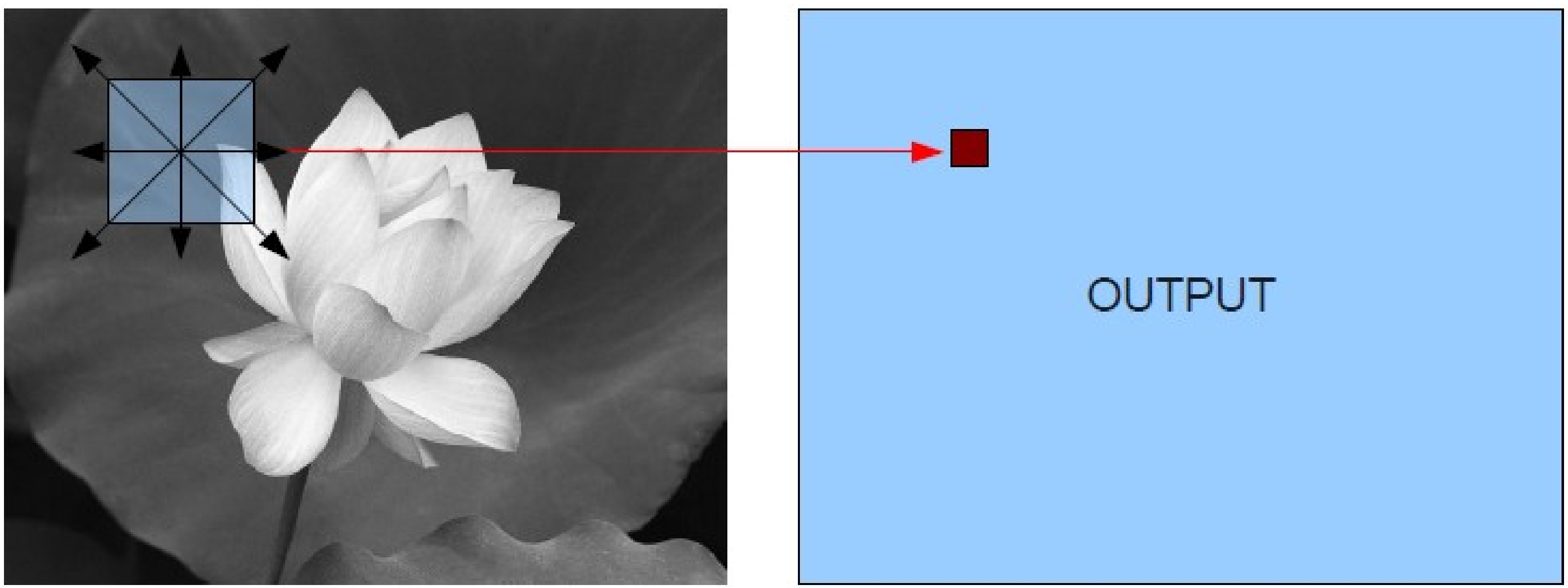
Filters

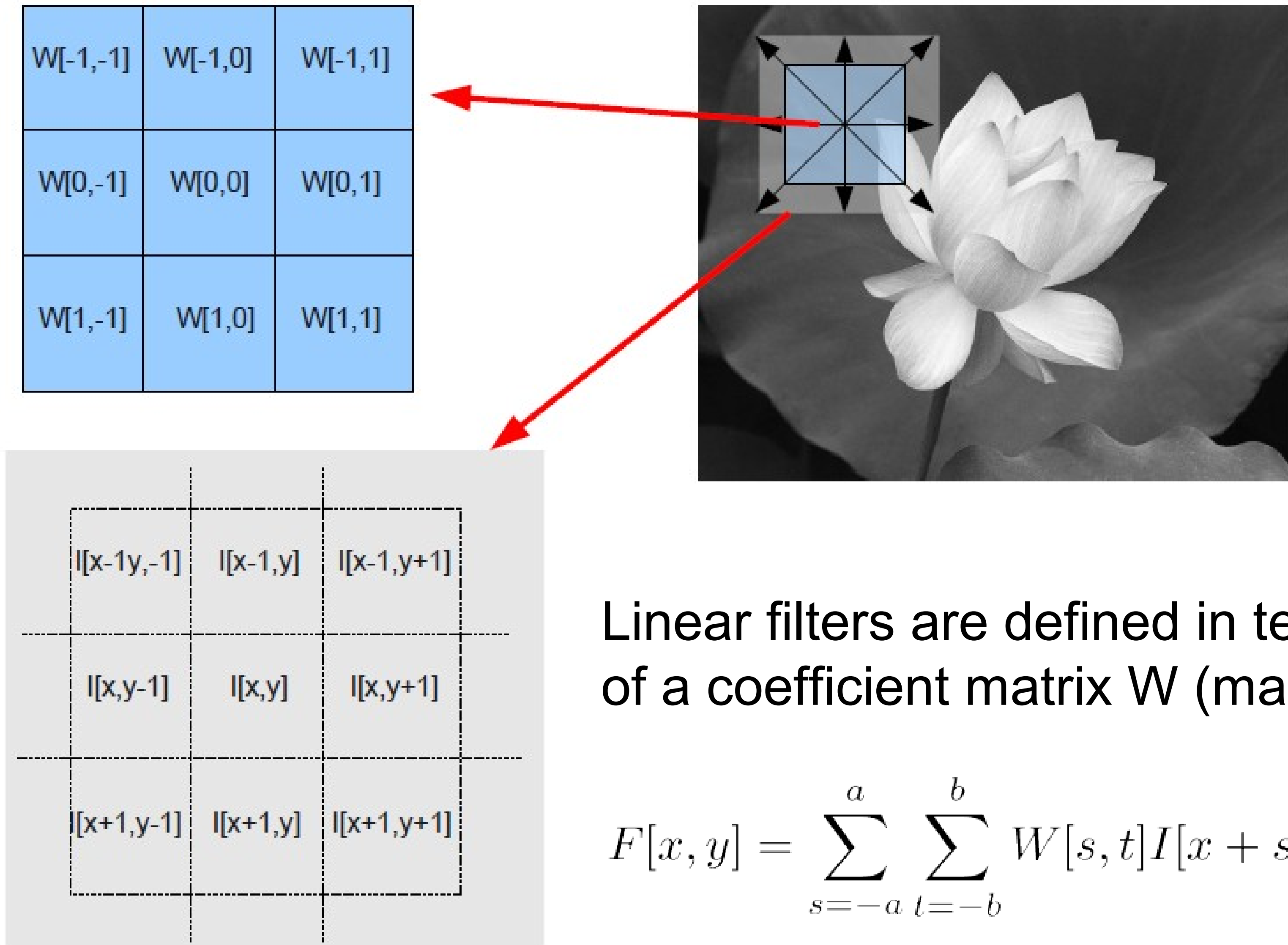
Andrea Torsello
DAIS
Università Ca' Foscari
via Torino 155,
30172 Mestre (VE)

Filters

- A filter is a transformation that is applied to a pixel and its neighborhood
- Generate a new image moving the filter over all the image



Linear Filters



Linear filters are defined in terms of a coefficient matrix W (mask)

$$F[x,y] = \sum_{s=-a}^{a} \sum_{t=-b}^{b} W[s,t] I[x+s, y+t]$$

Correlation and Convolution

- Correlation

$$(W \star I)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} W[s,t] I[x+s, y+t]$$

- Convolution

$$(W * I)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} W[s,t] I[x-s, y-t]$$

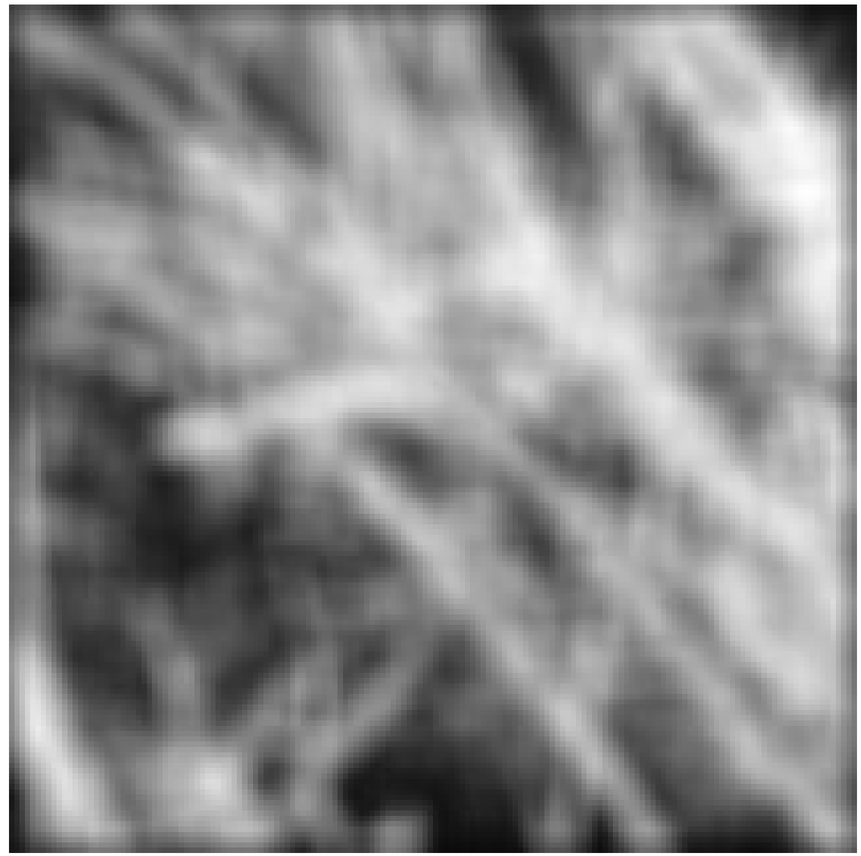
  – Convolution is equal to correlation with a mask rotated 180°

Properties of Convolution

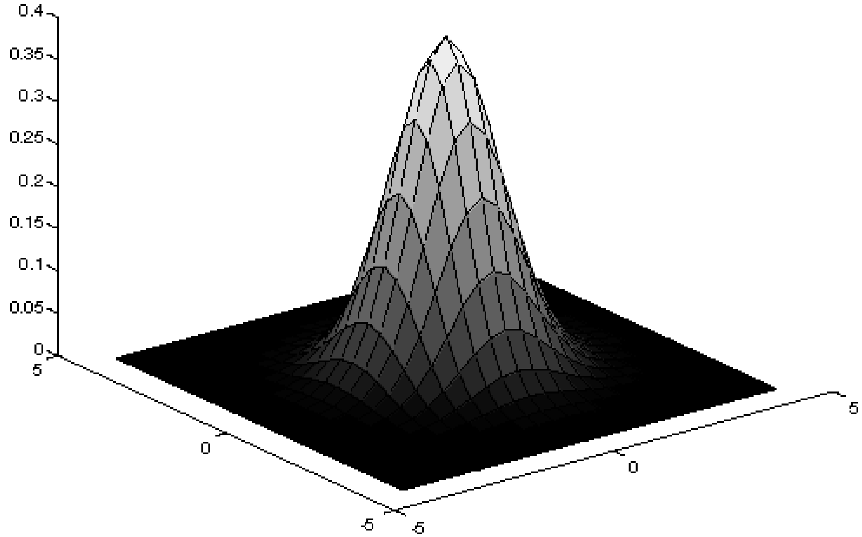
- Commutative
  - $F*G = G*F$
- Associative
  - $(F*G)*H = F*(G*H)$
- Linearity
  - $(a F + b G)*H = a F*H + b G*H$
- Translation invariance

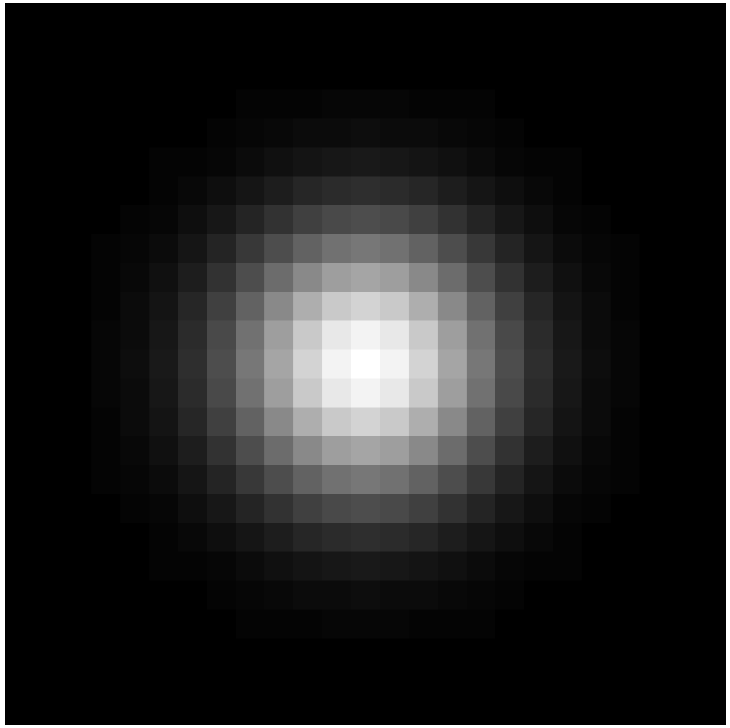
Average Filter

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



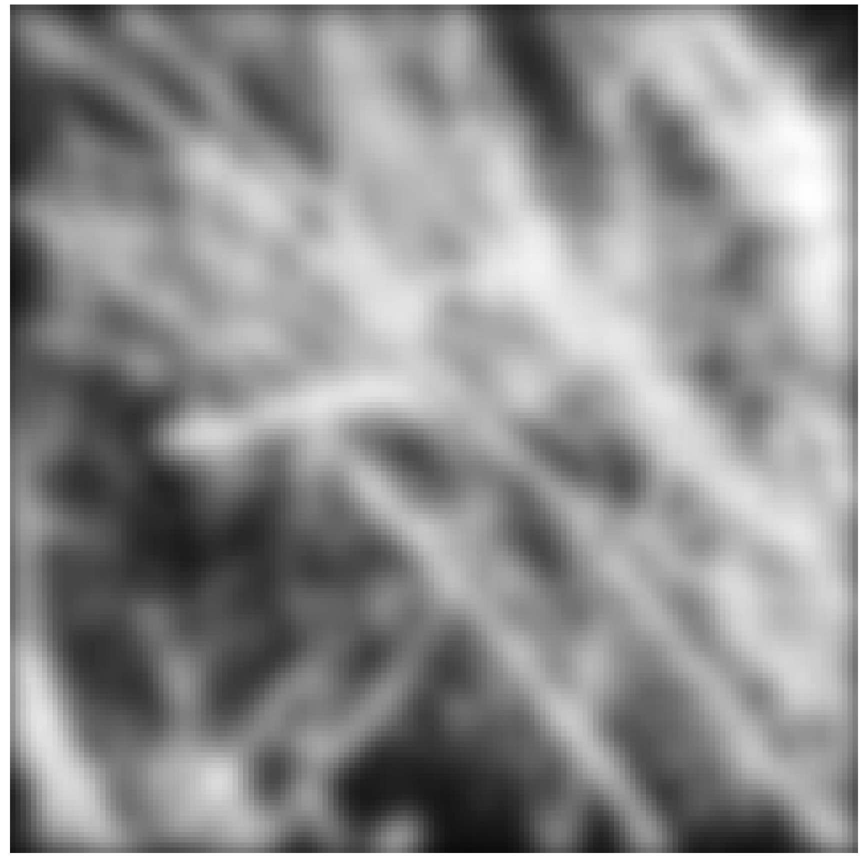
Gaussian Filter





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

Gaussian Filter



Unsharp Mask

UNIVERSITÀ
CA' FOSCARI
VENEZIA

- Smoothing can be used for sharpening

$$\bar{I}(x,y) = I(x,y) - f * I(x,y)$$
$$I_s(x,y) = I(x,y) + \bar{I}(x,y)$$

Noise

- Additive noise

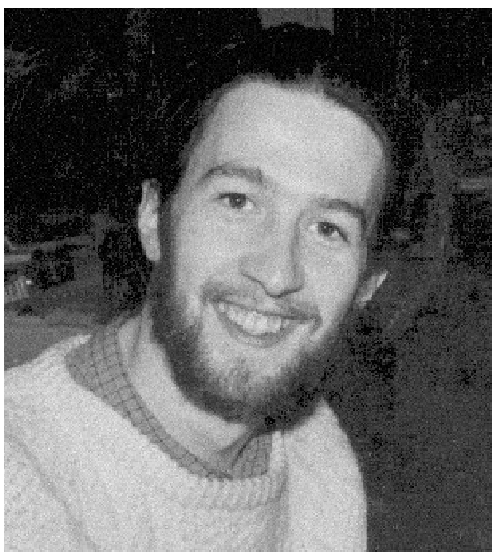
$$\tilde{I}(x,y) = I(x,y) + \omega$$

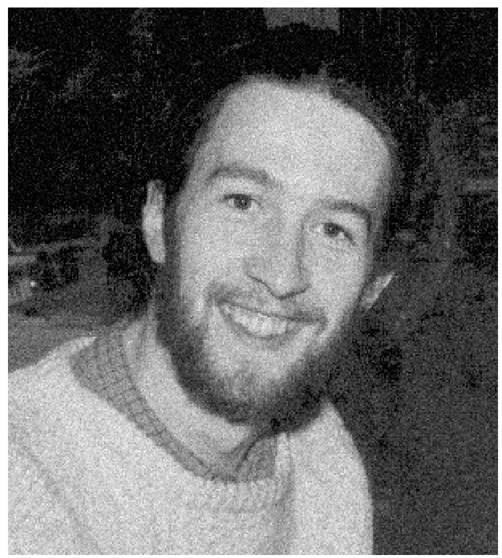
- Impulse noise

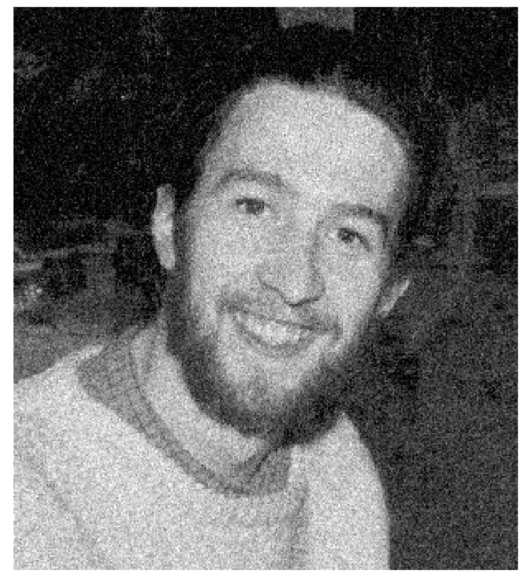
$$\tilde{I}(x,y) = \begin{cases} 0 \\ I(x,y) \\ 1 \end{cases}$$

- Other…

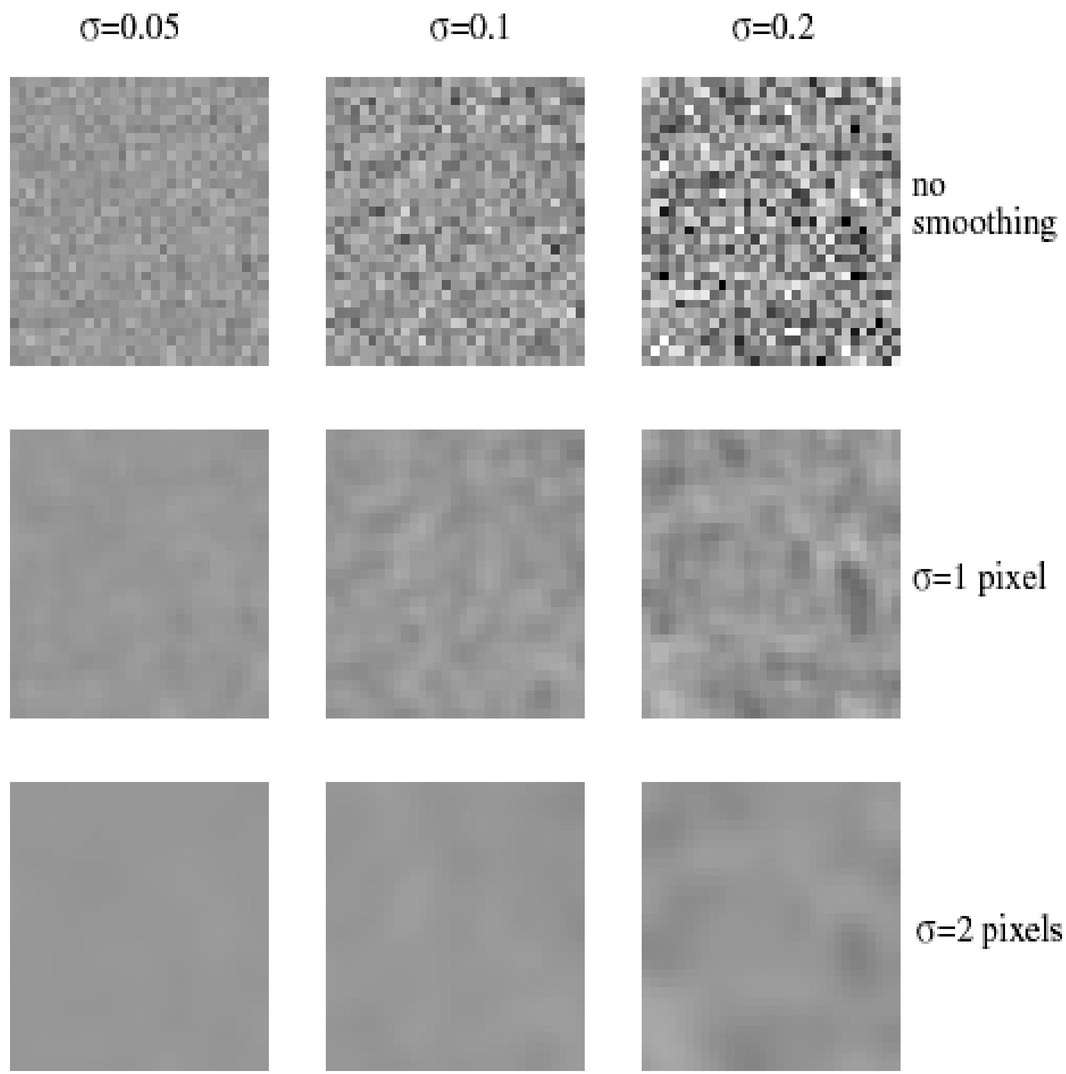
Additive noise







Smoothing and additive noise



Impulse noise

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Impulse - average

3 pixel

5 pixel

Impulso - Gaussian filter

5 pixel

Median Filter

- S_{xy} neighborhood of (x,y)
- Sort the intensity values of pixels S_{xy} into vector v_{xy}.

$$I'(x,y) = v_{xy}[½|S_{xy}|]$$

Median Filter

3x3 pixel

7x7 pixel

Median filter

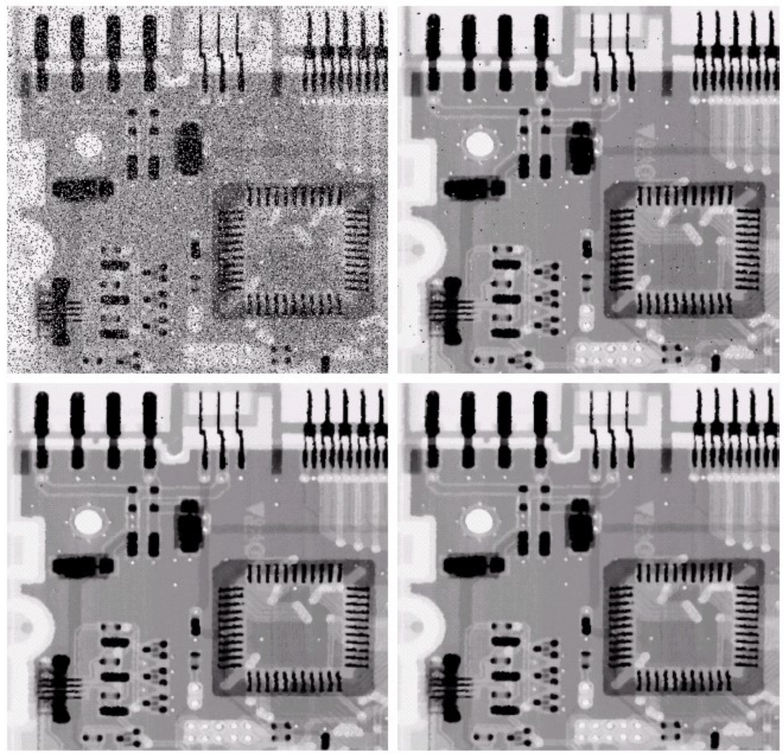
7x7 pixel

3x3 pixel applied 3 times

Alfa-trimmed mean filter

To eliminate bot additive and impulse noise use a robust estimate of the mean

- Eliminate the top and bottom $\alpha/2$ values
- Take the average of the remaining pixels

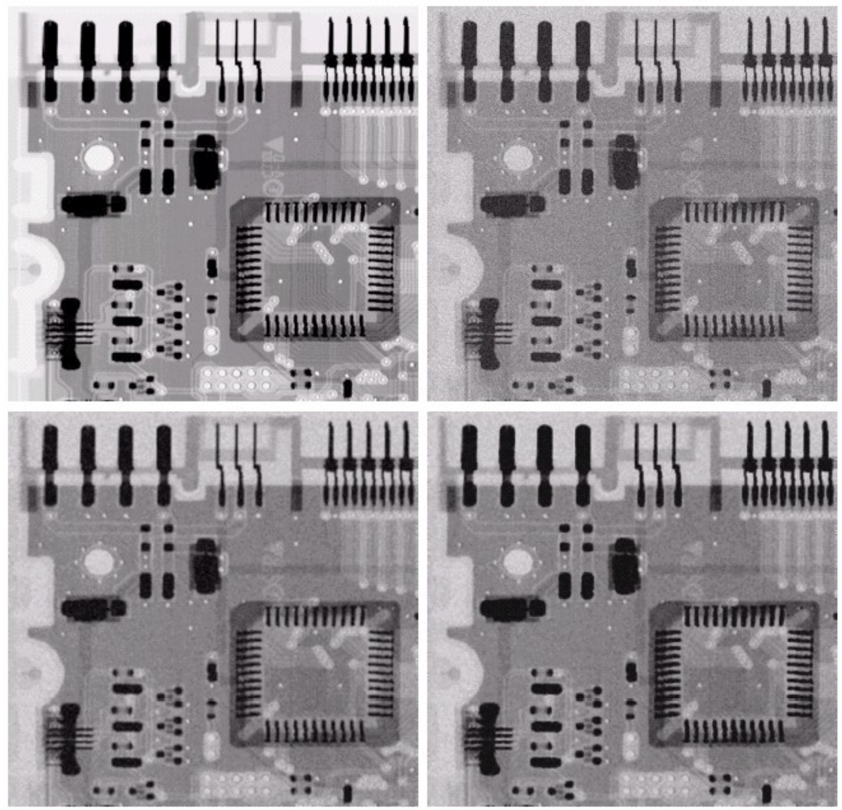


a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Geometric mean filter

$$\hat{f}(x,y)=\left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}}$$



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

UNIVERSITÀ CA' FOSCARI VENEZIA

Harmonic and Contraharmonic

Harmonic
(white impulse)

$$\hat{f}(x,y) = \frac{1}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

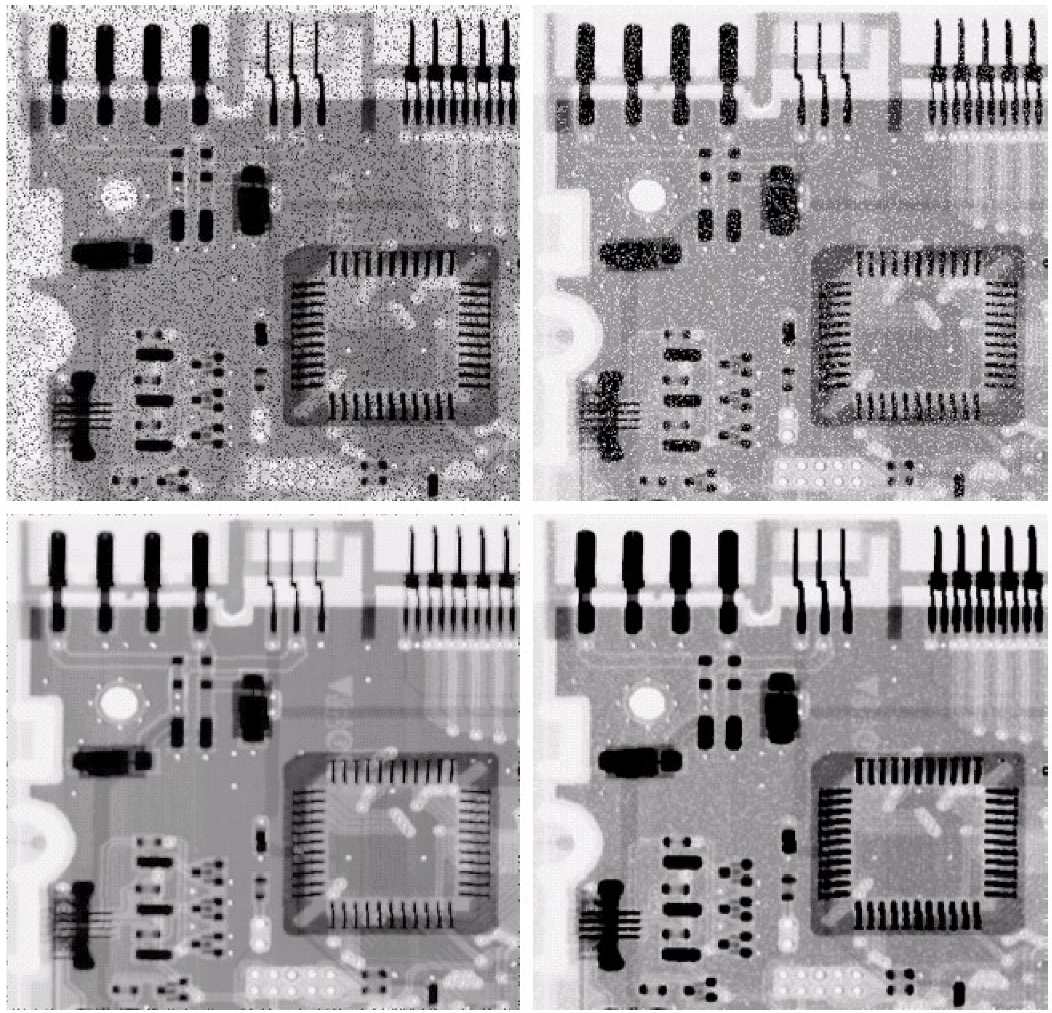
Contraharmonic
(black impulse)

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

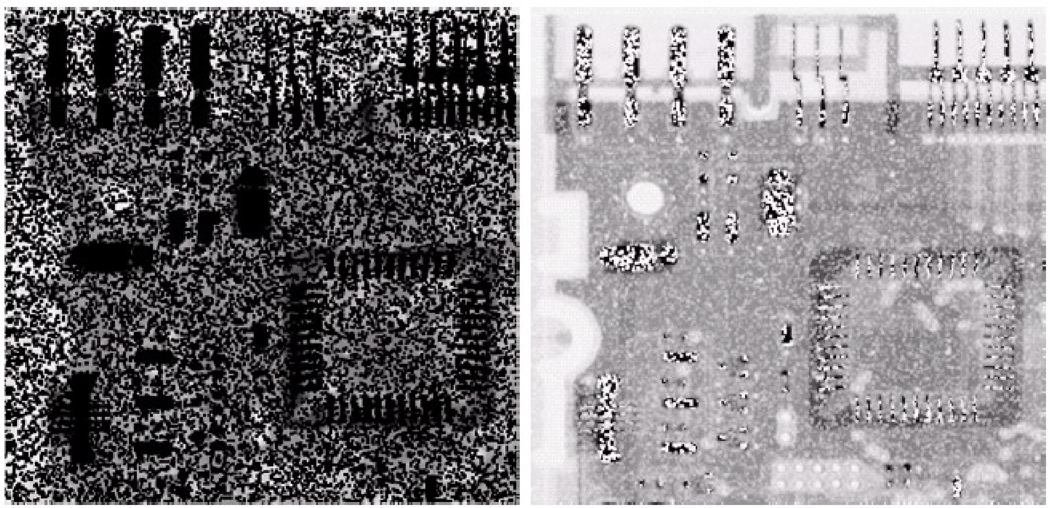
Harmonic and Contraharmonic

a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



Harmonic and Contraharmonic



a b

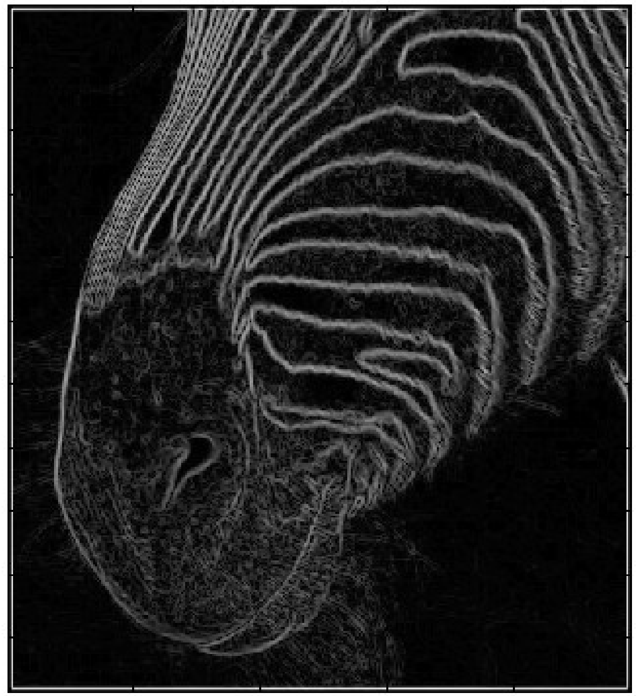
FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Differential filters

UNIVERSITÀ
CA' FOSCARI
VENEZIA

$$\frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n_1}, y_n) - f(x_n, y_n)}{\Delta x}$$



Differential filters

Roberts cross

−1	0
0	1

0	−1
1	0

Sobel

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Edge-detection

UNIVERSITÀ CA' FOSCARI VENEZIA

a
b c
d e
f g

FIGURE 10.8 A 3 × 3 region of an image (the z's are gray-level values) and various masks used to compute the gradient at point labeled z_5.

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

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

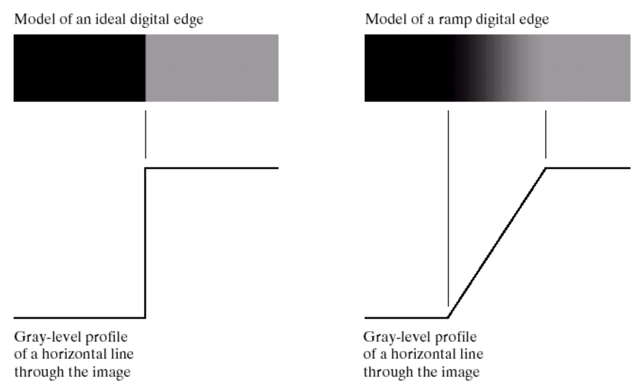
0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

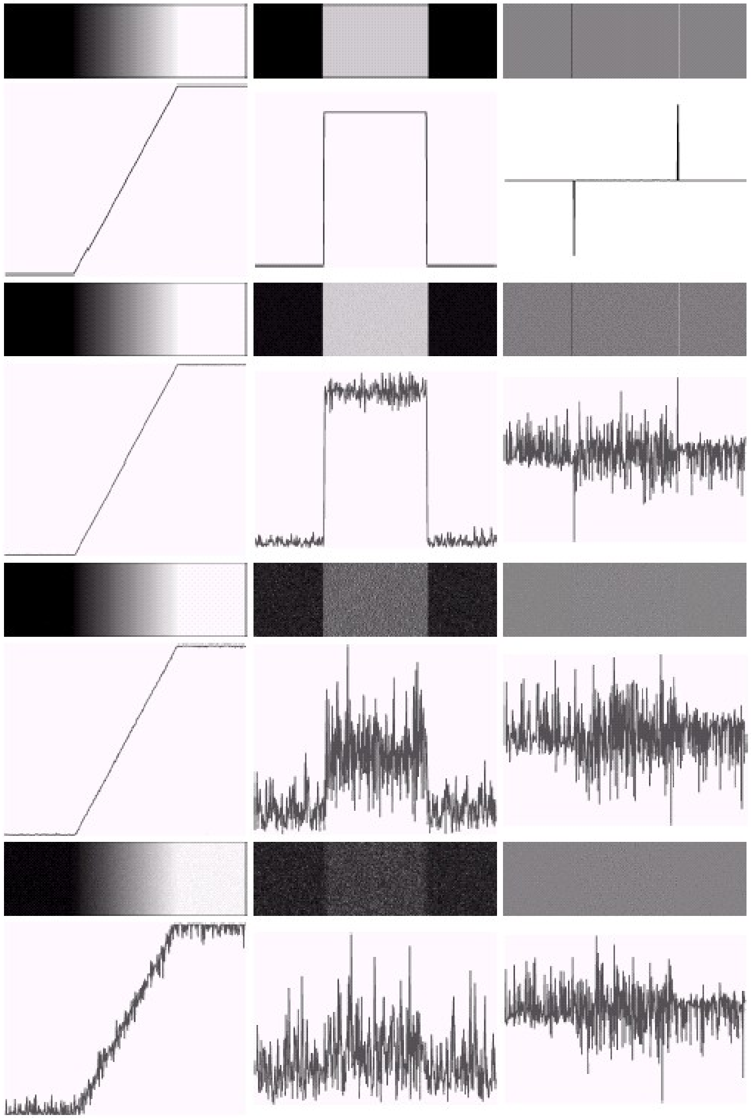
FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.



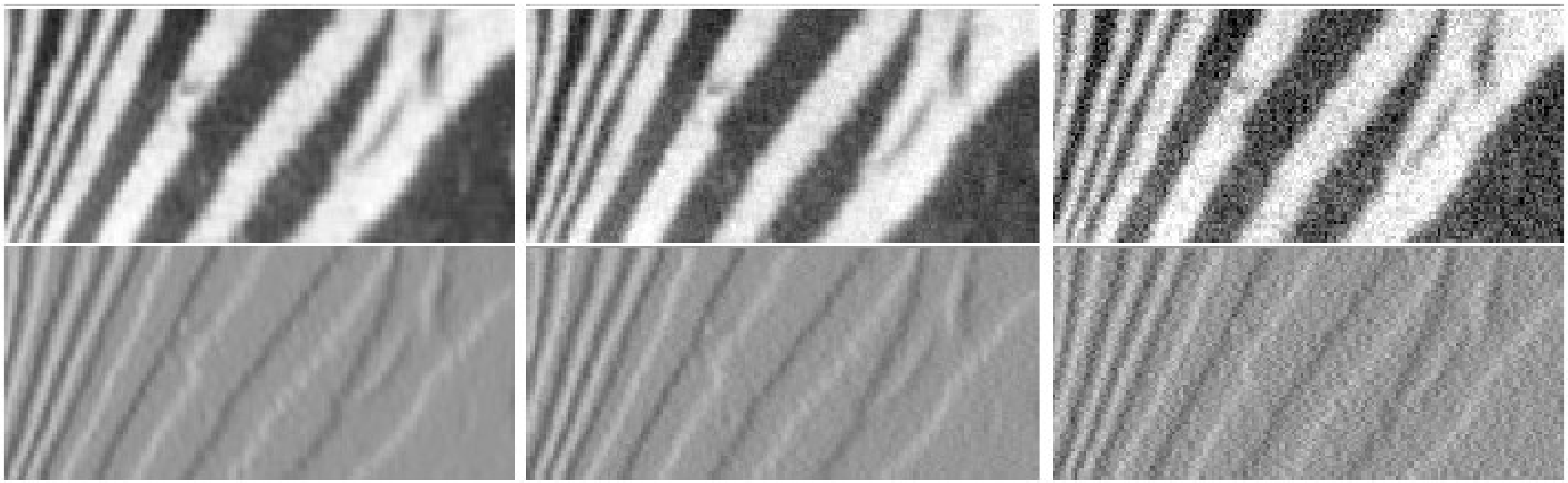
a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

Effects of Noise

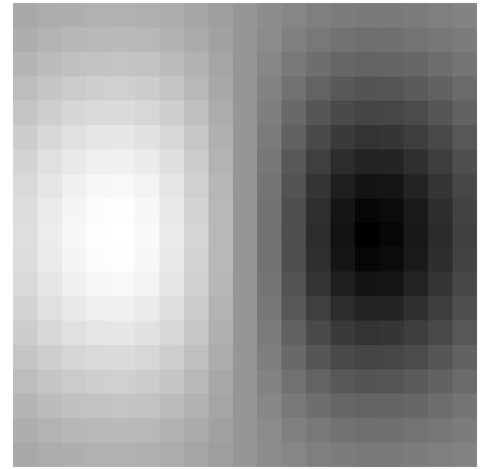


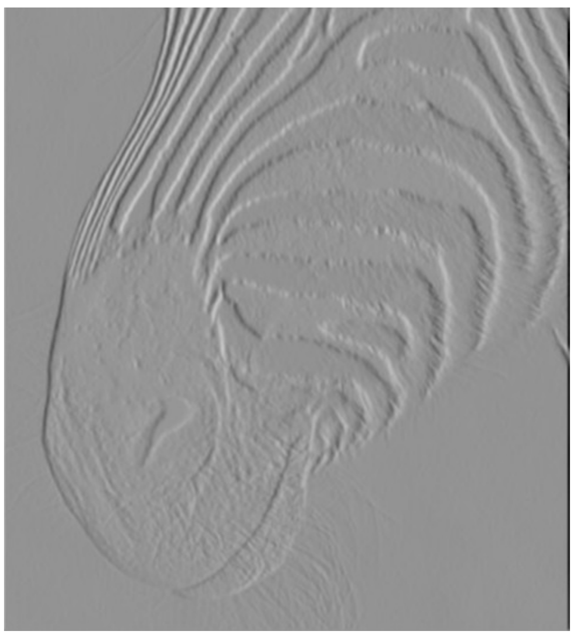
Derivatives and Noise

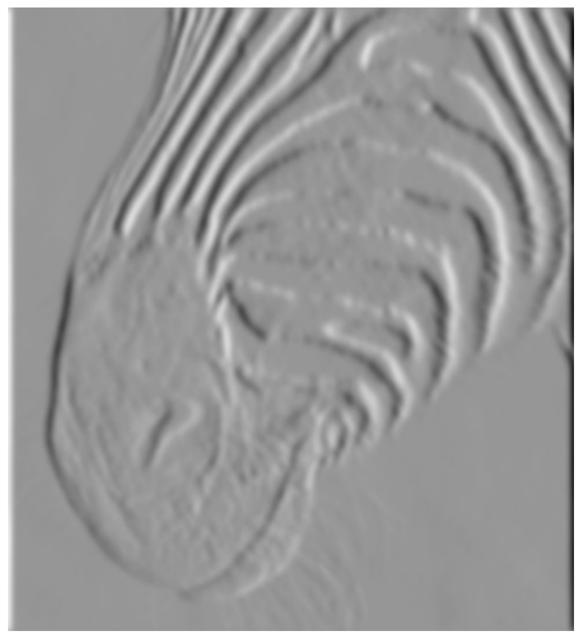


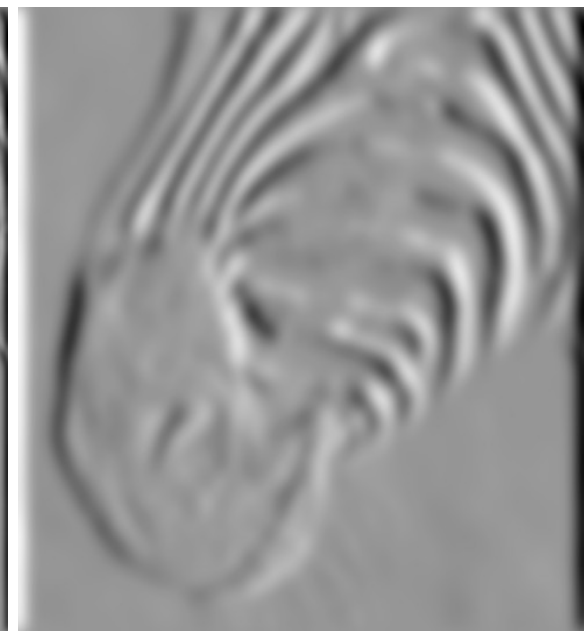
Smoothing + Differentiation

$$\frac{\partial}{\partial x}(f * I) = \left(\frac{\partial}{\partial x} f\right) * I$$

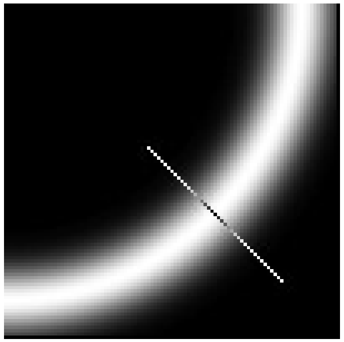






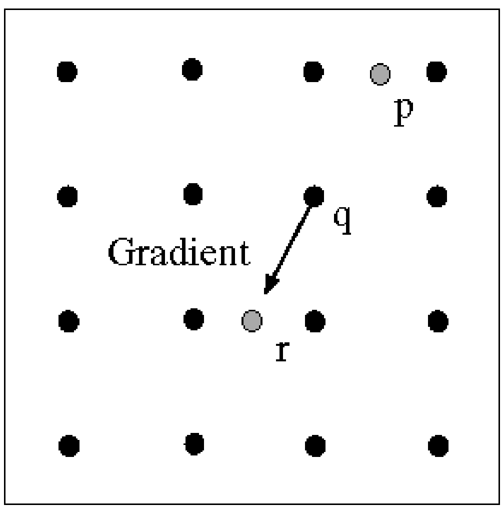


Non-maximal suppression

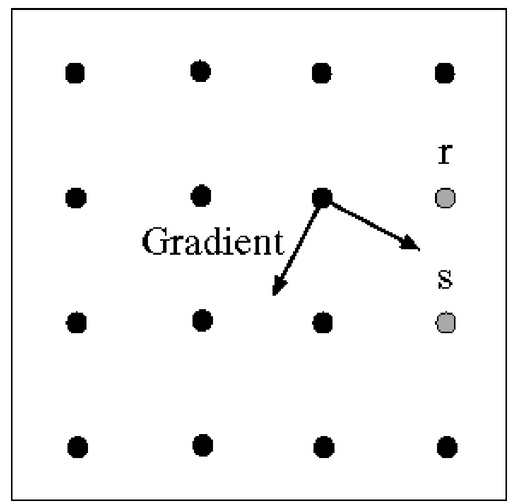


Boundaries are located at the maximum
of the gradient along the gradient direction

q is maximum if greater than p and r
(interpolation)



Follow the boundary profile along
Direction orthogonal to the gradient
(here r or s)



Laplacian Sharpening

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$I_s[x,y] = I[x,y] - c\nabla^2 I[x,y]$$

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

$\nabla^2 I = 0$ Along boundaries

- Look for pixels where the value of ∇^2I crosses 0