersecting R_m which divides the region into two subregions both of which have no more than $\beta|R|$ points, or
- no less than $\beta|R|$ points lie in either R_l or R_r .

Proof. Assume the contrary. Then, $|R_l| < \beta|R|$ implies that $|R_m| + |R_r| \geq \beta|R|$ or else the line l would be a suitable choice for m . Similarly, we know that $|R_l| + |R_m| \geq \beta|R|$. We now continually sweep a hyperplane m from l toward r . Let R_1 and R_2 be the two subregions of R to the left and right of m . $|R_1| = |R_l| + x$ for $0 \leq x \leq |R_m|$. This implies that there exists $x \in \{0, \dots, |R_m|\}$ such that $|R_1| = \beta|R|$ and $|R_2| = |R| - \beta|R| \leq \beta|R|$ for $\beta \geq 1/2$. Thus, there exists a hyperplane parallel to l and r which intersects R_m and divides R into two regions of size no more than $\beta|R|$, contradicting our original assumption. ■

In other words, either there exists a *dividing* cut that partitions a region into two subregions each with less than a constant fraction of the original number of points or one of the two outside regions has more than this fraction of points. In the latter case, we call this region the *dense outer region*, $\text{dor}_{l,r}(R)$, where l and r are the two parallel hyperplanes. If there exists a dividing cut, m , then $\text{dor}_{l,r}(R) = \emptyset$.

LEMMA 4.2. *Suppose we are given a convex region R and k pairs of parallel hyperplanes (l_i, r_i) , $i \in \{1, \dots, k\}$ (see Fig. 5b). For any $\beta \geq (k - 1)/k$ (and $\beta \geq 1/2$), one of the following is true:*

- *there exists a hyperplane m parallel to one of the pairs which divides the region into two subregions of size less than $\beta|R|$, or*
- *$|P| \geq (1 - (1 - \beta)k)|R|$, where $P = \bigcap_{i=1}^k \text{dor}_{l_i, r_i}(R)$.*

Proof. Assume that there does not exist a hyperplane m dividing the region into two small subregions. By Lemma 4.1, every pair must have a dense outer region for R or we would have a dividing hyperplane m . For each pair (l_i, r_i) , let $P_i = \text{dor}_{l_i, r_i}(R)$ be the dense outer region of R for each pair of hyperplanes.

If $k = 1$, we know from Lemma 4.1, $|P_1| \geq \beta|R| = (1 - (1 - \beta)k)|R|$. By induction, assume the theorem holds for all values less than k . Recall from set theory that for two sets A and B , $|A \cap B| = |A| + |B| - |A \cup B|$. Let $P' = \bigcap_{i=1}^{k-1} P_i$. We know that $|P' \cup P_k| \leq |R|$. Then, $P = P' \cap P_k \Rightarrow |P| \geq (1 - (1 - \beta)(k - 1))|R| + \beta|R| - |R| = (1 - (1 - \beta)k)|R|$. Thus, the result holds for all k . ■

One simple corollary that can be derived from this lemma is the following: if the intersection of all the dense outer regions is empty and $\beta > (k - 1)/k$, a hyperplane cut must exist that divides the region into two subregions of size less than $\beta|R|$ each, as there is no other alternative location for the points to lie.

DEFINITION 4.3. Given an α -balanced canonical region R and a canonical cut c with normal \vec{v}_i , sweep a cut c' from c toward the opposing cut b . Let P be the subregion of R between c and c' . If P is initially not a single point, which we consider to have good aspect ratio, sweep c' until the first time $\text{cas}_P(P) = \alpha$. Call the region P the *shield region of c in R* $s_c(R)$. Let the *dense outer shield* $\text{dos}_i(R)$ be the shield region $s_b(R)$ or $s_c(R)$, such that $|\text{dos}_i(R)| = \max(|s_b(R)|, |s_c(R)|)$.

Before we proceed, observe that the sweep must terminate. First, if the initial region P is not a point, $\text{cas}_P(P) = \infty > \alpha$. Since R is α -balanced, if c' reached b then $P = R$ and $\text{cas}_P(P) \leq \alpha$. Since the aspect ratio function is continuous, at some point during the sweep $\text{cas}_P(P)$ must equal α and the sweep terminates.

PROPERTY 4.1. *For any vector \vec{v}_i with canonical cut c , $w_i(s_c(R)) \leq \max(R)/\alpha$.*

COROLLARY 4.1. *Given a canonical region R , a vector $\vec{v}_i \in \mathcal{E}$, and a reduction factor $1 > \beta \geq 1/2$, let b and c be the two opposing canonical cuts*

in R with normal \vec{v}_i . If $s_b(R) \cap s_c(R) = \emptyset$, then one of the following holds:

- there exists a cut which divides R into two α -balanced regions R_1 and R_2 each with less than $\beta|R|$ points, or
- $|\text{dos}_i(R)| > \beta|R|$.

Proof. Since the two regions do not intersect, the cuts defining the two shields divide R into three regions, and as in Lemma 4.1, the size constraints hold. More importantly, by the definition of a shield region (Definition 4.3) and the fact that the two shield regions do not intersect, any cut lying between the two regions will produce two α -balanced regions.



The corollary above corresponds to Lemma 4.1, with the added guarantee that the two regions produced from the chosen cut are α -balanced. Furthermore, this corollary can be extended to multiple hyperplane directions:

COROLLARY 4.2. *Given a canonical region R , a subset $C \subseteq \mathcal{E}$, and a reduction factor $\beta \geq (|C| - 1)/|C| \geq 1/2$, if for all cuts $\vec{v}_i \in C$ with opposing bounding cuts for R , b and c , $s_b(R) \cap s_c(R) = \emptyset$, then one of the following holds:*

- R is one-cuttable with reduction factor β , or
- $|P| \geq (1 - (1 - \beta)|C|)|R|$, where $P = \bigcap_{\vec{v}_i \in C} \text{dos}_i(R)$.

DEFINITION 4.4. For a given canonical region R and a canonical bounding cut c , the *facet* of c , $f_c(R)$, is defined as the intersection of R with the hyperplane c . The *length* of c , $l_c(R)$, is defined as the maximum width of the facet of c , denoted by $\max(f_c(R))$.

Note in Fig. 6c that the size of the shield region is much smaller than in Figs. 6a and 6b. This is because the length of the diagonal facet is

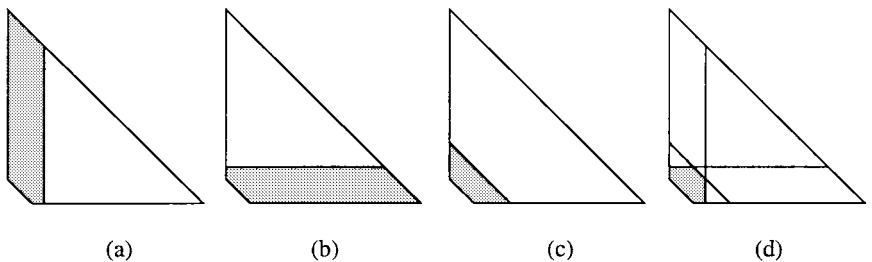


FIG. 6. Dense outer shields of R in the direction of (a) x , (b) y , and (c) the diagonal. The intersection region P of all of these shield regions (d), contains at least a fraction of the points.

significantly smaller than the other two sides. After proving two essential properties we show how to bound the width of a shield region by a quantity that depends only upon α , the dimension d , and the length of the shield's facet.

PROPERTY 4.2. *Suppose we are given a canonical region $R \in \mathbb{R}^d$. For any canonical cut $\vec{v}_i \in \mathcal{C}$ with bounding cut c and any other canonical cut $\vec{v}_j \in \mathcal{C}$ we have $w_j(R) - w_j(\mathfrak{f}_c(R)) \leq 2 dw_i(R)$ and $\max(R) \leq 2 dw_i(R) + \perp_c(R)$.*

Proof. This result follows from the fact that in all of the γ directions, there exists a minimum angle formed among any pair of planes. Let θ_i be the minimum angle between \vec{v}_i and any other vector in \mathcal{C} . Let $\delta = w_i(R)$, $f = \mathfrak{f}_c(R)$, and b be the opposing cut to c . Consider any point p on f . Define \mathcal{F}_p to be the d -dimensional cone extending from p in the direction \vec{v}_i that forms an angle θ_i with the hyperplane defined by c . Furthermore, let F_p be the portion of \mathcal{F}_p between the hyperplanes c and b . The height of this section is δ ; see Fig. 7a.

Let \vec{v}_j be any vector, not necessarily in \mathcal{C} , orthogonal to \vec{v}_i . Project the cone onto the 2-dimensional plane whose x - and y -axes are formed by the two vectors, \vec{v}_i and \vec{v}_j . Observe that $w_j(F_p) = 2\delta \cot \theta_i$. Define the conic union to be $F = \bigcup_{p \in f} F_p$. It follows that $w_j(T) = w_j(f) + w_j(F_p)$; see Fig. 7b.

Assume that $(R \cap F) \subset R$, i.e., part of R lies outside of F . Since R is convex there exists a facet f' of R , intersecting a point p on the facet f and lying (partially) outside the conic union F . Let us examine the angle formed between the two cut directions \vec{v}_i and \vec{v}_i' defining f and f' . Since f' lies outside of the cone which is directed along \vec{v}_i , the angle between \vec{v}_i and \vec{v}_i' is less than θ_i . Also since f' is a facet of the canonical region, \vec{v}_i'

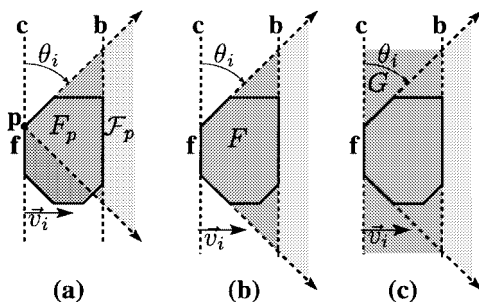


FIG. 7. (a) The shaded region F_p represents the cone extending distance δ from a point p on the diagonal facet f . (b) The shaded region F represents the union of all cones extending from every point on the diagonal facet f . (c) The shaded region G represents the cylinder encompassing F .

must be a canonical cut direction in \mathcal{E} . However, θ_i is defined to be the minimum angle between \vec{v}_i and every other vector in \mathcal{E} . Therefore, R must be contained inside the conic section F . It follows then that

$$\begin{aligned} w_j(R) &\leq w_j(F) \\ &= w_j(f) + w_j(F_p) \\ &= w_j(f) + 2\delta \cot(\theta_i) \\ &= w_j(\underline{f}_c(R)) + 2\cot(\theta_i)w_i(R). \end{aligned}$$

Let us now look at any other vector $\vec{v}_j \in \mathcal{E}$. Recall that $w_j(R) = \delta(b_j, c_j)$. Also, let $g = \underline{f}_b(F)$ be the end facet of the conic union F . To simplify our proof, let us create a d -dimensional ‘‘cylinder’’ G encompassing F with length δ and with a base equal to g ; see Fig. 7c. In other words, the base has diameter less than the maximum width of the cone in any orthogonal direction to \vec{v}_i . Therefore, since $G \supset F \supseteq R$, we get

$$\begin{aligned} w_j(R)^2 &= \delta(b_j, c_j)^2 \\ &\leq w_j(G)^2 \\ &\leq w_j(g)^2 + w_i(G)^2 \\ &= w_j(g)^2 + \delta^2 \\ &\leq (w_j(g)^2 + \delta)^2 \\ &\leq (w_j(f) + (2\cot(\theta_i)\delta) + \delta)^2 \\ &= (w_j(f) + (2\cot(\theta_i) + 1)\delta)^2. \end{aligned}$$

To establish our first bound, note that $\cot(\theta_i)$ is maximized when θ_i is minimized. This occurs between two corner cuts whose vectors are nearly identical, i.e., between vectors with normals $(1, 1, \dots, 1)$ and $(1, 1, \dots, -1)$. Solving for the cotangent, we see that $2\cot(\theta_i) + 1 < 2d$.

Let \vec{v}_j be the direction corresponding to the maximum width of R . Therefore, $\max(R) = w_j(R) \leq 2dw_i(R) + w_j(\underline{f}_c(R))$. Recall that $1_c(R) = \max(\underline{f}_c(R)) \geq w_j(\underline{f}_c(R))$ yielding the desired second bound. ■

PROPERTY 4.3. *Suppose we are given a canonical region R , a canonical cut $\vec{v}_i \in \mathcal{E}$ with bounding cut c , and any other canonical cut $\vec{v}_j \in \mathcal{E}$. Let c' be a cut parallel to c and intersecting R . Let R' be the subregion of R between c and c' . If $w_j(R') - w_j(\underline{f}_c(R')) < 2w_i(R')/\sqrt{d}$, then $w_j(R) = w_j(R')$.*

Proof. This is a simple extension of the proof of the previous property. To illustrate imagine incrementally increasing R' by sweeping c' from c out toward the opposing facet of c . In any given direction, $\vec{v}_j, w_j(R')$ certainly cannot decrease. As we move c' a further distance δ_i away from c , $w_j(R')$ also increases by an amount $\delta_j \geq 0$. Since the regions are convex, once δ_j becomes zero it must remain zero. This implies that $w_j(R) = w_j(R')$ for the remainder of the sweep after δ_j becomes zero.

As δ_i approaches zero, the ratio δ_j/δ_i represents the slope of δ_j . This slope is determined by the angles between the normals of the facets and \vec{v}_j and \vec{v}_i . Since the normals to the facets defining both R and R' are fixed, the possible values for this slope, δ_j/δ_i , are discrete. Therefore, there exists a lowest positive value before the change must be zero. Calculating the various angles from the canonical cut set shows that this value is at least $2/\sqrt{d}$. ■

LEMMA 4.3 (Corner cut shield lemma). *For an α -balanced canonical region R and a canonical cut $\vec{v}_i \in \mathcal{C}$ with bounding cut c , we have $w_i(s_c(R)) \leq \sqrt{d} \perp_c(R)/(2\alpha - 2d\sqrt{d})$, for $\alpha > d\sqrt{d}$.*

Proof. Let $\chi = \sqrt{d} \perp_c(R)/(2\alpha - 2d\sqrt{d})$. Since $s_c(R) \subseteq R$, if $w_i(R) \leq \chi$, then $w_i(s_c(R)) \leq w_i(R) \leq \chi$, and we are done.

Now, let c' be the canonical cut intersecting R parallel to c such that $\delta(c, c') = \chi$. Let $R' \subset R$ between c and c' . It suffices to show that R' is α -balanced. Note that $w_i(R') = \chi$. Let $\vec{v}_j \in \mathcal{C}$ be the canonical cut direction such that $w_j(R') = \max(R')$. From Property 4.2, we know that $\max(R') \leq 2d\chi + \perp_c(R')$. Let $\vec{v}_k \in \mathcal{C}$ be the canonical cut such that $w_k(R') = \min(R')$. If $k = i$, then $\min(R') = \chi$. Otherwise, if $w_k(R') = w_k(R)$, then $\min(R') = w_k(R') \geq \min(R)$. Finally, if $w_k(R') \leq w_k(R)$, then by Property 4.3, we know that $\min(R') = w_k(R') \geq 2w_i(R')/\sqrt{d} = 2\chi/\sqrt{d}$.

Recall that the canonical aspect ratio of R' is $\text{casp}(R') = \max(R')/\min(R')$. If $\min(R') \geq \min(R)$, we are done since $\max(R') \leq \max(R)$ and R is α -balanced. Otherwise, since $\min(R') \geq \chi/2$, we have $\text{casp}(R') \leq \sqrt{d}(2d\chi + \perp_c(R))/(2\chi) = \sqrt{d}d + (\alpha - \sqrt{d}d) = \alpha$. ■

In other words, suppose we are given some direction $\vec{v}_i \in \mathcal{C}$ with bounding cut c and assume we have some facet $f_c(R)$ with size x . We can make a cut c' close to c producing an α -balanced subregion R_1 between c and c' such that $w_i(R) = \delta(c, c')$ is dependent on x and some other constant values.

For simplicity, in this lemma we make a large overestimation, particularly for cases in the plane, where the equation can be reduced even further to $2\perp_c(R)/\alpha$. To illustrate this lemma further, let us look at the example in Fig. 8. Since it is in the plane, we use the simpler tighter

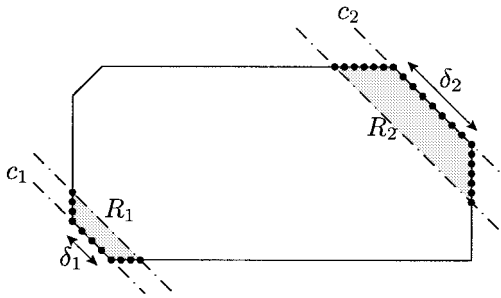


FIG. 8. Two opposing shield regions of R , in the direction $z \in \mathcal{E}$. Observe how the size of the shields depends on the lengths of the corresponding “faces,” δ_1 and δ_2 , and not on the size of R . Also, note that grid (marker) points are placed along each face to indicate distance.

formula. Let R be the main canonical region in the example. Let $\vec{v}_i \in \mathcal{E}$ be the direction creating the shield regions R_1 and R_2 , with aspect ratio $\alpha = 14/3$. Call the two opposing cuts c_1 and c_2 . First consider region R_1 . Note that $l_{c_1}(R) = l_{c_1}(R_1) = \delta_1 = 4\sqrt{2}$. Also, note that $w_i(R_1) = 3\sqrt{2}/2$, and consequently, $\max(R_1) = 7\sqrt{2}$. This gives us the aspect ratio, $\text{cas}_P(R_1) = \max(R_1)/w_i(R_1) = 14/3 = \alpha$. Now observe that $2 \cdot l_{c_1}(R)/\alpha = 8\sqrt{2}/(14/3) = 12\sqrt{2}/7 > 3\sqrt{2}/2 = w_i(R_1)$, which agrees with Lemma 4.3. Similarly, R_2 also has aspect ratio $\text{cas}_P(R_2) = \alpha$, but in this case $\delta_2 = 8\sqrt{2}$.

5. TWO-CUT EXISTENCE THEOREM

The corner cut shield lemma is very useful in constructing BAR trees but only if we can prove that there do exist cuts for a given canonical region whose length is below some constant fraction of the size of the region.

DEFINITION 5.1. For a canonical cut set \mathcal{E} and a canonical region R , an ϵ -corner $B \subseteq R$ is any ball with center q and radius ϵ such that, for every cut direction $\vec{v}_i \in \mathcal{E}$ with bounding cuts b and c , either b or c intersects B ; i.e., $\min(\delta(q, b), \delta(q, c)) \leq \epsilon$.

Note that this ϵ -corner is merely a conceptual extension of the standard corner of a rectangular box. In other words, for a point q , every pair of opposing canonical cuts has at least one cut relatively close to q . These ϵ -corners show worst-case performances for naive canonical cut sets. For example, Fig. 9a is a potential canonical region formed using the set of axis-orthogonal cuts in the plane, and Fig. 9b is a potential canonical

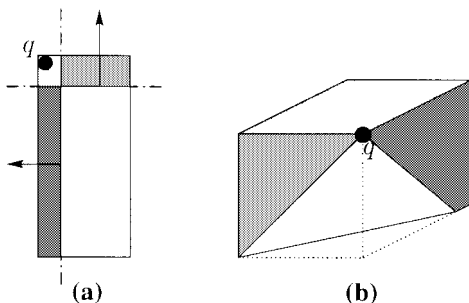


FIG. 9. (a) A bad corner using a canonical set of axis-orthogonal cuts, i.e., x and y directions. Notice a cut in either the x or y direction dividing the points located inside q would cause a skinny region. (b) A bad ϵ -corner in three-dimensions using a canonical set of axis-orthogonal cuts and one direction of the form $(1, 1, 1)$. Notice the ball at q touches all four possible facets.

region formed using the set of axis-orthogonal cuts and only one corner cut in space. Observe that the corner point q touches all four cut directions, and more importantly each facet has a “large” size. Place a large concentration of data points arbitrarily close to p . Notice that after any cut that preserves aspect ratio, the resulting subregion still has the corner p and the same concentration of data points. In fact, if we added two more corner cuts to this set, a similar corner could still be constructed. This does not occur, however, if we add every corner cut to the set.⁶ Let us now establish an important property of the corner cut canonical cut set, \mathcal{C} .

PROPERTY 5.1. *For any canonical region R , if there exists an ϵ -corner, $B \subseteq R$, then there exists a canonical corner cut $c \in R$ intersecting B such that $\perp_c(R) < 2\sqrt{2} d\epsilon$.*

Proof. Since B must intersect every axis-orthogonal cut plane, we know it is located near one of the 2^d corners of the bounding hyperbox of R . Because the corner cuts are all symmetrical, we can assume without loss of generality that this corner of R is the origin. Thus, the equations for the axis-orthogonal cut planes defining R and intersecting B are of the form $x_i = 0$, for all $i \in \{1, \dots, d\}$. Let $q = (q_1, q_2, \dots, q_d)$ be the center of B . Since B has radius ϵ and intersects each of the above axis-orthogonal planes, we have $-\epsilon \leq q_i \leq \epsilon$, for all $i \in \{1, \dots, d\}$.

Let us now examine the corner cut plane c with equation $(1, 1, \dots, 1)\vec{x} = a$. Since B intersects c , it follows that $0 \leq a < 2 d\epsilon$. Since the facet of c is bounded by the axis-orthogonal cuts defining R , we see that $\perp_c(R) = \max(\mathfrak{f}_c(R)) \leq \sqrt{2} a < 2\sqrt{2} d\epsilon$. ■

⁶ For dimensions higher than three, it is possible that a much smaller subset of corner cuts suffices.

This property states that if there exists an ϵ -corner for our canonical cut set, then there must exist a very small facet which intersects that corner. Note that we are again overestimating the actual size of the smallest facet, but for brevity we leave the bound as stated. The next two lemmas demonstrate the use of this ϵ -corner.

LEMMA 5.1. *For any α -balanced canonical region R with $\alpha > 2d\sqrt{d}$ and $\epsilon < \max'(R)/(6d)$, if there exists an ϵ -corner B such that $|R \cap B| \geq (1 - \beta)|R|$, then one of the following holds:*

- R is one-cuttable, or
- there exists a cut direction, $\vec{v}_i \in \mathcal{C}$, such that $\max(\text{dos}_i(R)) < 4\sqrt{2} d\epsilon$.

Proof. Let us assume that R is not one-cuttable. If there exists an ϵ -corner B , by Property 5.1, there exists a corner cut $c \in R$ with direction \vec{v}_i intersecting B such that $1_c(R) \leq 2\sqrt{2} d\epsilon$. Let b be the opposing bounding cut to c . Without loss of generality, let us assume that $\vec{v}_i = (1, 1, \dots, 1)$ and that the bounding box is translated to the origin. If the equation of the cut c is $\vec{v}_i \vec{x} = c'$, then $c' \leq 2 d\epsilon$. Let $m = \max'(R) = \max_{c \in \mathcal{C}'}(w_c(R))$; i.e., m is the length of the longest side of the bounding box of R . If the equation of the cut b is $\vec{v}_i \vec{x} = b'$, then $b' \geq m$. Also, recall the condition that $\epsilon < m/(6d)$. Thus,

$$\begin{aligned} w_i(R) &= (b' - c')/\sqrt{d} \\ &\geq (m - 2 d\epsilon)/\sqrt{d} \\ &> \frac{4m}{6\sqrt{d}}. \end{aligned}$$

Let $P = \mathfrak{s}_c(R)$ and $Q = \mathfrak{s}_b(R)$. From the corner cut shield lemma (4.3), Property 5.1, and the bounds on ϵ and α ,

$$\begin{aligned} w_i(P) &= w_i(\mathfrak{s}_c(R)) \\ &\leq \frac{\sqrt{d} 1_c(R)}{2\alpha - 2d\sqrt{d}} \\ &\leq \frac{d\sqrt{2d} \epsilon}{\alpha - d\sqrt{d}} \\ &< \frac{md\sqrt{2d}}{6d^2\sqrt{d}} \\ &= \frac{m\sqrt{2}}{6d}. \end{aligned}$$

Because R is α -balanced, we can also bound $w_i(Q)$ by

$$\begin{aligned} w_i(Q) &= w_i(\mathfrak{S}_b(R)) \\ &\leq \max(R)/\alpha \\ &\leq \frac{m\sqrt{d}}{6d}. \end{aligned}$$

Now, combining the two previous equations, and assuming $d \geq 2$, we see that

$$\begin{aligned} w_i(P) + w_i(Q) &< \frac{m\sqrt{2}}{6d} + \frac{m\sqrt{d}}{6d} \\ &< \frac{m}{6d}(2\sqrt{d}) \\ &< w_i(R). \end{aligned}$$

Thus, it follows that $P \cap Q = \emptyset$. Next, observe that

$$\begin{aligned} \delta(c, Q) &= \delta(b, c) - w_i(Q) \\ &= w_i(R) - w_i(Q) \\ &> \frac{4m\sqrt{d}}{6d} - \frac{m\sqrt{d}}{6d} \\ &\geq \frac{3m\sqrt{d}}{6d} \\ &> \frac{3m}{6d} \\ &> 3\epsilon. \end{aligned}$$

Since B intersects c and has radius ϵ , it follows that $B \cap Q = \emptyset$. Since R is not one-cuttable and $P \cap Q = \emptyset$, by Corollary 4.1, either P or Q has more than $\beta|R|$ points. If $|Q| > \beta|R|$, then $|Q \cup B| = |Q| + |B| - |Q \cap B| = |Q| + |B| > |R|$. This is impossible, since the points in Q and B are all points in R . Therefore, $|P| > \beta|R|$, and $P = \mathfrak{S}_c(R) = \text{dos}_i(R)$. Finally,

notice that we can bound the size of this dense outer shield by

$$\begin{aligned}
 & \max(\text{dos}_i(R)) \\
 &= \max(P) \\
 &\leq 2 dw_i(P) + 1_c(P) && \text{From Property 4.2} \\
 &\leq d\sqrt{d} 1_c(R)/(\alpha - d\sqrt{d}) + 1_c(R) && \text{From the Corner Cut Shield} \\
 & && \text{Lemma 4.3} \\
 &< 1_c(R) + 1_c(R) && \text{Since } \alpha > 2d\sqrt{d} \\
 &\leq 4\sqrt{2} d\epsilon && \text{From } 1_c(R) \leq 2\sqrt{2} d\epsilon.
 \end{aligned}$$

■

We now show that if a region is not one-cutttable there must exist an ϵ -corner with $\epsilon = O(\alpha)$ containing a large fraction of the points.

LEMMA 5.2. *For any α -balanced canonical region R and $\beta \geq d/(d + 1)$, one of the following holds:*

- R is one-cutttable, or
- there exists an ϵ -corner B such that $|R \cap B| \geq (1/(d + 1))|R| \geq (1 - \beta)|R|$ and $\epsilon < 2\sqrt{d} \max(R)/\alpha$.

Proof. If R is one-cutttable, we are done. So, assume R is not one-cutttable. First, let us examine the canonical subset $C \subseteq \mathcal{E}'$, where $\vec{v}_i \in C$ implies that $w_i(R) > 2 \max(R)/\alpha$. For any $\vec{v}_i \in C$, let c and b be the opposing bounding cuts of R . Recall that $w_i(s_c(R)) \leq \max(R)/\alpha$. Since $w_i(R) > 2 \max(R)/\alpha$, we know that $s_c(R) \cap s_b(R) = \emptyset$. Note that $d \geq |C|$ implies $\beta > d/(d + 1) > (|C| - 1)/|C|$. Let $P = \bigcap_{\vec{v}_i \in C} \text{dos}_i(R)$. Since R is not one-cutttable, by Corollary 4.2 we obtain

$$\begin{aligned}
 |P| &\geq (1 - (1 - \beta)|C|)|R| \\
 &\geq \left(1 - \left(1 - \frac{d}{d + 1}\right)|C|\right)|R| \\
 &= \left(1 - \frac{|C|}{d + 1}\right)|R| \\
 &\geq \left(1 - \frac{d}{d + 1}\right)|R| \\
 &\geq (1 - \beta)|R|.
 \end{aligned}$$

For all $\vec{v}_i \in \mathcal{E}'$, if $\vec{v}_i \in C$, then $w_i(P) \leq w_i(\text{dos}_i(R)) \leq \max(R)/\alpha$; otherwise $w_i(P) \leq w_i(R) \leq 2 \max(R)/\alpha$. So, we can bound P with a ball B' of radius ϵ' such that $\epsilon' \leq \sqrt{d} \max(R)/\alpha$. Let B be the ball concentric to B' with radius $\epsilon = 2\sqrt{d} \max(R)/\alpha$. We now show that for any $\vec{v}_i \in \mathcal{E}$ with opposing cuts $c, b \in R$, either c or b intersects B . This is already true for all axis-orthogonal directions, since one of each pair intersects B' and hence B . Let us look at the set of corner cut directions, \mathcal{E}'' . For any $\vec{v}_i \in \mathcal{E}''$, if $w_i(R) \leq 2 \max(R)/\alpha$, then both cuts intersect B . Otherwise, $s_b(R) \cap s_c(R) = \emptyset$. Since R is not one-cuttable and the shield regions do not intersect, we obtain from Corollary 4.1 that $|\text{dos}_i(R)| > \beta|R|$. Assume without loss of generality that c is the cut such that $s_c(R) = \text{dos}_i(R)$. As with all shield regions, $w_i(s_c(R)) \leq \max(R)/\alpha$. Since $|s_c(R)| = |\text{dos}_i(R)| > \beta|R|$ and $|P| > (1 - \beta)|R|$, $s_c(R)$ must intersect P and consequently B' . Therefore, $\delta(c, B') \leq w_i(s_c(R)) \leq \max(R)/\alpha$. Thus, cut c intersects B . By Definition 5.1, B is an ϵ -corner where $|R \cap B| \geq |R \cap B'| \geq |P| > (1 - \beta)|R|$ and $\epsilon = 2\sqrt{d} \max(R)/\alpha$. ■

Before we introduce and prove the two-cuttable theorem, we give a brief description of a technique for finding the two cuts, s_1 and s_2 , necessary to partition a given α -balanced canonical region R . The algorithm for constructing a BAR tree for a given point set follows directly from this technique. If R is one-cuttable, let s_1 be a one-cut in R dividing R into two subregions R_1 and R_2 , and note that s_2 is not needed. Otherwise, let $\vec{v}_i \in \mathcal{E}$ with bounding cut c be the canonical direction such that $\max(\text{dos}_i(R)) = \min_{\vec{v}_j \in \mathcal{E}}(\max(\text{dos}_j(R)))$. In other words, let c be the bounding cut associated with the *smallest* dense outer shield. Let $P = s_c(R)$. Let c' be the cut parallel to c intersecting R such that $\delta(c, c') = \perp_c(R)/(2d)$. Let P' be the region of R between c and c' . We prove that if R is not one-cuttable then P' is. Let s_1 be the cut producing the region P' . Let s_2 be the one-cut dividing P' into two subregions.

THEOREM 5.1. *For $\alpha > 18d\sqrt{d}$ and $\beta \geq (d + 1)/(d + 2)$, any α -balanced canonical region R is two-cuttable.*

Proof. If R is one-cuttable, we have nothing to prove, so assume R is not one-cuttable. By Lemma 5.2 there must exist an ϵ -corner B such that $|R \cap B| \geq (1 - \beta)|R|$ and $\epsilon < 2\sqrt{d} \max(R)/\alpha < 2d \max'(R)/(16d^2) < \max'(R)/(6d)$. Since R is not one-cuttable, from Lemma 5.1 there exists a cut direction $\vec{v}_i \in \mathcal{E}$ such that $\max(\text{dos}_i(R)) < 4\sqrt{2} d\epsilon$. In particular, let $\vec{v}_i \in \mathcal{E}$ be the direction such that $\max(\text{dos}_i(R)) \leq \max(\text{dos}_j(R))$ for all $\vec{v}_j \in \mathcal{E}$. Let $c, b \in R$ be the bounding canonical cuts associated with \vec{v}_i such that $s_c(R) = \text{dos}_i(R)$. Let $P = s_c(R)$ and $Q = s_b(R)$. As in the proof of Lemma 5.1, we know that $|P| > \beta|R|$.

Let c' be the cut parallel to c intersecting R such that $\delta(c, c') = 1_c(R)/(2d)$. Let P' be the region of R between c and c' . From Lemma 4.3, we know that $w_i(P) \leq \sqrt{d} 1_c(R)/(2\alpha - 2d\sqrt{d})$. Since $\alpha > 2d\sqrt{d}$, $w_i(P) < \sqrt{d} 1_c(R)/(2d\sqrt{d}) = 1_c(R)/(2d)$ and P must be a subset of P' . Following the same proof as in Lemma 5.1, we see $P' \cap Q = \emptyset$. Therefore, P' and $R \setminus P'$ are both α -balanced regions.

Now, assume P' is not one-cutttable. From Lemma 5.2 there exists an ϵ -corner $B' = (q', \epsilon')$ such that $|P' \cap B'| \geq (1/(d+1))|P'| \geq (1-\beta)|P'|$ and $\epsilon' < 2\sqrt{d} \max(P')/\alpha$. First, note that B' and P must intersect. Otherwise,

$$\begin{aligned} |P'| &\geq |(P' \cap B') \cup P| \\ &= |P' \cap B'| + |P| \\ &> (1-\beta)|P'| + \beta|R| \\ &\geq |P'|. \end{aligned}$$

From Property 4.2, we know that $\max(P') \leq 2dw_i(P') + 1_c(P') = 2d1_c(P')/(2d) + 1_c(P') = 21_c(P')$. We can now bound the radius of the ball B' as follows:

$$\begin{aligned} \epsilon' &< 2\sqrt{d} \max(P')/\alpha \\ &\leq 4\sqrt{d} 1_c(P')/\alpha \\ &< 4\sqrt{d} 1_c(P')/(18d\sqrt{d}) \\ &= 41_c(P')/(18d). \end{aligned}$$

Also, observe the following:

$$\begin{aligned} \delta(P, c') &= w_i(P') - w_i(P) \\ &\geq 1_c(P')/(2d) - \sqrt{d} 1_c(P')/(2\alpha - 2d\sqrt{d}) \\ &\quad \text{From the Corner Cut Shield Lemma 4.3} \\ &> 1_c(P')/(2d) - 1_c(P')/(34d) \\ &= 81_c(P')/(17d) \\ &> 2\epsilon'. \end{aligned}$$

Since B' is an ϵ -corner, it must intersect either c or c' . Because $\delta(P, c') > 2\epsilon$, B' cannot intersect both c' and P . As a result, B' intersects c . Since B' can then only intersect canonical cuts that are identical in both P' and R ,

B' must also be an ϵ -corner for R . However, note that

$$\begin{aligned}
 |R \cap B'| &\geq |P' \cap B'| \\
 &\geq \frac{1}{d+1} |P'| \\
 &\geq \frac{1}{d+1} \beta |R| \\
 &\geq \frac{1}{d+1} \frac{d+1}{d+2} |R| \\
 &= \frac{1}{d+2} |R| \\
 &\geq (1-\beta) |R|.
 \end{aligned}$$

Thus, from Lemma 5.1 there exists a cut direction $\vec{v}_j \in \mathcal{E}$ such that $\max(\text{dos}_i(R)) < 4\sqrt{2} d \epsilon' < \max(P)$. This contradicts the fact that P was the smallest dense outer shield. Therefore, P' must be one-cuttable. ■

This ability to always guarantee two-cuttable regions adds a lot of flexibility to the BAR tree construction. In particular, although it is typical that most axis-orthogonal cuts produce a sufficient one-cut, one may opt to first do a sequence of k , for some appropriate value of k , axis-orthogonal cuts which may not divide the point set into a small enough portion but still maintain good aspect ratio. In general, one may hope that such a sequence does divide the point set adequately. After k steps, however, the optional corner cuts may be incorporated as a safety net for two cuts which do produce proper balancing.

THEOREM 5.2. *For the given canonical set \mathcal{E} , a BAR tree with depth $O(d \log n)$ and balancing factor α can be constructed in $O(d\gamma n \log n)$ time, where γ is the size of the canonical set. In particular, the running time of the algorithm is $O(n \log n)$ for fixed dimensions.*

Proof. Notice that at any stage, using even a naive searching technique, the dense outer shields of a region R can be found in $O(|R|\gamma)$ time. Subsequently, it only takes $O(\gamma)$ time to determine the smallest such region. Similarly, P' can be constructed in $O(|R|\gamma)$ time. Therefore, we can find any one- or two-cut of a region R in $O(|R|\gamma)$ time. Because the depth of the BAR tree is bounded by $O(\log_{1/\beta} n)$, the running time becomes $O(\gamma n \log_{1/\beta} n)$. Finally, since $\beta = O((d+1)/(d+2))$ implies that $\log_{1/\beta} n = O(d \log n)$, we get the above stated running time. ■

6. CONCLUSION AND OPEN PROBLEMS

In this paper, we introduce the general framework of the BAR tree and describe an important application that can be solved using this type of tree. We also show that in fixed dimensions an (α, β) -BAR tree can be constructed in $O(n \log n)$ time, where n is the number of points in the data set.

These results, however, are only preliminary. There are still many open problems for this new type of data structure. We state only a few of them.

- Construct a canonical set with size $o(2^d)$ which guarantees that every canonical region is k -cuttable. We believe it can be done with a canonical set of size $O(d^2)$.

- Drop the dependency of α to $O(\sqrt{d})$, possibly using more careful analysis.

- Improve the running time bounds of the various geometric query problems already known.

- Prove other exact or approximate bounds on geometric problems. Are better results possible using randomization in the data set or in the construction?

- Explore applications outside geometry, particularly in those areas, such as computer graphics, geographical information systems, and pattern matching, already dependent on many geometric techniques.

For many of these problems, much research has already begun, but there are still many uses and questions for BAR trees left to be discovered.

ACKNOWLEDGMENTS

The authors would like to thank Rao Kosaraju and Dave Mount for their invaluable advice and guidance in the early stages of this research. We would also like to thank the anonymous referees for their help in improving the original conference paper.

REFERENCES

1. P. K. Agarwal and J. Erickson, Geometric range searching and its relatives, in "Advances in Discrete and Comput. Geom." (B. Chazelle, E. Goodman, and R. Pollack, Eds.), Amer. Math. Soc., Providence, RI, 1998.
2. P. K. Agarwal, E. F. Grove, T. M. Murali, and J. S. Vitter, Binary space partitions for fat rectangles, in "Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci., Oct. 1996," pp. 482–491.
3. P. K. Agarwal, M. Katz, and M. Sharir, Computing depth orders for fat objects and related problems, *Comput. Geom. Theory Appl.* **5** (1995), 187–206.

4. W. G. Aref and H. Samet, Perspective viewing of objects represented by octrees, *Comput. Graph Forum* **14** (1995), 59–66. [Also University of Maryland Computer Science TR-2757.]
5. S. Arya and D. M. Mount, Approximate range searching, in “Proc. 11th Annu. ACM Sympos. Comput. Geom., 1995,” pp. 172–181.
6. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, An optimal algorithm for approximate nearest neighbor searching, in “Proc. 5th ACM-SIAM Sympos. Discrete Algorithms, 1994,” pp. 573–582.
7. F. Aurenhammer, “Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure,” Report B-90-09, Fachber. Math., Free Univ. Berlin, Berlin, West Germany, 1990.
8. J. L. Bentley, Multidimensional binary search trees used for associative searching, *Commun. Assoc. Comput. Mach.* **18** (1975), 509–517.
9. J. L. Bentley, K -d trees for semidynamic point sets, in “Proc. 6th Annu. ACM Sympos. Comput. Geom., 1990,” pp. 187–197.
10. J. L. Bentley and J. H. Friedman, Data structures for range searching, *ACM Comput. Surv.* **11** (1979), 397–409.
11. J. L. Bentley and H. A. Maurer, Efficient worst-case data structures for range searching, *Acta. Inform.* **13** (1980), 155–168.
12. P. B. Callahan and S. R. Kosaraju, Algorithms for dynamic closest-pair and n -body potential fields, in “Proc. 6th ACM-SIAM Sympos. Discrete Algorithms, 1995,” pp. 263–272.
13. P. B. Callahan and S. R. Kosaraju, A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields, *J. Assoc. Comput. Mach.* **42** (1995), 67–90.
14. B. Chazelle, Lower bounds for orthogonal range searching, I: The reporting case, *J. Assoc. Comput. Mach.* **37** (1990), 200–212.
15. H. H. Chen and T. S. Huang, A survey of construction and manipulation of octrees, *Comput. Vision Graphics Image Process* **43** (1988), 409–431.
16. N. Chin and S. Feiner, Near real-time shadow generation using BSP trees, in “Proc. SIG-GRAPH ’89,” *Comput. Graph.* **23** (1989), 99–106.
17. M. de Berg, Linear size binary space partitions for fat objects, in “Proc. 3rd Annu. European Sympos. Algorithms,” Lecture Notes in Computer Science, Vol. 979, pp. 252–263, Springer-Verlag, Berlin/New York, 1995.
18. D. P. Dobkin and H. Edelsbrunner, Space searching for intersecting objects, *J. Algorithms* **8** (1987), 348–361.
19. C. A. Duncan, M. T. Goodrich, and S. G. Kobourov, Balanced aspect ratio trees and their use for drawing very large graphs, in “Sixth Symposium on Graph Drawing, 1998.”
20. C. A. Duncan, M. T. Goodrich, and S. G. Kobourov, Balanced aspect ratio trees: Combining the advantages of k -d trees and octrees, in “Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, Jan. 1999.”
21. A. Efrat and M. Sharir, On the complexity of the union of fat objects in the plane, in “Proc. 13th Annu. ACM Sympos. Comput. Geom., 1997,” pp. 104–112.
22. R. A. Finkel and J. L. Bentley, Quad trees: a data structure for retrieval on composite keys, *Acta Inform.* **4** (1974), 1–9.
23. S. Fortune, Voronoi diagrams and Delaunay triangulations, in “Handbook of Discrete and Computational Geometry” (J. E. Goodman and J. O’Rourke, Eds.), chap. 20, pp. 377–388, CRC Press, Boca Raton, FL, 1997.
24. H. Fuchs, Z. M. Kedem, and B. Naylor, On visible surface generation by a priori tree structures, in “Proc. SIGGRAPH ’80,” *Comput. Graph* **14** (1980), 124–133.
25. A. Henrich, Improving the performance of multi-dimensional access structures based on kd-trees, in “Proc. 12th IEEE Intl. Conf. on Data Engineering, 1996,” pp. 68–74.

26. C. Jackins and S. L. Tanimoto, Oct-trees and their use in representing 3-d objects, *Comput. Graph. Image Process.* **14** (1980), 249–270.
27. M. J. Katz, 3-D vertical ray shooting and 2-D point enclosure, range searching, and arc shooting amidst convex fat objects, *Comput. Geom. Theory Appl.* **8** (1997), 299–316.
28. J. Matoušek, “Geometric Range Searching,” Tech. Report B-93-09, Fachbereich Mathematik und Informatik, Free Univ. Berlin, 1993.
29. J. Matoušek, Geometric range searching, *ACM Comput. Surv.* **26** (1994), 421–461.
30. J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl, Fat triangles determine linearly many holes, *SIAM J. Comput.* **23** (1994), 154–169.
31. M. H. Overmars, Efficient data structures for range searching on a grid, *J. Algorithms* **9** (1988), 254–275.
32. M. H. Overmars and A. F. van der Stappen, Range searching and point location among fat objects, *J. Algorithms* **21** (1996), 629–656.
33. M. H. Overmars and J. van Leeuwen, Dynamic multi-dimensional data structures based on quad- and k -d trees, *Acta Inform* **17** (1982), 267–285.
34. M. S. Paterson and F. F. Yao, Efficient binary space partitions for hidden-surface removal and solid modeling, *Discrete Comput. Geom.* **5** (1990), 485–503.
35. M. S. Paterson and F. F. Yao, Optimal binary space partitions for orthogonal objects, *J. Algorithms* **13** (1992), 99–113.
36. H. Samet, The quadtree and related hierarchical data structures, *ACM Comput. Surv.* **16** (1984), 187–260.
37. H. Samet, An overview of quadtrees, octrees, and related hierarchical data structures, in “Theoretical Foundations of Computer Graphics and CAD” (R. A. Earnshaw, Ed.), NATO ASI Series F, Vol. 40, pp. 51–68, Springer-Verlag, Berlin, 1988.
38. H. Samet, “The Design and Analysis of Spatial Data Structures,” Addison-Wesley, Reading, MA, 1990.
39. E. Torres, Optimization of the binary space partition algorithm (BSP) for the visualization of dynamic scenes, in “Eurographics ’90,” pp. 507–518, North-Holland, Amsterdam, 1990.