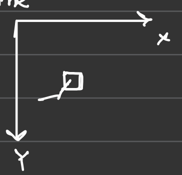
Fundamentals of spatial filtering.

we can perform processing under two principle domains

Image Processing

Spatial domain

we can perform processing directly on single pixel, present in image plane



Frequency domain

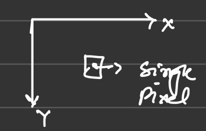
we can't perform any processing operation directly on the input image; we convert the original image into Fourier transform and we perform processing operations on that transformed image (Freq: domain) and then take the inverse Fourier transform to get back the original image.

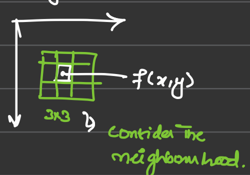
SPATIAL DOMAIN

Intensity transformation

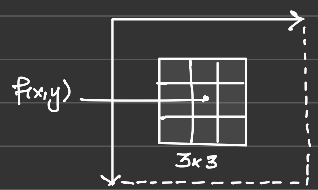
Spatial Filtering

1x1 correspondence b/w the spatial filtering and frequency domain processing.





3x3 "Filter" mask



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

coincide ..

h(-1,-1)	h(-1,0)	h(-1,1)
h(0,-1)	h(0,0)	h(0,1)
h(1,-1)	h(1,0)	h(1,1)

can be of size (3x3, 5x5, 7x7, ...)

3x 3 mask (filter, Template, window, kernel)

- If the operation perform on the neighbouring pixels is linear, it is called "linear spatial filter".

Response can be written as:

$$R = h(-1,-1)f(x-1,y-1) + h(-1,0)f(x-1,y) + h(-1,1)f(x-1,y+1) \ldots$$
$$\ldots\ h(1,1)f(x+1,y+1).$$

$$\boxed{g(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} h(i,j) f(x+i, y+j)} \longrightarrow (1)$$

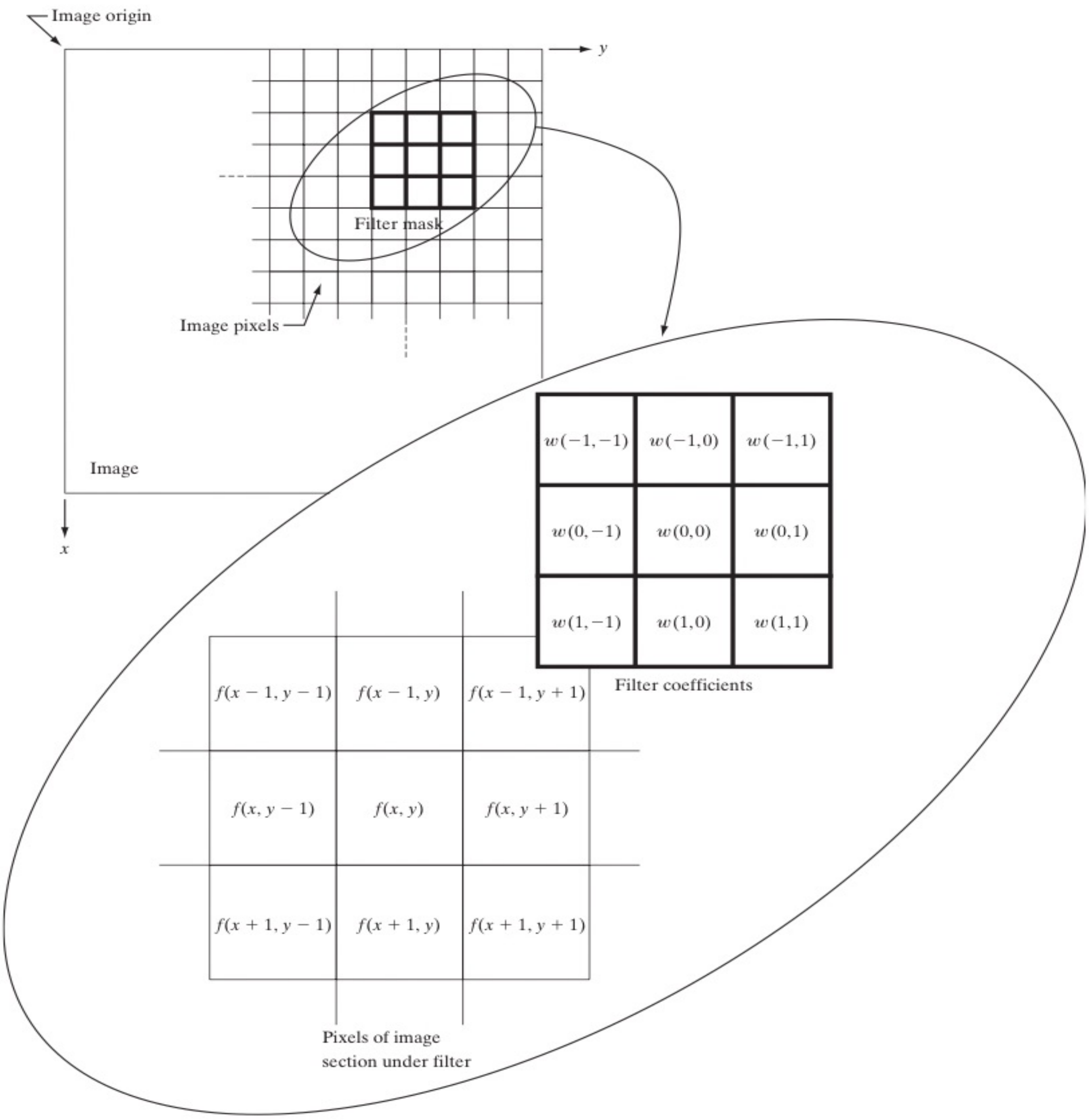


FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Where $m \times n \rightarrow$ size of the kernel & $M \times N \rightarrow$ size of the image

$$a = \frac{m-1}{2}, \quad b = \frac{n-1}{2} \qquad \therefore x = 0, 1, \ldots M-1, \quad y = 0, 1, \ldots N-1$$

- **CORRELATION AND CONVOLUTION IN SPATIAL FILTERS:**

Lets assume 1-D function be :

origin → 0 0 0 1 0 0 0 0

h
1 2 3 2 8

ZP ZP

0000 0 0 0 1 0 0 0 0 0000

1 2 3 2 8

h = 1 2 3 2 8

m = 5

m-1 = 4

No. of zero padding is required on both sides of the function

Full correlation : $\boxed{0\,0\,0\,8\,2\,3\,2\,1\,0\,0\,0\,0}$

crop correlation : $\boxed{0\,8\,2\,3\,2\,1\,0\,0}$

180° shift of the mask

CONVOLUTION:

f : 00010000

h : 8 2 3 2 1

m = 5

m-1 = 4 ZP

ZP ZP

0000 0 0 0 1 0 0 0 0 0000

8 2 3 2 1

Full convolution : $\boxed{0\,0\,0\,1\,2\,3\,2\,8\,0\,0\,0\,0}$

crop " : $\boxed{0\,1\,2\,3\,2\,8\,0\,0}$

2D Function:

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

h : 3x3

1 2 3

4 5 6

7 8 9

ZP :

m = 3 m-1 = 2

n = 3 n-1 = 2

ZP = 2x2

f(x,y) for the correlation:

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

Correlation:

moved

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

Full correlation results :

$$\begin{matrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&9&8&7&0&0&0 \\ 0&0&0&6&5&4&0&0&0 \\ 0&0&0&3&2&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

Full Convolution Result :

$$\begin{matrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&2&3&0&0&0 \\ 0&0&0&4&5&6&0&0&0 \\ 0&0&0&7&8&9&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{matrix}$$

Cropped correlation

$$\begin{matrix} 0&0&0&0&0 \\ 0&9&8&7&0 \\ 0&6&5&4&0 \\ 0&3&2&1&0 \\ 0&0&0&0&0 \end{matrix}$$

Cropped convolution

$$\begin{matrix} 0&0&0&0&0 \\ 0&1&2&3&0 \\ 0&4&5&6&0 \\ 0&7&8&9&0 \\ 0&0&0&0&0 \end{matrix}$$

General expression is :

Correlation : $h(s,t) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} h(s,t) f(x+s, y+t).$

Convolution : $h(s,t) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} h(s,t) f(x-s, y-t).$

Vector representation of the Spatial filters :

Suppose we have a mask of a size 3×3.

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

3×3 mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixels behind this given mask

3×3 neighborhood

Response : $h_1 z_1 + h_2 z_2 + \ldots\ldots + h_{mn} z_{mn}.$

$$\boxed{R = \sum_{i=1}^{mn} h_i z_i}$$

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

Spatial filters:

Now see the classifications of the spatial filters.
There are two types of spatial filters.

1. Smoothing spatial filters.
2. Sharpening spatial filters.

Smoothing spatial filter can further be classified as:
a. Linear filters.
b. non-Linear filter (order-statistics)

1: SMOOTHING SPATIAL FILTERS:

It can be used to remove the noise present in the original image.

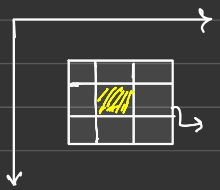
- To blur the image
- Extract the object in the image.
- Remove small objects in the image.
- Bridge the gap between the lines & curves present in the image.

LINEAR SMOOTHING SPATIAL FILTERS:

It is primarily used to:

- create the blurness &
- remove the noise present in the image.
- also called as "averaging spatial filter".

The response of this filter is averages the intensity values of its neighbouring pixels and assign it to the center pixel. This operation is perform on pixel to pixel by shifting it.



→ This finds the average of its neighboring pixels and assign it to the center pixel. This mainly provides the blurness and smoothens the image.
This mainly uses in the applications like dealing random noises in an image.

These random noises will be having very sharp intensity transition. In order to smooth these sharp intensity transitions, we have to find out the average of these neighbours by using Linear smoothing filters. It limitation is that it blurs the edges of the objects (important feature of an image).

To illustrate this if we consider a mask of a size 3x3,

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

normalized value & usually it is considered as "mn"

$$\boxed{R = \frac{1}{9}\sum_{i=1}^{9} h_i z_i}$$ → For this particular case.

In weighted linear smoothing filter, we normalized the mask by the weight of its coefficients. e.g.

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \boxed{R = \frac{1}{16}\sum_{i=1}^{9} h_i z_i}$$

Non-Linear Filter: Mainly used to remove the noise present in the image, also provides the considerable amount of blurness and it is primarily used in the case, where we are dealing with the impulse noise (salt & pepper noise).

In non-linear filter, we use median.

	20	

→ calculate the median of the mask and assign it to the center pixel → repeated for all the pixels in an image.

For example, if an image is

$\{10, 25, 20, 20, 20, 100\ 20, 20\ 15\}$

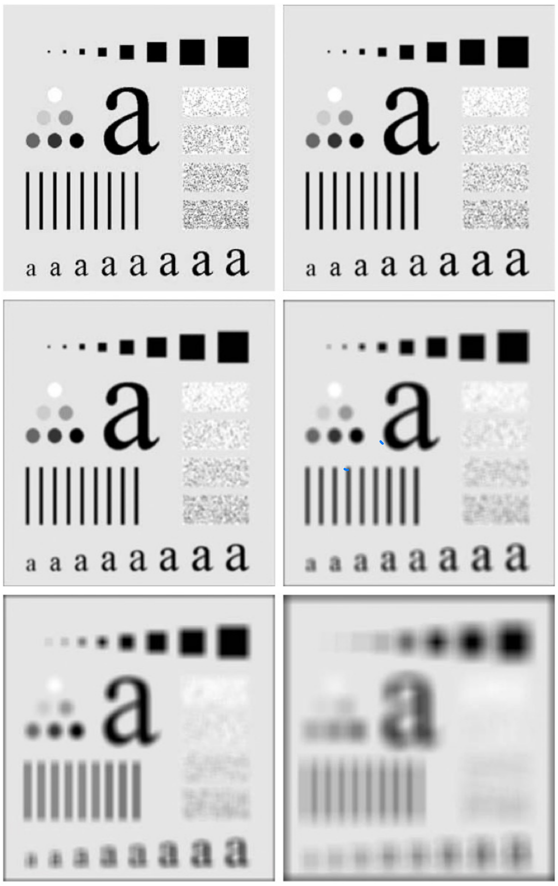
Sorted: $\{10, 15, 20, 20, \underline{20}, 20, 20, 25, 100\}$

MAX FILTER:- In max filter, the center pixel value will be the max value of the neighbourhood.

$$R = \text{Max}\ \{z_k \mid k = 1, 2, \ldots\ mn\}$$

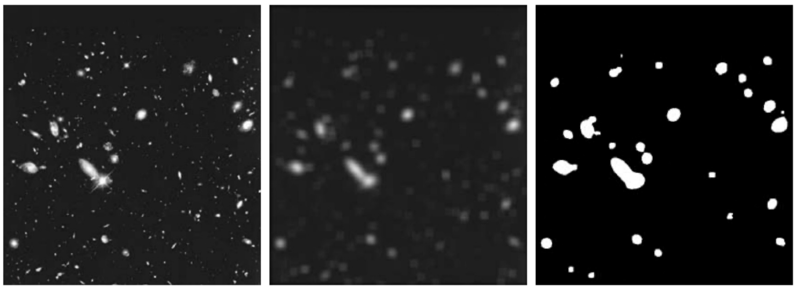
MIN FILTER :-

$$R = \text{Min}\ \{z_k \mid k = 1, 2, \ldots\ mn\}.$$



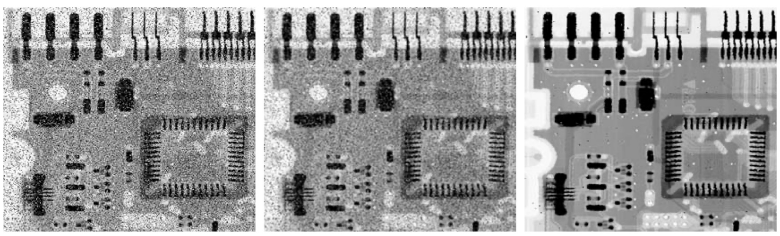
a b
c d
e f

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes m = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.



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.)



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.)

SHARPENING SPATIAL FILTERS: it is used to highlight the transitions in intensities. It performs the reverse operation of the smoothing spatial filters. In smoothing filter, we blur the image and in sharpening filter, we deblur / highlight the intensities in the image. In smoothing spatial filter, the blurness is accomplished by averaging the pixels of the neighbourhood. Averaging is analogous to the integration operation. In sharping spatial filtering, we perform "spatial differentiation".

Sharpening spatial filters are used mainly for:

- Highlight fine details.
- Enhance the blur image.
- Enhance the edges.

Sharpening spatial filter is usually used in the applications like:

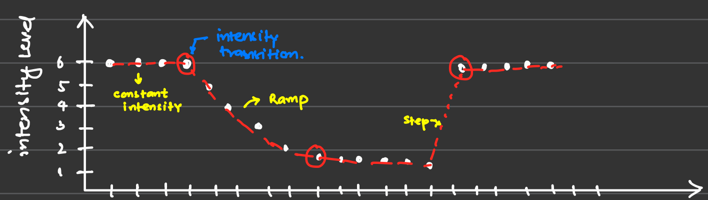
a. Electronic printing.
b. Medical imaging
c. Industrial inspection.
d. Autonomous guidance in military systems.

- a) First-order derivative:

Let assume a 1-D function.

The 1st-order derivative for 1-D function is given by:

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



Scan Line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st-order $\frac{\partial f}{\partial x}$	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0	0	0		
2nd-order $\frac{\partial^2 f}{\partial x^2}$	0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0		

1) So, for constant intensities, the $\frac{\partial f}{\partial x}$ is zero.
2) The onset (beginning) of ramp and steps must be non-zero.
3) Along the ramp, the $\frac{\partial f}{\partial x}$ must be non-zero.

b) Second-order derivative:-

The 2nd-order derivative for a 1-D function is given by:

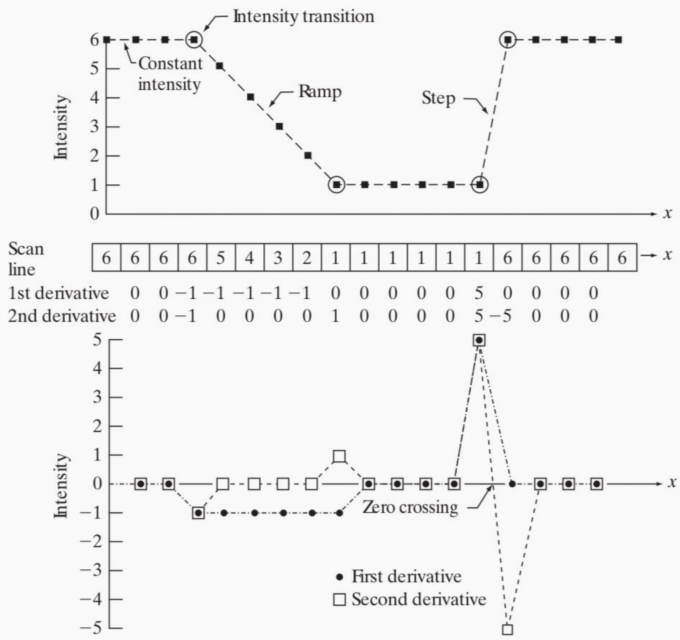
$$\frac{\partial^2 f}{\partial x^2} = f'(x+1) - f'(x)$$

$$= f(x+2) - f(x+1) - f(x+1) + f(x)$$

$$= f(x+2) + f(x) - 2f(x+1) \rightarrow (a)$$

subtracting "1" from (a), we get

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



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.

1) For constant intensities, the 2^{nd} order derivative is "zero"
2) The onset / end of ramp / step must be non-zero.
3) Along with ramp / constant slope, the 2^{nd}-order derivative must be zero.

Using the 2^{nd}-order derivatives for image sharpening – The Laplacian:

We will learn how to develop the laplacian mask using 2^{nd}-order derivative for a 2-D image.
Lets take a 2-D function.

$f(x, y) \rightarrow$ Isotropic Image (Rotation invariant)

The 2^{nd}-order derivative for a 2D $f(x,y)$ is given by:

$$\nabla^2 f = \underbrace{\frac{\partial^2 f}{\partial x^2}}_{x\text{-dir}} + \underbrace{\frac{\partial^2 f}{\partial y^2}}_{y\text{-dir}} \rightarrow (1)$$

In x-direction:

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

In y-direction:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \rightarrow (3)$$

Substitute (2) & (3) in eq (1)

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

This eq (4) is the discrete laplacian formulation.

$$\begin{bmatrix} f(x-1,y-1) & f(x-1,y) & f(x-1,y+1) \\ f(x,y-1) & f(x,y) & f(x,y+1) \\ f(x+1,y-1) & f(x+1,y) & f(x+1,y+1) \end{bmatrix}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

To find out the coefficients of the mask, the center pixel $z_5 = f(x,y)$

* Laplacian mask (a)

0	1	0
1	-4	1
0	1	0

· In sharpening spatial filter, we need to know that the sum of mask coefficient must be equal to "zero 0"

(b)

1	1	1
1	-8	1
1	1	1

→ considering diagonal elements.

(c)

0	-1	0
-1	4	-1
0	-1	0

(d)

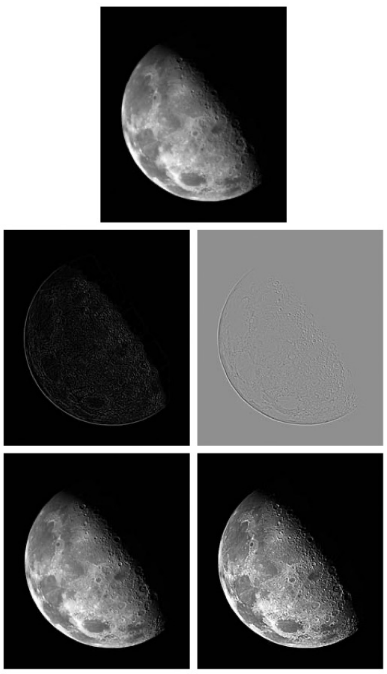
-1	-1	-1
-1	8	-1
-1	-1	-1

General expression to find the Laplacian mask:

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$

if $c = -1$ then the Laplacian masks are (a) & (b)
and if $c = 1$ then the " " " (c) & (d)

3.6 ■ Sharpening Spatial Filters 163



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.)