Digital Image Processing

Fundamentals

Thanh-Hai Tran

Electronics and Computer Engineering
School of Electronics and Telecommunications

Hanoi University of Science and Technology
1 Dai Co Viet - Hanoi - Vietnam

Outline

- Histogram revision
- Introduction to spatial filtering
- Smoothing filters
- Sharpening filters
- References

Histogram revision

- **Given an image as illustrated in the figure**
  - Compute the histogram of this image
  - Equalize the histogram and the converted image

4	4	4	4	4	4	4	0
4	5	5	5	5	5	4	0
4	5	6	6	6	5	4	0
4	5	6	7	6	5	4	0
4	5	6	6	6	5	4	0
4	5	5	5	5	5	4	0
4	4	4	4	4	4	4	0
4	4	4	4	4	4	4	0

Histogram implementation

- Given an image represented as 2D array I[H][W]
- Compute the histogram of this image and normalize it

Introduction

- Filtering is one of the principal tools used in DIP for a broad spectrum of applications
- It is highly advisable that you develop a solid understanding of these concepts.
- Filter is borrowed from frequency domain processing
- Filter refers to accepting (passing) or rejecting certain frequency components
- Spatial filtering is applying filter on spatial domain

The mechanics of spatial filter

- **A spatial filter**
  - Neighborhood: small rectangle
  - Pre-defined operation: linear and non-linear
- **Filtering creates a new pixel**
  - with coordinates equal to the coordinates of the center of the neighborhood,
  - and whose value is the result of the filtering operation
- **A processed (filtered) image is generated as the center of the filter visits each pixel in the input image**

The mechanics of spatial filter

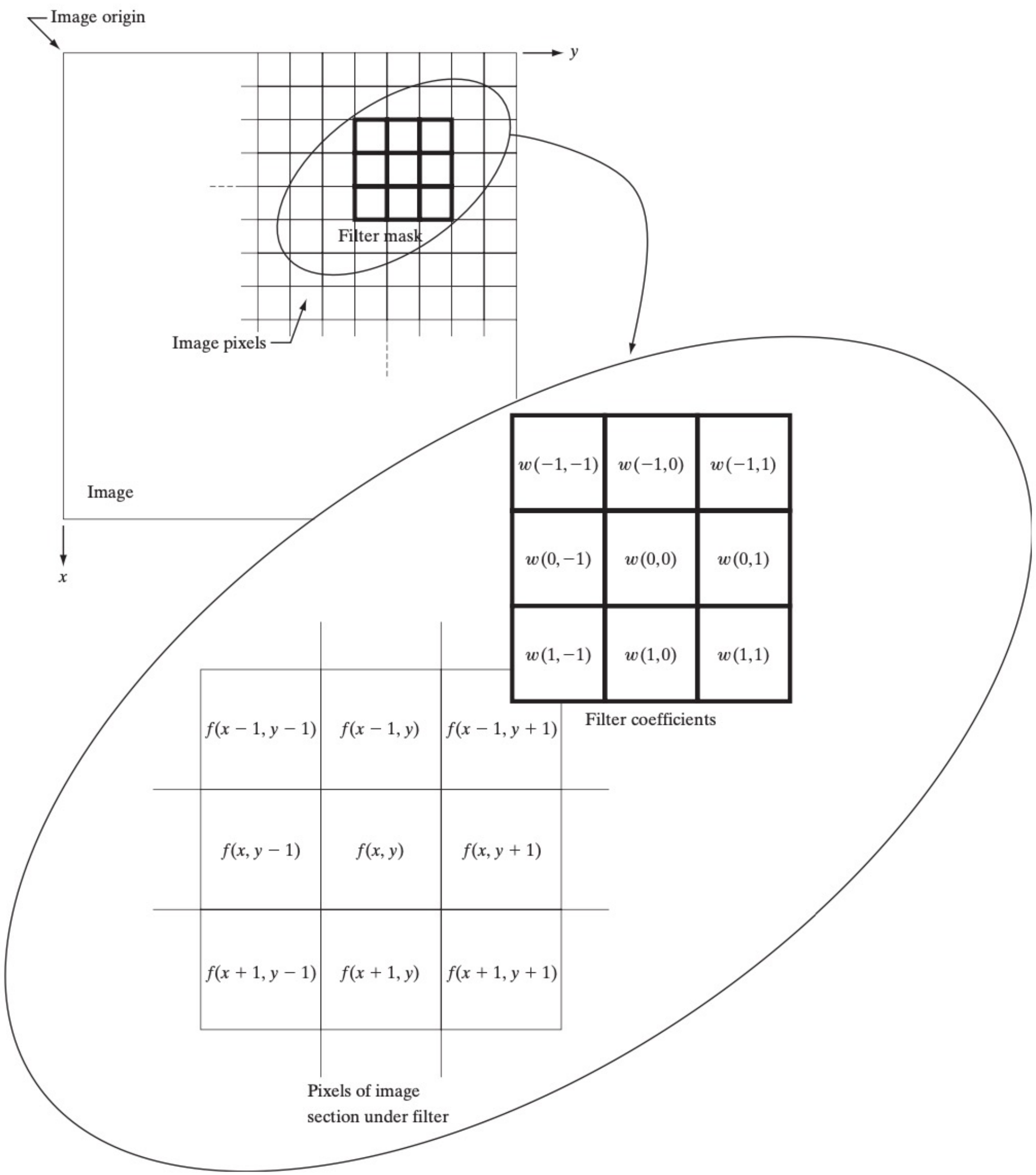
- **Notation:**
  - f(x,y) is the original image
  - w(x,y) is a filter mask
  - g(x,y) is the filtered image
- **At any point (*x*, *y*) in the image, the response, g(x, y) is the sum of products of the filter coefficients and the image pixels encompassed by the filter**

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

- **A general case, w has MxN size then a = M/2, b = N/2**

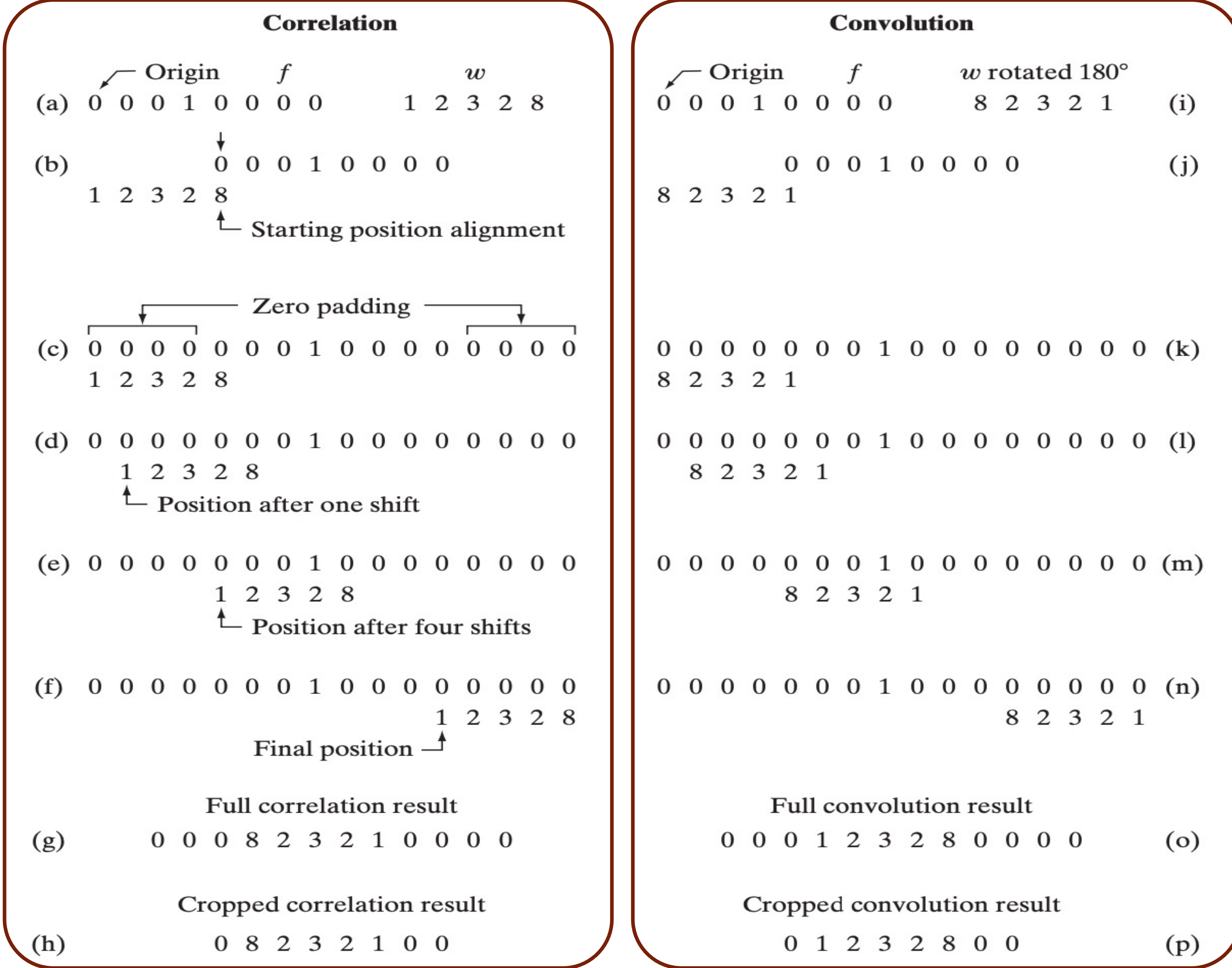
$$\boxed{g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x + s, y + t)}$$

The mechanics of spatial filter



Spatial correlation and convolution

- **Correlation:** is the process of moving a filter mask over the image and computing the sum of products at each location
- **Convolution:** the principle is the same but the filter is is first rotated by 180°


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)

Vector representation of linear filtering

- A filter of size mxn will be represented as

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$
$$= \sum_{k=1}^{mn} w_k z_k$$
$$= \mathbf{w}^T \mathbf{z}$$

- When m = n = 3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$
$$= \sum_{k=1}^{9} w_k z_k$$
$$= \mathbf{w}^T \mathbf{z}$$

Generating Spatial Filter Masks

- **Generating an mxn linear spatial filter** requires that we specify mxn mask coefficients.
- These coefficients are selected based on what the filter is supposed to do,
- All we can do with linear filtering is to implement **a sum of products**
- **For example:**
  - Replace the pixels in an image by the average intensity of a 3 * 3 neighborhood centered on those pixels.
  - The average value at any location (*x*, *y*) in the image is the sum of the nine intensity values in the 3 * 3 neighborhood centered on (*x*, *y*) divided by 9

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i \qquad w_i = 1/9$$

Generating Spatial Filter Masks

- In some applications, we have a continuous function of two variables, and the objective is to obtain a spatial filter mask based on that function
- For example, a Gaussian function of two variables has the basic form

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- **Solution:**
  - We sample it about its center
  - w1 = h(-1, -1), w2 = h(-1, 0), .. , w9 = h(1, 1)

Designing Gaussian Filters

- $g[\mathrm{i},\mathrm{j}] = c\, e^{-\frac{i^2+j^2}{2\sigma^2}}$
- **c is a normalizing constant**
- σ^2**= 2, n = 7**

$$h[i,j] = \frac{1}{1115}\left(f[i,j] \star g[i,j]\right)$$

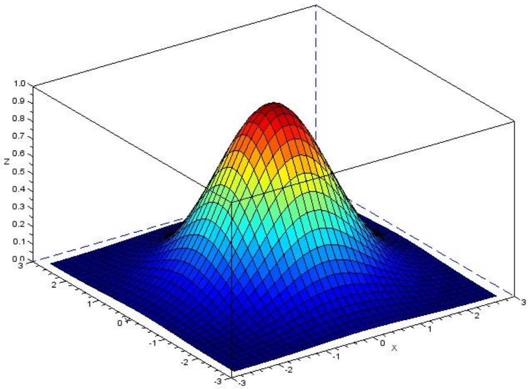
$$\sum_{i=-3}^{3}\sum_{j=-3}^{3} g[i,j] = 1115.$$

$[i,j]$	-3	-2	-1	0	1	2	3
-3	.011	.039	.082	.105	.082	.039	.011
-2	.039	.135	.287	.368	.287	.135	.039
-1	.082	.287	.606	.779	.606	.287	.082
0	.105	.368	.779	1.000	.779	.368	.105
1	.082	.287	.606	.779	.606	.287	.082
2	.039	.135	.287	.368	.287	.135	.039
3	.011	.039	.082	.105	.082	.039	.011

$[i,j]$	-3	-2	-1	0	1	2	3
-3	1	4	7	10	7	4	1
-2	4	12	26	33	26	12	4
-1	7	26	55	71	55	26	7
0	10	33	71	91	71	33	10
1	7	26	55	71	55	26	7
2	4	12	26	33	26	12	4
3	1	4	7	10	7	4	1

$$\frac{g[3,3]}{k} = e^{-\frac{(3^2+3^2)}{2.2}} = 0.011 \implies k = \frac{g[3,3]}{0.011} = \frac{1.0}{0.011} = 91.$$

Example of other Gaussian Filters



15 × 15 Gaussian mask

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

7 × 7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

Smoothing Spatial Filters

- Idea:
  - Replacing the value of every pixel in an image by the average of the intensity values in th neighborhood defined by mask center
  - It resuls in an image with reduced sharp transition
- Smoothing Spatial Filters are used for **blurring** or noise reduction
- These filters are called averaging filters / lowpass filters
- Side effect: edges are blurred too.

Smoothing Spatial Filters

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

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

Example

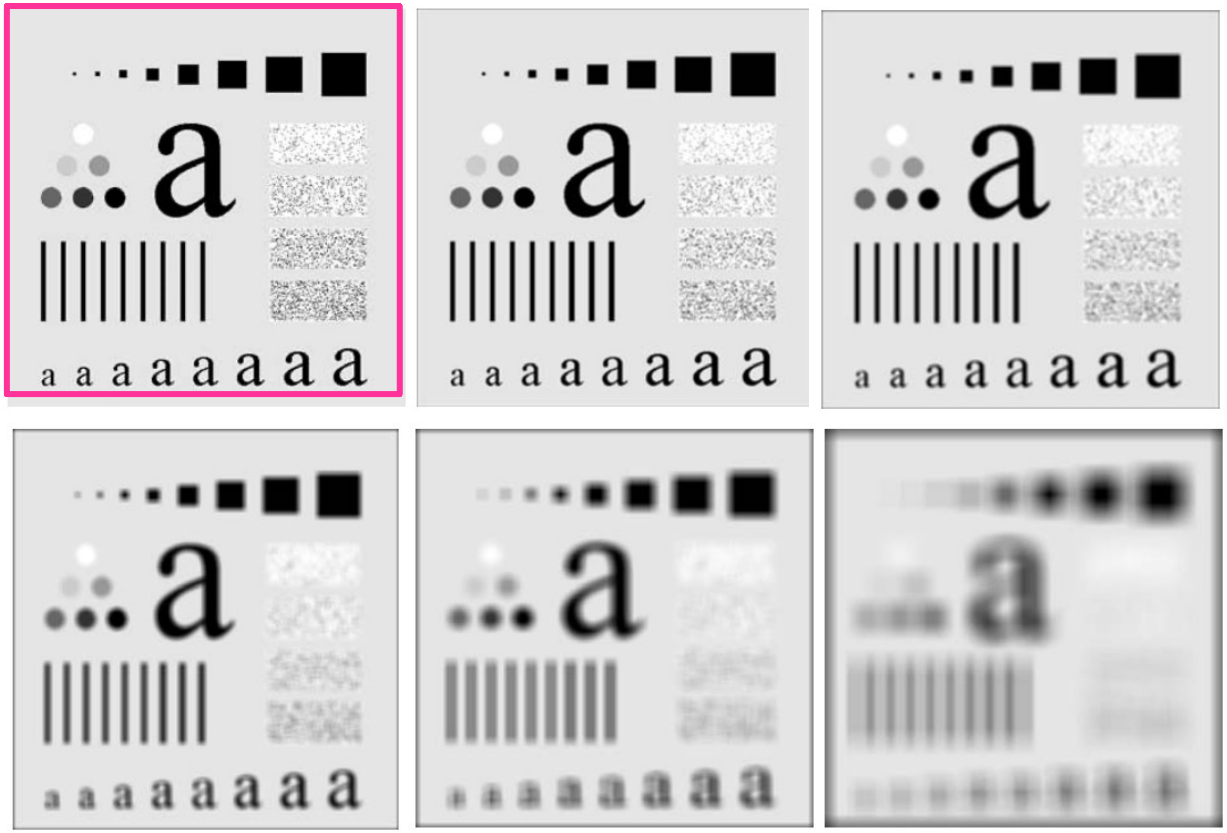
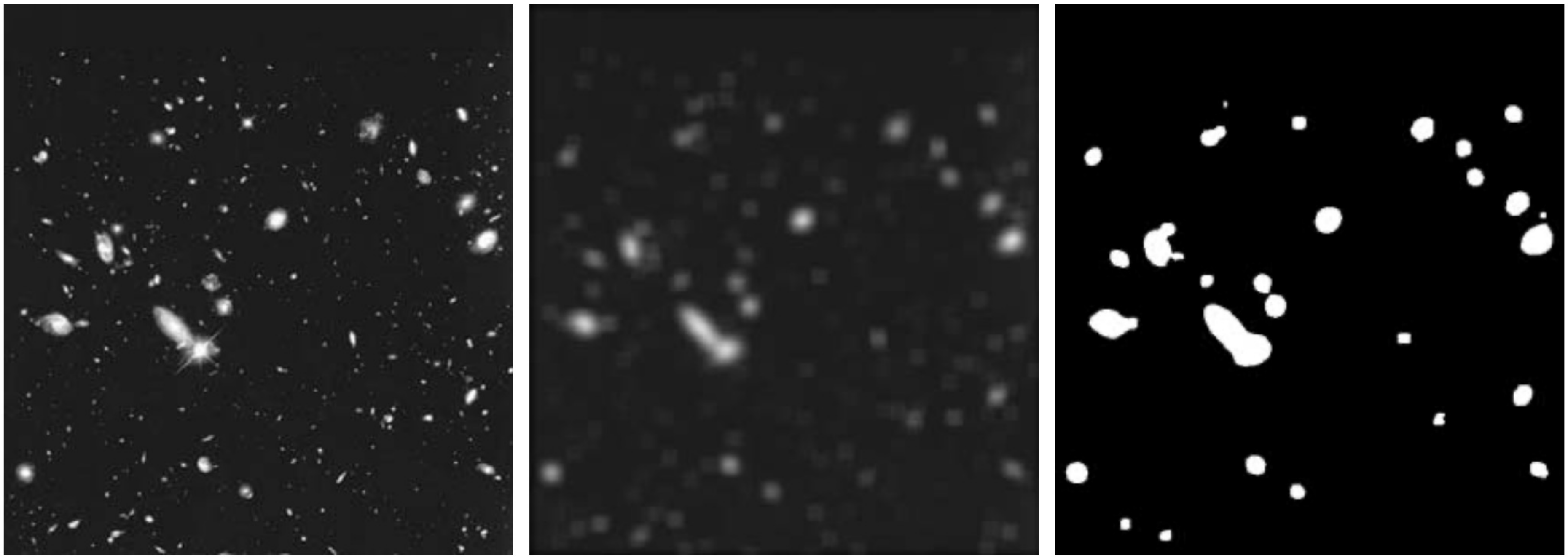


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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.

Example



a b c

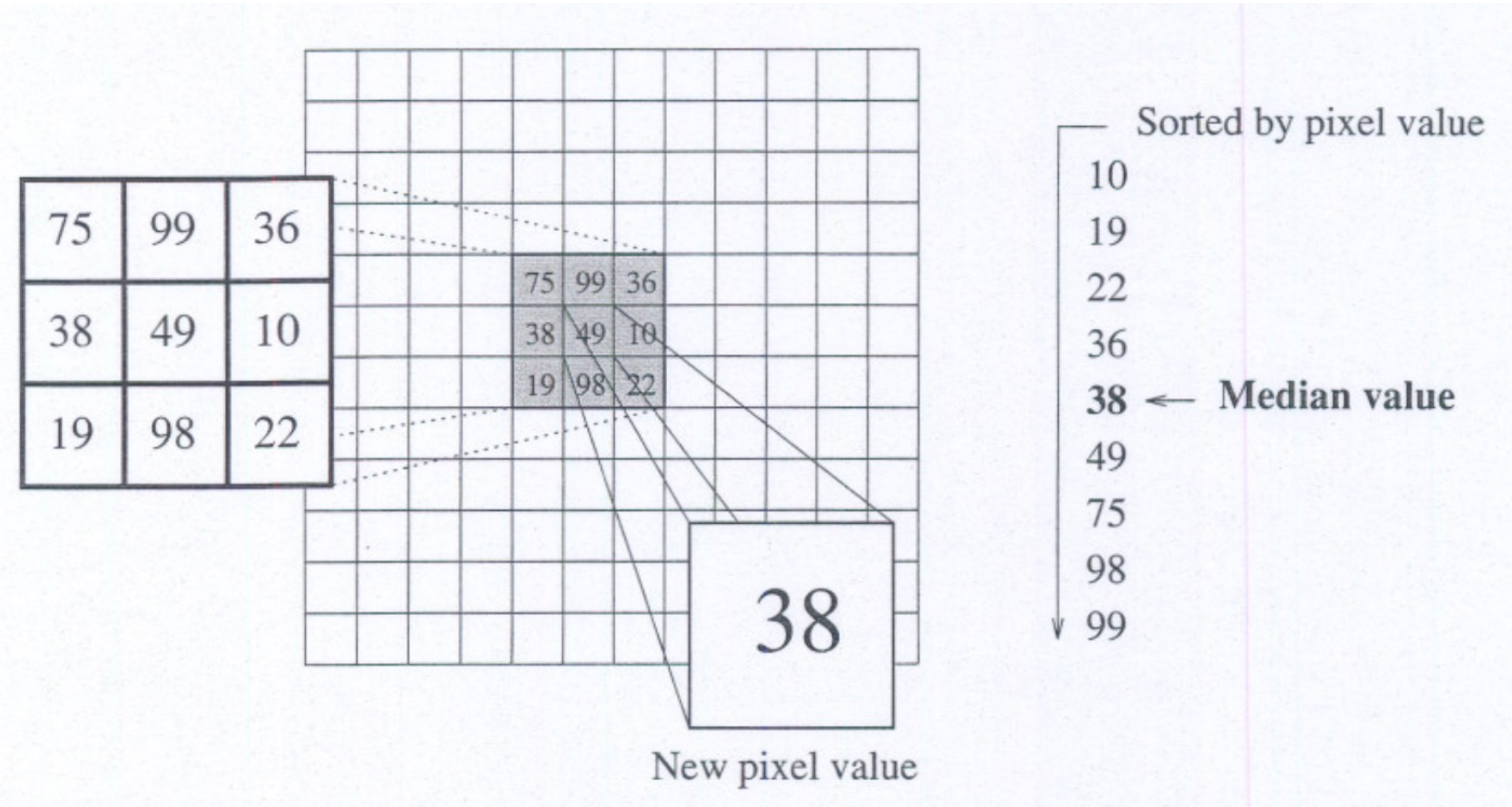
FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-statistic (non-linear) filters

- Order-statistic 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 result
- The best-known filter in this category is the median filter
- Median filters are particularly effective in the presence of impulse noise, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image

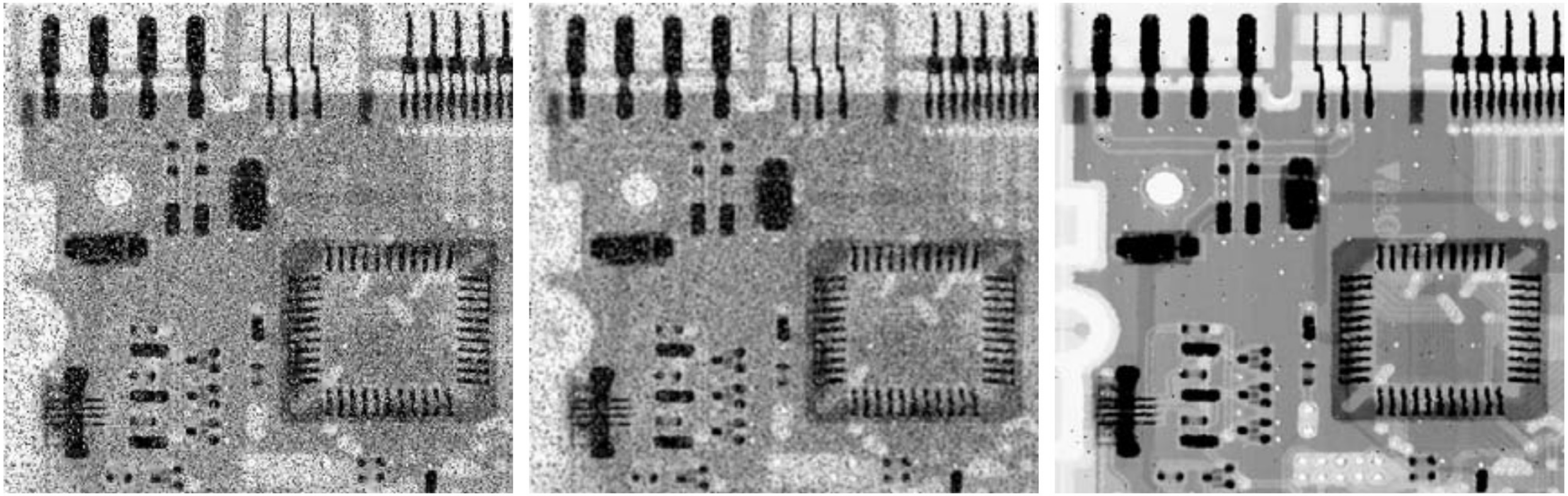
Order-statistic (non-linear) filters

- **Median filter 3x3**



Order-statistic (non-linear) filters

- **Median filter 3x3**
  - Neighbourhood: (10, 20, 20, 20, 15, 20, 20, 25, 100)
  - Sorted: (10, 15, 20, 20, **20**, 20, 20, 25, 100)



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.)

Example

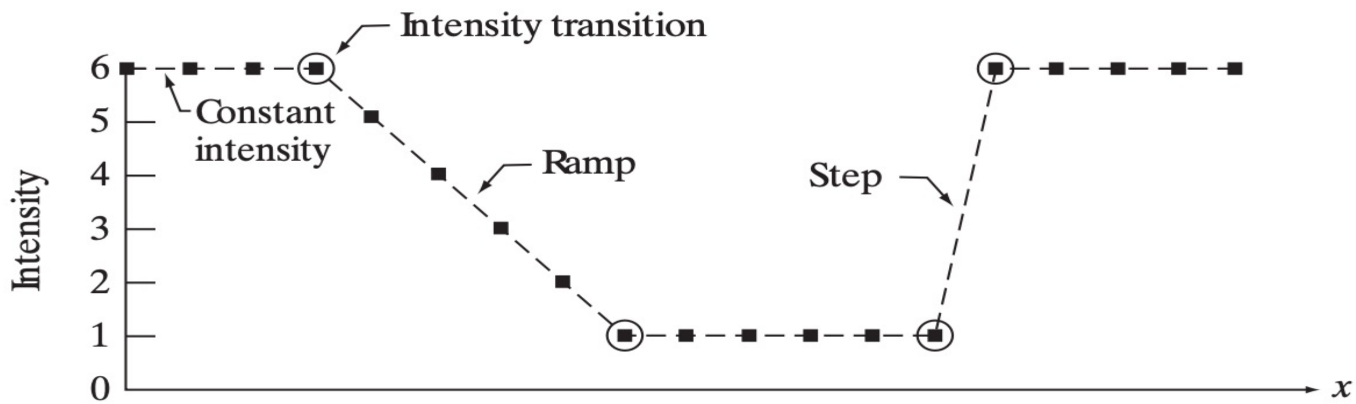
- An 8x8 image f(i,j) has gray levels given by the following equation: f[i, j] =|i-j| with i, j =0,1,2,3,4,5,6,7
- Illustrate the matrix representing the image
- Apply a 3x3 median filter on this image, find the output image; note that the border pixels remain unchanged.

Sharpening Spatial Filters

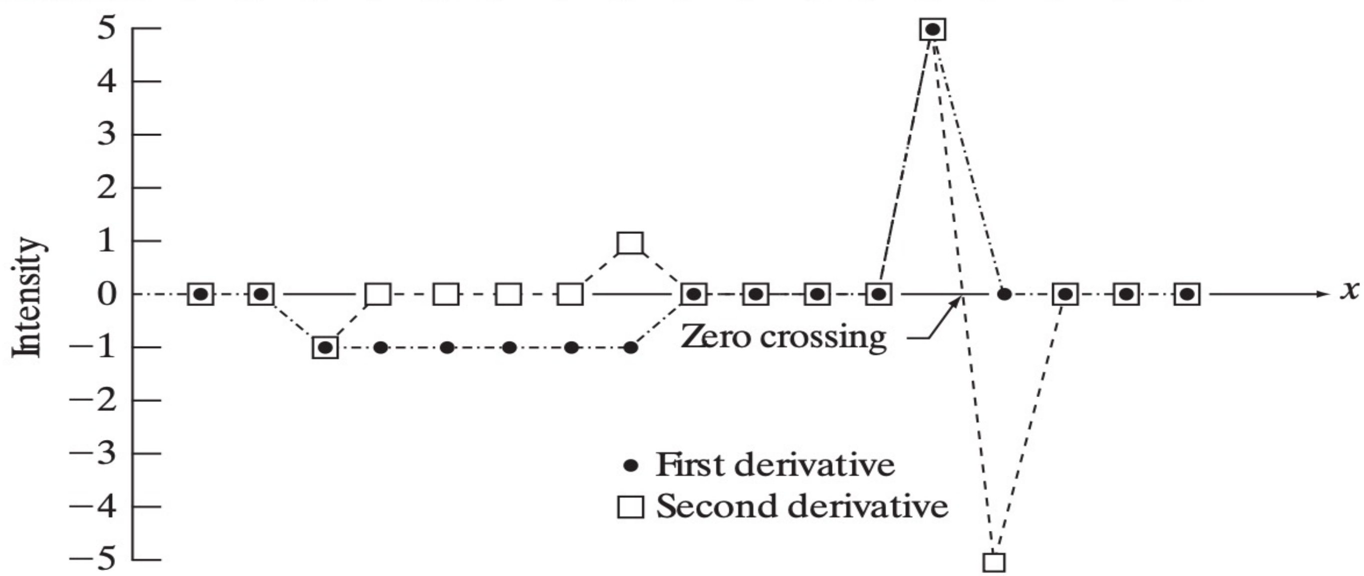
- The principal objective of sharpening is to highlight transitions in intensity.
- **Uses of image sharpening:** electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.
- **Foundation of sharpening spatial filter**
  - We focus attention initially on one-dimensional derivatives
  - Behavior of these derivatives in areas of constant intensity, at the onset and end of discontinuities
  - The derivatives of a digital function are defined in terms of differences.

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

First and second derivatives



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	



The Laplacian for Image Sharpening

- **Laplacian:**

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

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

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

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \qquad (3.$$

The Laplacian – Mask 3x3

- **f(x, y) and g(x, y) are the input and sharpened images, respectively.**
- **The constant is**
  - c = -1 if the Laplacian filters (first row)
  - c = 1 if either of the other two filters is used (second row)

0	1	0
1	−4	1
0	1	0

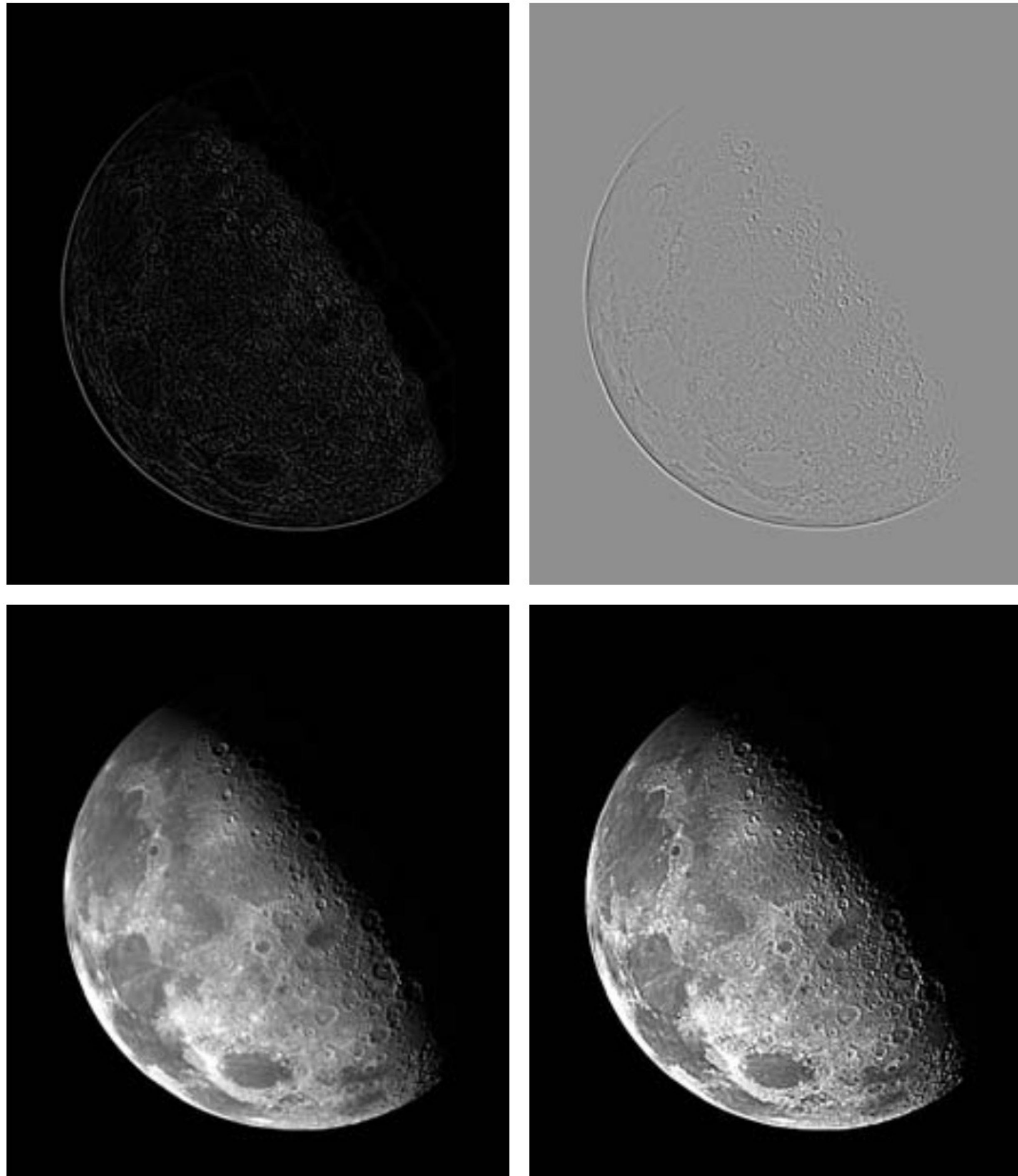
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

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

Example



Original image

Unsharp masking and Highboost Filtering

- A process that has been used for many years by the printing and publishing industry
- Sharpening images consists of subtracting an unsharp (smoothed) version of an image from the original image
- This process, called unsharp masking, consists of the following steps
  - Step 1: Blur the original image.
  - Step 2: Subtract the blurred image from the original (the resulting difference is called the *mask*.
  - Step 3: Add the mask to the original.

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

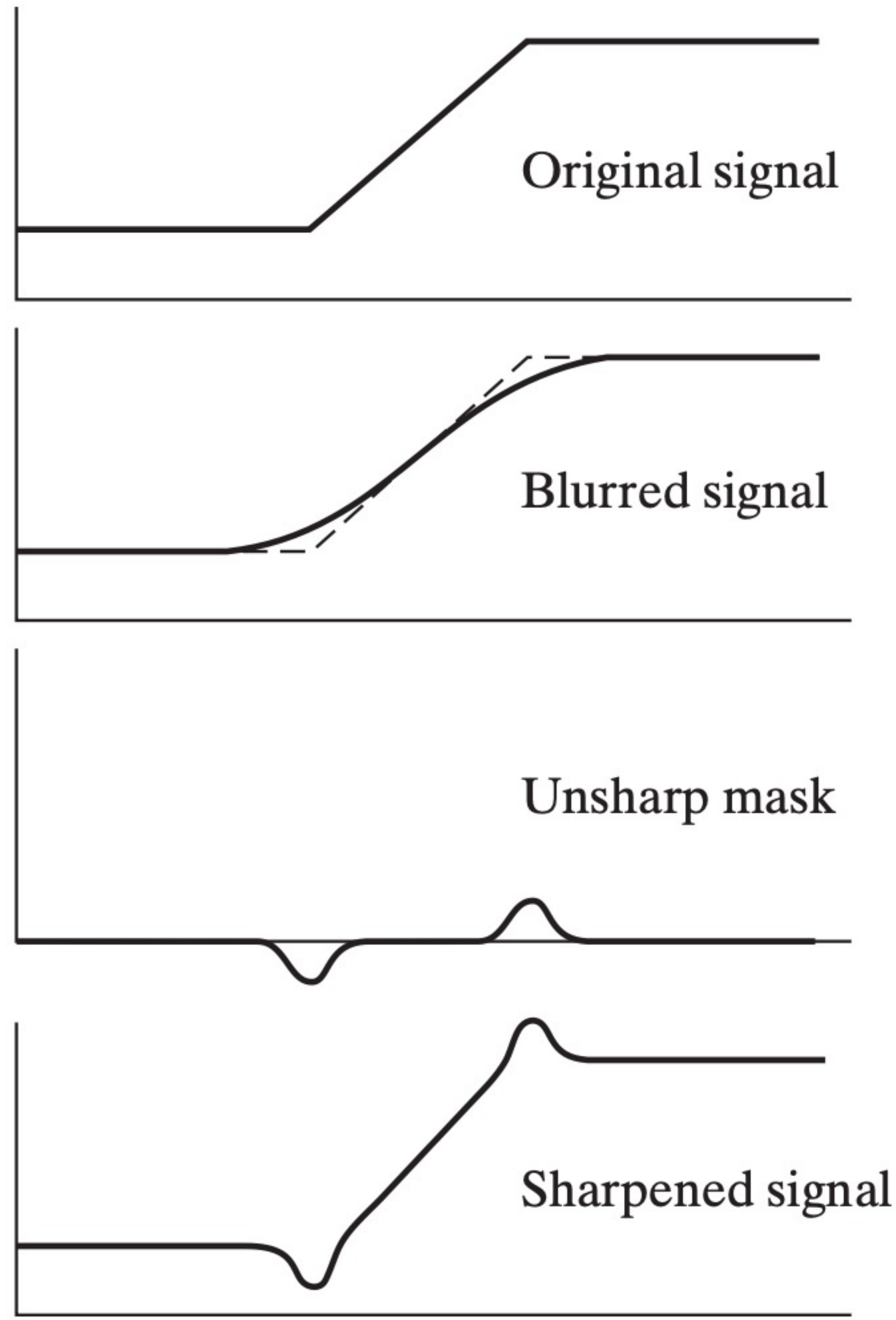
$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$

Unsharp masking

- Letting f(x, y) denote the blurred image
  - k = 1: unsharp masking
  - k > 1: highboost filtering
  - 0< k < 1:de-emphasizes the contribution of the un- sharp mask

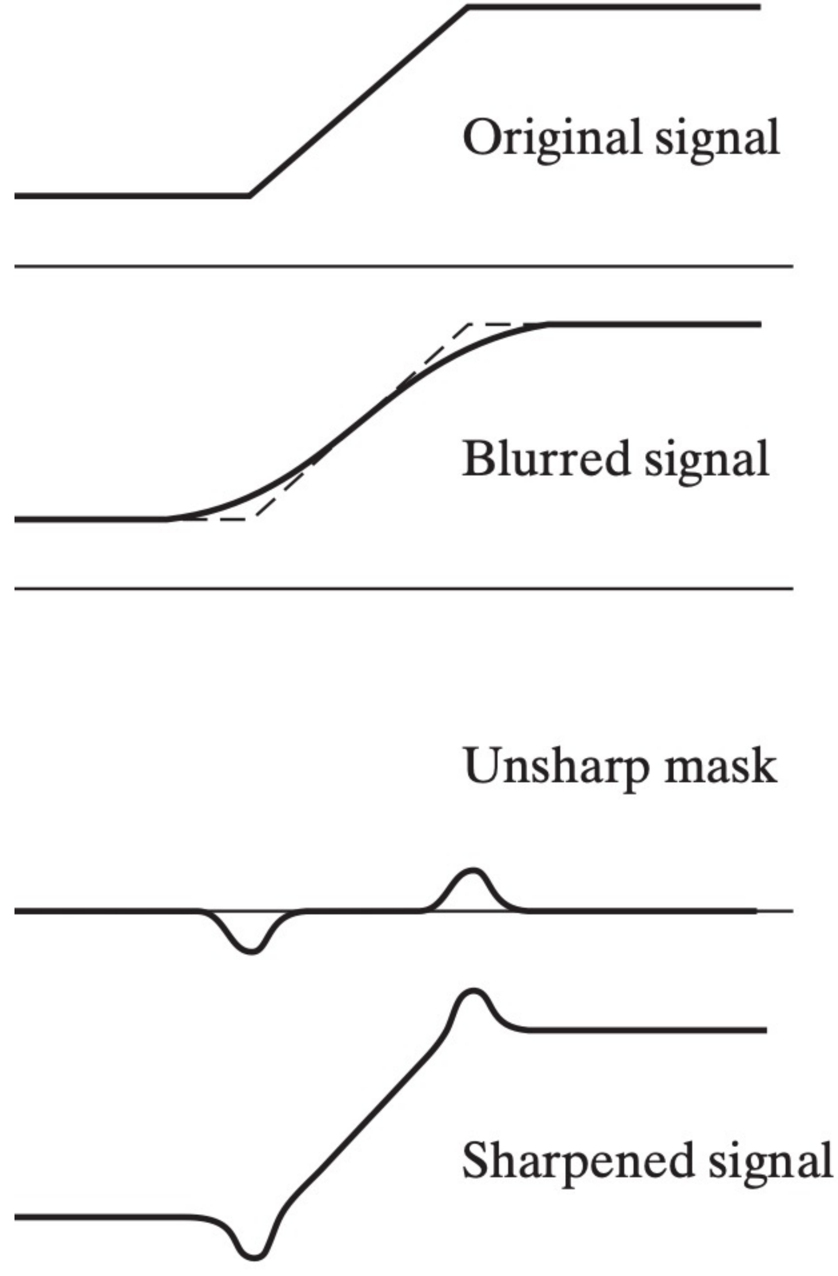
$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

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



Unsharp masking

(a) 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.



First order derivative for sharpening

- First derivatives in image processing are implemented using the magnitude of the gradient
- For a function f(x, y), the gradient of f at coordinates (x, y) is de- fined as the two-dimensional column vector

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$M(x, y) \approx |g_x| + |g_y|$$

gradient operators

- **Roberts cross**

−1	0
0	1

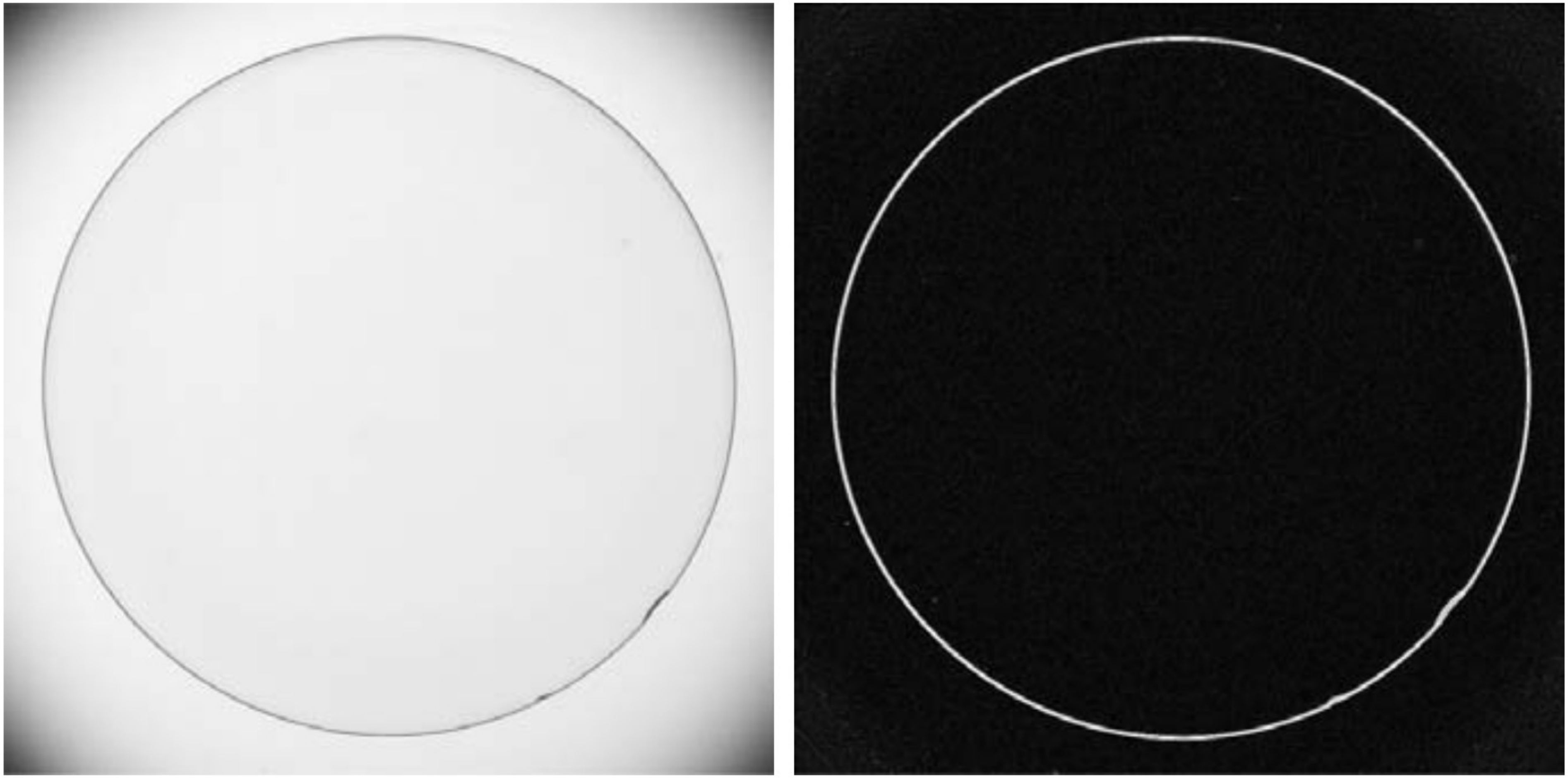
0	−1
1	0

- **Sobel operators**

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Gradient for image enhancement



(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.)

References

- Chapter 2, 3 - Digital Image Processing

Assigment 1 (week 1-2-3)

- Read, display and write an image using OpenCV
- Write a function to compute the histogram of an image and display the histogram
- Write a function to stretch constrast of an image
- Write a function to (3x3) filter an image, display and write the output image:
  - Median filter
  - Smoothing filter (Gaussian)
  - Sharpening filter (Laplacian)
- **You can implement in C/C++ or python/ java**
- **Don't use the pre-defined functions of the library**