上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Institute of Media, Information, and Network

生命科学技术学院
School of Life Sciences and Biotechnology

边缘检测与图像分割

Wenrui Dai
戴文睿
http://min.sjtu.edu.cn

Department of Computer Science & Engineering
Shanghai Jiao Tong University

April 8, 2020

Today's Topic

- Edge Detection

- Image Segmentation

Introduction & Overview

- Purpose
  - In medical image processing, as well as many other applications of image processing, it is necessary to identify the boundary between the objects in the image and separate the objects from each other.
- Approaches
  - Differences and dissimilarities of pixels in different regions
  - Similarities of the pixels within each region

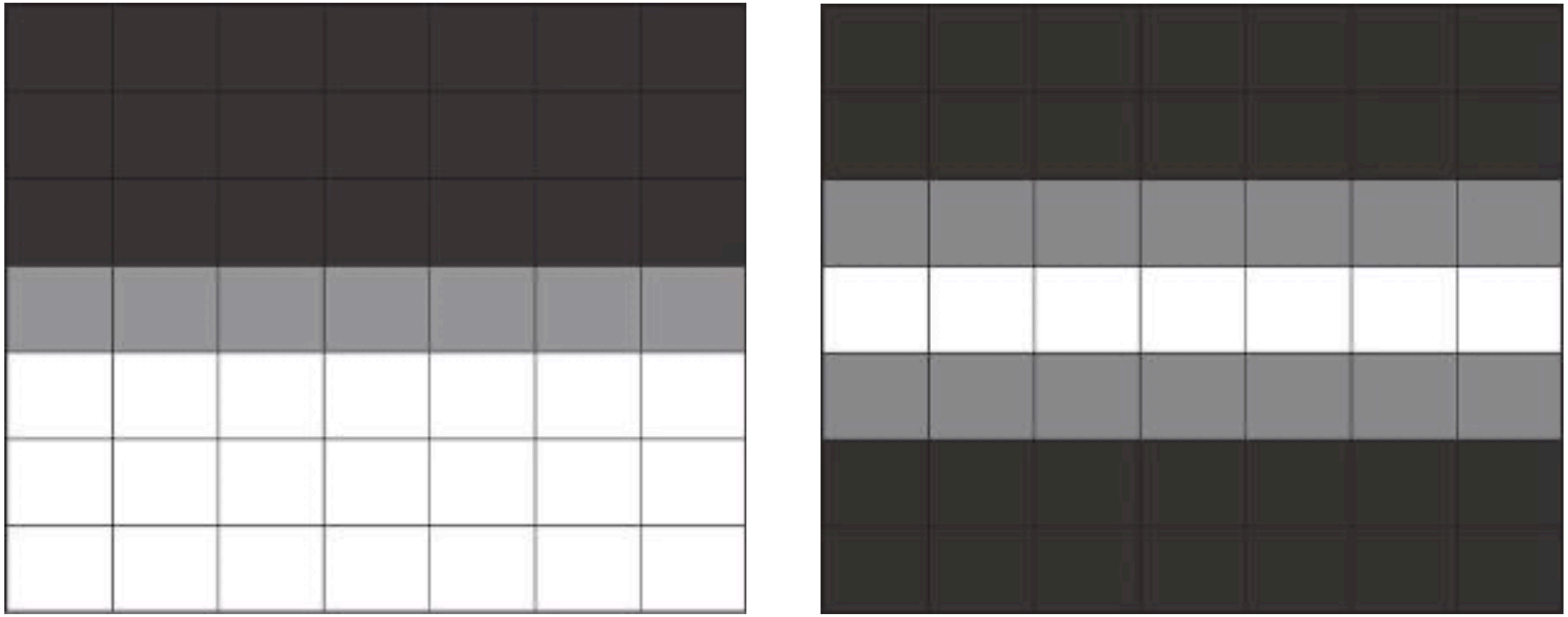
Edge Detection

- Sobel Edge Detection

$$S_H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad S_V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Edge Detection

- Example:

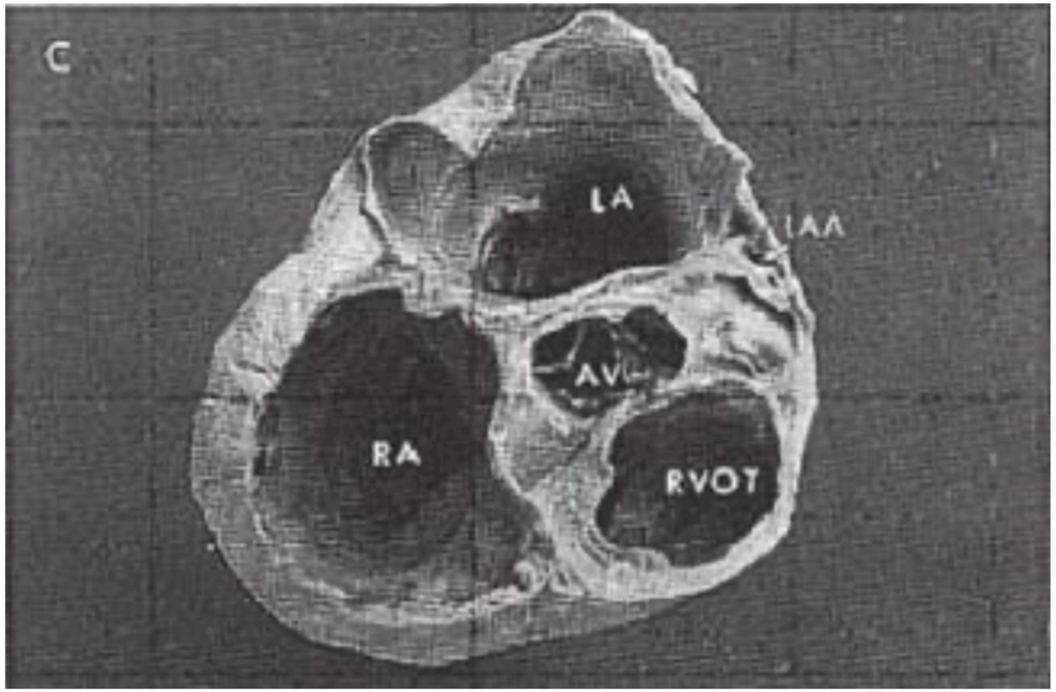


(a) (b)

(a) Original image and (b) edge-enhanced image

Edge Detection

- Example:



(a)

(b)

(a) Original image and (b) edge-detected image (Courtesy of Andre D'Avila, MD, Heart Institute (InCor), University of Sao Paulo, Medical School, Sao Paulo, Brazil)

Edge Detection

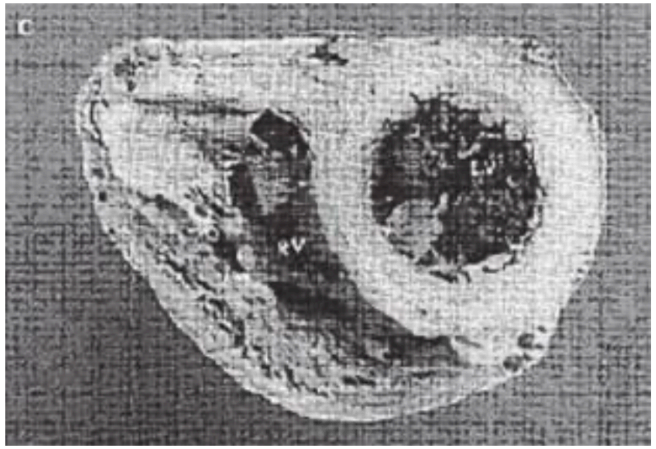
- Laplacian of Gaussian (LoG) Edge Detection
  - This edge detection technique, as the name may suggest, is a straightforward combination of a Laplacian operator and a Gaussian smoothing filter.

$$f(x, y) \longrightarrow \boxed{h(x, y)} \longrightarrow \boxed{\nabla^2} \longrightarrow g(x, y)$$

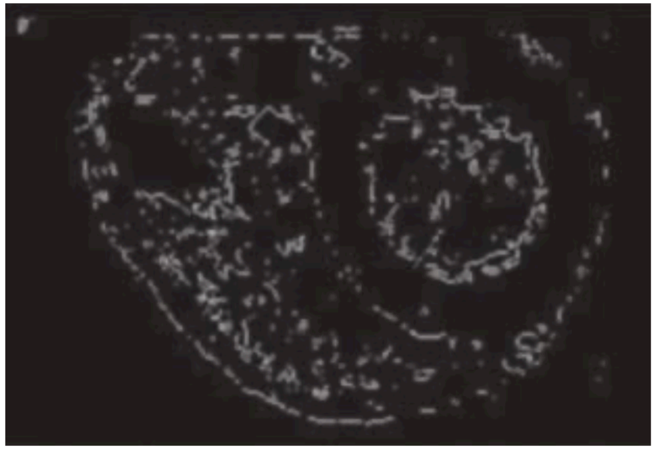
$$\nabla^2 f = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \qquad h(x, y) = \exp\left(-\frac{x^2 + y^2}{2s^2}\right)$$

Schematic diagram of edge detection using Laplacian of Gaussian

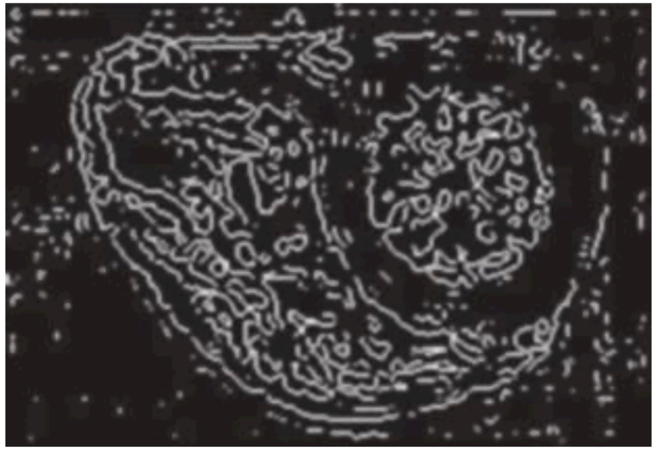
Example



(a)



(b)



(c)

(a) Original image,
(b) edge-detected image using Sobel, and
(c) edge-detected image using Laplacian of Gaussian
(Courtesy of Andre D'Avila, MD, Heart Institute (InCor), University of Sao Paulo, Medical School, Sao Paulo, Brazil)

Canny Edge Detection

- Four fundamental steps:

Step 1: The image is smoothed using a Gaussian filter (as defined earlier).

Step 2: The gradient magnitude and orientation are computed using finite-difference approximations for the partial derivatives (as discussed in the following).

Step 3: Non-maxima suppression is applied to the gradient magnitude to search for pixels that can identify the existence of an edge.

Step 4: A double thresholding algorithm is used to detect significant edges and link these edges.

Canny Edge Detection

- Details of the aforementioned steps:
  - Step 1

$$S(i,j) = G(i,j) * I(i,j)$$

$I(i,j)$: the input image
$G(i,j,\sigma)$: Gaussian smoothing filter where σ is the spread of the Gaussian controlling the degree of smoothing.
$S(i,j)$: the output of the smoothing filter.

Canny Edge Detection

- Details of the aforementioned steps:
  - Step 2

$$P(i,j) \approx \frac{S(i,j+1) - S(i,j) + S(i+1,j+1) - S(i+1,j)}{2}$$

$$Q(i,j) \approx \frac{S(i,j) - S(i+1,j) + S(i,j+1) - S(i+1,j+1)}{2}$$

$P(i,j)$ and $Q(i,j)$ are the horizontal and vertical partial (directional) derivatives, respectively. They are produced with the gradient of the smoothed image.

The magnitude $M(i,j)$ and orientation $q(i,j)$ of the gradient vector are given as follows:

$$M(i,j) = \sqrt{P(i,j)^2 + Q(i,j)^2} \qquad q(i,j) = \tan^{-1}\left[\frac{Q(i,j)}{P(i,j)}\right]$$

Canny Edge Detection

- Details of the aforementioned steps:
  - Step 3

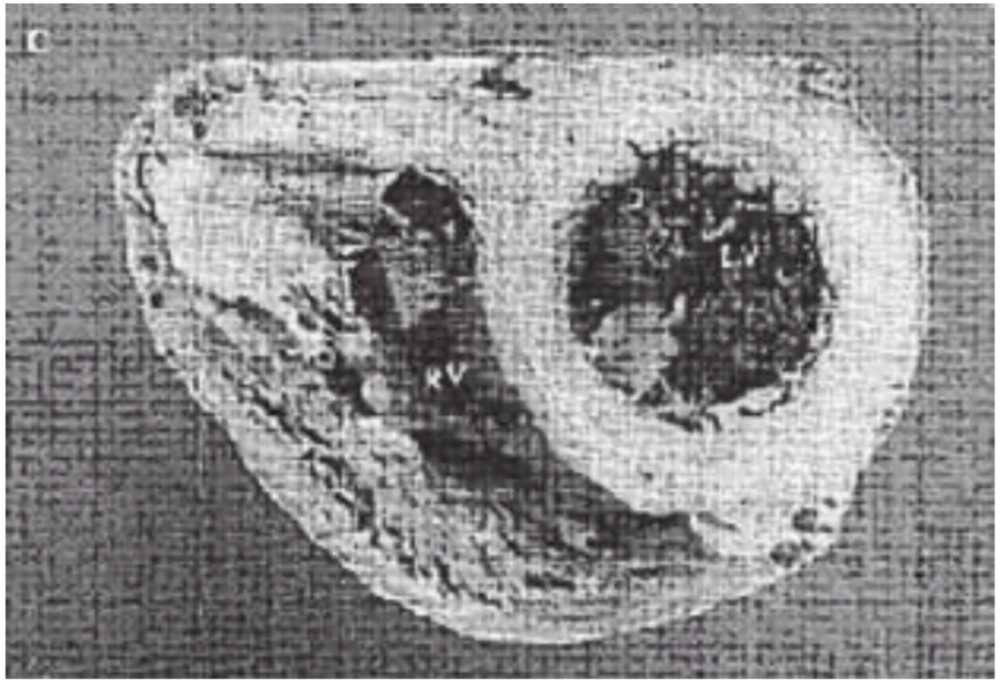
    In Canny method, an edge point is defined as a point whose gradient's magnitude identifies a local maximum in the direction of the gradient.The process of searching for such pixels, which is often called "non-maxima suppression," thresholds the gradient magnitude to find potential edge pixels.

  - Step 4

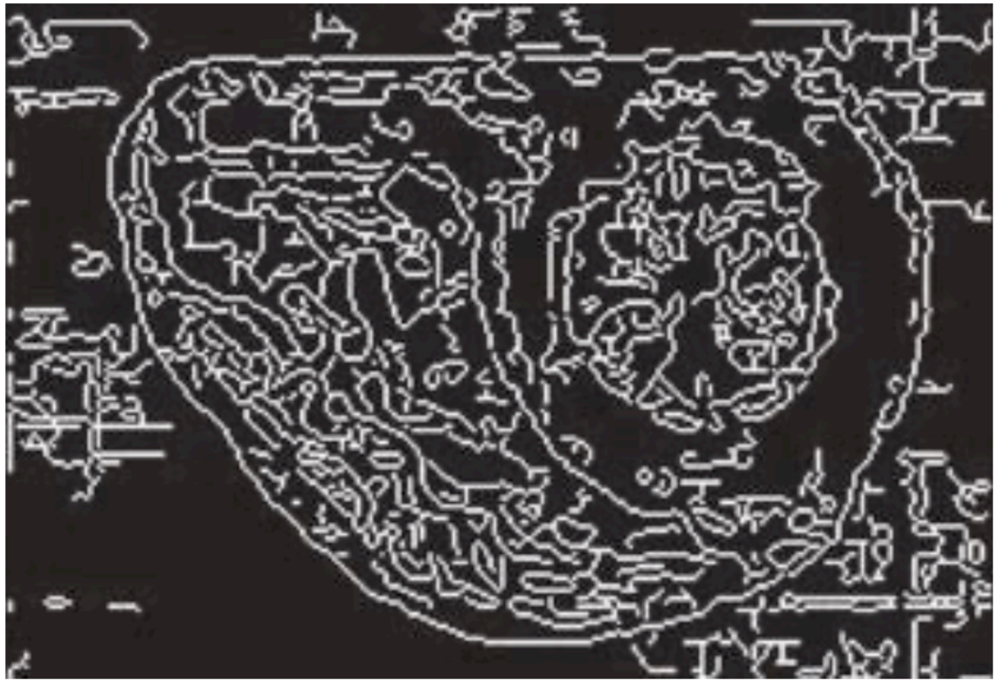
    After applying non-maxima suppression, there are often many false edge fragments in the image. To discard these false edge fragments, one can apply thresholds to $N(i,j)$ to discard false edge fragments.

Canny Edge Detection

- Examples:



(a)



(b)

(a) Original image and (b) Canny edge-detected image (Courtesy of Andre D'Avila, MD, Heart Institute (InCor), University of Sao Paulo, Medical School, Sao Paulo, Brazil).

Image Segmentation

- In almost all biomedical image processing applications, it is necessary to separate different regions and objects in an image. In fact, image segmentation is considered as the most sensitive step in many medial image processing applications.

Image Segmentation

- Categories
  - Based on the discontinuity of the points across two regions.
    - Detecting gray-level discontinuities (points, lines and edges)
    - Thresholding
  - Based on the similarities among the points in the same region.

Image Segmentation

- Point Detection
  - Intention: To detect the isolated points in an image.
  - The main factor that helps us detect isolated points is the difference between them and their neighboring pixels gray levels. This suggests using masks.
  - Supposes that the value obtained by applying mask is F, then a point is marked as an isolated point if $F \geq T$ with the prespecified threshold T.

Image Segmentation

- Point Detection
  - Example

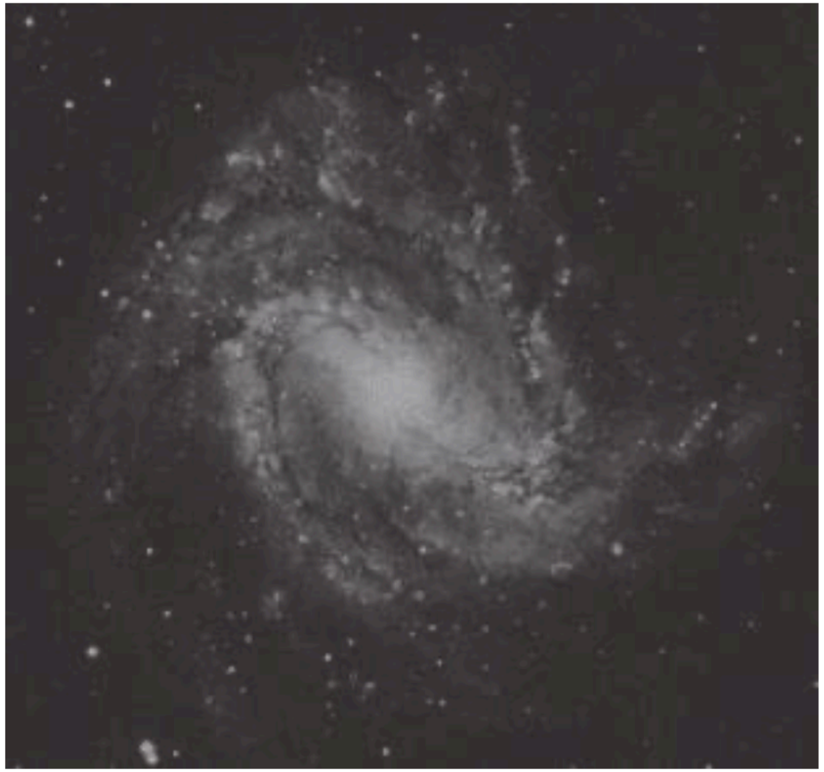
–1	–1	–1
–1	8	–1
–1	–1	–1

Mask for point detection (Courtesy of David Malin Images, Anglo-Australian Observatory [AAO], Epping, New South Wales, Australia http://www davidmalin com)

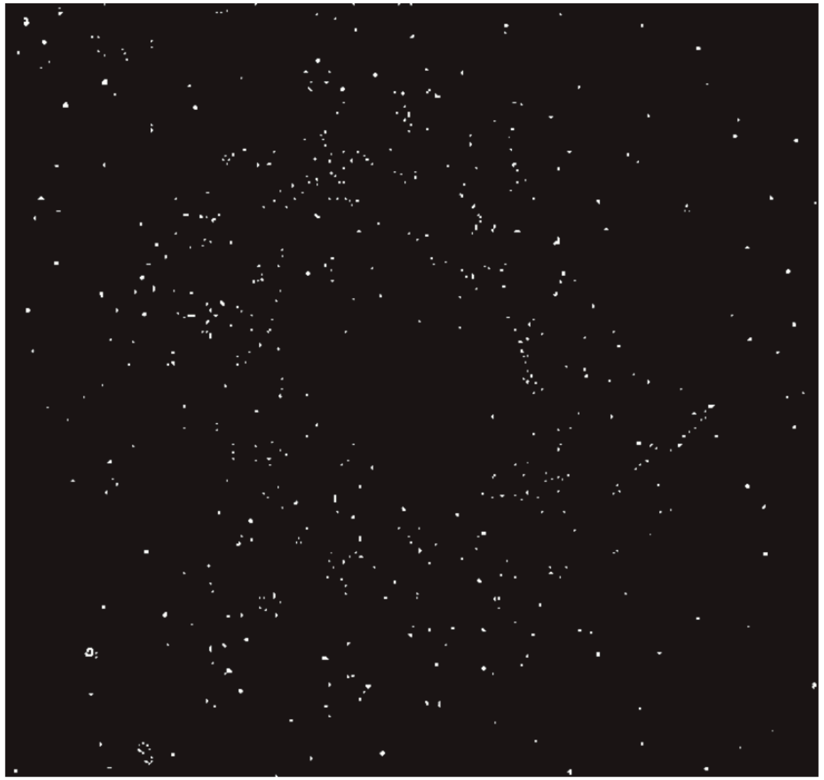
Image Segmentation

- Point Detection
  - Example

(a) Original image and
(b) image after point detection



(a)



(b)

Image Segmentation

- Line Detection
  - Magnifying and detecting lines with any prespecified angles with variety of line detection masks.
  - Example

–1	–1	–1
2	2	2
–1	–1	–1

–1	2	–1
–1	2	–1
–1	2	–1

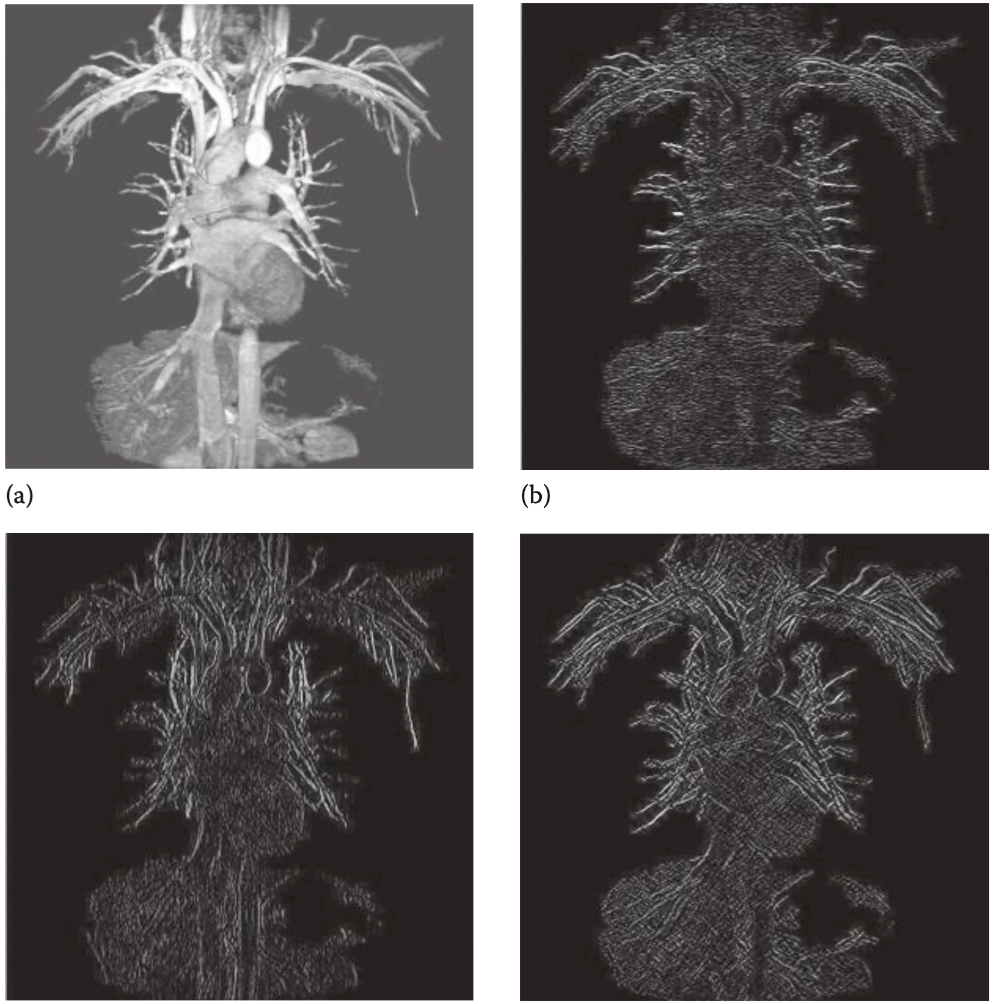
–1	–1	2
–1	2	–1
2	–1	–1

2	–1	–1
–1	2	–1
–1	–1	2

Line detection masks. From left to right: horizontal lines, vertical lines, rising lines with 45° angle, and falling lines with −45° angle

Image Segmentation

- Line Detection
  - Example



(a) (b) (c) (d)

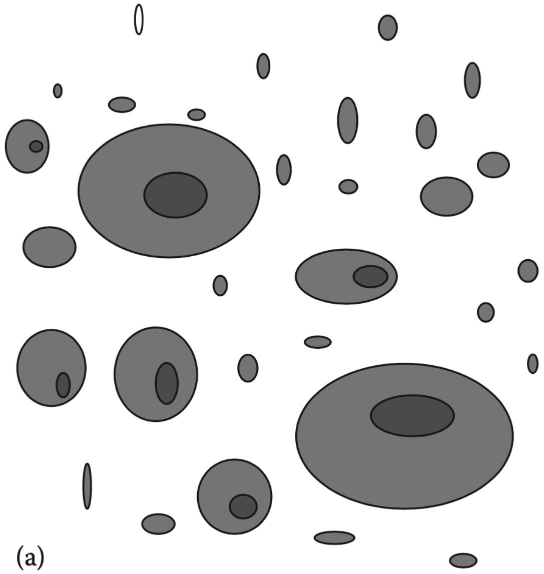
(a) Original image,
(b) image after horizontal line detection,
(c) image after vertical line detection, and
(d) image after 45° line detection
(Courtesy of Andre D'Avila, MD, Heart Institute (InCor), University of Sao Paulo, Medical School, Sao Paulo, Brazil)

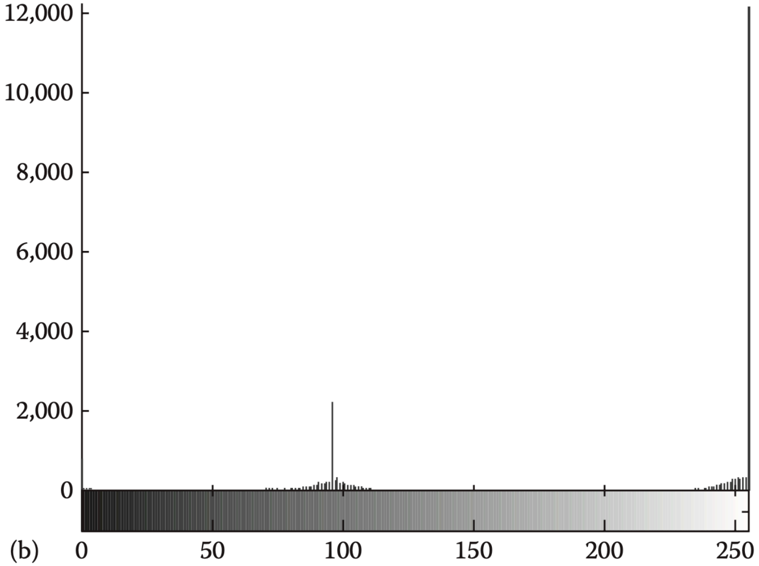
Image Segmentation

- Region and Object Segmentation
  - Distinguishing and detecting regions representing different objects are particularly important for biomedical image processing, because in a typical task of medical image analysis, one needs to detect regions representing objects, such as tumors, from background.

Image Segmentation

- Luminance Thresholding
  - The pixels in the objects of interest have gray levels that are either greater or smaller than the gray levels of the background pixels.





(a) A synthetic cell image that contains cells that are much darker than the bright background. (b) image histogram

Image Segmentation

- Luminance Thresholding
  - While in the synthetic images, such as the one shown in figure above, the separation process is rather easy. In real biomedical images, however, the gray-level separation process is much more complicated. Both the background and the interested object often occupy gray-level ranges that overlap each other.
  - A typical solution for these problem is dividing the original image into some sub-images.

Image Segmentation

- Region Growing
  - Methods above belong to the first category of segmentation algorithms, while region growing belongs to the second category.
  - Segmentation often starts by selecting a seed pixel for each region in the image in region growing methods. Seed pixels are often chosen close to the center of the region or object.
  - One of the most important factors are selecting a suitable similarity criterion.

Image Segmentation

- Region Growing
  - Example for criterion: In this example, we append to each seed all the pixels that (a) are 8-connected to that seed, and (b) are “similar” to it. Using absolute intensity differences as a measure of similarity, our predicate applied at each location (x, y) is:

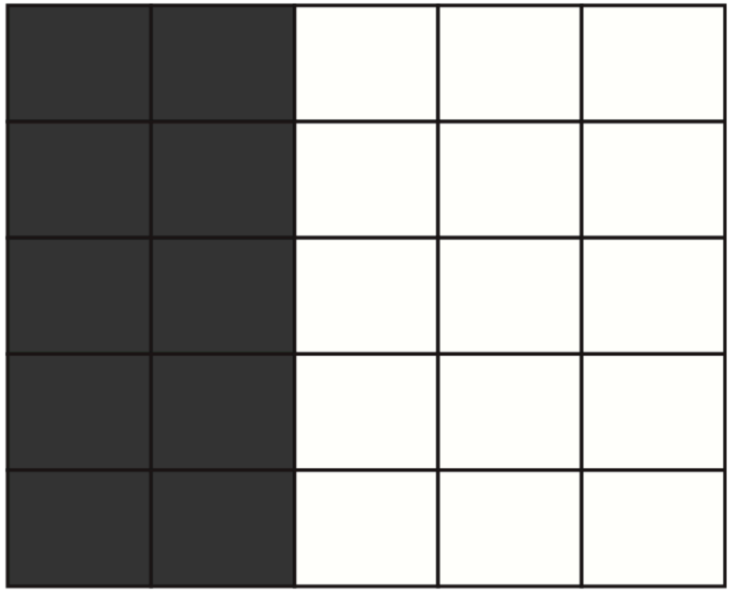
$$Q = \begin{cases} \text{TRUE} & \text{if the absolute difference of intensities} \\ & \text{between the seed and the pixel at } (x, y) \text{ is } \leq T \\ \text{FALSE} & \text{otherwise} \end{cases}$$

Image Segmentation

- Region Growing
  - Example

2	2	5	6	7
1	1	5	8	7
1	$\underline{1}$	6	$\underline{7}$	7
2	1	7	6	6
1	1	5	6	5

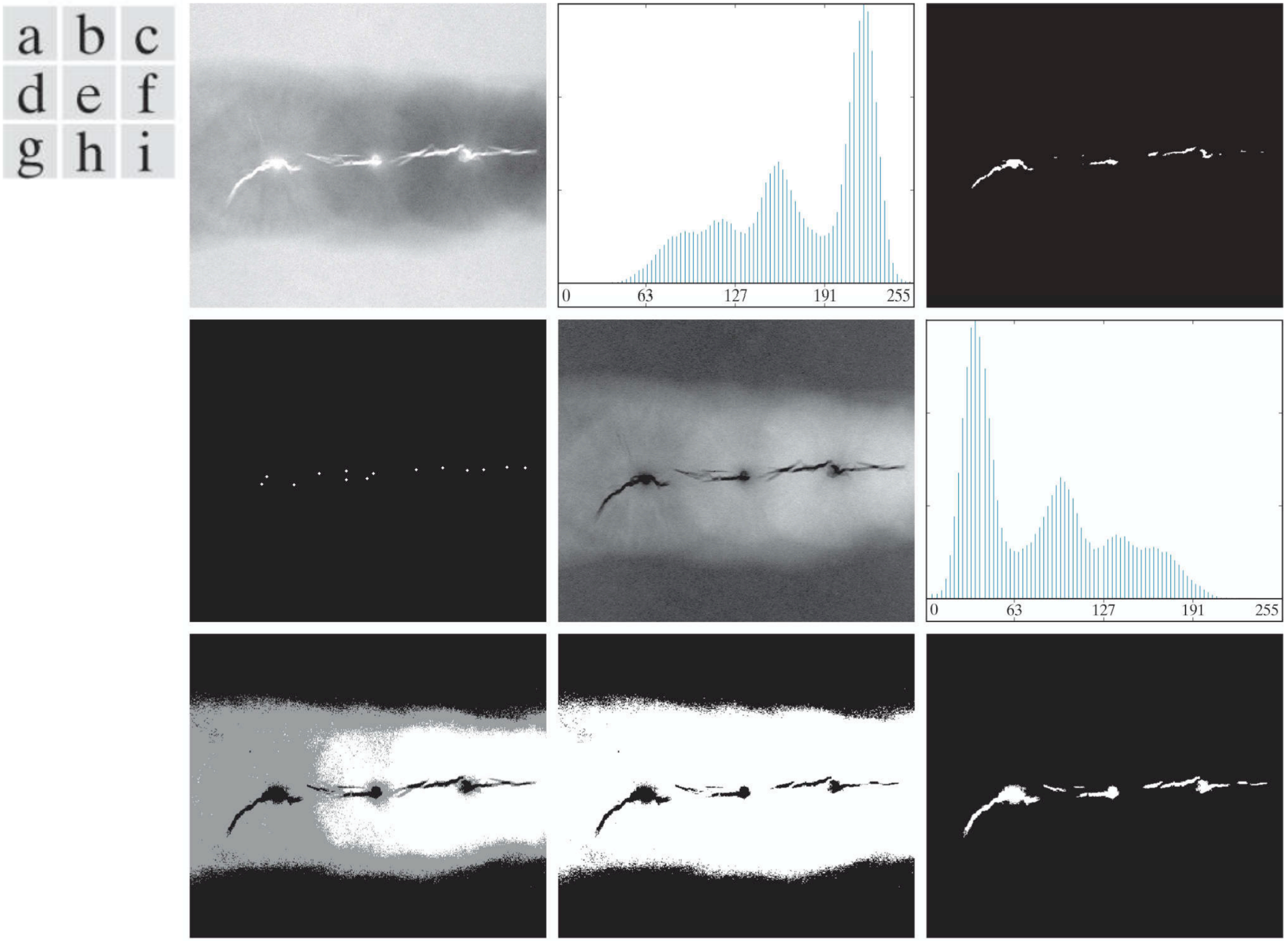
(a)



(b)

(a) Subimage with only the seed points and
(b) segmented subimage using region growing method

Image Segmentation



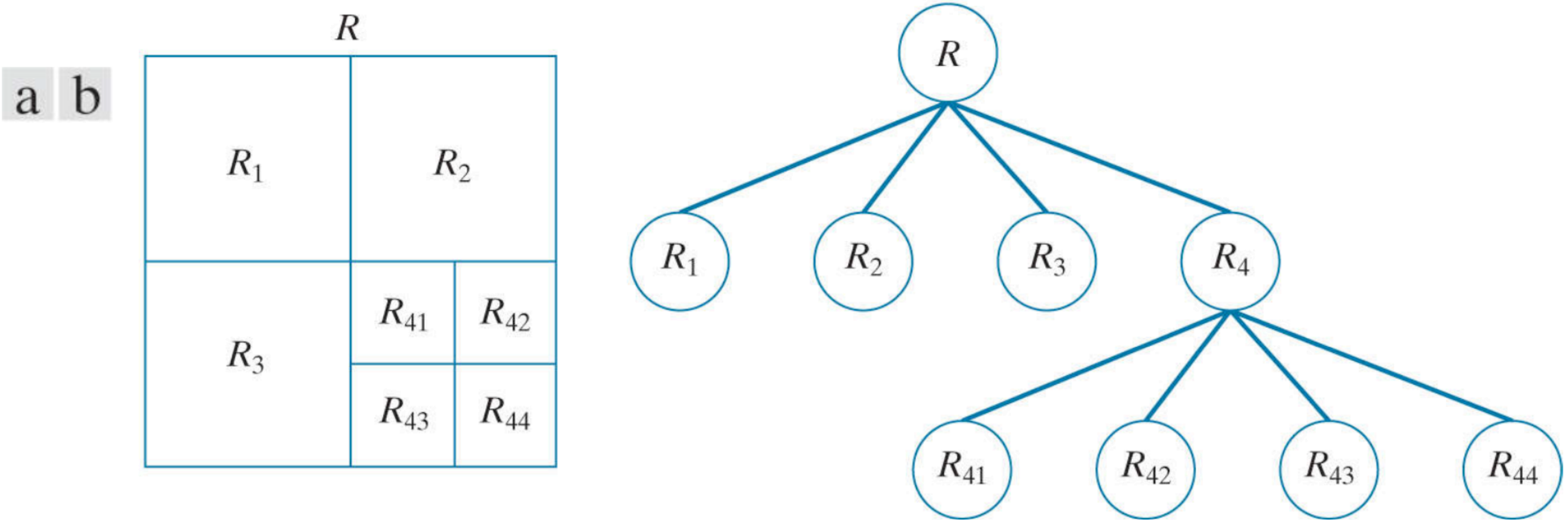
(a) X-ray image of a defective weld.
(b) Histogram.
(c) Initial seed image.
(d) Final seed image (the points were enlarged for clarity).
(e) Absolute value of the difference between the seed value (255) and (a).
(f) Histogram of (e).
(g) Difference image thresholded using dual thresholds.
(h) Difference image thresholded with the smallest of the dual thresholds.
(i) Segmentation result obtained by region growing.

Image Segmentation

- Region Splitting and Merging
  - Unlike region growing algorithm, an alternative is to subdivide an image initially into a set of disjoint regions and then merge and/or split the regions in an attempt to satisfy the conditions of segmentation

Image Segmentation

- Quad-Trees



(a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

Image Segmentation

- Quad-Trees
  - The preceding discussion can be summarized by the following procedure in which, at any step, we
    1) Split into four disjoint quadrants any region R_i for which $Q(R_i) = false$.
    2) When no further splitting is possible, merge any adjacent regions R_i and R_k for which $Q(R_j \cup R_k) = true$.
    3) Stop when no further merging is possible.

Image Segmentation

- Example for quad-trees algorithm

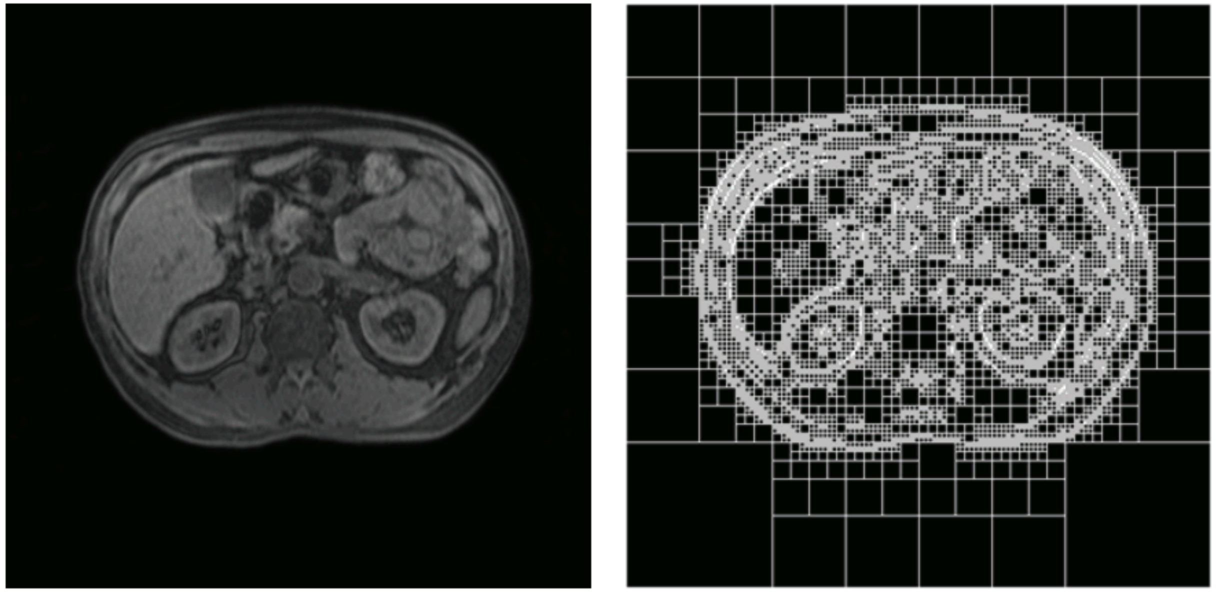
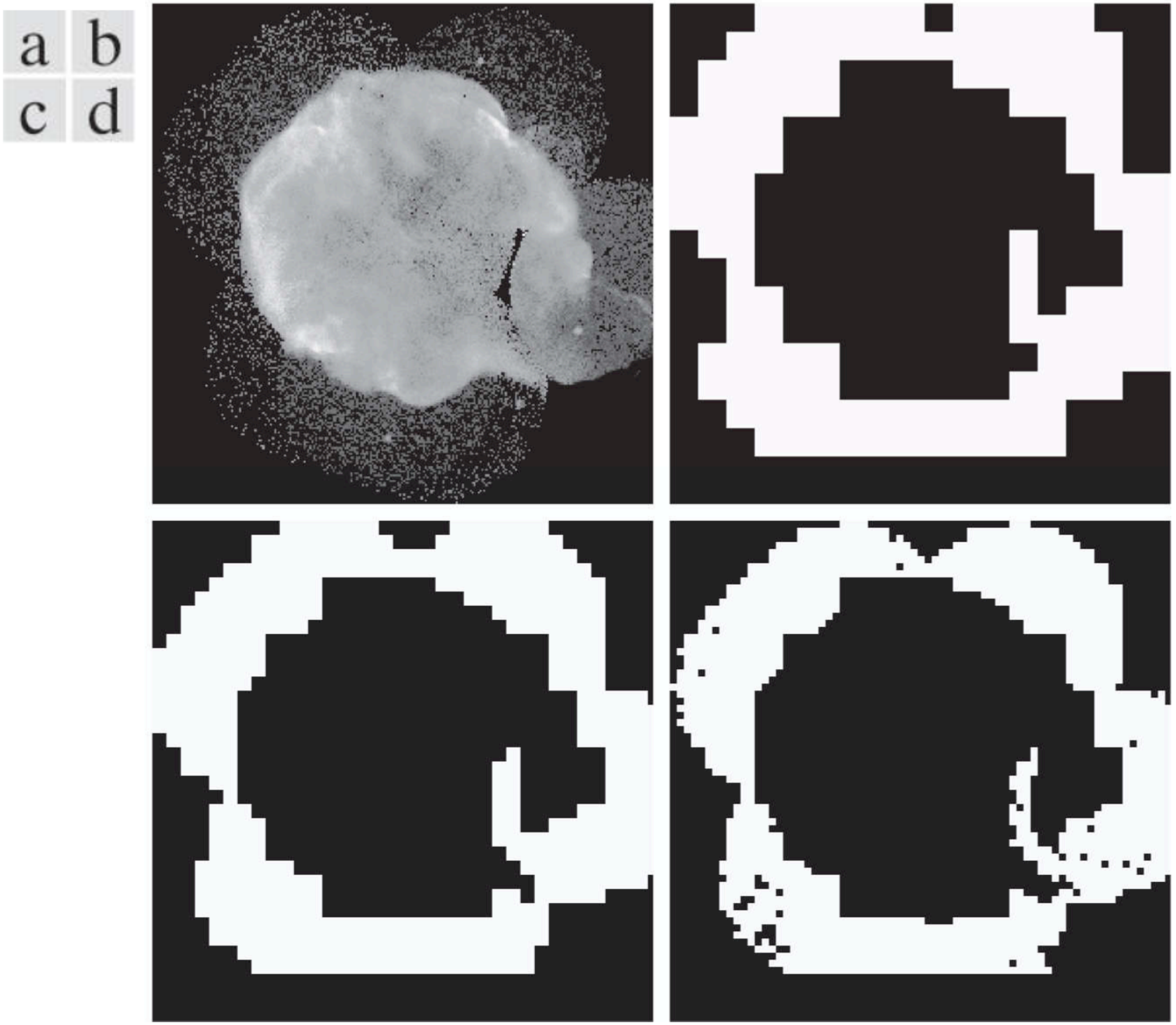


Figure (a).Expansion image   Figure (b).Quad tree decomposition

Lan H, Chen L, Hu W. An approach on liver medical image segmentation based on quad tree[C]//2011 International Conference on Multimedia Technology. IEEE, 2011: 152-154.

Image Segmentation



(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope.
(b) through (d): Results of limiting the smallest allowed quad-region to be of sizes of 32×32, 16×16, and 8×8 pixels, respectively.
The objective of this example is to segment (extract from the image) the "ring" of less dense matter surrounding the dense inner region.

The criterion:

$$Q(R) = \begin{cases} \text{TRUE} & \text{if } \sigma_R > a \text{ AND } 0 < m_R < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

Image Segmentation

- Clustering and Superpixels
  - K-means Clustering
  - Superpixels
  - Simple Linear Iterative Clustering (SLIC)

Image Segmentation

- K-Means Clustering
  - The basic idea behind the clustering approach is to partition a set Q of observations into a specified number k of clusters.
  - In K-means clustering, each observation is assigned to the cluster with the nearest mean and each mean is called the prototype of its cluster.
  - A K-means algorithm is an iterative procedure that successively refines the means until convergence is achieved.

Image Segmentation

- K-Means Clustering
  - The objective of K-means clustering is to partition the set of Q vector observations $\{z_1, z_2, \ldots, z_Q\}$ into k $(k \leq Q)$ disjoint cluster sets $\mathrm{C} = \{C_1, C_2, \ldots, C_k\}$. The following criterion of optimality is satisfied:

$$\arg\min_{C} \left(\sum_{i=1}^{k} \sum_{\mathbf{z} \in C_i} \|\mathbf{z} - \mathbf{m}_i\|^2 \right)$$

    $z = [z_1\ z_2 \cdots z_n]$ is the vector observation, m_i is the mean vector (or centroid) of the samples in set C_i and $\|\cdot\|$ is the vector norm.
  - NP-hard problem to find the minimum

Image Segmentation

- K-Means Clustering
  - A “standard” K-means algorithm: Given a set $\{z_1, z_2, \dots, z_Q\}$ of vector observation and a specified value of k
    - **Step 1: Initialize the algorithm**
      Specify an initial set of means $m_i(1), i = 1,2, \dots, k$.
    - **Step 2: Assign samples to clusters**
      Assign each sample to the cluster set whose mean is the closest:

$$\mathbf{z}_q \to C_i \text{ if } \| \mathbf{z}_q - \mathbf{m}_i \|^2 < \| \mathbf{z}_q - \mathbf{m}_j \|^2 \; j = 1, 2, \dots, k (j \neq i); q = 1, 2, \dots, Q$$

Image Segmentation

- K-Means Clustering
  - **Step 3: Update the cluster centers (means)**

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \quad i = 1, 2, \ldots, k$$

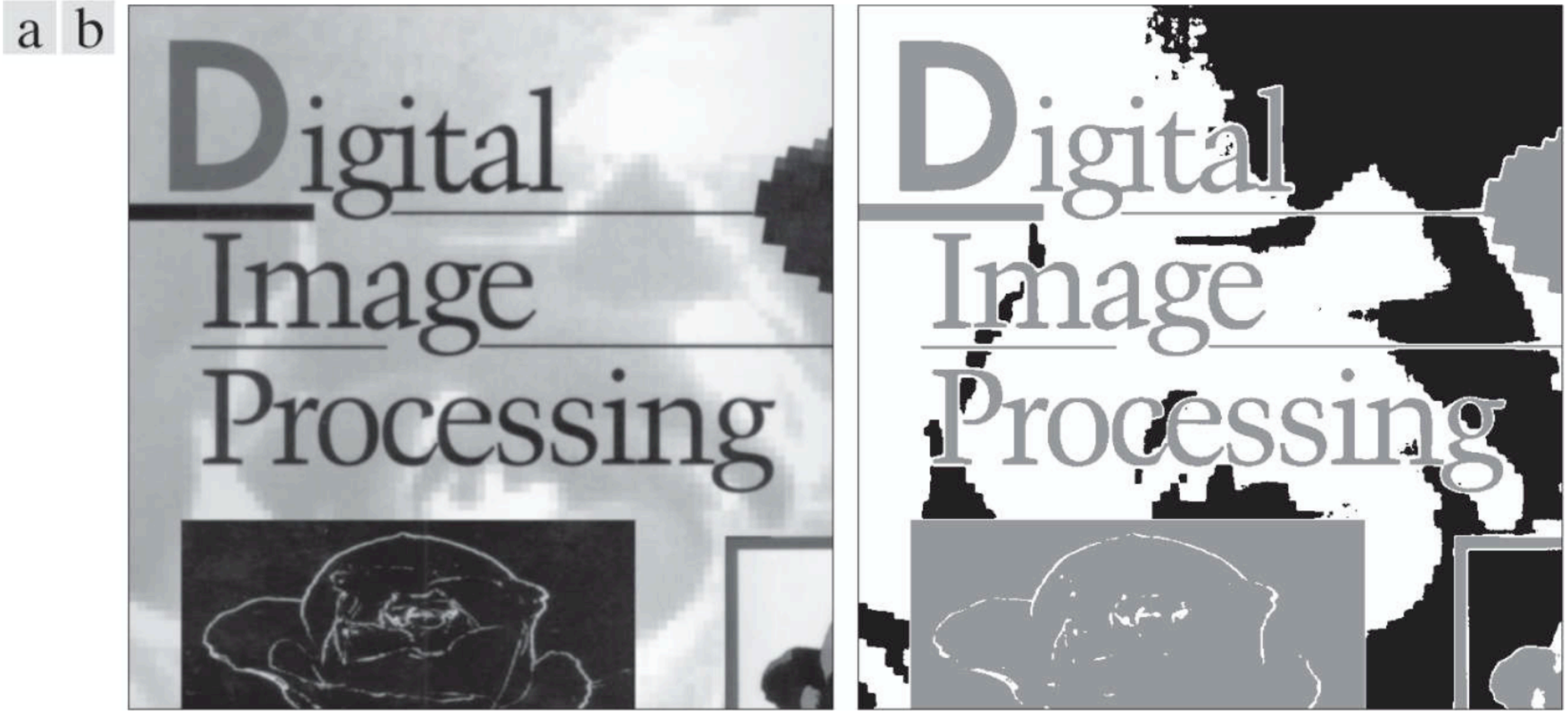
where $|C_i|$ is the number of samples in cluster set C_i.

  - **Step 4: Test for completion**

Compute the Euclidean norms of the differences between the mean vectors in the current and previous steps. Compute the residual error E as the sum of the k norms. Stop if $E \le T$, where T is a specified nonnegative threshold. Else, go back to Step 2.

Image Segmentation

- Examples for K-Means Clustering:



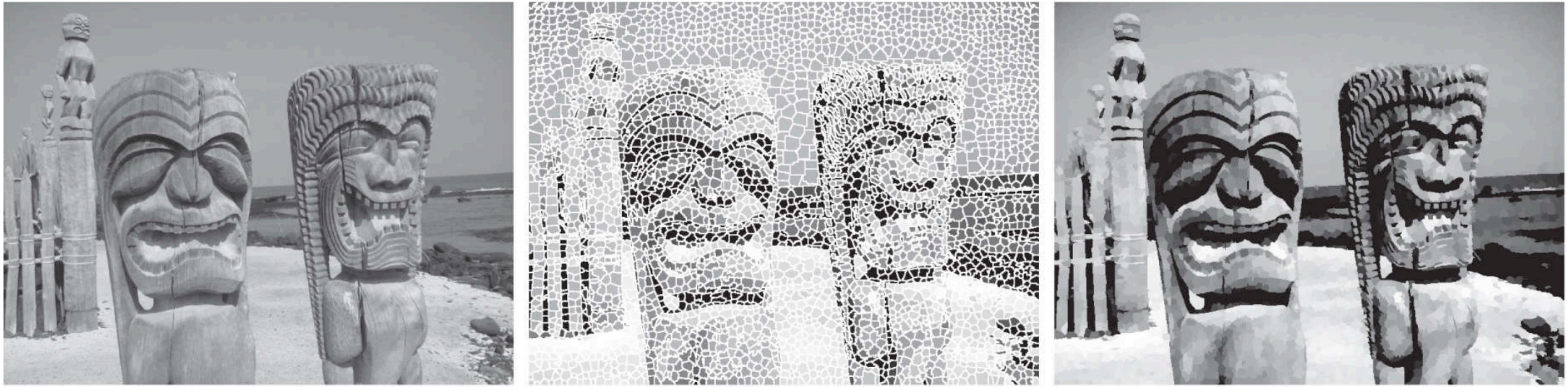
(a) Image of 688×688 pixels.
(b) Image segmented using the K-means algorithm with $k = 3$.

Image Segmentation

- Superpixels
  - The idea behind superpixels is to replace the standard pixel grid by grouping pixels into primitive regions that are more perceptually meaningful than individual pixels.
  - The objectives are to lessen computational load, and to improve the performance of segmentation algorithms by reducing irrelevant detail.

Image Segmentation

- Superpixels



(a) Image of size 600×480 (480,000) pixels. (b) Image composed of 4,000 superpixels (the boundaries between superpixels (in white) are superimposed on the superpixel image for reference—the boundaries are not part of the data). (c) Superpixel image.

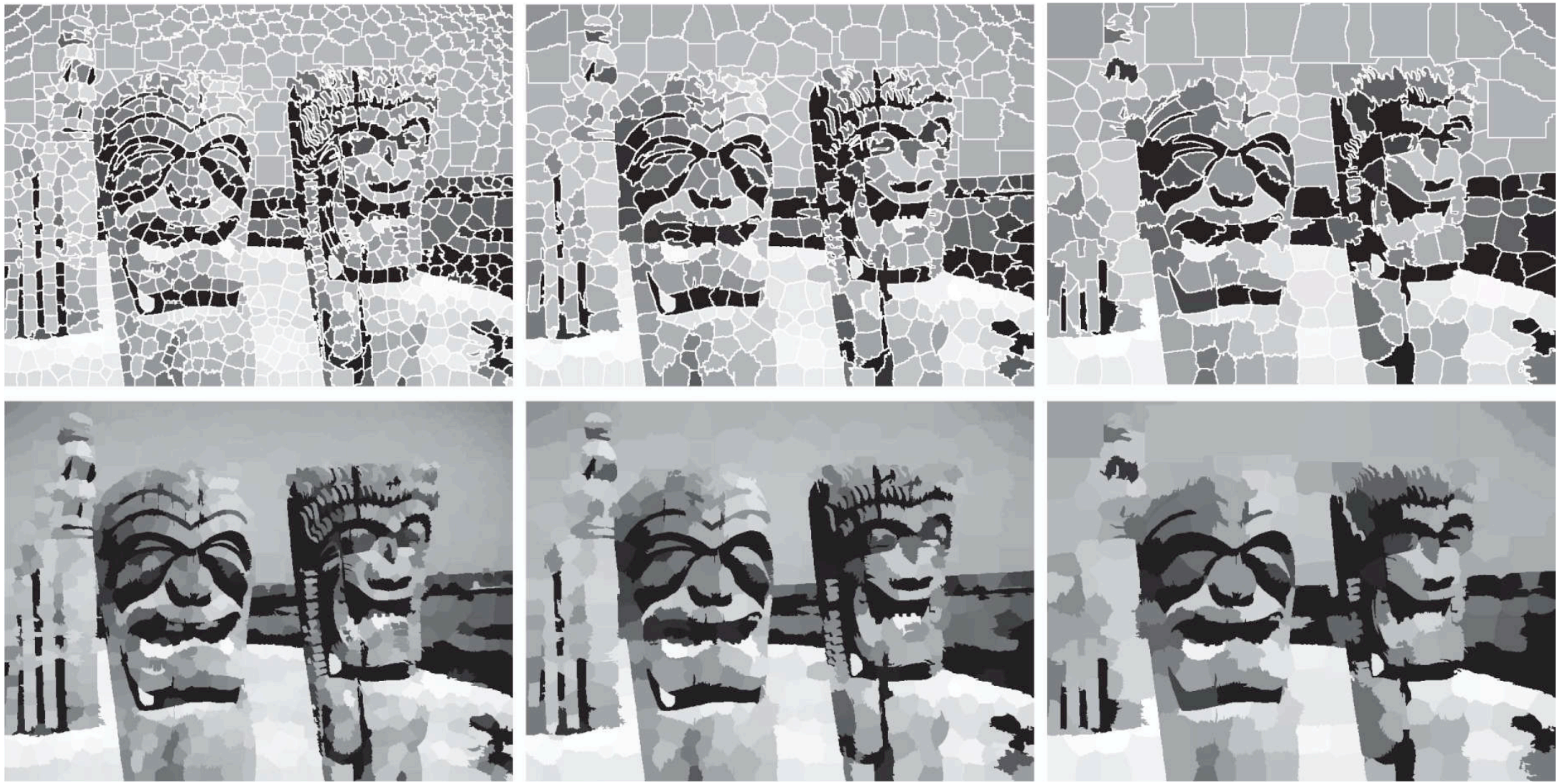
Image Segmentation

- Superpixels
  - It is possible to reduce the difference between a superpixel image and its parent image, and still achieve significant savings in storage and computation time.

a b c (a) Original image. (b) Image composed of 40,000 superpixels. (c) Difference between (a) and (b).

Image Segmentation

- Superpixels



Results of severely decreasing the number of superpixels. Top row: Results of using 1,000, 500, and 250 superpixels in the representation. As before, the boundaries between superpixels are superimposed on the images for reference. Bottom row: Superpixel images.

Image Segmentation

- Requirements for Superpixels Algorithm
  - Adherence to boundaries
  - Preservations of topological properties
  - Computational efficiency.

Image Segmentation

- Simple Linear Iterative Clustering (SLIC)
  - Simple linear iterative clustering (SLIC) is a modification of the K-means algorithm.
  - SLIC observations typically use (but are not limited to) 5-dimensional vectors containing three color components r, g, b and two spatial coordinates x, y.

$$z = [r\ g\ b\ x\ y]^T$$

Image Segmentation

- SLIC Superpixels Algorithm
  - **Step 1: Initialize the algorithm**

    Compute the initial superpixel cluster centers,

$$\mathbf{m}_i = [r_i \;\; g_i \;\; b_i \;\; x_i \;\; y_i]^T, i = 1, 2, \ldots, n_{sp}$$

by sampling the image at regular grid steps s. Move the cluster centers to the lowest gradient position in a 3×3 neighborhood. For each pixel p, in the image, set a label $L(p) = -1$ and a distance $d(p) = \infty$.

Image Segmentation

- SLIC Superpixels Algorithm
  - **Step 2: Assign samples to cluster centers**

    For each cluster center $m_i, i = 1,2, \ldots, n_{sp}$, compute the distance $D_i(p)$ between m_i and each pixel p in a $2s \times 2s$ neighborhood of m_i. Then, for each p and $i = 1,2, \ldots, n_{sp}$, if $D_i < d(p)$, let $d(p) = D_i$, and $L(p) = i$.

  - **Step 3: Update the cluster centers**

    Let C_i denote the set of pixels in the image with label $L(p) = i$. Update m_i

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \qquad i = 1, 2, \ldots, n_{sp}$$

    where $|C_i|$ is the number of pixels in set C_i.

Image Segmentation

- SLIC Superpixels Algorithm
  - **Step 4: Test for convergence**

    Compute the Euclidean norms of the differences between the mean vectors in the current and previous steps. Compute the residual error E, as the sum of the n_{sp} norms. If $E < T$, where T is a specified nonnegative threshold, go to Step 5. Else, go back to Step 2.

  - **Step 5: Post-process the superpixel regions**

    Replace all the superpixels in each region C_i by their average value m_i.

  - Note in Step 5 that superpixels end up as contiguous regions of constant value

Image Segmentation

- SLIC Superpixels Algorithm
  - Specifying the Distance Measure

$$d_c = [(r_j - r_i)^2 + (g_j - g_i)^2 + (b_j - b_i)^2]^{1/2} \qquad d_s = [(x_j - x_i)^2 + (y_j - y_i)^2]^{1/2}$$

- We then define D as the composite distance

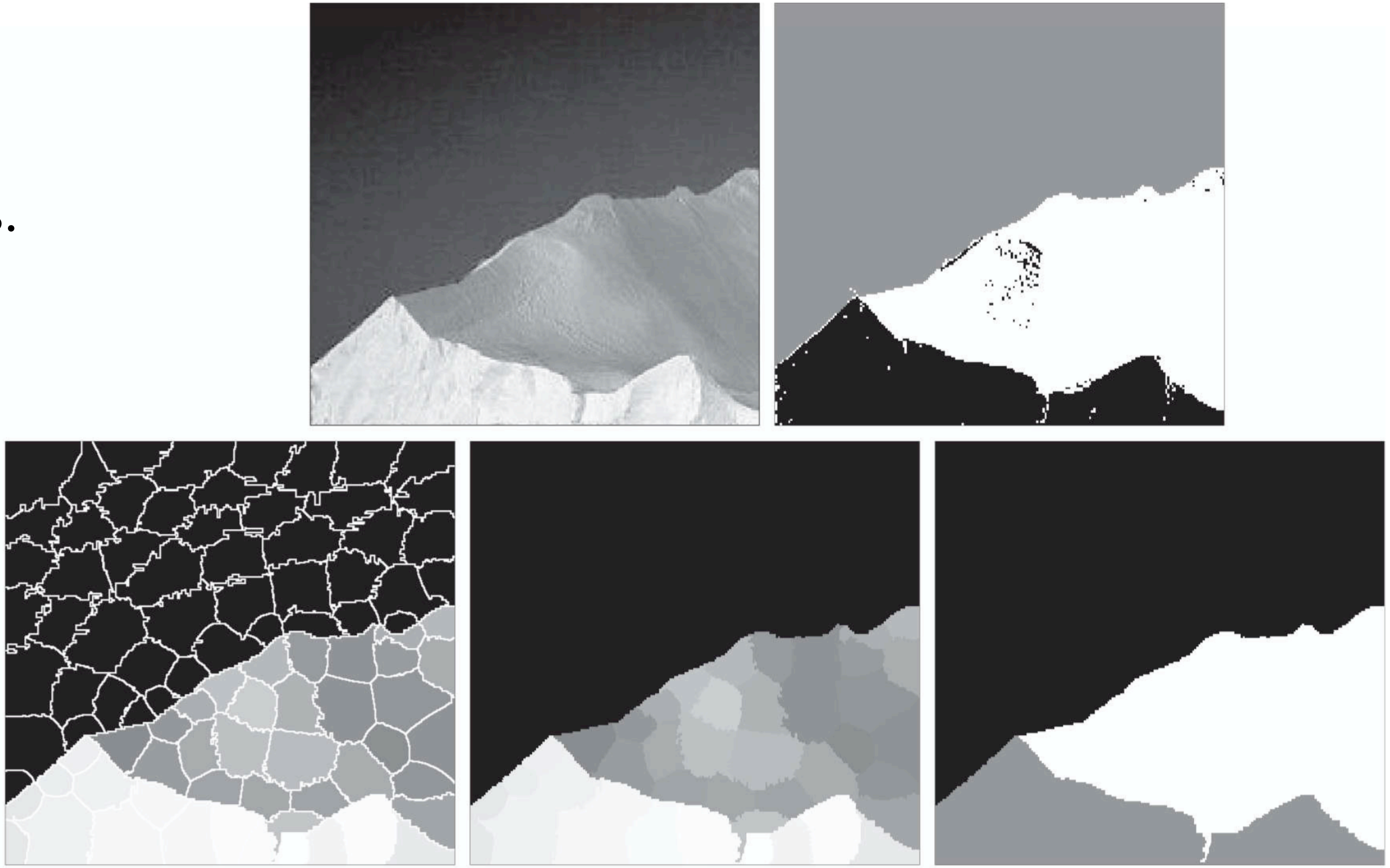
$$D = \left[\left(\frac{d_c}{d_{cm}}\right)^2 + \left(\frac{d_s}{d_{sm}}\right)^2\right]^{1/2}$$

- where d_{cm} and d_{sm} are the maximum expected values of d_c and d_s.

Image Segmentation

- Example for SLIC

(a) Image of 533×566 (301,678) pixels.
(b) Image segmented using the k-means algorithm.
(c) 100-element superpixel image showing boundaries for reference.
(d) Same image without boundaries.
(e) Superpixel image (d) segmented using the k-means algorithm.



a b
c d e

Image Segmentation

- Segmentation using Morphological Watersheds
  - Visualize an image in 3D:spatial coordinates and intensity
  - In such a topographic interpretation, there are 3 types of points:
    - Points belonging to a regional minimum
    - Points at which a drop of water would fall to a single minimum.
    - Points at which a drop of water would be equally likely to fall to more than one minimum.

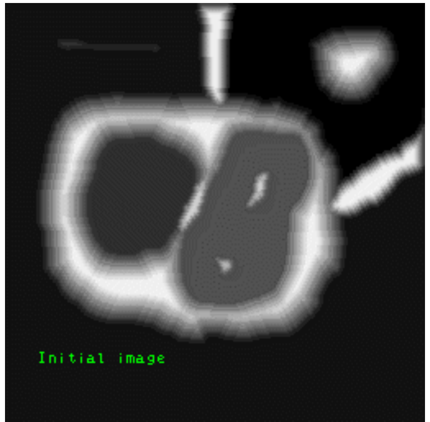


Image Segmentation

- Segmentation using Morphological Watersheds

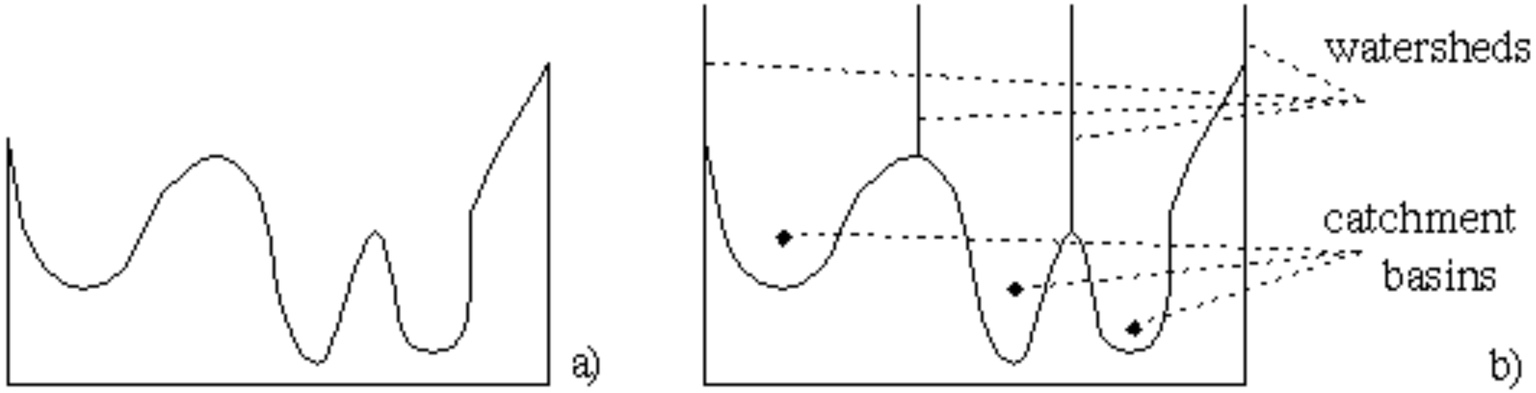


Figure 5.47 *One-dimensional example of watershed segmentation. (a) Gray level profile of image data. (b) Watershed segmentation – local minima of gray level (altitude) yield catchment basins, local maxima define the watershed lines.*

Image Segmentation

- Segmentation using Morphological Watersheds
  - The objective is to find watershed lines
  - The idea is simple:
    - Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
    - When rising water in distinct catchment basins is about the merge, a dam is built to prevent merging. These dam boundaries correspond to the watershed lines.

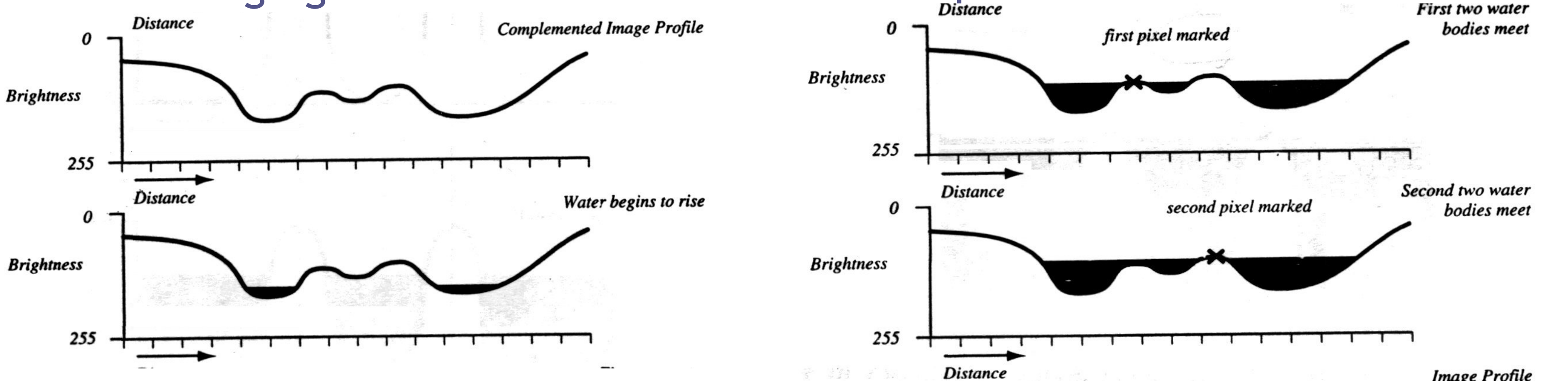
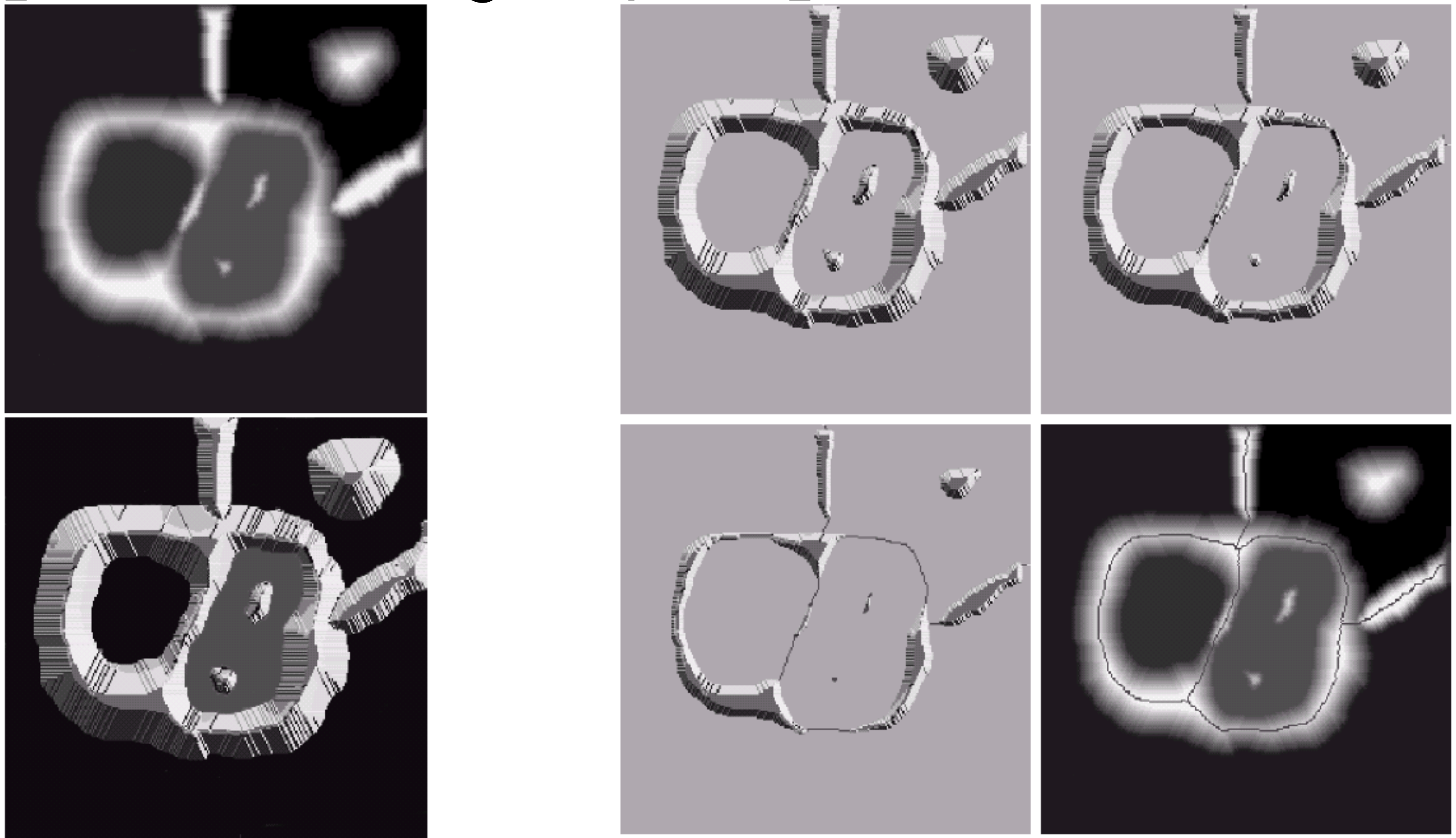


Image Segmentation

- Segmentation using Morphological Watersheds



e f
g h

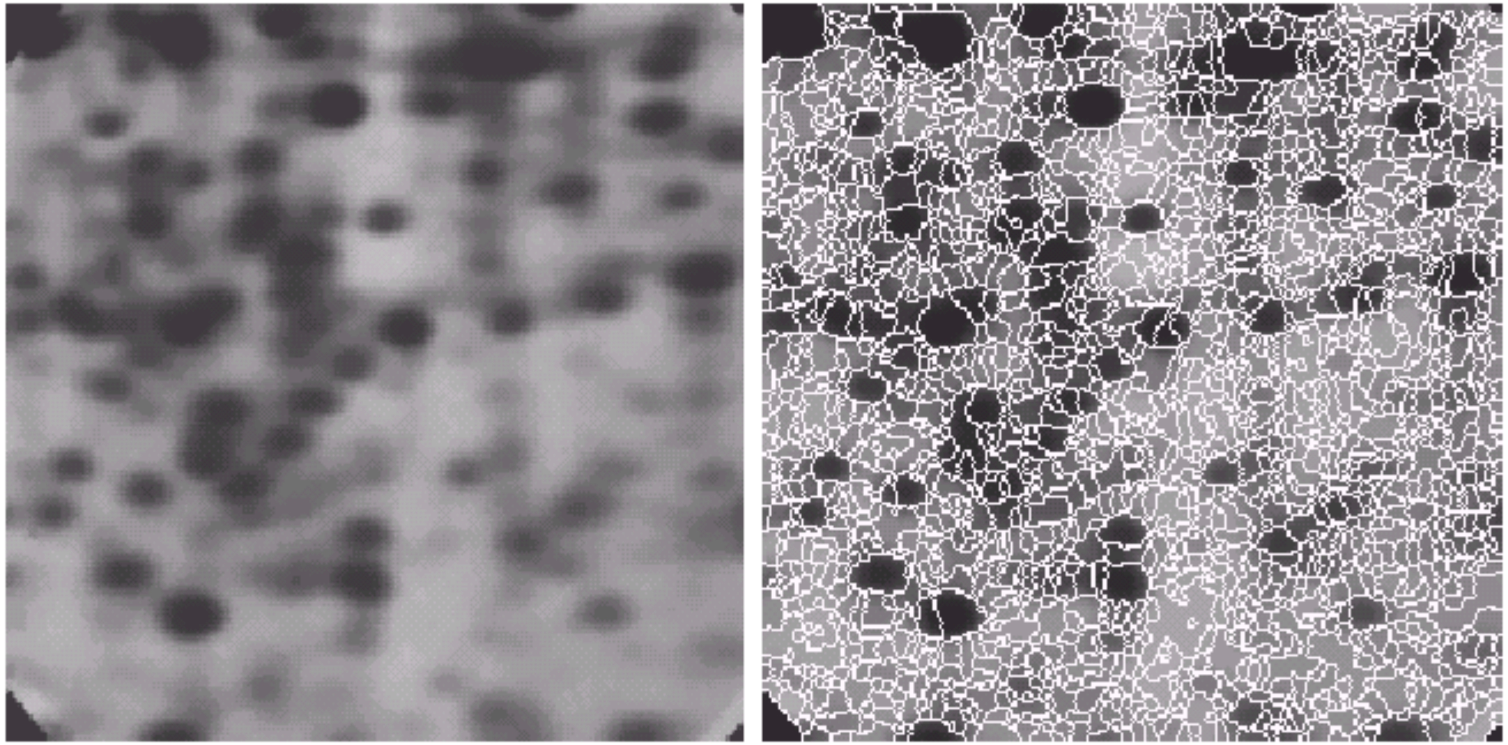
FIGURE 10.44 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Image Segmentation

- Segmentation using Morphological Watersheds、
  - Start with all pixels with the lowest possible value.
    - These form the basis for initial watersheds
  - For each intensity level k:
    - For each group of pixels of intensity k
      - If adjacent to exactly one existing region, add these pixels to that region
      - Else if adjacent to more than one existing regions, mark as boundary
      - Else start a new region

Image Segmentation

- Segmentation using Morphological Watersheds、
  - Due to noise and other local irregularities of the gradient, oversegmentation might occur.



a b

FIGURE 10.47 (a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Image Segmentation

- Segmentation using Morphological Watersheds、
  - A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima.

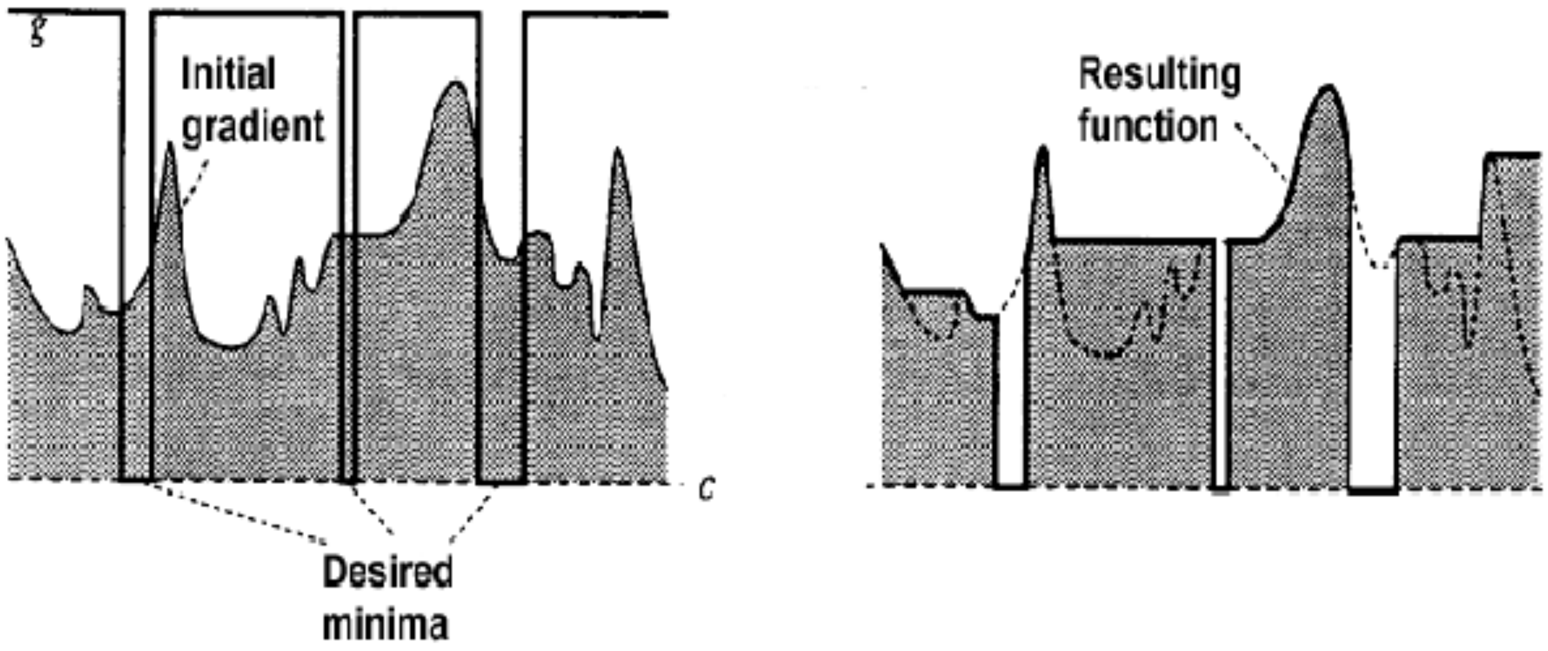
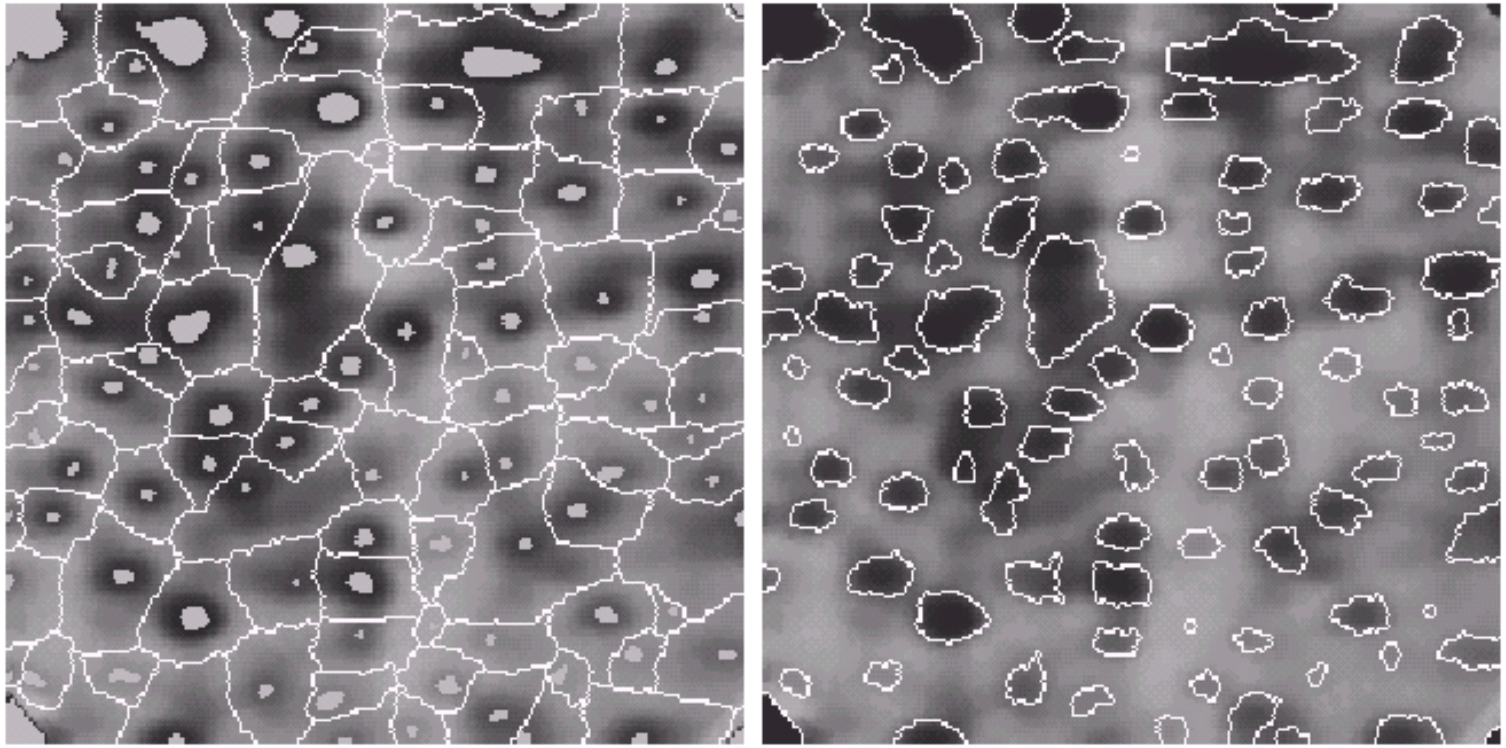


Image Segmentation

- Segmentation using Morphological Watersheds、
  - A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima.



a b

FIGURE 10.48 (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Image Segmentation

- Image Segmentation from a “modeling” point of view
  - Deformable models are physically based models of deformable curves, surfaces, and solids used traditionally in computer graphics.
  - Active contours (also called evolving fronts or evolving interfaces), are deformable models confined to the plane.
  - Work on active contours related specifically to image segmentation evolved along two different paths:
    - Snakes (Kass, Witkin, and Terzopoulus [1988])
    - Level sets (Osher and Sethian [1988])

Image Segmentation

- Snakes
  - Snakes are parametric representations of active contours.
  - The fundamental snake equation

$$\alpha \mathbf{c}''(s) - \beta \mathbf{c}''''(s) + \mathbf{F}(\mathbf{c}(s)) = 0$$

where the term F(c(s)) is a 2-D vector containing the force at points along the snake curve, c. This equation indicates that finding the snake contour can be interpreted as a process of balancing internal (elastic and bending) forces against an external force.

Image Segmentation

- Snakes
  - Iterative Solution of the Snake Equation

$$\mathbf{x}(t) = \mathbf{A}\left[\mathbf{x}(t-1) + \gamma\, \mathbf{F}_x(\mathbf{x}(t-1),\, \mathbf{y}(t-1))\right]$$

$$\mathbf{y}(t) = \mathbf{A}\left[\mathbf{y}(t-1) + \gamma\, \mathbf{F}_y(\mathbf{x}(t-1),\, \mathbf{y}(t-1))\right]$$

These two equations constitute the iterative form of the snake equation. They have reduced the problem of finding a segmentation snake to solving two straightforward iterative equations—a trivial task, especially in a matrix-oriented language, such as MATLAB.

Image Segmentation

- Snakes
  - External Force Based on the Magnitude of the Image Gradient (MOG)
    - The external (image) energy is defined as:

$$E_{\text{image}}(x, y) = -\|\nabla f(x, y)\|^2$$
$$= -\left[(\partial f(x, y)/\partial x)^2 + (\partial f(x, y)/\partial y)^2\right]$$

    - Then, the force corresponding to $E_{image}(x, y)$ is obtained by:

$$\mathbf{F}(x, y) = -\nabla E_{\text{image}}(x, y)$$
$$= \nabla\left[\|\nabla f(x, y)\|^2\right]$$

Image Segmentation

- Snakes
  - External Force Based on the Magnitude of the Image Gradient (MOG)
    - If needed, the force components can be normalized as follows:

$$F_x(x, y) = F_x(x, y)/(\|\mathbf{F}(x, y)\| + \varepsilon)$$

$$F_y(x, y) = F_y(x, y)/(\|\mathbf{F}(x, y)\| + \varepsilon)$$

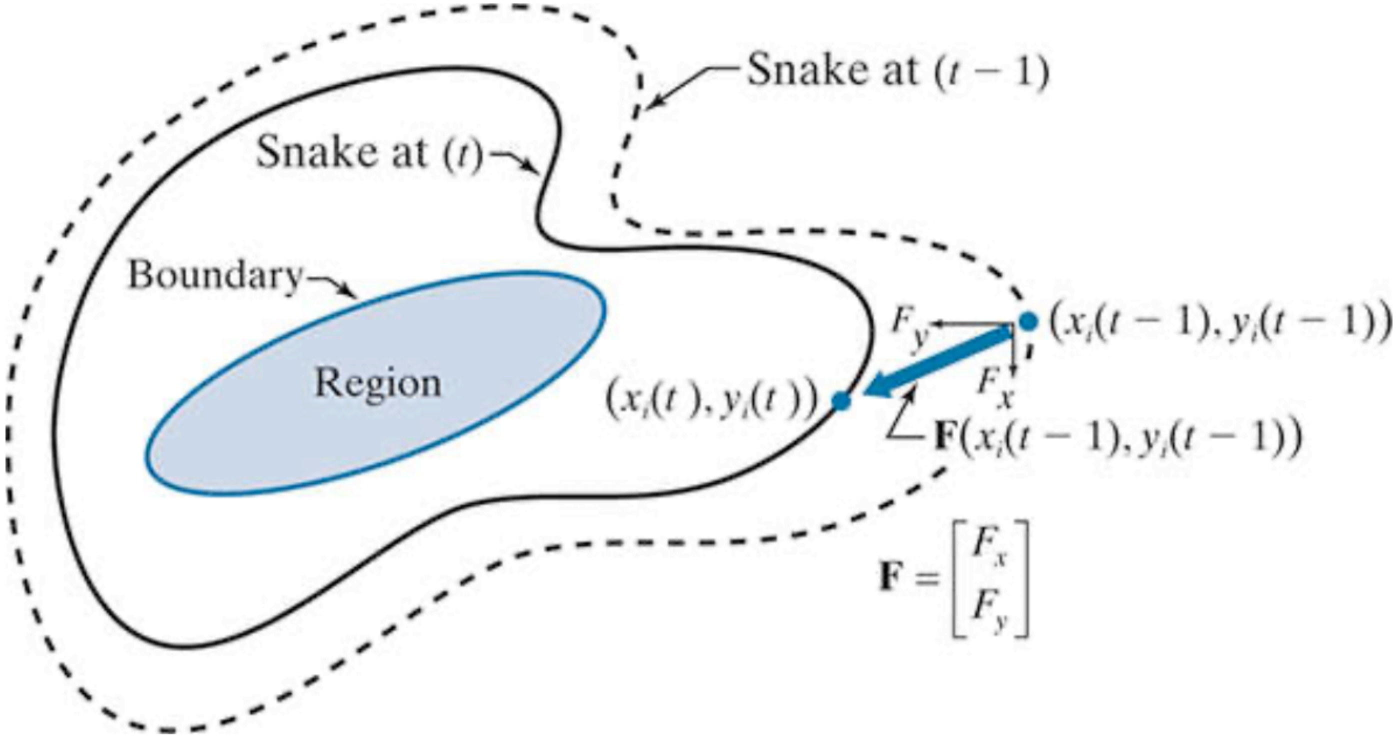
Where

$$\|\mathbf{F}(x, y)\| = \left[F_x(x, y)^2 + F_y(x, y)^2\right]^{1/2}$$

is the vector norm, and is a small constant used to prevent division by zero. This normalization helps in the selection of parameter , especially when experimenting with other snake variables.

Image Segmentation

- Illustration of how a snakes transitions from one time step to the next.

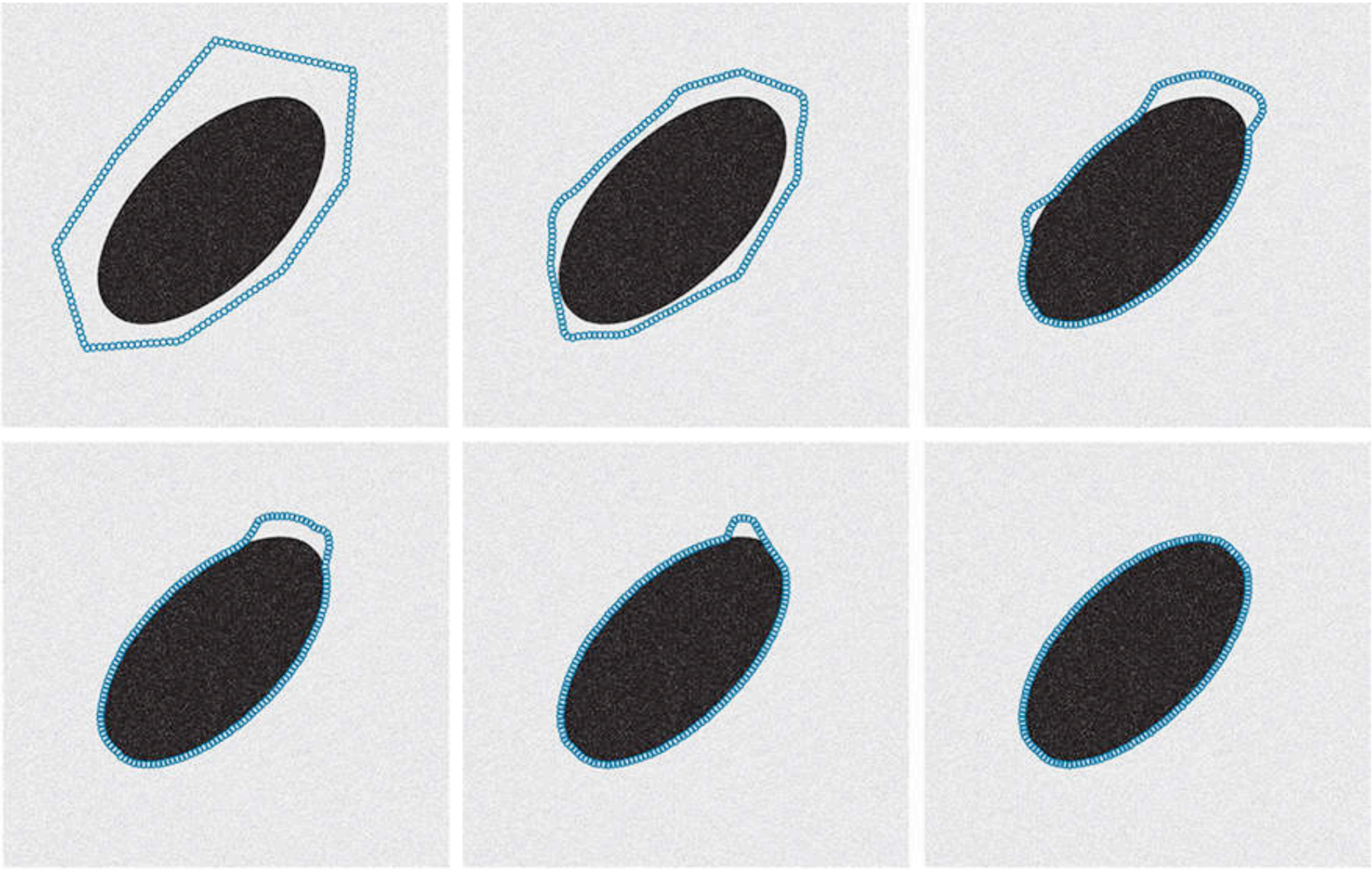


Only two corresponding snake points are shown for clarity. The snake consists of K such points, each influenced by a different component of the external force.

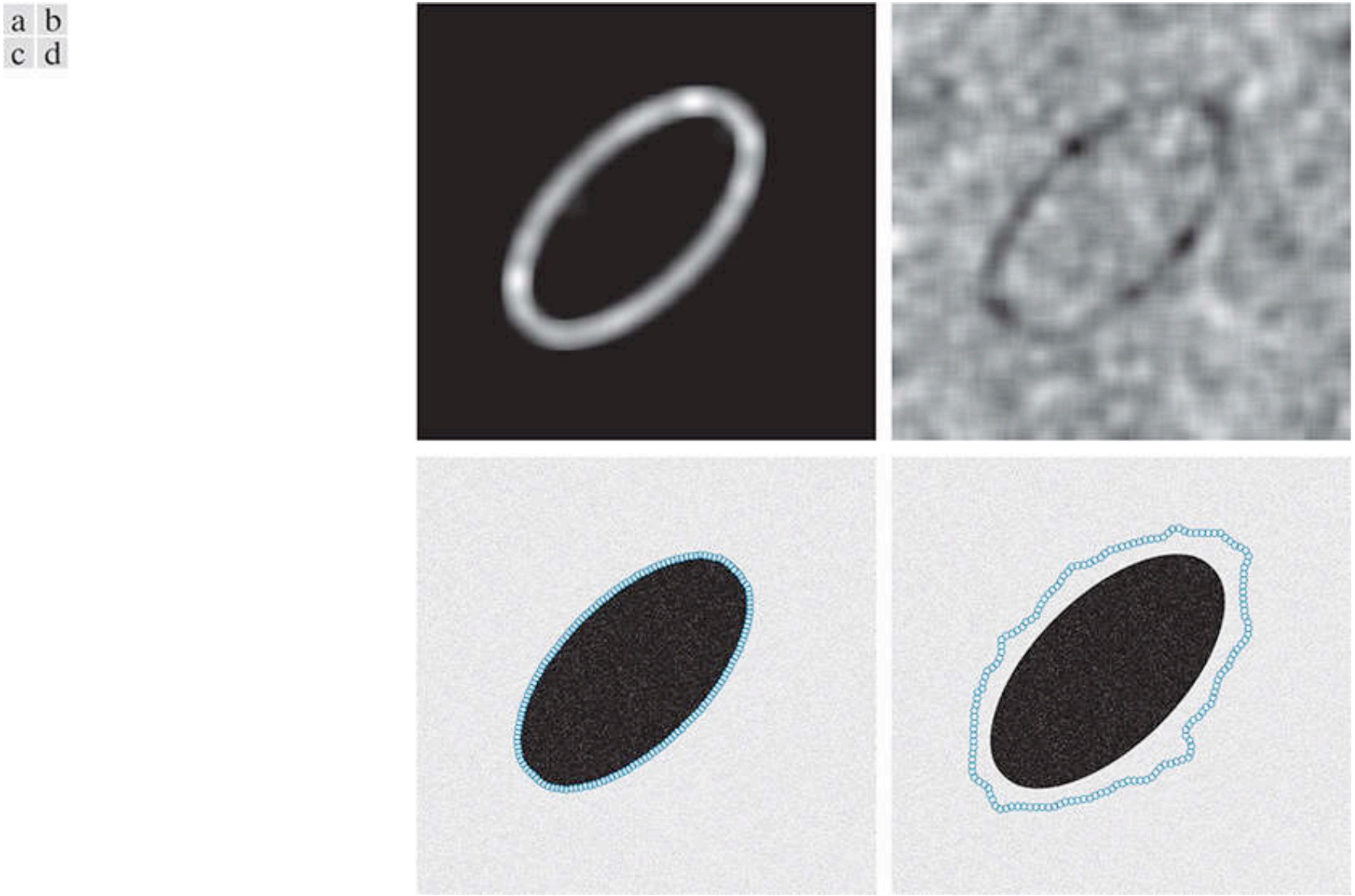
Image Segmentation

- Examples

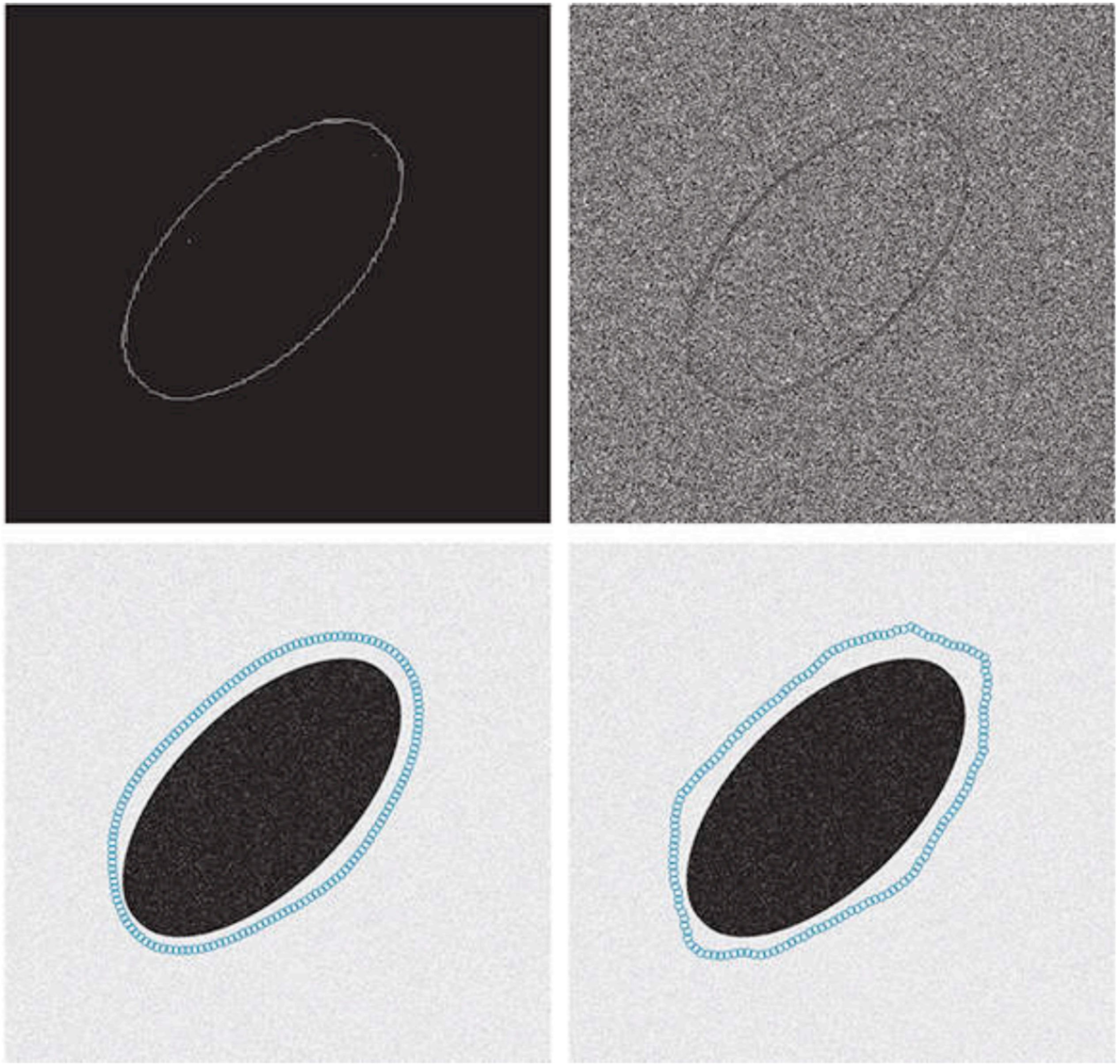
(a) Image and initial snake (the snake points are enclosed by small circles to make them easier to see). (b) Result after 10 iterations of the solution with α=0.5, β=0, and γ=0.6 . Note how the snake is beginning to become smooth. (c) through (f) Results after 50, 100, 150, and 200 iterations, respectively.



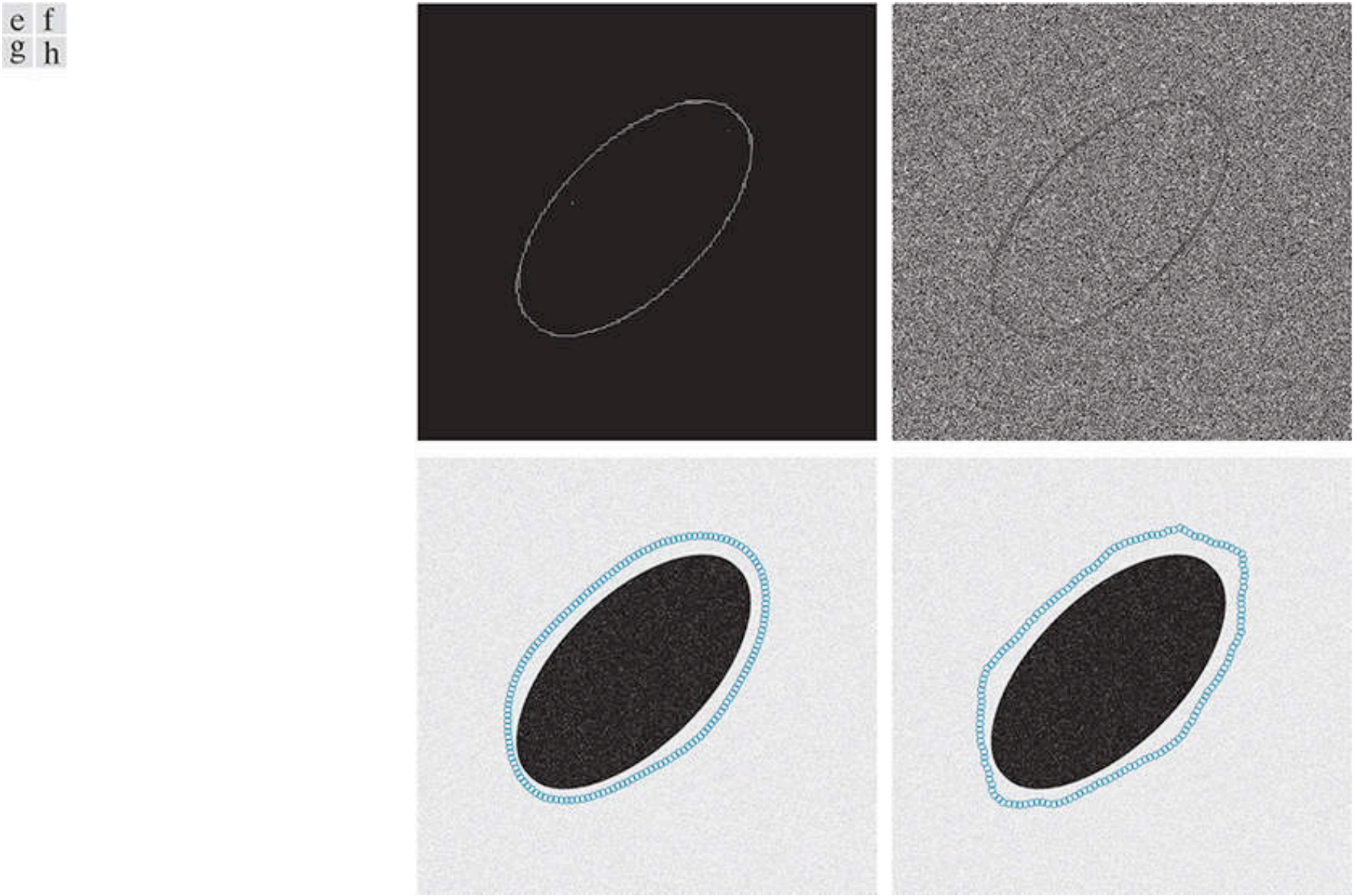
a b c
d e f



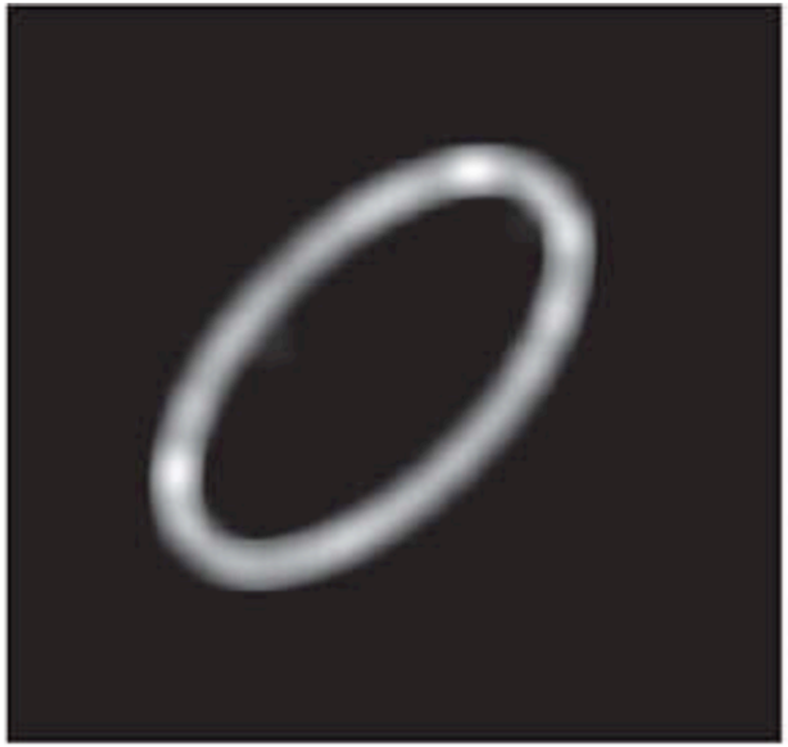
(a) Edge map used to generate the results in figure above. (b) Edge map with only the MOG filtered and then thresholded. (c) Result after 200 iterations using the forces based on (a). (d) Result after 200 iterations using the forces based on (b). The initial snake is shown in figure above (a).



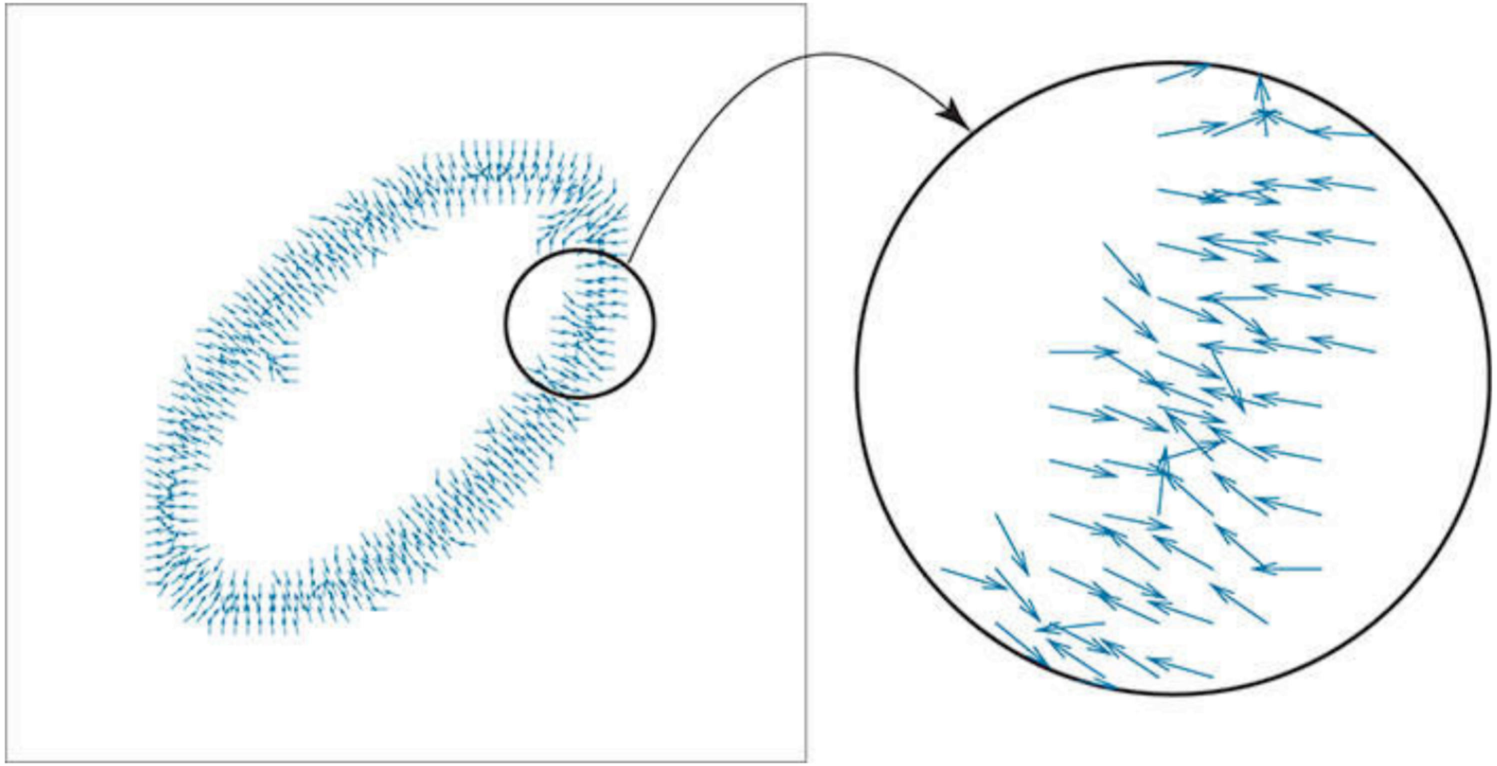
(e) Edge map with the image filtered and MOG thresholded (but not filtered). (f) Edge map with no filtering and the MOG thresholded. (g) Result after 200 iterations using the forces based on (e). (h) Result after 200 iterations using the forces based on (f). The initial snake.



(e) Edge map with the image filtered and MOG thresholded (but not filtered). (f) Edge map with no filtering and the MOG thresholded. (g) Result after 200 iterations using the forces based on (e). (h) Result after 200 iterations using the forces based on (f). The initial snake.



(a)



(b)

(a) Edge map. (b) Force field obtained using the edge map (a). All the arrows are of the same length because each element of the force field was normalized.

Image Segmentation

- Segmentation using Level Sets
  - Level sets in our context are sets of points of a 2-D curve formed by the intersection of a plane and a 3-D surface.
  - Unlike the parametric representation used for snakes, level sets are based on implicit representations. An important aspect of this approach is that it can adapt to changing topology during curve evolution.

Image Segmentation

- The implicitly representation of a 2-D contour
  - A 2-D contour can be defined as the intersection of a plane and a 3-D surface.

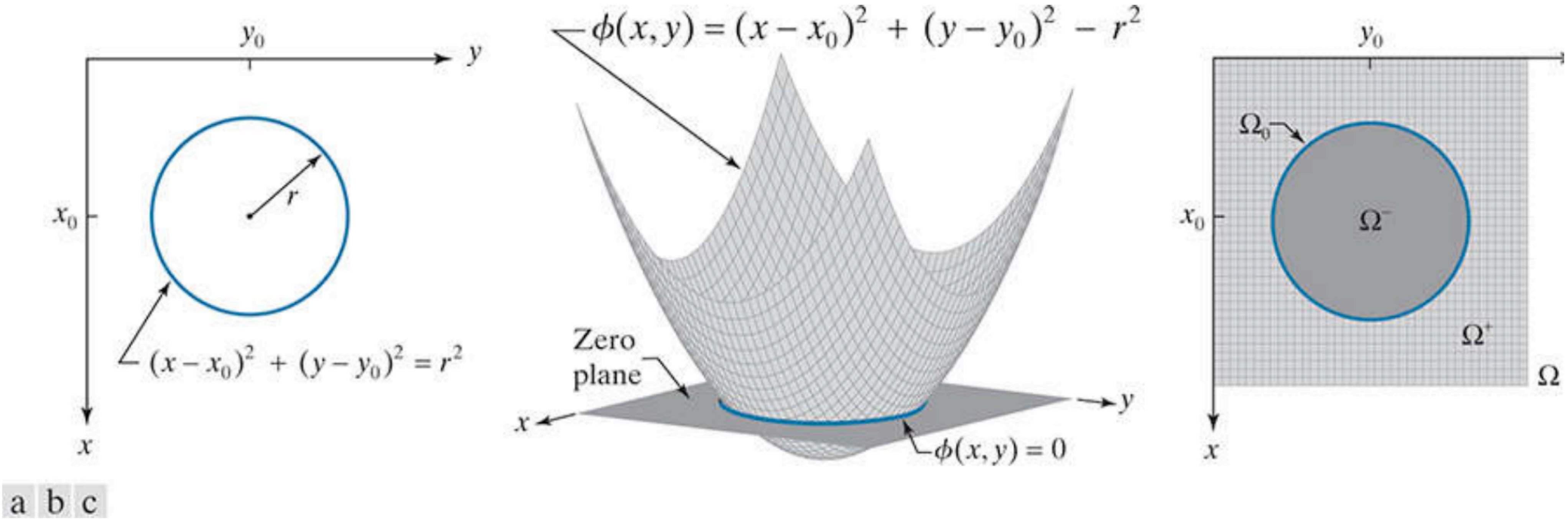


Image Segmentation

- Level Sets
  - The set of points in the intersection just mentioned is called a level set, and ϕ is usually referred to as a level set function.
  - When dealing with two variables, the level set reduces to a level set curve C which we define as:

$$C = \{(x, y) \mid \phi(x, y) = 0\}$$

  - Because the level set curves with which we work in this chapter are closed, it follows that (,)satisfies the following conditions for an arbitrary point (x, y):

$$\phi(x, y) = \begin{cases} > 0 & for (x, y) \in \Omega^{+} \\ = 0 & for (x, y) \in \Omega_0 \\ < 0 & for (x, y) \in \Omega^{-} \end{cases}$$

Illustration of how level sets are used for image segmentation conceptually.

(a) Conceptual image of a dry grass field containing three lakes. The point represents an initial fire. (b) Fire front expanding uniformly at some later time. (c) The fire front encounters a lake shore, causing it to burn around the edge of the lake. (d) and (e) Results after further burning. (f) Result after the fire has completely burned out. This simple concept is the foundation of interface boundary evolution based on level sets.

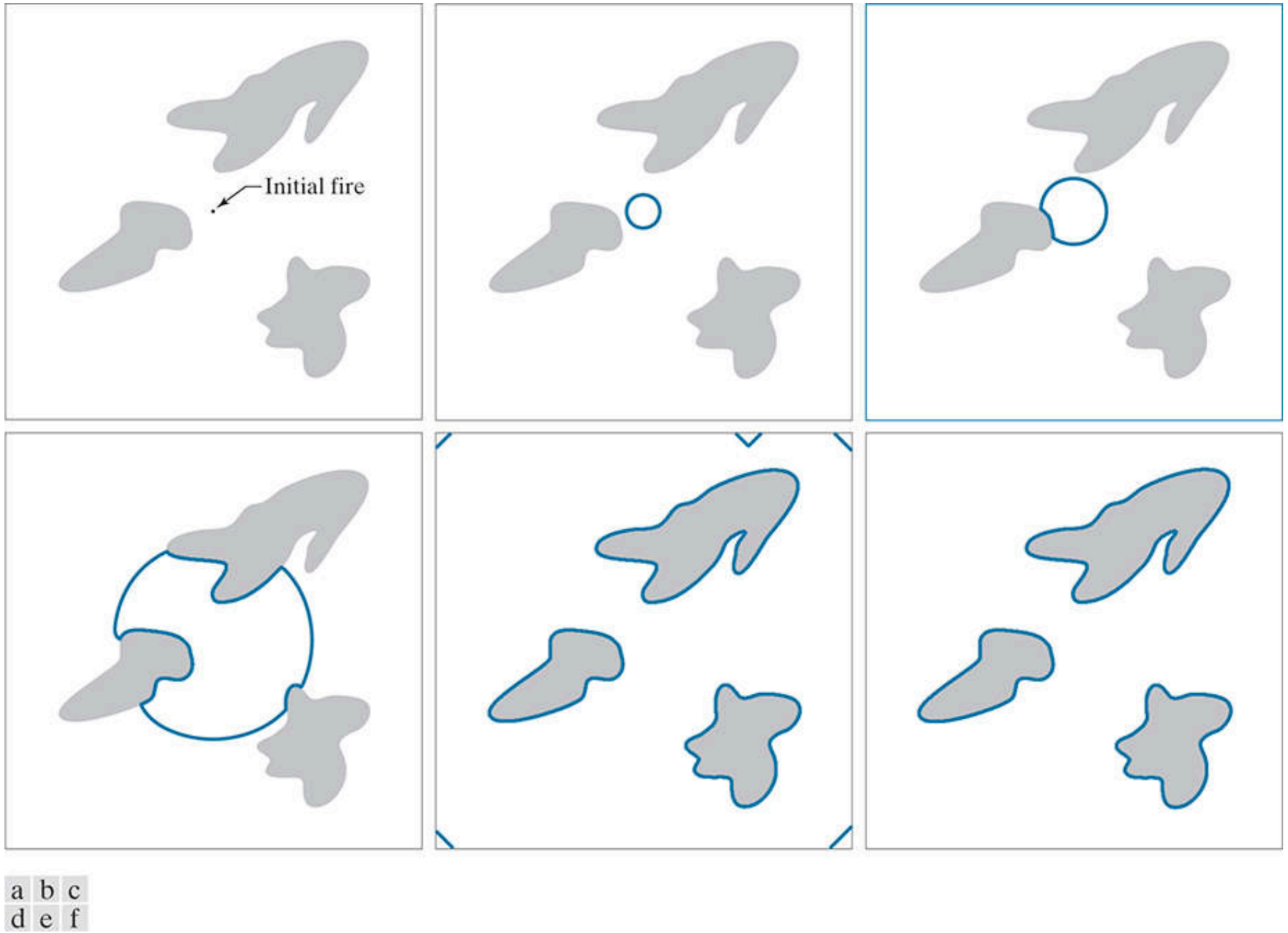


Image Segmentation

- Level Set Equation

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

  - The solution we are seeking is the zero level set of at steady state
    To apply this equation to image segmentation, we:
    1) Specify a suitable initial form for ϕ;
    2) Formulate F as a scalar field containing properties of interest in segmentation;
    3) Solve the level set equation to find a ϕ that satisfies equation above;
    4) Extract the segmentation contour as the zero level set of ϕ.

Image Segmentation

- Discrete (Iterative) Solution of the Level Set Equation

$$\phi^{n+1} = \phi^n - \Delta t\left\{\max(F, 0)\|\nabla\phi^n\|^+ + \min(F, 0)\|\nabla\phi^n\|^-\right\}$$

- This is the iterative implementation used to find a solution to the level set equation. As before, this expression properly chooses only one of the terms, $F\|\nabla\phi^n\|^+$ or $F\|\nabla\phi^n\|^-$, depending on whether F> 0 or F< 0 is true, respectively.

Image Segmentation

- Curvature
  - As we mentioned earlier, active contours often are interpreted as evolving fronts. In practice, fields are not perfectly uniform, and other factors such as wind currents and moisture affect how such a fire front evolves.
  - The curvature for implicit functions (such as the level set functions in this chapter) is defined as the divergence of the unit normal

$$\kappa = \mathrm{div}\left(\frac{\nabla\phi(x,y))}{\|\nabla\phi(x,y))\|}\right) = \nabla\cdot\left(\frac{\nabla\phi(x,y)}{\|\nabla\phi(x,y)\|}\right)$$

Image Segmentation

- Curvature
  - Equation above can be written equivalently as:

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

where, for example, $\phi_x = \frac{\partial\phi}{\partial x}, \phi_{xx} = \frac{\partial^2\phi}{\partial x^2}, \phi_x^2 = \left(\frac{\partial\phi}{\partial x}\right)^2$ *and* $\phi_{xy} = \frac{\partial^2\phi}{\partial x\partial y}$.

If is a signed distance function (see the next section), then $\|\nabla\phi\| = 1$,and we can write the equation as:

$$\kappa = \nabla\cdot\nabla\phi = \nabla^2\phi$$

$$= \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}$$

We recognize the last line of this equation as the Laplacian of ϕ.

Image Segmentation

- Specifying, Initializing, and Reinitializing Level Set Functions
  - The first step in implementing the iterative solution is to specify a form and initial value of the level set function .In theory, a level set function can be arbitrarily specified, provided that its zero level corresponds to the initially-specified contour. In practice, solving the level set equation requires iteration, which definitely is affected by the form selected for ϕ.
  - Let D(x, y) denote the Euclidean distance from an arbitrary point (x, y) on the plane to the closest point (x_B, y_B) on the boundary Ω:

$$D(x, y) = \min\left(\left[(x - x_B)^2 + (y - y_B)^2\right]^{1/2}\right)$$

Image Segmentation

- Specifying, Initializing, and Reinitializing Level Set Functions
  - where $(x_B, y_B) \in \Omega_0$. A class of functions used frequently for ϕ are signed distance functions, defined as:

$$\phi(x, y) = \begin{cases} D(x, y) > 0 & \text{i f}(x, y) \in \Omega^+ \\ 0 & \text{i f}(x, y) \in \Omega_0 \\ -D(x, y) > 0 & \text{i f}(x, y) \in \Omega^- \end{cases}$$

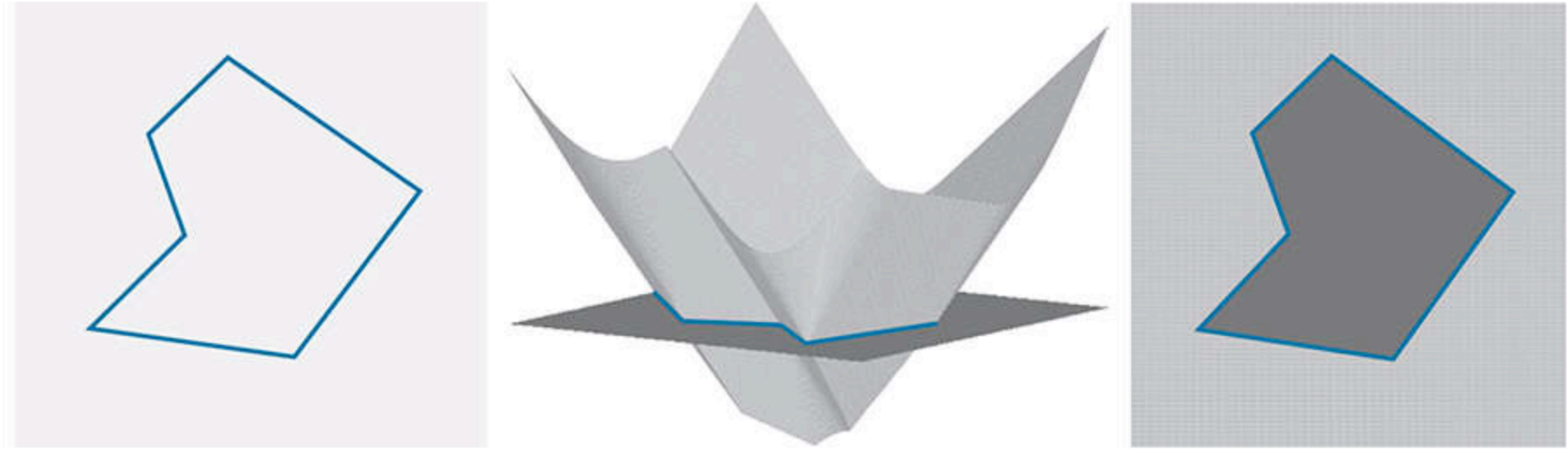
  - The fact that the value of $\phi(x, y)$ for any point (x, y) not on the interface is equal to the shortest distance from (x, y) to the interface implies that:

$$\|\nabla \phi(x, y)\| = 1$$

Properties of signed distance functions. Regions Ω_1^- and Ω_2^- are regions enclosed by the zero level sets (i.e. boundaries) of signed distance functions $\phi_1(x, y)$ and $\phi_2(x, y)$, respectively.

Property	Description
1) Unit gradient magnitude.	$\\|\nabla\phi(x, y)\\| = 1$
2) Unit normal to the boundary at point (x, y).	$\mathbf{n} = \frac{\nabla\phi(x, y))}{\\|\nabla\phi(x, y)\\|} = \nabla\phi(x, y)$
3) Mean curvature (equal to the Laplacian).	$\kappa = \nabla \cdot \left(\frac{\nabla\phi(x, y)}{\\|\nabla\phi(x, y)\\|}\right) = \nabla \cdot \nabla\phi(x, y) = \nabla^2\phi(x, y) = \text{Laplaci a}[\phi(x, y)]$
4) Point (x_B, y_B) on the boundary closest to an arbitrary point (x, y) on the plane (see **Fig. 11.14**).	$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \phi(x, y)\mathbf{n}$ where **n** is the unit normal from Property 2.
5) Convexity.	If Ω^- is convex, then its corresponding signed distance function, $\phi(x, y)$, is a convex function. (See **Section 9.5** regarding convexity.)

Examples:



a b c

(a) An arbitrarily specified, initial zero level set (interface) function. (b) Resulting signed distance function obtained with $\Delta = 0.5$and 100 iterations of Eq. (11-92) , starting with equal to the interface in (a). (c) Level set function viewed from above. (Figures (b) and (c) are not viewed from the same azimuth angle—(b) is viewed from an angle that shows the most detail in the 3-D plot.)