CS270: Digital Image Processing

Dr. Xavier LAGORCE

Email: xavier@shanghaitech.edu.cn

Office: SIST 1D-301B

ShanghaiTech University

Lecture 11
Image segmentation II: Watershed and active contours

Dr. Xavier LAGORCE
Email: xavier@shanghaitech.edu.cn
Office: 1D-301.B, SIST
ShanghaiTech University

上海科技大学 ShanghaiTech University

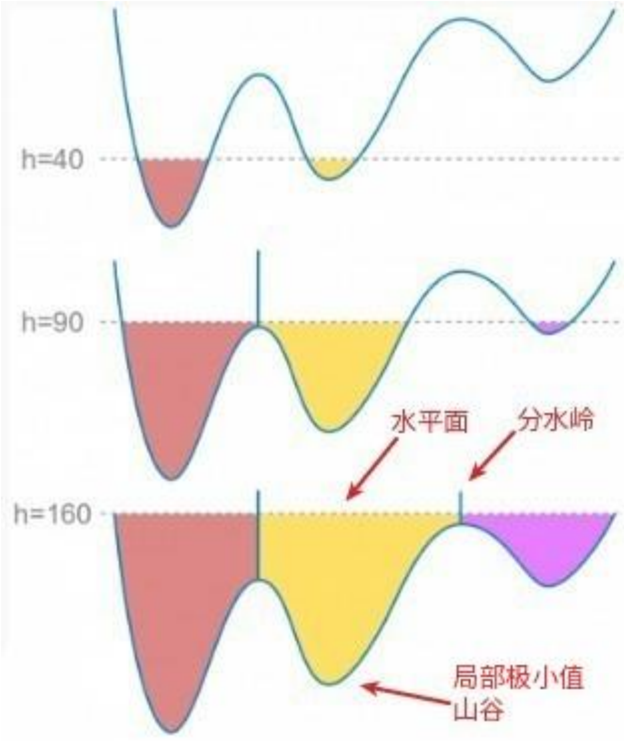
立志成才报国裕民

Previously

- Pixel-based segmentation:
  - each pixel is segmented based on gray-level values, no contextual information, only histogram.
- Edge-based segmentation:
  - Detects and links edge pixels to form contours.
- Region-based segmentation:
  - considers gray-levels from neighboring pixels by
    - including similar neighboring pixels (region growing),
    - split-and-merge,
    - super-pixel segmentation (k-means).

Watershed segmentation (分水岭)

- Look at the image as a 3D topographic surface, (x,y,intensity), with both valleys and mountains.
- Assume that there is a hole at each minimum, and that the surface is immersed into a lake.
- The water will enter through the holes at the minima and flood the surface.
- To avoid two different basins to merge, a dam is built.
- Final step: the only thing visible would be the dams.
- The connected dam boundaries correspond to the watershed lines.



Watershed segmentation

- Can be used on images derived from:
  - Intensity image
  - Edge enhanced image
  - Distance transformed image
  - Thresholded image: from each foreground pixel, compute the distance to a background pixel.
  - Gradient of the image
- Most common: **gradient image**

Watershed algorithm

- Let $g(x,y)$ be the input image (often a gradient image).
- Let $M_1, \cdots, M_R$ be the coordinates of the regional minima.
- Let $C(M_i)$ be a set consisting of the coordinates of all points belonging to the catchment basin associated with the regional minimum M_i.
- Let $T[n]$ be the set of coordinates (s,t) where $g(s,t) < n$

$$T[n] = \{(s,t) | g(s,t) < n\}$$

This is the set of coordinates lying below the plane $g(x,y) = n$

This is the set of candidate pixels for inclusion into the catchment basin, but we must take care that the pixels do not belong to different catchment basins.

Watershed algorithm

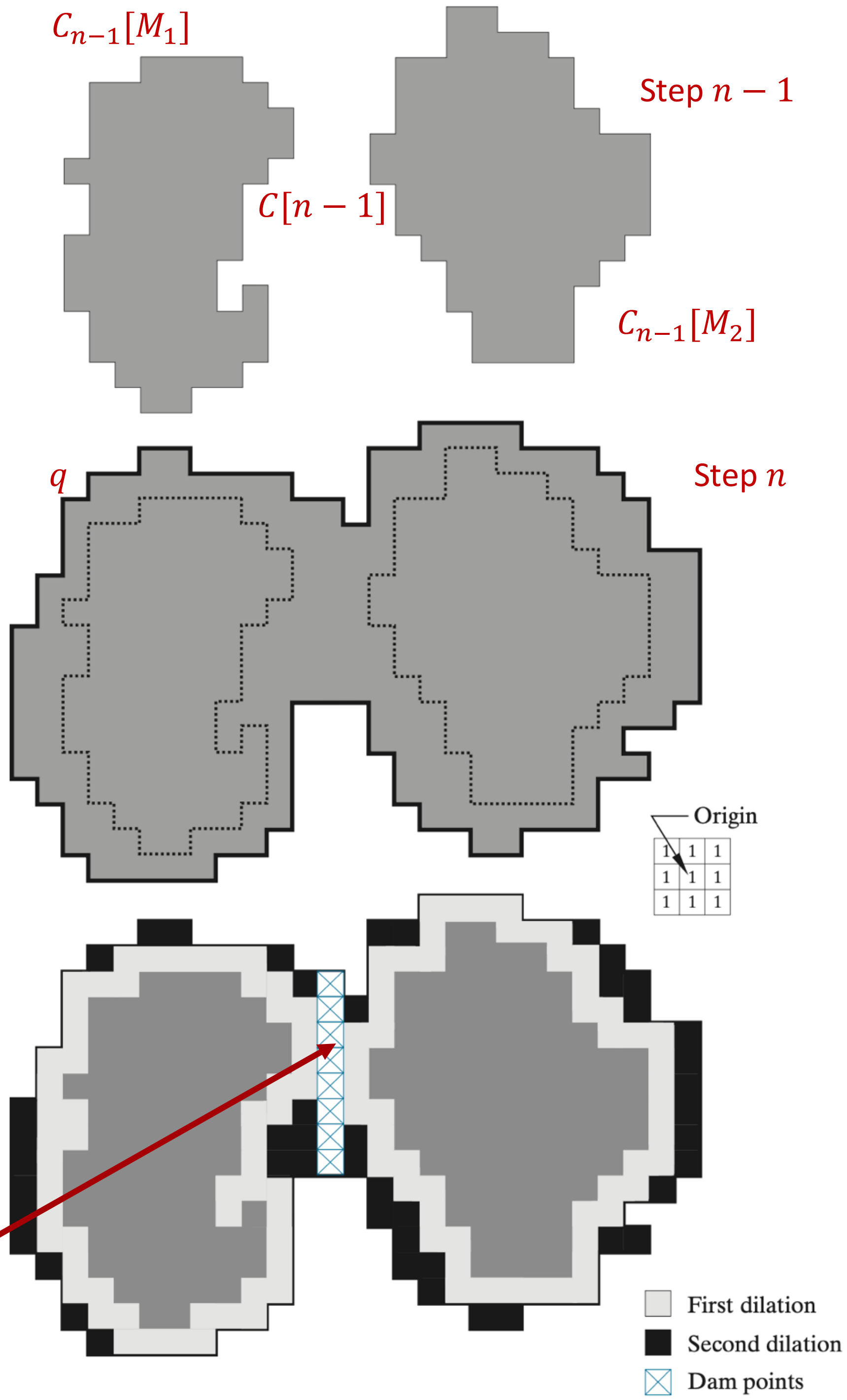
- The topography will be flooded with integer flood increments from $n = \min - 1$ to $n = \max + 1$.
- Let $C_n(M_i)$ be the set of coordinates of points in the catchment basin associated with M_i, flooded at stage n.
- This must be a connected component and can be expressed as $C_n(M_i) = C(M_i) \cap T[n]$ (only the portion of $T[n]$ associated with basin M_i)
- Let $T[n]$ be the set of flooded water, let $C[n]$ be union of all flooded catchments at stage n:

$$C[n] = \cup_{i=1}^{R} C_n(M_i) \text{ and } C[max + 1] = \cup_{i=1}^{R} C(M_i)$$

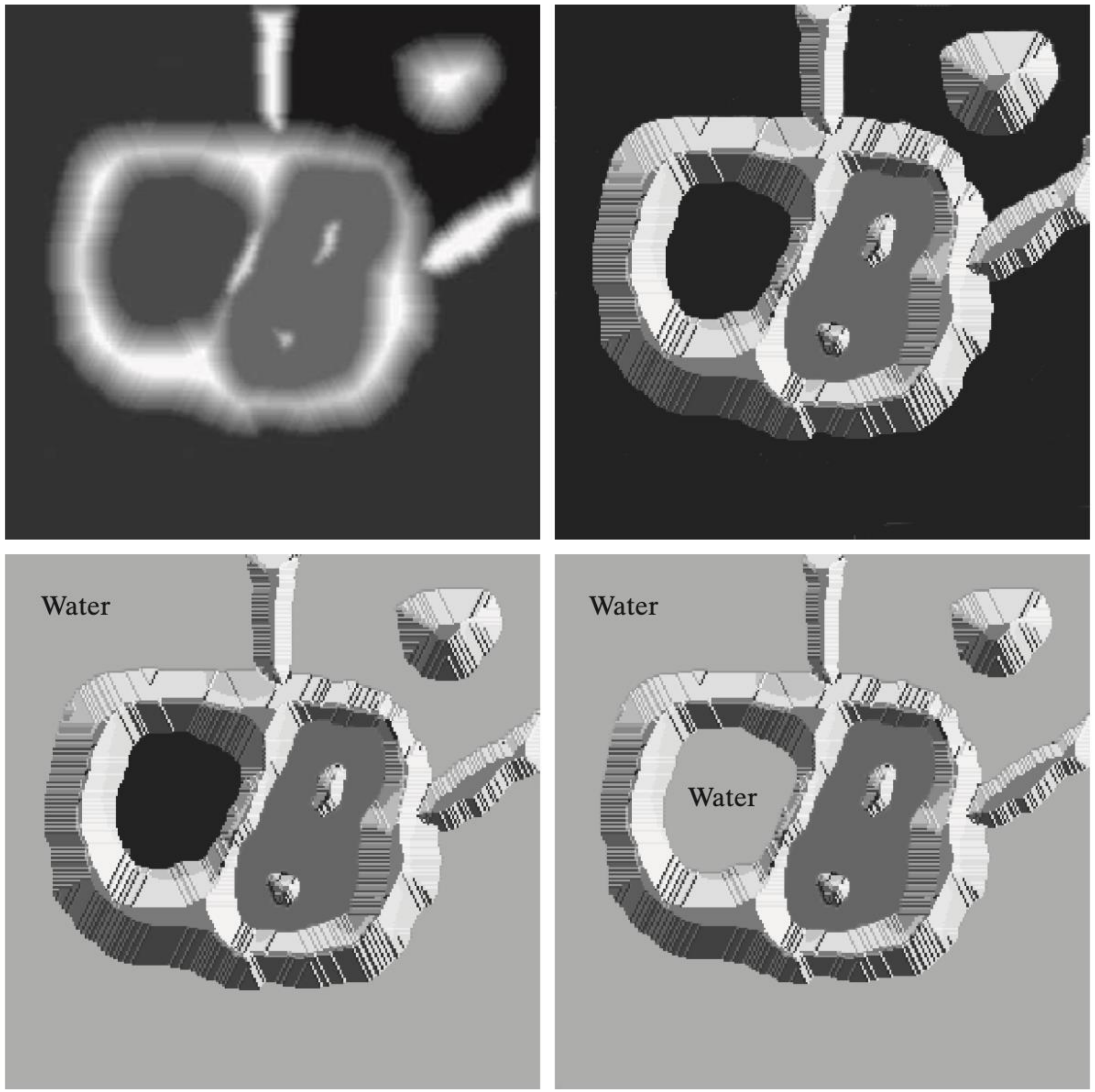
Dam construction

- Stage $n-1$: two basins forming separate connected components.
- To consider pixels for inclusion in basin k in the next step (after flooding), they must be part of $T[n]$, and also be part of the connected component q of $T[n]$ that $C_{n-1}[k]$ is included in.
- Use morphological dilation iteratively: Dam building
  - Dilation of $C[n-1]$ is constrained to q.
  - The dilation can not be performed on pixels that would cause two basins to be merged (form a single connected component)

Set to values larger than max image value (i.e. > 255 for a 8 bit image)



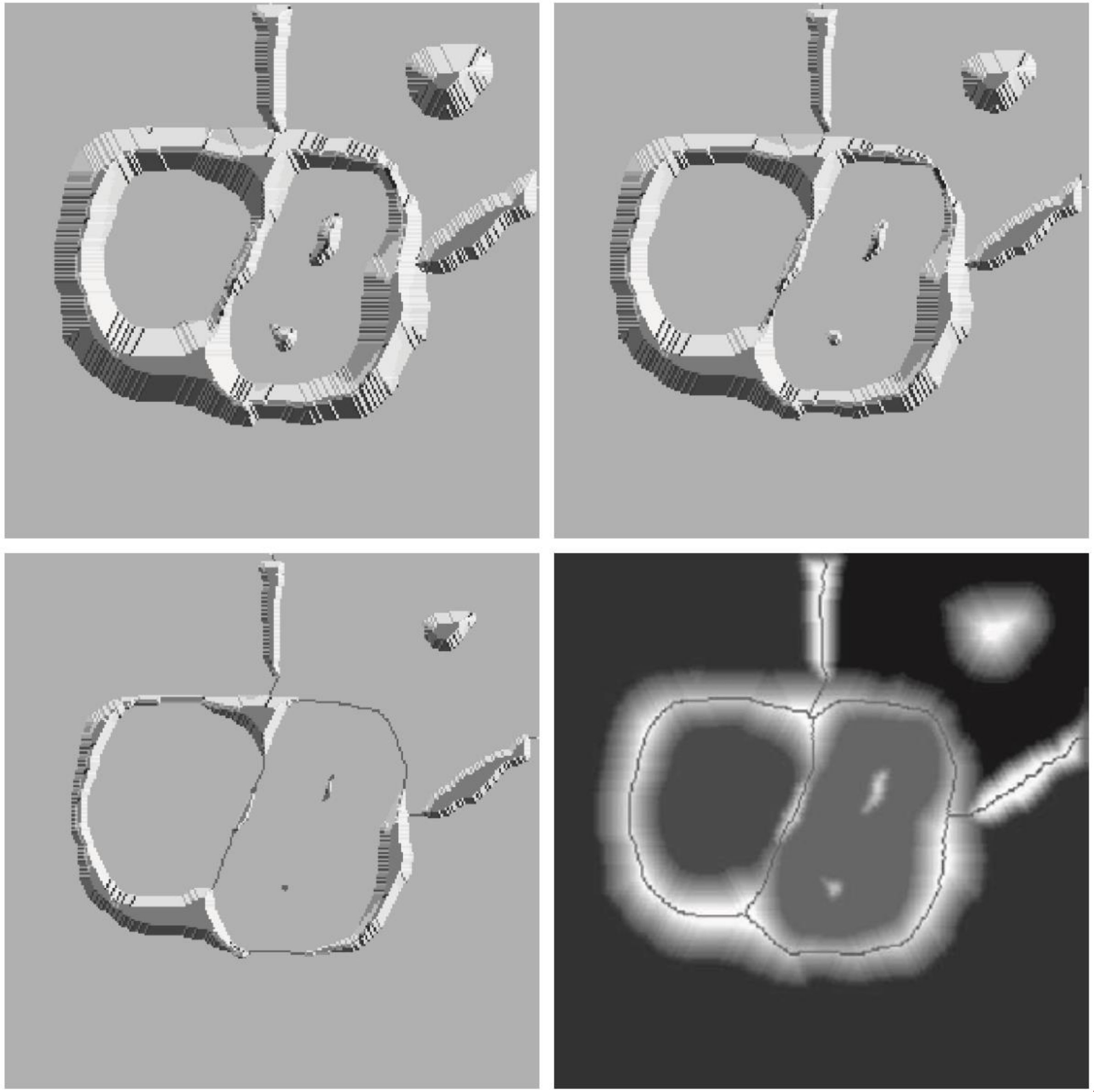
Watershed procedure (1)



a c
b d

FIGURE 10.57 (a) Original image. (b) Topographic view. Only the background is *black*. The basin on the left is slightly lighter than black. (c) and (d) Two stages of flooding. All constant dark values of gray are intensities in the original image. Only constant *light gray* represents "water." (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.) (*Continued on next page.*)

Watershed procedure (2)



e f
g h

FIGURE 10.57 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines superimposed on the original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Note that "dams" are always connected components (i.e. the boundaries resulting from the watershed segmentation are always connected components !)

Watershed Example

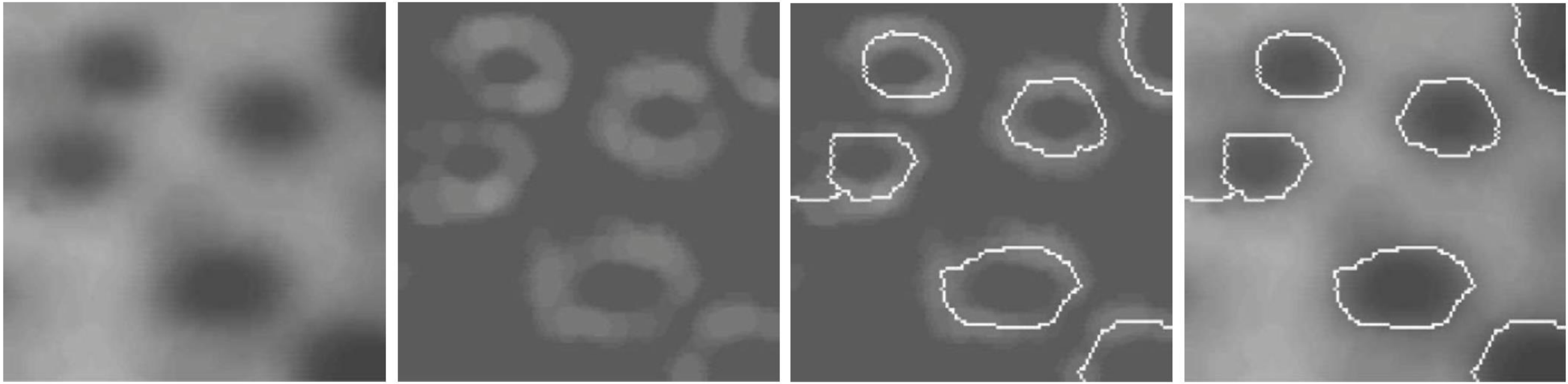
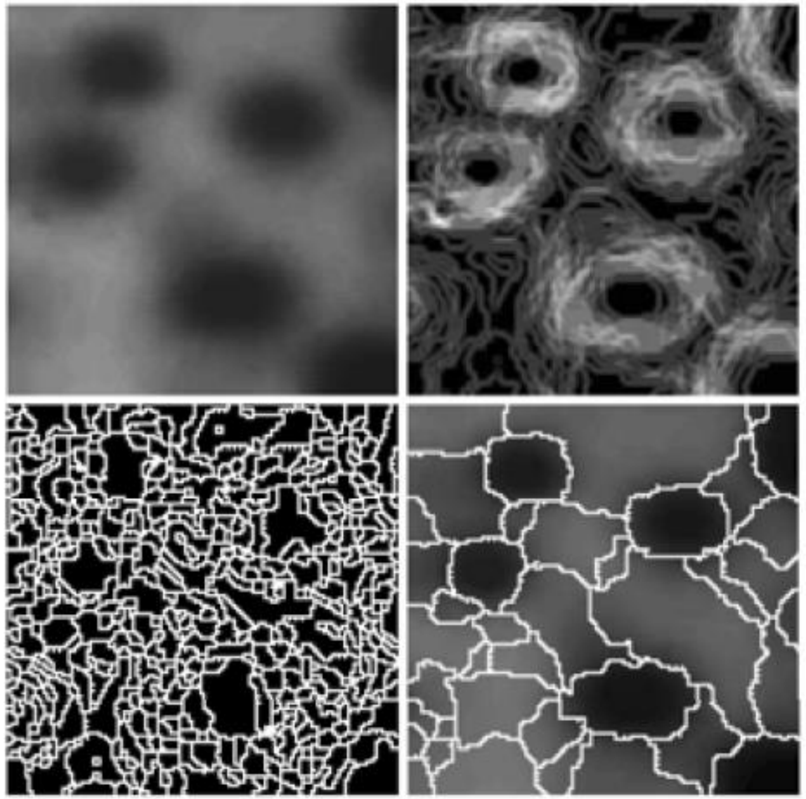


Image of blobs | Image gradient | Watershed lines (on gradient image) | Watershed lines (on original image)

Over-segmentation or fragmentation

Image I

Gradient magnitude image (G)

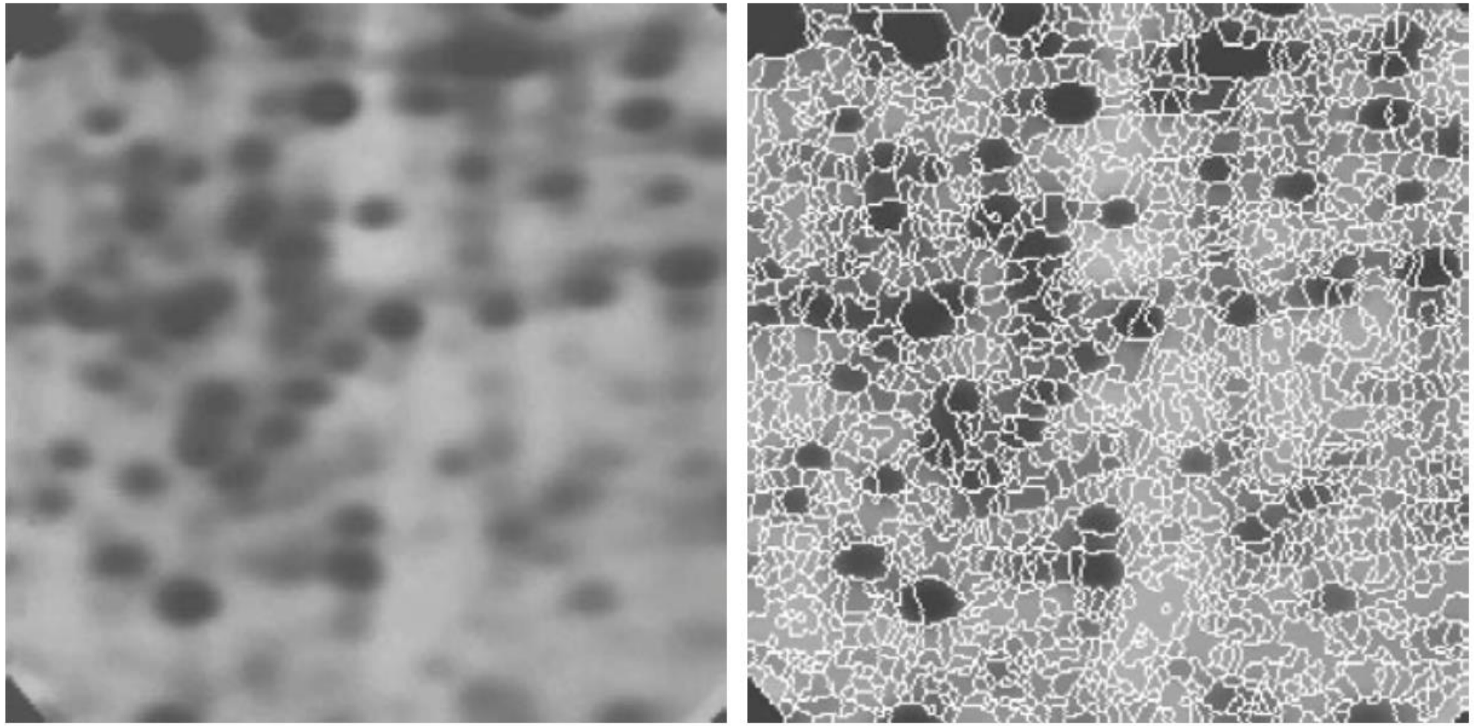


Watershed of G

Watershed of smoothed G

- Using the gradient image directly can cause over-segmentation because of noise and small irrelevant intensity changes.
- Improved by smoothing the gradient image or using markers

Another Over-segmentation example



a b

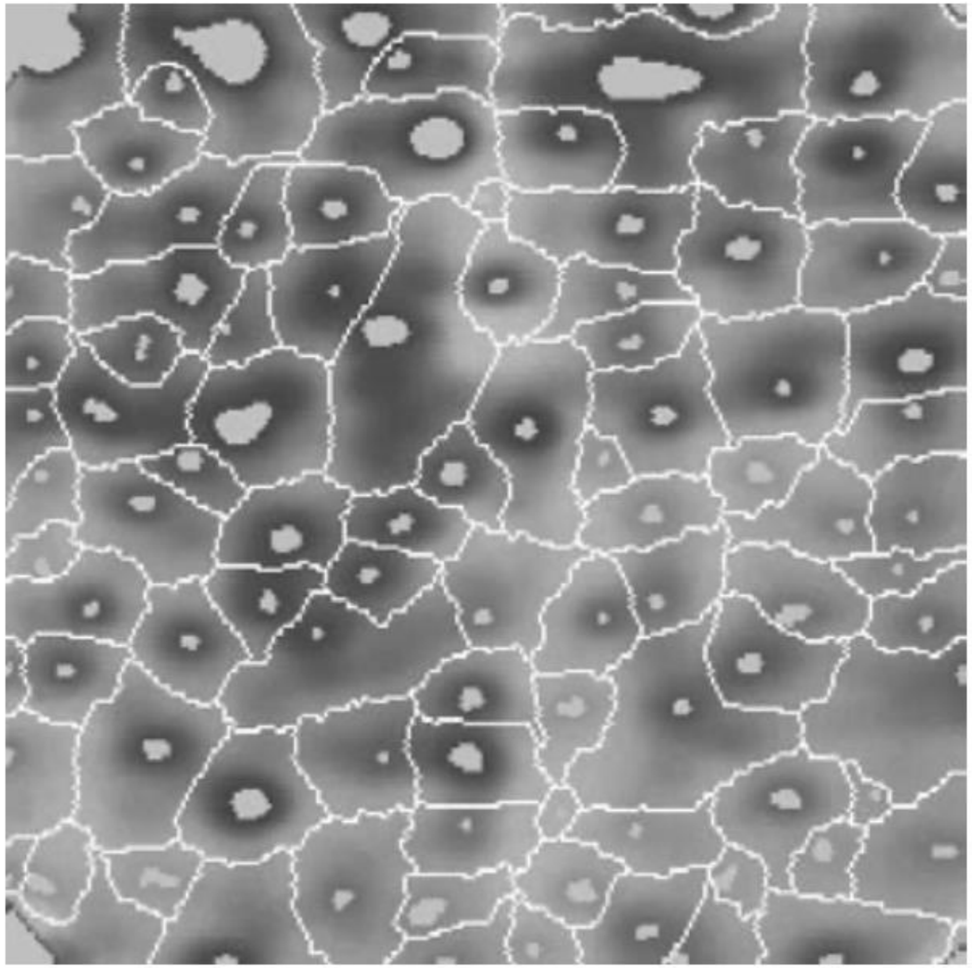
FIGURE 10.60
(a) Electrophoresis image.
(b) Result of applying the watershed segmentation algorithm to the gradient image. Over-segmentation is evident. (Courtesy of Dr. S. Beucher, CMM/ Ecole des Mines de Paris.)

- In this case over-segmentation is evident: too many minima leads to too many regions
- Solution → use markers:
  - Pre-process the image
  - Define a set of criteria that markers must satisfy

Solution: Watershed with markers

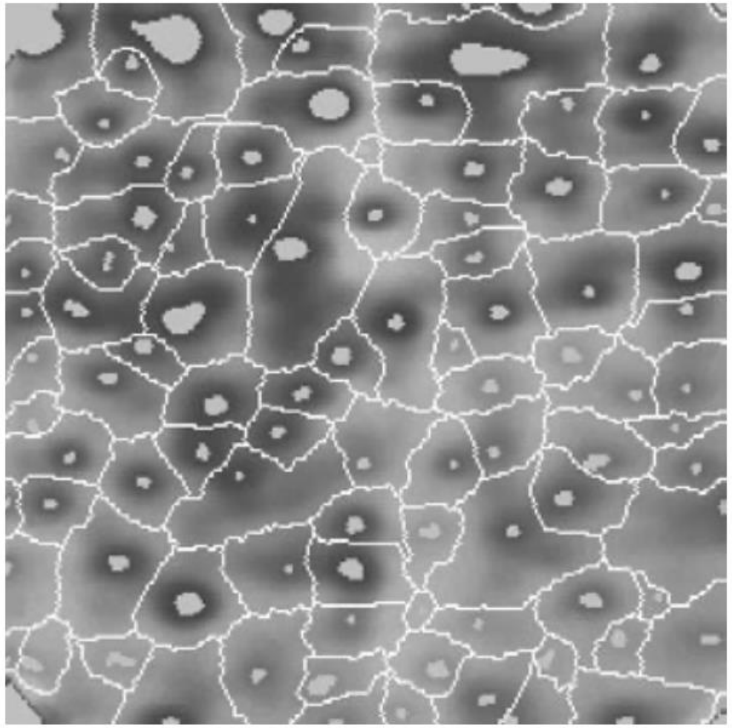
上海科技大学
ShanghaiTech University

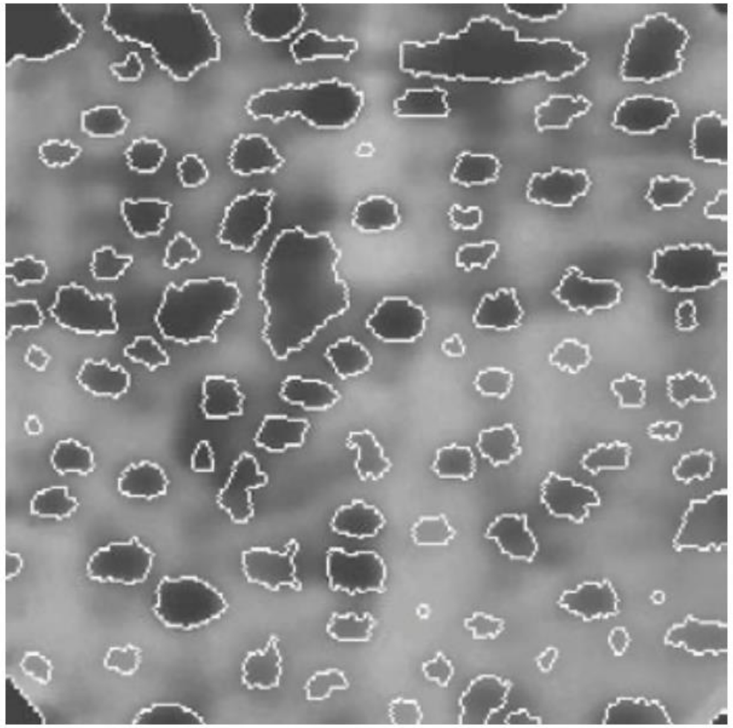
- Internal markers (light gray regions):
  1) A region surrounded by points of "higher" altitude
  2) Such that the points in the region form a connected component
  3) All connected points have the same value
- External markers (watershed lines):
  - Watershed algorithm applied to smoothed image with internal markers as only allowed regional minima
- Internal markers associated to objects
- External markers associated to the background
- Next step: segment each sub-region by any segmentation algorithm



立志成才 报国裕民

Solution: Watershed with markers





Watershed segmentation of the gradient image in each region

- Markers can have more complex descriptions involving size, shape, location, relative distances, texture content, …
- Bring *a priori* knowledge to segmentation

Watershed summary

- Advantages
  - Gives connected components (no broken segmentation lines)
  - A priori information can be implemented in the method using markers
- Disadvantages :
  - Often needs preprocessing to work well
  - Fragmentation or “over-segmentation” can be a problem

Active Contours (SNAKEs)

- Back to boundary detection
  - This time using perceptual grouping.
  - This is non-parametric
- We're not looking for a contour of a specific shape.
  - Just a good contour.

About SNAKEs

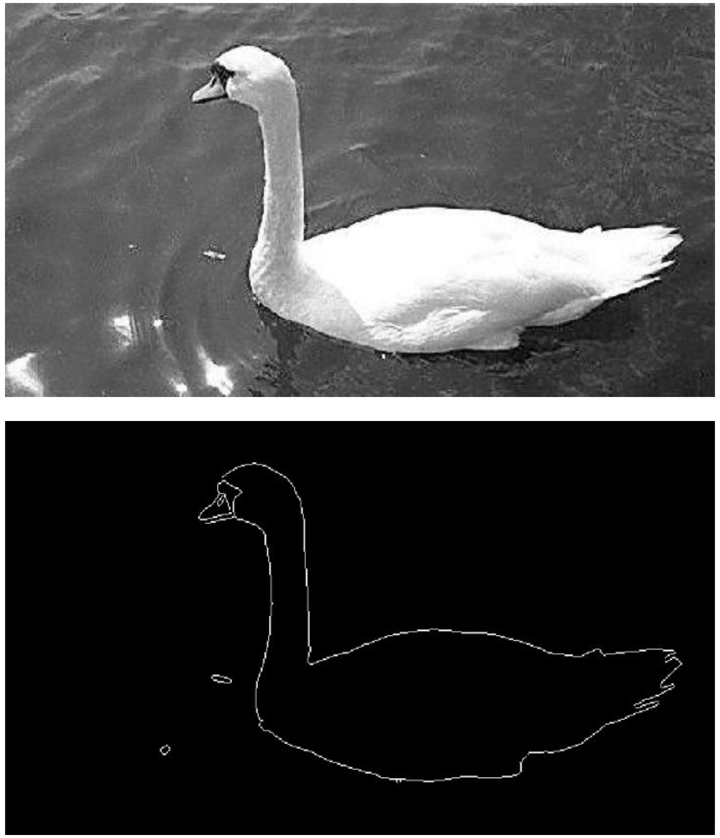
上海科技大学
ShanghaiTech University

- Kass, Witkin and Terzopoulos, IJCV.
- "Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours", by Geiger, Gupta, Costa, and Vlontzos, IEEE Trans. PAMI 17(3)294-302, 1995
- E. N. Mortensen and W. A. Barrett, Intelligent Scissors for Image Composition, in ACM Computer Graphics (SIGGRAPH `95), pp. 191-198, 1995

立志成才 报国裕民

Boundary following

Good case



Bad case

- Boundary following is not easy to use because it is a pixel method
- When looking for boundaries we are looking for overall information

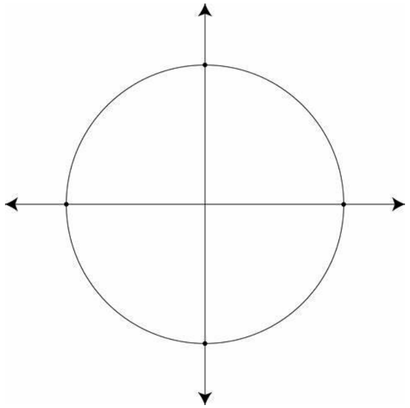
Improve Boundary Detection

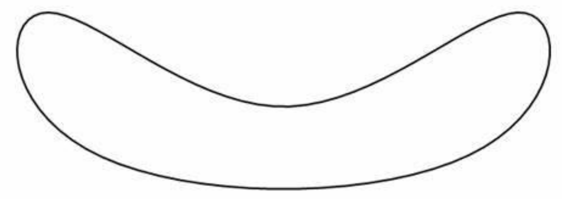
- Idea: segment using curves, not pixels.
- We want a segmentation curve that
  1. **Conforms to image edges.**
  2. **Generates a smooth and varying curve.**

Parametric Curves

- Consider $\begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$ $s \in [0, 1]$ continuous.

- e.g.,



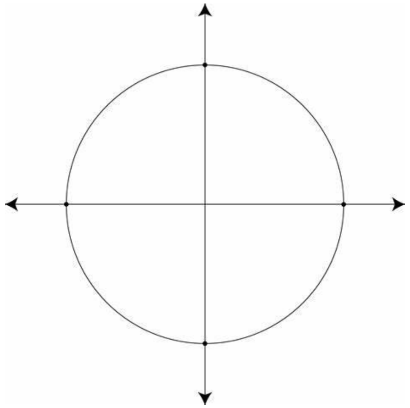


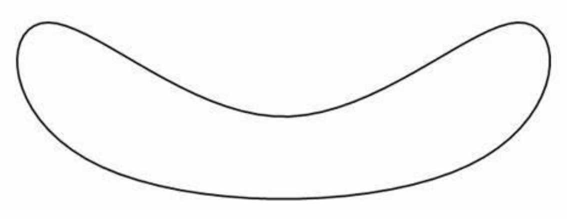
??

Parametric Curves

- Consider $\begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$ $s \in [0, 1]$ continuous.

- e.g.,



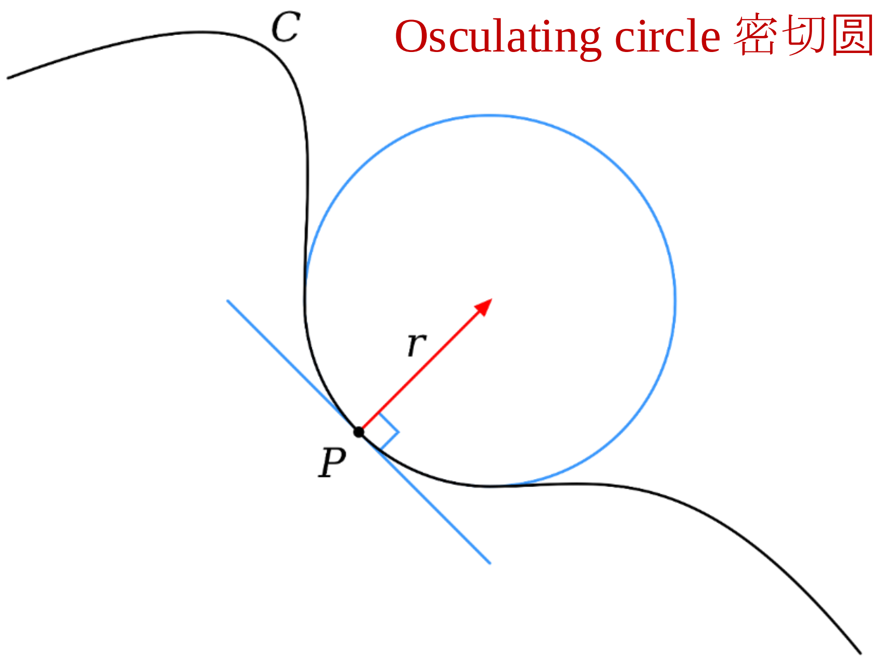


$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} r\cos(2\pi s) \\ r\sin(2\pi s) \end{bmatrix}$$

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} r(s)\cos(2\pi s) \\ r(s)\sin(2\pi s) \end{bmatrix}$$

- **We define a curve using $C(s) = [x(s), y(s)]$.**

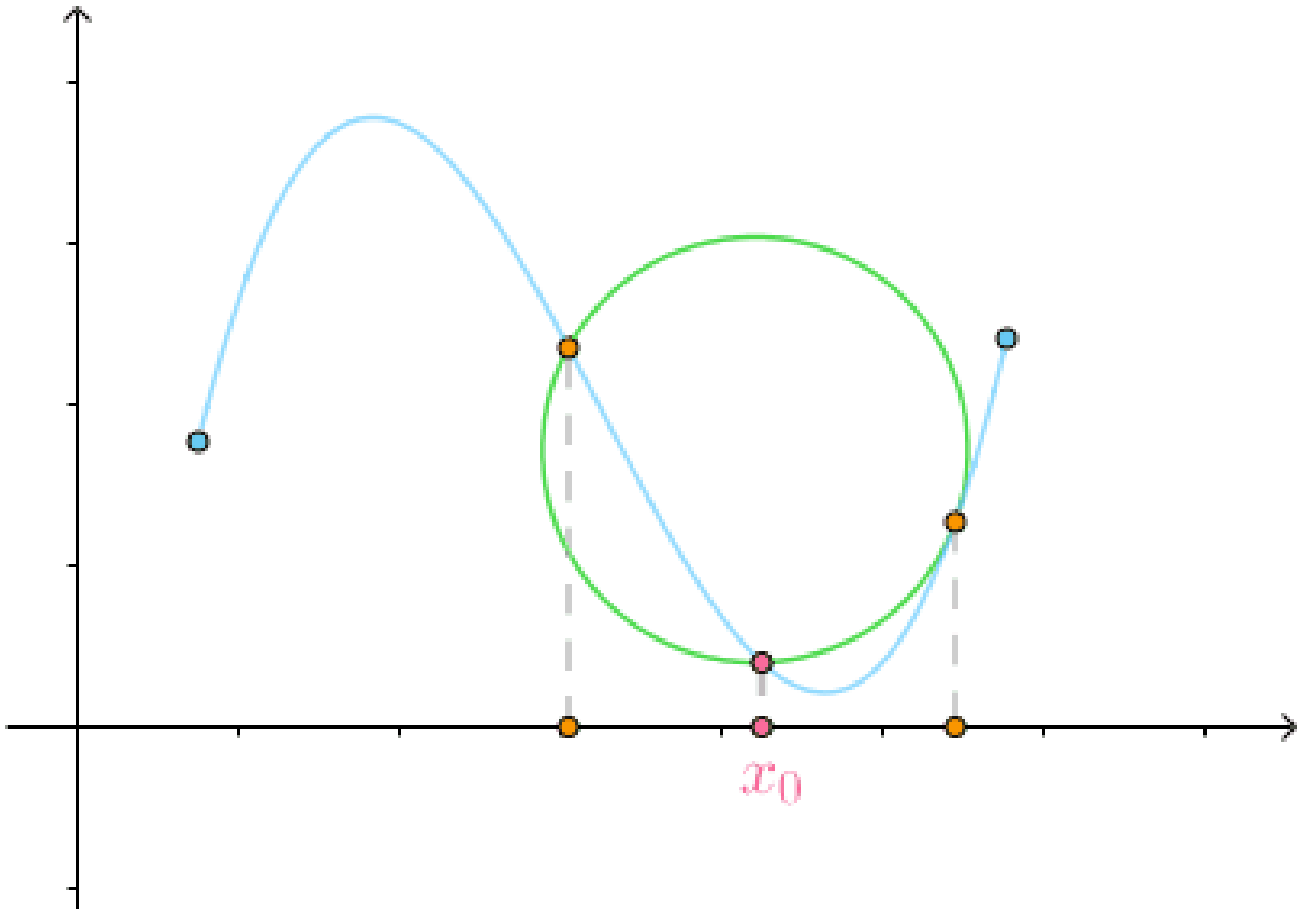
Curvature 曲率



Curvature $K(s) = \frac{1}{R(s)}$

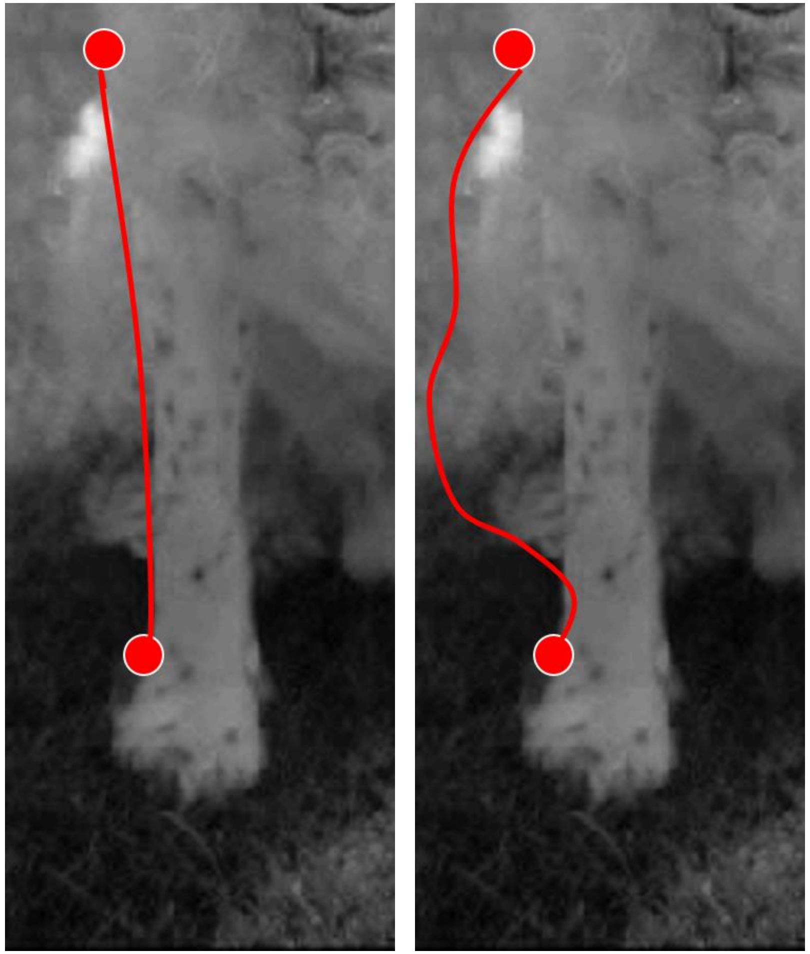
- Let C be a plane curve. The curvature of C at a point is a measure of how sensitive its tangent line is to moving the point to other nearby points.
- It is natural to define the curvature of a straight line to be constantly zero. The curvature of a circle of radius r should be large if r is small and small if r is large. Thus the curvature of a circle is defined to be the inverse of the radius.

Curvature 曲率



Curvature of plane curves

- How do we decide how good a path is? Which of the two paths is better?

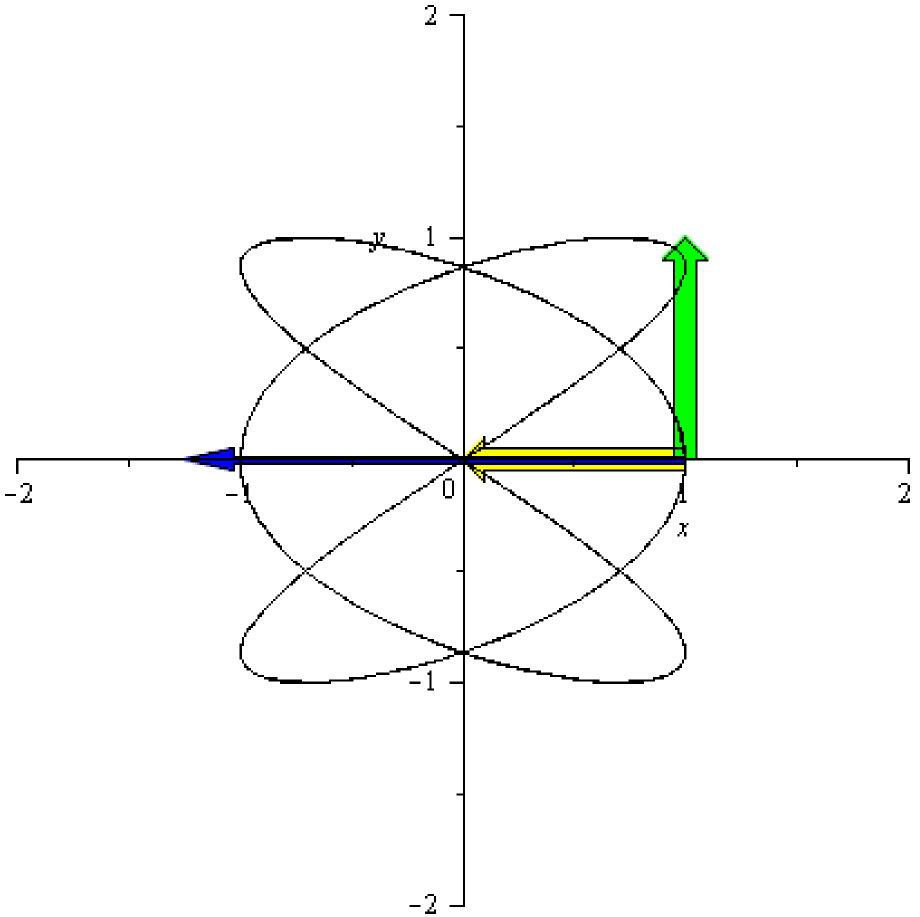


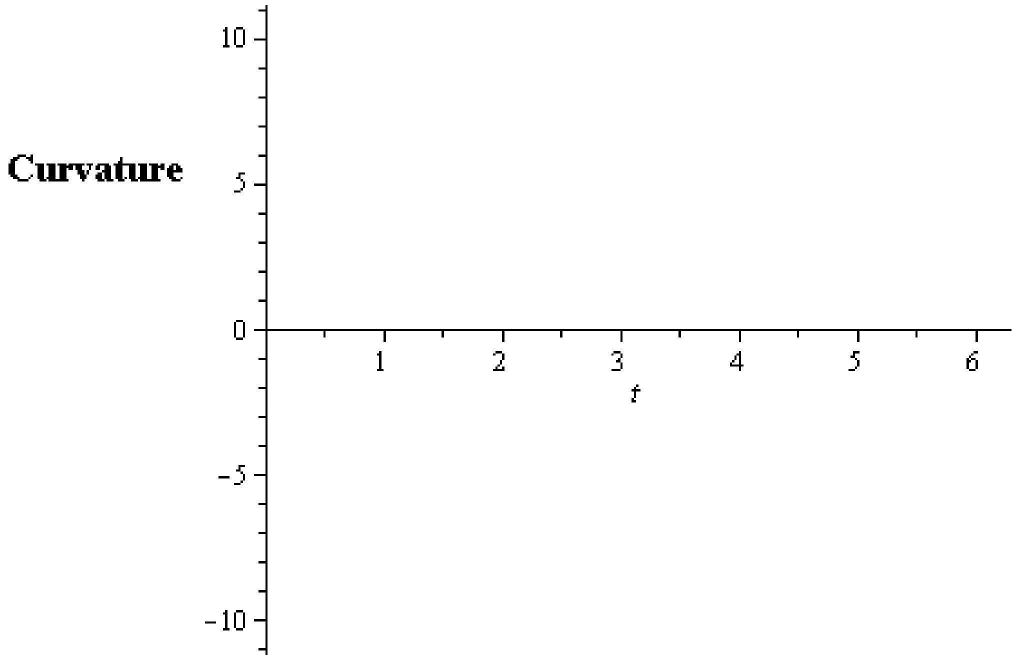
- $T(s) = C'(s)$ is considered as the velocity vector or the unit tangent vector of the curve $C(s)$.
- $k(s) = \frac{1}{R(s)} = C''(s)$ is the curvature of curve $C(s)$.

$$k(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'^2(t) + y'^2(t))^{3/2}} \qquad \boldsymbol{n}(t) = \frac{1}{\|\boldsymbol{\gamma}'(t)\|} \cdot \begin{bmatrix} -y'(t) \\ x't) \end{bmatrix}$$

Curvature of plane curves

Lissajous-Curve with tangent vector (green), normal vector (yellow), and "acceleration vector" (blue)





Find energy for the curve

- Idea: we want to define an energy function $\boldsymbol{E}(\boldsymbol{c})$ that matches our intuition about what makes a good segmentation.
- Curve will iteratively evolve to reduce/minimize $\boldsymbol{E}(\boldsymbol{c})$.
- $\boldsymbol{E}(\boldsymbol{c}) = \boldsymbol{E}_{\boldsymbol{internal}}(\boldsymbol{c}) + \boldsymbol{E}_{\boldsymbol{external}}(\boldsymbol{c})$.
  - $\boldsymbol{E}_{\boldsymbol{internal}}(\boldsymbol{c})$ depends only on the shape of the curve (i.e. make it nice).
  - $\boldsymbol{E}_{\boldsymbol{external}}(\boldsymbol{c})$ depends on image intensities (i.e. fit the image, like its gradient).

Energy for shape of the curve

$$E_{internal}(c) = \int_0^1 \alpha \|c'(s)\|^2 + \beta \|c''(s)\|^2 \, ds.$$

- ❑ ***Low*** $c'(s)$ **keeps curve not too "stretchy"**
- ❑ ***Low*** $c''(s)$ **keeps curve not too "bendy"**

Energy for image intensities

$$E_{external}(c) = \int_0^1 -\|\nabla I(c(s))\|^2 ds$$

$$= \int_0^1 -\{\left[\frac{\partial I}{\partial x}\big(x(s), y(s)\big)\right]^2 + \left[\frac{\partial I}{\partial y}\big(x(s), y(s)\big)\right]^2\}ds$$

- **No edge**, then ∇I=0 , $E_{external}(c)$ =0
- **Big edge**, then ∇I=big positive value , $E_{external}(c)$ = big negative value

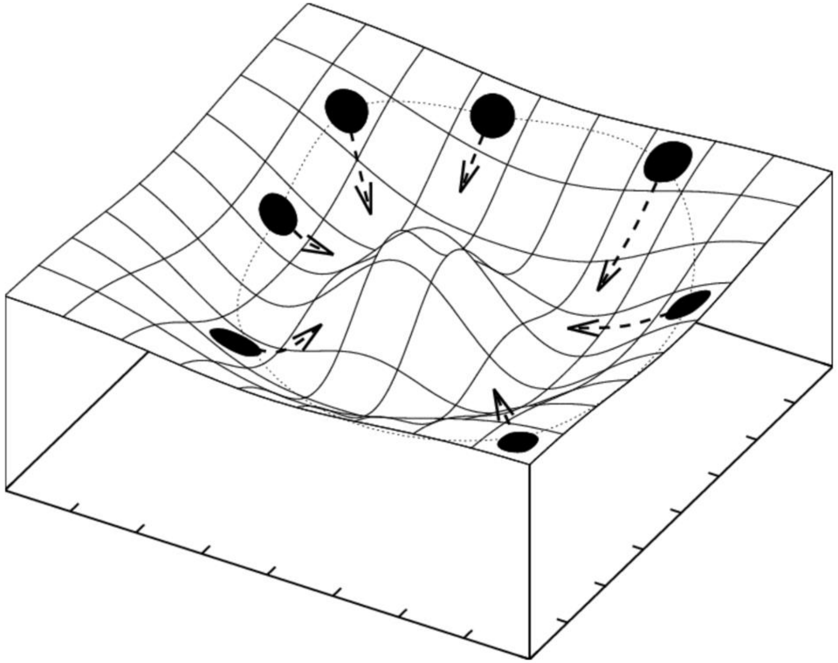
How to minimize *E(c)*?

- **Requires: variational calculus. (变分微积分)**

  1. In practice for digital images, we solve the problem by creating a curve $C(s,t)$. Where t represents the iteration.
  2. Curve approximated by k discrete points (x_i, y_i).
  3. Then we step $C(s,t-1)\ to\ C(s,t)$ by taking a step along gradient of $E(C)$: $\frac{\partial E}{\partial C}$.

- **Result: Curve inches along until points around the perimeter stop changing.**

Snake example



Problem with basic snake (E_{ext})

- Contour never "sees" strong edges that are far away.
- Small gradient: Snake gets hung up.
- When there is no gradient for external Energy, then only internal Energy working.
- Can not work for outer boundary.

Extensions

- Active shape models
- Active appearance models
- Level sets
- FAST: FMRIB's Automated Segmentation Tool

Graph-cut segmentation

- Images as Graphs:
  - A vertex for each pixel

Graph-cut segmentation

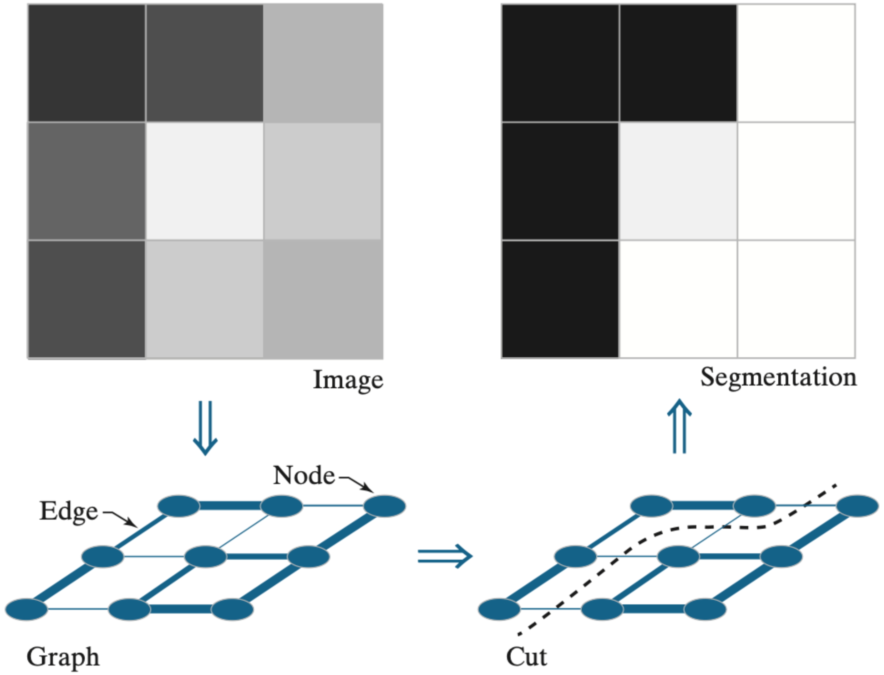
Images as Graphs:

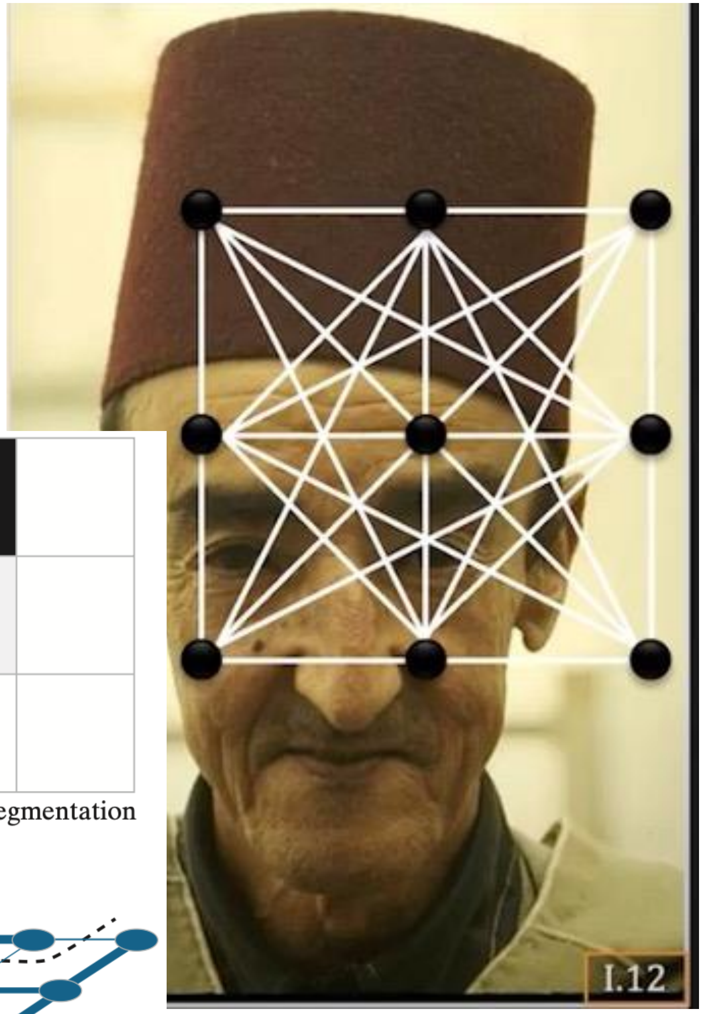
- A vertex for each pixel
- An edge between each pair of pixels
- Graph Notation: G = {V, E}, where V and E are the sets of vertices and edges, respectively: $E \subseteq V \times V$.

$$G = \{V, E\}$$

V: Graph nodes ⟷ Image ={pixels}

E: edges connection nodes　　Pixel similarity





Segmentation = Graph partition

Images as Graphs:

- A vertex for each pixel
- An edge between each pair of pixels
- Graph Notation: G = {V, E}, where V and E are the sets of vertices and edges, respectively: $\mathrm{E} \subseteq V \times V$.
- Pixel Dissimilarity

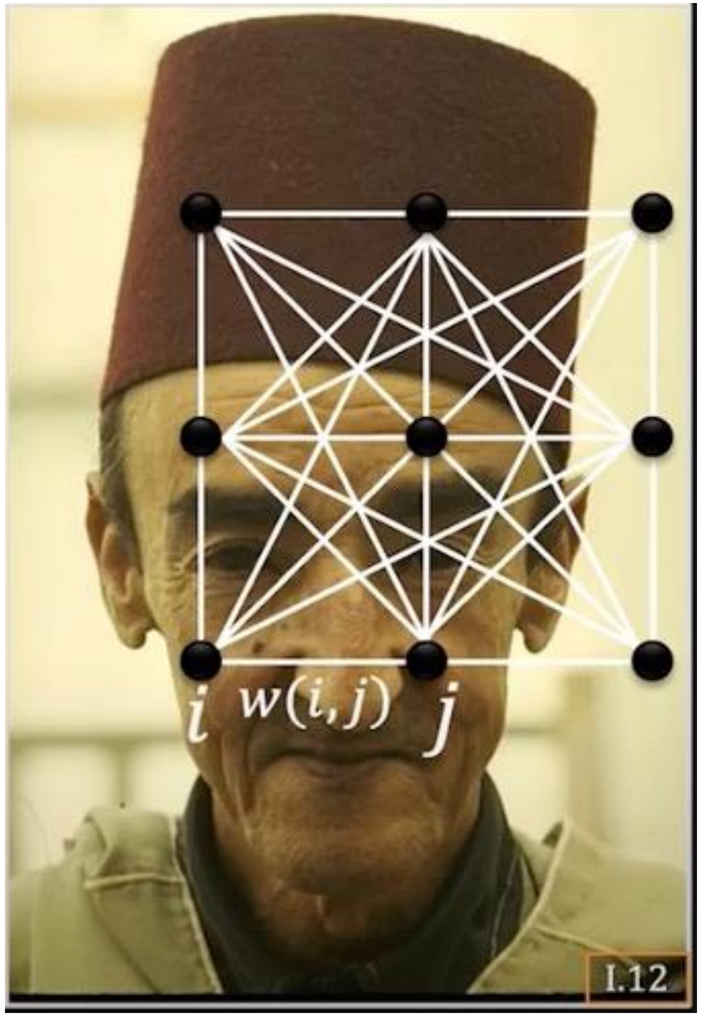
$$S(f_i, f_j) = \sqrt{\sum_k (f_{ik} - f_{jk})^2}$$

- Pixel Affinity or Similarity as weights:

$$w(i,j) = A(f_i, f_j) = e^{\frac{-1}{2\sigma^2} S(f_i, f_j)}$$

- Many different possibilities, for instance:

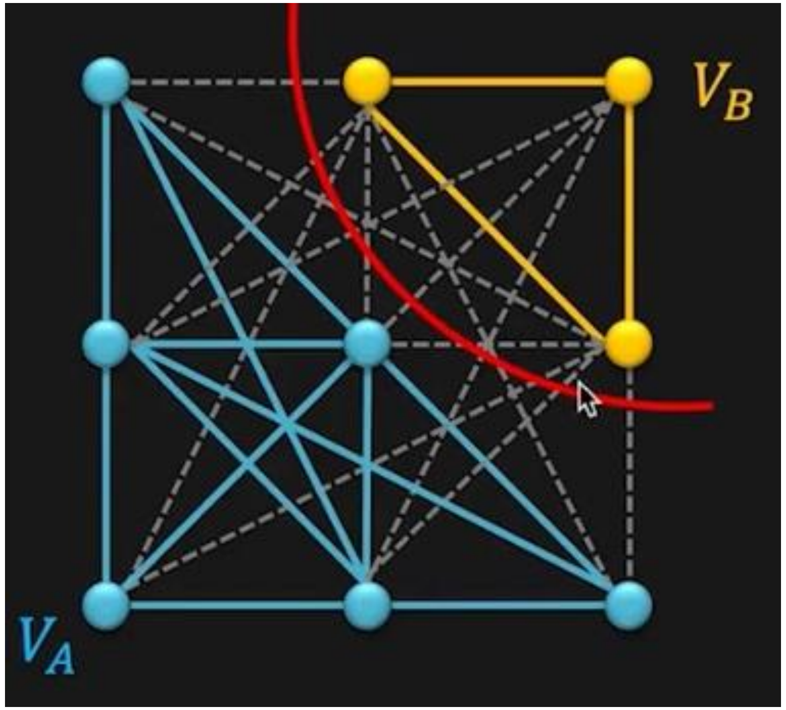
$$w(i,j) = \frac{1}{|I(n_i) - I(n_j)| + c}$$



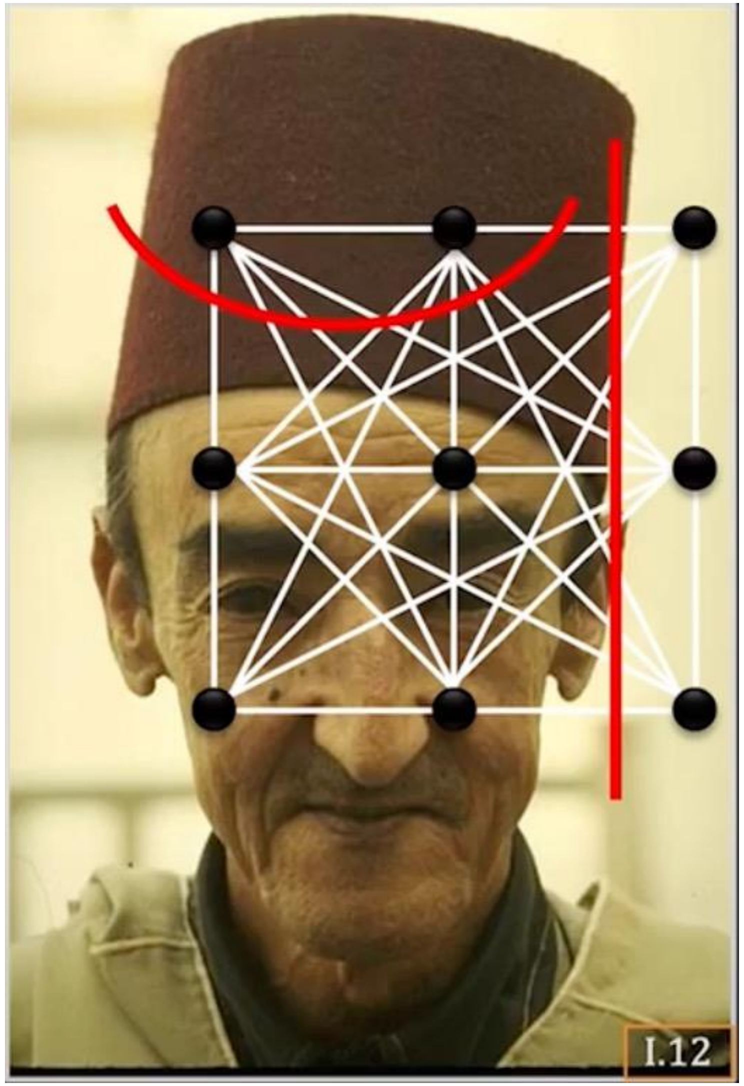
Graph-cut segmentation

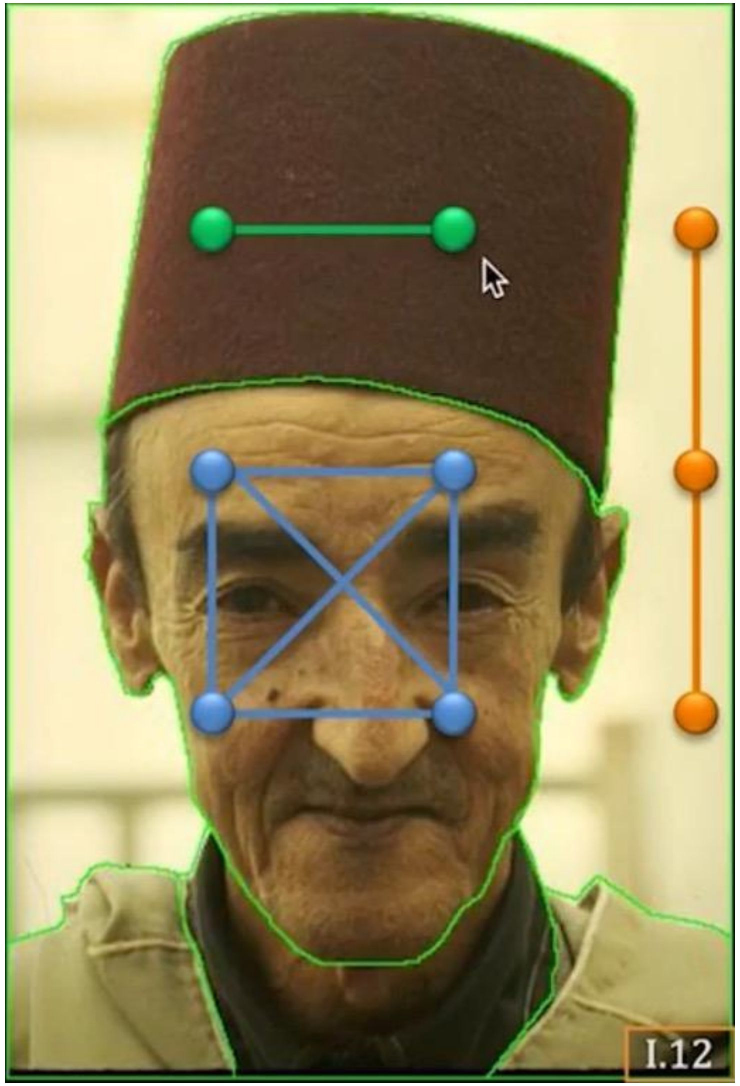
- Cut C = {V_A, V_B} is a partition of vertices V of a graph G = {V, E} into two disjoint subsets V_A, and V_B.
- Cut-set: set of edges whose vertices are in different subsets of partition
- Cost of Cut: Sum of weights of cut-set edges.

$$cut(V_A, V_B) = \sum_{u \in V_A, v \in V_B} w(u, v)$$

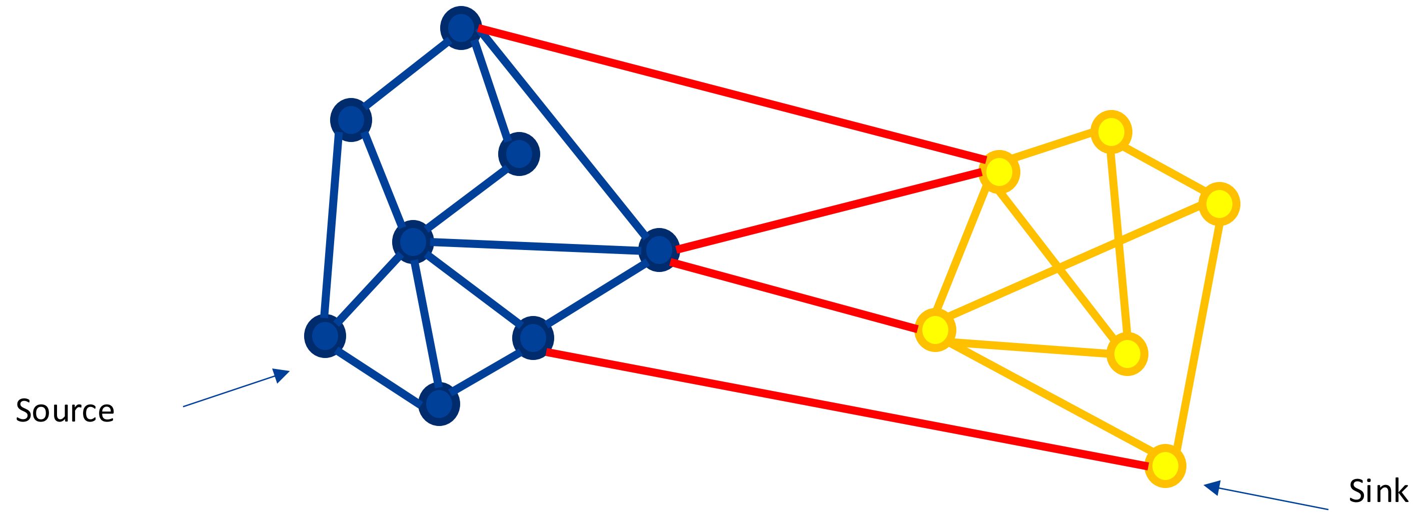


Graph-cut segmentation





Graph Cut and Flow

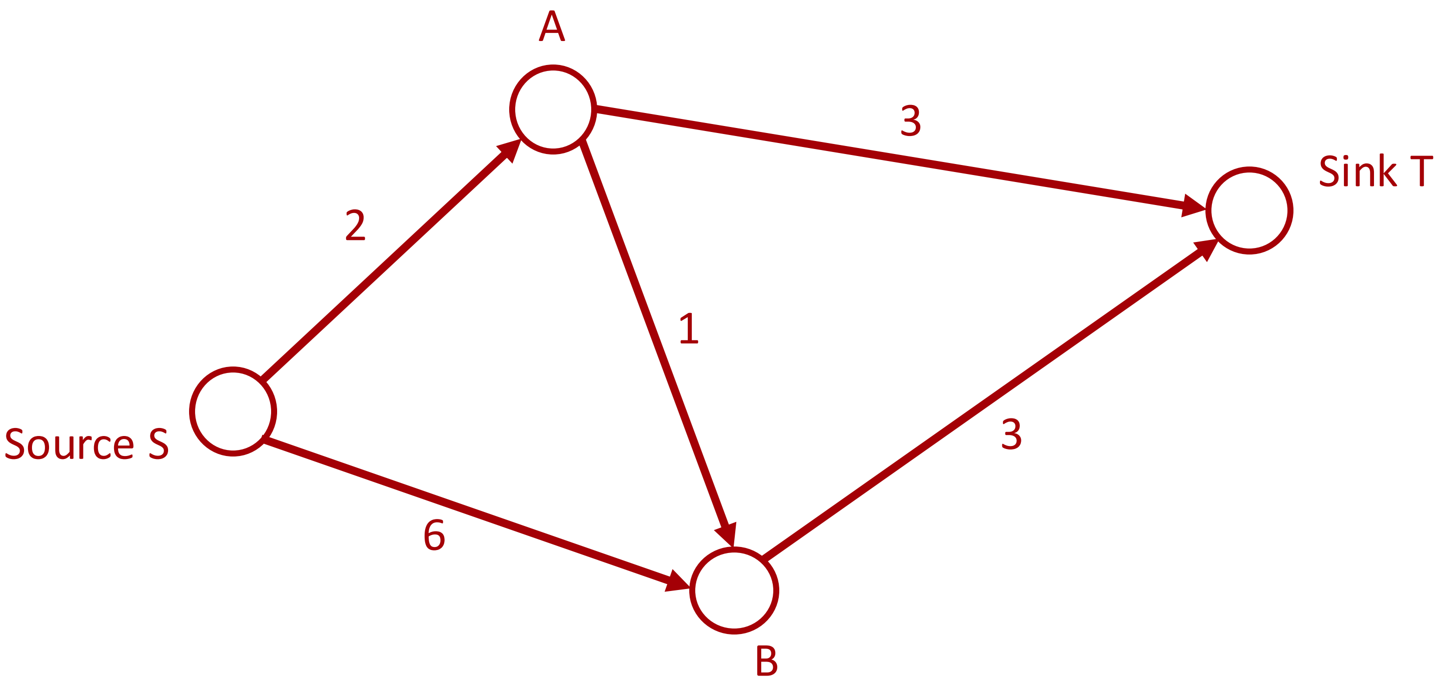


1) Given a source (s) and a sink node (t)

2) Define Capacity on each edge, $C_{ij} \; = W_{ij}$

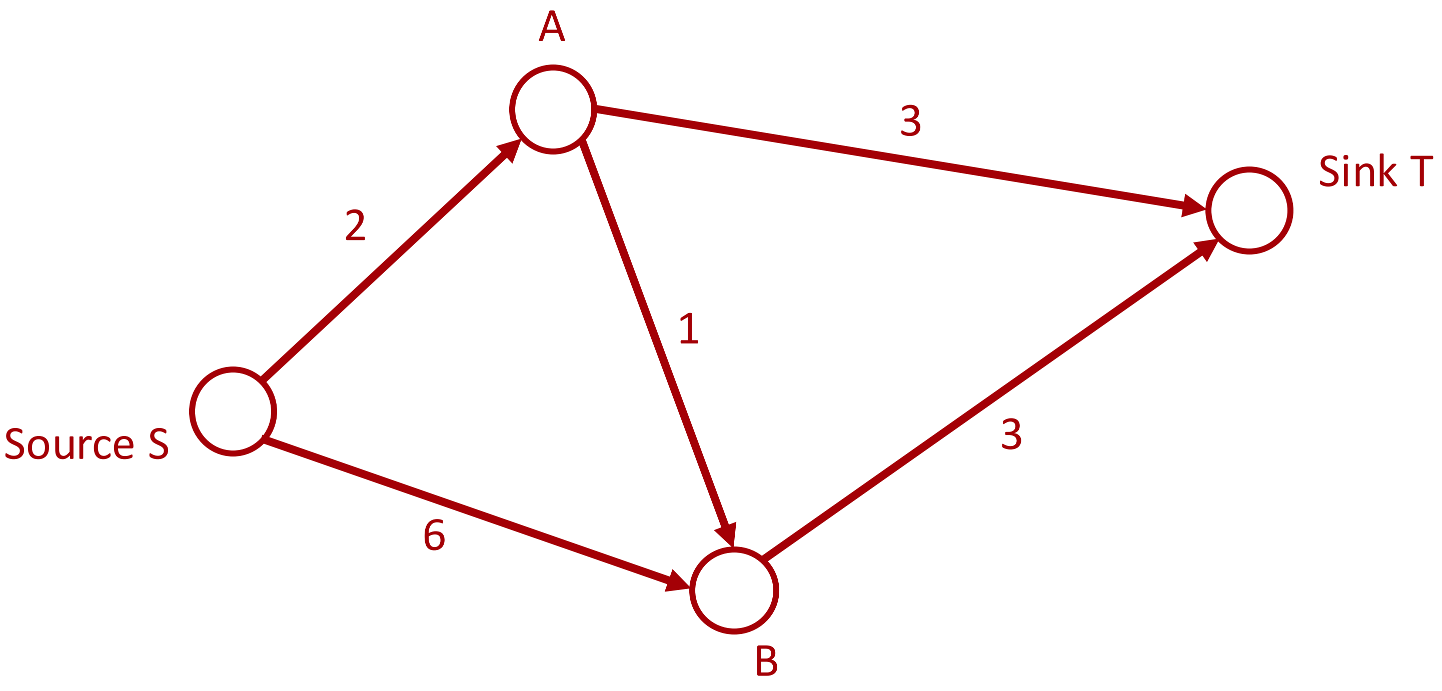
- What is the relationship with the min cut ?

Graph Cut and Flow: "Water pipes" Example



- Possible paths from S to T:
- Min cut ?
- Max flow ?

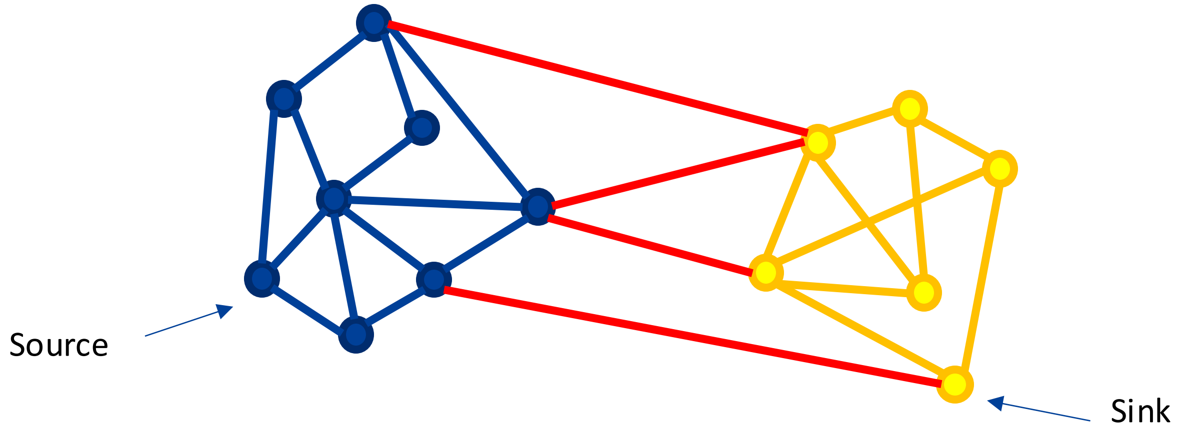
Graph Cut and Flow: "Water pipes" Example



Min-cut = Max-flow

- Possible paths from S to T:
  - S → A → T
  - S → A → B → T
  - S → B → T
- Min cut ?
  - S → A + B → T
  - Min cut = 2+3 = 5
- Max flow ?
  - B → T = 3
  - A → T = 2
  - Max flow = 2 + 3 = 5

Graph Cut and Flow

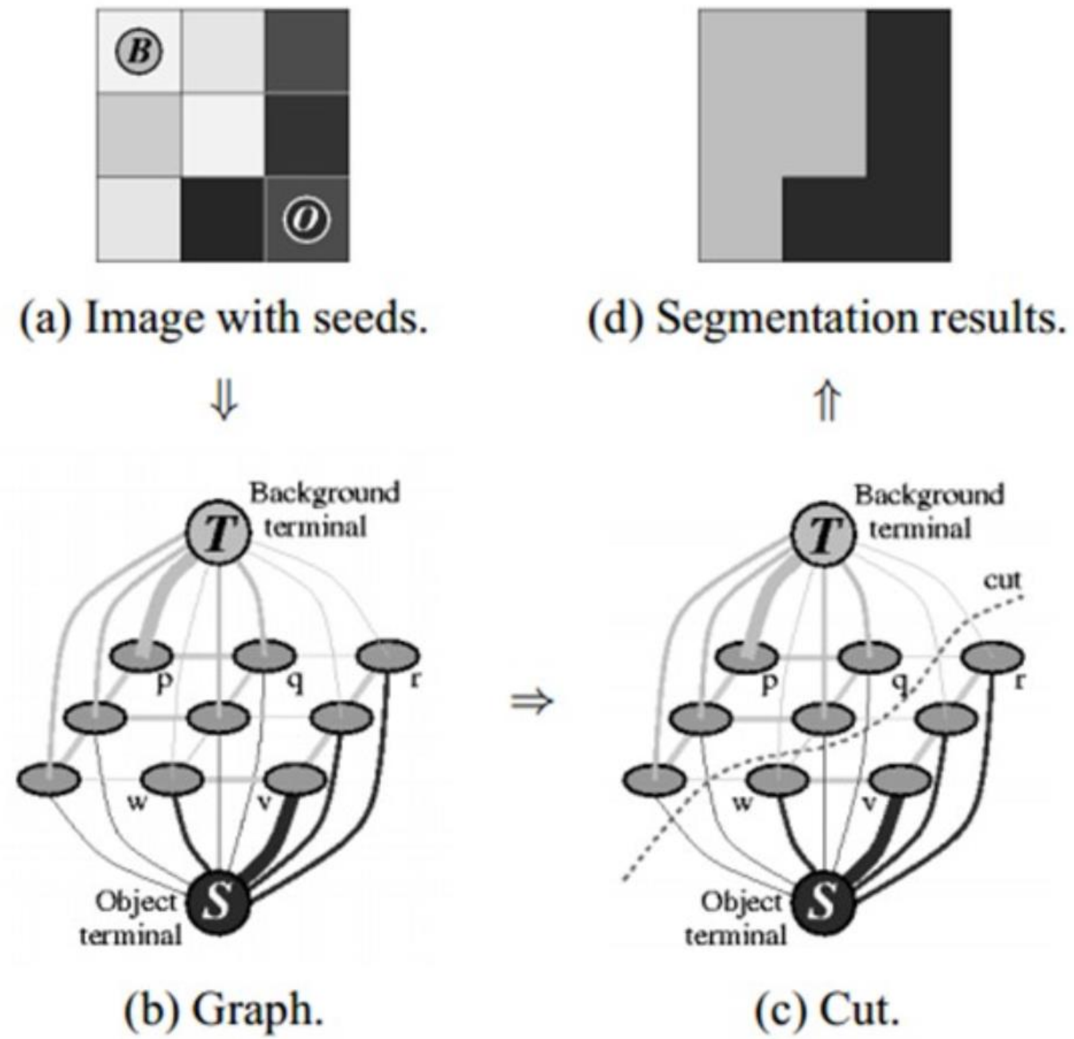


1) Given a source (s) and a sink node (t)
2) Define Capacity on each edge, $C_{ij} = W_{ij}$
3) Find the maximum flow from s->t, satisfying the capacity constraints:

$$\boldsymbol{Min\ Cut = Max\ \mathrm{Flow}}$$

- *Min-cut, max-flow Theorem*

Min Cut



B
O
(a) Image with seeds.
(d) Segmentation results.
Background
terminal
T
p
q
r
w
v
Object
terminal
S
(b) Graph.
cut
(c) Cut.

Min Cut refinements

上海科技大学 ShanghaiTech University

- Some similarity measures can lead to meaningless cuts:

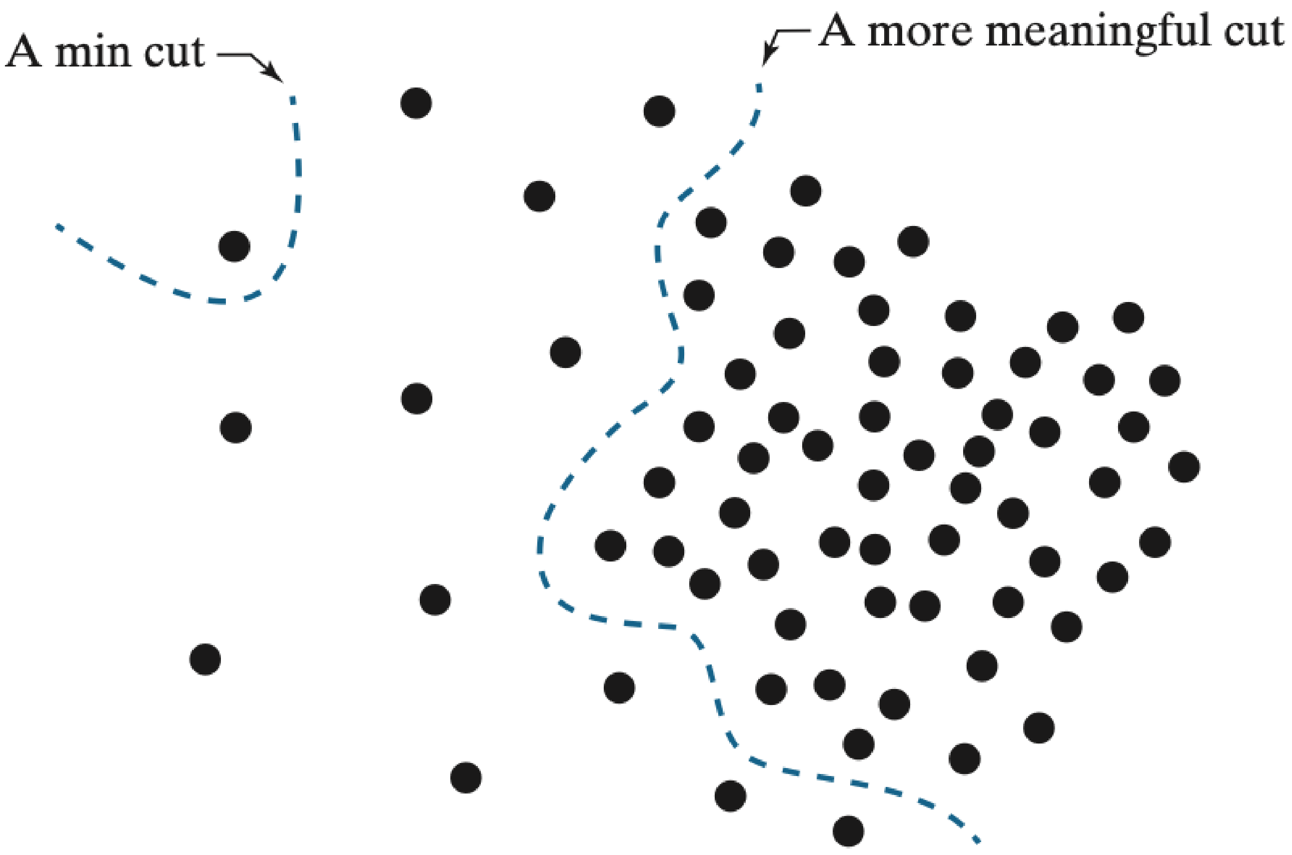


FIGURE 10.55 An example showing how a min cut can lead to a meaningless segmentation. In this example, the similarity between pixels is defined as their spatial proximity, which results in two distinct regions.

立志成才报国裕民

Min Cut refinements

- Solution: Use normalized cut costs
  - Normalized cost:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

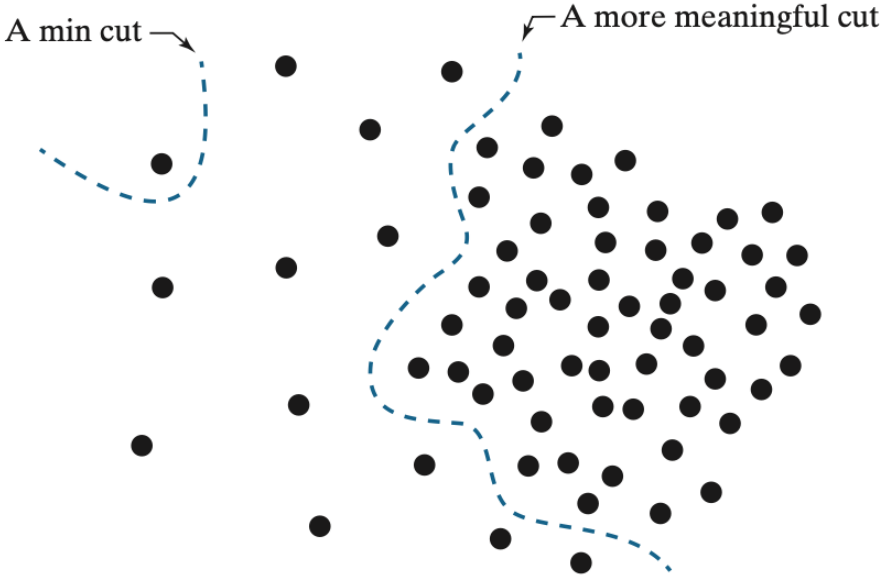
Where

$$assoc(A,V) = \sum_{u \in A, z \in V} w(u,z) \qquad assoc(B,V) = \sum_{v \in B, z \in V} w(v,z)$$

  - Or total normalized association:

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

  - It can be shown that $Ncut(A,B) = 2 - Nassoc(A,B)$



- So minimizing $Ncut$ is equivalent to maximizing $Nassoc$
  - But the problem is now more complicated than Max-Flow

Min Cut Example

$$w(i,j) = \begin{cases} e^{-\frac{[I(n_i)-I(n_j)]^2}{\sigma_I^2}} e^{-\frac{dist(n_i,n_j)}{\sigma_d^2}} & \text{if } dist(n_i,n_j) < r \\ 0 & \text{otherwise} \end{cases}$$

a b c

FIGURE 10.56 (a) Image of size 600 × 600 pixels. (b) Image smoothed with a 25 × 25 box kernel. (c) Graph cut segmentation obtained by specifying two regions.