Local Polyhedra and Geometric Graphs*

Jeff Erickson†

University of Illinois at Urbana-Champaign
jeffe@cs.uiuc.edu
http://www.cs.uiuc.edu/~jeffe

ABSTRACT

We introduce a new realistic input model for geometric graphs and nonconvex polyhedra. A geometric graph G is *local* if (1) the longest edge at every vertex v is only a constant factor longer than the distance from v to its Euclidean nearest neighbor and (2) the lengths of the longest and shortest edges differ by at most a polynomial factor. A polyhedron is local if all its faces are simplices and its edges form a local geometric graph. We show that any boolean combination of any two local polyhedra in $\mathbb{R}^d$, each with n vertices, can be computed in $O(n \log n)$ time, using a standard hierarchy of axis-aligned bounding boxes. Using results of de Berg, we also show that any local polyhedron in $\mathbb{R}^d$ has a binary space partition tree of size $O(n \log^{d-1} n)$. Finally, we describe efficient algorithms for computing Minkowski sums of local polyhedra in two and three dimensions.

Categories and Subject Descriptors: F.2.2 [Nonnumerical Algorithms and Problems]: Geometrical problems and computations; G.2.m [Discrete Mathematics]: Miscellaneous—*Discrete Geometry*; I.3.5 [Computational Geometry and Object Modeling]: Object hierarchies

General Terms: Algorithms, Theory

Keywords: realistic input models, collision detection, bounding volume hierarchies, binary space partition

1. INTRODUCTION

Nonconvex polyhedra are ubiquitous in computer graphics, solid modeling, computer aided design and manufacturing, robotics, and other geometric application areas. Unlike nonconvex polygons or convex objects in space, for which many problems can be solved easily, polyhedra are notoriously difficult to handle efficiently, at least in the worst case.

Collision detection is a textbook example of a problem that is relatively easy for polygons but hard for polyhedra. The fastest algorithm for deciding whether two static nonconvex polyhedra intersect, due to Pellegrini, runs in time $O(n^{8/5+\varepsilon})$ [31]. For polyhedral terrains, the time bound can be improved to $O(n^{4/3+\varepsilon})$ [7]. Pellegrini's algorithm was generalized by Schömer and Thiel [36] to find the first collision between two translating polyhedra in time $O(n^{8/5+\varepsilon})$, or between two rotating polyhedra in time $O(n^{5/3+\varepsilon})$. To avoid directly checking all $\Omega(n^2)$ edge pairs, these algorithms employ complex multilevel range-searching data structures that would be difficult (if not impossible) to implement efficiently. Erickson [11] proved that the polyhedral intersection problem is at least as hard as Hopcroft's problem: given a set of points and lines in the plane, does any point lie on a line? The main idea of the reduction is to replace each point and line with an arbitrarily thin spike. In light of this reduction and Erickson's $\Omega(n^{4/3})$ lower bound for Hopcroft's problem [12], an algorithm that detects intersections in $o(n^{4/3})$ worst-case time appears unlikely.

In practice, one of the most popular techniques for intersection detection uses a *hierarchy of bounding volumes*. For a given placement of two disjoint polyhedra, the algorithms refine their hierarchies only to the coarsest level at which the resulting bounding volumes are disjoint. Beginning with Guttmann's introduction of the R-tree in the early 1980s [17], several types of bounding volume hierarchies have been proposed and implemented [1, 15, 16, 19, 20, 22, 24]. Unfortunately, all of these methods—in fact, any related method that uses a hierarchy of convex bounding volumes—can be forced to spend $\Omega(n^2)$ time to determine whether two n-vertex polyhedra intersect. The worst-case example consists of two polyhedra whose edges approximate a twisted grid, similar to a construction of Chazelle [6, 29].

Since worst-case efficient algorithms for detecting intersections seem unlikely, many authors have analyzed heuristics under the assumption that the input objects satisfy certain realistic constraints. For example, Suri and others have shown that for large collections of objects, a standard bounding box heuristic culls out most non-intersecting pairs, provided the objects are fat (at least on average) and all about the same size [35, 43]. Guibas *et al.* [16] and independently Lotan *et al.* [22] recently showed that in

*See http://www.cs.uiuc.edu/~jeffe/pubs/local.html for the most recent version of this paper.

†Partially supported by a Sloan Foundation Fellowship, NSF CAREER award CCR-0093348, and NSF ITR grants DMR-0121695 and CCR-0219594. Portions of this work were done while the author was visiting Duke University.

SoCG'03, June 8–10, 2003, San Diego, California, USA.

a certain hierarchy of bounding spheres for well-behaved necklaces of balls, only $O(n^{4/3})$ pairs of balls can intersect. Haverkort *et al.* [18] also recently showed that storing a set of boxes with *low slicing number* in a certain bounding box hierarchy allows box-intersection and approximate range queries to be answered in polylogarithmic time.

This paper introduces a new realistic input model for nonconvex polyhedra, which we call *locality.* We prove that standard bounding volume hierarchy techniques can be used to detect whether two local polyhedra of any fixed dimension intersect in $O(n \log n)$ time. In fact, our algorithm can compute the intersection, union, or any other boolean combination of two local polyhedra in same time bound. Surprisingly, our model allows polyhedra that contain sharp spikes and folds; however, it forbids many long edges to be packed closely together.

The paper is organized as follows. In Section 2, we formally define our input model, prove some basic results, and relate local polyhedra to other realistic input models. We develop our intersection algorithm in Section 3. In Section 4, we show that our time bounds are asymptotically optimal; our lower bound construction also implies that two simple relaxations of our model allow the worst-case quadratic behavior of arbitrary polyhedra. In Section 5, we apply a result of de Berg [2] to show that any local polyhedron in $\mathbb{R}^d$ has a binary space partition tree of size $O(n \log^{d-1} n)$. We develop upper and lower bounds on the complexity of Minkowski sums of local polyhedra in low dimensions in Sections 6 and 7. Finally, we conclude by suggesting several open problems.

2. PRELIMINARIES

2.1 Definitions

A *geometric graph* $G = (V, E)$ is an undirected simple graph whose vertices V are distinct points in $\mathbb{R}^d$ and whose edges E are straight line segments. Planar straight-line graphs are examples of geometric graphs in the plane; however, the edges of a geometric graph are not required to have disjoint interiors. The vertices and edges of any (convex or non-convex) polyhedron or piecewise linear complex also form a geometric graph. The *size* $n(G)$ of a geometric graph G is the number of vertices.

Let $N(v)$ denote the set of neighbors of a vertex v in G. We define the *local scale factor* of a geometric graph G, denoted $\sigma(G)$, to be the maximum ratio between the length of the longest edge at a vertex v and the distance from that vertex v to its nearest Euclidean neighbor (which may not be a neighbor of v in the graph). Similarly, we define the *global scale factor* of G, denoted $\Sigma(G)$, as the ratio between the longest and shortest edge lengths in G:

$$\sigma(G) = \max_{v \in V} \frac{\max_{u \in N(v)} |uv|}{\min_{u \in V \setminus \{v\}} |uv|} \qquad \Sigma(G) = \frac{\max_{uv \in E} |uv|}{\min_{uv \in E} |uv|}$$

Intuitively, $\sigma(G)$ and $\Sigma(G)$ respectively bound the local and global variation in the 'scale' of the graph. We easily observe that $\Sigma(G) \leq \sigma(G)^{n(G)}$. We will write $n = n(G)$, $\sigma = \sigma(G)$, and $\Sigma = \Sigma(G)$ when the graph G is clear from context.

We call a geometric graph *local* if its local scale factor σ is less than some fixed constant and its global scale factor Σ is less than some fixed *polynomial* in the number of vertices. Local graphs are a generalization of *civilized* graphs [40], for which $\Sigma = O(1)$. The choice of the word "local" is meant to emphasize the much more important role of the local scale factor; all of our complexity bounds are polynomial in σ but at most polylogarithmic in Σ (for any fixed dimension).

A *polytope* is the convex hull of a finite number of points. A *polyhedron* is the union of a finite number of polytopes, all of the same dimension. The boundary of any d-dimensional polyhedron Π is a $(d-1)$-dimensional manifold, comprised of several connected $(d-1)$-dimensional polyhedra called the *facets* of Π. A *face* of Π is either Π itself or a face of a facet of Π; the latter are called *proper faces* of Π. In particular, the empty set is the unique (-1)-dimensional face of every polyhedron. A polyhedron is *simplicial* if its facets (and thus its faces) are all simplices. A *boundary triangulation* of a polyhedron decomposes its facets into simplices that meet face to face, with no additional vertices, Finally, we say that a polyhedron is *local* if it has a boundary triangulation whose edges form a local geometric graph.

Most of our bounds for simplicial polyhedra apply immediately to 'simplex soup': arbitrary collections of simplices in $\mathbb{R}^d$, possibly with shared faces. Thus, for example, we can replace the word 'polyhedron' with 'mesh' or 'piecewise linear complex' or 'immersed manifold' in almost all our results with no other changes. The only exceptions are the bounding volume hierarchy time bounds in Section 3.2, which require that the *spread* of the vertices is bounded by a polynomial, and the bounds for BSP trees in Section 5, which increase by a logarithmic factor if the simplices do not have disjoint interiors.

2.2 Basic Properties

Lemma 2.1. *In any geometric graph G in $\mathbb{R}^d$, every vertex has degree at most $O(\sigma^d \log \sigma)$, and this bound is tight in the worst case.*

Proof: Let v be an arbitrary vertex of G. Without loss of generality, suppose that the distance from v to its nearest neighbor is exactly 1. We define a sequence of open nested balls $B_0 \subset B_1 \subset B_2 \subset \cdots$, all centered at v, where each ball B_i has radius 2^i. Every neighbor of v lies inside the ball $B_{\lceil \lg \sigma \rceil}$. Let A_i denote the annulus $B_i \setminus B_{i-1}$.

Consider an edge uv of G where $u \in A_i$. The distance from u to v is at least 2^{i-1}, which implies that the ball of radius $2^{i-1}/\sigma$ centered at u does not contain any other vertex of G. The intersection of this empty ball and A_i has volume $\Omega(2^{id}/\sigma^d)$. On the other hand, the volume of A_i is $O(2^{id})$. A straightforward packing argument implies that each annulus A_i contains at most $O(\sigma^d)$ neighbors of v.

For the matching lower bound, let v be an arbitrary point, pack as many points as possible into each annulus A_i, and let G be the graph connecting v to every other point. $\square$

Lemma 2.2. *Any n-vertex geometric graph G in $\mathbb{R}^d$ has at most $O(\sigma^d n)$ edges, and this bound is tight in the worst case.*

Proof: Let u and v be the closest pair of vertices in G, and without loss of generality, assume that $|uv| = 1$. Let B be a ball of radius σ centered at v. Every neighbor of v in the graph lies inside B and is the center of an empty unit ball. A straightforward packing argument now immediately implies that v has $O(\sigma^d)$ neighbors. Let G' be the graph obtained from G by deleting v and all its incident edges. We easily observe that $\sigma(G') \leq \sigma(G)$ and $\Sigma(G') \leq \Sigma(G)$. By the inductive hypothesis, G' has $(n-1) \cdot O(\sigma^d)$ edges. The trivial base case is a one-vertex graph.

For the matching lower bound, consider the graph whose vertices are the integer grid $\{1, 2, \ldots, m\}^d$, where $m = \lfloor n^{1/d} \rfloor$, with an edge between any two vertices whose Euclidean distance is at most σ. This graph clearly has $\Omega(m^d \sigma^d) = \Omega(n\sigma^d)$ edges. $\square$

Corollary 2.3. *Any n-vertex polyhedron in $\mathbb{R}^d$ has at most $O(\sigma^{d(d-1)}n)$ faces.*

Proof: The *star* of any vertex v is the set of faces that contain v, along with their lower-dimensional faces. The *link* of v is the the set of faces in the star of v that do not contain v. We define the *rank* of a polyhedron to be the maximum dimension of any face. Thus, a polyhedron with rank 1 is just a geometric graph. The link of any vertex in any polyhedron with rank k is a polyhedron with rank at most $k-1$.

We claim that any n-vertex polyhedron in $\mathbb{R}^d$ with rank k has $O(\sigma^{kd}n)$ faces. We prove our claim by induction on k. The base case $k = 0$ is trivial. Let Π be a polyhedron with rank $k \geq 1$. Every k-dimensional face is the convex hull of a vertex v and a $(k-1)$-dimensional face in the link of v. It follows that

$$\#\text{faces}(\Pi) \leq 2 \sum_v \#\text{faces}(\text{link}(v)).$$

(This bound is quite conservative; most low-dimensional faces will be in several links.) The inductive hypothesis implies that the link of v has $O(\sigma^{d(k-1)})$ faces, so

$$\#\text{faces}(\Pi) = O(\sigma^{d(k-1)}) \sum_v \deg(v),$$

which is $O(\sigma^{dk}n)$ by Lemma 2.2. $\square$

Lemma 2.4. *Let G be a geometric graph in $\mathbb{R}^d$, all of whose edges have length at least 1. At most $O(\sigma^d \log \Sigma)$ edges of G intersect any unit-width hypercube.*

Proof: Let $\square$ be a hypercube of unit width, and let $E_\square$ be the set of edges in G that intersect $\square$. We define a sequence of nested hypercubes $\square_1 \subset \square_2 \subset \cdots$, all concentric with $\square$, where each hypercube $\square_i$ has width $2^{i+1}+1$. We partition the edges in $E_\square$ into into disjoint *length classes* $E_1 \cup E_2 \cup \cdots$, where each set E_i contains all edges in $E_\square$ whose length is between 2^{i-1} and 2^i. Any edge in E_i has at least one endpoint in $\square_i$, and that endpoint is the center of an empty ball of radius at least $2^{i-1}/\sigma$. Thus, by a straightforward packing argument, $O(\sigma^d)$ edges in any length class E_i intersect $\square$. At most $\lg \Sigma$ of the length classes are nonempty. $\square$

Finally, let $G = (V, E)$ and $G' = (V', E')$ be geometric graphs (or possibly the same graph). We say that two edges $uv \in E$ and $u'v' \in E'$ are *α-close* if their Euclidean distance is less than α times the sum of their lengths:

$$\min_{x \in uv} \min_{x' \in u'v'} |xx'| < \alpha\big(|uv| + |u'v'|\big).$$

Two edges are *close* if they are 1-close.

Lemma 2.5. *Let G be a geometric graph in $\mathbb{R}^d$, all of whose edges have length at least 1. At most $O((1+\alpha)^d \times \sigma^{2d} \log \sigma \log \Sigma)$ edges of G are α-close to any line segment of length 1.*

Proof: Let s be a line segment of length 1. As in the previous lemma, we partition the edges of G into disjoint length classes $E_1 \cup E_2 \cup \cdots$, where each edge class E_i contains edges whose length is between 2^{i-1} and 2^i. If an edge $e \in E_i$ is α-close to s, then the distance between e and s is at most $\alpha + \alpha 2^i$. In particular, some endpoint of e is within distance $\alpha + \alpha 2^i + 2^{i-1}$ of segment s.

Let C_i denote the Minkowski sum of s with a ball of radius $\alpha + \alpha 2^i + 2^{i-1} = O((1+\alpha)2^i)$. Any edge $e \in E_i$ that is close to s must have at least one endpoint in C_i. We charge the close edge e to this endpoint. Each charged endpoint must lie at the center of an ball of radius $2^{i-1}/\sigma$ that contains no other vertex of G. Since the volume of C_i is $O((1+\alpha)^d 2^{id})$, a standard packing argument implies that at most $O((1+\alpha)^d \sigma^d)$ endpoints of edges in E_i are charged. Lemma 2.1 implies that each endpoint is charged $O(\sigma^d \log \sigma)$ times. Finally, at most $\lceil \lg \Sigma \rceil$ of the length classes E_i are nonempty. $\square$

Corollary 2.6. *Two geometric graphs G and G' have at most $O(n(1+\alpha)^d \sigma^{2d} \log \sigma \log \Sigma)$ α-close edge pairs, where $\sigma = \max\{\sigma(G), \sigma(G')\}$, $\Sigma = \max\{\Sigma(G), \Sigma(G')\}$, and $n = n(G) + n(G')$.*

In the remainder of this extended abstract, we will assume that σ is a fixed constant and Σ is a fixed polynomial in n, and we will omit explicit dependence on these parameters from our upper bounds.

2.3 Relationship to Other Input Models

Several different models of realistic or well-shaped geometric data have been proposed in the past [3, 4]. Perhaps the most well-known realistic input model is *fatness* [42]. An object X is fat if any ball centered inside X either contains X or has a constant fraction of its volume inside X. Thus, fat objects have no sharp spikes or folds. Local polyhedra, however, can have arbitrarily sharp features, and thus need not be fat; conversely, fat objects can have vertices with edges of arbitrarily different length, and thus need not be local. See Figure 1.

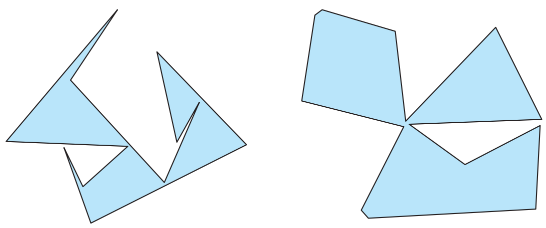

Figure 1. A local nonfat polygon, and a fat nonlocal polygon.

Another realistic input model, introduced by van der Stappen in the context of motion planning [34], is *low density*; see also [28, 32]. A set of objects have density λ if any ball of radius r intersects at most λ objects with diameter r or greater. Most bounds for low density scenes depend linearly on λ. Lemma 2.4 implies that a local polyhedron, viewed as a collection of facets, has density $O(\log n)$; thus, bounds for low-density environments apply to local environments with only a polylogarithmic penalty. On the other hand, low-density objects need not be local.

Two more realistic input models studied by de Berg *et al.* are *uncluttered* scenes and scenes with small *simple cover complexity* [4, 2]. Again, local polyhedra fit these models up to a logarithmic factor, but neither uncluttered nor easily-covered polyhedra are necessarily local. We discuss the connection between locality and clutter in more detail in Section 5.

We can also compare our model to quality metrics used for simplicial finite-element mesh generation. For example, Miller, Talmor, Teng, and others [21, 23, 33, 38, 40] define a triangulation to be *well-shaped* if the circumradius of each simplex is only a constant factor longer than the shortest edge of that simplex. Well-shaped triangulations have bounded local scale factor σ, but they are not necessarily local, since the global scale factor Σ could be exponential in the worst case. Conversely, local triangulations need not be well-shaped, even in two dimensions, since they can contain sharp angles. Talmor [38] proved that a well-shaped triangulation with n vertices, in any fixed dimension, has only $O(n)$ simplices. Corollary 2.3 implies that this linear upper bound actually holds for any triangulation whose local scale factor is bounded.

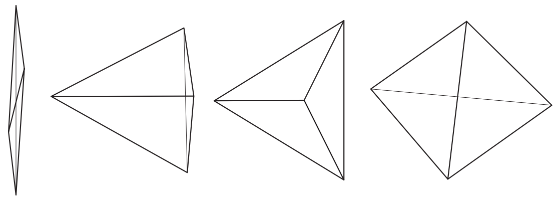

Figure 2. Local but badly-shaped tetrahedra. From left to right: spindle, wedge, cap, sliver.

3. INTERSECTING POLYHEDRA

3.1 Complexity

Intuitively, one of the reasons that collision detection is difficult for arbitrary nonconvex polyhedra in $\mathbb{R}^3$ is that two polyhedra can intersect, or nearly intersect, in a quadratic number of different locations [6, 29]. For local polyhedra, this quadratic behavior is impossible, even in higher dimensions.

Lemma 3.1. *If two simplices $\triangle$ and $\triangle'$ intersect, then at least one edge of $\triangle$ is close to at least one edge of $\triangle'$.*

Proof: Let uv and $u'v'$ be the longest edges of $\triangle$ and $\triangle'$, respectively, and let x be an arbitrary point in the intersection $\triangle \cap \triangle'$. We easily observe that the distance from any point in $\triangle$ to uv is at most $|uv|$. In particular, the distance from uv to x is at most $|uv|$. Similarly, the distance from $u'v'$ to x is at most $|u'v'|$. Thus, by the triangle inequality, uv and $u'v'$ are close. $\square$

Corollary 3.2. *Between any two local polyhedra in $\mathbb{R}^d$, each with at most n vertices, there are $O(n \log n)$ pairs of intersecting faces. Thus, any boolean combination of two local polyhedra has complexity $O(n \log n)$.*

Proof: Without loss of generality, we can assume that both polyhedra are simplicial; triangulating the boundary of each polyhedron can only increase the number of intersecting face pairs. Lemma 2.2 implies that each polyhedron has $O(n)$ edges, so by Corollary 2.6, there are $O(n \log n)$ close edge pairs. We charge each intersecting pair of faces to some close pair of edges with one edge from each face. By Lemma 2.1, each edge belongs to $O(1)$ faces. It follows that each close edge pair is charged $O(1)$ times. $\square$

Lemma 3.3. *If the axis-aligned bounding boxes of two simplices $\triangle$ and $\triangle'$ intersect, then at least one edge of $\triangle$ is close to at least one edge of $\triangle'$.*

Proof: Let $\square$ and $\square'$ denote the axis-aligned bounding boxes of $\triangle$ and $\triangle'$, respectively. Each facet of $\square$ contains at least one vertex of $\triangle$. Thus, there must be at least one edge e of $\triangle$ whose vertices lie on the furthest pair of parallel facets of $\square$. This edge is close to every point in $\square$, that is, the distance from e to any point in $\square$ is at most the length of e. Similarly, $\triangle'$ has at least one edge e' that is close to every point in the bounding box $\square'$. By the triangle inequality, e and e' are close. $\square$

Corollary 3.4. *Between any two local, simplicial polyhedra in $\mathbb{R}^d$, each with at most n vertices, there are $O(n \log n)$ pairs of faces whose axis-aligned bounding boxes intersect.*

Proof: Essentially the same as Corollary 3.2. $\square$

This corollary immediately suggests the following algorithm for detecting whether two local, simplicial polyhedra Π_1 and Π_2 intersect. Let B_1 and B_2 be the set of axis-aligned bounding boxes of facets of Π_1 and Π_2, respectively. Since each polyhedron has $O(n)$ facets, we can clearly calculate B_1 and B_2 in $O(n)$ time. Using multidimensional range trees and fractional cascading, we can find all pairs of intersecting boxes $(\square_1, \square_2) \in B_1 \times B_2$ in time $O(n \log^{d-1} n + k)$, where k is the number of intersecting pairs. Finally, for each pair of intersecting boxes, we can test in $O(1)$ time whether the corresponding pair of facets intersect. Corollary 3.4 implies that $k = O(n \log n)$, so the overall running time of this algorithm is $O(n \log^{d-1} n)$. This algorithm can be made extremely practical, at least in low dimensions, by combining it with simple heuristics [44]. In fact, we can actually compute the intersection, or any other boolean combination, of two local polyhedra within the same time bound, by performing an additional constant amount of work for each pair of intersecting facets, plus a constant number of point-in-polyhedron tests to handle the special case where no pair of facets intersects.

Theorem 3.5. *Any boolean combination of two local simplicial polyhedra in $\mathbb{R}^d$, each with n vertices, can be computed in $O(n \log^d n)$ time.*

3.2 The Graded Box-Tree

We can obtain an even faster intersection algorithm by constructing a bounding volume hierarchy, called a *graded box-tree*, for each polyhedron. A graded box-tree is (as usual) a rooted tree with constant degree, where the root corresponds to the entire polyhedron, and the leaves

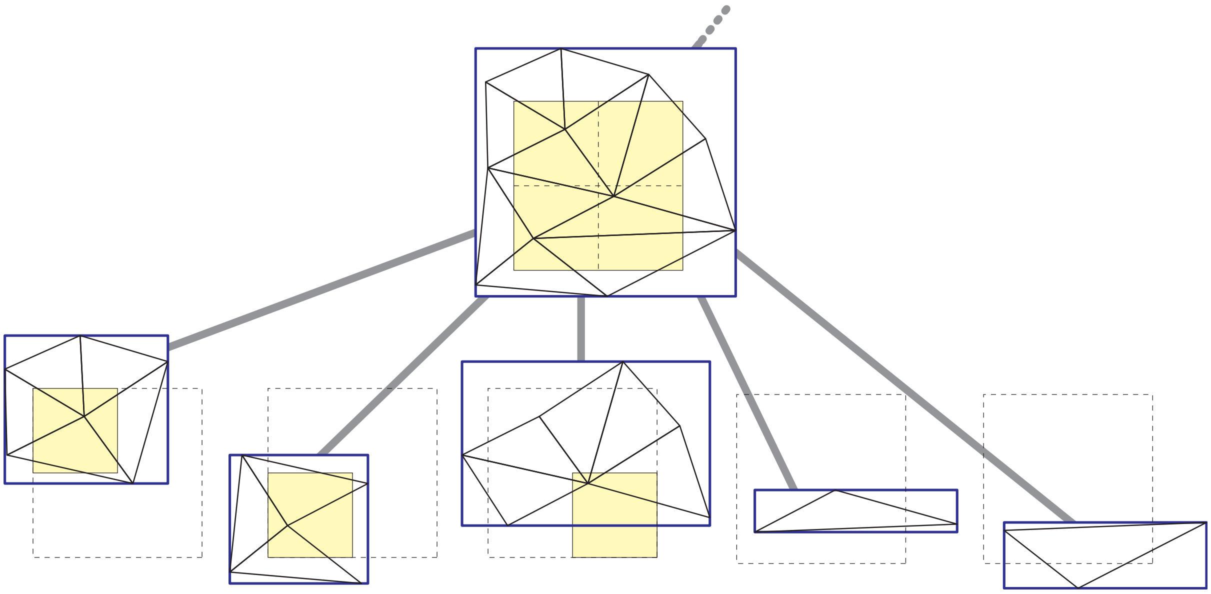

Figure 3. One node and its five children in the graded box-tree of a set of triangles in the plane. The shaded boxes are control cubes; the dashed boxes are the parent's control cube. See the text for further details.

correspond to individual facets. Each internal node v stores the axis-aligned bounding box of the facets (leaves) in its subtree. All the bounding volumes at the same level in a graded box-tree have approximately the same diameter. In the interest of simplifying the analysis, we will describe a bounding-volume hierarchy that can easily be improved in practice.

Let P be a local, simplicial polyhedron, or more generally, a local collection of $(d-1)$-simplices in $\mathbb{R}^d$, and let $\square(P)$ be a minimal axis-aligned bounding *cube* for P. To each internal node v in a graded box-tree, we associate an axis-aligned *control cube* $\square_v$ and a *facet set* P_v as follows. At the root, we have $\square_v = \square(P)$ and $P_v = P$. The control cubes are defined by a 2^d-tree of $\square(P)$—a quadtree in $\mathbb{R}^2$, an octtree in $\mathbb{R}^3$, and so forth. We emphasize that the control cubes are *not* the actual bounding volumes. For each internal node v, the facet set P_v contains the facets f of P such that

(1) f has at least one vertex inside $\square_v$, and

(2) the diameter of f is less than twice the width of $\square_v$.

A facet $f \in P_v$ is called an *outlier* if it is not a member of P_w for any child w of v, or equivalently, if its diameter is larger than the width of $\square_v$. Each outlier is attached to v as a new child, which becomes a leaf in our hierarchy. Because P is local, the proof of Lemma 2.4 implies that each node has $O(1)$ outlier children in addition to its 2^d subcube children. We easily verify that P_v is actually the set of facets that appear as leaves in the subtree rooted at v.

Finally, each node v actually stores the axis-aligned bounding box $\square(P_v)$ of its associated facet set. For internal nodes, this bounding box always intersects the corresponding control cube, but neither box is necessarily contained in the other.

Each facet is stored at most d times, once for each of its vertices. If we remove nodes w whose subsets P_w are empty and compress paths with no branches to single edges, we obtain a tree with constant degree, $O(dn)$ leaves, and therefore $O(dn)$ internal nodes.

An example of our construction is shown in Figure 3. The top of the figure shows a control square (shaded) and the bounding box for eleven triangles. This node has three normal children (northwest, southwest, and southeast) and two outliers. Because none of the vertices lie in the northeast quadrant of the control square, there is no northeast child. Two triangles belong to more than one child, since they have vertices in more than one quadrant of the original control square.

Theorem 3.6. *Given two local simplicial polyhedra P and Q, each with n vertices, we can determine whether they intersect in $O(n \log n)$ time.*

Proof: The diameter of any local polyhedron is at most $n\Sigma$ times the length of its shortest edge. Thus, the graded box-tree of any local polyhedron has has depth at most $\log_2(n\Sigma) = O(\log n)$. It follows that we can construct a graded box-tree for any local polyhedron in $O(n \log n)$ time.

Given their graded box-trees, we can decide whether P and Q intersect in the following standard fashion. If $\square(P)$ and $\square(Q)$ are disjoint, we can halt immediately. Otherwise we replace the larger of the two boxes, say $\square(P)$, with its $O(1)$ children, and recursively check for intersections between each of those children and Q. The base case for the recursion is two individual facets, which we can test for intersection in $O(1)$ time. The running time of this algorithm is clearly dominated by the number of recursive calls, each of which is caused by an intersection between two bounding boxes. Thus, to complete the proof, it suffices to show that between the two graded box-trees, there are only $O(n \log n)$ intersecting pairs of bounding boxes.

Let v be a node in P's tree, and let w be a node in Q's tree, such that the bounding boxes $\square(P_v)$ and $\square(Q_w)$ intersect. We charge this intersection to whichever node has the smaller control cube, say v. The bounding box $\square(P_v)$ is at most three times as wide in each direction as the corresponding control cube $\square_v$. A straightforward packing argument now implies that v is charged $O(1)$ times for each level of Q's hierarchy. We conclude that each bounding box in one graded box-tree intersects at most $O(\log n)$ larger boxes in the other graded box-tree. $\square$

As noted earlier, it is easy to modify this algorithm to actually compute any boolean combination of the two polyhedra in the same asymptotic running time. Almost any type of bounding volume can be used in place of axis-aligned boxes in our hierarchy with no loss of efficiency. The only requirement is that the diameter of each bounding volume is at most a constant factor larger than the diameter of the set of objects it encloses. Similarly, the underlying 2^d-tree can be replaced by any recursive decomposition into fat regions.

Finally, we observe that all the arguments in this section apply directly to local simplicial meshes, local self-intersecting polyhedra, or more generally, any local 'simplex soup' whose edge graph is connected. For disconnected sets of simplices, however, our running-time analysis requires that the *spread* of the vertices—the ratio between the largest and smallest pairwise distance [13]—is bounded by a polynomial in n.

4. THE HARPSICORDION

We now show that the results from the previous section are asymptotically optimal for polyhedra in $\mathbb{R}^3$ by constructing a pair of local polyhedra that intersect in $\Omega(n \log n)$ distinct points. Our lower bound construction also implies that our near-linear upper bounds do *not* hold under two obvious relaxations of our input model.

Our bad examples are variants of *Chazelle's polyhedron*, which was originally used to prove quadratic lower bounds for convex decomposition [6, 29]. Our version of Chazelle's construction consists of two polyhedra P and Q, each with total complexity $O(n)$. Each of these two polyhedra contains n edges on the saddle surface $z = xy$. Specifically, the 'vertical' saddle edges of P lie on the lines $z = iy$ for integers $1 \le i \le n$, and the 'horizontal' saddle edges of Q lie on the lines $z = xj$ for integers $1 \le j \le n$. Otherwise, P lies entirely above the saddle and Q lie entirely below. P and Q touch at $\Omega(n^2)$ distinct points of the form (i, j, ij). If we build bounding volume hierarchies for P and Q, the standard intersection algorithm must examine $\Omega(n^2)$ leaf pairs, no matter what shape the bounding volumes have.

A simple modification gives us a pair of *disjoint* polyhedra with similar worst-case behavior. Let $P^+ = P + (0, 0, \varepsilon)$ and $Q^- = Q - (0, 0, \varepsilon)$ be translations of P and Q away from the saddle, where $\varepsilon = O(1/n^2)$ is an arbitrarily small positive real number. Chazelle [6] proved that any convex decomposition of $\mathbb{R}^3 \setminus (P^+ \cup Q^-)$ has $\Omega(n^2)$ cells. We easily observe that the convex hull of any two saddle edges of P^+ intersects every saddle edge of Q^-. Thus, for any hierarchy of *convex* bounding volumes for P^+ and Q^-, there are $\Omega(n^2)$ pairs of intersecting bounding volumes, so the standard intersection algorithm requires $\Omega(n^2)$ time to prove that P^+ and Q^- are disjoint.

Theorem 4.1. *For any sufficiently large n, σ, and Σ, there are two n-vertex polyhedra P and Q, where $\sigma(P) = \sigma(Q) = \sigma$ and $\Sigma(P) = \Sigma(Q) = \Sigma$, that intersect in $\Omega(n\sigma \log \Sigma) = \Omega(n \log n)$ distinct points.*

Proof: In Chazelle's original construction, the edges that meet along the saddle are all roughly the same length, but this is clearly not necessary. Instead, we use two parallel sets of line segments of exponentially decaying length, resembling the strings of a harpsichord; specifically, the ith segment in each set has length $(2+\sigma)^{-i}$. See Figure 4. These segments can be placed arbitrarily close together without increasing the local scale factor above σ. If each set has m segments, the global scale factor of each set is $m^{(2+\sigma)/\sigma}$. Thus, for any desired σ and Σ, we can construct two local sets of m segments that meet in an $m \times m$ grid, where

$$m = \frac{\ln \Sigma}{\ln(2+\sigma) - \ln \sigma} = \frac{\ln \Sigma}{\ln(1 + \frac{2}{\sigma})} > \frac{\sigma \ln \Sigma}{2}.$$

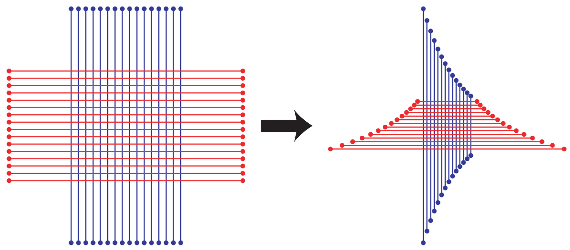

Figure 4. Replacing a regular grid with a harpsichord grid.

We now construct a local $O(m)$-vertex polyhedron with m edges on the saddle surface $z = xy$. Since our polyhedron resembles an accordion, with several saddle edges resembling the strings of a harpsichord, we call it a *harpsicordion*.

The harpsicordion is built by gluing together several copies of two local triangulated annuli $A^\sharp$ and $A^\flat$, shown in Figure 5. The annuli have the same convex outer boundary. The holes are similar triangles; the hole in $A^\flat$ is $(2+\sigma)/\sigma$ times as large as the hole in $A^\sharp$. Moreover, if we scale $A^\sharp$ by a factor of $(2+\sigma)/\sigma$ and align the two holes, the bottom edges of the annuli are collinear.

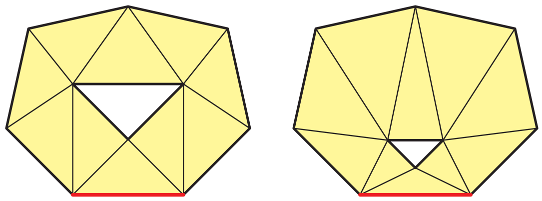

Figure 5. The template annuli $A^\sharp$ and $A^\flat$.

The harpsicordion consist of a sequence of m *folds*. Each fold is built by gluing a copy of $A^\sharp$ to a copy of $A^\flat$ along their common outer boundary, and translating the holes slightly away from the plane through this outer boundary. Successive folds, which differ in size by a factor of $(2+\sigma)/\sigma$, are glued together along a common hole boundary. We fill the holes in the first and last annuli with triangles to close the polyhedron. Finally, we slightly tilt the bottom edges of the folds to lie on the saddle surface. If $A^\sharp$ and $A^\flat$ are constructed carefully, the resulting polyhedron has local scale factor σ and global scale factor Σ, no matter how thin we make the folds.

Finally, polyhedra P and Q each consist of $O(n/m)$ harpsicordia, positioned so that each harpsicordion in P

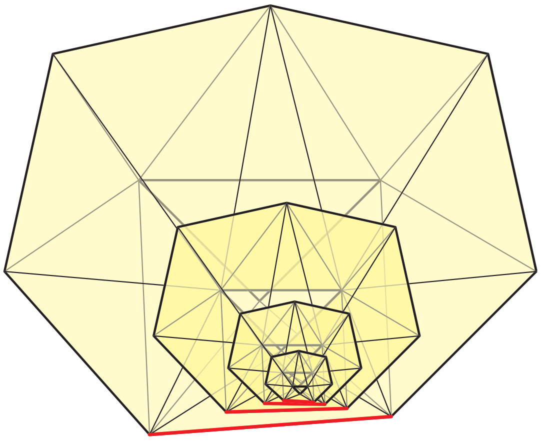

Figure 6. A harpsicordion with four folds.

meets one harpsicordion in Q in m^2 distinct points in some saddle. The total number of intersection points is $nm = \Omega(n\sigma \log \Sigma)$. We can make P and Q connected by adding prisms between pairs of harpsicordia. $\square$

Similar collections of harpsicordia can be used to prove the following lower bounds.

Theorem 4.2. *There are two disjoint, local, simplicial, n-vertex polyhedra in $\mathbb{R}^3$, such that for any hierarchy of convex bounding volumes, the standard intersection algorithm requires $\Omega(n \log n)$ time.*

Theorem 4.3. *There is a local, simplicial, n-vertex polyhedron P in $\mathbb{R}^3$ such that any convex decomposition of $\mathbb{R}^3 \setminus P$ has $\Omega(n \log n)$ cells.*

Our lower bound construction implies that both the local and global scale factors must be bounded in order to obtain our near-linear upper bounds. Specifically, we obtain pairs of n-vertex polyhedra with $\Omega(n^2)$ intersection points either by setting $\sigma = \Omega(n)$ and $\Sigma = 2$, or by setting $\Sigma = 2^{\Omega(n)}$ and $\sigma = 4$. In fact, for the case $\sigma = \Omega(n)$, we can simplify our lower bound construction by using a single annulus to construct a non-local accordion, as shown in Figure 7.

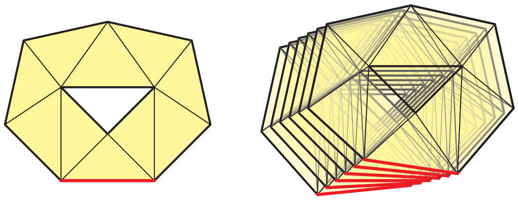

Figure 7. A non-local accordion and its template annulus.

5. BINARY SPACE PARTITIONS

A *binary space partition tree*, or *BSP*, is a binary tree where every internal node v has an associated *cutting hyperplane* h_v in $\mathbb{R}^d$. We can recursively associate an open convex polyhedral *cell* Δ_v with every node v in a BSP as follows. The cell associated with the root is $\mathbb{R}^d$. If u and w are the children of some internal node v, then $\Delta_u = \Delta_v \cap h_v^+$ and $\Delta_w = \Delta_v \cap h_v^-$, where h_v^+ and h_v^- are the open halfspaces bounded by the cutting hyperplane h_v. The (closures of the) leaf cells of a BSP form a convex decomposition of $\mathbb{R}^d$. We say that a BSP $\mathcal{B}$ *respects* a polyhedron Π—or less formally, that $\mathcal{B}$ is a BSP for Π—if no facet of Π intersects the interior of any leaf cell of $\mathcal{B}$. The *size* of a BSP is the number of cuts, or equivalently, one more than the number of leaf cells.

Fuchs *et al.* [14] introduced BSP trees, following earlier work by Schumacker *et al.* [37], as a tool for computing depth orders for rendering. Since their introduction, BSP trees have been used for many other applications in computer graphics, including shadow generation [8, 9], solid modeling [26, 41], geometric data repair [25], and visibility culling for interactive walkthroughs [39].

As in the case for intersection detection, the worst-case complexity bounds for BSPs of polyhedra in $\mathbb{R}^3$ are quite pessimistic. Chazelle's polyhedron, described in the previous section, gives an $\Omega(n^2)$ lower bound in general [29]. For orthogonal polyhedra in $\mathbb{R}^3$, a construction of Thurston gives a lower bound of $\Omega(n^{3/2})$ [30]. In both cases, matching upper bounds were first proved by Paterson and Yao [29, 30]. In fact, Paterson and Yao's techniques imply that any collection of n simplices in $\mathbb{R}^d$ has a BSP of size $O(n^{d-1})$.

De Berg [2] defines the *clutter factor* κ of a set of objects to be the largest number of objects that intersect a hypercube that does not contain a vertex of the axis-aligned bounding box of any object. He also proved that any scene of n objects whose clutter factor is constant has a BSP of linear size. By adapting de Berg's results, we show that any local polyhedron has a BSP of near-linear size.

Lemma 5.1. *The facets of any local simplicial polyhedron in $\mathbb{R}^d$ have clutter factor $O(\log n)$.*

Proof: Let $\triangle$ be a simplex and let $\square$ be a hypercube of width w, such that $\triangle$ and $\square$ intersect, but no vertex of the bounding box of $\triangle$ lies inside $\square$. The longest edge of $\triangle$ must have length greater than w. Following the proof of Lemma 3.3, we conclude that this edge must be close to some edge of $\square$.

Let Π be a local simplicial polyhedron, such that no facet of Π has a bounding box vertex inside $\square$. We charge each facet that intersects $\square$ to its longest edge, which is close to some edge of $\square$ by the previous argument. Lemma 2.5 implies that at most $O(\log n)$ edges of Π are close to any edge of $\square$, and by Lemma 2.1, each edge belongs to $O(1)$ facets. Finally, $\square$ has $d2^{d-1} = O(1)$ edges. $\square$

Theorem 5.2. *Any local, simplicial polyhedron in $\mathbb{R}^d$ with n vertices has a BSP of size $O(n \log^{d-1} n)$, which can be constructed in $O(n \log^{d-1} n)$ time.*

Proof: De Berg [2] describes a two-level BSP of linear complexity for any uncluttered collection of n objects. The first level is an orthogonal BSP of size $O(n)$ that covers the vertices of the bounding boxes of the objects; this bound does not depend at all on the clutter factor. Moreover, each leaf cell in this orthogonal BSP can be covered by $O(1)$ hypercubes that contain no bounding box vertices. This orthogonal BSP can be constructed in $O(n \log n)$ time.

Let Π be a local simplicial polyhedron with n vertices. Lemma 5.1 implies that each leaf cell in the orthogonal BSP intersects $O(\log n)$ facets of Π. Using an auxiliary data structure, de Berg describes how to determine the k leaf cells that intersect any constant-complexity query range in $O(k \log n)$ time. Using this structure, we can determine the facets of Π that intersect each leaf cell in $O(n \log^2 n)$ time.

Finally, in $O(\log^{d-1} n)$ time, we can build a BSP of size $O(\log^{d-1} n)$ for the facets that intersect any leaf cell, using the BSP-construction algorithm of Paterson and Yao [29]. (Paterson and Yao claim a running time of $O(k^{d+1})$, where k is the number of input facets, but this can be reduced to $O(k^{d-1})$ using standard randomized techniques [10].) □

Theorem 4.3 implies that this bound is tight up to a factor of $O(\log n)$ when $d = 3$. All of these results apply directly to any local collection of *interior-disjoint* simplices. If we allow self-intersections, however, both bounds increase by a single logarithmic factor.

Theorem 5.3. *Any local set of n $(d-1)$-simplices in $\mathbb{R}^d$ has a BSP of size $O(n \log^d n)$.*

Proof: Again we start with de Berg's linear-size orthogonal BSP. We can trivially construct a BSP of size $O(\log^d n)$ for the $O(\log n)$ simplices that intersect each leaf cell, in $O(\log^d n)$ time, by incrementally constructing the arrangement of planes through those simplices. □

Theorem 5.4. *For any n and d, there is a local set X of n $(d-1)$-dimensional simplices in $\mathbb{R}^d$ such that any BSP for X has $\Omega(n \log^{d-1} n)$ cells.*

Proof: Generalizing the planar harpsichord grid, we can construct a local set of $\log n$ simplices that intersect in a regular cubical grid. The complement of the union of these simplices has $\Omega(\log^d n)$ connected components. Collecting $n/\log n$ copies of this set, we obtain a local set of n simplices whose complement has $\Omega(n \log^{d-1} n)$ connected components. Any BSP has at least one leaf cell in each component. □

6. LARGER COMBINATIONS

Unions and intersections of local polyhedra are not necessarily local. Thus, if we want to efficiently construct a boolean combination of more than two local polyhedra, we cannot combine the objects in pairs; we must combine everything at once.

Theorem 6.1. *Any boolean combination of r local, simplicial polyhedra in $\mathbb{R}^3$, each with n vertices, has complexity $O(r^3 n \log^2 n)$.*

Proof: Let $P_1, P_2, \ldots, P_r$ be local polyhedra. We will show that the *arrangement* of these polyhedra has total complexity $O(r^3 n \log^2 n)$; any boolean combination of the polyhedra consists of a subset of the faces of the arrangement, possibly with some faces merged together. We can analyze the complexity of this arrangement using an argument similar to Paterson and Yao's analysis of their three-dimensional binary space partition trees [29].

First we count the vertices of the arrangement. Each arrangement vertex is either a vertex of a polyhedron P_i, the intersection of the edge of some polyhedron P_i with a facet of another polyhedron P_j, or the mutual intersection of three facets of three different polyhedra. There are clearly rn vertices of the first type, and Corollary 3.2 implies there are $O(r^2 n \log n)$ vertices of the second type. We charge each triple intersection point to the triangle whose longest edge is shortest among the three intersectors. By Lemma 3.1, the longest edge of the charged triangle is close to the longest edges of the other two triangles.

Consider three polyhedra P_i, P_j, P_k. Each polyhedron has $O(n)$ edges, and each edge lies on $O(1)$ facets. Each edge of P_i is close to $O(\log n)$ longer edges in P_j and $O(\log n)$ longer edges in P_j. Thus, each edge of P_i is charged $O(\log^2 n)$ times by triple intersections with P_j and P_k. Since there are $O(r^2)$ choices for j and k, each edge of P_i is charged $O(r^2 \log^2 n)$ times, so P_i is charged $O(r^2 n \log^2 n)$ times altogether. We conclude that the total number of triple intersections, and thus the total number of vertices, is $O(r^3 n \log^2 n)$.

The edges of the arrangement can be grouped into collinear *super-edges*, where each super-edge is either an edge of some polyhedron P_i, or the intersection of two facets of different polyhedra. Corollary 3.2 implies that there are $O(k^2 n \log n)$ super-edges. To count the actual arrangement edges, we charge each edge to one of its endpoints. Each triple intersection point is charged at most six times; all the remaining charges go to endpoints of super-edges. Thus, the total number of edges is $O(r^3 n \log^2 n)$.

Finally, each facet of each polyhedron P_i is decomposed into several arrangement facets, which we will call *fragments*, by the other polyhedra. Let F be a facet of P_i. Euler's formula implies that the number of fragments of F is less than $2v_F - 4$, where v_F is the number of arrangement vertices on F. If F charges two of its fragments to every vertex on F except its original vertices from P_i, only 2 fragments are uncharged. Except for polyhedra vertices, each vertex in the arrangement lies on exactly three polyhedron facets and thus is charged at most six times. Therefore, the total number of fragments is twice the number of facets plus six times the number of vertices, which by our earlier analysis is $O(r^3 n \log^2 n)$. □

Theorem 6.2. *We can compute any boolean function of any r local, simplicial polyhedra in $\mathbb{R}^3$, each with n vertices, in time $O(r^3 n \log^2 n)$.*

7. MINKOWSKI SUMS

Finally, we consider the complexity of Minkowski sums of local polyhedra in two and three dimensions. In the worst case, the Minkowski sum of two n-gons has complexity $\Theta(n^4)$, and the Minkowski sum of two polyhedra in $\mathbb{R}^3$, each with n faces, has complexity $\Theta(n^6)$.

Theorem 7.1. *The Minkowski sum of any two local n-gons in the plane has complexity $O(n^3 \log n)$ and $\Omega(n^3)$ in the worst case.*

Proof: Let P and Q be two local n-gons with vertices labeled $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$, respectively. The Minkowski sum $P + Q$ is the union of cells in the arrangement of the $r = 2n$ polygons $p_i + Q$ and $P + q_j$. Simplifying the proof of Theorem 6.1 to the two-dimensional case, we can prove that the arrangement of r local n-gons has complexity $O(r^2 n \log n)$.

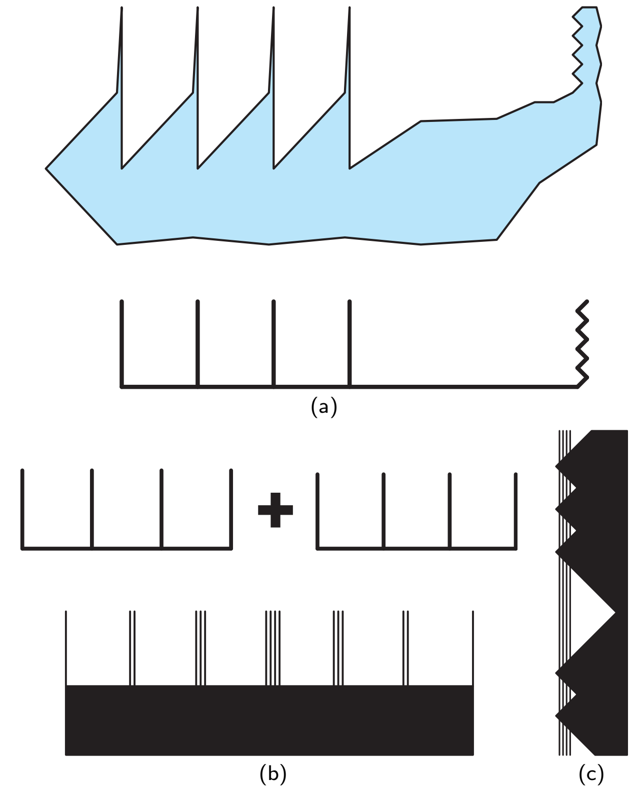

Figure 8. (a) One comb polygon and its salient features. (b) The Minkowski sum of two combs has several bundles of spikes. (c) The Minkowski sum of two zigzags cutting through a bundle.

For the lower bound, we construct two local $O(n)$-gons P and Q, with $\sigma \approx 2$ and $\Sigma = \Theta(n)$, whose Minkowski sum has complexity $\Omega(n^3)$. Each polygon consists of a comb with n widely-spaced extremely thin spikes, each of length $O(n)$, and a vertical zigzag of $2n$ edges, each of length 1. To maintain locality, a series of $O(\log n)$ edges interpolates between the large and small features of each polygon. Figure 8(a) shows the comb polygon, plus a simplified geometric graph with the same salient features.

The distance between the spikes in P is very slightly larger than in Q, so that the Minkowski sum of the two cones has n^2 spikes, grouped into $2n-1$ bundles. Similarly, the Minkowski sum of the two zigzags has n^2 teeth. The zigzags of P and Q are positioned so that their Minkowski sum cuts through the middle bundle of n spikes. Each of the n^2 teeth cuts all the way through this bundle, intersecting all n spikes. Thus, $P+Q$ has $\Omega(n^3)$ vertices. □

Theorem 7.2. *The Minkowski sum of any two local, simplicial polyhedra in $\mathbb{R}^3$ has complexity $O(n^4 \log^2 n)$ and $\Omega(n^4)$ in the worst case.*

Proof: Let P and Q be two local n-vertex polyhedra. The Minkowski sum $P+Q$ is the union of cells in the arrangement of the $r = 2n$ polyhedra $p_i + Q$ and $P + q_j$. The proof of Theorem 6.1 implies that this arrangement has complexity $O(r^3 n \log^2 n)$.

The lower bound construction is a generalization of the polygon case. Each polyhedron consists of two sets of 'shelves', one set parallel to the xz plane and one set parallel to the yz plane, along with a vertical 'staircase' with edges parallel to the plane $x = y$. The Minkowski sum of the two sets of shelves contains a tight $n \times n$ grid of planes, all parallel to the z axis. See Figure 9. The Minkowski sum of the two staircases cuts through this grid n^2 times to create $\Omega(n^4)$ vertices. We omit further details from this extended abstract. □

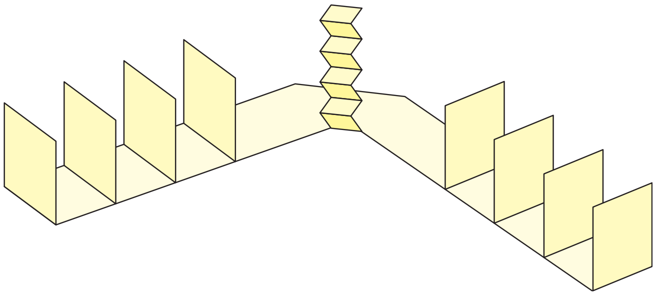

Figure 9. Salient features of the shelves and staircases polyhedron. Compare with Figure 8(a).

8. OPEN PROBLEMS

We have introduced a new realistic input model for simplicial nonconvex polyhedra: *locality*. Intersections and unions of local polyhedra can be computed in near-linear time using standard bounding volume hierarchy techniques. Surprisingly, our model allows polyhedra with sharp features.

Exactly the same techniques apply immediately to any other type of object represented by a simplicial piecewise linear complex. For example, given two local planar straight-line graphs, we can overlay them in $O(n \log n)$ time either by using the standard sweep-line algorithm, or by merging a graded (semi-)R-tree for each graph [5, 27]. Similarly, given two local finite element meshes, we can find all pairs of overlapping elements in near-linear time using recursive bisection. In particular, this technique is efficient for the well-shaped meshes produced by Delaunay refinement algorithms. These and other algorithmic applications of our results will be discussed in the full version of the paper.

We also derived upper and lower bounds for the Minkowski sum of two local polyhedra in two or three dimensions. The closeness of the upper and lower bounds is somewhat misleading, however, since they have very different dependencies on the parameter Σ; the two-dimensional bounds are more accurately written as $O(n^3 \log \Sigma)$ and $\Omega(n^2 \Sigma)$. How complex is the Minkowski sum of two *civilized* polyhedra, where $\Sigma = O(1)$? Our model allows vertices to be arbitrarily close to higher-dimensional facets. Can we obtain better bounds by replacing the nearest neighbor distance in the definition of σ with, say, the local feature size at each vertex?

Intersection, convex decomposition, and Minkowski sum are only two of many problems involving nonconvex polyhedra that are that are difficult in the worst case, but may be easier for local polyhedra. For example, can local polyhedra be triangulated using a near-linear number of simplices? How hard is constructing the triangulation of a local polyhedron with the minimum number of Steiner points or tetrahedra?

References

[1] G. Barequet, B. Chazelle, L. Guibas, J. Mitchell, and A. Tal. BOXTREE: A hierarchical representation for surfaces in 3D. *Comput. Graph. Forum* 15(3):C387–C396, C484, 1996. Proc. Eurographics'96.

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

[3] M. de Berg, M. Katz, M. Overmars, A. F. van der Stappen, and J. Vleugels. Models and motion planning. *Proc. 6th Scand. Workshop Algorithm Theory*, 83–94, 1998. Lecture Notes Comput. Sci. 1432, Springer-Verlag.

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

[5] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joins using R-trees. *Proc. ACM SIGMOD Conf. on Management of Data*, 237–246, 1993.

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

[7] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica* 11:116–132, 1994.

[8] N. Chin and S. Feiner. Near real-time shadow generation using BSP trees. *Comput. Graph.* 23:99–106, 1989. Proc. SIGGRAPH '89.

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

[10] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.* 4:387–421, 1989.

[11] J. Erickson. On the relative complexities of some geometric problems. *Proc. 7th Canad. Conf. Comput. Geom.*, 85–90, 1995. ⟨http://www.uiuc.edu/~jeffe/pubs/relative.html⟩.

[12] J. Erickson. New lower bounds for Hopcroft's problem. *Discrete Comput. Geom.* 16:389–418, 1996.

[13] J. Erickson. Nice point sets can have nasty Delaunay triangulations. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 96–105, 2001. Full version to appear in *Discrete Comput. Geom.*, 2003.

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

[15] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. *Comput. Graph.* ??:171–180, 1996. Proc. SIGGRAPH '96.

[16] L. Guibas, A. Nguyen, D. Russel, and L. Zhang. Collision detection for deforming necklaces. *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 33–42, 2002.

[17] A. Guttmann. R-trees: A dynamic index structure for spatial searching. *Proc. ACM SIGMOD Conf. on Management Data*, 47–57, 1984.

[18] H. H. Haverkort, M. de Berg, and J. Gudmundsson. Box-trees for collision checking in industrial applications. *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 53—62, 2002.

[19] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.* 15(3):179–210, 1996.

[20] J. Klosowski, M. Held, J. S. B. Mitchell, K. Zikan, and H. Sowizral. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Trans. Visualizat. Comput. Graph.* 4(1):21–36, 1998.

[21] X.-Y. Li and S.-H. Teng. Generating well-shaped Delaunay meshes in 3D. *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 28–37, 2001.

[22] I. Lotan, F. Schwarzer, D. Halperin, and J.-C. Latombe. Efficient maintenance and self-collision testing for kinematic chains. *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 43–52, 2002.

[23] G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. A Delaunay based numerical method for three dimensions: generation, formulation, and partition. *Proc. 27th Annu. ACM Sympos. Theory Comput.*, 683–692, 1995.

[24] B. Mirtich and J. Canny. Impulse-based dynamic simulation. *The Algorithmic Foundations of Robotics*, 1995. A. K. Peters.

[25] T. M. Murali and T. A. Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. *Proc. 1997 Sympos. Interactive 3D Graphics*, 1997.

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

[27] P. van Oosterom. An R-tree based map-overlay algorithm. *Proc. EGIS '94*, 318–327, 1994.

[28] M. H. Overmars and A. F. van der Stappen. Range searching and point location among fat objects. *J. Algorithms* 21:629–656, 1996.

[29] 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.

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

[31] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica* 9:471–494, 1993.

[32] O. Schwarzkopf and J. Vleugels. Range searching in low-density environments. *Inform. Process. Lett.* 60:121–127, 1996.

[33] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, 86–95, 1998.

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

[35] S. Suri, P. M. Hubbard, and J. F. Hughes. Collision detection in aspect and scale bounded polyhedra. *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, 127–136, 1998.

[36] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, 51–60, 1995.

[37] 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.

[38] D. Talmor. *Well-Spaced Points and Numerical Methods*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, August 1997. ⟨http://reports-archive.adm.cs.cmu.edu/anon/1997/abstracts/97-164.html⟩. Technical report CMU-CS-97-164.

[39] S. J. Teller and C. H. Séquin. Visibility preprocessing for interactive walkthroughs *Comput. Graph.* 25(4): 61–69, 1991. Proc. SIGGRAPH '91.

[40] S.-H. Teng. *Points, Spheres, and Separators: A Unified Geometric Approach to Graph Partitioning*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1992. Technical report CMU-CS-91-184.

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

[42] J. Vleugels. *On Fatness and Fitness — Realistic Input Models for Geometric Algorithms*. Ph.D. thesis, Dept. Comput. Sci., Univ. Utrecht, Utrecht, The Netherlands, 1997.

[43] Y. Zhou and S. Suri. Analysis of a bounding box heuristic for object intersection. *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 830–839, 1999.

[44] A. Zomorodian and H. Edelsbrunner. Fast software for box intersection. *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, 129–138, 2000.