Image Processing and Pattern Recognition (IPPR)

Lecture 5

Basanta Joshi, PhD

Asst. Prof., Depart of Electronics and Computer Engineering

Program Coordinator, MSc in Information and Communication Engineering

Member, Laboratory for ICT Research and Development (LICT)

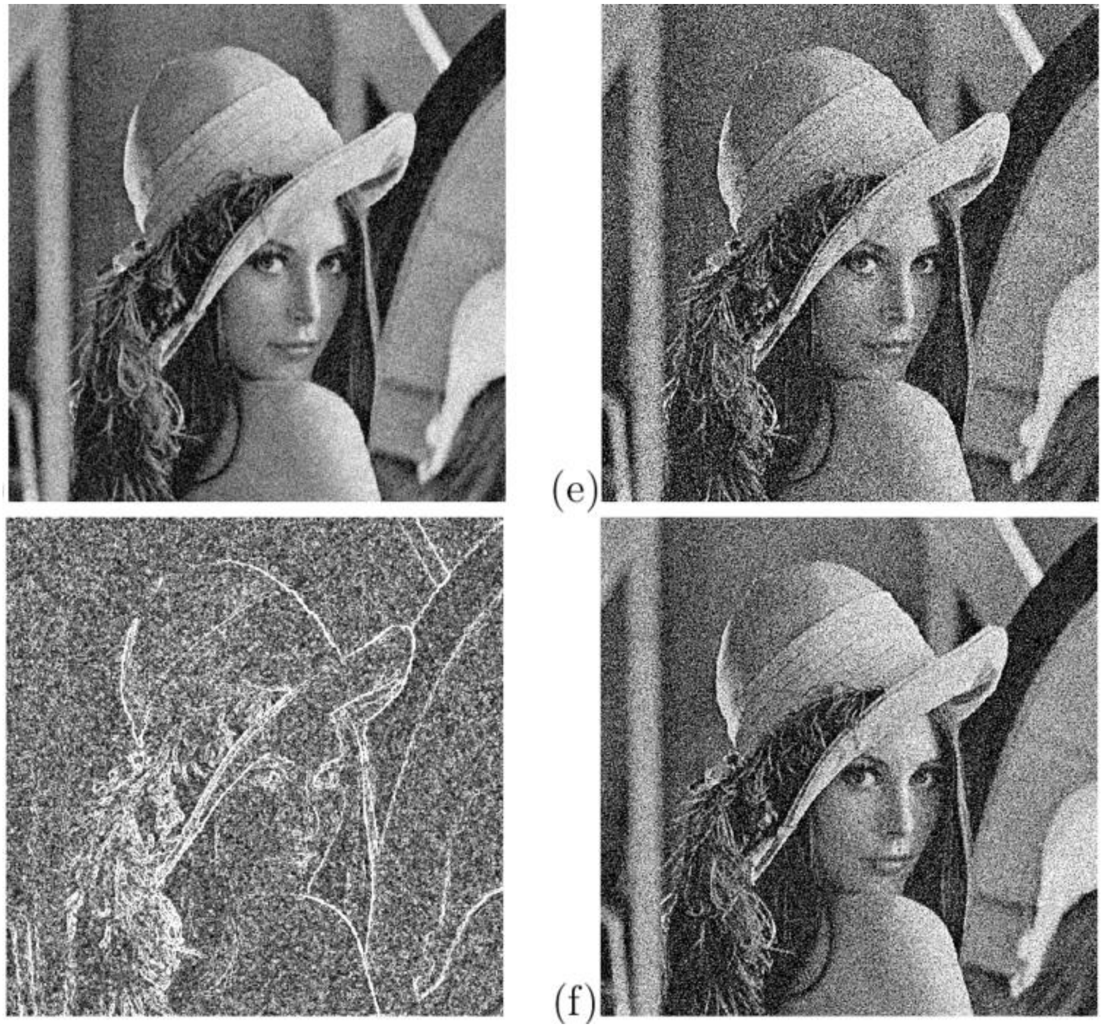
Member, Research Management Cell (RMC)

Institute of Engineering

basanta@ioe.edu.np

http://www.basantajoshi.com.np

https://scholar.google.com/citations?user=iocLiGcAAAAJ
https://www.researchgate.net/profile/Basanta_Joshi2

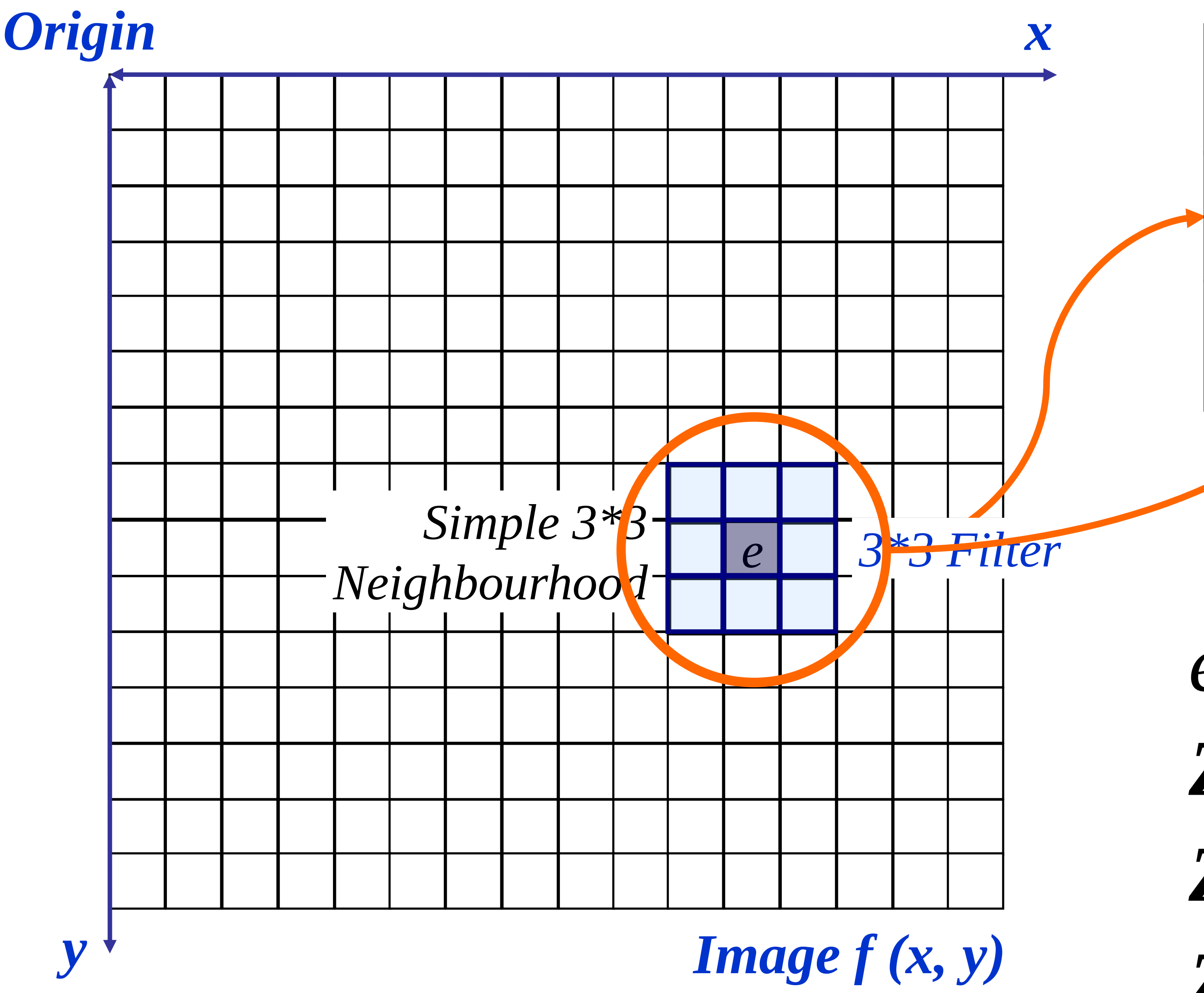


Contents

In this lecture we will look at more spatial filtering techniques

- Spatial filtering refresher
- Sharpening filters
  - 1st derivative filters
  - 2nd derivative filters
- Combining filtering techniques

The Spatial Filtering Refresher



Z_1	Z_2	Z_3
Z_4	Z_4	Z_5
Z_6	Z_7	Z_8

Original Image Pixels

*

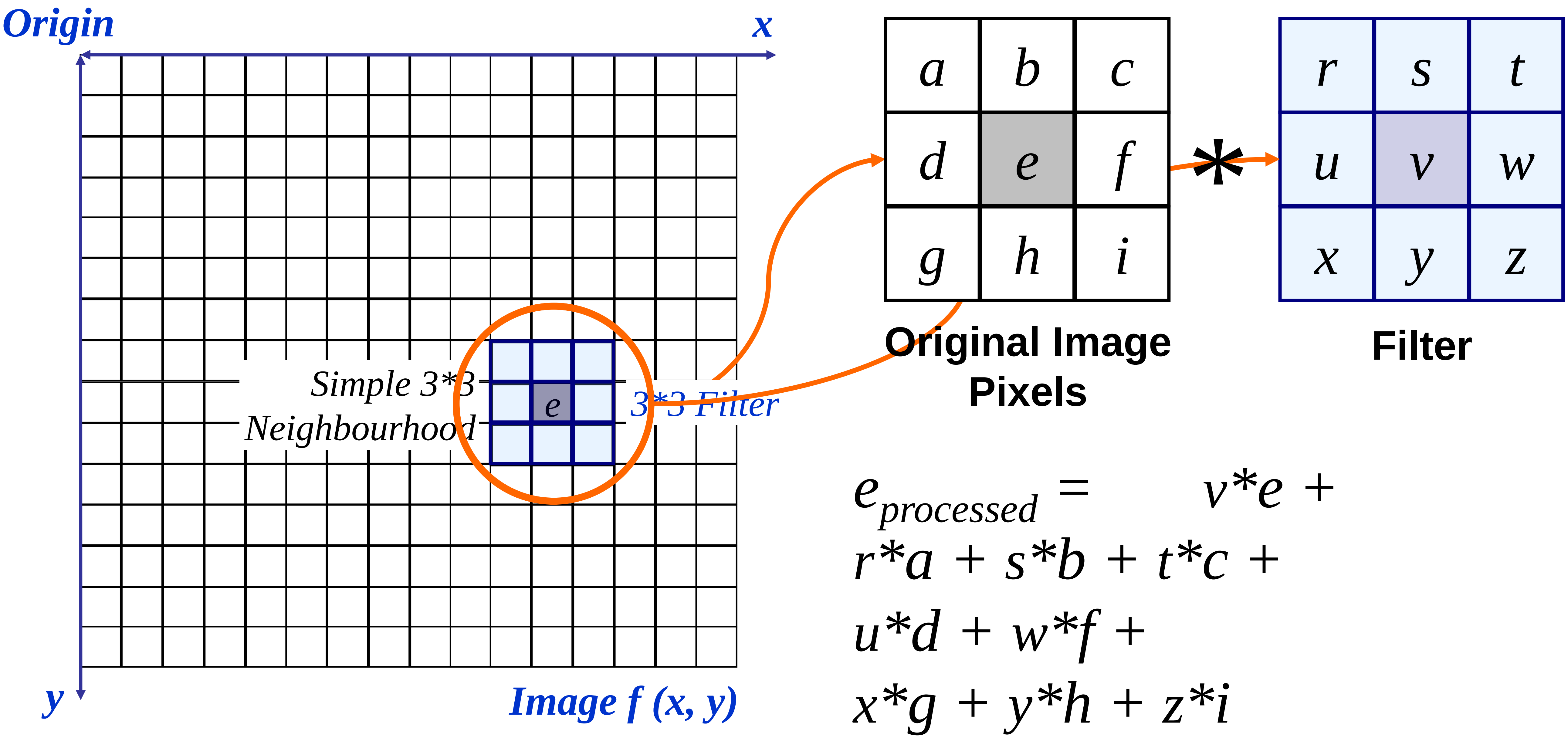
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Filter

$$e_{processed} = Z_1 * w_1 + Z_2 * w_{2+}\ Z_3 * w_{3+}\ Z_4 * w_4 + Z_5 * w_5 + Z_6 * w_6 + Z_7 * w_7 + Z_8 * w_8 + Z_9 * w_9$$

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

Spatial Filtering Refresher



$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

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

Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

Sharpening spatial filters seek to highlight fine detail

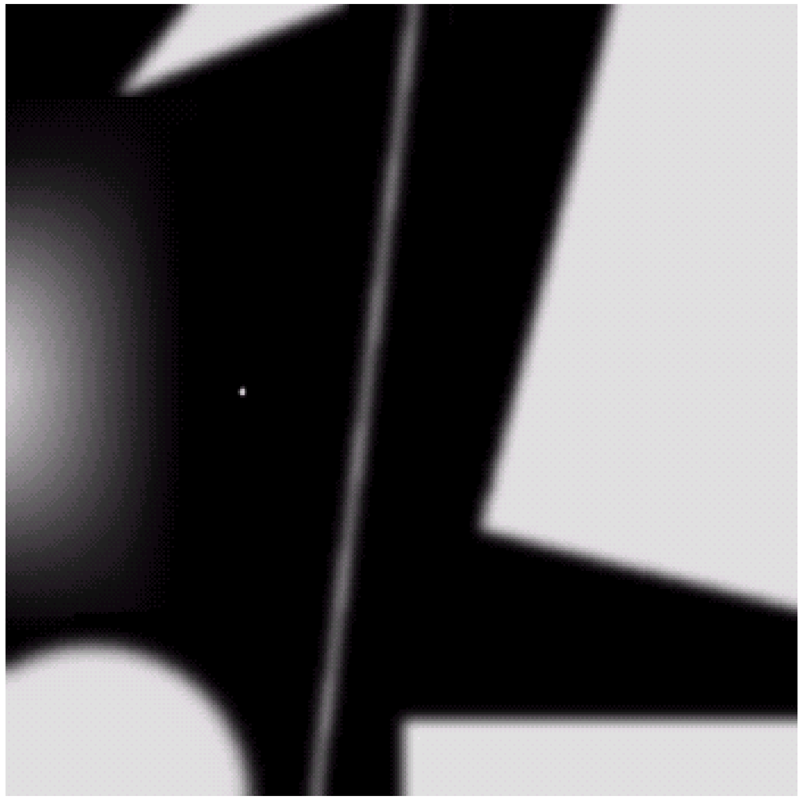
- Remove blurring from images
- Highlight edges

Sharpening filters are based on *spatial differentiation*

Spatial Differentiation

Differentiation measures the *rate of change* of a function

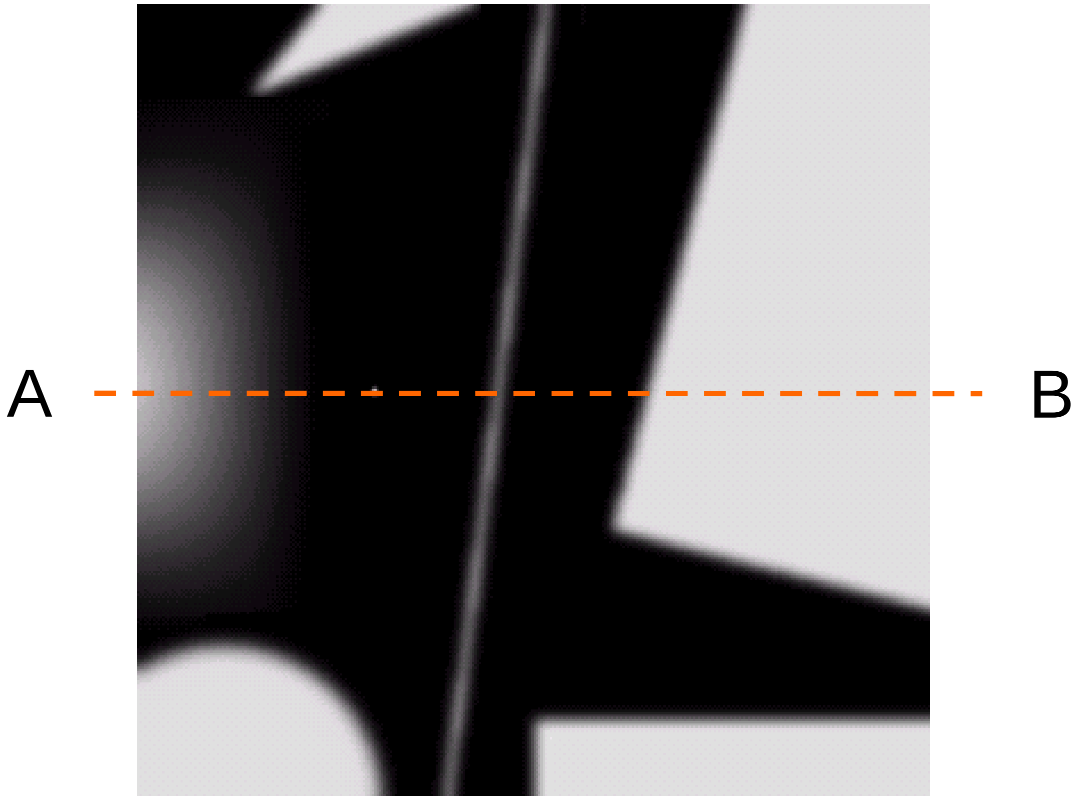
Let’s consider a simple 1 dimensional example



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Spatial Differentiation

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



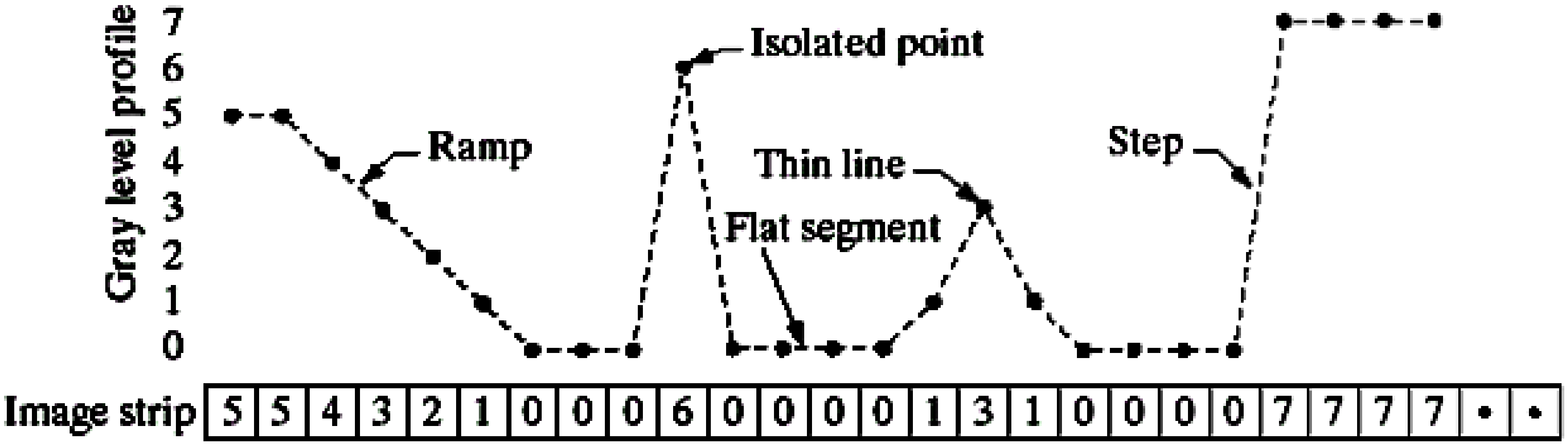


Image strip																										
	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•

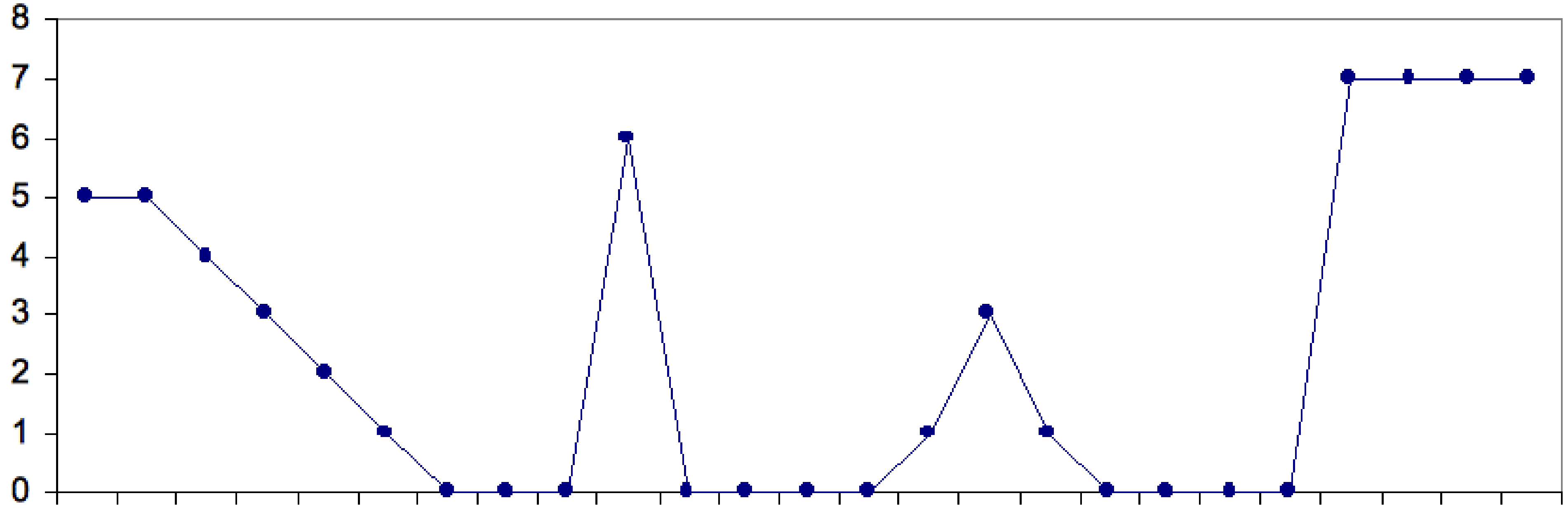
1st Derivative

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

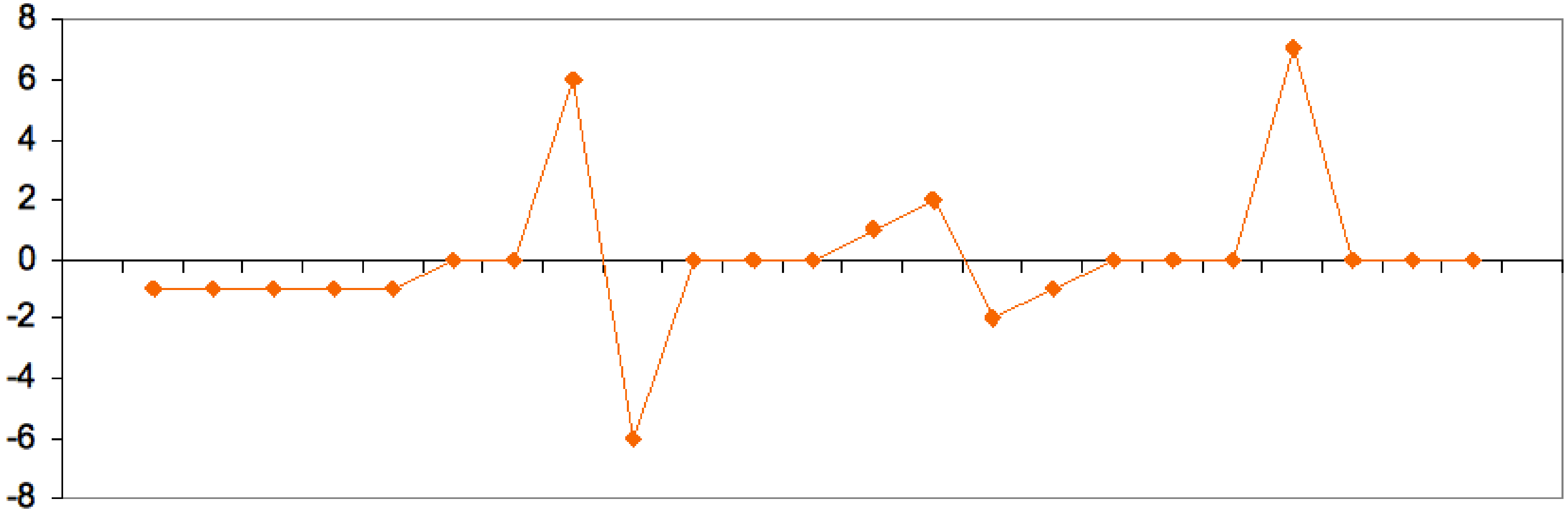
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

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

1st Derivative (cont…)



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	



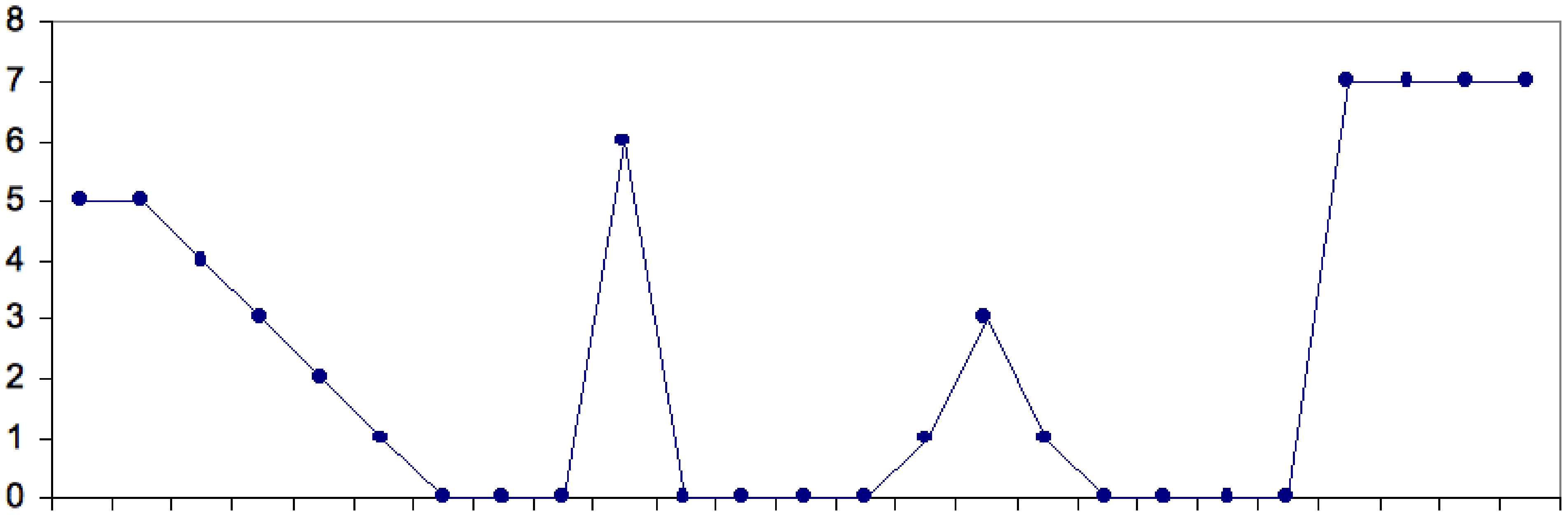
2nd Derivative

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

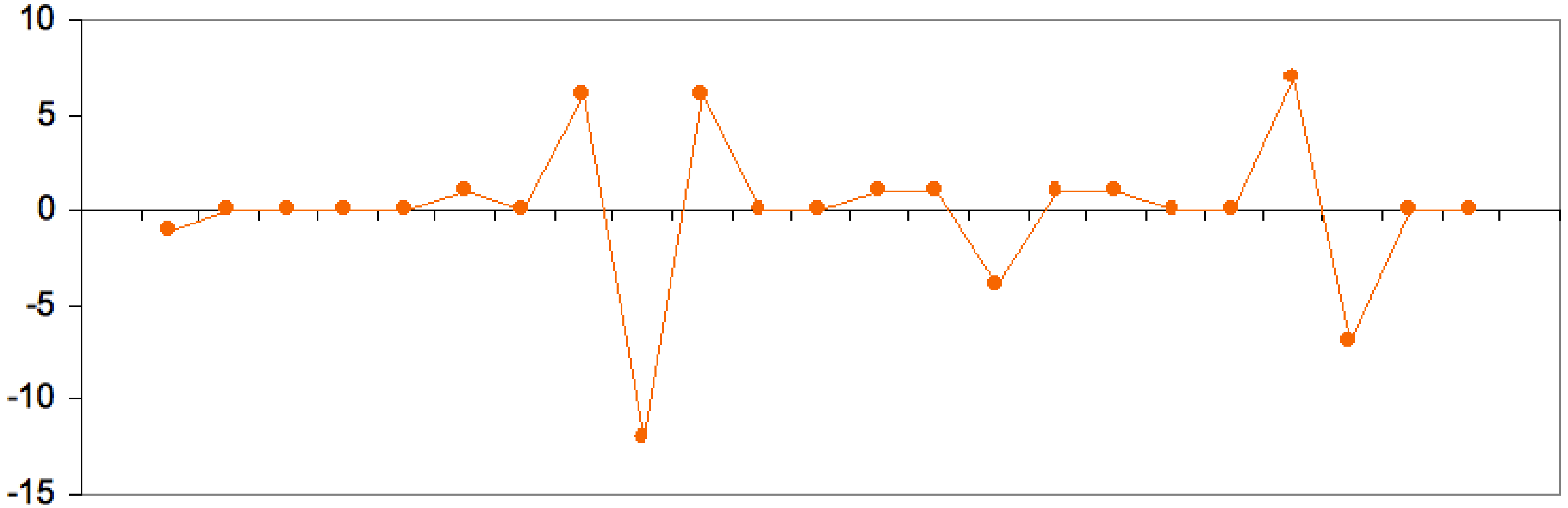
$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x$$

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

2nd Derivative (cont…)



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	



Using Second Derivatives For Image Enhancement

The 2nd derivative is more useful for image enhancement than the 1st derivative

- Stronger response to fine detail
- Simpler implementation
- We will come back to the 1st order derivative later on

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

- Isotropic
- One of the simplest sharpening filters
- We will look at a digital implementation

The 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 the partial 1st order derivative in the *x* direction is defined as follows:

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

and in the *y* direction as follows:

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

The Laplacian (cont…)

So, the Laplacian can be given as follows:

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

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

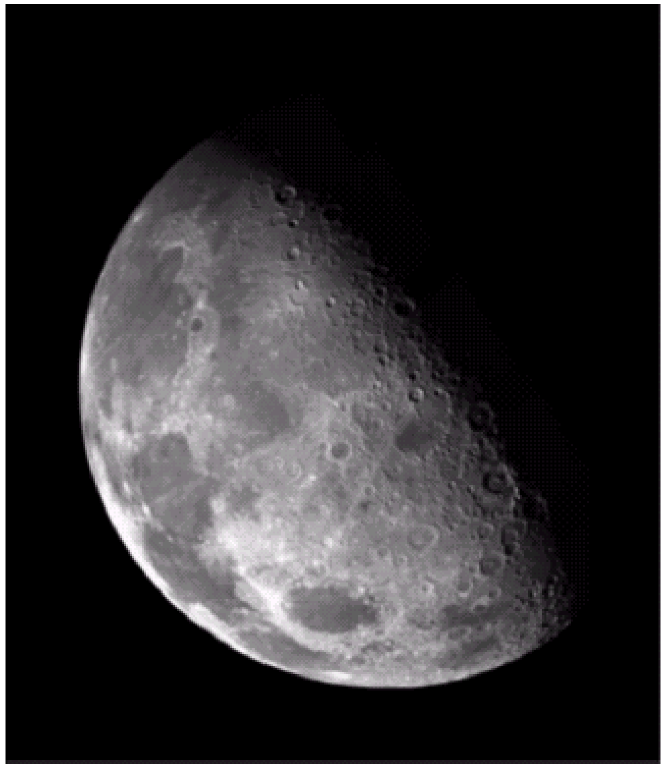
$$-4f(x,y)$$

We can easily build a filter based on this

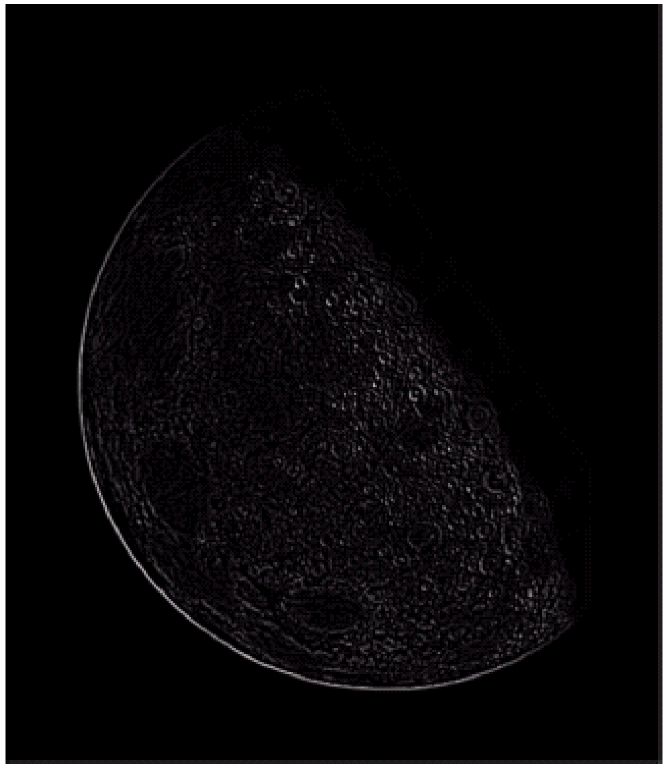
0	1	0
1	-4	1
0	1	0

The Laplacian (cont…)

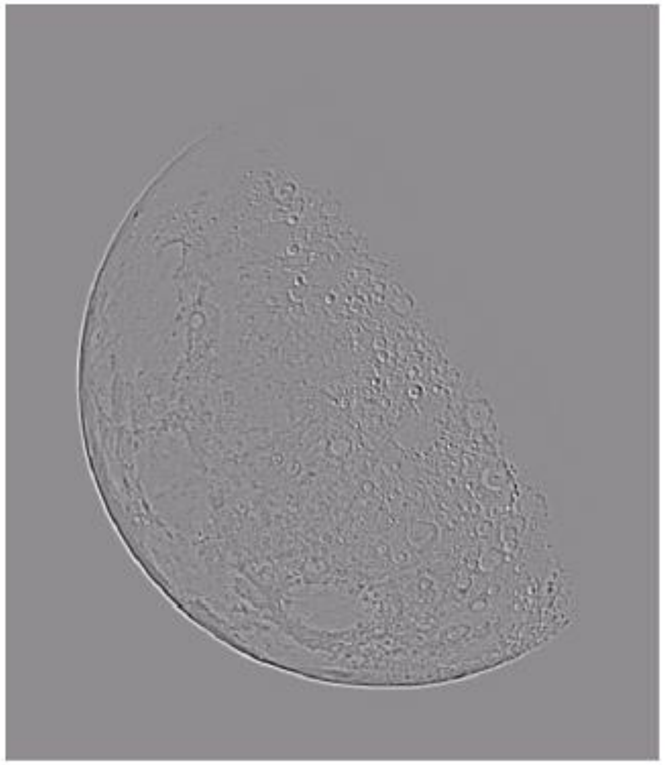
Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities



Original Image



Laplacian Filtered Image



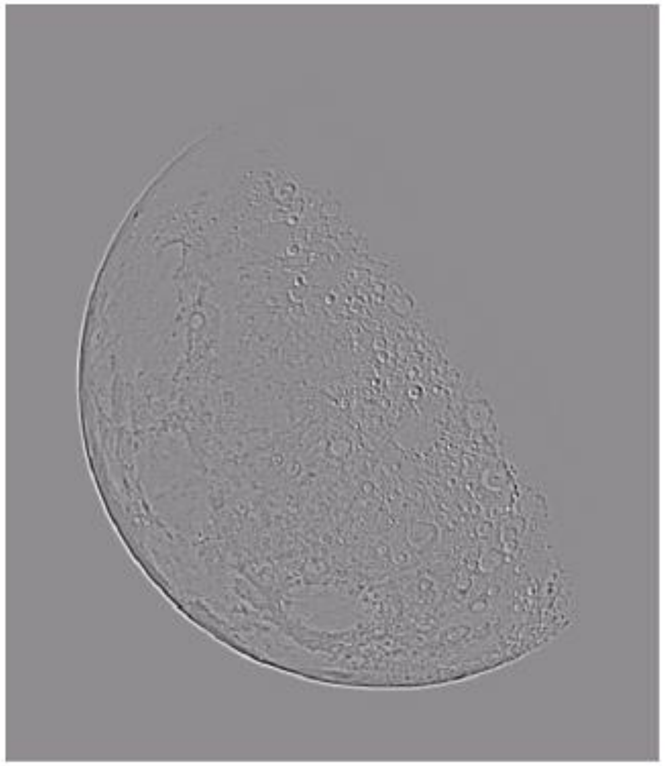
Laplacian Filtered Image Scaled for Display

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

But That Is Not Very Enhanced!

- The result of a Laplacian filtering is not an enhanced image
- We have to do more work in order to get our final image
- Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

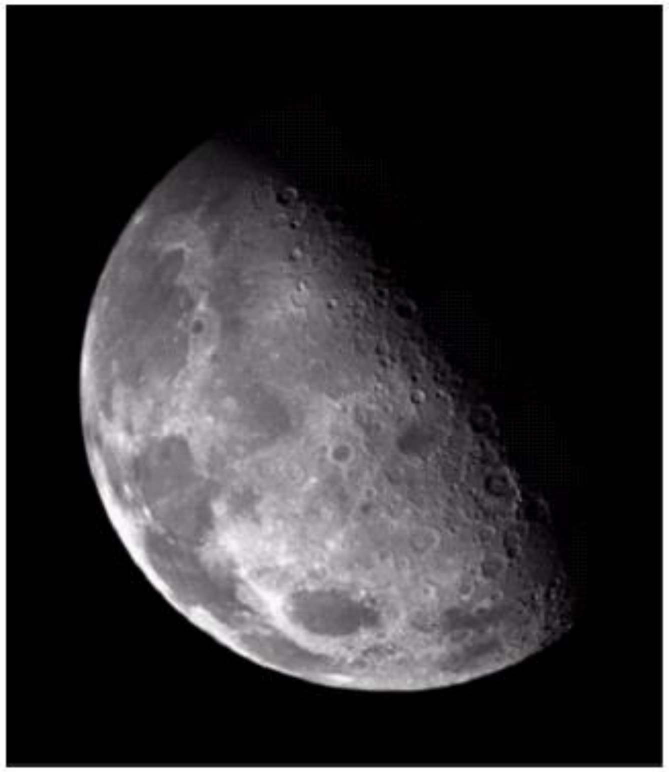
$$g(x, y) = f(x, y) - \nabla^2 f$$



Laplacian
Filtered Image
Scaled for Display

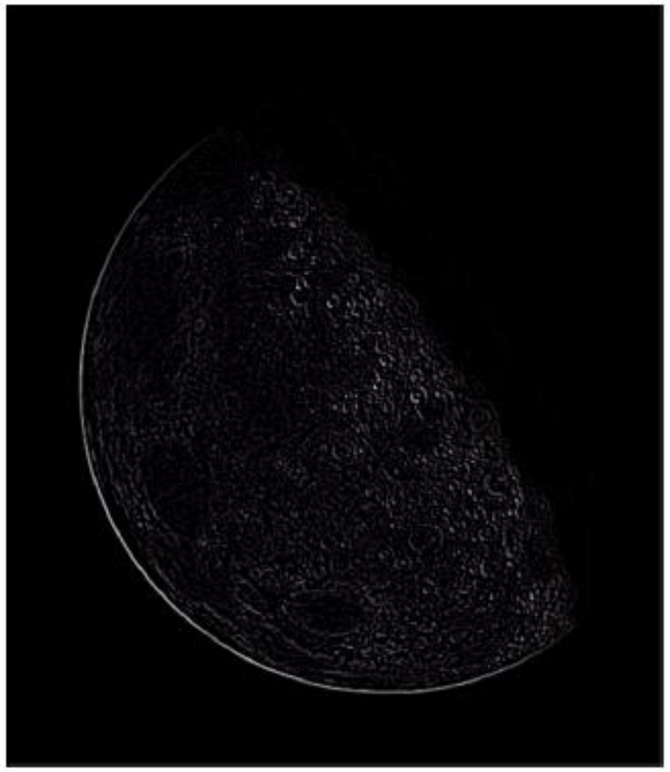
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Laplacian Image Enhancement



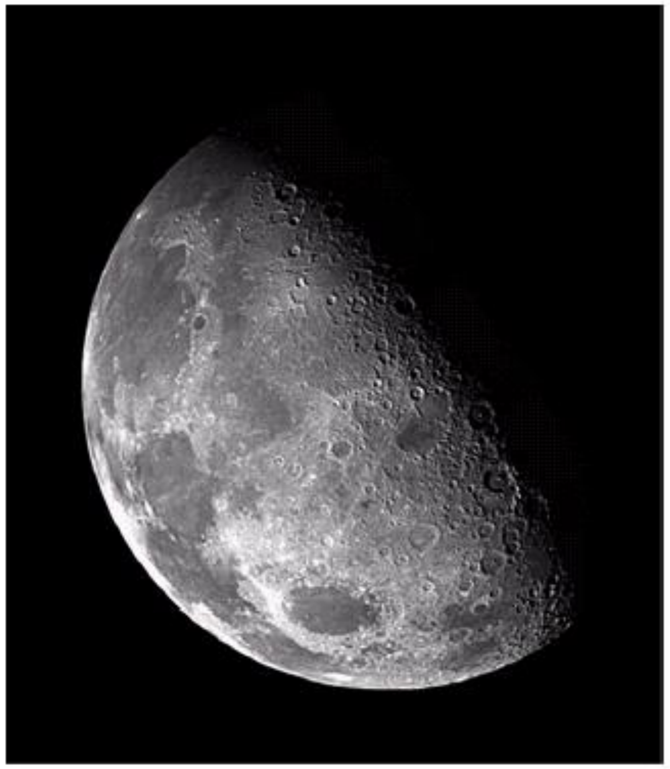
Original Image

-



Laplacian Filtered Image

=

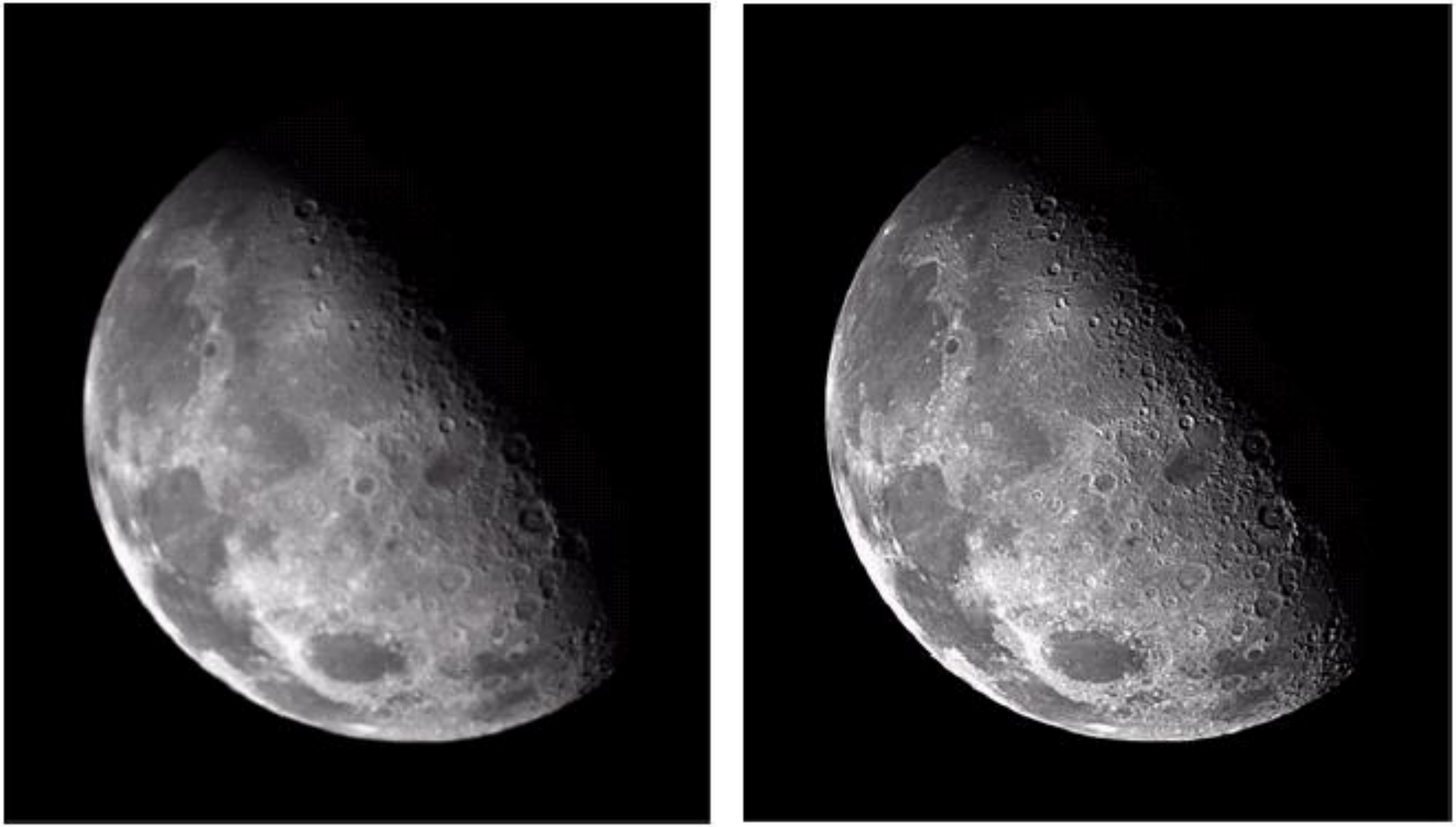


Sharpened Image

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

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Laplacian Image Enhancement



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simplified Image Enhancement

The entire enhancement can be combined into a single filtering operation

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

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

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

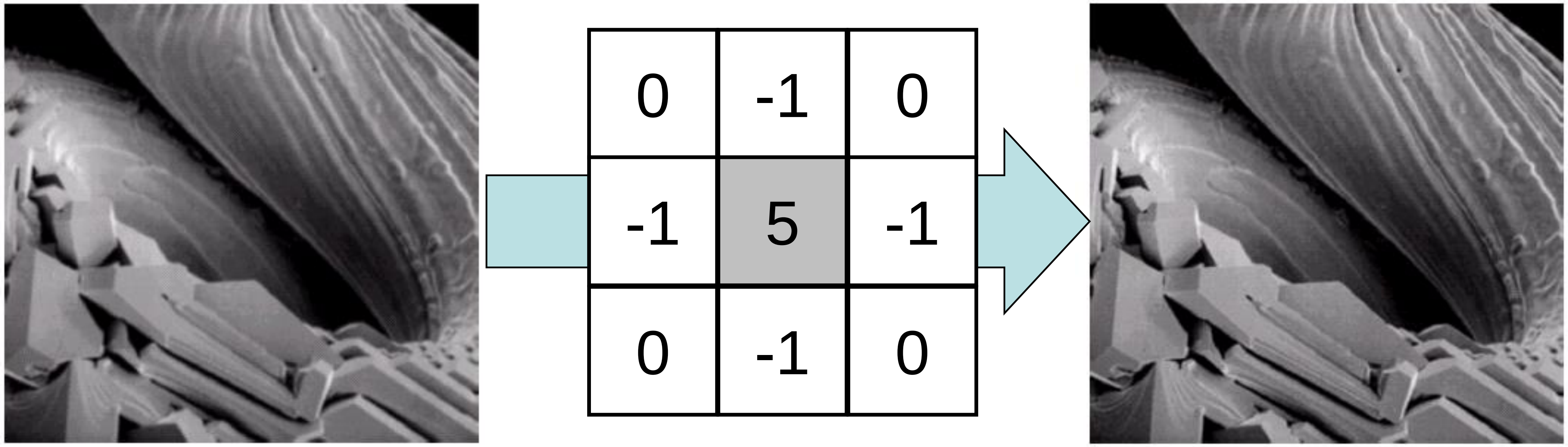
$$-4f(x,y)]$$

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

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

Simplified Image Enhancement (cont…)

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

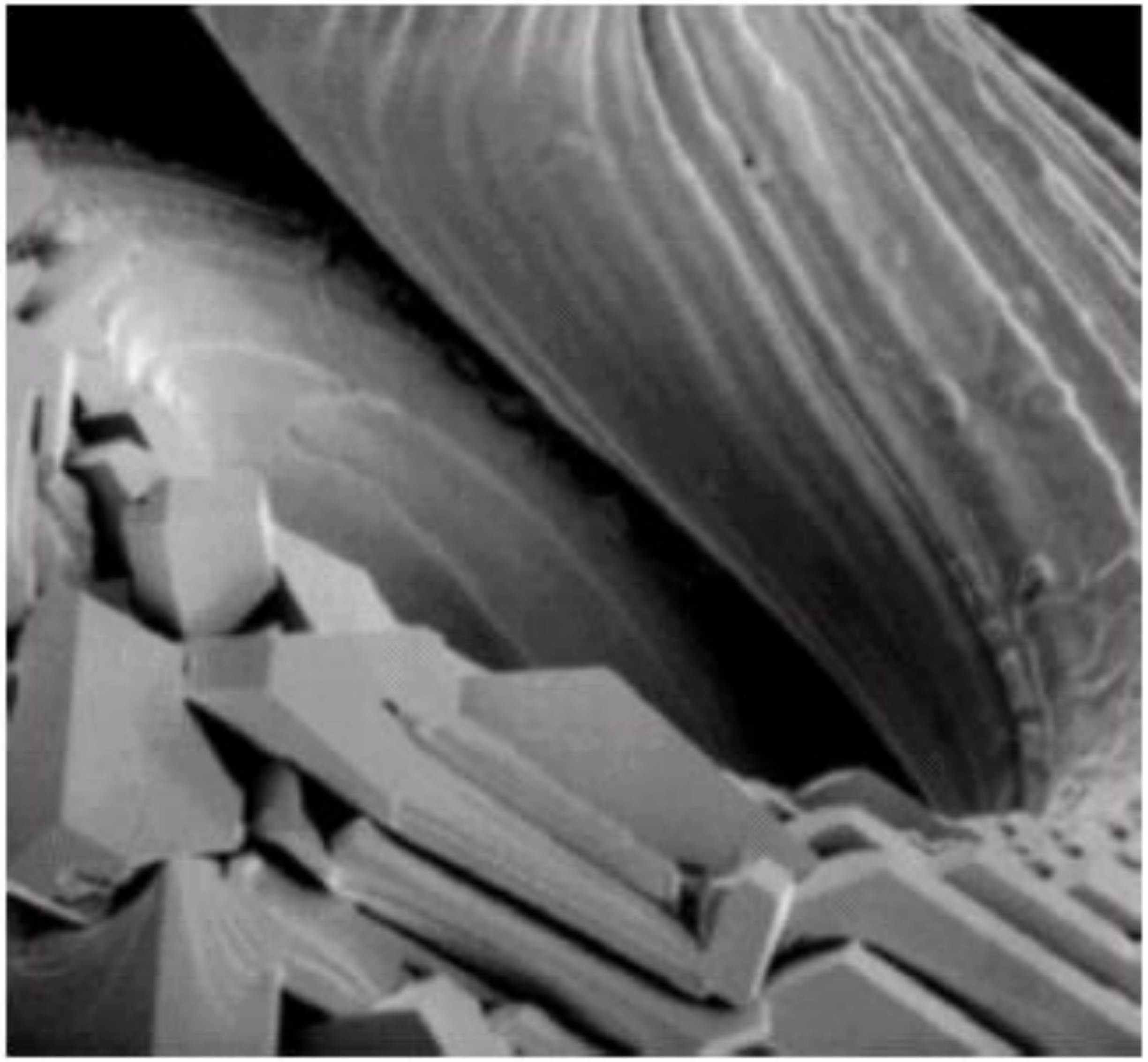


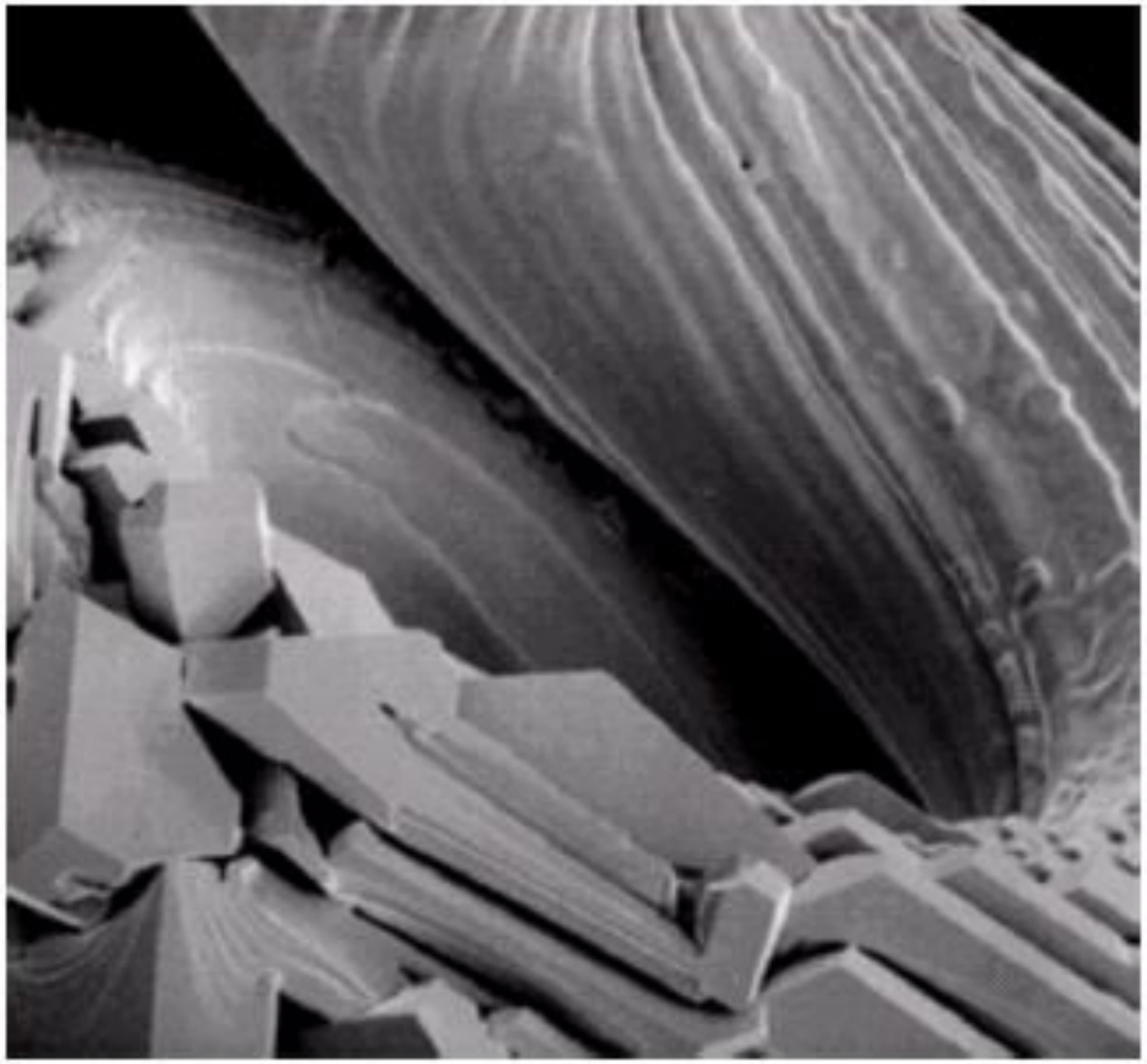
0	-1	0
-1	5	-1
0	-1	0

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simplified Image Enhancement (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)





Variants On The Simple Laplacian

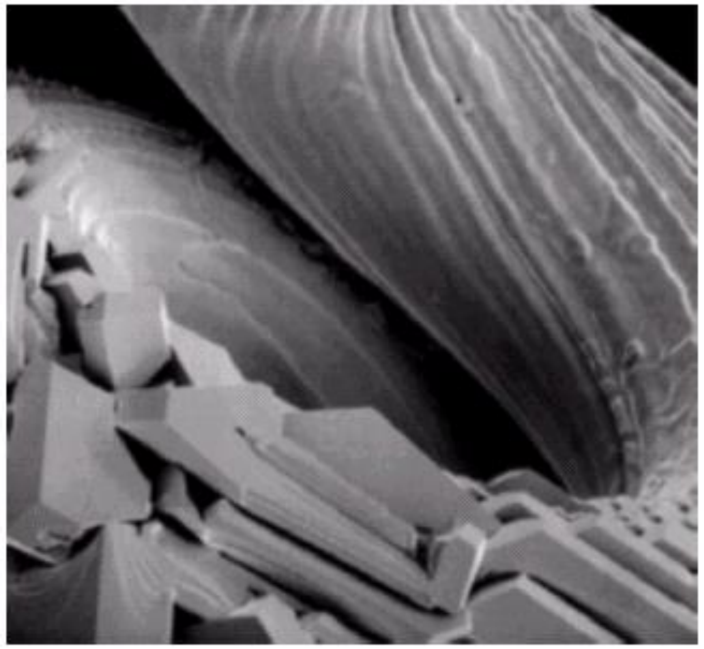
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

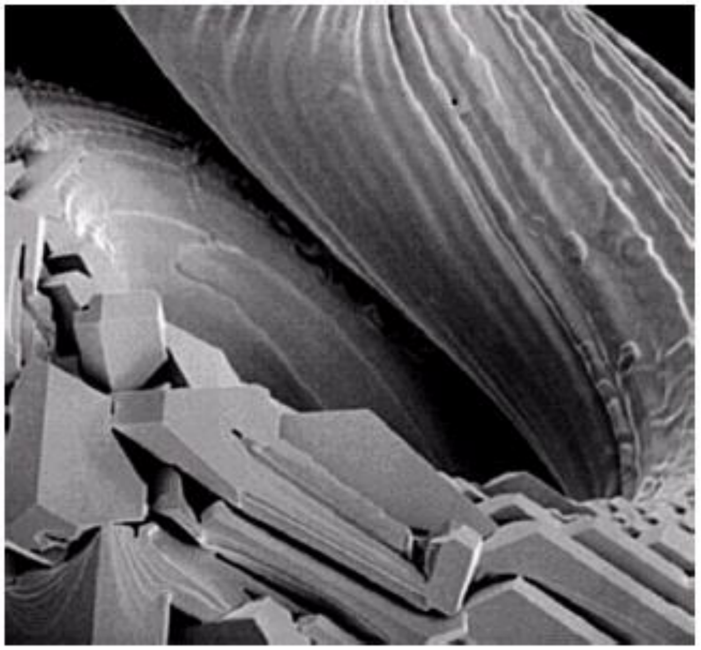
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



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



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

1st Derivative Filtering

Implementing 1st derivative filters is difficult in practice

For a function *f(x, y)* the gradient of *f* at coordinates *(x, y)* is given as the column vector:

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

1st Derivative Filtering (cont…)

The magnitude of this vector is given by:

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

For practical reasons this can be simplified as:

$$\nabla f \approx \left|G_x\right| + \left|G_y\right|$$

1st Derivative Filtering (cont…)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- The magnitude of the gradient at z_5 can be approximated in a number of ways
- The simplest is to use the difference $(z_5 - z_8)$ in the x direction and $(z_5 - z_6)$ in the y direction

$$\nabla f = [(z_5 - z_8)^2 + (z_5 - z_6)^2]^{1/2}$$

- or approximated as

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6|$$

1st Derivative Filtering (cont…)

- A cross difference may also be used to approximate the magnitude of the gradient

$$\nabla f = [(z_5 - z_9)^2 + (z_6 - z_8)^2]^{1/2}$$

$$\nabla f \approx | z_5 - z_9 | + | z_6 - z_8 |$$

- This can be implemented by taking the absolute value of the response of the following two masks (the *Roberts cross-gradient operators*) and summing the results

1	0
0	-1

0	1
-1	0

1st Derivative Filtering (cont…)

- Extension to a 3x3 mask yields the following

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

- The difference between the first and third rows approximates the derivative in the *x* direction
- The difference between the first and third columns approximates the derivative in the *y* direction
- The *Prewitt operator* masks may be used to implement the above approximation

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

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

1^{st} Derivative Filtering (cont…)

- The *Sobel operator* masks may also be used to implement the derivative approximation

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

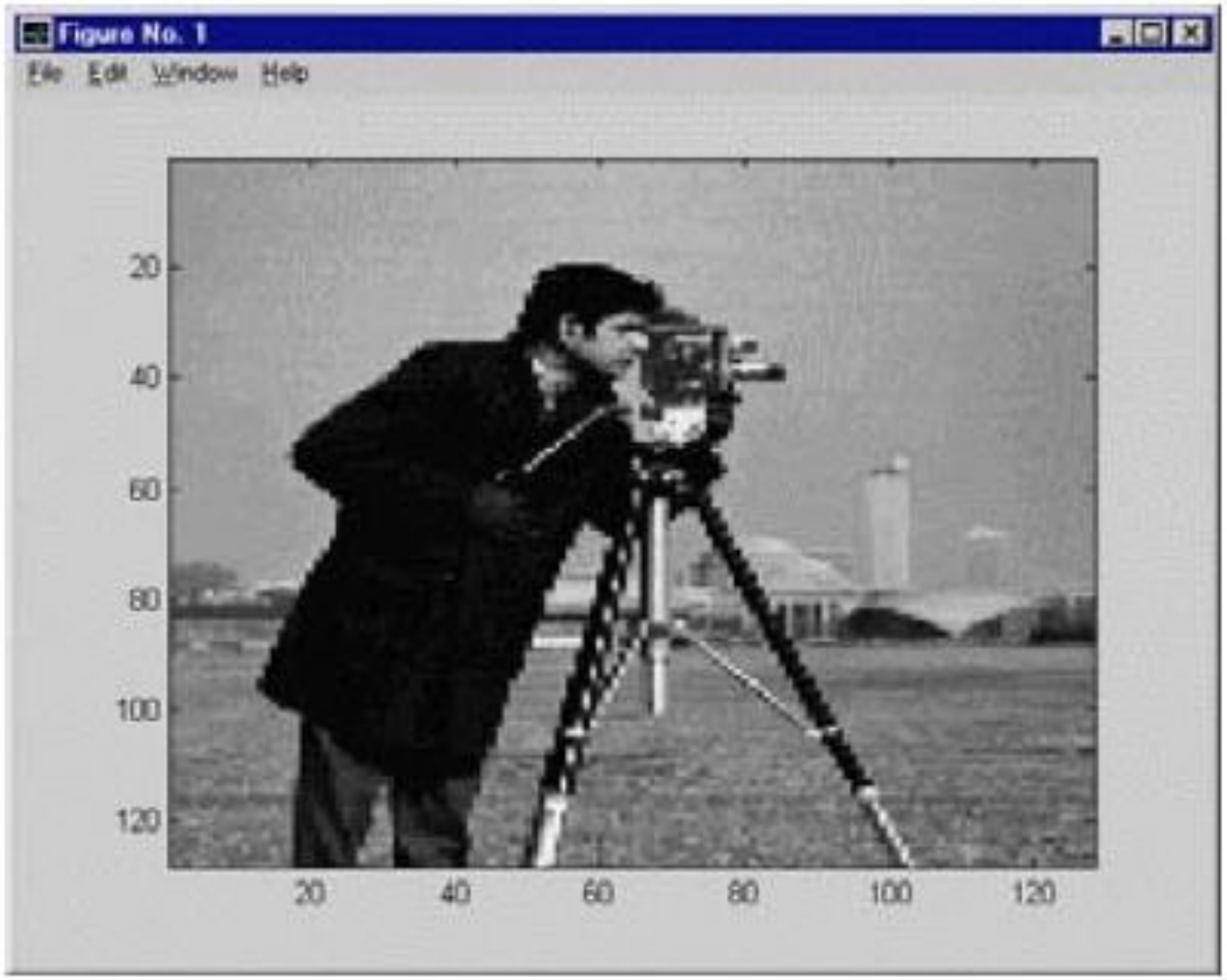
-1	0	1
-2	0	2
-1	0	1

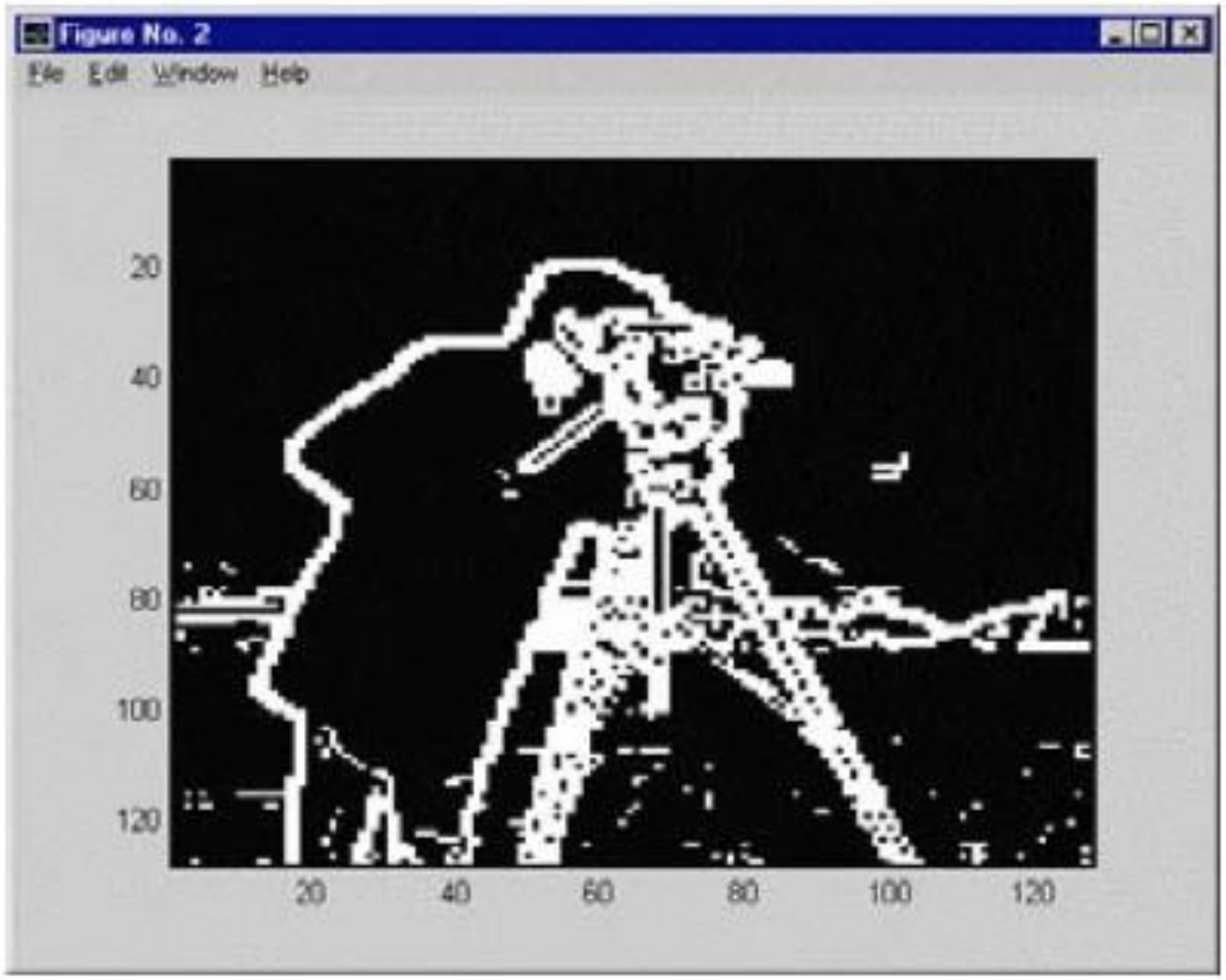
- The *Sobel operators* are widely used for edge detection (will be discussed in more detail later)
- NOTE: All the mask coefficients for all the derivative filters sum to zero, indicating a 0 response in a constant area (as expected of a derivative operator)

1^{st} Derivative Filtering (cont…)

- After application of a derivative filter, generally the output will be scaled (as in the lowpass and highpass cases)
- The result is then thresholded by setting any values above the threshold to white (256 in MATLAB with a 256 gray-level colormap) and all below the threshold are set to black (1 in MATLAB) or left with their initial values in *f(x,y)*
- The derivative filters may by applied:
  - Only in the *x* direction
  - Only in the *y* direction
  - In both directions (taking either the sum or maximum of the responses from the filter masks)

1^{st} Derivative Filtering (cont…)



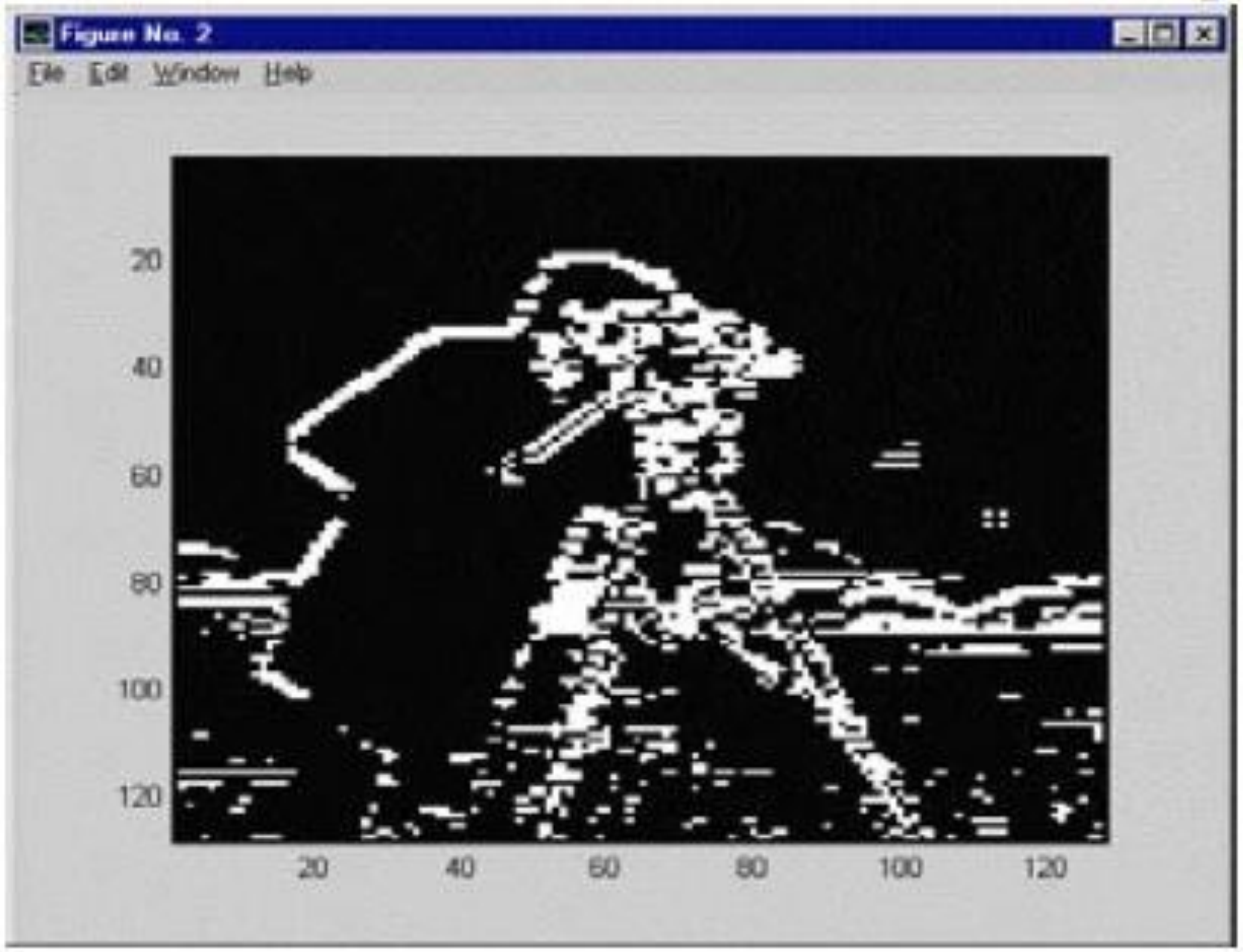


- Original image

- Sobel filtered image

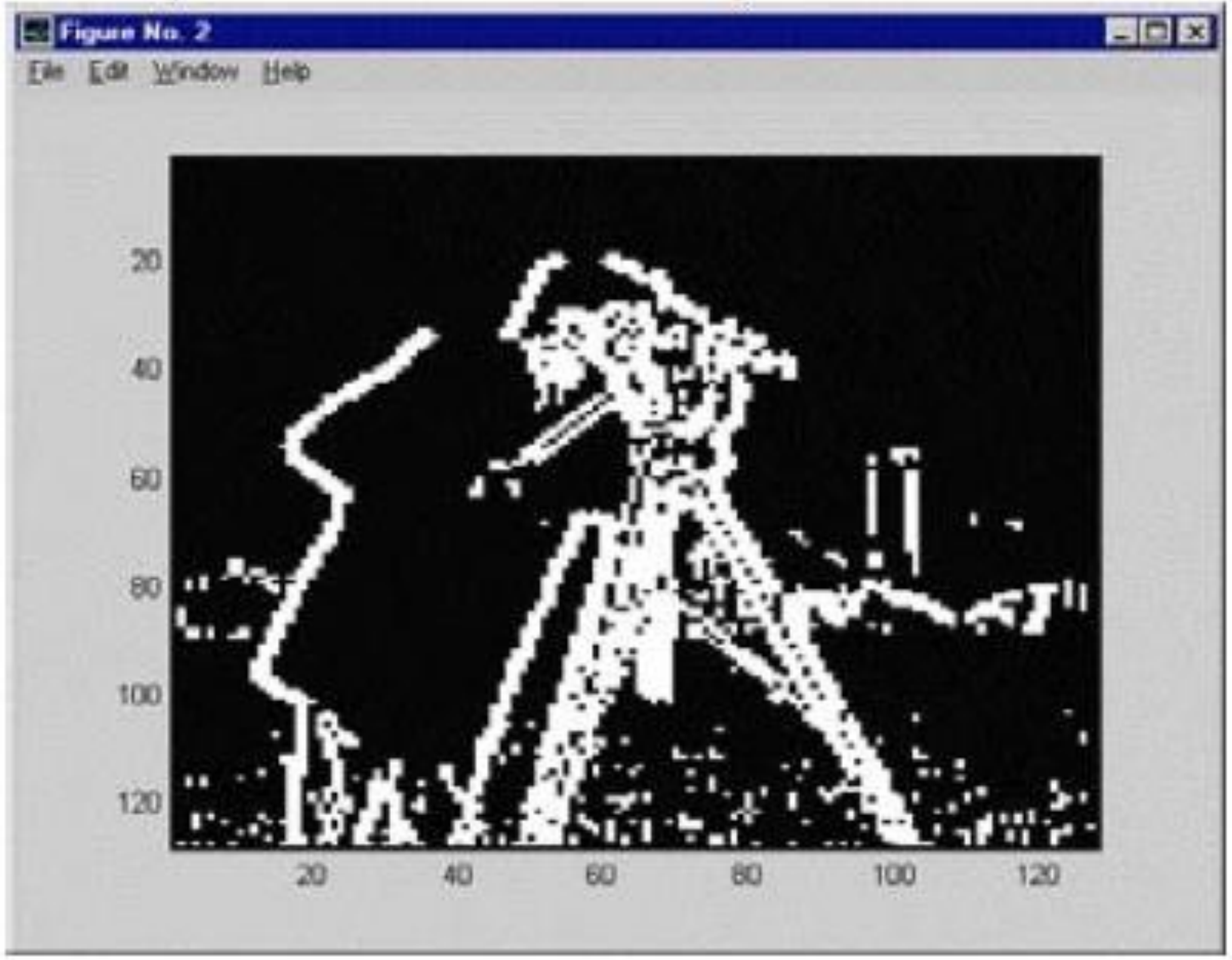
```
g=derivativefilter(f,'Sobel',25,'sum',0);
```

1st Derivative Filtering (cont…)



- Sobel filtered image

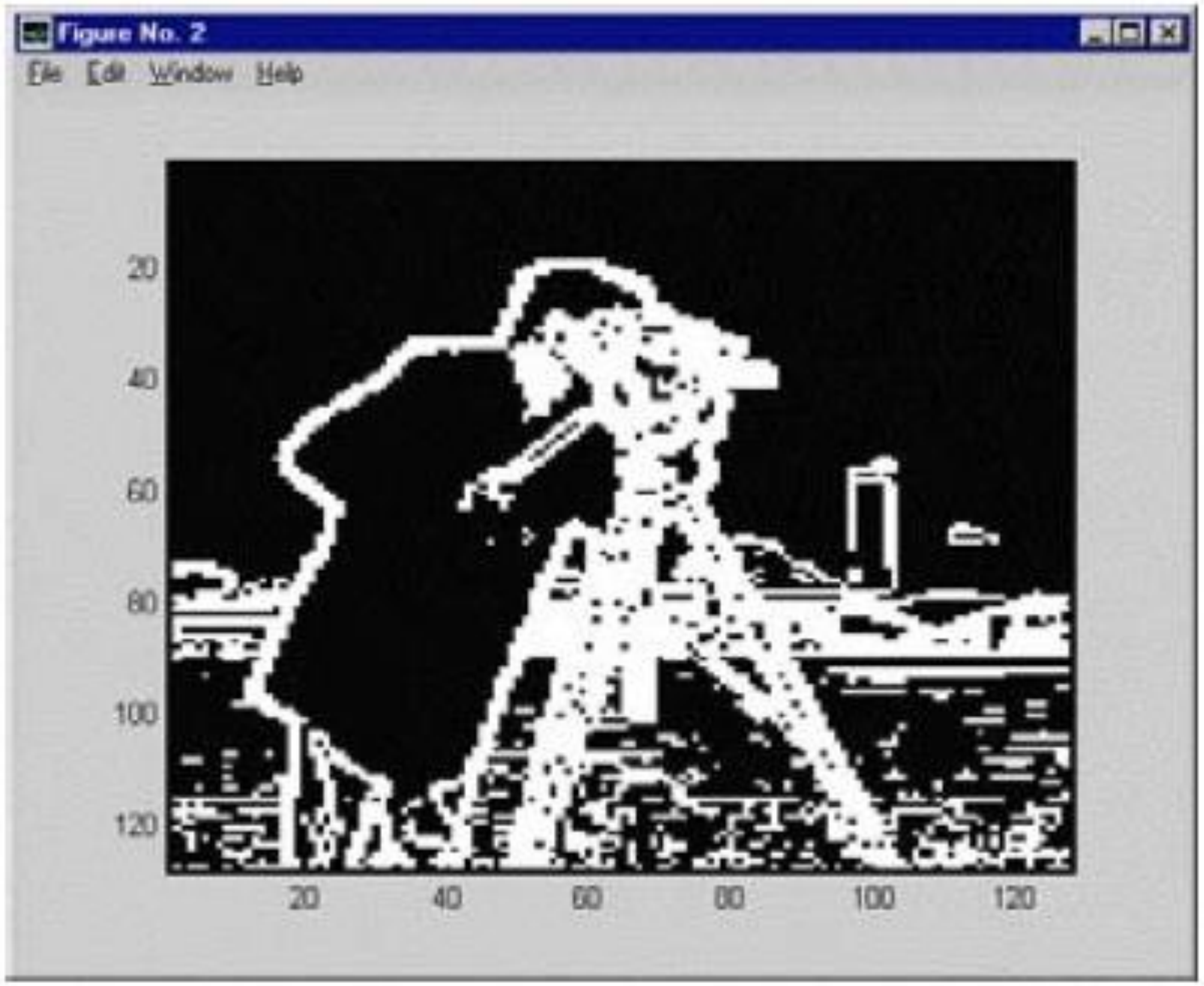
g=derivativefilter(f,'Sobel',15, 'horiz',0);



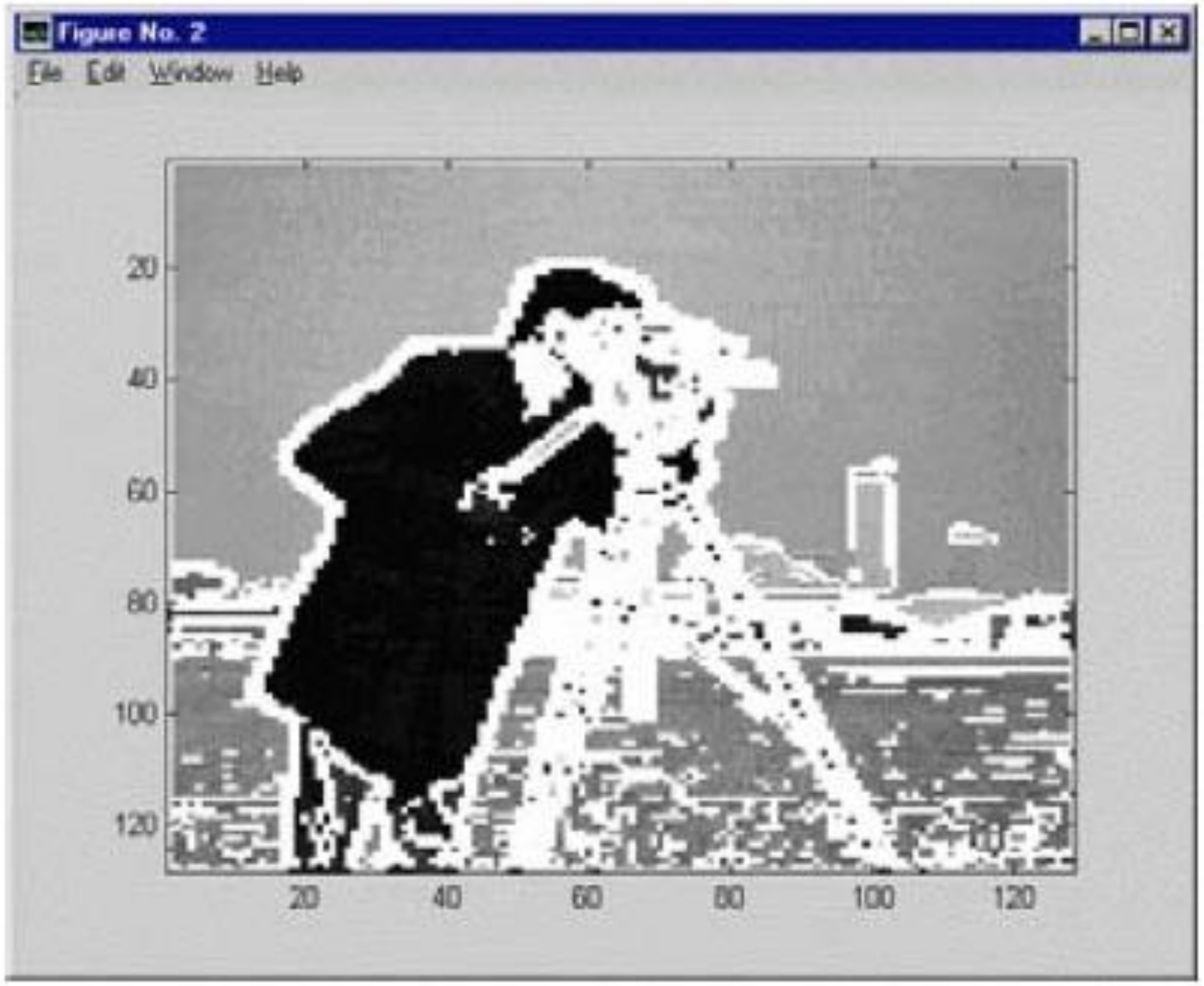
- Sobel filtered image

g=derivativefilter(f,'Sobel',10, 'vert',0);

1st Derivative Filtering (cont…)



`g=derivativefilter(f,'Sobel',10,'max',0);`



`g=derivativefilter(f,'Sobel',10,'max',1);`

1st Derivative Filtering (cont…)

There is some debate as to how best to calculate these gradients but we will use:

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

which is based on these coordinates

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel Operators

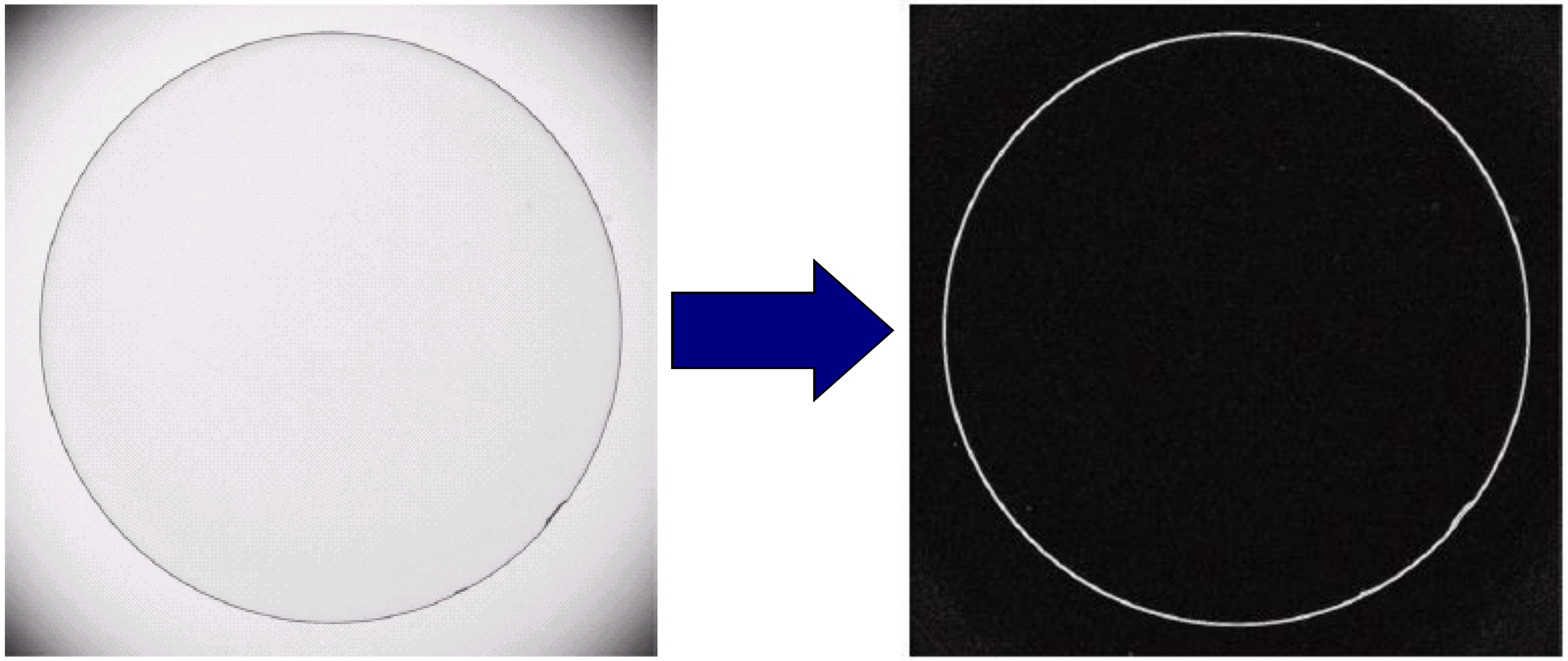
Based on the previous equations we can derive the *Sobel Operators*

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

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

To filter an image it is filtered using both operators the results of which are added together

Sobel Example



An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious

Sobel filters are typically used for edge detection

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

1st & 2nd Derivatives

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

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

Summary

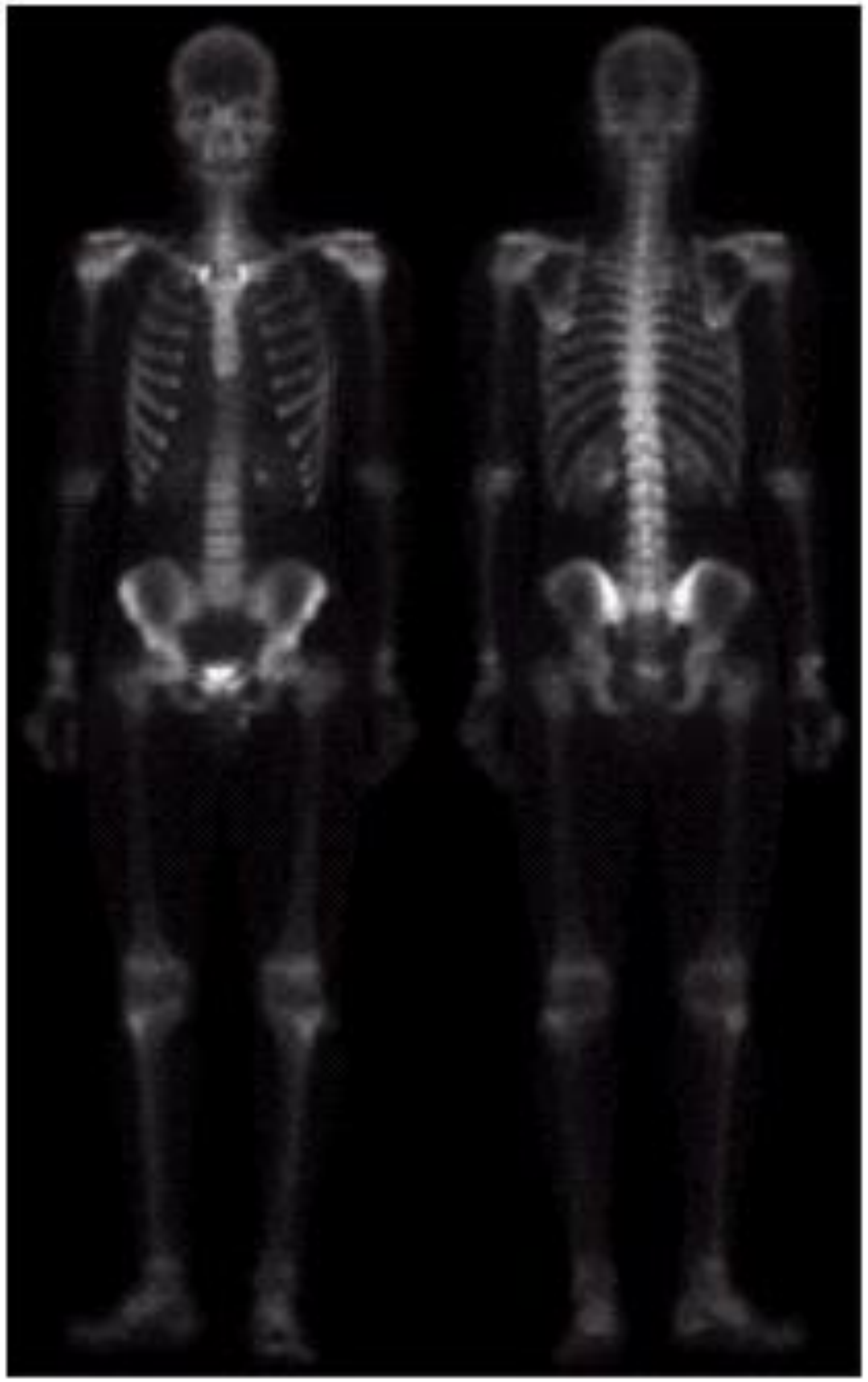
In this lecture we looked at:

- Sharpening filters
  - 1st derivative filters
  - 2nd derivative filters
- Combining filtering techniques

Combining Spatial Enhancement Methods

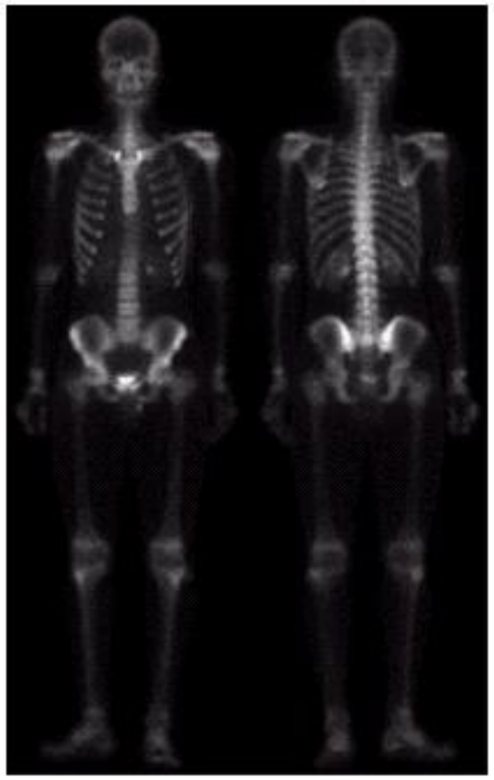
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

- 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 to the right



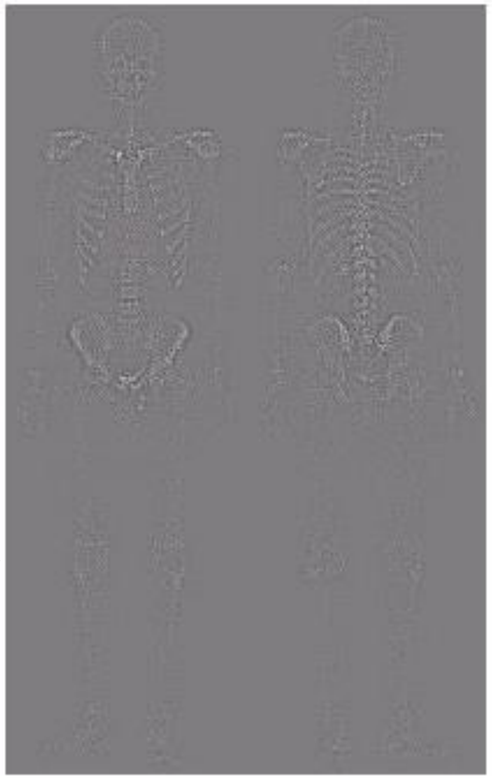
Combining Spatial Enhancement Methods (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



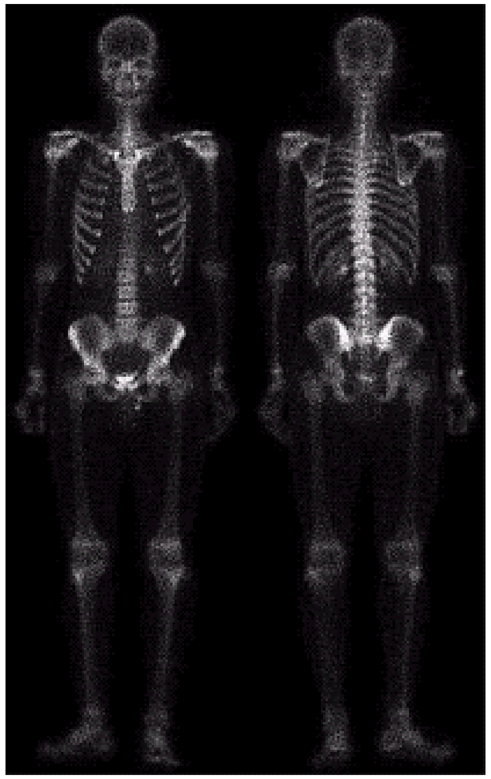
(a)

Laplacian filter of bone scan (a)



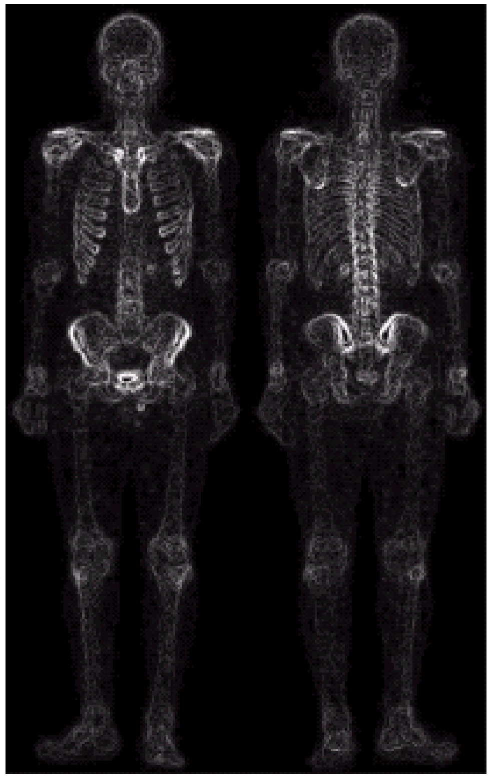
(b)

Sharpened version of bone scan achieved by subtracting (a) and (b)



(c)

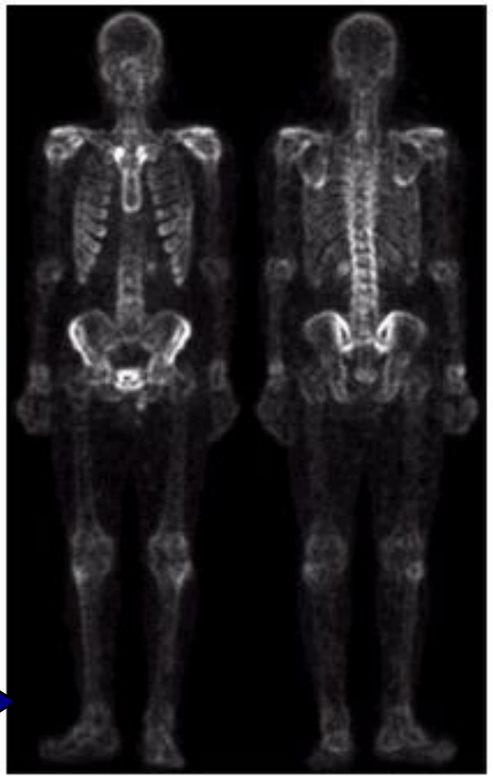
Sobel filter of bone scan (a)



(d)

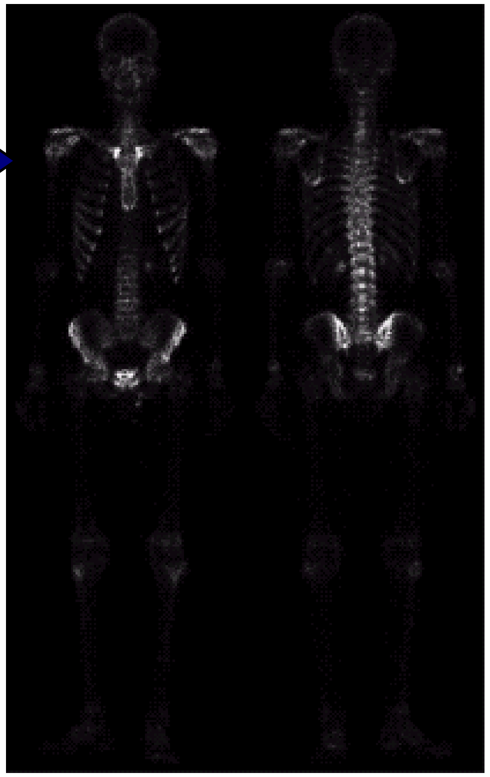
Combining Spatial Enhancement Methods (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



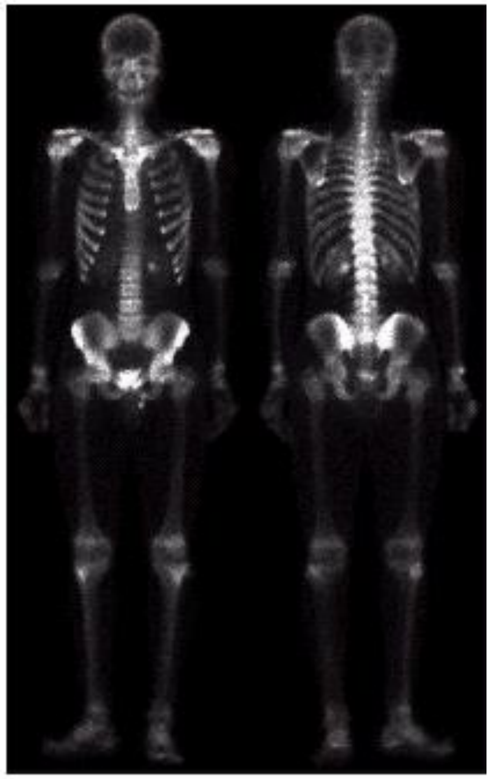
(e)

Image (d) smoothed with a 5*5 averaging filter



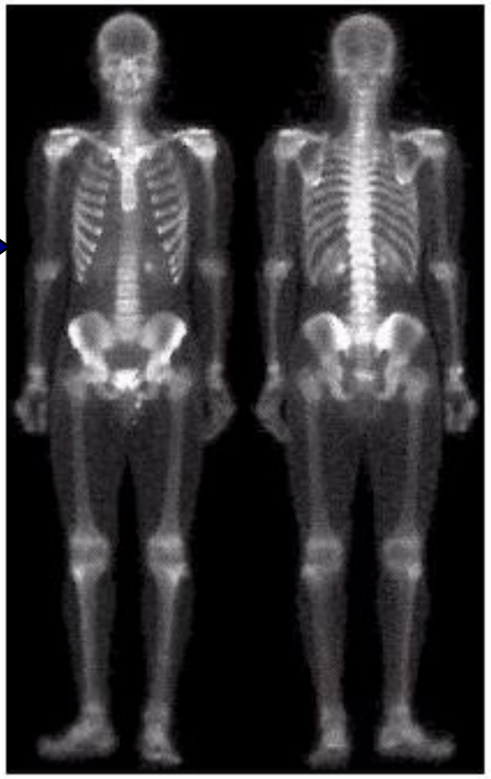
(f)

The product of (c) and (e) which will be used as a mask



(g)

Sharpened image which is sum of (a) and (f)

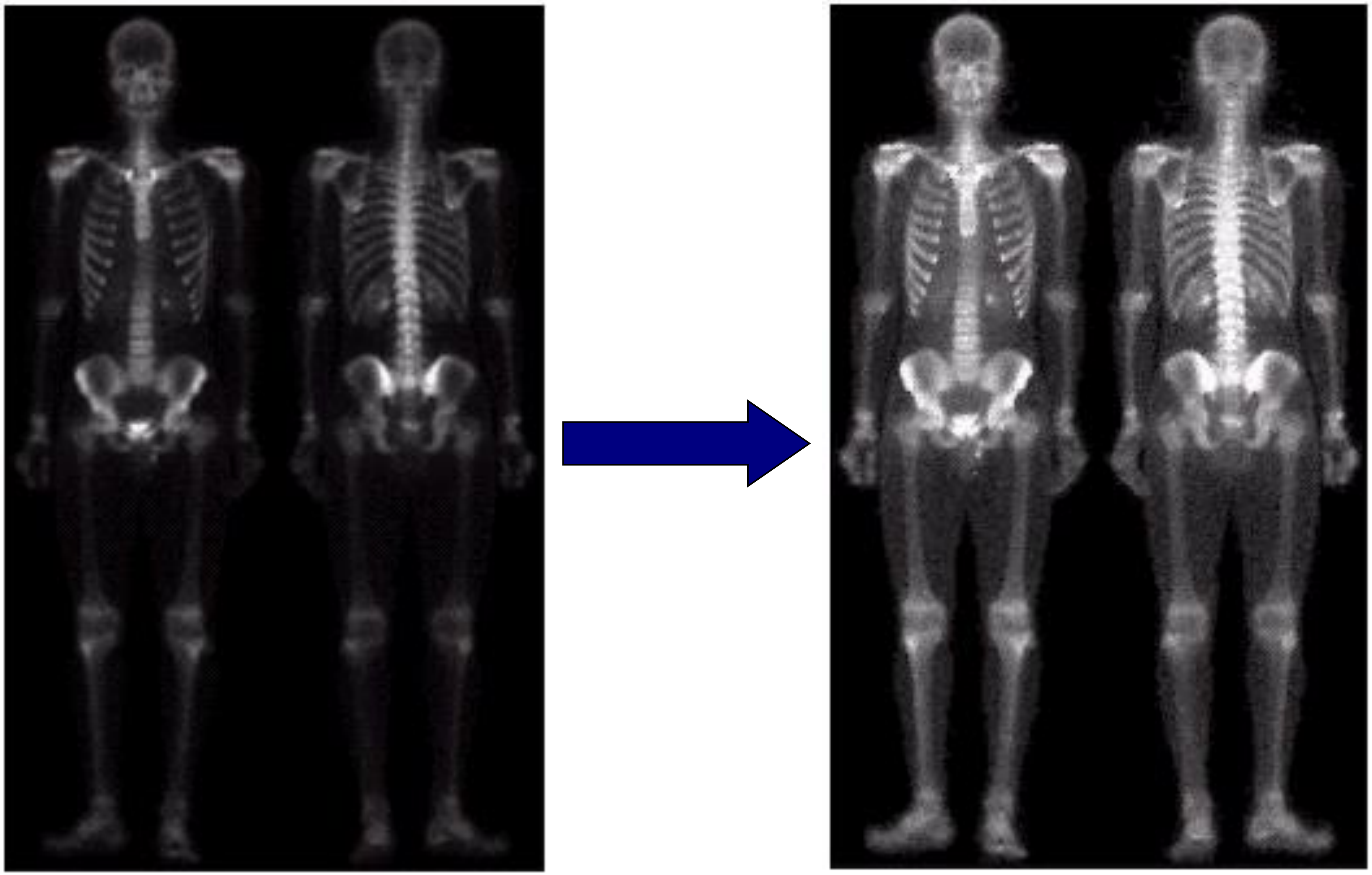


(h)

Result of applying a power-law trans. to (g)

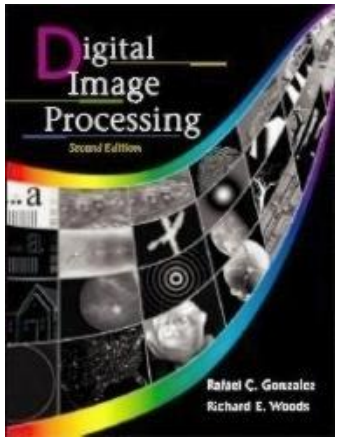
Combining Spatial Enhancement Methods (cont…)

Compare the original and final images



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

References



“Digital Image Processing”, Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

- –Much of the material that follows is taken from this book

–Image Processing and Pattern Recognition Slides of Dr. Sanjeeb Prasad Panday

Thank you !!!