

# Image Enhancement in spatial domain

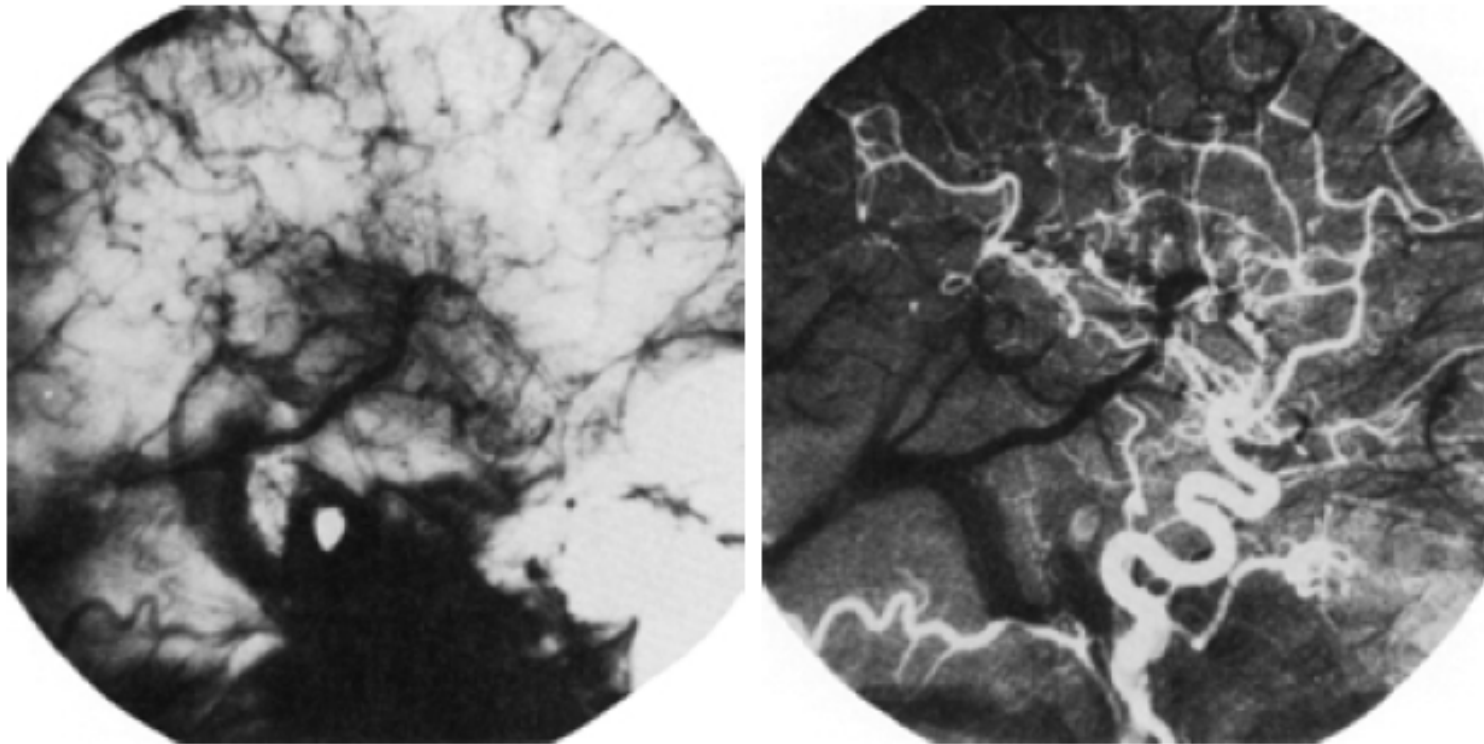
Digital Image Processing

GW Chapter 3 from Section 3.4.1 (pag 110)

Part 2: Filtering in spatial domain

# Mask mode radiography

- Image subtraction in medical imaging



a b

**FIGURE 3.29**  
Enhancement by image subtraction.  
(a) Mask image.  
(b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

# Range rescaling

- The values in a difference image can range from a minimum of  $-255$  to a maximum of  $255$ , so some sort of scaling is required to display the results. There are two principal ways to scale a difference image.
- One method is to add  $255$  to every pixel and then divide by  $2$ . It is not guaranteed that the values will cover the entire 8-bit range from  $0$  to  $255$ , but all pixel values definitely will be within this range. This method is fast and simple to implement, but it has the limitations that the full range of the display may not be utilized and, potentially more serious, the truncation inherent in the division by  $2$  will generally cause loss in accuracy.

# Range rescaling

- Second method
  - First, the value of the minimum difference is obtained and its negative added to all the pixels in the difference image (this will create a modified difference image whose minimum values is 0).
  - Then, all the pixels in the image are scaled to the interval **[0, 255]** by multiplying each pixel by the quantity  $255/\text{Max}$ , where Max is the maximum pixel value in the modified difference image. It is evident that this approach is considerably more complex and difficult to implement.

# Image averaging

- Consider an image sequence where each image is corrupted by noise

$$g(x,y)=f(x,y)+n(x,y)$$

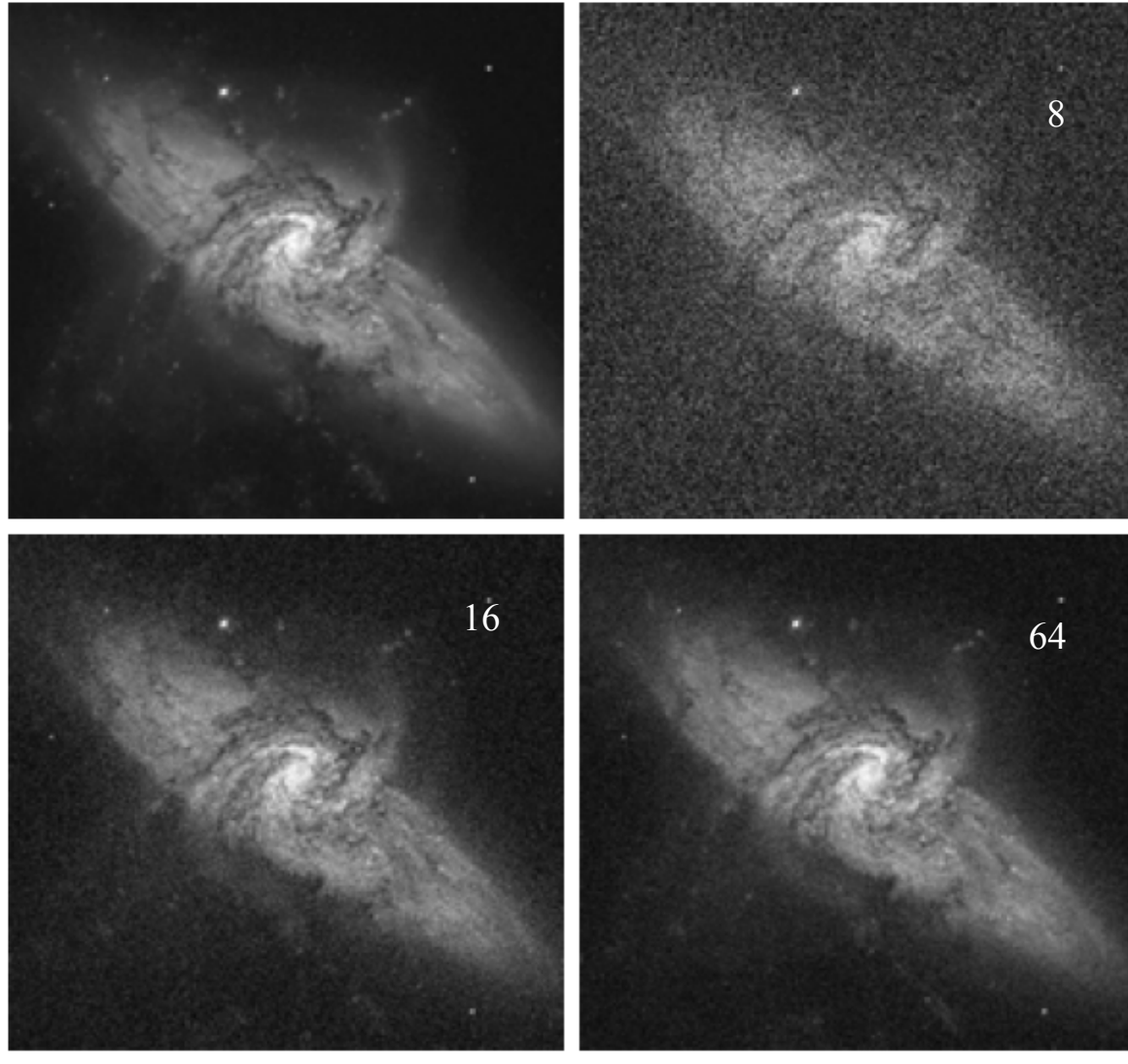
where the assumption is that at every pair of coordinates  $(x,y)$  the noise is uncorrelated and has zero average value

- It can be shown that the result of the following procedure is to reduce the noise content

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y) \quad (3.4-3)$$

- By adding a set of noisy images the noise “cancels out”. The more images we sum, the better
- Warning: images should be aligned (registered) to avoid defocusing

# Denoising by summation



# Basics of spatial filtering

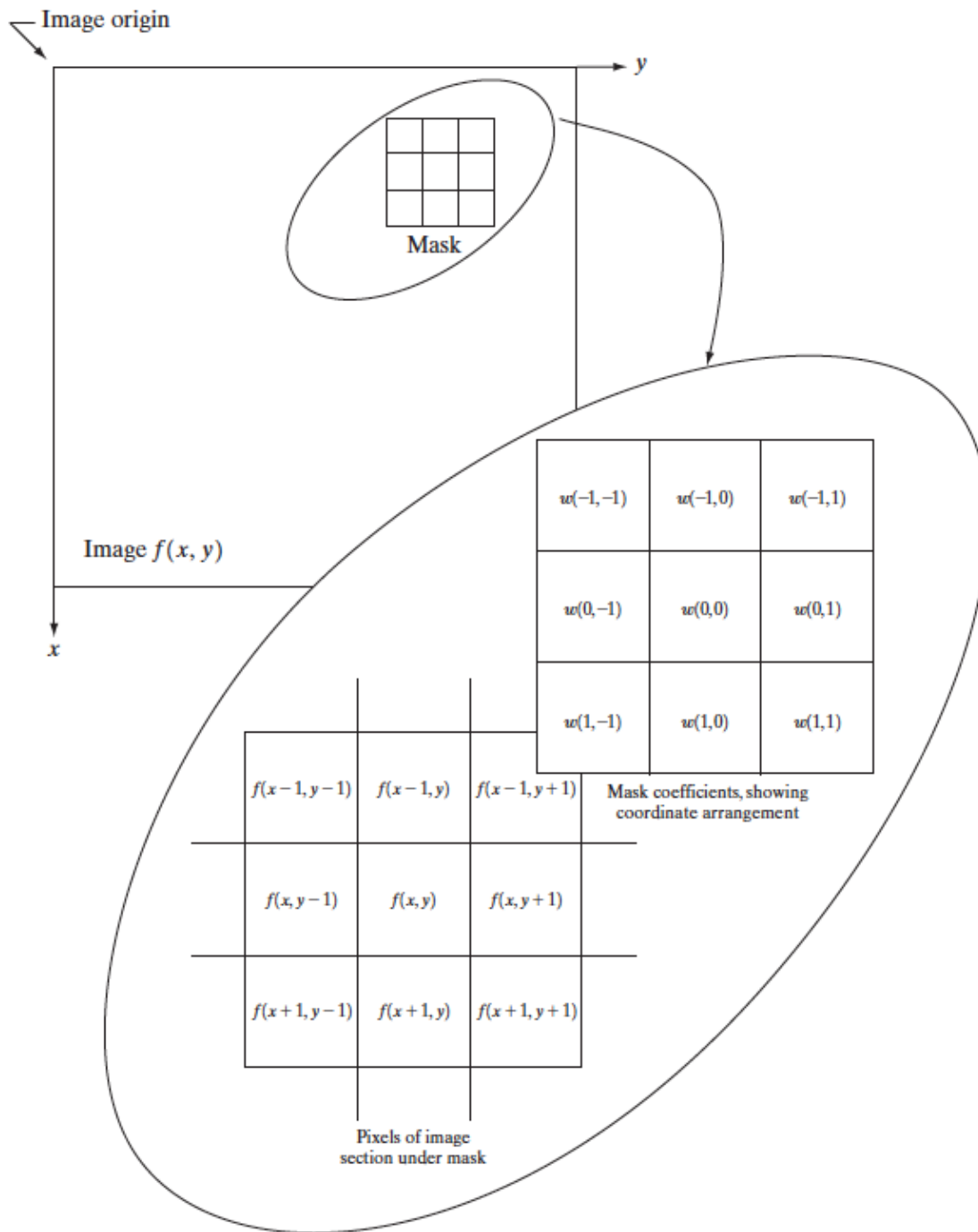
- Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a subimage that has the same dimensions as the neighborhood.
- The subimage is called *a filter, mask, kernel, template, or window*, with the first three terms being the most prevalent terminology.
- The values in a filter subimage are referred to as *coefficients*, rather than pixels.

# Spatial filtering

- The process consists simply of moving the filter mask from point to point in an image. At each point  $(x, y)$ , the response of the filter at that point is calculated using a predefined relationship.
- For linear spatial filtering (see Section 2.6 regarding linearity), the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.
- Example: with a 3x3 mask

$$R = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots \\ + w(0, 0)f(x, y) + \dots + w(1, 0)f(x + 1, y) + w(1, 1)f(x + 1, y + 1),$$





**FIGURE 3.32** The mechanics of spatial filtering. The magnified drawing shows a  $3 \times 3$  mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

# Spatial linear filtering

- In general, linear filtering of an image  $f$  of size  $M \times N$  with a filter mask of size  $m \times n$  is given by the expression

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

- where, from the previous paragraph,  $a=(m-1)/2$  and  $b=(n-1)/2$
- To generate a complete filtered image this equation must be applied for  $x=0, 1, 2, \dots, p, M-1$  and  $y=0, 1, 2, \dots, p, N-1$

# Smoothing linear filters qui

- Smoothing filters are used for blurring and for noise reduction.
  - Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves.
  - Anti-aliasing filters are smoothing filters
- Smoothing linear filters
  - The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

# Smoothing spatial filters

- Smoothing filters are used for blurring and for noise reduction.
- Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves.
- Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.
- Idea: By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced “sharp” transitions in gray levels
  - Advantage: noise is removed
  - Disadvantage: edges are blurred

# Image extension at borders

- What happens when the center of the filter approaches the border of the image?
  - Consider for simplicity a square mask of size  $n \times n$ . At least one edge of such a mask will coincide with the border of the image when the center of the mask is at a distance of  $(n-1)/2$  pixels away from the border of the image. If the center of the mask moves any closer to the border, one or more rows or columns of the mask will be located outside the image plane
- Possible solutions
  - To limit the excursions of the center of the mask to be at a distance no less than  $(n-1)/2$  pixels from the border.
  - Advantage: all the pixels in the filtered image will have been processed with the full mask, but
  - Shortcoming: the resulting filtered image will be smaller than the original

# Image extension at borders

- “padding” the image by
  - adding rows and columns of 0’s (or other constant gray level), or
  - padding by replicating rows or columns.
  - The padding is then stripped off at the end of the process.
  - Advantage: this keeps the size of the filtered image the same as the original, but
  - Disadvantage: the values of the padding will have an effect near the edges that becomes more prevalent as the size of the mask increases

# Smoothing spatial filters

- Example of smoothing averaging filter

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

BOX filter

$$\frac{1}{16} \times$$

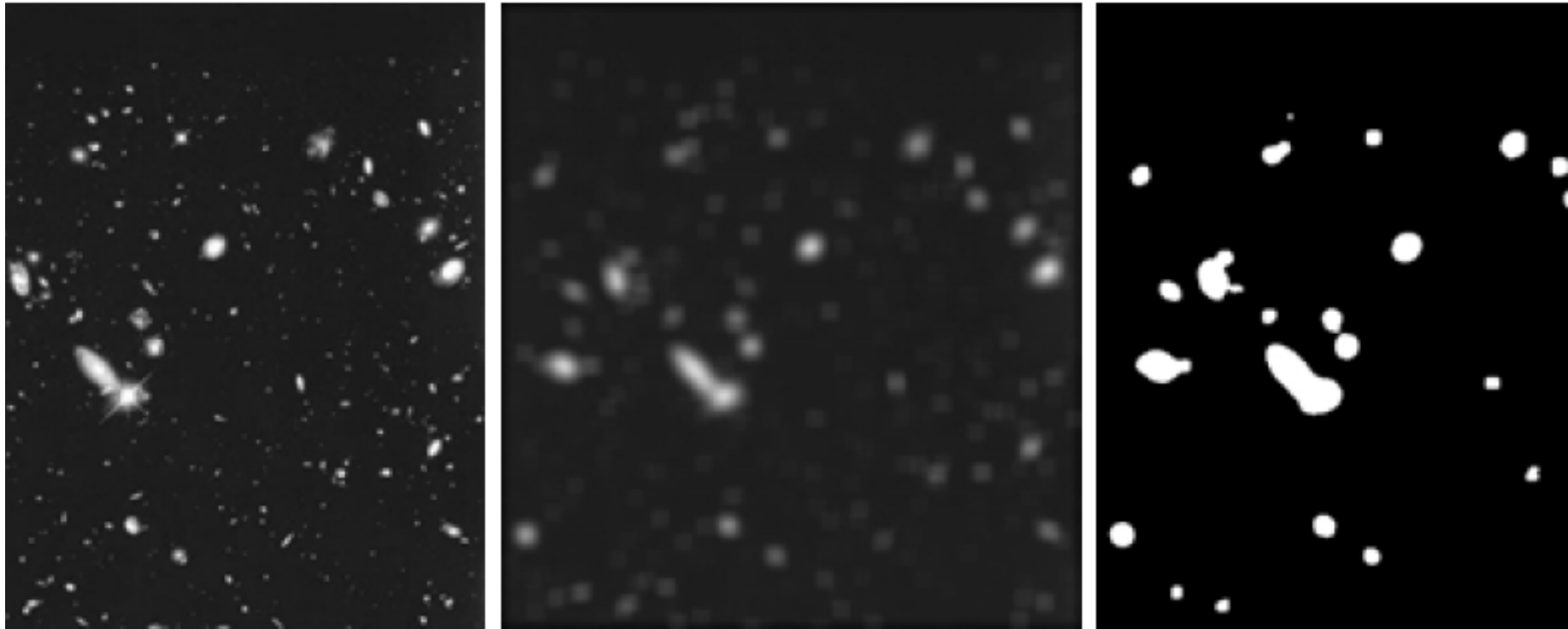
1	2	1
2	4	2
1	2	1

Weighted average filter

Mind the notations!!

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

## Example of segmentation by filtering + thresholding



c

**E 3.36** (a) Image from the Hubble Space Telescope. (b) Image processed by a  $15 \times 15$  averaging filter. (c) Binary image obtained by thresholding (b). (Original image courtesy of NASA.)



# Non-linear filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.
- In general, however, the filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration, and they do not explicitly use coefficients in the sum-of-products manner
- Example: noise reduction can be achieved effectively with a nonlinear filter whose basic function is to compute the median gray-level value in the neighborhood in which the filter is located.
- Computation of the median is a nonlinear operation

# Rank order filters

- Order-statistics filters are *nonlinear* spatial filters whose response is based on *ordering* (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking
- **Median filter**: replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median).
  - Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.
  - impulse noise, also called salt -and -pepper noise

# Median filter

- The median,  $m$ , of a set of values is such that half the values in the set are less than or equal to  $m$ , and half are greater than or equal to  $m$ .
- In order to perform median filtering at a point in an image
  1. sort the values of the pixel in question and its neighbors
  2. determine their median
  3. assign this value to that pixel
- Example (3x3 neighborhood)  
(10, 20, 20, 20, 15, 20, 20, 25, 100)      **sorting**  
(10, 15, 20, 20, **20**, 20, 20, 25, 100)      **median=result of filtering**
- **Consequence:** isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than  $n^2 / 2$  (one-half the filter area), are “eliminated” by an  $n*n$  median filter

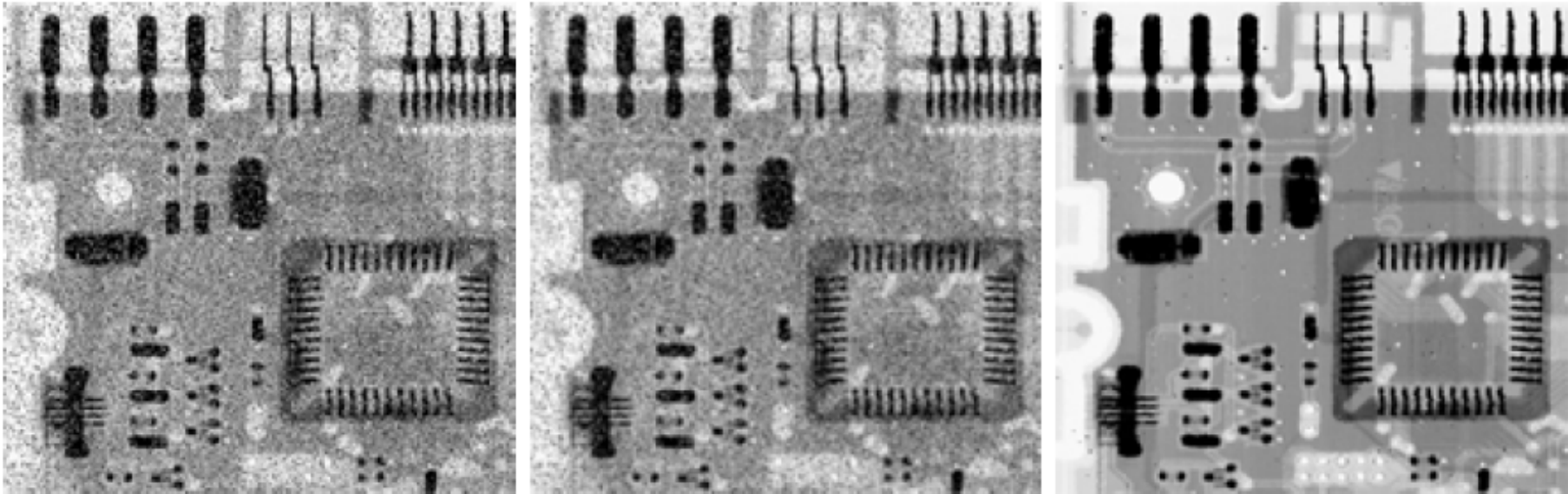
# Max and min filters

- Select the maximum and the minimum values in the window, respectively

$$R = \max\{z_k \mid k=1, 2, \dots, p, \dots, 9\}$$

$$r = \min\{z_k \mid k=1, 2, \dots, p, \dots, 9\}$$

# Example



a b c

**FIGURE 3.37** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a  $3 \times 3$  averaging mask. (c) Noise reduction with a  $3 \times 3$  median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

# Sharpening spatial filters

- Job: to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.
- Averaging  $\leftrightarrow$  Integration
- Sharpening  $\leftrightarrow$  Differentiation
- This means that
  - any definition we use for a first derivative (1) must be zero in flat segments (areas of constant gray-level values); (2) must be nonzero at the onset of a gray-level step or ramp; and (3) must be nonzero along ramps.
  - Similarly, any definition of a second derivative (1) must be zero in flat areas; (2) must be nonzero at the onset and end of a gray-level step or ramp; and (3) must be zero along ramps of constant slope.

# Numerical approximations

- First order derivative

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

- Second order derivative

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

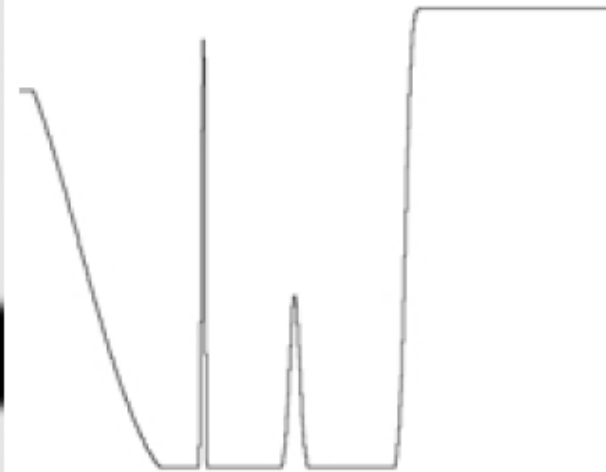
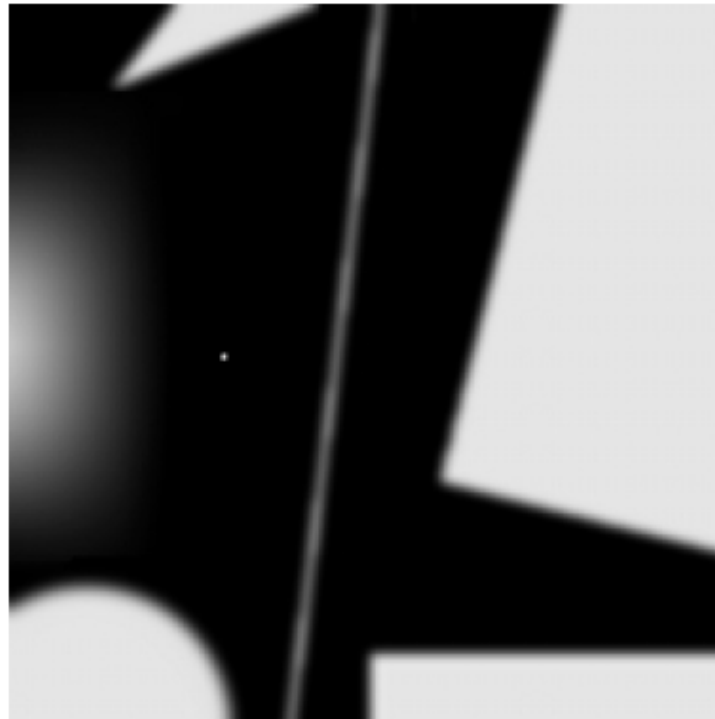
# Example

a b  
c

**FIGURE 3.38**

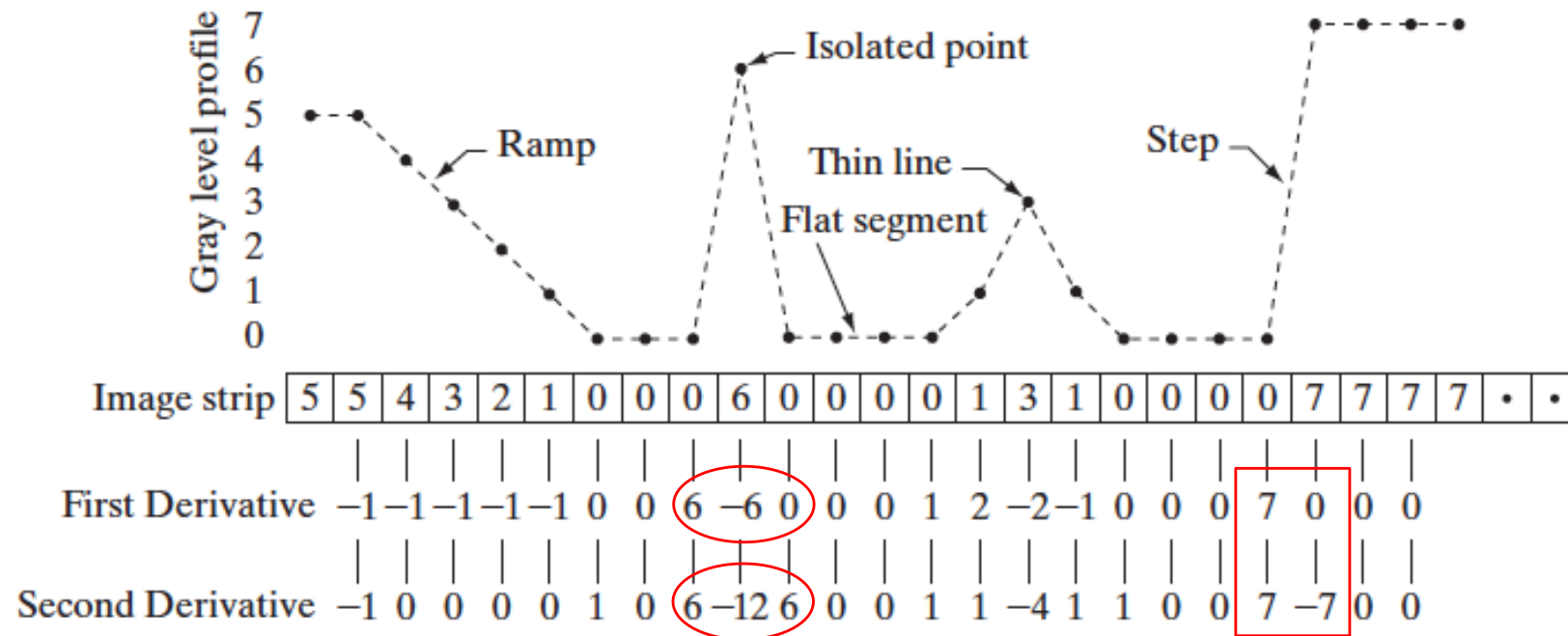
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.

(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).





# Example



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

# First vs Second order derivatives

- Ramp: the first-order derivative is nonzero along the entire ramp, while the second-order derivative is nonzero only at the onset and end of the ramp.
  - Because edges in an image resemble this type of transition, we conclude that **first-order derivatives produce “thick” edges and second-order derivatives, much finer ones.**
- Isolated noise point. Here, the response at and around the point is much **stronger for the second- than for the first-order derivative.** Of course, this is not unexpected.
  - A second-order derivative is much more aggressive than a first-order derivative in enhancing sharp changes.

# First vs Second order derivatives

- Grey-level step: the response of the two derivatives is similar at the gray-level step
  - in most cases when the transition into a step is not from zero, the second derivative will be weaker
  - the second derivative has a transition from positive back to negative. In an image, this shows as a thin double line.

# First versus second order derivatives

1. **First-order** derivatives generally produce **thicker** edges in an image.
2. **Second-order** derivatives have a stronger response to **fine details**, such as thin lines and isolated points.
3. **First order** derivatives generally have a stronger response to a **gray-level step**.
4. **Second-order** derivatives produce a **double** response at step changes in gray level.
  - We also note of second-order derivatives that, for similar changes in gray-level values in an image, their response is stronger to a line than to a step, and to a point than to a line.
  - First and second order derivatives can be used in combination

## 2<sup>nd</sup> order: Laplacian

- Premise: we are interested in *isotropic* filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied.
- In other words, isotropic filters are *rotation invariant*, in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result
- Laplacian operator

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

# Discrete approximations

- Partial second order derivative in the x direction

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

- Partial second order derivative in the y direction

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

- 2D Laplacian operator

$$\begin{aligned} \nabla^2 f = & [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] \\ & - 4f(x, y). \end{aligned} \tag{3.7-4}$$

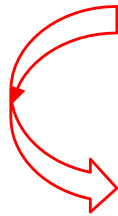
- This gives isotropic results for rotations of multiples of 90 deg.

# Discrete approximations

- The diagonal directions can be incorporated in the definition of the digital Laplacian by adding two more terms to Eq. (3.7-4), one for each of the two diagonal directions
  - The form is the same but the differences are calculated along the diagonal
  - The resulting global operator will account for this contribution and it will be covariant to rotations of multiples of 45 deg.

# Laplacian: examples

Opposite sign



0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-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.

Warning for sharpening

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the} \\ & \text{Laplacian mask is positive.} \end{cases} \quad (3.7-5)$$

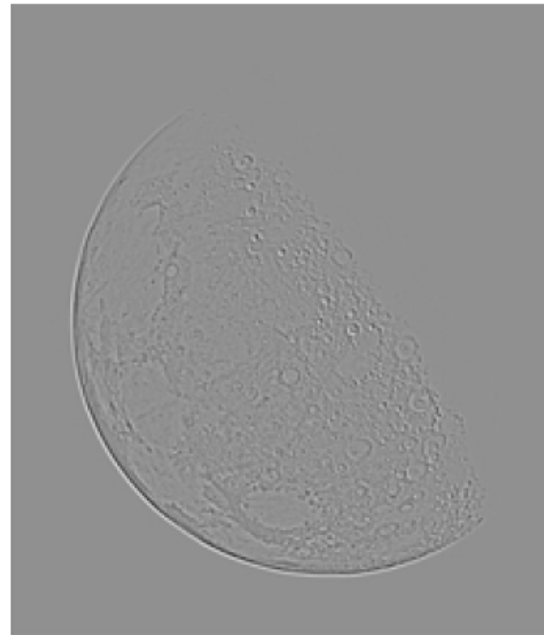
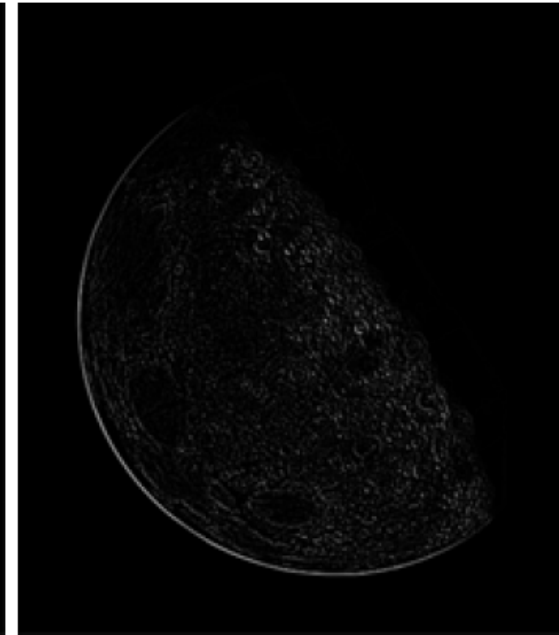


# Example: image enhance ment

a	b
c	d

**FIGURE 3.40**

(a) Image of the North Pole of the moon.  
(b) Laplacian-filtered image.  
(c) Laplacian image scaled for display purposes.  
(d) Image enhanced by using Eq. (3.7-5).  
(Original image courtesy of NASA.)



Laplacian increasing the contrast at the locations of gray-level discontinuities.

# Unsharp and high boost in spatial domain

- Unsharp masking

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

Blurred version of f



- High-boost filtering

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y)$$

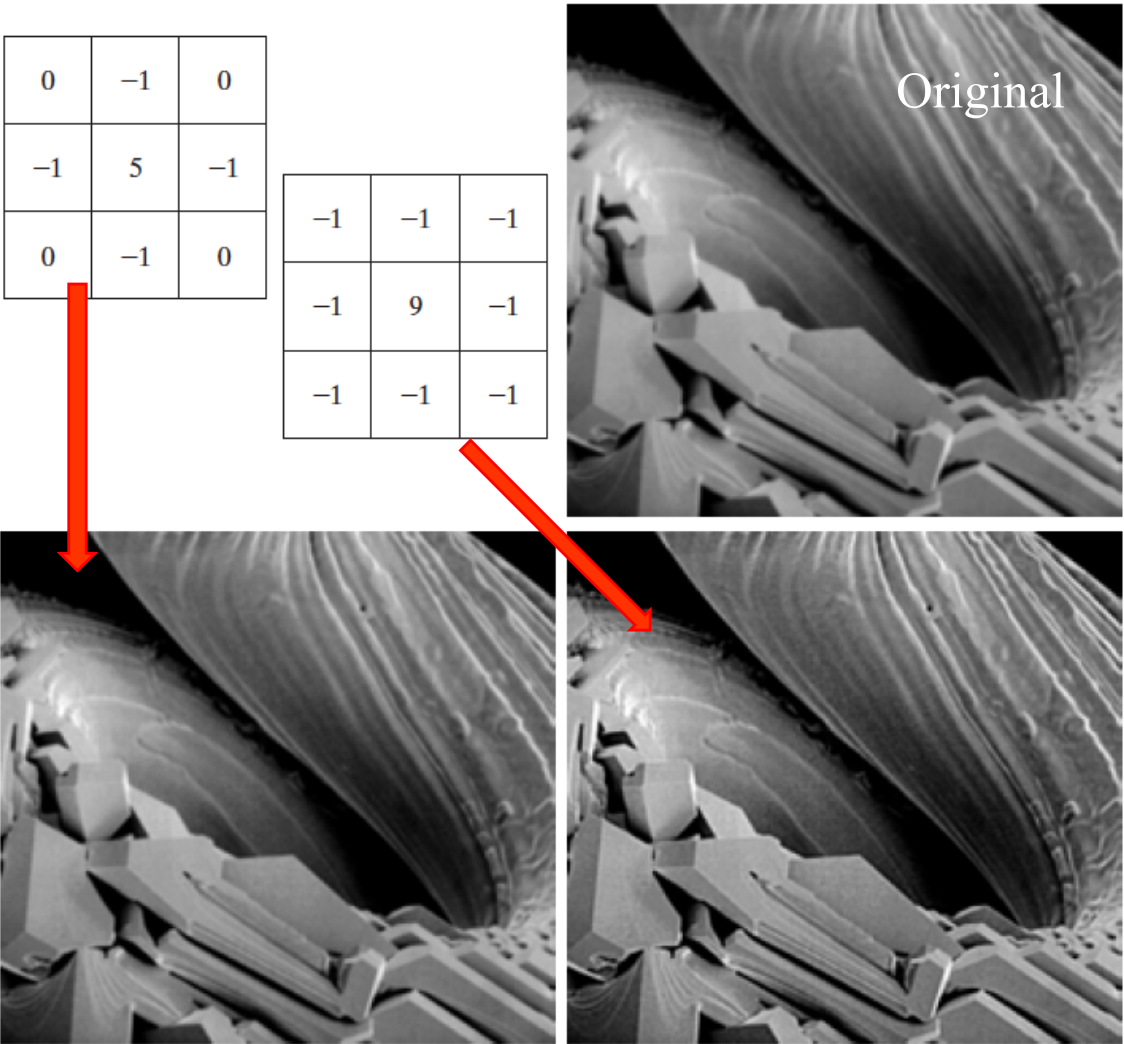
$$f_{hb}(x, y) = (A - 1)f(x, y) + f_s(x, y)$$

0	-1	0	-1	-1	-1
-1	$A + 4$	-1	-1	$A + 8$	-1
0	-1	0	-1	-1	-1

a b

**FIGURE 3.42** The high-boost filtering technique can be implemented with either one of these masks, with  $A \geq 1$ .

# Example



a b c  
d e

**FIGURE 3.41** (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

# Gradient

- The gradient in a point is a vector

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

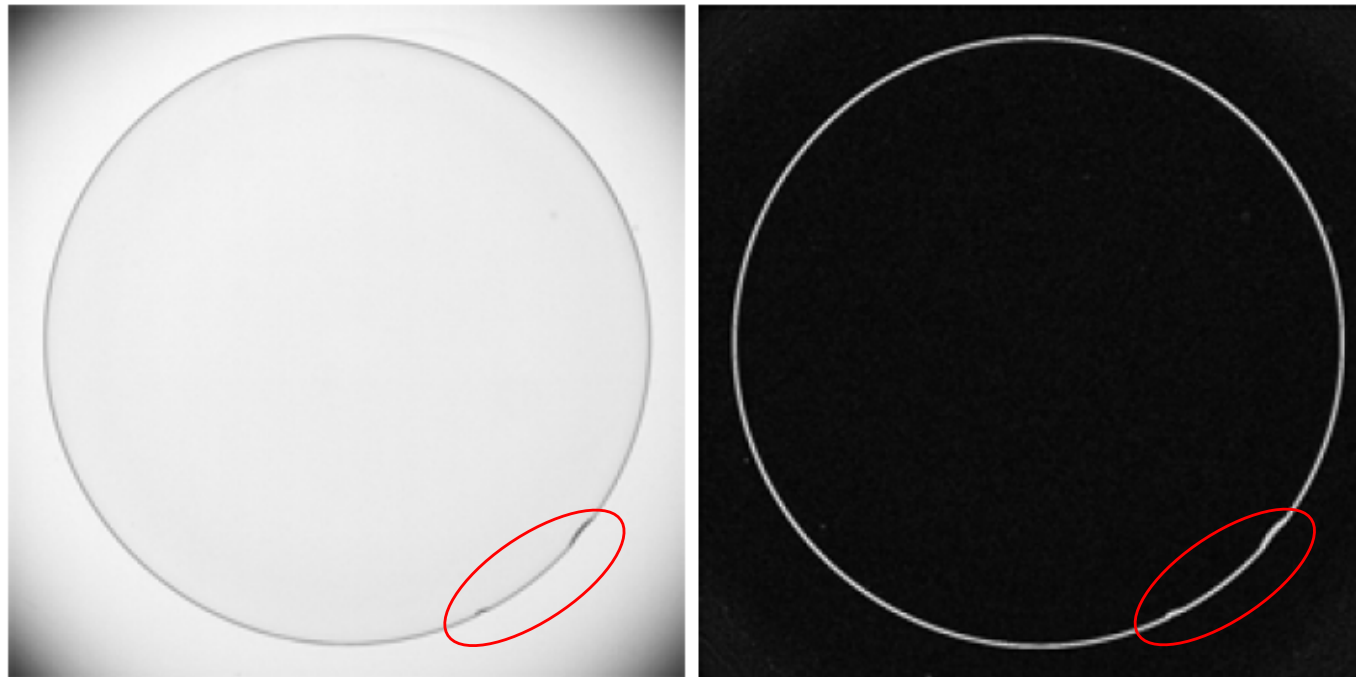
- Magnitude of the gradient

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

- Common simplification

$$\nabla f \approx |G_x| + |G_y|.$$

# Gradient: example



a b

**FIGURE 3.45**  
Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock).  
(b) Sobel gradient.  
(Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

The gradient enhances local graylevel discontinuities

# Combining spatial enhancement methods

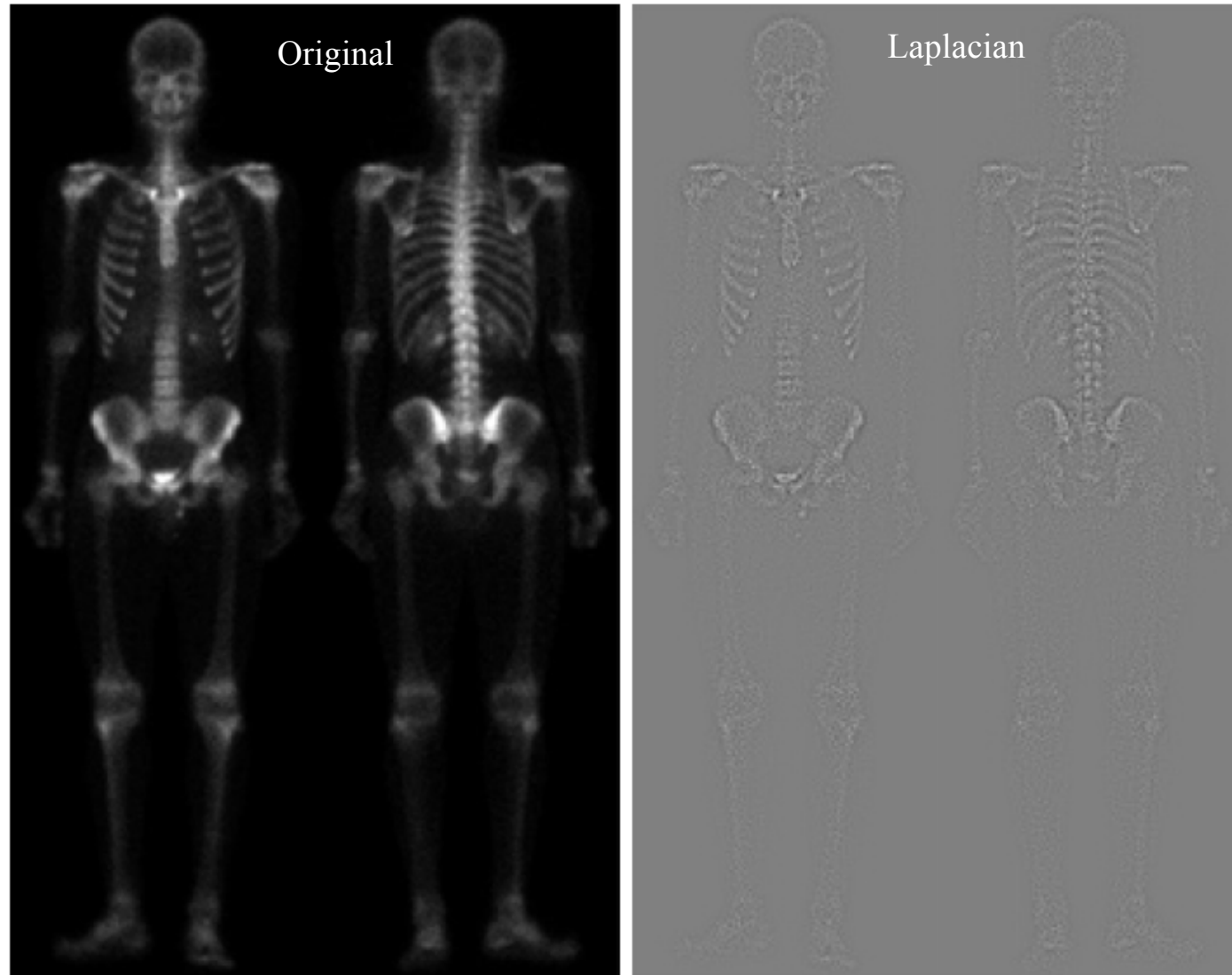
- The combination of different spatial enhancement methods leads to “better quality” images
- For instance
  - utilize the Laplacian to highlight fine detail, and the gradient to enhance prominent edges
  - a smoothed version of the gradient image can be used to mask the Laplacian image
  - increase the dynamic range of the gray levels by using a gray-level transformation

# Example

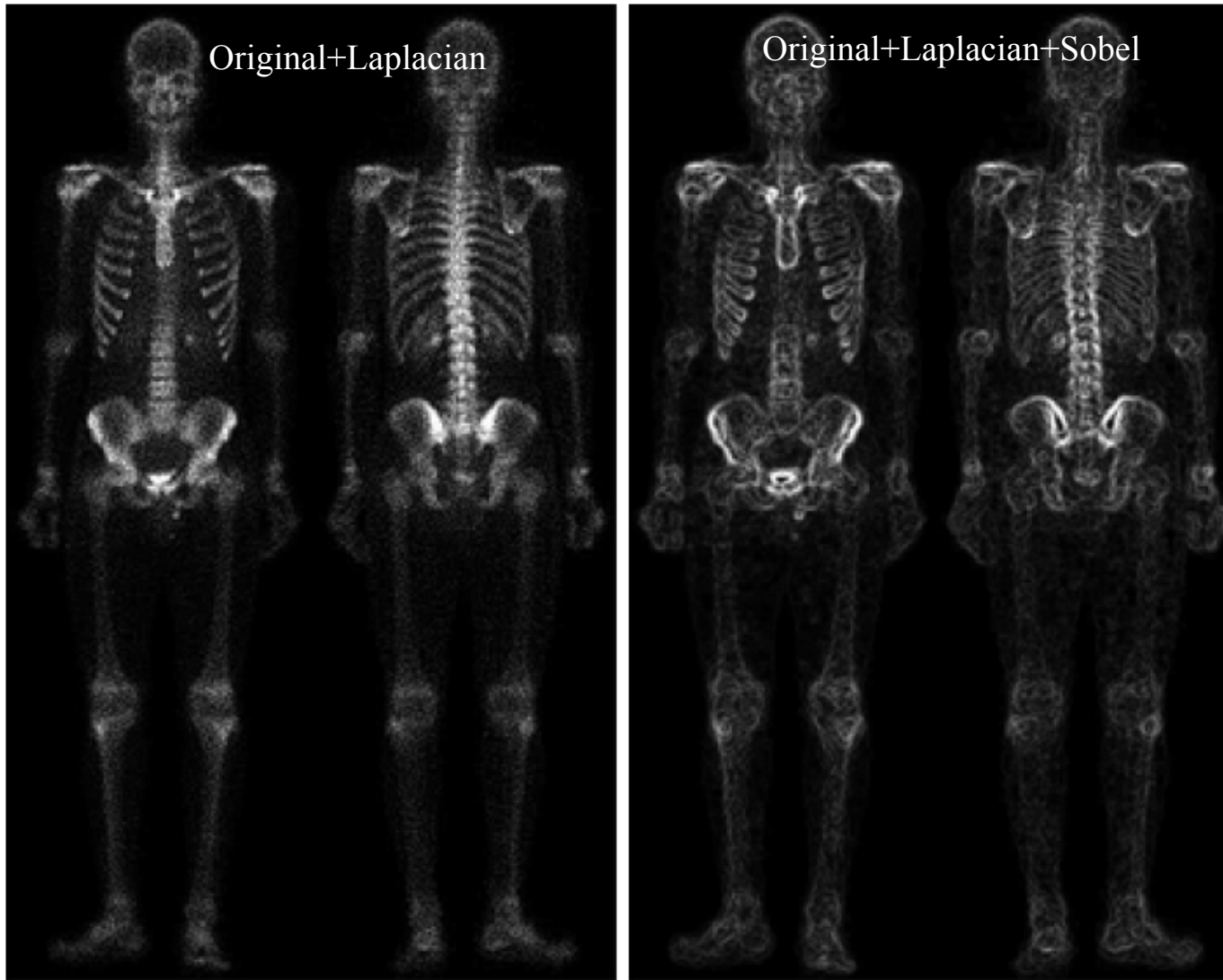
a	b
c	d

**FIGURE 3.46**

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

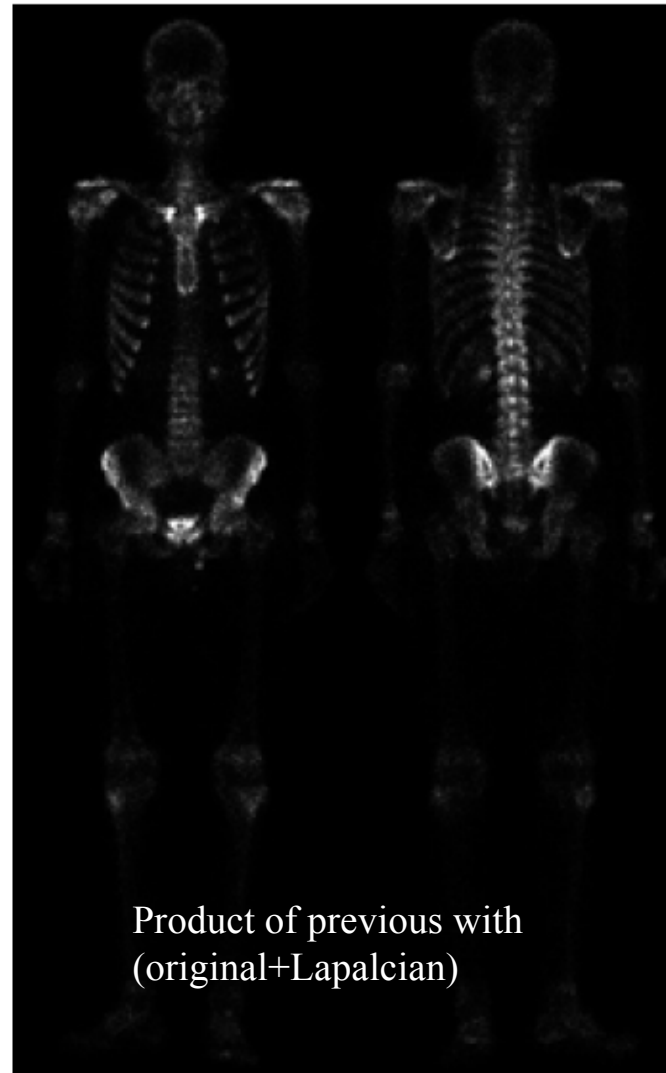
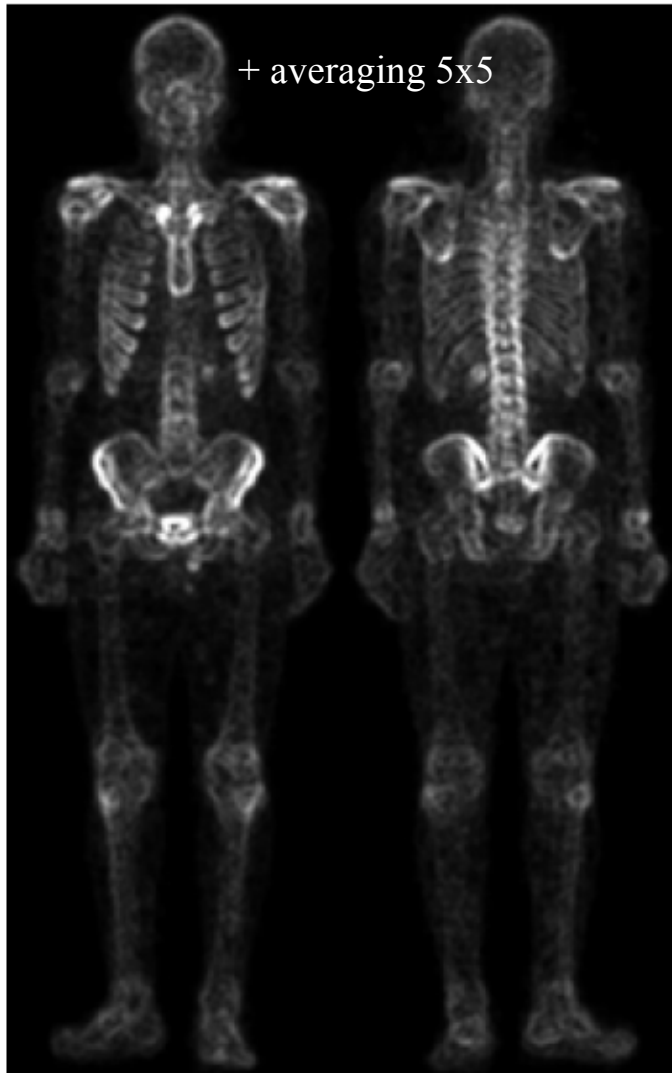


# Example





# Example



e	f
g	h

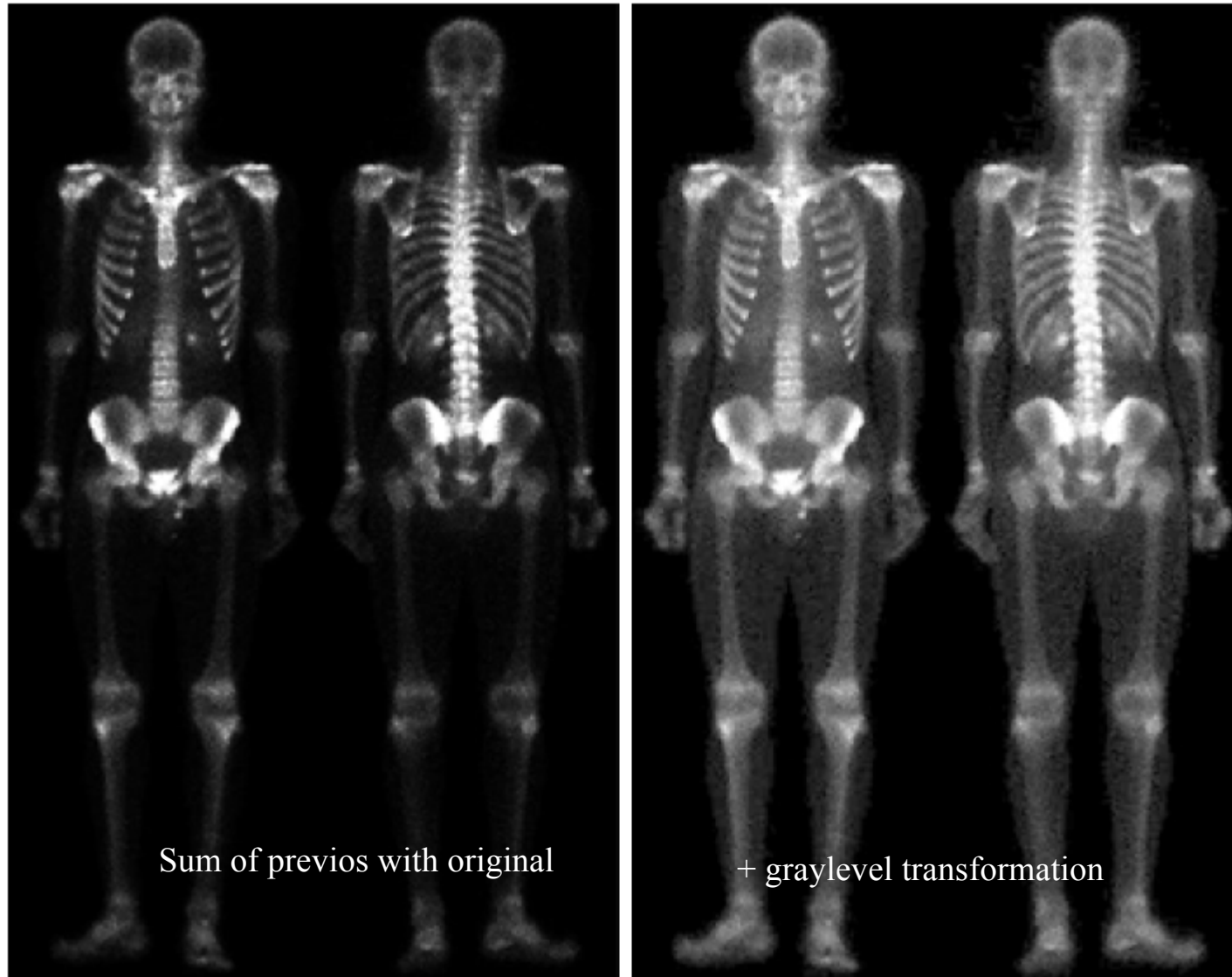
**FIGURE 3.46**

*(Continued)*

(e) Sobel image smoothed with a  $5 \times 5$  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.)

# Example



Sum of previos with original

+ graylevel transformation