Image Segmentation II

DD2423 Image Analysis and Computer Vision

Mårten Björkman

Computational Vision and Active Perception
School of Computer Science and Communication

November 25, 2015

Segmentation techniques

- Figure-ground segmentation: divide image in foreground and background regions. Methods:
  - Thresholding
  - Level-set methods
  - Energy minimization with graphs
- Image segmentation: divide image in regions with pixels of similar qualities. Methods:
  - K-means clustering
  - Split-and-merge
  - Watershedding
  - Mean-shift
  - Normalized cuts

Snakes – active contour models

- Idea: Represent a shape by a curve

$$X(s) = (x(s),\ y(s));\ s \in [0,1]$$

- Closed contour (common in practice):

$$X(0) = X(1)$$

- Find the curve by minimizing some energy.

$$E = E_{internal} + E_{image}$$

Snakes – active contour models

- Internal energy (makes curve as short and straight as possible):

$$E_{internal} = \int \alpha(s)\ ||X_s(s)||^2 + \beta(s)\ ||X_{ss}(s)||^2\ ds$$

  $\alpha(s)$ and $\beta(s)$ can control shape for different points.

- Image energy (ex. maximize image gradient along curve):

$$E_{image} = E_{edge} = -\int ||\nabla I(X(s))||^2\ ds$$

- Add whatever energy is suitable for the task. Examples:

  E_{line}: fit curve to a line in the image

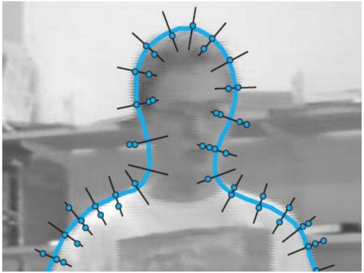
  E_{spring}: make curve pass some given anchor points

Snakes – active contour models

- Discretization: Divide contour into pieces

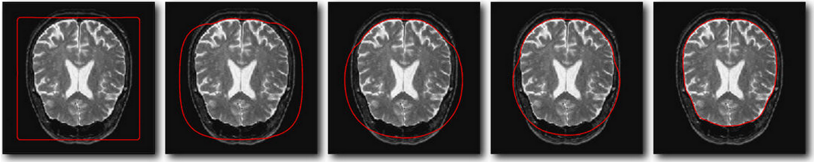
$$E = \sum_i \alpha_i ||X_i - X_{i+1}||^2 + \beta_i ||X_{i-1} - 2X_i + X_{i+1}||^2 - ||\nabla I(X_i)||^2$$

- Iteratively, move along normals to gradually decrease energy.



Snakes – active contour models

- Strengths:
  - Easy to add custom energies to suit problem
  - Very good results on some problems (eg. medical)
  - Can be extended to particular shapes (eg. bodies)
- Weaknesses:
  - Can easily get stuck on wrong edges
  - Hard to avoid self-intersection
  - Cannot change topology



Level set segmentation

- Write the curve a function of time t

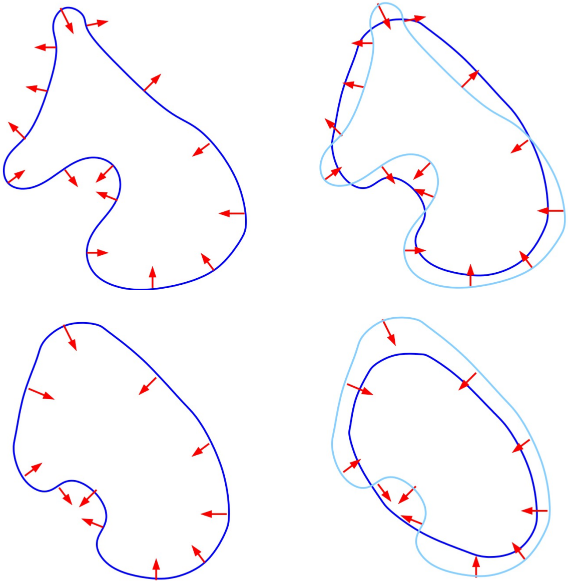
$$X(s,t) = (x(s,t),\ y(s,t))$$

- Let the curve move along normal N with speed given by the curvature κ.

$$\frac{\partial X}{\partial t} = \kappa N$$

- The effect of minimizing some internal energy.

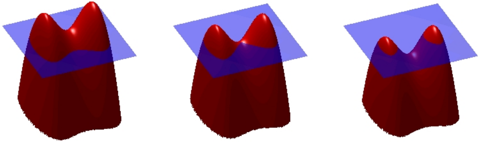
Level set segmentation



Level set segmentation

- Embed the curve $X(s,t)$ in a level set function ϕ:

$$\phi(X(s,t),t) = 0$$



- Idea: Update ϕ and let curve be given by level set $\phi = 0$.

Level set segmentation

- Apply the chain rule for derivation:

$$\frac{d\phi(X(s,t),t)}{dt} = \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial X}\frac{\partial X}{\partial t} = \frac{\partial\phi}{\partial t} + \nabla\phi\frac{\partial X}{\partial t},$$

where the gradient of the function is

$$\nabla\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}\right)^T$$

- The normal N can then be written as

$$N = \frac{\nabla\phi}{|\nabla\phi|}$$

Level set segmentation

- For the zero level set we have:

$$\frac{\partial \phi}{\partial t} + \nabla \phi \frac{\partial X}{\partial t} = 0$$

- With the curve moved as

$$\frac{\partial X}{\partial t} = \kappa N$$

  the level set function is updated as

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \kappa N = \kappa |\nabla \phi|$$

- Here the curvature κ is given by

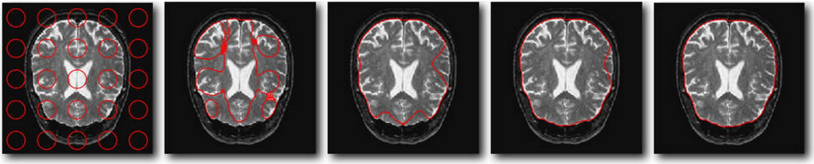
$$\kappa = \nabla N = \frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}$$

Level set segmentation

- Thus, you can get the same effect by updating a level-set function $\phi(x, y)$, as you would for an active contour $X(s)$. [Internal energy]
- To make it useful for image segmentation, we also need to $\phi(x, y)$ dependent on the image $I(x, y)$. [Image energy]

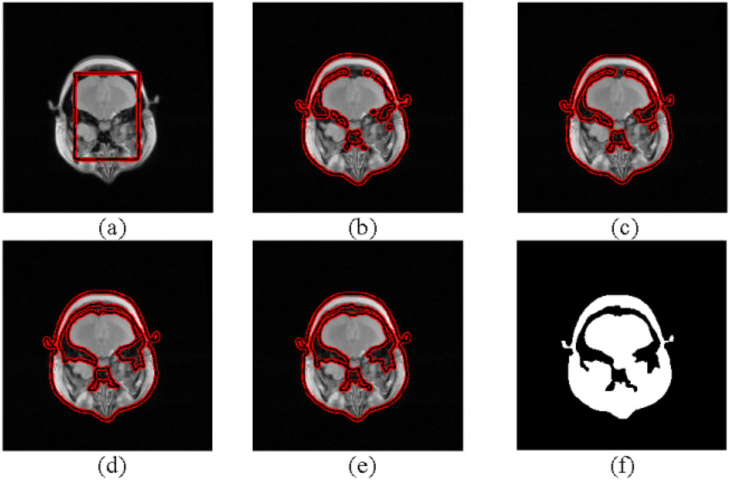
Level set segmentation

- Strengths:
  - Easy to add custom functions to suit problem
  - Can handle more complex shapes
  - Easily extended to high dimensions
  - Allows changes in topology (easier to initialize)
- Weaknesses:
  - Convergence speed can be slow
  - Can be hard to implement



Level-set based segmentation

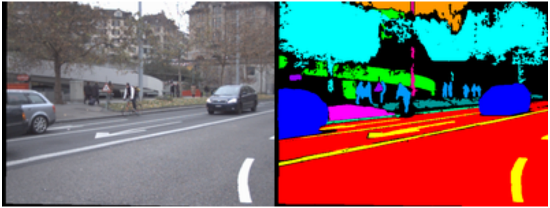
Very common in medical imaging.



From zero (a) to 200 iterations (e) with final segmentation (f).

Graph based segmentation using energy minimization

- Create a graph with one node per pixel and links between all neighbouring nodes (using e.g. 4-neighbours).
- Assume you have a set of models. Examples:
  - $L = \{foreground, background\}$
  - $L = \{sky, ground, vegitation, people, cars, houses\}$
- Each pixel x has a label $l_x \in L$ denoting which model it belongs to.
- Models are usually trained beforehand using a large database.
- A model can for example be represented by a color histogram.



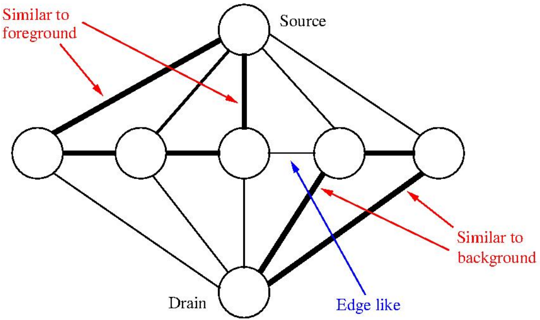
Graph based segmentation using energy minimization

- Idea: Set up the problem as a energy (cost) minimization problem.
- Find the combination of labels that minimizes the cost

$$E = \sum_x \Psi_x(l_x) + \sum_x \sum_{y \in N_x} \Psi_{x,y}(l_x, l_y)$$

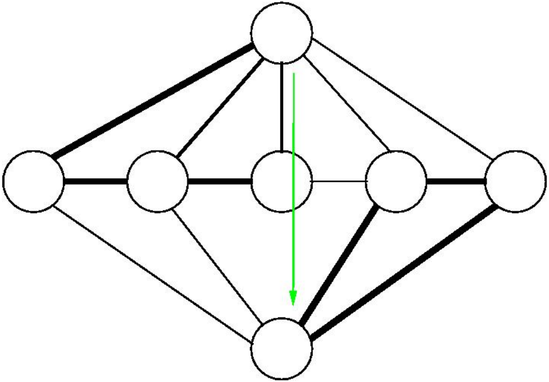
- Cost per node $\Psi_x(l_x)$
  - Low cost if pixel has a color similar to model l_x, high otherwise.
- Cost per link $\Psi_{x,y}(l_x, l_y)$
  - Normally $\Psi_{x,y}(l_x, l_y) = 0$, if $l_x = l_y$.
  - Low cost if pixels have different colors (edge)
  - High cost if pixels have similar colors (smooth surface)
- Result of energy minimization: assign pixels to most similar models, while aligning borders of segments to edges.

Graph cuts using 5 pixels



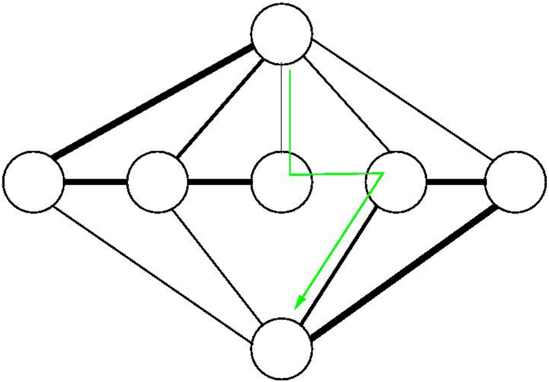
- Graph cuts can be used to minimize the energy E for $|L| = 2$.
- Add links to Source and Drain with costs given by $\Psi_x(l_x)$.
  Source = foreground, Drain = background.
- Goal: Find the lowest cost split of the graph into two pieces.

Graph cuts



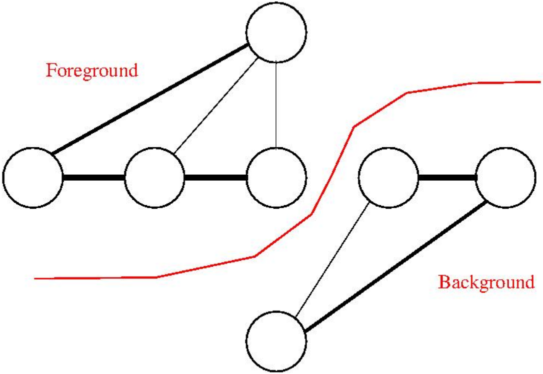
- Push flow from Source to Drain until links saturate.
- Imagine links are pipes and you push flow (water).
- Width of links illustrate how much more flow can be pushed.

Graph cuts



- Try all possible paths from Source to Drain.

Graph cuts

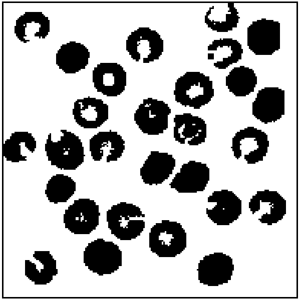


- In the end no more flow can be pushed from Source to Drain.
- Nodes have been divided into two segments.
- The saturated links correspond to the non-zero terms in E.

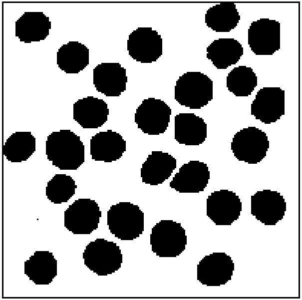
GrabCut: example of energy based segmentation

- Left: A couple of strokes are applied to create colour models of background and foreground.
- Right: Afterwards background can be changed to something else.

- Often there is a need to 'clean up' the segmentation.
  Some segments are simply to small and only due to noise.
  Others have holes, because colours look like background.
- Solution: Apply (non-linear) morphological operations.



A



B

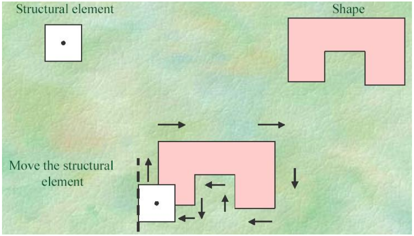
Morphology : *Britannica*

- Morphology (biology): the study of the size, shape, and structure of animals, plants, and microorganisms and the relationships of their internal parts.
- Morphology (linguistics): study of internal construction of words.
- Image processing: The basis of mathematical morphology is the description of image regions as sets.

A	the (usually binary) image
A^C	the complement of the image (inverse)
$A \cup B$	the union of images A and B
$A \cap B$	the intersection of images A and B
$A - B = A \cap B^C$	the difference between A and B (pixels in A not in B)
$\#A$	the cardinality of A (area of the object)

Mathematical morphology

- A morphological operator is defined by a structuring element (or kernel) of size $n \times n$ and a set operator.
- Kernel is shifted over the image and its elements are compared with the underlying pixels.
- If the two sets of elements match the condition defined by the set operator, the pixel underneath the center of the structuring element is set to a pre-defined value (0 or 1 for binary images).



Morphological operators

Dilation - grow image regions

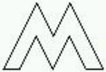
Erosion - shrink image regions

Opening - structured removal of image region boundary pixels

Morphological operators

Closing - structured filling in of image region boundary pixels

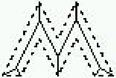
Hit and Miss Transform - image pattern matching and marking



Thinning - structured erosion using image pattern matching

Morphological operators

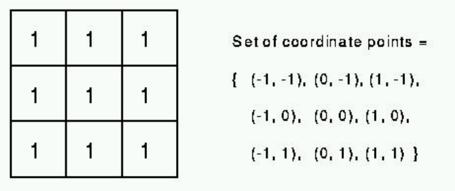
Thickening - structured dilation using image pattern matching



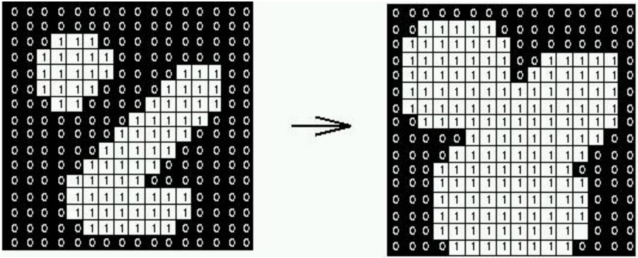
Skeletonization/Medial Axis Transform - finding skeletons of binary regions

Dilation (Minkowski addition)

- The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels.
- Areas of foreground pixels grow in size, while holes in regions become smaller.
- Let A and B denote sets in $\mathbb{R}^2$ with elements a and b.
  Then
  $$A \oplus B = \{c \in \mathbb{R}^2 : c = a + b\}$$
- Typically: A = binary image, B = mask (structuring element)
  Example: 3×3 structuring element



- If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value.
- If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

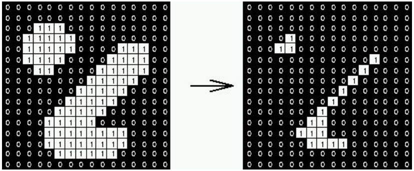


Erosion

- The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels.
- Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

$$A \ominus B = \{c \in \mathbb{R}^2 : c + b \in A, \forall b \in B\}$$

- If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is.
- If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.



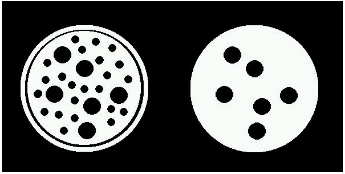
Operators

- Opening: an opening is defined as an erosion followed by a dilation using the same structuring element for both operations

$$A \circ B = (A \underbrace{\ominus}_{erosion} B) \underbrace{\oplus}_{dilation} B$$

- Closing:

$$A \bullet B = (A \oplus B) \ominus B$$



- Opening is the dual of closing: opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

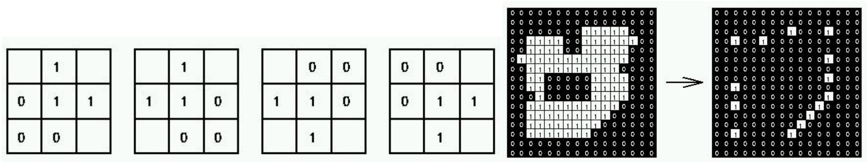
- Purpose: Extract objects of certain shape.
- Let J and K be structuring elements with $J \cap K = \emptyset$
- Hit-and-miss transform can then be defined as

$$A \otimes (J,K) = (A \ominus J) \cap (A^C \ominus K)$$

- Contains all points where

  \- J matches the object

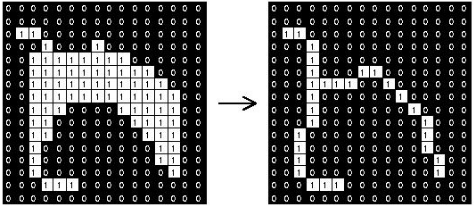
  \- K matches background
- Example: detect corners in a binary image.



Thinning

- The hit-and-miss transform has many applications in more complex morphological operations.
- It is used to construct the thinning and thickening operators.
- The thinning operation can be written in terms of a hit-and-miss transform

$$\text{thin}(A, B) = A - \text{hit-and-miss}(A, B)$$



Summary of good questions

- What is an active contour?
- How do level set methods represents segmentations?
- What are the difference and similarities between active contours and level set methods?
- What kinds of cost functions does an energy formulation for segmentation often include?
- What is the purpose of graph cuts for segmentation?
- How does a graph cut work?
- What does a morphlogical opening and closeing operation do?

Readings

- Gonzalez and Woods: Chapters 9.1-9.4
- Szeliski: Chapter 5