Surface Reconstruction

Power Diagrams, the Medial Axis Transform and the Power Crust Algorithm

Matthew Seegmiller and Adam Smith

\{xaco,adsmith\}@mit.edu.

6.838 Geometric Computation
Lecture 19 — 13 November 2001
Overview

- Introduction
- Weighted distance and power diagrams
- Medial Axis Transform
- PowerCrust Algorithm
Introduction
Introduction

- What is Surface Reconstruction?
- Applications
- Difficulties
- Survey of techniques
Surface Reconstruction

Given a set of points $X$ assumed to lie near an unknown surface $U$, construct a surface model $S$ approximating $U$. 
How it usually works

- Input points sampled from the surface either “by hand” or via a physical process (e.g. 3D scanning).
- Assume:
  - Real surface $U$ is “nice” (= “smooth”)
  - Samples $X$ are “dense enough”, especially near features such as edges, points, bumps, etc.
- Output $S$ in usable format for processing
  - Triangulation of $S$
  - Fitted “splines” (i.e. low-dimensional surfaces)
  - CSG model
Applications of 3D Scanning

- Reverse engineering / Industrial design
Applications of 3D Scanning

- Reverse engineering / Industrial design
- Performance analysis and simulations (e.g. drag)
Applications of 3D Scanning

- Reverse engineering / Industrial design
- Performance analysis and simulations (e.g. drag)
- Realistic virtual environments
Applications of 3D Scanning

- Reverse engineering / Industrial design
- Performance analysis and simulations (e.g. drag)
- Realistic virtual environments
- Medical Imaging
Applications of 3D Scanning

- Reverse engineering / Industrial design
- Performance analysis and simulations (e.g. drag)
- Realistic virtual environments
- Medical Imaging
- ...

Surface Reconstruction – p.7/60
Modeling a claw I
Modeling a Claw II
Modeling hand-made parts
Medical Shape Reconstruction
Difficulties

- Surface not smooth
- Noisy data
- Lack of orientation data
- Surface not watertight
Techniques
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fitting Parametric Surfaces

- Assume surface is from some known family (e.g. sphere, cylinder, plane, hyperboloid, etc)
- Find best parameters to fit data
Fitting Parametric Surfaces

- Fast, accurate for good data
- Useless when data is of unknown type
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Technique</td>
<td>Assumptions</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Contour Data Reconstruction

- Piece together image from parallel slices
- Assumes data is “pre-structured”
- Applications: medical, topographic terrain maps
<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
</tbody>
</table>

- shape Triangulation
- Noise-free

"Mesh" methods
- Dense Sample
- Crust Methods
- Dense Sample

Surface Reconstruction – p.19/60
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td><strong>Fit Gaussian Kernels</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fitting Gaussian balls

- Take linear combination of 3D Gaussians
  \[ f(\vec{x}) = \sum c_i e^{(\vec{x} - \vec{\mu})^\top K_i (\vec{x} - \vec{\mu})} \]
- Surface \( S = \{ \vec{s} \mid f(\vec{s}) = 0 \} \)
  (inside = positive, outside = negative)
Fitting Gaussian balls

Problems:

- Must know surface normal at each point
- Output always watertight, bubbly-shaped
- Useful for range scanner data
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td><strong>Given normal at each point</strong></td>
</tr>
</tbody>
</table>

- **Mesh** methods

- **Crust** methods
<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td></td>
</tr>
</tbody>
</table>
- Start with Delaunay triangulation
- Take subset of the edges “on” the surface $S$
  (In fact, just take shortest edges in graph)
\( \alpha \)-shape triangulation

- Bad when samples unevenly spaced (can be fixed using weights on sample points)
- Works only for noise-free data
# Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td>Noise-free</td>
</tr>
</tbody>
</table>

Surface Reconstruction – p.25/60
### Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td>Noise-free</td>
</tr>
<tr>
<td>“Mesh” methods</td>
<td></td>
</tr>
</tbody>
</table>
Mesh methods

- Exploit local information to find a mesh $S$ approximating surface $U$
- Simplify mesh afterwards
Mesh methods

- Handles noisy data
- Assumes only sample dense near features (edges, bumps)
- Methods ad hoc; Rigorous analysis difficult
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td>Noise-free</td>
</tr>
<tr>
<td>“Mesh” methods</td>
<td>Sample Dense near Features</td>
</tr>
</tbody>
</table>
## Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td>Noise-free</td>
</tr>
<tr>
<td>“Mesh” methods</td>
<td>Sample Dense near Features</td>
</tr>
<tr>
<td>Crust Methods</td>
<td></td>
</tr>
</tbody>
</table>
Crust methods

- Focus of this lecture
- Assume only dense sampling
- Provide other information on $S$: Volume, Skeletal Structure
# Techniques for Surface Reconstruction

<table>
<thead>
<tr>
<th>Technique</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit parametric surface</td>
<td>Data fits model</td>
</tr>
<tr>
<td>Piece together parallel contours</td>
<td>Data from known device</td>
</tr>
<tr>
<td>Fit Gaussian Kernels</td>
<td>Given normal at each point</td>
</tr>
<tr>
<td>$\alpha$-shape Triangulation</td>
<td>Noise-free</td>
</tr>
<tr>
<td>“Mesh” methods</td>
<td>Sample Dense Near Features</td>
</tr>
<tr>
<td>Crust Methods</td>
<td>Sample Dense Near Features</td>
</tr>
</tbody>
</table>
Overview

- Introduction
- Weighted distance and power diagrams
- Medial Axis Transform
- PowerCrust Algorithm
Weighted Distance
and Power Diagrams
Weighted Distance and Power Diagrams

- Weighted distance
- Power Diagrams ( = Weighted Voronoi)
- $\alpha$-shapes
Unions of balls

Key concept:

Solids can be roughly approximated (exact in the limit) as a union of balls (discs in 2D).

Given a set of points $X$, we can view $X$ as centers of balls. How can we use this?
Unions of balls

Key concept:

Solids can be roughly approximated (exact in the limit) as a union of balls (discs in 2D).

Given a set of points $X$, we can view $X$ as centers of balls. How can we use this?

Try to visualize “shape” of $X$. 
Adding weights

Some points are “bigger” than others.

- $X$ sampled from a surface $\rightarrow$ points where sampling is less dense are “bigger”.
- $X = \text{centers of atoms in a molecule} \rightarrow \text{heavier atoms are “bigger”}.$
- In Power Crust (later), this will be crucial.

Each point $x \in X$ gets a weight $r_x$ (its radius).
Weighted Distance

- Work with weighted distance. Distance from a point \( p \) to a ball \((x, r_x)\):

\[
d_{r_x}(x, p) = d(x, p)^2 - r_x^2
\]
Normal Distance: \((p - x)^2\)

In 1D, the (normal) squared distance induced by each point \(x\) gives a parabola centered at \(x\).
\[ d_r(y, p) = (p - y)^2 - r^2 \]

Point \( y \) has radius \( r_y \) \( \longrightarrow \) parabola gets lowered to intersect axis at distance \( r \) from \( y \).
Power Diagram

- **Distance:** \( d_{rx}(x, p) = d(x, p)^2 - r_x^2 \)

- Weighted Voronoi cell of \((x, r_x)\) is set of points \(p\) that have smaller weighted distance to \(x\) than to any other point in \(X\):

\[
\text{cell}(x) = \{ p \mid d_{rx}(x, p) \leq d_{rx'}(x', p) \text{ for all } x' \in X \}
\]

- When all weights are equal, get the usual Voronoi diagram.
Power Diagram Demo

Demo
\[ d_r(y, p) = (p - y)^2 - r^2 \]

Point \( y \) has radius \( r_y \) \( \iff \) parabola gets lowered to intersect axis at distance \( r \) from \( y \).
Weighted Voronoi cell for $z$ doesn’t necessarily contain $z$. 

$$d_r(x, y) = (x - y)^2 - r^2$$
Some Voronoi cells may be empty!
Power Diagram Demo

More Demo
Intersections

Power diagram edges always go through the intersections of circles
Weighted Delaunay Complex

- Add an edge \( \{x, y\} \) if cells of \( x, y \) intersect
- Add a triangle \( \{x, y, z\} \) if cells of \( x, y, z \) intersect
- Add a tetrahedron...
Weighted Delaunay Complex
Weighted Distance and Power Diagrams

- Weighted distance
- Power Diagrams ( = Weighted Voronoi)
- $\alpha$-shapes
Dual Complex

- **Subset** of weighted Delaunay graph
- Only keep edge \((x, y)\) if balls at \(x, y\) intersect:
  \[ d(x, y) \leq r_x + r_y. \]
Changing the radii

- Fix a parameter $\alpha^2 \in (-\infty, \infty)$ (i.e. $\alpha \in \mathbb{C}$)
- Consider new radii $r'_x = \sqrt{r_x^2 + \alpha^2}$
- Power diagram stays the same since $d_{r'}(x, p) = d_r(x, p) - \alpha^2$.
- Weighted Delaunay graph stays the same.
- Dual Complex
  - grows if $\alpha^2 > 0$
  - shrinks if $\alpha^2 < 0$
As $\alpha^2$ grows from $-\infty$ to $\infty$, progress from empty graph to full weighted Delaunay graph:
As $\alpha^2$ grows from $-\infty$ to $\infty$, progress from empty graph to full weighted Delaunay graph:
α-shapes
α-shapes
Who cares?

This ordering is useful for visualizing structure of the point set.
Example: Simple surface reconstruction

Other apps: chemical modeling, visualization
Overview

- Introduction
- Weighted distance and power diagrams
  - Medial Axis Transform
  - PowerCrust Algorithm
Selected References

- [http://www.alphashapes.org](http://www.alphashapes.org)
- [http://www.geomagic.com](http://www.geomagic.com)
- [http://pages.cpsc.ucalgary.ca/~laneb/Power/](http://pages.cpsc.ucalgary.ca/~laneb/Power/)