
Contents

List of algorithms	ii
7 Segmentation II	1
7.1 Mean Shift Segmentation	2
7.2 Fuzzy Connectivity	20
7.3 Towards 3D graph-based image segmentation	39
7.3.1 Simultaneous detection of border pairs	40
7.3.2 Sub-optimal surface detection	47
7.4 Graph cut segmentation	51
7.5 References	70

List of algorithms

7.1	Mean shift mode detection	10
7.2	Mean shift discontinuity preserving filtering	15
7.3	Mean shift image segmentation	18
7.4	Absolute fuzzy connectivity segmentation	27
7.5	Fuzzy object extraction	31
7.6	Graph cut segmentation	61

Chapter 7

Segmentation II

- Image segmentation methods rapidly developing
- New approaches must cope with increasing image sizes and increasing dimensionality
- three-dimensional or higher-dimensional capabilities are needed

7.1 Mean Shift Segmentation

- Optimal thresholding approaches introduced earlier are based on estimation of object and background statistics — frequently assuming normality of distributions.
- **Mean shift image segmentation** avoids estimation of the probability density function.
- Mean shift consists of 2 main steps
 1. **discontinuity preserving filtering**
 2. **mean shift clustering**

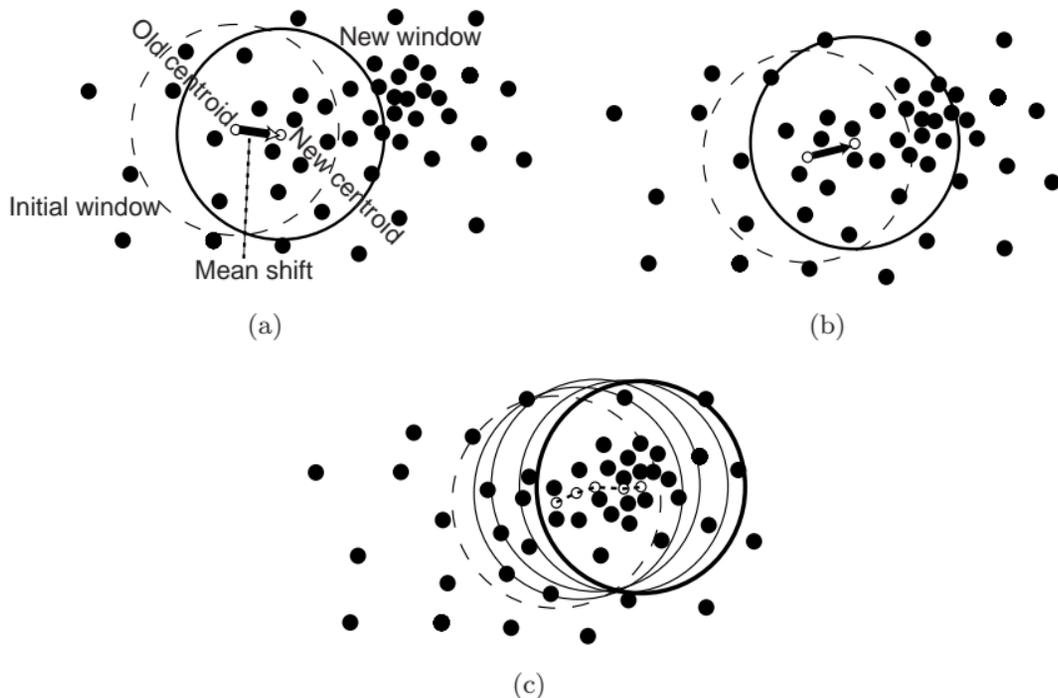


Figure 7.1: Principle of the mean shift procedure. The most dense region of data is identified in an iterative process. (a) The initial region of interest is randomly positioned over data and its centroid is determined. The new region is moved to the location of the identified centroid. The vector determining the region's positional change is the mean shift. (b) Next step of the mean shift procedure—a new mean shift vector is determined and the region is moved accordingly. (c) The mean shift vectors are determined in the remaining steps of the procedure until convergence. The final location identifies the local

- size and shape of ROI = size and shape of **multivariate density kernel estimator** – the only free parameters (Figure 7.1)
- radially symmetric kernels $K(\mathbf{x})$

$$K(\mathbf{x}) = c k(\|\mathbf{x}\|^2), \quad (7.1)$$

c is a strictly positive constant that makes $K(\mathbf{x})$ to integrate to one.

- Typical kernels
 - **normal** kernel $K_N(\mathbf{x})$

$$K_N(\mathbf{x}) = c \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right), \quad (7.2)$$

kernel profile $k_N(x)$

$$k_N(x) = \exp\left(-\frac{1}{2}x\right) \quad \text{for } x \geq 0. \quad (7.3)$$

often symmetrically truncated – finite support

– Epanechnikov kernel $K_E(\mathbf{x})$

$$K_E(\mathbf{x}) = \begin{cases} c(1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| \leq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (7.4)$$

kernel profile $k_E(x)$

$$k_E(x) = \begin{cases} 1 - x & \text{for } 0 \leq x \leq 1, \\ 0 & \text{for } x > 1, \end{cases} \quad (7.5)$$

not differentiable at the boundary.

- Given n data points \mathbf{x}_i in d -dimensional space R^d ,
- multivariate kernel density estimator $\tilde{f}_{h,K}(\mathbf{x})$ computed at point \mathbf{x} is

$$\tilde{f}_{h,K}(\mathbf{x}) = \frac{1}{n h^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \quad (7.6)$$

where h represents the kernel size = kernel **bandwidth**

- \Rightarrow Locating zeros of the gradient of $f_{h,K}(\mathbf{x})$, i.e., identifying \mathbf{x} for which $\nabla f_{h,K}(\mathbf{x}) = 0$ (Figure 7.1)
- Mean shift identifies these locations *without* estimating the underlying probability density function
- from estimating the *density*, the problem becomes one of estimating the **density gradient**

$$\nabla \tilde{f}_{h,K}(\mathbf{x}) = \frac{1}{n h^d} \sum_{i=1}^n \nabla K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right). \quad (7.7)$$

- $k(x)$ being the kernel's profile,
- assuming that its derivative exists $-k'(x) = g(x)$ for all $x \in [0, \infty)$ except for a finite set of points,

$$K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = c_k k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right), \quad (7.8)$$

c_k is a normalizing constant, h represents the kernel size

- profile $g_E(x)$ is uniform if $K(\mathbf{x}) = K_E(\mathbf{x})$
- for $K(\mathbf{x}) = K_N(\mathbf{x})$, profile of $g_N(x)$ is defined by the same exponential expression as $k_N(x)$

- $g(x)$ being a profile-defining kernel $G(\mathbf{x}) = c_g g(\|\mathbf{x}\|^2)$.
- successive locations $\{\mathbf{y}_j\}_{j=1,2,\dots}$ of kernel G are then

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}, \quad (7.9)$$

where \mathbf{y}_1 is the initial position of the kernel G

- first sum ... sum of “distances” from origin
- second sum ...how many points within kernel
- assume 1-D example
 - pixel values present 2,3,4,6,9,...
 - $h = 2.5$, $\mathbf{y}_0 = 0$
 - sequence of $\mathbf{y} = 0, 2, 3, 3 = \text{convergence}$
- corresponding sequence of density estimates computed with kernel K is therefore

$$\tilde{f}_{h,K}(j) = \tilde{f}_{h,K}(\mathbf{y}_j). \quad (7.10)$$

- if kernel K has a convex and monotonically decreasing profile, the sequences $\{\mathbf{y}_j\}_{j=1,2,\dots}$ and $\{\tilde{f}_{h,K}(j)\}_{j=1,2,\dots}$ converge while $\{\tilde{f}_{h,K}(j)\}_{j=1,2,\dots}$ increases monotonically [Comaniciu and Meer 02]
- guaranteed convergence of mean shift algorithm to local maximum of a probability density function due to the mean shift vector magnitude converging to zero
- convergence speed depends on kernel employed
- with *Epanechnikov* kernel on discrete data (uniform kernel profile), convergence achieved in a finite number of steps
- with data point weighting (e.g., using *normal* kernel) – mean shift procedure is infinitely convergent
- \Rightarrow small lower bound value of change between steps may stop the convergence process.
- use of a normal kernel, while typically slower, almost always outperforms the Epanechnikov kernel as far as quality of the result is concerned.
- Peter Meer = the differences if any - are very small

- set of all locations that converge to the same mode $\mathbf{y}_{\text{con}} ==$ **basin of attraction** associated with this mode
- convergence may also stop at a local plateau or a saddle point
- to avoid it, stationary points (seemingly a point of convergence) are perturbed by small random vectors and mean shift procedure is restarted
- if process converges to the same location (tolerance allowed) \rightarrow local maximum = density mode

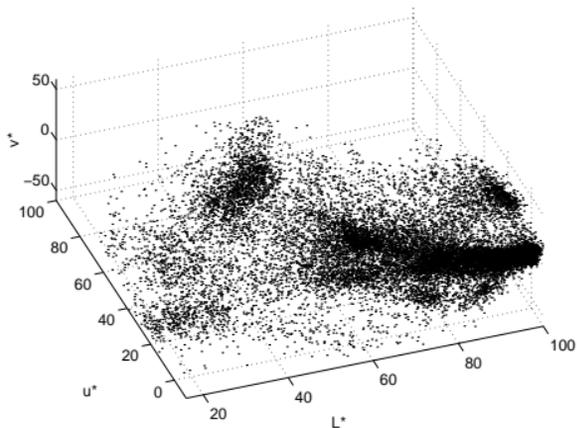
Algorithm 7.1: Mean shift mode detection

1. Using multiple initializations covering the entire feature space, employ the mean shift procedure to identify the stationary points of $\tilde{f}_{h,K}$.
2. Prune these points to only retain the local maxima corresponding to the density modes.

- Advantages:
 - generality
 - noise robustness (arbitrary cluster shapes and feature spaces)
 - the only parameter – size h of the kernel – actually has a physical and understandable meaning
- Disadvantages:
 - choice of h is not always trivial
 - too large a value of h may cause modes to be merged
 - too small a value allows introduction of insignificant additional modes \Rightarrow artificial cluster splitting

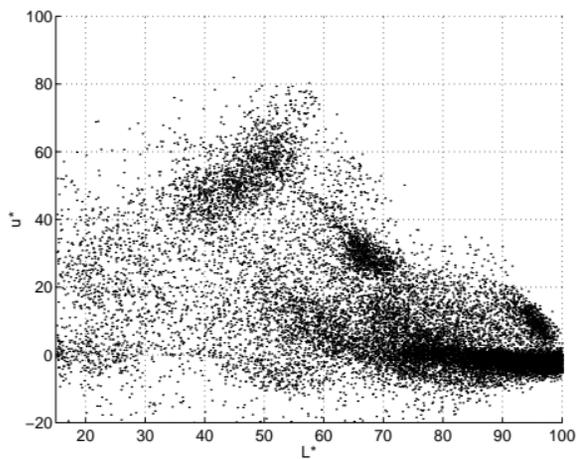


(a)

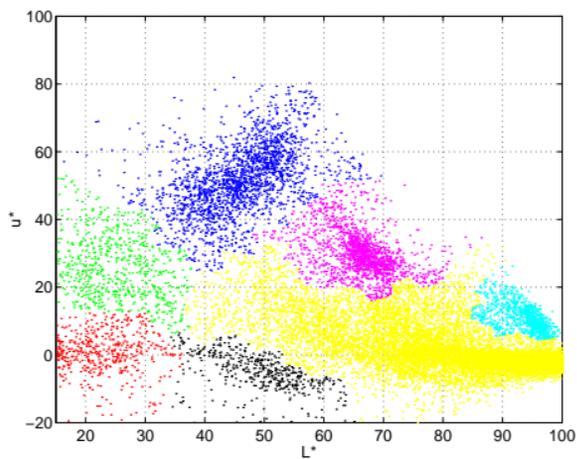


(b)

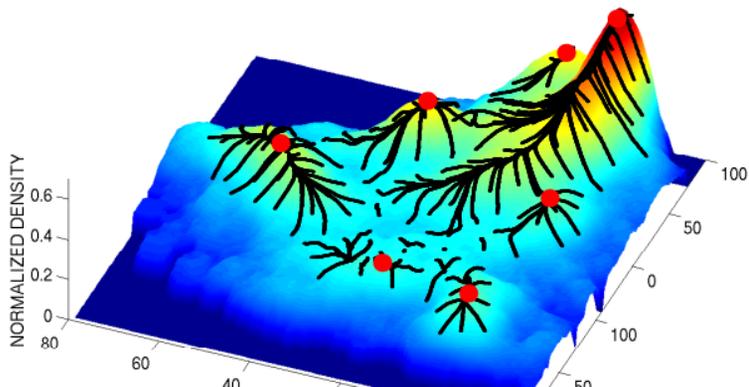
Figure 7.2: Color image L, u, v feature space. (a) Example color image. (b) Corresponding L, u, v feature space. *Courtesy of P. Meer, Rutgers University, ©2002 IEEE [Comanicu and Meer 02]. A color version of this figure may be seen in the color inset—Plate 1.*



(a)



(b)



- d -dimensional image represented by a d -dimensional grid (**spatial domain**) of p -dimensional pixels (voxels)
- p ... number of spectral bands (**range domain**)
- $p = 1$ for grayscale images, $p = 3$ for color images, etc.
- Assuming Euclidean metric for both domains, spatial and range vectors = complete information about pixel's location and properties \Rightarrow concatenate to form a joint spatial-range domain
- resulting joint-domain kernel $K_{h_s, h_r}(\mathbf{x})$ consists of two radially symmetric kernels with parameters h_s and h_r (spatial- and range-domain kernel sizes)
- p, d denote the space dimensionality.

$$K_{h_s, h_r}(\mathbf{x}) = \frac{c}{h_s^d, h_r^p} k \left(\left\| \frac{\mathbf{x}^s}{h_s} \right\|^2 \right) k \left(\left\| \frac{\mathbf{x}^r}{h_r} \right\|^2 \right), \quad (7.11)$$

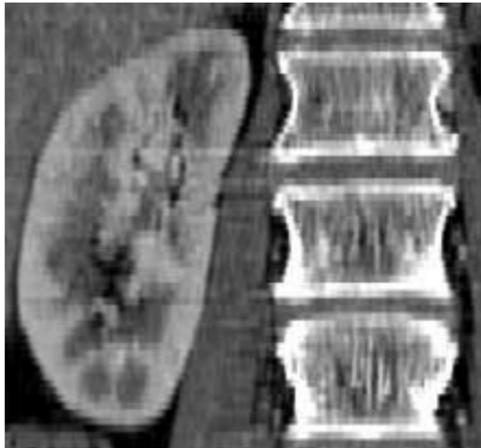
\mathbf{x}^s and \mathbf{x}^r ... spatial and range parts of a feature vector

- $k(x)$ is common profile used in both domains
- c = normalization constant
- resolution level of the mode detection is set by two parameters of a single vector $h = (h_s, h_r)$

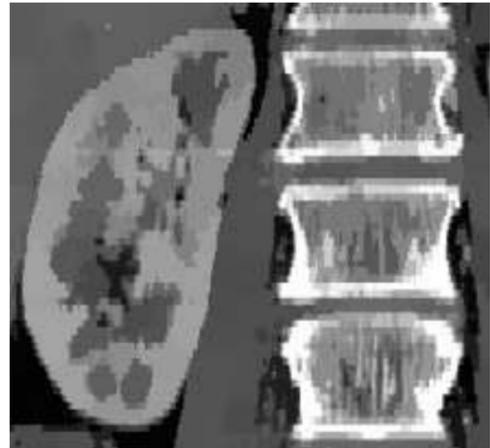
- *Mean shift image segmentation ...* 2-step sequence of *discontinuity preserving filtering* and *mean shift clustering*
- original pixels: \mathbf{x}_i
- filtered image pixels \mathbf{z}_i (in joint spatial–range domain)

Algorithm 7.2: Mean shift discontinuity preserving filtering

1. For each image pixel \mathbf{x}_i , initialize step $j = 1$ and $\mathbf{y}_{i,1} = \mathbf{x}_i$.
2. Compute $\mathbf{y}_{i,j+1}$ as given in equation (7.9) until convergence $\mathbf{y}_{i,\text{con}}$.
3. The filtered pixel values are defined as $\mathbf{z}_i = (\mathbf{x}_i^s, \mathbf{y}_{i,\text{con}}^r)$, i.e., the value of the filtered pixel at the location \mathbf{x}_i^s is assigned the image value of the pixel of convergence $\mathbf{y}_{i,\text{con}}^r$.



(a)



(b)

Figure 7.4: Meanshift filtering. (a) Original X-ray computed tomography image of human kidney and spine. (b) Filtered image. *Courtesy of R. Beichel, Graz University of Technology.*

DEMO - EDISON - FILTERING

- Once image is filtered, *mean shift image segmentation* takes advantage of association between filtered image pixels \mathbf{z}_i and significant modes of joint domain density
(in pixel's neighborhood after pruning nearby less significant modes)
- L_i ... segmentation label associated with pixel i in segmented image

Algorithm 7.3: Mean shift image segmentation

1. Employ the *mean shift discontinuity preserving filtering* and store all information about the d -dimensional convergence points $\mathbf{y}_{i,\text{con}}$.
2. Determine the clusters $\{C_p\}_{p=1,\dots,m}$ by grouping all \mathbf{z}_i , which are closer than h_s in the spatial domain and h_r in the range domain. In other words, merge the *basins of attraction* of these convergence points.
3. Assign $L_i = \{p | \mathbf{z}_i \in C_p\}$ for each pixel $i = 1, \dots, n$.
4. If desired, eliminate regions smaller than P pixels (post-processing).

DEMO - EDISON - SEGMENTATION



(a)



(b)



7.2 Fuzzy Connectivity

- many segmentation methods use crisp (or *hard-coded*) relationships between or within the individual regions
- these relationships may vary across image due to
 - noise
 - uneven illumination
 - limited spatial resolution
 - partial occlusions
 - etc.
- **Fuzzy connectivity** segmentation considers these uncertainties
- describes segmentation task with fuzzy rules
 - *if two regions have about the same gray-value and if they are relatively close to each other in space, then they likely belong to the same object*
- framework for such a reasoning approach ... **fuzzy logic**

- image pixels seem to *hang together* when forming an object
- **hanging togetherness** property is described using fuzzy logic
- spatial relationships determined for *each* pair of image elements in the entire image
- local and global image properties are considered

- *local* fuzzy relation = **fuzzy affinity**, $\psi \in [0, 1]$
- represents strength of *hanging togetherness* of nearby image elements = spels
- spel = pixel in 2-D, = voxel in 3-D
- *affinity* is a function of distance between two fuzzy adjacent image elements considering image-derived properties (e.g., image edges, intensities, etc.)
- image I can be represented by a pair $I = (C, f)$... C represents image domain and f represents local image properties
- $f(c) \in [0, 1]$... normalized image property (feature) associated with spel c

- **fuzzy adjacency** $\mu(c, d) \in [0, 1]$ of two elements c, d
- *Hard*-adjacency \Rightarrow binary adjacency values
- spels with a common face (e.g., 4-connectivity in 2D, 6-connectivity in 3D) ... fully adjacent (*adjacency value* = 1)
- other spel pairs considered non-adjacent (*adjacency value* = 0)
- using hard six-adjacency in 3D ($n = 3$):

$$\mu(c, d) = \begin{cases} 1 & \text{if } c \text{ and } d \text{ are identical or differ in exactly one coordinate by } 1, \\ 0 & \text{otherwise.} \end{cases} \quad (7.12)$$

- n -dimensional fuzzy spel adjacency [Udupa and Samarasekera 96]

$$\mu(c, d) = \begin{cases} \frac{1}{1+k_1 \sqrt{\sum_{i=1}^n (c_i - d_i)^2}} & \text{if } \sum_{i=1}^n |c_i - d_i| \leq n, \\ 0 & \text{otherwise,} \end{cases} \quad (7.13)$$

k_1 is a nonnegative constant

- non-binary definitions possible with adjacency values ranging from 0 to 1
- affinity function $\psi(c, d)$ is only determined for *fuzzy adjacent* spels = with adjacency value $\mu(c, d) \neq 0$

- **Fuzzy connectedness** $\mu_\psi =$ *global* fuzzy relationship
- every pair of image elements c and d assigned a value $\in [0, 1]$ based on affinity values ψ along all possible paths between these two elements
- elements c and d not expected to be nearby
- c and d connected by path $\pi = \langle c^{(0)}, \dots, c^{(N)} \rangle$ of spels, with $c = c^{(0)}$ and $d = c^{(N)}$
- each pair of consecutive spels characterized by fuzzy affinity $\psi(c^{(n)}, c^{(n+1)})$, $0 \leq n \leq N-1$
- strength of each path = **minimum** affinity value of all pairwise consecutive elements on the path = strength of its weakest local connection

$$\psi'(\pi) = \min_{0 \leq n \leq N-1} \psi(c^{(n)}, c^{(n+1)}) . \quad (7.14)$$

- many different paths may connect two c and d ... M = number of all path joining c, d
- fuzzy connectedness defined as

$$\mu_{\psi}(c, d) = \max_{\pi \in M} \psi'(\pi), \quad (7.15)$$

value of fuzzy connectedness (global hanging togetherness) of c and d is determined as the **maximum** of the strengths of all possible paths between c and d

- strength of connectedness of all possible pairs of elements defining a fuzzy connected object = determined via dynamic programming [Udupa and Samarasekera

- Starting from a seed-spel c
 - determining the fuzzy connectedness $\mu_\psi(c, d_i)$ to every other spel d_i in the image domain C
 - assigning the corresponding connectedness value to every spel
- resulting image is a fuzzy **connectedness map** representing the degree of connectedness of every spel in the image with the seed-spel c
- Any degree of connectedness in the range $[0, 1]$ is possible
 - very strong connectedness denoted by 1
 - no connectedness ... 0
 - thresholding the connectedness map ... keeps only certain pre-determined minimum degree of connectedness to the seed-spel
- Thresholding the connectedness map yields segmentation result

Algorithm 7.4: Absolute fuzzy connectivity segmentation

1. Define properties of fuzzy adjacency and fuzzy affinity.
2. Determine the affinity values for all pairs of fuzzy adjacent spels.
3. Determine the segmentation seed element c .
4. Determine all possible paths between the seed c and all other image elements d_i in the image domain C (not forming loops) considering the fuzzy adjacency relationship.
5. For each path, determine its strength according as the minimum affinity along the path (equation 7.14).
6. For each image element d_j , determine its fuzzy connectedness $\mu_\psi(c, d_j)$ to the seed point c as the maximum strength of all possible paths $\langle c, \dots, d_j \rangle$ (equation 7.15) and form an image connectedness map.
7. Threshold the connectedness map with an appropriate threshold t to segment the image into an object containing the seed c and the background.

- performance of fuzzy connectivity segmentation depends on design of fuzzy affinity, computed using local image properties
- let *fuzzy affinity* $\psi(c, d)$ quantify the hanging-togetherness of two spels c and d
- fuzzy affinity $\psi(c, d)$ is function of fuzzy adjacency $\mu(c, d)$, spel properties $f(c)$, $f(d)$, and—in spatially variant cases—of c and d

$$\psi(c, d) = \frac{\mu(c, d)}{1 + k_2 |f(c) - f(d)|}, \quad (7.16)$$

μ is fuzzy adjacency defined by equation (7.13), k_2 is a nonnegative constant

- general affinity function [Udupa and Samarasekera 96]

$$\psi(c, d) = \begin{cases} \mu(c, d) \left(\omega h_1(f(c), f(d)) + (1 - \omega) h_2(f(c), f(d)) \right) & c \neq d, \\ 1 & \text{otherwise,} \end{cases} \quad (7.17)$$

ω is a weighting factor

h_1 and h_2 are task dependent and may be constructed using:

$$g_1(f(c), f(d)) = \exp \left(-\frac{1}{2} \left(\frac{\frac{1}{2}[f(c) + f(d)] - m_1}{\sigma_1} \right)^2 \right), \quad (7.18)$$

$$g_2(f(c), f(d)) = \exp \left(-\frac{1}{2} \left(\frac{|f(c) - f(d)| - m_2}{\sigma_2} \right)^2 \right), \quad (7.19)$$

$$g_3(f(c), f(d)) = 1 - g_1(f(c), f(d)), \quad (7.20)$$

$$g_4(f(c), f(d)) = 1 - g_2(f(c), f(d)), \quad (7.21)$$

m_1 and m_2 are mean values

σ_1 and σ_2 standard deviations reflecting properties of the object of interest

- m and σ values can be calculated from spels that are a priori known to belong to the object or background

- affinity function behavior can be influenced by choice of h_1 and h_2
- e.g., choosing $h_1(f(c), f(d)) = g_1(f(c), f(d))$, $\omega = 1$ favors spels that are closer to an expected mean value μ_1
- choosing $h_1(f(c), f(d)) = g_1(f(c), f(d))$, $h_2(f(c), f(d)) = g_4(f(c), f(d))$, and $\omega = 0.5$ decreases the affinity $\psi(c, d)$ if the gradient between the spels is close to the mean value μ_2
- fuzzy connectedness $\mu_\psi(c, d)$ for every spel of the image domain $d \in C$, $c \neq d$
- assigning the respective connectedness value to every spel
- \Rightarrow *connectedness map* – can subsequently be thresholded at any value (possibly in an interactive way)
- output values f_c express strength of connectivity between the seed-spel c and all other image spels $d \in C$

Algorithm 7.5: Fuzzy object extraction

1. Define a seed-point c in the input image.
2. Form a temporary queue Q and a real-valued array f_c with one element $f_c(d)$ for each spel d .
3. For all spels $d \in C$, initialize array $f_c(d) := 0$ if $d \neq c$; $f_c(d) := 1$ if $d = c$.
4. For all spels $d \in C$ for which fuzzy spel adjacency $\mu_\psi(c, d) > 0$, add spel d to queue Q .
5. While the queue Q is not empty, remove spel d from queue Q and perform the following operations:
 - $f_{\max} := \max_{e \in C} \min(f_c(e), \psi(d, e))$
 - if $f_{\max} > f_c(d)$ then
 - $f_c(d) := f_{\max}$
 - for all spels g for which $\psi(d, g) > 0$, add g to queue Q
 - endif
6. Once the queue Q is empty, the connectedness map (C, f_c) is obtained.

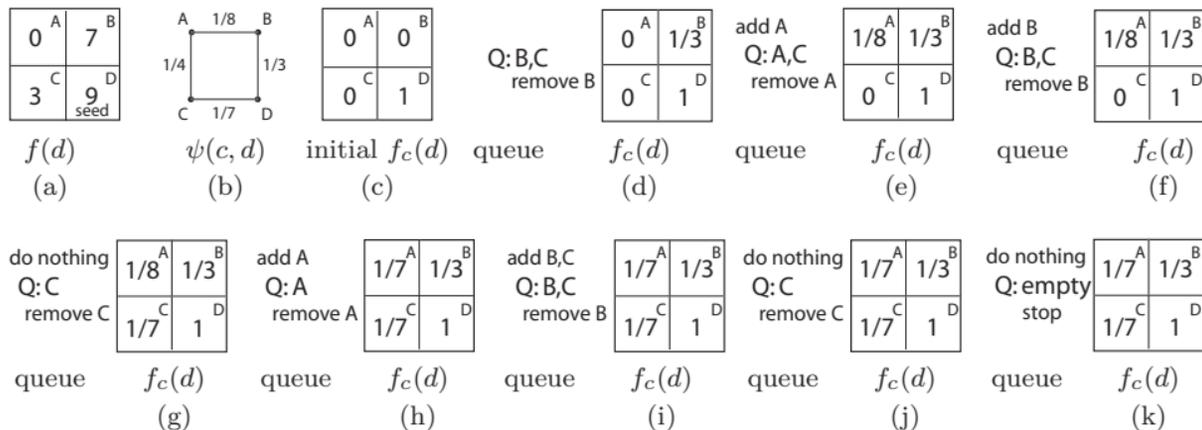


Figure 7.6: Fuzzy object extraction using Algorithm 7.5. (a) Image properties, which can be represented as image intensities. (b) Fuzzy affinity $\psi(c, d)$ calculated according to equation (7.16), $k_2 = 1$. (c) Initialized array $f_c(d)$. (d) Initial queue Q , temporary values of $f_c(d)$ after removal of spel B from queue Q . (e-j) Intermediate algorithm steps. (k) The queue Q is empty, stop. Values of array $f_c(d)$ represent the connectedness map.

- absolute fuzzy connectivity suffers from problems similar to traditional region growing algorithms [Jones and Metaxas 97]
- determining optimal threshold of connectivity map is difficult to automate
- **relative fuzzy connectivity** eliminates connectedness map thresholding step [Saha and Udupa 00b; Udupa et al. 99]
- instead of extracting a single object at a time as described above, two objects are extracted
- these two objects are competing against each other with each individual spel assigned to the object with a stronger affinity to this spel
- (simply using different affinities violates fundamental properties of fuzzy connectivity)
- affinities of different objects have to be combined into a single affinity by calculating fuzzy union of the individual affinities
- 2-object relative fuzzy connectivity was later refined to include **multiple objects** [Herman and Carvalho 01; Saha and Udupa 01; Udupa and Saha 01]
- extension to multiple object segmentation is a significant improvement compared to relative fuzzy connectivity.

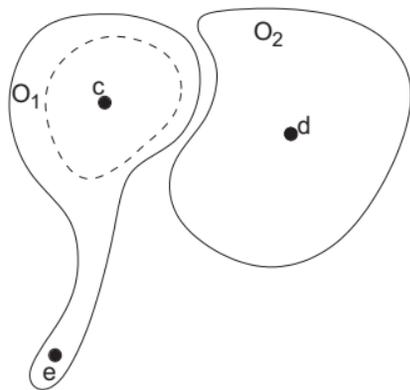


Figure 7.7: Segmentation task that will likely fail using fuzzy connectivity but can be solved by iterative fuzzy connectivity.

- if objects O_1 and O_2 are located very close to each other, border between them may be weak causing $\mu_{\psi}(d, e)$ to be of similar magnitude to $\mu_{\psi}(c, e) \rightarrow$ segmented as a single object
- **iterative fuzzy connectivity** [Saha and Udupa 00a; Udupa et al. 99] may help
- assume that optimal path between d and e passes through the core of O_1 (dashed line around c in Figure 7.7)
- this core can be segmented first - e.g., with a relative fuzzy connectivity algorithm
- paths for object O_2 between two spels not located in this core (like d and e) are not allowed to pass through the core of O_1
- objects are segmented in an iterative process
- the same affinity function must be used for all objects

- **Scale-based fuzzy connectivity** considers neighborhood properties of individual spels when calculating fuzzy affinity functions $\psi(c, d)$ [Saha and Udupa 99]
- $\psi(c, d)$ calculated in two hyperballs centered at c and d
- scale of calculation defined by radii of the hyperballs, which are derived from image content
- scale is adaptively varying location specific
- this approach generally leads to an improved segmentation, however with a considerable increase in computational cost

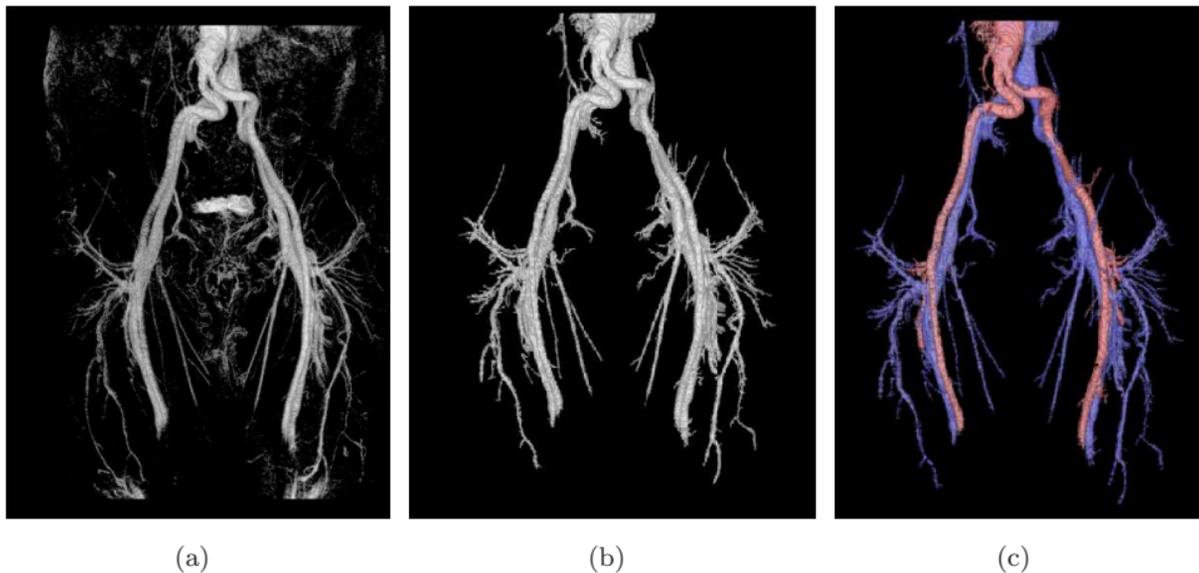
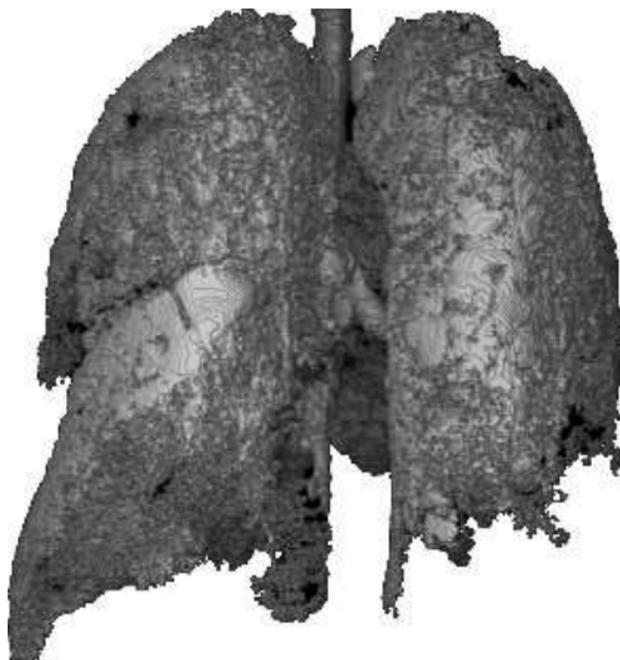
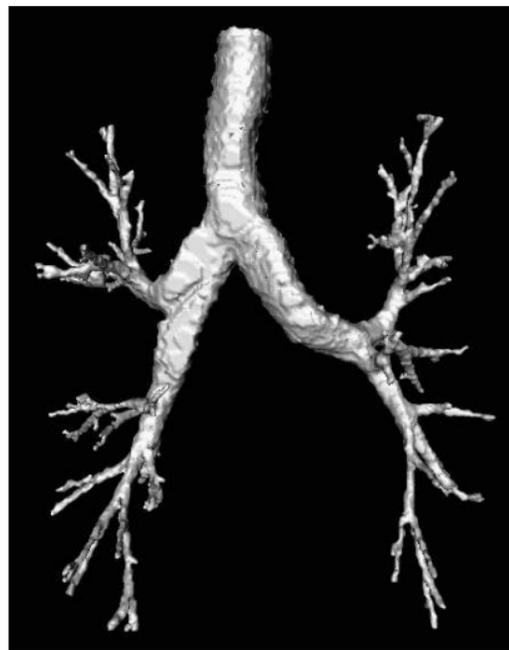


Figure 7.8: Segmentation and separation of vascular trees using fuzzy connectivity segmentation. (a) Maximum intensity projection image of the original magnetic resonance angiography data used for artery-vein segmentation in lower extremities. (b) Segmentation of the entire vessel tree using absolute fuzzy connectivity. (c) Artery-vein separation using relative fuzzy connectivity. *Courtesy of J. K. Udupa, University of Pennsylvania. A color version of this figure may be seen in the color inset—Plate 3.*



(a)



(b)

Figure 7.9: Segmentation result using multi-seeded fuzzy connectivity approach. (a) Region growing segmentation results in a severe segmentation leak. (Emphysema patient, segmented with standard 3D region growing algorithm—the leak was unavoidable). (b) Multi-seeded fuzzy connectivity succeeded with the image segmentation using a standard setting of the method.

7.3 Towards 3D graph-based image segmentation

- weighted graph $G = (V, E)$... *node set* V , *arc set* E
- nodes $v \in V$... image pixels (or voxels)
- arcs $\langle v_i, v_j \rangle \in E$ connect the nodes v_i, v_j according to some neighborhood system
- every node v and/or arc $\langle v_i, v_j \rangle \in E$ has a cost representing some measure of preference that the corresponding pixels belong to the object of interest
- constructed graph can be *directed* or *undirected*
- $\langle v_i, v_j \rangle$... node v_j is *successor* of v_i
- sequence of consecutive directed arcs $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \dots, \langle v_{k-1}, v_k \rangle =$ directed path (or *dipath*) from v_0 to v_k .

7.3.1 Simultaneous detection of border pairs

- for borders of elongated objects, searching for the pair of left and right borders **simultaneously** may take advantage of dual-border context
- more robust performance if the borders forming the border pair are interrelated
- information about one border helps identify the second
- advantageous if one border is locally noisy, ambiguous, or uncertain \Rightarrow identifying borders individually may fail

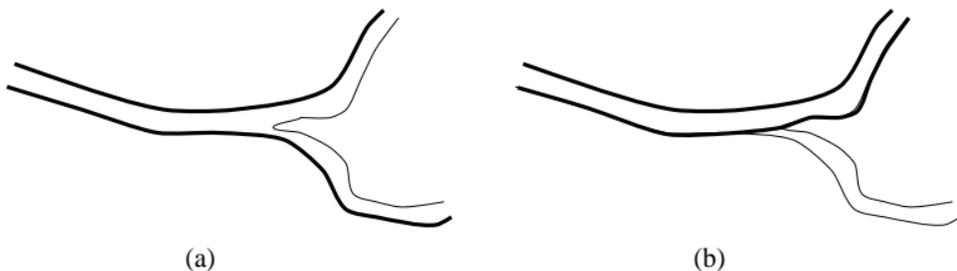


Figure 7.10: Individual and simultaneous border detection. (a) Individually identified borders may not be reasonable as a pair. (b) Simultaneously identified borders satisfy border-pair properties.

- search for optimal border pair requires a 3D graph

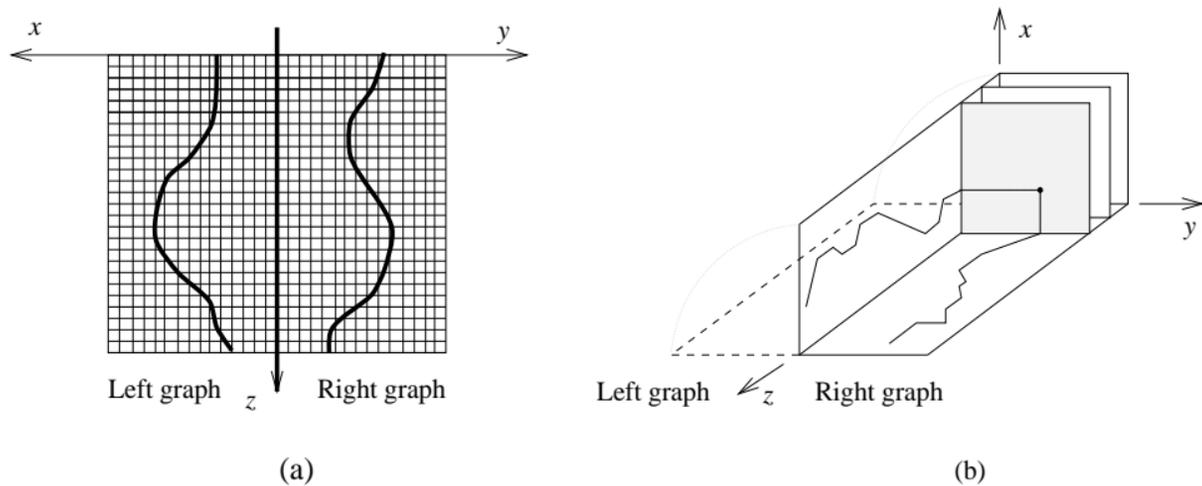


Figure 7.11: Three-dimensional graph construction. (a) Separate identification of the left and right borders by linking nodes in individual two-dimensional graphs corresponding to the left and right halves of the region segment of interest. (b) By rotating up the left graph, a three-dimensional graph results in which paths correspond to pairs of region borders.

- constructing 3-D graph ... rotating one of two 2D graphs
- 3-D graph in which nodes correspond to positions of left and right borders for a given point
- path through the graph corresponds to a pair of left and right borders
- nodes in 3-D graph (x, y, z)
- node successor rule – nine successors forming a 3×3 successor window
- all paths through the 3-D graph contain one and only one node from each **profile plane** in the 3D graph
- every path contains a single node derived from each of the left and right profile lines
- region borders are continuous in the straightened image space

Cost function design

- assignment of costs to pairs of candidate borders
- identification of optimal pair of region borders as lowest-cost path in the 3D graph
- node cost function derived by combining left/right edge costs that allows position of left border to influence position of right border and vice versa
- costs discriminate against border pairs that are unlikely to correspond to the true region borders
- costs prefer border pairs that have the greatest overall probability of matching the actual borders
- heuristic graph searching or dynamic programming methods can be used for optimal border detection
- path cost defined as sum of costs of nodes forming the path
- many different cost functions can be designed

Shape model

- $w(x, y, z)$ incorporates model so that positions of left and right borders follow certain preferred directions relative to the model
- discriminate against unlikely borders when considered as a pair
- include weighting factor that depends on the direction by which a node is reached from its predecessor

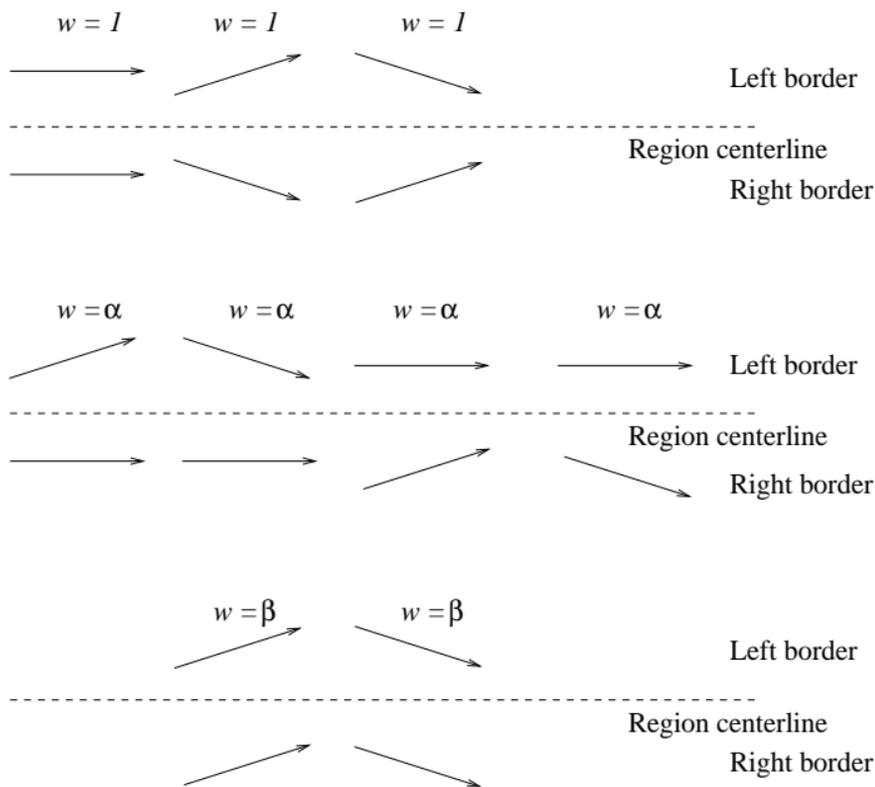


Figure 7.12: The weighting factors $w(x, y, z)$ associated with local directions of the potential border elements for a symmetric region model.

$$\begin{aligned}w(x, y, z) &= 1 && \text{for } (x, y) \in \{(\hat{x} - 1, \hat{y} - 1), (\hat{x}, \hat{y}), (\hat{x} + 1, \hat{y} + 1)\}, \\w(x, y, z) &= \alpha && \text{for } (x, y) \in \{(\hat{x} - 1, \hat{y}), (\hat{x} + 1, \hat{y}), (\hat{x}, \hat{y} - 1), (\hat{x}, \hat{y} + 1)\}, \\w(x, y, z) &= \beta && \text{for } (x, y) \in \{(\hat{x} - 1, \hat{y} + 1), (\hat{x} + 1, \hat{y} - 1)\},\end{aligned}\quad (7.22)$$

(x, y, z) is successor of $(\hat{x}, \hat{y}, z - 1)$

- influence of region model determined by $\alpha, \beta, \alpha > \beta$
- the larger α and β , the stronger the model's influence

7.3.2 Sub-optimal surface detection

- intuitively obvious that a set of 2D borders that were detected in individual slices may be far from optimal if the entire 3D volume is considered
- concurrent analysis of the entire 3D volume may give better results if a globally optimal surface is determined

Cost function

- surface cost calculated as sum of individual costs of all nodes forming the surface
- shall be possible to determine the optimal surface by application of optimal graph searching principles similar to those presented earlier
- standard graph searching approaches cannot be directly extended from a search for a path to a search for a **surface** [Thedens et al. 95] due to exponential increase of computational complexity
- cost minimization in a graph given in [Thedens et al. 90; Thedens et al. 95] used standard graph searching principles applied to a transformed graph
- guaranteed surface optimality, was impractical due to enormous computational requirements
- heuristic approach to surface detection was computationally feasible [Thedens et al.

Sub-optimal surface detection

- introduced in [Frank 96; Frank et al. 96]
- based on dynamic programming
- avoids the problem of combinatorial explosion by introducing local conditions that must be satisfied by all legal surfaces
- graph size corresponds directly to the image size
- resulting surfaces typically represent good solutions, surface optimality is not guaranteed

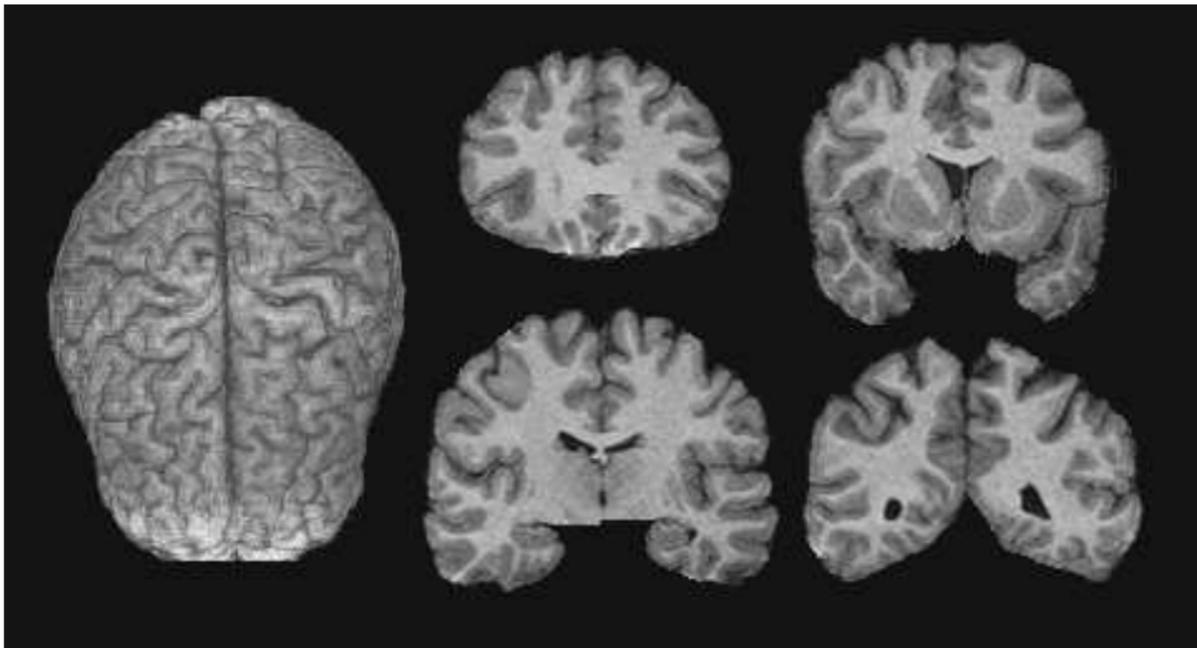
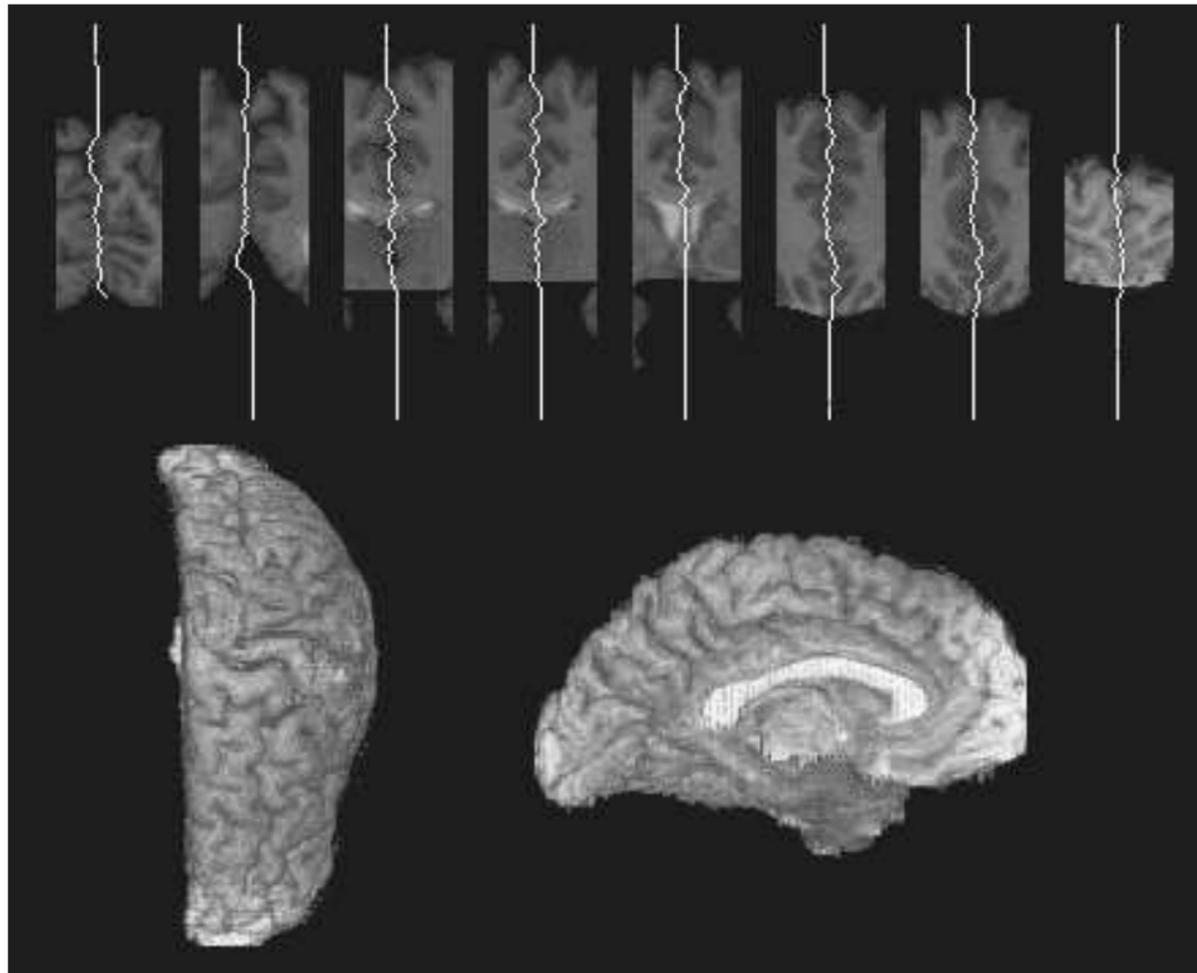


Figure 7.13: Magnetic resonance images of human brain. Left: Three-dimensional surface rendering of original MR image data after segmentation of the brain from the skull. Right: Four of 120 two-dimensional slices that form the three-dimensional image volume. *Courtesy of R. J. Frank and H. Damasio, The University of Iowa.*

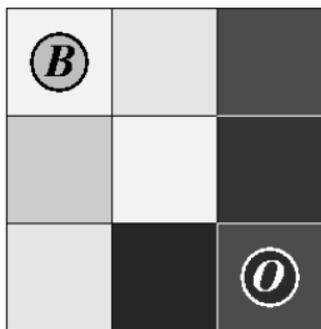


7.4 Graph cut segmentation

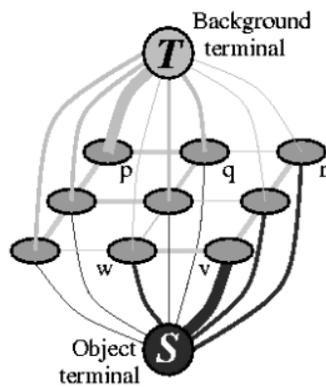
- optimal boundary and region segmentation in n-D image data [Boykov and Jolly 0] (<http://www.csd.uwo.ca/~yuri/>)
- initiated by interactive or automated identification of one or more points representing the ‘object’ and one or more points representing the ‘background’
- **seeds = hard constraints**
- **soft constraints** = boundary and/or region information
- segmentation solution is globally optimal with respect to an objective function
- cost function C (based on **Gibbs model**) [Geman and Geman 84]

$$C(f) = C_{\text{data}}(f) + C_{\text{smooth}}(f) \quad (7.23)$$

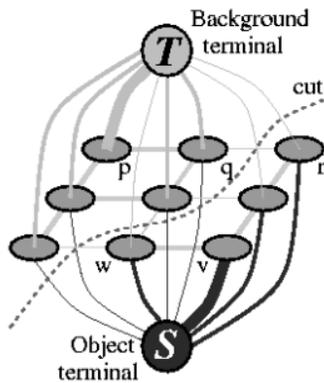
- $C(f)$ minimized in arc-weighted graphs $G_{st} = (V \cup \{s, t\}, E)$
- nodes $V \Rightarrow$ pixels (voxels) of image I
- node set G_{st} contains two *terminal* nodes
 - *source* s
 - *sink* t
- terminals hard-linked with segmentation seed points and represent segmentation labels (object, background)
- arcs E in G_{st}
 - *n-links* — connect pixels and terminals with costs derived from the data term $C_{\text{data}}(f)$
 - *t-links* — connect pairs of neighboring pixels whose costs are derived from the smoothness term $C_{\text{smooth}}(f)$



(a)



(b)



(c)



(d)

- s - t cut in G_{st} = set of arcs whose removal partitions nodes into two disjoint subsets S and T
 - such that $s \in S$ (all nodes linked to source) and $t \in T$ (all nodes linked to sink) and no directed path can be established from s to t
- cost of a cut is the total cost of arcs in the cut
- minimum s - t cut is a cut with minimal cost
- **minimum s - t cut** problem and its dual, **maximum flow** problem, are classic combinatorial problems that can be solved by various polynomial-time algorithms [Ford and Fulkerson 56; Goldberg and Tarjan 88; Goldberg and Rao 98]
- O, B – sets of image pixels corresponding to object and background seeds
- $O \subset V, B \subset V, O \cap B = \emptyset$
- seeds form hard t -links

- graph cut determined to form the object(s) and background from image pixels so that
 - all object pixels are connected to object seed terminal
 - all background pixels are connected to background seed terminal
 - accomplished by searching for a graph cut that minimizes a cost function (eq. 7.23)
 - = weighted combination of regional and boundary properties of the object with respect to the background
- all directed pairs of pixels (p, q) , $p, q \in I$ representing neighborhood pixel relationships ... N
 - 2D image example – rectangular 2D grid with 4- or 8-neighborhood connectivity links in N
 - 3D case – image voxels 3D grid (e.g., 26-connected) ... N
 - extendible to n -D
- cost of (p, q) may differ from cost of (q, p)
- \Rightarrow asymmetric neighborhood relationships possible

- each image pixel i_k ... binary label $L_k \in \{obj, bgd\}$
- labeling vector $\mathbf{L} = (L_1, L_2, \dots, L_{|I|})$ defines resulting binary segmentation
- cost function C minimized to achieve optimal labeling — may include regional terms $R(\mathbf{L})$ and boundary property terms $B(\mathbf{L})$

$$C(\mathbf{L}) = \lambda R(\mathbf{L}) + B(\mathbf{L}), \quad (7.24)$$

where

$$R(\mathbf{L}) = \sum_{p \in I} R_p(L_p), \quad (7.25)$$

$$B(\mathbf{L}) = \sum_{(p,q) \in N} B_{(p,q)} \delta(L_p, L_q), \quad (7.26)$$

and

$$\delta(L_p, L_q) = \begin{cases} 1 & \text{if } L_p \neq L_q, \\ 0 & \text{otherwise.} \end{cases}$$

- $R_p(obj)$ — cost of labeling pixel p as *object*
- $R_p(bgd)$ — cost of labeling the same pixel p as *background*
 - e.g., bright objects on a dark background
 - $R_p(obj)$ large in dark pixels (low I_p values)

- $R_p(obj)$ small in bright pixels
- $B_{(p,q)}$ cost – associated with local labeling discontinuity between neighboring pixels p, q
 - $B_{(p,q)}$ – large for both p and q belonging to either object or background
 - $B_{(p,q)}$ – small if one of p, q belongs to object and the other to background = across object/background boundaries
 - $\Rightarrow B_{(p,q)}$ – e.g., inverted image gradient magnitude between p and q
- complete graph ... n -links and t -links
- Table 7.1 – graph arc weight assignment

Table 7.1: Cost terms for Graph Cut segmentation. K may be interpreted as the maximum needed flow capacity of the arc from source s to $p \in O$ (or from $p \in B$ to sink t), increased by one so that the arc gets never saturated; $K = 1 + \max_{p \in I} \sum_{q: (p,q) \in N} B_{(p,q)}$.

Graph arc	Cost	
(p, q)	$B_{(p,q)}$	for $(p, q) \in N$
(s, p)	$\lambda R_p(bgd)$	for $p \in I, p \notin (O \cup B)$
	K	for $p \in O$
	0	for $p \in B$
(p, t)	$\lambda R_p(obj)$	for $p \in I, p \notin (O \cup B)$
	0	for $p \in O$
	K	for $p \in B$

- solution of minimum s - t cut problem = finding maximum flow from source s to sink t
- many algorithms exist
- most existing algorithms can be categorized in two groups
 - **push-relabel** methods [Goldberg and Tarjan 88]
 - **augmenting path** methods [Ford and Fulkerson 62]

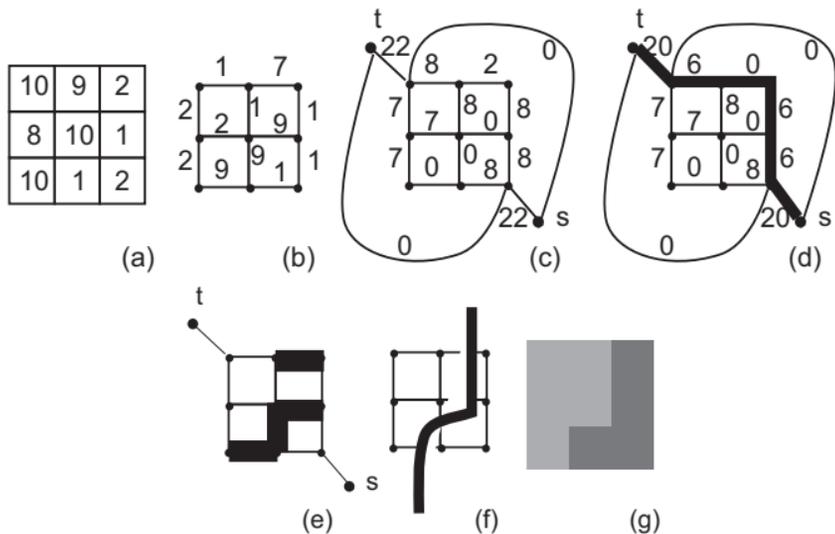


Figure 7.16: Image segmentation using graph cuts and maximum flow optimization. (a) Original image data—corresponding to Figure 7.15a. (b) Edge magnitudes calculated as image intensity differences in 4-connectivity. (c) G_{st} graph constructed according to Table 7.1; $\lambda = 0$; n -link costs use, e.g., edge-based costs; reverse path residual capacities are not shown. (d) Residual graph G_f after the one and only shortest path with non-saturated $s \rightarrow t$ connection was identified and saturated. No new non-saturated $s \rightarrow t$ path can be found. (e) Saturated graph arcs identified by thick black lines. (f) Resulting minimum $s-t$ cut separating S and T nodes. (g) Corresponding image segmentation.

Algorithm 7.6: Graph cut segmentation

1. Create an arc-weighted directed graph corresponding in size and dimensionality to the image to be segmented.
2. Identify object and background seeds—example points required to be part of the background or object(s) in the final segmentation. Create two special graph nodes—source s and sink t ; connect all seeds with either the source or the sink node based on their object or background label.
3. Associate appropriate arc cost with each link of the formed graph according to Table 7.1.
4. Use one of the available maximum flow graph optimization algorithms to determine the graph cut.
5. The minimum s – t cut solution identifies the graph nodes that correspond to the image boundaries separating the object(s) and the background.

DEMO - GRAPH CUT SEGMENTATION - MOVIE

- \Rightarrow ability to interactively improve previous segmentation efficiently
- segmentation can be improved by adding supplemental object or background seeds
 - possible to recompute graph cut segmentation from scratch
 - efficient way does not require restarting
 - previous status of the graph optimization initializes next graph cut optimization process
- How to do that?
 - algorithmic $s - t$ cut solution is characterized by saturation of graph by maximum flow
 - adding new object seed p requires forming corresponding hard t -links according to Table 7.1:
 - * weight of $(s, p) \rightarrow K$
 - * weight $(p, t) \rightarrow 0$
 - * may lead to negative capacities in residual network of current flow
 - * compensated for by increasing values c_p of t -links as specified in Table 7.2

Table 7.2: Cost term $c_p = \lambda(R_p(bgd) + R_p(obj))$ modification for sequential improvement of graph cut segmentation after adding object seed p .

t -link	initial cost	added cost	new cost
(s, p)	$\lambda R_p(bgd)$	$K + \lambda R_p(obj)$	$K + c_p$
(p, t)	$\lambda R_p(obj)$	$\lambda R_p(bgd)$	c_p

- new costs remain consistent with costs of pixels in O — additional constant c_p appears at both t -links — does not change the optimal cut
- \Rightarrow new optimal cut can be efficiently obtained starting from the previous flow solution without starting from scratch
- the same approach can be used if new background seed is added — cost constants added to new t -links must be consistent with cost table and need to be modified by the same constant

Cost Function

- influences method's performance in real-world applications
- e.g., seeds (object and background) may consist of small patches — used to sample the object and background image properties
 - e.g., calculating histograms of object and background patches
 - $P(I|O)$ and $P(I|B)$ — probabilities of particular gray level belonging to object or background can be derived from patch histograms
 - more complex probabilities possible

- regional R_p and boundary $B(p, q)$ costs [Boykov and Jolly 01]

$$\begin{aligned}R_p(obj) &= -\ln P(I_p|O), \\R_p(bgd) &= -\ln P(I_p|B), \\B(p, q) &= \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \frac{1}{\|p, q\|},\end{aligned}\tag{7.27}$$

$\|p, q\|$... distance between pixels p, q

- cost $B(p, q)$ high for small differences between image values $|I_p - I_q| < \sigma$ (within object or background)
- cost $B(p, q)$ low for boundary locations where $|I_p - I_q| > \sigma$
- σ — allowed or expected intensity variation within the object and/or background

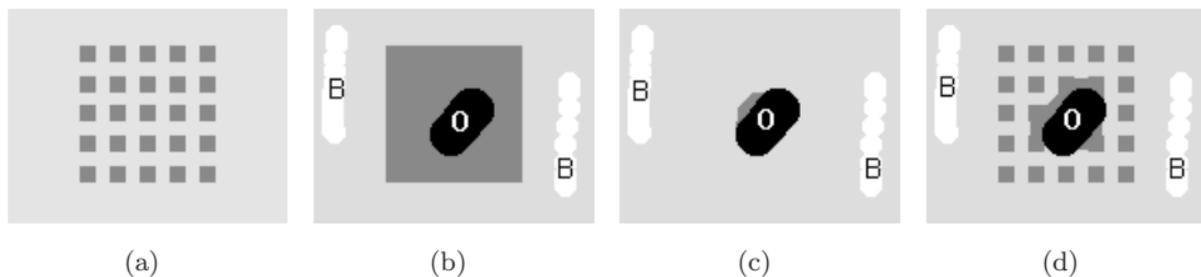
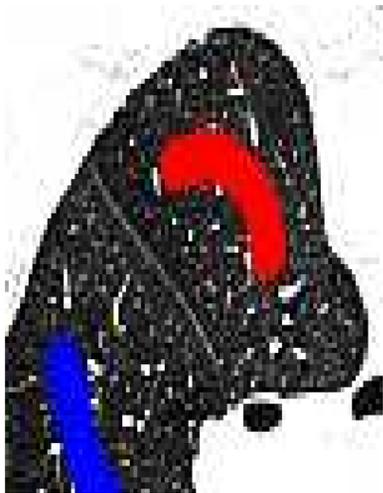


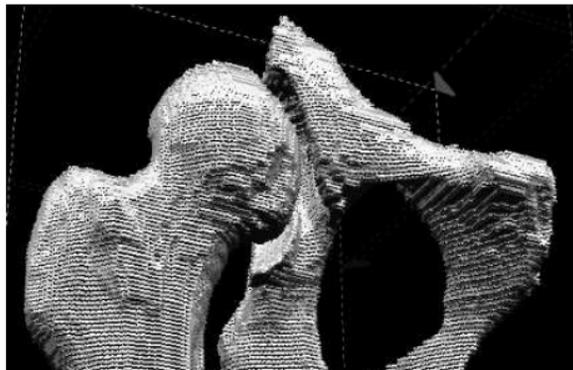
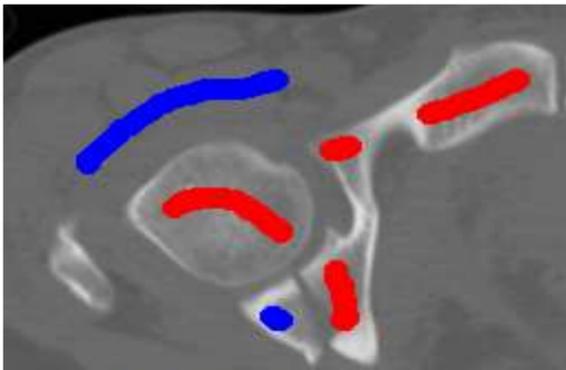
Figure 7.17: Graph cut segmentation behavior on a synthetic image. In all cases, the segmentation was initialized using the object patch as marked in black and background patch marked in white. The resulting segmentation is shown in light gray (background) and dark gray (objects). The initialization patches are parts of the segmented object(s) or background. (a) Original image. (b) Segmentation result for $\lambda \in [7, 43]$, i.e., only using a wide weighting range of region and boundary cost terms. (c) Segmentation result for $\lambda = 0$, i.e., only using the boundary cost term. (d) Segmentation result for $\lambda = 60$, i.e., using almost solely the region cost term. Notice the ability of the method to change the topology of the segmentation result. *Courtesy of Y. Boykov, University of Western Ontario, ©2001 IEEE [Boykov and Jolly 01].*



(a)



(b)



- applications include stereo problems
 - disparity maps interpreted as separating hypersurfaces
- multi-view image stitching
- video texture synthesis
- image reconstruction
- n -dimensional image segmentation

Boykov and Jolly's graph cut

- flexible
- shares elegance of level set methods
- similar to level sets, results are topology-unconstrained and may be sensitive to initial seed point selections unless a priori shape knowledge about the objects is incorporated
- while graph cut approach provides inherently binary segmentation, it can be extended to multi-label segmentation
- unfortunately, multi-way cut problem is NP -complete and α -expansion algorithm may be used to obtain a good approximate solution [Boykov et al. 01]
- development ongoing
 - combination of graph cuts and geodesic active contours [Boykov and Kolmogorov 01]
 - connection between discrete graph cut algorithms and global optimization of a wide class of continuous surface functionals [Kolmogorov and Boykov 05]
 - associations between level set and graph cut approaches [Boykov and Kolmogorov 01; Boykov and Funka-Lea 06; Boykov et al. 06]
- experimental comparison of performance of several min-cut / max-flow algorithms for energy minimization in vision applications [Boykov et al. 01]

7.5 References

- Boykov and Funka-Lea 06] Y Boykov and G Funka-Lea. Graph-cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70:109–131, 2006.
- Boykov and Jolly 01] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. International Conference on Computer Vision (ICCV)*, volume 1935-I, pages 105–112, July 2001.
- Boykov and Kolmogorov 03] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. International Conference on Computer Vision (ICCV)*, pages 26–33, Nice, France, October 2003.
- Boykov et al. 01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- Boykov et al. 06] Yuri Boykov, Vladimir Kolmogorov, Daniel Cremers, and Andrew Delong. An integral solution to surface evolution PDEs via geo-cuts. In *European Conference on Computer Vision (ECCV)*, pages 409–422, Springer, Graz, Austria, 2006.
- Christoudias et al. 02] M Christoudias, B Georgescu, and P Meer. Synergism in low level vision. In *Proc. ICPR*, pages 150–155, Quebec City, Canada. Code available at <http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html>, 2002.
- Comaniciu and Meer 02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

- [Ford and Fulkerson 56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
- [Ford and Fulkerson 62] L R Ford and D R Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.
- [Frank 96] R. J. Frank. Optimal surface detection using multi-dimensional graph search: Applications to Intravascular Ultrasound. Master's thesis, University of Iowa, Iowa City, IA, USA, 1996.
- [Frank et al. 96] R J Frank, D D McPherson, K B Chandran, and E L Dove. Optimal surface detection in intravascular ultrasound using multi-dimensional graph search. In *Computers in Cardiology*, pages 45–48, IEEE, Los Alamitos, CA, 1996.
- [Geman and Geman 84] S Geman and D Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [Goldberg and Rao 98] Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45:783–797, 1998.
- [Goldberg and Tarjan 88] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35:921–940, 1988.
- [Herman and Carvalho 01] Gabor T. Herman and Bruno M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):460–474, 2001.
- [Jones and Metaxas 97] Timothy N. Jones and Dimitris N. Metaxas. Automated 3D Segmentation Using Deformable Models and Fuzzy Affinity. In *Information Processing in Medical Imaging Conference (IPMI)*, pages 113–126, 1997.

- [Kolmogorov and Boykov 05] V Kolmogorov and Y Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *International Conference on Computer Vision (ICCV)*, vol. I, pages 564–571, Springer, Beijing, China, 2005.
- [Saha and Udupa 99] Punam K. Saha and Jayaram K. Udupa. Scale-based fuzzy connectivity: a novel image segmentation methodology and its validation. In *SPIE Conference on Image Processing, San Diego, California*, pages 246–257, 2 1999.
- [Saha and Udupa 00a] Punam K. Saha and Jayaram K. Udupa. Iterative relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '00)*, pages 254–269, 2000.
- [Saha and Udupa 00b] Punam K. Saha and Jayaram K. Udupa. Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation. *Computer Vision and Image Understanding*, 77:145–174, 2000.
- [Saha and Udupa 01] Punam K. Saha and Jayaram K. Udupa. Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82(1):42–56, 2001.
- [Thedens et al. 90] D R Thedens, D J Skorton, and S R Fleagle. A three-dimensional graph searching technique for cardiac border detection in sequential images and its application to magnetic resonance image data. In *Computers in Cardiology*, pages 57–60, IEEE, Los Alamitos, CA, 1990.

- Thedens et al. 95] D R Thedens, D J Skorton, and S R Fleagle. Methods of graph searching for border detection in image sequences with application to cardiac magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 14:42–55, 1995.
- Udupa and Saha 01] Jayaram K. Udupa and Punam K. Saha. Multi-Object Relative Fuzzy Connectedness and its Implications in Image Segmentation. In *Medical Imaging 2001: Image Processing; Proceedings of SPIE, Vol. 4322*, pages 204–213, 2001.
- Udupa and Samarasekera 96] J K Udupa and S Samarasekera. Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.
- Udupa et al. 99] Jayaram K. Udupa, Punam K. Saha, and Roberto A. Lotufo. Fuzzy connected object definition in images with respect to co-objects. In *SPIE Conference on Image Processing, San Diego, California*, pages 236–245, 1999.