Segmentation-1

Classification, Segmentation, Labeling and Modeling: Classical Approaches

Jerry L. Prince
Image Analysis and Communications Laboratory
Johns Hopkins University

Slide deck structure and many slides taken from Russell H. Taylor “Segmentation and Modeling” slides.
Segmentation & Modeling

Images → Segmented Images → Models

Credit: Russell H. Taylor
Stages of IGS

1. Multiple images
2. Segmentation
3. Model construction
4. Tissue properties
5. Visualization
6. Planning
7. Navigation

Credit: Eric Grimson
Example: Bone Modeling from CT

Contours → Tetrahedral Mesh → Density Model

Density Function \( f_n \)

Tetrahedral Mesh Simplification

Credit: Russell H. Taylor

CT Slices → Multiple Resolution Model

© Jerry L. Prince
Segmentation Variants

- Image segmentation: divide an image into regions
- Voxel classification: cluster voxel intensities by type
- Voxel labeling: identify voxel anatomy/type by name

Object versus background

Gray matter intensities

Cortical gyri
Case Study: Cluster histograms on T2
Case Study: Cluster histograms on T2
Case Study: Cluster histograms on T2
Case Study: Cluster histograms on T2

T2 image histogram

Wm
Gm
Csf

count

Intensity

0 50 100 150 200 250
Manual Segmentation (Outlining)

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

- **Voxel-based versus boundary-based:**
  - Voxel-based: voxels are labeled (object vs. background)
  - Boundary-based: boundaries are identified

- **Supervised versus unsupervised:**
  - Supervised: examples are provided
  - Unsupervised: the data itself provides all information

- **Automatic versus semi-automatic**
  - Automatic: no user information provided
  - Semi-automatic: some user information is provided
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</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>5</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Credit: Russell H. Taylor
Thresholding (T>10)

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

Credit: Russell H. Taylor
Thresholding: $T > 9$

Credit: Russell H. Taylor
Thresholding: $T > 11$
Region Growing

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</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>5</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Region Growing: Start with $T>10$

100% sure these are in the object

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</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>5</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Region Growing: +/-2 from 10

Seed value = 10

Go around the boundary and include values within 2 of 10

Using nearest neighbor rule = 4-connectivity (not 8-connectivity)

<table>
<thead>
<tr>
<th>3</th>
<th>5</th>
<th>7</th>
<th>3</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<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>5</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Region Growing: +/-2 from 10 again

Seed value = 10

Go around the boundary and include values within 2 of 10

Region complete
Region Growing: Start with $T>10$

An alternative approach looks for differences across the boundary. Kind of like looking for large gradients on the boundary.

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

100% sure these are in the object
Region Growing: (bndy within +/- 2)

Checking along the boundary for weak transitions.

Using nearest neighbor rule = 4-connectivity (not 8-connectivity)
Region Growing: (same rule)

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</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>5</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>17</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>18</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Region Growing: (again and again)

Creates a fairly large object. Defined primarily by starting points and boundary tolerance: +/-2

This is a common problem called “weak edges”
“Partial volume” effects: true object

Credit: Russell H. Taylor
"Partial volume" effects: digital object

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td>90</td>
<td>55</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td>55</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>70</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>45</td>
</tr>
</tbody>
</table>

Credit: Russell H. Taylor
“Partial volume” effects: problem?

Credit: Russell H. Taylor
“Partial volume” effects: segmentation

Credit: Russell H. Taylor
“Partial volume” effects: bad topology

Credit: Russell H. Taylor
What is Voxel Classification?

- Classification = label each voxel according to the type of tissue that it represents.
- Two distinct and important results in medical imaging:
  - hard classification
  - soft classification
- Soft classification accounts for partial content of tissues.
Uses of Classification

- Automatic brain isolation
- Cortical segmentation
- Quantification of tissue volumes
  - population differences
  - individual changes in health and disease
Desirable Properties

- Automated; efficient
- Accurate and reproducible

Robust to
- noise,
- partial volume effects,
- intensity shading

- Accommodate variability of anatomy
- Use multiple tissue contrasts when available
Broad Categories

• Supervised classification
  – uses training data with known labels to partition feature space—e.g., image intensities—into tissue classes

• Unsupervised classification
  – does not use training data
  – instead relies on data models and clustering of the feature space
Supervised classifiers

- **Supervised Classifiers** partition a feature space derived from the image by using data with known labels.
- Known as supervised methods because **training data** is required.
- Applicable to multichannel image data, multiple classes.
Supervised classifier steps

1) Acquire data
2) Extract feature space
3) Determine number of classes
4) Generate training data
5) Define classifier criteria
6) Apply classifier to data

• Common supervised classifiers: K-NN, decision trees and random forests, Bayesian classifiers, neural networks, support vector machines
Training data

- Training data is segmented and labeled data that is used to guide the classifier
- Typically obtained from manual segmentation
k-Nearest Neighbor classifier

- Nonparameteric classifier
- Classification by a majority vote scheme
- $k$ is the number of nearest neighbors from which to allow a vote
k-Nearest Neighbor steps

• Can carry out:
  • on one data point at a time
  or
  • on the histogram space of the image:

1) Specify $k$

2) Determine distance measure

3) Find the $k$ data points within the training data with the smallest distance from the data point to be classified

4) Classify the data point according to a majority vote, set $k=k-1$ if tie occurs
Balancing selection of $k$

- What happens when $k$ is too small?
- What happens when $k$ is too big?
Statistical Supervised Classifiers

• Goal: determine posterior probability $p(class \ label \ | \ data)$

• Refresher (conditional probability)

$$p(C|f) = \frac{p(C, f)}{p(f)} = \frac{p(f|C)p(C)}{p(f)}$$

• How do you determine $p(f)$?

$$p(C|f) = \frac{p(f|C)p(C)}{\sum_C p(f, C)} = \frac{p(f|C)p(C)}{\sum_C p(f|C)p(C)}$$

• Why is Bayes’ theorem useful?

• What is $\sum_C p(C|f)$?
Gaussian supervised classifier

- Assume each class consists of image intensities represented by a Gaussian distribution

- Univariate Gaussian distribution

\[ p_f(f | c) = \frac{1}{\sqrt{2\pi \sigma_c^2}} \exp\left(- \frac{(f - \mu_c)^2}{2\sigma_c^2}\right) \]

- Multivariate Gaussian distribution

\[ p_f(f | c) = \frac{1}{(2\pi)^{N/2}|\Sigma_c|^{1/2}} \exp \left( -\frac{1}{2}(f - \mu_c)^T \Sigma_c^{-1}(f - \mu_c) \right) \]
Gaussian supervised classifier

- Assume each class consists of image intensities represented by a Gaussian distribution

- Univariate Gaussian distribution

\[ p_f(f|c) = \frac{1}{\sqrt{2\pi \sigma^2_c}} \exp\left(-\frac{(f-\mu_c)^2}{2\sigma^2_c}\right) \]

- Multivariate Gaussian distribution

\[ p_f(f|c) = \frac{1}{(2\pi)^{N/2}|\Sigma_c|^{1/2}} \exp\left(-\frac{1}{2}(f - \mu_c)^T \Sigma_c^{-1}(f - \mu_c)\right) \]
Gaussian classifier steps

1) Estimate the parameters for each class

\[ \hat{\mu}_c = \frac{1}{N_c} \sum_{k=1}^{N_c} f_c \quad \hat{\Sigma}_c = \frac{1}{N_c} \sum_{k=1}^{N_c} (f_c - \mu_c)(f_c - \mu_c)^T \]

where \( N_C \) is the number of data values in class \( C \).

\[ p(C') = \frac{1}{K} \quad \text{or} \quad \frac{N_c}{N_t} \]

where \( K \) is the total number of classes, and \( N_t \) is total data values.

2) Assign data to class with largest posterior probability

\[ p(C|f) = \frac{p(f|C)p(C)}{\sum_C p(f|C)p(C)} \]
Unsupervised classifiers

• Classifiers that do not require training data

• Typically iterative, in order to train themselves

• Also called *clustering algorithms*

• Common unsupervised classifiers include K-means, fuzzy C-means, and Gaussian clustering via the EM algorithm

• They require an initialization

• Nonparametric unsupervised classifiers are difficult to derive
K-means clustering algorithm

- Also known as the ISODATA algorithm
- \( K \) refers to the number of classes, which is assumed to be known
- Formulated as a minimization of a cost function. Find the centroids \( \mu_k \) and class labels for each \( j \) to minimize

\[
J_{KM} = \sum_{k=1}^{K} \sum_{j \in \text{Class } k} \left\| f_j - \mu_k \right\|^2
\]

- How do you minimize a cost function?
K-means clustering steps

1) Provide initial means for each class

2) Classify each pixel/voxel in the image such that

\[
\text{Class of pixel } j = \arg \min_k \| f_j - \mu_k \|^2
\]

3) Recompute means

\[
\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j
\]

4) Iterate 2 & 3 until small number of pixels change classes or cost function no longer changes
K-means Example

Class of pixel $j = \arg \min_k \| f_j - \mu_k \|^2$

$$\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j$$
K-means Example

Class of pixel $j = \arg \min_k \| f_j - \mu_k \|^2$

$$
\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j
$$
K-means Example

Class of pixel j = \arg \min_k \| f_j - \mu_k \|^2

\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j
K-means Example

Class of pixel $j = \arg \min_k \| f_j - \mu_k \|^2$

$$\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j$$
K-means Example

Class of pixel $j = \arg\min_k \| f_j - \mu_k \|^2$

$$\mu_k = \frac{1}{N_k} \sum_{j \in \text{Class } k} f_j$$
O Brain has already been isolated (segmented!)
O Assume there are three classes remaining.
O Essentially, classify into three intensity groups

Original

CSF Indicator

GM Indicator

WM Indicator
Why “Fuzzy”?

- K-means model says that “non-pure” tissue intensities arise from noise and only noise:

\[ y_j = g_j \sum_{k=1}^{K} z_{jk} \nu_{jk} + \eta_j \]

- But in reality, intermediate values also arise from partial voluming.

There is a substantial number of voxels with “intermediate” intensity values.
Membership Functions

- **Indicator functions** $z_{jk}$ lead to hard classification
- **Membership functions** $u_{jk}$ lead to soft or fuzzy classification

$$u_{jk} \in [0, 1] \quad \text{and} \quad \sum_{k=1}^{K} u_{jk} = 1$$
Fuzzy K-means

• Generalization of K-means that allows for soft segmentations

• Interpretation based on fuzzy set theory

• Computes *membership functions*, like characteristic functions in traditional set theory but allow for partial membership within a set

• Membership functions characterize vagueness in data, probability characterizes frequency
Properties of membership values

• For membership $u_{jk}$ of pixel $j$, class $k$, the following properties must be satisfied:

\[ 0 \leq u_{jk} \leq 1 \quad \forall j \in \Omega, k = 1, \ldots, K \]

\[ \sum_{k=1}^{K} u_{jk} = 1, \quad \forall j \]

• Membership values generalize $z_{jk}$

• Posterior probabilities can be used exactly like membership functions
Fuzzy K-means algorithm

- Formulated as the minimization of the following cost function

\[ J_{FKM} = \sum_{k=1}^{K} \sum_{j \in \Omega} u_{jk}^q \| f_j - \mu_k \|^2 \]

- For \( q=1 \), equivalent to standard K-means

- For \( q>1 \), memberships become more fuzzy
Fuzzy K-means cost function

• Minimize with respect to memberships $u$ and means $\mu_k$

$$J_{FKM} = \sum_{k=1}^{K} \sum_{j \in \Omega} u_{jk}^2 \| f_j - \mu_k \|^2$$

• Subject to the constraints

$$0 \leq u_{jk} \leq 1, \quad \forall j \in \Omega, \ k = 1, \ldots, K$$

$$\sum_{k=1}^{K} u_{jk} = 1, \quad \forall j$$
Fuzzy K-means results

Image slice

Hard classification

Cerebrospinal Fluid Membership

Gray matter Membership

White Matter Membership
Deformable Models / Active Contours / Snakes

- Contours that deform within digital images
  - 2D: contour = curve
  - 3D: contour = surface
- Two broad classes
  - parametric
  - geometric or implicit

- Competing “forces”:
  - the contour is driven by “forces” derived from the image
  - its shape is controlled by “forces” that are internal to the snake

- Alternative formulations:
  - energy minimizing principle
  - direct dynamical equations
# Deformable Surfaces

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

Credit: Russell H. Taylor
Deformable Surfaces

Credit: Russell H. Taylor
Deformable Surfaces

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

Credit: Russell H. Taylor
Deformable Surfaces

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

Credit: Russell H. Taylor
Traditional Active Contour

- Initialize a curve $X(s)$ around or near the object boundary
- Find $X(s)$ that minimizes:

$$E = \int_0^1 \left[ \frac{1}{2} \left\{ \alpha |X'(s)|^2 + \beta |X''(s)|^2 \right\} + E_{\text{ext}} \{X(s)\} \right] ds$$

- Where $\alpha = 0.001$, $\beta = 0.09$ and

$$E_{\text{ext}}(x, y) = -\|\nabla f(x, y)\|^2$$

- How to find $X(s)$?
Dynamic Equation From E-L Equation

- Euler-Lagrange equation
  \[ \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{X}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{X}}{\partial s^2} \right) - \nabla P(\mathbf{X}) = 0 \]

- Make \( \mathbf{X} \) dynamic: \( X(s) \rightarrow X(s, t) \)

  \[ \mathbf{X}(s, t) = [X(s, t), Y(s, t)] \]
  where \( s \in [0, 1] \)

- Now set "in motion" – gradient descent
  \[ \gamma \frac{\partial \mathbf{X}}{\partial t} = \frac{\partial}{\partial s} \left( \alpha \frac{\partial \mathbf{X}}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left( \beta \frac{\partial^2 \mathbf{X}}{\partial s^2} \right) - \nabla P(\mathbf{X}) \]

- General dynamical equation for snake:
  \[ \gamma \mathbf{X}_t = \mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}} \]
Basic External Forces

\[ \gamma X_t = F_{\text{int}} + F_{\text{ext}} \]

- Edge “potential”

\[
P(x) = \frac{1}{1 + |\nabla(G_\sigma(x) \ast I(x))|}
\]

- \(I(x)\) is the image and \(G_\sigma(x)\) is a Gaussian convolution kernel

- Forces derived from edge potential

\[ F_{\text{ext}}(x) = -\nabla P(x) \]

- Add pressure forces

\[ F_{\text{ext}}(x) = -\nabla P(x) + w_{\text{pres}}(s, t)N \]
Basic External Forces

- Edge "potential"

\[ P(x) = \frac{1}{1 + |\nabla(G_\sigma(x) \ast I(x))|} \]

- \( I(x) \) is the image and \( G_\sigma(x) \) is a Gaussian convolution kernel
- Forces derived from edge potential
  \[ F_{\text{ext}}(x) = -\nabla P(x) \]
- Add pressure forces
  \[ F_{\text{ext}}(x) = -\nabla P(x) + w_{\text{pres}}(s) N \]

\[ \gamma X_t = F_{\text{int}} + F_{\text{ext}} \]
Deformable Surfaces

Figure 4.4 Evolution of the 3D surface “folding” on a 3D MRI image of a head. The initial surface is a plane on the border of the image.

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

Credit: Russell H. Taylor
Deformable Surfaces

Credit: Prince & Davatzikos
Example: Bone Modeling from CT

Contours → CT Slices → Tetrahedral Mesh → Density Function $f_n$ → Tetrahedral Mesh Simplification → Multiple Resolution Model → Credit: Yao and Taylor
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

Needle graph of Image force

Bone Contours

Credit: Yao and Taylor
Bone Contour Extraction
Closer-up view

Needle graph of Image force
Bone Contours

Credit: Yao and Taylor
3D Deformable Surface Model

Commonly done with triangle mesh

- Added complexity, time, especially to avoid self-intersection
Critique of Parametric Models

• Advantages:
  – explicit equations, direct implementation
  – automatic topology control

• Disadvantages:
  – costly to prevent overlaps
  – requires reparameterization to space out triangles
**Basic Idea of Geometric Active Contours**

- \( X(s, t) \): The parametric curve

- \( \phi(x, t) \): A level set function

- The zero level set: \( \{ x \mid \phi(x, t) = 0 \} \)

- **The level set function is usually a signed distance function**

- **Convention:**
  - positive on outside
  - negative on inside
GDM: Geometric Deformable Model

• Conventional level set function $\phi(x,t)$
  – signed distance function
• Change the values of $\phi \Rightarrow$ move the contour
Parametric to Geometric

Contour Deformation:

\[
\frac{\partial \mathbf{C}(p, t)}{\partial t} = \mathbf{F}
\]

\[
\Phi(\mathbf{C}(p, t), t) = 0
\]

\[
\frac{\partial \Phi}{\partial t} + \nabla \Phi \cdot \frac{\partial \mathbf{C}}{\partial t} = 0
\]

Define:

\[
F = \mathbf{F} \cdot \frac{\nabla \Phi}{\|\nabla \Phi\|} = \frac{\partial \mathbf{C}(p, t)}{\partial t} \cdot \frac{\nabla \Phi}{\|\nabla \Phi\|}
\]

Rearrange:

\[
\frac{\partial \Phi}{\partial t} + F \|\nabla \Phi\| = 0
\]

[Osher & Sethian 1988]
Philosophy of GDMs

• Curve is not parameterized until the end of evolution
  – tangential forces are meaningless
  – forces must be derived from “spatial position” and “time” because location on the curve is meaningless
  – Final contour is an “isocurve” (2D) or “isosurface” (3D)
  – It has a “Eulerian” rather than “Lagrangian” framework

• Speed function incorporates internal and external forces
  – Design of geometric model is accomplished by selection of F(x), the speed function
  – curvature terms takes the place of internal forces

• “Action” is near the zero level set
  – “narrowband” methods are computationally more efficient
General GDM

• A very useful model encompassing common forces is

\[ \gamma \phi_t = [\alpha \kappa + \beta \kappa^3 - \rho] |\nabla \phi| + w_R R |\nabla \phi| - F_{ext} \cdot \nabla \phi \]

• So-called “curvature” forces are actually related to the tangential tension \( \Rightarrow \alpha \)

• Bending forces require computation of \( \kappa^3 \)
  – rarely used

• Advection forces arise from “force” vectors applied in the normal direction

• Region “forces” arise from prior classification \( T(x) \)

\[ R(x) = \begin{cases} 
  +1 & T(x) = T_i \\
  -1 & T(x) \neq T_i 
\end{cases} \]

• A region force drives the contour outward when inside and inward when outside

\[ F_R = w_R R(x) N \]
Ventricle Segmentation
Cortical Surface Segmentation
Critique of Geometric Deformable Models

• Advantages:
  – Produce closed, non-self-intersecting contours
  – Independent of contour parameterization
  – Easy to implement: numerical solution of PDEs on regular computational grid
  – Stable computations

• Disadvantages:
  – topologically flexible
  – some numerical difficulties with narrowband and level set function reinitialization
Topology Preserving Geometric Deformable Model (TGDM)

- Evolve level set function according to GDM PDE
- If level set function is going to change sign, check whether the point is a simple point
  - If simple, permit the sign-change
  - If not simple, prohibit the sign-change
  - (replace the grid value by epsilon with same sign)
  - (Roughly, this step adds 7% computation time.)
- Extract the final contour using a connectivity consistent isocontour algorithm
Nested Deformable Surfaces

Initial WM Isosurface

Inner Surface

Central Surface

Pial Surface

TGDM-1

TGDM-2

TGDM-3
TGDM for Inner Surface

Initial WM Isosurface

Evolving GM/WM Interface

\[ \Phi_t = (\omega_1 R(\bar{x}) + \omega_2 \kappa(\bar{x})) \| \nabla \Phi \| \]

[Han et al., NeuroImage, 2004]
TGDM for Central Surface

Initialize with GM/WM surface

\[ \Phi_t = (\omega_1 R(\bar{x}) + \omega_2 \kappa(\bar{x})) \| \nabla \Phi \| + \omega_3 F_{GVF}(\bar{x}) \cdot \nabla \Phi \]

Evolving toward Central Surface
TGDM for Outer Surface

\[ \Phi_t = (\omega_1 R(\bar{x}) + \omega_2 \kappa(\bar{x})) || \nabla \Phi || + \omega_3 F_{GVF}(\bar{x}) \cdot \nabla \Phi \]

Start from Central Surface

Evolving toward Outer Surface
3D Digital Connectivity

- In 3D there are three connectivitys: 6, 18, and 26
- Four consistent connectivity pairs:
  \((\text{foreground,background}) \rightarrow (6,18), (6,26), (18,6), (26,6)\)
Topology Preservation Principle

- Preserving topology is equivalent to maintaining the topology of the digital object.
- The digital object can only change topology when the level set function changes sign at a grid point.
- To prevent the digital object from changing topology, the level set function should only be allowed to change sign at simple points.

[Han et al., PAMI, 2003]
Simple Point

• **Definition:** a point is simple if adding or removing the point from a binary object will not change the digital object’s topology.

• **Determination:** can be characterized locally by the configuration of its neighborhood (8- in 2D, 26- in 3D) [Bertrand & Malandain 1994]

![Simple vs. Non-Simple Points](image-url)
$x$ is a Simple Point

\[ \Phi(x) > 0 \iff \Phi(x) < 0 \]

(Connectivity happens to be irrelevant in this case)
x is Not a Simple Point (if n=4)

\[ \Phi(x) < 0 \quad \leftrightarrow \quad \Phi(x) > 0 \]

Digital connectivity assumption is crucial in this case
Topology Preserving Geometric Deformable Model (TGDM)

- Evolve level set function according to GDM PDE
- If level set function is going to change sign, check whether the point is a simple point
  - If simple, permit the sign-change
  - If not simple, prohibit the sign-change
  - (replace the grid value by epsilon with same sign)
  - (Roughly, this step adds 7% computation time.)
- Extract the final contour using a *connectivity consistent isocontour algorithm*
Marching Cubes Isosurface Algorithm

• How to “tile/triangulate” the zero level set?
• Consider values on corners of voxel (cube)
• Label as
  – above isovalue
  – below isovalue
• Determine the position of a triangular mesh surface passing through the voxel
  – Linear interpolation
Connectivity Errors

Most isosurface codes use rules that lead to connectivity errors

- Multiple meshes
  - typically solved by selecting the largest mesh
- Touching vertices, edges, and faces
  - typically solved isovalue choice
- Ambiguous faces and cubes
  - solved by use of a specially coded connectivity consistent MC algorithm
Ambiguous Faces

Two possible tilings:
Ambiguous Cubes

Two possible tilings:
Connectivity Consistent MC

(18,6) \rightarrow choose b, f \quad (6,18) \rightarrow choose c, f

(26,6) \rightarrow choose b, e \quad (6,26) \rightarrow choose c, f
Partial Inflation
Next Lecture on Segmentation: More advanced and modern approaches

- Graph cuts and Markov random fields
- Random walker algorithm
- Segmentation by registration and multi-atlas segmentation
- Deep networks