Segmentation and Modeling

Russell H. Taylor
John C. Malone Professor of Computer Science, with joint appointments in Mechanical Engineering, Radiology & Surgery
Director, Laboratory for Computational Sensing and Robotics
The Johns Hopkins University
rht@jhu.edu

Segmentation & Modeling

Images  →  Segmented Images  →  Models
Example: Bone Modeling from CT

Contours

CT Slices

Density Function $f_n$

Density Model

Tetrahedral Mesh

Multiple Resolution Model

Tetrahedral Mesh Simplification

Multiple Images

Segmentation

Model construction

Visualization

Planning

Navigation
Segmentation

- Process of identifying structure in 2D & 3D images
- Output may be
  - labeled pixels
  - edge map
  - set of contours
Automation Approaches

- Pixel-based
  - Thresholding
  - Region growing
- Edge/Boundary based
  - Contours/boundary surface
  - Deformable warping
  - Deformable registration to atlases

- Extremely time-consuming (~6 hours per case)
- 3D Imagery – Performed slice at a time
- Some structures near impossible (blood vessels)
### Thresholding

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>22</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>12</td>
<td>11</td>
<td>15</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Thresholding

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>22</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>12</td>
<td>11</td>
<td>15</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### "Partial volume" effects

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>80</th>
<th>60</th>
<th>90</th>
<th>100</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td>90</td>
<td>55</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>90</td>
<td>55</td>
<td>0</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>55</td>
<td>0</td>
<td>70</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>50</td>
<td>45</td>
<td>100</td>
<td>98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
“Partial volume” effects

Copyright © 1999-2015 R. H. Taylor
“Partial volume” effects

Segment statistically

- Measure distribution of intensities at known tissue locations
- Use nearest neighbor style classifiers for all other voxels
Standard Scans

Statistical segmentation
**Between Scylla and Charybdis**

- Problem: imagery contains non-linear gain artifacts that shift the intensity values in a non-stationary way
- If one knew the gain field, could correct image and use standard statistical method
- If one knew the tissue types, could predict the image and find the gain field correction
- Solution: Use Expectation/Maximization method to iteratively solve for gain field and tissue class, using probabilistic models

---

**EM-Segmentation [Wells 1994]**

**E-Step**
- Compute tissue posteriors using current intensity correction.

**M-Step**
- Estimate intensity correction using residuals based on current posteriors.
Standard Scans

Statistical segmentation
Deformable Surfaces

Geodesic Active Contours

- Snake methodology defines an energy function $E(C)$ over a curve $C$ as
  \[ E(C) = \beta \int |C'(q)|^2 dq - \lambda \int |\nabla I(C(q))| dq \]

- Caselles, et al. reduced the minimization problem to the expression
  \[ \min_{C(q)} \int g(|\nabla I(C(q))|) |C'(q)| \, dq \]
  where $g$ is a function of the image gradient of the form $\frac{1}{1 + |\nabla I|^p}$.

- The following curve evolution equation can be derived using Euler-Lagrange
  \[ \frac{\partial C}{\partial t} = g \kappa N' - (\nabla g \cdot N)N' \]
  where $\kappa$ is the curvature and $N$ is the normal.

- By defining an embedding function $u$ of the curve $C$, the update equation for
  the higher dimensional surface is given by (Osher, Sethian ‘88):
  \[ \frac{\partial u}{\partial t} = g \kappa |\nabla u| + \nabla u \cdot \nabla g \]
Deformable Surfaces

Figure 4.7. Segmentation of vertebrae defined by a set of CT slices. Four steps of the deformation of a roughly spherical medullary spline toward the vertebrae are shown.

Credit: Prince & Davatzikos
Example: Bone Modeling from CT

- Contours
- CT Slices
- Tetrahedral Mesh
- Density Model
- Density Function $f_n$
- Tetrahedral Mesh Simplification
- Multiple Resolution Model

Bone Structure

- Compact bone
- Spongy bone
- Medullary Cavity

Credit: Yao and Taylor
Bone Contour Extraction

- Deformable Contour Algorithm (Snake)
- \( F = F_{\text{internal}} + F_{\text{image}} + F_{\text{external}} \)
  - \( F_{\text{internal}} \): the spline force of the contour
  - \( F_{\text{image}} \): the image force
  - \( F_{\text{external}} \): an external force
- Semi-automatic
Bone Contour Extraction
Closer-up view

Needle graph of Image force
Bone Contours
Credit: Yao and Taylor

Curve Representation:
(Level Sets)

Credit: Eric Grimson
Example of current research:
• New data structure (“Springls”) and associated algorithmic formulation combining aspects of level sets and meshes
• Blake Lucas, October 2010

Anatomy of a Springl
Blake Lucas

Deformable Surfaces & Level Sets

Segmenting with Spatial Priors

1. Step
- Generic Case
- Registration
- Align Atlas
- EM-MF

2. Step
- Label Map

- Given standard scan, and probability maps of tissue types
- Elastically register standard scan to new case
- Apply transformation to all probability maps
- Use as prior probabilities in EM-MF segmentation
- Apply in hierarchical manner (first segment out major structures, then substructures)

Credit: Eric Grimson
Modeling

- Representation of anatomical structures
- Models can be
  - Images
  - Labeled images
  - Boundary representations

FROM VOXELS TO SURFACES

Representing solids:
- B-REP - surface representation,
  d/s of vertices, edges, faces.
- CSG- composition of primitive solids

binary image  \rightarrow  B-REP representation

Surface construction algorithms:
- 2D-based algorithms
- 3D-based algorithms
Surface Representations

- Implicit Representations
  \[ \{ \mathbf{x} | f(\mathbf{x}) = 0 \} \]
- Explicit Representations
  - Polyhedra
  - Interpolated patches
  - Spline surfaces
  - ...

Source: CIS p 73 (Lavallee image)

Polyhedral Boundary Reps

- Common in computer graphics
- Many data structures.
  - FEV lists
  - Winged edge
  - Connected triangles
  - etc.
FEV lists

- Explicit linked lists of faces, edges, vertices
- Many variations
- Key properties
  - Convenient to traverse
  - Lists are variable length
  - Can be tricky to maintain consistency

Winged Edge

- Baumgart 1974
- Basic data structures
  - winged edge (topology)
  - vertex (geometry)
  - face (surfaces)
- Key properties
  - constant element size
  - topological consistency
Connected Triangles

- Basic data structures
  - Triangle (topology, surfaces)
  - Vertex (geometry)
- Properties
  - Constant size elements
  - Topological consistency
Tetrahedral Mesh Data Structure

- **Vertex list**
  - x, y, z coordinates
  - reference to one tetrahedron
- **Tetrahedron list**
  - references to four vertices
  - references to four face neighbors
- **Properties such as density functions**

Advantages of Tetrahedral Mesh

- Greatest degree of flexibility
- Data structure, data traversal, and data rendering are more involved
- Ability to better adapt to local structures
- Computational steps such as interpolation, integration, and differentiation can be done in closed form
- Finite element analysis
- Hierarchical structure of multiple resolution meshes
2D-based Methods for Shape Reconstruction

• Treat 3D volume as a stack of slices
• Outline
  – Find contours in each 2D slice
  – Match contours in successive slices
  – Connect contours to create tiled surfaces (for boundary representation)
  – Use contours to guide subdivision of space between slices into tetrahedra (for volumes)

SURFACE CONSTRUCTION ALGORITHMS

2D-based algorithms

1. 2D contour extraction
2. tiling of contours


Contour extraction

• Sequential scanning
• boundary following (random access to pixels)
Example: Bone Modeling from CT

Contours → Tetrahedral Mesh → Density Model

Density Function $f_n$ → Tetrahedral Mesh Simplification

Multiple Resolution Model

CT Slices

Construct Tetrahedral Mesh from Contours

CT Slices → Contour Extraction

Tiling → Branching → $f(x)$ → Raw Tetrahedral Mesh Model → Smoothing

Tetrahedral Mesh Reconstruction from Contours

Credit: Yao and Taylor
Tetrahedral Mesh Tiling

- **Objectives**
  - Subdivide the space between adjacent slices into tetrahedra, slice by slice
- **Method**
  - Two-steps tiling strategy
    - 2D tiling and medial axis tiling
    - 3D tiling
**Metric Functions**

- Maximize Volume, $f_v$
- Minimize Area, $f_a$
- Minimize Density Deviation, $f_d$
- Minimize Span Length, $f_s$

Current Metric Function:
- Combination of minimizing density deviation and span length
- Minimize $F = w_1 * f_d + w_2 * f_s$

**Tiling Constraints**

- Non-intersection between tetrahedra
- Continuity between slices
- Continuity between layers
**Correspondence Problem**

- Examining the overlap and distance between contours on adjacent slices
- Graph based method

![Contour Correspondence](Image)

**Branching Problem**

- Branching Between layers
  - Convert to tiling of 3 contours
- Branching Between contours
  - Composite contour
  - Split contour

![Composite Contour](Image)  ![Split Contour](Image)
3D-based methods for Surface Reconstruction

- Segment image into labeled voxels
- Define surface and connectivity structure
- Can treat boundary element between voxels as a face or a vertex

3D-BASED ALGORITHMS

Block-form and Beveled-form representations of surface:
Block form methods

- “Cuberille”-type methods
- Treat voxels as little cubes
- May produce self-intersecting volumes
- E.g., Herman, Udupa

Ref: Udupa, CIS Book, p47
Beveled form methods

- “Marching cubes” type
- Voxels viewed as 3D grid points
- Vertices are points on line between adjacent grid points
- E.g. Lorensen&Cline, Baker, Kalvin, many others

Block form to beveled form

Segmented voxels
Block form to beveled form

Duality between voxels and vertices on adjacency graph

Block form to beveled form

Label vertices based on segmentation labels
Block form to beveled form

Label vertices based on segmentation labels

Boundary crosses edges between occupied and empty vertices
Block form to beveled form

Boundary crosses edges between occupied and empty vertices

Note: Choice of exact vertex placement is somewhat arbitrary. One choice is linear interpolation along edge based on density.

Beveled-form Algorithms and medical Imaging

Classification by definition of vertex adjacency (boundary element adjacency).

Vertex adjacency can be calculated:
1. Inconsistently
2. Tetrahedral tessellation
3. Supertrapping
4. Voxel topology best for 3D medical applications.
Beveled form basic approach

- Segment the 3D volume
- Scan 3D volume to process “8-cells” sequentially
- Use labels of 8 cells as index in (256 element) lookup table to determine where surfaces pass thru cell
- Connect up topology
- Use various methods to resolve ambiguities

Marching Cubes

- Lorensen & Kline
- Probably best known
- Used symmetries to reduce number of cases to consider from 256 to 15
- BUT there is an ambiguity
Wyvill, McPheters, Wyvill

Step 1: determine edges on each face of 8 cube

![Figure 6: The seven cases for calculating vertices and faces](image)

Step 2: Connect the edges up to make surfaces

**Ambiguities**

- Arise when alternate corners of a 4-face have different labels
- Ways to resolve:
  - supersampling
  - look at adjacent cells
  - tetrahedral tessellation
Tetrahedral Tessalation

- Many Authors
- Divide each 8-cube into tetrahedra
- Connect tetrahedra
- No ambiguities

Figure 8: The two tetrahedral partitionings of an 8-cell.

Figure 9: The two cases used for surface construction.

Bounded-form algorithms based on the tetrahedral decomposition of the 3D volume have been developed by Payne et al [33], and Wicke [34]. While this approach does provide a neat resolution to the ambiguous 8-cell problem, it

Alligator Algorithm

- Phase 1: Initial Construction
- Phase 2: Adaptive Merging
ALLIGATOR ALGORITHMS

Phase 2 - Adaptive face merging

Algorithm exploits the following:
1. beveled-form property:
   • each vertex lies on 4 faces
   • only 2 possible ways for a vertex to lie on 4 regular faces.
2. Euler operators
   • simple, high-level operations
   • efficient
   • simplifies proof of correctness (e.g. topological genus)

Source: C. Cutting, CIS Book
Tetrahedral Mesh Smoothing

- Motivations
  - Noise/discretazition in CT data set
  - Artifacts during segmentation
Classic Laplacian Smoothing Method

- Equation
  \[ v'_i = \frac{1}{|N_i|} \sum_{j \in N_i} v_j \]

- Advantages
  - Fast and easy

- Drawbacks
  - Shrinkage
  - Invalid elements

Enhanced Laplacian Smoothing Method

- Objective
  - Reduce shrinkage

- Method
  - Project back to boundary

\[ v'_i = proj \left( \frac{1}{|N_i|} \sum_{j \in N_i} v_j \right) \]
Average and reproject

Average and reproject
Enhanced Laplacian Smoothing Method

- **Objective**
  - Prevent invalid element
- **Method**
  - Iterative assignment

\[
v_i^{(0)} = \text{proj}\left(\frac{1}{|N_i|} \sum_{j \in N_i} v_j\right)
\]

\[
v_i^{(k)} = \alpha \cdot v_i + (1 - \alpha) v_i^{(k-1)}, 0 \leq \alpha \leq 1
\]

Mesh Smoothing Results

a) Before Smoothing  
b) After Smoothing
Tetrahedral Mesh Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Num of Vertices</th>
<th>Num of Tetrahedra</th>
<th>Num of Slices</th>
<th>Total Num of Voxels inside</th>
<th>Avg Num of Voxels Per Tetra</th>
<th>Volume (mm$^3$)</th>
<th>Avg Vol. Per Tetra (mm$^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Femur</td>
<td>6163</td>
<td>31,537</td>
<td>83</td>
<td>1,802,978</td>
<td>57.1</td>
<td>312,107</td>
<td>9.9</td>
</tr>
<tr>
<td>Pelvis</td>
<td>8219</td>
<td>32,741</td>
<td>110</td>
<td>1,941,998</td>
<td>59.3</td>
<td>347,070</td>
<td>10.6</td>
</tr>
</tbody>
</table>

Example: Bone Modeling from CT

Contours → Tetrahedral Mesh → Density Model

CT Slices → Tetrahedral Mesh Simplification → Multiple Resolution Model
Density Functions

- n-degree Bernstein polynomial in barycentric coordinate

\[ D(\mu) = \sum_{i+j+k+l=n} C_{i,j,k,l} B^n_{i,j,k,l}(\mu) \]

- \( C_{i,j,k,l} \) polynomial coefficient

\[ B^n_{i,j,k,l}(\mu) = \frac{n!}{i!j!k!l!} \mu_i^i \mu_j^j \mu_k^k \mu_l^l \] barycentric Bernstein basis

Barycentric Coordinate of Tetrahedron

- Local coordinate system
- Symmetric and normalized
- Every 3D position can be defined by an unique coordinate \((x, y, z, w)\)

\[ V = x*V_a + y*V_b + z*V_c + w*V_d \]

\[ x+y+z+w=1, \ V_a, \ V_b, \ V_c, \ V_d \text{ are coordinate of Tetrahedron vertices} \]

\( x, y, z, w \text{ within}[0,1] \) if \( V \) is inside the tetrahedron
Density Functions

- Advantages
  - Efficient in storage
  - Continuous function
  - Explicit form
  - Convenient to integrate, to differentiate, and to interpolate

Fitting Density Function

- Minimize the density difference between the density function and CT data set

\[
\min_{\rho_i \in \Omega} \left( \sum_{i,j,k,l=1}^{s} C_{i,j,k,l} \cdot B_{i,j,k,l}(\mu, \rho) - T(\rho) \right)^2
\]

\(\Omega\) is the set of sample voxels, \(T(\rho)\) is the density value from the CT data set.

\[
\begin{bmatrix}
B_1(\mu_{p1}) & B_2(\mu_{p1}) & \ldots & B_m(\mu_{p1}) \\
B_1(\mu_{p2}) & B_2(\mu_{p2}) & \ldots & B_m(\mu_{p2}) \\
\vdots & \vdots & \ddots & \vdots \\
B_1(\mu_{ps}) & B_2(\mu_{ps}) & \ldots & B_m(\mu_{ps}) \\
\end{bmatrix}
\begin{bmatrix}
C_1 \\
C_2 \\
\vdots \\
C_m \\
\end{bmatrix}
= 
\begin{bmatrix}
T(\rho_{p1}) \\
T(\rho_{p2}) \\
\vdots \\
T(\rho_{ps}) \\
\end{bmatrix}
\]

\(s\): number of sample voxels
\(m\): number of density function coefficient,
\(s > 2m\)
Accuracy vs Degree of Density Function

- Use CT data set as ground truth
- Cut an arbitrary plane through the model

Arbitrary Cutting Plane

Partitions by tetrahedra on cutting plane

Accuracy vs Degree of Density Function (cont’)

<table>
<thead>
<tr>
<th>Degree</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coeff Number</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>20</td>
<td>35</td>
<td>56</td>
<td>84</td>
<td>120</td>
<td>165</td>
</tr>
<tr>
<td>Avg. Density Err (%)</td>
<td>3.291</td>
<td>1.583</td>
<td>0.766</td>
<td>0.442</td>
<td>0.298</td>
<td>0.216</td>
<td>0.167</td>
<td>0.149</td>
<td>0.128</td>
</tr>
</tbody>
</table>


Example: Bone Modeling from CT

Contours → Tetrahedral Mesh → Density Model → Tetrahedral Mesh Simplification → Multiple Resolution Model

Density Function $f_n$

CT Slices

Model Simplification

- Models used in CIS must be guaranteed to be accurate within known bounds
- But 3D models from medical images often are very complex, and require representations with large data structures.
- Algorithms using models are often computationally expensive, and computation costs go up with model complexity
- **PROBLEM:** reduce model complexity while preserving adequate accuracy

~350,000 triangles!
Model simplification

- Problem is also common in computer graphics
  - There is a growing literature
  - But many graphics techniques only care about appearance, and do not necessarily preserve accuracy or other properties (like topological connectivity) important for computational analysis

- Broadly, two classes of approaches
  - Top down
  - Bottom-up

Top down

- Active surfaces used in segmentation

- Deformable registration of an atlas to a patient
  - E.g., skull atlas discussed in craniofacial lecture had about 5000 polygons (perhaps 15-20,000 triangles)

- Recursive approximations
  - E.g., Pizer et al. “cores”
Bottom up methods

• Typically, start with very high detail model generated from CT images
  – Large number of elements a consequence of small size of pixels & way model is created

• Then merge nearby elements into larger elements
  – E.g., “decimation” (Lorensen, et. al.)
  – E.g., “superfaces” (Kalvin & Taylor)
  – E.g., Gueziec

• An excellent tutorial may be found in:

Source: David Luebke; A Developer’s Survey of Polygonal Simplification Algorithms; IEEE Computer Graphics and Application, May 2001