SYDE 575: Introduction to Image Processing

Image Enhancement and
Restoration in Spatial Domain

Chapter 3

Spatial Filtering

- Recall 2D discrete convolution

$$g[m,n] = f[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} f[i,j] \cdot h[m-i, n-j]$$

output input impulse function

- Impulse function *h[m,n]* can be viewed as a spatial filter for an input image *f[m,n]* to produce output image *g[m,n]*

Spatial Filtering

- Let us represent *h[m,n]* as a 2D convolution mask

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

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

Source: Gonzalez and Woods

Spatial Filtering

- Spatial filtering of an image *f* with 2D convolution mask *w* of size *m x n* can be expressed as

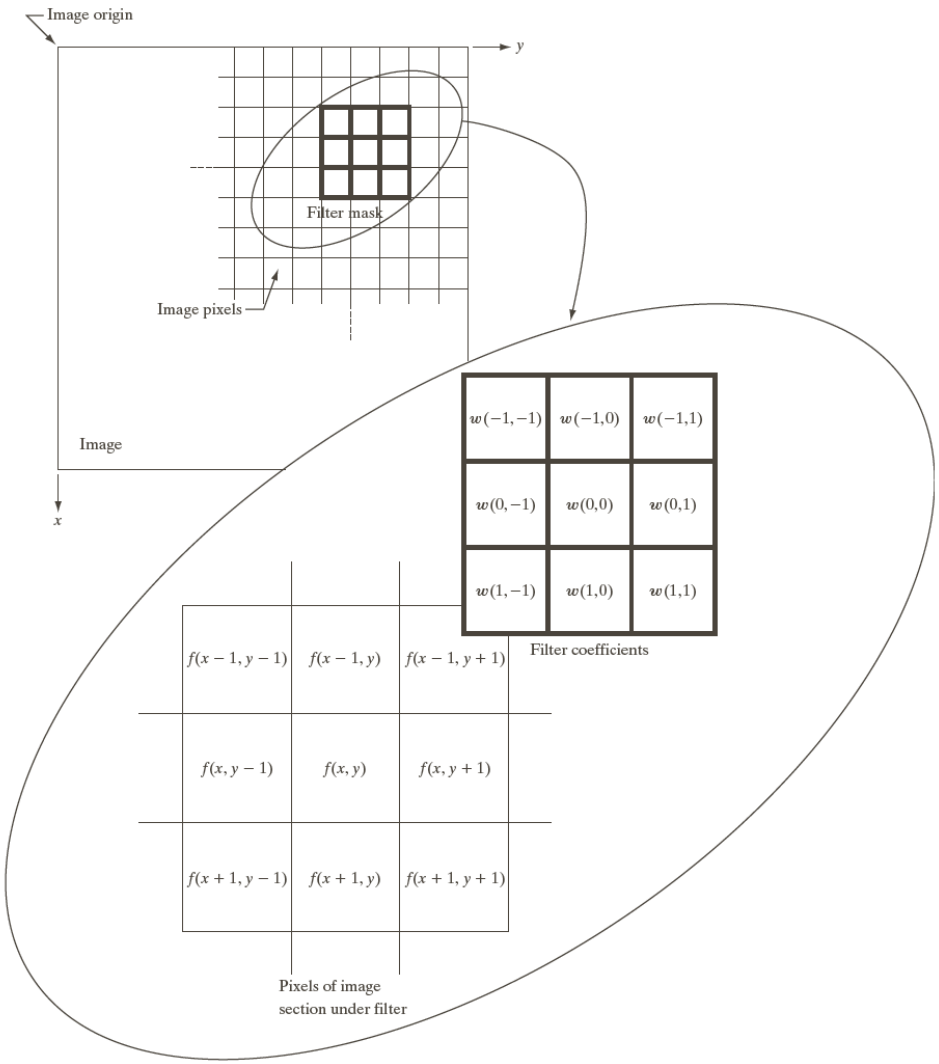
$$g(x,y) = \sum_{s=-(m-1)/2}^{(m-1)/2} \sum_{t=-(n-1)/2}^{(n-1)/2} w(s,t) f(x+s, y+t)$$

output

convolution mask

input

Spatial Filtering



Source: Gonzalez and Woods

Spatial Filtering: Noise Reduction

- Simple image noise degradation models
- Additive noise model

$$g(x,y) = f(x,y) + \eta(x,y)$$

output input noise

- Multiplicative noise model

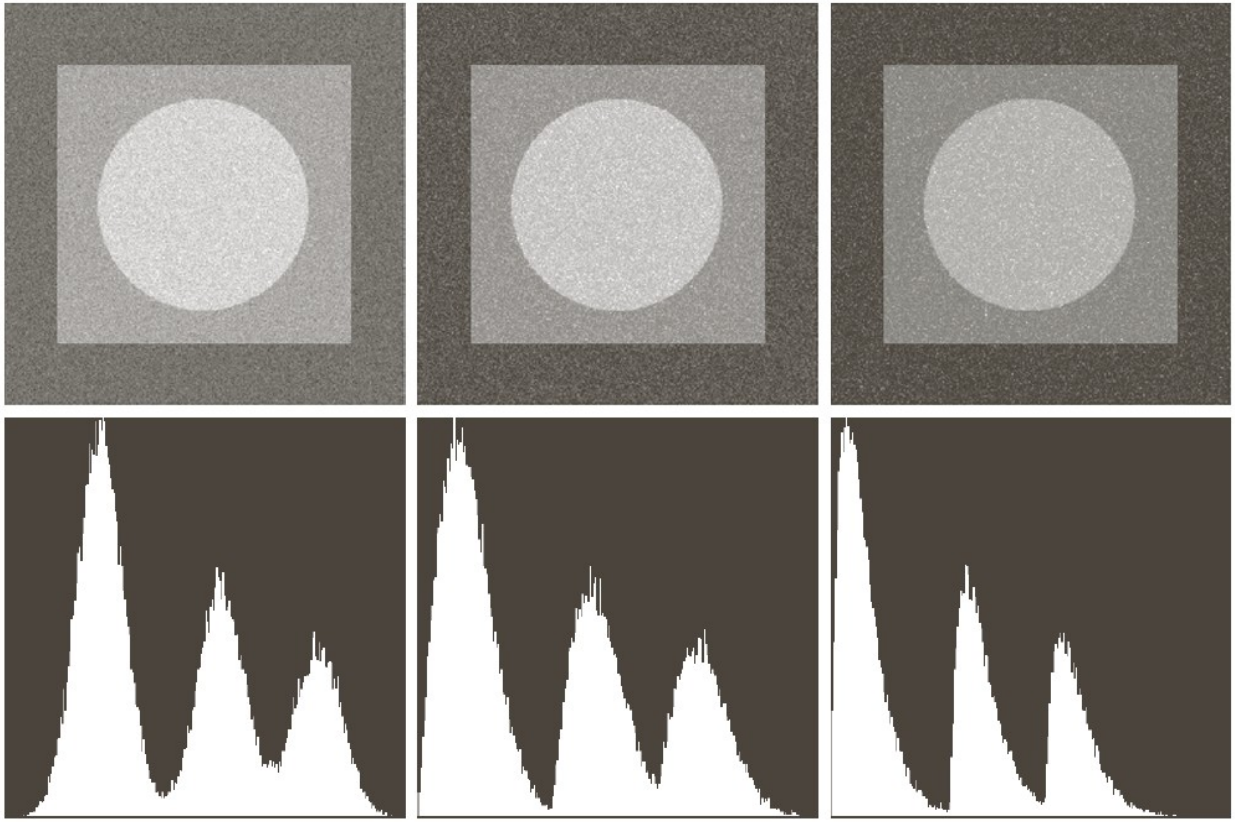
$$g(x,y) = f(x,y)\eta(x,y)$$

output input noise

Spatial Filtering: Noise Models

- Principal sources of digital image noise arise during image acquisition and/or transmission
  - Thermal noise
  - Shot noise
  - Corruption during transmission over network
- In simple noise models, noise is assumed to be independent, uncorrelated, and white

Some Noise Models



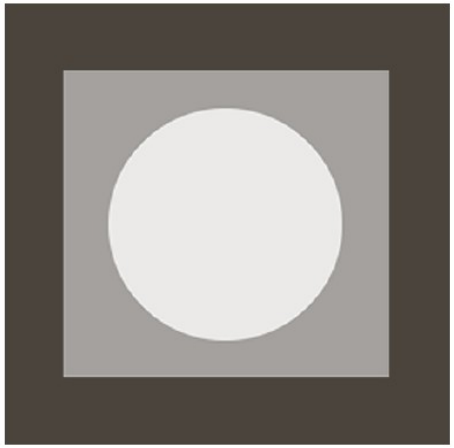
Gaussian Rayleigh Gamma

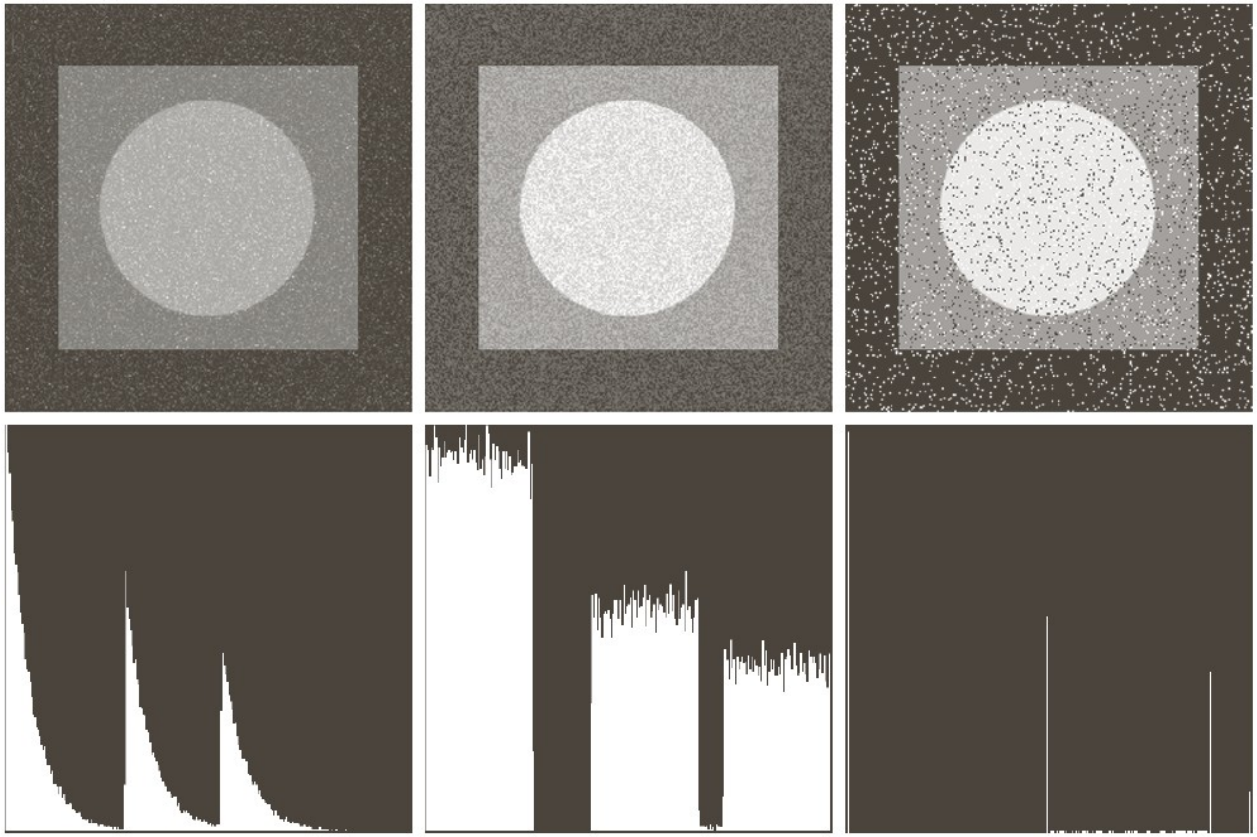
a b c
d e f

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Source: Gonzalez and Woods

Some Noise Models





Exponential Uniform Salt & Pepper

g h i
j k l

FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Source: Gonzalez and Woods

Noise reduction by Image Averaging

- Assume noise is additive and Gaussian distributed with zero mean

$$n \sim N(0, \sigma)$$

- Suppose we take the average of *q* number of noise samples $n_1, n_2, ..., n_q$ at a point in the image

$$m = \frac{1}{q} \sum_{k=i}^{q} n_k$$

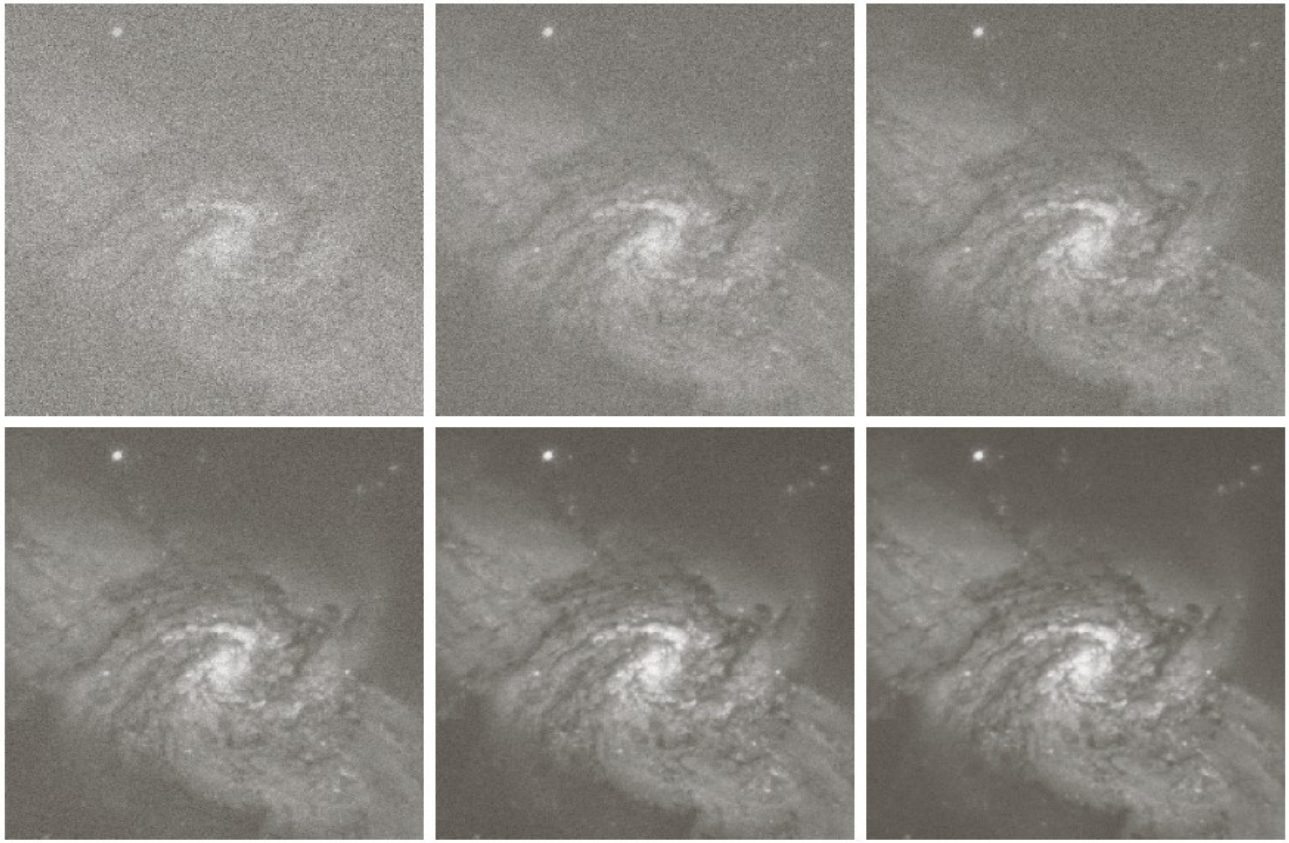
Noise reduction by Image Averaging

- Given an infinite number of noise samples, the average approaches the mean of the distribution, which in this case is 0.

$$g(x,y) = \frac{1}{q}\sum_{k=1}^{q} f(x,y) + \frac{1}{q}\sum_{k=1}^{q} n_k(x,y)$$

$$\text{As } q \to \infty, \ \frac{1}{q}\sum_{k=1}^{q} n_k(x,y) \to 0, \ g(x,y) \to f(x,y)$$

Example



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Source: Gonzalez and Woods

Noise reduction by Spatial Filtering

- Image averaging takes advantage of information redundancy from the individual images to reduce noise
- Not always possible to acquire so many images!
- Alternative option: Take advantage of information redundancy from different pixels within the same image to reduce noise

Averaging Filter

- Instead of averaging between images, we can average neighboring pixels

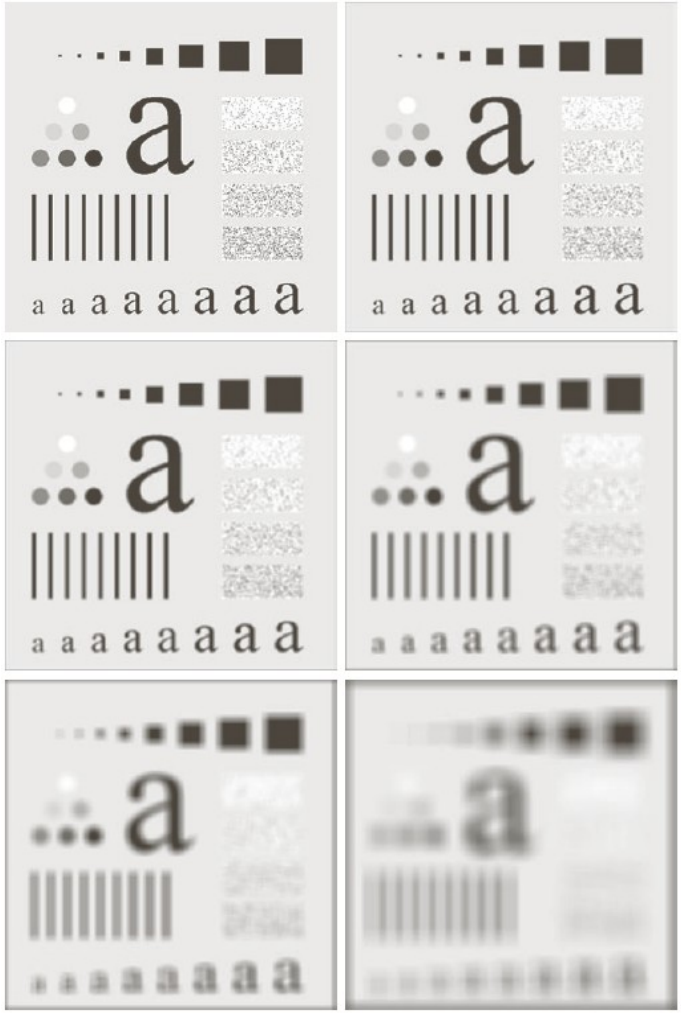
$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

- Also provides a blurring effect



Source: Gonzalez and Woods

Weighted Average Filter

- Problem: Simple averaging of neighboring pixels lead to over-smoothing
- Possible solution: Instead of weighting all neighboring pixels equally, assign higher weights to pixels that are closer to the pixel being convolved

Weighted Average Filter

weight input

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

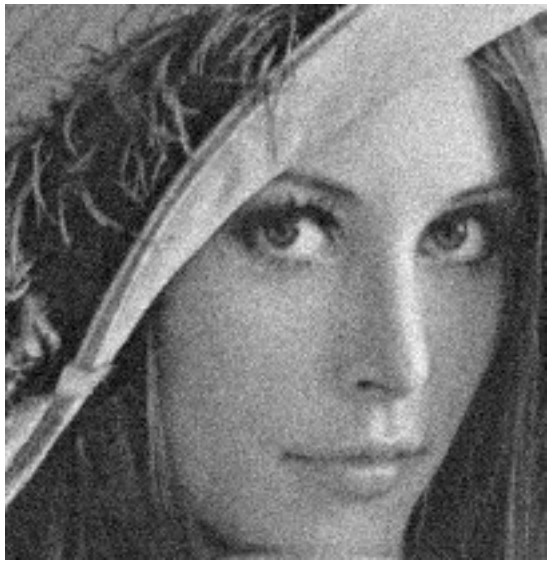
output

Weighted Averaging Filter: Example

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Source: Gonzalez and Woods

Weighted Averaging Filter: Example



Noisy

Average

Weighted Average

Order-Statistic Filters

- Nonlinear spatial filters

$$H(af + bg) \neq aH(f) + bH(g)$$

- Steps
  - Order pixels within an area
  - Replace value of center pixel with value determined by ordering
- Best known example: median filters

Median Filter

- Provides good noise reduction for certain types of noise such as impulse noise
- Considerably less blurring than weighted averaging filter
- Forces a pixel to be like its neighbors

- Steps
  - Order pixels within an area
  - Replace value of center pixel with median value (half of all pixels have intensities greater than or equal to the median value)

Median Filter: Example

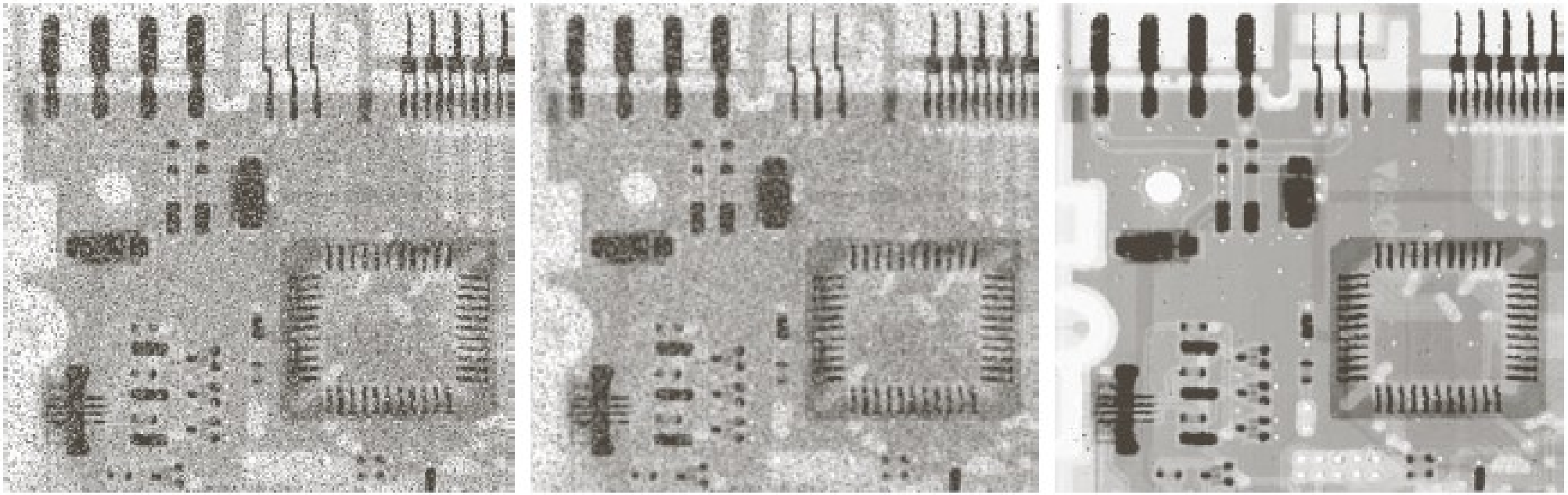
255	10	9
10	255	10
8	10	10

255	10	9
10	10	10
8	10	10

8	9	10	10	10	10	10	255	255

median=10

Median Filter: 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.)

Source: Gonzalez and Woods

Spatial Filtering: Sharpening

- Goal: highlight or enhance details in images
- Some applications:
  - Photo enhancement
  - Medical image visualization
  - Industrial defect detection

Spatial Filtering: Sharpening

- Goal: highlight or enhance details in images
- Some applications:
  - Photo enhancement
  - Medical image visualization
  - Industrial defect detection
- Basic principle:
  - Averaging (blurring) is analogous to integration
  - Therefore, logically, sharpening accomplished by differentiation

Derivatives of digital function

- First-order derivative

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

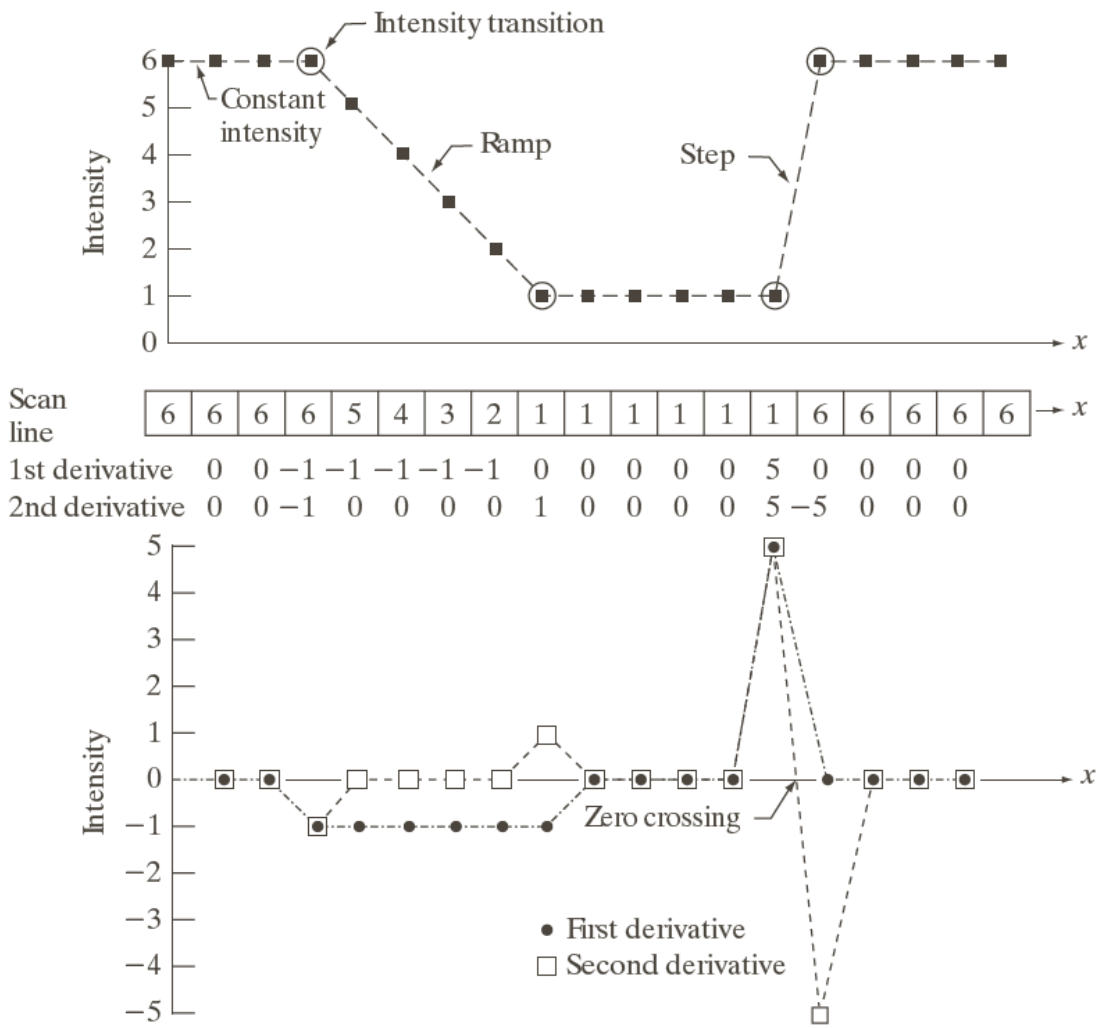
- Second-order derivative

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

Derivatives of digital function

- First-order derivatives generally produce thicker edges
- Second-order derivatives have stronger response to fine detail (e.g., thin lines and points)
- First-order derivatives have stronger response to step changes
- Second-order derivatives produce double response at step changes

Example



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.

Source: Gonzalez and Woods

Sharpening using Laplacian

- Second-order derivatives is better suited for most applications for sharpening
- How?
  - By constructing a filter based on discrete formulation of second-order derivatives
- Simplest isotropic derivative operator: Laplacian

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

How to create Laplacian filter

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

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

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

Laplacian Filter

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 3.37 (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.

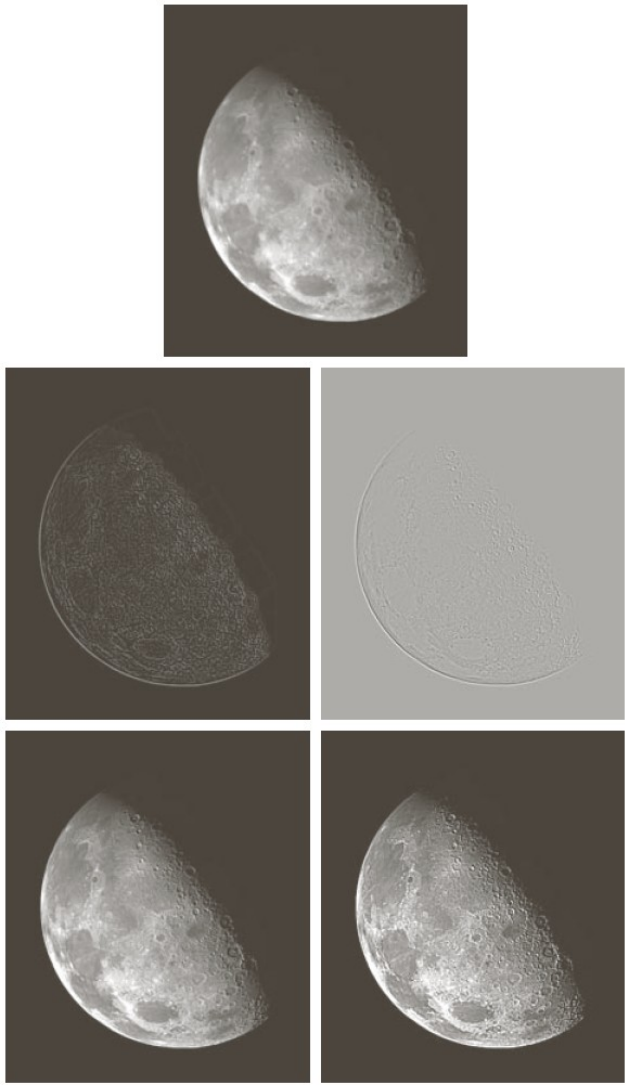
Source: Gonzalez and Woods

Sharpening using Laplacian

- Problem: While applying Laplacian highlights fine detail, it de-emphasizes smooth regions (e.g., background features)
- Results in featureless background with greyish fine details
- Solution: Add original image to recover background features

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y), & \text{negative filter center} \\ f(x,y) + \nabla^2 f(x,y), & \text{positive filter center} \end{cases}$$

Example



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.)

Source: Gonzalez and Woods

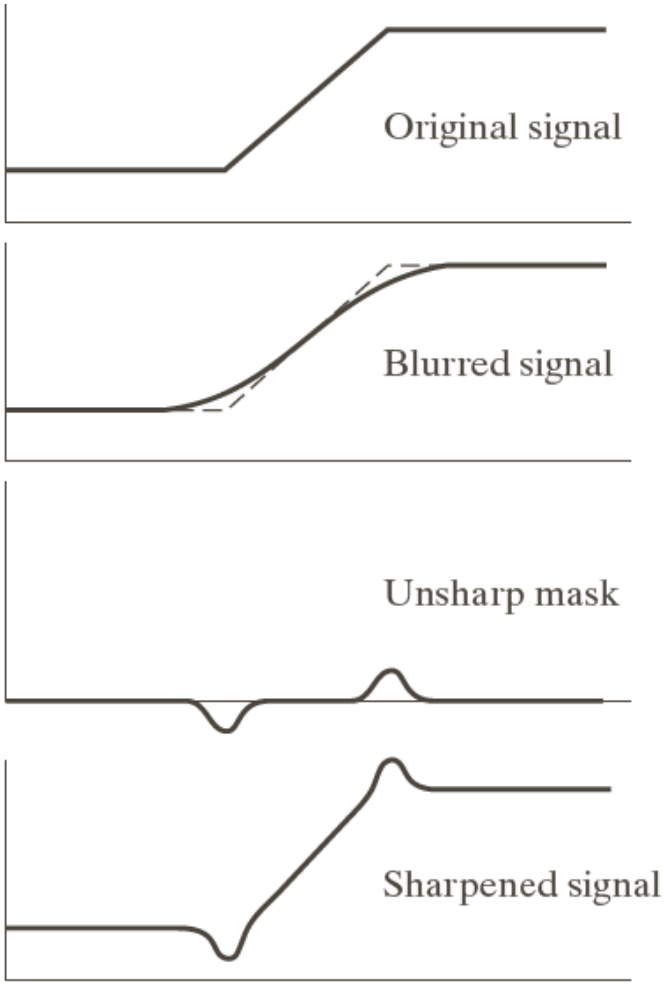
Sharpening using Unsharp Masking

- Process used for many years in publishing
- Subtract blurred version of image from the image itself to produce sharp image

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

output input Blurred output

Example



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).

Source: Gonzalez and Woods

Sharpening using High-boost Filtering

- Generalization of unsharp masking

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

output input Blurred output

- As *A* increases, contribution of sharpening decreases

Example

a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

Source: Gonzalez and Woods

First Derivatives for Enhancement

- Implemented as magnitude of gradient in image processing

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad \nabla f = \left[\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2 \right]^{1/2}$$

Implementation

- Problem: Expensive to implement directly
- Solution: Approximate using absolute values

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

  - Can be implemented using two 2x2 filters (Roberts cross-gradient operators)
  - Even filters difficult to implement, so 3x3 filters are used for approximation

Gradient Masks

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

−1	−2	−1
0	0	0
1	2	1

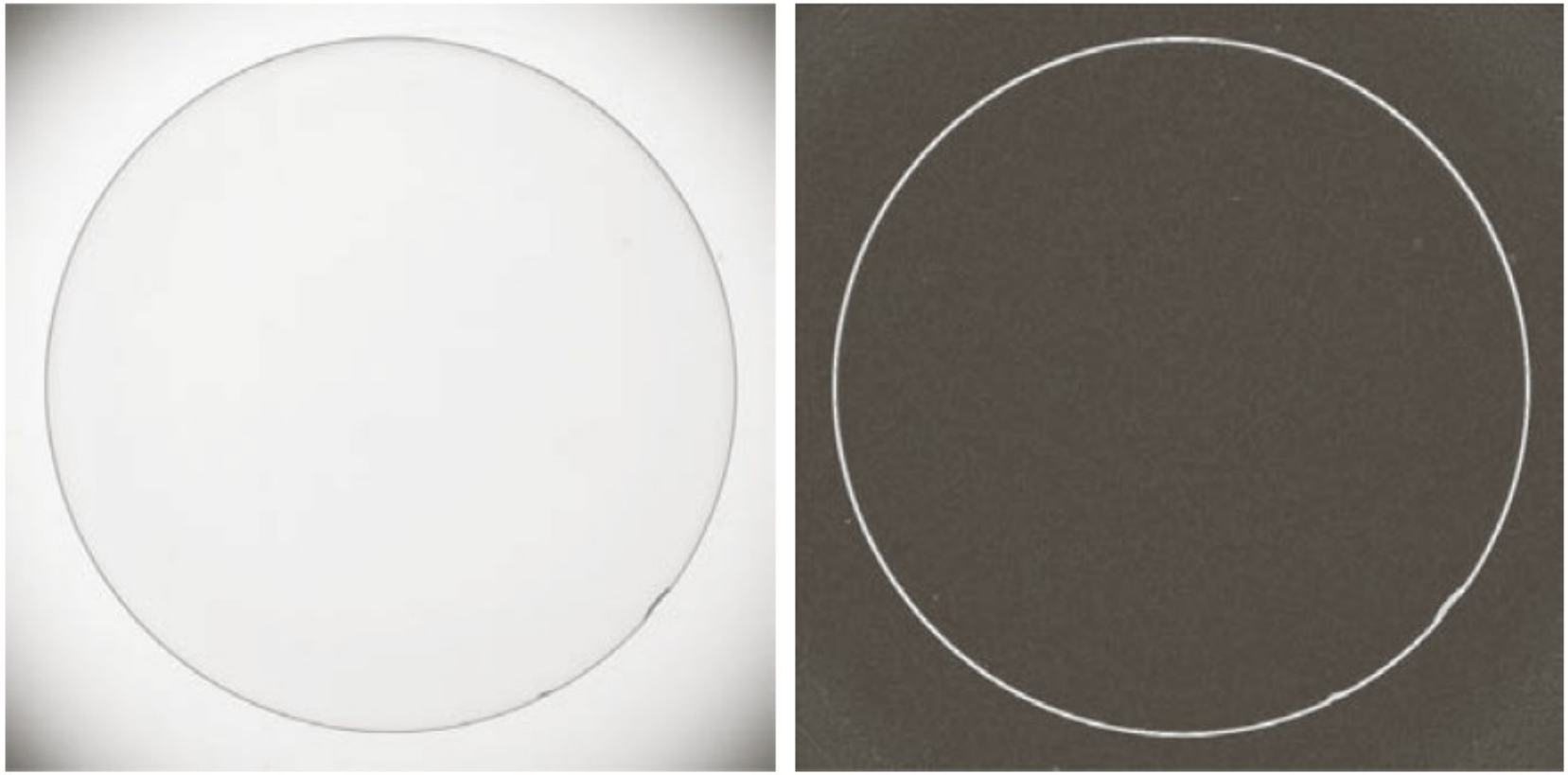
−1	0	1
−2	0	2
−1	0	1

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.

Source: Gonzalez and Woods

Example: Defect Detection



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.)

Source: Gonzalez and Woods