Digital Image Processing

Lecture # 5
Sharpening Filters & Edge Detection

Smoothing Spatial Filters

Simply average all of the pixels in a neighbourhood around a central value

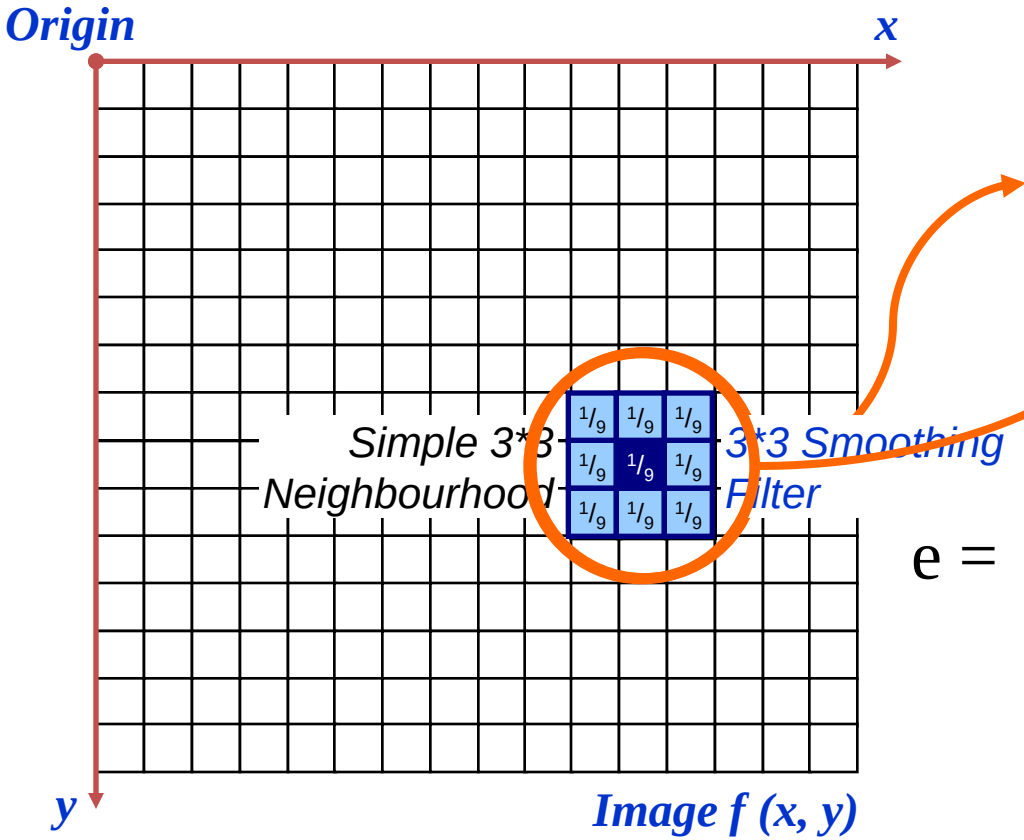
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Simple averaging filter

Smoothing Spatial Filters

- For blurring/noise reduction
- Blurring is usually used in preprocessing steps, e.g., to remove small details from an image prior to object extraction, or to bridge small gaps in lines or curves
- Equivalent to Low-pass spatial filtering in frequency domain because smaller (high frequency) details are removed based on neighborhood averaging (averaging filters)

Smoothing Spatial Filters



104	100	108
99	106	98
95	90	85

Original Image Pixels

*

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Filter

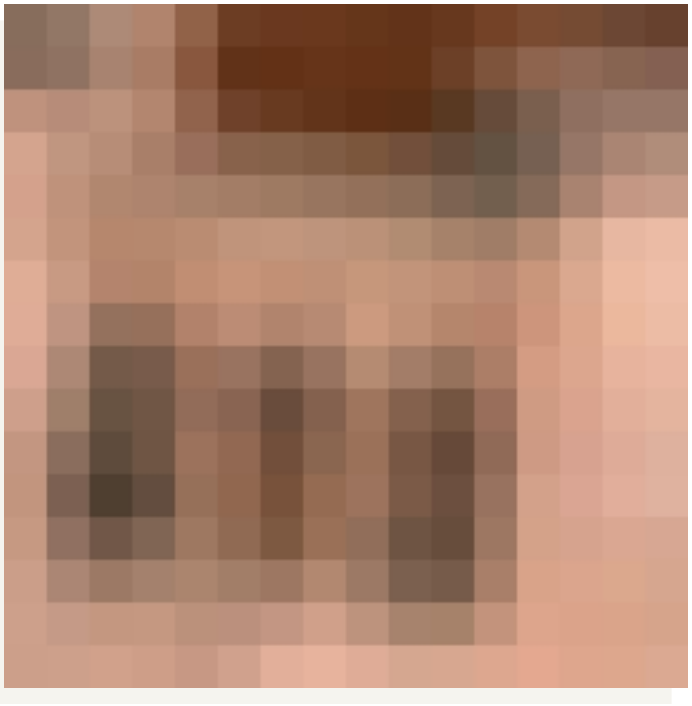
$$e = \tfrac{1}{9}*106 + \tfrac{1}{9}*104 + \tfrac{1}{9}*100 + \tfrac{1}{9}*108 + \tfrac{1}{9}*99 + \tfrac{1}{9}*98 + \tfrac{1}{9}*95 + \tfrac{1}{9}*90 + \tfrac{1}{9}*85$$

$$= 98.3333$$

The above is repeated for every pixel in the original image to generate the smoothed image

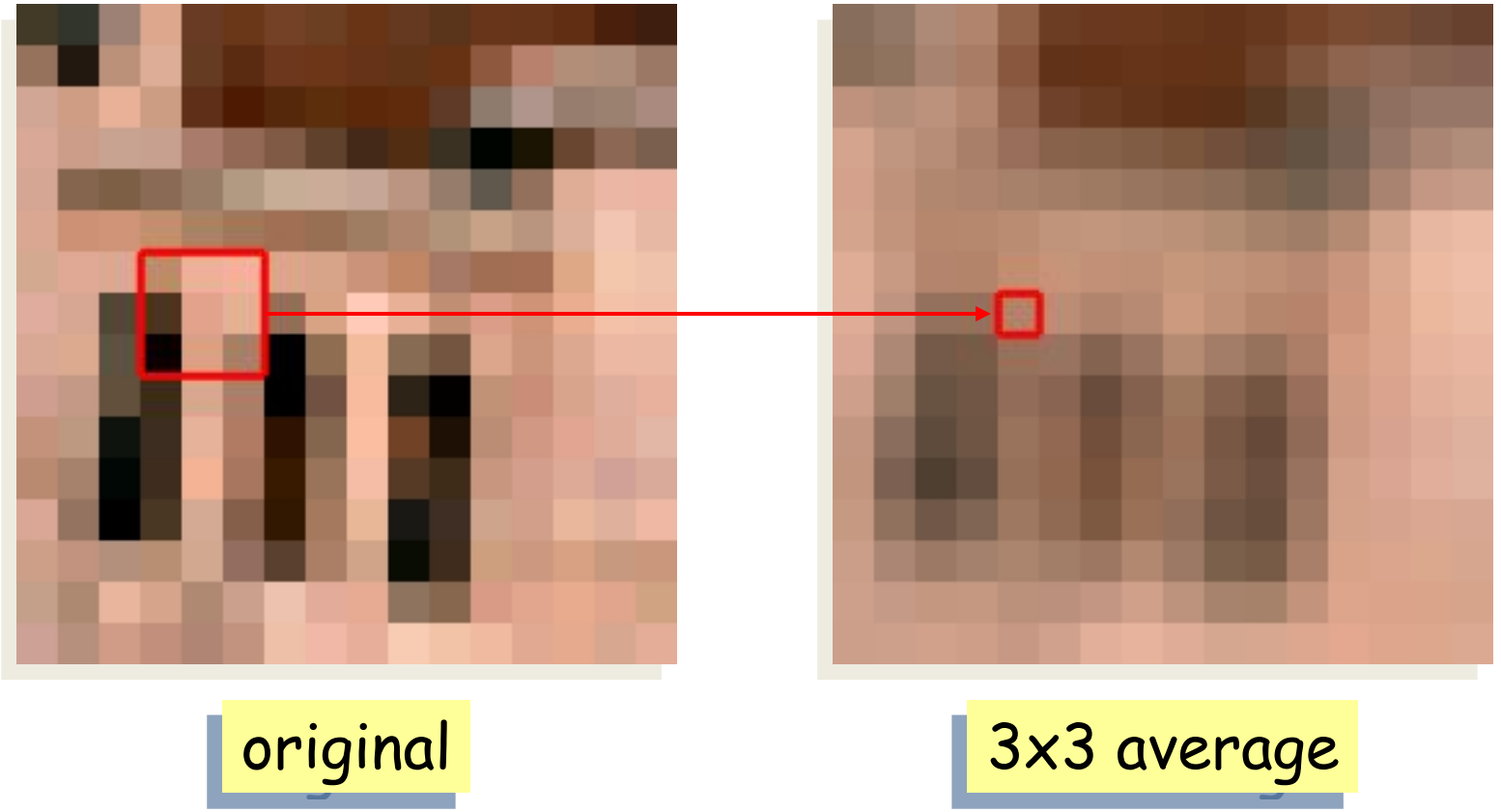
Smoothing Filter: Example

original



3x3 average

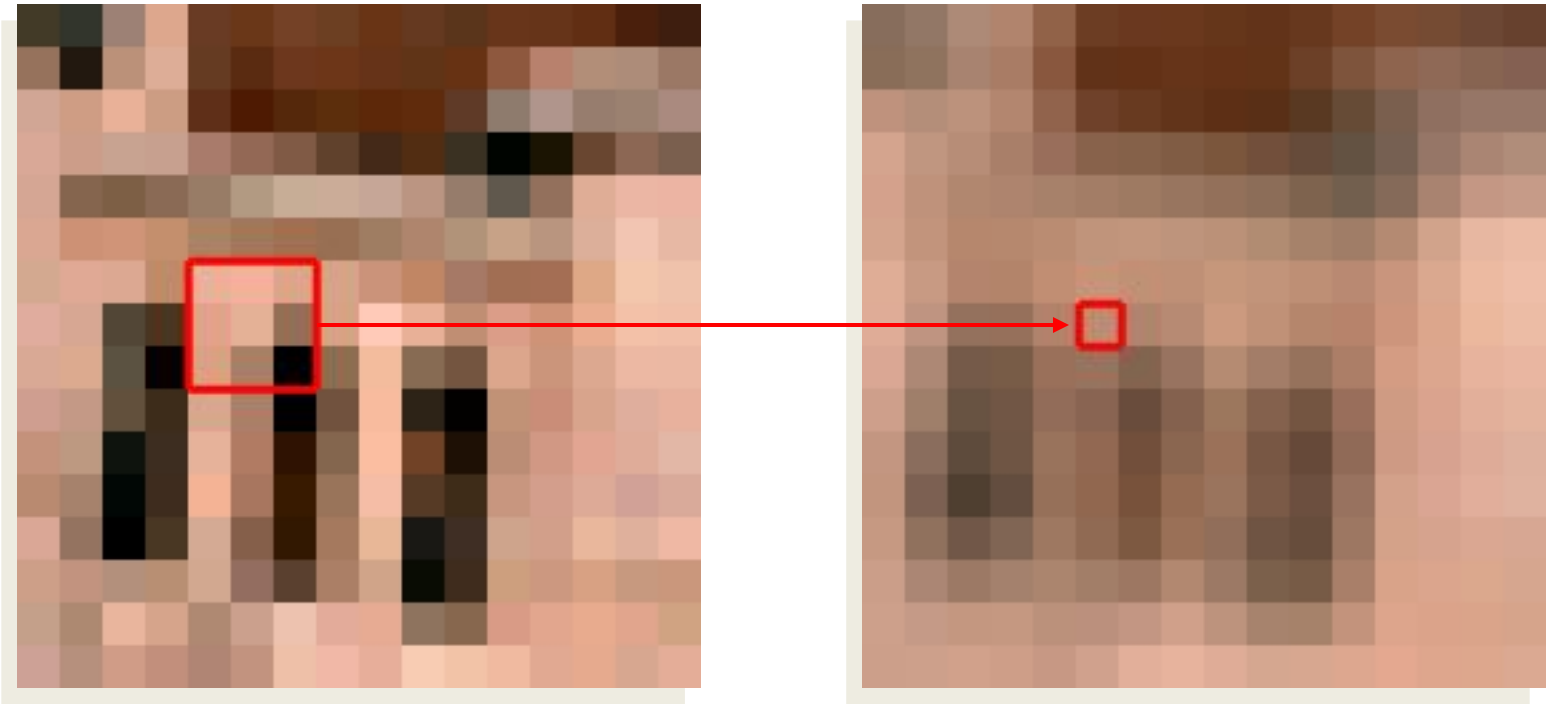
Smoothing Filter: Example



original

3x3 average

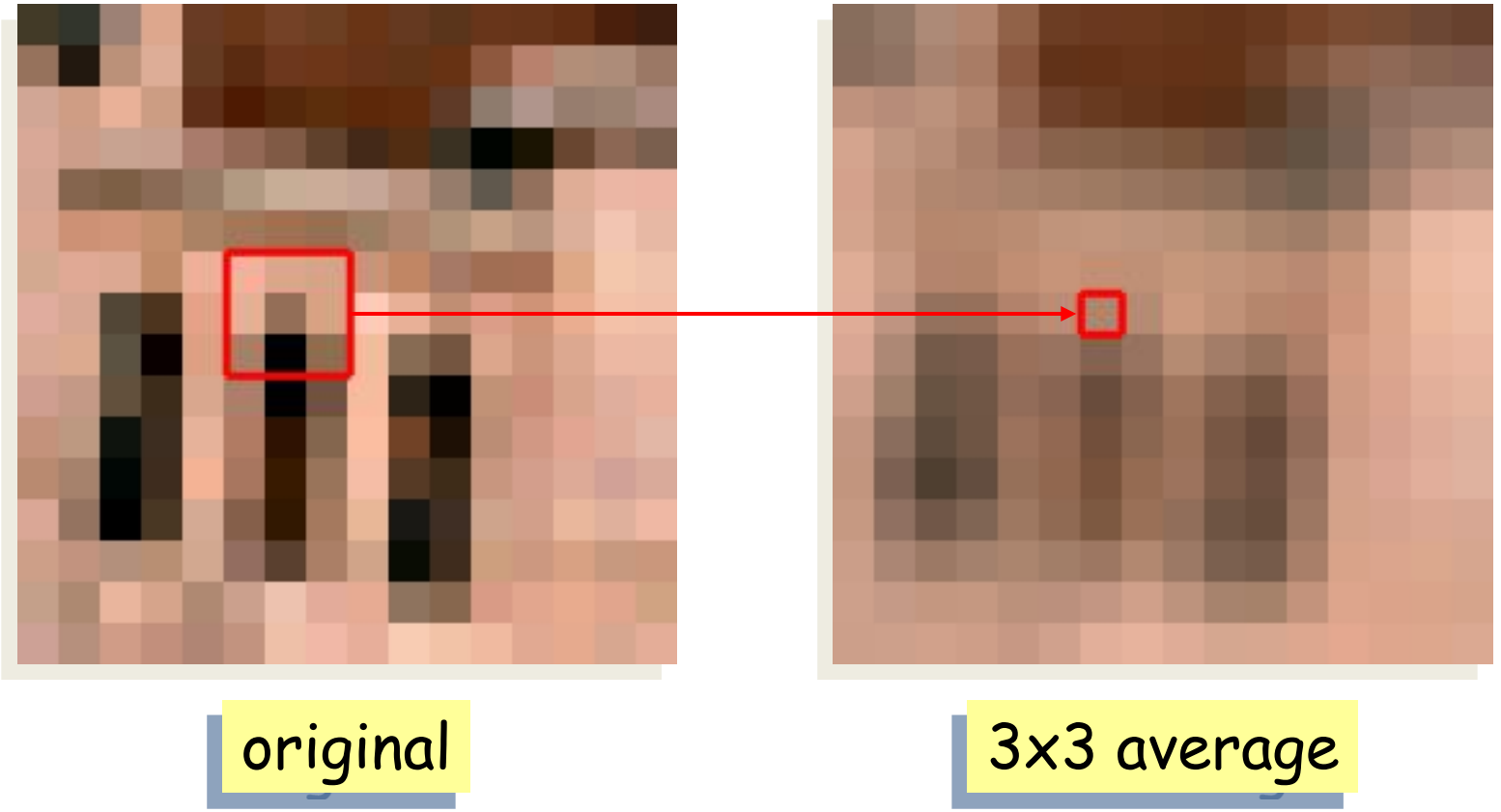
Smoothing Filter: Example



original

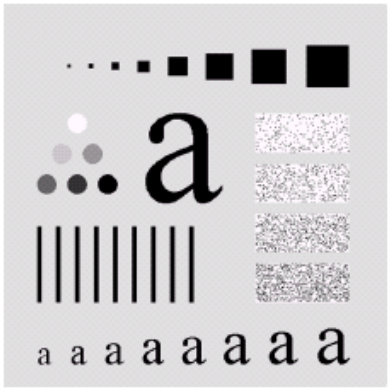
3x3 average

Smoothing Filter: Example



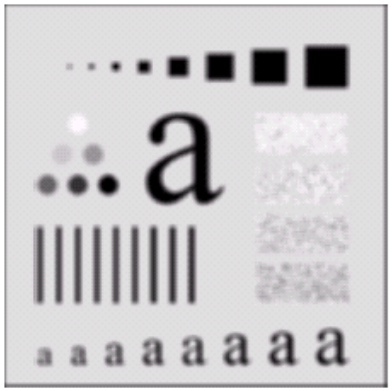
Original image

Size: 500x500



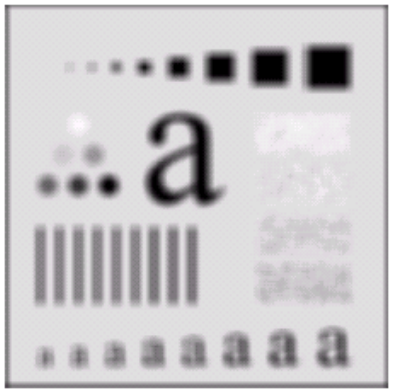
Smooth by 3x3 box filter

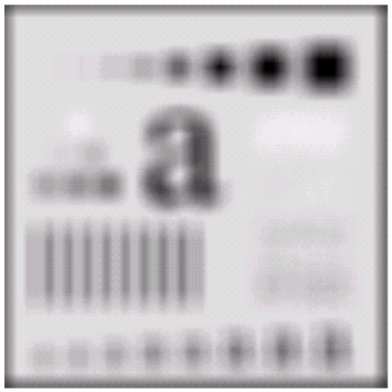
Smooth by 5x5 box filter



Smooth by 9x9 box filter

Smooth by 15x15 box filter

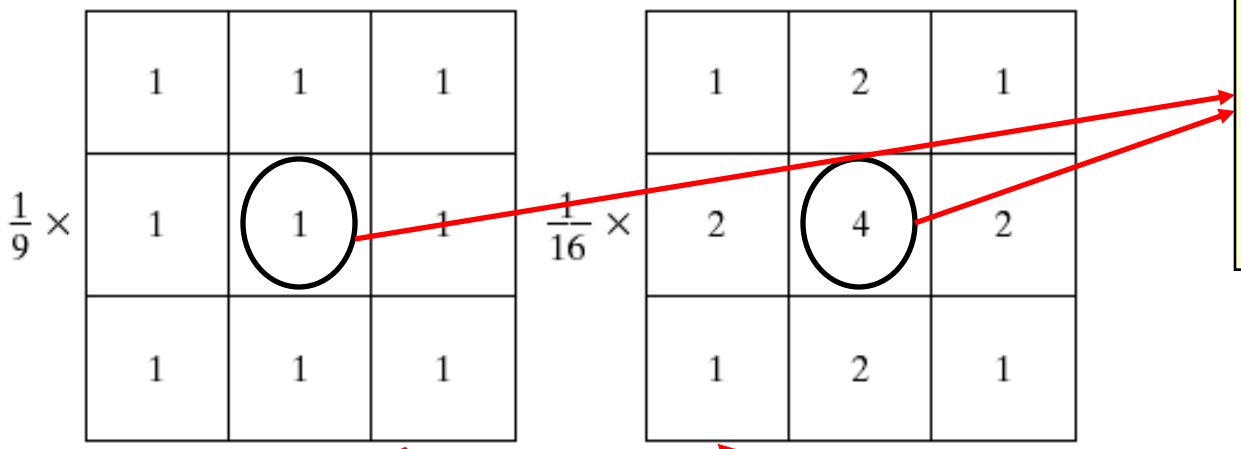




Smooth by 35x35 box filter

Notice how detail begins to disappear

Smoothing Spatial Filters



Consider the output pixel is positioned at the center

Box Filter all coefficients are equal

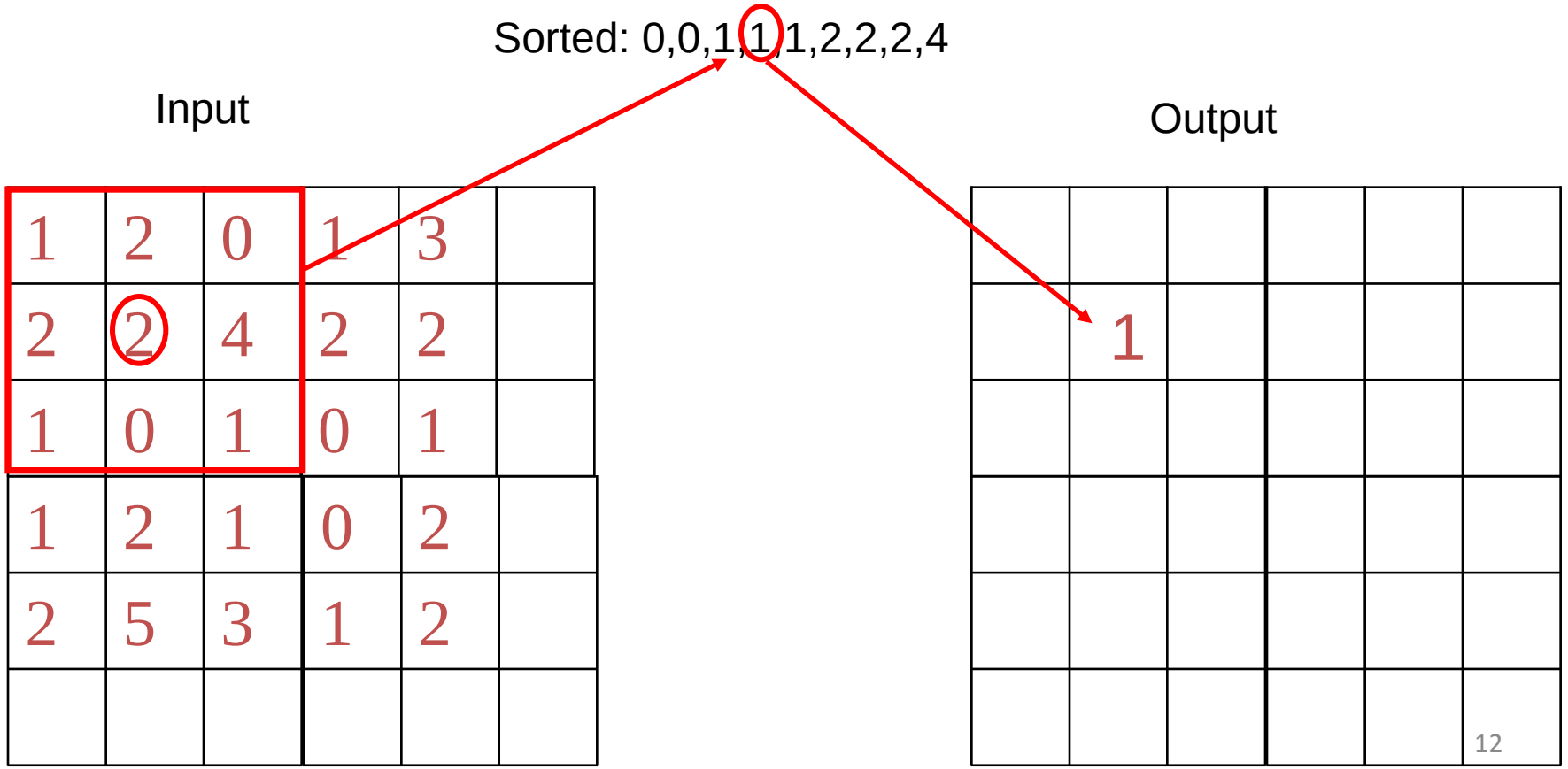
Weighted Average give more (less) weight to near (away from) the output location

Order-Statistic Filtering

- Output is based on order of gray levels in the masked area
- Some simple neighbourhood operations include:
  - **Min:** Set the pixel value to the minimum in the neighbourhood
  - **Max:** Set the pixel value to the maximum in the neighbourhood
  - **Median:** The median value of a set of numbers is the midpoint value in that set

Median Filter

- For an image, mask symmetric: 3x3, 5x5, etc.



Median Filtering

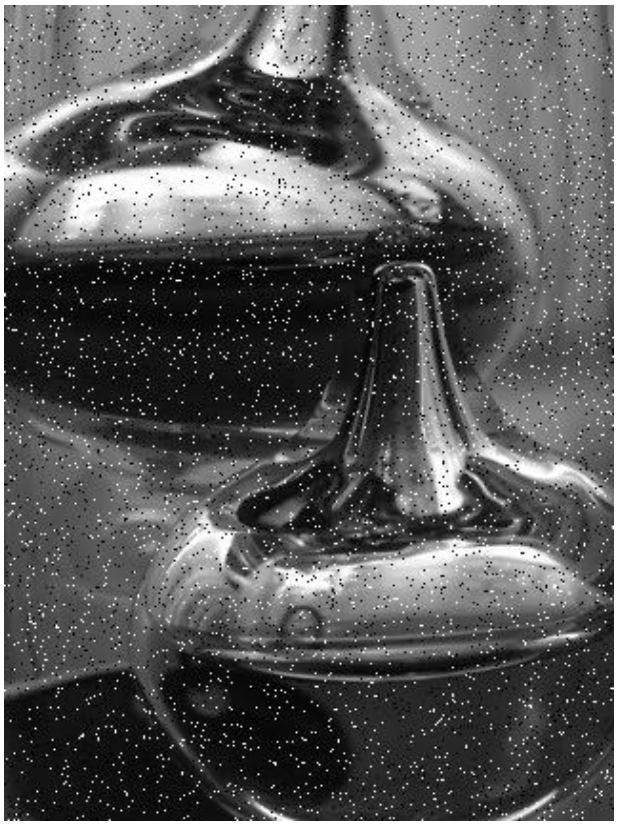
10	20	20
20	15	20
20	25	100

Sort the values
Determine the median

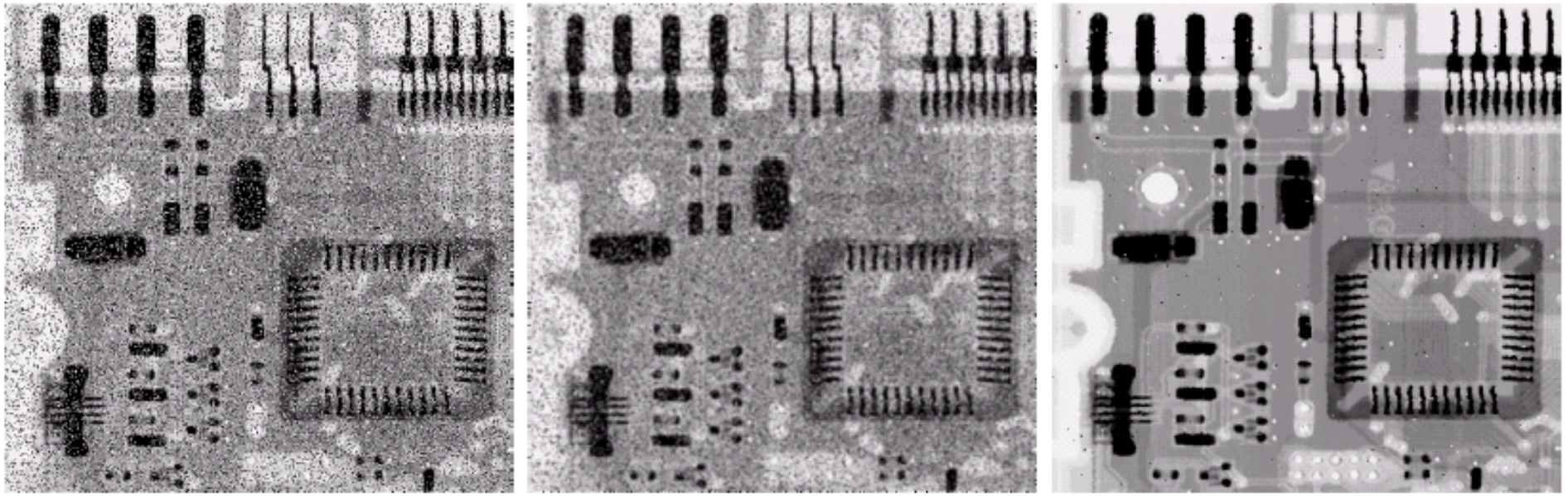
Median = ? 20

- Particularly effective when
  - The noise pattern consists of strong impulse noise (salt-and-pepper)

Salt and Pepper Noise



Median Filtering



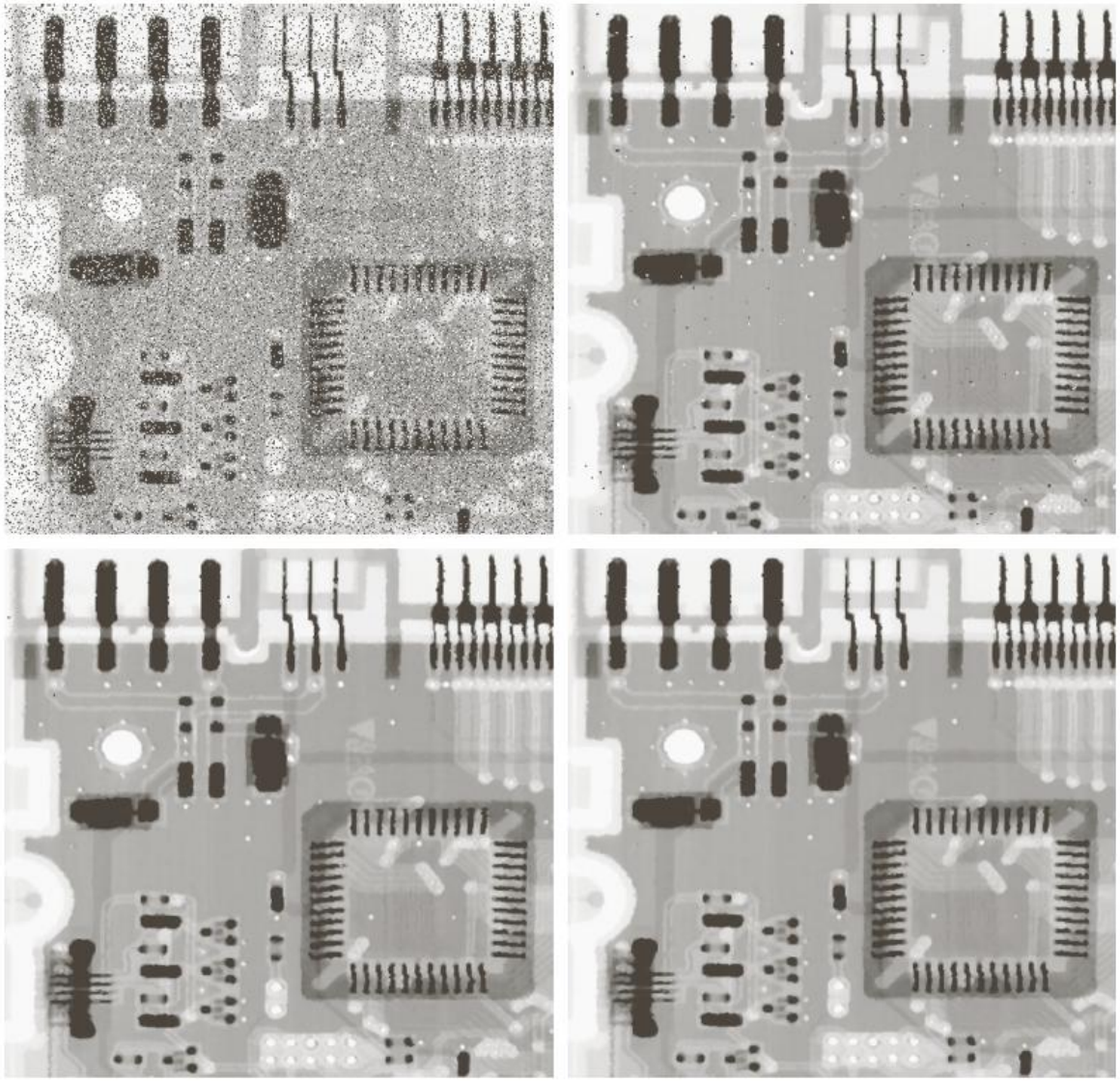
a b c

FIGURE 3.37 (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.)

Min/Max Filtering

a b
c d

FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.
(b) Result of one pass with a median filter of size 3×3.
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.



Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

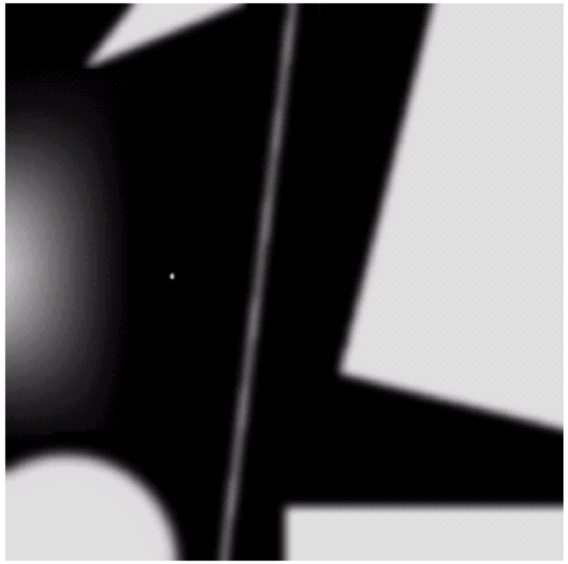
Sharpening spatial filters seek to highlight fine detail

- Remove blurring from images
- Highlight edges

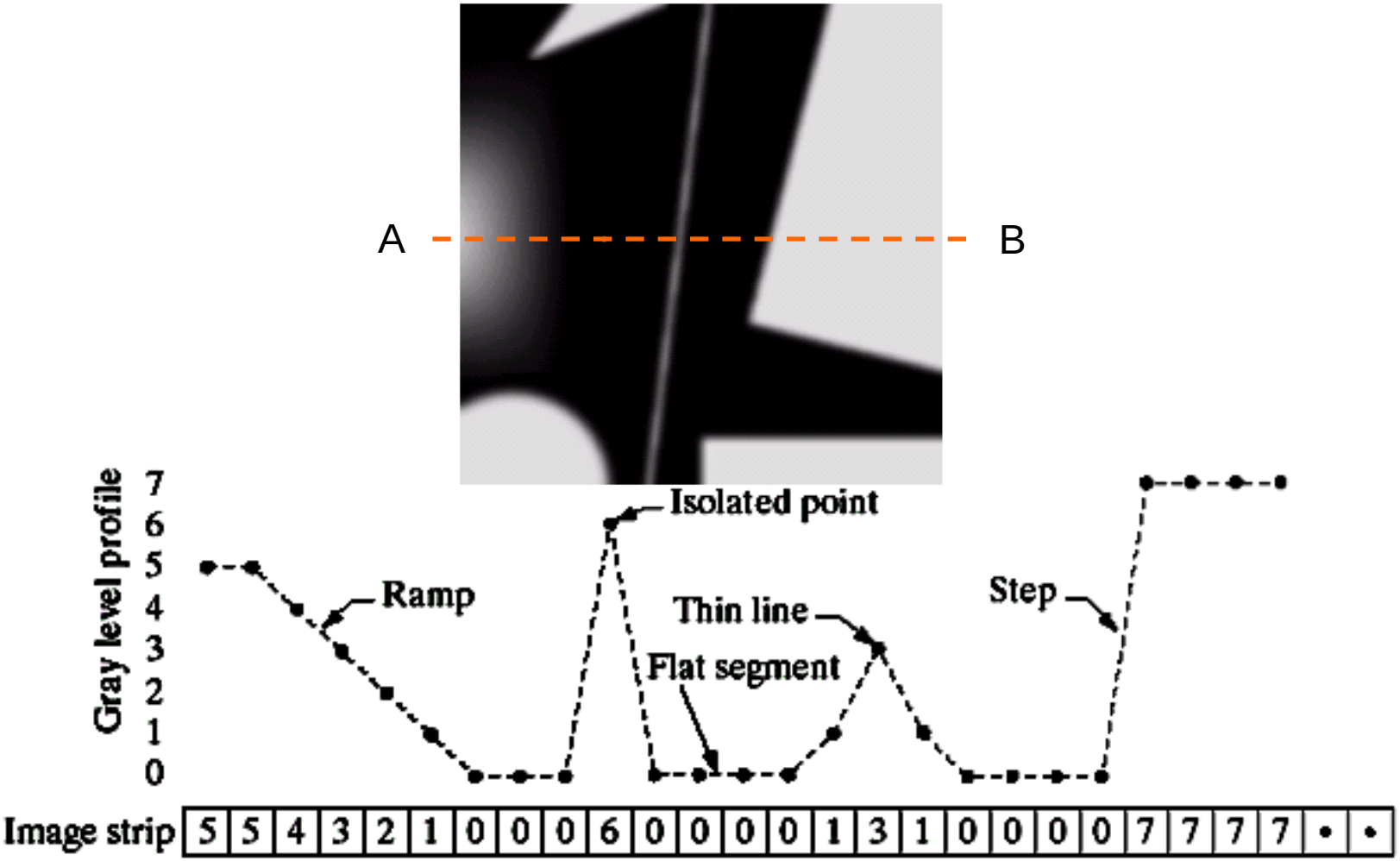
Sharpening filters are based on *spatial differentiation*

Spatial Differentiation

- Let's consider a simple 1 dimensional example



Spatial Differentiation



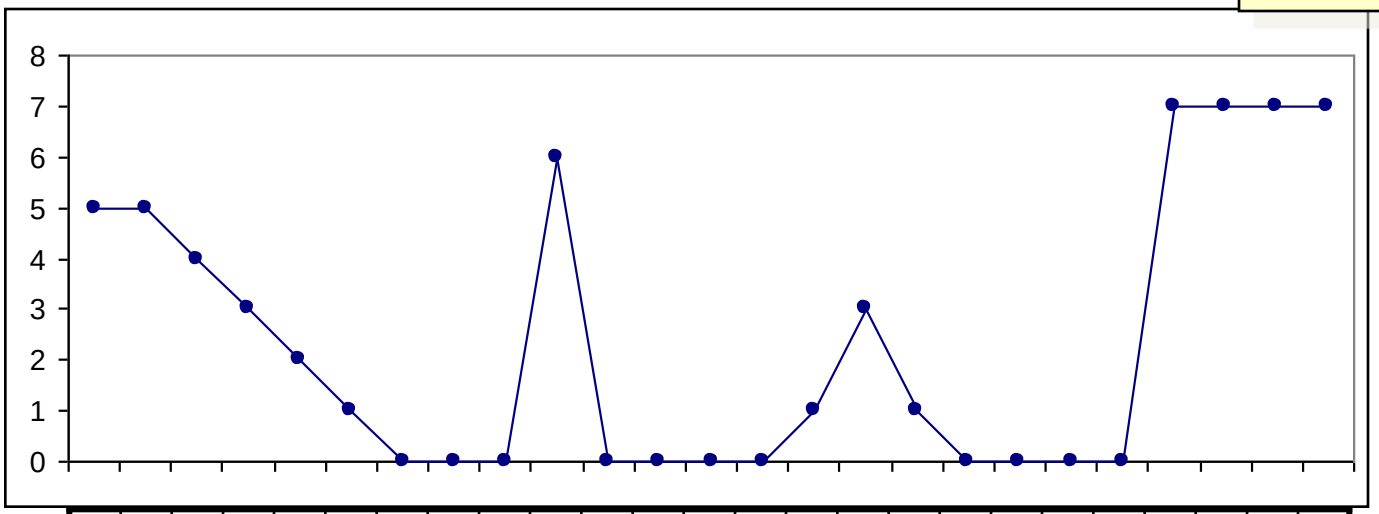
1st Derivative

The 1st derivative of a function is given by:

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

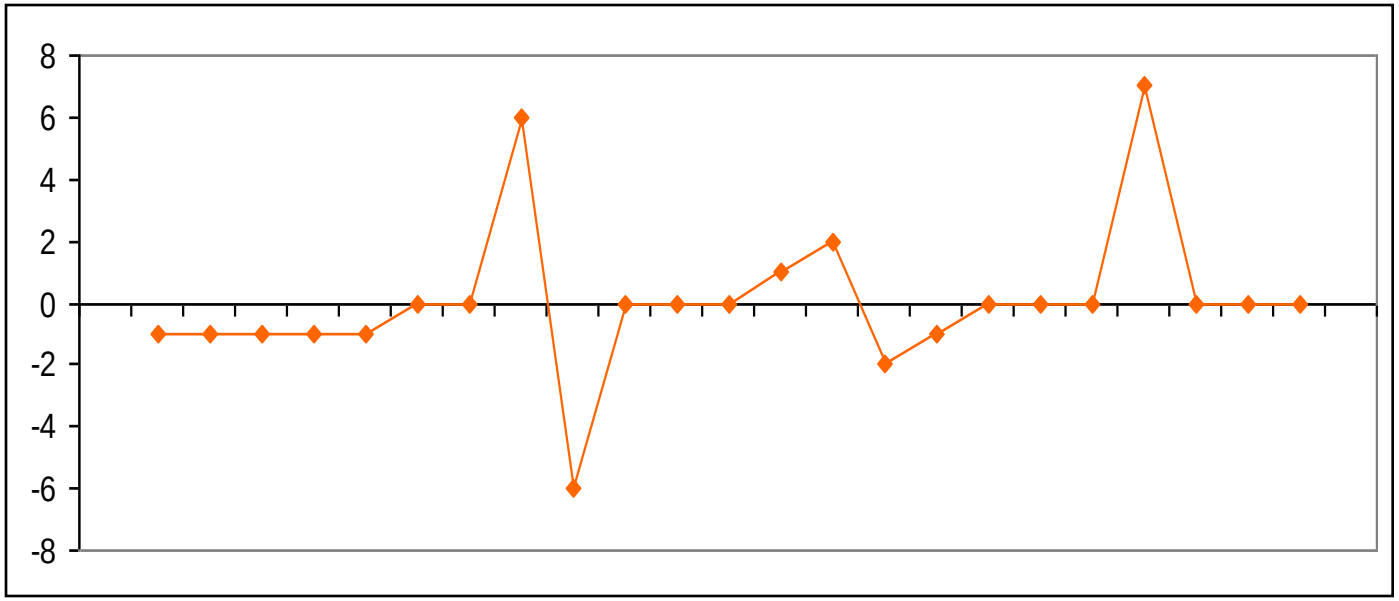
Its just the difference between subsequent values and measures the rate of change of the function

1st Derivative



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

	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



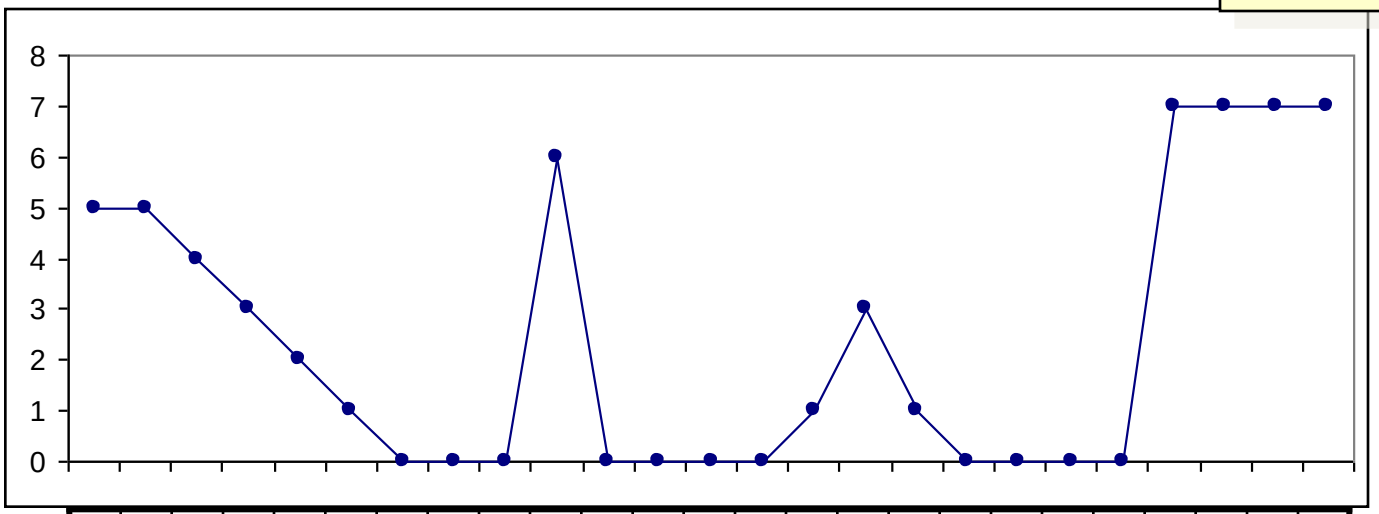
2nd Derivative

The 2nd derivative of a function is given by:

Simply takes into account the values both before and after the current value

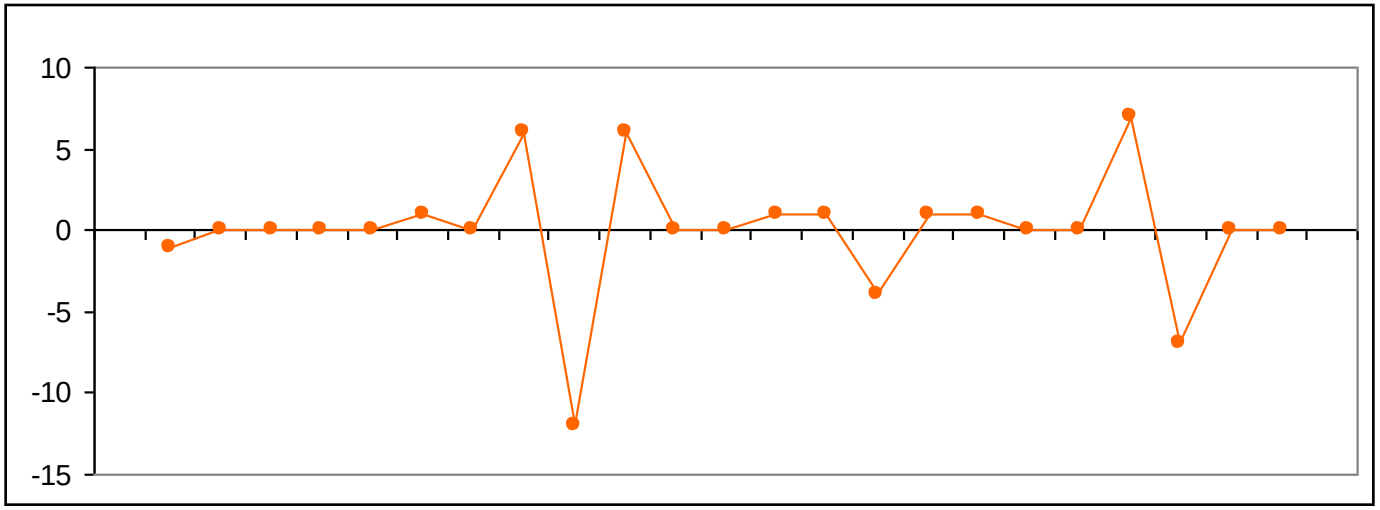
$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

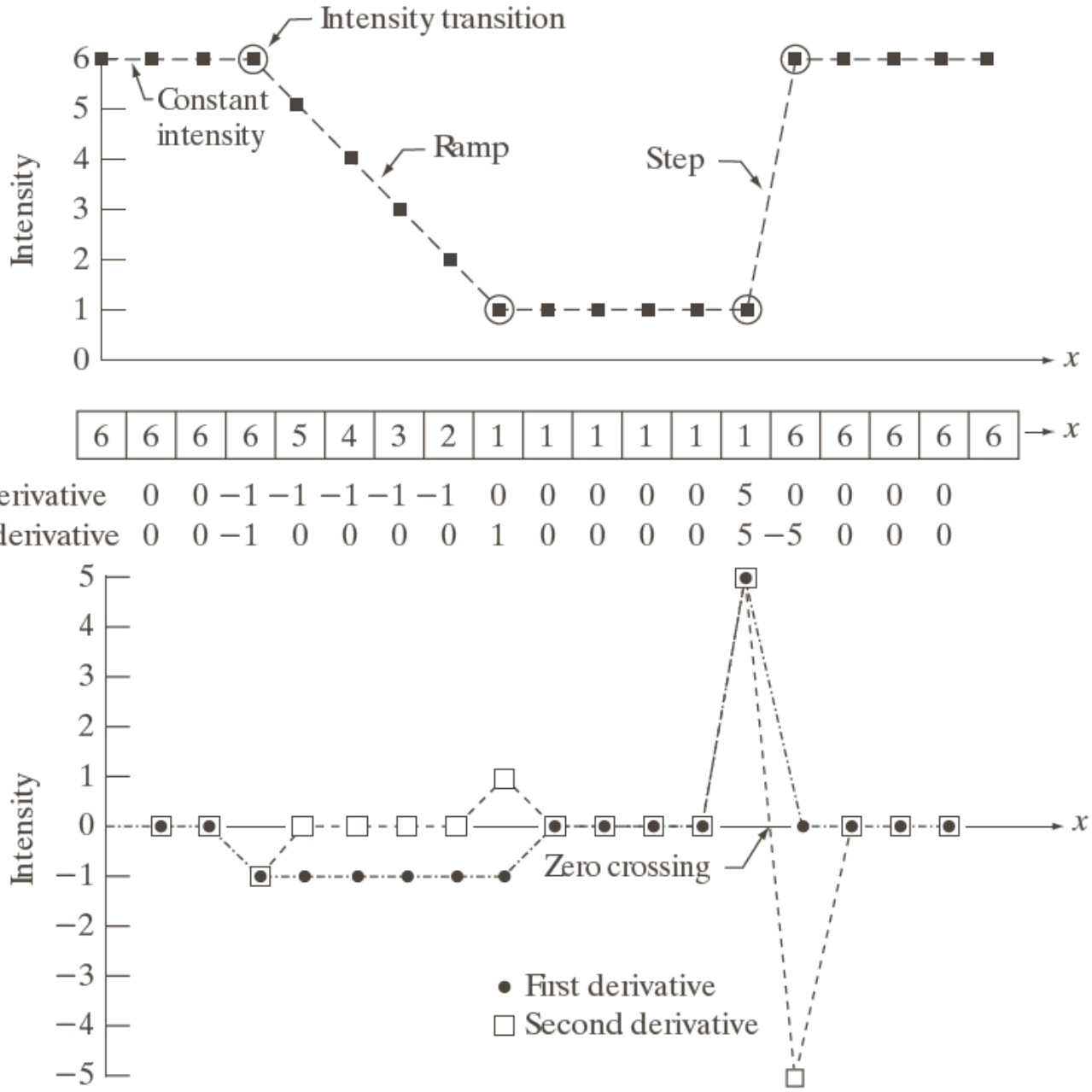
2nd Derivative



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

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	




Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

2nd Derivative for Image Enhancement

The 2nd derivative is more useful for image enhancement than the 1st derivative - S*tronger response to fine detail*

We will come back to the 1st order derivative later on

The first sharpening filter we will look at is the *Laplacian*

Laplacian Filter

The Laplacian is defined as follows:

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

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

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

Laplacian Filter

So, the Laplacian can be given as follows:

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

Can we implement it using a filter/ mask?

0	1	0
1	-4	1
0	1	0

Laplacian Filter

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

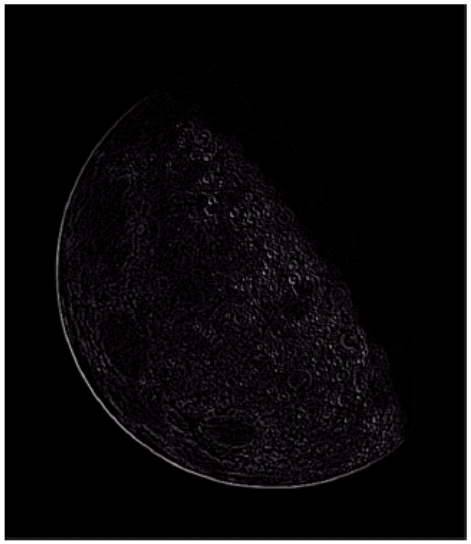
a b
c d

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.

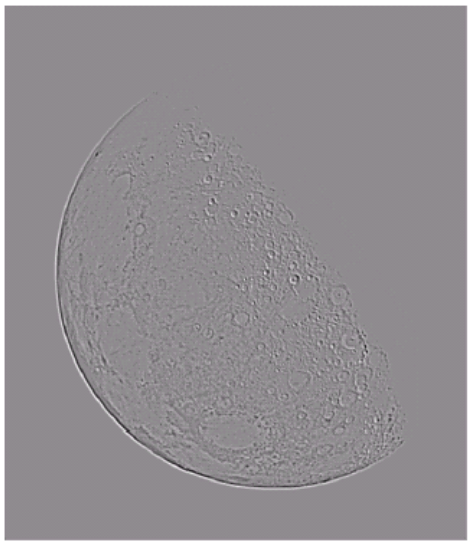
Laplacian Filter

Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities

Original Image



Laplacian Filtered Image



Laplacian Filtered Image Scaled for Display

Laplacian Image Enhancement

The result of a Laplacian filtering is not an enhanced image

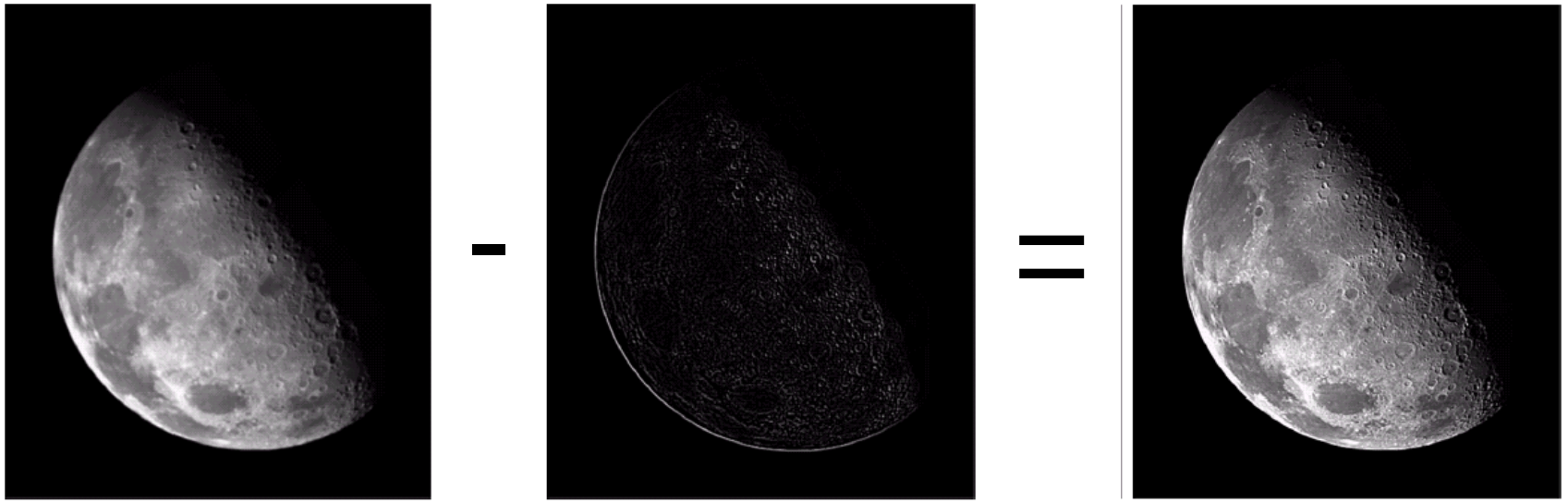
To generate the final enhanced image

Laplacian
Filtered Image
Scaled for Display

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f, w_5 < 0 \\ f(x,y) + \nabla^2 f, w_5 > 0 \end{cases}$$

Laplacian Image Enhancement



Original Image

Laplacian Filtered Image

Sharpened Image

In the final sharpened image edges and fine detail are much more obvious

Laplacian Image Enhancement

Simplified Image Enhancement

- The entire enhancement can be combined into a single filtering operation

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

$$= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$$

Simplified Image Enhancement

- The entire enhancement can be combined into a single filtering operation

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

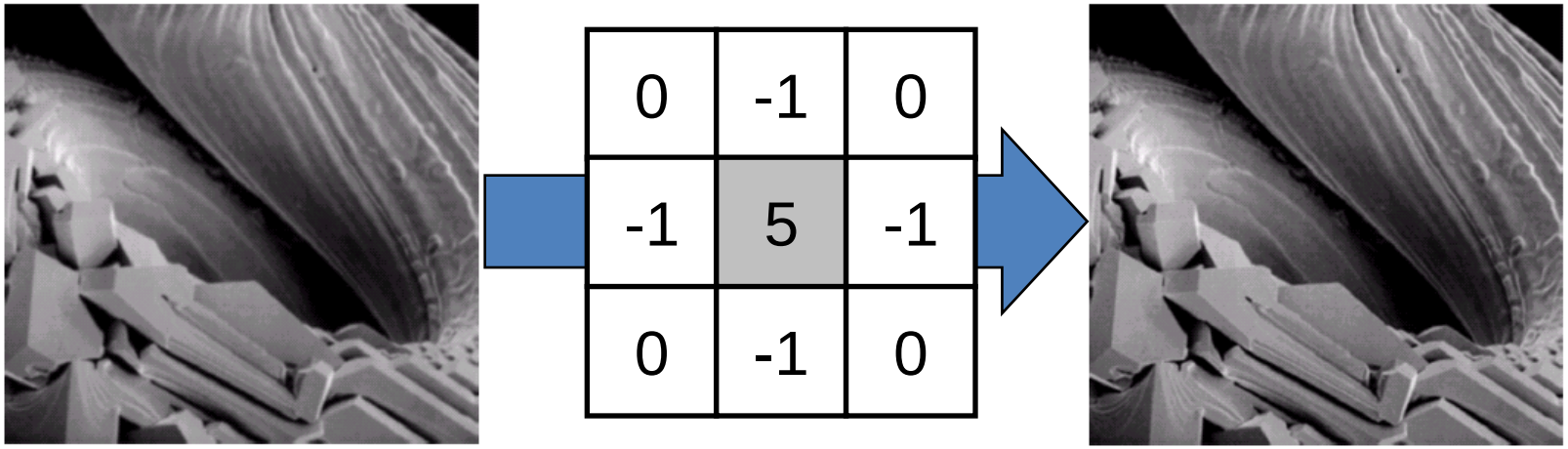
$$= 5f(x,y) - f(x+1,y) - f(x-1,y)$$

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

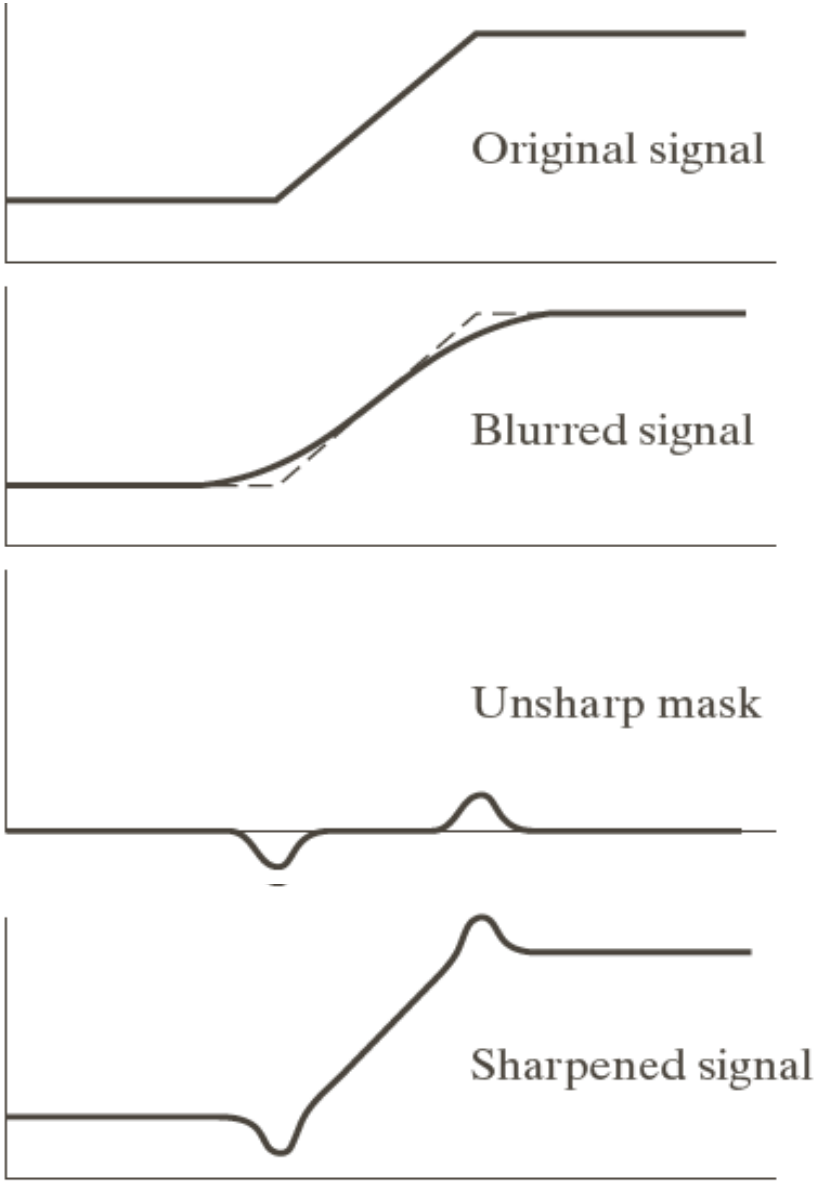
0	-1	0
-1	5	-1
0	-1	0

Simplified Image Enhancement

- This gives us a new filter which does the whole job for us in one step



Unsharp Masking



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.

Use of first derivatives for image enhancement: The Gradient

- The **gradient** of a function f(x,y) is defined as

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

Gradient Operators

Sobel Operator

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

Extract horizontal edges

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

Extract vertical edges

Emphasize more the current point (y direction)

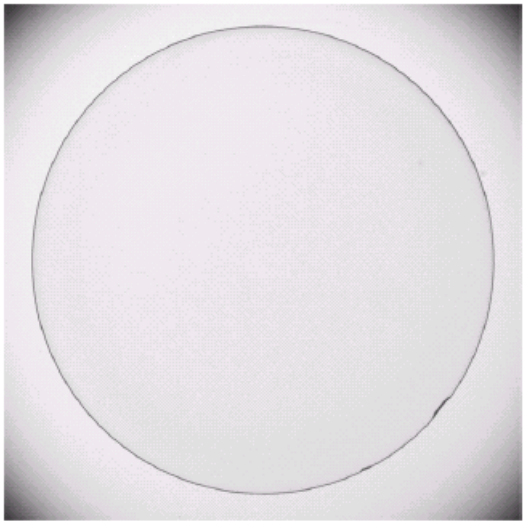
$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

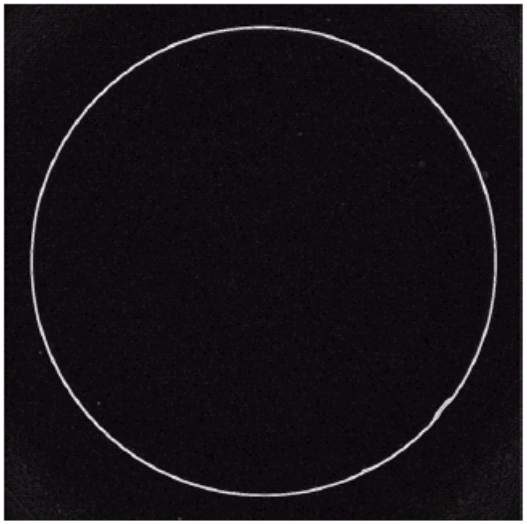
Emphasize more the current point (x direction)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel Arrangement

Sobel Operator: Example





An image of a contact lens which is enhanced in order to make defects more obvious

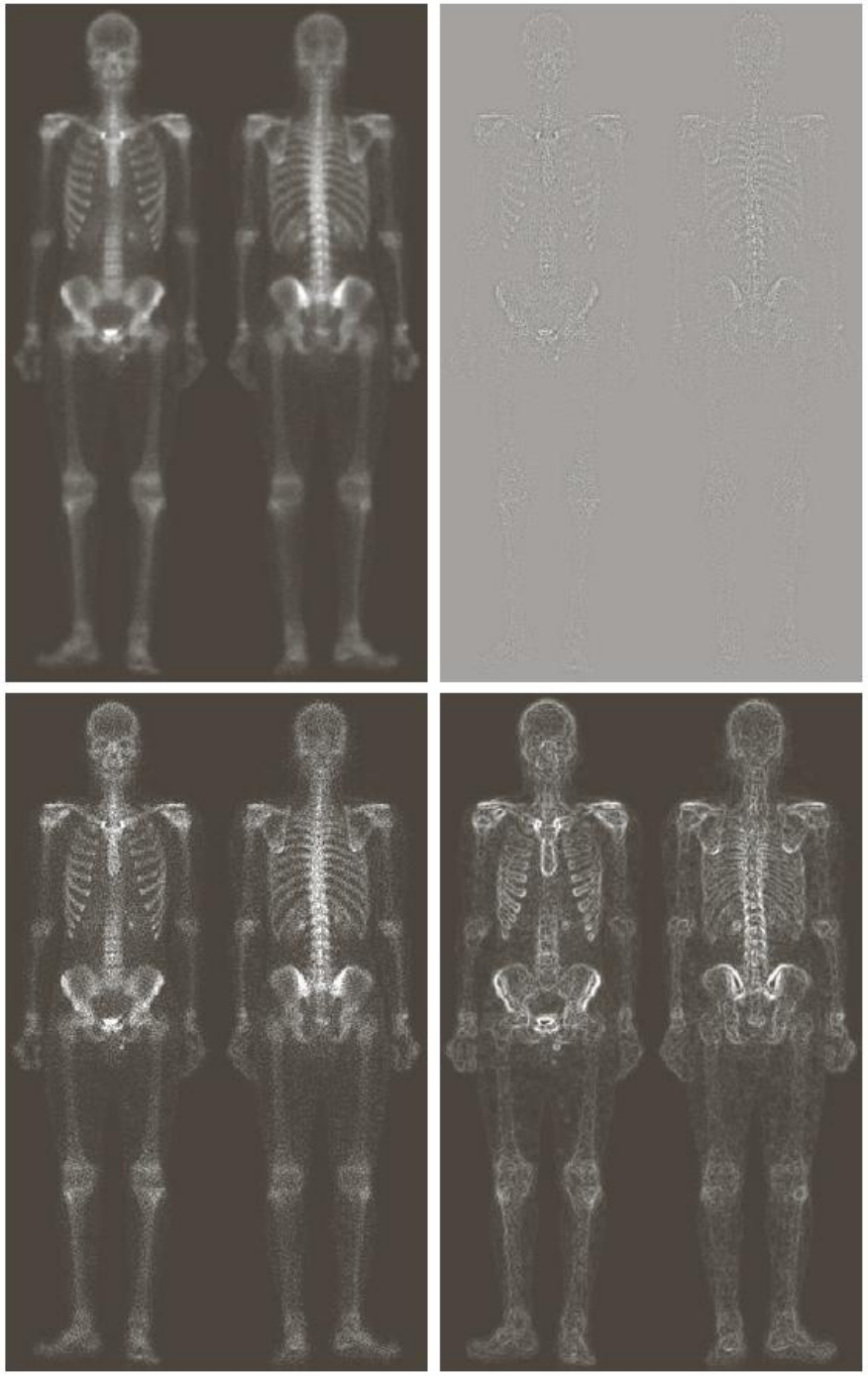
Sobel filters are typically used for edge detection

Combining Spatial Enhancement Methods

Successful image enhancement is typically not achieved using a single operation

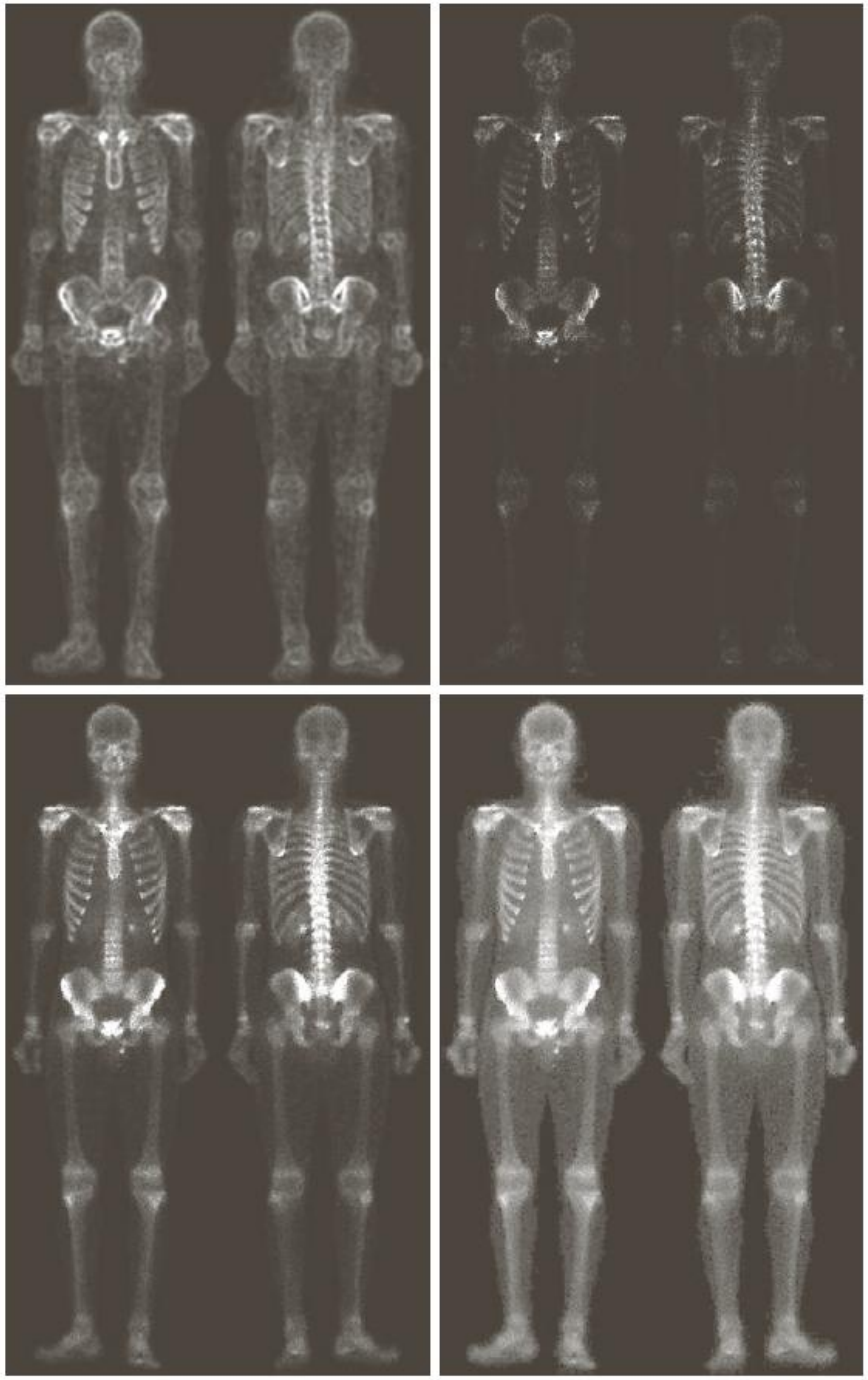
Rather we combine a range of techniques in order to achieve a final result

This example will focus on enhancing the bone scan



a b
c d

FIGURE 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of (a).



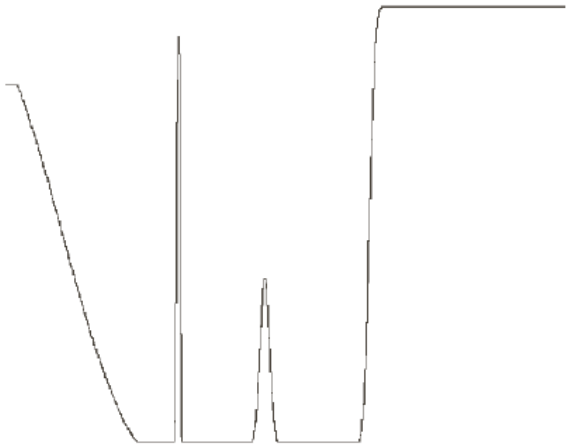
e f
g h

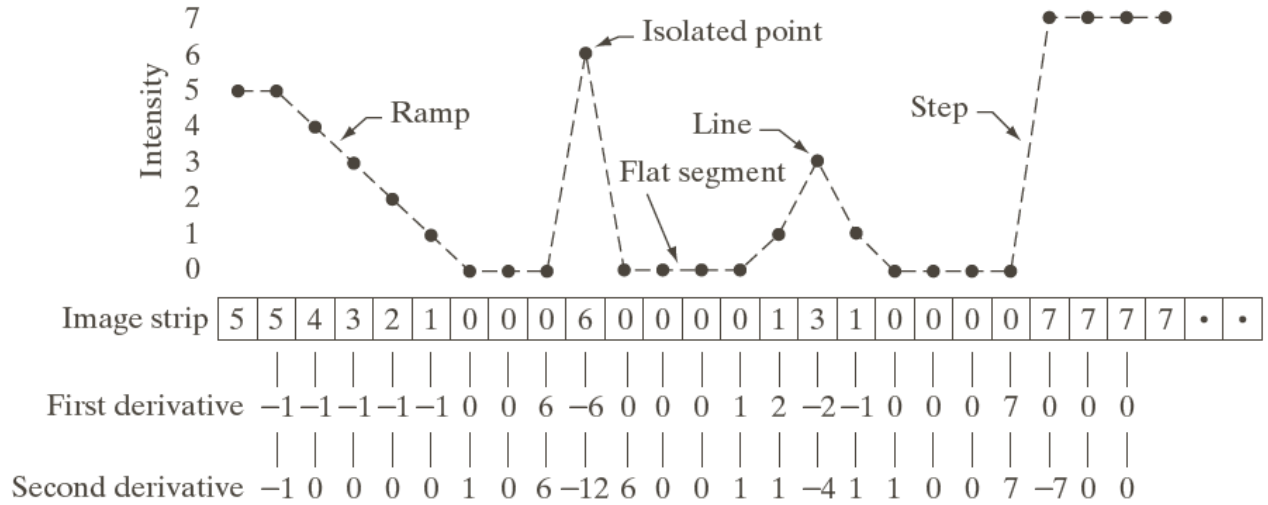
FIGURE 3.43
(Continued)
(e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e).
(g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Compare the original and final images

Edge Detection

- Edge detectors can be based on the first and second derivatives which can detect abrupt intensity changes.
- Derivates of digital functions are defined in terms of differences
- See approximations on using first and second derivatives in Gonzalez section 10.2.1

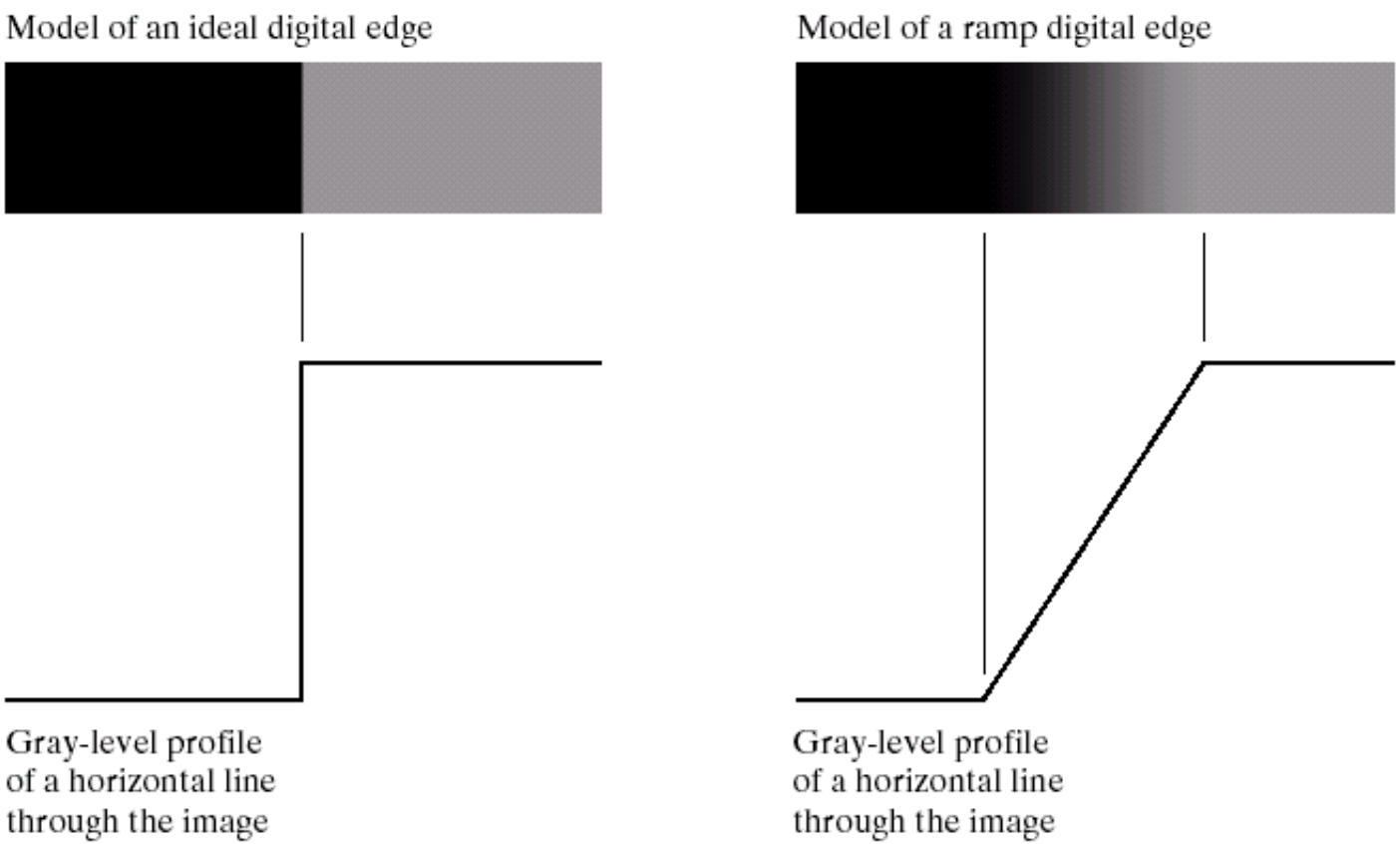



Intensity
7
6
5
4
3
2
1
0
Ramp
Isolated point
Line
Flat segment
Step
Image strip
5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •
First derivative
−1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second derivative
−1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

Edge Detection

- First order derivatives produce thick edges while second order derivatives produce finer ones. See ramp edge on the figure 10.2 on previous slide
- Second order derivatives enhance sharp changes and fine details more aggressively than first order derivatives see the isolated point and the line in the same figure. This can be a problem if the noise is present in the image
- Second derivative changes its sign as it transitions into and out of a ramp or step edge. See the step edge. This “double edge” effect can be used to locate edges.
- The sign of the second derivative is also used to determine whether an edge is a transition from light to dark (-ve value) or from dark to light (+ve value). See step edge

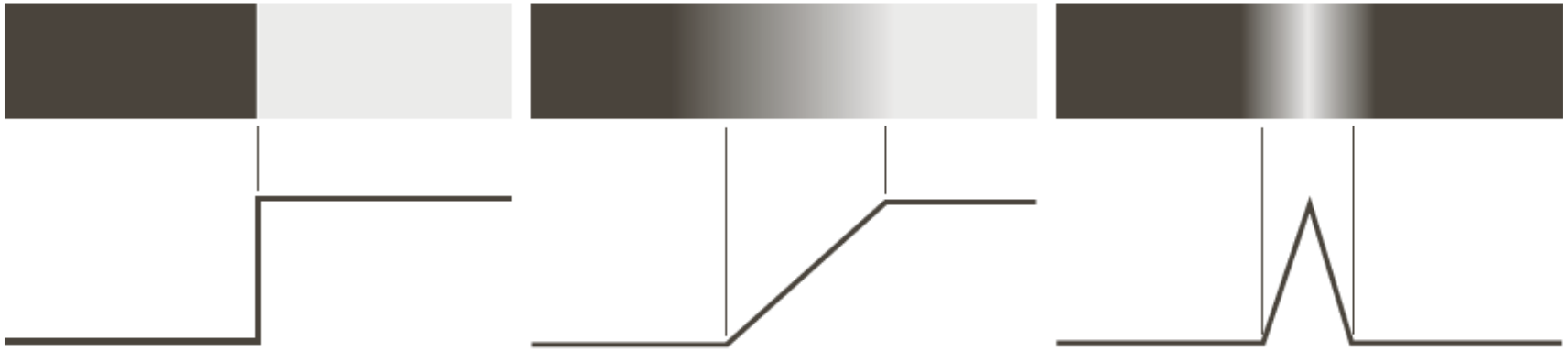
Edge Detection



a b

FIGURE 10.5
(a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

Edges



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.

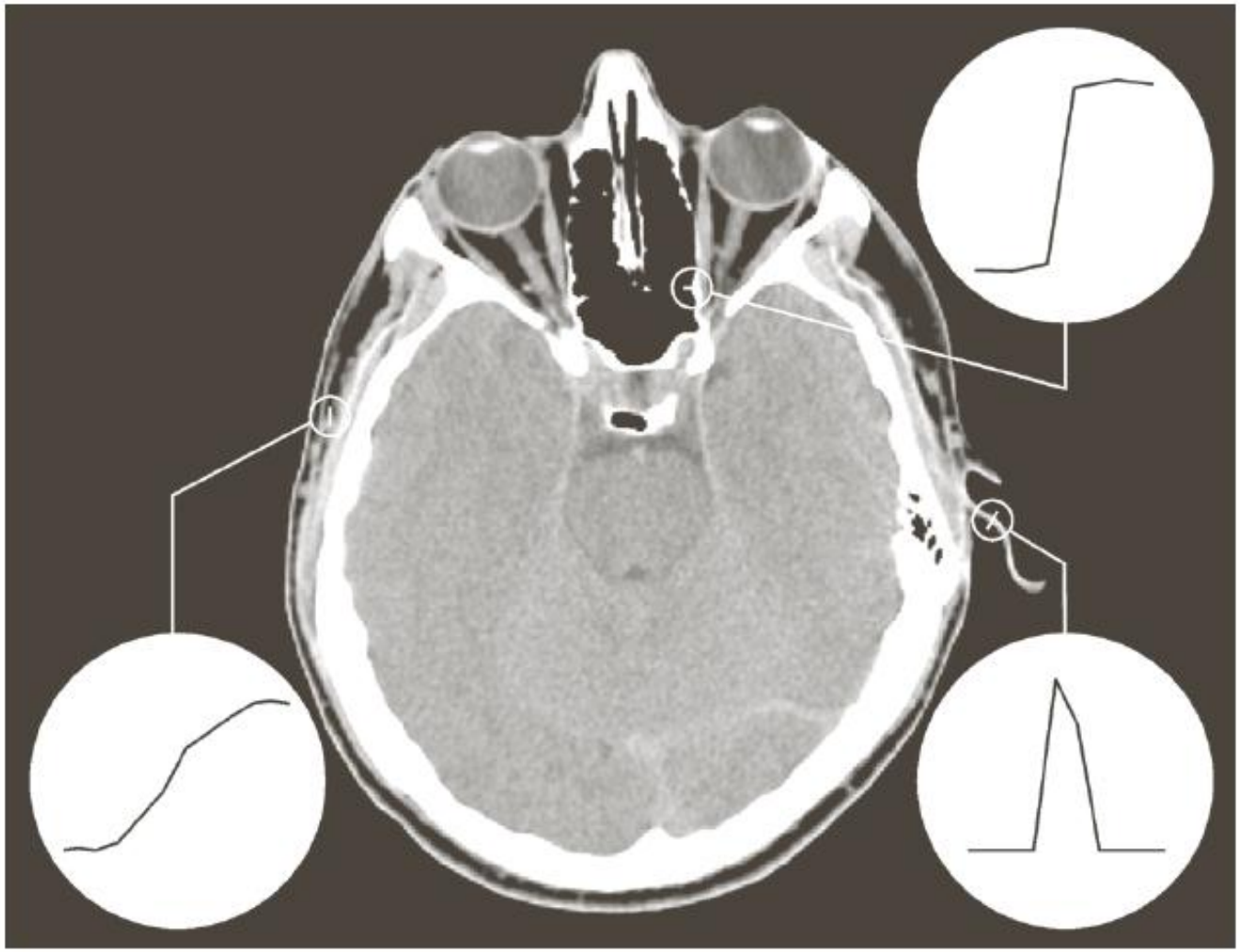
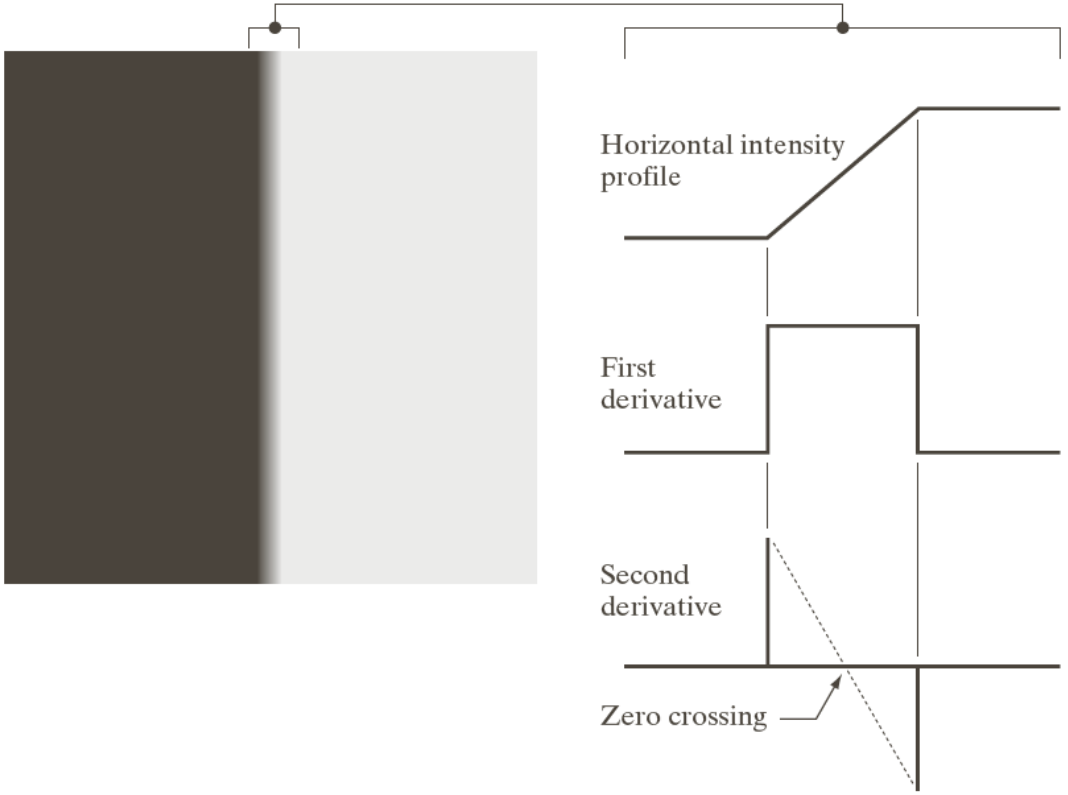


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Edge Detection

- First and second derivative for smooth noiseless edge
- The zero crossings of the second derivative can be used for locating edges in an image.

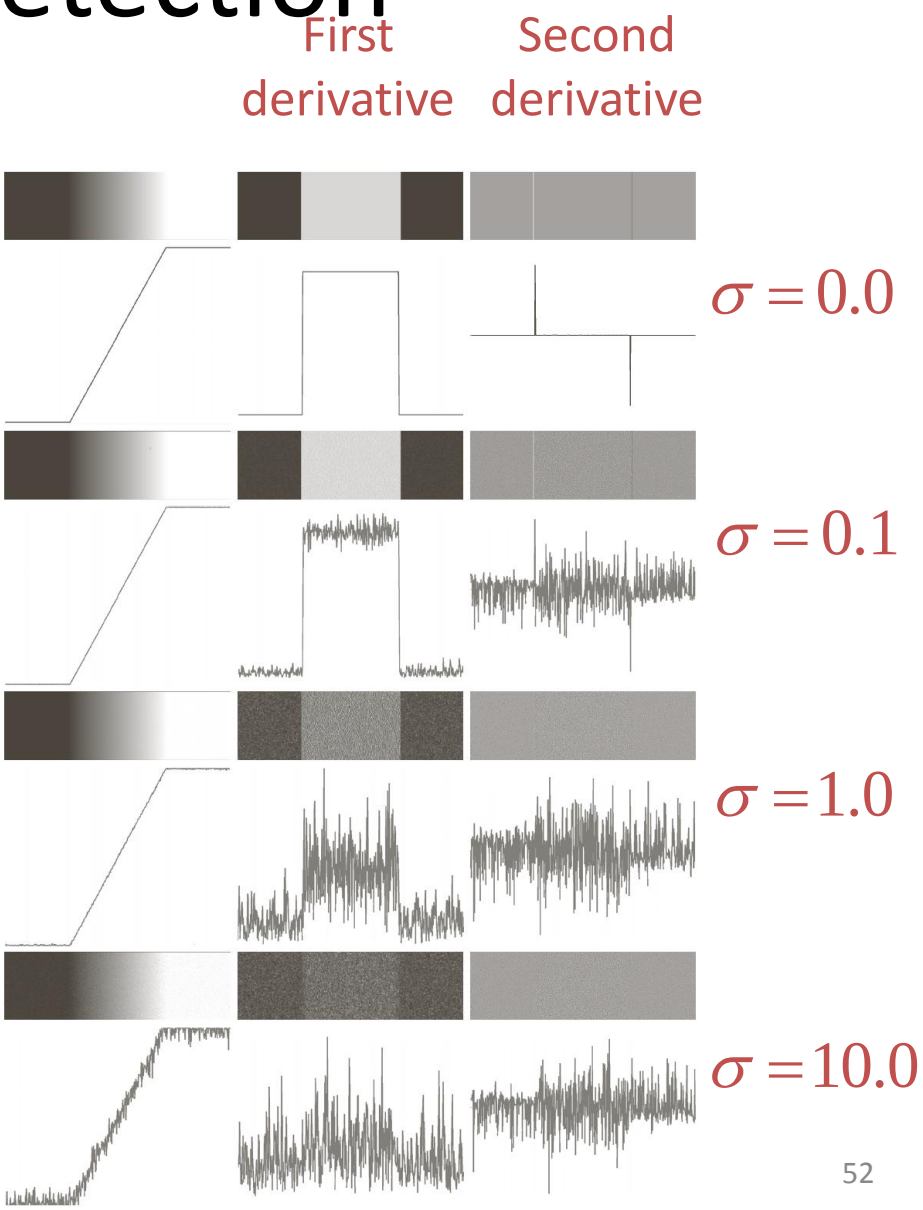


a b

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.

Edge Detection

Results of first and second derivative for edges with Gaussian noise of mean = 0.



•Smoothing step is a must before taking derivative for edge detection

Gradient Operators

- Most common differentiation operator is the gradient vector.

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

Magnitude:

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

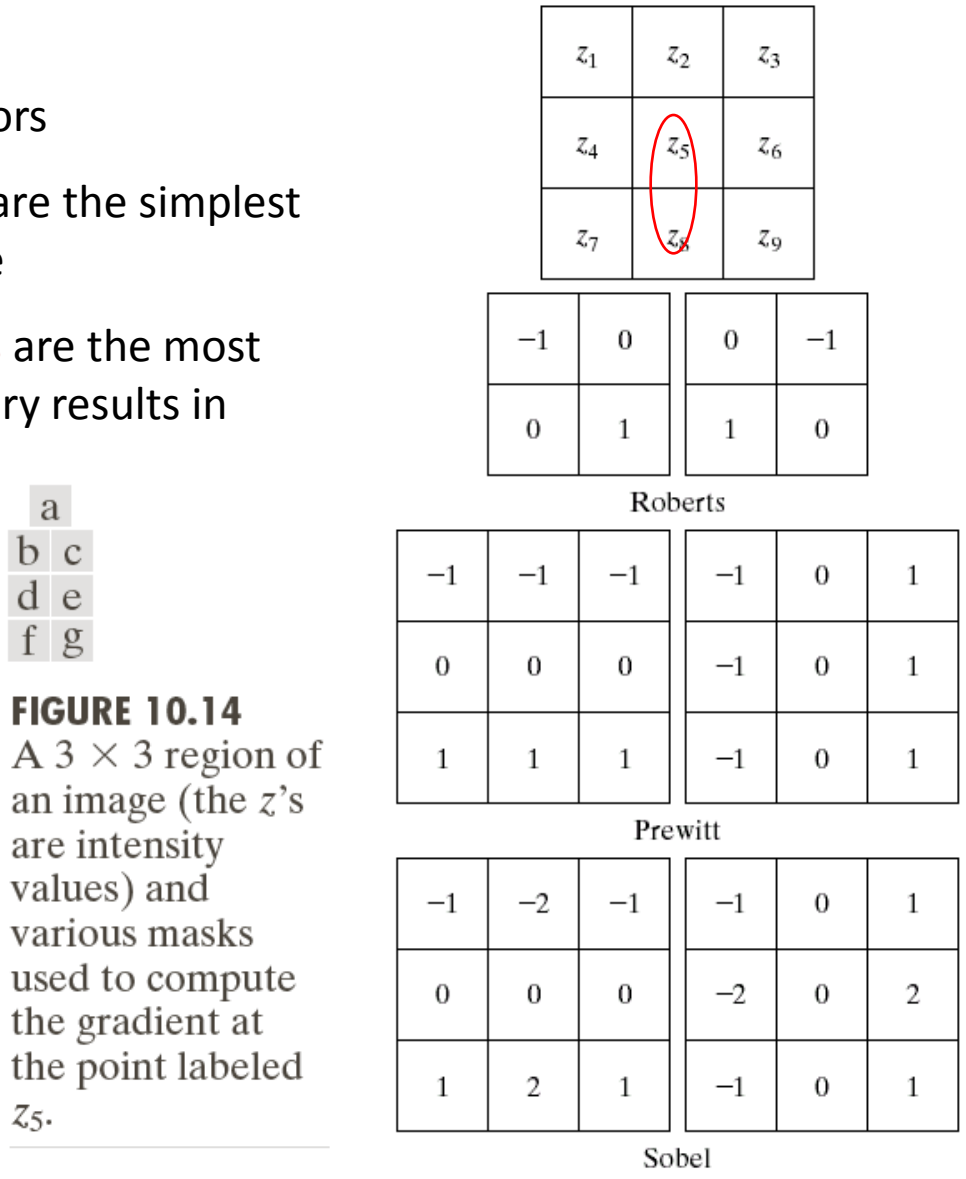
Direction:

$$\angle f(x,y) = \tan^{-1}\left[\frac{G_y}{G_x} \right]$$

Gradient Operators

Some common gradient operators

- Roberts and Prewitt masks are the simplest but not robust against noise
- Sobel edge detection masks are the most common and give satisfactory results in presence of noise



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.

Gradient Operators

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.

Direction of an Edge

•The direction of an edge at a point is orthogonal to the direction of the gradient vector at the point

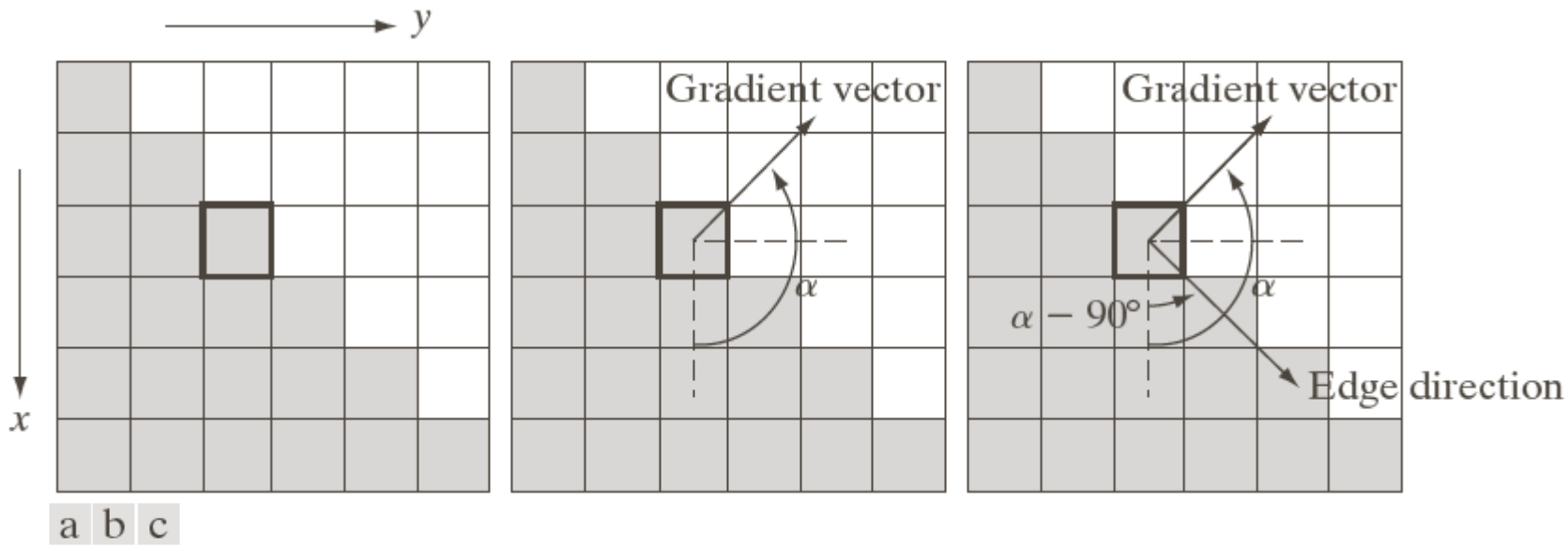


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

a b
c d

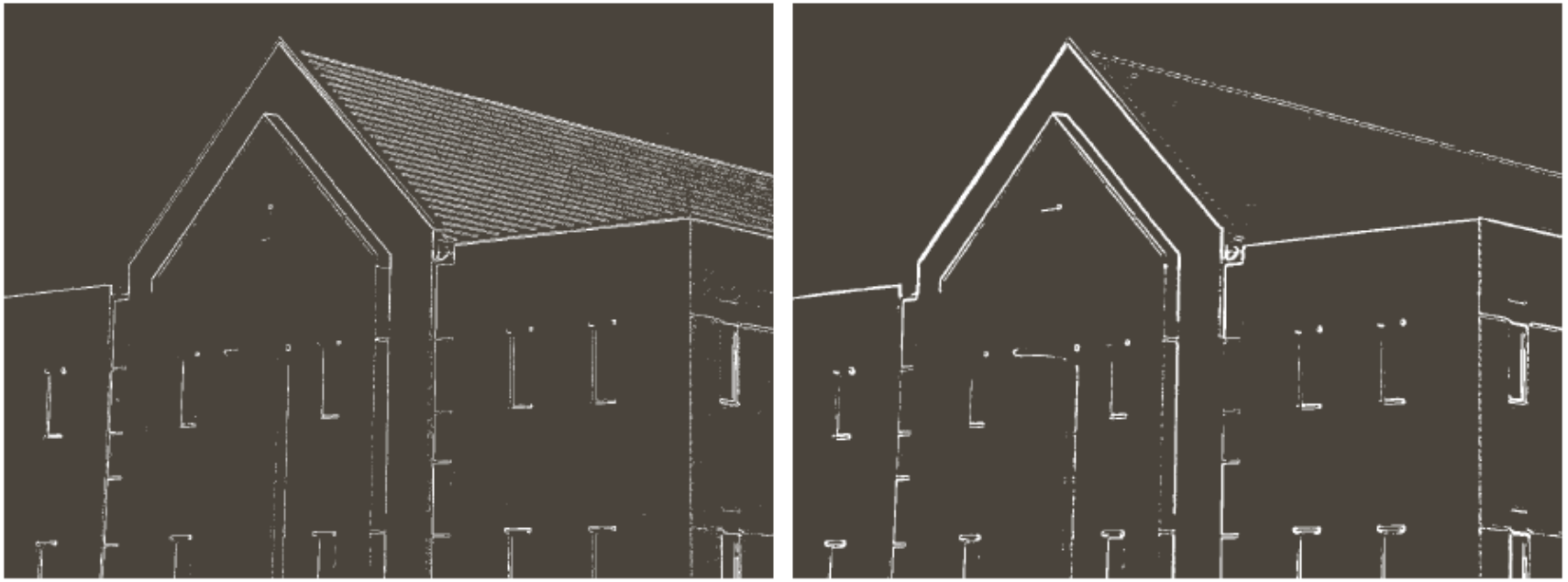
FIGURE 10.16
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The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

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 10.19
Diagonal edge detection.
(a) Result of using the mask in Fig. 10.15(c).
(b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).



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.

Laplacian of a Gaussian (LoG)

- A filter which combines the smoothing function (Gaussian) with the Laplacian is called Laplacian of a Gaussian (LoG) filter.
- Robust against noise.
- Consider a smoothing Gaussian function:

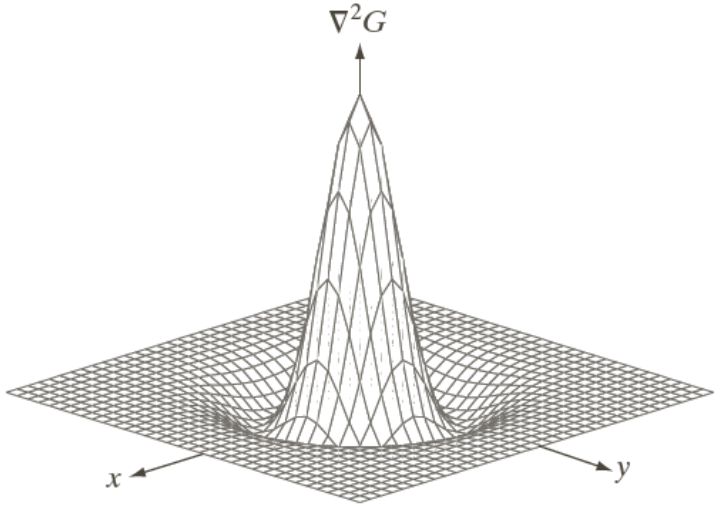
$$G(r) = e^{-\frac{r^2}{2\sigma^2}}$$

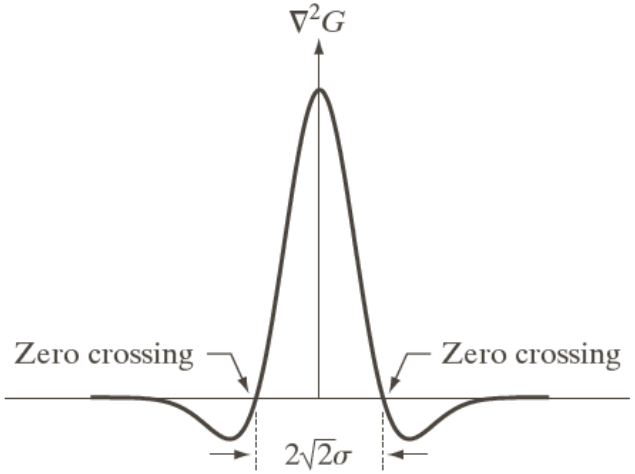
where $r^2 = x^2 + y^2$, σ : standard deviation

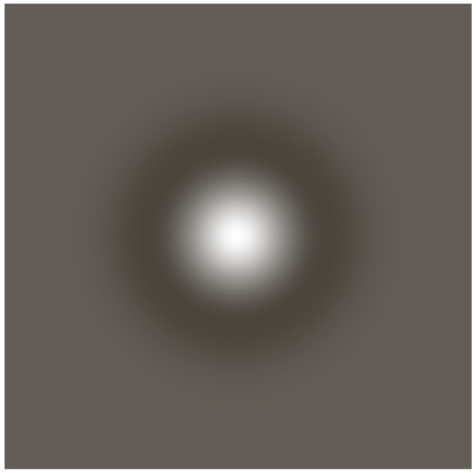
- The Laplacian of this function gives the LoG function:

$$\nabla^2 G(r) = \frac{\partial^2 G(r)}{\partial r^2} = \left[\frac{r^2 - 2\sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

Laplacian of a Gaussian (LoG) Filter







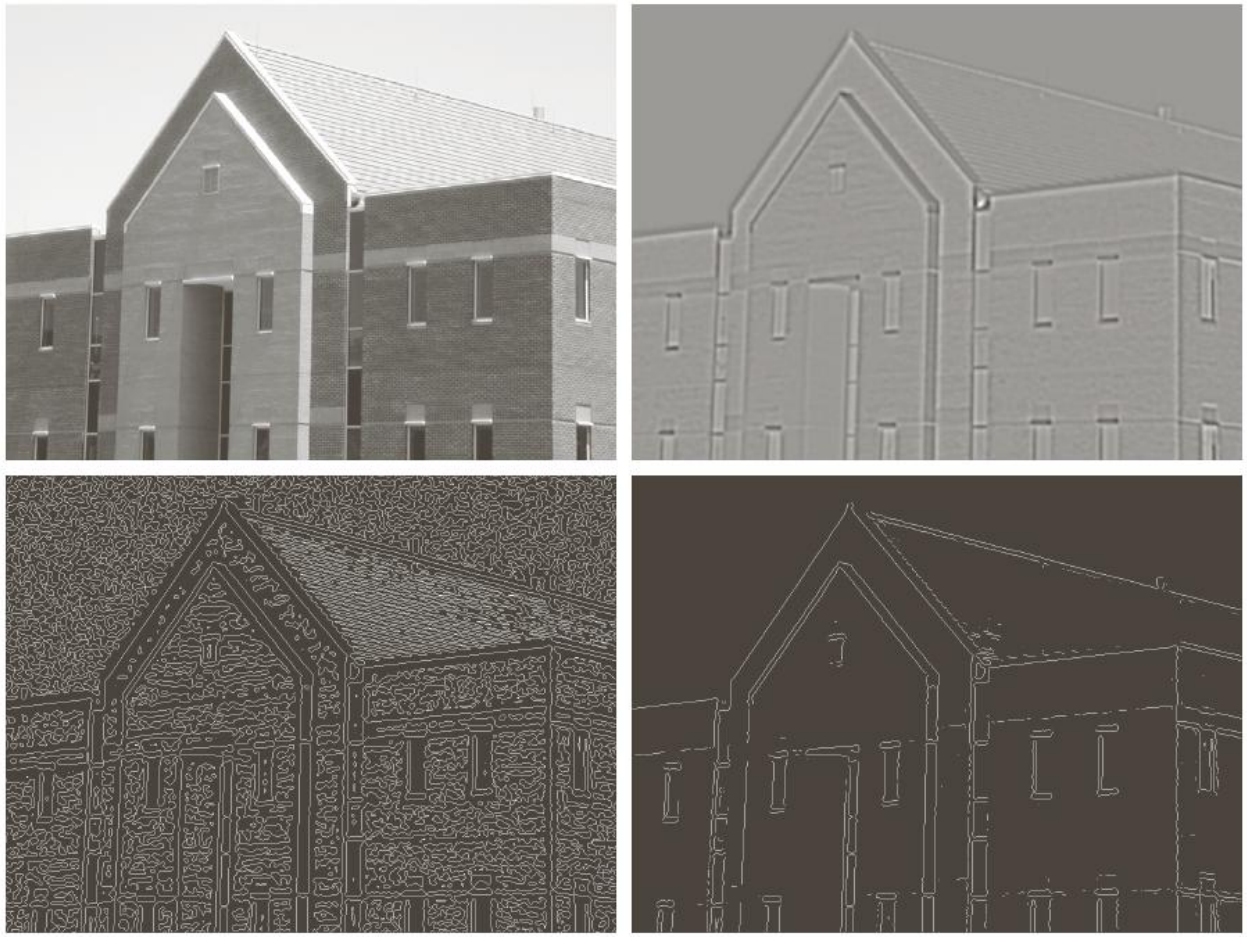
a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Edge detection by LoG

- Due the shape of this function it is also called Mexican hat function (or Mexican hat filters).
- Better performance against noise. Reduces the intensity of structures or noise, which are at scales much smaller than sigma.
- Choose size of Gaussian mask to be n >= 6*sigma
- Then use a 3x3 Laplacian
- Find the zero crossings



a b
c d

FIGURE 10.22
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Readings from Book (3rd Edn.)

- Sharpening Filters (Chapter – 3)
- Edge Detection (Chapter – 10)

Reading Assignment

- High Boost Filtering (Chap – 3)
- Laplacian of Gaussian (LoG) (Chap – 10)

Acknowledgements

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Computer Vision for Computer Graphics, Mark Borg

Material in these slides has been taken from, the following resources