Binary Space Partitions of Orthogonal Subdivisions*

John Hershberger
Mentor Graphics Corp.
8005 SW Boeckman Road
Wilsonville, OR 97070
john_hershberger@mentor.com

Subhash Suri
Computer Science Dept.
University of California
Santa Barbara, CA 93106
suri@cs.ucsb.edu

Csaba D. Tóth
Computer Science Dept.
University of California
Santa Barbara, CA 93106
toth@cs.ucsb.edu

Abstract

We consider the problem of constructing binary space partitions (BSPs) for orthogonal subdivisions (space filling packings of boxes) in d-space. We show that a subdivision with n boxes can be refined into a BSP of size $O(n^{\frac{d+1}{3}})$, for all $d \geq 3$, and that such a partition can be computed in time $O(K \log n)$, where K is the size of the BSP produced. Our upper bound on the BSP size is tight for 3-dimensional subdivisions; in higher dimensions, this is the first nontrivial result for general full-dimensional boxes. We also present a lower bound construction for a subdivision of n boxes in d-space that requires a BSP of size $\Omega(n^{\beta(d)})$, where $\beta(d)$ converges to $(1+\sqrt{5})/2$ as $d \to \infty$.

Categories and Subject Descriptors: F.2.2 [Theory of Computation]: Nonnumerical Algorithms and Problems; G.2.1 [Discrete Mathematics]: Combinatorics

General Terms: Algorithms, Theory

Keywords: Binary Space Partitions, Tilings

1 Introduction

Many algorithms in computational geometry, robotics, and computer graphics decompose a set of objects and the ambient space for efficient processing of certain queries. A binary space partition (BSP) is a popular scheme for constructing such decompositions. Given an open convex region of space containing a set of objects S, a BSP partitions the region and objects with a cutting hyperplane, then recursively partitions the two resulting subproblems. The process stops when each open partition region intersects at most one object of S. See Figure 1 for an example in two dimensions. In the ideal case, the number of regions in the final BSP would be at most n, the number of input objects. In general, however, the recursive partitioning may fracture input objects many times, and the size of the partition can be much larger than n.

Binary space partitions were introduced in the computer graphics community [8, 15] to solve hidden surface removal problems. But they are now used for a wide variety of applications, including set operations in solid modeling, visibility preprocessing for interactive walkthroughs, shadow generation, and cell decomposition methods in robotics, to name only a representative sample [2, 5, 6, 11, 12, 16]. The *size* of a BSP is the total number of pieces the input objects are partitioned into during the BSP, and it is a measure of the *fragmentation* caused by the partition. Because BSPs are often used to decompose large data sets, their size can be crucial to the performance of the underlying applications. Theoretical analyses have therefore focused primarily on BSP algorithms that produce small size partitions.

A theoretical study of BSPs began in earnest with two influential papers by Paterson and Yao in the late 1980s [13, 14]. In two dimensions, Paterson and Yao gave an $O(n)$ size BSP construction for orthogonal (axis-parallel rectangular) objects, and an $O(n \log n)$ size construction for general polygons with a total of n edges. Recently, Tóth [17] has shown an almost-matching lower bound of $\Omega(n \log n / \log\log n)$ for arbitrary polygons in the plane. In three dimensions, Paterson and Yao gave an $O(n^{3/2})$ size BSP construction for orthogonal boxes, and an $O(n^2)$ size construction for arbitrarily oriented polyhedra with a total of n edges. They also gave lower bound constructions matching these upper bounds.

The results of Paterson and Yao have been extended and improved in several important directions, yet the complexity of BSPs in three and higher dimensions is not fully understood. We briefly review the previous research most relevant to our work. De Berg shows that any *uncluttered* scene of n objects in d-space admits a linear size BSP [4]; informally, "uncluttered" means that any cubical region intersecting more than a constant number of objects must include a vertex of one of the objects. For instance, a collection of hypercubes (or, more generally, of *fat* objects) is uncluttered. Such an assumption may be practical in some situations, but it seems quite restrictive in general.

Paterson and Yao [13] consider the complexity of BSPs for 1-dimensional objects (line segments) in d-space, and show that a worst-case set of n axis-aligned line segments requires a BSP of size $\Theta(n^{d/(d-1)})$, for $d > 2$. Dumitrescu, Mitchell, and Sharir [7] give an upper bound of $O(n^{d/(d-k)})$ for the BSP size of disjoint k-dimensional orthogonal objects in d-space, which is asymptotically tight *for* $k < d/2$

*Research by the last two authors was partially supported by National Science Foundation grants CCR-0049093 and IIS-0121562.

only. For axis-aligned boxes, one can obtain an upper bound of $O(n^{d/2})$ by using the result of [7] in conjunction with Paterson-Yao's observation that a set of n full-dimensional boxes in d-space has the same BSP complexity as the set of their $(d-2)$-dimensional faces. Agarwal et al. [1] consider BSPs of 2-dimensional *fat rectangles* in 3-space—each rectangle has sides parallel to the axes and a constant aspect ratio. They show that a collection of n such rectangles admits a BSP of size $n2^{O(\sqrt{\log n})}$. This bound was recently improved to $O(n \log^8 n)$ by Tóth [18].

Our Results

In an attempt to understand the true complexity of d-dimensional BSPs, we consider a natural but restricted setting: *orthogonal subdivisions.* An orthogonal subdivision is a collection of interior-disjoint rectilinear boxes that fill their containing space. (That is, a subdivision is a space-filling packing of boxes.) Our interest in subdivisions is motivated by the observation that all the known lower bound constructions involve intertwined rod-like objects that create many "holes." This raises a natural question: what is the complexity of the BSP for scenes where the complement space can be "tiled" with few boxes, say a linear number? Our main result is the following theorem:

> *Given an orthogonal subdivision with n boxes in d-space, we can construct a BSP for it of size $O(n^{\frac{d+1}{3}})$, for any $d \geq 2$.*

Thus, for collections of orthogonal objects that fill their containing space, we are able to improve the worst-case bound on the BSP size from $O(n^{d/2})$ to $O(n^{\frac{d+1}{3}})$. As a corollary, for any collection of n orthogonal boxes whose complement space can be tiled with m boxes, there exists a BSP of size $O((n+m)^{\frac{d+1}{3}})$. Our subdivision BSP can be constructed in time $O(K \log n)$, where K is the output size. We also exhibit a lower bound construction for a subdivision that requires a BSP of size $\Omega(n^{\beta(d)})$ in $\mathbb{R}^d$, where $\beta(d)$ converges to $(1+\sqrt{5})/2$ as d goes to infinity. The value of $\beta(d)$ is $4/3$ for $d = 3$, and thus our upper bound is tight in 3-space, settling an open question from [9].

Our BSP algorithms are quite simple and therefore easy to implement. They follow a conventional scheme—a round robin partitioning phase followed by efficient BSP construction in the terminal regions. Such a two-phase partitioning framework has been used earlier in several papers [1, 7, 9], but in each case it requires problem-specific insights to find the right stopping rule for the round robin phase as well as an analysis for the terminal case. In our case, we show that "round robin cutting until no interior $(d-3)$-dimensional face remains" is a good stopping rule. One tricky part of analyzing this round robin scheme is that each cut may also *increase* the number of $(d-3)$-faces. The second phase requires efficient binary space partitions for regions that do not contain a $(d-3)$-face. To this end, we prove the following theorem, which may have independent appeal: *Consider a box R in d-space, and a subdivision of R into n boxes such that no $(d-3)$-dimensional face of any box lies in the interior of R. Then there is a BSP of size $O(n)$ for this subdivision.*

2 Geometric Preliminaries

A *d-dimensional box B* is the cross product of d real-valued intervals. Given a box R and a set of boxes $S = \{B_1, \ldots, B_n\}$, we say that S is a *subdivision* of R if each B_i lies in R, the union of the B_i's covers R, and the B_i's have pairwise disjoint interiors. Thus, an orthogonal subdivision of R is a packing by boxes that completely fills R. In this paper, we consider BSPs for d-dimensional box subdivisions only.

A BSP for problem instance (R, S) partitions R with a hyperplane into sub-boxes R_1 and R_2, and recursively solves the problems (R_1, S_1) and (R_2, S_2), where $S_i = S \cap R_i$ is the set of fragments of the input objects contained in R_i, for $i = 1, 2$. The partitioning stops when every subproblem contains at most one object (fragment). A BSP is naturally modeled as a binary tree: the root corresponds to the problem (R, S), and its two children correspond to the subproblems (R_1, S_1) and (R_2, S_2). Every internal node stores the splitting hyperplane for the corresponding problem; the leaf nodes correspond to the regions in the final partition. See Figure 1 for a small example in two dimensions. For ease of reference, we will often call R the *container box* for the problem instance. Thus, R_i is the container box for the ith subproblem, for $i = 1, 2$, produced at the root.

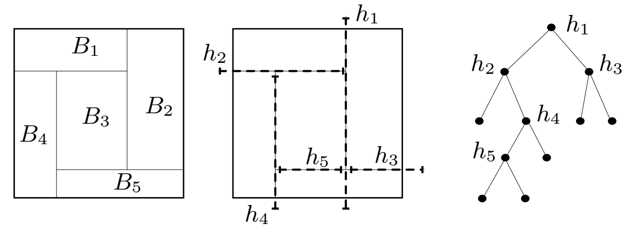

Figure 1: An orthogonal subdivision S (left), a BSP for S (middle), and the corresponding binary tree (right).

Free cuts are important for the construction of small BSPs. A free cut is a hyperplane that separates the object set without splitting any object. When a subproblem (R, S) has a free cut, it is always worth splitting it with the free cut, since it does necessary work—separating objects that must be separated by the BSP—without increasing the complexity of the subproblems (and hence the final BSP size). In Figure 1, the cuts along h_2, h_3, h_4, and h_5 are free cuts.

A d-dimensional box B has 2^d vertices, $d2^{d-1}$ edges, and $2d$ facets. B also has many faces of intermediate dimensions j, for $2 \leq j \leq d-2$. A *k-face* of B is a k-dimensional box, and it can be characterized as follows: a k-face of B is obtained from the cross-product B by fixing $(d-k)$ coordinates at one of the two extremes of the corresponding intervals; the remaining k coordinates maintain the same extent as B. Thus, vertices are the 0-faces, edges are the 1-faces, and facets are the $(d-1)$-faces of B. It is easy to see that every k-face of the box B, for $0 \leq k < d$, lies on the boundary of B. Our algorithm exploits the structure of boxes whose extents in certain dimensions match those of the container box. We therefore introduce the notions of *pass-through* and *k-rod.*[1]

Consider a container box R, and a box B in a subdivision of R. We say that B is *pass-through* for R in dimension j if the extent of B in dimension j equals that of R. We say that box B is a *k-rod* in R if B is pass-through in exactly k dimensions. If B is a k-rod in R, then its *orientation* $\sigma(B)$

[1]The notion of pass-through also appears in [7]. Our analysis, however, requires a more detailed investigation into the properties of rods.

is the set of the k dimensions in which B is pass-through for R. Whenever the container box R is clear from the context, we will simply say that B is a k-rod, without mentioning R. Figure 2 illustrates k-rods in three dimensions, for $k = 1, 2$.

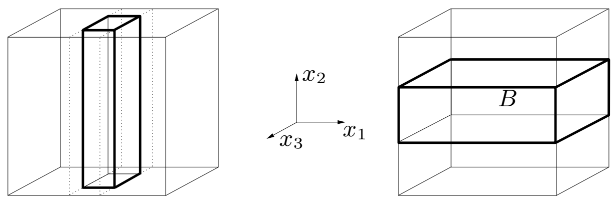

Figure 2: A 1-rod of orientation $\{x_2\}$ on the left and a 2-rod of orientation $\{x_1, x_3\}$ on the right in $\mathbb{R}^3$.

Suppose B is a rod in a box subdivision of some container R. If B is a d-rod, then, of course, B fills up R, and it is the only box in R. If B is a $(d-1)$-rod, with orientation $\sigma(B) = \{x_1, x_2, \ldots, x_{d-1}\}$, then at least one of the two facets of B orthogonal to the x_d-axis intersects the interior of R, separating it into two parts. A cut along this facet is a free cut for the subdivision of R. For instance, in the right half of Figure 2, the box B is a 2-rod in 3-space, and there are free cuts along its top and bottom faces. We summarize this fact for future reference.

Lemma 2.1. *If a box subdivision contains a $(d-1)$-rod, then there is a free cut along one of the facets of that rod.*

We next establish an elementary but useful characterization of k-rods: a box B is a k-rod in a container R *if and only if* no j-face of B lies in the interior of R, for all $j < k$. For instance, in Figure 2 (right), B is a 2-rod, and all of its vertices and edges (0- and 1-faces) lie on the boundary of the container box.

Lemma 2.2. *Suppose B is a box in the subdivision of R. Then B is a k-rod if and only if no j-face of B intersects the interior of R, for exactly those $j < k$.*

Proof. Consider a k-rod B, and let F be one of its j-faces, where $j < k$. By definition, F has a fixed coordinate in $(d-j)$ dimensions, and the same extent as B in the remaining j dimensions. Because B is a k-rod, it is pass-through for R in at least k dimensions, and $k \geq j+1$. Thus, at least one of the fixed coordinates of F is at one of the extremes of a pass-through dimension of B, and so F lies on the boundary of R. Now consider $j \geq k$. Choose F to be a j-face whose varying coordinates include all the pass-through directions of the k-rod B, and whose fixed coordinates are chosen inside the corresponding intervals of the cross product of the container box R (possible because B is pass-through in none of those $d - j \leq d - k$ dimensions). By construction, this j-face F intersects the interior of R. $\square$

3 An Upper Bound in $\mathbb{R}^3$

Our first result is an optimal binary space partition for 3-dimensional subdivisions. Our BSP has worst-case size $O(n^{4/3})$, which is optimal because a modified Paterson-Yao construction gives a matching lower bound for orthogonal subdivisions [9]. The 3-dimensional bound is central to our main result, because our algorithm for constructing the d-dimensional BSP uses projection of the d-dimensional subdivision onto an appropriate 3-space (after a suitable number of round-robin cuts).

Our algorithm (Algorithm 3-BSP) is a round robin partitioning scheme that iteratively slices the subdivision by planes passing through interior box vertices, producing a collection of smaller subdivisions, each of which ultimately contains only j-rods, for $j \geq 1$. (A similar round robin scheme is also used by Murali [10] in his construction of BSPs for fat axis-aligned rectangles in $\mathbb{R}^3$. Interestingly, while Murali also achieves an $O(n^{4/3})$ size BSP, that complexity is suboptimal for his problem. In our case, the round robin algorithm produces an optimal BSP.) The partition tree T created by the algorithm 3-BSP is *not* a BSP—the cells output by the algorithm may contain multiple boxes. These terminal regions, however, do not contain any vertices of the subdivision. We can refine each such box into a proper BSP with only a constant factor increase in complexity, and append the tree associated with this refinement to each leaf of T. Therefore, the key to efficiency is bounding the total complexity of all the regions produced by the slicing procedure.

The algorithm of Hershberger and Suri [9] follows the same general framework, but that algorithm chooses the cutting planes by *local optimality*: each cut minimizes the number of boxes partitioned at each step. However, this seemingly simple rule leads to complex analysis and a suboptimal ($O(n^{1.41504} \log^2 n)$) upper bound. The algorithm described below is not locally optimal, but its regularity and simplicity enable an analysis that (surprisingly) gives a tight upper bound.

Suppose we have a container box R and its subdivision $S = \{B_1, B_2, \ldots, B_n\}$ into n boxes, so that there are m vertices in the interior of R; clearly, $m \leq 8n$. Our partitioning scheme cuts the subdivision using planes normal to each of the three axes in turn, cycling through the three possible orientations in a round robin fashion. In the following pseudo-code description of our algorithm, we use the term *median x_j plane* of a point set to denote the plane normal to the x_j-axis and passing through the median x_j coordinate of the point set.

Algorithm 3-BSP

- Input is the box subdivision $S = \{B_1, B_2, \ldots, B_n\}$ of a container box R.
- Initialize $i = 0$, and $\mathcal{C}_0 = \{R\}$.
- While there is a container box $C \in \mathcal{C}_i$ with a subdivision vertex in its interior, do
  1. For each container box $C \in \mathcal{C}_i$ with at least one vertex of the subdivision in its interior, split C by the median x_p plane of the vertices in the interior of C, where $p = 1 + (i \bmod 3)$.
  2. Apply all possible free cuts.
  3. Let $\mathcal{C}_{i+1}$ be the set of all container boxes resulting from these cuts, and set $i := i + 1$.
- Return $\mathcal{C}_i$.

Using an argument similar to one of Paterson and Yao [14], we derive an upper bound on the total number of box fragments generated by our algorithm by considering how many times the edges of the original subdivision S are cut.

LEMMA 3.1. *Consider the partition tree T generated by the algorithm 3-BSP. At depth i in T, there are $O(n2^{i/3})$ fragments of the original edges of S.*

PROOF. Each edge is parallel to one of the three directions x_1, x_2, and x_3. It can be cut only by planes orthogonal to its direction. Since our splitting planes cycle through the three directions, an edge can be cut at every third level of T. Thus, the number of fragments of a given original edge at most doubles at every third level. Since the number of edges in the subdivision at the root of T is $O(n)$, the total number of edge fragments at depth i is $O(n2^{i/3})$. □

LEMMA 3.2. *Suppose S is a 3-dimensional box subdivision with n boxes, and $m = O(n)$ vertices lie in the interior of the container box. Then the partition created by the algorithm 3-BSP produces $O(nm^{1/3}) = O(n^{4/3})$ fragments of the input boxes.*

PROOF. We analyze the algorithm in *rounds* of three consecutive steps: round j corresponds to the steps $i = 3j, 3j+1, 3j+2$. In one round, cuts are made in all three directions, and a box of a subproblem can be split into at most eight pieces.

We observe that if a box is split in a subproblem, then at least one edge of the box must be interior to the container box for this subproblem. This is because only a 3-rod or a 2-rod can have all its edges on the boundary of the container box (Lemma 2.2), but no 3-rods are ever split and all 2-rods are eliminated in step 2 by free cuts. Lemma 3.1 implies that $O(n2^j)$ box fragments can be further partitioned at round j. Since there is no vertex in the interior of any container cell after $\lceil \log m \rceil$ steps, the algorithm terminates in $\lceil \frac{1}{3} \log m \rceil$ rounds. The number of box fragments produced is

$$O\left(n \cdot \sum_{j \le \lceil \frac{1}{3} \log m \rceil} 2^j\right) = O\left(n \cdot 2^{(\log m)/3}\right) = $$
$$= O\left(nm^{1/3}\right) = O\left(n^{4/3}\right). \quad \square$$

All that remains now is to show that the terminal cells output by algorithm 3-BSP can be refined into linear size BSPs. This was already shown by Hershberger and Suri [9]:

LEMMA 3.3. *Consider a subdivision of a 3-dimensional container box R into a set of boxes $S = \{B_1, B_2, \dots, B_n\}$. If none of the vertices of any box B_i is in the interior of R, then there is a BSP of size $O(n)$ for S.*

Thus, combining the previous two lemmas, we get the main result of this section.

THEOREM 3.4. *A 3-dimensional box subdivision with n boxes and m interior vertices has a BSP of size $O(nm^{1/3}) = O(n^{4/3})$. We can construct such a BSP in time $O(K \log n)$, where K is the size of the final BSP.*

The implementation that achieves the $O(K \log n)$ construction time is quite simple. We maintain all the extents of all the boxes in all containers. It is easy to determine from the extents which boxes are split into two, where the median planes lie, and which faces are free cuts. If we sort the extents initially in all dimensions, we can find medians trivially in linear time, and can create sorted extent lists for the child containers in the same time. We spend $O(n \log n)$ time for the initial sorting and $O(K)$ time for each of the $O(\log n)$ steps. Because $K \ge n$, this gives a running time of $O(K \log n)$.

The upper bound $O(n^{4/3})$ in Theorem 3.4 is tight in the worst case: Hershberger and Suri [9] exhibit a 3-dimensional subdivision requiring a BSP of size $\Omega(n^{4/3})$. In Section 6 we give a more general construction that establishes lower bounds on the size of BSPs in all dimensions.

4 The Structure of Rods in $\mathbb{R}^d$

The algorithm of [9], with its complex analysis, seems quite difficult to extend to higher dimensions. By contrast, the simplification achieved in Section 3 makes a higher-dimensional generalization much easier. In order to extend our 3-dimensional BSP algorithm to higher dimensions, we need two key ingredients: (1) a suitable stopping rule for our round robin algorithm, that is, one that does not result in too much fragmentation, and (2) a linear-size BSP construction for the cells obtained when the round robin phase ends. In three dimensions, the correct stopping rule was the elimination of interior vertices. In d dimensions, we show that the *elimination of interior $(d-3)$-dimensional faces* is a good stopping rule. The analysis of the total fragmentation caused by this stopping rule as well as the construction of BSPs in terminal cells requires a better understanding of higher-dimensional geometry. In this section, we establish two key facts about the d-dimensional subdivisions that are central to our analysis.

LEMMA 4.1. *Consider two boxes B_1, B_2 that are rods for their container box R in d-space. Let $D_1, D_2 \subset \{1, 2, \dots, d\}$ be the sets of dimensions in which B_1 and B_2, respectively, are pass-through. If $D_1 \cup D_2 = \{1, 2, \dots, d\}$, then B_1 and B_2 have intersecting interiors.*

PROOF. Let ℓ_i be the extent of dimension x_i *common* to B_1 and B_2. Since in each dimension, at least one of the rods is pass-through, ℓ_i is non-empty for every i. Thus a full-dimensional box defined by the cross-product $\ell_1 \times \cdots \times \ell_d$ lies in the common interior of B_1 and B_2. □

This lemma implies that, for any two boxes B_i, B_j in a subdivision, there is at least one direction in which neither is a rod. With this basic property, we next classify the set system of all possible orientations of $(d-2)$-rods in a box subdivision. The $(d-2)$-rods are the rods especially important to us because our ultimate goal is to consider subdivisions without any interior $(d-3)$-faces. Recall that a set system is a pair $(X, \mathcal{F})$, where X is a ground set, and $\mathcal{F}$ is a family of subsets of X. We consider the set system $(D, \mathcal{F})$, where $D = \{1, 2, \dots, d\}$, and $|F| = d-2$ for all $F \in \mathcal{F}$. We are interested in the following two configurations:

- **Cycle Configuration:** $\mathcal{F}$ is a *cycle configuration* if $\mathcal{F}$ has at least 3 sets and there is some $x \in D$ that is not an element of any set of $\mathcal{F}$.
- **Star Configuration:** $\mathcal{F}$ is a *star configuration* if $\mathcal{F}$ has at most three sets and they all share a common $(d-3)$ element subset of D.

Lemma 4.2. *Consider a set system* $(D, \mathcal{F})$, *with* $D = \{1, 2, \ldots, d\}$ *and* $|F| = d - 2$ *for every* $F \in \mathcal{F}$. *If every two sets* $F_1, F_2 \in \mathcal{F}$ *have* $F_1 \cup F_2 \neq D$, *then* $\mathcal{F}$ *is either a star or a cycle configuration.*

Proof. If $|\mathcal{F}| \leq 2$, then $\mathcal{F}$ is trivially a star configuration. So let us assume that $|\mathcal{F}| \geq 3$, and let F_1, F_2 be two distinct sets in $\mathcal{F}$. Since $F_1 \neq F_2$ and $F_1 \cup F_2 \neq D$, the union $F_1 \cup F_2$ has exactly $d-1$ elements and $F_1 \cap F_2$ has $d-3$ elements. Let $a = F_1 \setminus F_2$, $b = F_2 \setminus F_1$, and $c = D \setminus (F_1 \cup F_2)$.

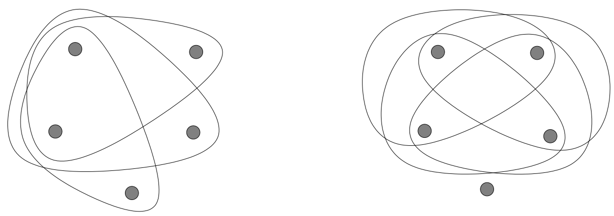

Figure 3: A star configuration (left) and a cycle configuration (right) on a five-element set.

First suppose that there is a set $F \in \mathcal{F}$ such that $c \in F$. We observe that F contains neither a nor b: otherwise, we get $F \cup F_2 = D$ or $F \cup F_1 = D$. Since $|F| = d - 2$, we must have $F = \{c\} \cup (F_1 \cap F_2)$. The rôles of F_1, F_2, and F are symmetric, so the sets containing a, b, and c are uniquely determined. Thus, $\mathcal{F} = \{F_1, F_2, F\}$, and $\mathcal{F}$ is a star configuration.

On the other hand, if there is no set in $\mathcal{F}$ that contains c, then $\mathcal{F}$ has at least three and at most $\binom{d-1}{d-2} = d - 1$ subsets of $D \setminus \{c\}$ of size $(d-2)$. In this case, $\mathcal{F}$ is a cycle configuration. □

A 4-dimensional version of our star-cycle property was observed by Dumitrescu, Mitchell, and Sharir [7]. It is somewhat surprising that it extends to arbitrary dimensions. This lemma turns out to be a critical technical piece in constructing linear size BSPs in higher dimensions whenever the subdivision is free of $(d-3)$-dimensional faces.

5 An Upper Bound in $\mathbb{R}^d$

We begin by showing that if a d-dimensional subdivision is free of all interior $(d-3)$-faces, then it admits a linear size BSP. This fact ensures that if we use the stopping rule "recurse until no interior $(d-3)$-dimensional faces remain," the second phase of the algorithm incurs only a linear additional fragmentation.

Lemma 5.1. *Consider a set of boxes* $S = \{B_1, B_2, \ldots, B_n\}$ *that forms a subdivision of a* d*-dimensional container box* R. *If none of the* $(d-3)$*-faces of any box* B_i *is in the interior of* R, *then there is a BSP of size* $O(n)$ *for* S.

Proof. Our proof is by induction on n. The base case is $n = 1$, for which there is nothing to prove. If any of the boxes of S is a $(d-1)$-rod, then we can make a free cut along a facet of this box by Lemma 2.1; each of the subproblems on either side of the cut contains strictly fewer than n boxes, so our lemma follows by induction. Lemma 2.2 tells us that all boxes of S are $(d-2)$-rods. By Lemmas 4.1 and 4.2, the orientations of these $(d-2)$-rods form either a star or a cycle configuration. We handle these two cases separately.

First, suppose that the orientations of the rods form a cycle configuration. Without loss of generality, let us assume that none of the $(d-2)$-rods is pass-through in direction x_d. We will show that there is a free cut orthogonal to the x_d-axis, and the rest of the proof then follows by induction. We observe that all the $(d-2)$-rods properly intersecting a hyperplane H orthogonal to x_d must have the same orientation; otherwise Lemma 4.1 is violated for the $(d-1)$-dimensional subdivision on H. Now consider a point p on the common boundary of two $(d-2)$-rods of different orientations, and let H be a hyperplane passing through p and orthogonal to x_d. Any (closed) box of S intersecting H lies on one side of H because there are $(d-2)$-rods of different orientations on the two sides of H. Every box intersecting H has a facet along H, and therefore H is a free cut.

Next, suppose that the orientations of the rods form a star configuration. Without loss of generality, assume that all boxes are pass-through in $d-3$ dimensions $x_4, x_5, \ldots, x_d$. We project the boxes of S onto the subspace spanned by the remaining three axes, using the map $\pi : (x_1, x_2, \ldots, x_d) \to (x_1, x_2, x_3)$. Any BSP for the projection corresponds to a BSP of equal size for S, since the pre-image of a plane h in the projection is a hyperplane H in $\mathbb{R}^d$, and H cuts only those $(d-2)$-rods that are the pre-images of the 3-dimensional boxes cut by h. Observe that the subdivision $\pi(S)$ does not have any interior vertices (interior in $\mathbb{R}^3$), because an interior vertex v corresponds to an interior $(d-3)$-face in S, namely, $\pi^{-1}(v)$. By Lemma 3.3, there is a linear size BSP for the projected subdivision, which when lifted to $\mathbb{R}^d$ gives a linear size BSP for S. □

We are now ready to describe our BSP algorithm for d-dimensional subdivisions. In three dimensions, our round robin cuts were made along planes that evenly partition the set of interior vertices. In d dimensions, we choose hyperplanes that "evenly split" the interior $(d-3)$-dimensional faces. Unfortunately, these median hyperplanes can also introduce new $(d-3)$-dimensional faces by splitting some old ones. For instance, in 4-space, a cutting hyperplane can split many 1-faces (line segments). Our fragmentation lemma (Lemma 5.2) below will bound the total number of box fragments created in the round robin phase. We need to introduce a bit of notation first to discuss the objects used in the algorithm.

Given a d-dimensional box B, let $f_k(B)$ denote the set of k-faces of B, where $0 \leq k \leq d$. We use the notation $f_k(S)$ to refer to the (multi) set of k-faces in all the boxes of $S = \{B_1, B_2, \ldots, B_n\}$; that is, $f_k(S)$ contains multiple copies of a face if that face is incident to multiple boxes. Finally, given a container box R, we use the notation $f_k(R, S)$ to denote the (fragments of) k-faces of $f_k(S)$ that lie in the *interior* of R. In our d-dimensional BSP scheme, the elements of the set $f_{d-3}(R, S)$ will play the rôle that vertices played in the algorithm 3-BSP.

Algorithm d-BSP

- Input is a box subdivision $S = \{B_1, B_2, \ldots, B_n\}$ of a container box R in d-dimensions.
- Initialize $j = 0$, and $\mathcal{C}_0 = \{R\}$.
- While there is a container box $C \in \mathcal{C}_j$ with non-empty $f_{d-3}(C, S)$, do
  1. For each container box $C \in \mathcal{C}_j$ with non-empty $f_{d-3}(C, S)$, do

(a) Set $\mathcal{D}_1 = \{C\}$ and let V be the multiset of vertices of all faces in $f_{d-3}(C, S)$.
(b) For $i = 1$ to d, do
  i. Split every $D \in \mathcal{D}_i$ by the median x_i hyperplane of the multiset $V \cap D$.
  ii. Remove the vertices on the splitting hyperplane from V.
  iii. Apply all possible free cuts.
  iv. Let $\mathcal{D}_{i+1}$ be the set of all container boxes resulting from these cuts.

2. Let $\mathcal{C}_{j+1}$ be the set of all container boxes resulting from these cuts, and set $j := j + 1$.

- Return $\mathcal{C}_j$.

The algorithm d-BSP constructs a partition tree T, whose leaves correspond to regions that do not contain any fragment of a $(d-3)$-face of the subdivision S. By Lemma 5.1, each of these regions can be refined into a BSP of size linear in the number of box fragments contained in it. Thus, it suffices to bound the total complexity of all the box fragments produced by the algorithm d-BSP.

Lemma 5.2. *Given a d-dimensional box subdivision with n boxes, the algorithm d-BSP produces a partition containing $O(n^{\frac{d+1}{3}})$ box fragments.*

Proof. Let us define $m = |f_{d-3}(R, S)|$. We call a *round* of the algorithm the work done between increments of j. We show that Algorithm d-BSP terminates in $\lceil(\log m)/3\rceil$ rounds, and then we bound the number of box fragments produced in $\lceil(\log m)/3\rceil$ rounds.

During a round, every $(d-3)$-face F of a subproblem (C, S) can be cut into at most 2^{d-3} fragments—it is cut at most once for each of the $(d-3)$ directions orthogonal to F. We focus on the fragments of F that are not contained in any splitting hyperplane of the round. We argue that at least one vertex of every fragment of F is a vertex of F that does not lie on any of the splitting hyperplanes of the round. A fragment is itself a $(d-3)$-face, and so it is the cross product of 3 fixed coordinates and $d-3$ nontrivial intervals. Each nontrivial interval contains at least one of the two endpoints of the corresponding original interval of F (because there is at most one cut orthogonal to the interval's direction). The cross product of the 3 fixed coordinates and the $d-3$ original interval endpoints is a vertex of F but does not lie on any splitting hyperplane of the round.

The container box C is divided into sub-boxes during the round, each containing at most $2^{d-3} \cdot |f_{d-3}(C, S)|/2^d = |f_{d-3}(C, S)|/8$ vertices of V. As noted, each face fragment produced by the round is incident to a vertex from V, and so $|f_{d-3}(C', S)| \le |f_{d-3}(C, S)|/8$ for each sub-container C' produced. By induction, the number of $(d-3)$-faces in the interior of a container box at the beginning of round j is at most $m/8^j$. Therefore, after $\lceil\log_8 m\rceil = \lceil(\log m)/3\rceil$ rounds, no $(d-3)$-face intersects the interior of any subproblem's container box, and the algorithm terminates.

Now consider a $(d-2)$-dimensional face of a box B. During a round of the algorithm, it can be cut recursively in $d-2$ steps. Thus, the total number of fragments of $(d-2)$-faces of $f_{d-2}(R, S)$ at round j is $O(n2^{(d-2)j})$.

Finally, in each round j when a box fragment B in some subproblem is split, the container box for the subproblem must have a fragment of a $(d-2)$-face of B in its interior. This is because only a d-rod or a $(d-1)$-rod can have all its $(d-2)$-faces on the boundary of the container by Lemma 2.2. But a d-rod is never split and step 1(b)iii eliminates all $(d-1)$-rods. This implies that $O(n2^{(d-2)j})$ box fragments can be further split in round j. The algorithm terminates in $\lceil(\log m)/3\rceil$ rounds, and so the number of box fragments produced by the algorithm is

$$O\left(n \cdot \sum_{j \le \lceil \log m/3 \rceil} 2^{(d-2)j}\right) = O\left(n \cdot 2^{(d-2)(\log m)/3}\right) = $$
$$= O\left(nm^{(d-2)/3}\right) = O\left(n^{(d+1)/3}\right). \quad \Box$$

Since m is at most $2^3\binom{d}{3} \cdot n$, our bound in terms of n and d would be $O(d^{d+1} \cdot n^{(d+1)/3})$.

Finally, this result, combined with Lemma 5.1 and standard implementation techniques similar to those used for Algorithm 3-BSP, yields our main theorem.

Theorem 5.3. *A d-dimensional box subdivision of n boxes has a BSP of size $O(n^{\frac{d+1}{3}})$. The BSP can be constructed in time $O(K \log n)$, where K is the size of the BSP.*

A simple but practical corollary of this theorem is the following: Any collection of n disjoint boxes in d-space whose complement space can be tiled with m additional boxes admits a BSP of size $O((n+m)^{\frac{d+1}{3}})$.

6 Lower Bounds

We now describe a construction establishing a nontrivial lower bound on the BSP size in any dimension $d \ge 3$. Paterson and Yao [13, 14] gave a construction establishing an $\Omega(n^{3/2})$ lower bound on 3-dimensional orthogonal box sets. However, their construction does not say anything meaningful for subdivisions, because converting their construction into a subdivision requires $\Omega(n^{3/2})$ additional boxes. Hershberger and Suri [9] modified the Paterson-Yao construction to get an $\Omega(n^{4/3})$ lower bound for 3-dimensional subdivisions. Our construction is similar in spirit to these constructions, but more involved. Additionally, our construction is recursive—we use the lower bound construction in $d-1$ dimensions to build the subdivision in d dimensions. We begin with an informal description, then formally describe the construction in four dimensions. The construction in d dimensions is a straightforward extension of this construction.

We partition the container box in a grid-like fashion into disjoint regions that we call *junctions*, and give a lower bound for the BSP complexity for the fragments of objects within each region. Then we bound the complexity of a BSP for all objects by the sum of complexities over all the regions.

We have $d-1$ families of congruent $(d-2)$-rods whose orientations form a cycle configuration, and a family of 1-rods whose orientation is the complement of the orientations of the $(d-2)$-rods. The rods assure our complexity bound in each region. (Intuitively, it is efficient to use rods of larger orientations, because a k-rod of the container box is a k-rod in each junction it intersects.) On the other hand, rods of such orientations cannot fill the container box, and therefore we need to add a number of filler boxes to obtain a box subdivision.

In order to balance the number of rods and the number of filler boxes, we allocate the rods in *bundles*, so that the union of a bundle of rods is again a rod with the same orientation. The $(d-2)$-rods in a bundle are all congruent; the $\mathbb{R}^{d-1}$-projection of a bundle of 1-rods, however, is the worst-case box subdivision we obtain in $\mathbb{R}^{d-1}$.

We prove our lower bound for $\mathbb{R}^4$ in detail. The construction then generalizes to any dimension $d > 3$.

THEOREM 6.1. *There is a four-dimensional box subdivision with n boxes such that the size of any BSP is $\Omega(n^{13/9})$.*

PROOF. First, we describe an auxiliary construction of $\Theta(\ell^{9/2})$ boxes such that all vertices of the boxes lie on a $2\ell \times 2\ell \times 2\ell \times 3\ell^{3/2}$ grid. The rods of the auxiliary subdivision correspond to the bundles of rods in the final subdivision.

We have ℓ^3 long 1-rods that are pass-through in x_4:

$$R_1 = \{r_1(i,j,k) : i,j,k = 0,\ldots,\ell-1\}, \text{ where}$$
$$r_1(i,j,k) = [2i, 2i+1] \times [2j, 2j+1] \times [2k, 2k+1] \times [0, 3\ell^{3/2}].$$

We also have three families of boxes, each containing $\ell^{5/2}$ 2-rods with orientations $\{x_1, x_2\}$, $\{x_1, x_3\}$, and $\{x_2, x_3\}$, respectively:

$$R_{1,2} = \{r_{1,2}(k,t) : 0 \le k \le \ell-1;\ 0 \le t \le \ell^{3/2}-1\},$$
$$r_{1,2}(k,t) = [0, 2\ell] \times [0, 2\ell] \times [2k+1, 2k+2] \times [3t, 3t+1];$$

$$R_{1,3} = \{r_{1,3}(j,t) : 0 \le j \le \ell-1;\ 0 \le t \le \ell^{3/2}-1\},$$
$$r_{1,3}(j,t) = [0, 2\ell] \times [2j+1, 2j+2] \times [0, 2\ell] \times [3t+1, 3t+2];$$

$$R_{2,3} = \{r_{2,3}(i,t) : 0 \le i \le \ell-1;\ 0 \le t \le \ell^{3/2}-1\},$$
$$r_{2,3}(i,t) = [2i+1, 2i+2] \times [0, 2\ell] \times [0, 2\ell] \times [3t+2, 3t+3].$$

Figure 4 illustrates the intersection of our auxiliary construction with three (3-dimensional) hyperplanes orthogonal to x_4. The intersection of the 1-rods in R_1 with every hyperplane consists of ℓ^3 unit cubes in a grid. The 2-rods of three distinct orientations appear in three different slices. In each slice, congruent 2-rods fill ℓ slots between the grid of unit cubes.

Since all the rods have pairwise disjoint interiors, we can obtain a rectilinear subdivision of the container box $R = [0, 2\ell] \times [0, 2\ell] \times [0, 2\ell] \times [0, 3\ell^{3/2}]$ by filling up the space between the rods with $9\ell^{9/2}$ unit cubes. This completes the description of our auxiliary construction of $\Theta(\ell^{9/2})$ boxes. We obtain a subdivision S of $n = \Theta(\ell^{9/2})$ boxes in the following way: partition each 2-rod into ℓ^2 congruent rods with the same orientation, and replace each 1-rod by $\ell^{3/2}$ pairwise disjoint 1-rods whose projection to the (x_2, x_3, x_4) hyperplane is the worst-case construction known for a subdivision in $\mathbb{R}^3$.

Consider the $\ell^{9/2}$ boxes $\Pi(i,j,k,t) = [2i, 2i+2] \times [2j, 2j+2] \times [2k, 2k+2] \times [3t, 3t+3]$, where $i,j,k = 0,1,\ldots,\ell-1$ and $t = 0,1,\ldots,\ell^{3/2}-1$. These are the *junction* regions. Note that the junctions have pairwise disjoint interiors. We show that any rectilinear BSP for S dissects the rods of S such that pieces of rods have at least ℓ^2 vertices in the interior of every junction. This implies that the total number of pieces is $\Omega(\ell^{9/2} \cdot \ell^2) = \Omega(\ell^{13/2}) = \Omega(n^{13/9})$.

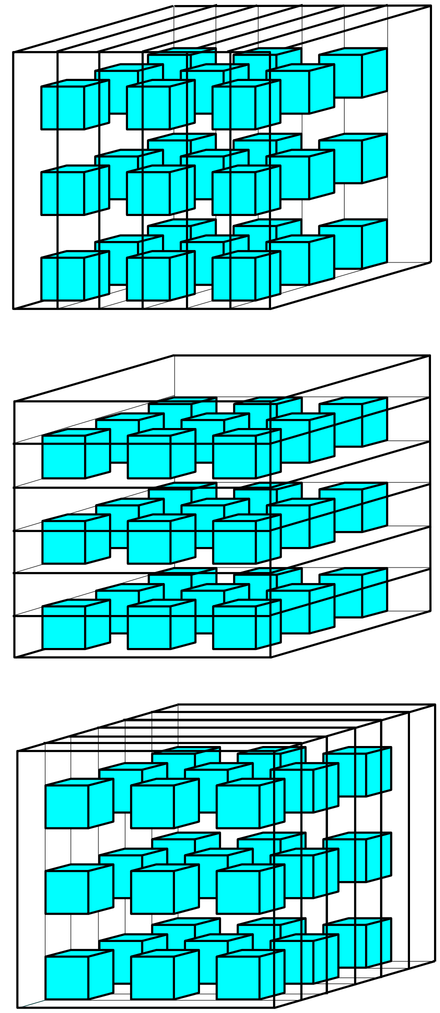

Figure 4: Three different 3-dimensional slices of the subdivision along three hyperplanes orthogonal to x_4.

First suppose that each of the $\ell^{3/2}$ 1-rods in $r_1(i,j,k) \in R_1$ is cut by some hyperplane $x_4 = c$, for $c \in (3t, 3t+3)$. (Not all 1-rods might be cut by the same hyperplane, though.) Consider an intersection of the bundle of 1-rods with a hyperplane $x_4 = c_0$ such that each 1-rod has a vertex in the interior of $\Pi(i,j,k,t)$ below $x_4 = c_0$. A BSP for S, restricted to the intersection of the bundle with $x_4 = c_0$, must be a BSP for the 3-dimensional subdivision to which the bundle projects. Hence the intersection BSP must have $\Omega((\ell^{3/2})^{4/3}) = \Omega(\ell^2)$ vertices, by the lower bound of Hershberger and Suri [9]. Each vertex corresponds to a vertex in the interior of $\Pi(i,j,k,t)$—in fact, it is connected to its corresponding vertex by an edge of the BSP parallel to the x_4-axis.

Otherwise let r be a rod in $R_1(i,j,k)$ that is not cut by any hyperplane $x_4 = c$, for $c \in (3t, 3t+3)$. Let e be a segment in the interior of r such that e is pass-through in x_4 within $\Pi(i,j,k,t)$ and does not lie on any hyperplane of the BSP. Since e is not cut by the BSP, one subproblem contains e all through the partitioning procedure. Notice that the container box for e cannot be separated from all the other original boxes by hyperplanes orthogonal to only one axis. Let H_1 and H_2 be hyperplanes orthogonal to, say, x_1 and x_2 on the boundary of the final container box of e. The 2-face $F = H_1 \cap H_2$ is pass-through in x_3 and x_4 and lies in the interior of $\Pi(i, j, k, t)$. Therefore F has a common vertex with all ℓ^2 2-rods in $r_{1,2}(k, t)$ in the interior of $\Pi(i, j, k, t)$. $\square$

Generalizing this recursive construction, we obtain a recursive formula for the exponent of the lower bound $\beta(d)$ we can reach for a box subdivision in $\mathbb{R}^d$.

THEOREM 6.2. *There is a d-dimensional box subdivision S with n boxes such that the size of any BSP for S is $\Omega(n^{\beta(d)})$, where $\beta(3) = 4/3$, $\beta(4) = 13/9$, and $\beta(d)$ converges to $(1 + \sqrt{5})/2$ as $d \to \infty$.*

PROOF. In $\mathbb{R}^d$, our auxiliary construction consists of ℓ^{d-1} 1-rods that are pass-through in x_d, and $d-1$ families of $\ell^{1+\alpha(d)}$ $(d-2)$-rods such that each slice $x_d = c$ intersects ℓ $(d-2)$-rods and there are $\ell^{\alpha(d)}$ levels in the x_d extent ($\alpha(d)$ is a parameter to be specified later). We need $\Theta(\ell^{(d-1)+\alpha(d)})$ unit cubes to convert this configuration into a subdivision.

We balance the number of filler unit cubes and rods by replacing each auxiliary $(d-2)$-rod by a a bundle of ℓ^{d-2} congruent $(d-2)$-rods and each auxiliary 1-rod by a bundle of $\ell^{\alpha(d)}$ 1-rods. We arrange the 1-rods in each bundle such that their $(d-1)$-dimensional projection gives the $\mathbb{R}^{d-1}$ subdivision for which any BSP has size $\Omega((\ell^{\alpha(d)})^{\beta(d-1)})$.

We define $\ell^{(d-1)+\alpha(d)}$ junctions, arranged in a grid. Each junction contains either $\Omega(\ell^{d-2})$ vertices on fragments of $(d-2)$-rods or $\Omega((\ell^{\alpha(d)})^{\beta(d-1)})$ vertices on fragments of 1-rods. To achieve our bound, we set $\alpha(d) = (d-2)/\beta(d-1)$, implying that any BSP for our subdivision has at least $\ell^{(d-1)+\alpha(d)} \cdot \ell^{d-2} = \ell^{2d-3+\alpha(d)}$ vertices. That gives a lower bound of

$$\Omega\left(n^{\frac{2d-3+\alpha(d)}{d-1+\alpha(d)}}\right) = \Omega\left(n^{\frac{2d-3+(d-2)/\beta(d-1)}{d-1+(d-2)/\beta(d-1)}}\right).$$

The exponent in this bound is

$$\begin{aligned} \beta(d) &= \frac{2d-3+(d-2)/\beta(d-1)}{d-1+(d-2)/\beta(d-1)} \\ &= 1 + \frac{d-2}{d-1+(d-2)/\beta(d-1)}. \end{aligned}$$

The first values of this sequence are $\beta(4) = 13/9 = 1.444$, $\beta(5) = 118/79 = 1.494$, $\beta(6) = 689/453 = 1.521$, and $\beta(7) = 9844/6399 = 1.538$. The sequence converges to $\lim_{d\to\infty} \beta(d) = (1 + \sqrt{5})/2 = 1.618$.

Note that the d-dimensional lower bound subsumes the lower bound for three dimensions. If $d = 3$, then $\beta(d-1) = \beta(2) = 1$, and $\alpha(3) = (3-2)/\beta(2) = 1$. The lower bound we obtain is

$$\Omega\left(n^{\frac{2\cdot 3-3+\alpha(3)}{3-1+\alpha(3)}}\right) = \Omega(n^{4/3}),$$

matching the result of Hershberger and Suri [9]. $\square$

7 Concluding Remarks

Binary space partitions are a popular space-decomposition method in computational geometry, computer graphics, and other fields dealing with geometric shapes. However, their worst-case complexity remains poorly understood in dimensions three and higher. Even for the BSP of n orthogonal boxes in $\mathbb{R}^d$, the best upper bound known is $O(n^{d/2})$, while no lower bound better than $\Omega(n^2)$ is known.

In an attempt to understand the complexity of higher-dimensional BSPs, we considered a natural special case: orthogonal subdivisions. We showed that every d-dimensional subdivision with n boxes admits a BSP of size $O(n^{\frac{d+1}{3}})$. We also exhibited subdivisions that require BSPs of size $\Omega(n^{\beta(d)})$ in d dimensions, where $\beta(d)$ converges to $(1+\sqrt{5})/2$ as $d \to \infty$. Our result shows that if the objects do not fracture the complement space too badly, then their BSP size can be significantly smaller than the current worst-case bounds indicate.

Clearly, the most interesting open problem suggested by this paper is to close the gap between the upper and lower bounds. The absence of super-quadratic lower bounds is intriguing. Can it really be the case that in any dimension d, a set of n boxes (forming a subdivision, or not) admits a quadratic size BSP? If not, then what is the true complexity of the BSP of n d-dimensional boxes? Is this bound necessarily worse than that for n boxes forming a subdivision in higher dimensions?

References

[1] P. K. Agarwal, E. F. Grove, T. M. Murali and J. S. Vitter, Binary space partitions for fat rectangles, *SIAM J. of Computing* **29** (2000), 1422–1448.

[2] C. Ballieux, Motion planning using binary space partitions, Tech. Rep. Inf/src/93-25, Utrecht University, 1993.

[3] J. L. Bentley, Multidimensional binary search trees used for associative searching, *Communication of the ACM* **18** (9) (1975), 509–517.

[4] M. de Berg, Linear size binary space partitions for uncluttered scenes, *Algorithmica* **28** (2000), 353–366.

[5] N. Chin and S. Feiner, Near real-time shadow generation using BSP trees, in: *Proc. SIGGRAPH, 1989*, vol. 23, ACM SIGGRAPH, pp. 99–106.

[6] N. Chin and S. Feiner, Fast object-precision shadow generation for areal light sources using BSP trees, *Comput. Graph.* **25** (1992), 21–30, Proc. 1992 Sympos. Interactive 3D Graphics.

[7] A. Dumitrescu, J. S. B. Mitchell, and M. Sharir, Binary space partitions for axis-parallel segments, rectangles, and hyperrectangles, *Discrete Comput, Geom.* **31** (2) (2004), 207–227.

[8] H. Fuchs, Z. M. Kedem, and B. Naylor, On visible surface generation by a priori tree structures, *Comput. Graph.* **14** (3) (1980), 124–133, Proc. SIGGRAPH '80.

[9] J. Hershberger and S. Suri, Binary space partitions for 3D subdivisions, in: *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, ACM Press, 2003, pp. 100–108.

[10] T. M. Murali, Efficient hidden-surface removal in theory and in practice, Ph. D. Thesis, Department of Computer Science, Brown University, Providence, 1998, pp. 80-81.

[11] B. Naylor, J. A. Amanatides, and W. Thibault, Merging BSP trees yields polyhedral set operations, *Comput. Graph.* **24** (4) (1990), 115–124, Proc. SIGGRAPH '90.

[12] B. Naylor and W. Thibault, Application of BSP trees to ray-tracing and CSG evaluation, Technical Report GIT-ICS 86/03, Georgia Institute of Tech., School of Information and Computer Science, 1986.

[13] 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.

[14] M. S. Paterson and F. F. Yao, Optimal binary space partitions for orthogonal objects, *J. Algorithms* **13** (1992), 99–113.

[15] R. A. Schumacker, R. Brand, M. Gilliland, and W. Sharp, Study for applying computer-generated images to visual simulation, Tech. Rep. AFHRL–TR–69–14, U.S. Air Force Human Resources Laboratory, 1969.

[16] W. C. Thibault and B. F. Naylor, Set operations on polyhedra using binary space partitioning trees, *Comput. Graph.* **21** (4) (1987), 153–162, Proc. SIGGRAPH '87.

[17] Cs. D. Tóth, A note on binary plane partitions, *Discrete Comput. Geom.* **30** (2003), 3–16.

[18] Cs. D. Tóth, Binary space partitions for orthogonal fat rectangles, in: *Proc. 11th European Sympos. Algorithms*, vol. 2832 of LNCS, Springer-Verlag, Berlin, 2003, pp. 494–505.