Reconstruction of 3D Meshes from Point Clouds

Ming Zhang
Patrick Min

cs598b, Geometric Modeling for Computer Graphics
Feb. 17, 2000

Outline

- problem statement
- motivation
- applications
- challenges
- three approaches:
  - signed distance function (Hoppe '92)
  - incremental construction (Mencl '98)
  - Voronoi crust (Amenta '98)
- previous work, classification of methods
- summary

Problem Statement

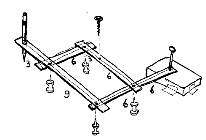
Given:
a set P of unorganized sample points from an unknown surface S

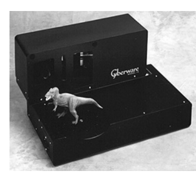
Produce:
a surface which approximates S

Motivation

Relatively easy to point-sample objects, using for example:

- pantograph





- laser range scanner

Applications

- create model from exisiting part (CAD/CAM)
- analysis of used parts
- modeling for virtual worlds

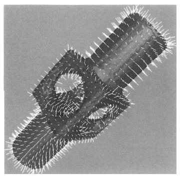
- surfaces from slices of biological specimens
- from laser range data
- from interactive sketching

Challenges

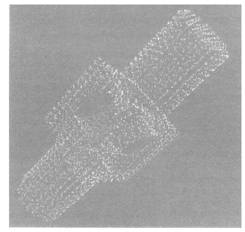
- reconstruction should cover wide range of shapes
- what is a sufficient sampling density?
- how to deal with arbitrary topology
- surface orientation
- inside/outside determination
- mesh optimization/simplification
- "sharp" features (sharp edges and boundaries)
- continuity guarantees

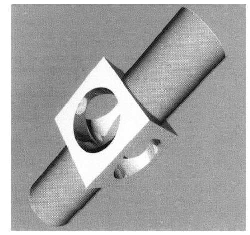
Surface Reconstruction from Unorganized Points

Hugues Hoppe, Tony DeRose, Tom Duchamp,
John McDonald, Werner Stuetzle, SIGGRAPH 1992



Goal of surface reconstruction





δ noisy and ρ-dense

- Two Definitions: δ noisy and ρ-dense
- xi = yi + ei, |ei| < δ
- ρ-dense: sphere with radius ρ contains >= 1 sample point
- This is a general approach

Algorithm

1. define a signed distance function f:D -> R
   - associate oriented plane with each point: compute "tangent planes" from neighbouring points
2. use a contour tracing algorithm to approximate Z(f)

Computing tangent planes

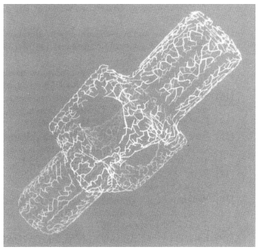
- k-Nbhd(xi) is the k points of X nearest to xi
- Oi is the centroid of K-Nbhd(xi)
- Choose Ni such that best fitting to Nbhd(xi)
- Use covariance to compute Ni
- tangent plane at xi has center Oi, normal Ni

Finding consistent orientation

- Model this problem as graph optimization
- Each Oi (center) has a corresponding Vi (vertex in graph)
- Connect Vi and Vj is Oi and Oj are close
- Cost on edge is Ni*Nj

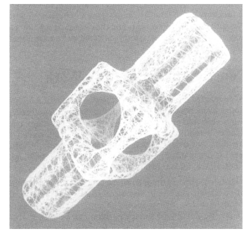
Euclidian Minimum Spanning Tree

First compute EMST of tangent plane vertices



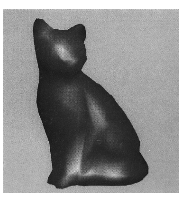
Riemannian Graph

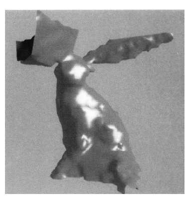
- add edges to EMST:
  add edge (i,j) if Oi or Oj are in the K-nbhd of the other
- Resulting graph is Riemannian Graph



Compute orientation from graph

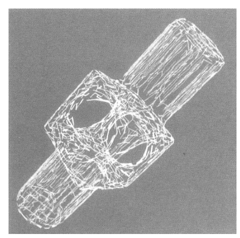
- Maximize the total cost of the graph
- The problem is reducible to MAX CUT
- propagation order is important

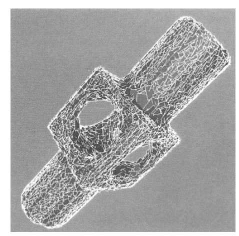




Obtaining good propagation order

- Assign cost 1-|Ni*Nj| to edge (i,j)
- traverse Minimum Spanning Tree: tends to propagate along directions of low curvature





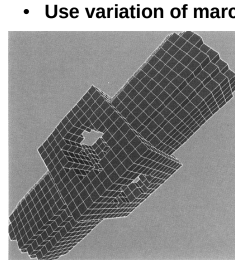
Computing distance function

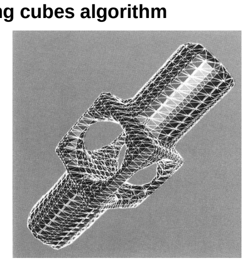
- Find Tp(xi) whose Oi is closest to p
  z = p – ((p – Oi) * ni) * ni
  if d(z,X) < ($\rho + \delta$) then
  f(p) = (p – Oi) * Ni
  else
  f(p) = undefined

- creates a Zero Set Z(f), piecewise linear, but contains discontinuities

Extracting isosurface

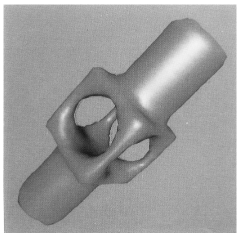
- Contour Tracing is to extract an isosurface from a scalar function
- Use variation of marching cubes algorithm



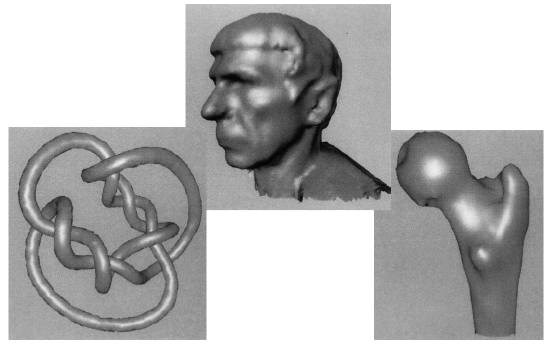


Collapse edges

- collapse edges in a post processing step

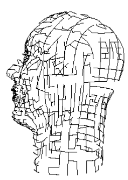


Sample results



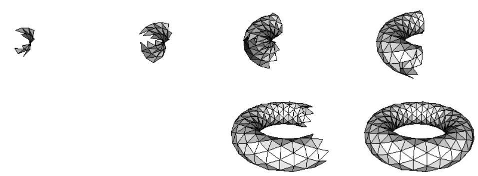
Graph-Based Surface Reconstruction Using Structures in Scattered Point Sets

Robert Mencl and Heinrich Mueller, CGI 1998



Method

Class: incremental surface oriented construction



Iteratively augment Euclidian Minimum Spanning Tree

Each step is based on (heuristic) rules acting on features

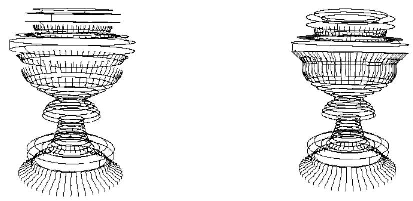
Algorithm

1. compute EMST
2. extend leaves
3. recognize ring and path features
4. extract different parts
5. connect similar features
6. connect associated edges
7. fill wireframe with triangles

1. Compute EMST

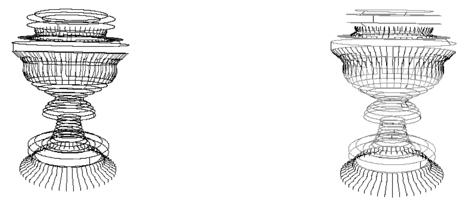
“distribution of points should allow human observer to understand structure of surface”

2. Extend leaves



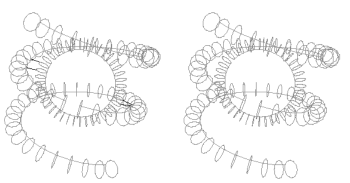
connect leaf edges to edges in neighbourhood
- prevent intersecting edges
- prevent thin triangles

3. Recognize ring and path features



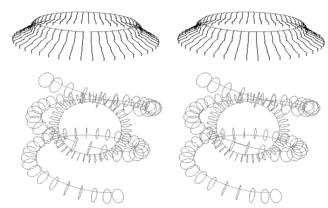
ring:
- angle between consecutive edges > 135
- normals similar
- projection does not produce intersecting edges
- all turn the same way

4. Extract different parts



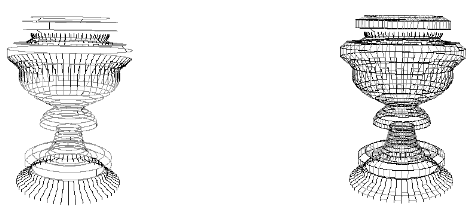
- edge connects two rings or ring with path:
  - only if similar orientation
  - distance between ring centers <=
    distance between closest edges

5. Connect similar features



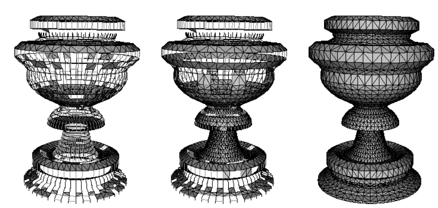
add edges connecting ends of paths with similar orientation
only if similar orientation, and edge length <=
max(both ending edges) * factor

6. Connect associated edges



create quadrilaterals by connecting “almost parallel” edges
- all angles > certain minimum
- don’t reconnect previously disconnected edges

7. Fill wireframe with triangles



In order of smallest enclosing angle
- avoid flat tetrahedra
- maximize sum of dihedral angles of edges around point

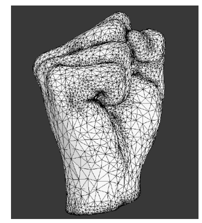
Evaluation

+ exactly interpolates
+ handles variations in point density
+ handles non-orientable surfaces
+ handles arbitrary topology

- no sampling conditions
- most rules are intuitive, not clear how method performs for different shapes

A New Voronoi-Based Surface Reconstruction Algorithm

Nina Amenta, Marshall Bern,
Manolis Kamvysselis, SIGGRAPH 1998

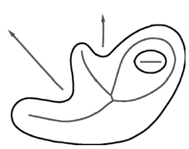


Method

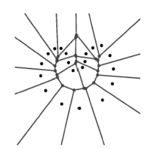
Class: surface oriented spatial subdivision

Use "crust" triangles of Delaunay triangulation of sample point and Voronoi vertices

Use medial axis to define "good sample"

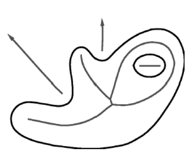


Voronoi vertices and medial axis



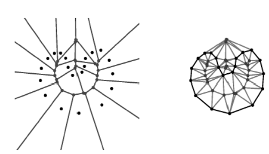
- Voronoi vertex equidistant to 3 points
- Voronoi vertices approximate medial axis

Sampling criterion



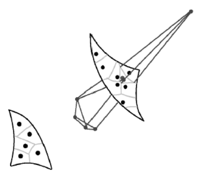
- density at least inversely proportional to distance to medial axis
=> distance to nearest sample <= r * distance to medial axis
theory: r <= 0.06, practice r <= 0.5

Creating the "crust"



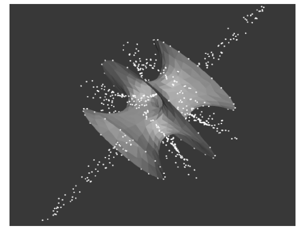
- compute Delaunay triangulation of set S of sample points and Voronoi vertices
- edges between points in S: crust edges
  called "Voronoi filtering"

Voronoi vertices in 3D



- *some* Voronoi vertices lie near medial surface
=> use only 2 opposing vertices of Voronoi cell for filtering step (called “poles”)

Using poles



- compute Delaunay triangulation of S and P (poles)
- keep only triangles with v1, v2 and v3 in S

Evaluation

+ exactly interpolates
+ topologically correct
+ converges to original surface
+ handles varying density
+ all proven

- crust is not necessarily manifold (use poles to do “normal filtering”)
- problem with sharp edges: Voronoi cell is not long and thin.
  heuristic: use farthest and 2nd farthest vertex

Classification of methods

• spatial subdivision
  • surface oriented
  • volume oriented

• distance functions

• warping

• incremental surface oriented

Warping

Terzopoulos, Witkin and Kass (1988, 1991):
  deformable superquadrics
Miller et al. (1991):
  deformation based on set of constraints
  “inflating balloon in object”
Algorri and Schmitt (1996):
  mass in points, springs between points,
  degree 2 LDE, iterative solution
Baader and Hirzinger (1993):
  Kohonen feature map, training data is
  derived from coordinates of points

Volume oriented

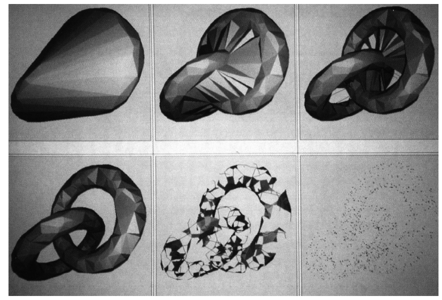
Boissonat (1984):
  Delaunay triangulation
  tetrahedra with certain properties are
  successively removed
  restriced to genus 0 objects
Isselhard et al. (1997):
  addition of rule to allow holes

Volume oriented

Bajaj, Bernardini (1995):
approximate signed distance function
using alpha solids
1. Delaunay triangulization
2. alpha shape
3. alpha solid (alpha s.t. solid is connected)

build piecewise polynomial approximation in
tetrahedral cells (least sq. Bezier patches)
smooth surface to C1

Alpha shapes



Summary

- many different methods
- most use Delaunay/Voronoi

- Amenta: “need reliable techniques to identify sharp edges and boundaries”

for shape analysis:
- heuristics used are interesting
- alpha shapes may be useful