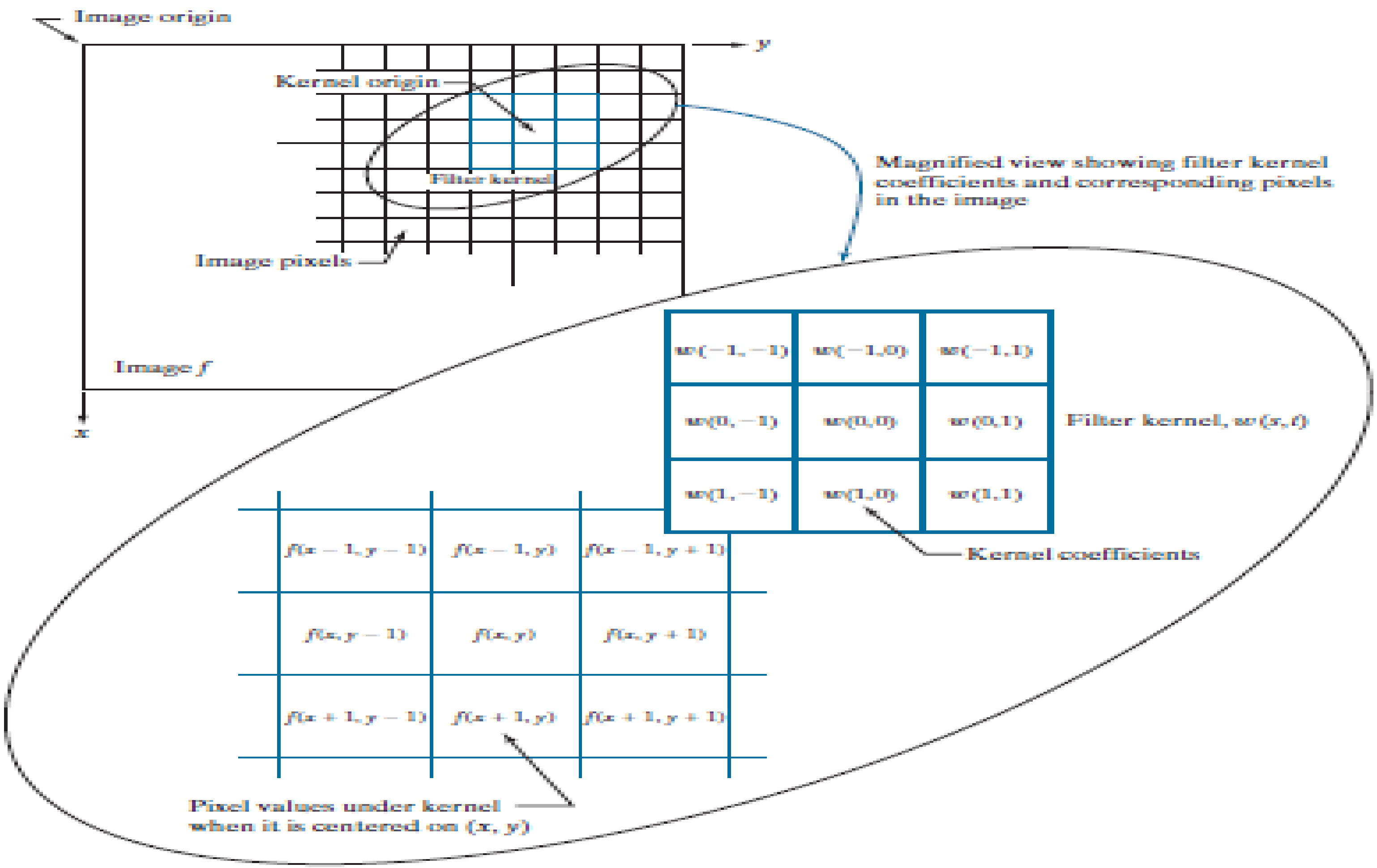
SHARPENING FILTERS

MECHANISM OF LINEAR SPATIAL FILTERING



Images taken from Gonzalez & Woods, Digital Image Processing

SHARPENING FILTERS

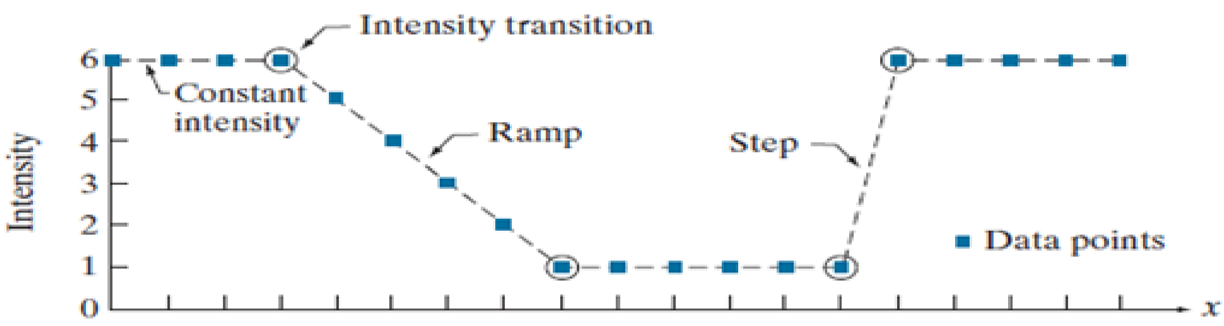
- ✓ **Smoothing filters remove fine details**
- ✓ **Sharpening filters seek to highlight fine details**
  - ✓ Remove blurring from images
  - ✓ Highlight edges
- ✓ ***Applications***
  - electronic printing
  - medical imaging
  - industrial inspection
  - Autonomous guidance in military systems.

SHARPENING FILTERS

- ✓ sharpening is often referred to as ***highpass* filtering**.
- ✓ In this case**,** **high frequencies** (which are responsible for fine details) **are passed**, while **low frequencies are attenuated or rejected.**
- ✓ **Sharpening filters are based on *spatial differentiation*.**
- ✓ **Differentiation measures the *rate of change* of a function**
- ✓ The strength of the response of a **derivative operator** is proportional to the **magnitude of the intensity discontinuity** at the point at which the operator is applied.
- ✓ Hence, **image differentiation enhances edges and other discontinuities (such as noise) and de-emphasizes areas with slowly varying intensities**.

DERIVATIVES

- ✓ Sharpening filters based on **first- and second-order derivatives**.
- ✓ In one dimensional, lets focus on behavior of these derivatives in areas of constant intensity, at the onset and end of discontinuities (*step* and *ramp discontinuities*), and along *intensity ramps.*



FIRST ORDER AND SECOND ORDER DERIVATIVE FUNCTION

- ✓ Derivatives of a digital function are defined in terms of differences.

First order and second order derivative function

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

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

DERIVATIVES

- **Definition we use for a *first derivative:***
  - Must be zero in areas of constant intensity.
  - Must be nonzero at the onset of an intensity step or ramp.
  - Must be nonzero along intensity ramps.

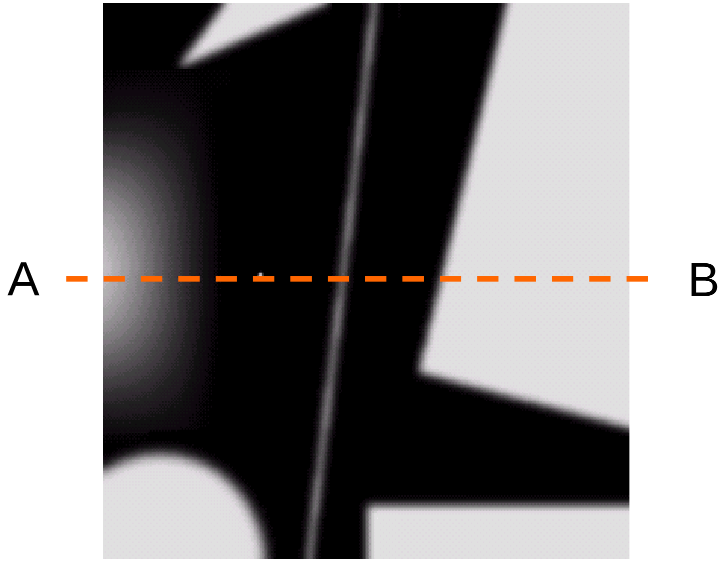
- **Definition of second Derivative**
  - Must be zero in areas of constant intensity.
  - Must be nonzero at the onset *and* end of an intensity step or ramp.
  - Must be zero along intensity ramps.

- We deal with digital quantities whose values are finite.
- Therefore, the maximum possible intensity change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.

SPATIAL DIFFERENTIATION

Images taken from Gonzalez & Woods, Digital Image Processing

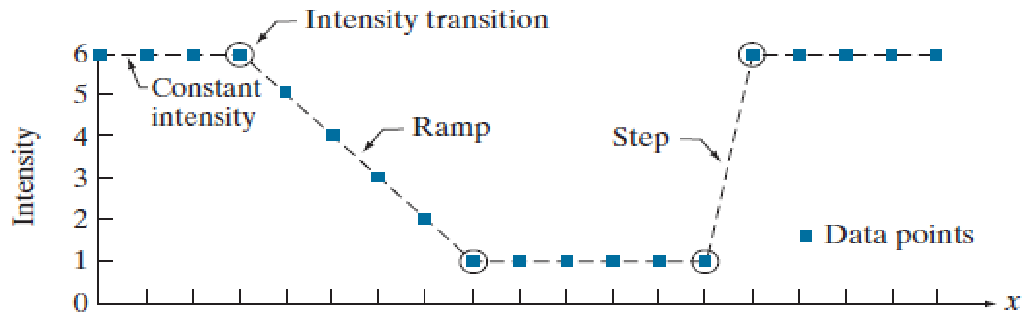
Let's consider a simple 1 dimensional example



SPATIAL DIFFERENTIATION

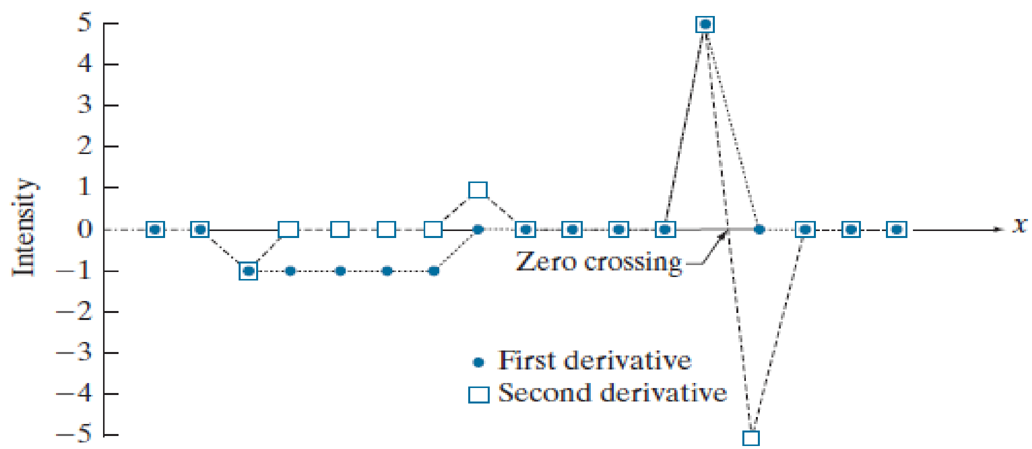
a
b
c

FIGURE 3.44
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b)Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.



Values of 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	

→ x



SPATIAL DIFFERENTIATION

- ✓ As we traverse the profile from left to right we encounter first an area of constant intensity and, as (b) and (c) show, both derivatives are zero there, so condition (1) is satisfied by both.
- ✓ Next, we encounter an intensity ramp followed by a step, and we note that the first-order derivative is nonzero at the onset of the ramp and the step; similarly, the second derivative is nonzero at the onset and end of both the ramp and the step; therefore, property (2) is satisfied by both derivatives.
- ✓ Finally, see that property (3) is satisfied also by both derivatives because the first derivative is nonzero and the second is zero along the ramp. Note that the sign of the second derivative changes at the onset and end of a step or ramp.
- ✓ A step transition a line joining these two values crosses the horizontal axis midway between the two extremes. This ***zero crossing*** **property is quite useful for locating edges**

SPATIAL DIFFERENTIATION

- ✓ **Edges** in digital images often are ramp-like transitions in intensity, in which case the first derivative of the image would result in thick edges because the derivative is nonzero along a ramp.
- ✓ On the other hand, the second derivative would produce a double edge one pixel thick, separated by zeros.
- ✓ Second derivative enhances fine detail much better than the first derivative, a property ideally suited for sharpening images.
- ✓ Second derivatives require fewer operations to implement than first derivatives

SHARPENING SPATIAL FILTERS

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**

In general the second derivative is better than the first derivative for image enhancement. The principle use of first derivative is for edge extraction.

LAPLACIAN FILTER

Sharpening filter - *Laplacian*

- ✓ **Isotropic -** whose response is independent of the direction of intensity discontinuities in the image to which the filter is applied
- ✓ **One of the simplest sharpening filters**

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…)

Laplacian

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

Filter Designed

0	1	0
1	-4	1
0	1	0

LAPLACIAN KERNELS

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

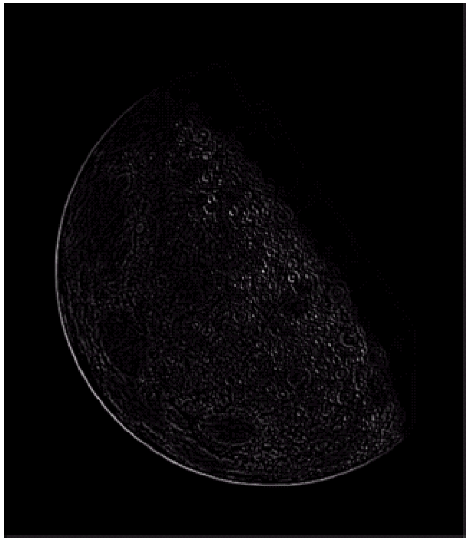
a b c d

FIGURE 3.45 (a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

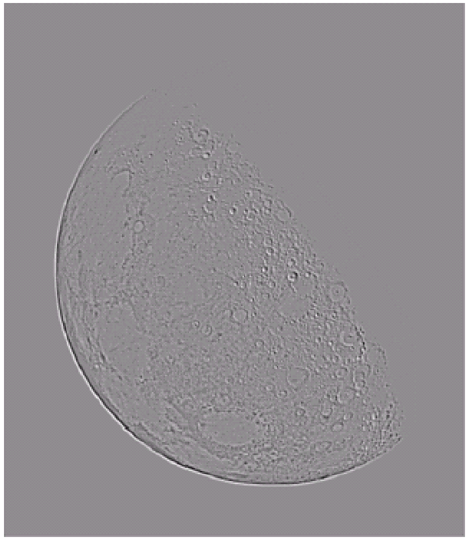
THE LAPLACIAN (CONT…)

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 (it contains negative values).**
- ✓ **when an image is convolved with a kernel whose coefficients sum to zero, the elements of the resulting filtered image sum to zero also, so images convolved with the kernels 0 will have negative values in general.**
- ✓ **Do some more processing in order to get the final enhanced image**
- ✓ **Subtract the Laplacian result from the original image to generate the final sharpened enhanced image (Fig 3.45 kernel (a) or (b) is used)**

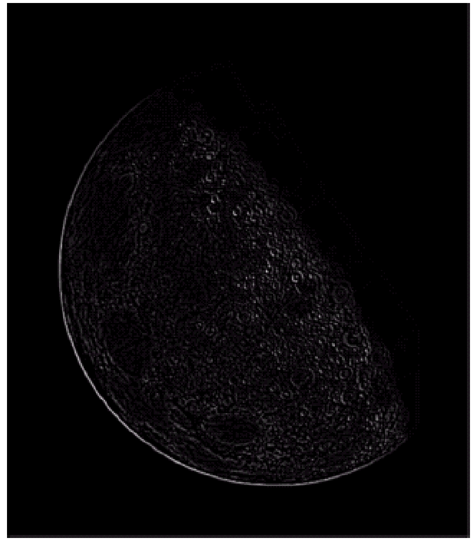
$$g(x,y) = f(x,y) - \nabla^2 f$$

Laplacian Filtered Image Scaled for Display

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)

LAPLACIAN FOR IMAGE SHARPENING

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

where $f(x, y)$ and $g(x, y)$ are the input and sharpened images

***c* = −1 if the Laplacian kernels in Fig. 3.45(a) or (b) is used, and**

***c* = 1 if either of the other two kernels in Fig. 3.45(c) or (d) is used.**

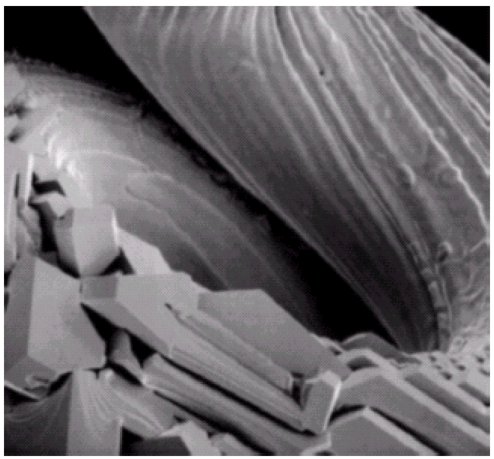
SIMPLIFIED IMAGE ENHANCEMENT

The entire enhancement can be combined into a single filtering operation

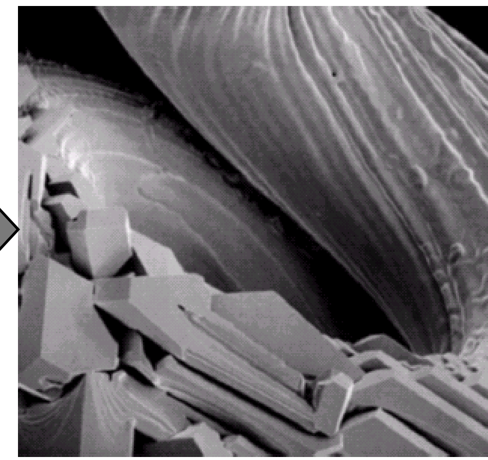
$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) \\
&\qquad - 4f(x,y)] \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) \\
&\qquad - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

SIMPLIFIED IMAGE ENHANCEMENT (CONT…)

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

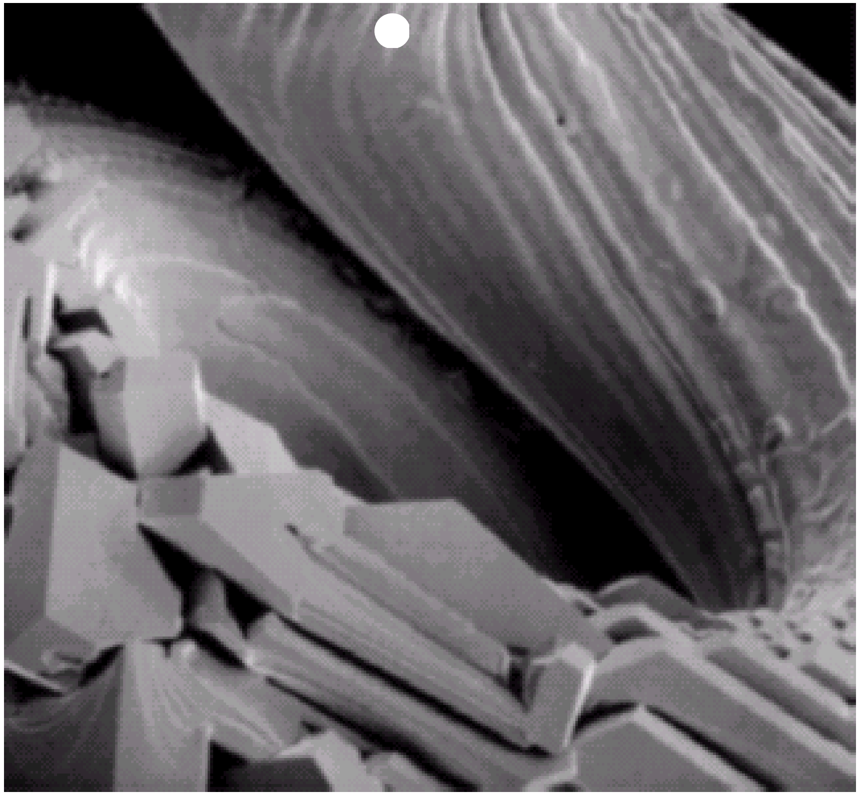


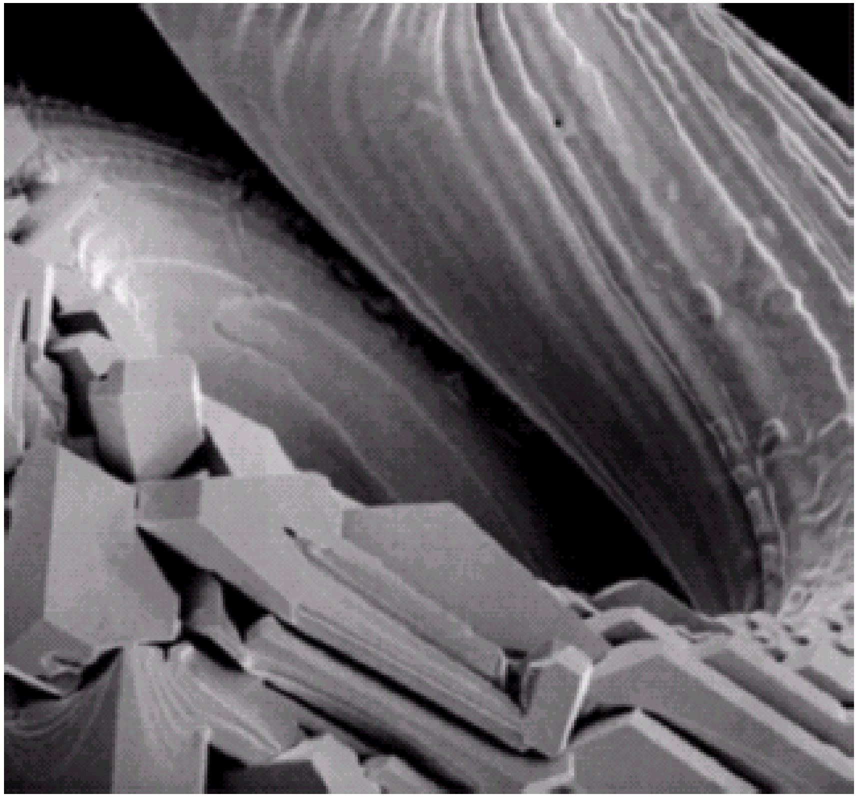
0	-1	0
-1	5	-1
0	-1	0



Images taken from Gonzalez & Woods, Digital Image Processing

SIMPLIFIED IMAGE ENHANCEMENT (CONT…)





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

VARIANTS ON THE SIMPLE LAPLACIAN

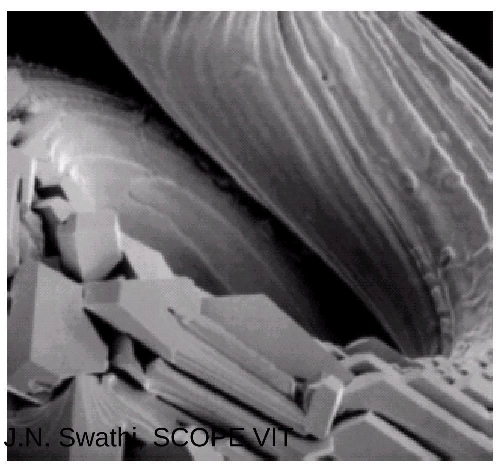
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

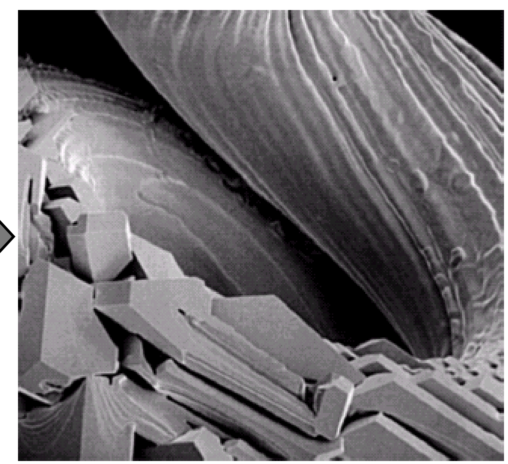
Simple Laplacian

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)

UNSHARP MASKING AND HIGHBOOST FILTERING

- ✓ Subtracting an unsharp (smoothed) version of an image from the original image is process that has been used since the 1930s by the printing and publishing industry to sharpen images.
- ✓ Steps involved in *unsharp masking:*

1. Blur the original image.

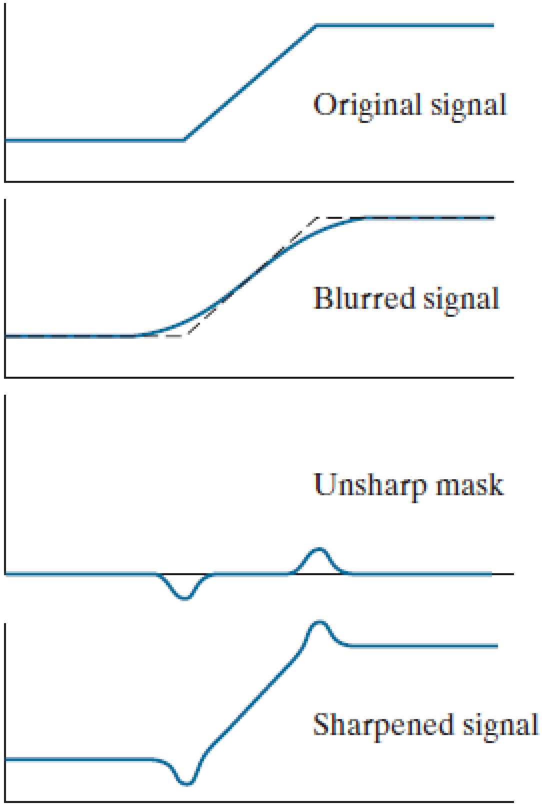
2. Subtract the blurred image from the original (the resulting difference is called the *mask*.)

3. Add the mask to the original.

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

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

UNSHARP MASKING



a
b
c
d

FIGURE 3.48
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).

UN SHARP MASKING

- $G_{mask}(x,y) = f(x,y) – f_{lp}(x,y)$

High Boost Filtering

- $F_{hb}(x,y) = A.\ (G_{mask}(x,y)$ **or unsharp mask)**
- $F_{hb}(x,y) = A.\ (f(x,y) – f_{lp}(x,y))$

0	-1	0
-1	A+4	-1
0	-1	0

If A=0 → Laplacian filter
If A=1 → Sharpened filter
If A>1 → High boost filter

UN SHARP MASKING AND HIGH BOOST FILTERING

Images taken from Gonzalez & Woods, Digital Image Processing

a b c
d e

FIGURE 3.49 (a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) Result of highboost filtering with $k = 4.5$.

USE OF FIRST DERIVATIVE FOR EDGE EXTRACTION – THE GRADIENT

First derivatives in image processing are implemented using the magnitude of the gradient.

$$\nabla \mathbf{f} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^t$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$\nabla f = mag(\nabla \mathbf{f}) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{0.5} \approx |G_x| + |G_y|$$

M(x, y) is an image of the same size as the original, created **when *x* and *y* are allowed to vary over all pixel locations in *f***.

It is common practice to refer to this image as the ***gradient image***

APPROXIMATE THE SQUARES AND SQUARE ROOT OPERATIONS BY ABSOLUTE VALUES

- ✓ Because the **components of the gradient vector** are derivatives, they **are linear operators**.
- ✓ However, the **magnitude of this vector is not linear**, because of the squaring and square root operations.
- ✓ On the other hand, the **partial derivatives are not rotation invariant**, but the **magnitude of the gradient vector is rotation invariant**.

$$\nabla \mathbf{f} = \begin{bmatrix} \dfrac{\partial f}{\partial x} & \dfrac{\partial f}{\partial y} \end{bmatrix}^t$$

$$\nabla f = mag(\nabla \mathbf{f}) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{0.5} \approx |G_x| + |G_y|$$

ROBERTS CROSS DIFFERENCE OPERATOR

The simplest approximations to a first-order derivative

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

$$g_x = (z_8 - z_5) \text{ and } g_y = (z_6 - z_5)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts cross operator

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

ROBERTS OPERATOR

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

Roberts

$$\nabla f = mag\,(\nabla \mathbf{f}) \approx \left|G_x\right| + \left|G_y\right|$$

PREWITT OPERATOR

Kernels of size 2 × 2 are simple conceptually, but they are not as useful for computing edge direction as kernels that are symmetric about their centers, the smallest of which are of size 3 × 3.

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

PREWITT OPERATORS

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

Prewitt

$$\nabla f = mag\,(\nabla \mathbf{f}) \approx |G_x| + |G_y|$$

SOBEL OPERATORS

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

which is based on these coordinates

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

SOBEL OPERATORS

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

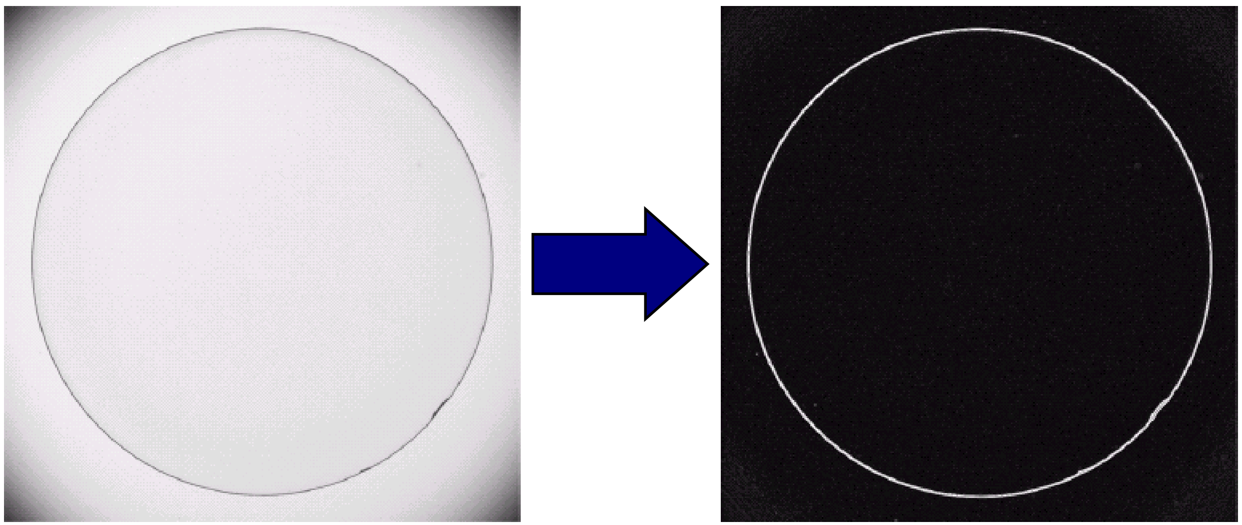
-1	-2	-1
0	0	0
1	2	1

-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

$$\nabla f = mag\,(\nabla \mathbf{f}) \approx \left|G_x\right| + \left|G_y\right|$$

SOBEL EXAMPLE



Sobel filters are typically used for edge detection

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

SUMMARY OF FILTERS

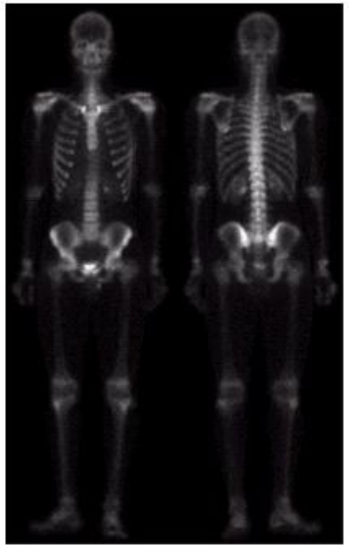
Filter type	Spatial kernel in terms of lowpass kernel, lp
Lowpass	$lp(x,y)$
Highpass	$hp(x,y) = \delta(x,y) - lp(x,y)$
Bandreject	$br(x,y) = lp_1(x,y) + hp_2(x,y)$ $= lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]$
Bandpass	$bp(x,y) = \delta(x,y) - br(x,y)$ $= \delta(x,y) - [lp_1(x,y) + [\delta(x,y) - lp_2(x,y)]]$

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 to the right

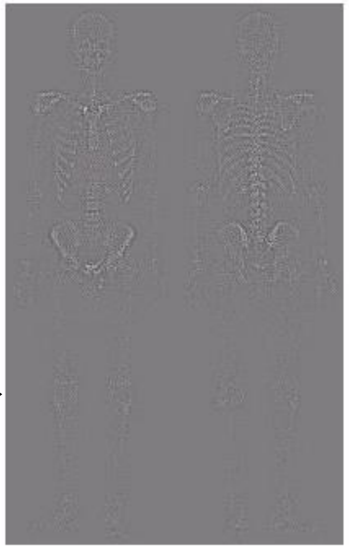
COMBINING SPATIAL ENHANCEMENT METHODS

Input
Image



(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

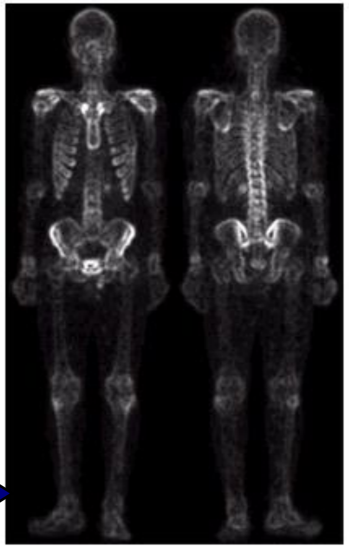
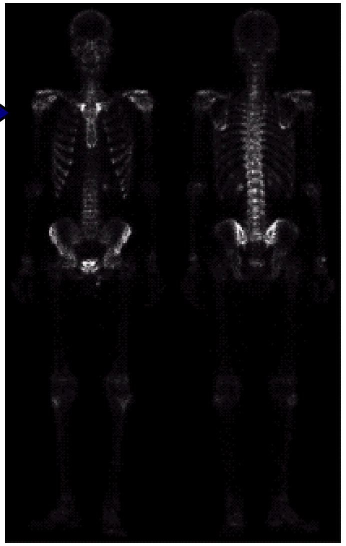


Image (d) smoothed with a 5*5 averaging filter

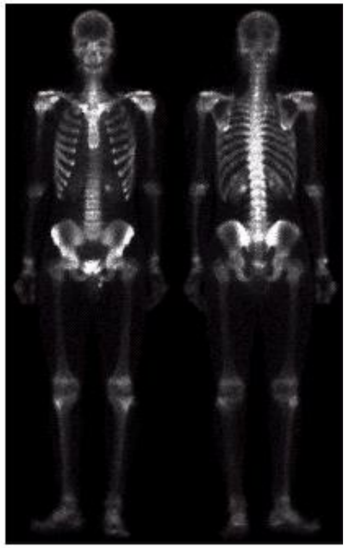
The product of (c) and (e) which will be used as a **mask**

(e)



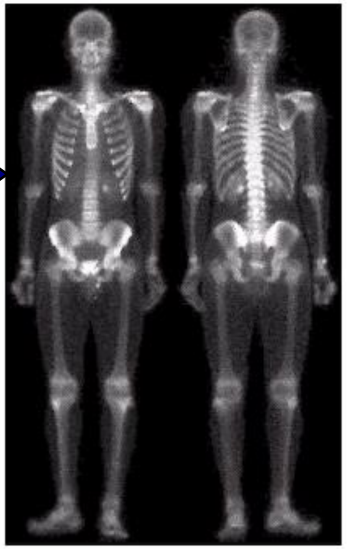
Sharpened image which is sum of (a) and (f)

(f)



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

(g)



COMBINING SPATIAL ENHANCEMENT METHODS