The Powercrust Algorithm for Surface Reconstruction

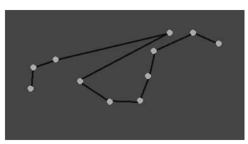
Nina Amenta   Sunghee Choi   Ravi Kolluri
University of Texas at Austin

Correctness

- Boundary of a solid
- Close to original surface
- Homeomorphic to original surface

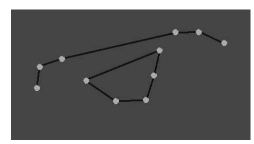
Correctness

- Boundary of a solid
- Close to original surface
- Homeomorphic to original surface

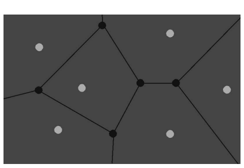


Correctness

- Boundary of a solid
- Close to original surface
- Homeomorphic to original surface



Tools - Voronoi Diagram



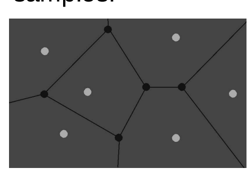
Points closest to each sample form cells.

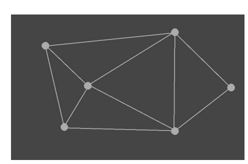
Cell boundaries have more than one closest sample.

Adjacent cells define adjacent samples.

Delaunay Triangulation

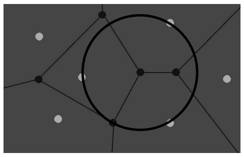
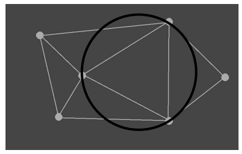
Delaunay triangles connect adjacent samples.





Delaunay Triangulation

Delaunay triangles connect adjacent samples.

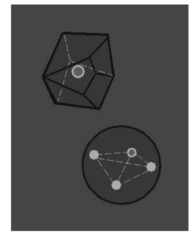
Voronoi balls centered at Voronoi vertices pass through closest samples.

3D Voronoi/Delaunay

Voronoi cells are convex polyhedra.

Voronoi balls pass through 4 samples.

Delaunay tetrahedra.



Voronoi-based Surface Reconstruction

Boissonnat, 84

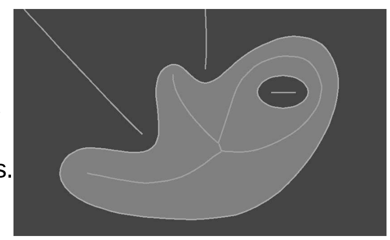
Edelsbrunner and Mucke, 94

Bernardini et al, 98

Amenta and Bern, 98

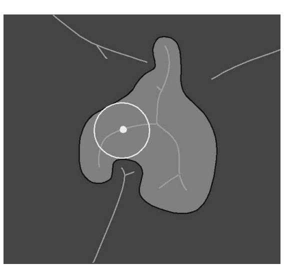
Medial Axis

Think of object surface as infinite set of samples.



Medial axis is set of points with more than one closest sample.

Medial Axis

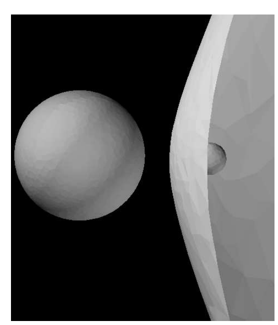


Maximal ball avoiding surface is a medial ball.

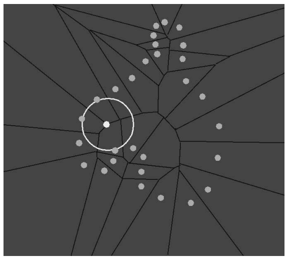
Every solid is a union of balls !

3D Medial Axis

Medial axis of a surface forms a dual surface.

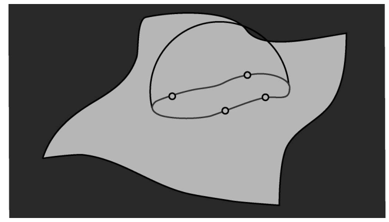


2D Medial Axis Approximation



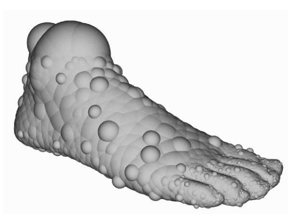
Voronoi balls approximate medial balls.

Sliver tetrahedra



In 3D, some Voronoi vertices are not near medial axis …

Poles

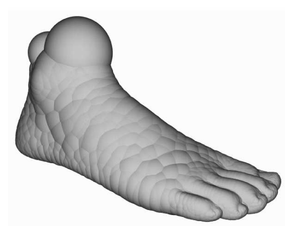


Interior Voronoi balls

Problem in 3D:

Not all Voronoi vertices are near medial axis, even when samples are arbitrarily dense.

Poles



Interior *polar* balls

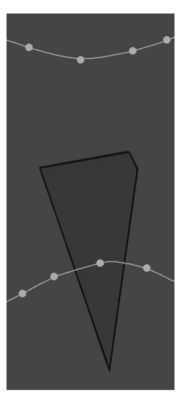
Subset of Voronoi vertices, the poles, approximate medial axis.

Amenta & Bern, 98

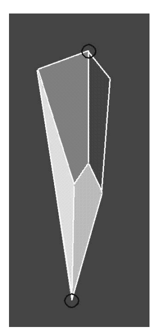
Poles

For dense surface samples, Voronoi cells are:

- long and skinny,
- perpendicular to surface,
- with ends near the medial axis.



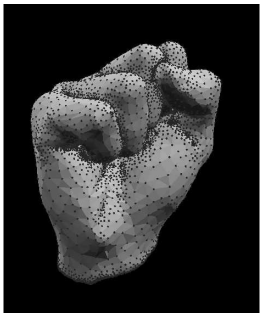
Poles



Poles are Voronoi vertices at opposite ends.

To find: farthest Voronoi vertex from sample, farthest on opposite side.

Crust Algorithm



Surface reconstruction with theoretical guarantees.

Uses poles to find Delaunay triangles eligible for surface.

Amenta, Bern and Kamvysselis, 98

Improvments on Crust

Amenta et al, 00: Simpler algorithm, simpler proof, topological guarantees.

Dey and Giesen, 01: Sharp corners and boundaries.

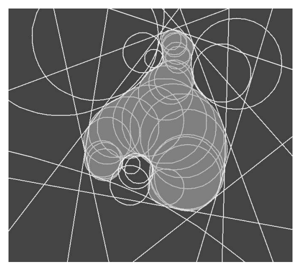
Ramos, 01: O(n lg n) algorithm, replacing Delaunay with well-separated pair decomposition.

Practical Crust Drawbacks

- Fails when sample is not sufficiently dense: holes in surface, errors at sharp corners.
- Need to select surface from set of eligible triangles. Hard to do in a way that is provably correct, makes nice surface, etc.

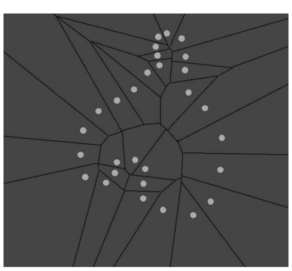
Project: Algorithm which is robust, has no post-processing, and is still correct.

Power Crust



Idea: Approximate object as union of balls, compute polygonal surface from balls.

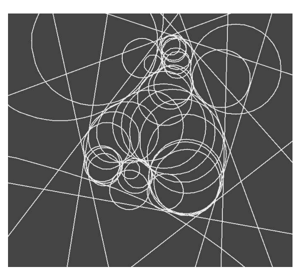
Power Crust



Compute Voronoi diagram of samples.

Select poles to approximate object and its complement by finite unions of balls.

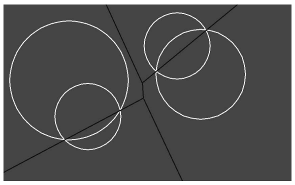
Power Crust



Compute polygonal surface from polar balls using power diagram.

Power Diagram

Power diagram is Voronoi diagram of balls.



Voronoi diagram program can be easily modified to produce power diagrams.

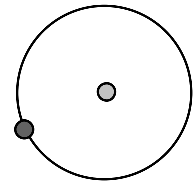
Has polyhedral cells.

Power Diagram

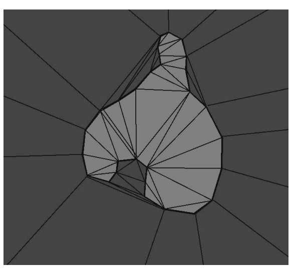
Ball B, center c, radius r

Power Distance from B to point x:

$d_{pow} = d^2(c,x) - r^2$



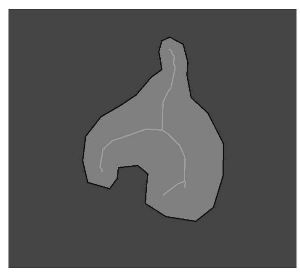
Power Crust



Label power diagram cells inside or outside object (skipping details).

Inside cells form polyhedral solid.

Power Crust



Boundary of solid approximates surface: power crust.

Connect inner poles with adjacent power diagram cells: power shape approximates medial axis.

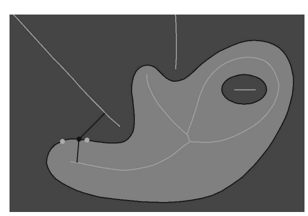
Power Crust

Robust: Always boundary of a solid.

Simple: No surface extraction or hole-filling steps required.

Correct: Theoretical results relate geometric and topological quality of approximation to quality of sample.

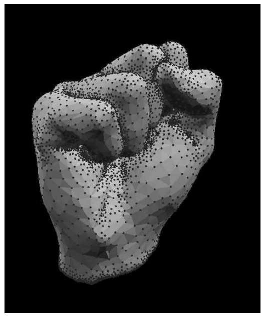
Sampling Requirement



Sample is sufficiently dense when distance from any surface point **x** to nearest sample is at most small constant **r** times distance to medial axis.

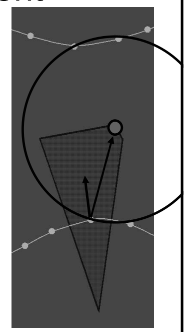
Sampling Requirement

Captures intuition that we need dense sampling where curvature is high or where there are nearby features.



Large balls tangent

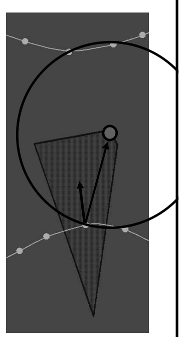
Any large ball (with respect to distance to medial axis) touching sample s has to be nearly tangent to the surface at s.



Specifically

Given an ε-sample from a surface F:

Angle between normal to F at sample s and vector from s to either pole = $O(\varepsilon)$



Theoretical Results

Amenta, Choi, Kolluri, CGTA 01.

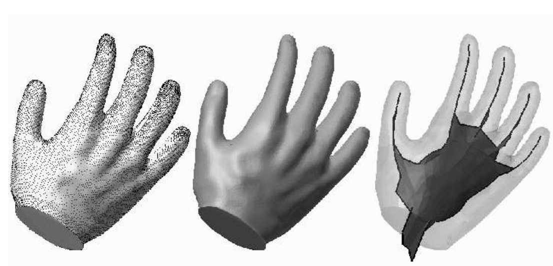
Assume sufficiently dense sampling, smooth surface.

- Power crust approaches object surface linearly as sampling density increases.
- Power crust normals converge to surface normals linearly.
- Power crust is homeomorphic to surface for dense enough samples.

Theoretical Results

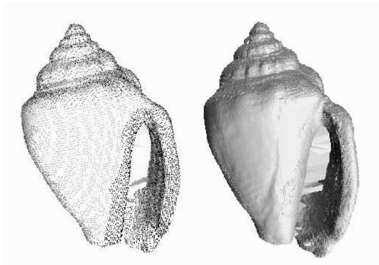
- Similar results for union of balls.
- Power shape is homotopy equivalent to solid object.
- Set of poles converges to medial axis, faster in some places than in others.
  - also Boissonnat and Cazals, 01; and Dey, 02 gives polygonal MA approximation.

Results



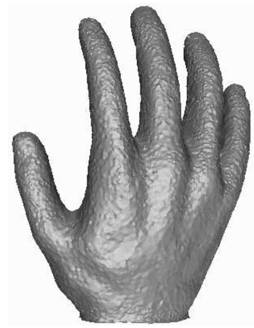
Laser range data, power crust, approximate medial axis.

Results



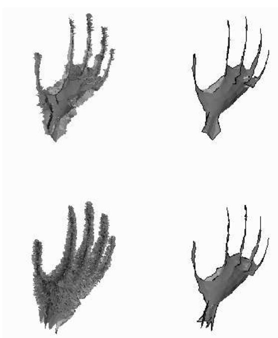
Four laser range scans merged.
Hole deep inside object filled.

Robust



Good reconstruction even with lots of added noise.

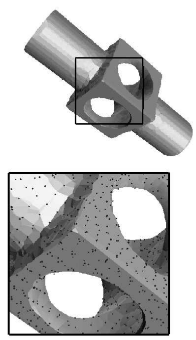
Medial Axis Simplification



Simplification of original power shape.

Simplification of power shape of noisy hand.

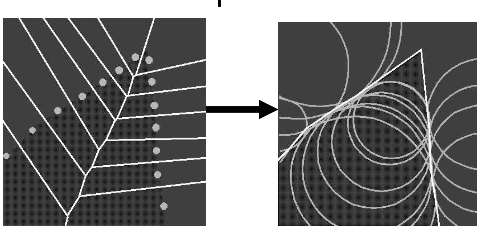
Sharp Corners



With additional hack, sharp edges can be resolved when they are far enough away from other features.

No need to have samples on the edge!

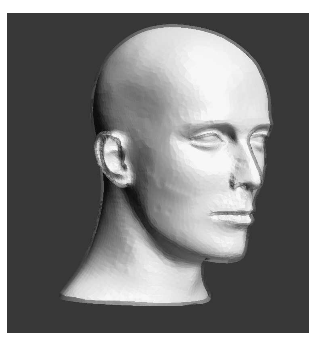
Sharp Corners



Detect badly shaped Voronoi cells and omit those poles.

Power diagram cells of nearby good poles form a sharp corner!

Approximate Offset Surface



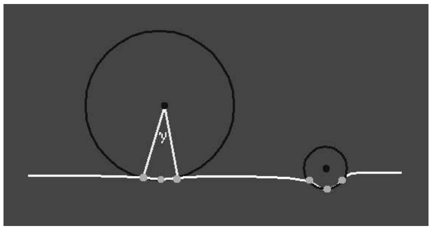
Shrink inner balls.

Grow outer balls.

Compute power crust - always solid output!

Only accurate for small offsets.

Medial Axis Simplification



Samples determining noise balls are closer together than noise threshold.

Remove noise balls before computing surface.

Software

Software, papers, models….

www.cs.utexas.edu/users/amenta/powercrust

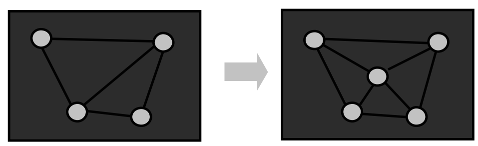
Incremental Constructions con BRIO

Nina Amenta (UC-Davis)

Sunghee Choi (UT-Austin)

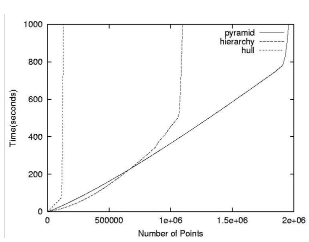
Günter Rote (Freie Univ. Berlin)

Randomized Incremental Delaunay Algorithm



Add points one by one in random order, update triangulation.
Simple and optimal.

Drawback



Performs great…until !

Idea

Partially randomized insertion order

- increase locality of reference, especially as data structure gets large
- retain enough randomness to guarantee optimality

Result

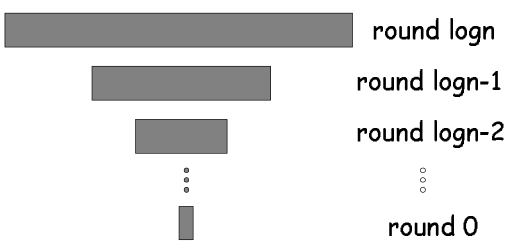
- We give a new ordering called *BRIO* (biased randomized insertion order) that is still optimal.
- Size of input we could compute increased 500K → 10M.

Biased Randomized Insertion Order (BRIO)

- Choose each point with prob = 1/2.
- Insert chosen points recursively con BRIO.
- Insert the remaining points in arbitrary order.

BRIO

log n rounds of insertion



BRIO

- Which round a point is in is random.
- In each round, points are inserted in arbitrary order.
- The arbitrary order allows us to introduce locality.

Implementation

- Divide points into local cells (oct-tree)
- In each round, visit cells in fixed order and add points in cell together

Analysis

Randomness has two benefits:

- Bound total number of tetrahedra
- Bound time required for locating new points in triangulation

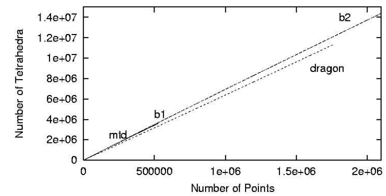
Analysis

Two cases:

Worst-case - size of Delaunay triangulation is $O(n^2)$

Realistic-case - size of Delaunay triangulation is $O(n)$. Assume for any random subset R, DT(R) = $O(|R|)$

"Realistic" case



Linear size, linear expected size of intermediate triangulations .

Results

In "realistic case":

Expected total number of tetrahedra $= O(n)$

Expected running time $= O(n \lg n)$

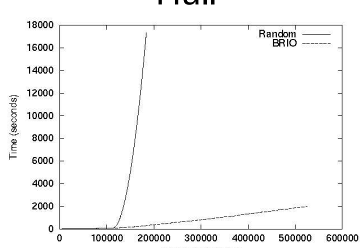
Results

In worst case:

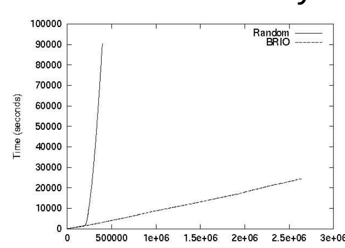
Expected total number of tetrahedra $= O(n^2)$

Expected running time $= O(n^2)$

Hull



CGAL Hierarchy



128M RAM, 360 MHz, 4 G virtual memory

Pyramid

More space-efficient but $O(n^{1/4})$ point location.

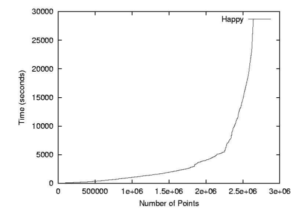
Use smaller memory, slower machine and much larger data. Multiple “Happy buddha”. 4096 kd-cells.

360 MHz

128 M RAM

4 GB Virtual memory

Pyramid

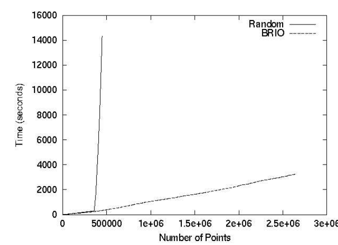


More space-efficient but $O(n^{1/4})$ point location.

Point Location Hack

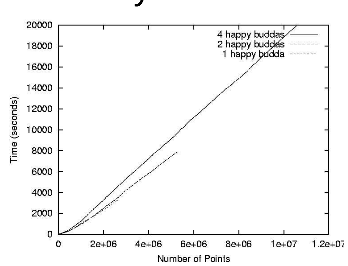
- Instead of $O(n^{1/4})$ jump-and-walk, just walk from last inserted point.
- As size grows, locality increases, so point location time remains roughly constant.

Pyramid



128M RAM, 360 MHz, 4 G virtual memory

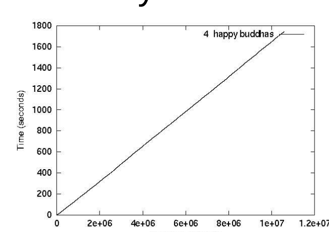
Pyramid



4 million points in 2 hours

10 million points in 5.5 hours

Pyramid



512 M RAM, 10 million points in 30 min