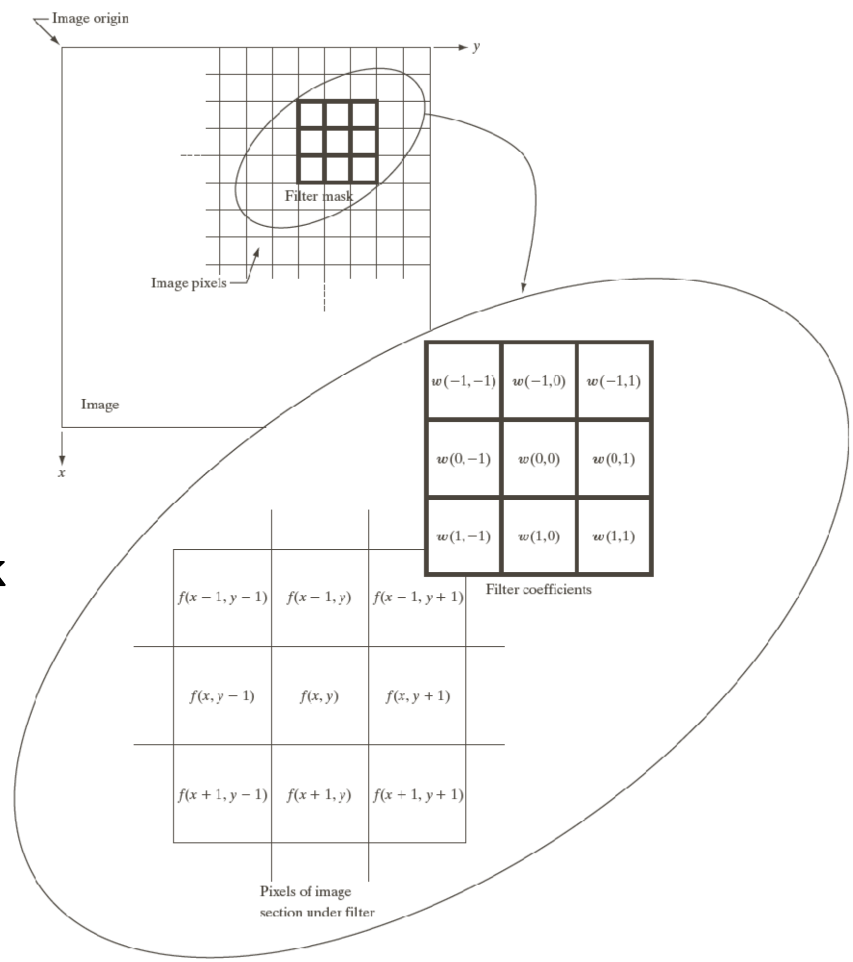
3.4 & 3.4.1 Fundamentals & mechanics of spatial filtering *(page 166)*

- **Spatial filter (mask)** • **Filter coefficients** • **Filter response**

Example: 3×3 **mask**



Linear spatial filtering

$$g(x,y) = w(-1,-1)f(x-1,y-1) + \cdots + w(1,1)f(x+1,y+1)$$

3.4.2 Spatial correlation and convolution *(page 168)* (1-dimensional)

	Correlation	Convolution	
	Origin f / w	Origin f / w rotated 180°	
(a)	0 0 0 1 0 0 0 0 / 1 2 3 2 8	0 0 0 **1** 0 0 0 0 / 8 2 3 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 3 2 1	(j)
	Zero padding		
(c)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(k)
(d)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after one shift	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(l)
(e)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after four shifts	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(m)
(f)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Final position	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(n)
	Full correlation result	Full convolution result	
(g)	0 0 0 8 2 3 2 1 0 0 0 0	0 0 0 1 2 3 2 8 0 0 0 0	(o)
	Cropped correlation result	Cropped convolution result	
(h)	0 8 2 3 2 1 0 0	0 1 2 3 2 8 0 0	(p)

FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

(2-dimensional)

Origin $f(x, y)$

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

$w(x, y)$

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	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	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	0	0
4	5	6	0	0	0	0	0	0
7	8	9	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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

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

(d)

Cropped 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

(e)

Rotated w

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

(f)

Full convolution result

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

(g)

Cropped 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

(h)

• **Convolving a function with a unit impulse yields a copy of the function at the location of the impulse (Chapter 4)**

• **If the filter mask is symmetric, correlation and convolution yield the same result**

• **If f contains a region identical to w, the value of the correlation function is a maximum when w is centered on that region. Application in pattern matching (Chapter 12)**

Equations for an image $f(x,y)$ and a general $m \times n$ filter mask $w(x,y)$:

$$\text{(Correlation)} \quad w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$\text{(Convolution)} \quad w(x,y) \bigstar f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Common terms used interchangeably: convolution filter/mask/kernel

3.4.3 Vector representation of linear filtering *(page 172)*

General linear filter: (Response)

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} \\ &= \sum_{k=1}^{mn} w_k z_k \\ &= \mathbf{w}^T \mathbf{z} \end{aligned}$$

3x3 example: (Response)

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 \\ &= \sum_{k=1}^{9} w_k z_k \\ &= \mathbf{w}^T \mathbf{z} \end{aligned}$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3.4.4 Generating spatial filter masks *(page 173)*

(Read)

- **Based on what we want to accomplish**

3.5 Smoothing spatial filters

3.5.1 Smoothing linear filters

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

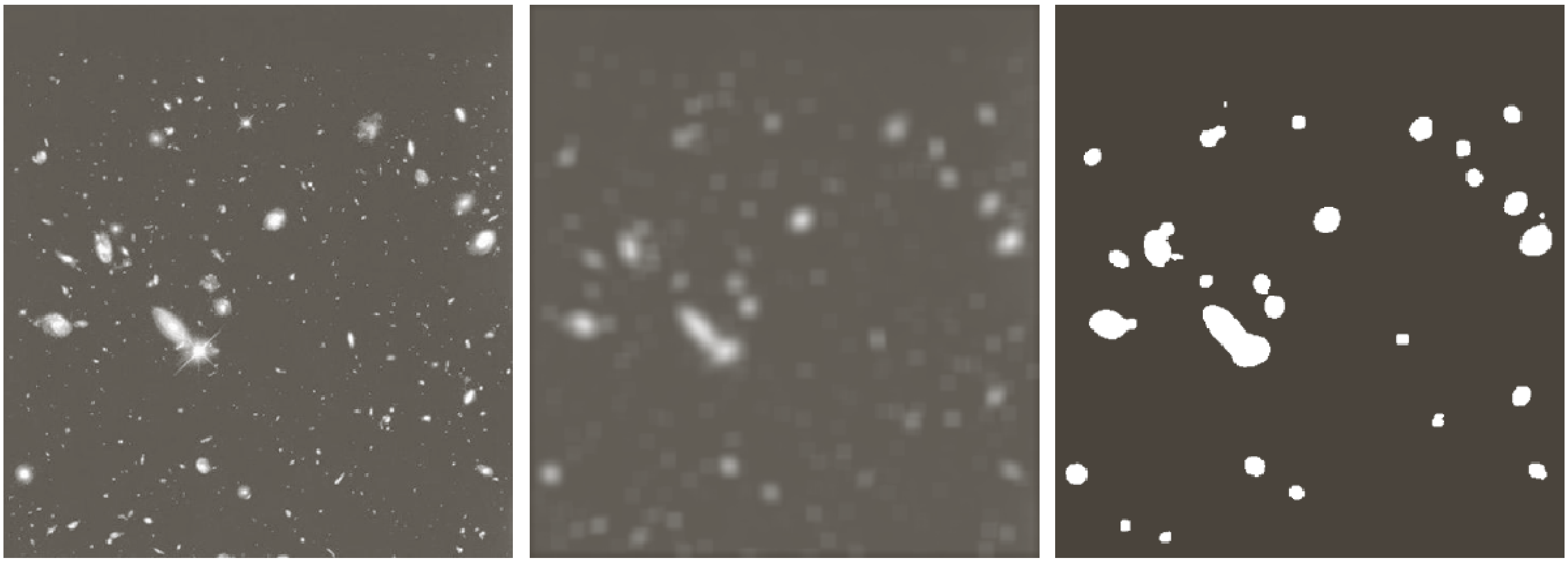
$$R = \frac{1}{9} \sum_{i=1}^{9} z_i$$

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

1x1 3x3
5x5 9x9
15x15 35x35

Example

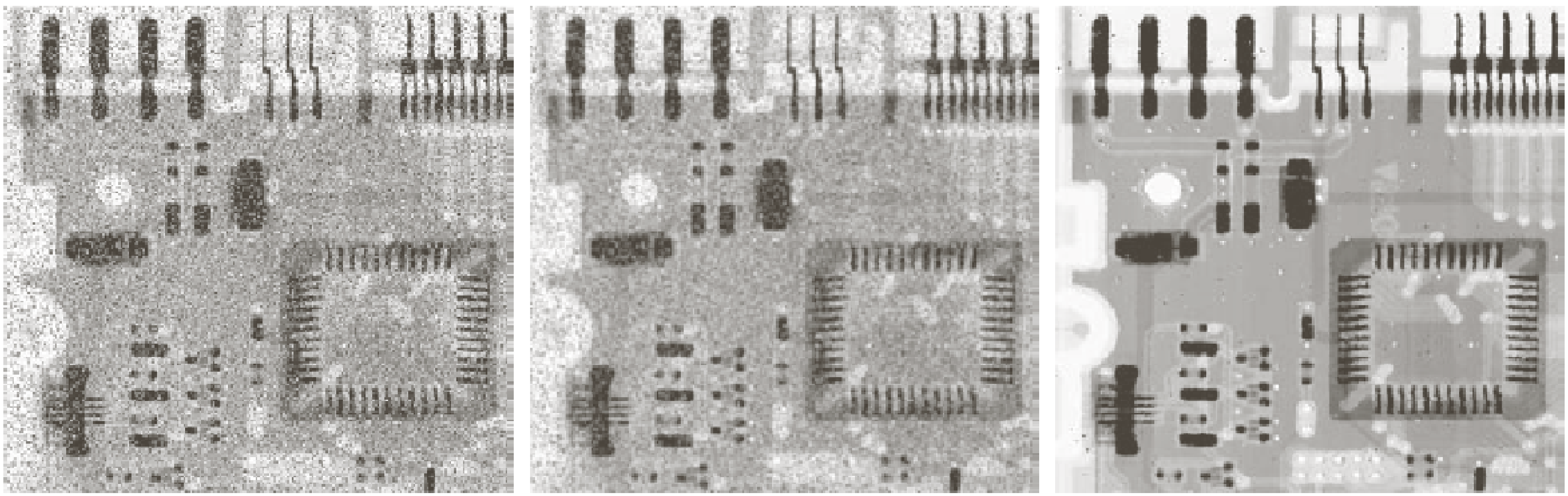


a b c

FIGURE 3.34 (a) Image of size 528×485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

3.5.2 Order-statistic nonlinear filters

- **Non-linear filters: Order (rank) pixels, e.g. median filter**
- **Noise reduction: Salt-and-pepper noise**



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

3.6 Sharpening spatial filters *(page 179)*

Purpose: highlight fine detail

Smoothing: pixel averaging: integration

Sharpening: differentiation: enhances edges and deemphasizes slowly varying areas

3.6.1 Foundation

One dimensional derivatives

Derivatives of digital functions: differences

Requirements for definitions:

First-order derivative

(1) Zero in flat segments
(2) Nonzero at onset of step or ramp
(3) Nonzero along ramp

Second-order derivative

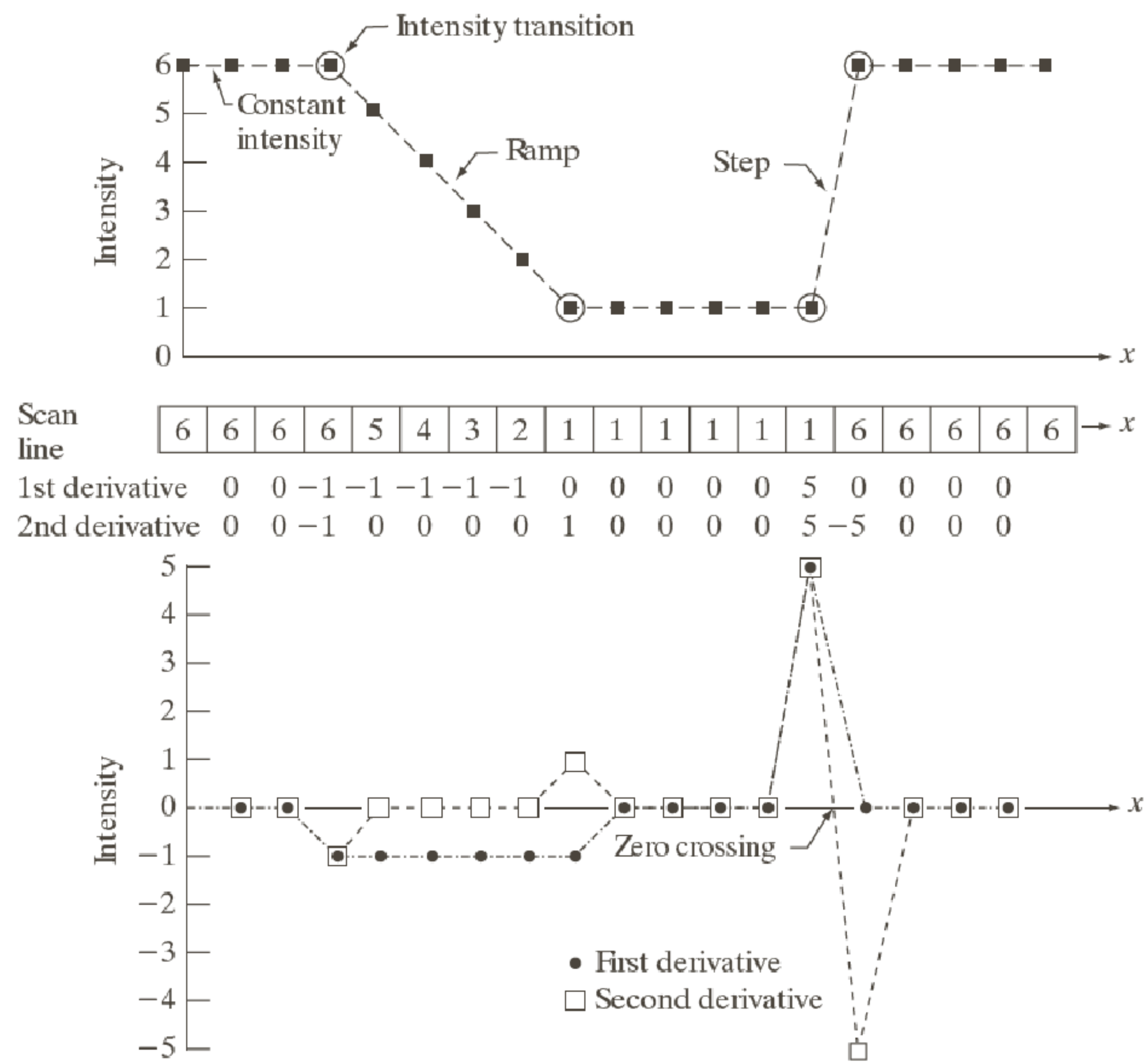
(1) Zero in flat segments
(2) Nonzero at onset of step or ramp
(3) Zero along ramp of constant slope

First-order derivative:

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

Second-order derivative:

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



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

- **First derivative $\rightarrow$ edge detection**
- **Second derivative $\rightarrow$ sharpening**

3.6.2: Using the second derivative for image sharpening - the Laplacian

Isotropic filters: Rotation invariant

Development of the method

Leplacian derivative operator

Continuous form:

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

Discrete form: x**-direction**

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

Discrete form: y**-direction**

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

Discrete form: 2-D Laplacian - sum of the two components

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Implementation

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 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

(a) and (c): Isotropic results for increments of $90^{\mathbf{o}}$
(b) and (d): Isotropic results for increments of $45^{\mathbf{o}}$

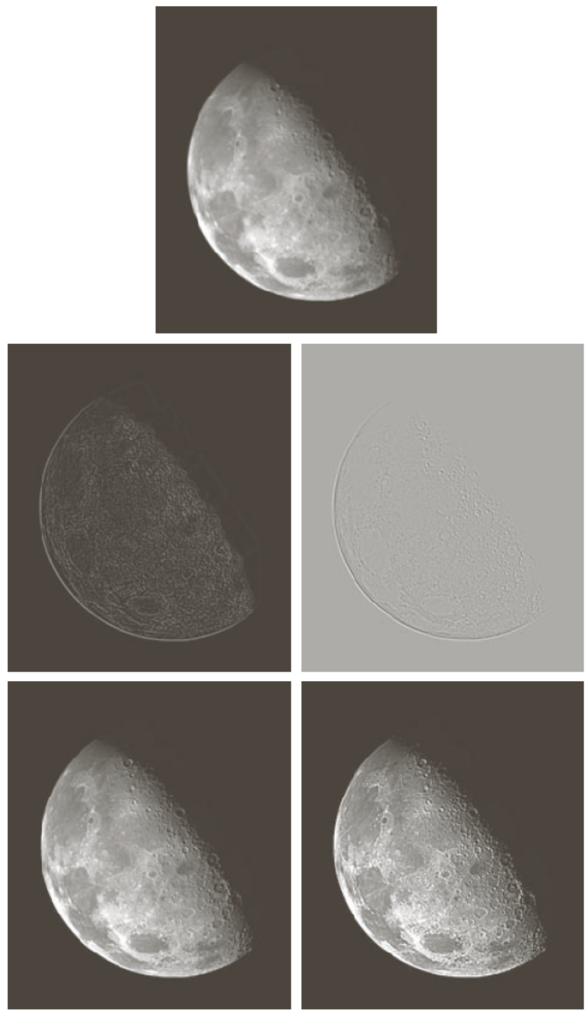
In order to recover background features, while still preserving the sharpening effect...

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

The constant is $c=-1$ if Fig 3.37 (a) or (b) is used, and $c=+1$ if (c) or (d) is used

Example 3.15

- **Image sharpening**



a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

3.6.3: Unsharp masking and highboost filtering

Unsharp masking

Used in printing industry

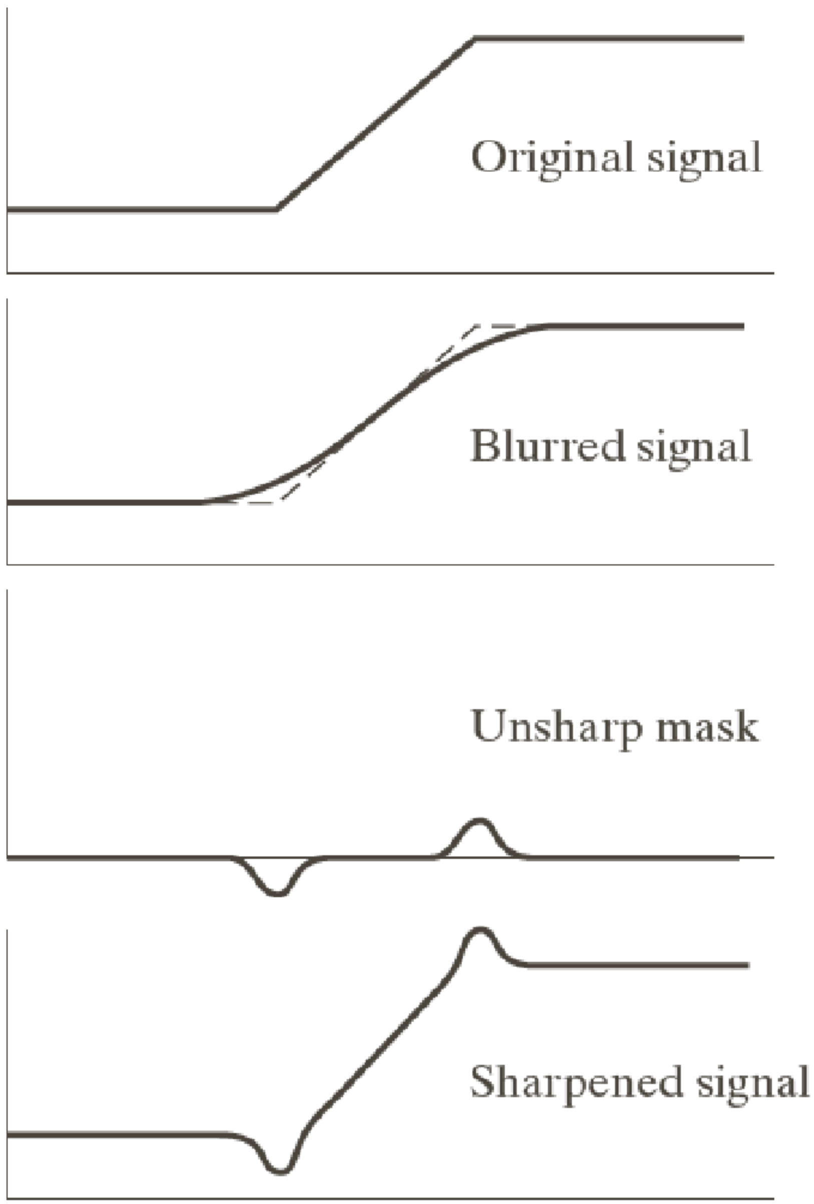
(a) Blur original image
(b) Subtract blurred image from original (difference is called mask)
(c) Add mask to original

$$g_{\text{mask}}(x,y) = f(x,y) - \underbrace{\overline{f}(x,y)}_{\textbf{Blurred}}$$

$$g(x,y) = f(x,y) + k \cdot g_{\text{mask}}(x,y)$$

Unsharp masking: Choose $k = 1$

Highboost filtering: Choose $k > 1$



a
b
c
d

FIGURE 3.39 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).

Example 3.16

a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

3.6.4: First-order derivatives for (nonlinear) image sharpening - the gradient

Continuous form: Gradient

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

Magnitude of gradient

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

Approximation: $M(x,y) \approx |g_x| + |g_y|$

Discrete form: Roberts: $M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$

Sobel:

$$\begin{aligned} M(x,y) \approx\ & |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ & + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \end{aligned}$$

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

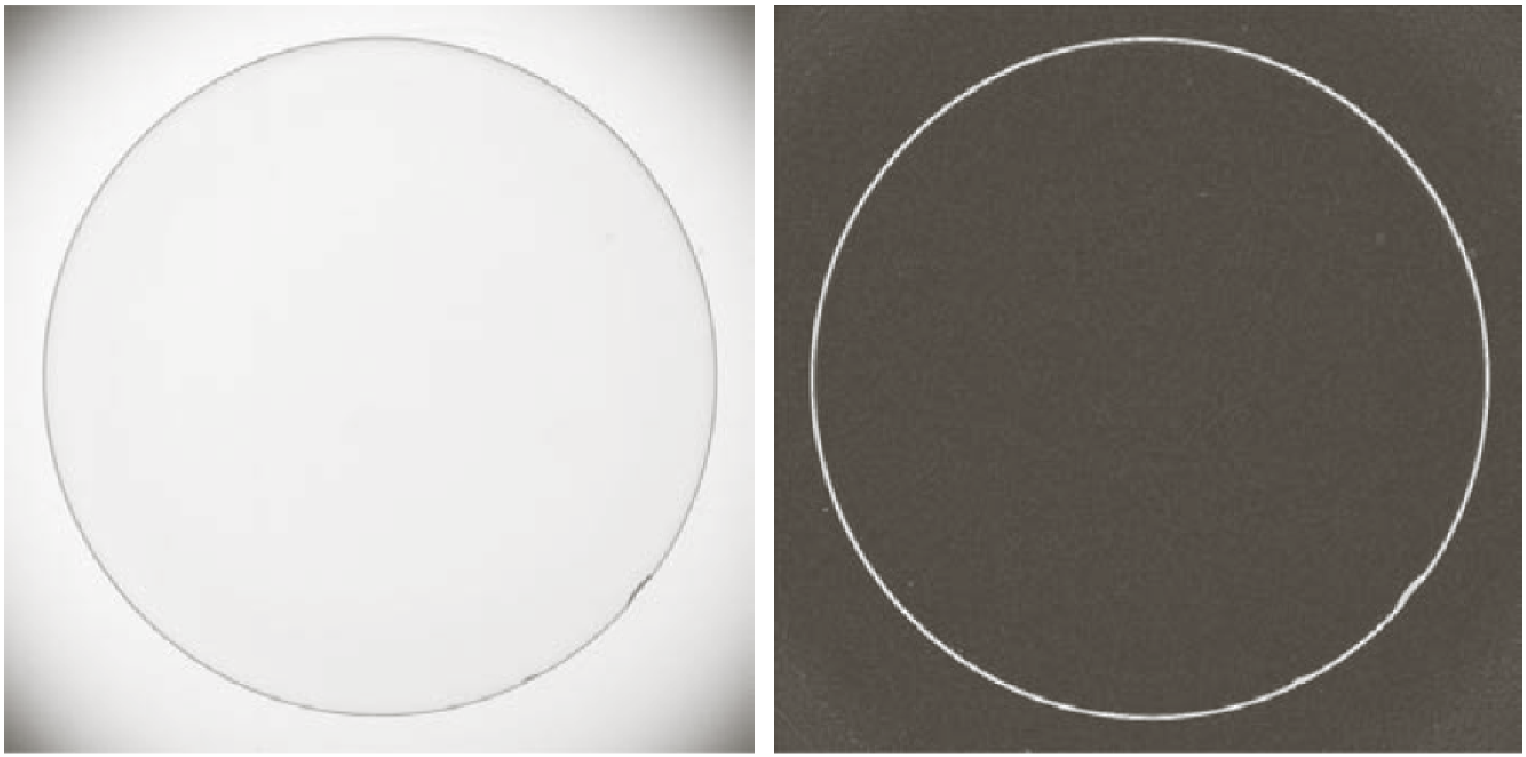
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

FIGURE 3.41 A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Example 3.17



a b

FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

3.7: Combining spatial enhancement methods...