Implementation of a Generalized Nonoverlapping Unfolding Algorithm

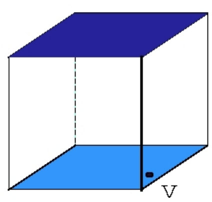
Nathaniel Born[1]   Ezra Miller[2]

1. Introduction

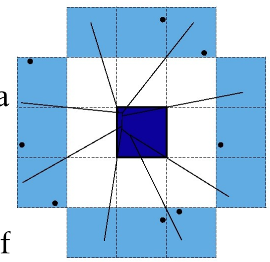
Algorithms for computing non-overlapping foldouts of 3-dimensional convex polyhedra have been developed [1,2] and implemented [3,5] for quite some time. Recently, Miller and Pak [4] gave an algorithm to compute certain non-overlapping foldouts for convex polyhedra of arbitrary dimension. This presentation will describe an implementation of the Miller-Pak algorithm, including a discussion of the algorithm itself, issues that were raised during the implementation process, and sample results of 3-dimensional foldouts of boundaries of 4-dimensional polytopes.

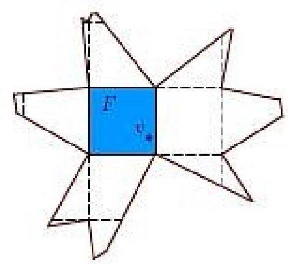
2. Overview of the algorithm

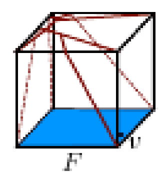
The input is a polyhedron of dimension d+1 and a *source point* on the boundary of the polyhedron. For example, the 3-dimensional cube to the right. The source point v emits a signal, sending out a spherical wavefront of dimension d along the boundary of the polyhedron. At certain times and locations, called *events*, the geometry of the wavefront changes. An event is a 3-tuple (w, F, R) such that w is a *source image*, F is a facet, and R is a ridge with the property that R lies in the hyperplane between w and F. The source image is the point, in the hyperplane containing F, where the source point appears to be if you look from F through R.



The algorithm computes the events in chronological order as follows. After some events have been computed, identify a set of *potential events*. Declare the “closest” potential event to be the next event. As the wavefront wraps itself around lower dimensional faces of the polyhedron, it looks, from the point of view of an observer in any given facet, that there are multiple source images in the hyperplane of the facet, each emitting its own wavefront. These source images induce a Voronoi subdivision of the facet. To the right is an example of the Voronoi subdivision of the top facet of the cube. The regions in the subdivision are then rearranged to form a foldout, such as the one below and to the left. Below and to the right is an image of the cube with the cuts of the foldout shown.







3. Pseudocode

Input: a d+1-dimensional polyhedron P and a source point v on some facet F of P.

Step 1. Initialize the set E_F of potential events of F to be the triples (source image, facet, ridge) for the facets adjacent to F. Initialize Y_F, the set of source images of F, to be v. For all other facets F' of P, set $Y_{F'} = \{\}$, $E_{F'} = \{\}$. Define $\mathcal{E}$ to be the union of all potential events of P.

Step 2. Compute the dihedral angles for all pairs of adjacent facets.

Step 3. From $\mathcal{E}$, choose the potential event E that occurs earliest (so E is the “next event”), and

1 Department of Mathematics, University of Minnesota
email: born0063@umn.edu

2 Department of Mathematics, University of Minnesota
email: ezra@math.umn.edu

remove E from E_F and $\mathcal{E}$.

Step 4. For the facet F' on the other side of the relevant ridge of F, add to $Y_{F'}$ the source image corresponding to E.

Step 5. Compute the Voronoi diagram for F' with its source images $Y_{F'}$. For all possible potential events involving F', find the potential events (w, F', R) such that *w can see R through F*. That is, such that a line segment can be drawn from w through the interior of F to R, without leaving its respective Voronoi subdivision. Add all such events to $\mathcal{E}$ and $E_{F'}$.

Step 6. Continue to repeat steps 3-5 until there are no more potential events to process.

Step 7. Calculate the Voronoi diagram for each facet F using its corresponding source images, Y_F. The different regions of the Voronoi subdivisions are then sequentially rotated to the hyperplane containing the original facet F using the angles computed in Step 2.

4. The closest point problem

Step 3 requires us to be able to do the following: Given a point v and a polyhedron P, find the point on P that is closest to v. Experts we consulted claimed that existing algorithms to do this are not particularly efficient. Furthermore, it would have been difficult to integrate the optimization methods of the existing algorithms into our implementation. Therefore we developed a naïve geometric algorithm to compute these closest points:

Place the origin arbitrarily in the interior of P. Then, take the ray from the origin through v, and travel infinitely far away to a point x. The point x will be in a unique *outer normal polyhedron* to P. As x travels back along the ray towards the origin, x will cross from one outer normal polyhedron to another. When x is in the same outer normal polyhedron C_F as v, project x orthogonally onto F, the face of P whose outer normal polyhedron is C_F.

5. The finiteness theorem

In order to determine the "next event" to process (as in Step 3), we need to look at the points x where multiple wavefronts converge on each other. A theorem [4] indicated that a subroutine for determining this exists, but did not produce sufficient pseudocode for implementation. At these points, we iterate the following procedure for each potential event:

Input: a point x; an outer support vector to x, v; and a cone V with apex x.

Step 1. Initialize an ordered sequence of mutually orthogonal unit vectors $\mathcal{L}$ to be empty.

Step 2. Choose any nonzero vector ζ in V that minimizes the angle with v, and add it to $\mathcal{L}$.

Step 3. Take the orthogonal hyperplane to ζ, move it one unit along ζ, intersect it with V, then move it back one unit. This is the new V, with the same support vector.

Step 4. Repeat steps 2-3 until Step 2 is no longer possible.

The next potential event to process will be the event with the lexicographically minimal angle sequence (Theorem 4.11 in [4]).

6. References

[1] A. D. Aleksandrov, Vnutrennyaya geometriya vypuklykh poverkhnostey (in Russian), M.–L.: Gostekhizdat, 1948; English translation: Selected works. Intrinsic geometry of convex surfaces, Vol. 2, Chapman & Hall/CRC, Boca Raton, FL, 2005.

[2] A. D. Aleksandrov, Vypuklye mnogogranniki (in Russian), M.: Gostekhizdat, 1950; English translation: Convex polyhedra, Springer, Berlin, 2005.

[3] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, The discrete geodesic problem, SIAM J. Comp. 16 (1987), no. 4, 647–668.

[4] E. Miller and I. Pak, *Metric Combinatorics of Convex Polyhedra: cut loci and nonoverlapping unfoldings*, Discrete and Computational Geometry (2008), no. 1-3, 339-388. DOI: 10.1007/s00454-006-1249-0[5]

[5] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, SIAM J. Comp. 15 (1986), no. 1, 193–215.