Segmentation II
Segmentation

Separate image into coherent “regions”

Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench
Interactive Segmentation

Today: separate image into “foreground” and “background” with user assistance

Input

Foreground Segmentation
Applications

Why would you want to do this?

Input  ➔  Foreground Segmentation
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking
etc.
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking

Input
Foreground Segmentation
Composition
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking

Mortensen and Barret, 1995
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking

Wang, 2012
Applications

Image composition
Image processing
**Image analysis**
Object labeling
Object tracking

Traced Fracture Edges

Crocus Gatherer and Potnia, Akrotiri
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking

Spinal Vertebrae

Mortensen and Barret, 1995
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking

Corpus Callosum

[Davatzikos and Prince]
Applications

Image composition
Image processing
Image analysis
Object labeling
Object tracking
User Interface

What input should the user provide?

Input

Foreground Segmentation
User Interface

Magnetic lasso
Approximate contour
Surrounding contour
Labeling strokes
etc.
User Interface

Magnetic lasso
Approximate contour
Surrounding contour
Labeling strokes

http://www.devarticles.com/c/a/Photoshop/Using-Adobe-Photoshop-CS-Part-1/5/
User Interface

Magnetic lasso
Approximate contour
Surrounding contour
Labeling strokes

User-Drawn Contour
User Interface

Magnetic lasso
Approximate contour
Surrounding contour
Labeling strokes
User Interface

Magnetic lasso
Approximate contour
Surrounding contour
Labeling strokes
Different Algorithms for Different Interfaces

Labeling strokes
Surrounding contour
Magnetic lasso
Approximate contour
Outline for Today

Labeling strokes
Surrounding contour
Magnetic lasso
Approximate contour
Labeling strokes:

- User sketches out a few strokes on foreground and background, and asks computer to find segmentation.

What algorithm?
Graph Cuts

Node $x_i$: pixel label

Edge: connected pixels (e.g., neighbors)

User-specified Background

User-specified Foreground
Graph Cuts

Unary Potential
0: background
1: foreground

Edge Potential
\[ k_1 + k_2 \exp \left( \frac{-|c(x) - c(y)|^2}{2\sigma^2} \right) \]

User-specified Background
User-specified Foreground
Simple Graph Cuts

Source (Label 0)

Cut

Cost to assign to 0

Cost to split nodes

Cost to assign to 1

Sink (Label 1)
Graph Cuts

Observation: user input provides examples from which a discriminative model can be learned
Graph Cuts

Can interpret connection to source/sink as a probability that a pixel has a particular label

\[
- \log \left( \frac{P(c(x); \theta_{\text{foreground}})}{P(c(x); \theta_{\text{background}})} \right)
\]

Unary Potential

Source (Label 0)

Cost to assign to 0

Sink (Label 1)

Cost to assign to 1
Graph Cuts

Can interpret connections between neighbor pixels as probabilities that they share a label (edge potentials)

\[ k_1 + k_2 \exp \left\{ \frac{-\|c(x) - c(y)\|^2}{2\sigma^2} \right\} \]

Cost to split nodes
Graph Cuts

Can combine unary and edge probabilities (potentials) into a joint probability (energy function)

\[ \text{Energy}(x; \theta, \text{data}) = \sum_i \psi_1(x_i; \theta, \text{data}) \sum_{i,j \in \text{edge}} \psi_2(x_i, x_j; \theta, \text{data}) \]

Markov Random Field

Source (Label 0)

Cost to assign to 0

Cost to split nodes

Cost to assign to 1

Sink (Label 1)
Graph Cuts

Cut is the labeling that has maximum posterior probability (minimum energy)

\[ \text{Energy}(x; \theta, \text{data}) = \sum \psi_1(x_i; \theta, \text{data}) + \sum_{i,j \in \text{edge}} \psi_2(x_i, x_j; \theta, \text{data}) \]
Graph Cuts

Application: segmenting noisy images

\( \phi(x_i, y_i) \)

\( \psi(x_i, x_j) \)

y (image pixels)

x (region labels)

real image

label image
Graph Cuts

Application: medical imaging
Different Algorithms for Different Interfaces

Labeling strokes

**Surrounding contour**

Magnetic lasso

Approximate contour
GrabCut

Application: user draws contour around foreground
  • Learn unary and edge potentials iteratively
1. Define graph
   − usually 4-connected or 8-connected

2. Define unary potentials
   − Color histogram or mixture of Gaussians for background and foreground

   \[
   \text{unary \_ potential}(x) = -\log \left( \frac{P(c(x); \theta_{\text{foreground}})}{P(c(x); \theta_{\text{background}})} \right)
   \]

3. Define pairwise potentials

   \[
   \text{edge \_ potential}(x, y) = k_1 + k_2 \exp \left( -\frac{\|c(x) - c(y)\|^2}{2\sigma^2} \right)
   \]

4. Apply graph cuts

5. Return to 2, using current labels to compute foreground, background models
Step 2: Define unary potentials

\[ \text{unary\_potential}(x) = -\log \left( \frac{P(c(x); \theta_{\text{foreground}})}{P(c(x); \theta_{\text{background}})} \right) \]

Gaussian Mixture Model
(typically 5-8 components)
GrabCut

Step 3: Define pairwise potentials

\[ edge\_potential(x, y) = k_1 + k_2 \exp \left\{ \frac{-\|c(x) - c(y)\|^2}{2\sigma^2} \right\} \]
Step 4: Graph cut

Source: Rother
GrabCut

Step 5: Iterate

1st Iteration

Last Iteration

Gaussian Mixture Models
GrabCut

Relatively easy examples:
GrabCut

Difficult examples

Camouflage & Low Contrast

Initial Rectangle

Fine structure

Initial Result

Harder Case

GrabCut – Interactive Foreground Extraction
Outline for Today

Graph cuts
GrabCuts
Magnetic lasso
Deformable contours
Magnetic Lasso

User traces segment outline, and computer “snaps” to closest edge
Magnetic Lasso

User traces segment outline, and computer “snaps” to closest edge
Magnetic Lasso

User traces segment outline, and computer “snaps” to closest edge
Magnetic Lasso

How does it work?
Magnetic Lasso

Move user-traced path along gradient vector field
Application: Fracture Pattern Analysis

Goal: learn statistics of fragment arrangements from previously reconstructed wall paintings

- Develop model of fracture formation
- Guide matching algorithms
Application: Fracture Pattern Analysis

Approach

• Trace fragment contours in image of reconstructed wall painting

• Describe relationships between adjacent fragments statistically
Application: Fracture Pattern Analysis

Fragment tracing
Application: Fracture Pattern Analysis

Fragment tracing

Crocus Gatherer and Potnia
Application: Fracture Pattern Analysis

Fragment tracing
Application: Fracture Pattern Analysis

Statistical analysis of fragment contours

- Number of Adjacent Fragments
- Normalized Area
- Angle (degrees)
- Log frequency
Application: Fracture Pattern Analysis

Resulting Hypothesis

• Sequential, hierarchical fracture process
• Fragments broke recursively into two nearly equal size pieces, along nearly orthogonal cracks
Outline for Today

Graph cuts
GrabCuts
Magnetic lasso
Approximate contours
A deformable contour (snake) is defined by:

- A set of \( n \) points,
- An internal energy term (tension, bending, plus optional shape prior)
- An external energy term (gradient-based)

To use to segment an object:

- Initialize in the vicinity of the object
- Modify the points to minimize the total energy
Deformable Contours

a.k.a. active contours, snakes

**Given:** initial contour (model) near desired object

Figure credit: Yuri Boykov

[Snakes: Active contour models, Kass, Witkin, & Terzopoulos, ICCV1987]
Deformable Contours

**Given:** initial contour (model) near desired object

**Goal:** evolve the contour to fit exact object boundary

**Main idea:** elastic band is iteratively adjusted so as to

- be near image positions with high gradients, **and**
- satisfy shape “preferences” or contour priors

Figure credit: Yuri Boykov
Deformable Contours: Intuition
Why Deformable Contours?

May be able to acquire initial contour without (much) user input
Why Deformable Contours?

Non-rigid, deformable objects can change their shape over time, e.g. lips, hands...

Figure from Kass et al. 1987
Why Deformable Contours?

Non-rigid, deformable objects can change their shape over time, e.g. lips, hands…
Why Deformable Contours?

• Non-rigid, deformable objects can change their shape over time.

Figure credit: Julien Jomier
Deformable Contours

Algorithm:
- Initialize contour as a list of vertices
- Define a cost function ("energy" function) that says how good a contour is.
- Adjust vertex positions to minimize the cost function.
Deformable Contours

Energy definition

Energy minimization
Deformable Contours

Energy definition:

\[ E_{\text{total}} = E_{\text{external}} + E_{\text{internal}} \]

**External** energy ("image" energy): encourages contour to align with structures in the image

**Internal** energy: encourage *prior* shape preferences: e.g., smoothness, elasticity, particular known shape.

A good fit between the current deformable contour and the target shape in the image will yield a **low** value for this cost function.
Deformable Contours

External energy: Intuition

- Measure how well the curve matches the image data
- “Attract” the curve toward image features
External energy: formulation

Magnitude of gradient

\[ G_x(I)^2 + G_y(I)^2 \]

- (Magnitude of gradient)

\[ - \left( G_x(I)^2 + G_y(I)^2 \right) \]
Deformable Contours

External energy at a point on the curve could be defined as:

\[ E_{\text{external}}(\mathbf{v}) = -\left( |G_x(\mathbf{v})|^2 + |G_y(\mathbf{v})|^2 \right) \]

External energy for the whole curve:

\[ E_{\text{external}} = -\sum_{i=0}^{n-1} |G_x(x_i, y_i)|^2 + |G_y(x_i, y_i)|^2 \]

- (Magnitude of gradient)

\[ -\left( G_x(I)^2 + G_y(I)^2 \right) \]
Deformable Contours

This simple external energy would not really affect contour unless object boundary is already very close.

\[ \nabla I \]

image gradients are large only directly on the boundary

Kristen Grauman
External energy can instead be a gradient vector field

- e.g., **distance transform** of the edge image

Value at \((x,y)\) tells how far that position is from the nearest edge point (or other binary image structure)

Kristen Grauman
Deformable Contours

External energy with gradient vector field

Gradient vector field
Deformable Contours

External energy with gradient vector field

With simple image gradient  With gradient vector field
Deformable Contours

Internal energy
Internal energy: intuition

• *A priori*, we want to favor …
  • Contours with *smooth* shapes
  • Contours similar to a *known shapes*
Deformable Contours

Internal energy: formulation

• Common internal energy term is the “bending energy”.

\[ E_{\text{internal}}(\nu(s)) = \alpha \left| \frac{d \nu}{d s} \right|^2 + \beta \left| \frac{d^2 \nu}{d^2 s} \right|^2 \]

Tension, Elasticity

Stiffness, Curvature
Deformable Contours

Internal energy: formulation

- Other internal energy terms might consider shape priors, etc.

Fig from Y. Boykov
Deformable Contours

Total energy: weighted sum of terms

\[ E_{\text{total}} = E_{\text{internal}} + \gamma E_{\text{external}} \]

\[ E_{\text{external}} = - \sum_{i=0}^{n-1} \left| G_x(x_i, y_i) \right|^2 + \left| G_y(x_i, y_i) \right|^2 \]

\[ E_{\text{internal}} = \sum_{i=0}^{n-1} \left[ \alpha (\bar{d} - \| \nu_{i+1} - \nu_i \|)^2 + \beta \| \nu_{i+1} - 2\nu_i + \nu_{i-1} \| \right]^2 \]
Deformable Contours

Energy minimization:

• Greedy (steepest decent)
• Dynamic programming
• Others

Xu and Prince
Deformable Contours

Application: Tracking:

• Use final contour/model extracted at frame $t$ as an initial solution for frame $t+1$

• Evolve initial contour to fit exact object boundary at frame $t+1$

• Repeat, initializing with most recent frame.

Tracking Heart Ventrices (multiple frames)
Deformable Contours

Limitations:

- **Choice of weights affects results**

  ![Examples of different alpha values](image)

  - **large** $\alpha$
  - **medium** $\alpha$
  - **small** $\alpha$

- **Cannot follow topological changes of objects**

  ![Examples of topological changes](image)
Deformable Contours

Pros:
- Useful to track and fit non-rigid shapes
- Contour remains connected and well shaped
- Flexibility in how energy function is defined, weighted.

Cons:
- Must have decent initialization near true boundary, may get stuck in local minimum
- Parameters of energy function must be set well based on prior information
- No topological changes to contour
Summary

Interactive segmentation algorithms

• Graph cuts
• GrabCuts
• Magnetic lasso
• Deformable contours

Applications

• Image editing
• Image analysis
• Object tracking
• etc.

Which algorithm is best depends on user and application