Surface Reconstruction

Ligang Liu
Graphics&Geometric Computing Lab
USTC

http://staff.ustc.edu.cn/~lgliu
Introduction

• Rendering:

• Reconstruction:
Shape from “X”

- Shape from Structured Light
- Shape from Illumination
- Shape from Silhouette
- Shape from Stereo
- **Shape from Data**
Applications

• Reverse engineering
  – Industrial design
• Augmented reality
• Medical Imaging
• Human computer interaction
  – Realistic virtual environments
• Animation
• …
Problem

• **Input**
  – A set of points in 3D that sampled from a model surface

• **Output**
  – A 2D manifold mesh surface that closely approximates the surface of the original model
Problem
Desirable Properties

- No restriction on topological type
- Representation of range uncertainty
- Utilization of all range data
- Incremental and order independent updating
- Time and space efficiency
- Robustness
- Ability to fill holes in the reconstruction
Solutions

• Methods that construct triangle meshes directly — *Explicit methods*
  – Voronoi diagram and Delaunay triangulation
  – Zippering in 3D

• Methods that construct volumetric implicit functions — *Implicit methods*
  – Signed distances
  – Radial basis function reconstruction
  – Poisson reconstruction
General Pipeline

- Sampling
- Sample-Fusing
- Preprocessing
- Surface Reconstruction
- Postprocessing
Voronoi Diagram

- Voronoi Cell of $\mathbf{x}$
  - The set of points that are closer to $\mathbf{x}$ than to any other sample point
Medial Axis

Medial Axis:
- Find all circles that tangentially touch the curve in at least 2 points
- Medial axis = centers of all those circles
Simple Examples
Medial Axis vs. Voronoi Diagram
Definitions (1)

- **Voronoi cell**
  - A cell where all points in the cell are closer to a given sample point than any other point

- **Voronoi diagram**
  - A space partitioned into Voronoi cells

- **Voronoi vertex**
  - A point equidistant to d+1 sample points in $\mathbb{R}^d$
Definitions (2)

- Delaunay triangulation
  - Dual of Voronoi diagram
  - Each triangle’s circumcircle contains no other vertices

- Medial axis
  - Set of points with more than one closest point
Range Data
Optical Range Acquisition

• Strengths
  – Non-contact
  – Safe
  – Inexpensive (?)
  – Fast

• Limitations
  – Can only acquire visible portions of the surface
  – Sensitivity to surface properties
    • transparency, shininess, rapid color variations, darkness
  – Confused by interreflections
Range Images

• Construct a range surface

Range image  Tessellation  Range surface
Range Images
Registration

• Any surface reconstruction algorithm should strive to use all of the detail in all the available range data.
• Accurate registration may require:
  – Calibrated scanner/object positioning
  – Software-based optimization
  – Both
Registration
Registration as Optimization (1)

- Given two overlapping range scans, we wish to solve for the rigid transformation, $T$, that minimizes the distance between them.

\[ E = \sum_{i}^{N_P} \| Tq_i - p_i \|^2 \]
Registration as Optimization (2)

• If the correspondences are known a priori, then there is a closed form solution for T.

• This is not the case.
Registration as Optimization (3)

- Iterative close-point (ICP) solutions
  - Identify nearest points
  - Compute the optimal $T$
  - Repeat until $E$ is small
Registration as Optimization (4)

- In 3D:
Registration as Optimization (5)

• Sequential registration and integration is not optimal.

• Multiple range scans could be simultaneously registered. This provides greater information to assist in generating a more accurate registration.
Global Registration

- Network of range views
Registration Result

• Registration Error < Measurement Uncertainty
Curve from Points
Curve from Points
- Connect the Dots (1)

• Given unordered set of points P
  – connect them by linear segments
Curve from Points
- Connect the Dots (2)

• Can be ambiguous
• Harder when topology not known
Curve from Points
- Connect the Dots (3)

- Use Voronoi Diagram
- Construct Delaunay triangulation
- Which edges to choose?
Medial Axis

• Medial axis of \((d-1)\)-dimensional surface in \(R^d\) - set of points with more than one closest point on the surface

• Alternative definition: locus of centers of maximal inscribed spheres
Medial Axis and VD

- Voronoi diagram of set of points on curve approximates Medial Axis
  - if points sampled densely enough
Sampling Criterion (1)

- Good sample - sampling density (at least) inversely proportional to distance from medial axis
- \( r \)-sample: distance from any point on surface to nearest sample point \( \leq r \) * distance from point to medial axis
- In general, \( r \in (0, 1] \)
- \( r = 0.5 \) good enough
Sampling Criterion (2)

• Inherently takes into account
  – curvature of the surface
  – proximity of other parts of the surface
2D Crust Algorithm
Idea

• Adopt Delaunay edges which are “far” from MA
• To represent MA use Voronoi vertices
• Edge e in crust <=> circumcircle of e contains no other sample points or Voronoi vertices of S
Algorithm Process
Crust Algorithm

• Compute Voronoi diagram of $S$
  – let $V$ be set of Voronoi vertices
• Compute Delaunay triangulation of SUV
• Return all Delaunay edges between points of $S$
Theory

- **Theorem**: The crust of an $r$-sample from a smooth curve $F$, for $r \leq 0.25$ connects only adjacent samples of $F$
• The algorithm may fail when $r$ is too large
3D Crust Algorithm

A New Voronoi-Based Surface Reconstruction Algorithm
N. Amenta, M. Bern, and M. Kamvysselis
Siggraph 1998
Idea

• Extend 2D approach
• Voronoi cells are polyhedra
• Voronoi vertex is equidistant from 4 sample points
• BUT in 3D not all Voronoi vertices are near medial axis (regardless of sampling density)
Concepts

• Poles
  – Farthest Voronoi vertices for a sample point that are on opposite sides

• Crust
  – Shell created to represent the surface
Problem

- But some vertices of the Voronoi cell are near medial axis
- Intuitively – cell is closed not just from the sides but also from both sides of the surface
Observation

- $p^+ \equiv \text{pole of } p = \text{point in the Voronoi cell farthest from } p$
- $\varepsilon < 0.1 \rightarrow$
  - the vector from $p$ to $p^+$ is within $\pi/8$ of the true surface normal
  - The surface is nearly flat within the cell
Solution

• Solution
  – use only two farthest vertices of Vs - one on each side of the surface

• Call vertices poles of s: $p^+(s)$, $p^-(s)$
3D Algorithm

• Compute Voronoi diagram of $S$
• For each $s \in S$, identify the poles $p^+(s)$ and $p^-(s)$
  – $p^+(s)$ is the vertex of $V_s$ most distant from $s$
  – $p^-(s)$ is the vertex of $V_s$ most distant from $s$ in the opposite direction
• Let $P$ be the set of all poles
• Compute Delaunay triangulation $T$ of $S \cup P$
• Add to crust all triangles in $T$ with vertices only in $S$
Post-processing

• Delete triangles whose normals differ too much from the direction vectors from the triangle vertices to their poles
• Orient triangles consistently with its neighbors and remove sharp dihedral edges to create a manifold
Example

- Crust of set of points and poles used in its reconstruction
Examples

- Femur
  - 939 points
  - 2 minutes

- Golf club
  - 16,864 points
  - 12 minutes

- Bunny
  - 35,947 points
  - 23 minutes
Examples

Foot images from Prof. Dey
Advantages

• No need for experimental parameters in basic algorithm
• Not sensitive to distribution of points
Problems

• Sampling of points needs to be dense
  – Undersampling causes holes
• Problems at sharp corners
• Another way to choose poles gives better reconstruction
  – choosing the farthest and the second farthest Voronoi vertices, regardless of direction
• Correct, BUT slow
Q&A
Zippered Polygon Meshes From Range Images

Greg Turk and Mark Levoy
Siggraph’1994
Idea

- Use range scanner properties for reconstruction
- Single scan from given direction produces regular lattice of points in X and Y with changing depth (Z)
- Take multiple scans to create complete model
Algorithm

• Generate separate mesh from each scan (range image)
  – Use X & Y adjacency info
• Combine
  – Register positions
  – Merge meshes
Steps (1)

- Find quadruples of lattice points
- Form triangles
  - Find shortest diagonal
  - Form two triangles (test depth)
Steps (2)

• Avoid connecting depth discontinuities:
  – Test 3D distance between points when generating triangles
  – Do not generate if depth $\gg S$
Registration of Range Images

- Align corresponding portions of different range images
- Modified *iterated closest-point* (ICP) algorithm
- Initial alignment from camera positions (user)
Alignment (ICP)

- Find nearest position on mesh A to each vertex of mesh B
- Discard pairs of points that are too far apart
- Find rigid transformation that minimizes weighted least-squared distance
- Iterate until convergence
Point Matching

• Input:
  – 2 matching sets of 3D points \((M, D)\)
• Find rigid transformation (rotation+translation) which minimizes the distance between \(M\) and \(D\)
• Use Least-Squares

\[
\min_{R,T} \sum_i \left\| M_i - (RD_i + T) \right\|^2
\]
Weight Assignment

- Final surface will be weighted combination of range images
- Weights assigned at each vertex to:
  - Favor views with higher sampling rates
    - View direction is parallel to surface normal
  - Encourage smooth blends between range images
Weights for Smooth Blending

• To assure smooth blends, weights are forced to taper in the vicinity of boundaries
Integration: Mesh Zippering

• After registration have two overlapping meshes
• Need to combine into single connectivity
• Zippering
  – Remove overlapping portion of the mesh
  – Clip one mesh against another
  – Remove small triangles introduced during clipping
Removing Redundant Surfaces

• Remove “redundant” triangles until the meshes just meet
  – Triangle T in mesh A is redundant if
    • Hausdorff distance from it to B is within tolerance
    • Nearest points on B are not on boundary
Redundancy Removal

<Mesh A> clip boundary

Retain

Discard

Final triangles
Mesh Clipping

• Find intersection between boundary of B and mesh A
• Add intersection nodes to A and B
• Discard overlapping triangles
Example
Post-Processing

- Fill holes – local triangulation
- Remove small triangles – vertex removal
- Consensus geometry
  - Move each vertex to consensus position given by weighted average of positions from original range images
Consensus Geometry

Zippered geometry + range surfaces

Find vertex positions on range surfaces by intersection with consensus normal

Compute consensus normal

Compute weighted average of vertex positions
Examples (1)

- 10 range images, more than 160,000 triangles
Examples (2)

- 14 range images, more than 360,000 polygons
Comparison

• Crust is more formal (no heuristics)
  – Proven to generate good results
  – Slow
  – Require “correct” data

• Zippering – heuristic
  – Fast
  – Lot of small “fixes” / “tricks”
  – In practice works well
Discussions
Co-cones
- Cone with apex at sample point and aligned with poles
- Algorithm only requires one Voronoi diagram computation
- Eliminates normal trimming step
- Still does not support sharp edges
The power crust

- Use polar balls and power diagrams to separate the inside and outside of the surface
- Approximates medial axis
Detecting Undersampling
- Fat Voronoi cells or dissimilarly oriented neighboring Voronoi cells imply undersampling. Add sample points to accommodate
- This accounts for sharp edges and boundaries

Tight Co-cone
- After detecting undersampling, stitch up holes
Discussions
Volumetrically Combining Range Images

Siggraph 1996
Overview

• Convert range images to signed distance functions
• Combine signed distance functions
• Carve away empty space
• Extract hole-free isosurface
Signed Distance Function
Combining SDF
Merging Surface
Least Squares Solution

Range surface #1
Least Squares Solution

\[ E(f) = \sum_{i=1}^{N} \int d_i^2 (x, f) dx \]

Finding the \( f(x) \) that minimizes \( E \) yields the optimal surface.
This \( f(x) \) is exactly the zero-crossing of the combined signed distance functions.
Isosurface Extraction
Hole Filling

• The procedure so far will reconstruct a mesh from the observed surface. Unseen portions will appear as holes in the reconstruction

• We can fill holes in the polygonal model directly, but such methods:
  – are hard to make robust
  – do not use all available information
Space Carving

Without space carving

With space carving
Space Carving
Results (1)

No hole filling

Hole filling – no backdrop

Hole filling with backdrop

Smoothed
Local Result

- No hole filling
- No backdrop
- With backdrop
- Smoothed
Results (2)
Local Result
Limitations

• Minimum thickness and edge sharpness have limits
Merging 12 views of a drill bit
Discussions
More (1)

- Zero sets
  - Using input points, define implicit signed distance function
  - Distance function is interpolated and polygonized using marching cubes
  - Approximation rather than interpolation
More (2)

• Delaunay Sculpting
  – Remove tetrahedra from Delaunay triangulation
  – Associate values to tetrahedra and eliminate largest valued ones
More (3)

• Alpha Shapes
  – Given a parameter, $\alpha$, connect vertices within $\alpha$ units
  – Subset of Delaunay triangulation
  – Generalized convex hull
More (4)

• Moving least square fitting – Siggraph05
More (5)

• RBF fitting – Siggraph01
More (6)

- Greg Turk – Siggraph99, TOG01
More (7)

- Hughes Hoppe – Siggraph92
More

• So many papers
# Summary

<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>
Marching Cubes
Overview

- Marching cubes: method for approximating surface defined by isovalue $\alpha$, given by grid data

- **Input:**
  - Grid data (set of 2D images)
  - Threshold value (isovalue) $\alpha$

- **Output:**
  - Triangulated surface that matches isovalue surface of $\alpha$
Voxels

- Voxel – cube with values at eight corners
  - Each value is above or below isovalue $\alpha$
  - Method processes one voxel at a time
- $2^8 = 256$ possible configurations (per voxel)
  - reduced to 15 (symmetry and rotations)
- Each voxel is either:
  - Entirely inside isosurface
  - Entirely outside isosurface
  - Intersected by isosurface
Algorithm

- First pass
  - Identify voxels which intersect isovalue

- Second pass
  - Examine those voxels
  - For each voxel produce set of triangles
    - approximate surface inside voxel
Configurations
Configurations

- For each configuration add 1-4 triangles to isosurface

- Isosurface vertices computed by:
  - Interpolation along edges (according to pixel values)
    - better shading, smoother surfaces
  - Default – mid-edges
Example
MC Problems

- Marching Cubes method can produce erroneous results
  - E.g. isovalue surfaces with “holes”
- Example:
  - voxel with configuration 6 that shares face with complement of configuration 3:

![Diagram](image.png)

*Figure 3. An example illustrating the flaw in the marching cubes method.*
Discussions
Postprocessing

• Smoothing
  – Coming soon…
References

• Greg Turk and Marc Levoy, Zippered Polygon Meshes from Range Images, SIGGRAPH 94, 311-318.
• Brian Curless and Marc Levoy, A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH 96.
Q&A