Image Segmentation

Fundamentals

- Let R represent the entire spatial region occupied by an image. Image segmentation is a process that partitions R into *n* sub-regions, R_1, R_2, …, R_n, such that

(a) $\bigcup_{i=1}^{n} R_i = R.$

(b) R_i is a connected set. $i = 1, 2, \ldots, n.$

(c) $R_i \cap R_j = \Phi.$

(d) $\mathbb{Q}(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots, n.$

(e) $\mathbb{Q}(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j.

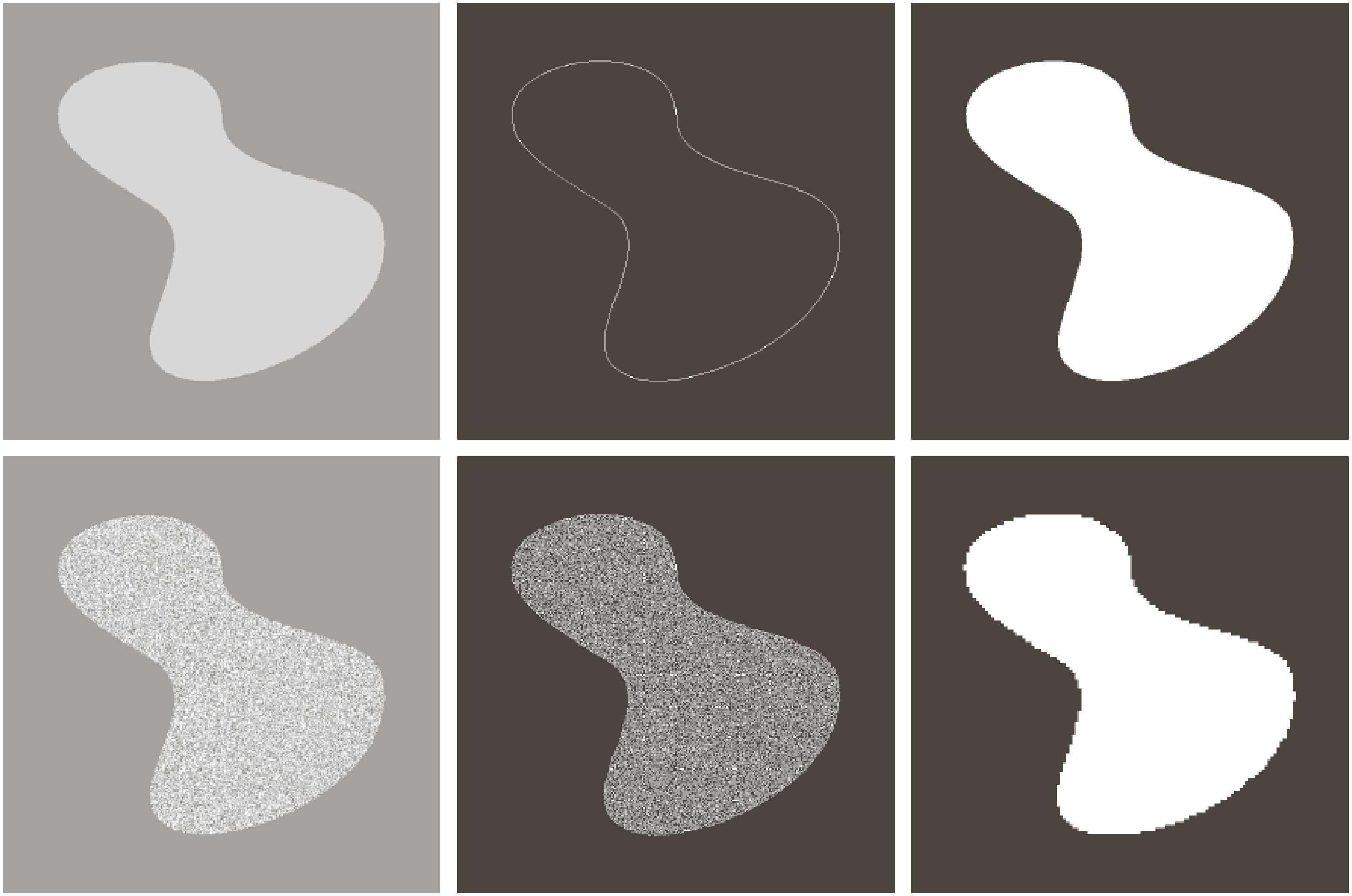


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Background

- First-order derivative

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- Second-order derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

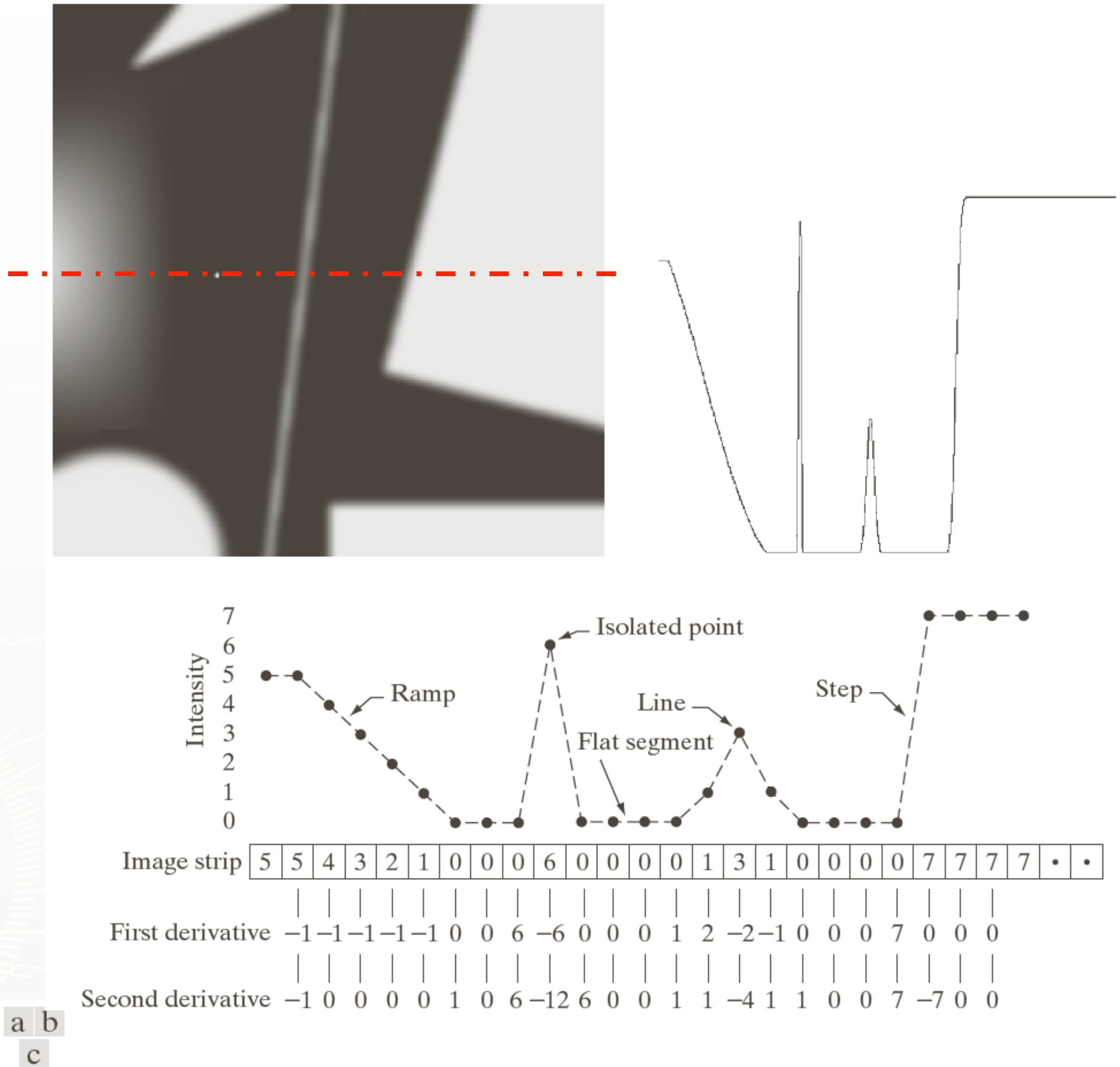


FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Characteristics of First and Second Order Derivatives

- First-order derivatives generally produce thicker edges in image
- Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- Second-order derivatives produce a double-edge response at ramp and step transition in intensity
- The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light

Detection of Isolated Points

- The Laplacian

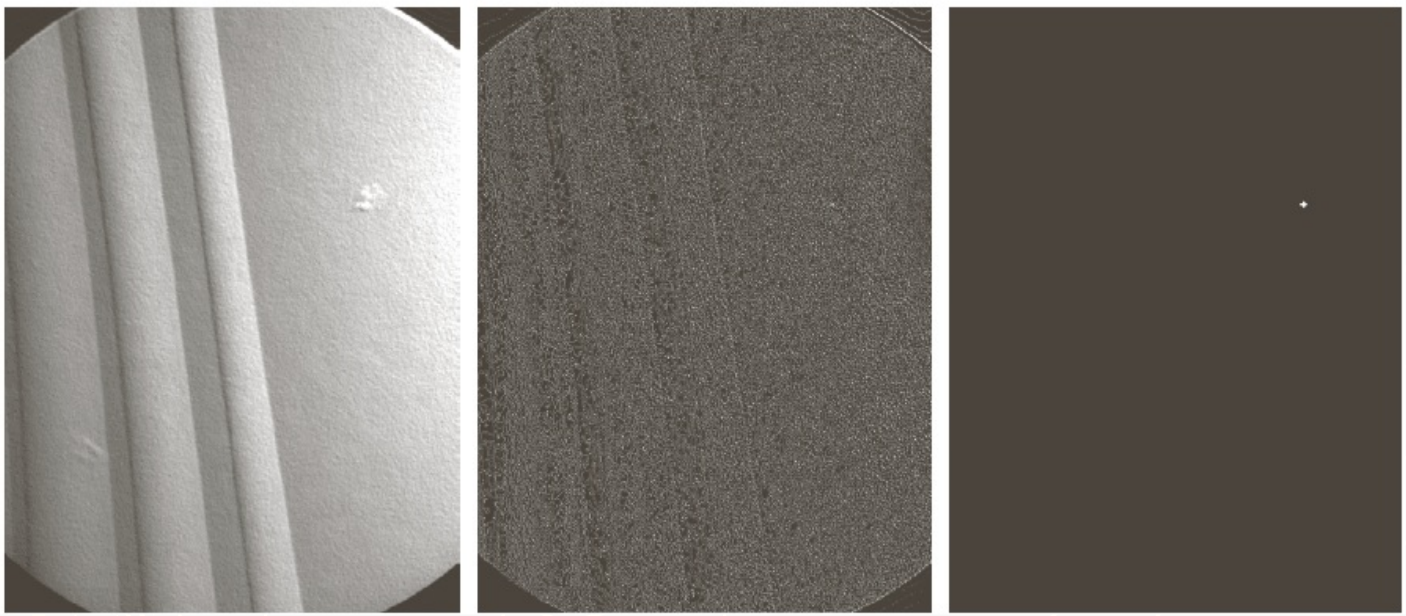
$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)$$

$$-4f(x,y)$$

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases} \qquad R = \sum_{k=1}^{9} w_k z_k$$

1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Line Detection

- Second derivatives to result in a stronger response and to produce thinner lines than first derivatives

- Double-line effect of the second derivative must be handled properly

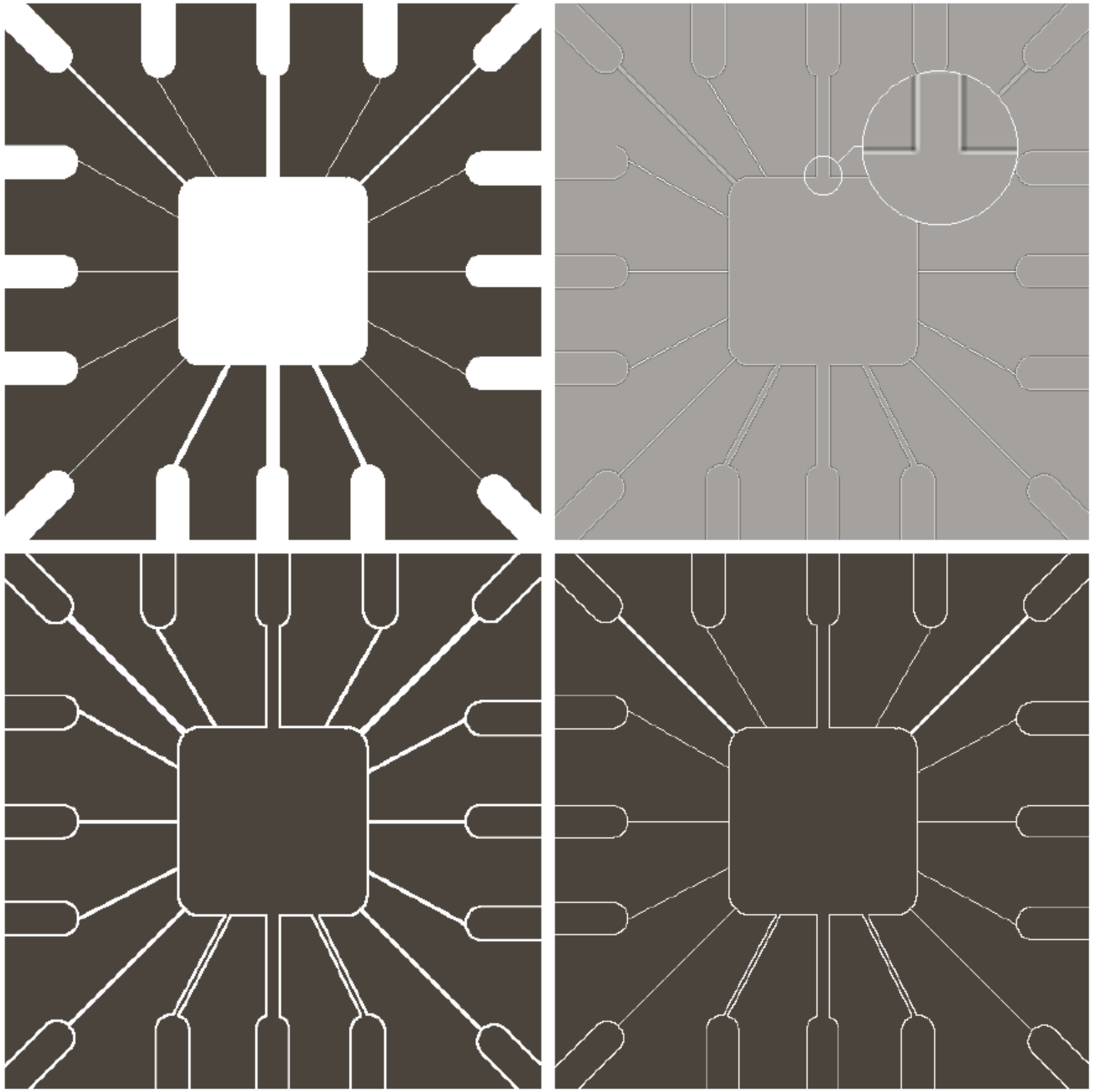


FIGURE 10.5 (a) Original image. (b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian. (c) Absolute value of the Laplacian. (d) Positive values of the Laplacian.

Detecting Line in Specified Directions

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- Let R_1, R_2, R_3, and R_4 denote the responses of the masks in Fig. 10.6. If, at a given point in the image, $|R_k|>|R_j|$, for all $j \neq k$, that point is said to be more likely associated with a line in the direction of mask k.

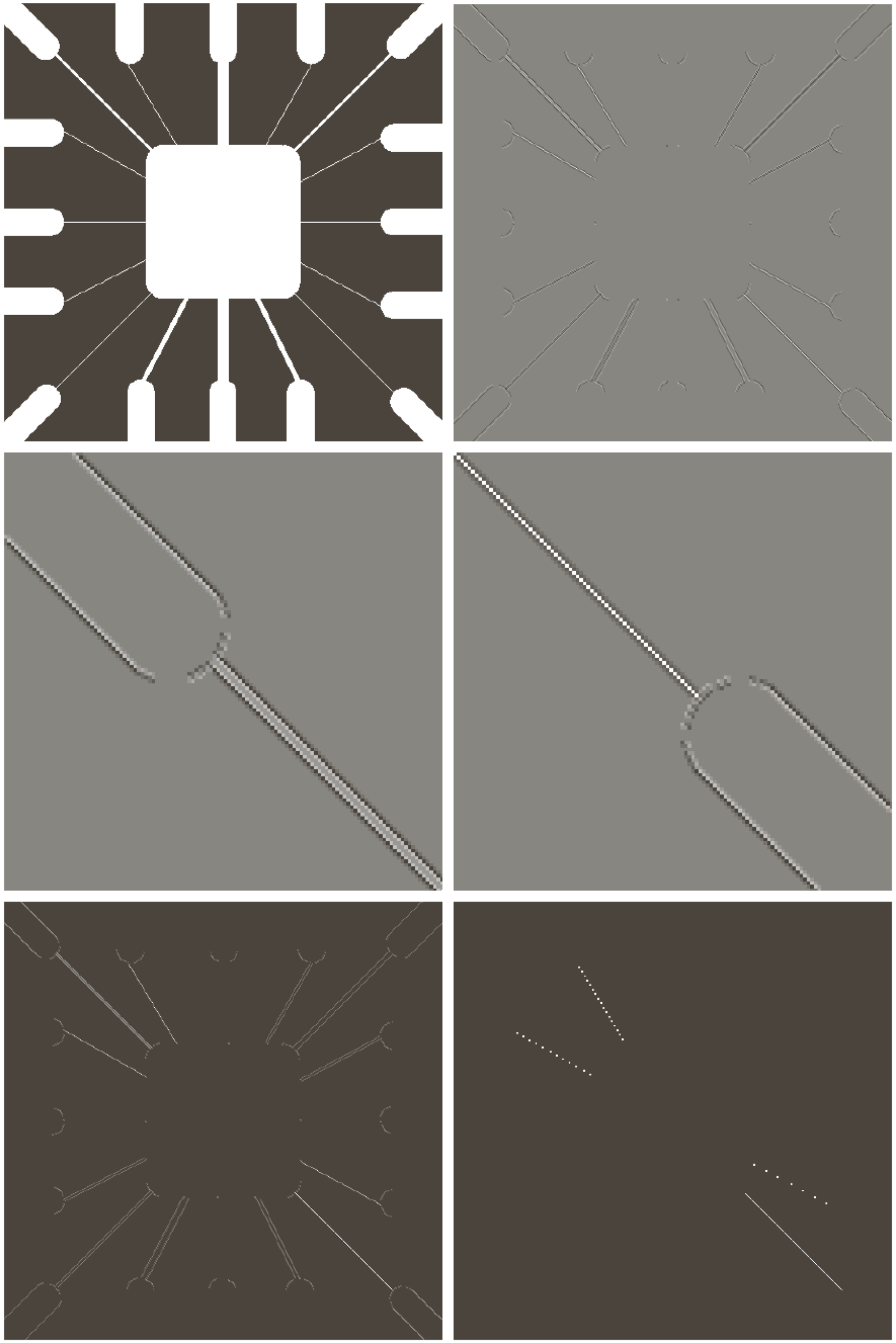
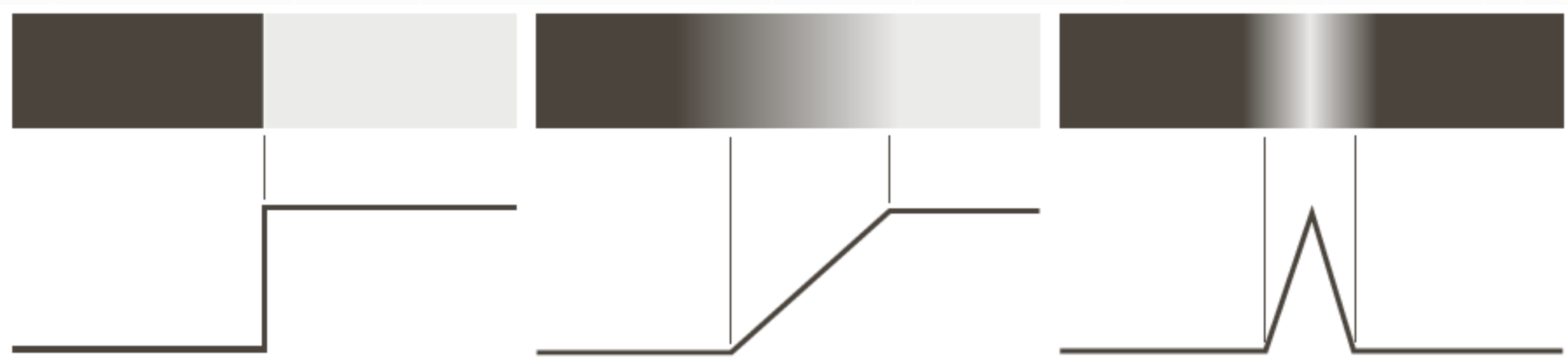


FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection

- Edges are pixels where the brightness function changes abruptly
- Edge models



a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

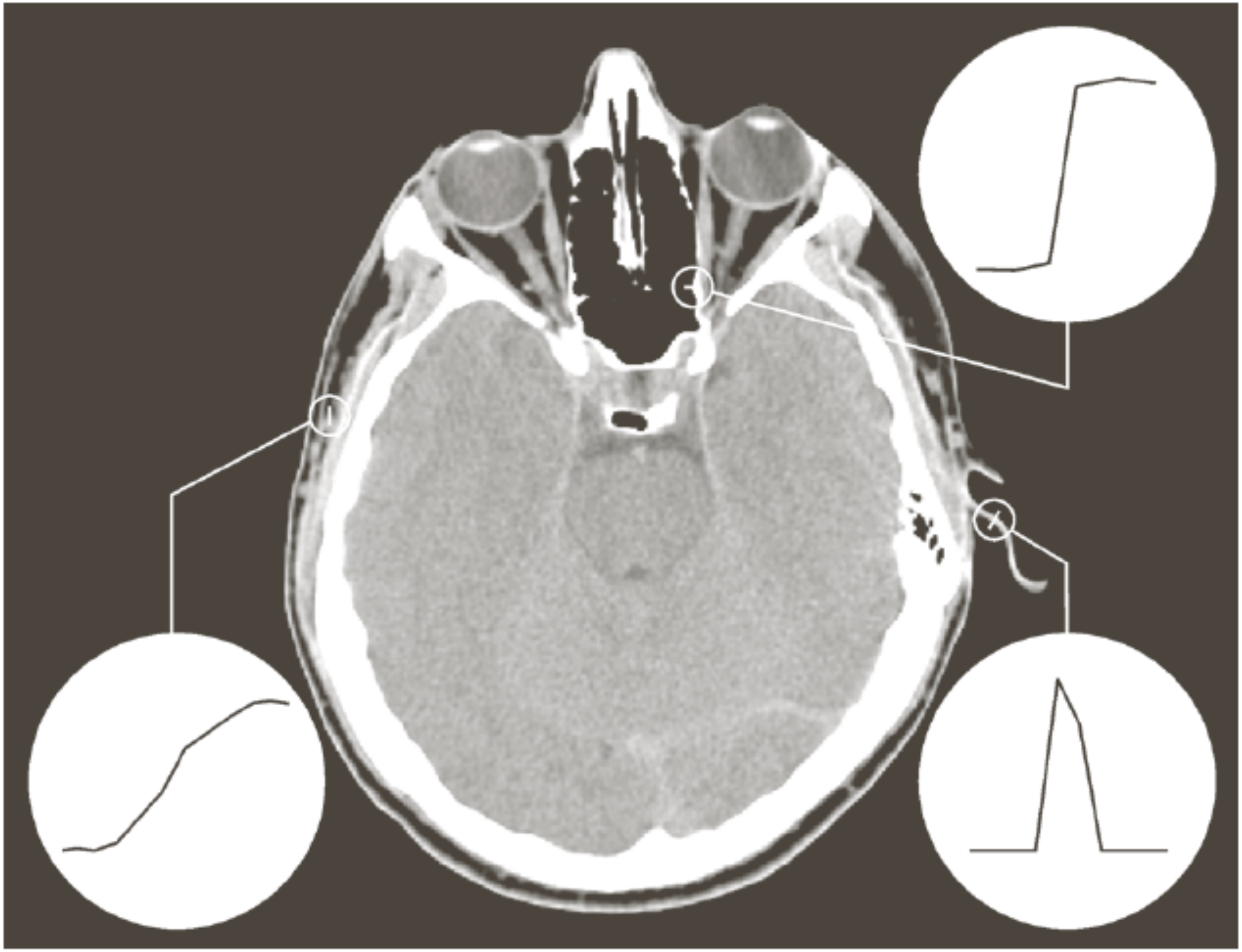
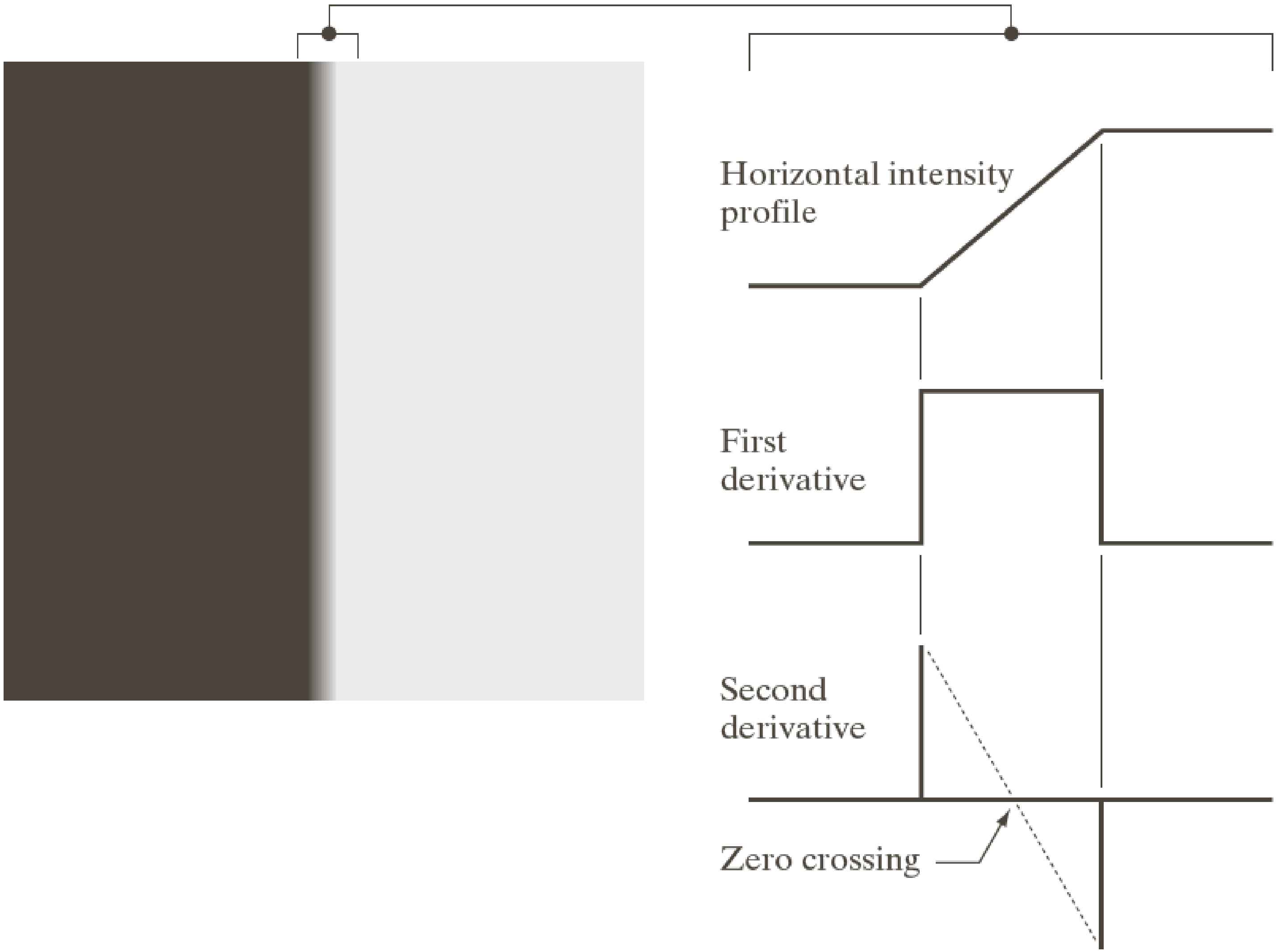


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)



a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal vertical ramp edge.
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

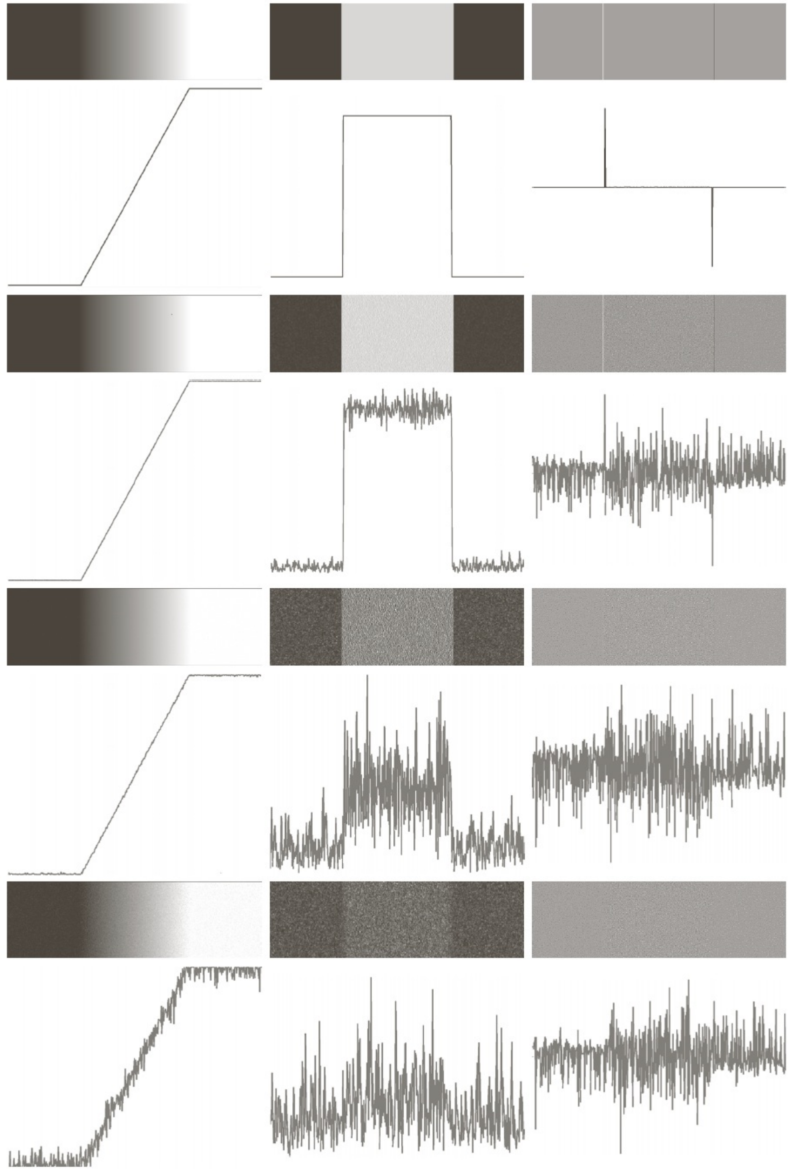


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Basic Edge Detection by Using First-Order Derivative

$$\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of ∇f

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\ 2} + g_y^{\ 2}}$$

The direction of ∇f

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_x}{g_y}\right]$$

The direction of the edge

$$\phi = \alpha - 90^{\text{o}}$$

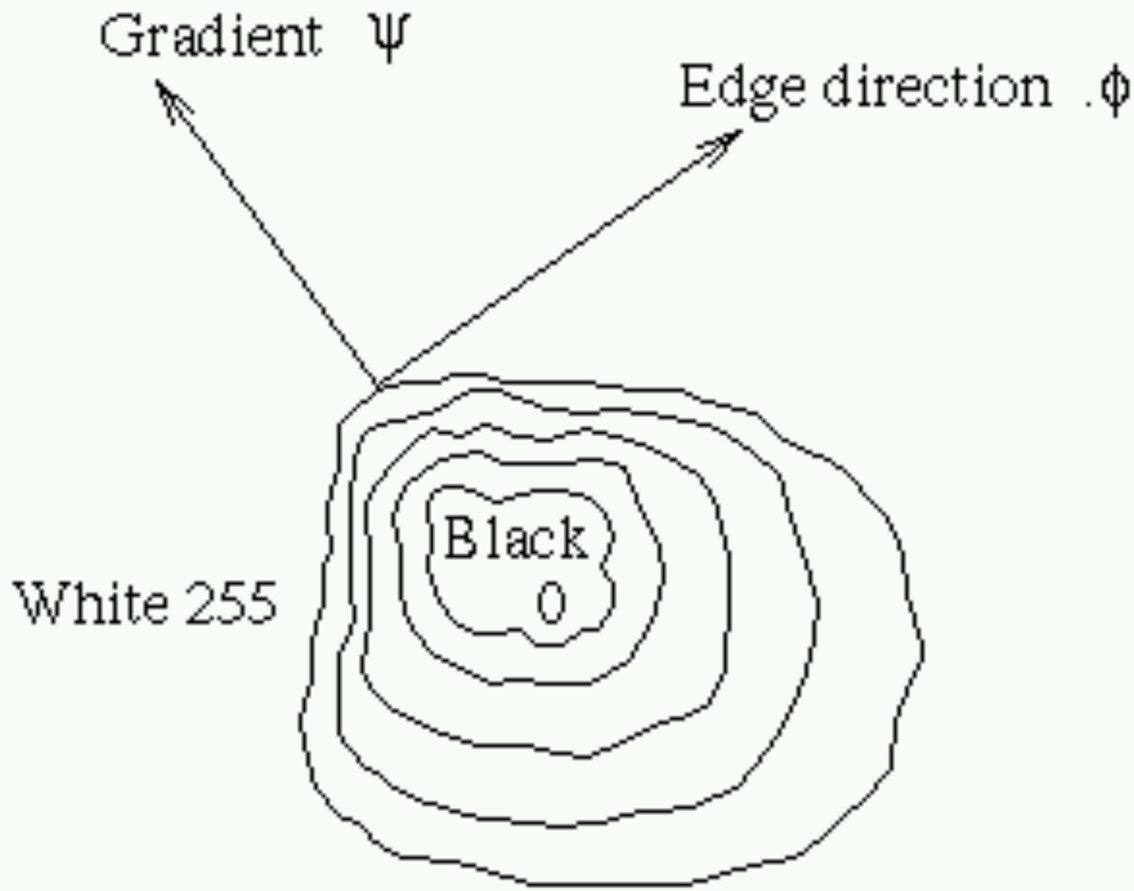
Basic Edge Detection by Using First-Order Derivative

Edge normal: $\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$

Edge unit normal: $\nabla f \,/\, \mathrm{mag}(\nabla f)$

In practice,sometimes the magnitude is approximated by

$$\mathrm{mag}(\nabla f)=\left|\frac{\partial f}{\partial x}\right|+\left|\frac{\partial f}{\partial y}\right| \text{ or } \mathrm{mag}(\nabla f)=\max\left(\left|\frac{\partial f}{\partial x}\right|,\left|\frac{\partial f}{\partial y}\right|\right)$$


Gradient ψ
Edge direction .ϕ
Black
0
White 255

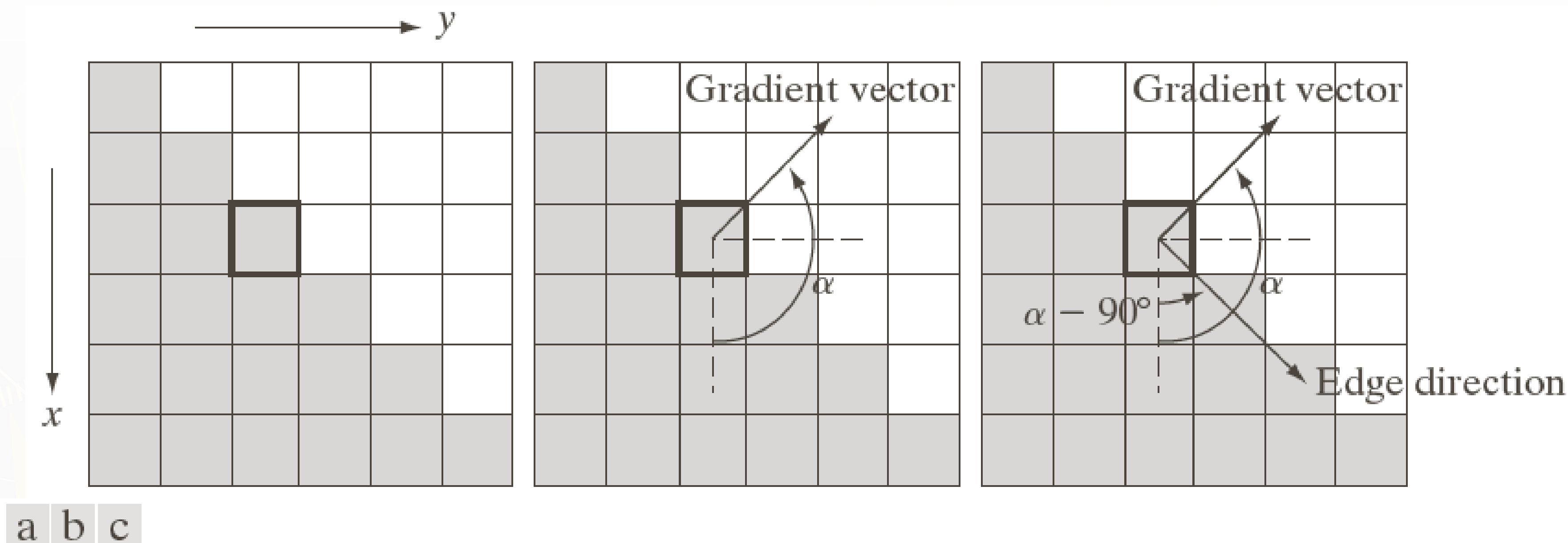


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

−1
1

−1	1

a b

FIGURE 10.13 One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

a
b c
d e
f g

FIGURE 10.14 A 3 × 3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

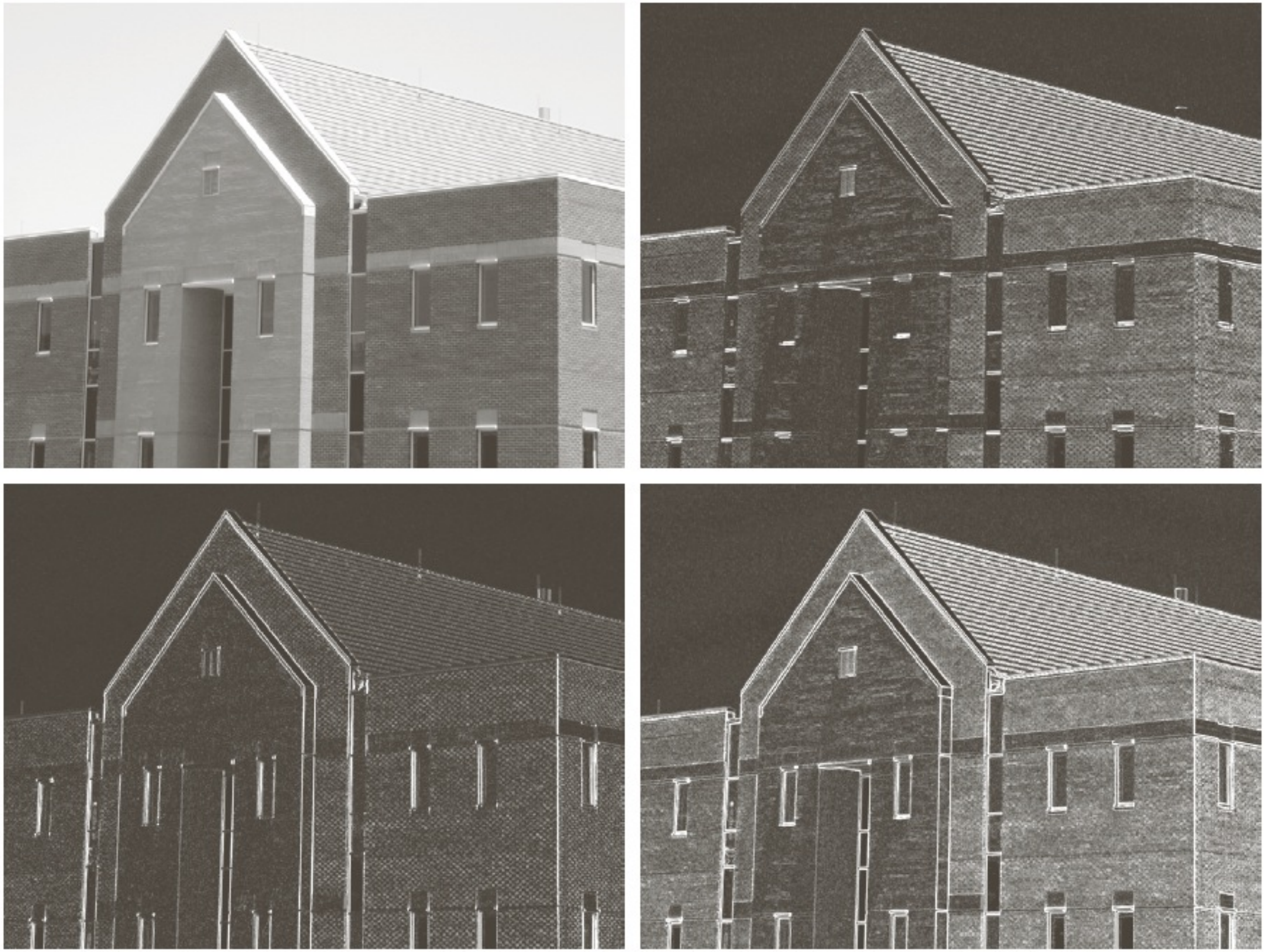
0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a	b
c	d

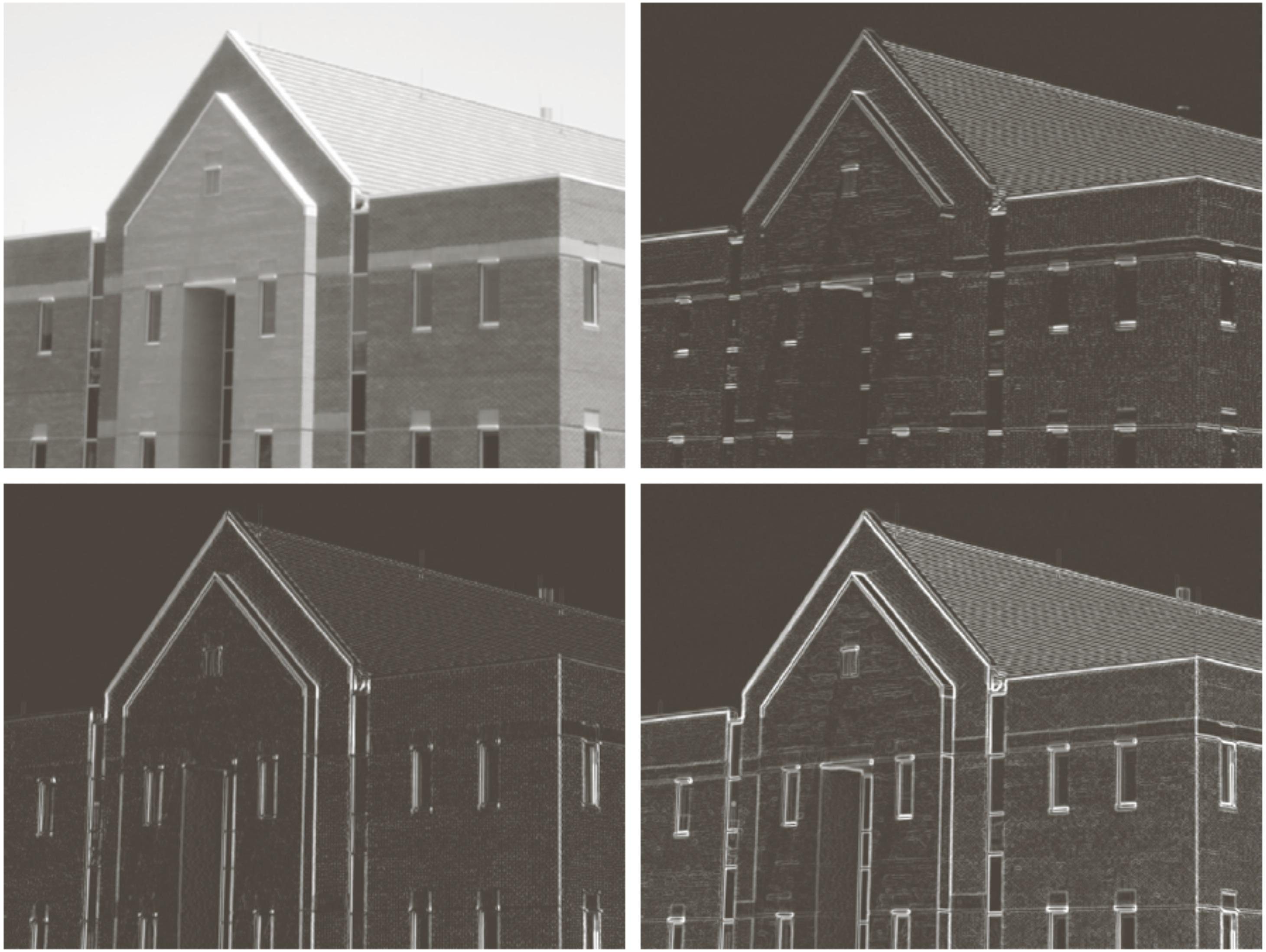
FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.



a	b
c	d

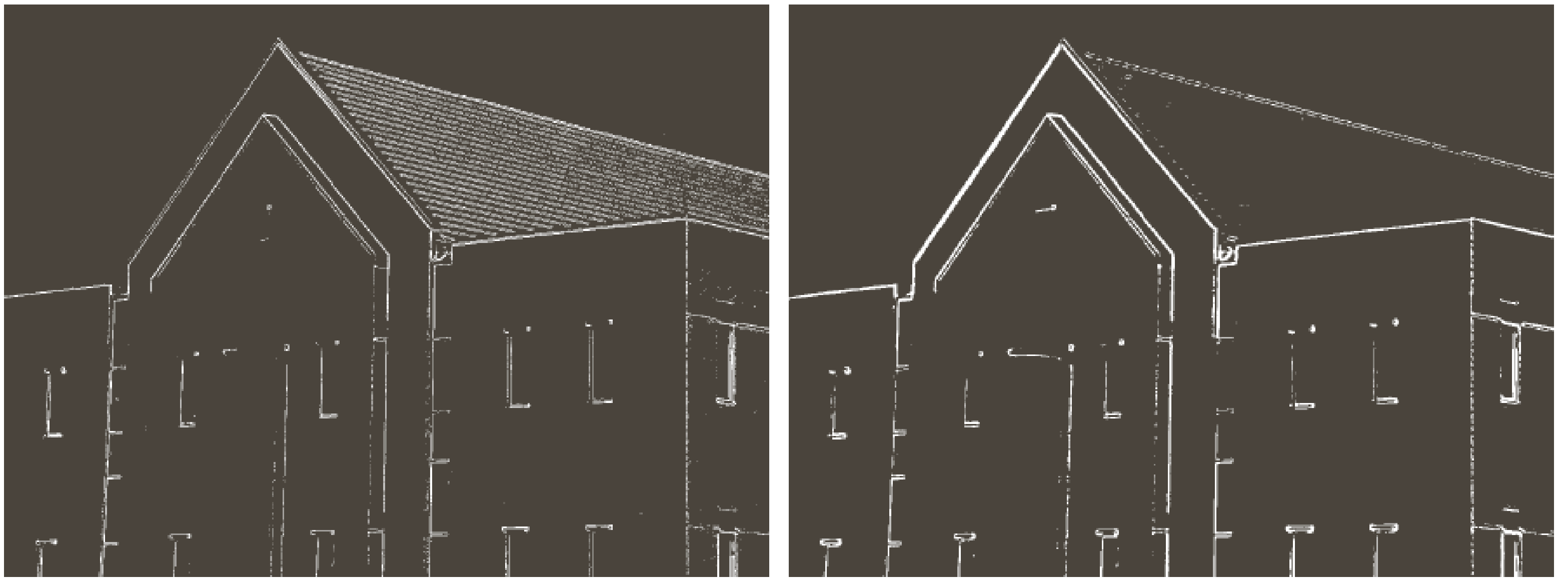
FIGURE 10.16
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.



a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.

a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Edge Linking and Boundary Detection

- Edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries

- Three approaches to edge linking

  Local processing

  Regional processing

  Global processing

Local Processing

- Analyze the characteristics of pixels in a small neighborhood about every point (x,y) that has been declared an edge point

- All points that similar according to predefined criteria are linked, forming an edge of pixels.

  Establishing similarity: (1) the strength (magnitude) and (2) the direction of the gradient vector.

  A pixel with coordinates (s,t) in S_{xy} is linked to the pixel at (x,y) if both magnitude and direction criteria are satisfied.

Local Processing

Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image. An edge pixel with coordinate (s,t) in S_{xy} is similar in *magnitude* to the pixel at (x, y) if

$$\left| M(s,t) - M(x, y) \right| \le E$$

An edge pixel with coordinate (s,t) in S_{xy} is similar in *angle* to the pixel at (x, y) if

$$\left| \alpha(s,t) - \alpha(x, y) \right| \le A$$

Local Processing: Steps (1)

1. Compute the gradient magnitude and angle arrays, M(x,y) and $\alpha(x, y)$, of the input image f(x,y)

3. Form a binary image, g, whose value at any pair of coordinates (x,y) is given by

$$g(x, y) = \begin{cases} 1 & \text{if } M(x, y) > T_M \text{ and } \alpha(x, y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

T_M : threshold $\quad A$: specified angle direction

T_A: a "band" of acceptable directions about A

Local Processing: Steps (2)

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K.

5. To detect gaps in any other direction, rotate g by this angle and apply the horizontal scanning procedure in step 3.

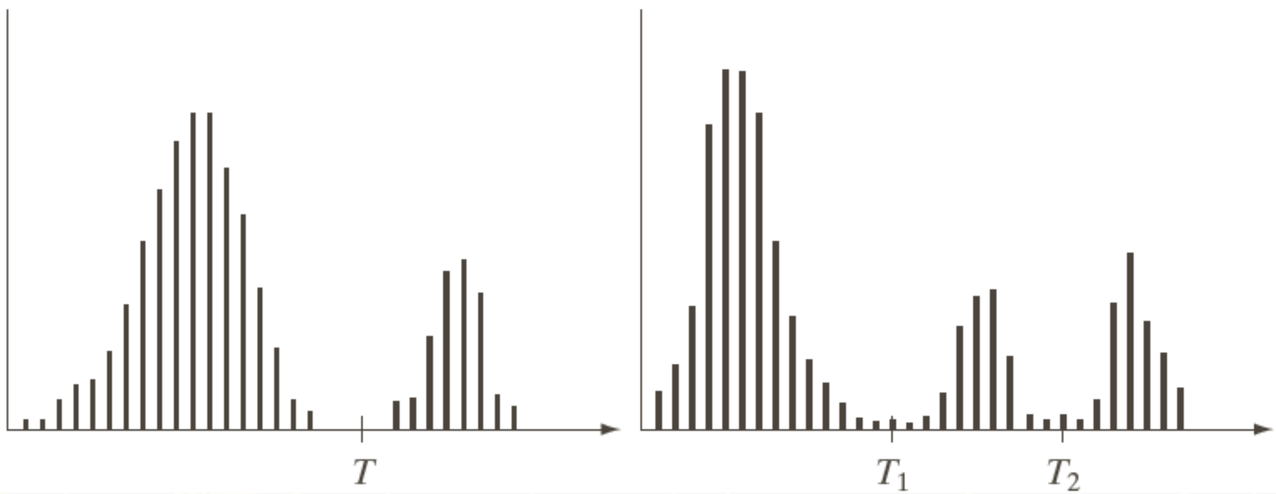
Thresholding

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T \quad \text{(object point)}\\ 0 & \text{if } f(x,y)\le T \quad \text{(background point)}\end{cases}$$

T : global thresholding

Multiple thresholding

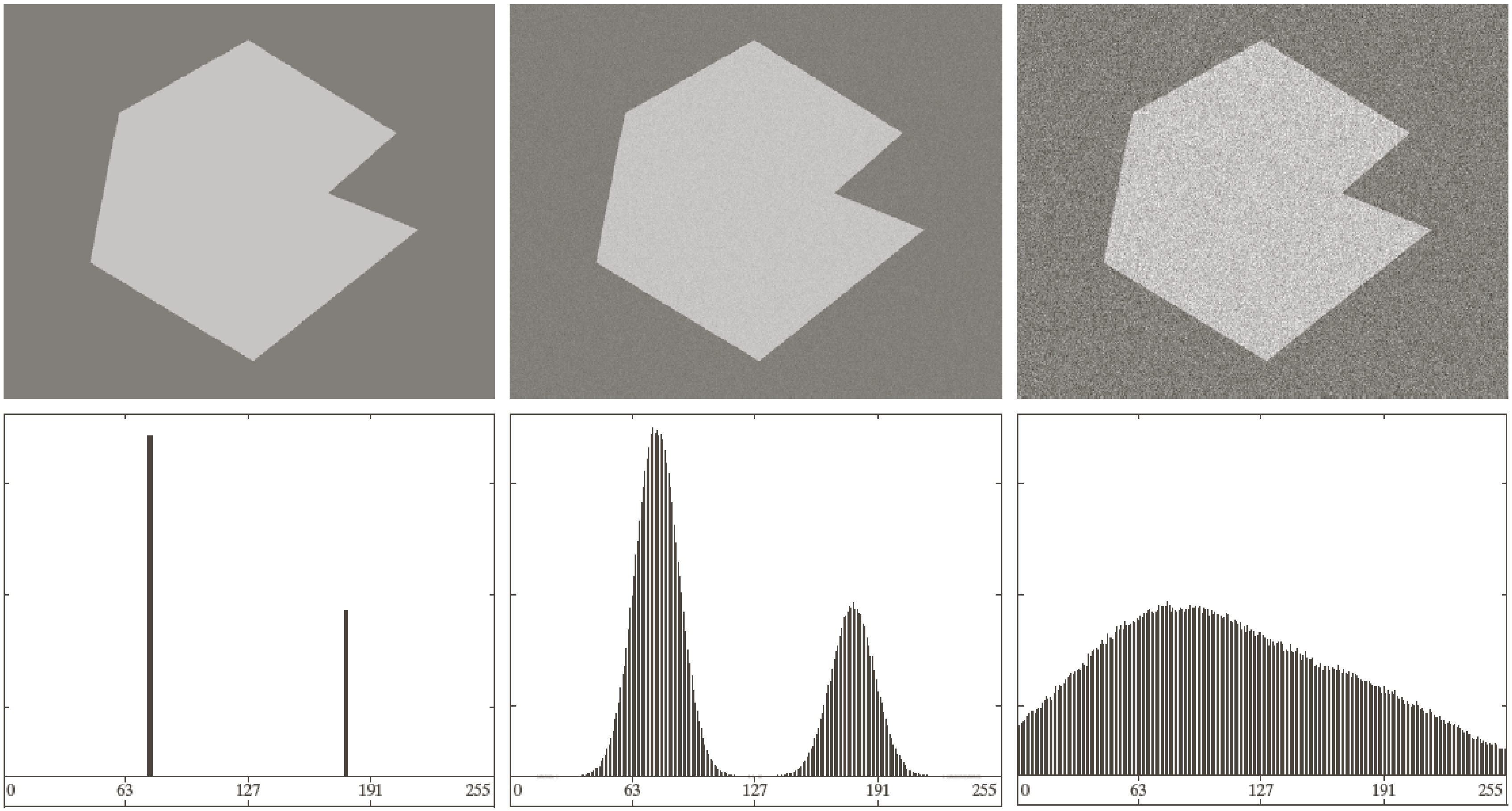
$$g(x,y)=\begin{cases}a & \text{if } f(x,y)>T_2\\ b & \text{if } T_1< f(x,y)\le T_2\\ c & \text{if } f(x,y)\le T_1\end{cases}$$



a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

The Role of Noise in Image Thresholding



a	b	c
d	e	f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

The Role of Illumination and Reflectance

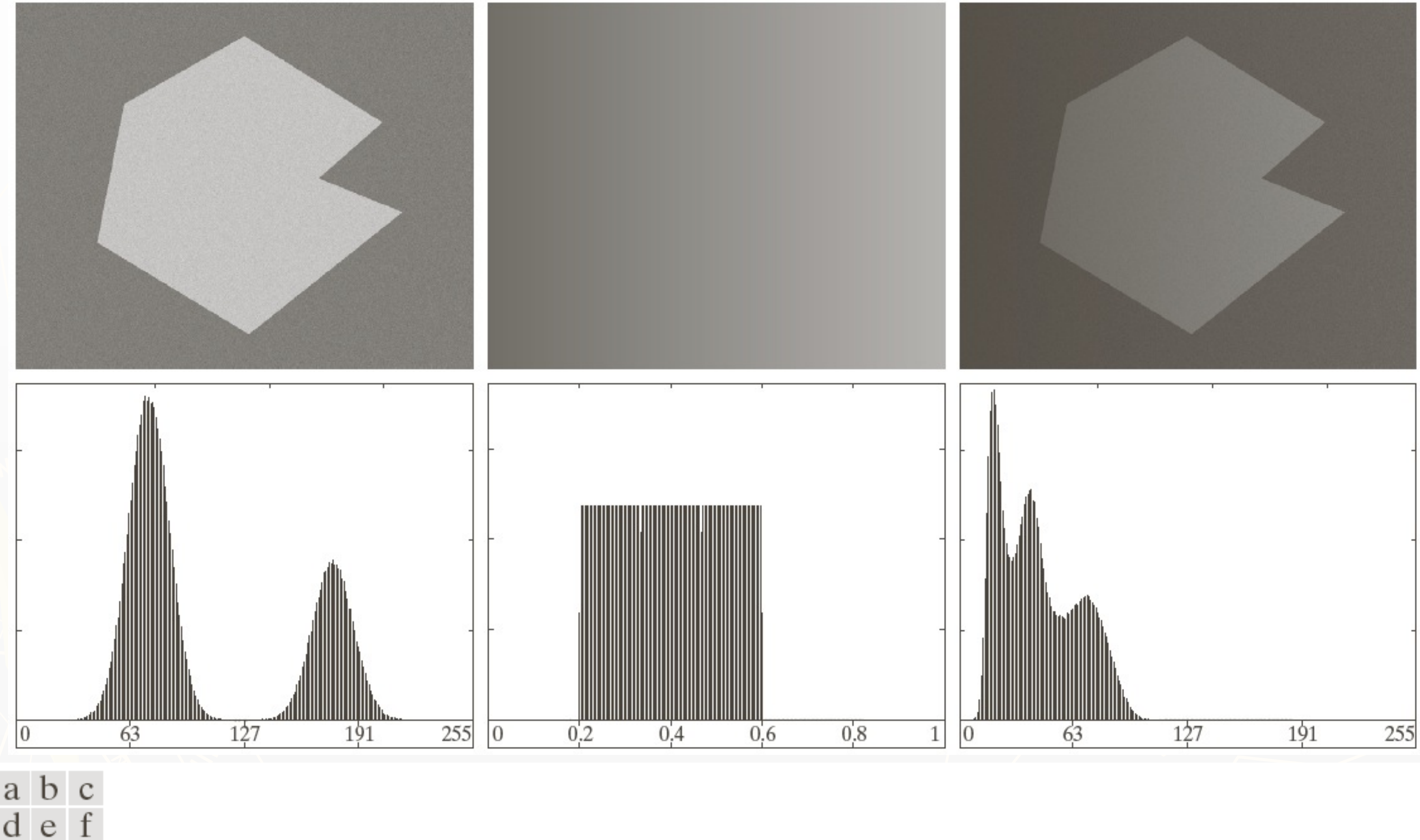


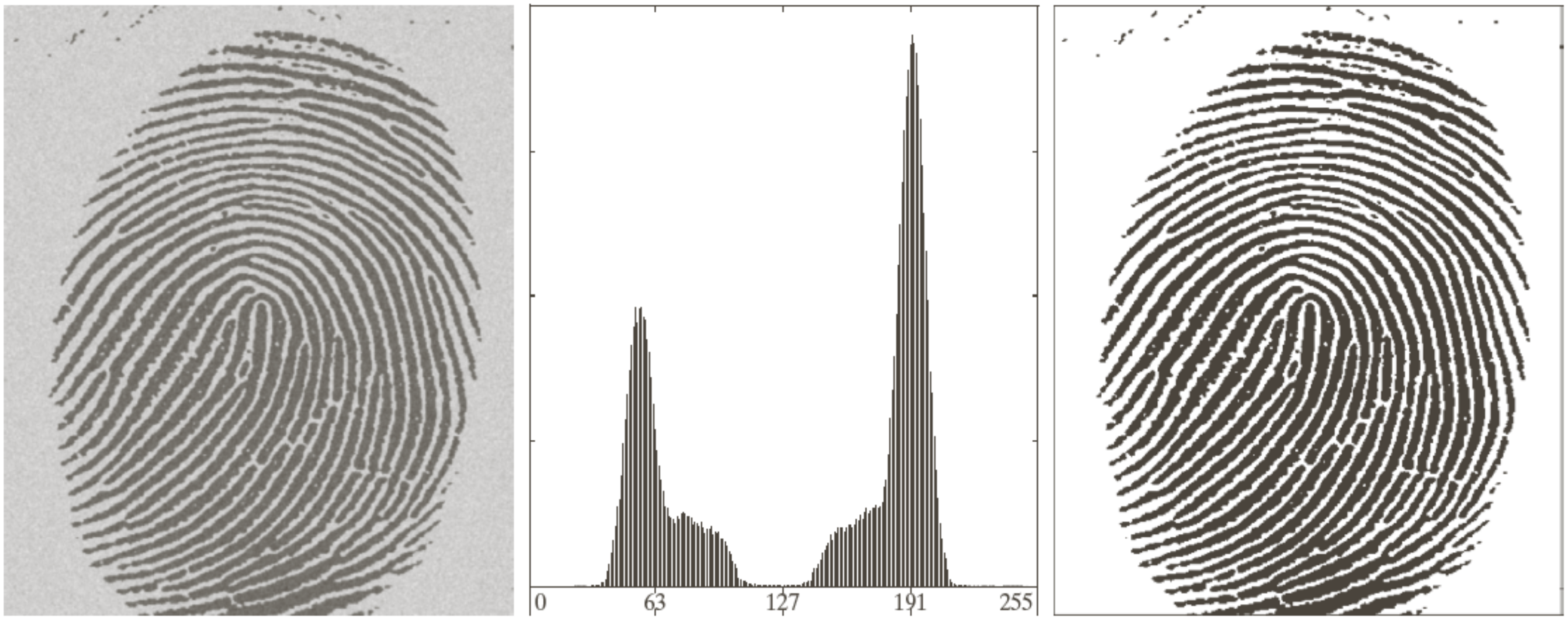
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Basic Global Thresholding

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T. It will produce two groups of pixels: G1 consisting of all pixels with intensity values > T and G2 consisting of pixels with values $\leq$ T.
3. Compute the average intensity values m1 and m2 for the pixels in G1 and G2, respectively.
4. Compute a new threshold value.

$$T = \frac{1}{2}\left(m1 + m2\right)$$

5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT .

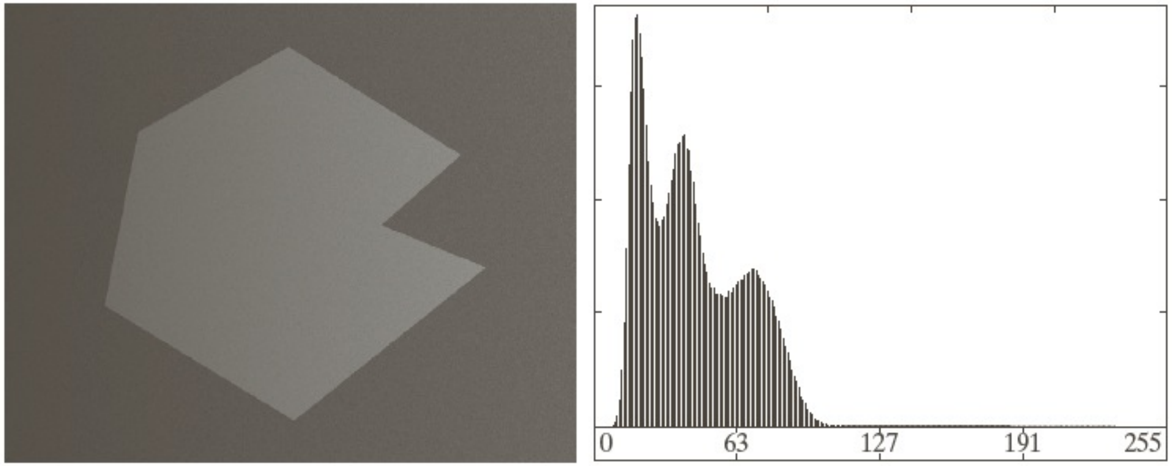


a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Variable Thresholding: Image Partitioning

- Subdivide an image into nonoverlapping rectangles
- The rectangles are chosen small enough so that the illumination of each is approximately uniform.



a	b	c
d	e	f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

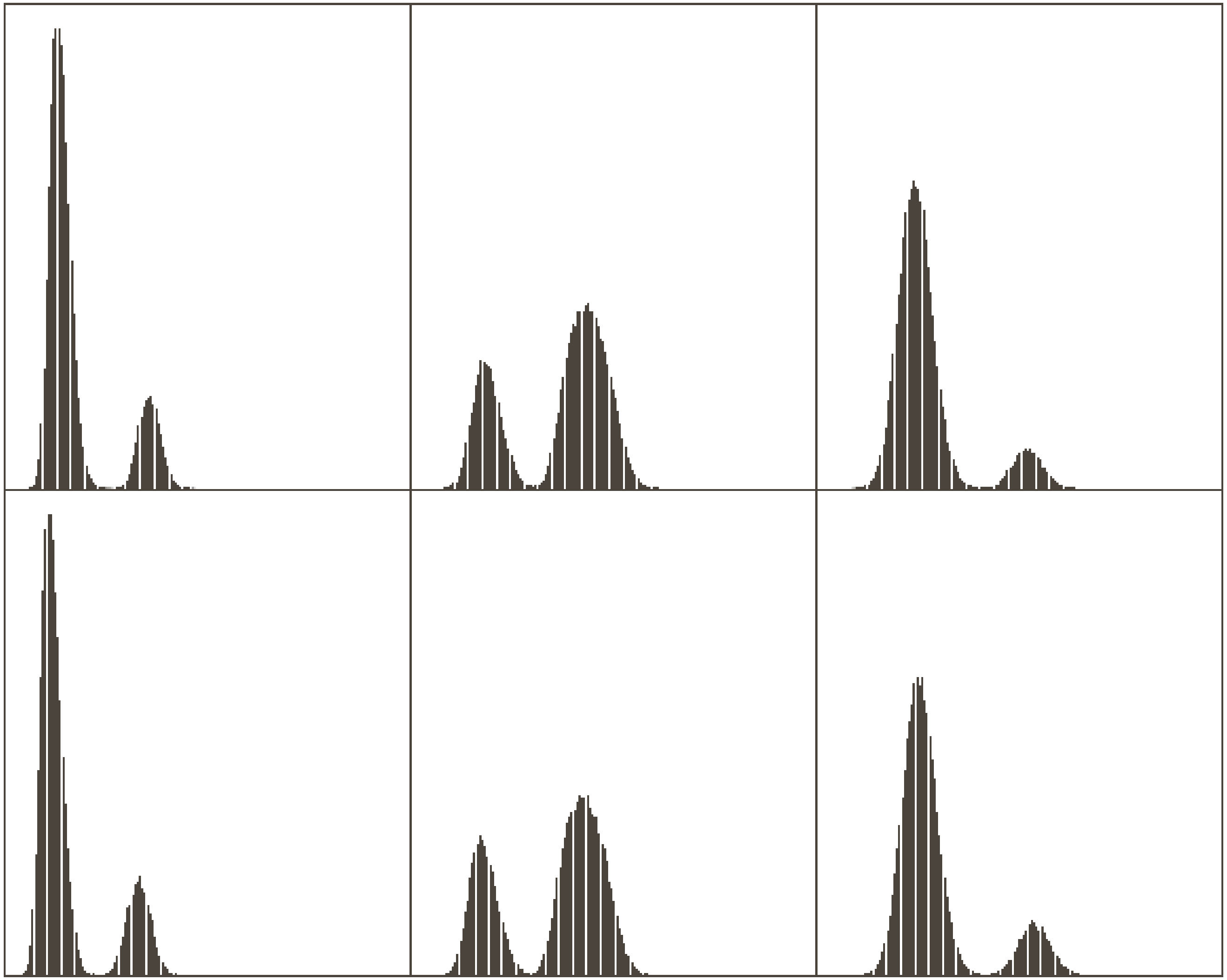


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

Variable Thresholding Based on Local Image Properties

Let σ_{xy} and m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood S_{xy}, centered at coordinates (x, y) in an image. The local thresholds,

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

If the background is nearly constant,

$$T_{xy} = a\sigma_{xy} + bm$$

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T_{xy} \\ 0 & \text{if } f(x,y) \le T_{xy} \end{cases}$$

Variable Thresholding Based on Local Image Properties

A modified thresholding

$$g(x, y) = \begin{cases} 1 & \text{if } Q(\text{local parameters}) \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

e.g.,

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{true} & \text{if } f(x, y) > a\sigma_{xy} \text{ AND } f(x, y) > bm_{xy} \\ \text{false} & \text{otherwise} \end{cases}$$

a	b
c	d

FIGURE 10.48
(a) Image from Fig. 10.43.
(b) Image segmented using the dual thresholding approach discussed in Section 10.3.6.
(c) Image of local standard deviations.
(d) Result obtained using local thresholding.

a=30
b=1.5
$m_{xy} = m_G$

Region-Based Segmentation

- Region Growing

1. Region growing is a procedure that groups pixels or subregions into larger regions.

2. The simplest of these approaches is ***pixel aggregation***, which starts with a set of “seed” points and from these grows regions by appending to each seed points those **neighboring pixels** that have **similar properties** (such as gray level, texture, color, shape).

3. Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

Region-Based Segmentation

Example: Region Growing based on 8-connectivity

$f(x, y)$: input image array

$S(x, y)$: seed array containing 1s (seeds) and 0s

$Q(x, y)$: predicate

Region Growing based on 8-connectivity

1. Find all connected components in $S(x, y)$ and erode each connected components to one pixel; label all such pixels found as 1. All other pixels in S are labeled 0.

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

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

4-connectivity

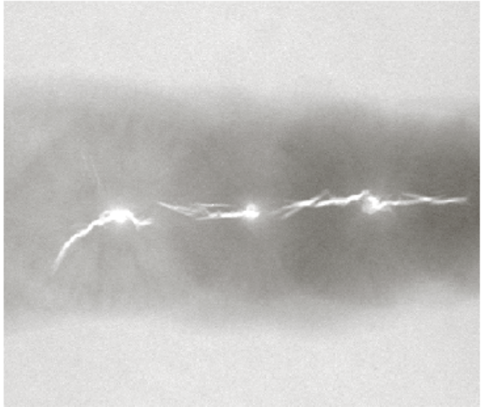
Suppose that we have the image given below.

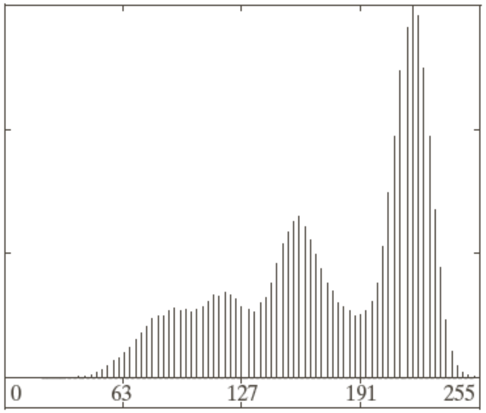
(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

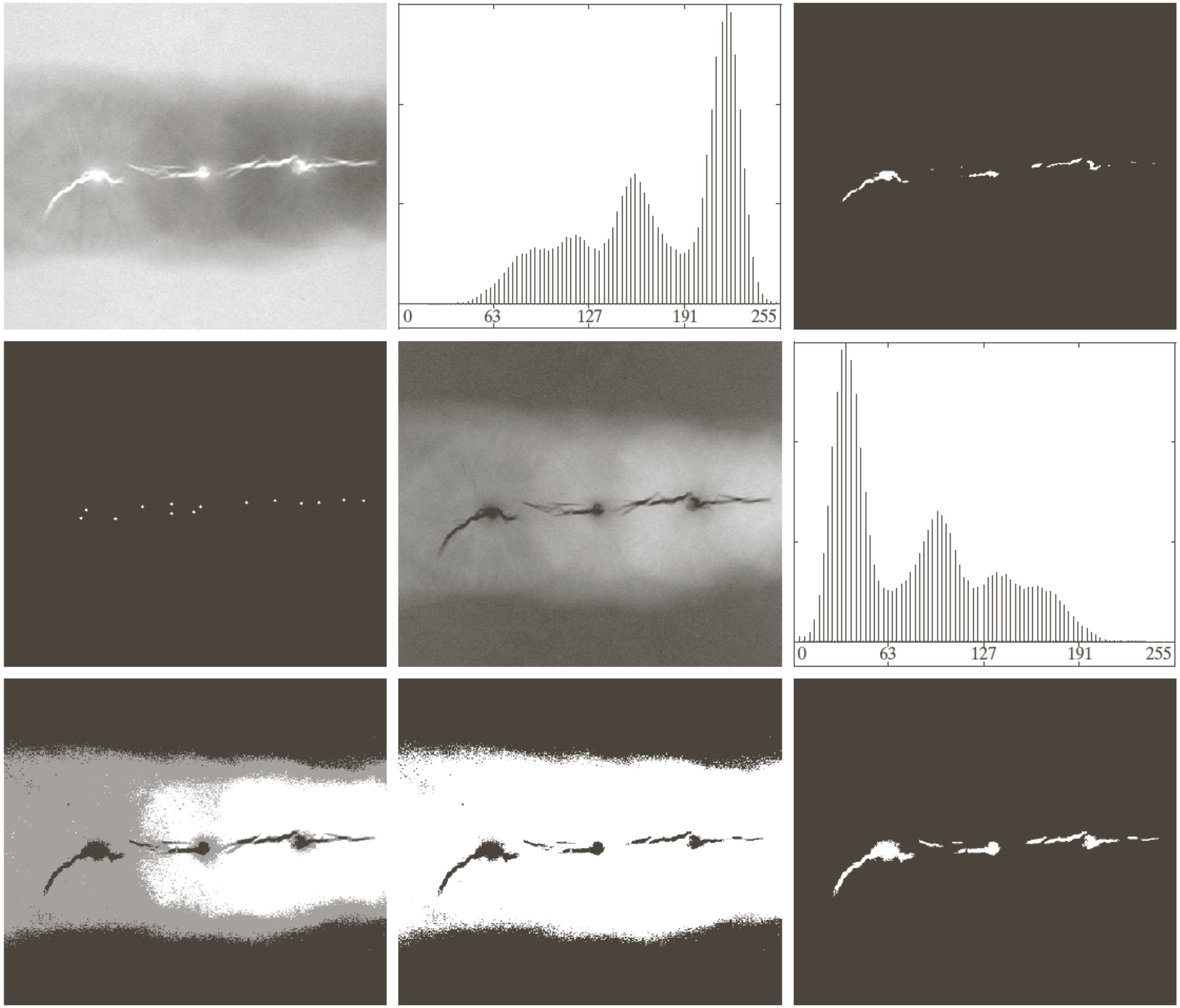
Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

8-connectivity




0
63
127
191
255

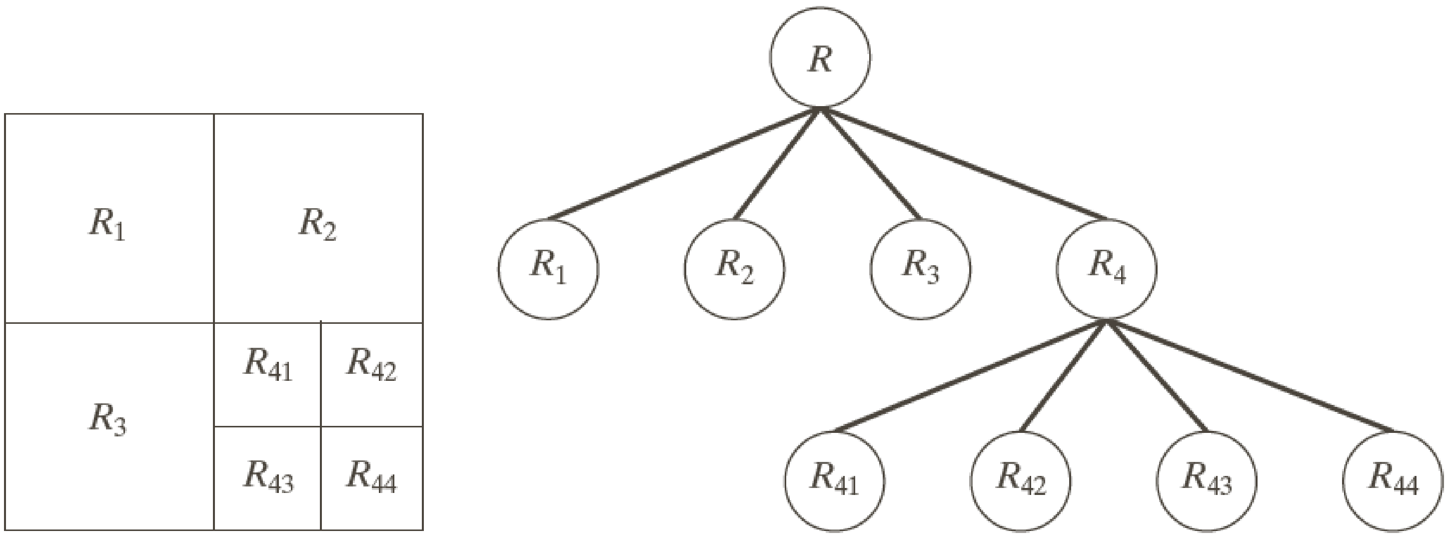


a	b	c
d	e	f
g	h	i

FIGURE 10.51 (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 (a) and (c). (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. (Original image courtesy of X-TEK Systems, Ltd.)

Region Splitting and Merging

R : entire image R_i :entire image Q: predicate



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

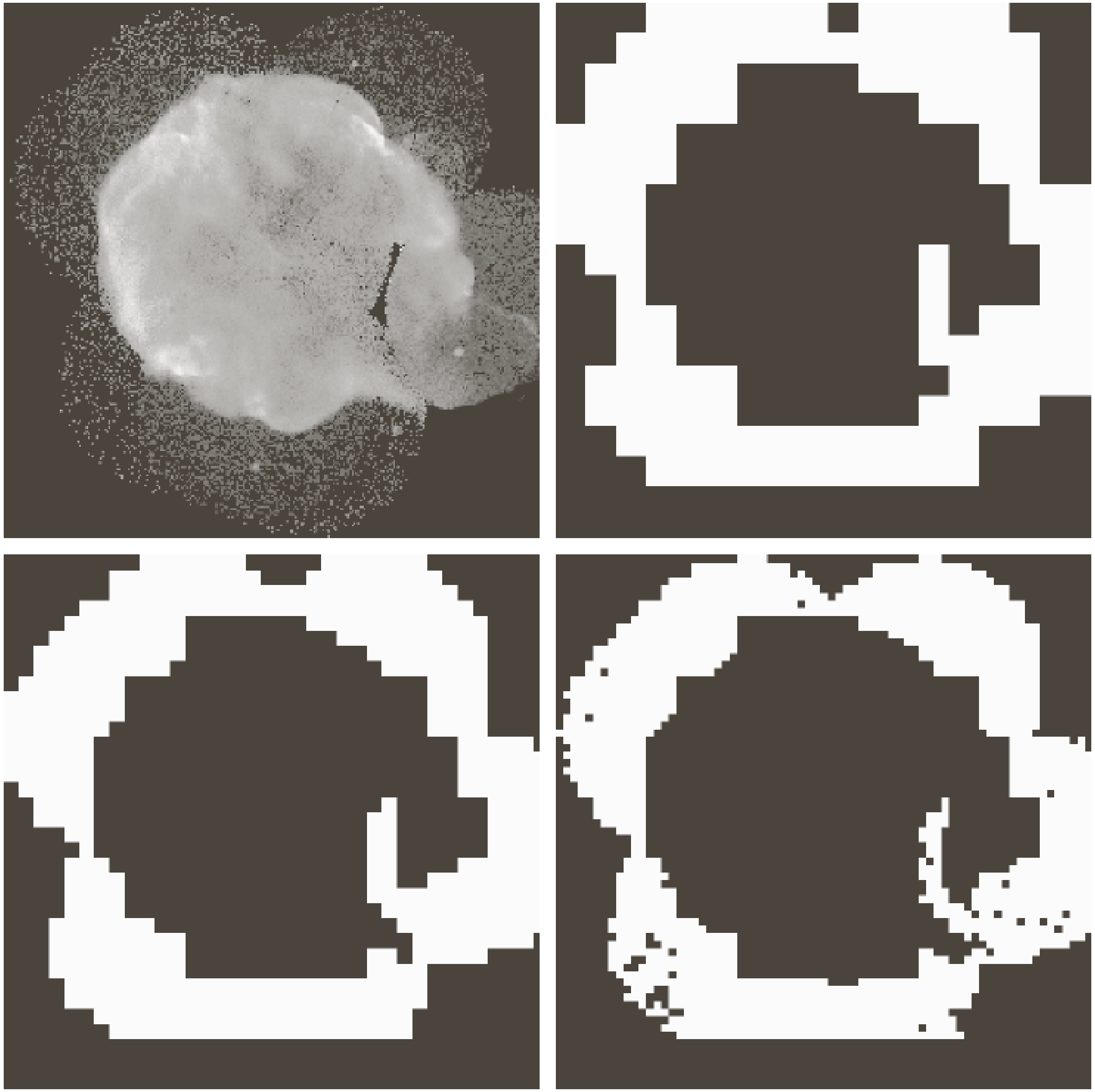


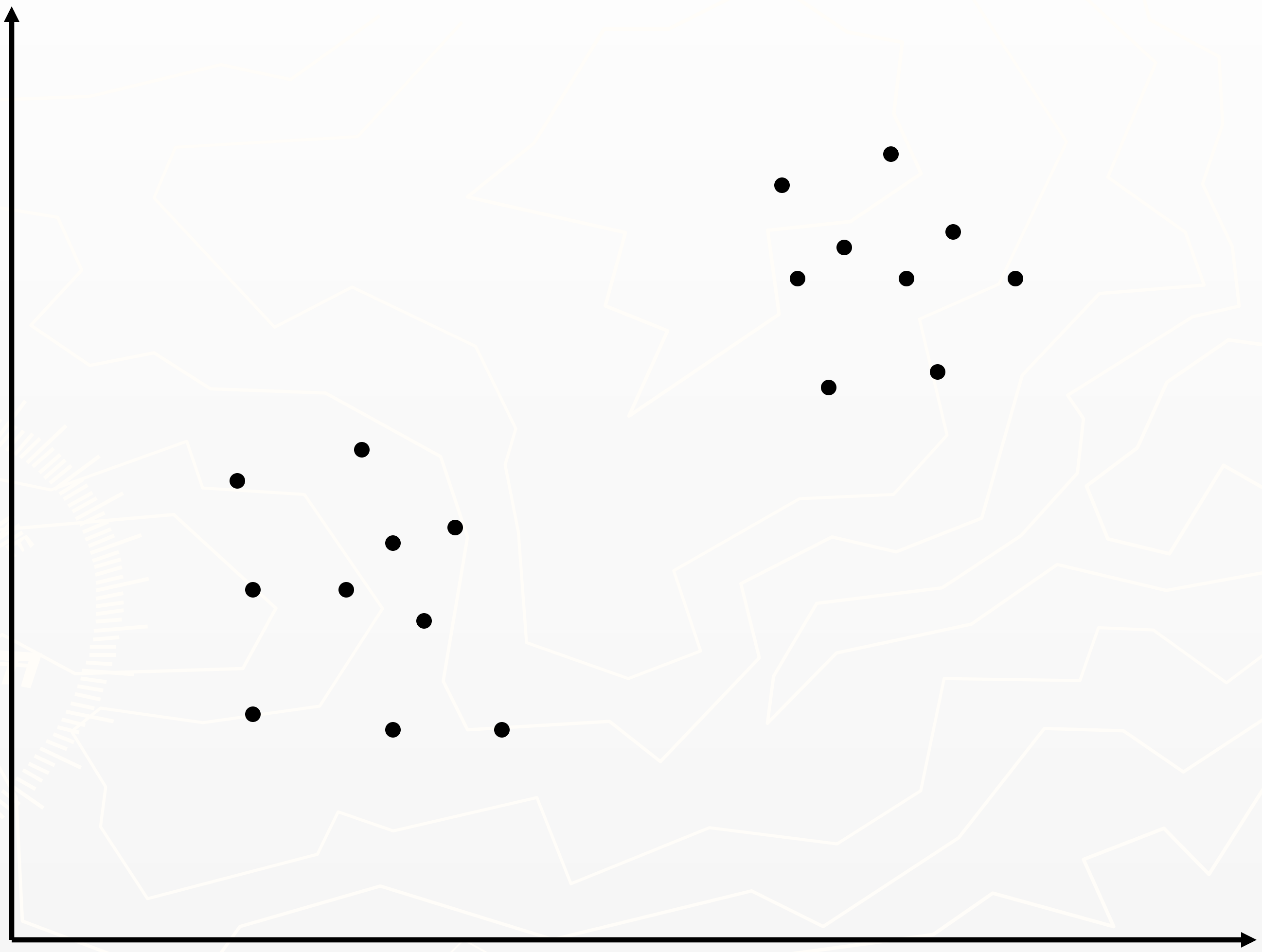
FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

$$Q = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ and } 0 < m < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

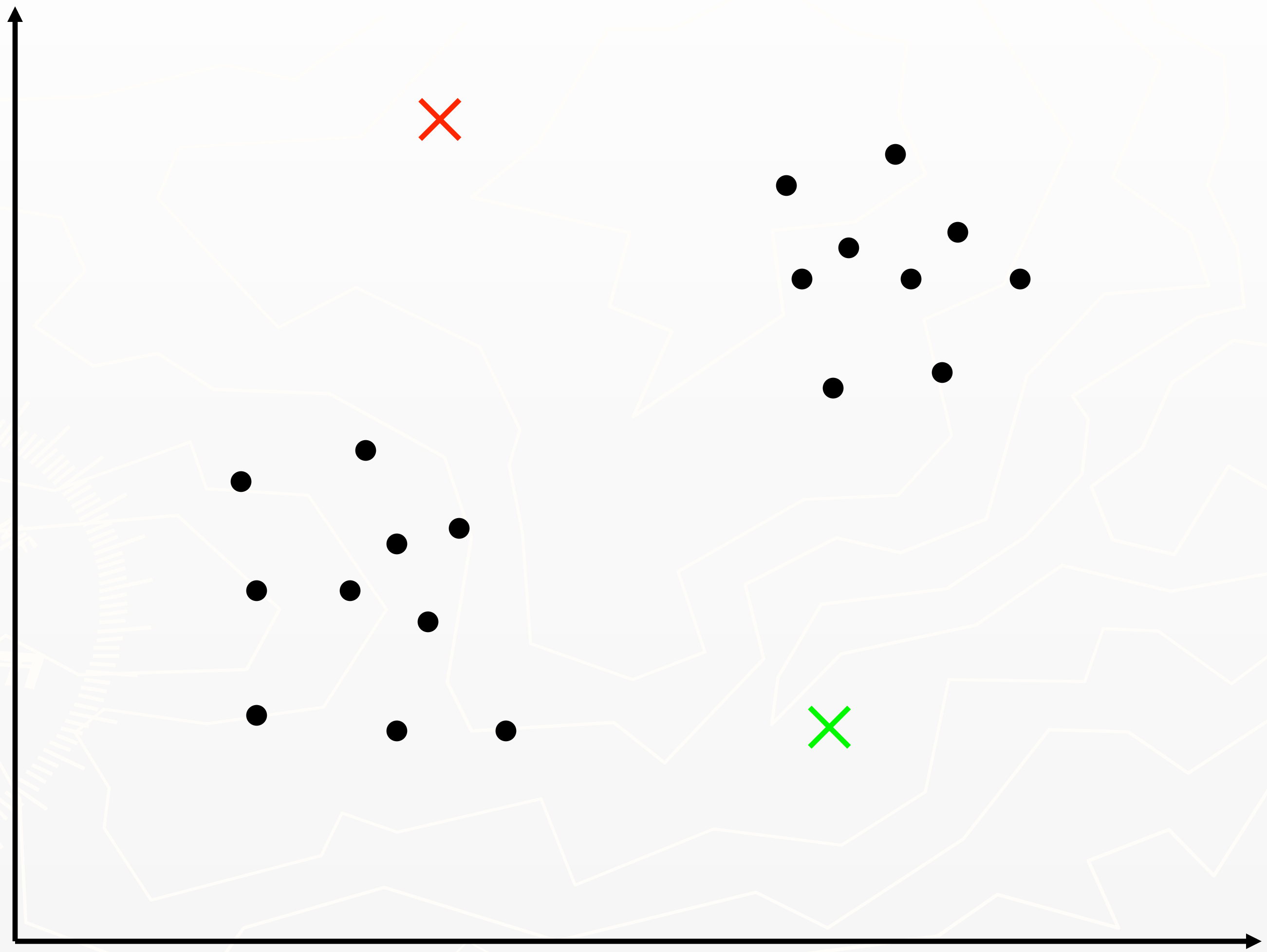
K-means Clustering

- Partition the data points into K clusters randomly. Find the centroids of each cluster.
- For each data point:
  - Calculate the distance from the data point to each cluster.
  - Assign the data point to the closest cluster.
- Recompute the centroid of each cluster.
- Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

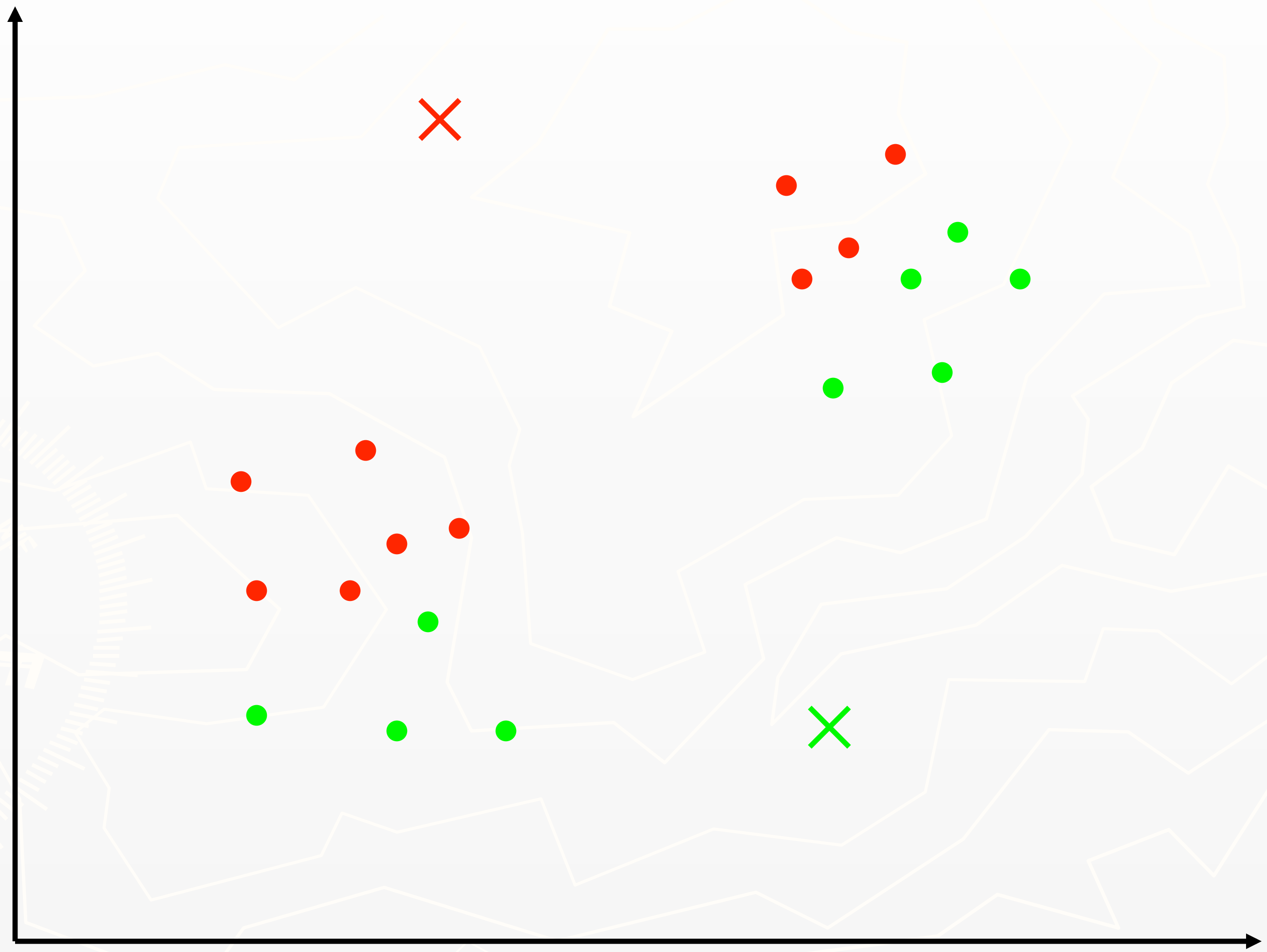
K-Means Clustering



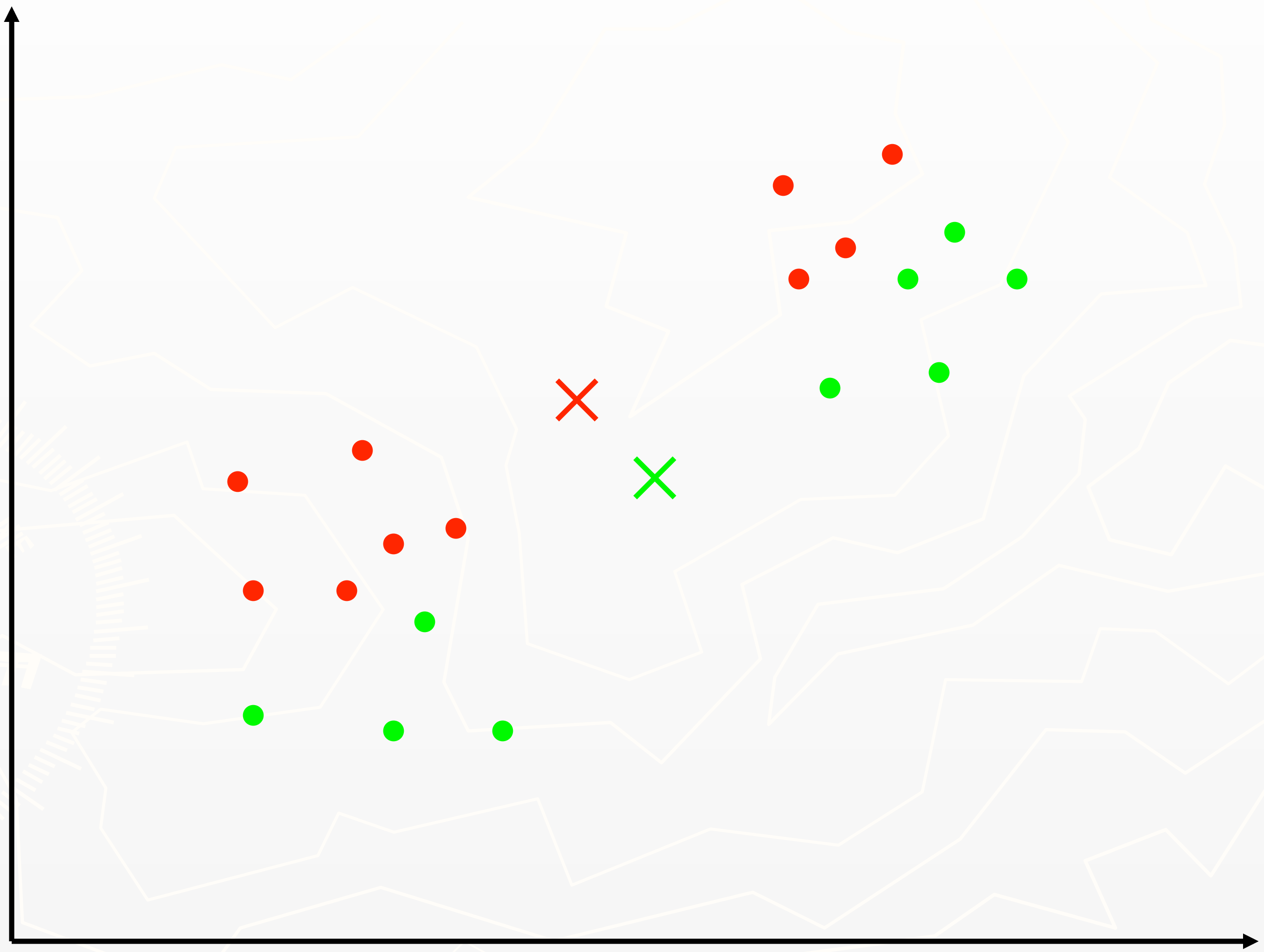
K-Means Clustering



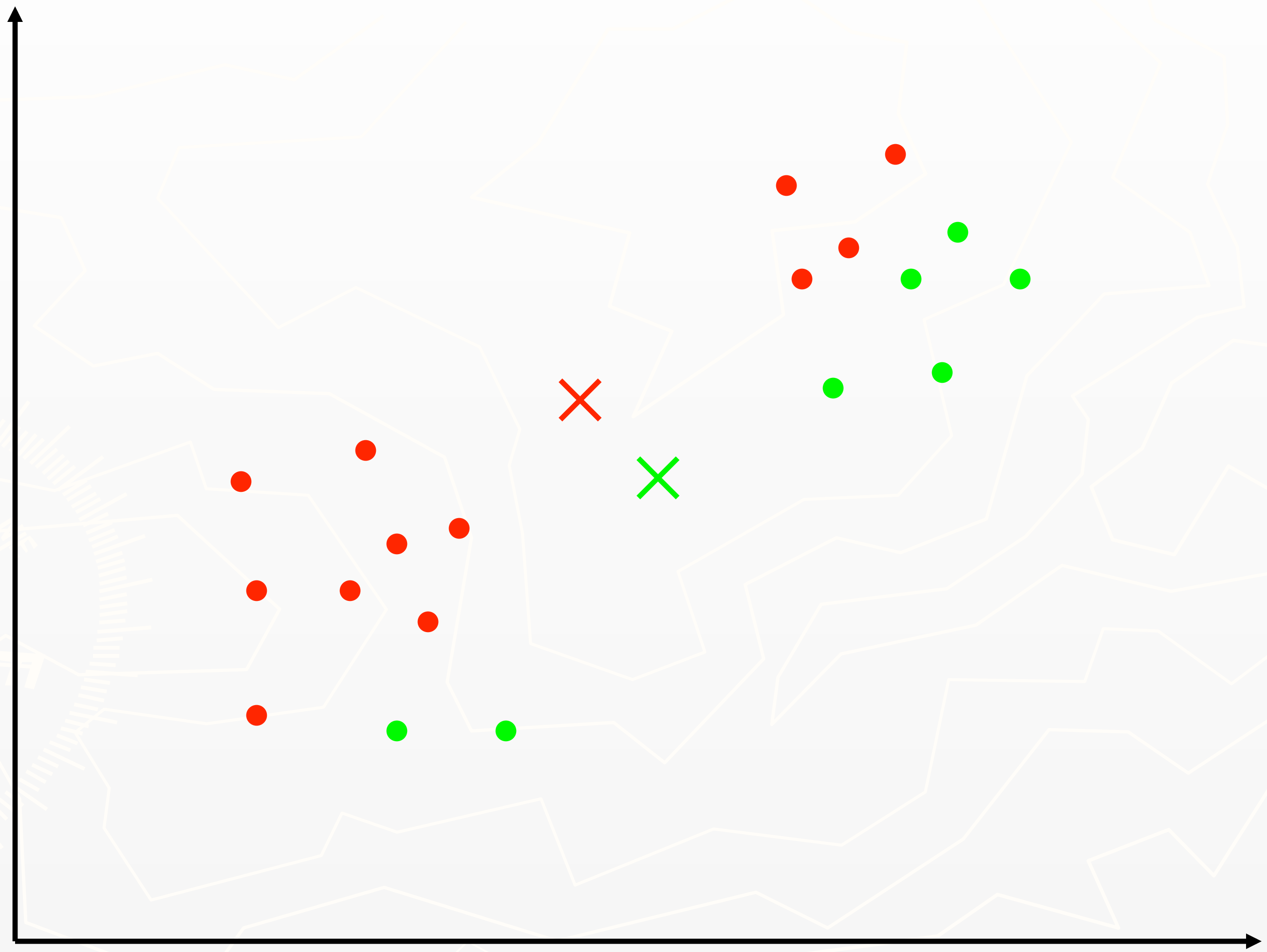
K-Means Clustering



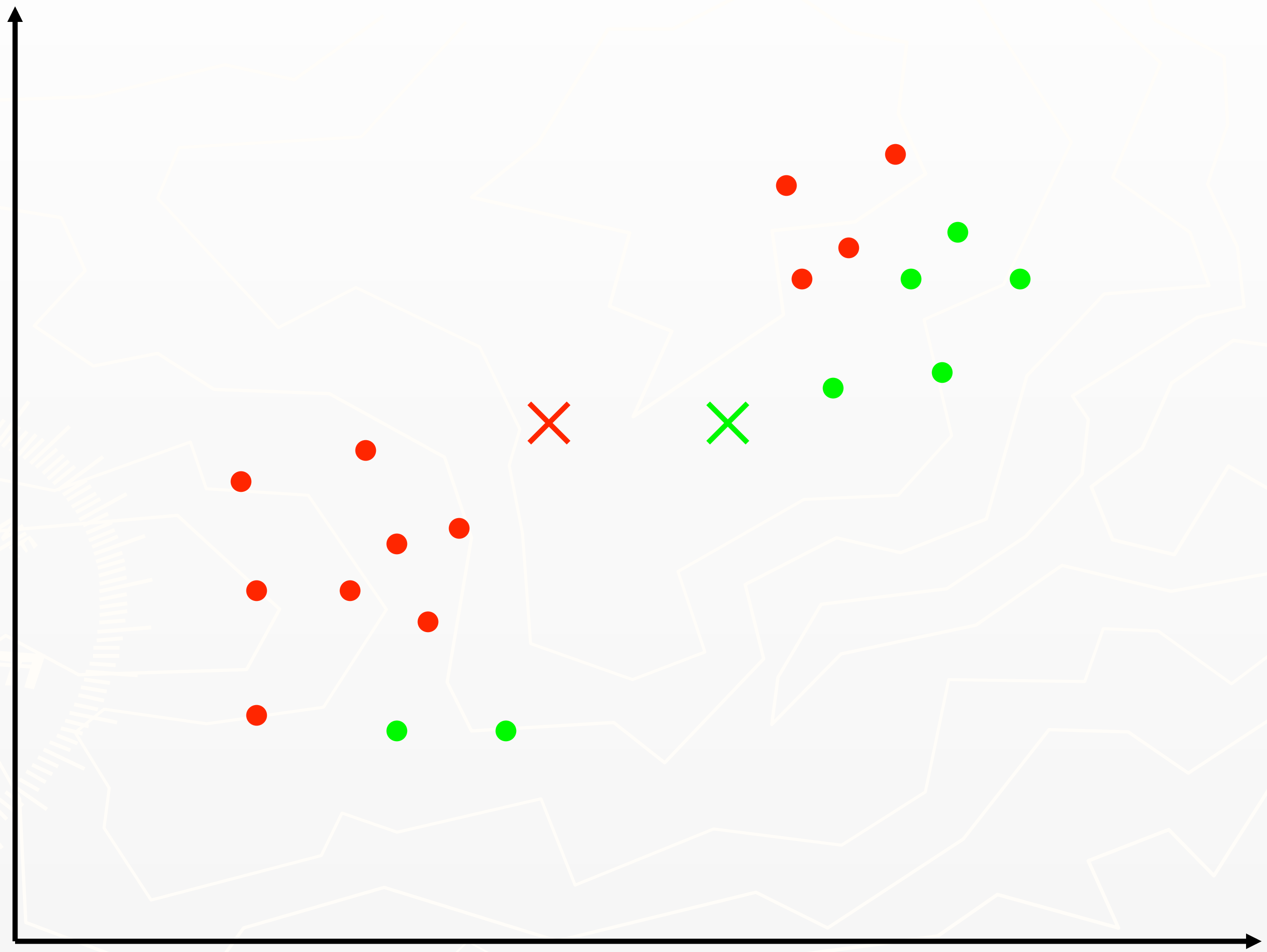
K-Means Clustering



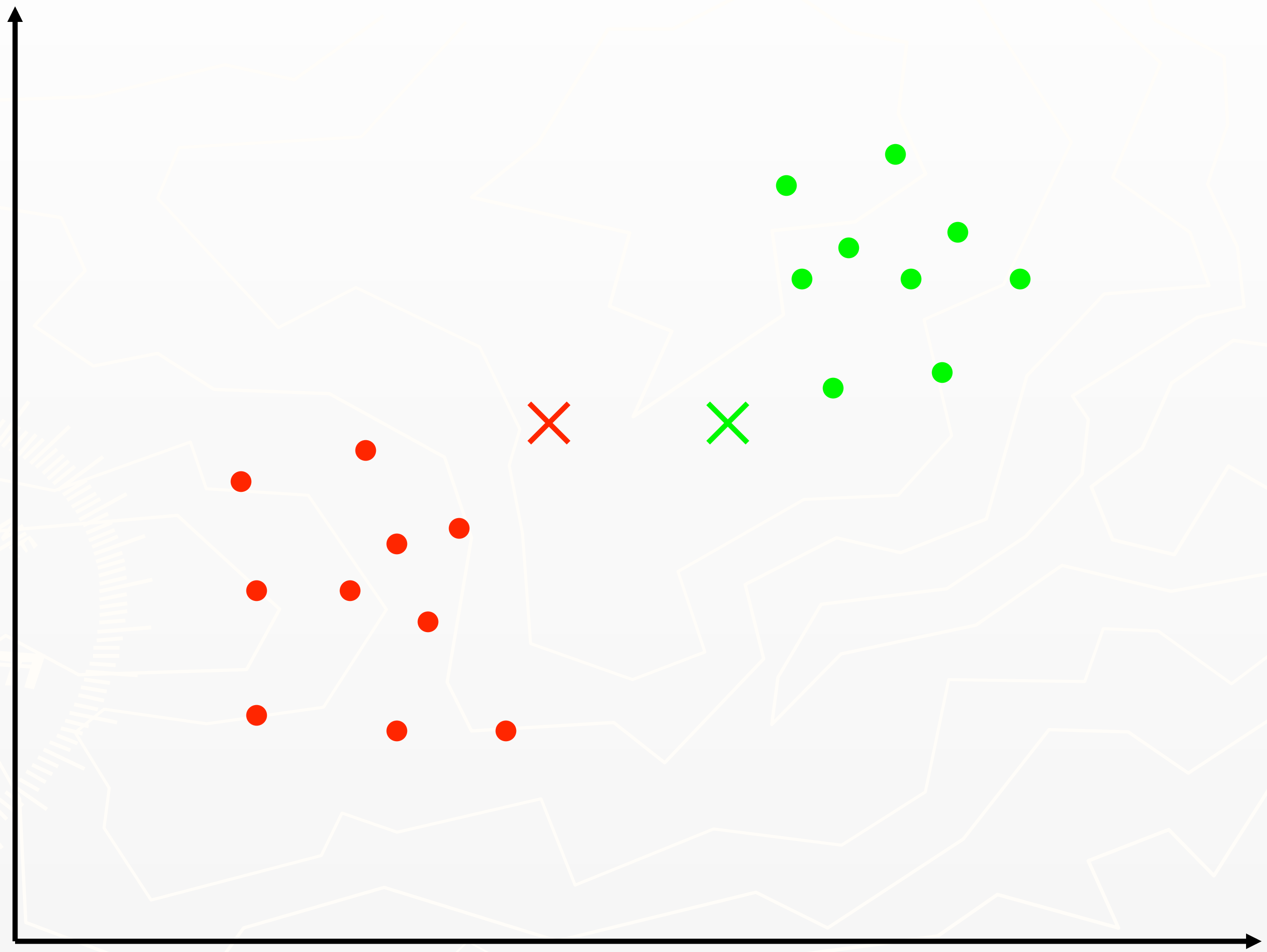
K-Means Clustering



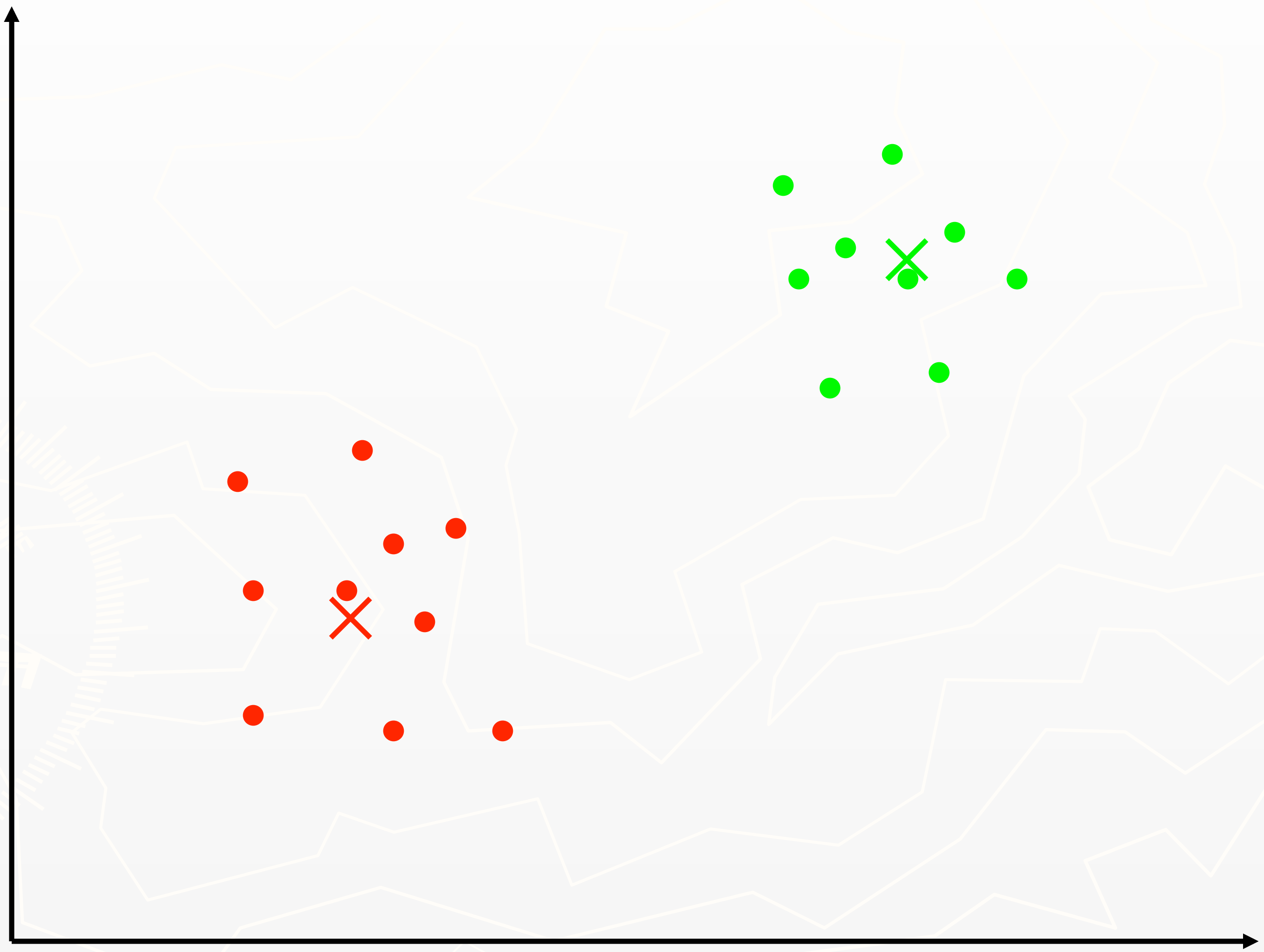
K-Means Clustering



K-Means Clustering

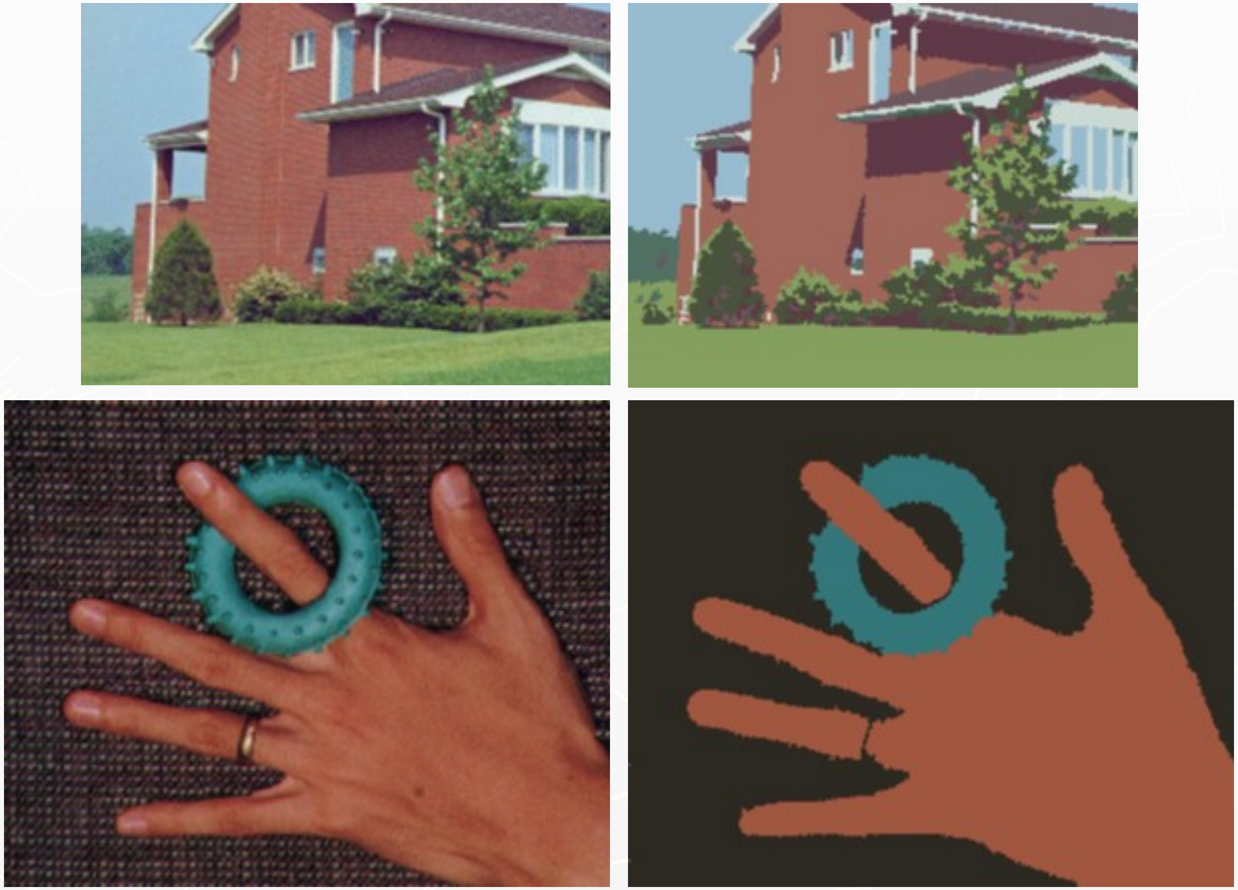


K-Means Clustering



Clustering

- Example



D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.