Sinogram Polygonizer for Reconstructing 3D Shapes

Hiromasa Suzuki, Daiki Yamanaka, Yutaka Ohtake, The University of Tokyo, Japan

Industrial X-rays CT for 3D Scanning

- Scanning
- CT Reconstruction
- Surface Extraction
- Dimensional X-rays CT
- Dimensional measurement
- reverse engineering
X-ray CT Scanning

micro focus
cone beam type

X-ray source
Rotation table
Detector
Sinogram
(projection images)
Sinogram vs Tomogram

Sinogram (projection images)

Tomogram (volume)

pixel: projection value

voxel: CT value

CT reconstruction
Filtered Back-Projection

CT value at point $p$ in the tomogram:

$$f(p) = \int_{0}^{2\pi} h \ast \text{ProjectionValue}(\theta, p) d\theta$$
Filtered Back-Projection

CT value at point $p$ in the tomogram:

$$f(p) = \int_0^{2\pi} h \ast \text{ProjectionValue}(\theta, p) \, d\theta$$
Volume Model

Sinogram

Volume model

voxel value = CT value at a grid point

Compute CT values at all the grid points

\[ f(p) = \int_0^{2\pi} h * \text{ProjectionValue}(\theta, p) d\theta \]

grid point
Surface Extraction

- Grid-based polygonizer
  - Marching cubes  [Lorensen and Cline 1987]
  - Dual contouring  [Ju et al. 2002]

- Generated surface has grid artifact
  - Grid is not aligned to material boundary
**Sinogram Polygonizer**

**Surface Extraction from Sinogram**

![Diagram showing sinogram polygonizer and surface extraction process.]

- **Sinogram**
- **Our approach**
  - Directly generate surface from **sinogram**
- **no grid artifact**
Surface Extraction from Sinogram

CT values computed on regular grid

\[ f(p) = \int_0^{2\pi} h \cdot \text{ProjectionValue}(\theta, p) \, d\theta \]

grid point
Surface Extraction from Sinogram

CT value can be computed at arbitrary positions
Surface from Unstructured Grid

Use unstructured grid (tet mesh) for CT reconstruction
Surface from Unstructured Grid

Surface can be aligned to the material boundary.

Sinogram

grid based

ours

sharp features
Algorithm
Algorithm Overview

Sinogram → CT reconstr. → Thresholding → Surface extraction → Vertex updating → Surface mesh

Remeshing
Iterative Optimization
Iterative Optimization
Thresholding & Boundary Extraction

• Computer CT values at the centers of the triangles

• Binarize with threshold CT value
  ‣ Assume two material (air, object)

\[
F_{\text{threshold}} = \frac{F_{\text{air}} + F_{\text{material}}}{2}
\]

- \( F_{\text{air}} \): CT value of air
- \( F_{\text{material}} \): CT value of object

• Extract boundary mesh
Move Vertex to Iso-surface

\[ f(m'_j) = F_{\text{threshold}} \]

\[ E(p_i) = \sum_{t_j \in N(p)} \left\{ n_j \cdot (p_i - m'_j) \right\}^2 \]

QEM energy

\[ n_j = \nabla f(m'_j)/|\nabla f(m'_j)| \]
Move Vertex to Iso-surface

Boundary vertices $p_i$

$f(m'_j) = F_{\text{threshold}}$

Boundary mesh

Material boundary

$E(p_i) = \sum_{t_j \in \mathcal{N}(p)} \{n_j \cdot (p_i - m'_j)\}^2$

QEM energy

$n_j = \nabla f(m'_j)/|\nabla f(m'_j)|$
Remeshing

- Low quality triangles can be generated

- Optimal Delaunay Triangulation (ODT) [Alliez 2005]
  - Vertices are moved to area-weighted average of 1-ring neighbor circumcenters
  
  \[ x_i^* = \frac{1}{|\Omega_i|} \sum_{T_j \in \Omega_i} T_j c_j \]

  \( T_i \) : Area of triangle

  \( c_j \) : Circumcenter

  \( \Omega_i \) : Neighbor triangles

Low quality triangle

After several iterations
Gradient from Sinogram

• Gradient of CT value must be estimated to minimize QEM energy

\[ n_i = \frac{\nabla f(m')}{|\nabla f(m')|} \quad E(p) = \sum_{i \in N(p)} \{n_i \cdot (p - m_i')\}^2 \]

• Standard gradient estimation
  ▶ Finite difference method (numerical)

• Our method

\[ f(p) = \int_0^{2\pi} h \ast \text{ProjectionValue}(\theta, p) \, d\theta \]

Differentiate with \( p \) (analytically)

\[ \nabla f(p) = \int_0^{2\pi} \alpha(\theta, p) R_{\theta} \left( \nabla_{x,y} h \ast \text{ProjectionValue}(\theta, p) \right)_0 \, d\theta \]

gradient of the sinogram

more accurate gradient
Initial adaptive meshing

- Adaptive sampling near boundary

almost same quality

uniform meshing (#tet 58M)

x7 faster

adaptive meshing (#tet 8.5M)
Convergence

- Test for three different models
- Experimentally, 20 iterations are enough to converge
Results
Sharp features

broken squirrel model  “Real” sharp feature
Comparison

• Simulation data & real data
  - Simulated sinogram
  - Pedal (Al)
  - Sinogram

• Compare 3 methods
  1. Marching Cubes
  2. Dual Contouring
  3. Sinogram Polygonizer
Comparison: simulation data

<table>
<thead>
<tr>
<th>Method</th>
<th>Hausdorff error [%]</th>
<th>#vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marching Cubes</td>
<td>0.625</td>
<td>24,654</td>
</tr>
<tr>
<td>Dual Contouring</td>
<td>0.578</td>
<td>24,567</td>
</tr>
<tr>
<td>Sinogram Polygonizer</td>
<td>0.156</td>
<td>5,649</td>
</tr>
</tbody>
</table>

Accurate shape with fewer mesh, good quality!
Comparison: real data

Marching Cubes #vertex:20K
Dual Contouring #vertex:20K
Sinogram Polygonizer #vertex:6K
Applications

- Meshing extremely complex objects
- Porous material
- 3D copy fabrication
- FEM simulation
Conclusion

• “Sinogram Polygonizer“
  ▶ Use **Sinogram** which is raw data of CT scanner
  ▶ Reconstruct on **unstructured** grid and deform it
  ▶ Generate accurate shape with **sharp features** and **good approximation**
Limitation & future work

• Only single material
  - Multi-material segmentation method should be included

• Inner mesh quality
  - Proposed method can generate tetrahedral mesh
  - Slivers occasionally appear
  - If post-processing is integrated, generated mesh can be used for FEM
Thank you for your kind attention!