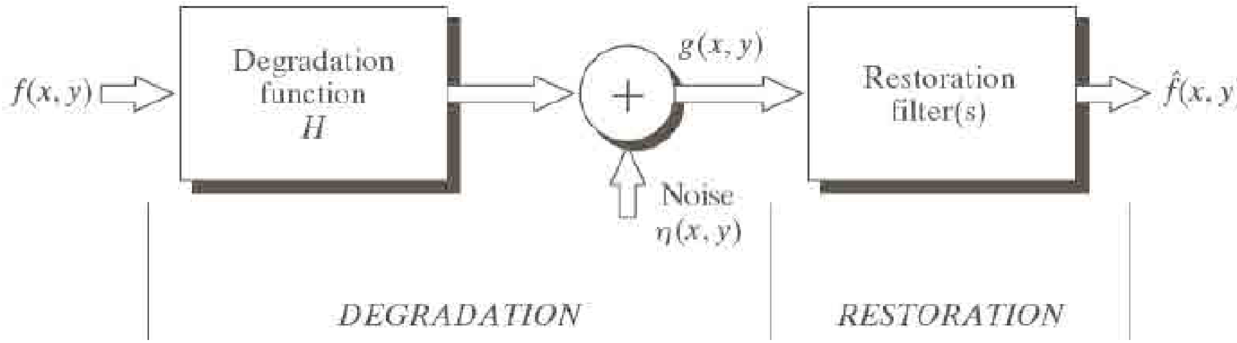
FIGURE 5.1 A model of the image degradation/restoration process.



If H is a linear, position-invariant process

Spatial domain representation of the degraded image:

$$g(x,y) = \boxed{h(x,y)} \star f(x,y) + \boxed{\eta(x,y)}$$

Frequency domain representation:

$$G(u,v) = \boxed{H(u,v)} F(u,v) + \boxed{N(u,v)}$$

Some Important Noise Probability Density Functions

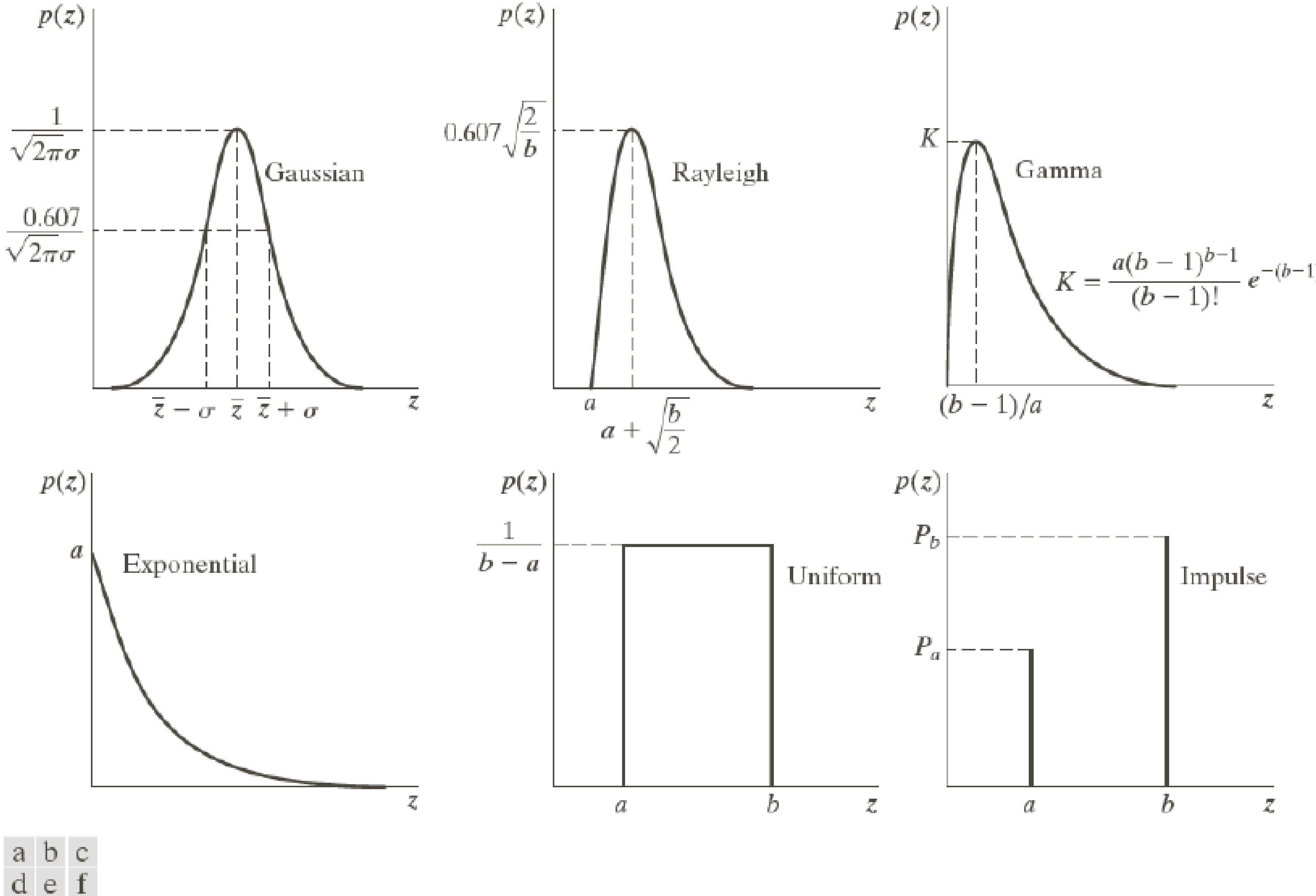


FIGURE 5.2 Some important probability density functions.

3. Restoration in the Presence of Noise Only - Spatial Filtering

When the only degradation is noise, the corrupted image is:

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

$$G(u, v) = F(u, v) + N(u, v)$$

When only additive noise present: *spatial filtering*

3.1 Mean Filters

Arithmetic mean filter

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

Geometric mean filter

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

3.1 Mean Filters

Harmonic mean filter

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

Works well for salt noise or Gaussian noise, but fails for pepper noise

Contraharmonic mean filter

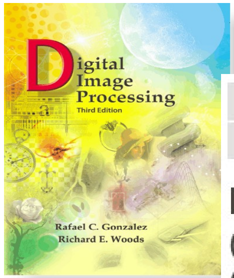
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

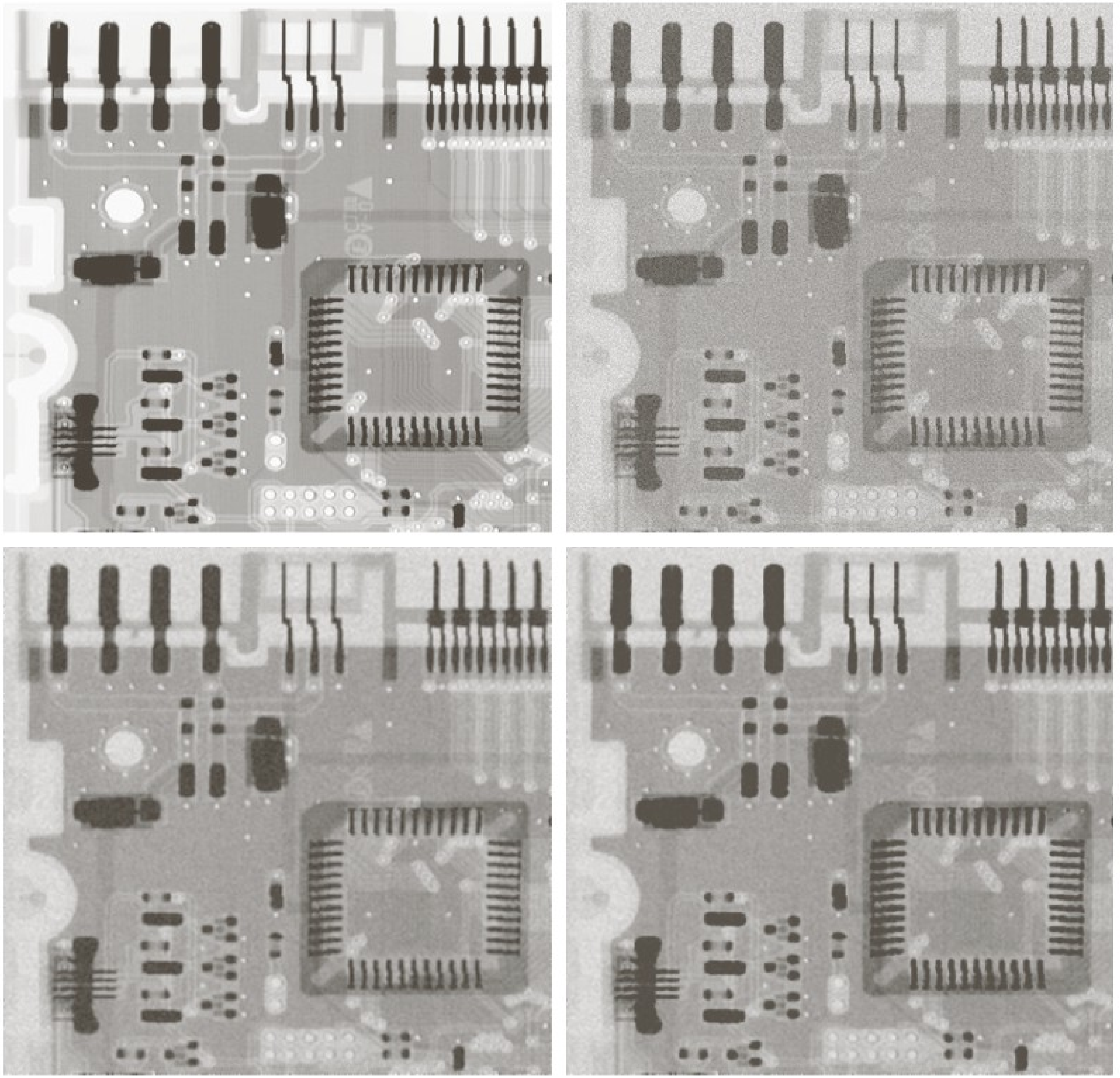
Q = *order* of the filter

Good for salt-and-pepper noise.

Eliminates pepper noise for $Q > 0$ and salt noise for $Q < 0$

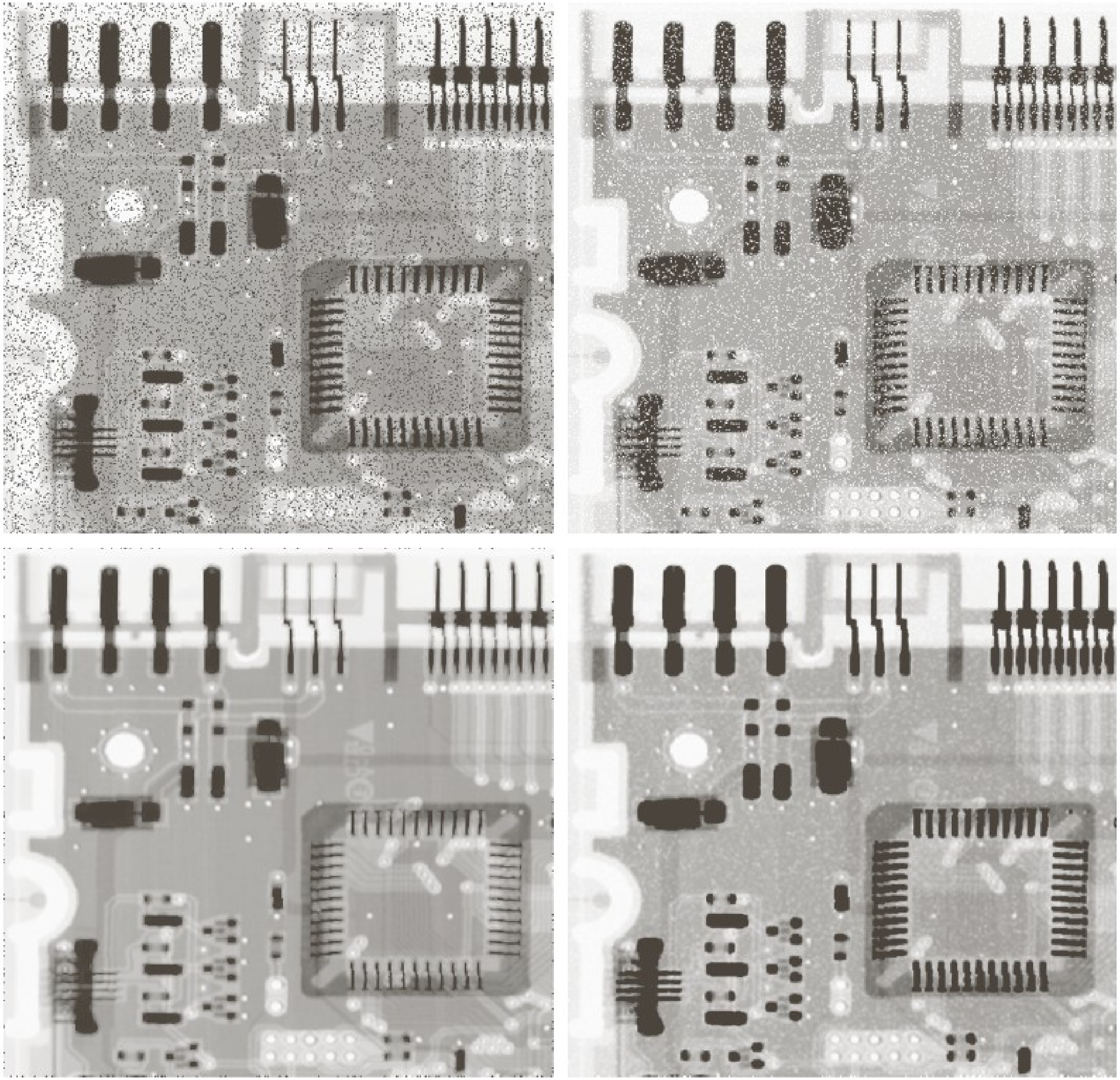
NB: cf. arithmetic filter if $Q = 0$, harmonic mean filter if $Q = -1$





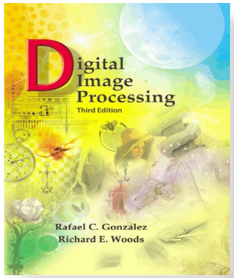
a b
c d

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.)



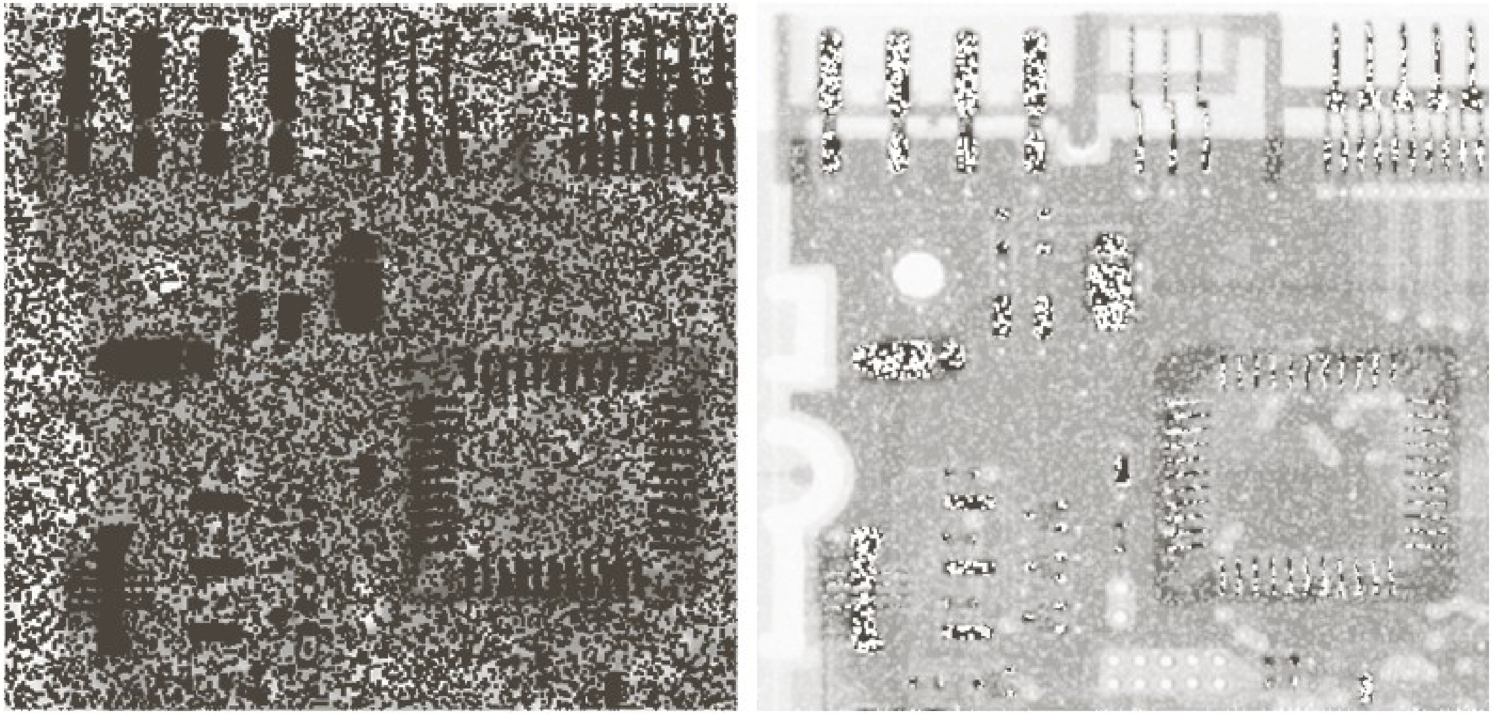
a b
c d

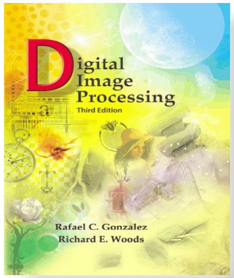
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



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 $Q = 1.5$.





3.2 Order-Statistic Filters

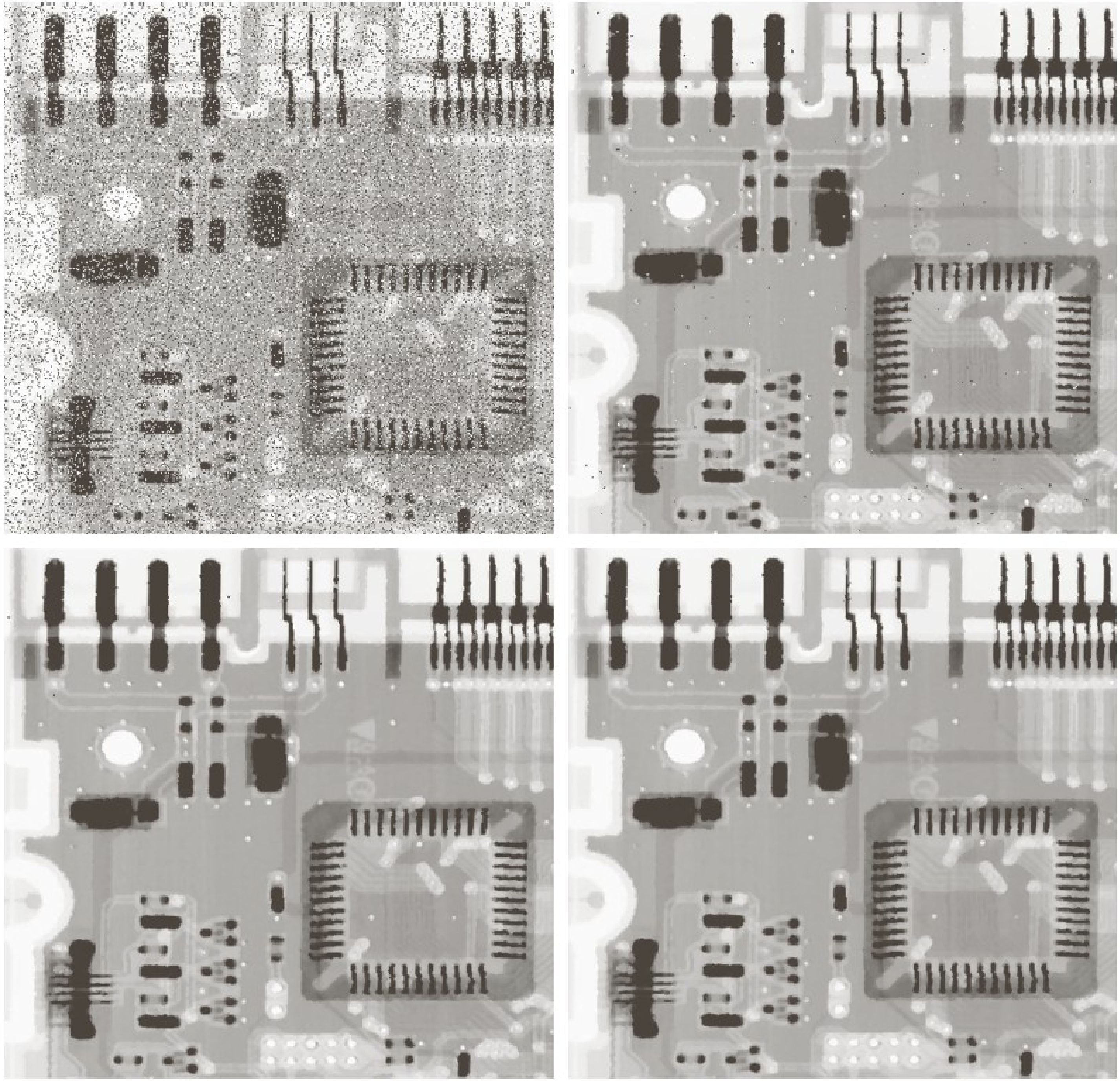
<u>Median filter</u> $\hat{f}(x,y) = \text{median}\{g(s,t)\}_{(s,t)\in S_{xy}}$

Particularly effective with bipolar and unipolar impulse noises

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.

2d pass

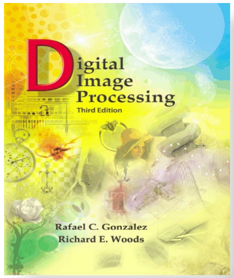


3.2 Order-Statistic Filters

Max filter: $\hat{f}(x, y) = \max\{g(s, t)\}_{(s,t)\in S_{xy}}$

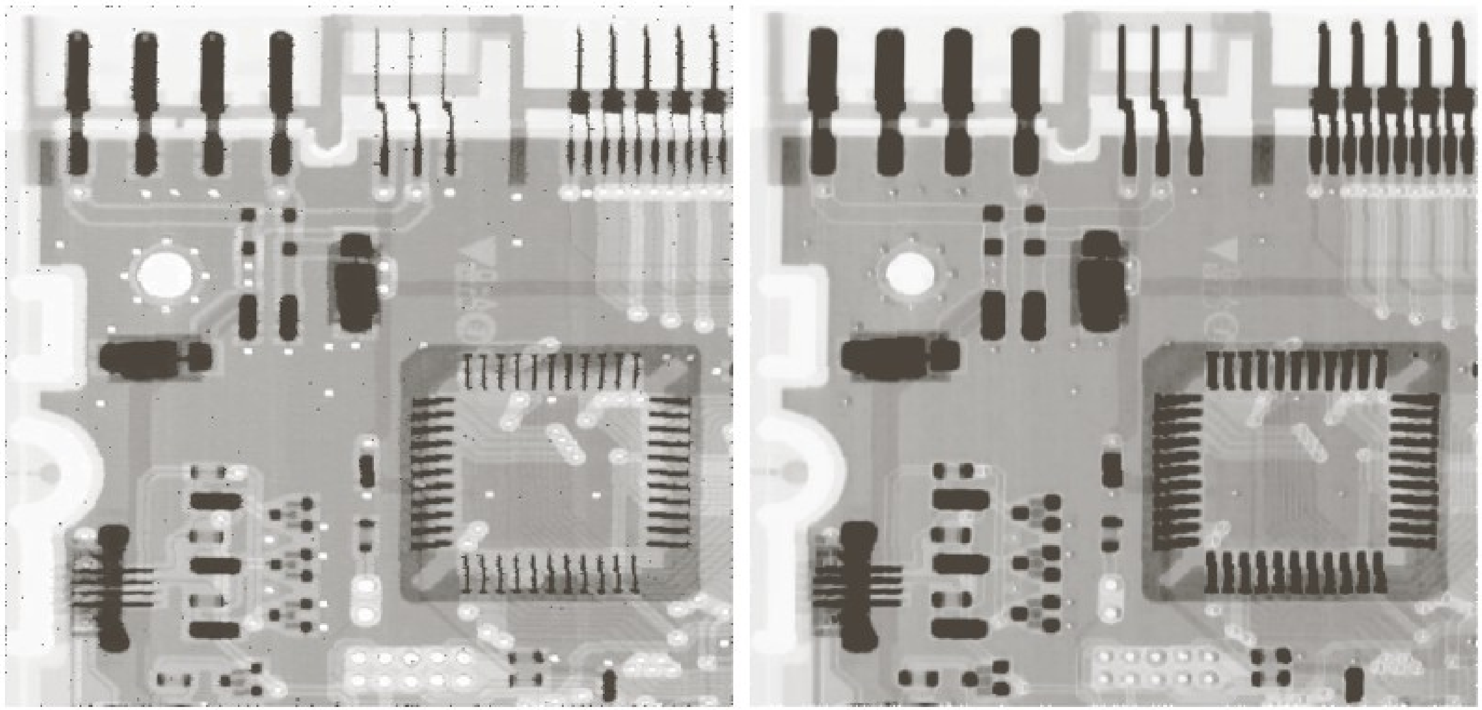
Useful for finding the brightest points in an image

Min filter: $\hat{f}(x, y) = \min\{g(s, t)\}_{(s,t)\in S_{xy}}$



Chapter 5

Image Restoration and Reconstruction



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Midpoint filter

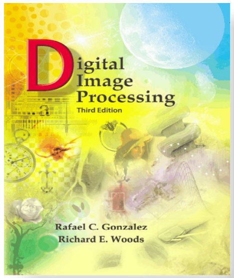
$$\hat{f}(x,y) = \frac{1}{2}\left[\max\{g(s,t)\}_{(s,t)\in S_{xy}} + \min\{g(s,t)\}_{(s,t)\in S_{xy}}\right]$$

NB: combines order statistics and averaging.
Works best for randomly distributed noise such as Gaussian or uniform

Alpha-trimmed mean filter

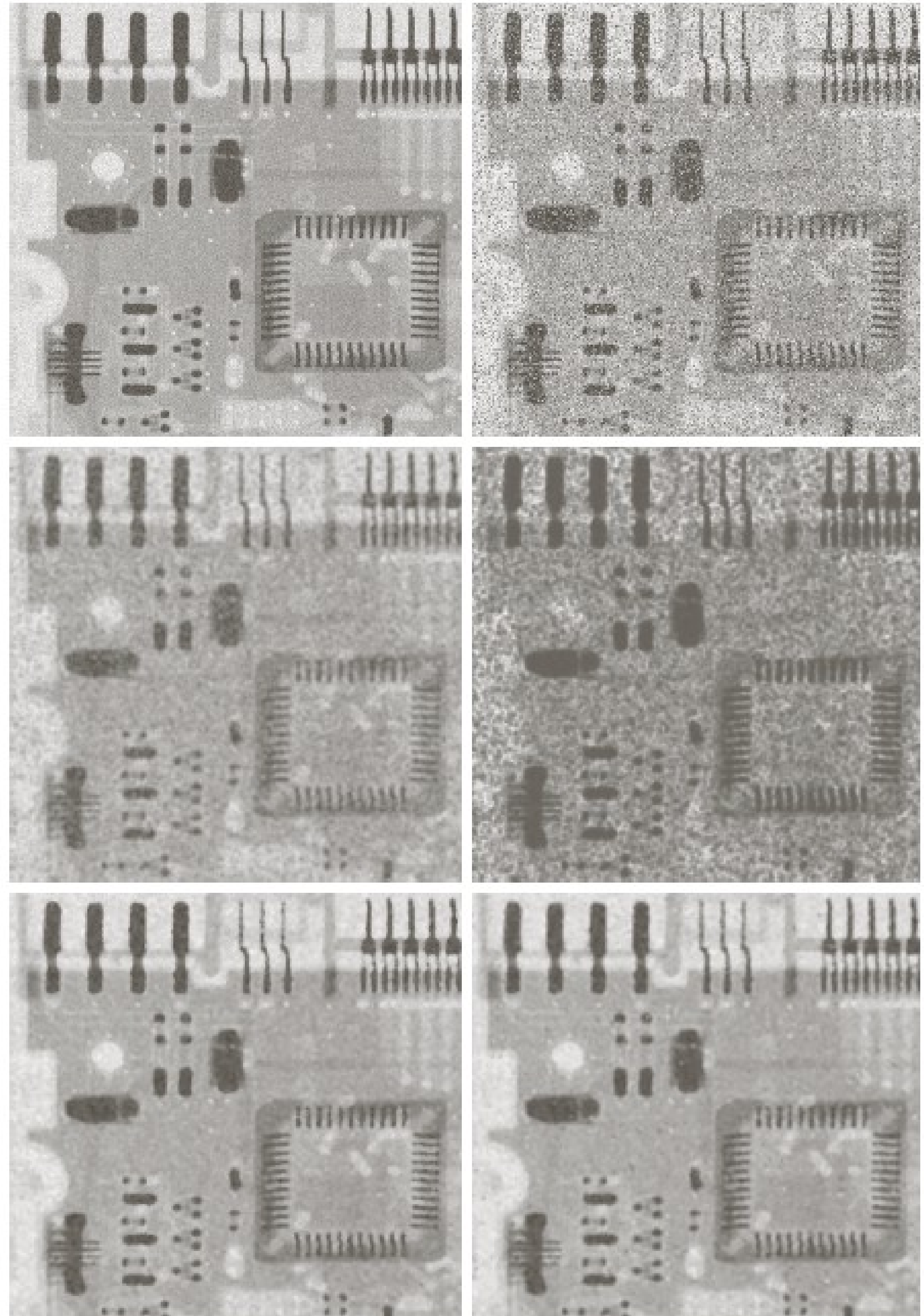
$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$$

Where g_r represents the image g in which the $d/2$ lowest and $d/2$ highest intensity values in the neighbourhood S_{xy} were deleted
NB: $d = 0$ => arithmetic mean filter, $d = mn-1$ => median filter
For other values of d, useful when multiple types of noise (e.g. combination of salt-and-pepper and Gaussian Noise)



a	b
c	d
e	f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5×5; (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

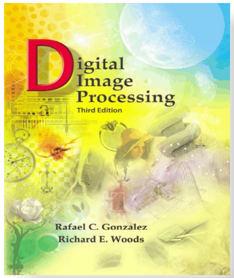


The filters discussed thus far are non-adaptive filters.

- whose coefficients are static, collectively forming the transfer function
- applied to an image regardless of how image characteristics vary from one point to another

In this section, two adaptive filters are discussed.

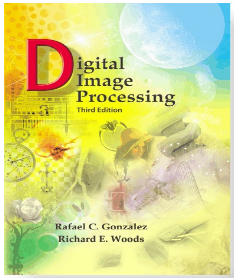
- whose behavior changes according to statistical characteristics of the image inside the filter window
- whose performance is superior to that of non-adaptive filters having discussed



5.3.3 Restoration in the Presence of Noise Only — **Example 5.4: Illustration of adaptive, local noise-reduction filtering**

Adaptive, local noise reduction filter

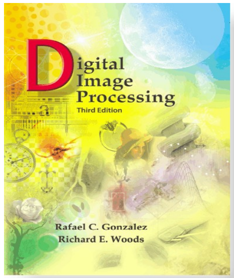
- The response of the filter at any point(x,y) on which the region is centered is to be based on four quantities:
  - g(x,y), the value of the noisy image at (x,y);
  - σ_η^2 , the variance of the noise corrupting f(x,y) to form g(x,y);
  - m_L , the local mean of the pixels in S_{xy};
  - σ_L^2 , the local variance of the pixels in S_{xy}.



Adaptive LNR Filters

- We want the behavior of the filter to be as follows:
  - If σ_η^2 is zero, the filter should return simply the value of g(x, y). This is the trivial, zero-noise case in wnich g(x, v) is equal to f(x, y).
  - If the local variance is high relative to σ_η^2 , the filter should return a value close to g(x,y). A nigh local variance typically is associated with edges, and these should be preserved.
  - If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in Sxy· This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced simply by averaging.
- An adaptive expression for obtaining $\hat{f}(x,y)$ based on these assumptions may be written as

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$



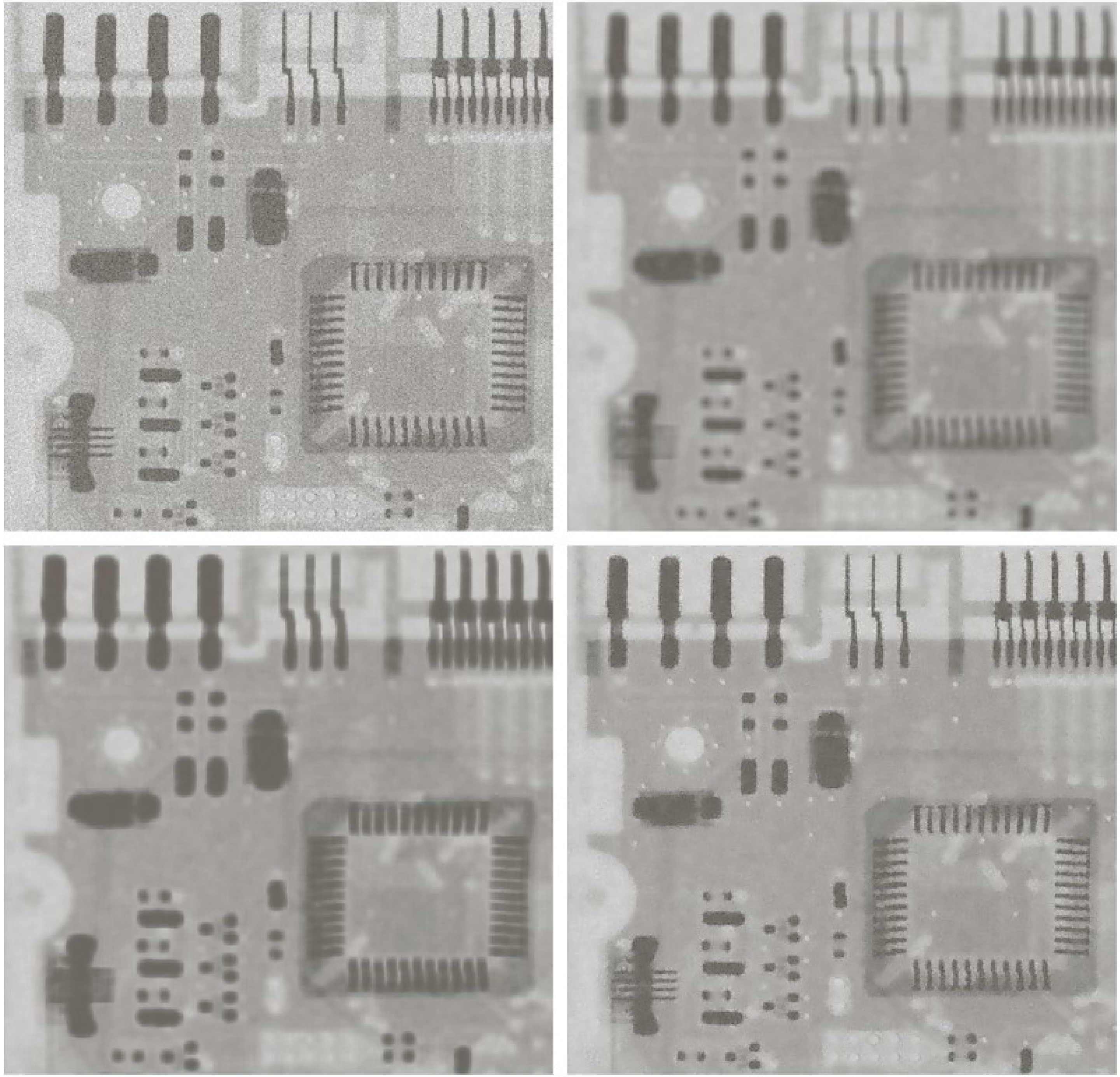
Adaptive LNR Filters

Adaptive, local noise reduction filter

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

- based on local mean (average intensity) m_L and local variance (contrast) σ_L^2
- if $\sigma_\eta^2 = 0$, no change
- if $\sigma_L^2 > \sigma_\eta^2$, edge, keep unchanged or less changed
- if $\sigma_L^2 \approx \sigma_\eta^2$, the m_L returns

- only the variance of corrupting noise $\hat{f}(x,y) \geq 0$ needed to be known or estimated
- assume $\sigma_\eta^2 \leq \sigma_L^2$, otherwise, set the ratio =1



a b
c d

FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

Adaptive Median Filter

Adaptive median filter

Stage A: $A1 = z_{med} - z_{min}$
$A2 = z_{med} - z_{max}$
If $A1 > 0$ AND $A2 < 0$, go to stage B
Else increase the window size
If window size $\leq S_{max}$ repeat stage A
Else output z_{med}

Stage B: $B1 = z_{xy} - z_{min}$
$B2 = z_{xy} - z_{max}$
If $B1 > 0$ AND $B2 < 0$, output z_{xy}
Else output z_{med}

z_{min} = minimum intensity value in S_{xy}
z_{max} = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity value at coordinates (x, y)
S_{max} = maximum allowed size of S_{xy}

- representing the restored pixel value at (x,y) by executing pseudocode
- the size of filter window is adaptive
- three purposes: to remove salt-and-pepper noise (capable of handling large P_a and P_b), to smooth non-impulsive noise, and to reduce distortion, e.g., excessive thinning or thickening of object boundaries
- performance is better than un-adaptive median filter
- for more detail, read text

Simple median filter does not generally work well if P_a or $P_b > 0.2$

Adaptive Median Filter

The key to understanding the mechanics of this algorithm is to keep in mind that it has three main purposes: to remove salt-and-pepper (impulse) noise, to provide smoothing of other noise that may not be impulsive, and to reduce distortion, such as excessive thinning or thickening of object boundaries.

Adaptive median filter can handle large probabilities (spatial density of impulse noise)

- Also, it helps preserving details better while smoothing non-impulse noise
- And this also changes the size of S_{xy} during operation, depending on some conditions as discussed

Adaptive Median Filter

$z_{\min}$ = minimum gray level value in S_{xy}

$z_{\max}$ = maximum gray level value in S_{xy}

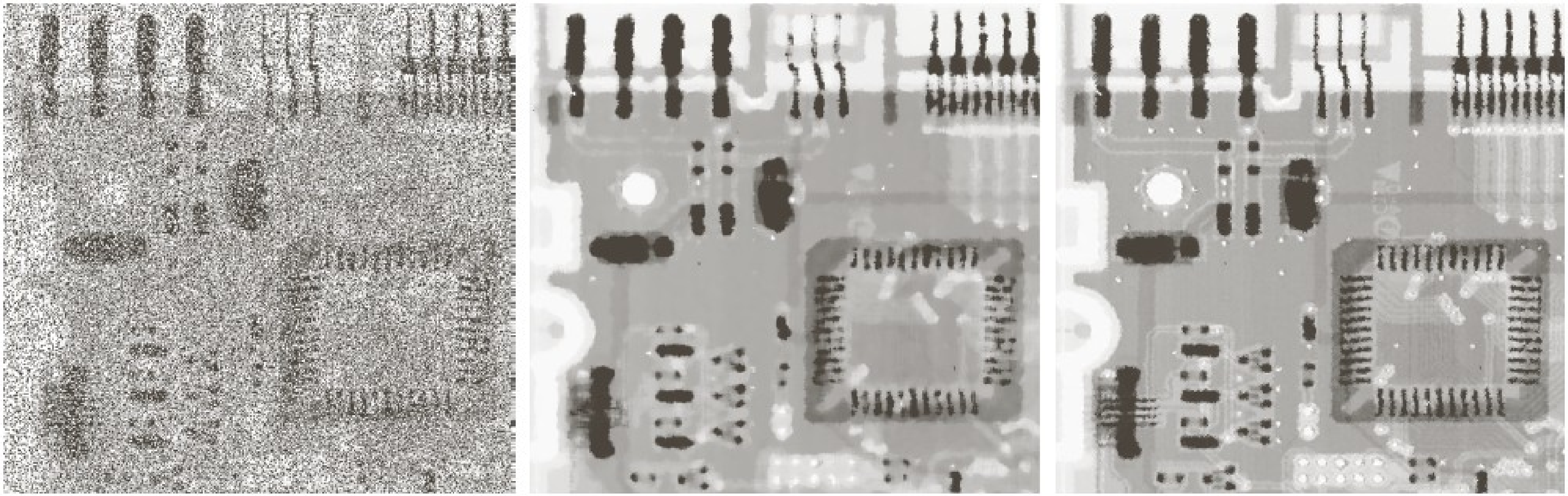
z_{med} = median of gray levels in S_{xy}

z_{xy} = gray level at coordinates (x, y)

$S_{\max}$ = maximum allowed size of S_{xy}.

Level A:
- $A1 = z_{\text{med}} - z_{\min}$
- $A2 = z_{\text{med}} - z_{\max}$
- If $A1 > 0$ AND $A2 < 0$, Go to level B
- Else increase the window size
- If window size $\leq S_{\max}$ repeat level A
- Else output z_{med}.

Level B:
- $B1 = z_{xy} - z_{\min}$
- $B2 = z_{xy} - z_{\max}$
- If $B1 > 0$ AND $B2 < 0$, output z_{xy}
- Else output z_{med}.



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

Non-local means

- A local neighborhood is specified by spatial position onlyA
- A non-local neighborhood is specified by the image pixels (or patches) of similar grayscale values
- Non-local techniques rely on redundancy in natural images
- Reference:

  Buades et al., “A non-local algorithm for image denoising”, CVPR 2005

Non-Local Means Adaptive Filter

Main idea - NL means

- Weighted average of pixels with similar neighborhood
- Weights depend on the similarity between the neighborhoods
- Assumes non-local neighborhood based on grayscale similarity
- Preserves edges better

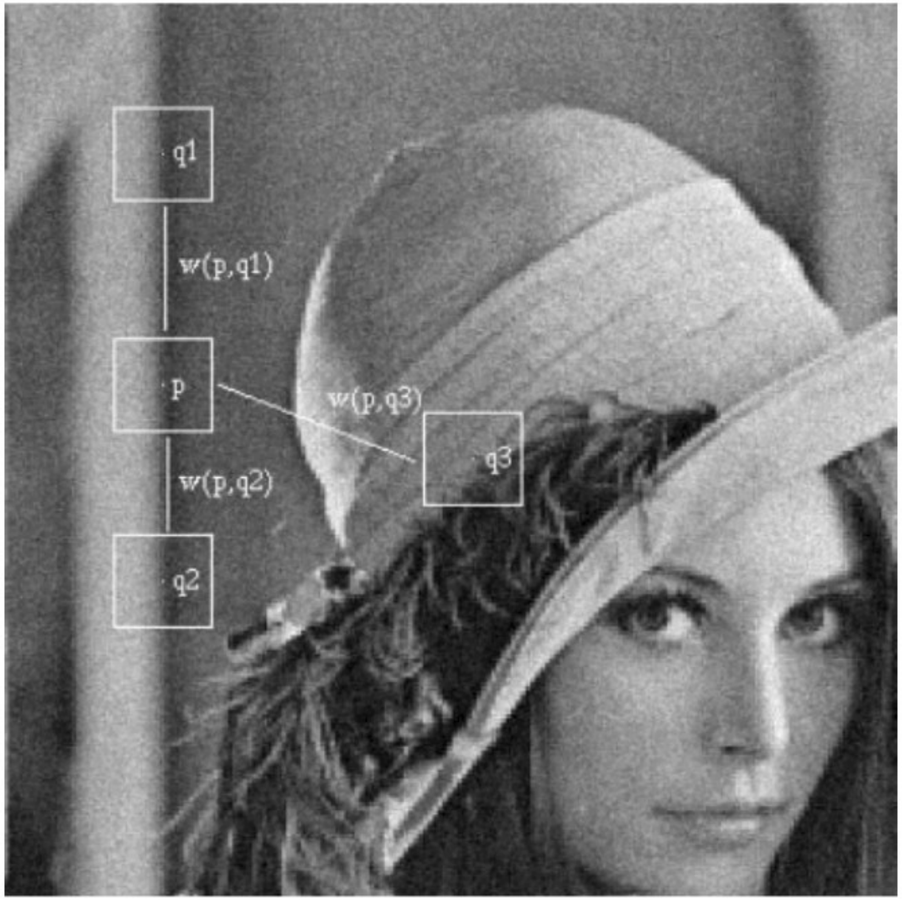


image source: Buades et al

Credits: T. Guha, EE604 (2016) Lecture notes/slides, IIT Kanpur and Buades et al. "A non-local algorithm for image denoising", CVPR 2005

Non-Local Means Adaptive Filter

- Consider a noisy image **I**. The grayscale value at pixel i is $v(i)$.

$$v(i) = u(i) + \eta(i)$$

- The NL-denoised estimate of $u(i)$:

$$\hat{u}(i) = \sum_{j \in I} w(i,j) v(j)$$

where $0 \leq w(i,j) \leq 1$ and $\sum_j w(i,j) = 1$

- $w(i,j)$ depends on the notion of similarity between i and j

Credits: T. Guha, EE604 (2016) Lecture notes/slides, IIT Kanpur and Buades et al. "A non-local algorithm for image denoising", CVPR 2005

Non-Local Means Adaptive Filter

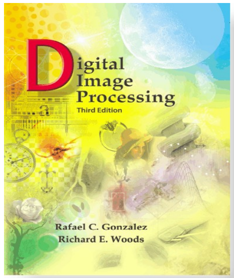
- Define neighborhoods $\mathcal{N}(i)$ and $\mathcal{N}(j)$

$$w(i,j) = \frac{1}{C} e^{-\frac{\|v(\mathcal{N}(i)) - v(\mathcal{N}(j))\|_2^2}{h^2}}$$

where C is a normalizing constant,

and h controls the decay of weight.

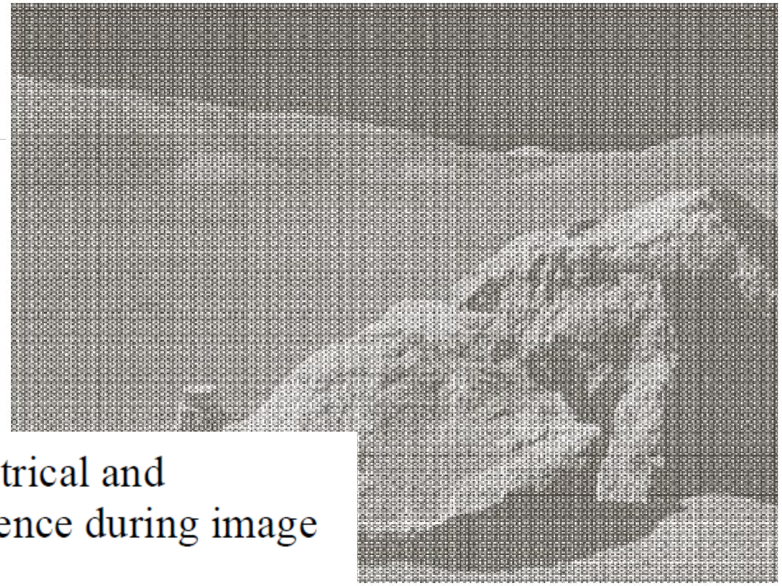
Credits: T. Guha, EE604 (2016) Lecture notes/slides, IIT Kanpur and Buades et al. "A non-local algorithm for image denoising", CVPR 2005

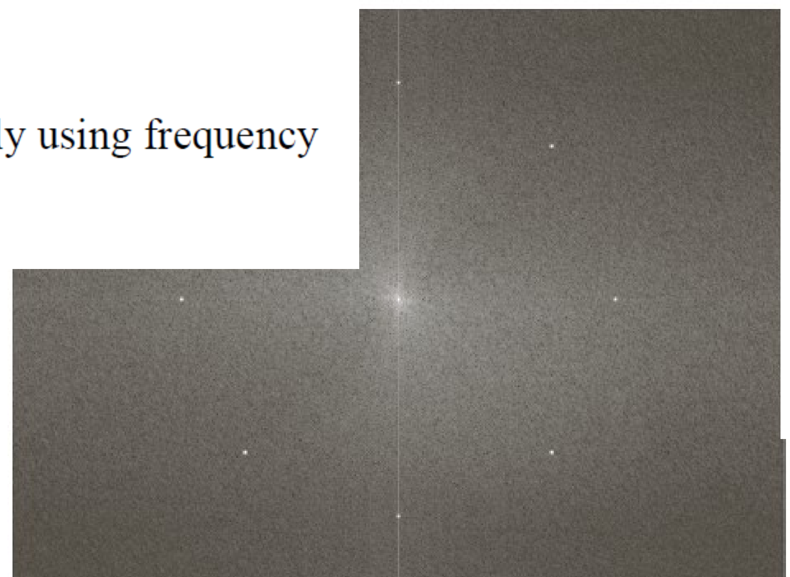


Periodic Noise

Typically comes from electrical and electromechanical interference during image acquisition

Can be reduced significantly using frequency domain filtering





a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Periodic Noise Reduction by Frequency Domain Filtering

- Bandreject filters remove or attenuate a band of frequencies about the origin of the Fourier transform. An ideal bandreject filter is given by the expression

$$H(u,v) = \begin{cases} 1 & \text{if } \; D(u,v) < D_0 - \dfrac{W}{2} \\ 0 & \text{if } \; D_0 - \dfrac{W}{2} \le D(u,v) \le D_0 + \dfrac{W}{2} \\ 1 & \text{if } \; D(u,v) > D_0 + \dfrac{W}{2} \end{cases}$$

- Where D(u,v) is the distance from the origin of the centered frequency rectangle, W is the width of the band, and D0 is its radial center.

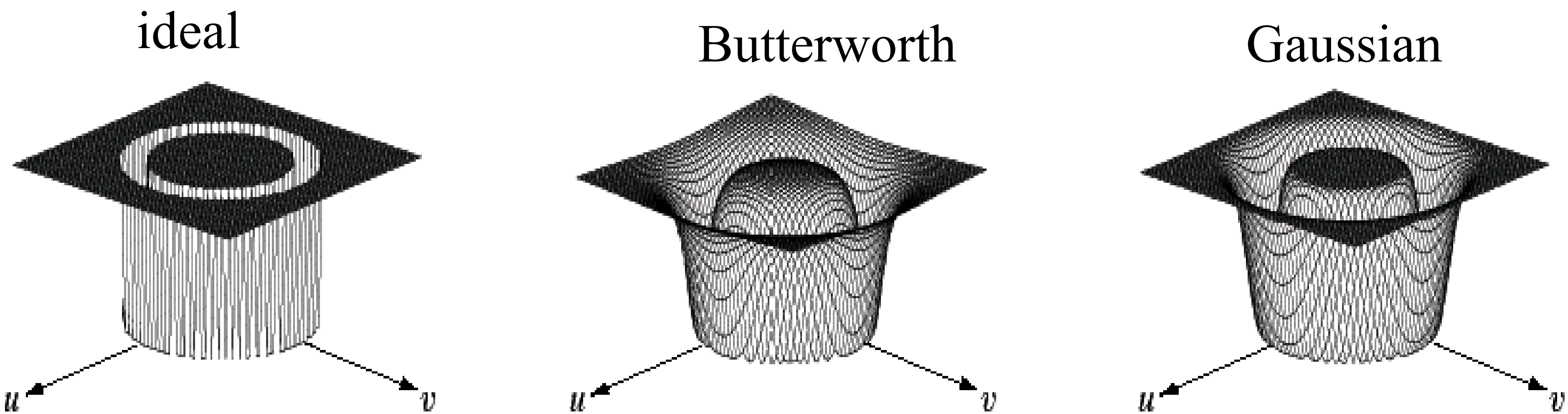
TABLE 4.6
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

A *bandpass* filter is obtained from a *bandreject* filter as:

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Band-Reject filters



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

(a) Ideal bandpass filter:

$$H_{\mathrm{IBP}}(u,v) = \begin{cases} 0 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 1 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 0 & D(u,v) > D_0 + \frac{W}{2} \end{cases}.$$

(b) Butterworth bandpass filter:

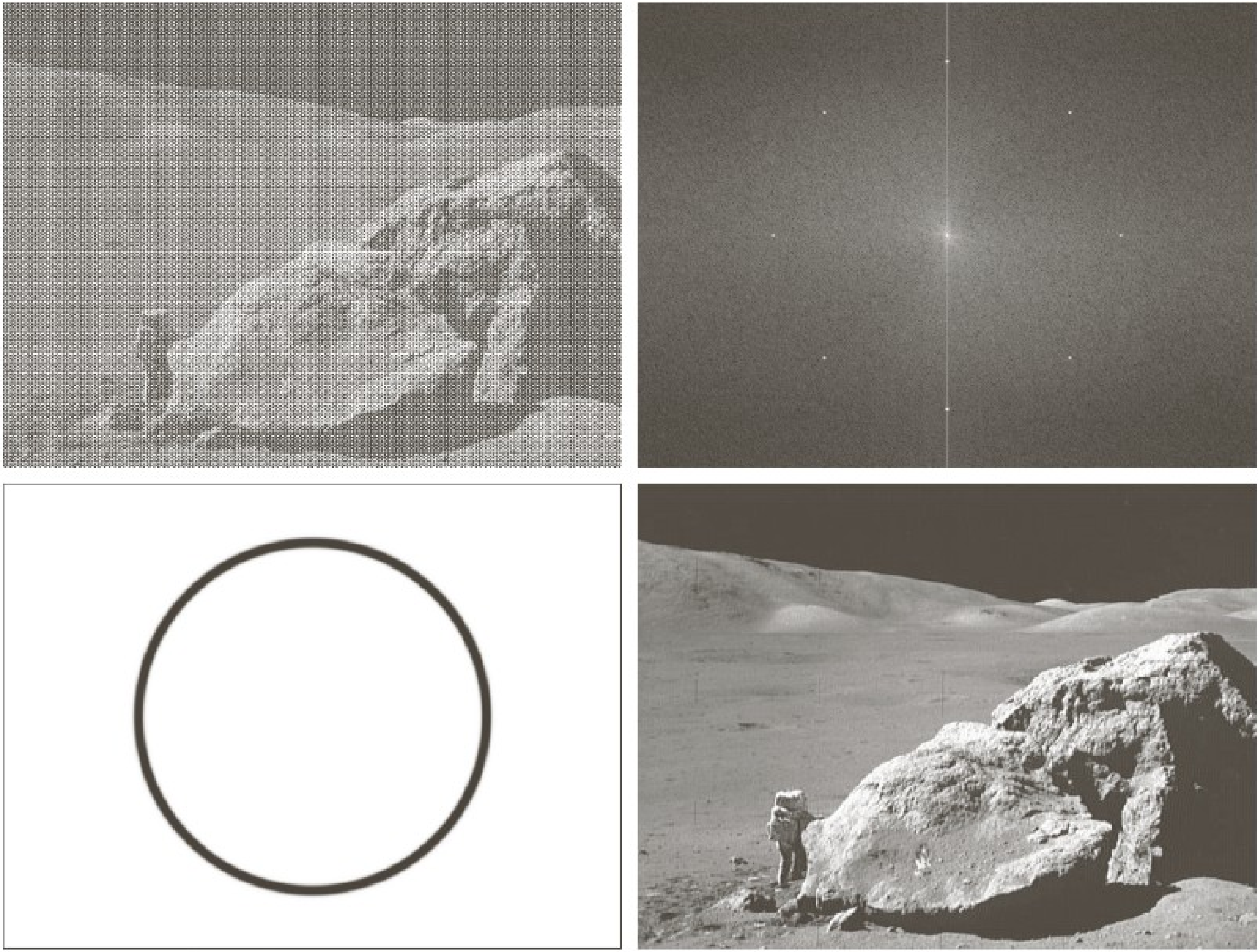
$$H_{\text{BBP}}(u,v) = 1 - \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}$$

$$= \frac{\left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}.$$

(c) Gaussian bandpass filter:

$$H_{\text{GBP}}(u,v) = 1 - \left[1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]^2}\right]$$

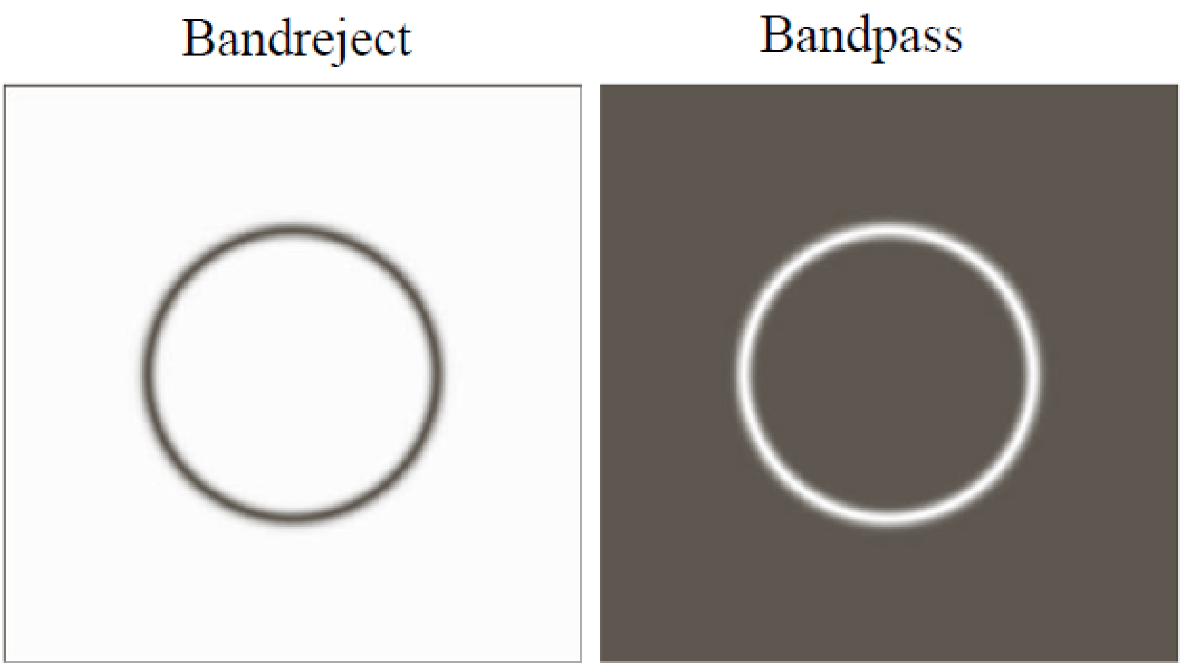
$$= e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]^2}$$



a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Example: Bandreject Gaussian filter



a b

FIGURE 4.63 (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

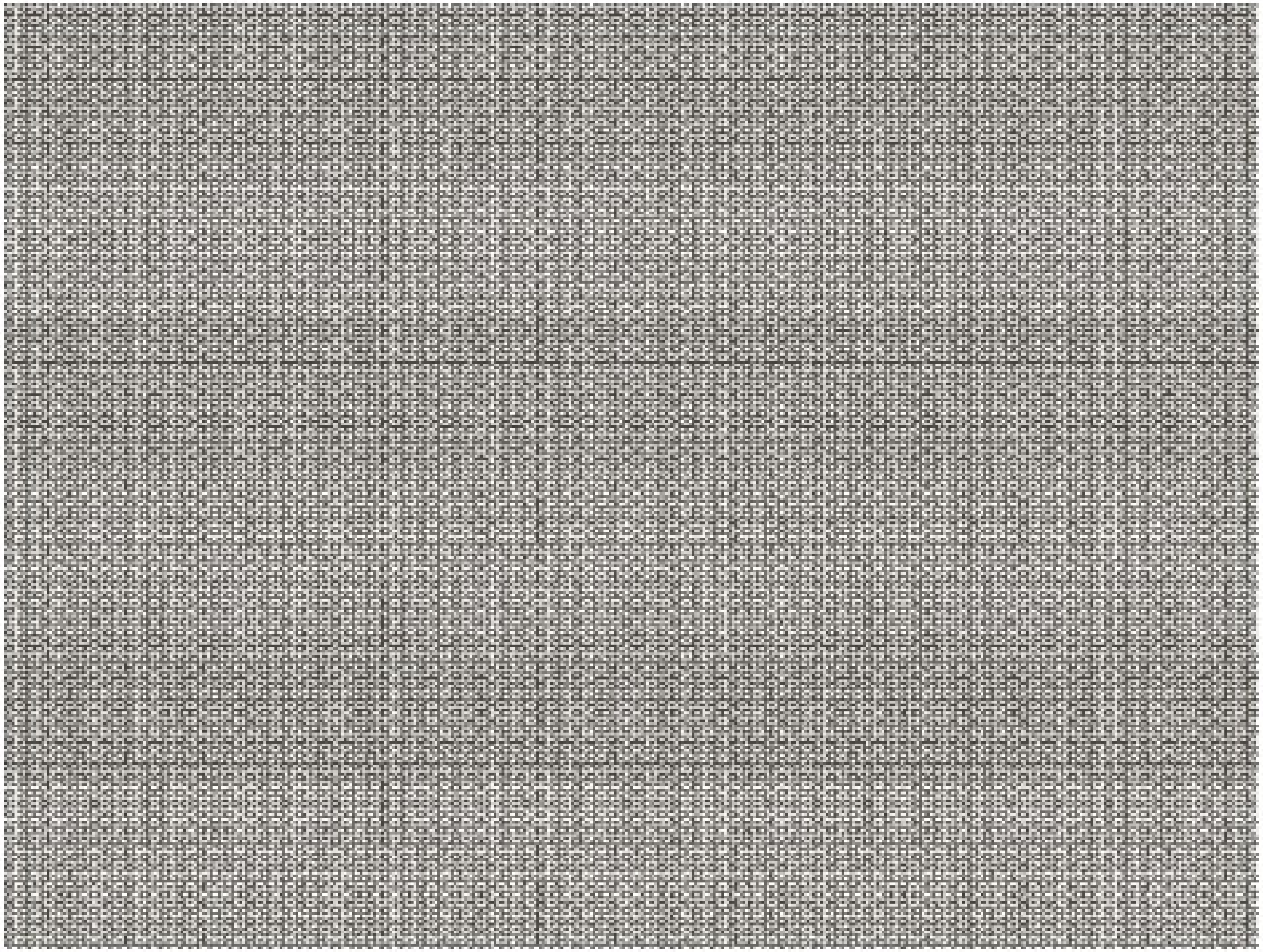
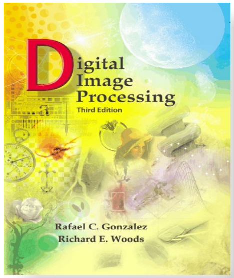
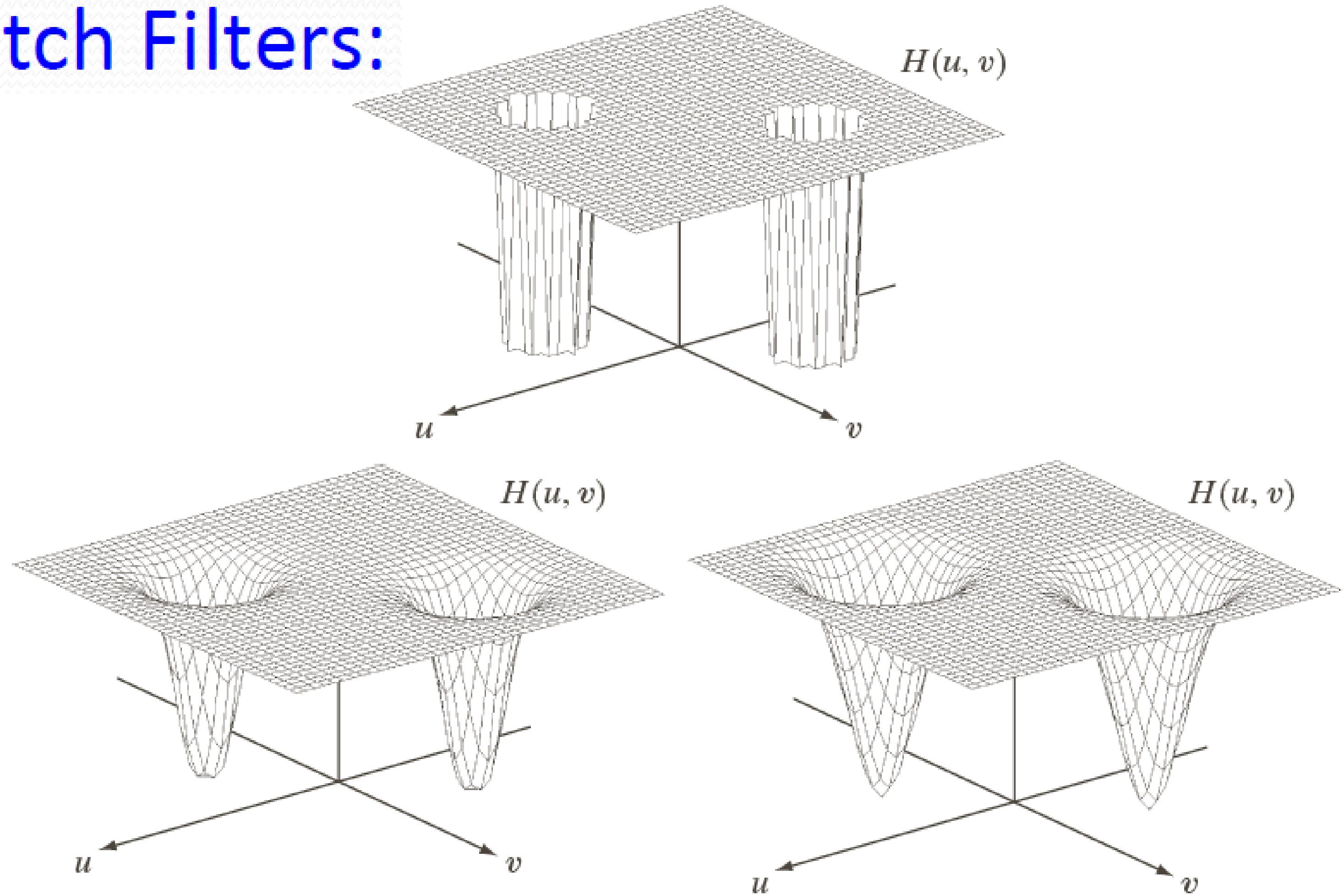


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



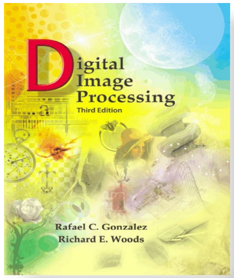
■ Performing straight bandpass filtering on an image is not a common procedure because it generally removes too much image detail. However, bandpass filtering is quite useful in isolating the effects on an image caused by selected frequency bands. This is illustrated in Fig. 5.17. This image was generated by (1) using Eq. (5.4-1) to obtain the bandpass filter corresponding to the bandreject filter used in Fig. 5.16; and (2) taking the inverse transform of the bandpass-filtered transform. Most image detail was lost, but the information that remains is most useful, as it is clear that the noise pattern recovered using this method is quite close to the noise that corrupted the image in Fig. 5.16(a). In other words, bandpass filtering helped isolate the noise pattern. This is a useful result because it simplifies analysis of the noise, reasonably independently of image content. ■

Notch Filters:



a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



Notch Filters

Reject (or pass) frequencies in a predefined neighbourhood about the centre of the frequency rectangle

Constructed as products of highpass filters whose centres have been translated to the centres of the notches

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

centre at (u_k, v_k)

centre at $(-u_k, -v_k)$

=> Distances computations:

$$D_k(u,v) = \left[(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2\right]^{1/2}$$

$$D_{-k}(u,v) = \left[(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2\right]^{1/2}$$

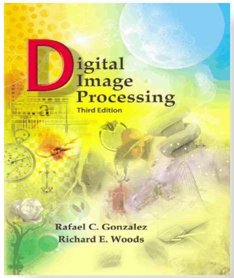
Notch Filters

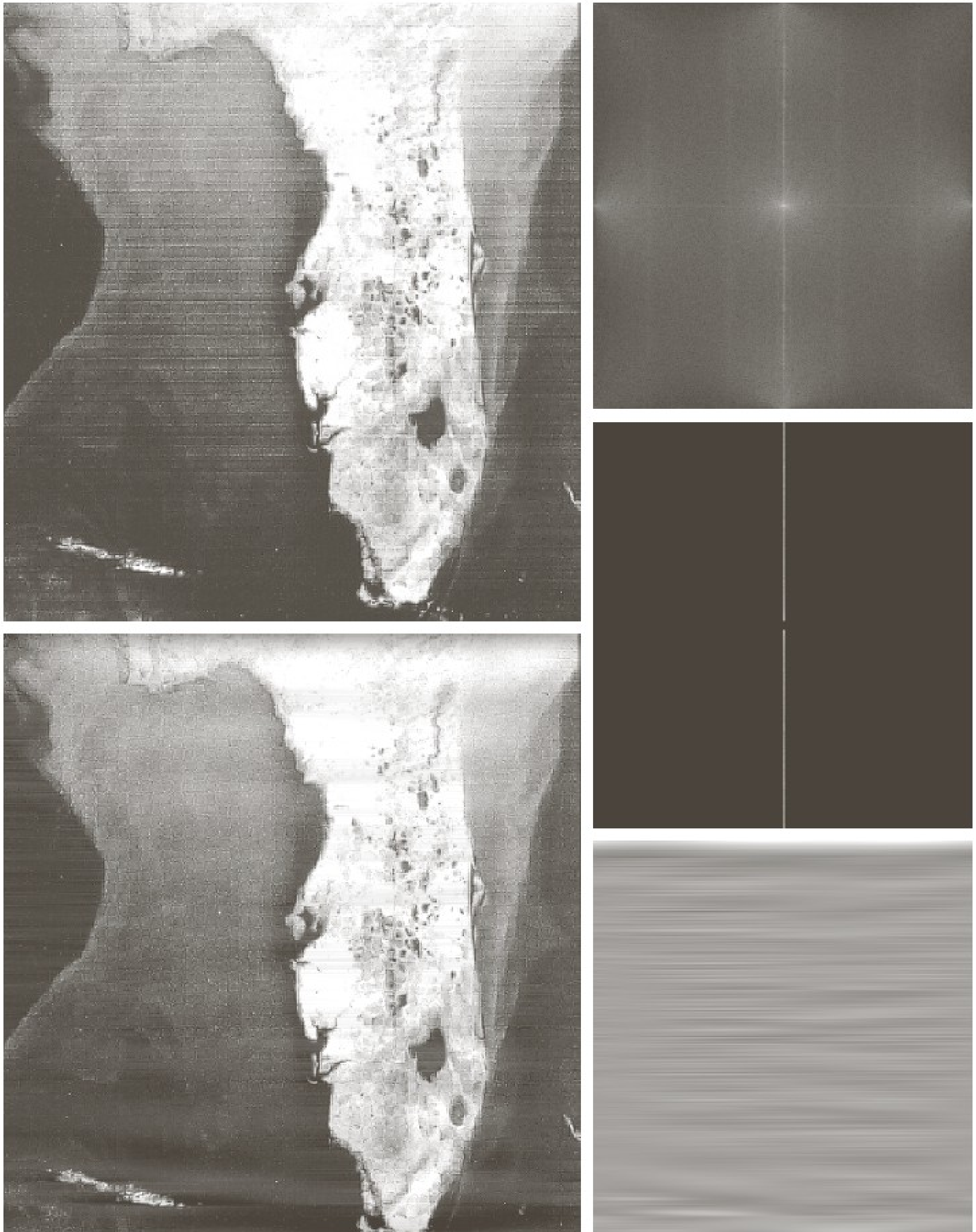
Example: Butterworth notch reject filter of order n, containing 3 notch pairs:

$$H_{NR}(u,v) = \prod_{k=1}^{3} \left[\frac{1}{1 + [D_{0k}/D_k(u,v)]^{2n}} \right] \left[\frac{1}{1 + [D_{0k}/D_{-k}(u,v)]^{2n}} \right]$$

A *Notch Pass filter* (NP) is obtained from a *Notch Reject filter* (NR) using:

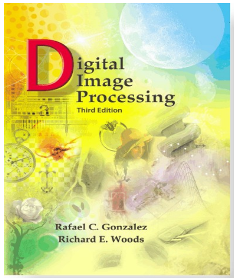
$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$





a b
c
e d

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

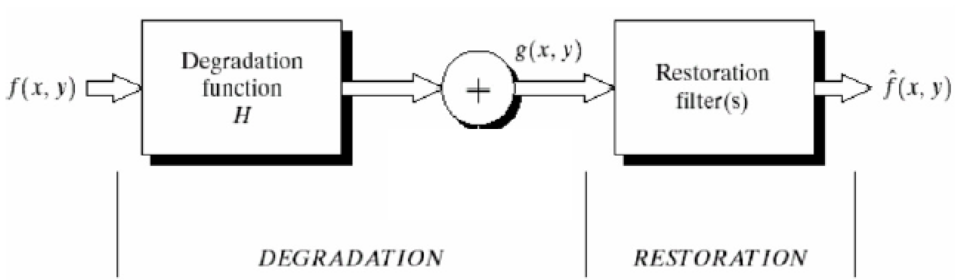


Linear Position-Invariant Degradations

- Degradation function
  - Linear (eq 5.5-3, 5.5-4)
    - Homogeneity $H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$
    - Additivity $H[af_1(x, y)] = aH[f_1(x, y)]$
  - Position-invariant (in cartesian coordinates, eq 5.5-5)

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta)$$

→ linear filtering with H(u,v)
convolution with h(x,y) – point spread function



Divide-and-conquer step #2: linear degradation, noise negligible.

$$g(x, y) = H[f(x, y)] = H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)\, d\alpha\, d\beta\right] \quad (5.5\text{-}7)$$

If H is a linear operator and we extend the additivity property to integrals, then

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H[f(\alpha, \beta)\delta(x - \alpha, y - \beta)]\, d\alpha\, d\beta \quad (5.5\text{-}8)$$

Because $f(\alpha, \beta)$ is independent of x and y, and using the homogeneity property, it follows that

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)H[\delta(x - \alpha, y - \beta)]\, d\alpha\, d\beta \quad (5.5\text{-}9)$$

The term

$$h(x, \alpha, y, \beta) = H[\delta(x - \alpha, y - \beta)] \quad (5.5\text{-}10)$$

is called the *impulse response* of H. In other words, if $\eta(x, y) = 0$ in Eq. (5.5-1), then $h(x, \alpha, y, \beta)$ is the response of H to an impulse at coordinates (x, y). In

Substituting Eq. (5.5-10) into Eq. (5.5-9) yields the expression

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)\, d\alpha\, d\beta \qquad (5.5\text{-}11)$$

which is called the *superposition* (or *Fredholm*) *integral of the first kind*. This expression is a fundamental result that is at the core of linear system theory. It states that if the response of H to an impulse is known, the response to *any* input $f(\alpha, \beta)$ can be calculated by means of Eq. (5.5-11). In other words, a linear system H is completely characterized by its impulse response.

If H is position invariant, then, from Eq. (5.5-5),

$$H[\delta(x - \alpha, y - \beta)] = h(x - \alpha, y - \beta) \qquad (5.5\text{-}12)$$

Equation (5.5-11) reduces in this case to

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)\, d\alpha\, d\beta \qquad (5.5\text{-}13)$$

This expression is the *convolution integral* introduced for one variable in

In the presence of additive noise, the expression of the linear degradation model [Eq. (5.5-11)] becomes

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x, \alpha, y, \beta)\, d\alpha\, d\beta + \eta(x, y) \qquad (5.5\text{-}14)$$

If H is position invariant, Eq. (5.5-14) becomes

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)\, d\alpha\, d\beta + \eta(x, y) \qquad (5.5\text{-}15)$$

The values of the noise term $\eta(x, y)$ are random, and are assumed to be independent of position. Using the familiar notation for convolution, we can write Eq. (5.5-15) as

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \qquad (5.5\text{-}16)$$

or, based on the convolution theorem (see Section 4.6.6), we can express it in the frequency domain as

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \qquad (5.5\text{-}17)$$

5 Estimating the Degradation Function

3 main ways to estimate the degradation function for use in an image restoration:

1. Observation
2. Experimentation
3. Mathematical modeling

5.1 Estimation by Image Observation

The degradation is assumed to be *linear* and *position-invariant*

• Look at a small rectangular section of the image containing sample structures, and in which the signal content is strong (e.g. high contrast): subimage $g_s(x,y)$

• Process this subimage to arrive at a result as good as possible: $\hat{f}_s(x,y)$

Assuming the effect of noise is negligible in this area: $H_s(u,v) = \dfrac{G_s(u,v)}{\hat{F}_s(u,v)}$

=> deduce the complete degradation function $H(u,v)$ (position invariance)

5.2 Estimation by Experimentation

If an equipment similar to the one used to acquire the degraded image is available:

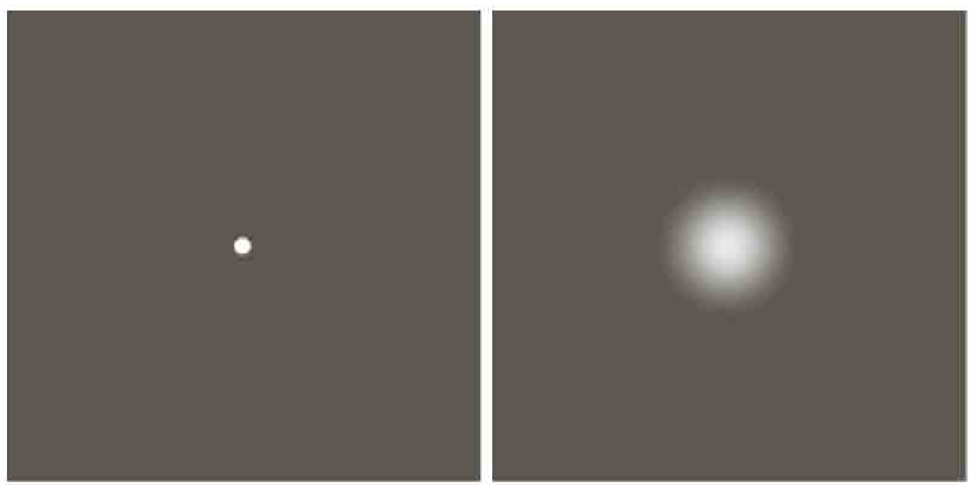
- Find system settings reproducing the most similar degradation as possible
- Obtain an impulse response of the degradation by imaging an impulse (dot of light)

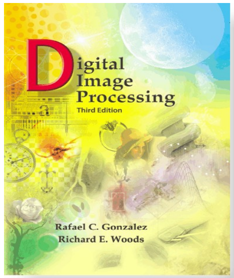
FT of an impulse = constant => $$H(u,v) = \frac{G(u,v)}{A}$$

A = constant describing the strength of the impulse

a b

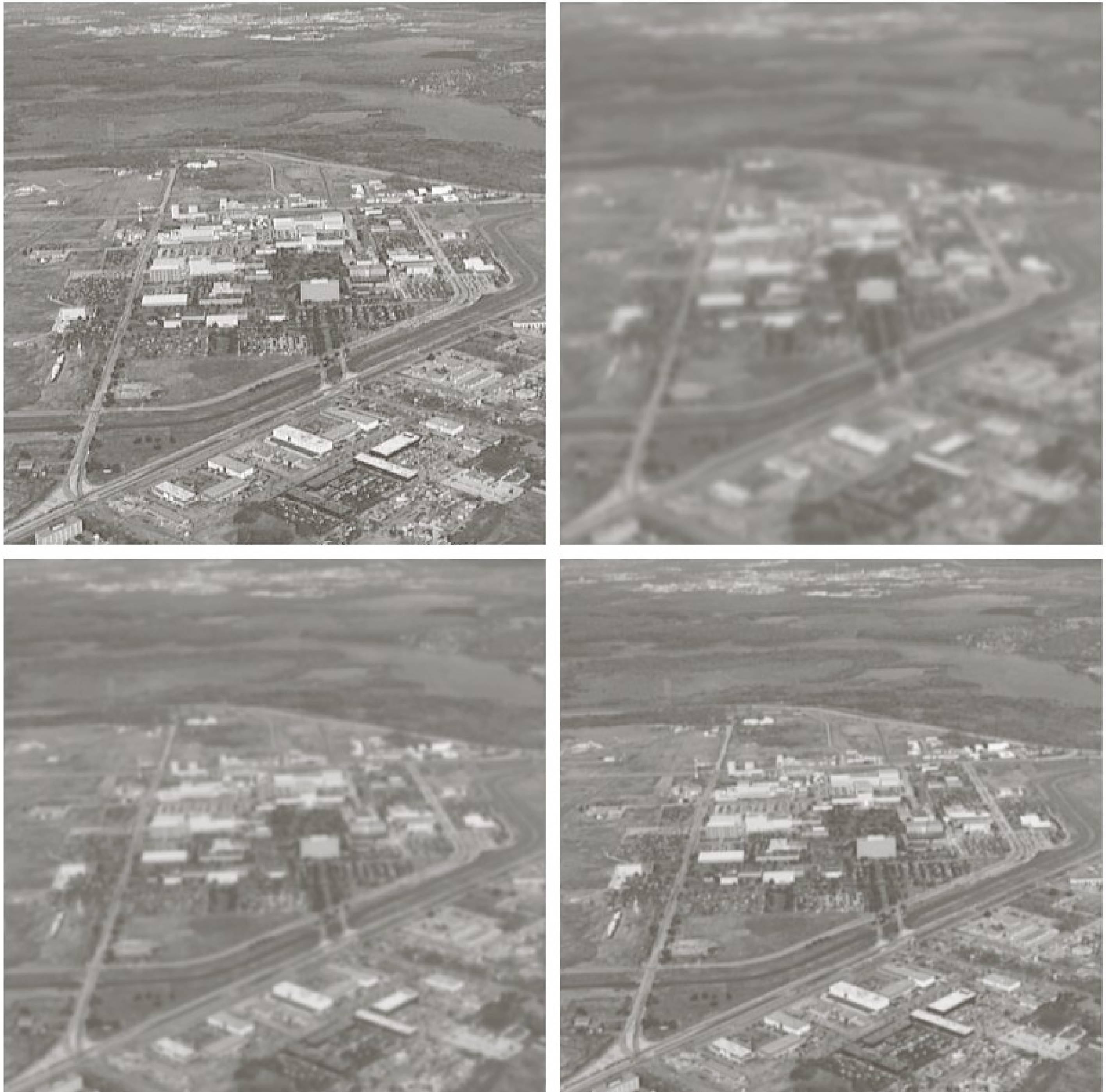
FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.





5.3 Estimation by Modeling

Example 1: degradation model proposed by Hufnagel and Stanley [1964] based on the physical characteristics of atmospheric turbulence: $H(u,v) = e^{-k(u^2+v^2)^{5/6}}$



a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Example 2: derive a mathematical model starting form basic principles

Illustration: image blurring by uniform linear motion between the image and the sensor during image acquisition

If T is the duration of exposure the blurred image can be expressed as:

$$g(x,y) = \int_0^T f\left[x - x_0(t), y - y_0(t)\right] dt$$

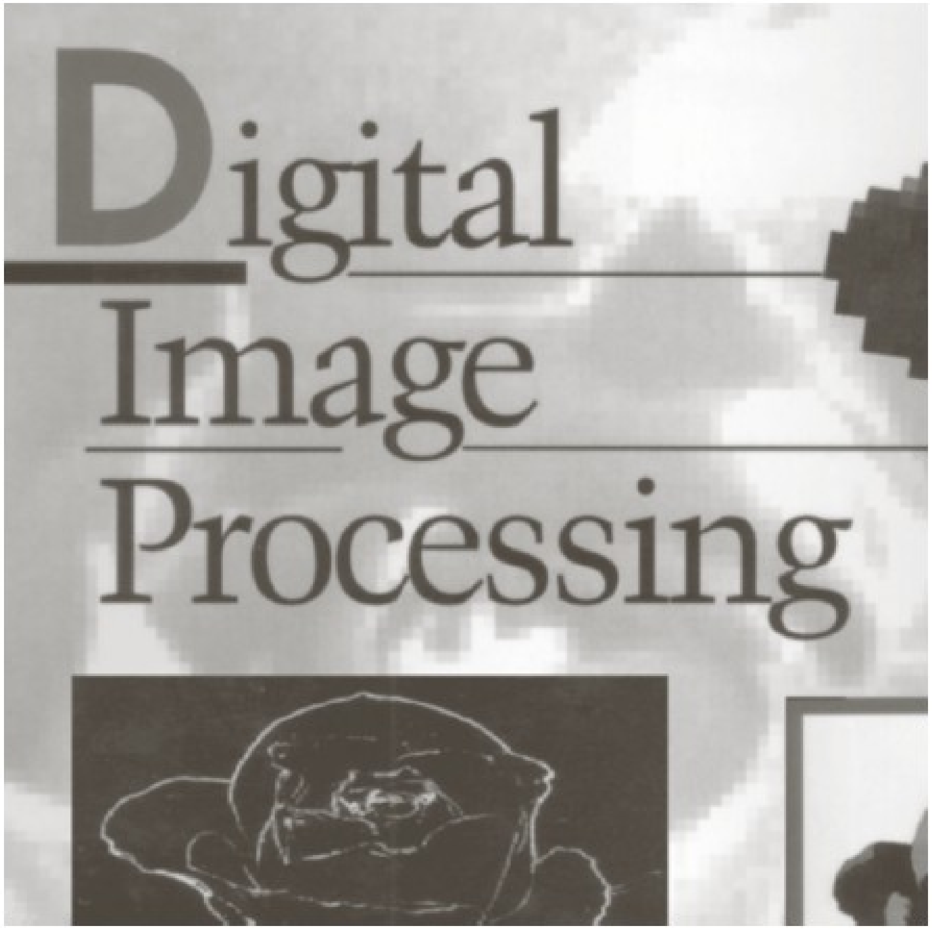
FT[g(x,y)] => $G(u,v) = F(u,v) \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt$

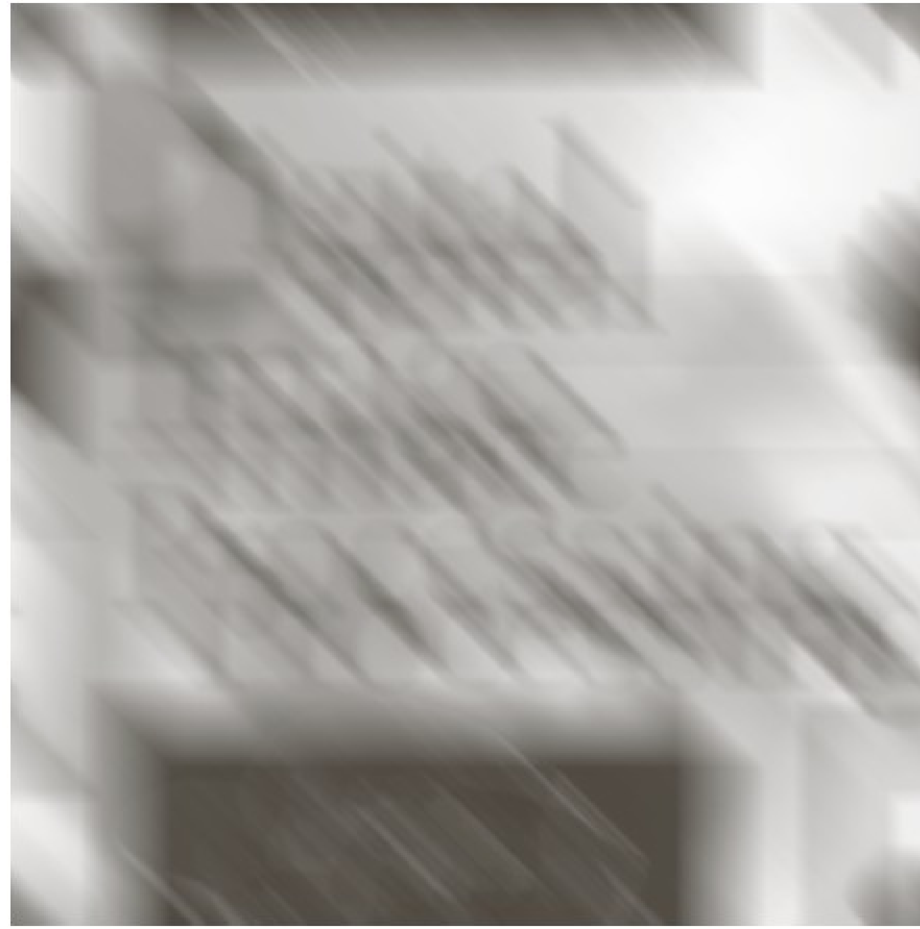
$$H(u,v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \qquad \Rightarrow \qquad G(u,v) = H(u,v)F(u,v)$$

E.g. if uniform linear motion in the *x*-direction only, at a rate $x_0(t) = at/T$

$$H(u,v) = \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua}$$

NB: $H = 0$ for $u = n/a$





a b

FIGURE 5.26
(a) Original image.
(b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

If motion in y as well:

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin\left[\pi(ua+vb)\right] e^{-j\pi(ua+vb)}$$

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

6 Inverse Filtering

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \qquad \text{(array operation)}$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \qquad \Rightarrow \hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

$\Rightarrow$ Even if we know *H(u,v)*, we cannot recover the "undegraded" image exactly because *N(u,v)* is not known

$\Rightarrow$ If *H* has zero or very small values, the ration *N/H* could dominate the estimate

One approach to get around this is to limit the filter frequencies to values near the origin

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.