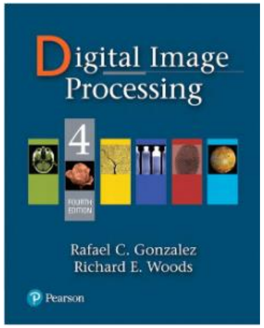


Digital Image Processing

Image Enhancement(II)

Dr. Tun-Wen Pai

1) Local Enhancement
2) Spatial Filtering
3) Spatial Convolution and Correlation
4) Smoothing Filtering

Image Enhancement Techniques

Point Operations

- Image Negative
- Contrast Stretching
- Compression of dynamic range
- Graylevel slicing
- Image Subtraction
- Image Averaging
- Histogram operations

Mask Operations

- Smoothing operations
- Median Filtering
- Sharpening operations
- Derivative operations
- Histogram operations

Transform Operations

- Low pass filtering
- High pass filtering
- Band pass filtering
- Homomorphic filtering
- Histogram operations

Coloring Operations

- False coloring
- Full color processing

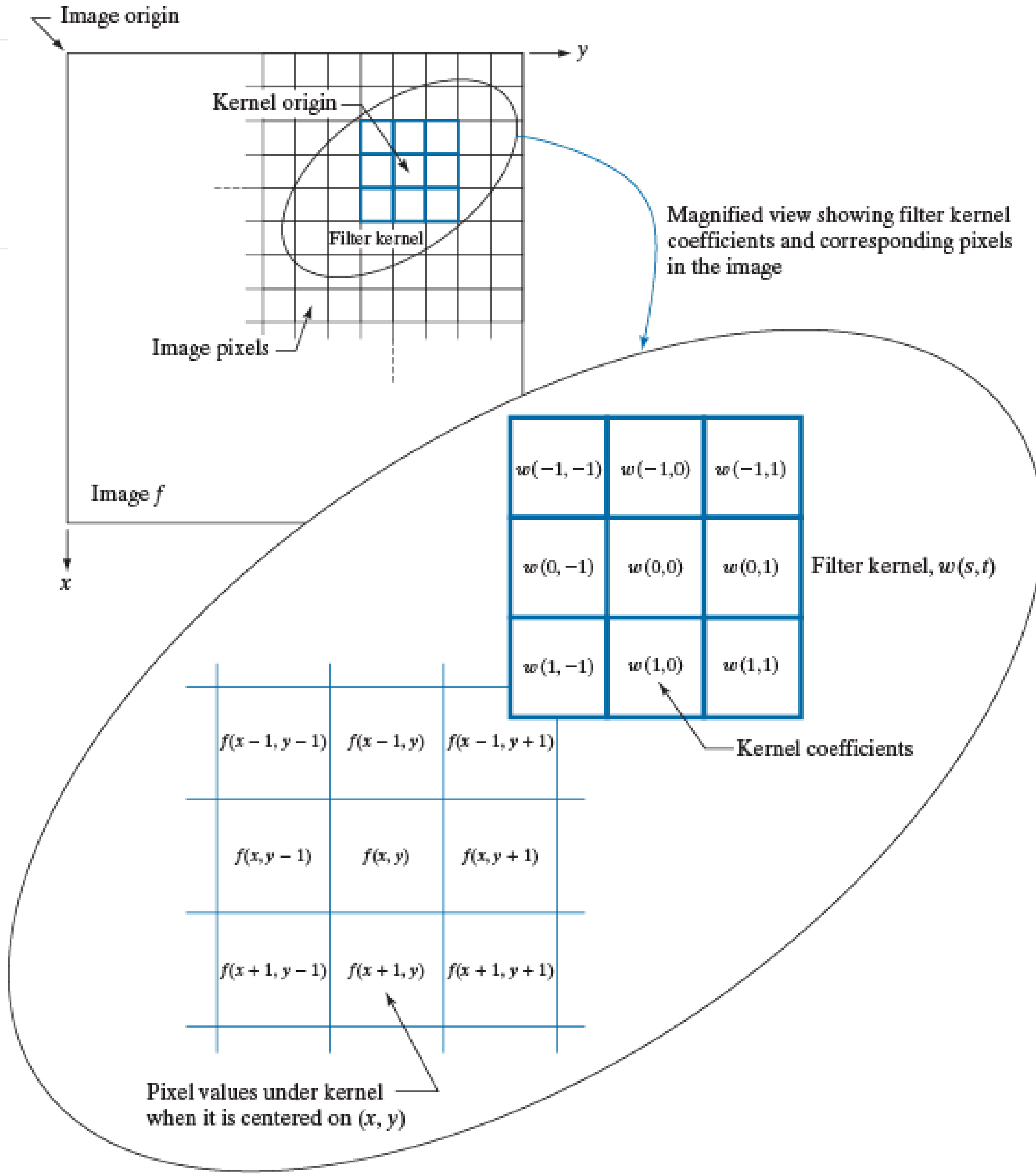


FIGURE 3.34 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

FIGURE 3.35 Illustration of 1-D correlation and convolution of a kernel, w, with a function f consisting of a discrete unit impulse. Note that correlation and convolution are functions of the variable x, which acts to *displace* one function with respect to the other. For the extended correlation and convolution results, the starting configuration places the right-most element of the kernel to be coincident with the origin of f. Additional padding must be used.

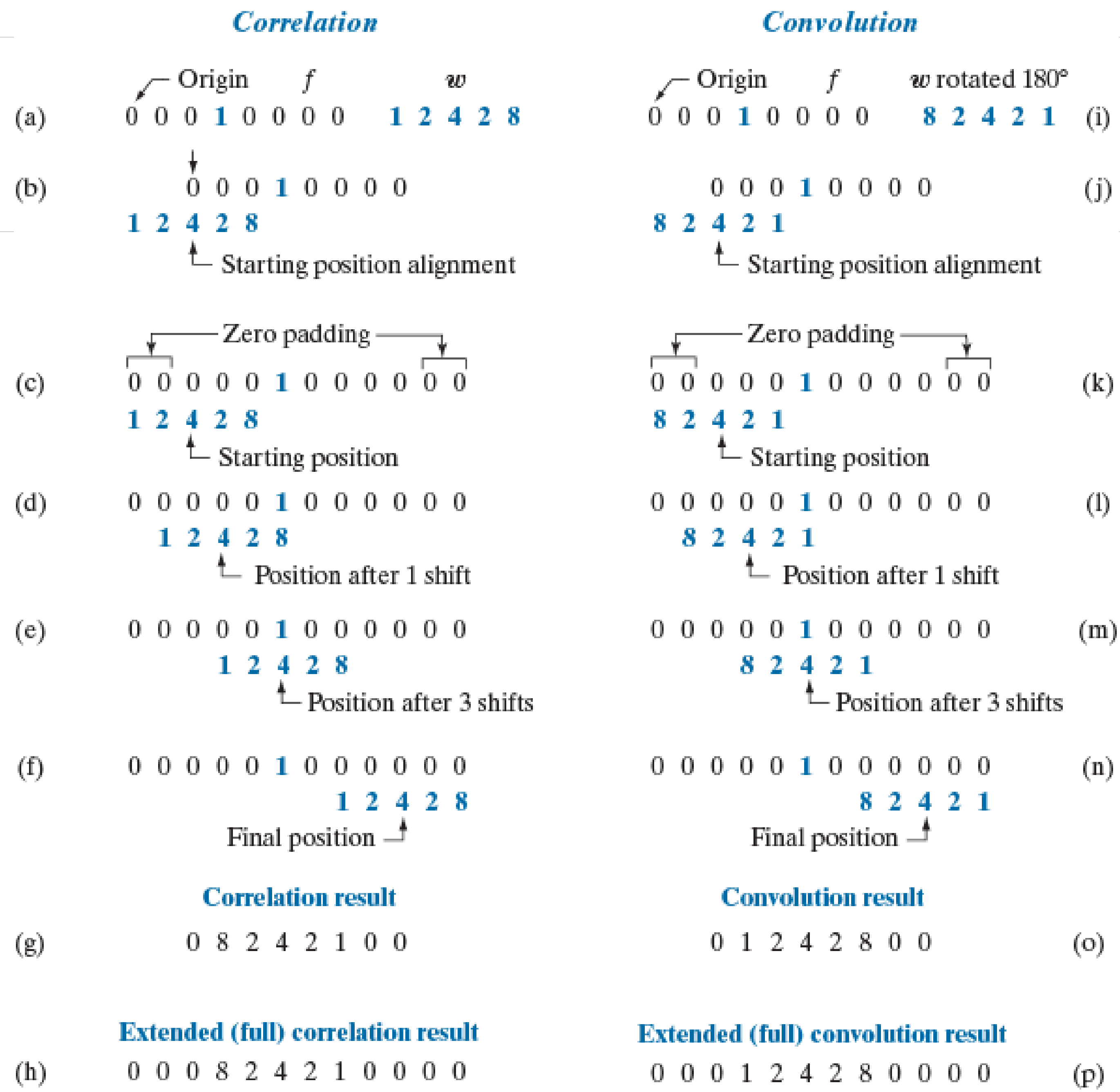


FIGURE 3.36 Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0's are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of x and y. As these variable change, they *displace* one function with respect to the other. See the discussion of Eqs. (3-45) and (3-46) regarding full correlation and convolution.

Origin f

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

w

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0
4	5	6	0	0	0	0
7	8	9	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(c)

Correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(d)

Full correlation result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0
6	5	4	0	0	0	0
3	2	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(f)

Convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(g)

Full convolution result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(h)

TABLE 3.5 Some fundamental properties of convolution and correlation. A dash means that the property does not hold.

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \,☆\, g) + (f \,☆\, h)$

a b

FIGURE 3.37 Examples of smoothing kernels: (a) is a *box* kernel; (b) is a *Gaussian* kernel.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

Smoothing Filters

- Image smoothing refers to any image-to-image transformation designed to “smooth” or flatten the image by reducing the rapid pixel-to-pixel variation in grayvalues.

- Smoothing filters are used for:
  - **Blurring**: This is usually a preprocessing step for removing small (unwanted) details before extracting the relevant (large) object, bridging gaps in lines/curves,
  - **Noise reduction**: Mitigate the effect of noise by linear or nonlinear operations.

- Smoothing is accomplished by applying an **averaging mask**.

- An averaging mask is a mask with positive weights, which sum to 1. It computes a weighted average of the pixel values in a neighborhood. This operation is sometimes called **neighborhood averaging**.
- Some 3 x 3 averaging masks:

$$\frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{32}\begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix} \quad \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- This operation is equivalent to lowpass filtering.

Example of Image Blurring

Original Image

$$\frac{1}{N^2}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{N\times N}$$

Avg. Mask

$N = 3$

$N = 5$

$N = 7$

$N = 11$

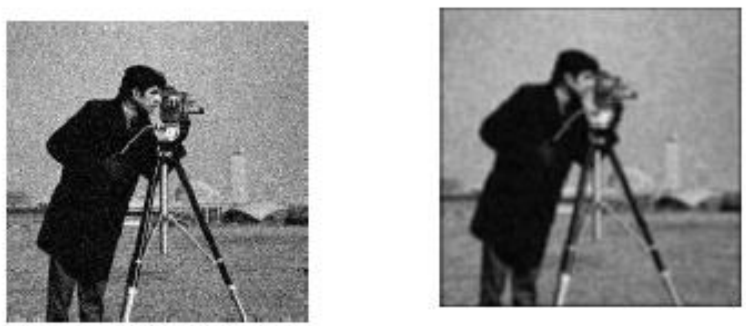
$N = 15$

$N = 21$

Example of noise reduction

$$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Noise-free Image

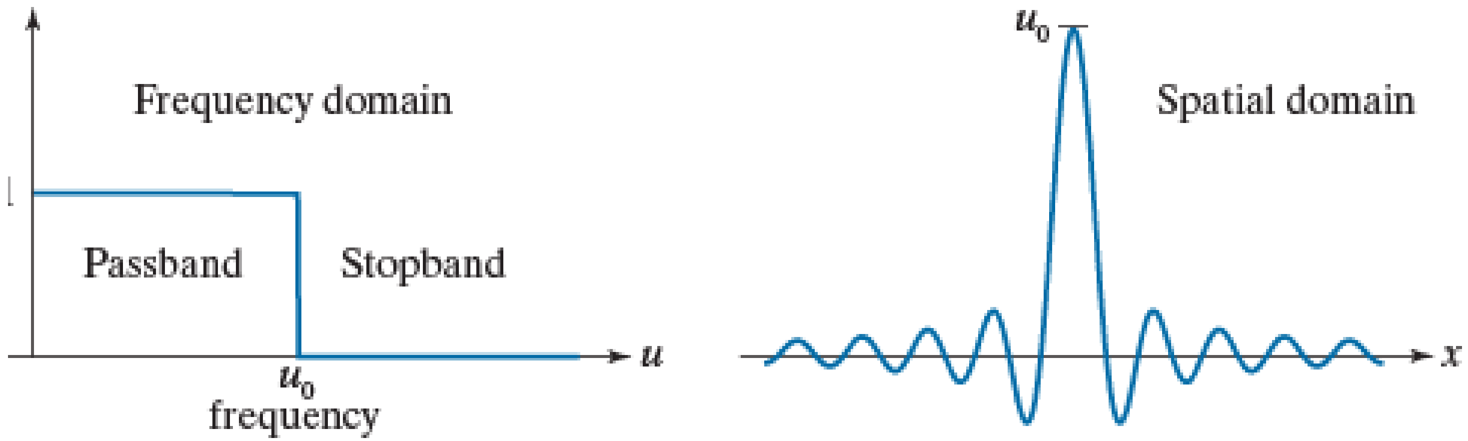


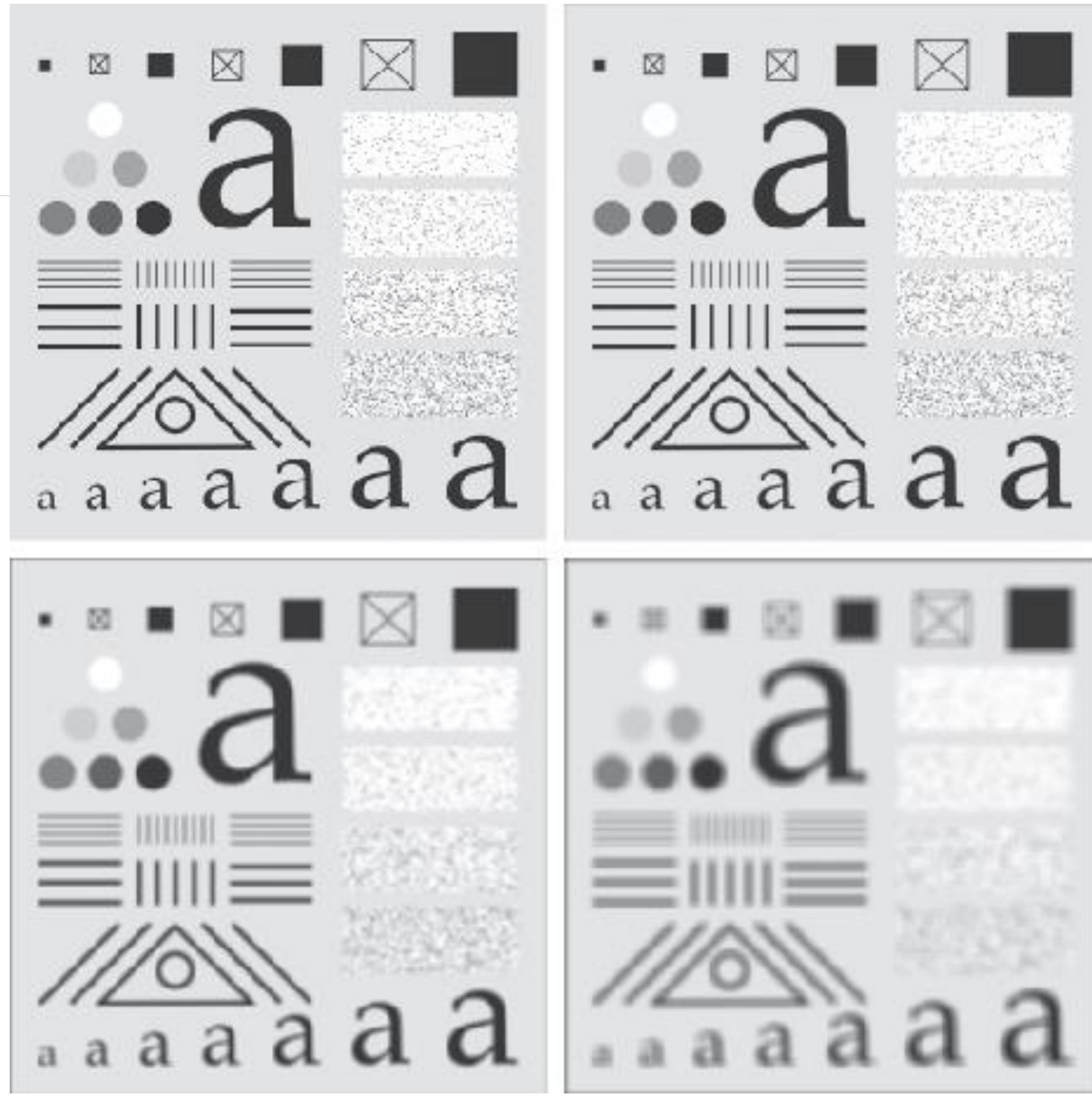
Zero-mean Gaussian noise, Variance = 0.01

Zero-mean Gaussian noise, Variance = 0.05

a b

FIGURE 3.38 (a) Ideal 1-D lowpass filter transfer function in the frequency domain. (b) Corresponding filter kernel in the spatial domain.

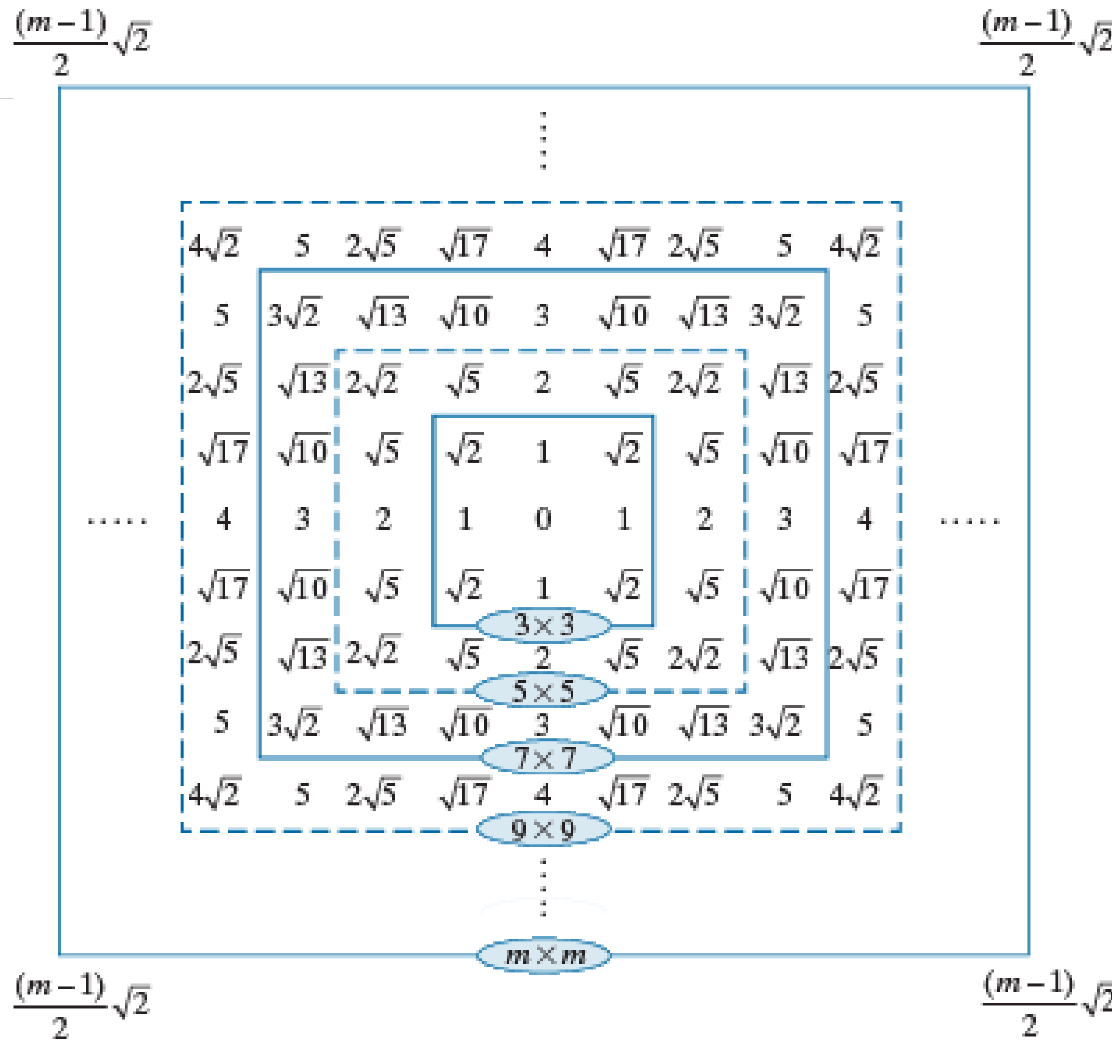




a b
c d

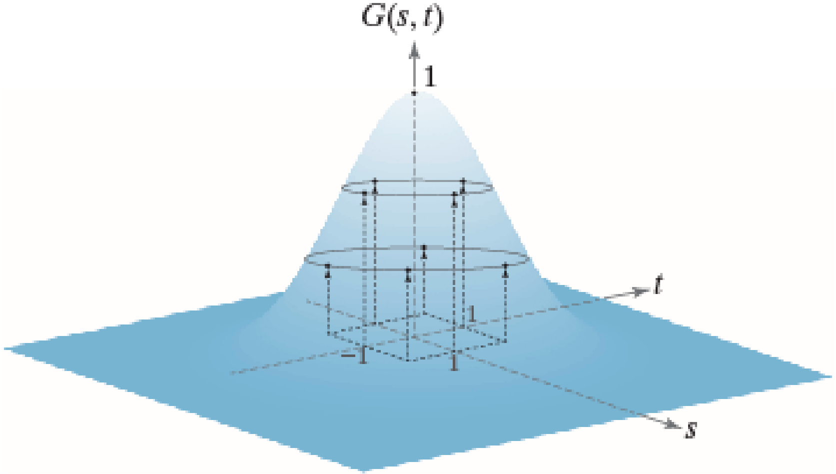
FIGURE 3.39
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.

FIGURE 3.40
Distances from the center for various sizes of square kernels.



a b

FIGURE 3.41
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.37(b)].

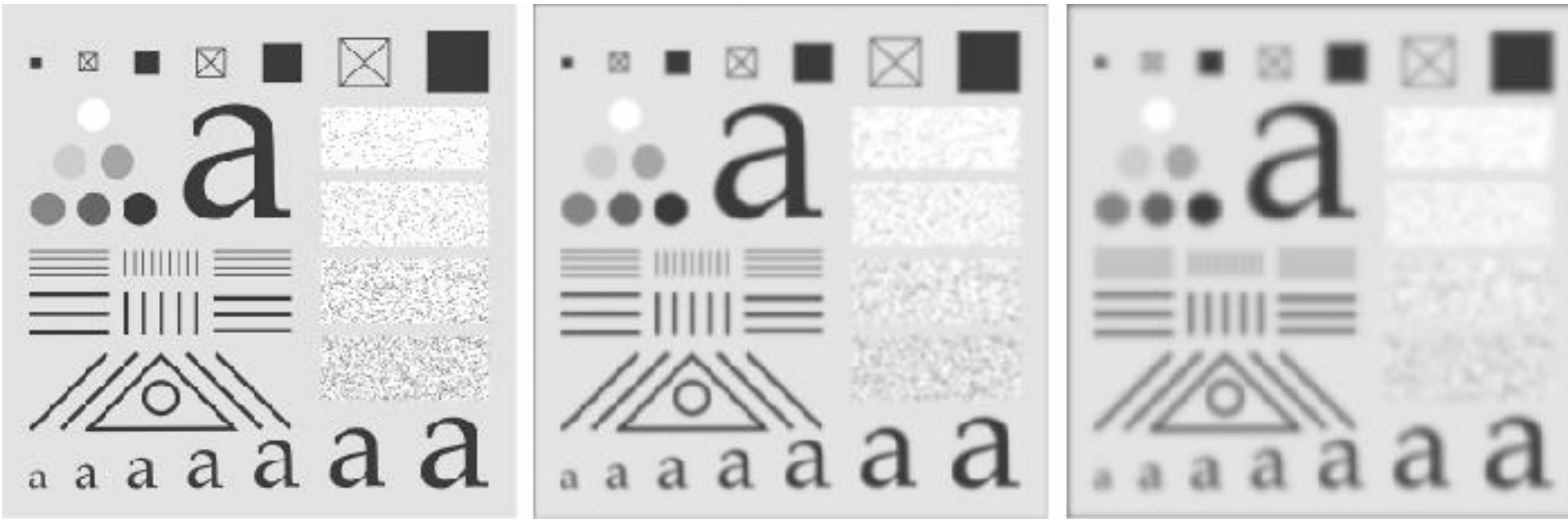


$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

TABLE 3.6 Mean and standard deviation of the product ($\times$) and convolution ($\star$) of two 1-D Gaussian functions, f and g. These results generalize directly to the product and convolution of more than two 1-D Gaussian functions (see Problem 3.33).

	f	g	$f \times g$	$f \star g$
Mean	m_f	m_g	$m_{f\times g} = \dfrac{m_f\sigma_g^2 + m_g\sigma_f^2}{\sigma_f^2 + \sigma_g^2}$	$m_{f\star g} = m_f + m_g$
Standard deviation	σ_f	σ_g	$\sigma_{f\times g} = \sqrt{\dfrac{\sigma_f^2\sigma_g^2}{\sigma_f^2 + \sigma_g^2}}$	$\sigma_{f\star g} = \sqrt{\sigma_f^2 + \sigma_g^2}$

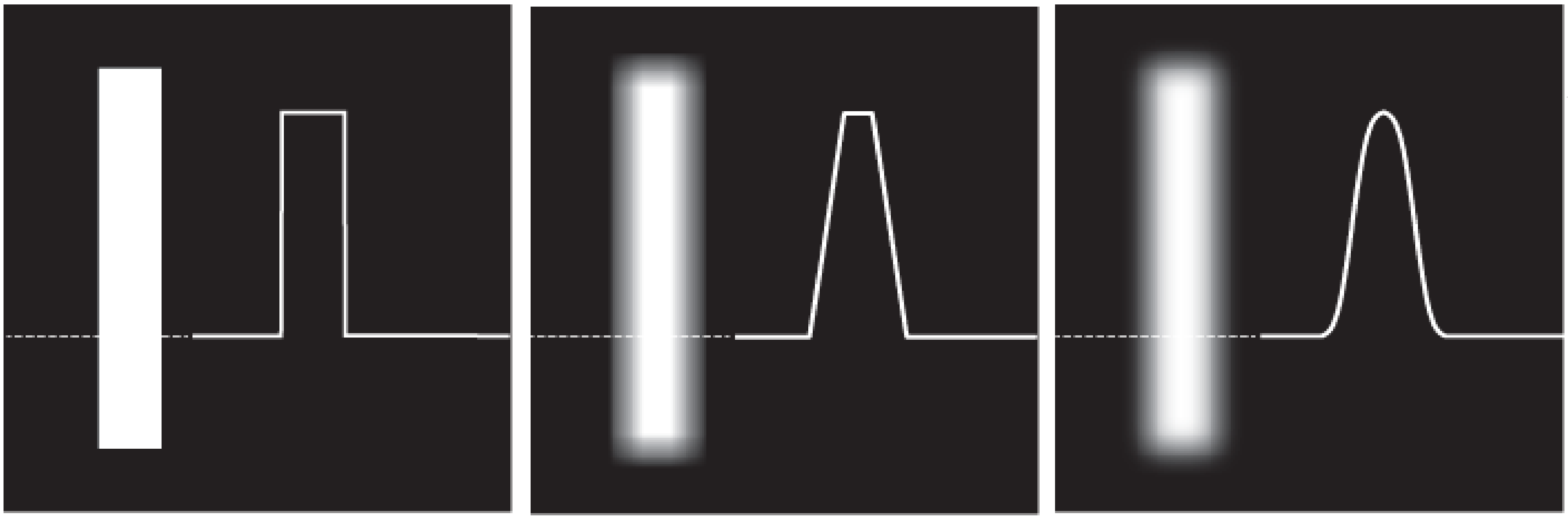


a b c

FIGURE 3.42 (a) A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.39(d). We used $K = 1$ in all cases.

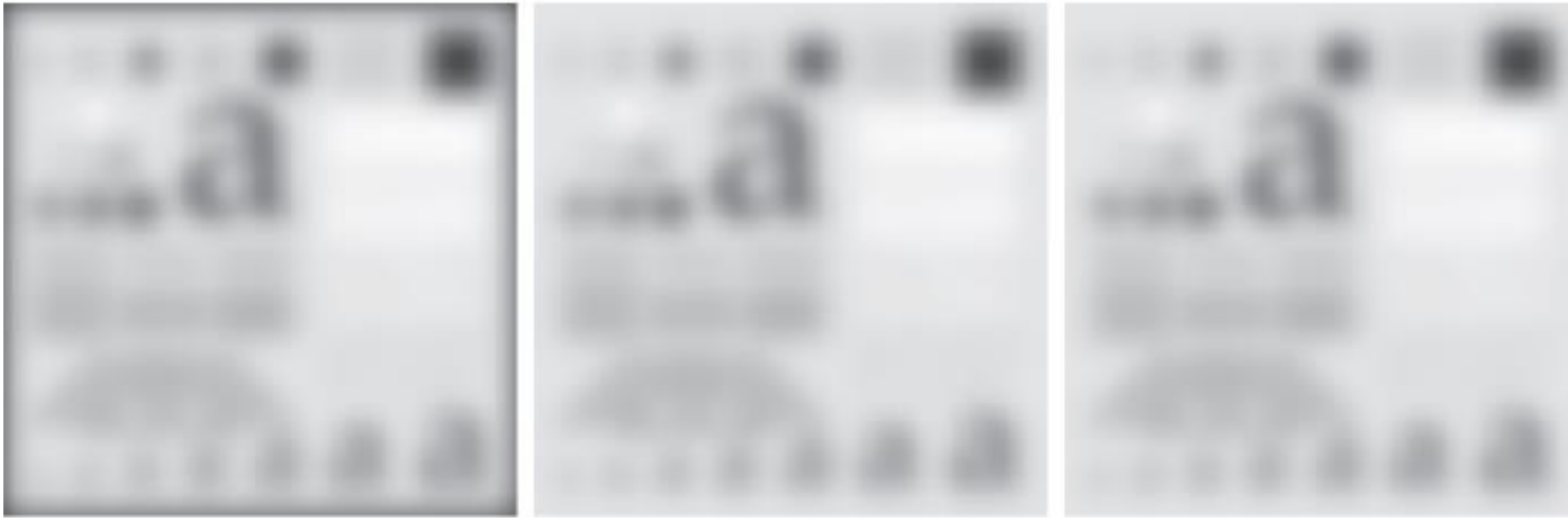
a b c

FIGURE 3.43 (a) Result of filtering Fig. 3.42(a) using a Gaussian kernels of size 43×43, with $\sigma = 7$. (b) Result of using a kernel of 85×85, with the same value of σ. (c) Difference image.



a b c

FIGURE 3.44 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.

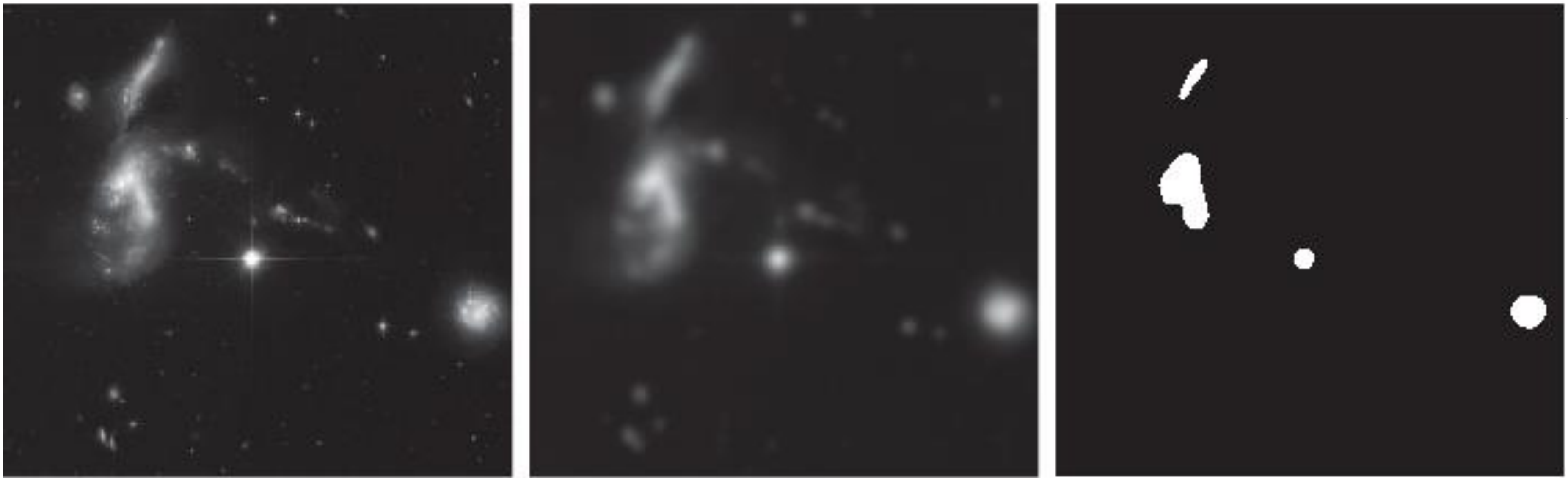


a b c

FIGURE 3.45 Result of filtering the test pattern in Fig. 3.42(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size 187×187, with $K = 1$ and $\sigma = 31$ was used in all three cases.

a b c

FIGURE 3.46 (a) Test pattern of size 4096×4096 pixels. (b) Result of filtering the test pattern with the same Gaussian kernel used in Fig. 3.45. (c) Result of filtering the pattern using a Gaussian kernel of size 745×745 elements, with $K = 1$ and $\sigma = 124$. Mirror padding was used throughout.



a b c

FIGURE 3.47 (a) A 2566×2758 Hubble Telescope image of the *Hickson Compact Group*. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range $[0, 1]$). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)

a b c

FIGURE 3.48 (a) Image shaded by a shading pattern oriented in the $-45°$ direction. (b) Estimate of the shading patterns obtained using lowpass filtering. (c) Result of dividing (a) by (b). (See Section 9.8 for a morphological approach to shading correction).

Median Filtering

- The averaging filter is best suited for noise whose distribution is Gaussian:

$$p_{noise}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

- The averaging filter typically blurs edges and sharp details.
- The median filter usually does a better job of preserving edges.
- Median filter is particularly suited if the noise pattern exhibits strong (positive and negative) spikes. **Example**: salt and pepper noise.
- Median filter is a nonlinear filter, that also uses a mask. Each pixel is replaced by the median of the pixel values in a neighborhood of the given pixel.
- Suppose A = $\{a_1, a_2, \ldots, a_k\}$ are the pixel values in a neighborhood of a given pixel with $a_1 \leq a_2 \leq \ldots \leq a_k$. Then

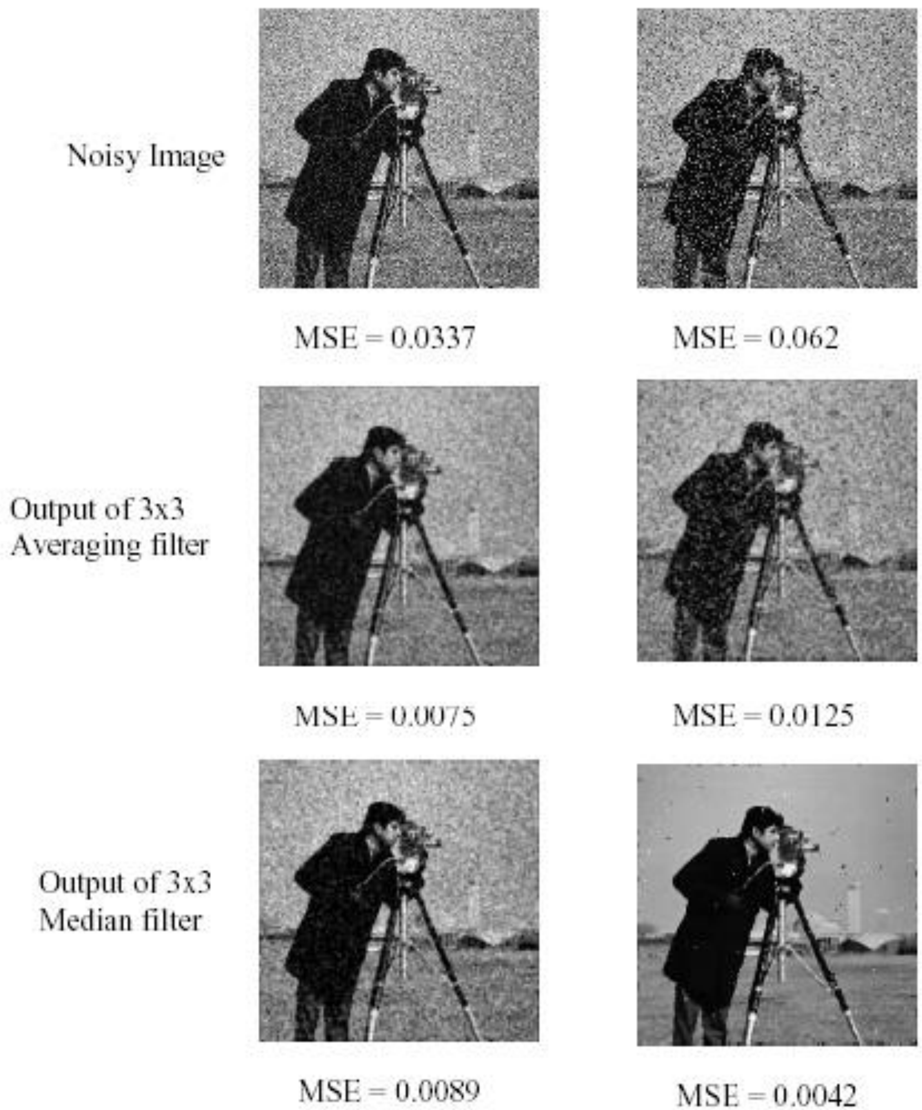
$$\text{median}\,(A) = \begin{cases} a_{k/2}, \text{for K even} \\ a_{(k+1)/2}, \text{for K odd} \end{cases}$$

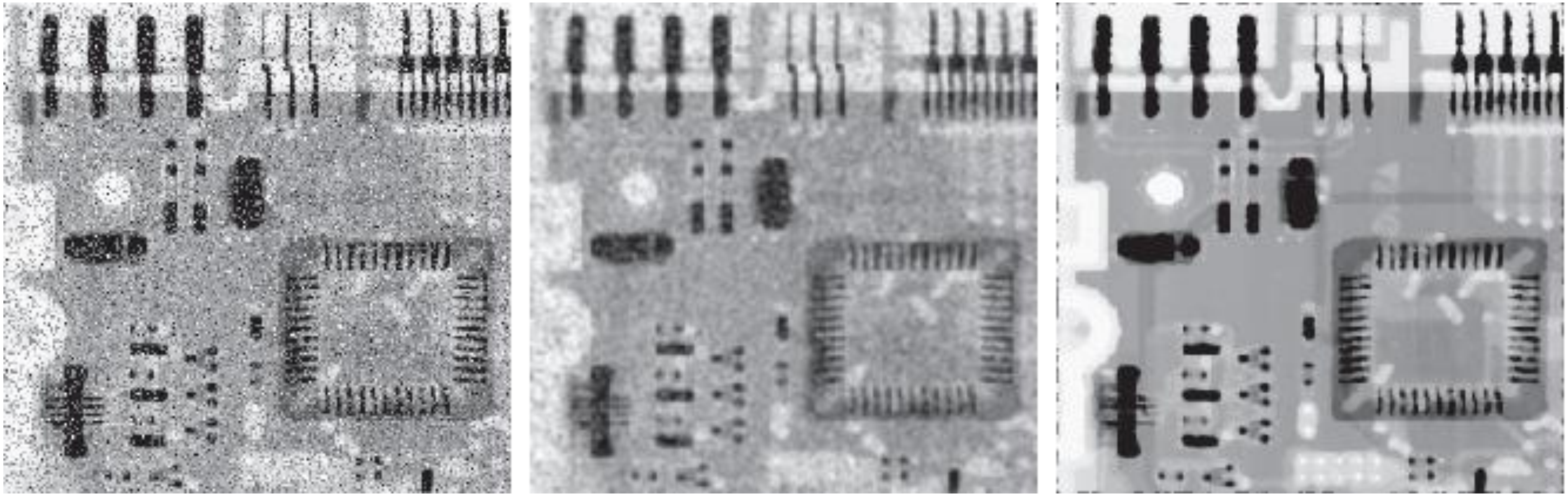
Note: Median of a set of values is the "center value," after sorting.

- For example: If A = {0, 1, 2, 4, 6, 6, 10, 12, 15}, then median(A) = 6.

Example of noise reduction

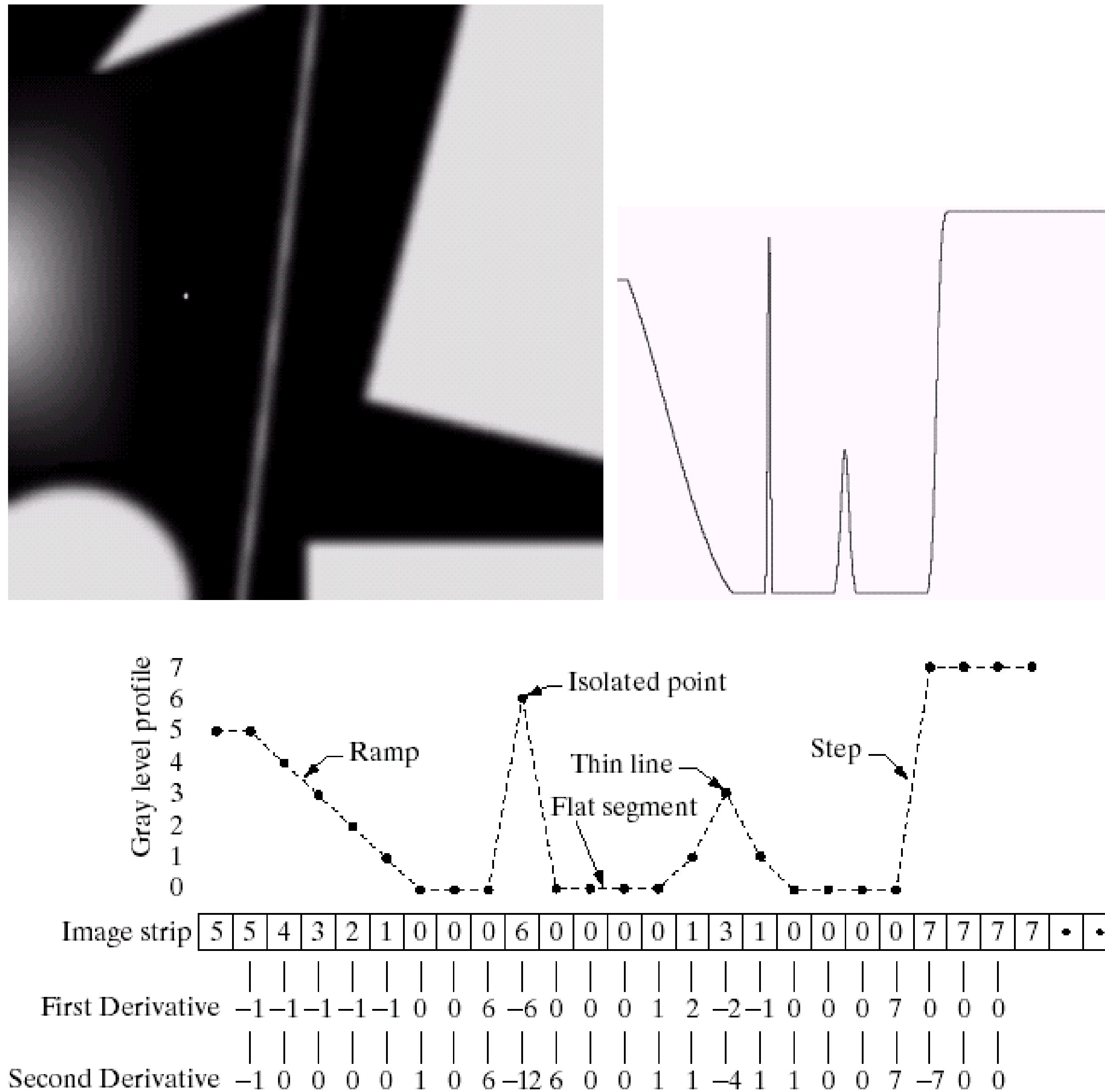
Gaussian nose: σ = 0.2 Salt & Pepper noise: prob. = 0.2





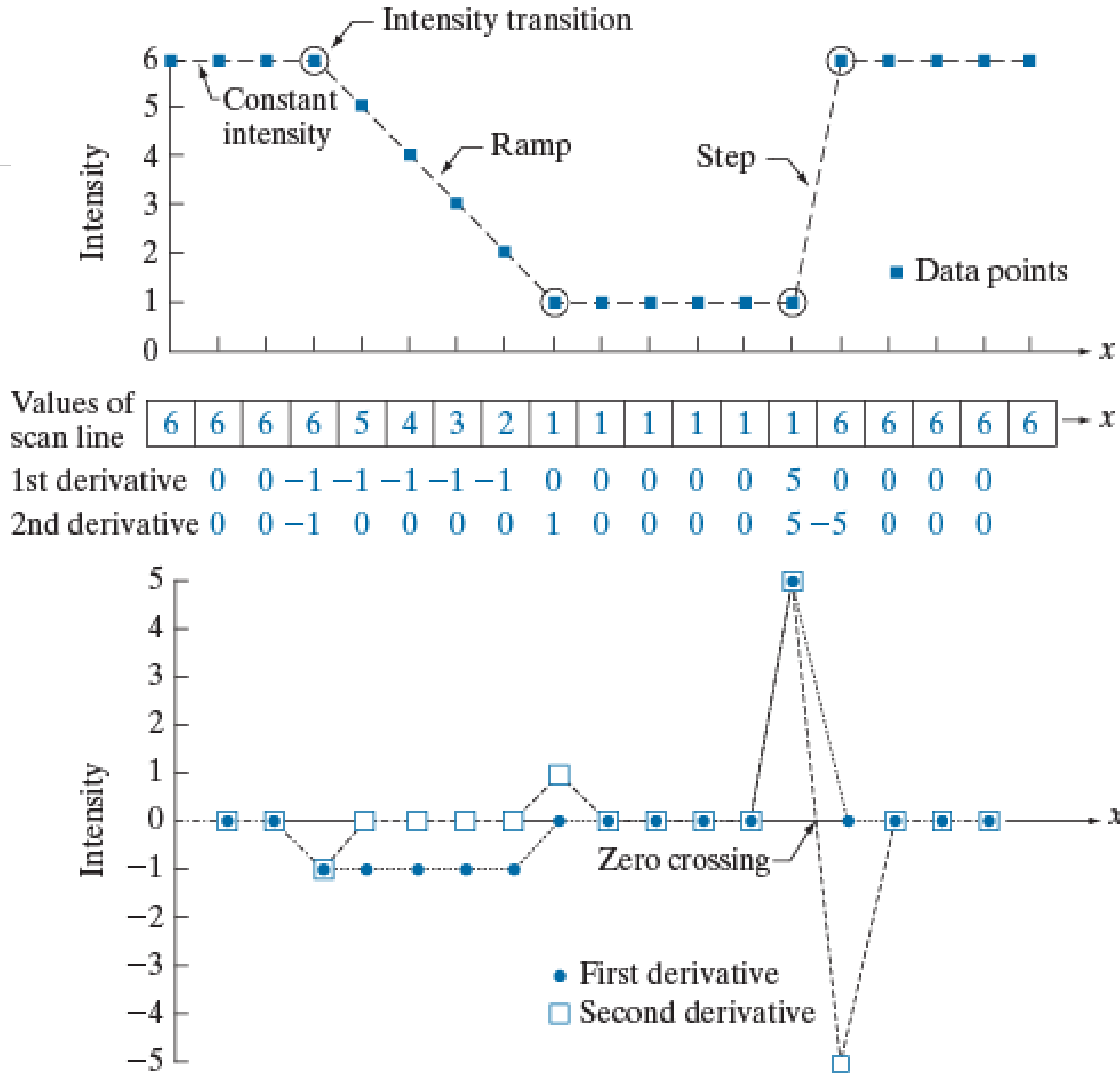
a b c

FIGURE 3.49 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



a
b
c

FIGURE 3.50
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b) Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

Image Sharpening

- This involves highlighting fine details or enhancing details that have been blurred.

Basic highpass spatial filtering

- This can be accomplished by a linear shift-invariant operator, implemented by means of a mask, with positive and negative coefficeitns.

- This is called a sharpening mask, since it tends to enhance abrupt graylevel changes in the image.

- The mask should have a positive coefficient at the center and negative coefficients at the periphery. The coefficients should sum to zero. Example:

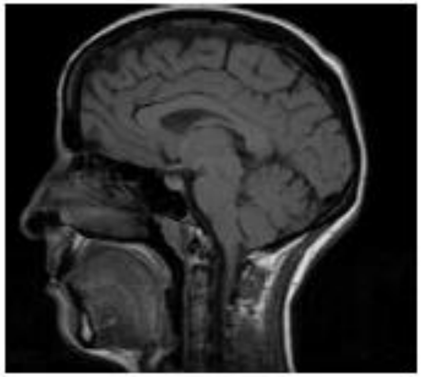
$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

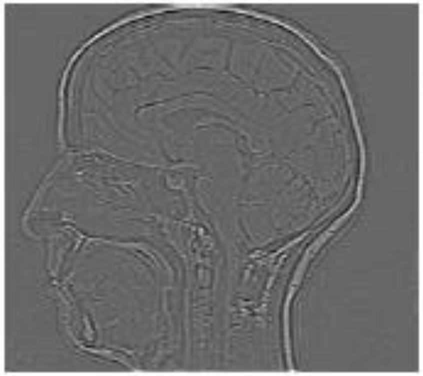
- This is equivalent to **highpass filtering**.

• A highpass filtered image g can be thought of as the difference between the original image f and a lowpass filtered version of f:

$$g(m, n) = f(m, n) - \text{lowpass}(f(m,n))$$

Example





0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

FIGURE 3.39
(a) Filter mask used to implement the digital Laplacian, as defined in Eq.(3.7-4).
(b) Mask used to implement an extension of this equation that includes the diagonal neighbors.
(c) and (d) Two other implementations of the Laplacian.

High-boost filtering

- This is a filter whose output g is produced by subtracting a lowpass (blurred) version of f from an amplified version of f

$$g(m,n) = Af(m,n) - lowpass(f(m,n))$$

This is also referred to as **unsharp masking**.

- Observe that

$$\begin{aligned} g(m,n) &= Af(m,n) - lowpass(f(m,n)) \\ &= (A-1)f(m,n) + f(m,n) - lowpass(f(m,n)) \\ &= (A-1)f(m,n) + highpass(f(m,n)) \end{aligned}$$

- For A>1, part of the original image is added back to the highpass filtered version of f.

- The result is the original image with the edges enhanced relative to the original image.
Example:

Original Image

Highpass filtering

High-boost filtering

0	−1	0
−1	$A + 4$	−1
0	−1	0

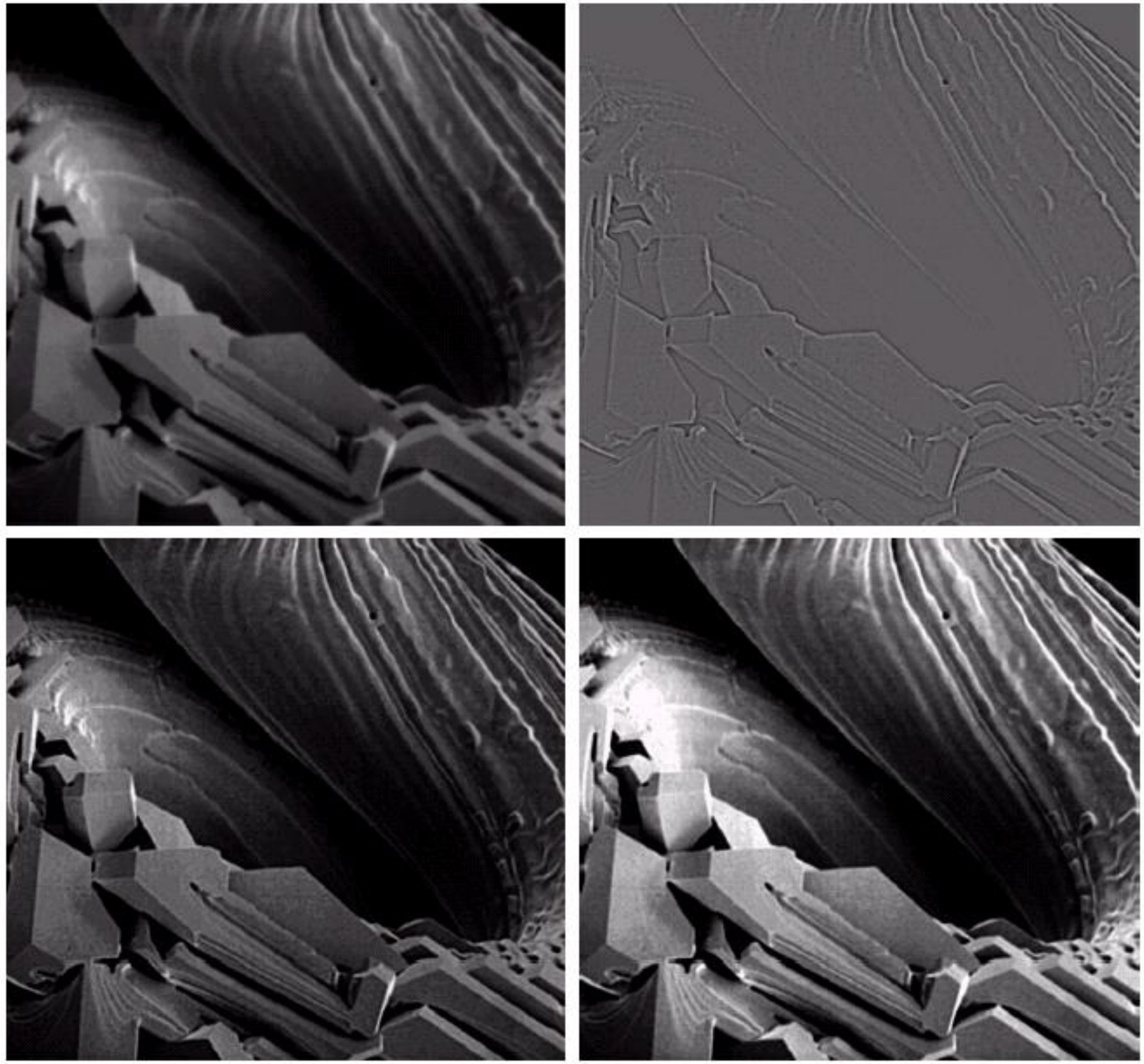
−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

a b

FIGURE 3.42
The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

a b
c d

FIGURE 3.43
(a) Same as Fig.3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig.3.42 (b) using A = 0.
(c) Laplacian enhanced image using the mask in Fig.3.42(b) with A = 1.
(d) Same as (c), but using A = 1.7.



Derivative filter

- Averaging tends to blur details in an image. Averaging involves summation or integration.
- Naturally, differentiation or "differencing" would tend to enhance abrupt changes, i.e., sharpen edges.
- Most common differentiation operator is the gradient.

$$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

- The magnitude of the gradient is:

$$|\nabla f(x,y)| = \left[\left(\frac{\partial f(x,y)}{\partial x} \right)^2 + \left(\frac{\partial f(x,y)}{\partial y} \right)^2 \right]^{1/2}$$

- Discrete approximations to the magnitude of the gradient is normally used. Consider the following image region:

z_1	z_1	z_1
z_1	z_1	z_1
z_1	z_1	z_1

- We may use the approximation

$$\left|\nabla f(x,y)\approx\left[(z_5-z_8)^2+(z_5-z_6)^2\right]^{1/2}\right.$$

- This can implemented using the masks:

$$h_1=\begin{bmatrix}1\\-1\end{bmatrix} and\ h_2=\begin{bmatrix}1 & -1\end{bmatrix}$$

As follows:

$$|\nabla f(x,y)|\approx\left[(f*h_1)^2+(f*h_2)^2\right]^{1/2}$$

- Alternatively, we may use the approximation:

$$|\nabla f(x,y)|\approx\left[(z_5-z_9)^2+(z_6-z_8)^2\right]^{1/2}$$

- This can implemented using the masks:

$$h_1=\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} and\ h_2=\begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}$$

As follows:

$$|\nabla f(x,y)|\approx\left[(f*h_1)^2+(f*h_2)^2\right]^{1/2}$$

- The resulting maks are called **Roberts cross-gradient operators**.

- The Roberts operators and the Prewitt/Sobel operators (described later) are used for edge detection and are sometimes called edge detectors.

Example: Roberts cross-gradient operator

$$h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \; and \; h_2 = \begin{bmatrix} 1 & -1 \end{bmatrix} \qquad h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \; and \; h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

- Better approximations to the gradient can be obtained by:

$$\left|\nabla f(x,y) \approx \left[((z_7+z_8+z_9)-(z_1+z_2+z_3))^2+((z_3+z_6+z_9)-(z_1+z4+z_7))^2\right]^{1/2}\right|$$

- This can be implemented using the masks:

$$h_1=\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix} \; and \; h_2=\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$$

as follows:

$$|\nabla f(x,y)| \approx \left[(f*h_1)^2+(f*h_2)^2\right]^{1/2}$$

- The resulting masks are called **Prewitt operators**.

- Another approximation is given by the masks:

$$h_1=\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix} \; and \; h_2=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$$

- The Resulting masks are called **Sobel operators**.


Prewitt


Sobel

a
b c
d e

FIGURE 3.44
A 3 x 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

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

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

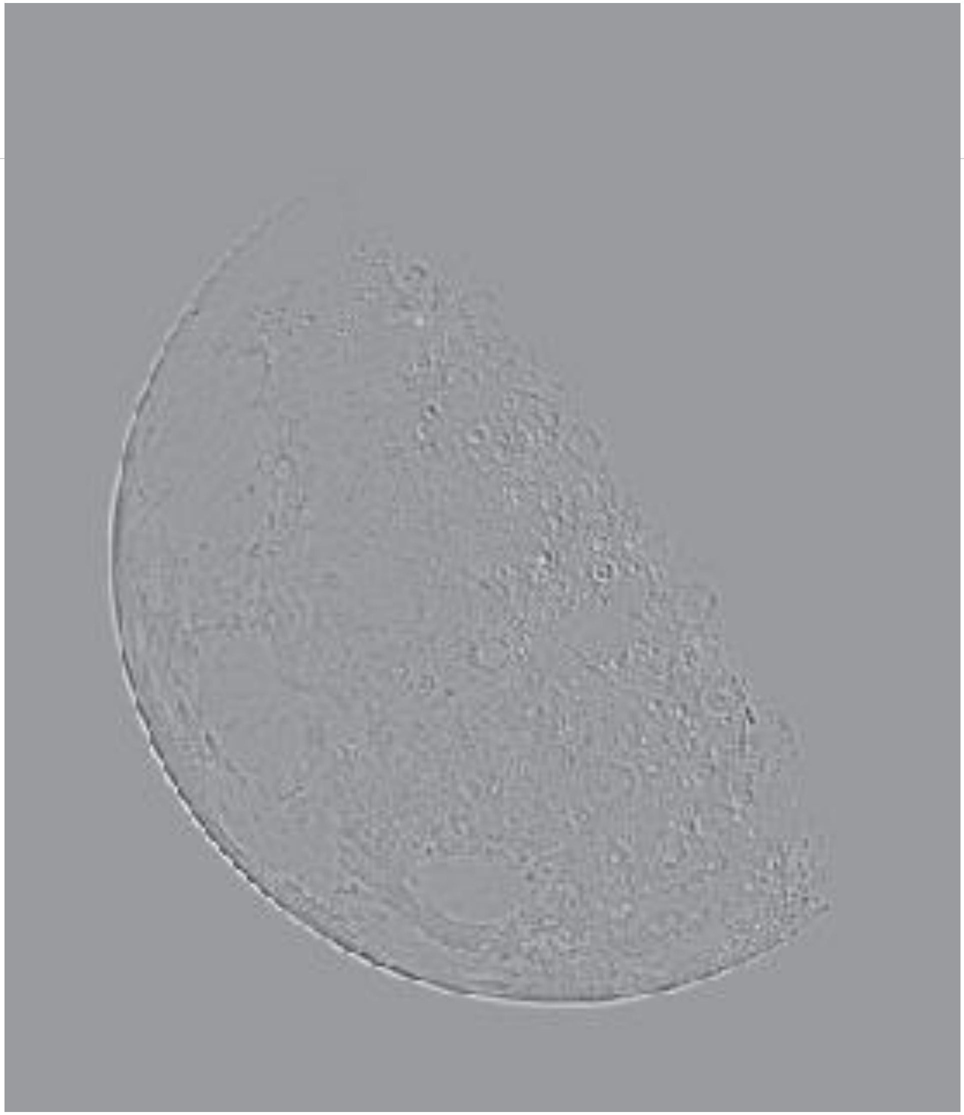
a b c d

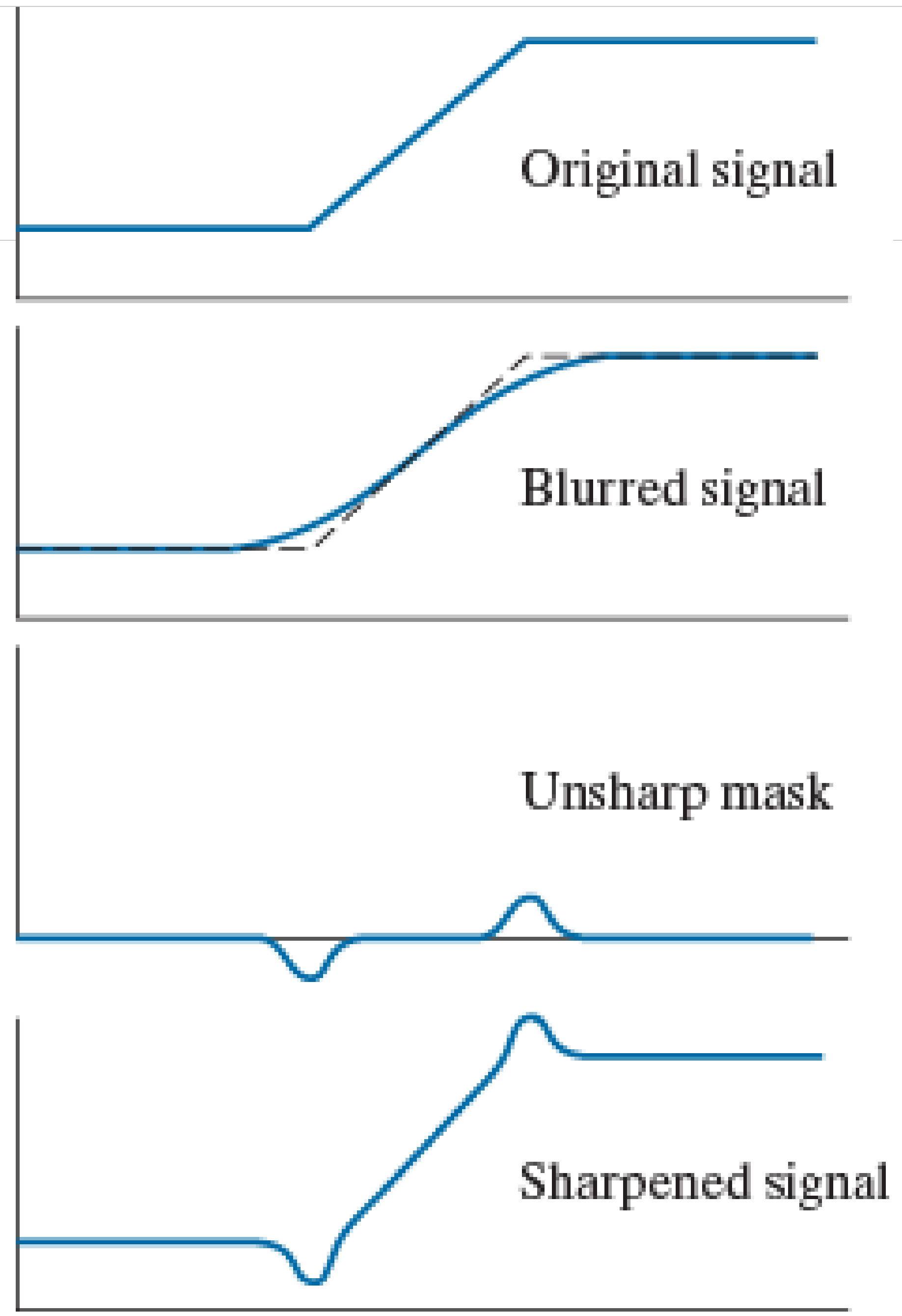
FIGURE 3.51 (a) Laplacian kernel used to implement Eq. (3-62). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

a b
c d

FIGURE 3.52
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.51(a).
(c) Image sharpened using Eq. (3-63) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.51(b).
(Original image courtesy of NASA.)

FIGURE 3.53
The Laplacian image from Fig. 3.52(b), scaled to the full [0, 255] range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels corresponds to the highest positive value.





a
b
c
d

FIGURE 3.54
1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

a b c
d e f

FIGURE 3.55 (a) Unretouched "soft-tone" digital image of size 469×600 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-65) with $k = 1$. (e) and (f) Results of highboost filtering with $k = 2$ and $k = 3$, respectively.

a
b c
d e

FIGURE 3.56 (a) A 3×3 region of an image, where the zs are intensity values. (b)–(c) Roberts cross-gradient operators. (d)–(e) Sobel operators. All the kernel coefficients sum to zero, as expected of a derivative operator.

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(b)

−1	0
0	1

(c)

0	−1
1	0

(d)

−1	−2	−1
0	0	0
1	2	1

(e)

−1	0	1
−2	0	2
−1	0	1

a b

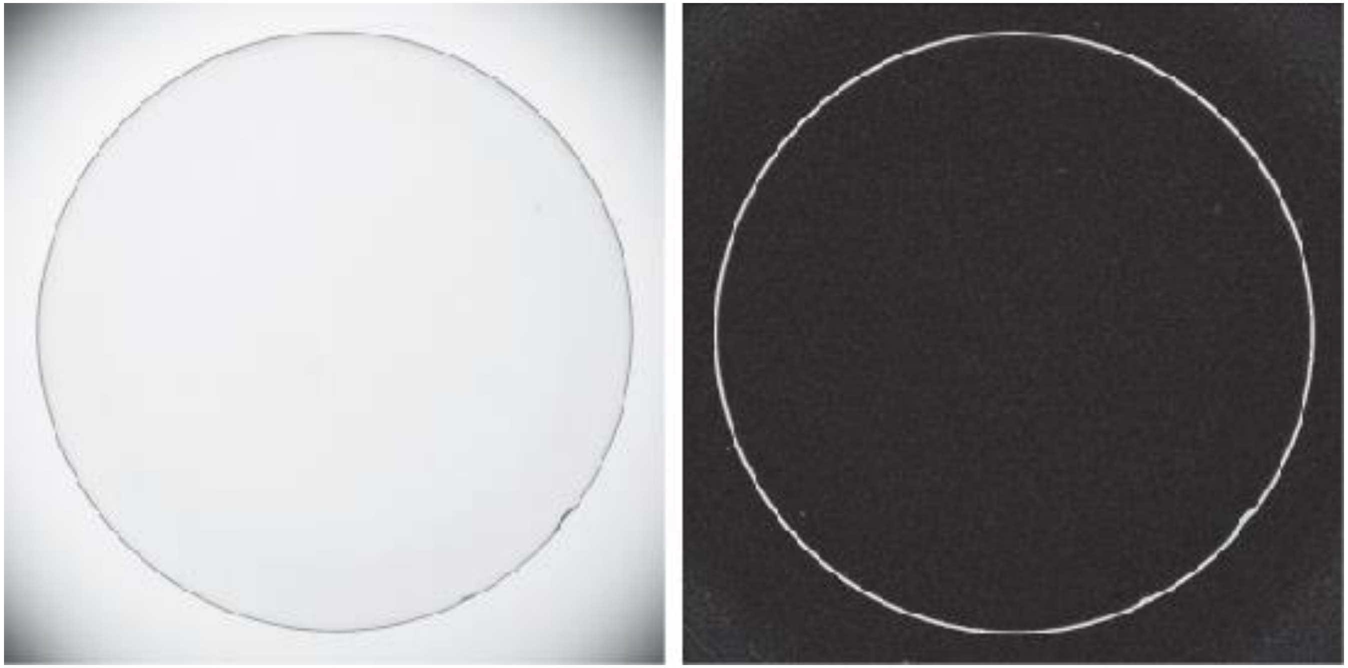
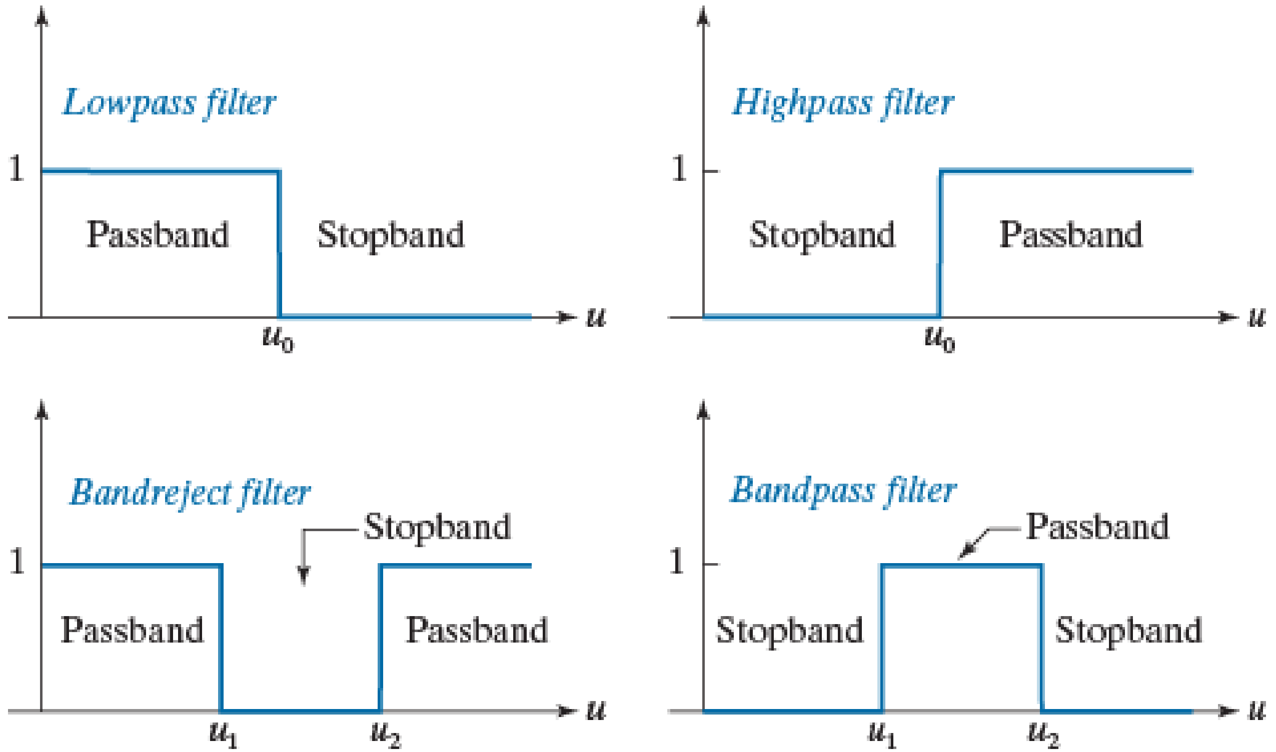


FIGURE 3.57
(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Perceptics Corporation.)

a b
c d

FIGURE 3.58
Transfer functions of ideal 1-D filters in the frequency domain (u denotes frequency).
(a) Lowpass filter.
(b) Highpass filter.
(c) Bandreject filter.
(d) Bandpass filter.
(As before, we show only positive frequencies for simplicity.)



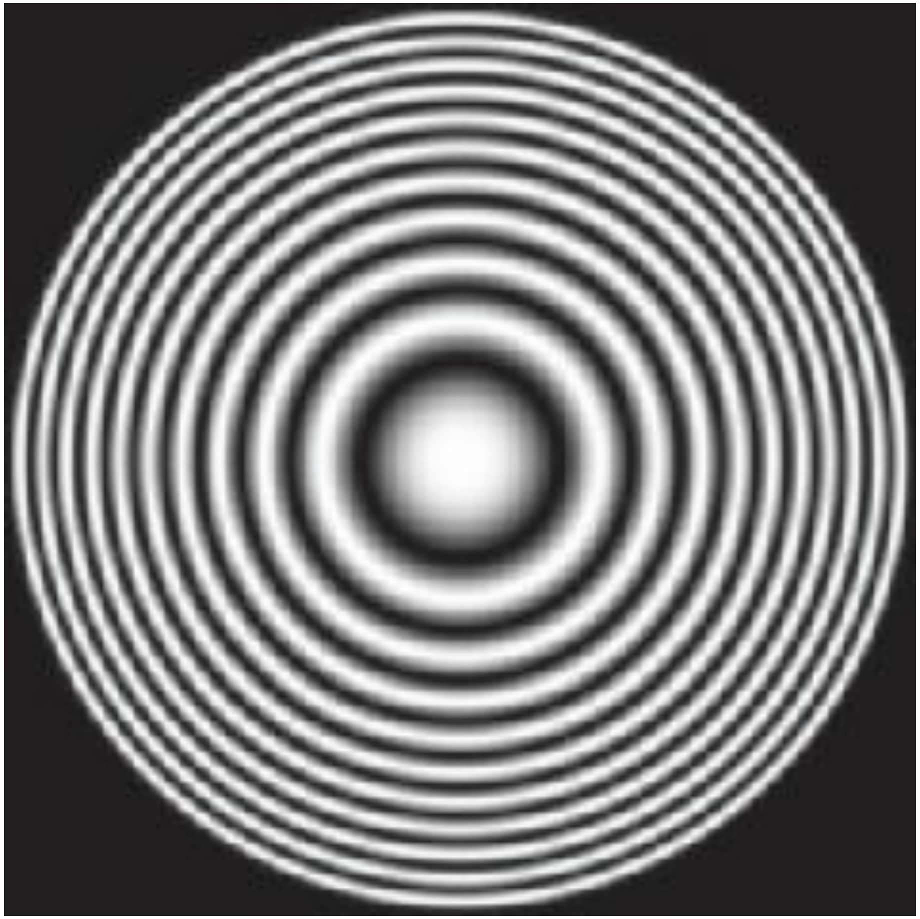
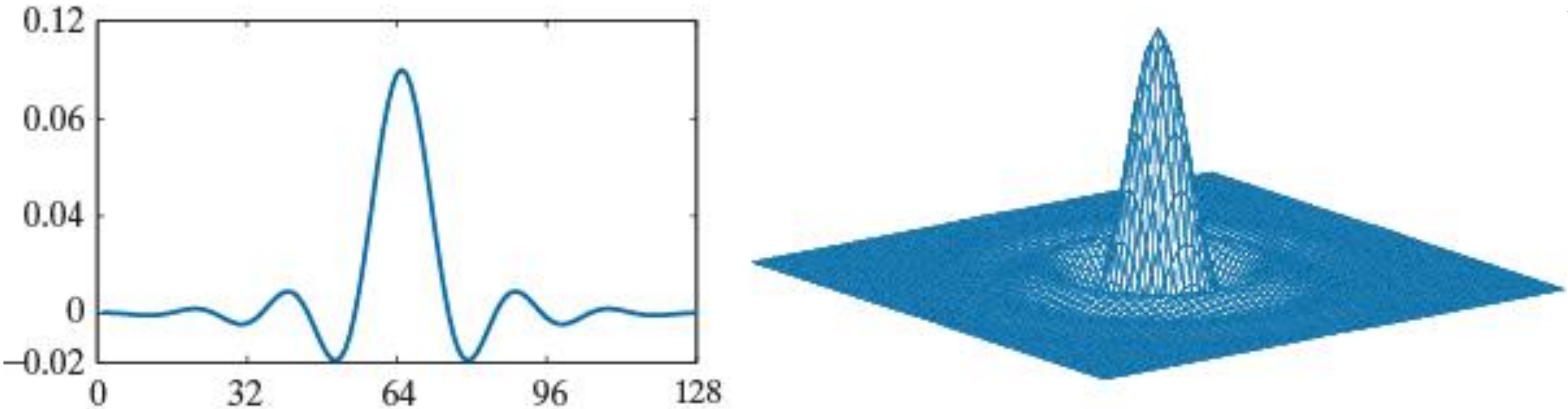
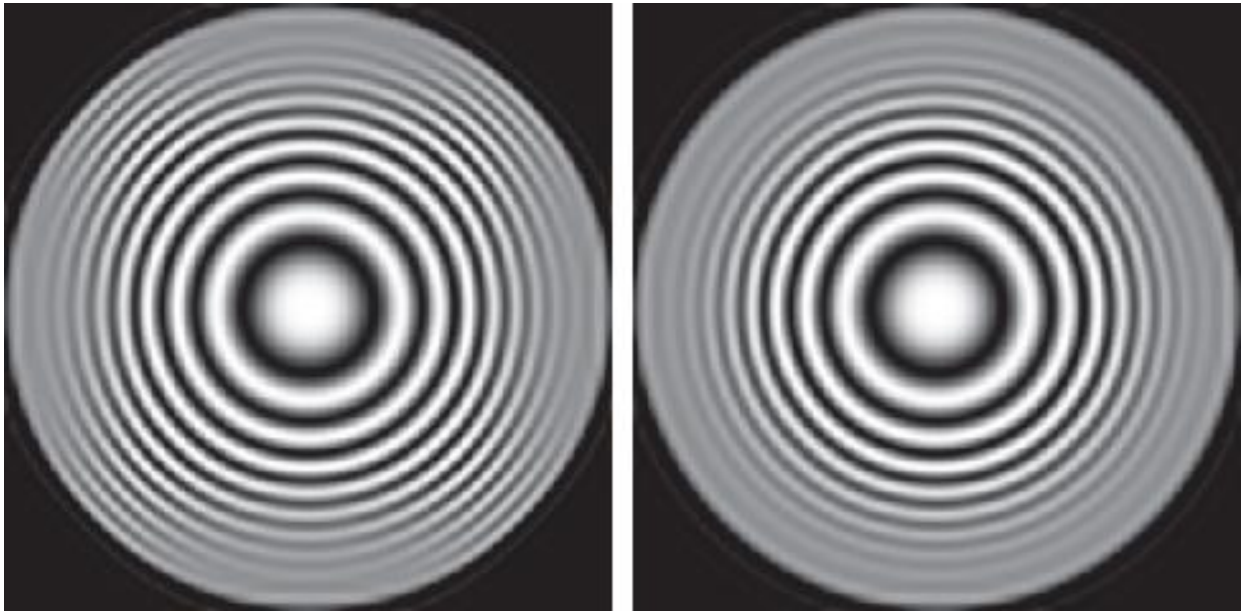


FIGURE 3.59
A zone plate image of size 597×597 pixels.



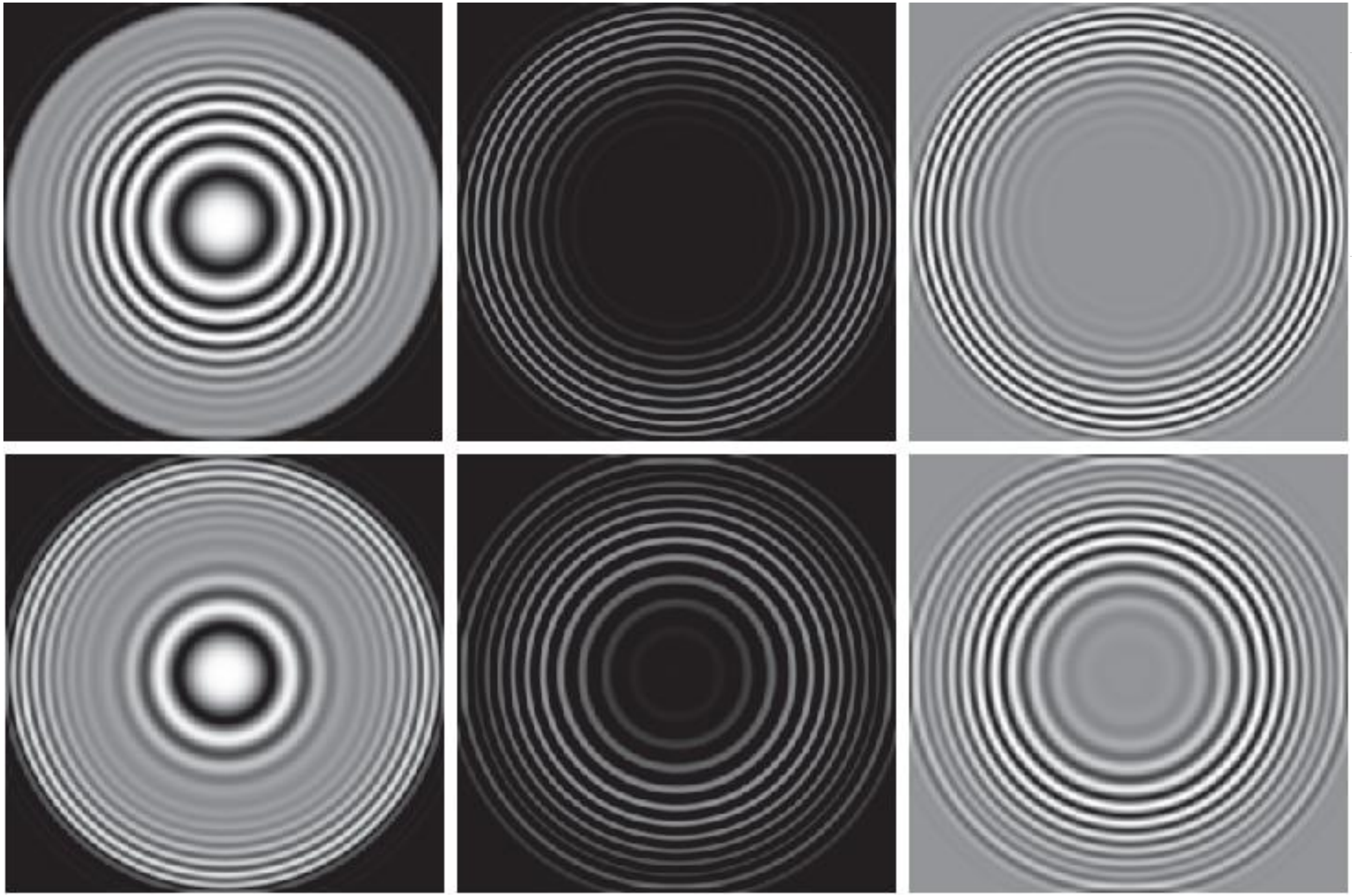
a b

FIGURE 3.60
(a) A 1-D spatial lowpass filter function. (b) 2-D kernel obtained by rotating the 1-D profile about its center.



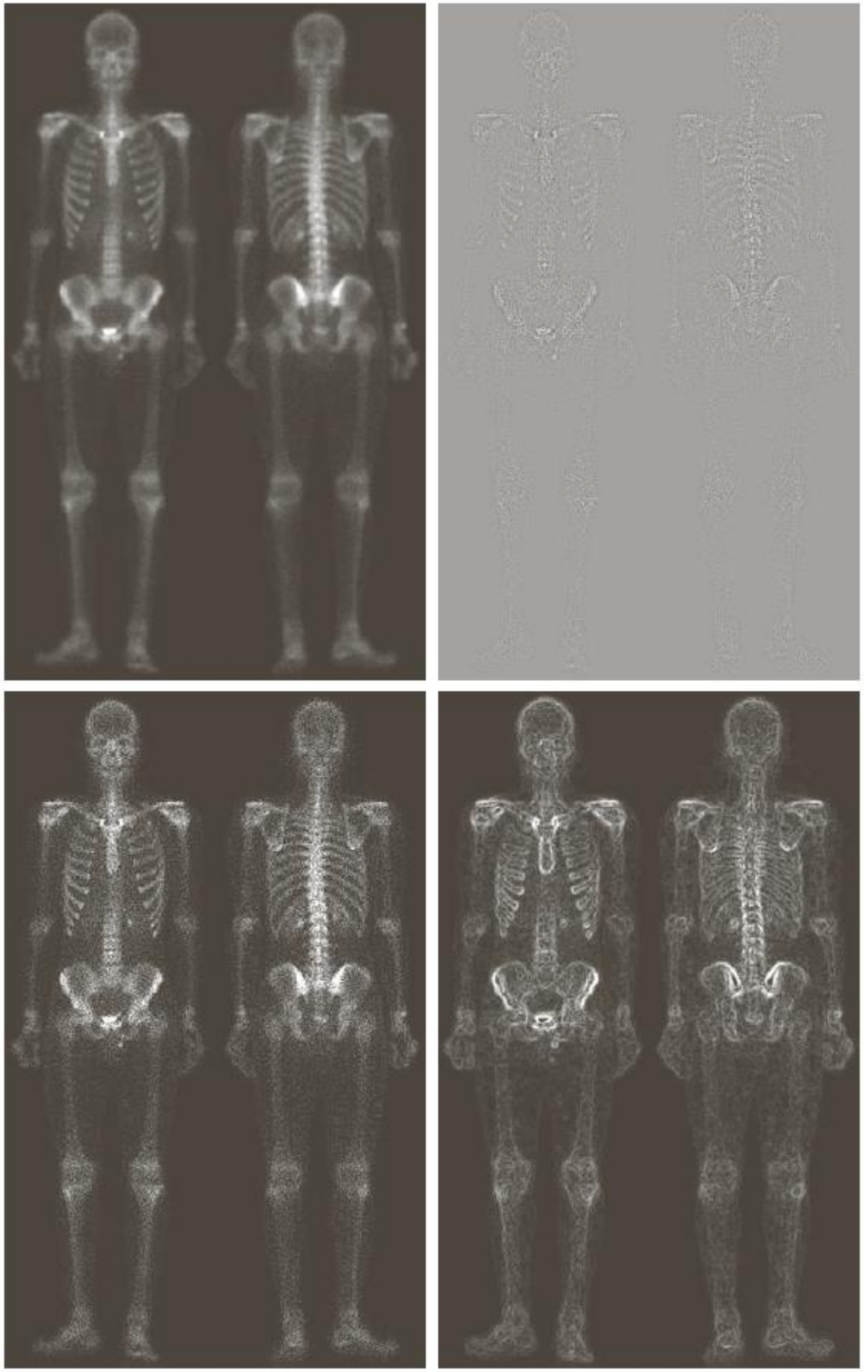
a b

FIGURE 3.61
(a) Zone plate image filtered with a separable lowpass kernel. (b) Image filtered with the isotropic lowpass kernel in Fig. 3.60(b).



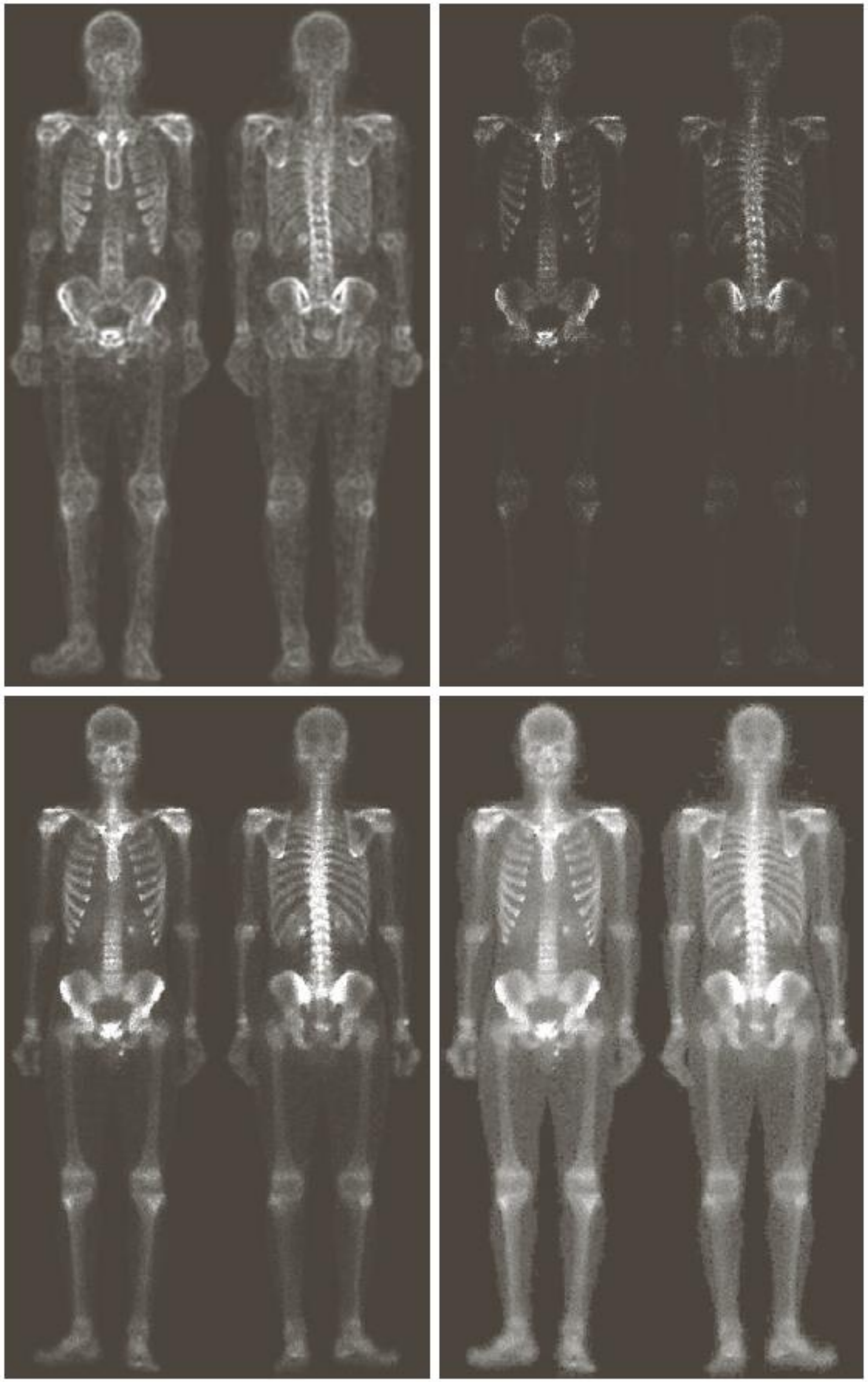
a	b	c
d	e	f

FIGURE 3.62
Spatial filtering of the zone plate image. (a) Lowpass result; this is the same as Fig. 3.61(b). (b) Highpass result. (c) Image (b) with intensities scaled. (d) Bandreject result. (e) Bandpass result. (f) Image (e) with intensities scaled.



a b
c d

FIGURE 3.63
(a) Image of whole body bone scan.
(b) Laplacian of (a).
(c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of image (a). (Original image courtesy of G.E. Medical Systems.)



e f
g h

FIGURE 3.63 *(Continued)* (e) Sobel image smoothed with a 5×5 box filter. (f) Mask image formed by the product of (b) and (e). (g) Sharpened image obtained by the adding images (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare images (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)