An Efficient Algorithm for Generating Zonohedra

Paul Heckbert

3-D Technical Memo 11
24 February 1985

Three Dimensional Animation Systems Group
Computer Graphics Lab
New York Institute of Technology

Abstract

An $O(n \log n)$ algorithm for generating order n zonogons and an $O(n^2 \log n)$ algorithm for generating zonohedra are described.

A *k-dimensional hyper-zonohedron of order n* is the convex hull of an n-cube after it has been projected into k-space. Another way to think of this is to visualize n *defining vectors* in k-space radiating from the origin and the 2^n points corresponding to all the combinations of these vectors. These 2^n points, which we shall call *zonopoints*, can be thought of as the affine projection of an n-cube into k-space. When their convex hull is found, a k-dimensional hyper-polyhedron results, called a *zonohedron*. An excellent, concise introduction to zonohedra can be found in [Coxeter].

Each zonopoint can be expressed as:

$$b_1 D_1 + b_2 D_2 + \cdots + b_n D_n \qquad (b_i = 0 \text{ or } 1)$$

The sequence $b_1 b_2 \cdots b_n$ is called the *binary code* for the point. It can be thought of as the zonopoint's coordinates in n-space before projection. The neighbors of a zonopoint are the n zonopoints whose binary codes differ in one bit (those whose graph distance is one). A zonopoint will be on the surface of the zonohedron iff there is a hyperplane passing through it such that one side of the hyperplane contains no zonopoints.

1. Zonogons

A two dimensional zonohedron is a *zonogon*.

Some examples:

The order 1 zonogon is degenerate.

The order 2 zonogon is a parallelogram.

The order 3 zonogon is a hexagon (the zonopoints are a projected box).

The order 4 zonogon is an octagon (the zonopoints are a projected tessaract).

Some properties of zonogons:

It is a convex polygon with $2n$ vertices and $2n$ edges.

Opposite edges are equal and parallel.

It has central symmetry (rotation 180 degrees about its centroid leaves it unchanged).

The zonogon is degenerate if all the defining vectors are collinear (parallel).

Collinear vectors cause degenerate collinear edges in the zonogon which have arbitrary order.

The edges are all equal to one of the defining vectors; each vector appears twice (on opposite sides of the zonogon).

Any parallel-sided $2n$-gon is a zonogon.

We will consider two algorithms for generating zonogons†.

Brute Force Zonogon Algorithm:

1. Generate the 2^n points as the sums of all possible combinations of vectors. This requires $O(2^n)$ time and $O(2^n)$ space.

2. Find their convex hull. This requires $O(n\,2^n)$ time worst case [Shamos].

An Efficient Zonogon Algorithm:

1. Sort the n defining vectors and their opposites by angle. This requires $O(n \log n)$ time and $O(n)$ space.

2. Pick a starting position in the plane and a starting vector. Go through the vector list in sorted order repeatedly adding vectors to the current position. This generates a closed polygon, the zonogon, and requires $O(n)$ time.

† Generation of a zonogon is defined to be computation of lists of vertices and edges and their adjacencies. Generation of a zonohedron is defined to be computation of lists of vertices, edges, and faces and their adjacencies.

Since their time costs are $O(n2^n)$ and $O(n \log n)$ respectively, the second algorithm is clearly faster for large n, and also no harder to implement than the first.

2. Zonohedra

Highly symmetric zonohedra can be generated using the segments joining opposite vertices of a Platonic solid as the defining vectors. Call this set of vectors the *skeleton* of a polyhedron.

Some examples:

The order 1 and order 2 zonohedra are degenerate.

The order 3 zonohedron is a parallelepiped (8V, 12E, 6F). The skeleton of an octahedron generates a cube.

The order 4 zonohedron defined by the skeleton of a cube is a rhombic dodecahedron (14V, 24E, 12F).

The order 6 zonohedron defined by the skeleton of an icosahedron is a triacontahedron (32V, 60E, 30F).

Some properties of zonohedra:

Zonohedra are convex polyhedra with $n(n-1)+2$ vertices, $2n(n-1)$ edges, and $n(n-1)$ parallelogram faces.

All faces are parallelograms.

The vertices can have any degree ≥ 3.

Opposite faces are congruent and parallel.

It has central symmetry.

The zonohedron is degenerate if all the defining vectors are coplanar.

A group of m coplanar vectors causes a degenerate coplanar face with $2m$ sides (an order m zonogon!). It is best to think of this $2m$-gon as m parallelograms for consistency.

The zonohedron edges are all equal to one of the defining vectors; each vector appears $2(n-1)$ times along a *zone* which forms a belt around the zonohedron.

There are n zones, each corresponding to one of the defining vectors.

There are two faces determined by each pair of vectors which are located where their zones cross, opposite each other. The face normal is the cross product of the two vectors.

Any convex polyhedron with zonogon faces is a zonohedron [Coxeter].

It is easy to generalize the brute force zonogon algorithm to 3-D:

Brute Force Zonohedron Algorithm:

1. Generate the 2^n points as the sums of all possible combinations of vectors. This requires $O(2^n)$ time and $O(2^n)$ space.

2. Find their convex hull. This requires $O(n2^n)$ time worst case [Shamos].

Generalizing the efficient zonogon algorithm is subtler. In the 2-D algorithm the critical data structure is a list of all edge directions. In 3-D the analogous information is a list of all face normals. Each face normal can be represented by a point on a unit sphere. These $n(n-1)$ points can be generated by taking cross products of all ordered pairs of the defining vectors D_i for $i=1...n$. The face normals determined by D_i are $D_i \times D_j$ for all $j \neq i$ and these points lie on a great circle perpendicular to D_i. Therefore, the face normals are the intersections of pairs of great circles corresponding to defining vectors. The great circles on the unit sphere correspond to the zones and their intersections correspond to the faces. The graph of ring intersections determines the topology of the zonohedron, in fact, they are duals!

An Efficient Zonohedron Algorithm:

1. Generate the $n(n-1)$ face normals by computing the cross products of all ordered pairs (i,j) of defining vectors. These are put in great circle intersection point lists. Each face normal $D_i \times D_j$ goes in two lists i and j. This step requires $O(n^2)$ time and $O(n^2)$ space.

2. Sort the $2(n-1)$ points on each of the n great circles. This requires $O(n^2 \log n)$ time.

3. At this point the graph of great circle intersections is known. It may be necessary to translate the graph into a convenient polyhedron format, such as winged edge. Take the dual to arrive at the zonohedron topology. Mark each edge with the number of the zone (great circle) to which it belongs. This requires $O(n^2)$ time.

4. The remaining task is to set the geometry. The vertex coordinates and perhaps face plane equations must be computed. An arbitrary vertex is identified and positioned, after which geometric information can be propagated across the surface. Identifying a vertex means finding its binary code. There are several ways to do this. The simplest one involves averaging the face normals around an arbitrary vertex to find a ''vertex normal'' N. The precise value of this normal is unimportant; it can be anywhere within the spherical polygon defined by the surrounding face normals. Dot products tells us which side of each great circle N is on: $b_i = 1$ if $N \cdot D_i > 0$ else 0. Given this vertex's binary code, it is a simple matter to incrementally determine the binary codes and positions of neighboring vertices. This step requires $O(n^2)$ time.

Based on the great circle synthesis method we can make the following observations:

The zonohedron's topology is independent of the lengths of the defining vectors.

When a new defining vector is added it splits the zonohedron and inserts a new zone between the two halves dependent on where the new great circle falls.

The dual of any polyhedron topologically equivalent to a graph of great circle intersections is a zonohedron. The octahedron, cuboctahedron, and icosidodecahedron are examples.

Degenerate zonohedron faces result when 3 or more great circles intersect in one point. If such multiple intersection points are not handled specially they will cause ambiguous sorting decisions during step 2 which might cause errors in step 3.

Another potential difficulty is parallel defining vectors since they would lead to coincident great circles.

Our algorithm for zonogons required sorting $2n$ defining vectors by angle. If instead their unit perpendiculars are sorted, these *edge normals* will be equivalent to points on a unit circle, like the intersection points on each great circle in the 3-D algorithm. This variation suggests that sorting $2(n-1)$ points on each of the great circles in the 3-D algorithm is the same complexity as generating an order $n-1$ zonogon. This explains why our zonogon algorithm required $O(n \log n)$ time and the zonohedron algorithm required n times that, $O(n^2 \log n)$.

We conjecture that our algorithm can be generalized to k dimensions as follows: A k-dimensional hyper-zonohedron has $2\binom{n}{k-1}$ hyperfaces, each perpendicular to $k-1$ defining vectors. On the hypersphere containing the hyperface unit normals, there will be $\binom{n}{k-2}$ great circles, each containing

$2(n-k+2)$ intersection points. Sorting these takes $O(n^{k-2} \cdot n \log n)$ time, which is probably the dominant cost of the algorithm.

3. Acknowledgements

Thanks to Veit Elser for introducing me to zonohedra, and to Tom Shermer and Pat Hanrahan for their comments on this paper.

4. References

Coxeter, H. S. M., *Regular Polytopes,* Dover, New York, 1973.

Shamos, Michael I., *Computational Geometry,* PhD thesis, CS Dept., Yale University, May 1978.