The Crust Algorithm for 3D Surface Reconstruction

Nina Amenta*
University of Texas
Austin, TX 78712, USA
amenta@cs.utexas.edu

1 Introduction

This video demonstrates the *crust* algorithm for surface reconstruction [1],[2] and gives some intuition about why it works. The input to the algorithm is a set S of sample points from the surface of a three-dimensional object. The output, the *crust* of S, is a set of triangles, with the samples as their vertices. When the object is smooth and the samples are distributed sufficiently densely, we can show that all the crust triangles are close to the object surface, and that the crust covers the surface.

2 Algorithm and Sampling

The basic crust algorithm is very simple. First, we compute the three-dimensional Voronoi diagram of the sample set S. Then for each sample s, we select its two farthest Voronoi vertices, one on either side of the surface - the *poles* of s. Let P be the set of all poles. Next, we compute the Delaunay triangulation of $S \cup P$, the union of samples and poles. From this triangulation, we discard any triangle with a pole as a vertex, keeping only triangles connecting a triples of samples. These form the crust.

The crust is guaranteed to be a good reconstruction of a surface W provided that the sample set $S \subset W$ is sufficiently dense. The definition of "sufficiently dense" is related to the *medial axis* of W, which is defined as the closure of the set of points with more than one closest point on W. We say S is an r-sample if, for any point $x \in W$, the distance from x to the nearest sample is at most a constant fraction r times the least distance from x to the medial axis. For our (theoretical) purposes, S is sufficiently dense when $r \leq .01$. In practice, larger values of r work well; see Figure 2. Notice that the sampling needs to be dense only near sharp

Figure 1: A reconstruction from a dense set of about 20,000 samples.

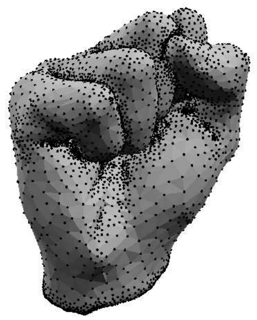

Figure 2: The samples here are distributed using a value of $r \approx 1/2$.

* Partially supported by NSF grant CCR-9731977.

features.

3 Intuition

Although the proofs are non-trivial [1], the intuition behind the crust algorithm is simple. When the sampling is dense, the Voronoi cell $Vor(s)$ of any sample s has to be long, skinny and perpendicular to the surface. This is because some of the facets of $Vor(s)$ are supported by the perpendicular bisectors separating s from its nearby neighbors on the surface. Since $Vor(s)$ clearly has two ends, we can select the poles unambiguously. One nice lemma is that the vectors from a sample s to its poles approximate the surface normal at s to within an additive factor, linear in r.

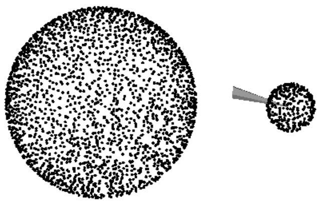

Figure 3: One long skinny cell in the Voronoi diagram. The samples are densely distributed on the surface of a pair of spheres.

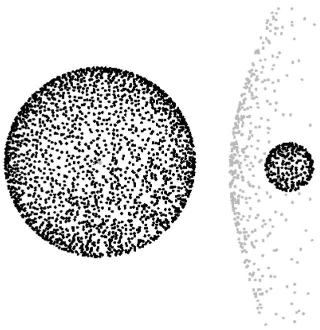

Figure 4: The poles are distributed near the medial axis of the pair of spheres (which is a sheet of a hyperboloid, along with the two circumcenters).

$Vor(s)$ extends perpendicularly from the surface in either direction, and ends near the medial axis of W (since past the medial axis samples on the opposite patch of surface are closest). The medial axis separates opposite patches of surface, and the set P of poles forms something like a sampling of the medial axis. This is why, when we find the Delaunay triangulation of $S \cup P$, two samples on opposite patches are never connected by a Voronoi triangle; there is always some pole in the way!

4 Implementation, Visualization and Acknowlegments

The crust algorithm is surprisingly robust. Manolis Kamvysselis developed this implementation during a summer internship at Xerox PARC, working with Marshall Bern and Nina Amenta. We used Ken Clarkson's `Hull` program to generate Delaunay triangulations. Computing the Voronoi center of a tetrahedron can be unstable; but since the algorithm needs only a subset of the Voronoi centers, we could eliminate unstable (very flat) tetrahedra from consideration. We successfully computed some very large, detailed models. The video concludes by examining a few of them.

We used `Geomview` for visualization, using interactive screen capture for some shots and producing individual frames for others. Production assistant John Liechty generated some of the footage, recorded the sound, and was a great source of moral support. Tamara Munzner provided prompt long-distance advice. The (quite simple!) hardware was an SGI O2 and a nifty Panasonic AG1980 VCR. We used SGI `Moviemaker` and `Mediaplayer` for editing and recording.

We decided to use a real chalkboard over dinner with Julien Bausch and John Liechty. At the time, it seemed humorous.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. To appear in *Discrete and Computational Geometry.* An earlier version appears in the *14th ACM Symposium on Computational Geometry*, (1998) pages 39–48.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. *Siggraph '98*, pages 415–421.