Title: Binary Space Partitions
Name: Adrian Dumitrescu[1], Csaba D. Tóth[2,3]
Affil./Addr. 1: Computer Science, Univ. of Wisconsin–Milwaukee, Milwaukee, WI, USA
Affil./Addr. 2: Mathematics, California State University Northridge, Los Angeles, CA, USA
Affil./Addr. 3: Computer Science, Tufts University, Medford, MA, USA
Keywords: Computational geometry; Recursive partition; Convex decomposition; BSP tree
SumOriWork: No Years and Authors of Summarized Original Work Given

Binary Space Partitions

Adrian Dumitrescu[1], Csaba D. Tóth[2,3]

[1] Computer Science, Univ. of Wisconsin–Milwaukee, Milwaukee, WI, USA
[2] Mathematics, California State University Northridge, Los Angeles, CA, USA
[3] Computer Science, Tufts University, Medford, MA, USA

Years aud Authors of Summarized Original Work

No Years and Authors of Summarized Original Work Given

Keywords

Computational geometry; Recursive partition; Convex decomposition; BSP tree

Problem Definition

The *binary space partition* (for short, *BSP*) is a scheme for subdividing the ambient space $\mathbb{R}^d$ into open convex sets (called *cells*) by hyperplanes in a recursive fashion. Each subdivision step for a cell results in two cells, in which the process may continue, independently of other cells, until a stopping criterion is met. The binary recursion tree, also called *BSP-tree*, is traditionally used as a data structure in computer graphics for efficient rendering of polyhedral scenes. Each node v of the BSP-tree, except for the leaves, corresponds to a cell $C_v \subseteq \mathbb{R}^d$ and a partitioning hyperplane H_v. The cell of the root r is $C_r = \mathbb{R}^d$, and the two children of a node v correspond to $C_v \cap H_v^-$ and $C_v \cap H_v^+$, where H_v^- and H_v^+ denote the open halfspaces bounded by H_v. Refer to Fig. 1.

A binary space partition *for* a set of n pairwise disjoint (typically polyhedral) objects in $\mathbb{R}^d$ is a BSP where the space is recursively partitioned until each cell intersects at most one object. When the BSP-tree is used as a data structure, every leaf v stores the fragment of at most one object clipped in the cell C_v, and every interior node v stores the fragments of any lower-dimensional objects that lie in $C_v \cap H_v$.

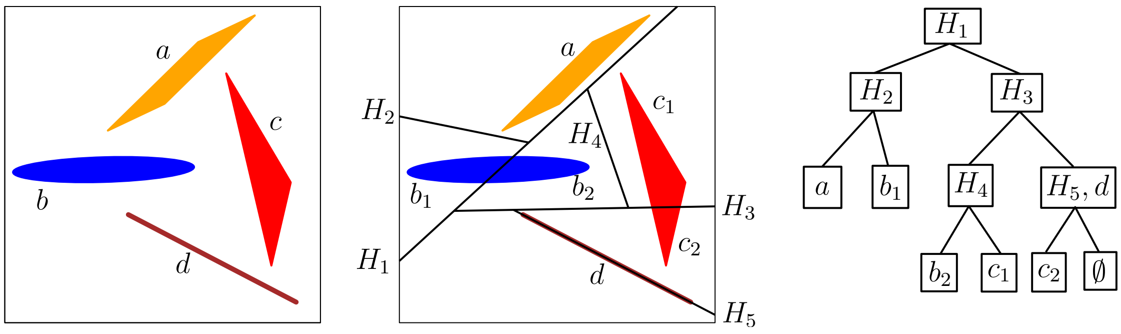

Fig. 1. Three 2-dimensional convex objects and a line segment (left), a binary space partition with five partition lines $H_1, \ldots, H_5$ (center), and the corresponding BSP tree (right).

A BSP for a set of objects has two parameters of interest: the *size* and the *height* of the corresponding BSP-tree. Ideally, a BSP partitions space so that each object lies entirely in a single cell or in a cutting hyperplane, yielding a so-called *perfect* BSP [4]. However, in most cases this is impossible, and the hyperplanes H_v partition some of the input objects into *fragments*. Assuming that the input objects are k-dimensional, for some $k \leq d$, the BSP typically stores only k-dimensional fragments, i.e., object parts clipped in leaf cells C_v or in $C_v \cap H_v$ at interior nodes.

The size of the BSP-tree is typically proportional to the number of k-dimensional fragments that the input objects are partitioned into, or the number of nodes in the tree. Given a set S of objects in $\mathbb{R}^d$, one would like to find a BSP for S with small size and/or height. The *partition complexity* of a set of objects S is defined as the minimum size of a BSP for S.

GLOSSARY

- **Autopartition**: a class of BSPs obtained by imposing the constraint that each cut is along a hyperplane containing a facet of one of the input objects.
- **Axis-aligned BSP**: a class of BSPs obtained by imposing the constraint that each cut is orthogonal to a coordinate axis.
- **Round-robin BSP:** An axis-aligned BSP in $\mathbb{R}^d$ where any d consecutive recursive cuts are along hyperplanes orthogonal to the d coordinate axes.
- **Tiling** in $\mathbb{R}^d$: a set of interior-disjoint polyhedra that partition $\mathbb{R}^d$.
- **Axis-aligned tiling**: a set of full-dimensional boxes that partition $\mathbb{R}^d$.
- d**-dimensional box**: the cross product of d real-valued intervals.

Key Results

The theoretical study of BSPs was initiated by Paterson and Yao [12, 13].

Line segments in the plane. A classical result of Paterson and Yao [12] is a simple and elegant randomized algorithm, which, given n disjoint segments, produces a BSP whose expected size is $O(n \log n)$; see also [3, Ch. 12]. It was widely believed for decades that every set of n disjoint line segments in the plane admits a BSP of size $O(n)$, see e.g. [12, p. 502]; this was until Tóth proved a tight super-linear bound for this problem, first by constructing a set of segments for which any BSP must have size $\Omega(n \log n / \log\log n)$, and later by matching this bound algorithmically:

Theorem 1. [14, 17] *Every set of n disjoint line segments in the plane admits a BSP of size $O(n \log n / \log\log n)$. This bound is the best possible, and a BSP of this size can be computed using $O(n \log^2 n)$ time.*

$(d-1)$-dimensional simplices in $\mathbb{R}^d$. The randomized partition technique of Paterson and Yao generalizes to higher dimensions yielding the following.

Theorem 2. [12] *Every set of n $(d-1)$-dimensional simplices in $\mathbb{R}^d$, where $d \geq 3$ admits a BSP of size $O(n^{d-1})$.*

While there exist n disjoint triangles in $\mathbb{R}^3$ that require a BSP of size $\Omega(n^2)$ [7], no super-quadratic lower bound is known in any dimension d. Near-linear upper bounds are known for "realistic" input models in $\mathbb{R}^3$ such as uncluttered scenes [5] or fat axis-aligned rectangles [16].

Axis-parallel segments, rectangles, and hyperrectangles.

Theorem 3. [1, 8, 12] *Every set of n pairwise disjoint axis-parallel line segments in the plane admits an auto-partition of size at most $2n-1$. Such a BSP can be computed using $O(n \log n)$ time and space, and has the additional property that no input segment is cut more than once. The upper bound on the size is the best possible apart from lower order terms.*

Theorem 4. [8, 13] *Let Γ be a collection of n line segments in $\mathbb{R}^d$, where $d \geq 3$, consisting of n_i segments parallel to the x_i-axis, for $i = 1, \ldots, d$. Then Γ admits a BSP of size at most*

$$4^{1/(d-1)}(d-1)(n_1 n_2 \cdots n_d)^{1/(d-1)} + 2n.$$

Theorem 5. [8] *For constants $1 \leq k \leq d-1$, every set of n axis-parallel k-rectangles in d-space admits an axis-aligned BSP of size $O(n^{d/(d-k)})$. This bound is the best possible for $k < d/2$ apart from the constant factor.*

For $k \geq d/2$, the best known upper and lower bounds do not match. No super-quadratic lower bound is known in any dimension d. In $\mathbb{R}^4$, Dumitrescu et al. [8] constructed n 2-dimensional disjoint rectangles whose partition complexity is $\Omega(n^{5/3})$.

Tilings. Already in the plane, the worst case partition complexity of axis-aligned tilings is smaller than that for disjoint boxes. Berman, DasGupta, and Muthukrishnan [6] showed that every axis-aligned tiling of size n admits an axis-aligned BSP of size at most $2n$; apart from lower order terms this bound is the best possible. For higher dimensions, Hershberger, Suri, and Tóth obtained the following result.

Theorem 6. [11] *Every axis-aligned tiling of size n in $\mathbb{R}^d$, where $d \geq 2$, admits a round-robin BSP of size $O(n^{(d+1)/3})$. On the the hand, there exist tilings of size n in $\mathbb{R}^d$ for which every BSP has size $\Omega(n^{\beta(d)})$, where $\beta(3) = 4/3$, and $\lim_{d\to\infty} \beta(d) = (1+\sqrt{5})/2 \approx 1.618$.*

In dimensions $d = 3$, the partition complexity of axis-aligned tilings of size n is $O(n^{4/3})$, which is tight by a construction of Hershberger and Suri [10].

Applications

The initial and most prominent applications are in computer graphics: BSPs support fast *hidden-surface removal* and *ray tracing* for moving viewpoints [9, 12]. Rendering is used for visualizing spatial opaque surfaces on the screen. A common and efficient rendering technique is the so-called *painter's algorithm*. Every object is drawn sequentially according to the *back-to-front* order, starting with the deepest object and continuing with the objects closer to the viewpoint. When all the objects have been drawn, every pixel represents the color of the object closest to the viewpoint. Further computer graphics applications include *constructive solid geometry* and *shadow generation.* Other applications of BSP trees include *range counting*, *point location*, *collision detection*, *robotics*, *graph drawing*, and *network design*; see for instance [15] and the references therein.

In the original setting, the input objects of the BSP were assumed to be static. Recent research on BSPs for moving objects can be seen in the context of *kinetic data structures* (KDS) of Basch, Guibas, and Hershberger [2]. In this model, objects move continuously along a given trajectory (flight plan), typically along a line or a low degree algebraic curve. The splitting hyperplanes are defined by faces of the input objects, and so they move continuously, too. The BSP is updated only at discrete *events*, though, when the combinatorial structure of the BSP changes.

Open Problems

◇ What is the maximum partition complexity of n disjoint $(d-1)$-dimensional simplices in $\mathbb{R}^d$ for $d \geq 3$?

◇ What is the maximum partition complexity of n disjoint (axis-aligned) boxes in $\mathbb{R}^d$ for $d \geq 3$?

◇ What is the maximum (axis-aligned) partition complexity of a tiling of n axis-aligned boxes in $\mathbb{R}^d$ for $d \geq 4$?

◇ Are there families of n disjoint objects in $\mathbb{R}^d$ whose partition complexity is super-quadratic in n?

◇ How many combinatorial changes can occur in the kinetic BSP of n points moving with constant velocities in the plane?

In all five open problems, the dimension $d \in \mathbb{N}$ of the ambient space $\mathbb{R}^d$ is constant, and asymptotically tight bounds in terms of n are sought.

Cross-References

Kinetic Data Structures

Recommended Reading

1. F. d'Amore and P. G. Franciosa, On the optimal binary plane partition for sets of isothetic rectangles, *Inform. Process. Lett.* **44** (1992), 255–259.
2. J. Basch, L.J. Guibas, and J. Hershberger, Data structures for mobile data, *J. Algorithms* **31(1)** (1999), 1–28.
3. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*, 3rd edition, Springer Verlag, 2008.
4. M. de Berg, M. de Groot, and M.H. Overmars, Perfect binary space partitions, *Comput. Geom. Theory Appl.* **7** (1997), 81–91.
5. M. de Berg, M.J. Katz, A.F. van der Stappen, and J. Vleugels, Realistic input models for geometric algorithms, *Algorithmica* **34** (2002), 81–97.
6. P. Berman, B. DasGupta, and S. Muthukrishnan, On the exact size of the binary space partitioning of sets of isothetic rectangles with applications, *SIAM J. Disc. Math.* **15(2)** (2002), 252–267.
7. B. Chazelle, Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm, *SIAM J. Computing* **13** (1984), 488–507.
8. 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.

9. H. Fuchs, Z. M. Kedem, and B. Naylor, On visible surface generation by a priori tree structures, *Comput. Graph.* **14(3)** (1980), 124–133.
10. J. Hershberger and S. Suri, Binary space partitions for 3D subdivisions, in *Proc. 14th ACM-SIAM Sympos. on Discrete Algorithms*, ACM Press, 2003, pp. 100–108.
11. J. Hershberger, S. Suri and Cs. D. Tóth, Binary space partitions of orthogonal subdivisions, *SIAM J. Computing* **34(6)** (2005), 1380–1397.
12. 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.
13. M. S. Paterson and F. F. Yao, Optimal binary space partitions for orthogonal objects, *J. Algorithms* **13** (1992), 99–113.
14. Cs. D. Tóth, A note on binary plane partitions, *Discrete Comput. Geom.* **30** (2003), 3–16.
15. Cs. D. Tóth, Binary space partitions: recent developments, in *Combinatorial and Computational Geometry (J.E. Goodman, J. Pach, and E. Welzl, eds.)*, pp. 529–556, vol. 52 of MSRI Publications, Cambridge Univ. Press, Cambridge, 2005.
16. Cs. D. Tóth, Binary space partition for axis-aligned fat rectangles, *SIAM J. Comput.* **38(1)** (2008), 429–447.
17. Cs. D. Tóth, Binary plane partitions for disjoint line segments, *Discrete Comput. Geom.* **45(4)** (2011), 617–646.