Linear Filtering – Part I

Selim Aksoy
Department of Computer Engineering
Bilkent University
saksoy@cs.bilkent.edu.tr

Importance of neighborhood

- Both zebras and dalmatians have black and white pixels in similar numbers.
- The difference between the two is the characteristic appearance of small group of pixels rather than individual pixel values.

Adapted from Pinar Duygulu, Bilkent University

Outline

- We will discuss neighborhood operations that work with the values of the image pixels in the neighborhood.

- Spatial domain filtering
- Frequency domain filtering
- Image enhancement
- Finding patterns

Spatial domain filtering

3	3	3
3	3	3
3	3	3

- What is the value of the center pixel?

3	4	3
2	3	3
3	4	2

- What assumptions are you making to infer the center value?

Spatial domain 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 (or mask, kernel, template, window).
- The values in a filter subimage are referred to as coefficients, rather than pixels.

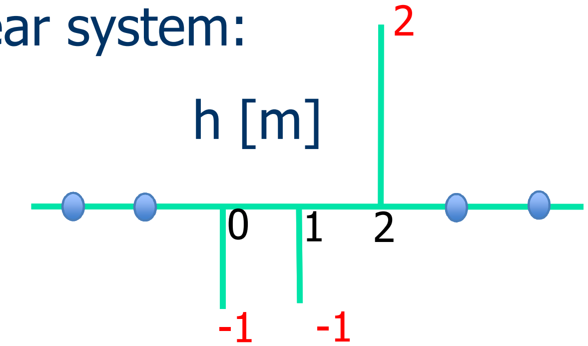
Spatial domain filtering

- Operation: modify the pixels in an image based on some function of the pixels in their neighborhood.
- Simplest: linear filtering (replace each pixel by a linear combination of its neighbors).
- Linear spatial filtering is often referred to as "convolving an image with a filter".

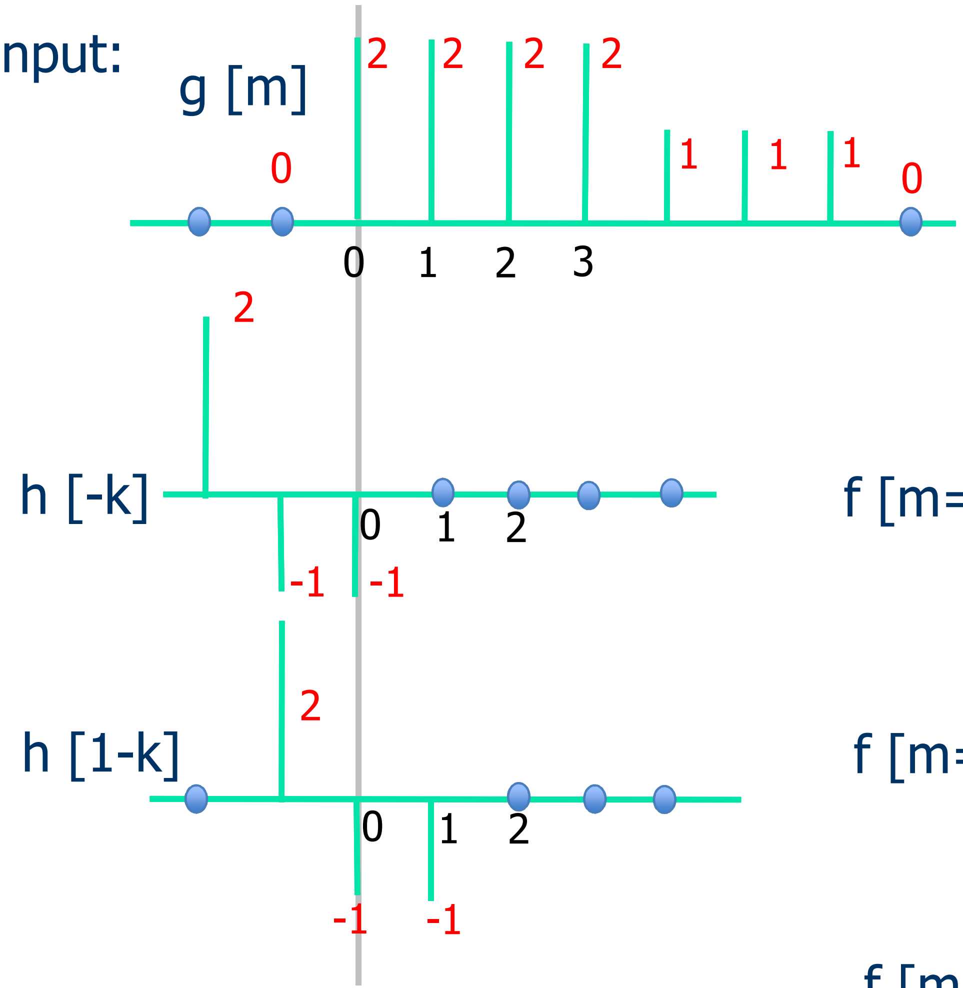
Linear filtering

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k,n-l]g[k,l]$$

Linear system:



Input:



Output?

$$f[m=0] = \sum_{k} h[-k]g[k]$$

f [m=0]=-2

$$f[m=1] = \sum_{k} h[1-k]g[k]$$

f [m=1]=-4

$$f[m=2] = \sum_{k} h[2-k]g[k]$$

f [m=2]=0

Linear filtering

For a linear spatially invariant system

$$f[m,n] = I \otimes g = \sum_{k,l} h[m-k, n-l]\, g[k,l]$$

m=0 1 2 …

111	115	113	111	112	111	112	111
135	138	137	139	145	146	149	147
163	168	188	196	206	202	206	207
180	184	206	219	202	200	195	193
189	193	214	216	104	79	83	77
191	201	217	220	103	59	60	68
195	205	216	222	113	68	69	83
199	203	223	228	108	68	71	77

g[m,n]

$\otimes$

-1	2	-1
-1	2	-1
-1	2	-1

h[m,n]

=

?	?	?	?	?	?	?	?
?	-5	9	-9	21	-12	10	?
?	-29	18	24	4	-7	5	?
?	-50	40	142	-88	-34	10	?
?	-41	41	264	-175	-71	0	?
?	-24	37	349	-224	-120	-10	?
?	-23	33	360	-217	-134	-23	?
?	?	?	?	?	?	?	?

f[m,n]

Linear filtering

- Filtering process:
  - Masks operate on a neighborhood of pixels.
  - The filter mask is centered on a pixel.
  - The mask coefficients are multiplied by the pixel values in its neighborhood and the products are summed.

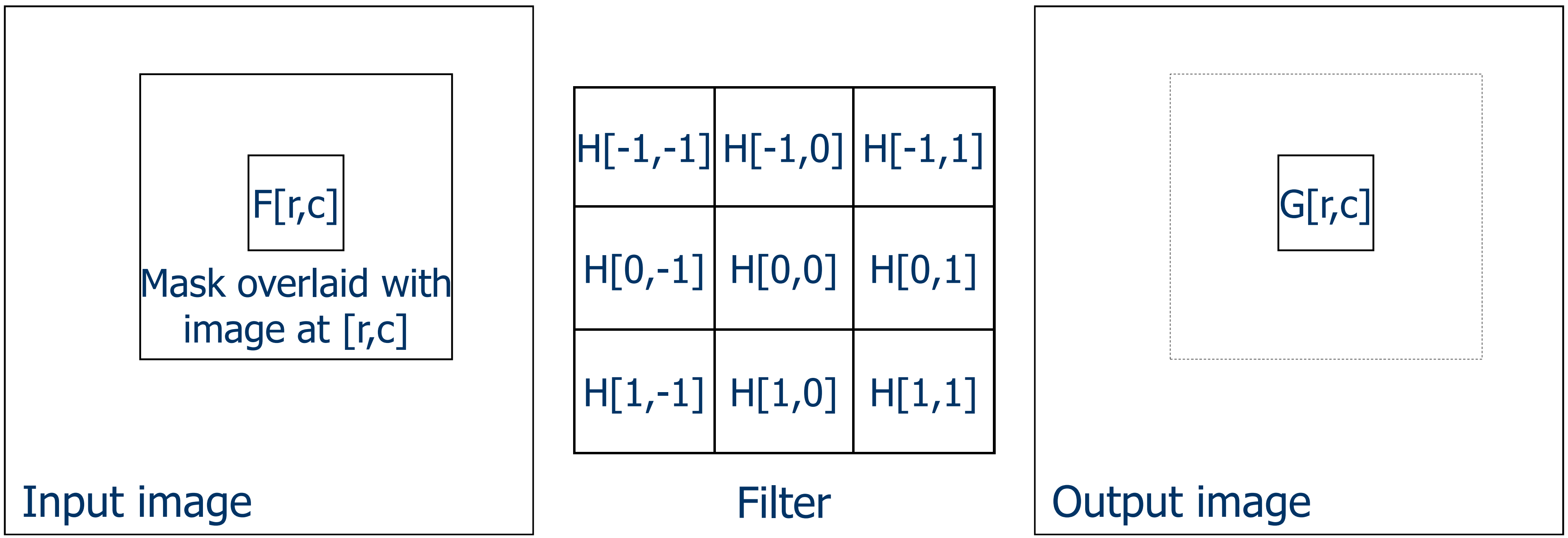
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[i+u, j+v]$$

  - The result goes into the corresponding pixel position in the output image.
  - This process is repeated by moving the filter mask from pixel to pixel in the image.

Linear filtering

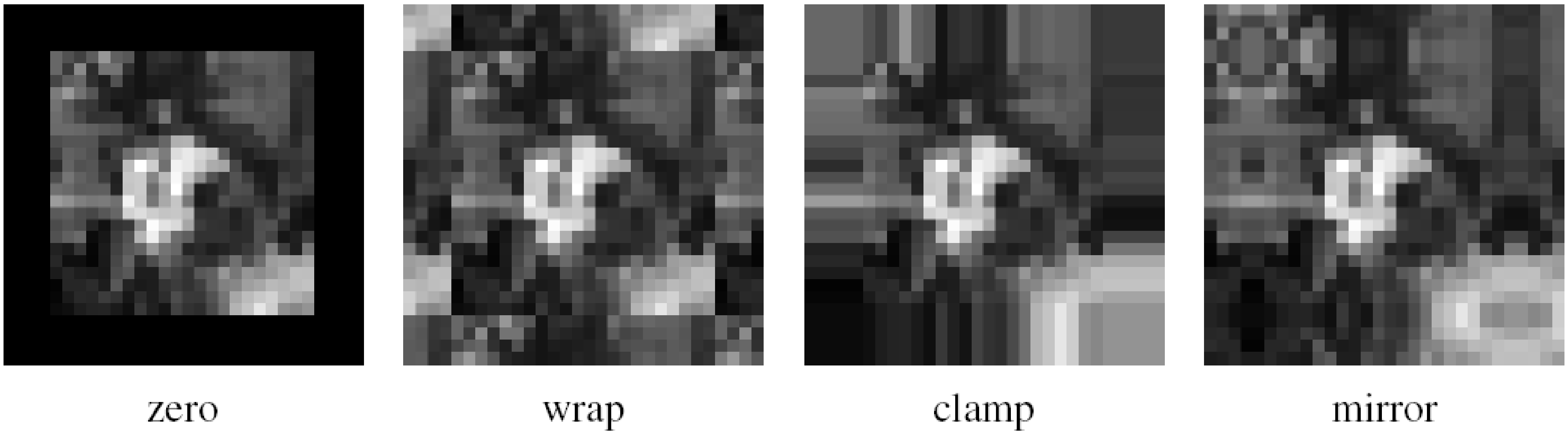
- This is called the cross-correlation operation and is denoted by

$$G = H \otimes F$$



Linear filtering

- Be careful about indices, image borders and padding during implementation.



Border padding examples.

Smoothing spatial filters

- Often, an image is composed of
  - some underlying ideal structure, which we want to detect and describe,
  - together with some random noise or artifact, which we would like to remove.
- Smoothing filters are used for blurring and for noise reduction.
- Linear smoothing filters are also called averaging filters.

Smoothing spatial filters

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

Averaging (mean) filter

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

Weighted average

Smoothing spatial filters

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

O

X	X	X	X	X	X
X	10				X
X					X
X					X
X					X
X	X	X	X	X	X

F

1/9

1	1	1
1	1	1
1	1	1

1/9.(10x1 + 11x1 + 10x1 + 9x1 + 10x1 + 11x1 + 10x1 + 9x1 + 10x1) =
1/9.(90) = 10

Adapted from Octavia Camps, Penn State

Smoothing spatial filters

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

F

1/9

1	1	1
1	1	1
1	1	1

O

X	X	X	X	X	X
X					X
X					X
X					X
X				20	X
X	X	X	X	X	X

1/9.(10x1 + 9x1 + 11x1 + 9x1 + 99x1 + 11x1 + 11x1 + 10x1 + 10x1) =
1/9.(180) = 20

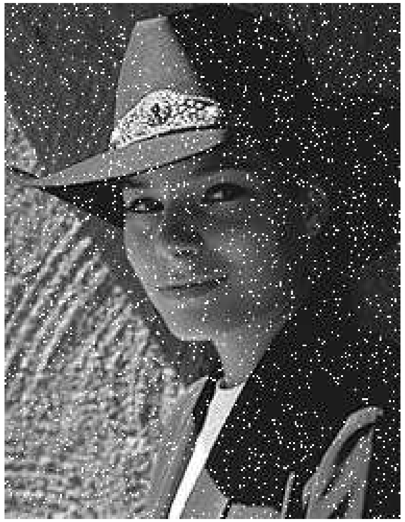
Adapted from Octavia Camps, Penn State

Smoothing spatial filters

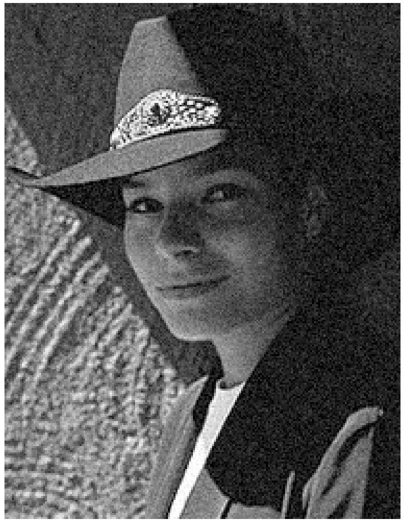
- Common types of noise:
  - Salt-and-pepper noise: contains random occurrences of black and white pixels.
  - Impulse noise: contains random occurrences of white pixels.
  - Gaussian noise: variations in intensity drawn from a Gaussian normal distribution.

Original

Salt and pepper noise



Impulse noise

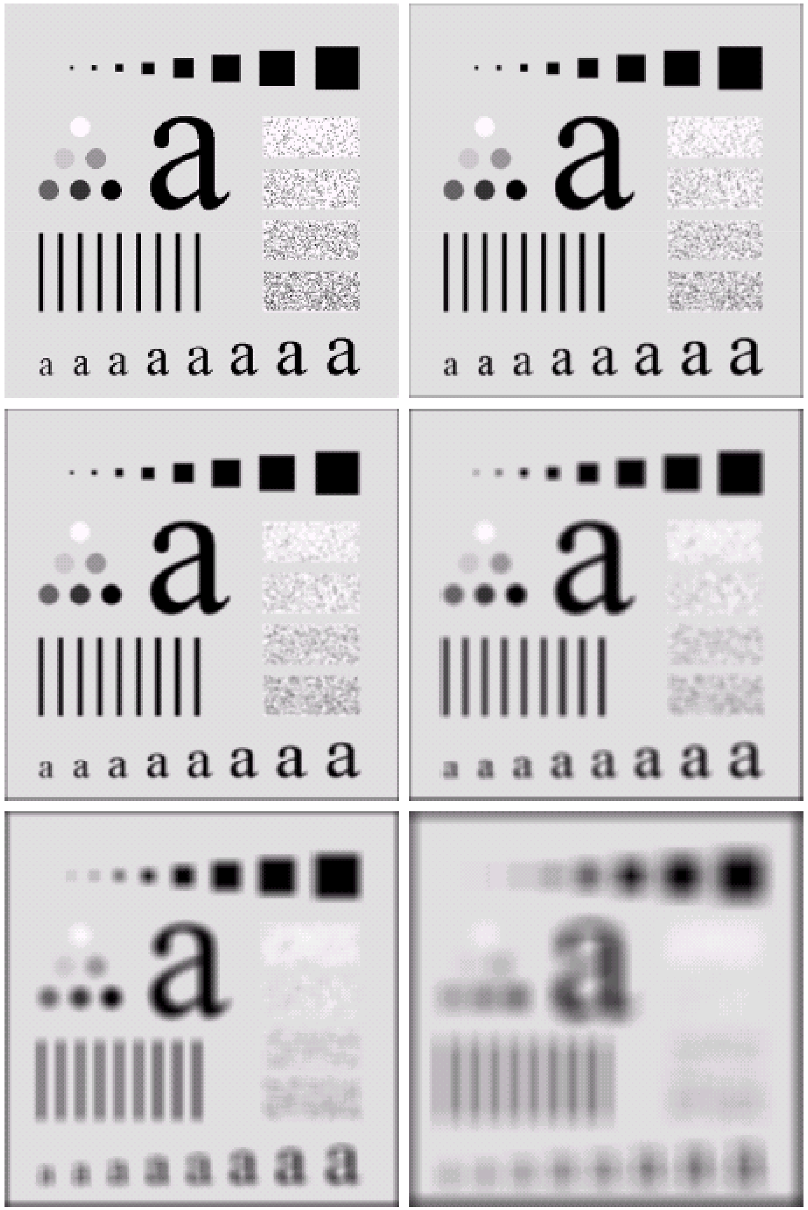


Gaussian noise

Adapted from Linda Shapiro, U of Washington

Adapted from Linda Shapiro, U of Washington

Smoothing spatial filters

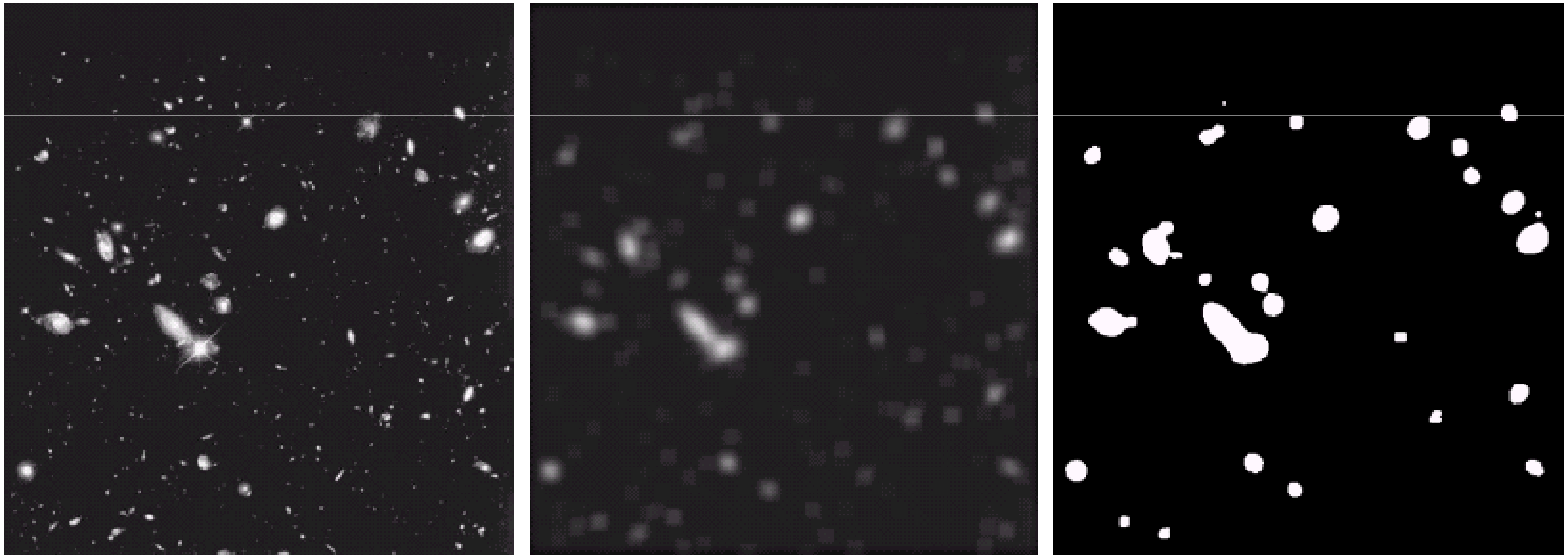


a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ 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 gray 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.

Adapted from Gonzales and Woods

Smoothing spatial filters

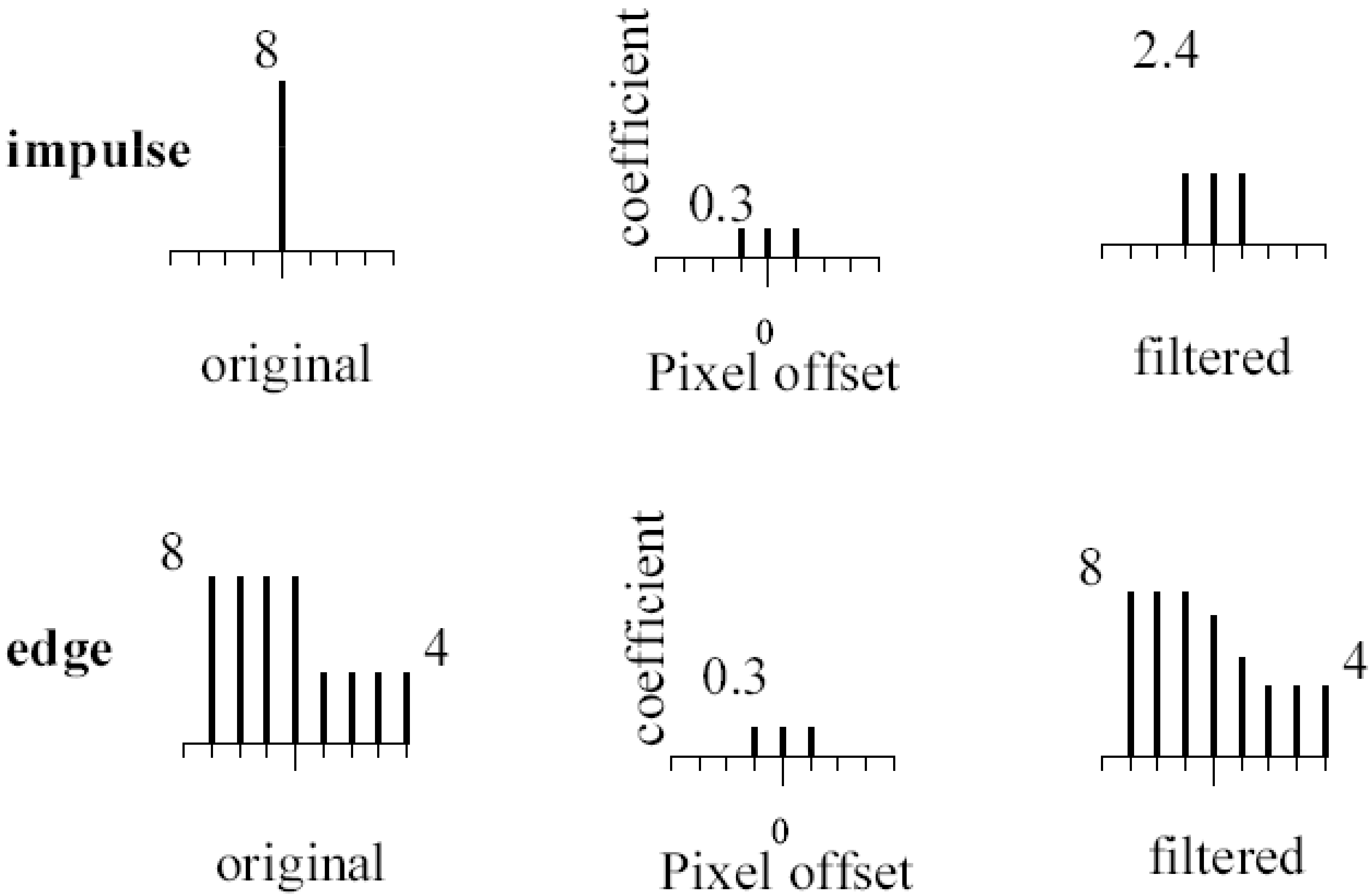


a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

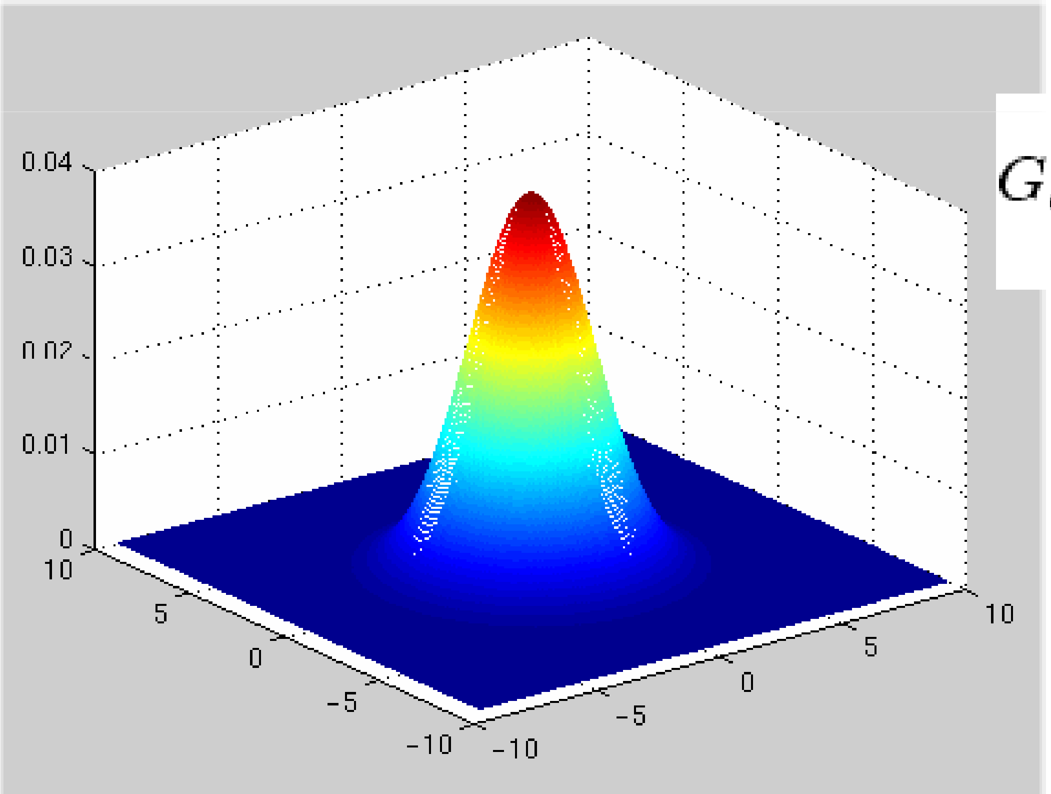
Adapted from Gonzales and Woods

Smoothing spatial filters



Adapted from Darrell and Freeman, MIT

Smoothing spatial filters



$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

A weighted average that weighs pixels at its center much more strongly than its boundaries.

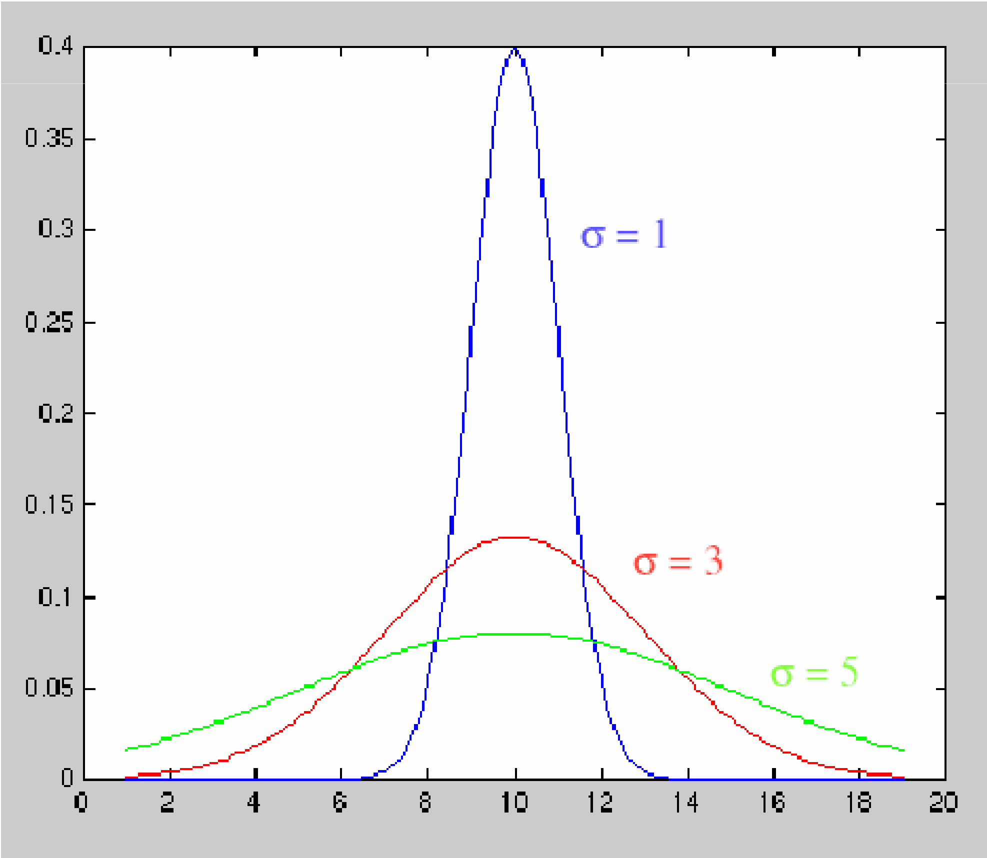
2D Gaussian filter

Adapted from Martial Hebert, CMU

Smoothing spatial filters

- If σ is small: smoothing will have little effect.
- If σ is larger: neighboring pixels will have larger weights resulting in consensus of the neighbors.
- If σ is very large: details will disappear along with the noise.

Effect of σ

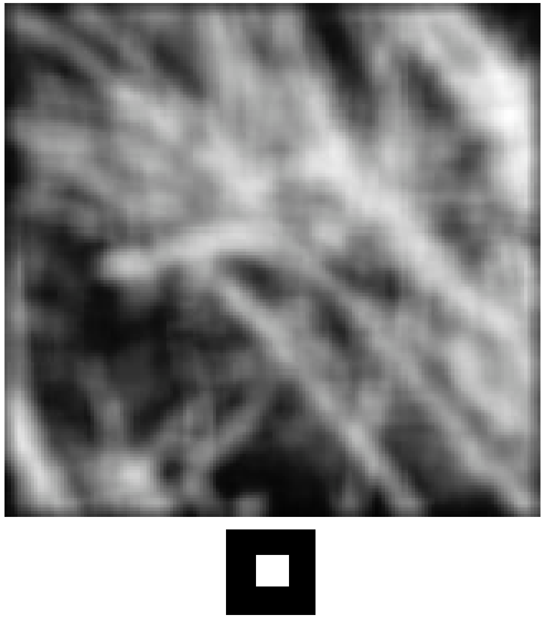


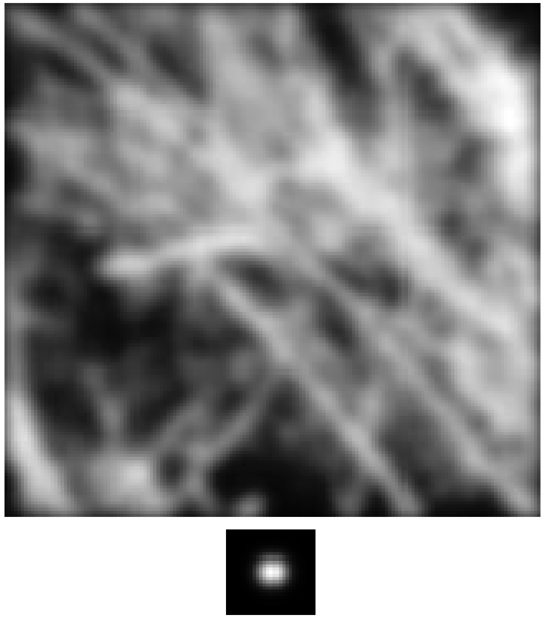
Adapted from Martial Hebert, CMU

Smoothing spatial filters

Result of blurring using a uniform local model.

Produces a set of narrow horizontal and vertical bars – ringing effect.



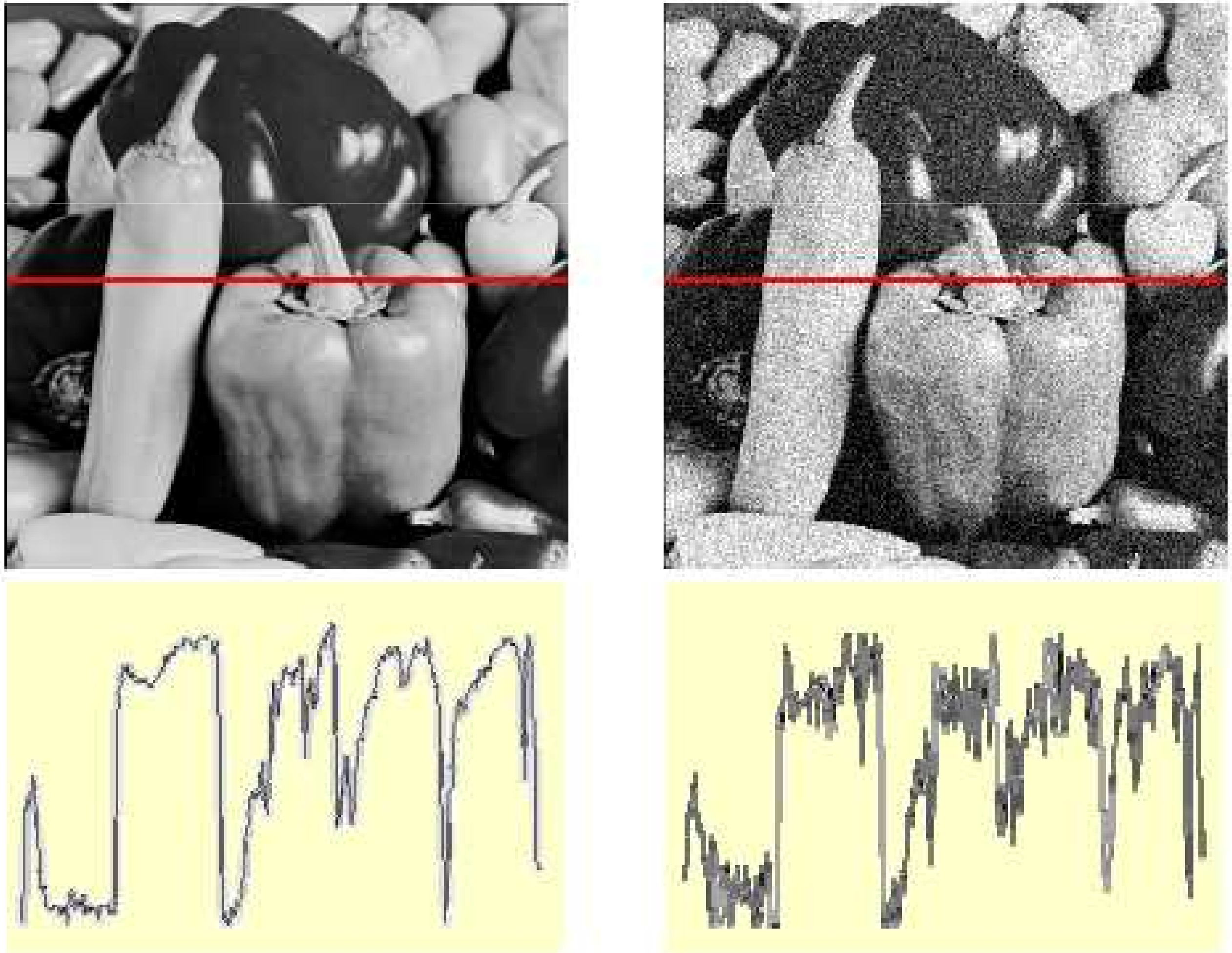


Result of blurring using a Gaussian filter.

Adapted from David Forsyth, UC Berkeley

Smoothing spatial filters

Image Noise



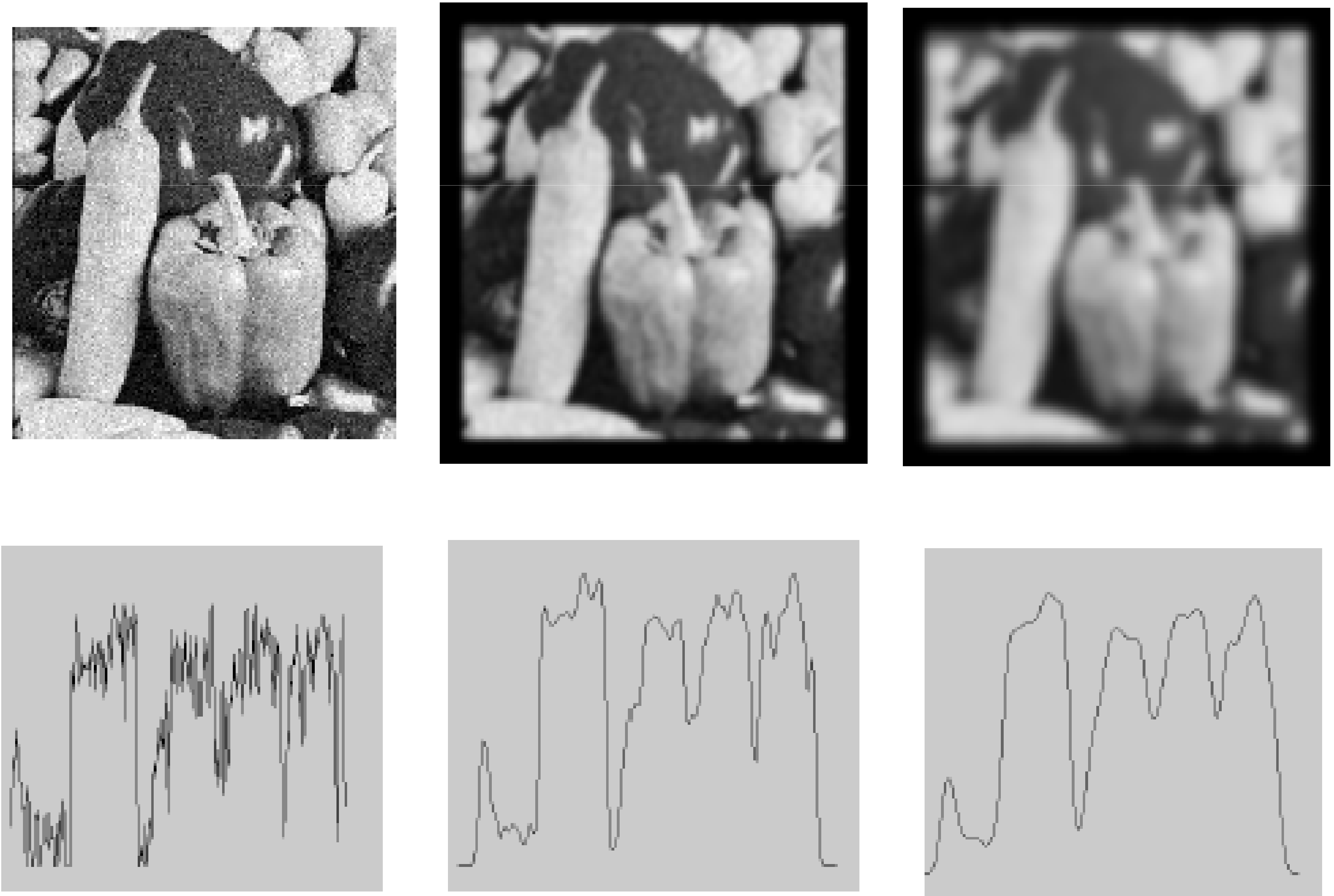
$$f(x,y) = \overbrace{\tilde{f}(x,y)}^{\text{Ideal Image}} + \overbrace{\eta(x,y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:

$\eta(x,y) \sim \mathcal{N}(\mu,\sigma)$

Adapted from Martial Hebert, CMU

Smoothing spatial filters



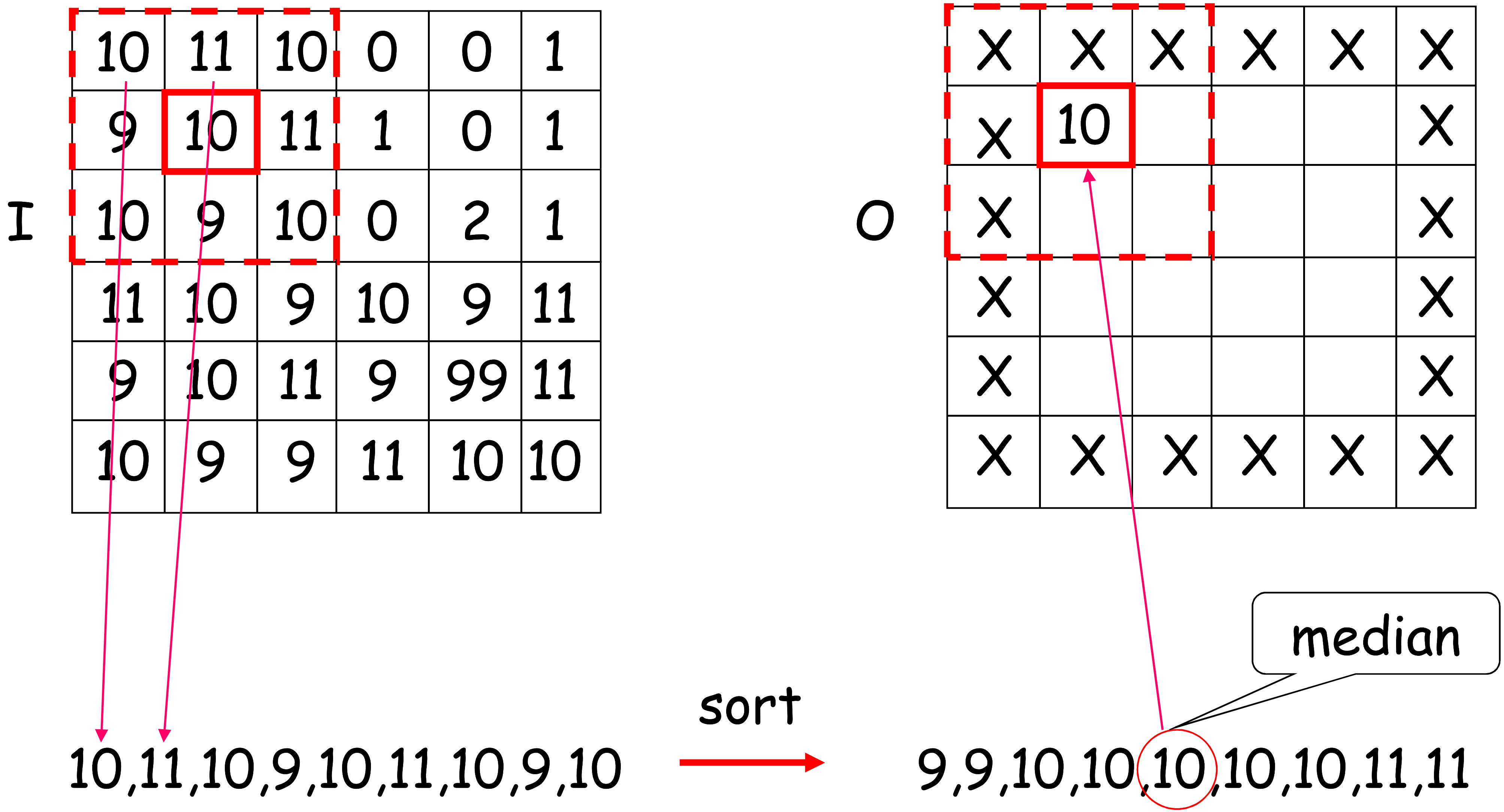
No smoothing | $\sigma = 2$ | $\sigma = 4$

Adapted from Martial Hebert, CMU

Order-statistic filters

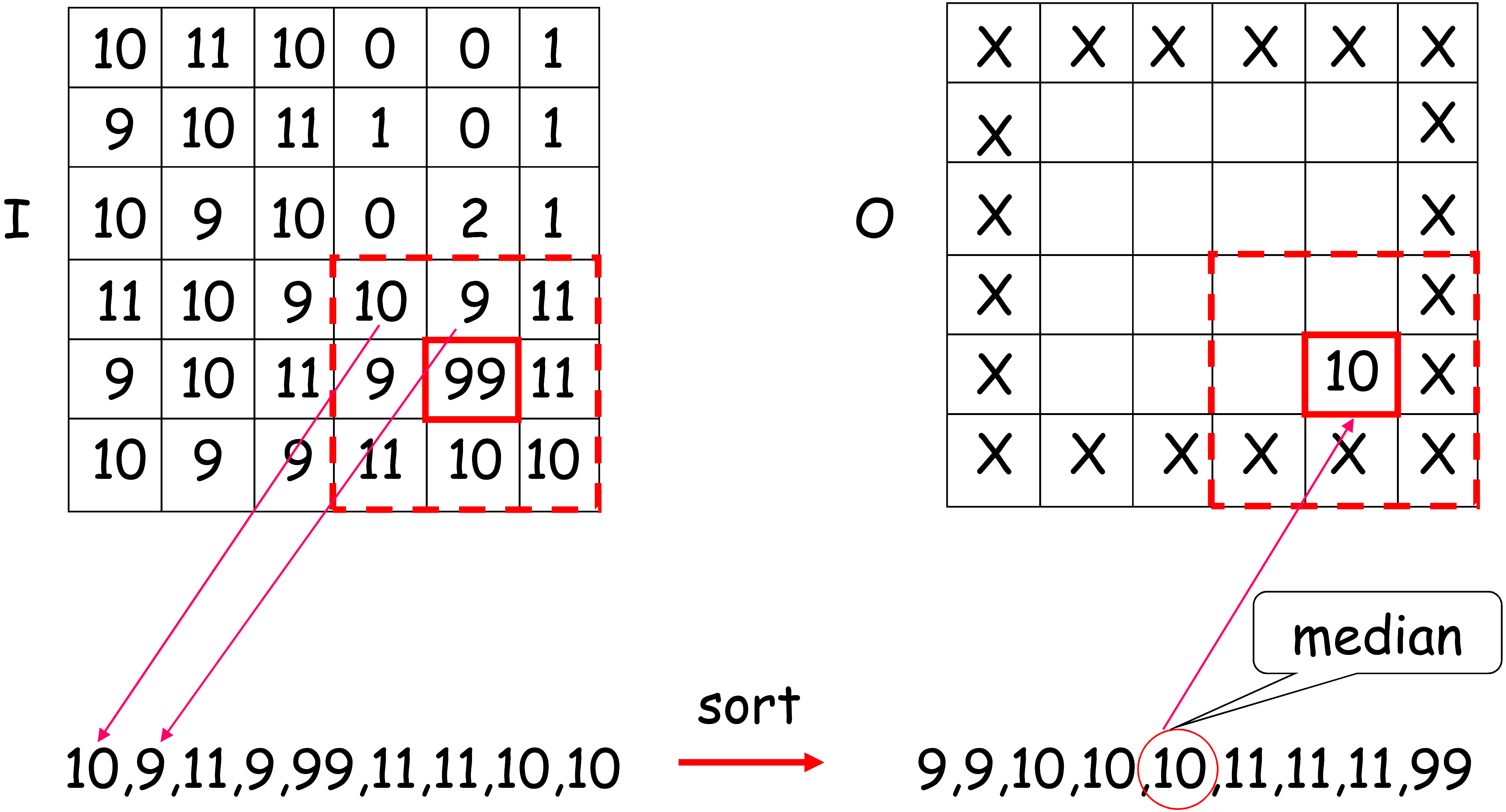
- Order-statistic filters are nonlinear spatial filters whose response is based on
  - ordering (ranking) the pixels contained in the image area encompassed by the filter, and then
  - replacing the value of the center pixel with the value determined by the ranking result.
- The best-known example is the median filter.
- It is particularly effective in the presence of impulse or salt-and-pepper noise, with considerably less blurring than linear smoothing filters.

Order-statistic filters



Adapted from Octavia Camps, Penn State

Order-statistic filters



Adapted from Octavia Camps, Penn State

Salt-and-pepper noise

Mean | Gaussian | Median

3x3

5x5

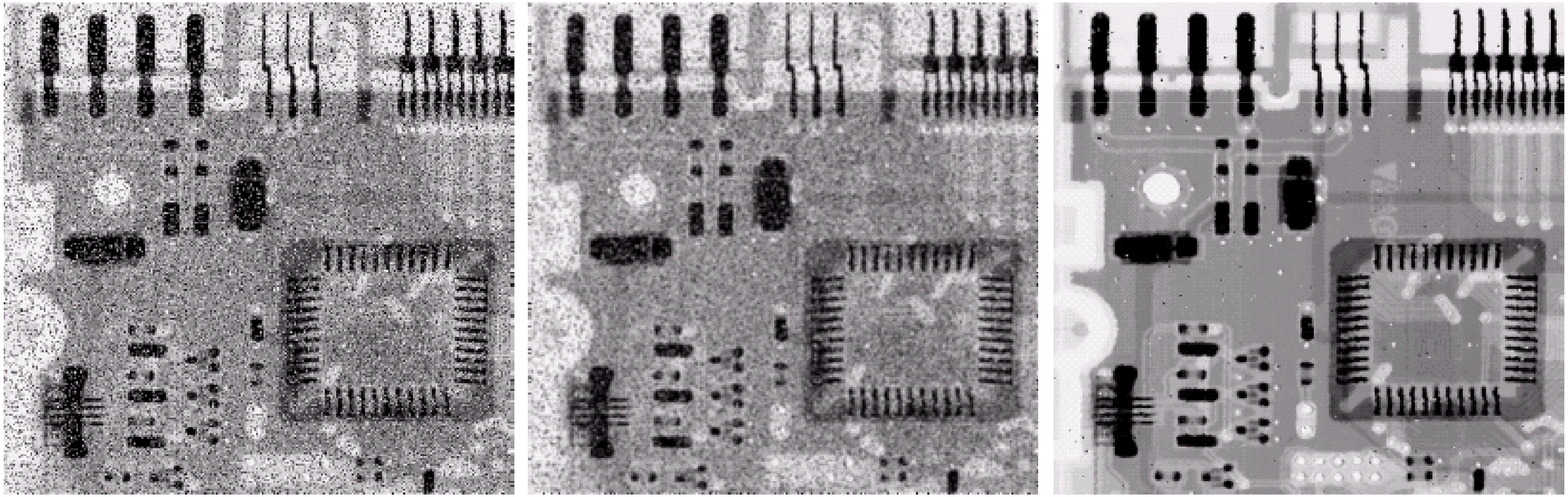
7x7

Adapted from Linda Shapiro,
U of Washington

Gaussian noise

Adapted from Linda Shapiro, U of Washington

Order-statistic filters



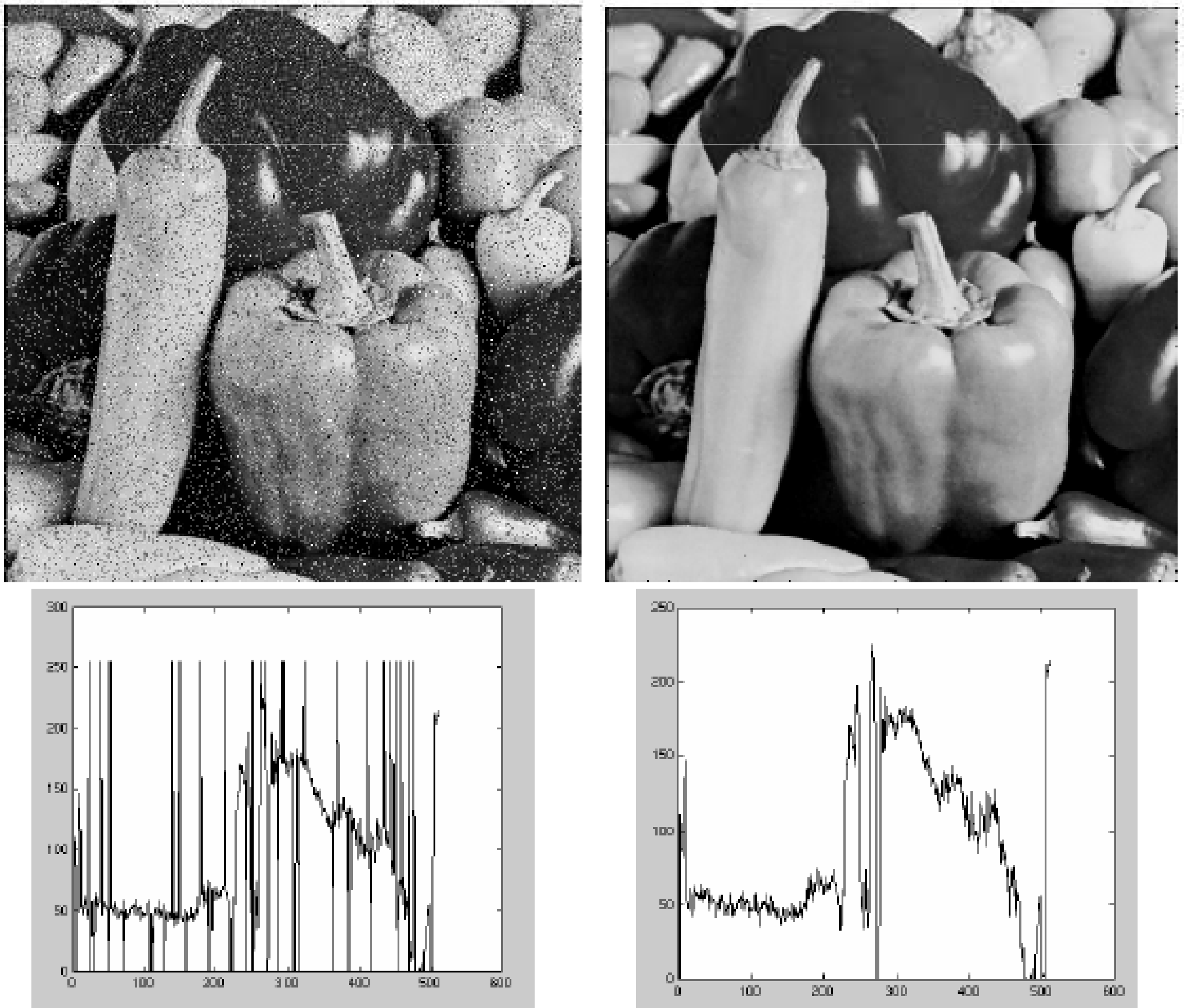
a b c

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

Adapted from Gonzales and Woods

Order-statistic filters

Effect of median filter on salt and pepper noise



Adapted from Martial Hebert, CMU

Sharpening spatial filters

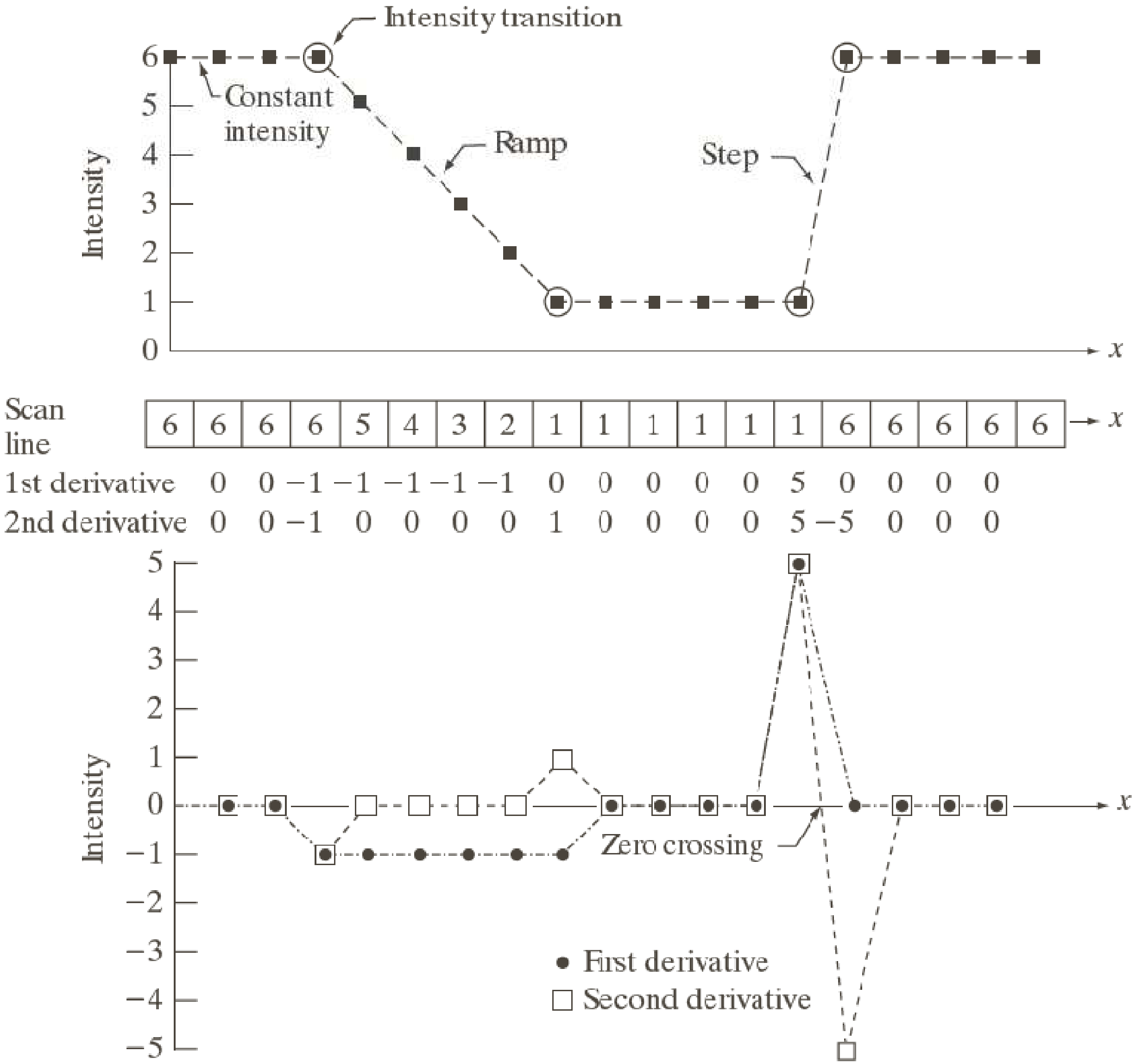
- Objective of sharpening is to highlight or enhance fine detail in an image.
- Since smoothing (averaging) is analogous to integration, sharpening can be accomplished by spatial differentiation.
- First-order derivative of 1D function f(x)

$$f(x+1) - f(x).$$

- Second-order derivative of 1D function f(x)

$$f(x+1) - 2f(x) + f(x-1).$$

Sharpening spatial filters



a
b
c

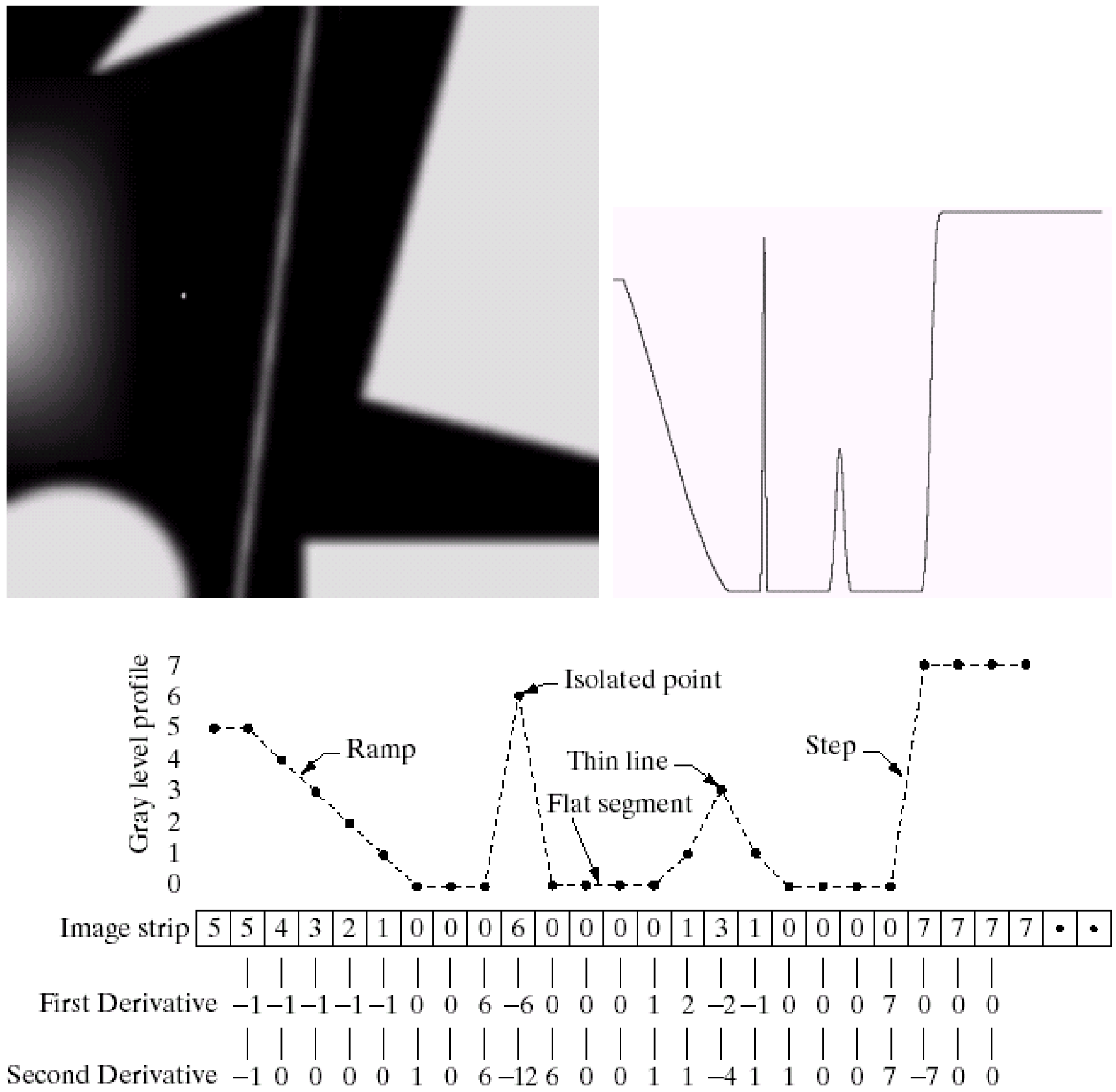
FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Adapted from Gonzales and Woods

Sharpening spatial filters

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).



Adapted from Gonzales and Woods

Sharpening spatial filters

- Observations:
  - First-order derivatives generally produce thicker edges in an image.
  - Second-order derivatives have a stronger response to fine detail (such as thin lines or isolated points).
  - First-order derivatives generally have a stronger response to a gray level step.
  - Second-order derivatives produce a double response at step changes in gray level.

Sharpening spatial filters

- *Laplacian* of a function (image) $f(x, y)$ of two variables x and y

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

is a second-order derivative operator.

(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 3.39 (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.

Sharpening spatial filters

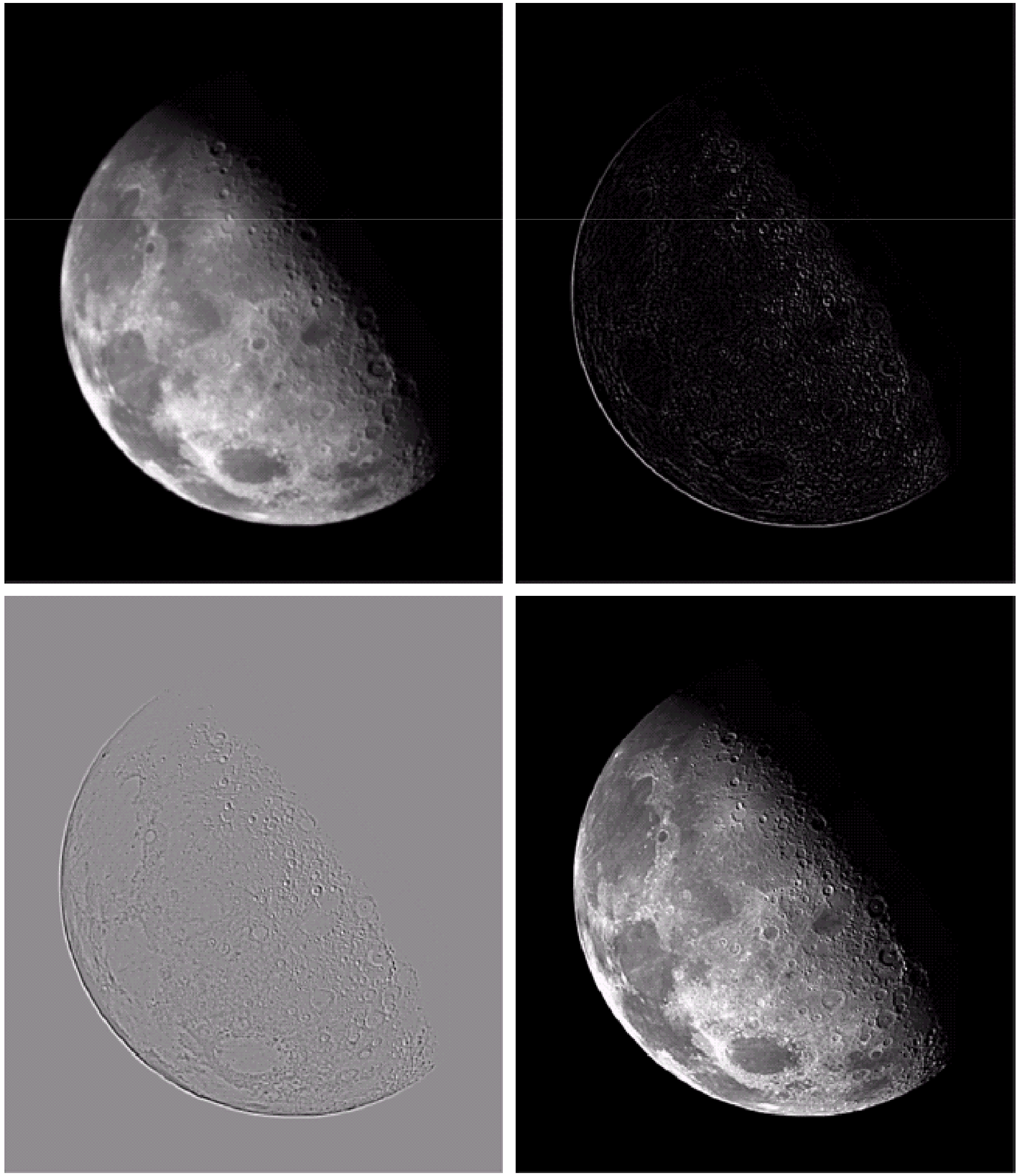


FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

Adapted from Gonzales and Woods

Sharpening spatial filters

- For a function $f(x, y)$, the *gradient* at (x, y) is defined as

$$\nabla f = \left[\frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y} \right]^T$$

where its magnitude can be used to implement first-order derivatives.

−1	0
0	1

0	−1
1	0

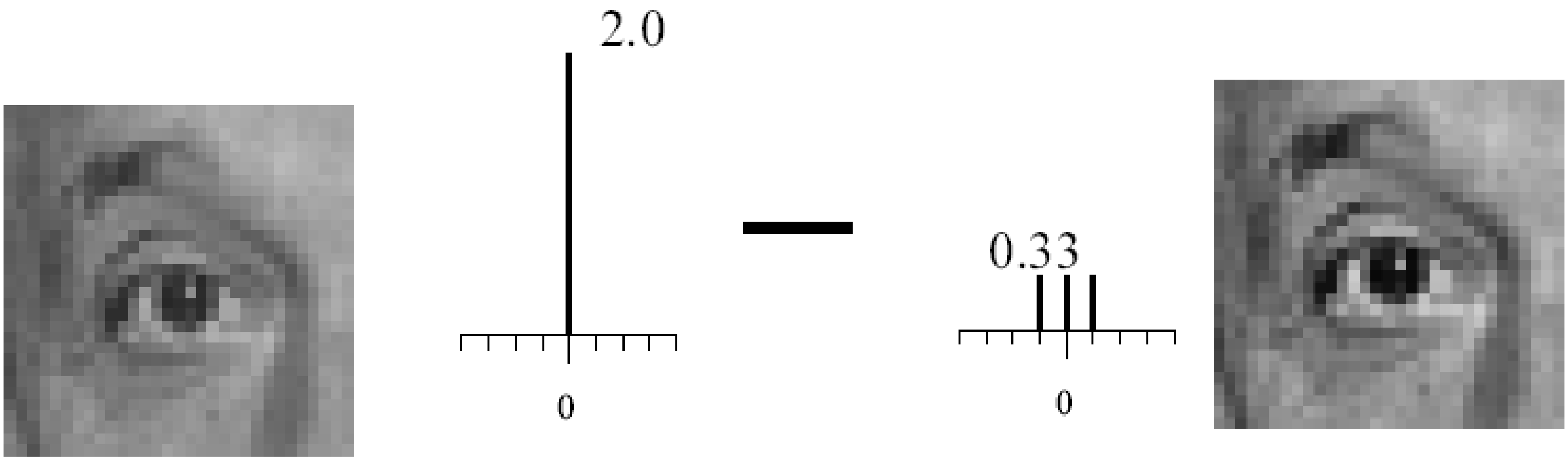
Robert's cross-gradient operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel gradient operators

Sharpening spatial filters



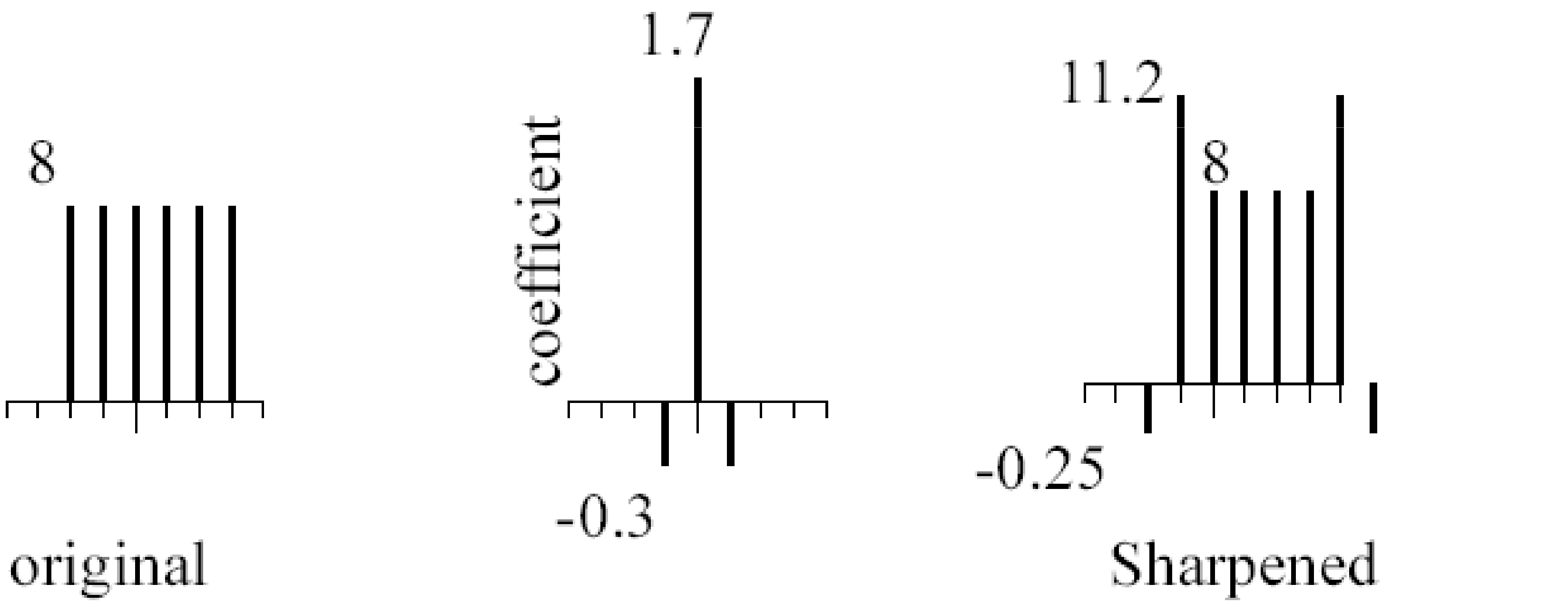
original

Sharpened original

High-boost filtering

Adapted from Darrell and Freeman, MIT

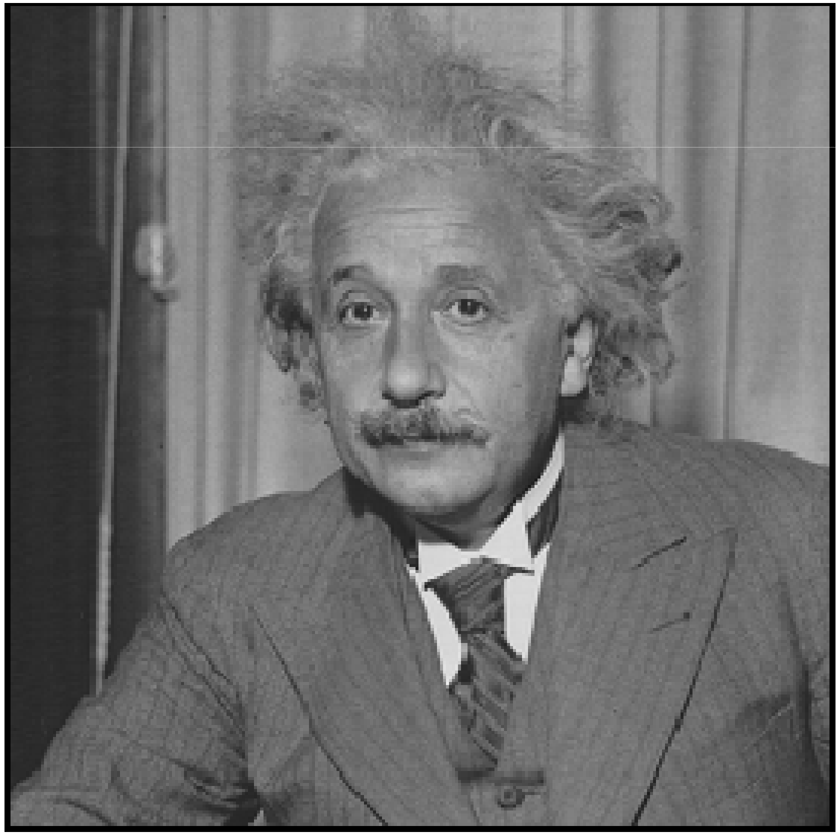
Sharpening spatial filters



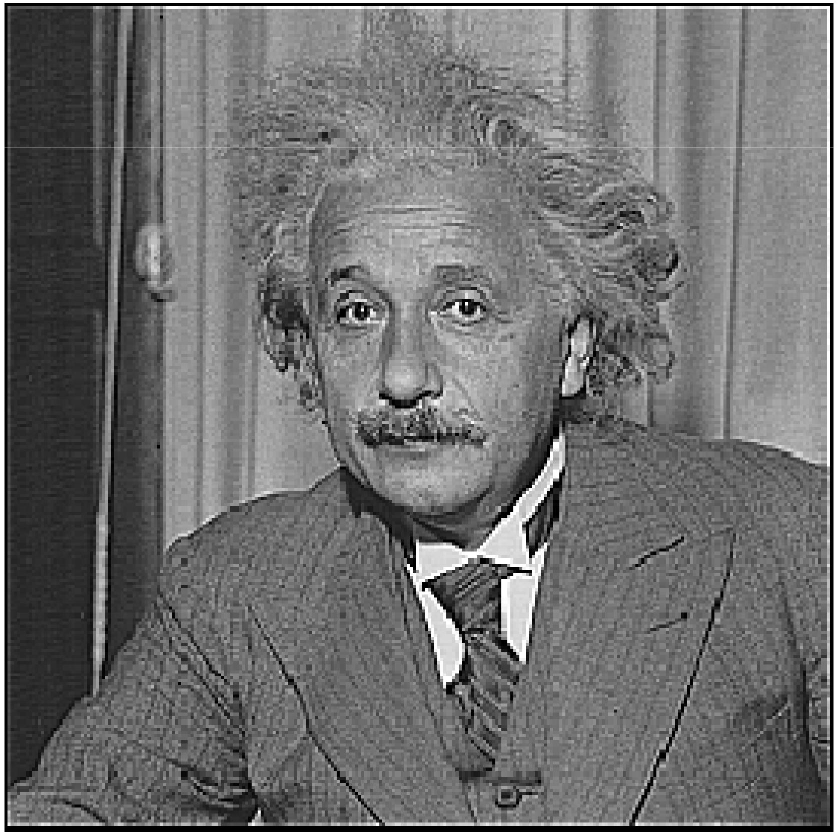
Sharpened (differences are accentuated; constant areas are left untouched).

Adapted from Darrell and Freeman, MIT

Sharpening spatial filters



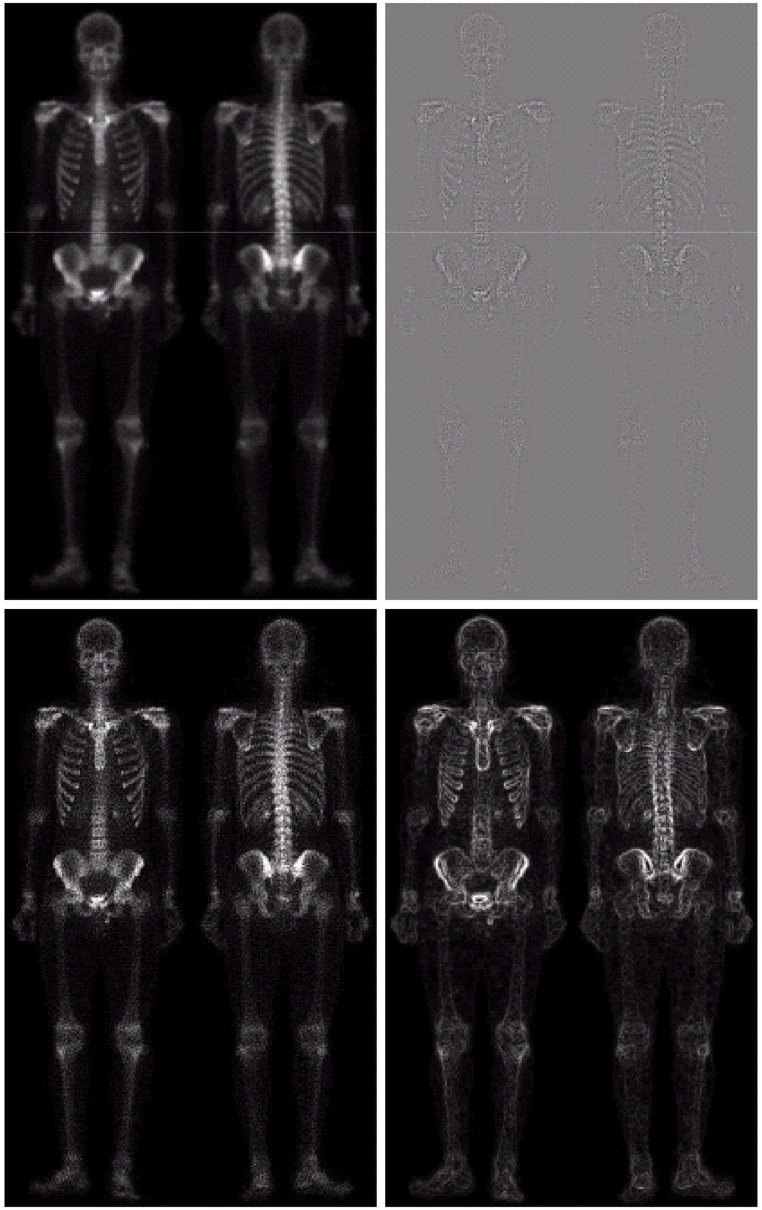
before



after

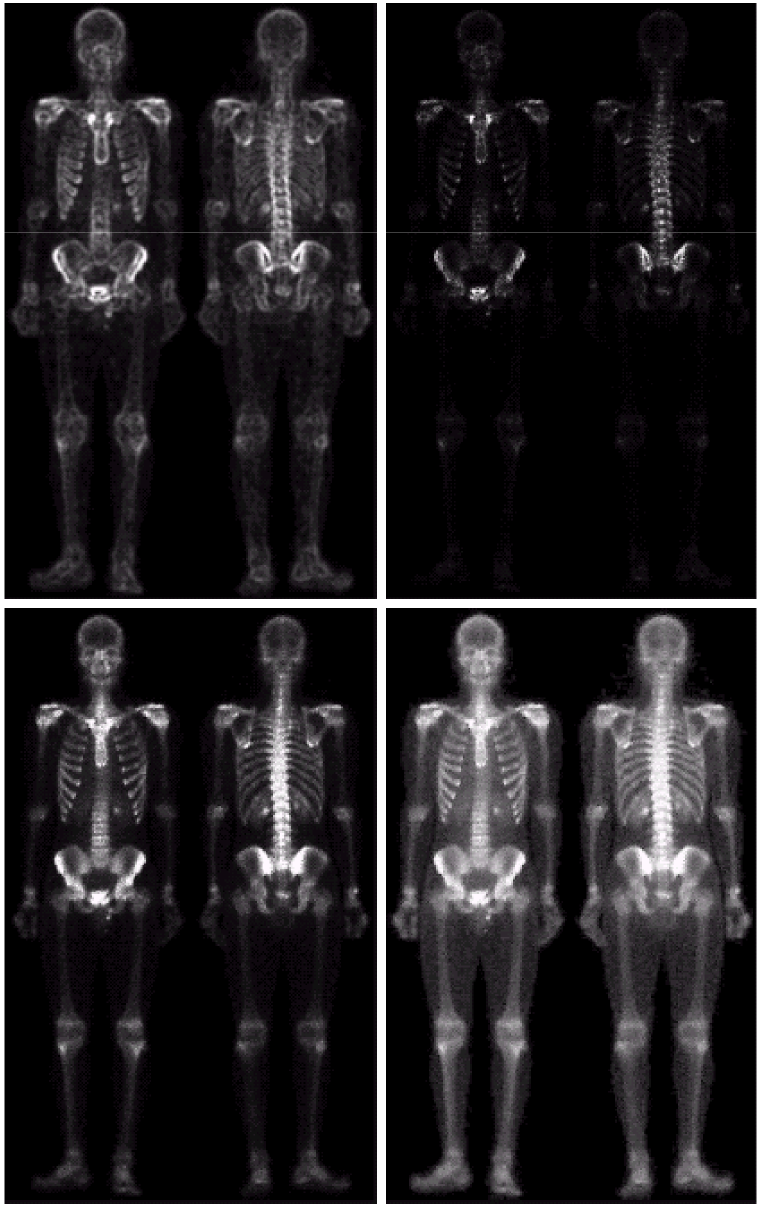
Adapted from Darrell and Freeman, MIT

Combining spatial enhancement methods



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).



e f
g h

FIGURE 3.46
(Continued)
(e) Sobel image smoothed with a 5 × 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.)