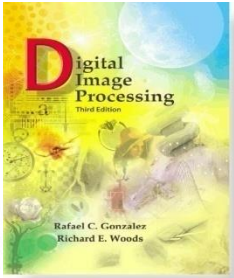


Intensity Transformations and Spatial Filtering

- Fundamentals of Spatial Filtering:

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

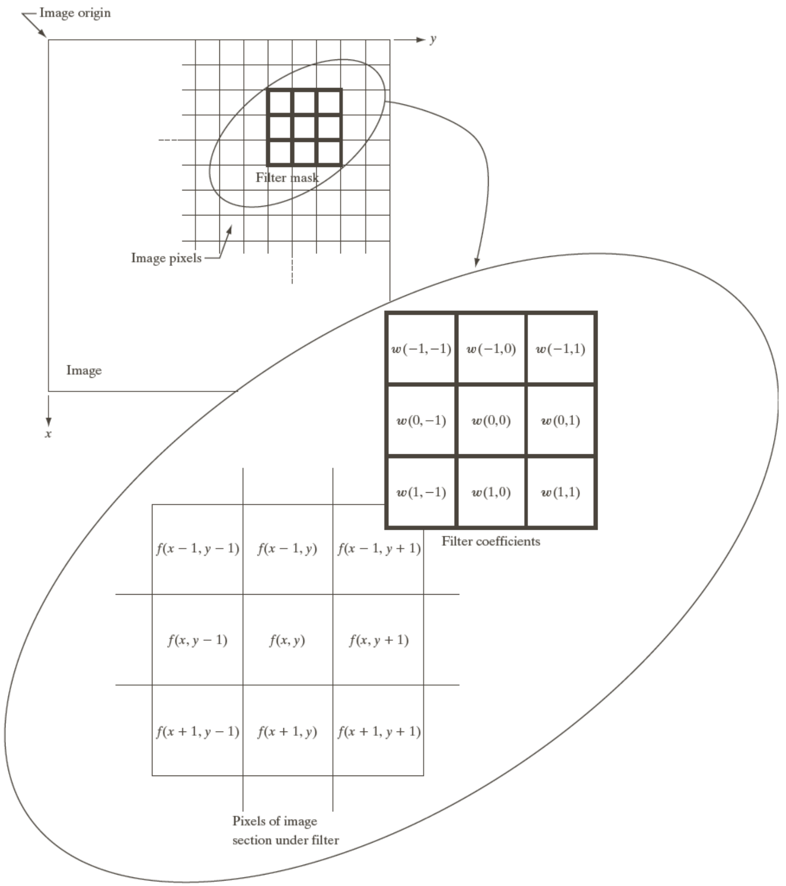
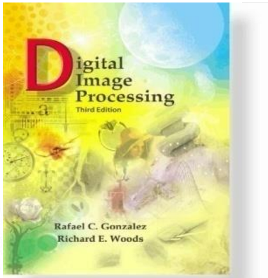


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.



Intensity Transformations and Spatial Filtering

- Spatial Correlation (☆) and Convolution (★)

Correlation

Origin f w
(a) 0 0 0 1 0 0 0 0 1 2 3 2 8

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
Starting position alignment

Zero padding
(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(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

(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

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Final position

Full correlation result
(g) 0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result
(h) 0 8 2 3 2 1 0 0

Convolution

Origin f w rotated 180°
0 0 0 1 0 0 0 0 8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

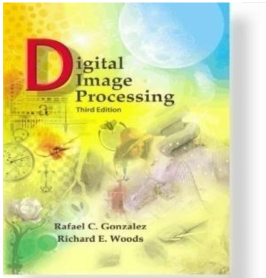
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result
0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result
0 1 2 3 2 8 0 0 (p)

FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.



Intensity Transformations and Spatial Filtering

- Spatial Correlation (☆) and Convolution (★)

Origin $f(x, y)$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} & w(x, y) & \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(a)

Padded f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(b)

Initial position for w

$$\begin{matrix} 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 8 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(c)

Full correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 8 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 5 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(d)

Cropped correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 8 & 7 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(e)

Rotated w

$$\begin{matrix} 9 & 8 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 5 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(f)

Full convolution result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(g)

Cropped convolution result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(h)

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Intensity Transformations and Spatial Filtering

- Spatial Filters

$$\begin{array}{|c|c|c|}\hline w_1 & w_2 & w_3 \\ \hline w_4 & w_5 & w_6 \\ \hline w_7 & w_8 & w_9 \\ \hline \end{array}$$

FIGURE 3.31 Another representation of a general 3×3 filter mask.

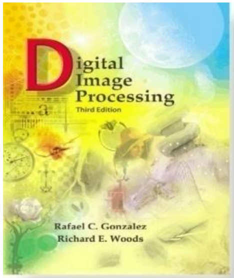
Linear Filters (averaging, lowpass)

Blur effect generates from average filter.
Low pass filter is one in which constant change is passed or the filter which passes low frequency

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

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

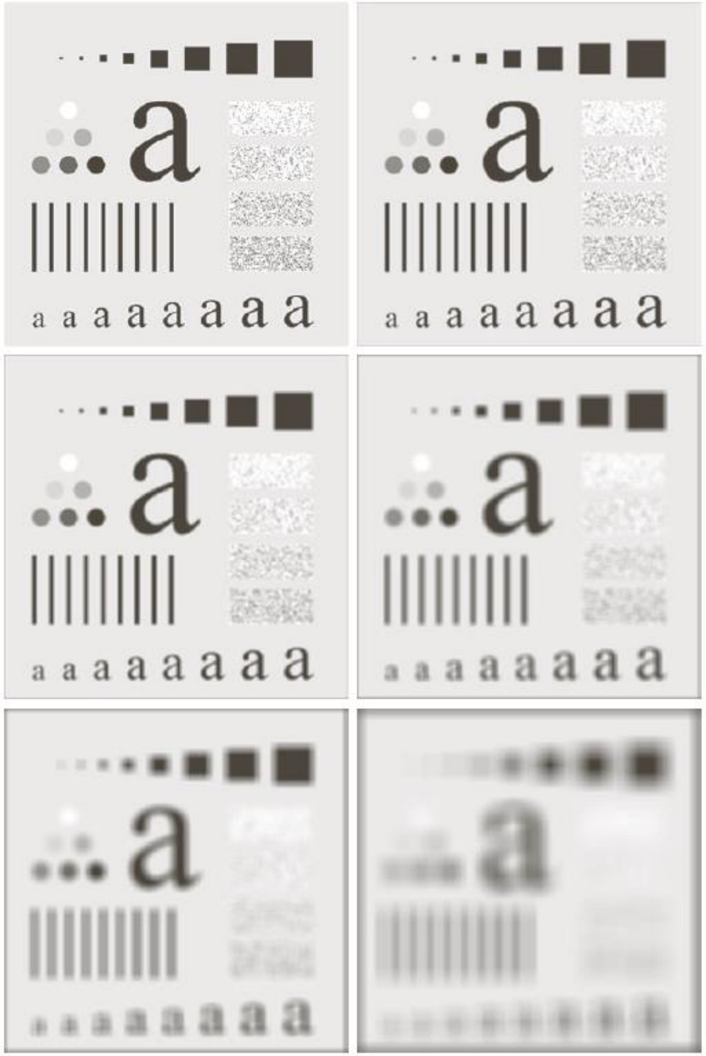


Intensity Transformations and Spatial Filtering

Square averaging Filter:

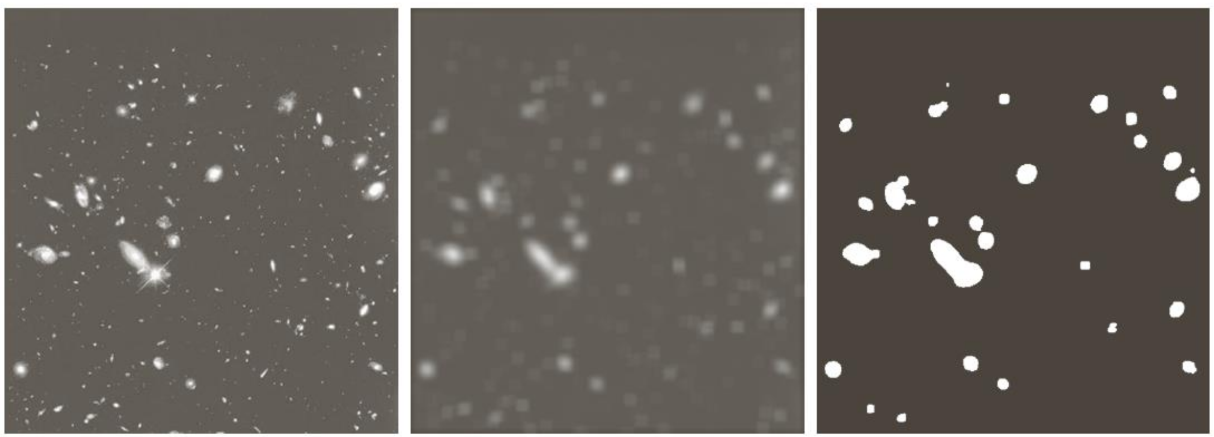
FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f



Increasing kernel/mask size makes image more blur depending upon the dimension of kernel/mask. But it is recommended that dimenshion of mask must be odd like 3x3, 5x5. Because center can be measured from mask of odd dimenshion and convolution and correlation becomes easy to apply.

- Blurring Usage:
  - Delete unwanted (small) subjects.



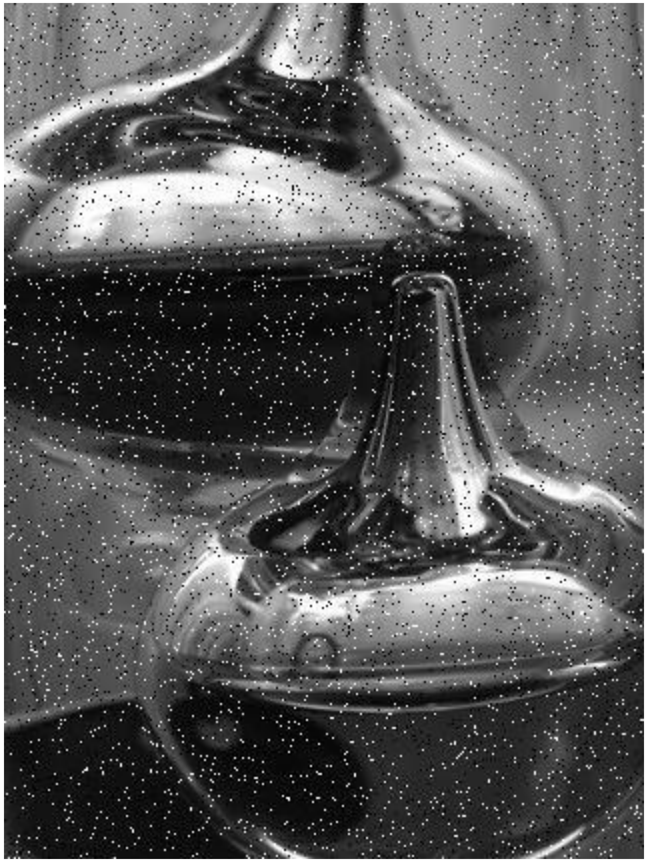
a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

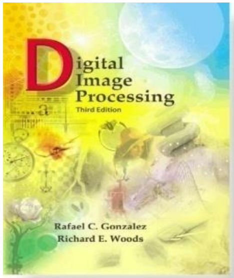
Thresholding is the process of converting color or any other image to binary image depending upon the type of image. For example, if we have a grey scale image whose range is 0-255, values below the threshold 127 will be assigned black and greater the 127 will be assigned white. Sometimes, white color is said to be foreground class and black as background class.

Blurring is also known as smoothing process. It is used to remove irrelevant detail from image.

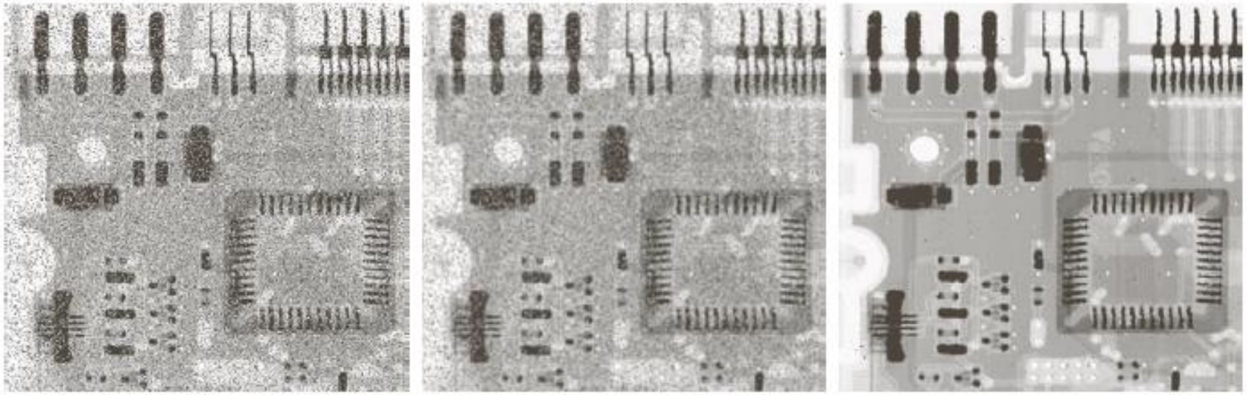
- Order Statistics Filters:
  - Impulsive noise:
    - Mono Level : Salt, Pepper noises
    - Bi Level: Salt-Pepper noises
  - Filter
    - Median
    - Max
    - Min



For median, sort the pixel intensities and find mid using formula:
For odd length of total pixels: (n+1)/2
For even length of total pixels: ((n/2)+(n+1/2))/2



- Example:



a b c

FIGURE 3.35 (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.)

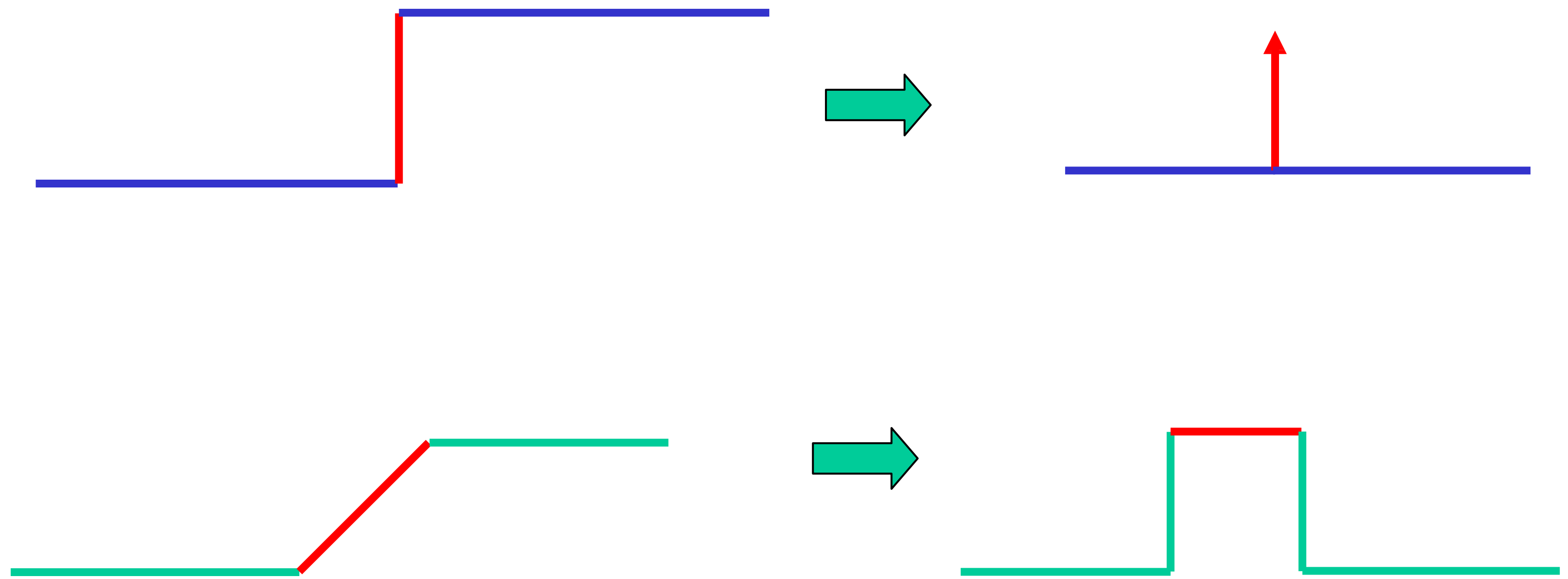
- Sharpening Spatial Filter
  - Highlights Intensity Transitions
  - First and Second order Derivatives

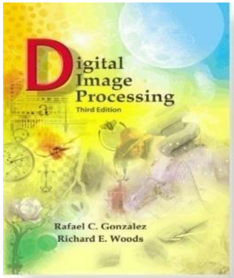
$$\frac{\partial f}{\partial x} \approx \begin{cases} f(x+1,y) - f(x,y) \\ f(x,y) - f(x-1,y) \\ 0.5\big(f(x+1,y) - f(x-1,y)\big) \end{cases}$$

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

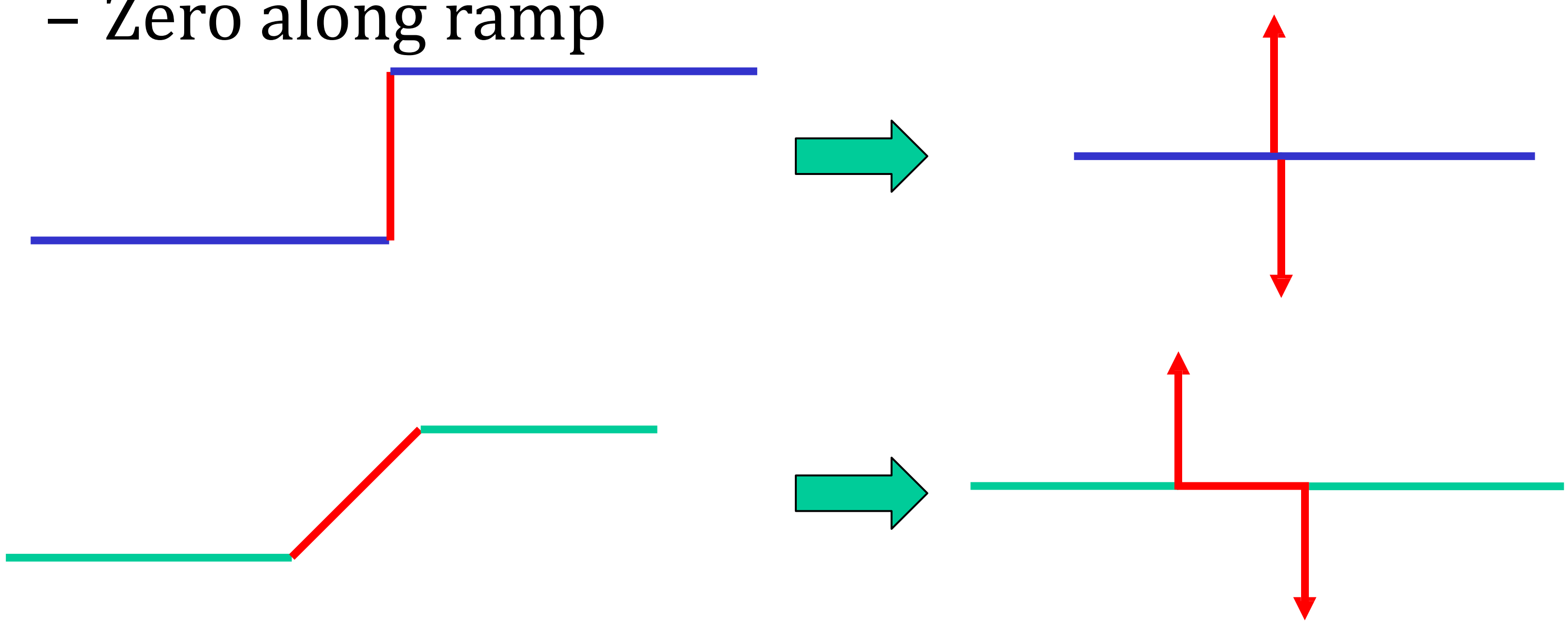
Ramp is a waveform that starts at a low value and gradually increases in amplitude over time. Ramps are often used as test signals to measure the frequency response of electronic circuits or to simulate the behavior of mechanical systems.

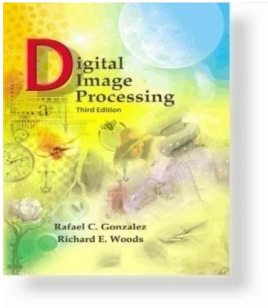
- First Order Derivative:
  - Zero in flat region
  - Non-zero at start of step/ramp region
  - Non-zero along ramp





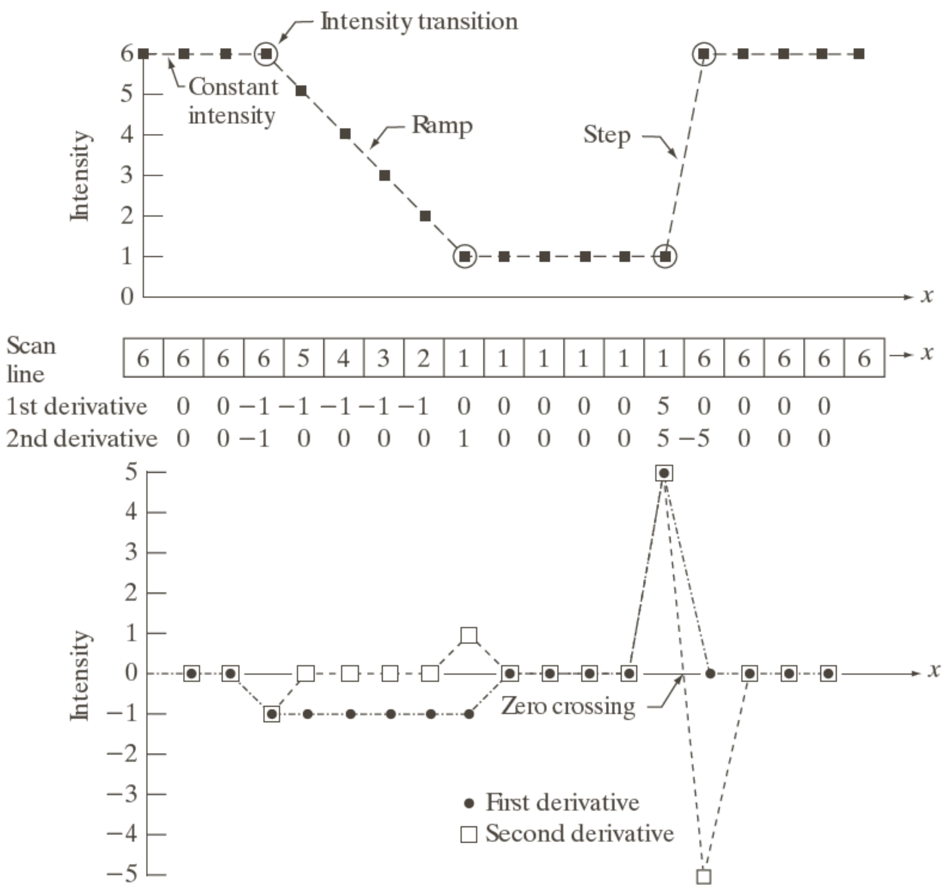
- Second Order Derivative:
  - Zero in flat region
  - Non-zero at start/end of step/ramp region
  - Zero along ramp





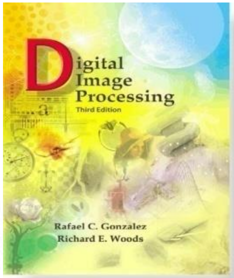
- Comparison:

1st order derivative:
f(x,y) - f(x-1,y)
2nd order derivative:
f(x-1,y)+f(x+1,y)-2f(x,y)



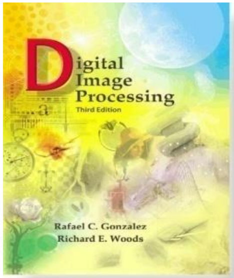
a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.



- 1st and 2nd Order Derivative Comparison:
  - First Derivative:
    - Thicker Edge;
    - Strong Response for step changes;
  - Second Derivative:
    - Strong response for fine details and isolated points;
    - Double response at step changes.

Edge detection: The first order derivative (gradient) of an image can be used to detect edges, which are areas of rapid intensity change in an image. The magnitude and direction of the gradient can be used to identify the location and orientation of edges in an image.
Image enhancement: The first and second order derivatives of an image can be used to enhance image contrast and improve image quality. The Laplacian of an image, which is the second order derivative, can be used to sharpen an image and highlight fine details.
Feature extraction: The first and second order derivatives can be used to extract features from an image that can be used for image classification, object recognition, and other computer vision tasks. For example, the Hessian matrix, which is the second order derivative of an image, can be used to extract features such as corners and blobs.
Noise reduction: The first and second order derivatives can be used to filter noise from an image. For example, the gradient of an image can be used to detect noisy areas of an image, which can then be smoothed using a filter.



- Laplacian as an isotropic Enhancer:

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

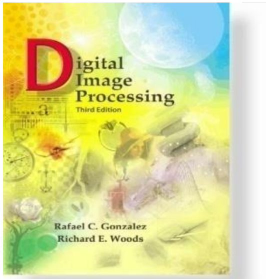
2 times 2nd order derivative to sharp image 4 times on a single operation.

- Discrete Implementation:

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad 90^\circ \; \textit{isotropic} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad 45^\circ \; \textit{isotropic}$$

The term "isotropic" is often used to describe filters or operations that treat all directions in an image equally. For example, an isotropic filter would produce the same result regardless of the orientation of the features in the image, while an anisotropic filter would produce different results depending on the orientation of the features.



- Laplacian Masks

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

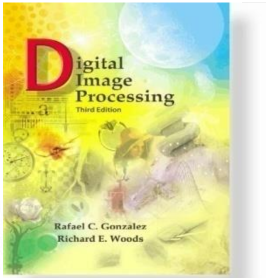
(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Practically use:



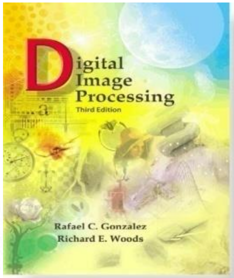
- Background Recovering:

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

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & +5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad 90^\circ \; isotropic \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad 45^\circ \; isotropic$$

Background recovery is the process of separating the background signal from the foreground signal in an image. This is often necessary in applications such as image segmentation and object detection, where the foreground objects are of interest and need to be isolated from the background.

One common approach for background recovery is to use a constant value c to represent the background signal. This assumes that the background signal is relatively uniform across the image, and that it can be represented by a single constant value.



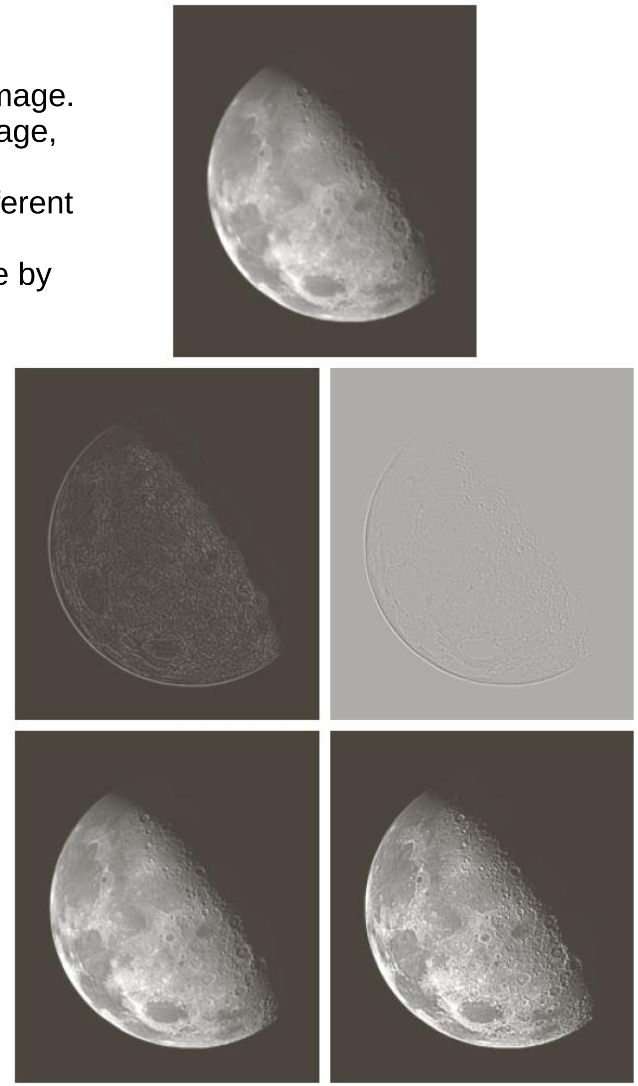
Intensity Transformations and Spatial Filtering

- Example:

Scaling refers to the process of changing the size of an image.
Scaling can either increase or decrease the size of an image, and can be done for various reasons, such as:
Display: Scaling can be used to display an image at a different size on a screen or print it at a different size.
Compression: Scaling can be used to compress an image by reducing its size, which can help in reducing storage and transmission requirements.
Up-sampling, down-sampling can be performed using interpolation method nearest Neighbour, bi-linear, bi-cubic or any other one.

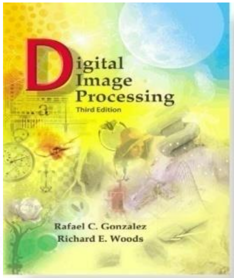
Non-Scaled and scaled Laplacian

Sharpened using 90º and 45º degree isotropic Laplacian



a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)



- Unsharp Masking and High-Boost Filtering:

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

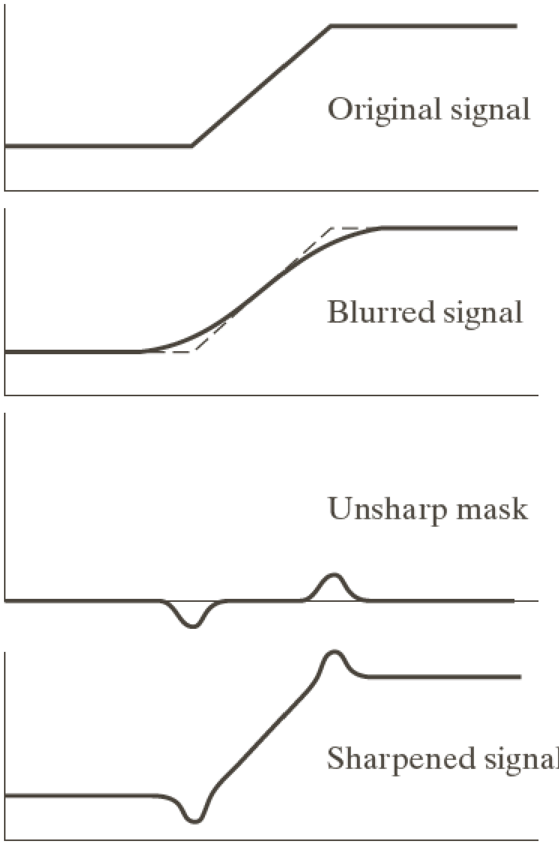
$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

- $k \geq 0$
  - $k=1$: Unsharp Masking
  - $k>1$: High Boost

Subtracting blurred image from original means removing low-frequency details from image and only high frequency details are visible which contains more sharp edges. Blurred image
can be gathered using average/low pass filtering.

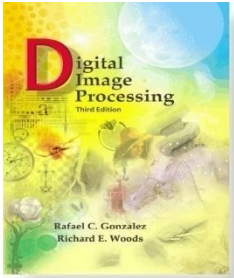
- Another mask:
  - Laplacian and any highpass filter

- One Dimensional Illustration



a b c d

FIGURE 3.39 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).



- Two Dimensional Example

The variance of intensity values in an image is a measure of the amount of contrast or variation in the image. Images with high variance in intensity have a large range of pixel values, indicating a high degree of contrast or variation, while images with low variance have a smaller range of pixel values, indicating less contrast or variation.

The variance of intensity can affect an image in several ways, including:

Image Quality: Images with high variance in intensity tend to have better image quality and appear sharper, since they have a wider range of pixel values that capture more detail and contrast. Low variance images, on the other hand, may appear flat or lack detail.

Noise: Images with high variance in intensity are more resistant to noise, since the noise will be relatively small compared to the overall range of pixel values. Low variance images, however, may be more prone to noise, as the noise may be a larger proportion of the overall range of pixel values.

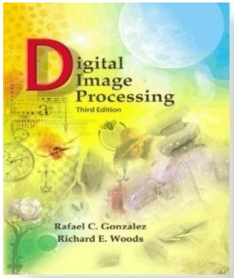
Image Analysis: The variance of intensity can be used as a feature in image analysis tasks such as object recognition, segmentation, and classification. High variance images may contain more distinctive features that are easier to recognize, while low variance images may require more sophisticated algorithms to extract meaningful

Blurred, 5×5, σ=3

Unsharp mask

Unsharp masking , k=1

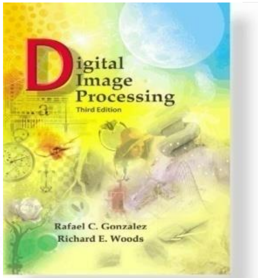
Highboost, k=4.5



- First Derivative - Gradient:

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

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



- Discrete Implementation

Roberts Cross Gradient

Takes 2x2 filter

$$G_x = (z_9 - z_5) \qquad G_y = (z_8 - z_6)$$

Sobel Gradient

Takes 3x3 filter

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Gradient refers to the rate of change of pixel intensity.
Different gradient techniques:
1) Sobel operator 2) Prewitt operator 3) Robert cross
4) Laplacian operator

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(b)

−1	0
0	1

(c)

0	−1
1	0

(d)

−1	−2	−1
0	0	0
1	2	1

(e)

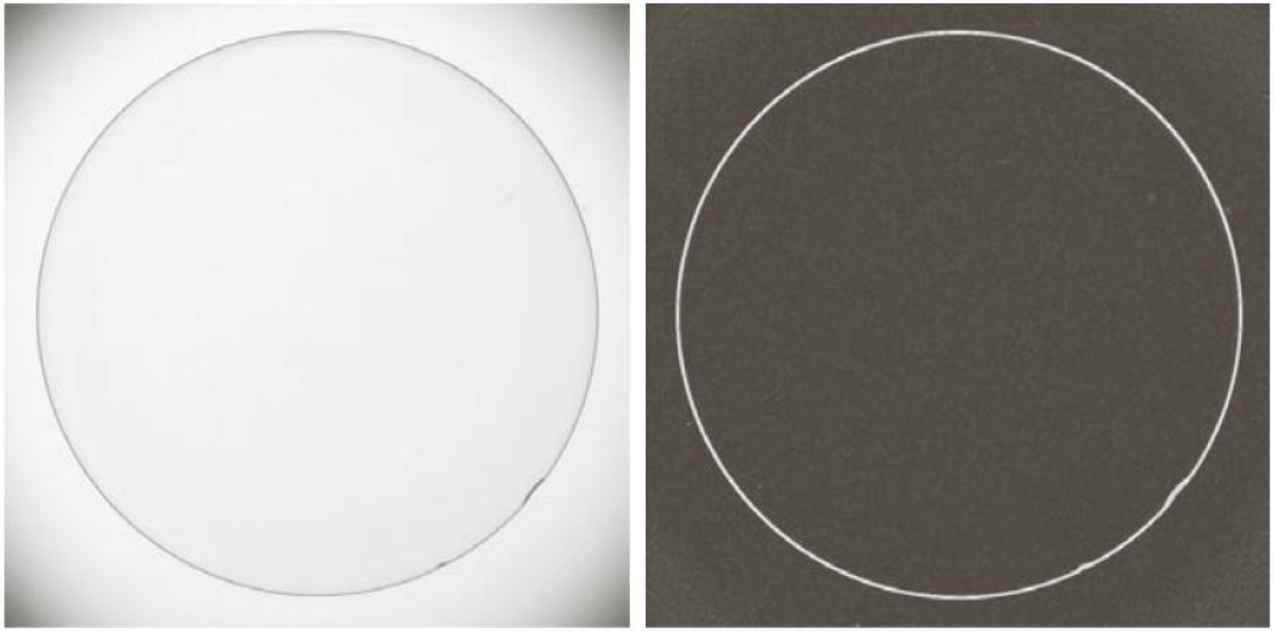
−1	0	1
−2	0	2
−1	0	1

a
b c
d e

FIGURE 3.41 A 3 × 3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Intensity Transformations and Spatial Filtering

- Example (Sobel Mask):



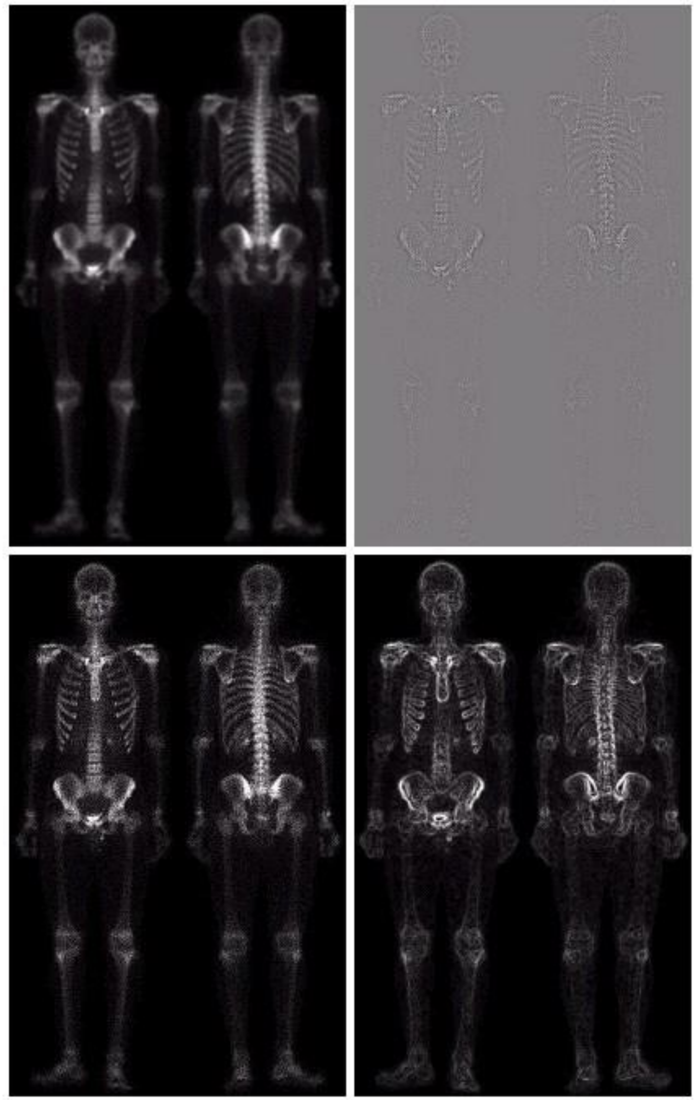
a b

FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

Sobel is useful in reducing noice from image and edge detection.

Intensity Transformations and Spatial Filtering

- Combination:



Bone Scan

Scaled Laplacian

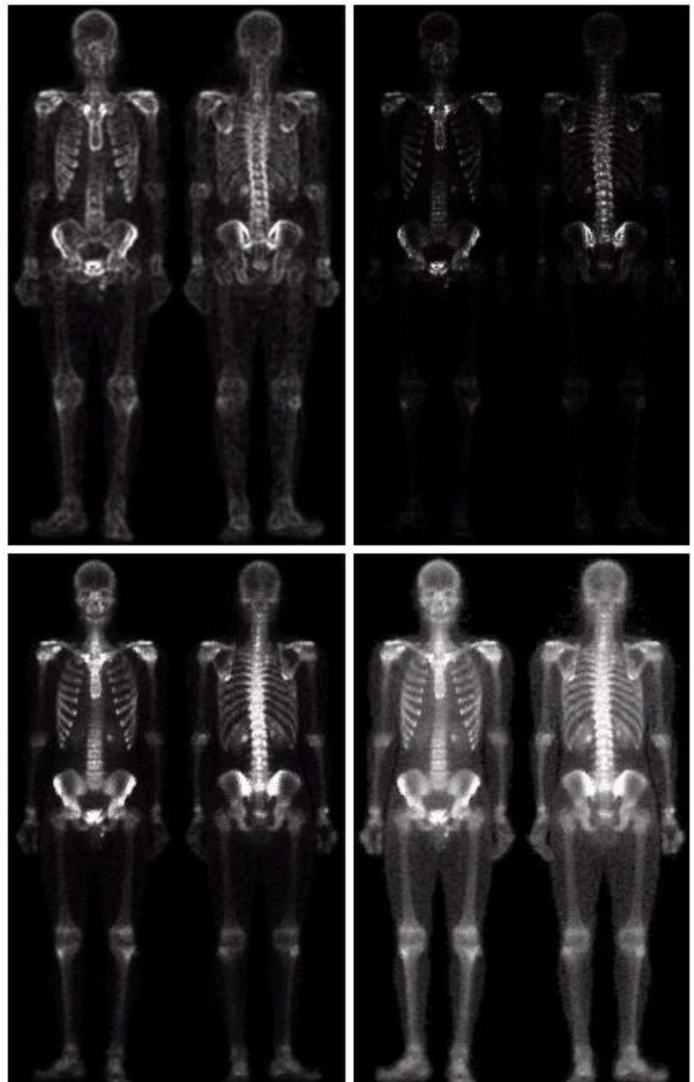
Original+Laplacian

Sobel of Original

- Combination:

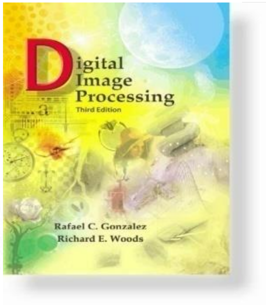
Smoothed Sobel

(Orig. + L.)*S.Sobel



Orig.+ (Orig.+L.)*S.Sobel

Apply Power-Law



Chapter # 3

- # Assignment statement:

  Apply the concepts of image enhancement in spatial domain to solve exercise problems related to spatial filtering for smoothing and sharpening and histogram processing.

- # End problems:

  All questions except: 2,3,8,9,11,14,17,22,27,30-34

Benefits of using Sobel operator:
Faster computation: The Roberts Cross operator uses smaller filters and requires less computation than the Sobel operator, making it faster and more suitable for real-time applications, such as video processing or robotics.

Simplicity: The Roberts Cross operator is simpler to implement and understand than the Sobel operator, since it uses only two 2x2 filters.

Diagonal edges: The Roberts Cross operator is more suitable for detecting diagonal edges, since its filters are applied diagonally to the image.