CSC 420 Digital Image Processing

Dr. Maryam Al-Berry

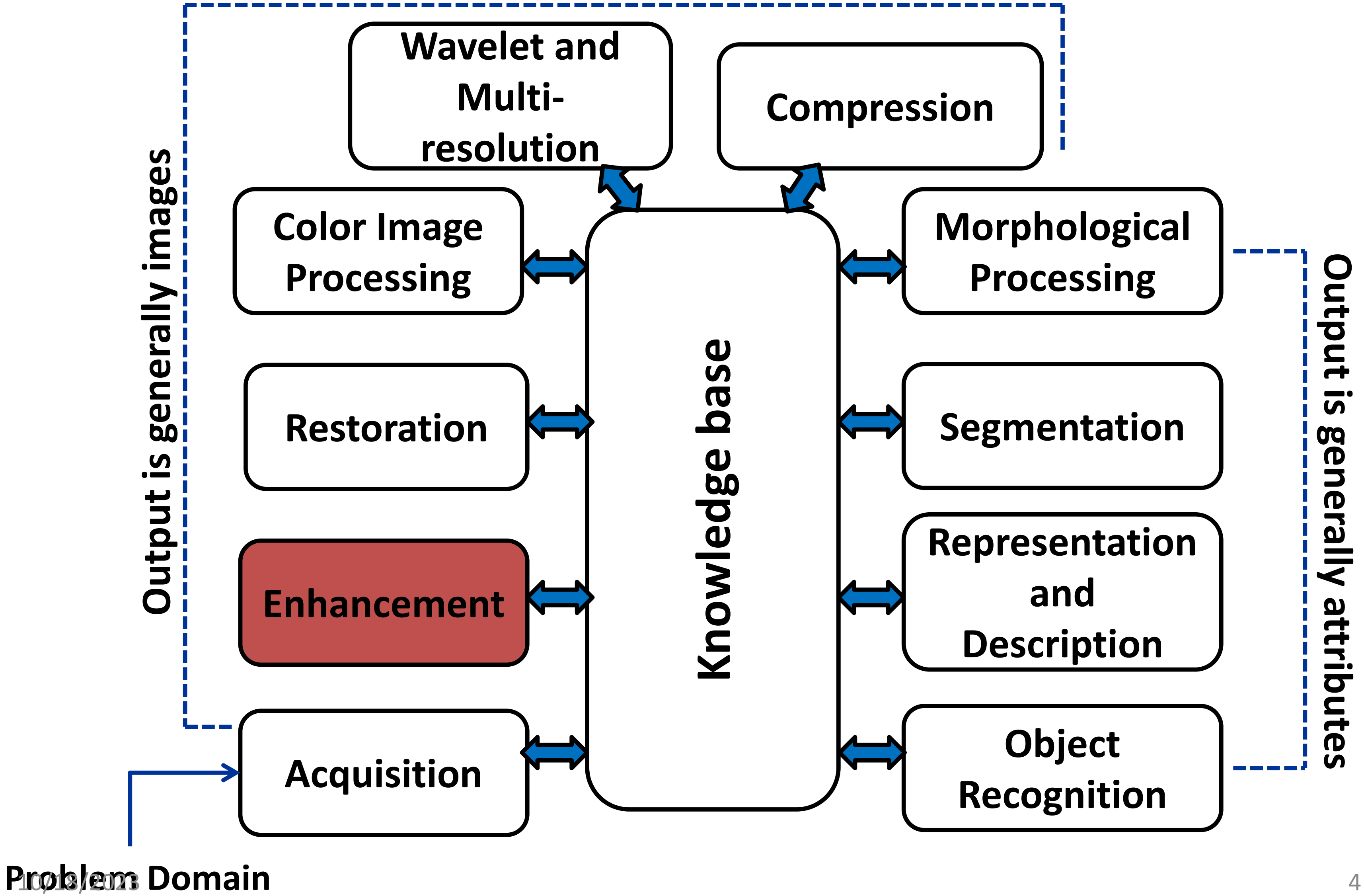
Maryam_nabil@cis.asu.edu.eg

Recall

1. What is image enhancement
2. Enhancement vs. restoration
3. Brightness vs. contrast
4. Image Histogram
5. Intensity transformation functions

Image Enhancement II

Fundamental Steps of DIP



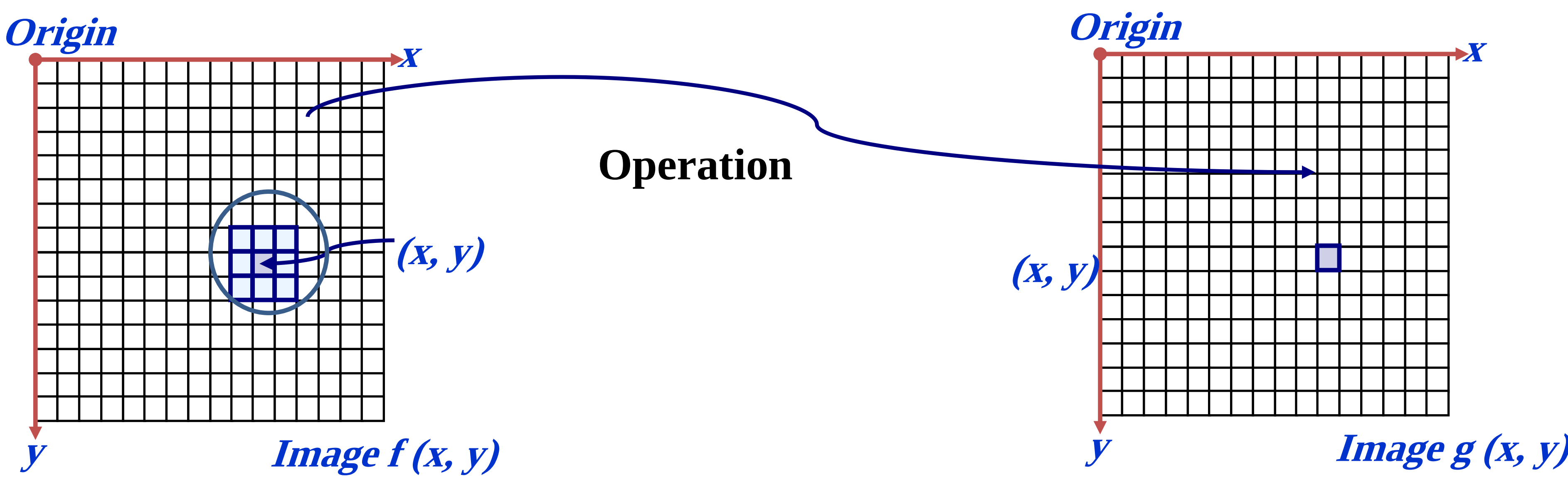
3. Spatial Filtering (cont.)

Contents

1. Neighborhood Operations
2. What is Spatial Filtering? How?
   Correlation and Convolution
3. Smoothing Filters
   Linear and Nonlinear
4. Sharpening Filters
   First Derivatives
   Second Derivatives

Neighborhood Operations

- For any specific location *(x, y)*, the value of *g* at that location is the result of applying an operation to the pixels in the neighborhood with origin *(x, y)*.



Neighborhood Operations – (cont.)

Some Simple Neighborhood Operations:

- **Min:** Set the pixel value to the minimum in the neighbourhood.

- **Max:** Set the pixel value to the maximum in the neighbourhood.

- **Average:** Set the pixel value to the sum of pixel divided by the size of the neighborhood.

- **Median:** Set the pixel value to the midpoint value in the ORDERED set. Sometimes the median works better than the average.

Neighborhood Operations – (cont.)

Example

Original Image

x

123	127	128	119	115	130
140	145	148	153	167	172
133	154	183	192	194	191
194	199	207	210	198	195
164	170	175	162	173	151

y

Average

Output Image

x

	142				

y

original

3x3 average

Neighborhood Operations – (cont.)

Example

Original Image

x

123	127	128	119	115	130
140	145	148	153	167	172
133	154	183	192	194	191
194	199	207	210	198	195
164	170	175	162	173	151

y

original

Average

Output Image

x

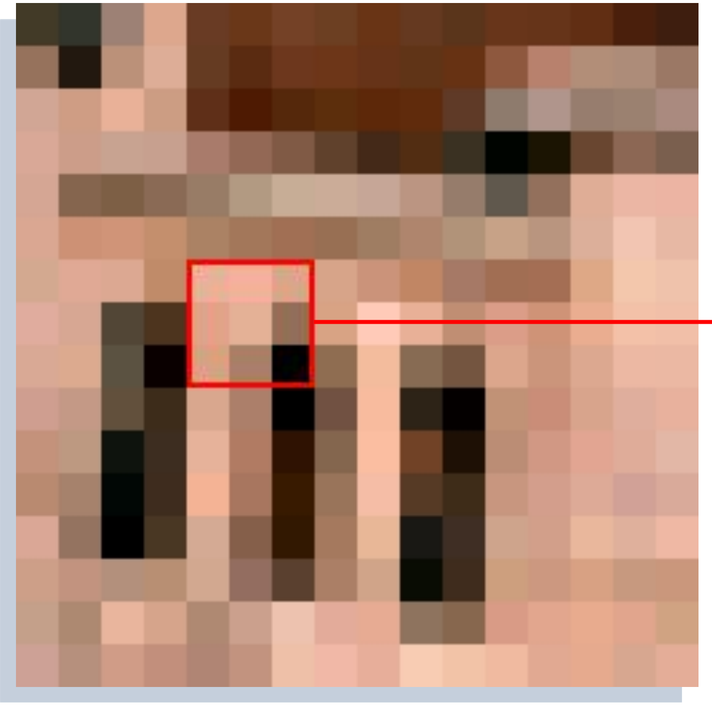
	142	150			

y

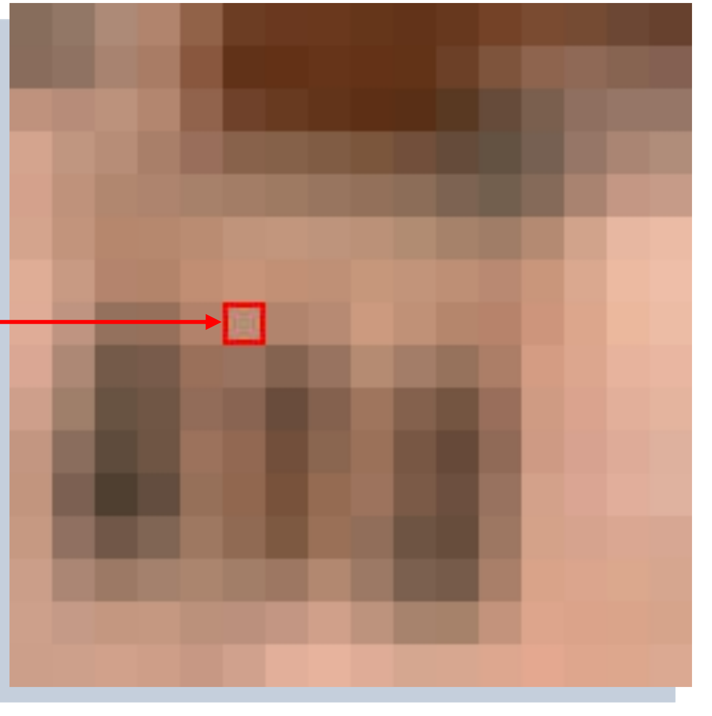
3x3 average

Neighborhood Operations – (cont.)

Example



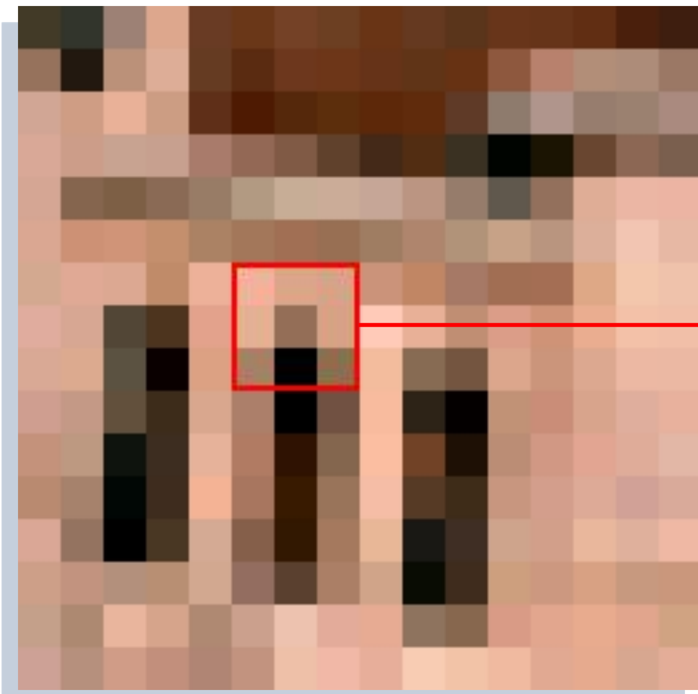
original



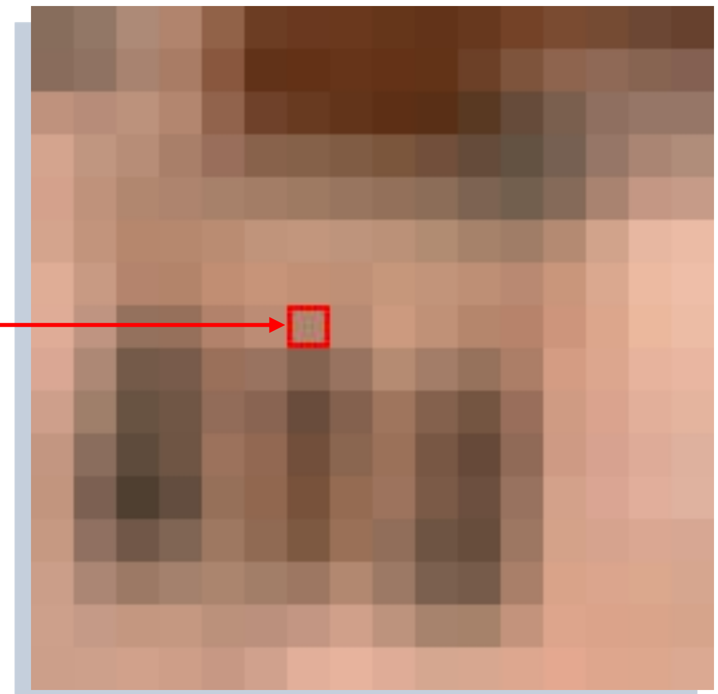
3x3 average

Neighborhood Operations – (cont.)

Example



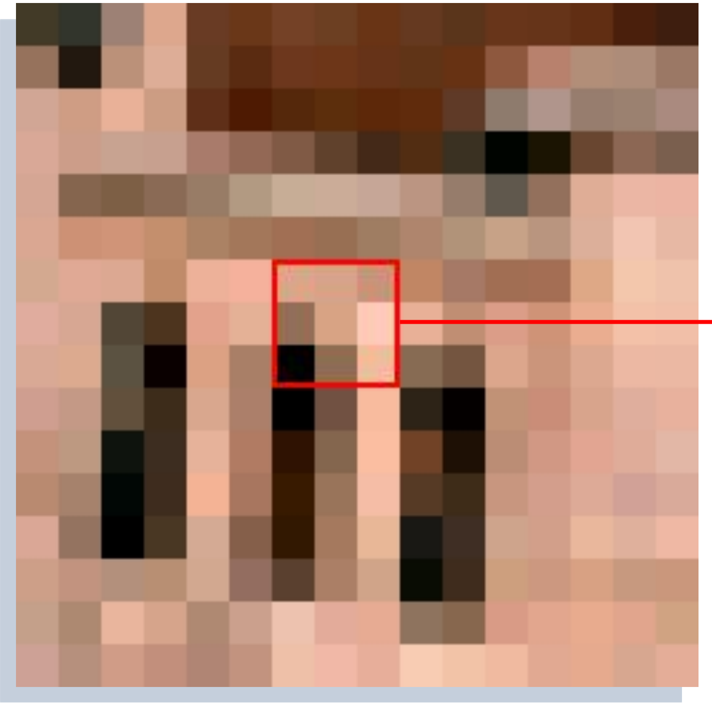
original



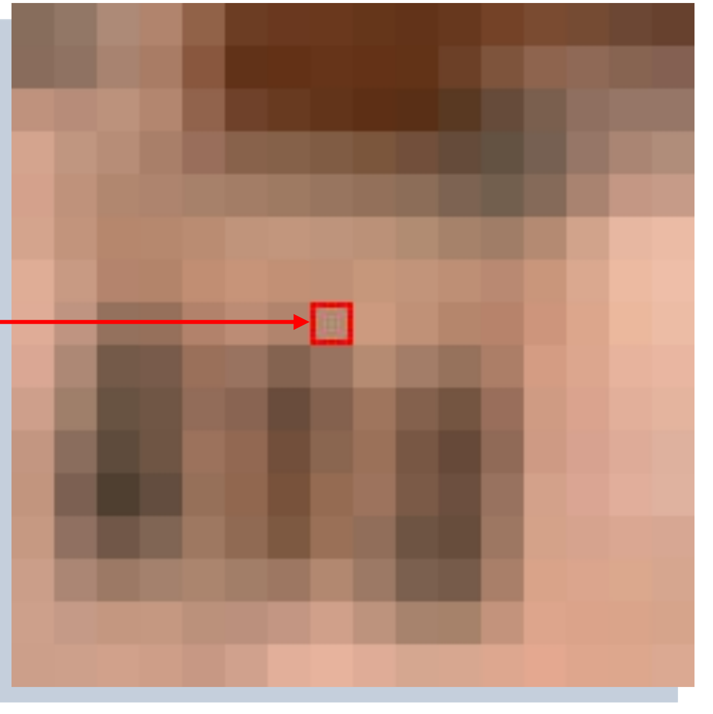
3x3 average

Neighborhood Operations – (cont.)

Example



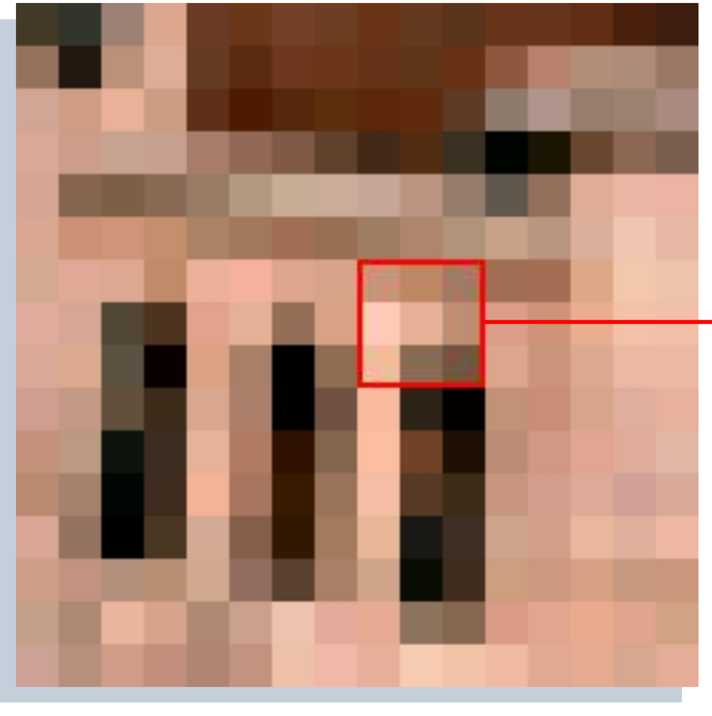
original



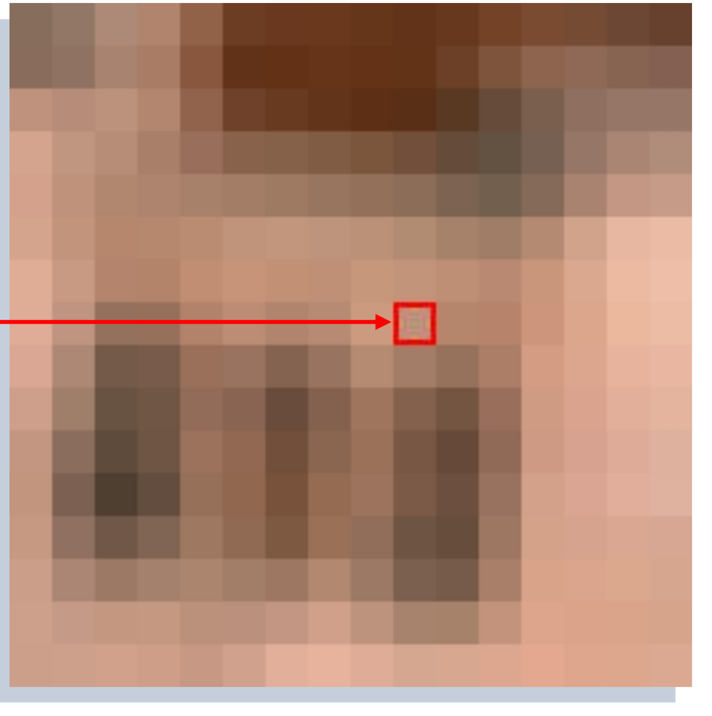
3x3 average

Neighborhood Operations – (cont.)

Example



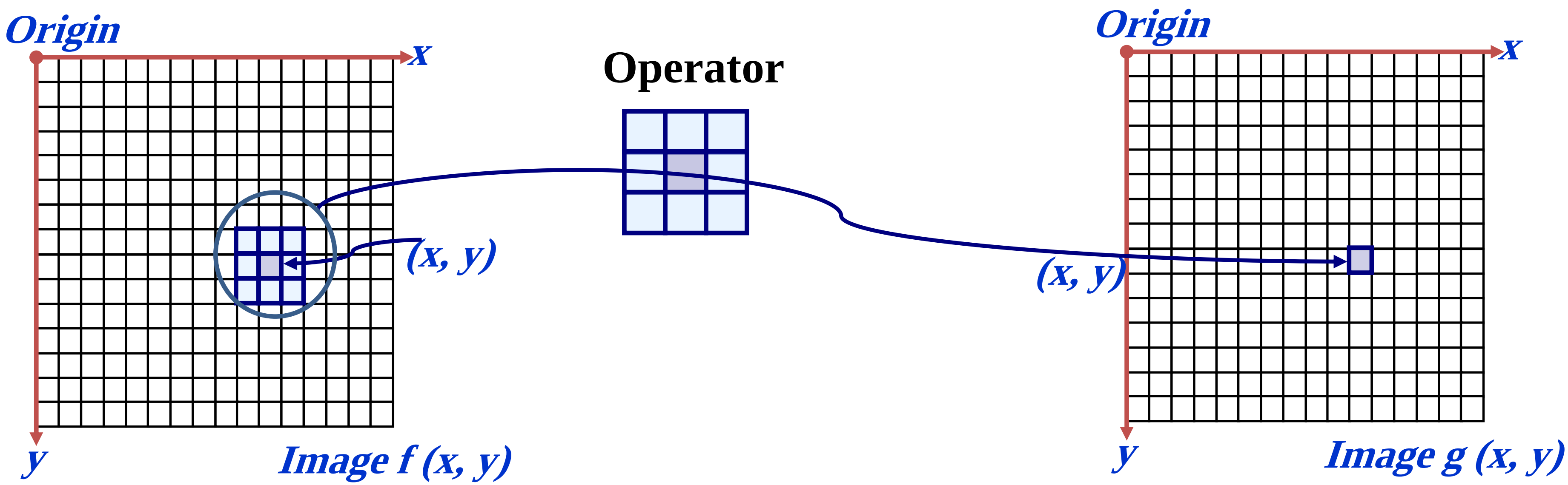
original



3x3 average

What is Spatial Filtering?

- Neighborhood size.
- Operator (filter/mask/weight matrix).



What is Spatial Filtering? – (cont.)

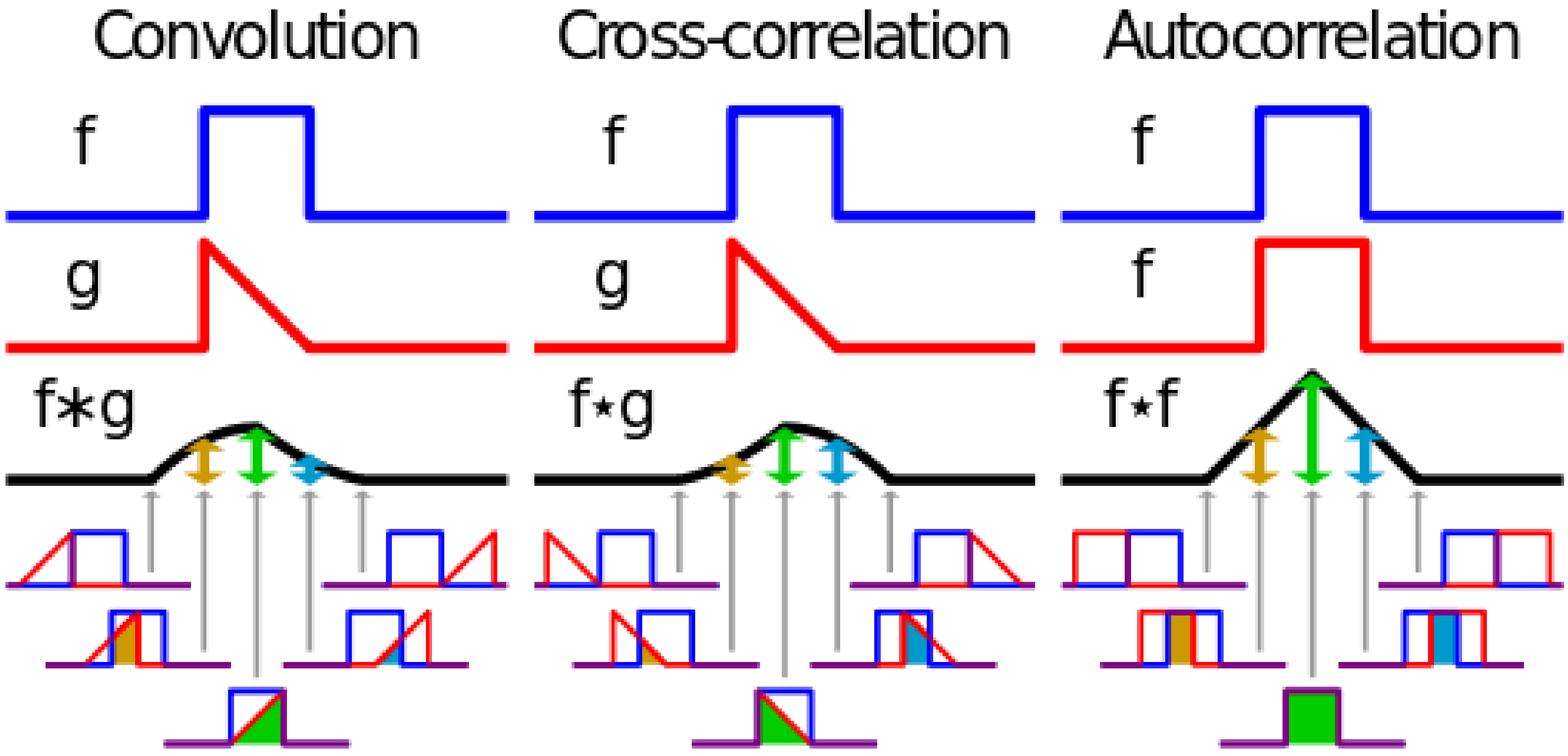
Basic Filtering Processes

- **Convolution is used to linearly filter a signal, for example to smooth a spike train.**

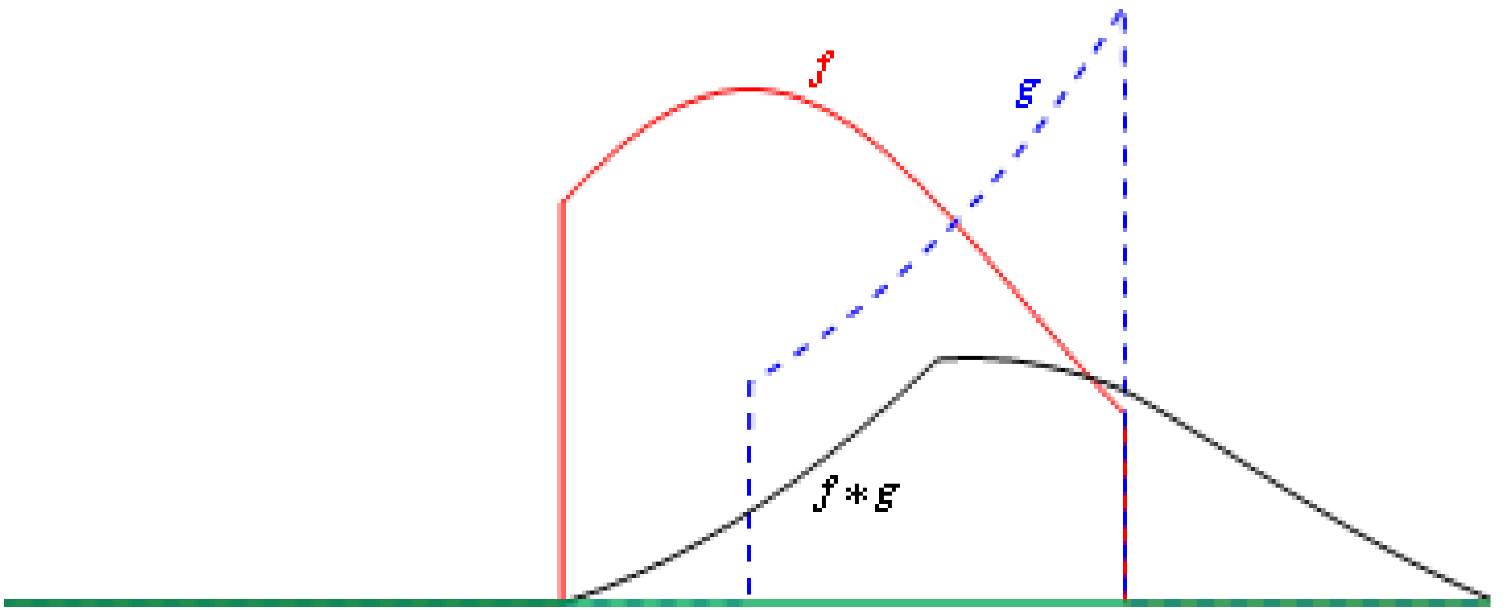
- **Correlation is used to characterize the statistical dependencies between two signals.**

- **<u>Sum of products between two signals.</u>**

What is Spatial Filtering? – (cont.)



What is Spatial Filtering? – (cont.)

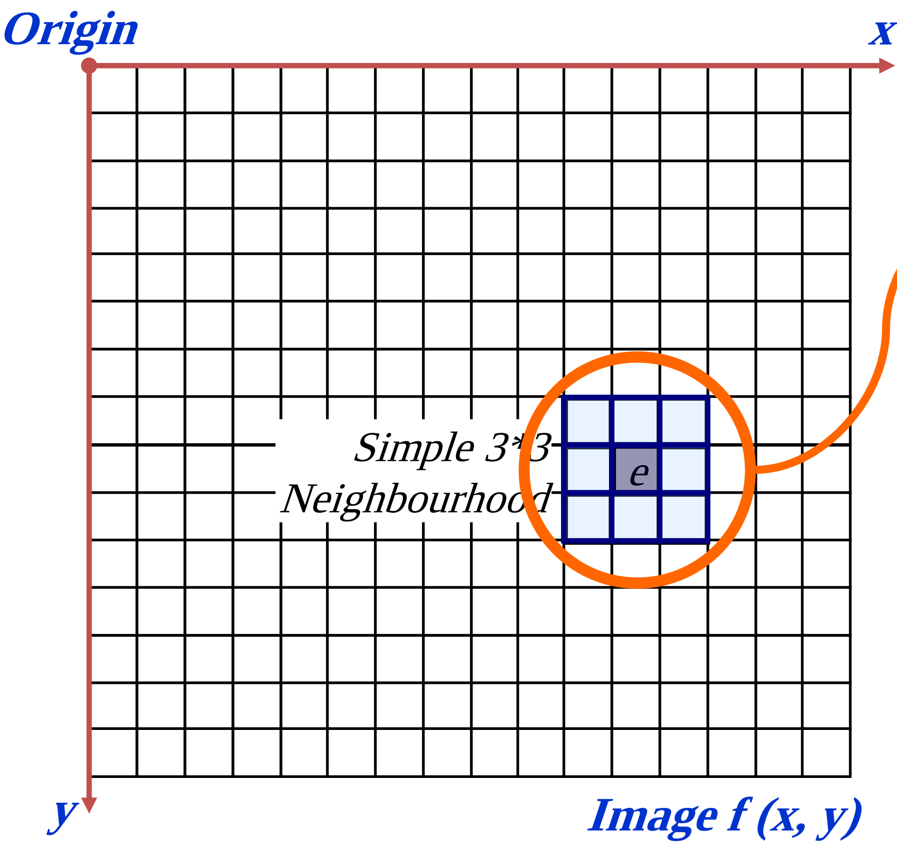


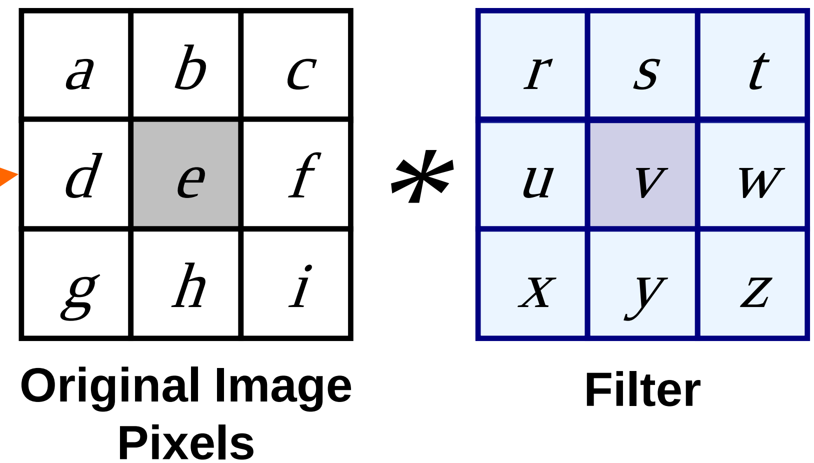
Correlation		Convolution	
(a)	Origin f: 0 0 0 1 0 0 0 0 w: 1 2 3 2 8	Origin f: 0 0 0 1 0 0 0 0 w rotated 180°: 8 2 3 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 3 2 1	(j)
(c)	Zero padding 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 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)
(g)	Full correlation result 0 0 0 8 2 3 2 1 0 0 0 0	Full convolution result 0 0 0 1 2 3 2 8 0 0 0 0	(o)
	Cropped correlation result	Cropped convolution result	

FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse.

What is Spatial Filtering? – (cont.)

Correlation





$$
\begin{aligned}
e_{processed} = {} & v \times e + \\
& r \times a + s \times b + t \times c + \\
& u \times d + w \times f + \\
& x \times g + y \times h + z \times i
\end{aligned}
$$

What is Spatial Filtering? – (cont.)

Correlation

two remarks

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)

A 3×3 nighborhood

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

An m×n = 3×3 filter

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

$$m=2a+1,\, n=2b+1$$

What is Spatial Filtering? – (cont.)

Convolution

two remarks

a	b	c
d	e	f
g	h	i

Original Image Pixels

$*$

r	s	t
u	v	w
x	y	z

Filter

$$\begin{aligned} e_{processed} = {} & v \times e + \\ & z \times a + y \times b + x \times c + \\ & w \times d + u \times f + \\ & t \times g + s \times h + r \times i \end{aligned}$$

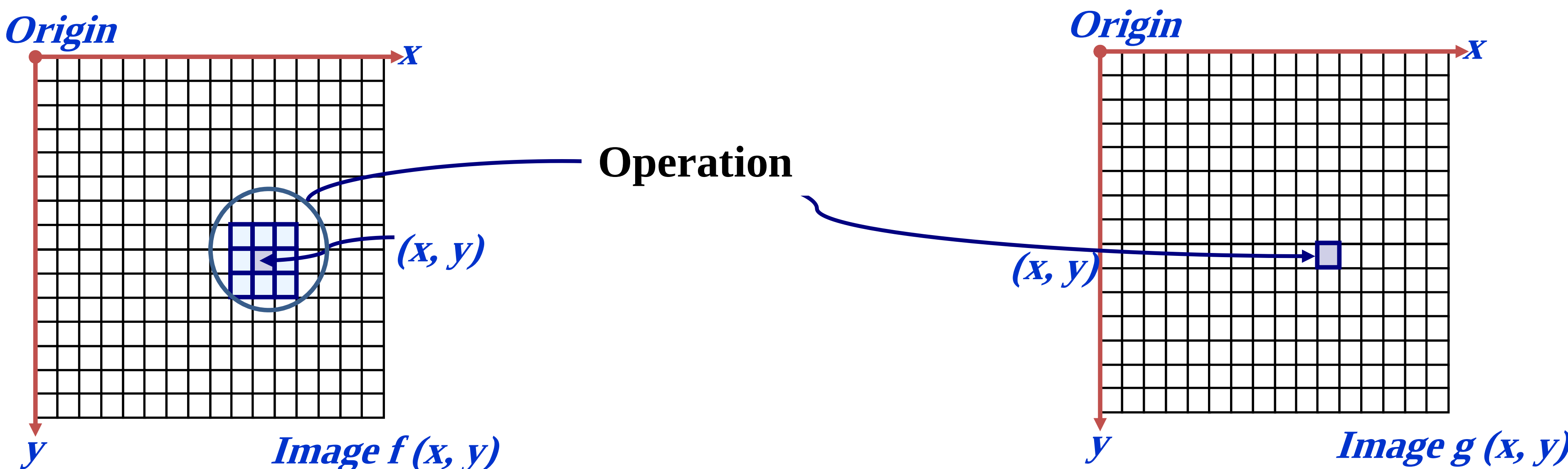
Mask is pre-rotated by 180^0 before operating (flipping along one axis then the other).

Unless mask is symmetric

What is Spatial Filtering? – (cont.)

Moving Window Transform

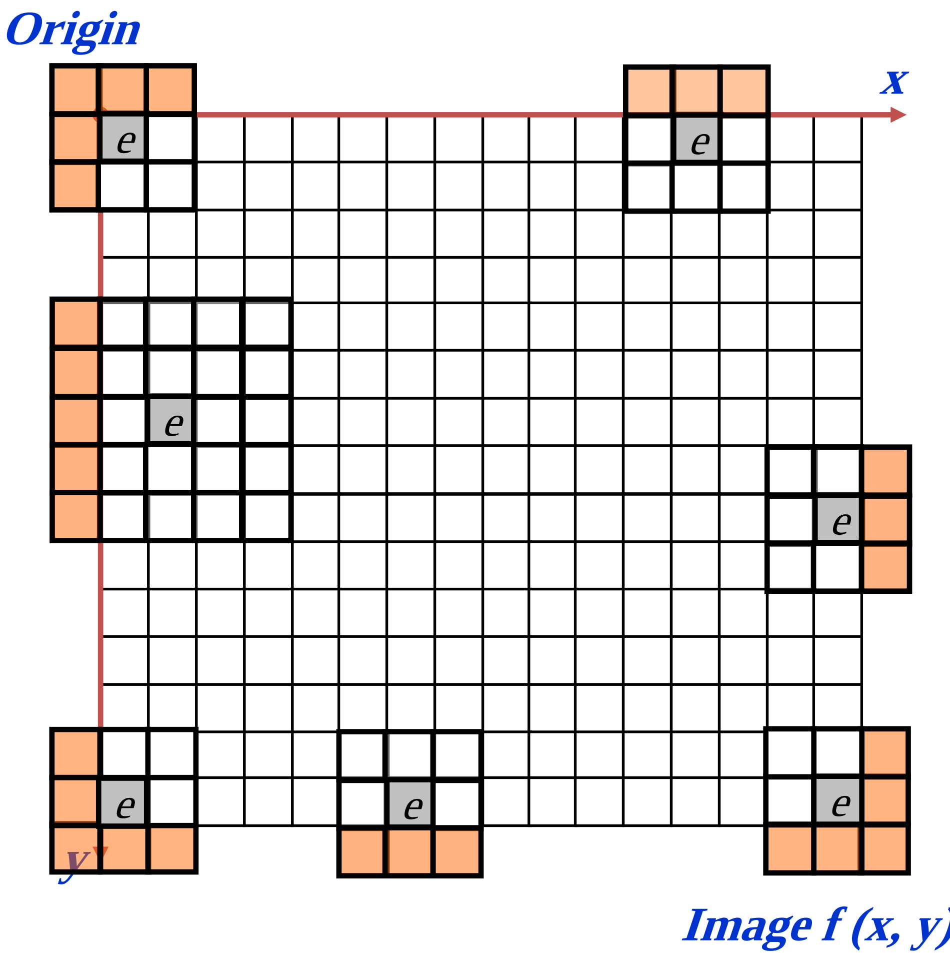
For any specific location *(x, y)*, the value of *g* at that location is the result of applying an operation to the pixels in the neighborhood with origin *(x, y)*.



What is Spatial Filtering? – (cont.)

Dealing with Image Borders

At the edges of an image we are missing pixels to form a neighbourhood.



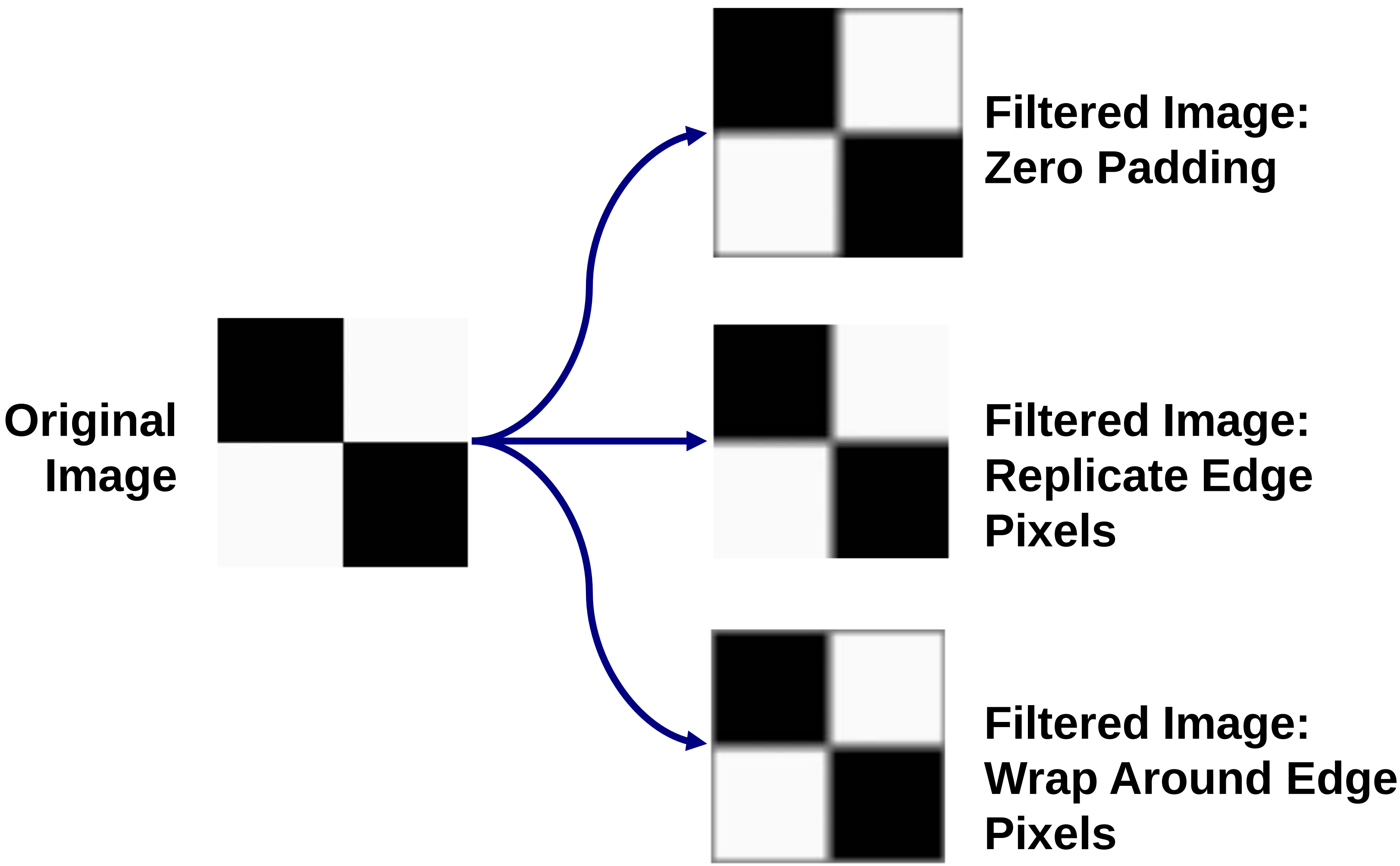
What is Spatial Filtering? – (cont.)

Dealing with Image Borders

- **Omit missing pixels.**
  - Only works with some filters.
  - Can add extra code and slow down processing.
- **Pad the image.**
  - Typically with either all white or all black pixels.
- **Replicate border pixels.**
- **Truncate the image.**
- **Allow pixels *wrap around* the image.**
  - Can cause some strange image artifacts.

What is Spatial Filtering? – (cont.)

Dealing with Image Borders



What is Spatial Filtering? – (cont.)

Basic rule

- To generate an $m \times n$ linear spatial filter, we need to specify mn mask coefficients (weights).
- These are selected based on what the filter is intended to do.

1. Smoothing Filters

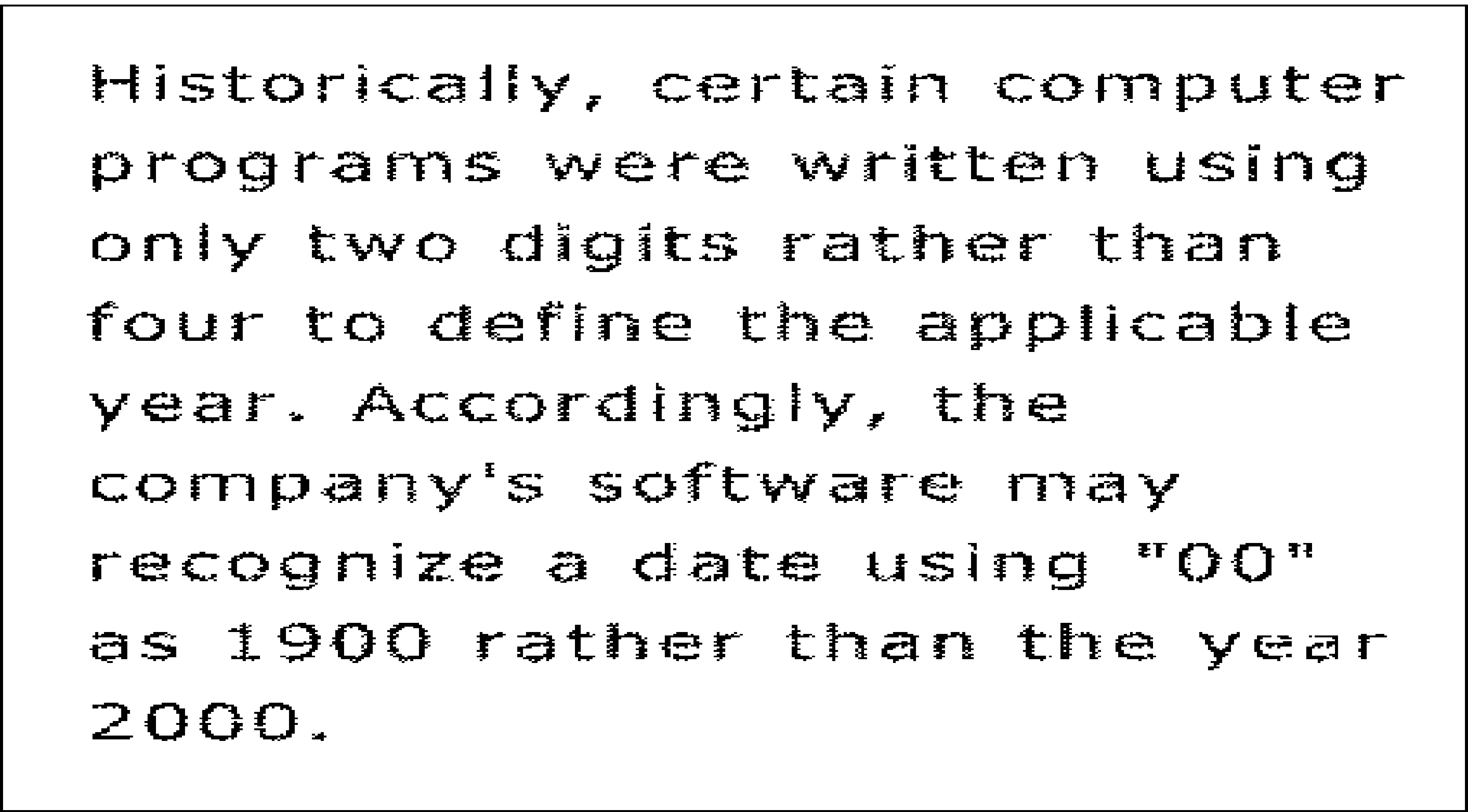
Smoothing Filters

Main applications:

- **Used for blurring and noise reduction.**
- **Removing small details prior to object extraction.**
- **Bridging small gaps in lines and curves.**
- **Linear and nonlinear filters.**

Smoothing Filters – (cont.)

Example: OCR preprocessing.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Smoothing Filters – (cont.)

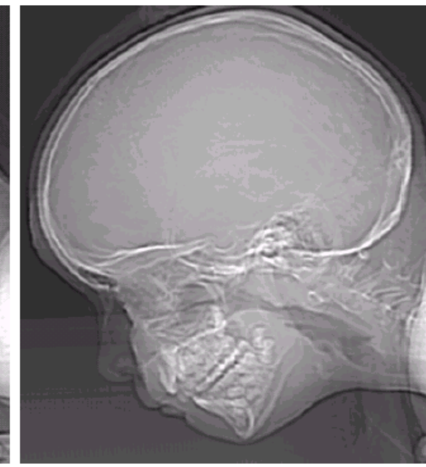
Example: False contours

256 grey levels (8 bits per pixel)

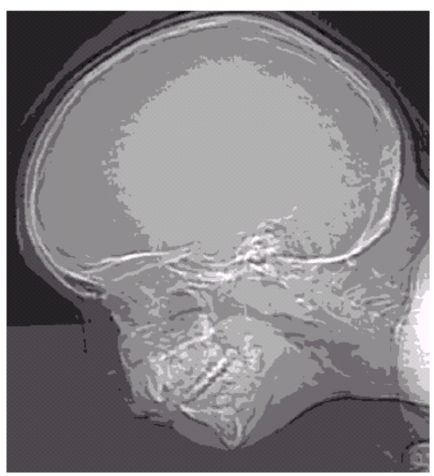
128 grey levels (7 bpp)

64 grey levels (6 bpp)

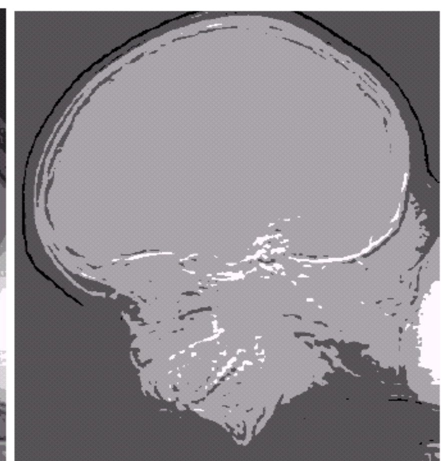
32 grey levels (5 bpp)



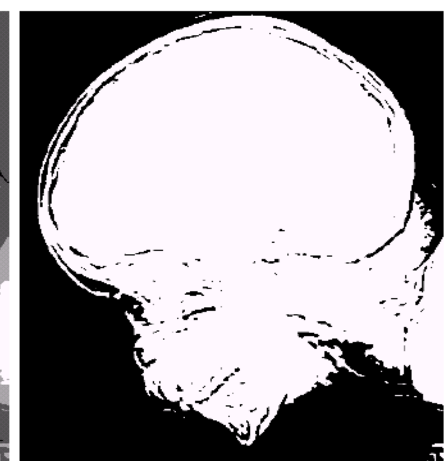
16 grey levels (4 bpp)



8 grey levels (3 bpp)



4 grey levels (2 bpp)



2 grey levels (1 bpp)

Smoothing Filters – (cont.)

Averaging Filter

- **One of the simplest linear spatial filtering operations.**
- ***Response=*** **average all of the pixels in a neighbourhood.** $$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$
- **Called low-pass filter.**

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Simple averaging filter

Smoothing Filters – (cont.)

Averaging Filter

- *Weighted average:* allowing different pixels in the neighbourhood different weights in the averaging function.
- Pixels closer to the central pixel are more important.
- More effective in smoothing.

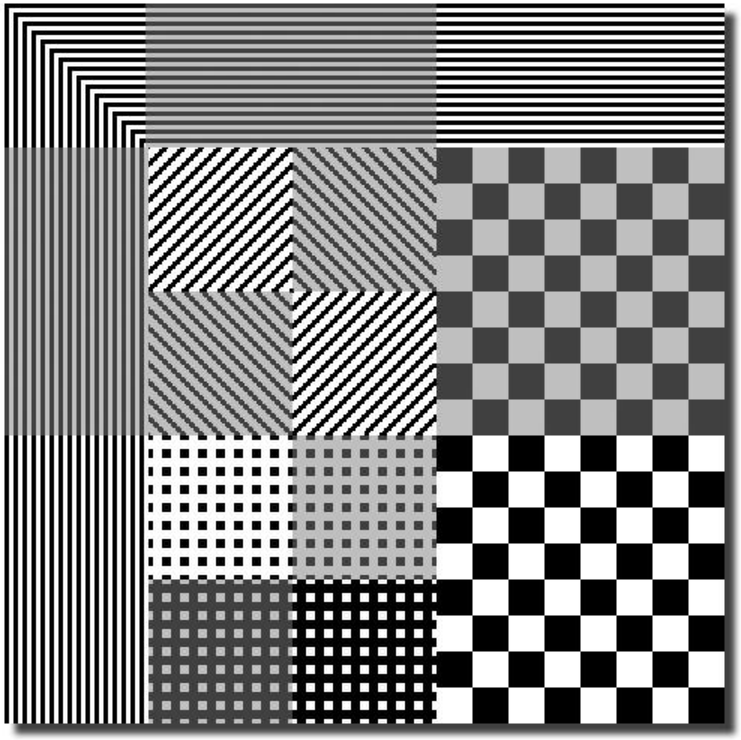
$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$
$^{2}/_{16}$	$^{4}/_{16}$	$^{2}/_{16}$
$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$

Weighted averaging filter

Smoothing Filters – (cont.)

Averaging Filter

- **Original**

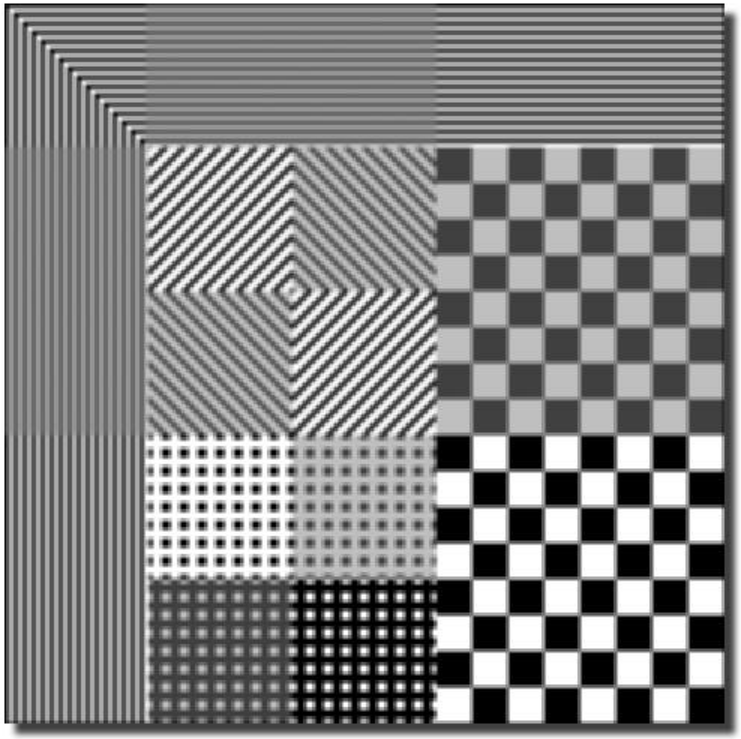


Smoothing Filters – (cont.)

Averaging Filter

- **3×3**

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

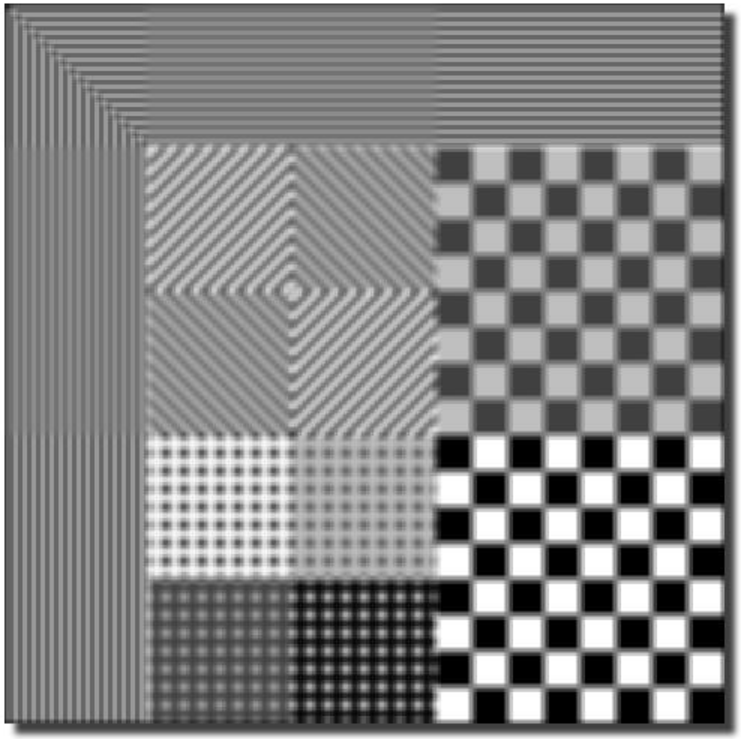


Smoothing Filters – (cont.)

Averaging Filter

- 5×5

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

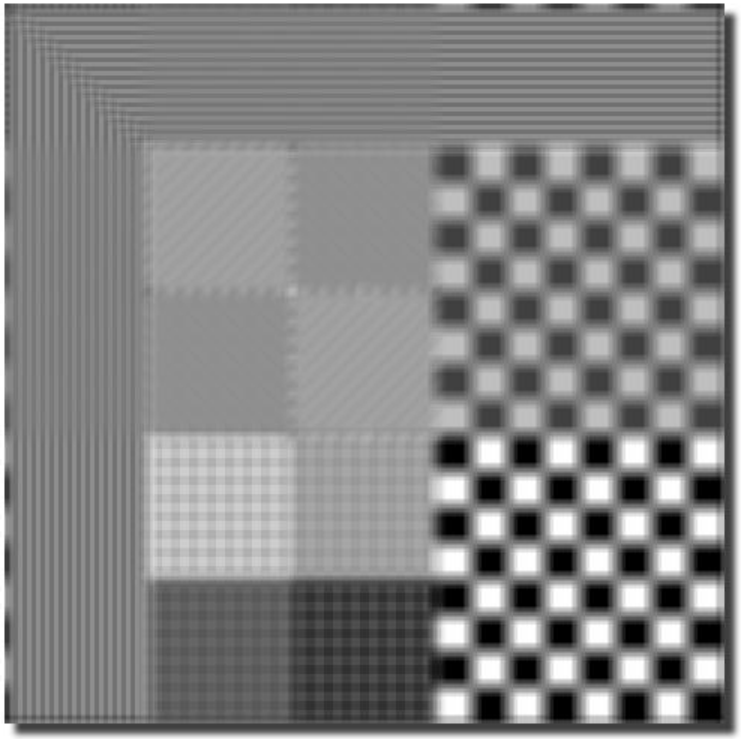


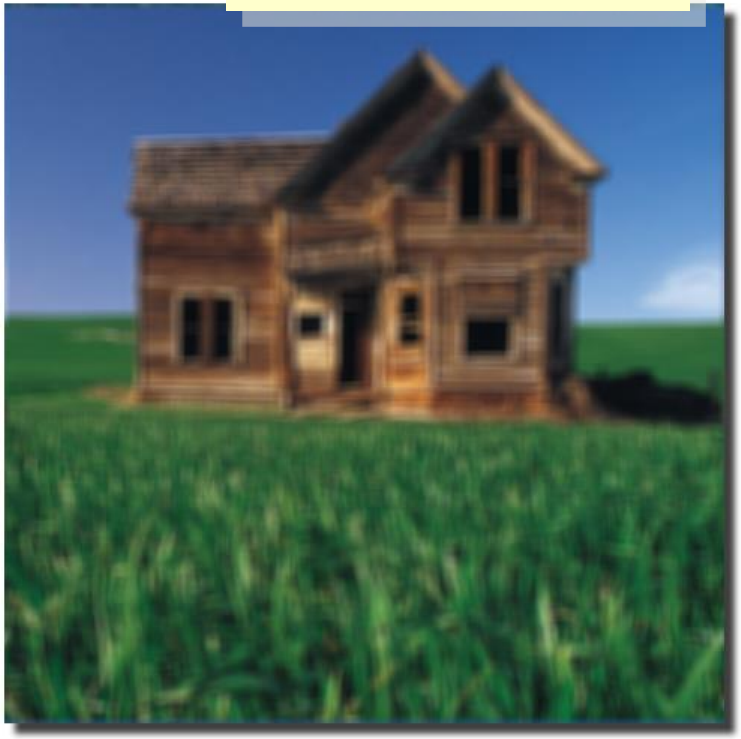
Smoothing Filters – (cont.)

Averaging Filter

- **9×9**

$$\frac{1}{81}\begin{bmatrix}1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\end{bmatrix}$$



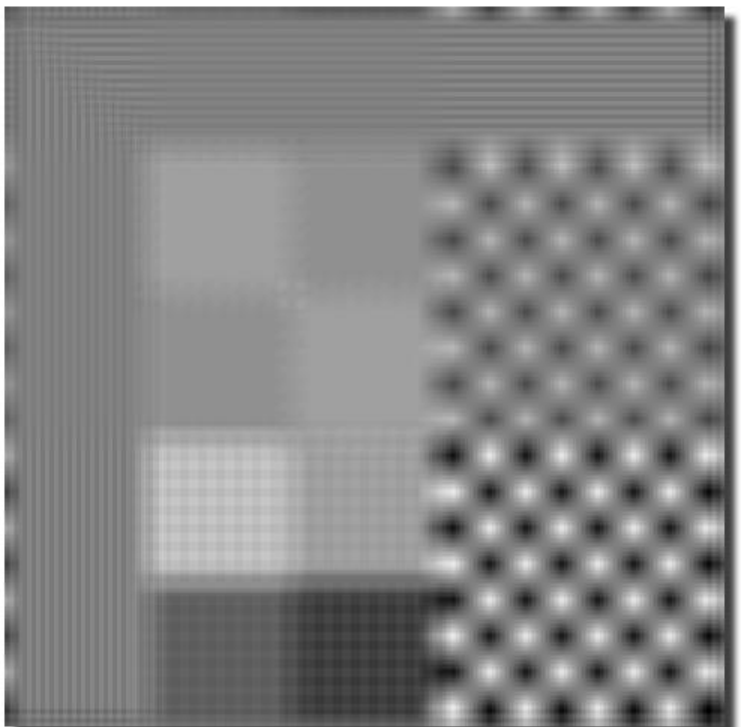


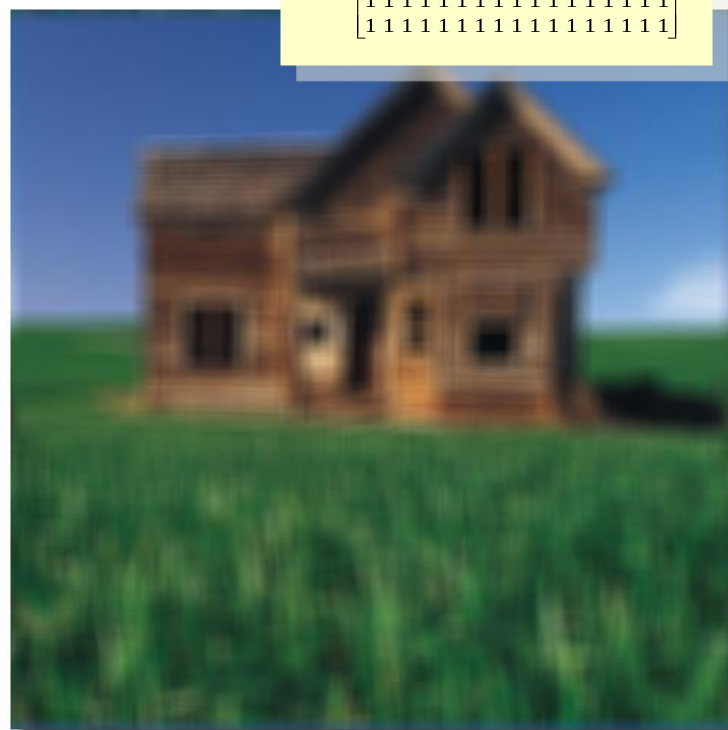
Smoothing Filters – (cont.)

Averaging Filter

- **17×17**

$$\frac{1}{289}\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

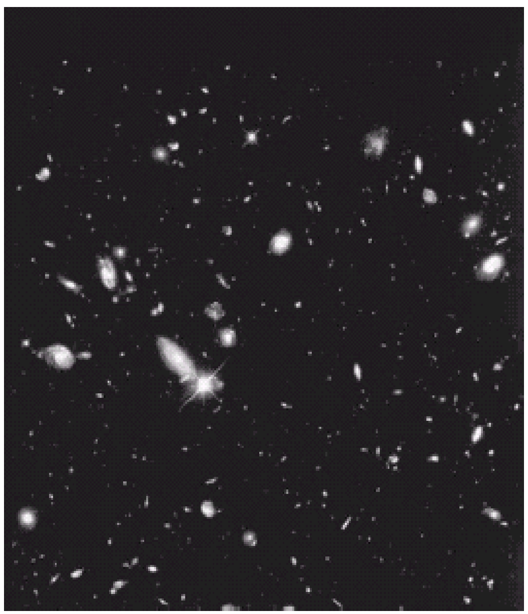




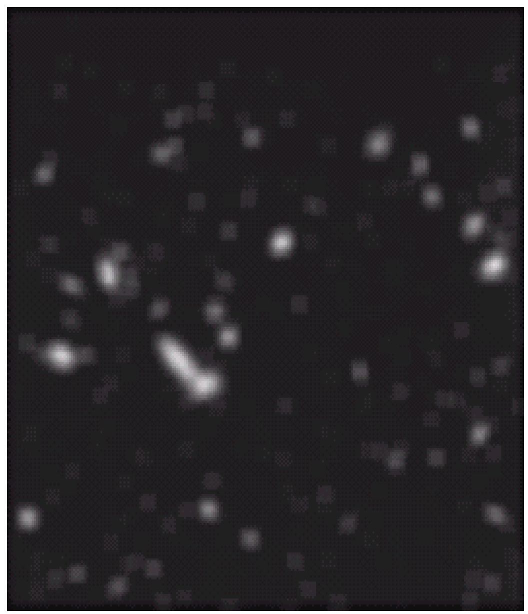
Smoothing Filters – (cont.)

Averaging Filter

- **Highlighting gross detail:** the size of the mask determines the relative size of the objects blended.



Original Image



Smoothed Image

Thresholded Image

Smoothing Filters – (cont.)

Averaging Filter Usage

- **Results an image with reduced sharp transitions in intensities.**
- **Useful in removing noise and reducing irrelevant detail (highlighting gross detail).**
- **Side effect: blurred edges.**

Smoothing Filters – (cont.)

Nonlinear Filters

- **Spatial filters whose response is based on ordering (ranking) the pixels in the image area encompassed by the filer, and replacing the center pixel with the value determined by the ranking result.**
- **Examples:**
  - min filter, max filter, <u>median filter</u>.

Smoothing Filters – (cont.)

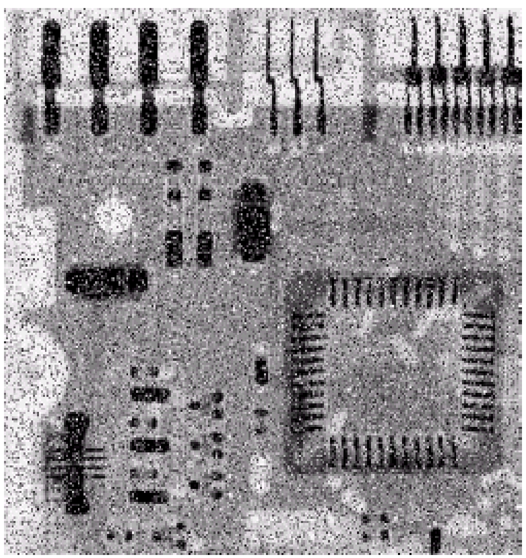
Nonlinear Filters

- **<u>Median Filter</u>: sort pixels in neighborhood, determine median, assign to center pixel.**
- **Forces points with distinct intensity levels to be like their neighbors.**
- **Eliminates isolated clusters of pixels that are light or dark with respect to their neighbors.**

Smoothing Filters – (cont.)

Median Filter

- **Excellent salt-and-pepper noise reduction.**



Original Image With Noise

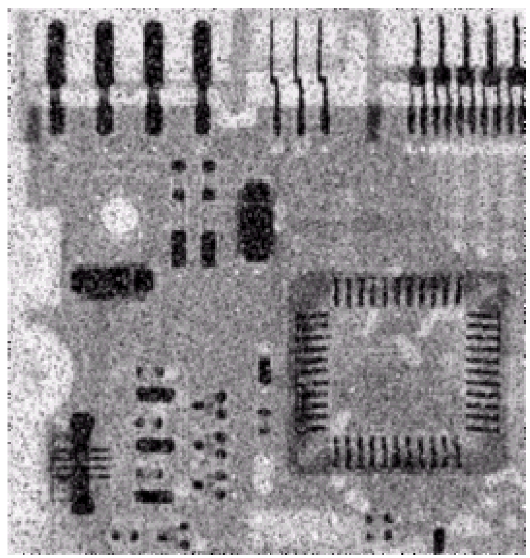


Image After Averaging Filter

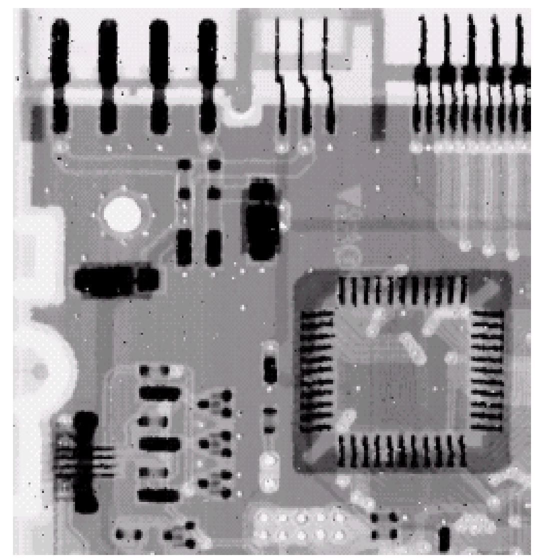
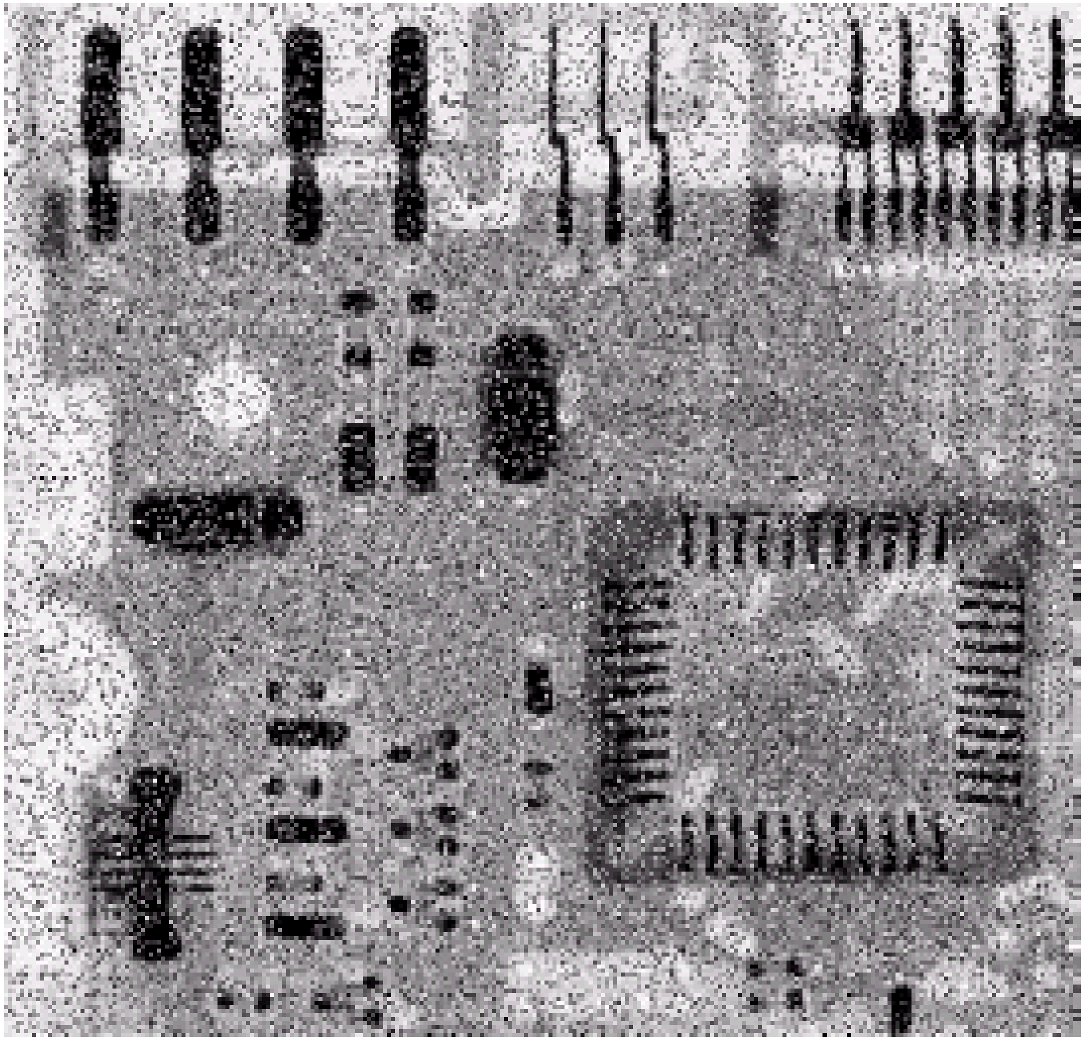


Image After Median Filter

Smoothing Filters – (cont.)

Example

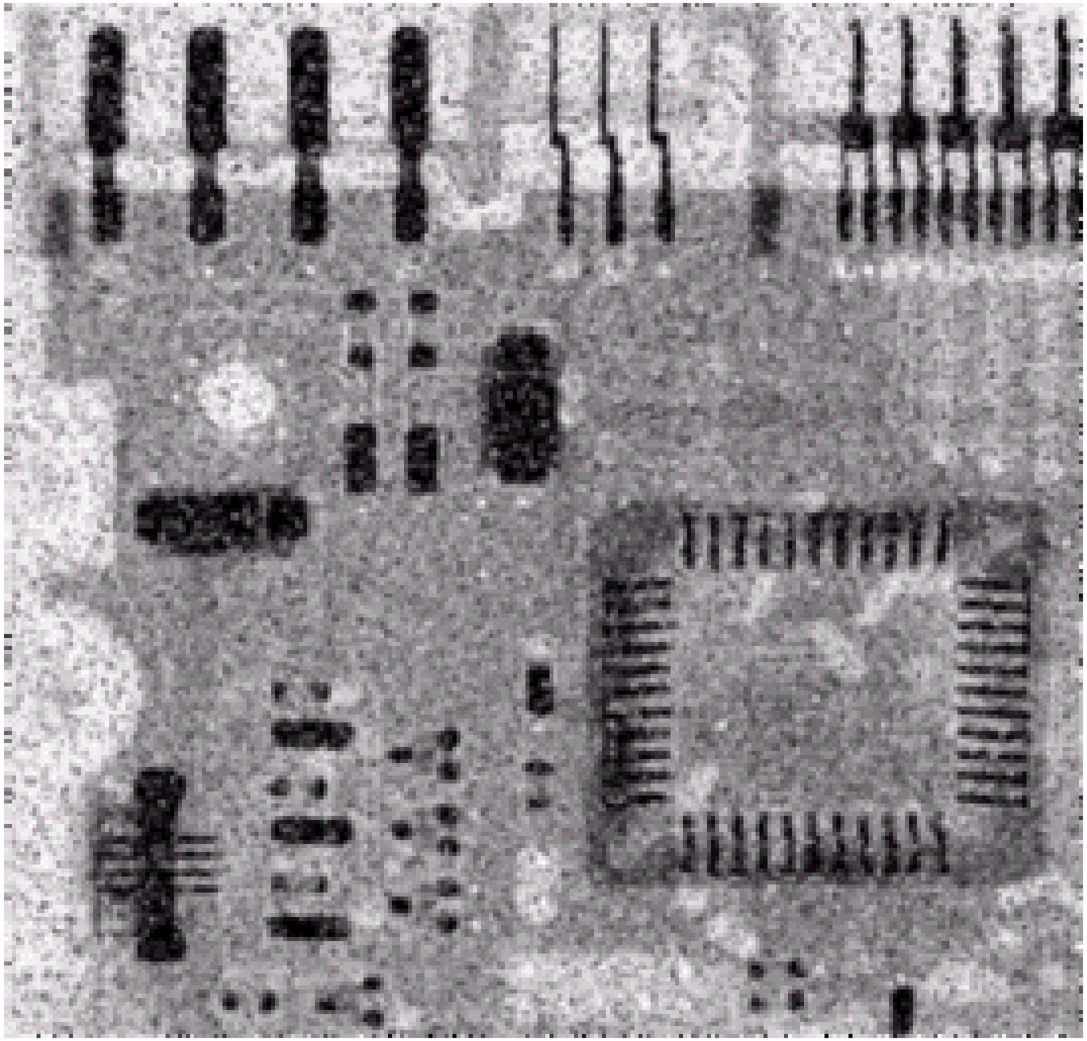
- **Original**



Smoothing Filters – (cont.)

Example

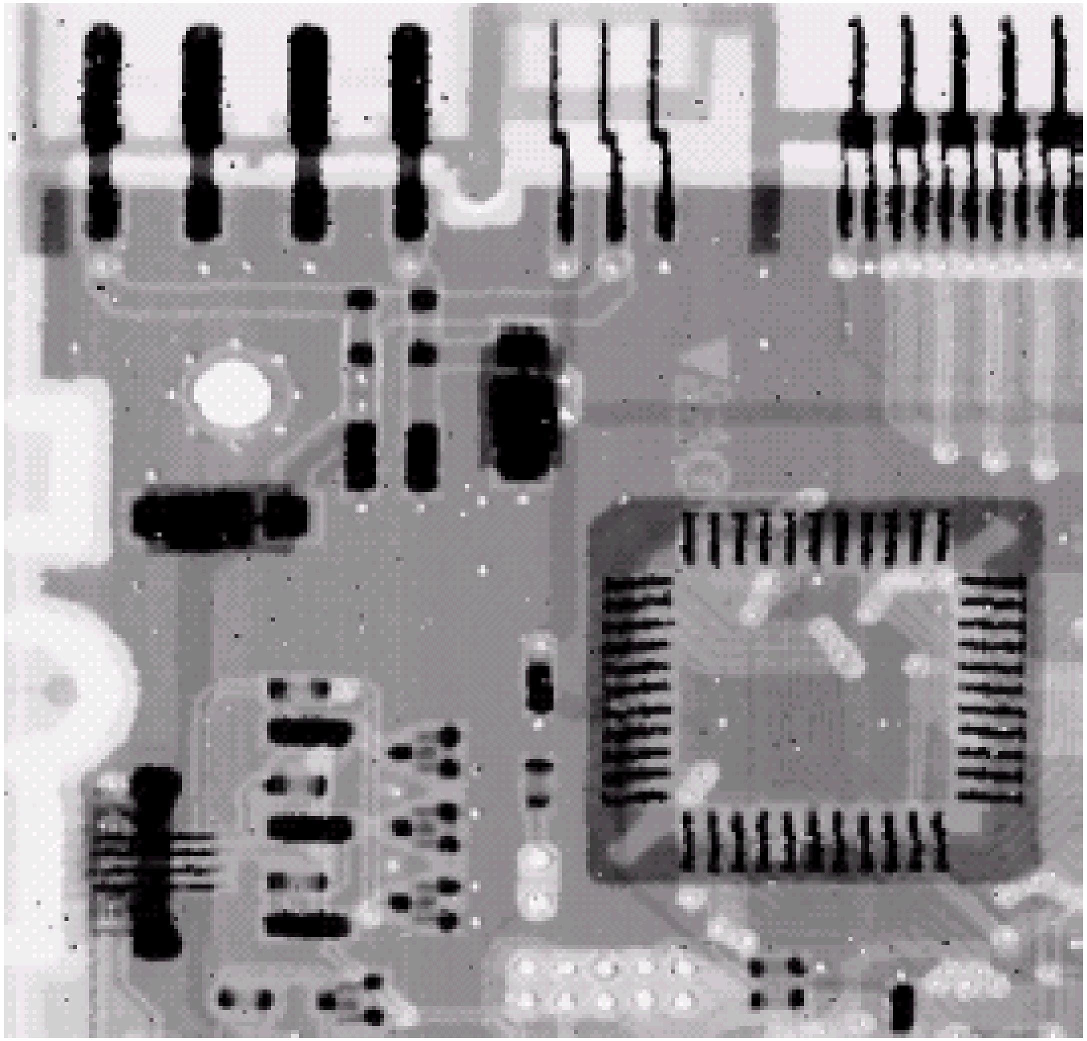
- **Average**



Smoothing Filters – (cont.)

Example

- **Median**



2. Sharpening Filters and Edge Detectors

Sharpening Filters

Main objectives:

- **Highlight transitions in intensities → highlight edges.**
- **Remove blurring → enhance details.**
- **Sharpening filters are based on *spatial differentiation*, which measures the rate of change of a function.**
- **First and second derivatives are used for image enhancement.**

Sharpening Filters – (cont.)

1st Derivative:

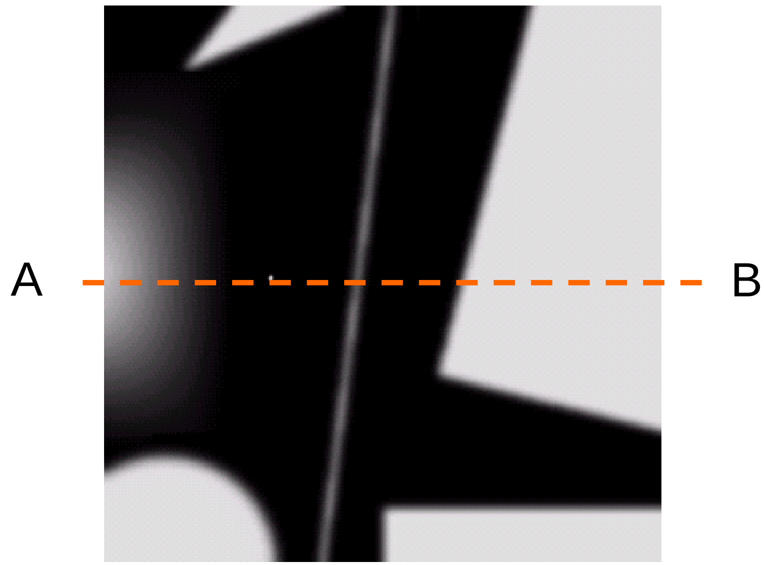
The formula for the 1st derivative of a function is as follows:

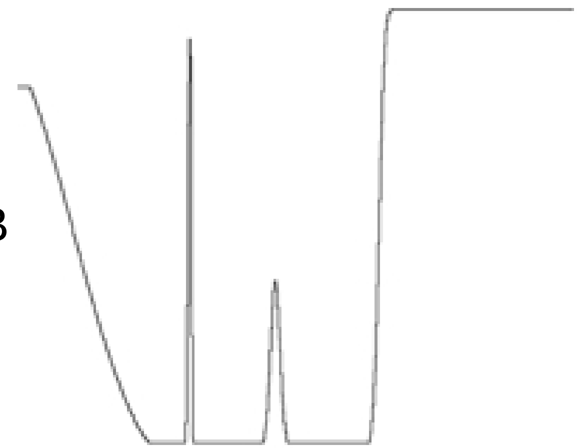
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

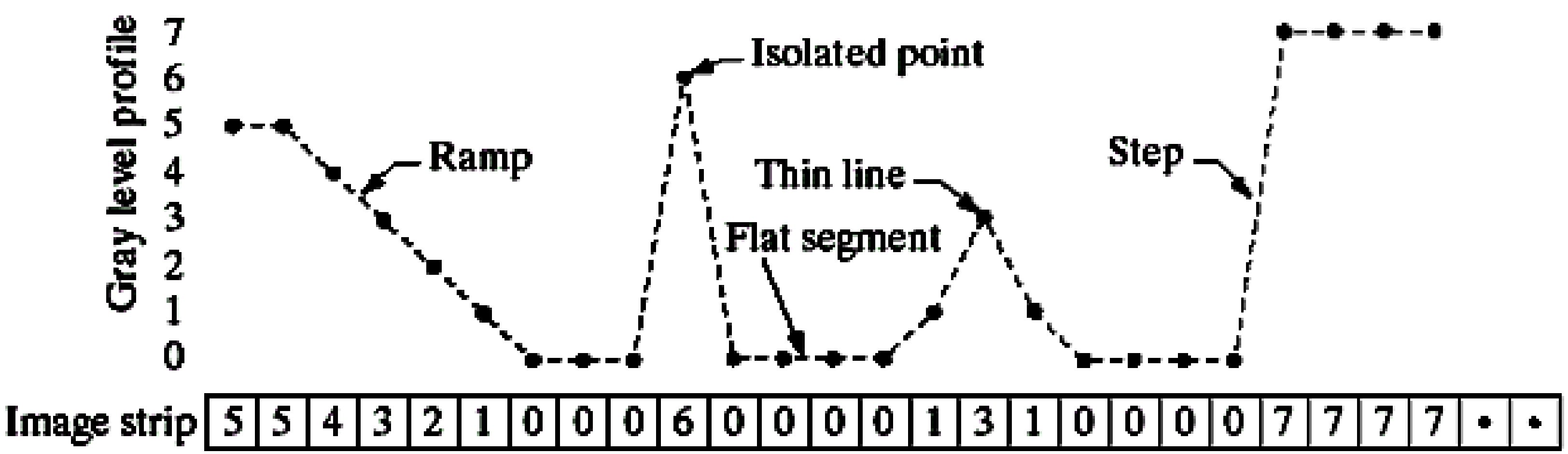
It's just the difference between subsequent values and measures the rate of change of the function.

Sharpening Filters – (cont.)

1st Derivative Example in 1D:

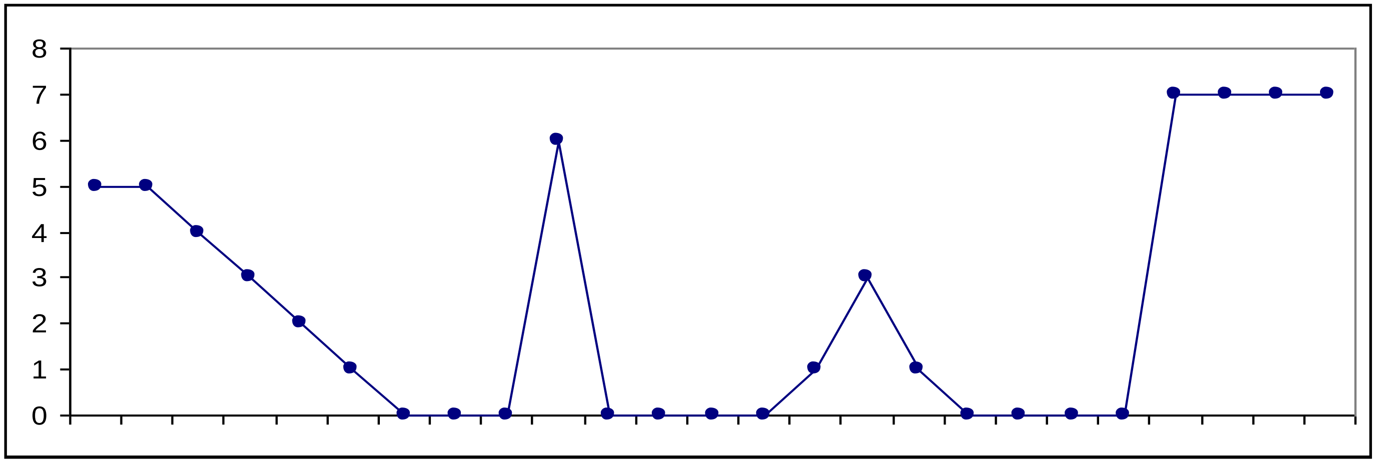






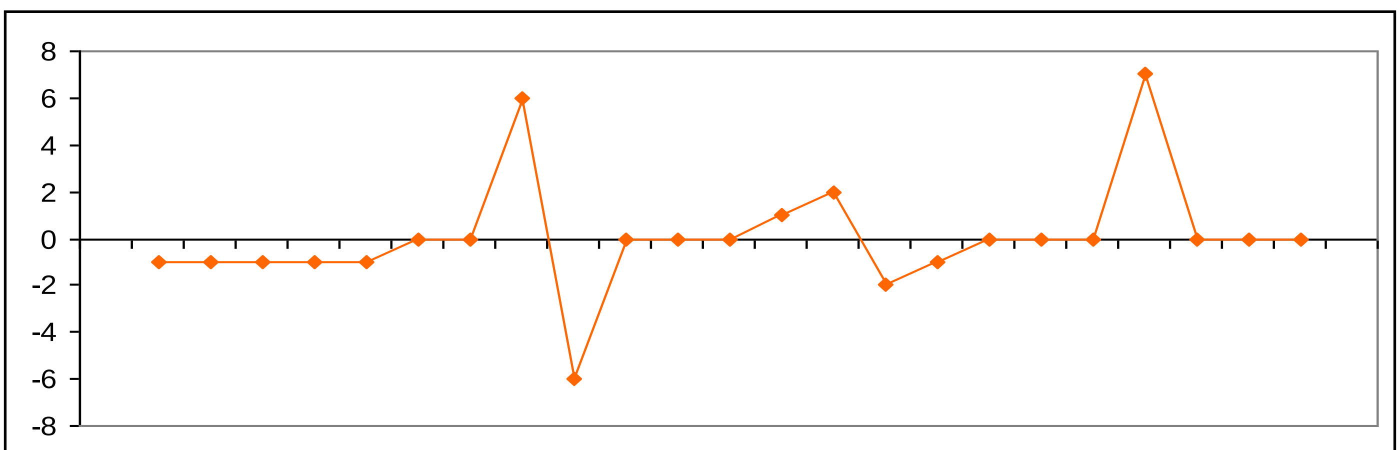
Sharpening Filters – (cont.)

1^{st} Derivative Example in 1D:



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

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



Sharpening Filters – (cont.)

2nd Derivative:

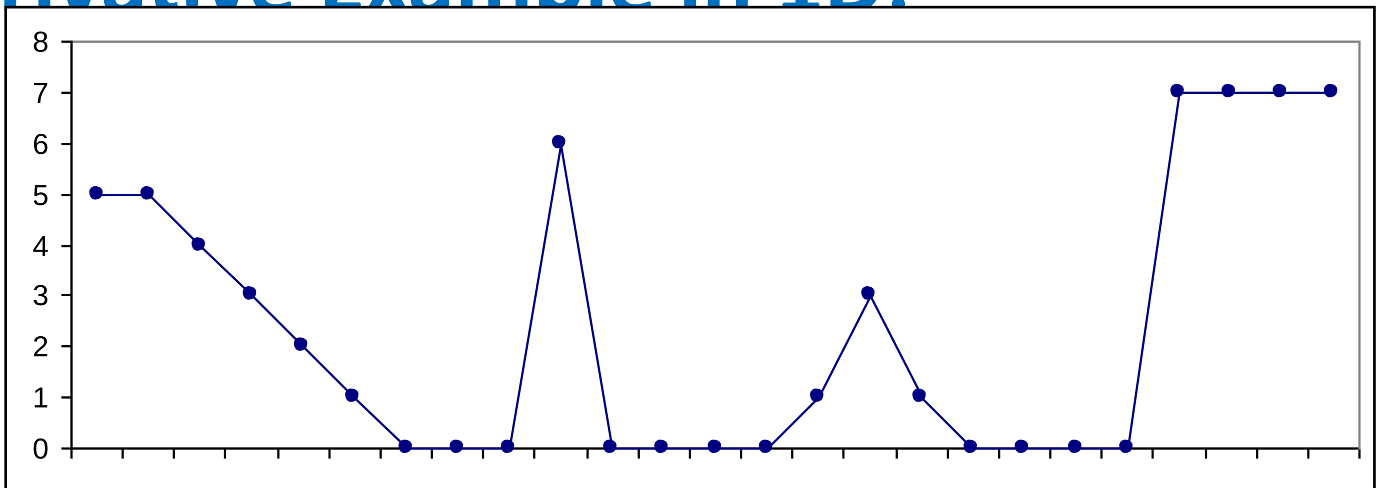
The formula for the 2nd derivative of a function is as follows:

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

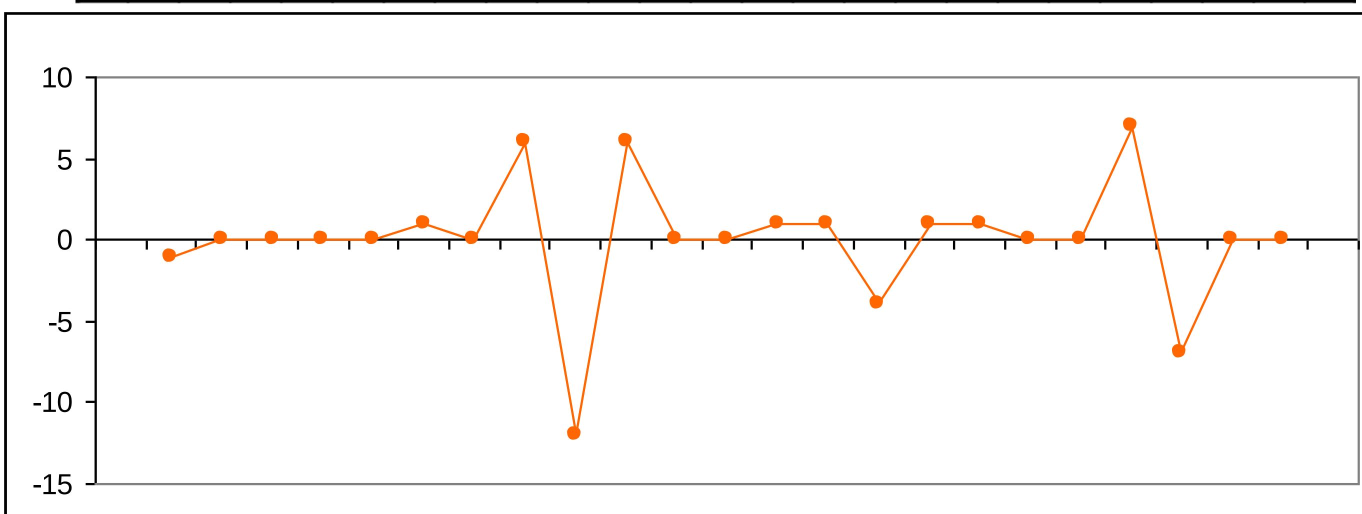
Simply takes into account the values both before and after the current value.

Sharpening Filters – (cont.)

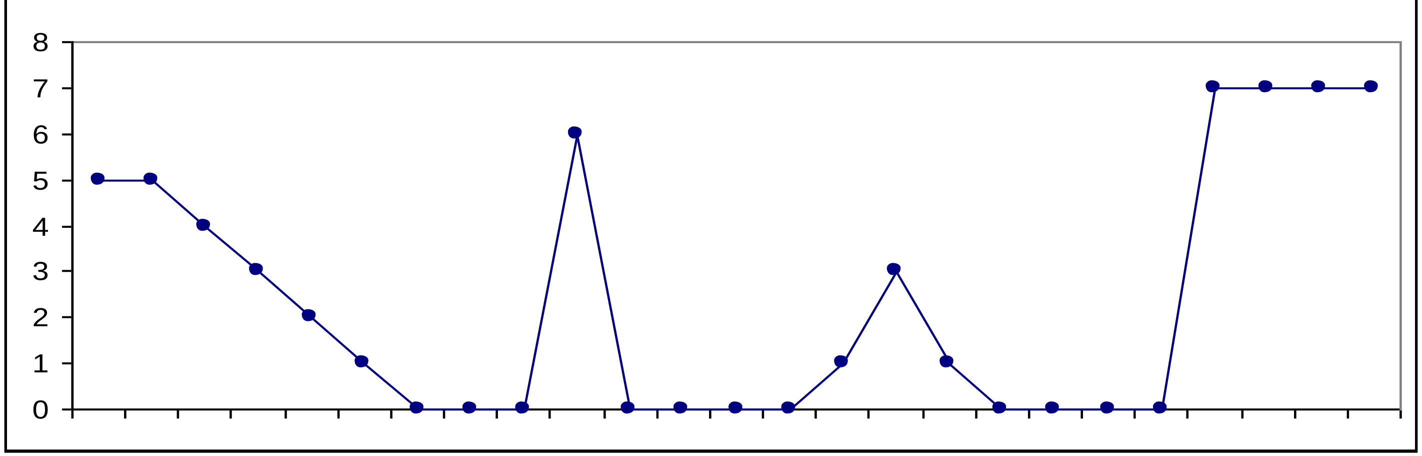
2nd Derivative Example in 1D:



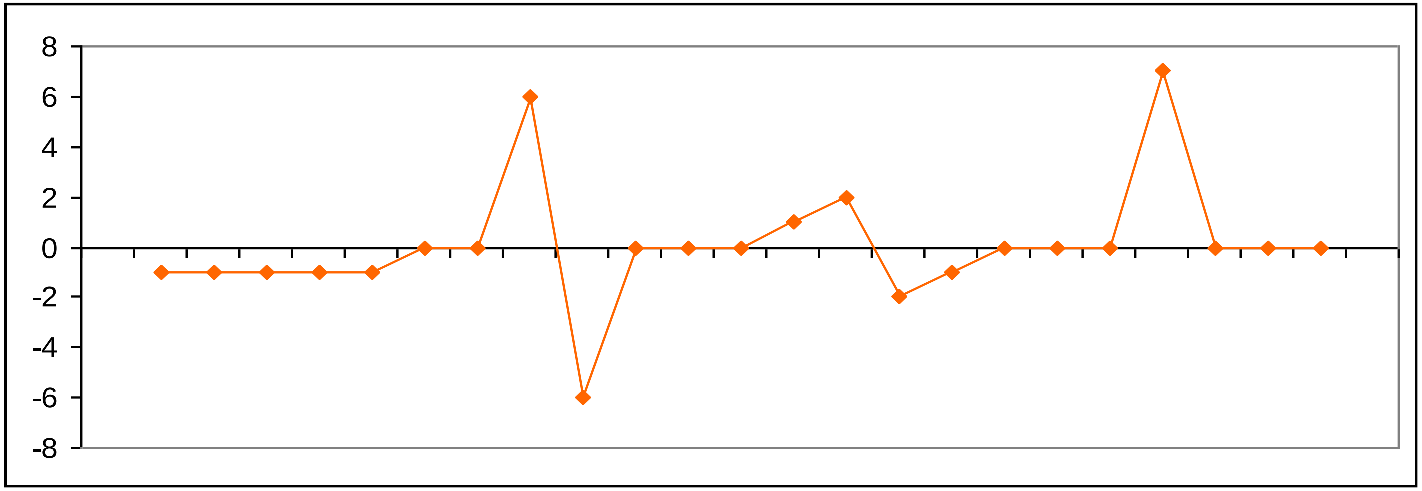
5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



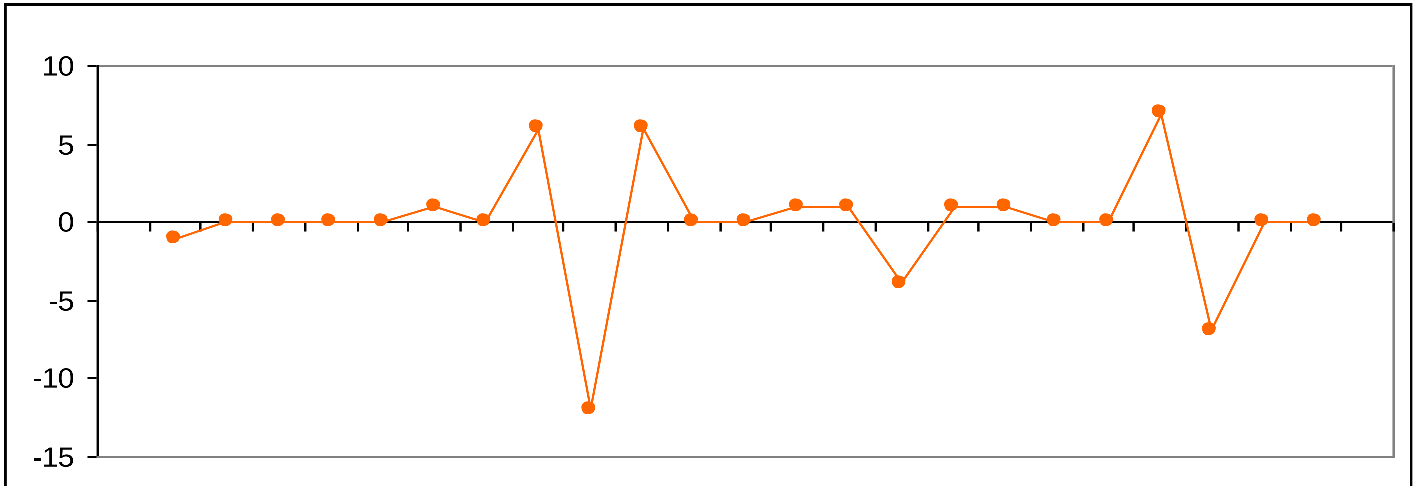
Profile



1st



2nd



Sharpening Filters – (cont.)

Example in 1D:

- **Note that the sign of the 2nd derivative changes at the onset and end of ramp and step → zero crossing → useful in locating edges.**
- **The 2nd derivative is more useful for image enhancement than the 1st derivative**
  - **Stronger response to fine detail.**
  - **Simpler implementation.**

Sharpening Filters – (cont.)

Second Derivative (Laplacian)

The Laplacian is defined as follows:

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

where in the x direction:

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

and in the y direction:

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

Sharpening Filters – (cont.)

Second Derivative (Laplacian)

Hence the Laplacian is given by:

$$\begin{aligned}\nabla^2 f = [\, & f(x+1, y) + f(x-1, y) \\ & + f(x, y+1) + f(x, y-1)] \\ & - 4f(x, y)\end{aligned}$$

We can easily build a filter based on this.

Sharpening Filters – (cont.)

Second Derivative (Laplacian)

Hence the Laplacian is given by:

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

We can easily build a filter based on this.

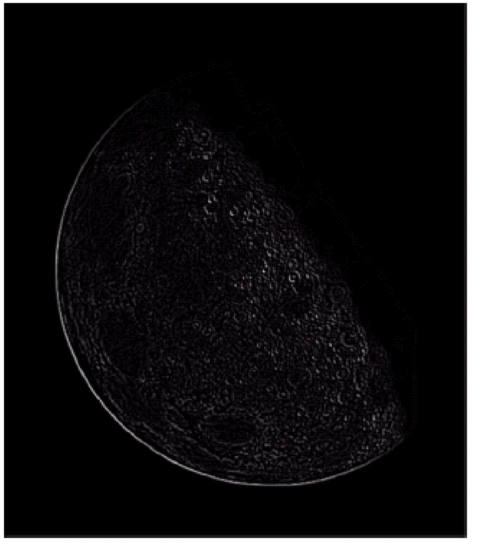
0	1	0
1	-4	1
0	1	0

Sharpening Filters – (cont.)

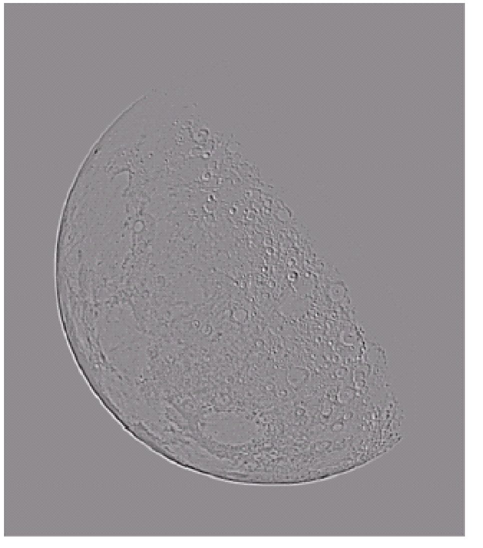
Second Derivative (Laplacian)

Applying the Laplacian gives a new image that highlights edges and other discontinuities.

Original Image



Laplacian Filtered Image



Laplacian Filtered Image Scaled for Display

Sharpening Filters – (cont.)

Second Derivative (Laplacian)

Subtract a weighted Laplacian result from the original image to obtain the sharpened image.

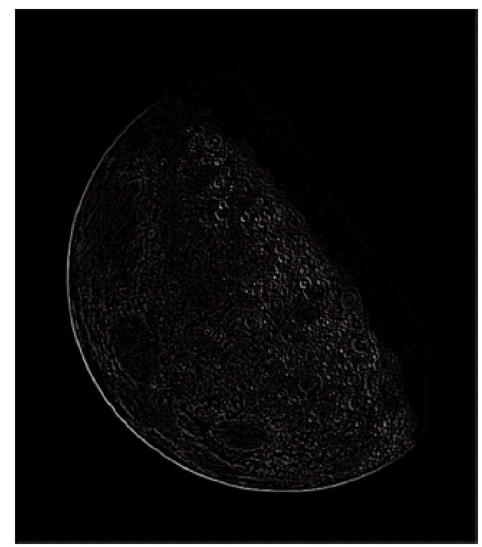
$$g(x,y) = f(x,y) - \nabla^2 f$$

Sharpened Image

=

Original Image

-



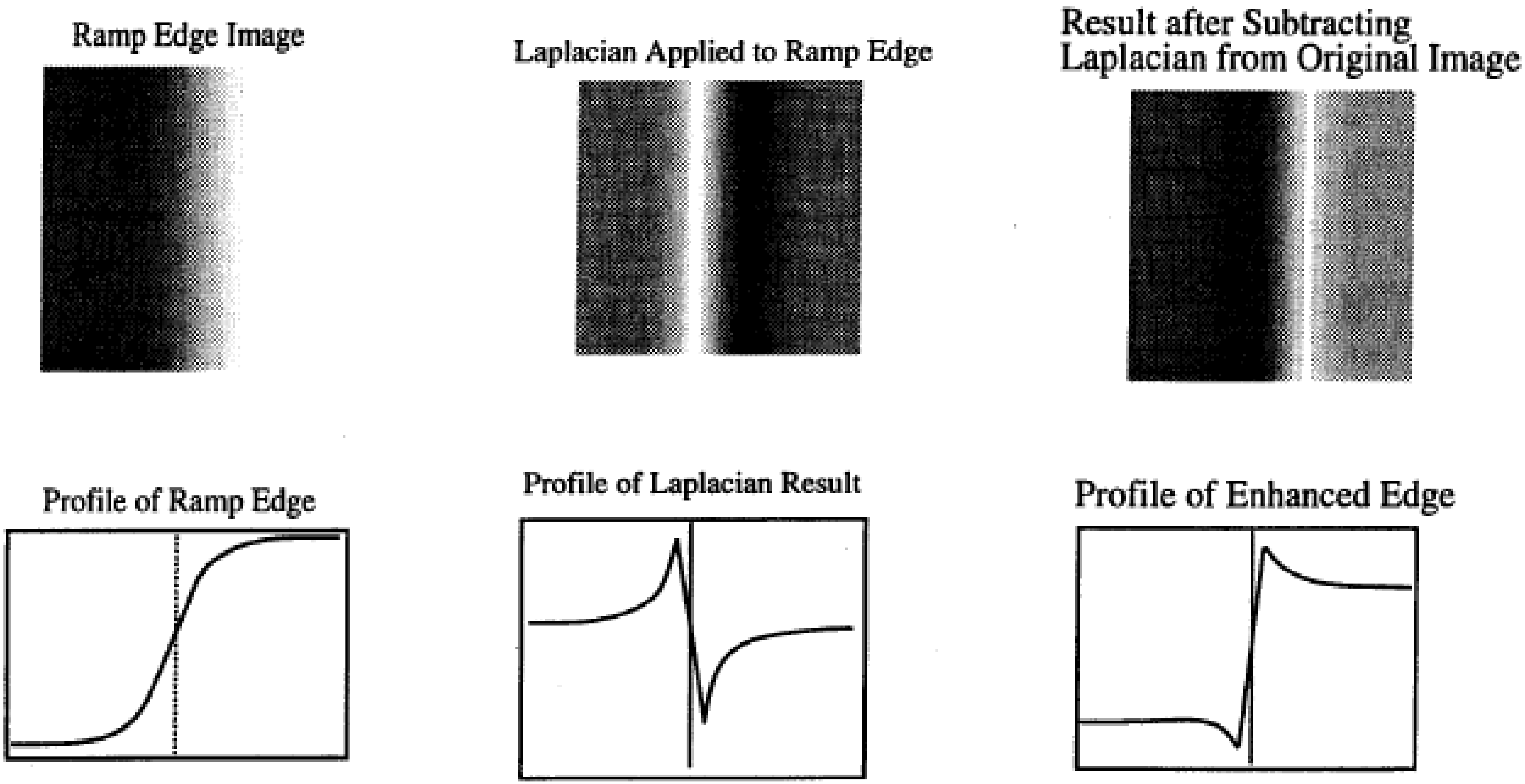
Laplacian Filtered Image

Sharpening Filters – (cont.)

Second Derivative (Laplacian)

Sharpening Filters – (cont.)

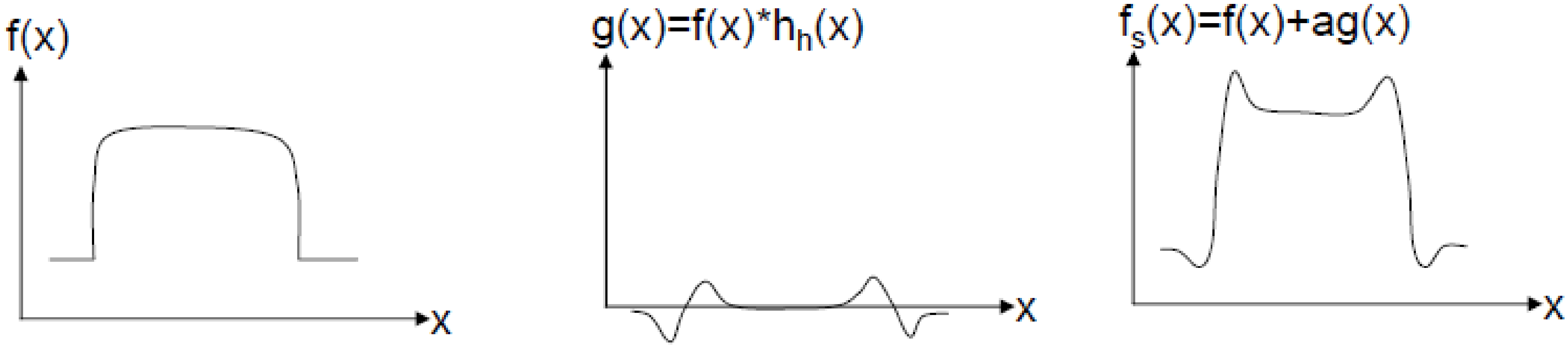
Second Derivative (Laplacian)



Sharpening Filters – (cont.)

Second Derivative (Laplacian)

Subtracting or adding a weighted Laplacian depending on the sign of the mask coefficients.



Sharpening Filters – (cont.)

Second Derivative (Laplacian)

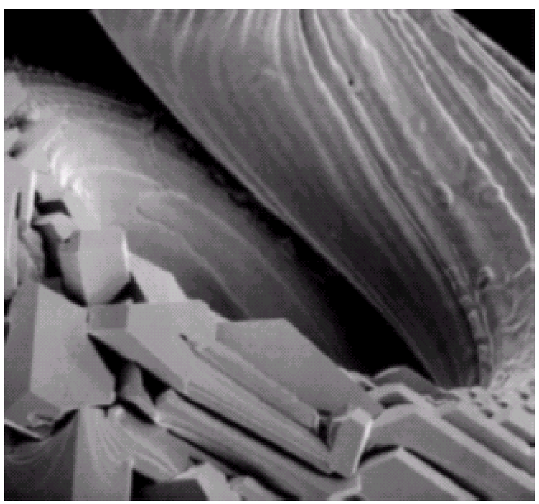
The entire enhancement can be combined into a single filtering operation,

$$
\begin{aligned}
g(x, y) &= f(x, y) - \nabla^2 f \\
&= f(x, y) - [f(x+1, y) + f(x-1, y) \\
&\qquad + f(x, y+1) + f(x, y-1) \\
&\qquad - 4f(x, y)] \\
&= 5f(x, y) - f(x+1, y) - f(x-1, y) \\
&\qquad - f(x, y+1) - f(x, y-1)
\end{aligned}
$$

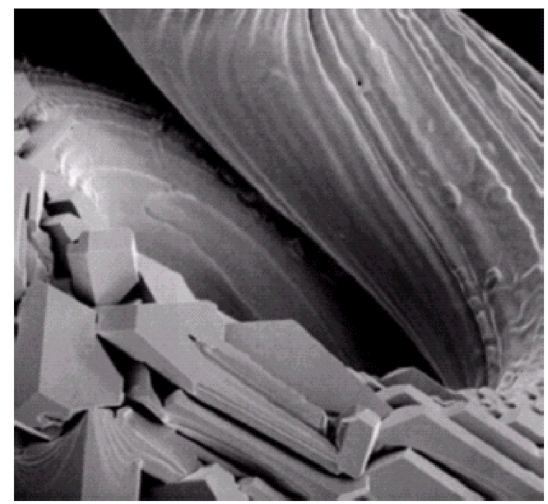
Sharpening Filters – (cont.)

Second Derivative (Laplacian)

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



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



Sharpening Filters – (cont.)

Second Derivative (Laplacian)

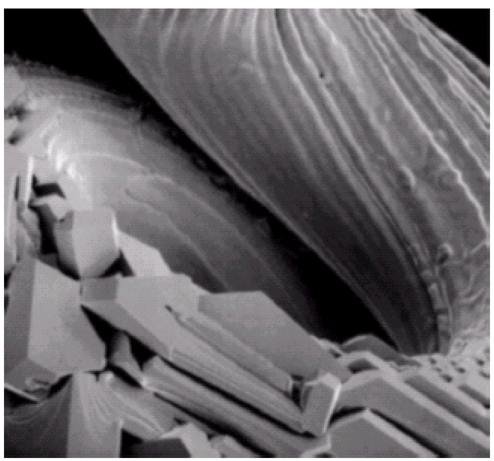
There are lots of slightly different versions:

0	1	0
1	-4	1
0	1	0

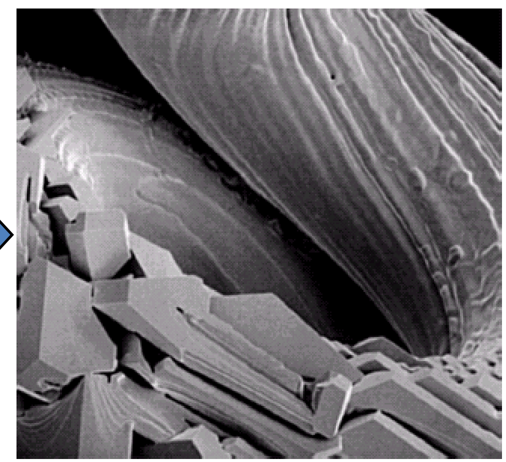
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



-1	-1	-1
-1	9	-1
-1	-1	-1



Sharpening Filters – (cont.)

First Derivative (Gradient)

The Gradient of $f(x, y)$ **at coordinates** (x, y) **is defined as the column vector:**

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

Sharpening Filters – (cont.)

First Derivative (Gradient)

The magnitude of this vector is given by:

$$
\begin{aligned}
M(x,y) &= mag(\nabla \mathrm{f}) \\
&= \left[G_x^2 + G_y^2\right]^{1/2} \\
&= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}
\end{aligned}
$$

For practical reasons this can be simplified as:

$$M(x,y) \approx \left|G_x\right| + \left|G_y\right|$$

Sharpening Filters – (cont.)

First Derivative (Gradient)

Approximating the partial derivative in the *x*-direction == third row – first row

Approximating the partial derivative in the *y*-direction == third column – first column

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)

=

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sharpening Filters – (cont.)

First Derivative (Gradient)

Approximating the partial derivative in the *x*-direction == third row – first row

Approximating the partial derivative in the *y*-direction == third column – first column

$$M(x,y) \approx \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Perwitt masks

Sharpening Filters – (cont.)

First Derivative (Gradient)

Based on the previous equations we can derive the *Sobel Operators*

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

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

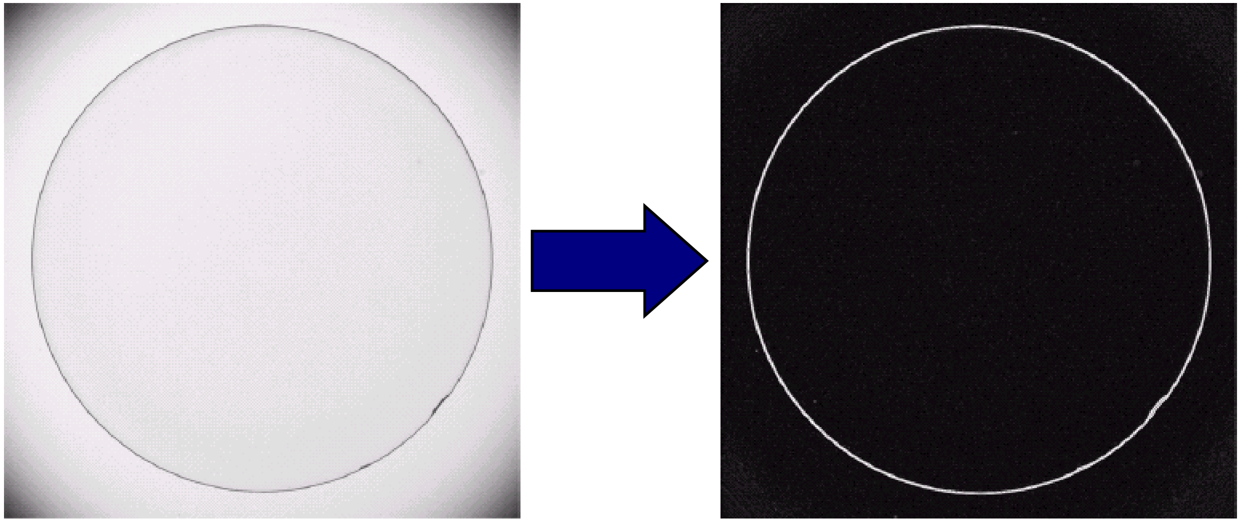
An image is filtered using both operators and the results are added together.

Note that they do not need to be rotated before convolution.

Sharpening Filters – (cont.)

First Derivative (Gradient)

Sobel filters are typically used for edge detection.



Sharpening Filters – (cont.)

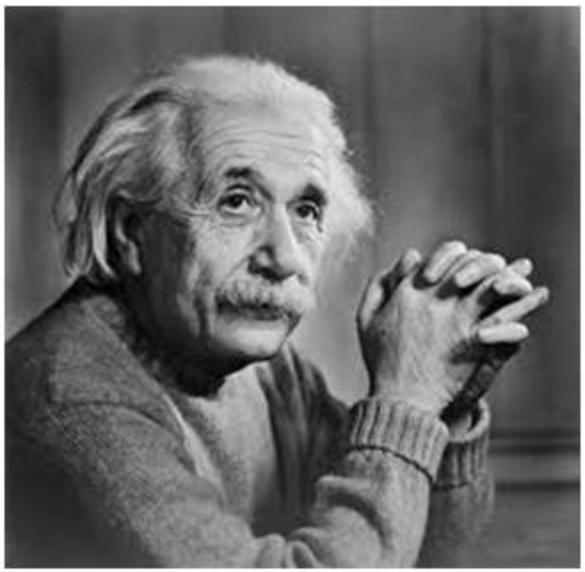
Original Image

Horizontal Gradient Component

Vertical Gradient Component

Combined Edge Image

Sharpening Filters – (cont.)



Original Image

Vertical Gradient Component

Horizontal Gradient Component

Sharpening Filters – (cont.)

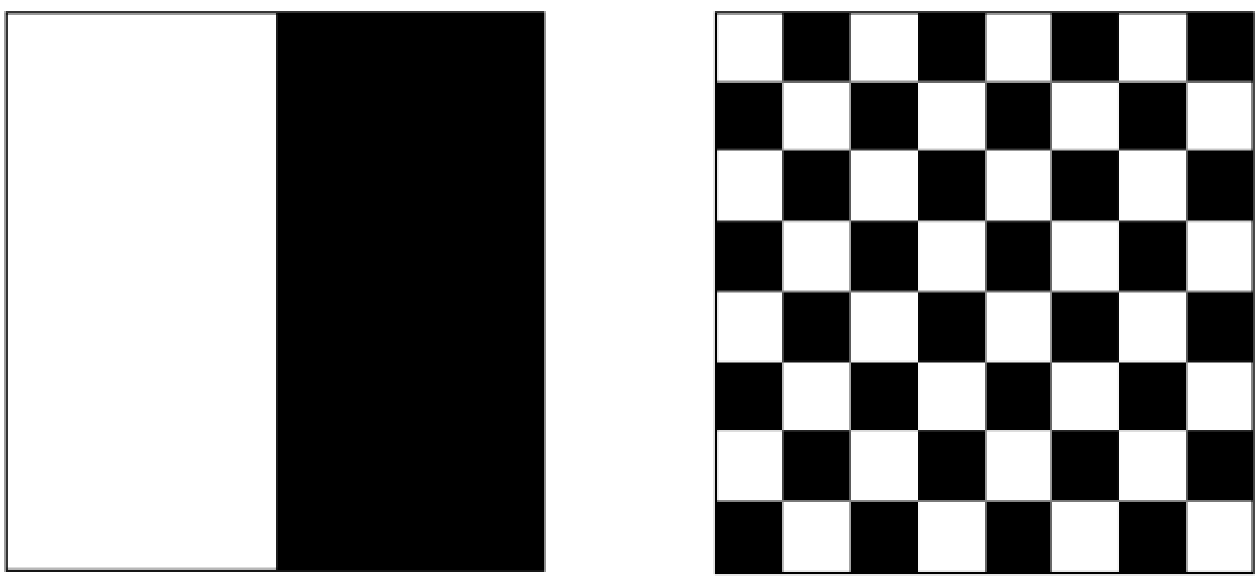
Comparing the 1^{st} and 2^{nd} derivatives we can conclude the following:

- **1^{st} order derivatives generally produce thicker edges.**
- **1^{st} order derivatives have stronger response to grey level step.**
- **2^{nd} order derivatives produce a double response at step changes in grey level.**
- **2^{nd} order derivatives have a stronger response to fine detail e.g. thin lines.**

Selected Problems

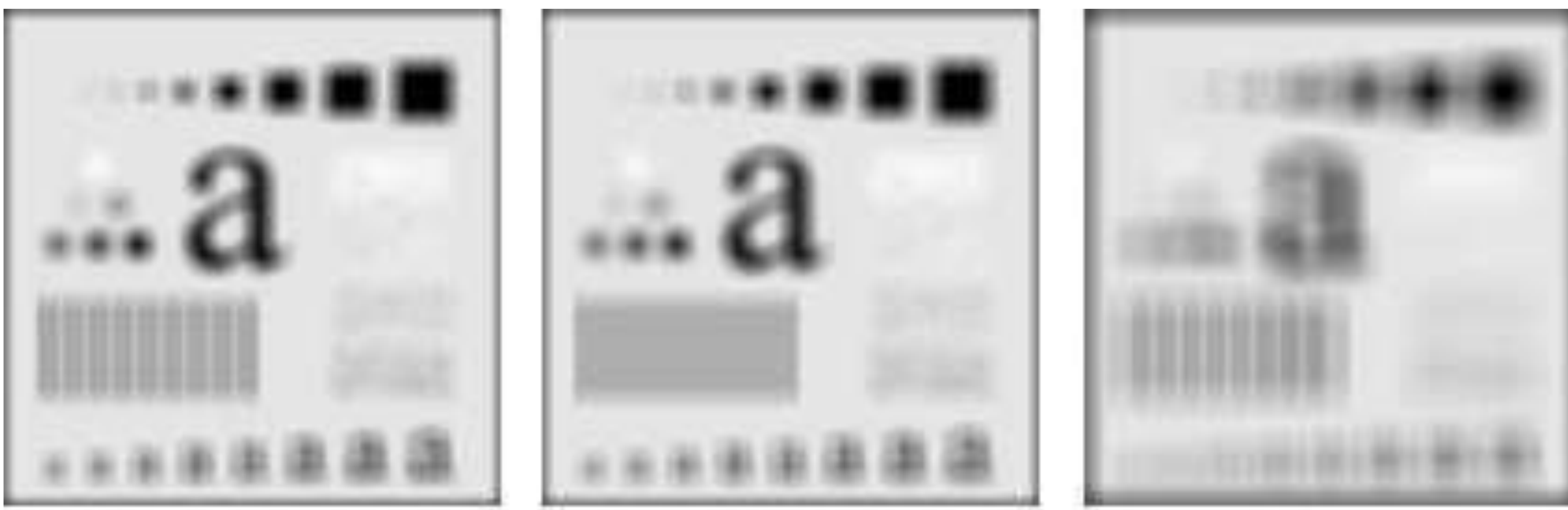
3.14 The images shown on the next page are quite different, but their histograms are the same. Suppose that each image is blurred with a 3×3 averaging mask.

(a) Would the histograms of the blurred images still be equal? Explain.



a b

3.21 The three images shown were blurred using square averaging masks of sizes $n = 23$, 25, and 45, respectively. The vertical bars on the left lower part of (a) and (c) are blurred, but a clear separation exists between them. However, the bars have merged in image (b), in spite of the fact that the mask that produced this image is significantly smaller than the mask that produced image (c). Explain the reason for this.



(a) (b) (c)

3.25 You saw in Fig. 3.38 that the Laplacian with a -8 in the center yields sharper results than the one with a -4 in the center. Explain the reason in detail.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

3.26 With reference to Problem 3.25,

(a) Would using a larger "Laplacian-like" mask, say, of size 5×5 with a -24 in the center, yield an even sharper result? Explain in detail.

Next Lecture

Frequency Domain

Assignment

- **Textbook Chapter 3:** 3, 4, 5, 6, 7
- **Check associated problems**

Chapter 3	14, 17, 18, 20, 21, 22, 23, 25, 26, 27, 30

References

- Gonzalez and Woods, Digital Image Processing.

- Peters, Richard Alan, II, “Spatial Filtering 1 and 2", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available on the web at the Internet Archive, http://www.archive.org/details/Lectures_on_Image_Processing.