Binary Space Partitions for 3D Subdivisions

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

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

Abstract

We consider the following question: Given a subdivision of space into n convex polyhedral cells, what is the worst-case complexity of a binary space partition (BSP) for the subdivision? We show that if the subdivision is rectangular and axis-aligned, then the worst-case complexity of an axis-aligned BSP is $\Omega(n^{4/3})$ and $O(n^\alpha \log^2 n)$, where $\alpha = 1 + \log_2(4/3) = 1.4150375\ldots$. By contrast, it is known that the BSP of a collection of n rectangular cells *not* forming a subdivision has worst-case complexity $\Theta(n^{3/2})$. We also show that the worst-case complexity of a BSP for a general convex polyhedral subdivision of total complexity $O(n)$ is $\Omega(n^{3/2})$.

1 Introduction

A binary space partition (BSP) is a recursive convex subdivision of space, defined with respect to some set of objects S. Given an open convex region of space containing S, a BSP partitions the region and objects with a cutting plane, then recursively partitions the two subproblems that result. The process stops when each open partition region intersects at most one object of S. In the ideal case, the number of regions in the final BSP would be at most n, the number of input objects. However, because a BSP may fracture input objects many times, the BSP size may be much larger than n.

Binary space partitions were introduced in the graphics community [8, 13] to solve hidden surface removal problems, and since then have been used for a wide variety of applications, including solid modeling, ray tracing, shadow generation, and robotics, to name only a representative sample [1, 4, 5, 9, 10, 14]. Because most computer graphics is concerned with visualizing 3-space, $\mathcal{R}^3$ is the most important setting for BSPs.

Computational geometers have studied BSPs since the work of Paterson and Yao in the late 1980s [11, 12]. Because BSPs are often used to decompose large data sets, the size of a BSP (the number of regions in the final decomposition) can be crucial to the performance of BSP-based applications. Thus theoreticians have focused on finding upper and lower bounds on the sizes of BSPs in various settings.

In two dimensions, there are linear-size BSPs for axis-aligned segments [11], segments with a fixed number of orientations [15], segments with lengths in a bounded range [6], and fat objects or homothets [6]. For arbitrary segments in the plane, an optimal BSP has size $O(n \log n)$ and $\Omega(n \log n / \log \log n)$ [11, 16].

In three dimensions, a collection of n arbitrarily-oriented segments requires a BSP of worst-case size $\Theta(n^2)$; axis-aligned segments or rectangles have a smaller BSP of worst-case size $\Theta(n^{3/2})$ [11, 12]. Bounds can also be obtained for axis-aligned hyperrectangles in $\mathcal{R}^d$, for $d > 3$ [7].

In all of these cases, the input consists of disjoint objects that do not cover their containing space. In this paper we consider the case in which we are given a *convex subdivision* of space as input. (BSPs of rectilinear subdivisions in the plane have previously been considered; however, in two dimensions, the improvement is only in the constant factors, not in asymptotic complexity [2].) We are interested in the complexity of refining an arbitrary convex subdivision into a binary space partition In this case, the known lower bounds do not tell us anything nontrivial. For example, in the lower bound construction for axis-aligned BSPs [12], just filling in the space around the n line segments requires $\Theta(n^{3/2})$ cells. Thus for that construction, the BSP size is *linear* in the size of the *subdivision.* Similarly, converting the lower bound construction for arbitrary segments [11] into a subdivision requires $\Theta(n^2)$ additional cells, trivializing

*Subhash Suri's research on this paper was partially supported by National Science Foundation grants CCR-9901958 and ANI-9813723.

the $\Omega(n^2)$ lower bound for arbitrary BSPs.

Our main results apply to BSPs for axis-aligned convex subdivisions. We show that the complexity bounds for segments and rectangles do not apply in this case. If every cell in the input subdivision is an axis-aligned box, then the worst-case size of a BSP for the subdivision is strictly less than the $\Theta(n^{3/2})$ bound for non-subdivision input. In particular, we show that the BSP size is $O(n^\alpha \log^2 n)$, for $\alpha = 1 + \log_2(4/3) = 1.4150375\ldots$. On the other hand, we exhibit subdivisions such that any axis-aligned BSP must have size $\Omega(n^{4/3})$. If the vertices of the input subdivision are constrained to lie on c axis-aligned planes, for $c \le n^{1/3}$, then the BSP has worst-case size $\Omega(nc)$ and $O(nc \log c)$. If the input is an unconstrained convex subdivision, then we show that there are linear-size subdivisions whose BSPs must have size $\Omega(n^{3/2})$. We have no improvements in the upper bound at the moment.

2 Preliminaries

The input to a binary space partition problem consists of an open convex region C of $\mathcal{R}^d$, along with a set S of interior-disjoint objects that are contained in C. (The restriction to disjoint objects can be relaxed, but it complicates the bounds.) A BSP partitions C and S with a hyperplane into $\{C_1, C_2\}$ and $\{S_1, S_2\}$, then recursively partitions (C_1, S_1) and (C_2, S_2). The partitioning stops when for every subproblem (C_i, S_i), $|S_i| \le 1$.

A BSP corresponds to a binary tree. Each internal node of the tree represents an open region C_i, its corresponding set of objects S_i, and the plane that splits C_i and S_i in two. A leaf of the tree represents a *leaf cell* C_i whose set S_i contains at most one object. The height of the tree is important for applications such as point location; however, we focus our attention on the number of leaves, that is, the size of the final partition.

Free cuts are important for efficient construction of BSPs. A free cut is a partition that separates the object set without splitting any object. See Figure 1. When a subproblem (C_i, S_i) has a free cut, it is always worth splitting at 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). A BSP that uses only free cuts necessarily has size less than n. Of course, not all subdivisions admit a free cut.

We focus on BSPs for convex subdivisions of 3-space. The objects to be partitioned are the convex polyhedral cells of the subdivision. It follows that every subdivision face must be included in a splitting plane of the BSP. Because most of our attention will be devoted to axis-aligned convex subdivisions, we give a few definitions specific to this case:

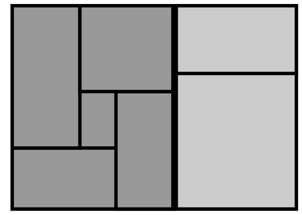

Figure 1: A free cut (shown thickened) in a planar subdivision.

> A *box* is an axis-aligned rectangular parallelepiped. A *box subdivision* is a partition of some box (called the *outer box* or *boundary* of the subdivision) into smaller boxes called *cells*.
>
> A *rod* is a cell of a box subdivision that is incident to two opposite faces of the subdivision's outer box. If a rod is incident to two opposite faces F_1 and F_2 of the subdivision's outer box, the rod's *orientation* is the direction normal to F_1 and F_2. A rod may have one, two, or three orientations, depending on the number of opposite face pairs it touches. We call such rods *singly-oriented*, *doubly-oriented*, or *triply-oriented*, respectively.

LEMMA 2.1. *A cell C of a box subdivision is a rod if and only if it has all its vertices on the outer box of the subdivision.*

Proof. If C is a rod, then it has two opposite faces F_1 and F_2 that are contained in faces of the subdivision's outer box. But all the vertices of C belong to one or the other of F_1 and F_2, and so all the vertices of C lie on the subdivision's outer box.

Suppose C is not a rod. Then C has three faces, one for each dimension, that do not lie on the outer box. These three faces (denoted F_x, F_y, and F_z) share a common vertex, since none of them is opposite the other. Each of the planes supporting one of F_x, F_y, and F_z intersects the interior of the outer box. Therefore, their common intersection, which is a vertex of C, must lie in the interior of the outer box. ∎

3 Lower bounds for box subdivisions

Paterson and Yao gave a configuration of $3n$ (non-space-filling) axis-aligned rods such that any BSP for the configuration cuts the rods into $\Omega(n^{3/2})$ subcells [12]. We describe a variant of the Paterson-Yao construction in some detail, because it is the basis of our lower bound for box subdivisions.

The rods of the Paterson-Yao construction belong to three families, each parallel to one of the coordinate axes, and form an interlocking grid. See Figure 2. Without loss of generality, assume that $n = m^2$ for some integer m. The x-, y-, and z-parallel families consist of boxes indexed by $1 \le i, j, k \le m$, described by the following Cartesian products:

$$[0.5, m + 0.5] \times [j, j + 0.5] \times [k, k + 0.5]$$
$$[i, i + 0.5] \times [0.5, m + 0.5] \times [k - 0.5, k]$$
$$[i - 0.5, i] \times [j - 0.5, j] \times [0.5, m + 0.5]$$

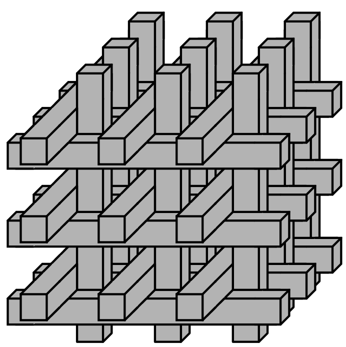

Figure 2: The Paterson-Yao lower bound construction, with the rods slightly separated and lengthened to make them easier to see.

It is straightforward to observe that the rods in each family are disjoint. Furthermore, rods in different families are disjoint, because for each pair of families there is one of the three dimensions (x, y, or z) in which the two families lie in disjoint ranges: for one family, any rod's coordinates lie in the range $[A, A + 0.5]$, for some integer A; for the other family, any rod's coordinates lie in $[B - 0.5, B]$, for integral B. Further observe that each grid point (i, j, k), for $1 \le i, j, k \le m$, is incident to one rod from each of the three families.

For each grid point (i, j, k), for $1 \le i, j, k \le m$, consider the box

$$[i - 0.5, i + 0.5] \times [j - 0.5, j + 0.5] \times [k - 0.5, k + 0.5].$$

These boxes are interior-disjoint, and each must be cut by a BSP plane passing through (i, j, k), since the BSP must separate the three rods in the neighborhood of (i, j, k). The first plane that passes through (i, j, k) must cut at least one of the three rods inside the box centered on (i, j, k). Since all the boxes are disjoint, the total number of rod cuts is at least $m^3 = n^{3/2}$.

The Paterson-Yao construction is *opaque*, in the sense that every axis-parallel line passing through the big cube

$$[0.5, m + 0.5] \times [0.5, m + 0.5] \times [0.5, m + 0.5]$$

intersects the closure of at least one rod. However, the configuration of rods is not a subdivision: the rods do not fill space. This is easy to prove by observing that the total volume of the big cube is m^3, and the total volume of each rod inside the cube is $m/4$. There are $3m^2$ rods, for a total rod volume inside the cube of $0.75m^3$, which is less than m^3. The empty space is distributed in $2m^3$ little cubes. In particular, for each grid point (i, j, k) inside the big cube, the little cubes

$$[i - 0.5, i] \times [j, j + 0.5] \times [k - 0.5, k]$$

and

$$[i, i + 0.5] \times [j - 0.5, j] \times [k, k + 0.5]$$

are empty. Each of these little cubes has volume 1/8. See Figure 3.

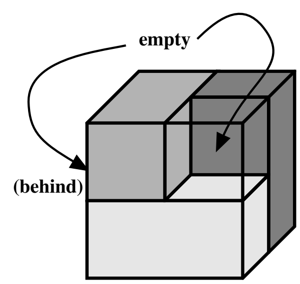

Figure 3: A unit cube centered on a grid point in the Paterson-Yao construction.

Converting the Paterson-Yao configuration into a subdivision requires adding $\Theta(m^3)$ new cells—two cubic cells for each triple (i, j, k). Thus the $\Omega(m^3)$ lower bound on the BSP complexity becomes trivial—it is linear in the input size. However, with a little more work, we can extend the construction to give a nontrivial lower bound.

THEOREM 3.1. *There is a three-dimensional subdivision of space into n axis-aligned boxes such that any axis-aligned binary space partition for the subdivision cuts the boxes into $\Omega(n^{4/3})$ subcells.*

Proof. We begin with a Paterson-Yao construction consisting of $3m^2$ rods, and then extend it to a subdivision of the cube $[0.5, m+0.5]^3$ by adding the $2m^3$ little cube cells needed to fill in the empty space between the rods. We then subdivide each rod longitudinally into m parallel sub-rods.

As in the original Paterson-Yao argument, each of the unit cubes centered on a grid point (i, j, k) must be cut by at least one BSP plane, since the BSP must separate the cells incident to (i, j, k). The first plane that cuts a unit cube must completely cross one of the original rods, and hence it cuts at least m sub-rods. The total number of rod cuts is at least $m^3 \times m = m^4$. The total number of cells in our subdivision is $n = 3m^2 \times m + 2m^3 = 5m^3$, and hence the number of subcells produced by the BSP is $\Omega(n^{4/3})$. ∎

4 Upper bounds for box subdivisions

We begin by showing that the difficulty of computing a BSP for a box subdivision is due to cells with vertices not on the boundary of the subdivision, that is, non-rod cells. We first establish a useful technical lemma.

Lemma 4.1. *Consider a box subdivision with n cells, all of them rods. If each rod in the subdivision is singly-oriented, then all the rods, taken together, have at most two orientations.*

Proof. Suppose to the contrary that A, B, and C are three rods with three different orientations. Let ℓ_A, ℓ_B, and ℓ_C be lines running down the centers of A, B, and C, respectively, parallel to their rods' orientations. These are three axis-parallel skew lines. Let P_{AB} be the axis-aligned plane containing ℓ_A and intersecting ℓ_B (see Figure 4). Define P_{BC} and P_{CA} similarly. Note that P_{AB} does not intersect ℓ_C, because it is parallel to it. The point $P_{AB} \cap \ell_B$ lies inside B—P_{AB} lies between the planes supporting the outer box faces that ℓ_B pierces. Now $P_{AB} \cap P_{BC}$ is a line parallel to ℓ_A that intersects ℓ_B. It follows that $P_{AB} \cap P_{BC}$ is disjoint from A and intersects B. Similar claims hold for $P_{BC} \cap P_{CA}$ and $P_{CA} \cap P_{AB}$.

Let p be the point $P_{AB} \cap P_{BC} \cap P_{CA}$. Point p is disjoint from $A \cup B \cup C$; p also lies inside the outer box, since it is connected by three orthogonal lines to the points $P_{AB} \cap \ell_B$, $P_{BC} \cap \ell_C$, and $P_{CA} \cap \ell_A$, all of which lie inside the outer box. Now we claim that the cell containing p cannot be a rod: for each of the axis-parallel directions, there is a line (e.g., $P_{AB} \cap P_{BC}$) that intersects one of A, B, and C, and hence the cell

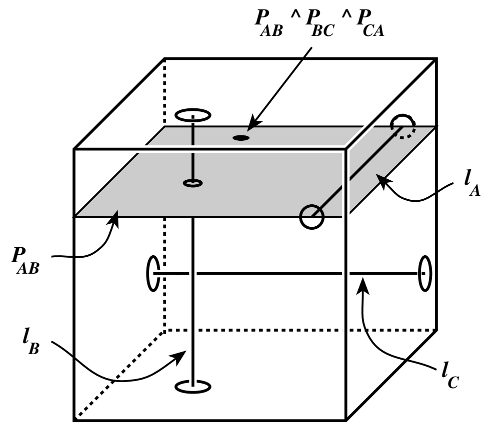

Figure 4: Singly-oriented rods with three different orientations imply the existence of an interior vertex.

containing p cannot touch any pair of opposite faces of the outer box. This completes the proof. ∎

Lemma 4.2. *If a box subdivision with n cells has no vertices in the interior of its outer box, then it has a BSP of size $O(n)$.*

Proof. All the cells in the subdivision are rods, by Lemma 2.1. A triply-oriented rod completely fills the outer box, and no BSP cuts are needed. If a rod is doubly-oriented, any of its faces not on the outer box touches four faces of the outer box. That is, we can make a free cut along such a face and recursively partition the resulting two subdivisions. Thus, we may assume that all rods are singly-oriented. By Lemma 4.1, therefore, all the rods in the subdivision have at most two different orientations. We handle the two cases below separately.

If the rods in the subdivision have two different orientations, we produce a free cut. See Figure 5. Suppose without loss of generality that the rods are x- and y-oriented, and that the rod R_1 with greatest z-coordinate is x-oriented. Note that no x-oriented rod can intersect the z-interval of a y-oriented rod, or else the two rods would intersect. The y-oriented rod R_2 with greatest z-coordinate lies strictly below R_1, and hence strictly below the top of the outer box. Let z_2 be the maximum z-coordinate of R_2. The part of the outer box with $z \geq z_2$ contains only x-oriented rods, and no x-oriented rod lies in the z-interval of R_2. Hence the plane $z = z_2$ is a free cut.

If all the rods in the cell have the same orientation, then we project them onto the plane perpendicular to their orientation. This gives a two-dimensional subdivision, which has a two-dimensional BSP of size

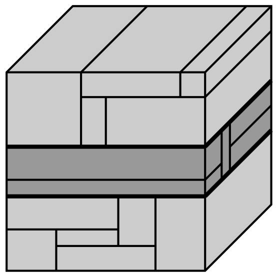

Figure 5: Singly-oriented rods with two different orientations imply the existence of a free cut.

at most cn, for some small constant c [7, 12]. Extending each linear cut of the two-dimensional BSP into a planar cut parallel to the rod orientation, we produce a BSP for the box subdivision of size at most cn.

Putting the pieces together, we have shown that a subdivision containing only n rod cells always has a free cut unless all the rods have the same orientation, in which case a BSP of size cn is possible. The total size of a BSP for the subdivision is therefore

$$\begin{aligned} f(n) &\leq \max\left(\max_{0<i<n}\big(f(i)+f(n-i)\big),\, cn\right) \\ &\leq cn. \end{aligned}$$

∎

We apply the preceding lemma by slicing a subdivision through its interior vertices, producing a collection of smaller subdivisions, each of which contains only rod cells. The lemma implies that each of these subdivisions has a linear-size BSP. The key to efficiency is bounding the total complexity of the rod subdivisions produced by the slicing procedure.

Given a subdivision with n cells and k interior vertices, we define a potential for each cell, depending on k and the incidences between the cell and the outer box of the subdivision.

Definition 4.3. *For a given cell b, define $F(b)$ to be the number of faces of b that lie on the sides of the outer box. Note that $0 \leq F(b) \leq 6$.*

Definition 4.4. *For a given cell b, let $\delta(b)$ be a 0/1 variable that is 1 if b is a rod, and 0 otherwise.*

The potential of a cell b is

$$C(b,k) = \begin{cases} 0 & \text{if } b \text{ is a doubly- or triply-oriented rod} \\ 2^{4-F(b)}\lceil \lg k\rceil^{2-\delta(b)} & \text{otherwise} \end{cases}$$

where $\lg k \equiv \log_2 k$. The maximum value of $C(b,k)$ is $16\lceil \lg k\rceil^2$ for a cell completely interior to the subdivision. The total potential of a subdivision is the sum of the potentials of its cells. Thus the potential of a subdivision with n cells and k interior vertices is at most $16n\lceil \lg k\rceil^2$.

We use the potentials to guide our BSP construction, ensuring that our BSP cuts do not increase the potential by too much. We argue that the total complexity of the final BSP is asymptotically bounded by the total potential added during the slicing procedure, which we bound in Theorem 4.8. Our partition algorithm is as follows:

1. If there are any free cuts available, choose one and then recursively partition the two remaining subdivisions. (If this step is not executed, no rod is doubly- or triply-oriented.)
2. If the subdivision has at most one interior vertex, slice through it (if it exists) with a single BSP cut, then use Lemma 4.2 on the remaining pieces to compute a linear-size BSP.
3. (The subdivision has $k > 1$ interior vertices, and all rods are singly-oriented.)
   (a) For each orientation $\theta \in \{x, y, z\}$, compute the total potential of the rods with orientation θ.
   (b) For the orientation θ with minimum potential σ, cut the subdivision with a plane perpendicular to θ and passing through the median internal vertex ($\leq k/2$ internal vertices lie on either side).
   (c) Recursively partition the two subdivisions produced by the cut.

We first show that the potential of a cell increases only when it is cut.

Lemma 4.5. *For a given cell b that is not partitioned by a particular BSP cut, its potential after the cut is no greater than its potential before the cut.*

Proof. If b is not touched by the cutting plane, the exponents in

$$C(b,k) = 2^{4-F(b)}\lceil \lg k\rceil^{2-\delta(b)}$$

are not changed, and the value of k cannot increase; hence the potential does not increase. If b is adjacent to the cutting plane but not intersected by it, then $F(b)$ increases, $\delta(b)$ may increase, and k does not increase. Therefore $C(b,k)$ decreases. ∎

Corollary 4.6. *The total potential is not increased by the free cuts performed in Step 1.*

Lemma 4.7. *Suppose that Step 3 of the algorithm is applied to a subdivision with potential Σ, and t cells are split. Then the two resulting subdivisions have total potential at most $\frac{4}{3}\Sigma - t$.*

Proof. By Lemma 4.5, we need to consider only cells that are partitioned by the cutting plane; the potentials of the other cells do not increase. For convenience, let us refer to a θ-oriented rod as a θ-rod, and refer to a cell face that is contained in a θ-extreme face of the outer box as a θ-face. Thus a θ-rod has two θ-faces. Consider a cell b partitioned by the cutting plane. We analyze the potential of the two child subcells produced from b in the following cases:

1. b is a θ-rod. It initially has potential $C(b,k) = 2^{4-F(b)}\lceil \lg k \rceil$; note that $F(b) \le 4$, since Step 1 of the algorithm was not applied. The two child cells are both θ-rods, and have total potential at most

$$\begin{aligned} 2 \cdot 2^{4-F(b)} \left\lceil \lg \frac{k}{2} \right\rceil &= 2 \cdot 2^{4-F(b)}(\lceil \lg k \rceil - 1) \\ &= 2\,C(b,k) - 2. \end{aligned}$$

2. b has no θ-face. In this case $F(b) \le 3$. Each child cell has one θ-face, and the total potential for both of them is at most

$$\begin{aligned} & 2 \cdot 2^{4-F(b)-1} \left\lceil \lg \frac{k}{2} \right\rceil^{2-\delta(b)} \\ & \quad \le 2^{4-F(b)}(\lceil \lg k \rceil - 1)^{2-\delta(b)} \\ & \quad \le C(b,k) - 1. \end{aligned}$$

3. b has one θ-face. In this case one child is a θ-rod, and the other has one θ-face. We distinguish two subcases:

   (a) b is not a rod. In this case $F(b) \le 3$, and the children have potential at most

$$\begin{aligned} & 2^{4-F(b)} \left\lceil \lg \frac{k}{2} \right\rceil^2 + 2^{4-F(b)-1} \left\lceil \lg \frac{k}{2} \right\rceil \\ & \quad \le 2^{4-F(b)} \left\lceil \lg \frac{k}{2} \right\rceil \left(\lceil \lg k \rceil - 1 + \frac{1}{2} \right) \\ & \quad \le 2^{4-F(b)}(\lceil \lg k \rceil - 1)\lceil \lg k \rceil \\ & \quad \le C(b,k) - 1. \end{aligned}$$

   (b) b is a rod with a non-θ orientation. Then $F(b) \le 4$. After the cut, the child that is a doubly-oriented rod has zero potential. The potential of the other child is at most

$$\begin{aligned} & 2^{4-F(b)} \left\lceil \lg \frac{k}{2} \right\rceil \\ & \quad = 2^{4-F(b)}\left(\lceil \lg k \rceil - 1\right) \le C(b,k) - 1. \end{aligned}$$

In each case except the first, the potential of the child cells is at least 1 less than the potential of the parent cell that is split. In the first case, the potential is at least 1 less than twice the potential of the parent cell. The total potential of the cells split in the first case is $\sigma \le \Sigma/3$. (Since the rod potentials are partitioned among the three orientations, the potential σ associated with the minimum-potential orientation is at most $\Sigma/3$.) Summing over all the split cells proves the lemma. ∎

Theorem 4.8. *A box subdivision with n cells has a BSP of size $O(n^\alpha \log^2 n)$, where $\alpha = 1 + \log_2(4/3) = 1.4150375\ldots$.*

Proof. Our partition algorithm creates a BSP tree T. At the leaves of T are individual BSP regions. Each internal node corresponds to a subdivision that is further partitioned. For analysis purposes, we will consider the subtrees that correspond to invocations of Step 2 separately from the rest of the tree. Let T^- be the partial BSP tree obtained by removing all of the Step 2 trees from the bottom fringe of T.

By Lemma 4.2, the number of leaves in a Step 2 subtree is proportional to the total number of cells in the subdivision at its root. The root of a Step 2 subtree has at most as many cells as the leaf of T^- that is its parent. Thus the total number of leaves of T (the complexity of the final BSP) is proportional to the number of cells at the leaves of T^-.

We bound the number of cells at the leaves of T^- using a credit scheme. Our partition algorithm builds T^- recursively, starting from a single node (the root). Each recursive invocation of the algorithm applies Step 1 or Step 3 to add two children to some leaf of the current tree. We assign some number of credits to the root initially, then maintain the invariant that at each stage of the construction, the total number of credits assigned to the tree is at least as large as the number of cells plus the potential of the cells in the leaves' subdivisions. Thus at the end of the construction of T^-, the number of credits bounds the number of cells at the leaves of T^-, which in turn asymptotically bounds the complexity of the final BSP.

We assume that $k > 1$, since otherwise Step 2 applies immediately, and the total BSP size is $O(n)$, by Lemma 4.2. We assign $n + 16n\lceil \lg k \rceil^2$ credits to the initial subdivision (stored at the root of T^-), so the invariant holds initially.

Whenever Step 1 is applied, the total number of cells remains constant, and their total potential does not increase (Lemma 4.5), so the invariant continues to hold.

Whenever we split some θ-rods with total potential $\sigma \leq \Sigma/3$ in Step 3, we increase the number of credits by σ. Let t be the total number of cells cut by Step 3 (not just θ-rods). Let C_{pre} be the total potential of the cells in the leaves' subdivisions before Step 3 is applied and let C_{post} be the potential afterward. By Lemma 4.7, $C_{\text{post}} + t \leq C_{\text{pre}} + \sigma$. If the number of cells before the split was N, and the number of credits before the split was A, the initial condition $A \geq C_{\text{pre}} + N$, together with the preceding observation, implies that the invariant continues to hold: $A + \sigma \geq C_{\text{post}} + N + t$.

Because Step 3 splits the current subdivision through a median interior vertex, the number of such splits on the path from the root of T^- to any leaf of T^- is at most $\lg k$. If we remove every non-root Step 1 node in T^- by collapsing the edge joining it to its parent, we are left with a tree T^* of height at most $\lg k$ in which all the nodes (except possibly the root) correspond to Step 3 splits. We bound the credits assigned to T^* (and hence to T^-) level by level. The total rod potential at a single level of T^* is bounded by the number of credits at that level, and so the number of credits assigned to the next level is at most 4/3 of that at the current level. In the worst case, the total number of credits assigned to T^- is at most

$$(n + 16\lceil \lg k \rceil^2) \cdot (4/3)^{\lg k}.$$

But $(4/3)^{\lg k} = 2^{\lg(4/3)\lg k} = k^{\lg(4/3)} = O(n^{\lg(4/3)})$. Thus the number of credits, and hence the final size of the BSP, is $O(n^\alpha \log^2 n)$, where $\alpha = 1 + \lg(4/3) = 1.4150375\ldots$. ∎

5 Bounds for constrained box subdivisions

In this section we consider the special case of box subdivisions whose vertices are constrained to lie on a fixed set of c axis-aligned planes, where $c \leq n^{1/3}$. Note that this is less restrictive than requiring all the *faces* to lie on the fixed set of planes. We are able to prove nearly matching lower and upper bounds on the worst-case BSP complexity.

THEOREM 5.1. *For any $c \leq n^{1/3}$, there exists an n-cell box subdivision with all of its vertices on a fixed set of c axis-aligned planes such that any axis-aligned BSP for the subdivision has size $\Omega(nc)$.*

Proof. Let c have the form $c = 3(2d+1)$. Our construction is based on the proof of Theorem 3.1. We construct a Paterson-Yao $d \times d \times d$ grid of rods, where each rod extends from one side to the other of the cube

$$[0.5, d+0.5] \times [0.5, d+0.5] \times [0.5, d+0.5].$$

We fill in the gaps around the rods with $2d^3$ little cube cells. All the vertices of the subdivision lie on the $3(2d+1)$ half-integer planes that intersect the outer box. We further subdivide each rod into m parallel sub-rods, for a value of m to be determined below. All the sub-rod vertices lie on the outer box of the subdivision. The total number of sub-rods is $3md^2$. As in the proof of Theorem 3.1, the BSP complexity is $\Omega(md^3)$.

To determine m, we set $2d^3 + 3md^2 = n$. This solves to

$$m = \frac{n - 2d^3}{3d^2} = \frac{n}{3d^2} - \frac{2d}{3}.$$

The BSP lower bound is therefore $\Omega(nd/3 - 2d^4/3)$. We assumed that $c \leq n^{1/3}$, which implies that $d \leq \frac{1}{6}n^{1/3}$, and hence $\Omega(nd/3 - 2d^4/3) = \Omega(nc)$. ∎

THEOREM 5.2. *For any $c \leq n^{1/3}$, any n-cell box subdivision with all of its vertices on a fixed set of c axis-aligned planes has an axis-aligned BSP of size $O(nc \log c)$.*

Proof. Every cell vertex lies on one of c chosen planes. We begin by slicing the subdivision along the chosen planes perpendicular to the x-axis. This produces $O(nc)$ fragments that are x-rods in their respective subdivisions, and $O(n)$ fragments that are not x-rods.

Within each of the $O(c)$ subdivisions produced by the first round of cuts, we perform cuts along the chosen planes perpendicular to the y-axis. We perform these cuts in a balanced hierarchical order: we first cut at the median y-plane, then recursively perform y-cuts in each of the two child subdivisions. Whenever a cell is cut by two y-cuts, the part of the cell between the two cuts becomes a y-rod. If the cell was already an x-rod, then it is a doubly-oriented rod, and we can remove it immediately by free cuts along the other two (z) faces. No further cuts are necessary for a cell removed by free cuts. Because the y-cuts are performed in a balanced hierarchical order, each x-rod can be cut only $O(\log c)$ times, producing $O(\log c)$ fragments that are removed by free cuts and at most two fragments that are not removed by free cuts. Thus the y-cuts produce $O(nc \log c)$ fragments from the cells that were x-rods after the x-cuts, and $O(nc)$ from the cells that were not x-rods. All but $O(nc)$ fragments in the first set will not be cut again (because they are doubly-oriented rods, removed by free cuts). Of the second set, $O(nc)$ fragments are y-rods, and $O(n)$ are not y-rods.

Within each of the nontrivial subdivisions remaining after the y-cuts, we perform z-cuts in a balanced hierarchical order. As in the previous round of cuts, each cell that was initially an x-rod or a y-rod can be cut into $O(\log c)$ fragments, all but two of which will be removed by free cuts. Each cell that was not originally an x-rod or a y-rod may be cut up to c times. The total complexity of the BSP that results is $O(nc \log c)$, since each round of cuts produces $O(nc \log c)$ cells that are removed by free cuts and $O(nc)$ cells that are considered in the next round of cuts. ∎

6 Bounds for general convex subdivisions

Paterson and Yao gave a lower bound construction of n line segments in 3-space such that any BSP for the segments has size $\Omega(n^2)$ [11]. Their construction does not immediately give a bound for subdivisions, because filling in the space around the segments with convex cells increases the input size to $\Theta(n^2)$. However, we can extend the construction by using bundles of rods, just as we did in Theorem 3.1, to get a nontrivial lower bound.

THEOREM 6.1. *There is a three-dimensional subdivision of space into n convex cells with total complexity $O(n)$ such that any binary space partition for the subdivision has size $\Omega(n^{3/2})$.*

Proof. We first recap Paterson and Yao's lower bound for segments, which is based on a polyhedral construction first used by Chazelle [3]. Consider two sets of lines: *red* lines have equation $z = jx + \epsilon, y = j$, for $j \in \{1, \ldots, N/2\}$, and *blue* lines have equation $z = iy - \epsilon, x = i$, for $i \in \{1, \ldots, N/2\}$. The red lines lie just above the hyperboloid $z = xy$, and the blue lines lie just below it. (We can choose $\epsilon = 0.1$ in this construction.) The lower bound segments are obtained by clipping the red and blue lines to the range $0 \leq x, y \leq (N/2) + 1$. See Figure 6. If we project any pair of red and blue segments into the xy-plane, the projections intersect at an integer grid point (i, j). The projections of any quadruple of two red and two blue segments form a rectangle with corners on the integer grid. Paterson and Yao argued, based on earlier work by Chazelle [3], that for any quadruple of two red and two blue segments, any BSP must break at least one of the subsegments that project to edges of the rectangle. Since we can choose $\Theta(N^2)$ quadruples that project to disjoint unit squares, any BSP must have $\Omega(N^2)$ size.

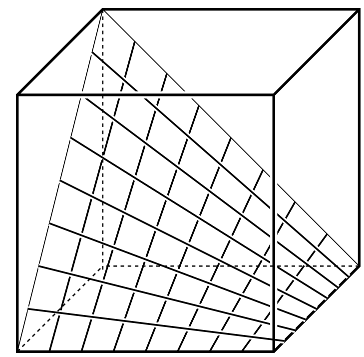

Figure 6: The $\Omega(n^2)$ lower bound for BSPs of arbitrary line segments.

To get a lower bound on BSPs for convex subdivisions, we replace each red or blue segment by a bundle of rods. The bundle is a triangular prism, and the rods inside the bundle are also triangular prisms whose union is the bundle. The cross section of the bundles can be chosen to be ϵ. Thus all the longitudinal (parallel to the rod axis) rod edges are within ϵ of the position of the original segment from which the rod is derived. The argument of Paterson and Yao can be applied to any quadruple of two red and two blue longitudinal rod edges. (A rod edge is incident to multiple cells of the subdivision. A BSP that separates the cells must also include the edges in its splitting planes.) Thus, if every bundle contains M rods, any BSP must break the rod edges (and hence the rods themselves) into $\Omega(N^2 M)$ pieces. The BSP must have size $\Omega(N^2 M)$.

Chazelle showed that $\Theta(N^2)$ convex polyhedra with total complexity $\Theta(N^2)$ are needed to fill in the space around the N bundles [3]. In fact, we can use a BSP for the bundle edges to produce such a convex subdivision of size $\Theta(N^2)$ [11]. To balance the number of rods and the size of the remaining subdivision, we set $M = N$. The size of the subdivision is therefore $n = \Theta(N^2)$. The BSP size is $\Omega(N^2 M) = \Omega(n^{3/2})$. ∎

7 Future work

Several tantalizing questions remain to be answered for subdivision BSPs:

- What is the worst-case complexity of an axis-aligned BSP for a box subdivision? Though the lower and upper bounds presented in this paper are both tighter than what was previously known, there is still a significant gap between them.

In joint work with Csaba Tóth, we have recently been able to reduce the upper bound to match the $\Omega(n^{4/3})$ lower bound (in preparation).

- Is there an upper bound better than $O(n^2)$ on the size of a BSP for a general convex subdivision with n cells of total complexity $O(n)$? The charging scheme we used to prove an upper bound for box subdivisions does not seem to apply directly to this case, but perhaps there is some generalization that will work.

- Do the box subdivision bounds extend to higher dimensions? It seems likely that our techniques will give nontrivial results in $\mathcal{R}^4$ and above, but we have not worked out the details.

References

[1] C. Ballieux. Motion planning using binary space partitions. Technical Report Inf/src/93-25, Utrecht University, 1993.

[2] P. Berman, B. DasGupta, and S. Muthukrishnan. Slice and dice: A simple, improved approximate tiling recipe. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 455–464, 2002.

[3] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM J. Comput.*, 13:488–507, 1984.

[4] N. Chin and S. Feiner. Near real-time shadow generation using BSP trees. In *Proc. SIGGRAPH '89*, volume 23, pages 99–106, New York, 1989.

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

[6] M. de Berg, M. de Groot, and M. Overmars. New results on binary space partitions in the plane. *Comput. Geom. Theory Appl.*, 8:317–333, 1997.

[7] A. Dumitrescu, J. S. B. Mitchell, and M. Sharir. Binary space partitions for axis-parallel segments, rectangles, and hyperrectangles. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 141–149, 2001.

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

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

[10] 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, February 1986.

[11] M. S. Paterson and F. F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5:485–503, 1990.

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

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

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

[15] C. D. Tóth. Binary space partitions for line segments with a limited number of directions. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 465–471, 2002.

[16] C. D. Tóth. A note on binary plane partitions. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 151–156, 2001.