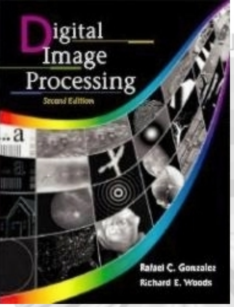
Chapter 5
Image Restoration

- Degraded digital image restoration
  - Spatial domain processing

    Additive noise
  - Frequency domain

    Blurred image



5. 1 Model of Image degradation and restoration

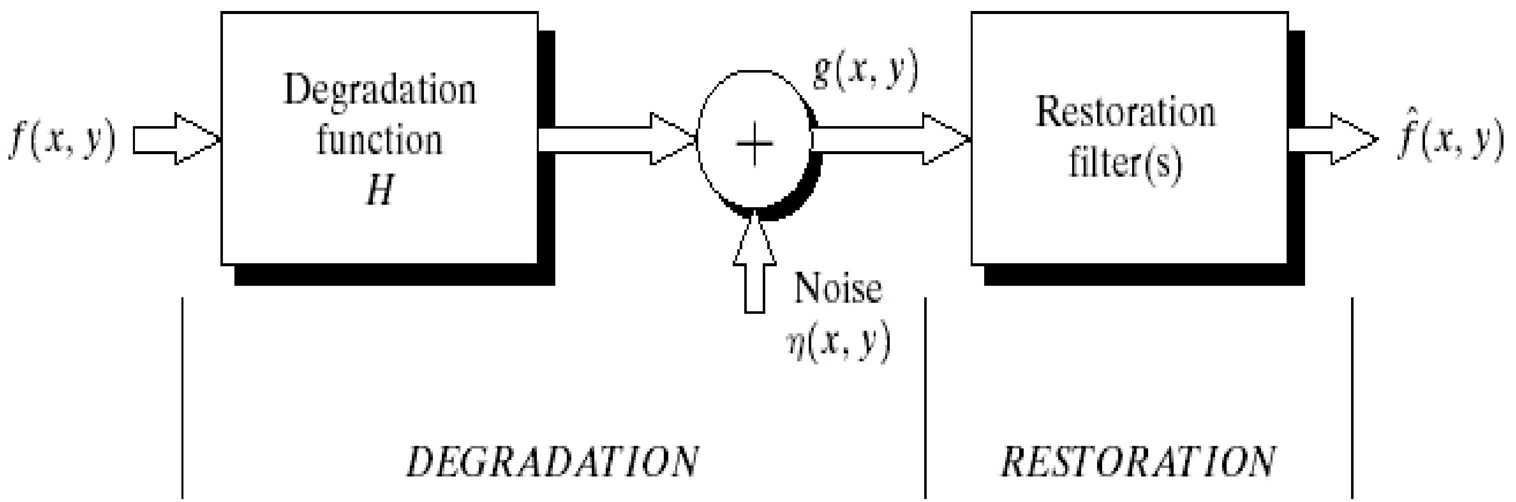
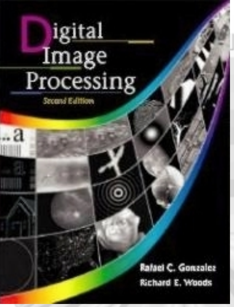


FIGURE 5.1 A model of the image degradation/restoration process.

- $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$
- $G(u, v) = H(u, v)F(u, v) + N(u, v)$



5.2 Noise Model

- Spatial and frequency property of noise
  - White noise (random noise)
  - Gaussian noise

$$p(z)=\frac{1}{\sqrt{2\pi\sigma}}e^{-(z-\mu)^2/2\sigma^2}$$

  - μ: mean ; σ :variance
  - *70%* $[(\mu\text{-}\sigma), (\mu+\sigma)]$
  - *95 %* $[(\mu\text{-}2\sigma), (\mu+2\sigma)]$

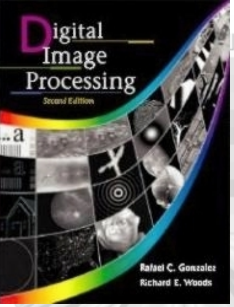
5.2 Noise Model

- *Rayleigh noise*

  The PDF is

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

- $\mu = a + (\pi b/4)^{1/2}$
- $\sigma = b(4-\pi)/4$



5.2 Noise Model

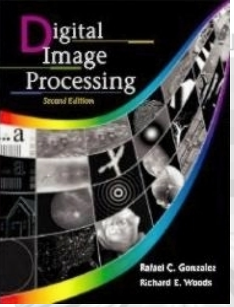
- *Erlang (gamma) noise*

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

- $\mu = b/a; \quad \sigma = b/a^2$
- *Exponential noise*

$$p(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

- $\mu = 1/a; \quad \sigma = 1/a^2$



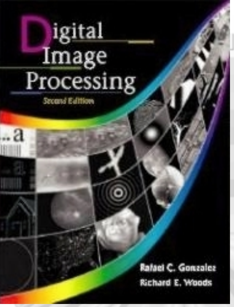
5.2 Noise Model

- *Uniform noise*

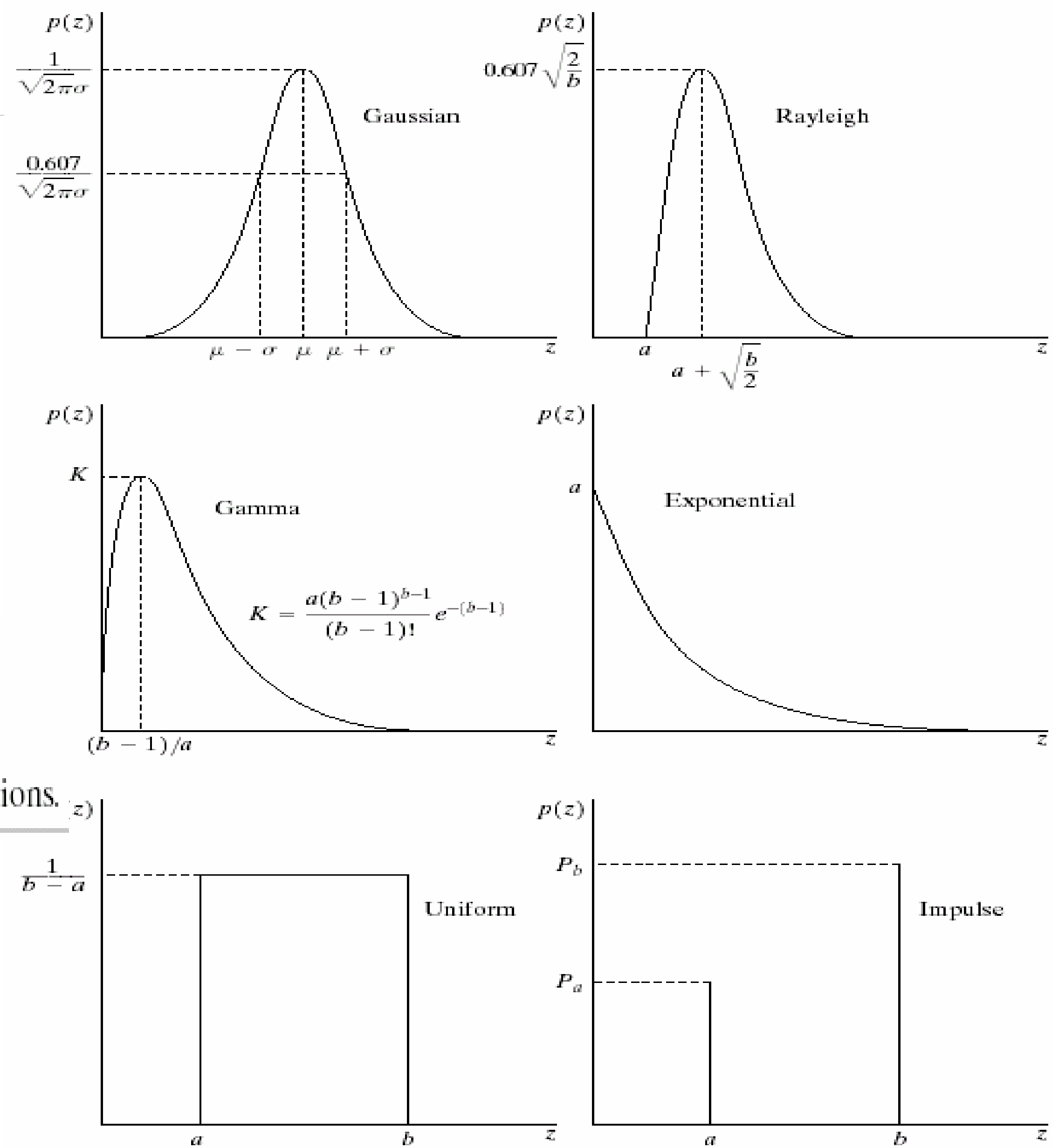
$$p(z)=\begin{cases}\dfrac{1}{b-a} & \text{for } a\le z\le b\\ 0 & \text{otherwise}\end{cases}$$

- $\mu=(a+b)/2;\quad \sigma=(b-a)^2/12$
- *Impulse noise (salt and pepper noise)*

$$p(z)=\begin{cases}P_a & \text{for } z=a\\ P_b & \text{for } z=b\\ 0 & \text{otherwise}\end{cases}$$



5.2 Noise Model



a	b
c	d
e	f

FIGURE 5.2 Some important probability density functions.

5.2 Noise Model

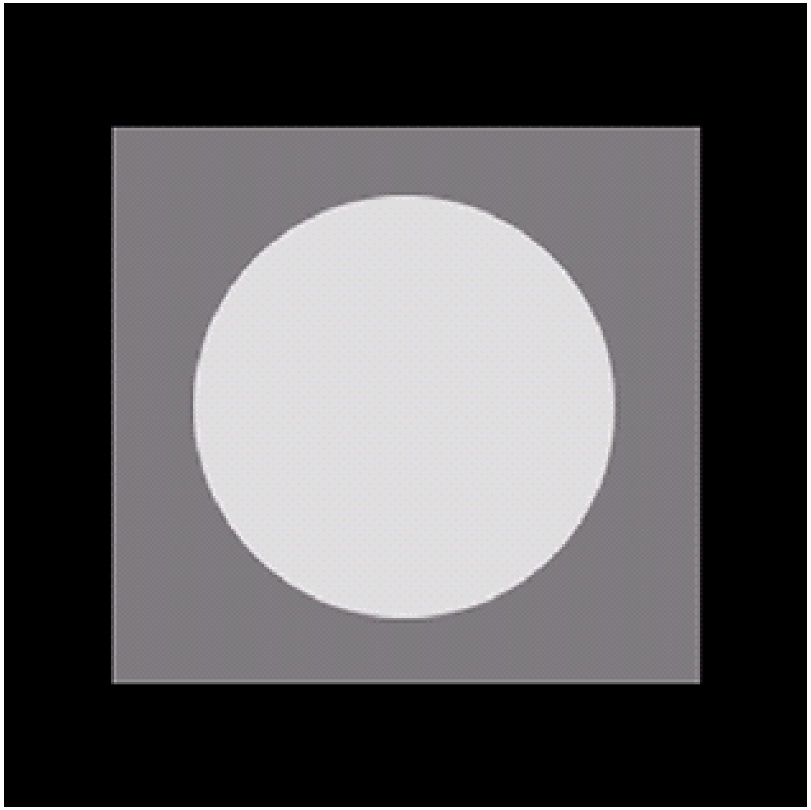


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

5.2 Noise Model

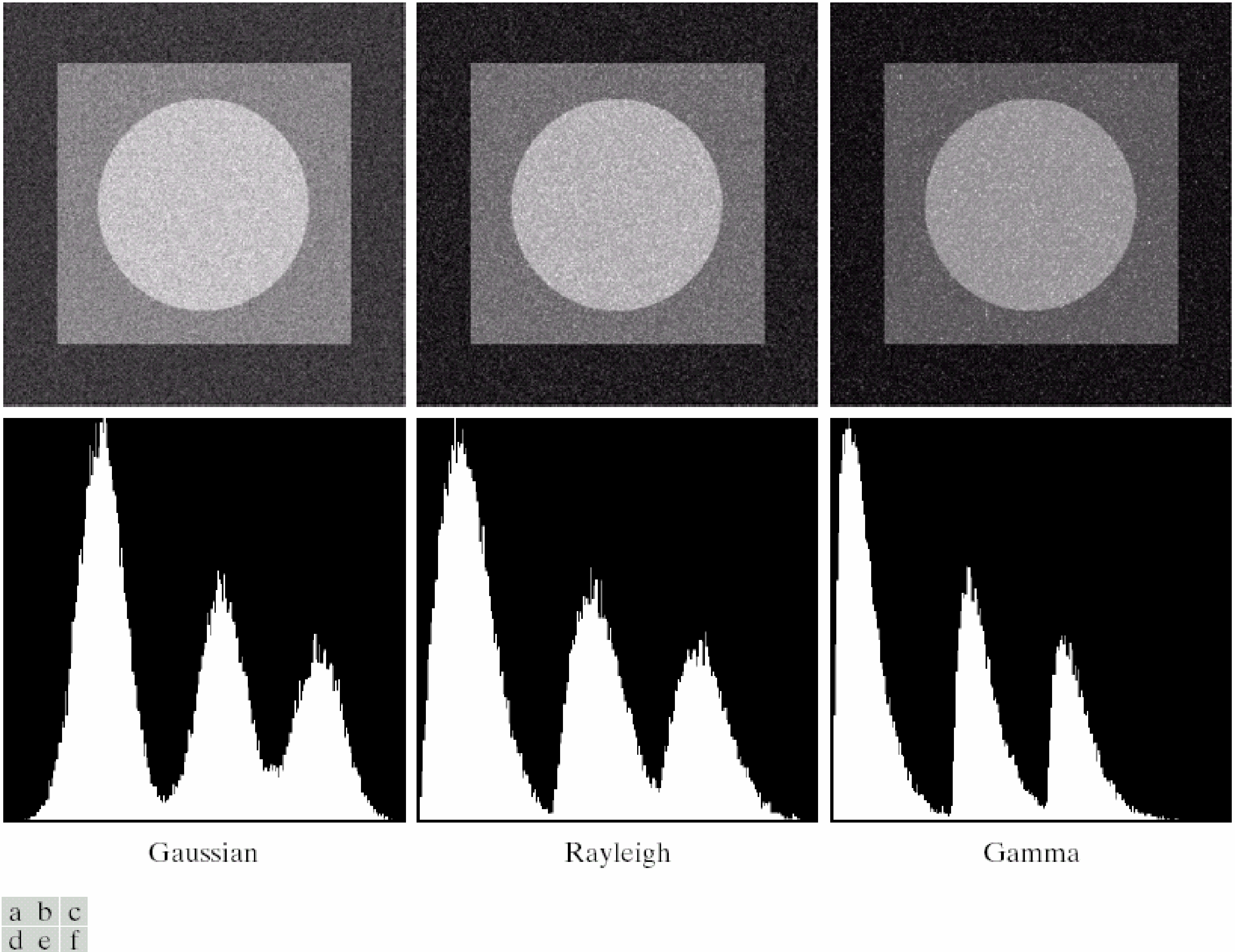
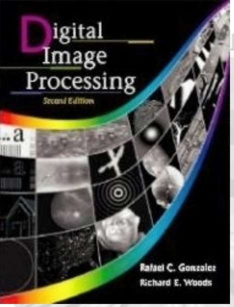


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.



5.2 Noise Model

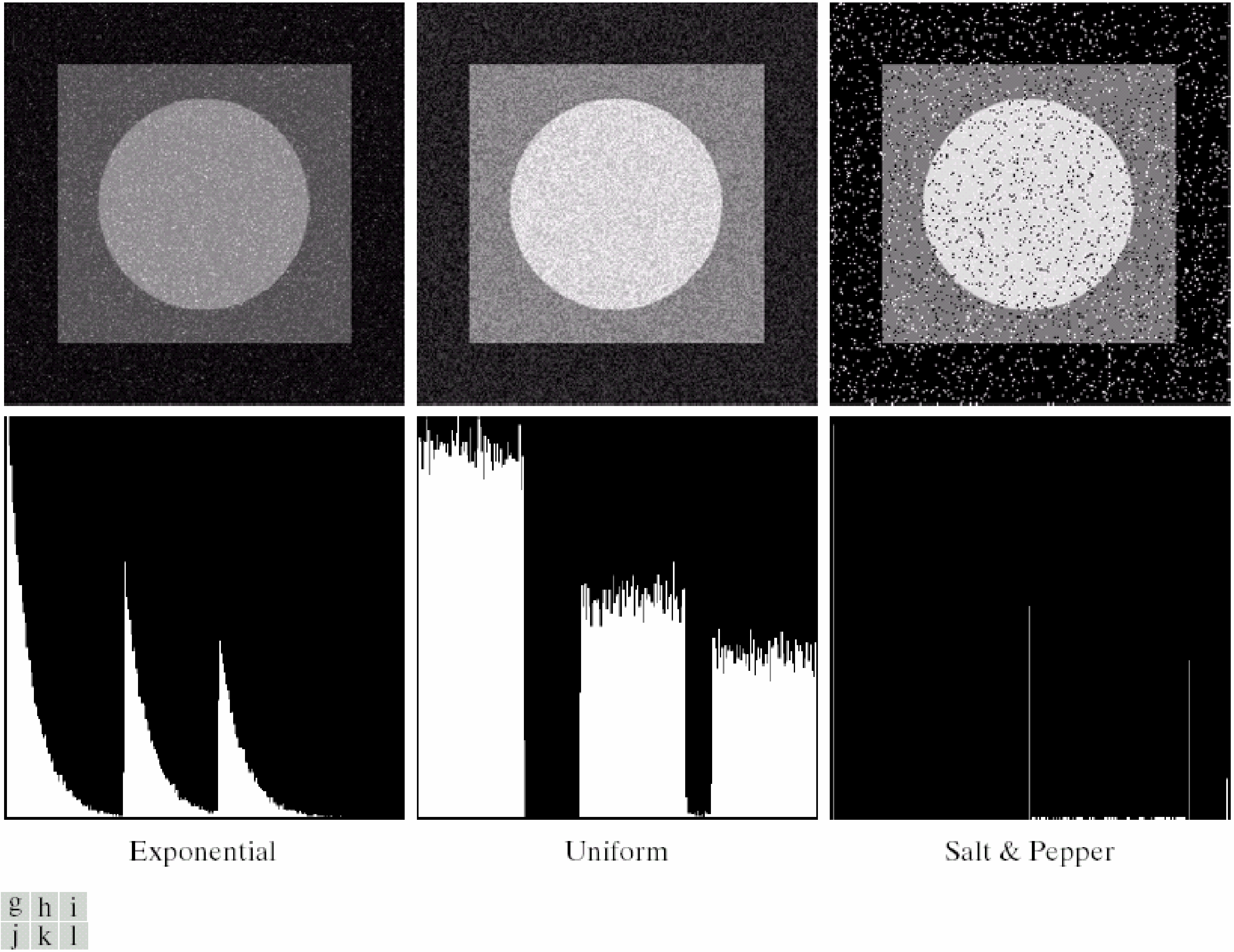
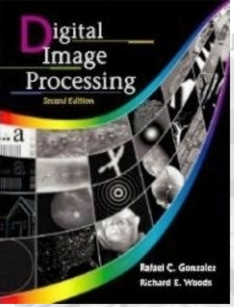


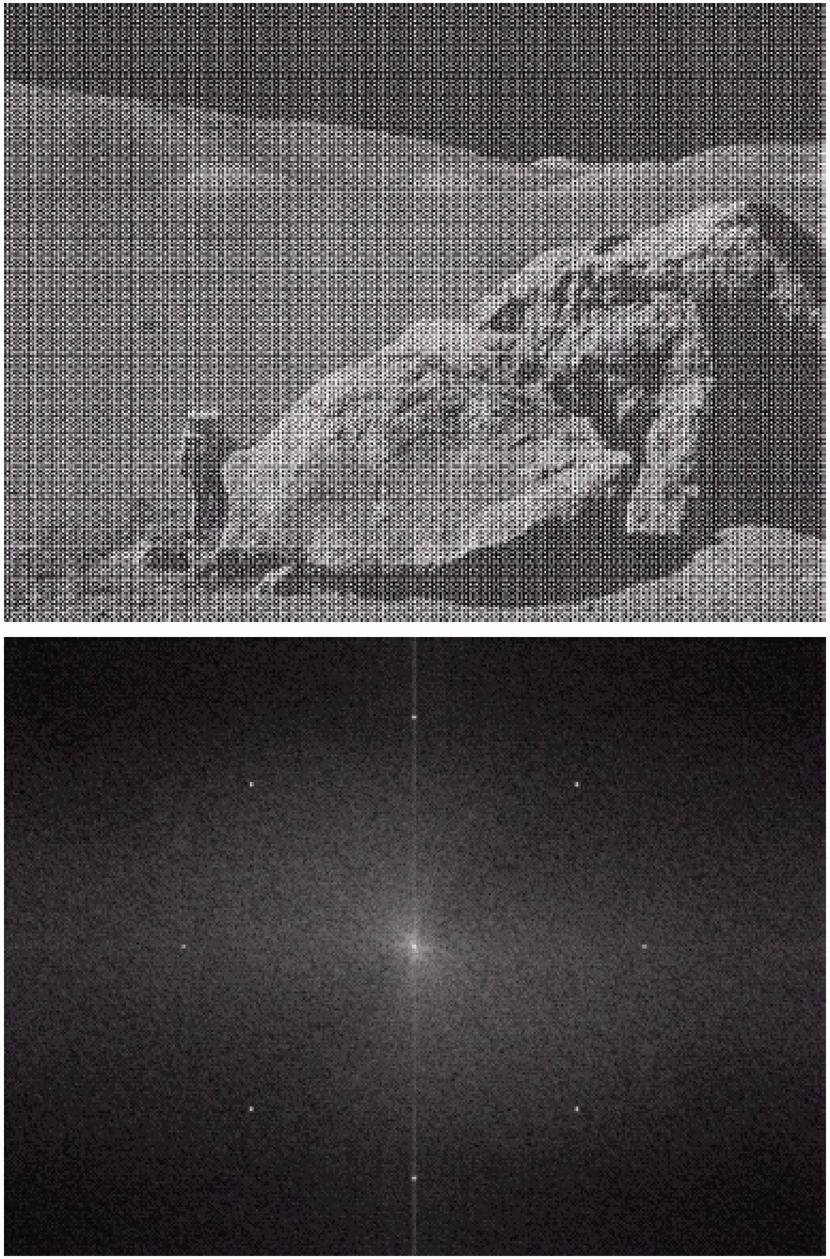
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.



5.2.3 Noise Model-periodic noise

- Periodic noise is due to the electrical or electromechanical interference during image acquisition.
- Can be estimated through the inspection of the Fourier spectrum of the image.

5.2.3 Noise Model-periodic noise



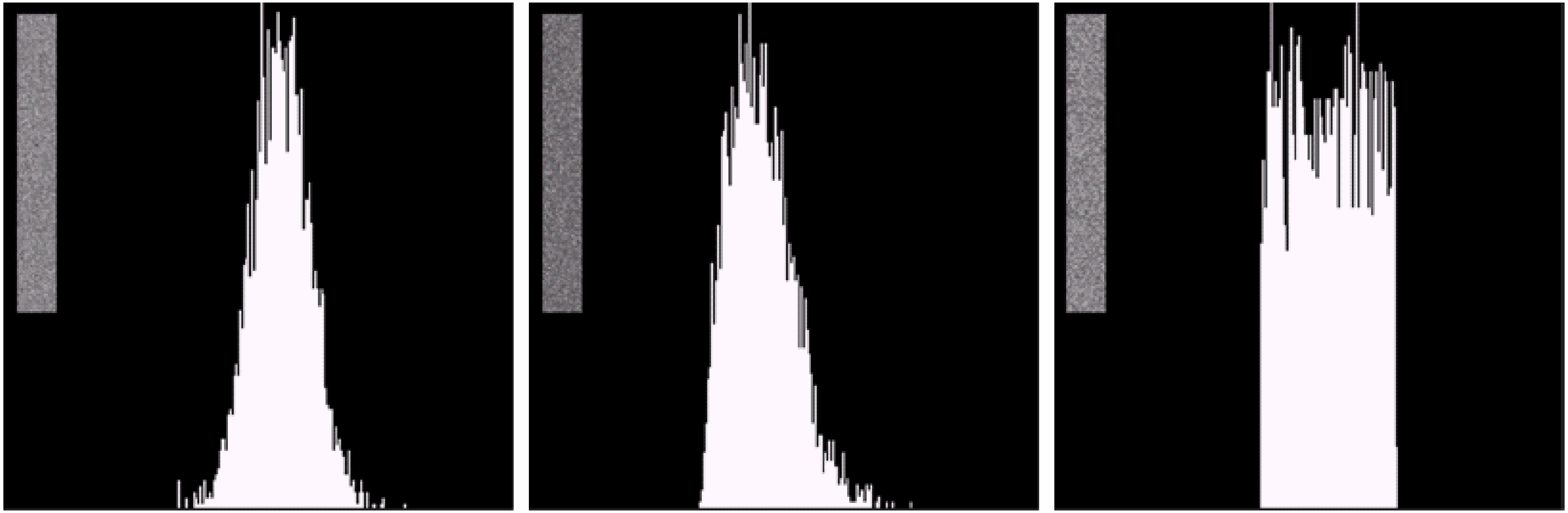
a
b

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.)

5.2.4 Noise Model- Noise parameters estimation

- Use a optical sensor to capture the image of a solid gray board that is illuminated uniformly.
- The resulting image is a good indicator of system noise.
- Find the mean μ and standard deviation σ of the histogram of the resulting image.
- From the μ and σ we can calculate the a and b, the parameter of that specific noise distribution.

5.2 Noise Model



a b c

FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

5.3 Restoration using spatial filter

- Mean filters
  - Arithmetic mean filter

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

  - Geometric mean filter

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

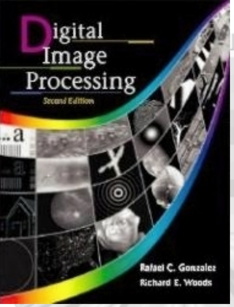
5.3 Restoration using spatial filter

– ***Harmonic mean filter*** is good for salt noise not for pepper noise.

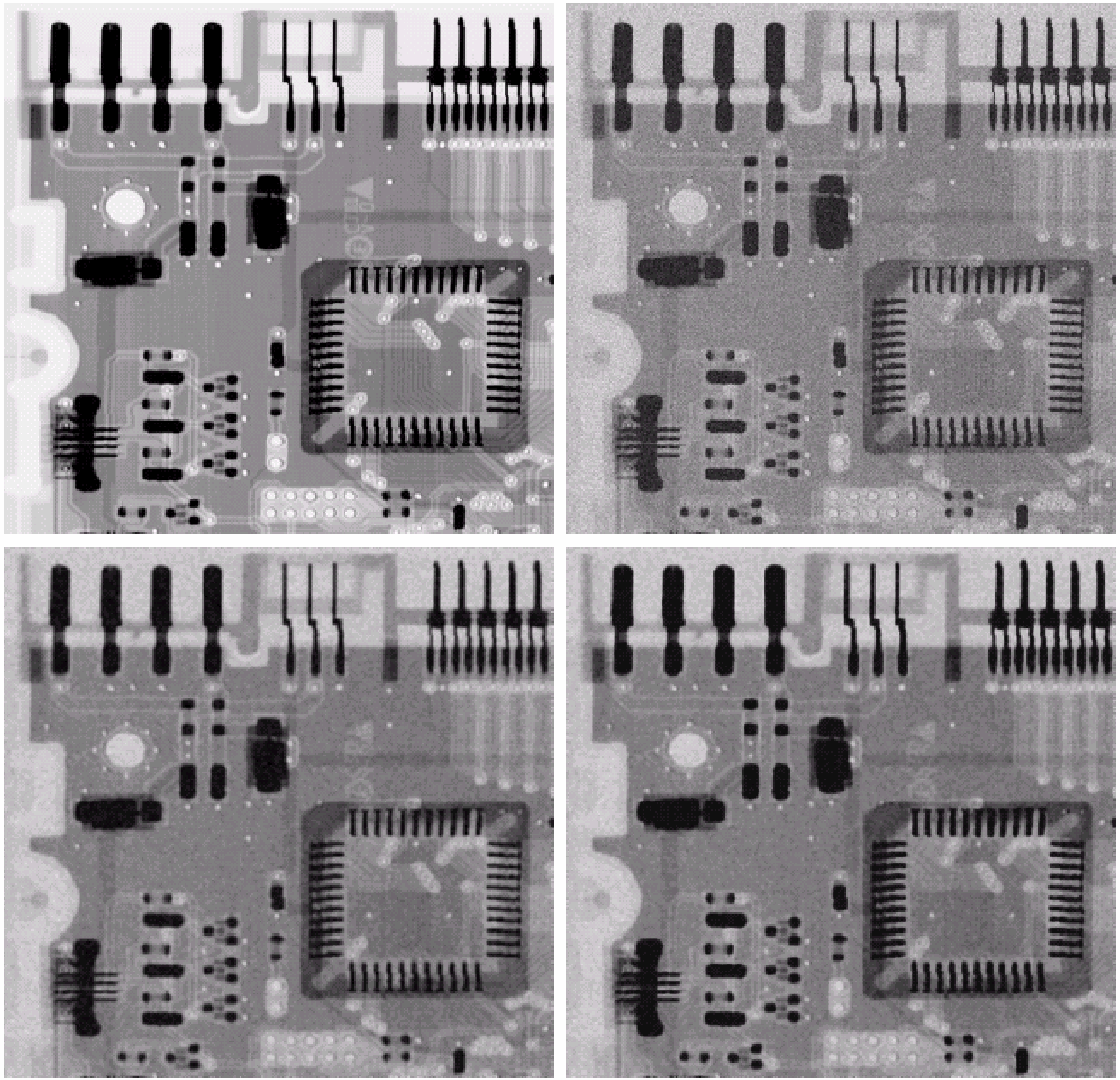
$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s, t)}}$$

– ***Contra-harmonic filter***: Q>0 reduce pepper noise Q<0 reduce salt noise.

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



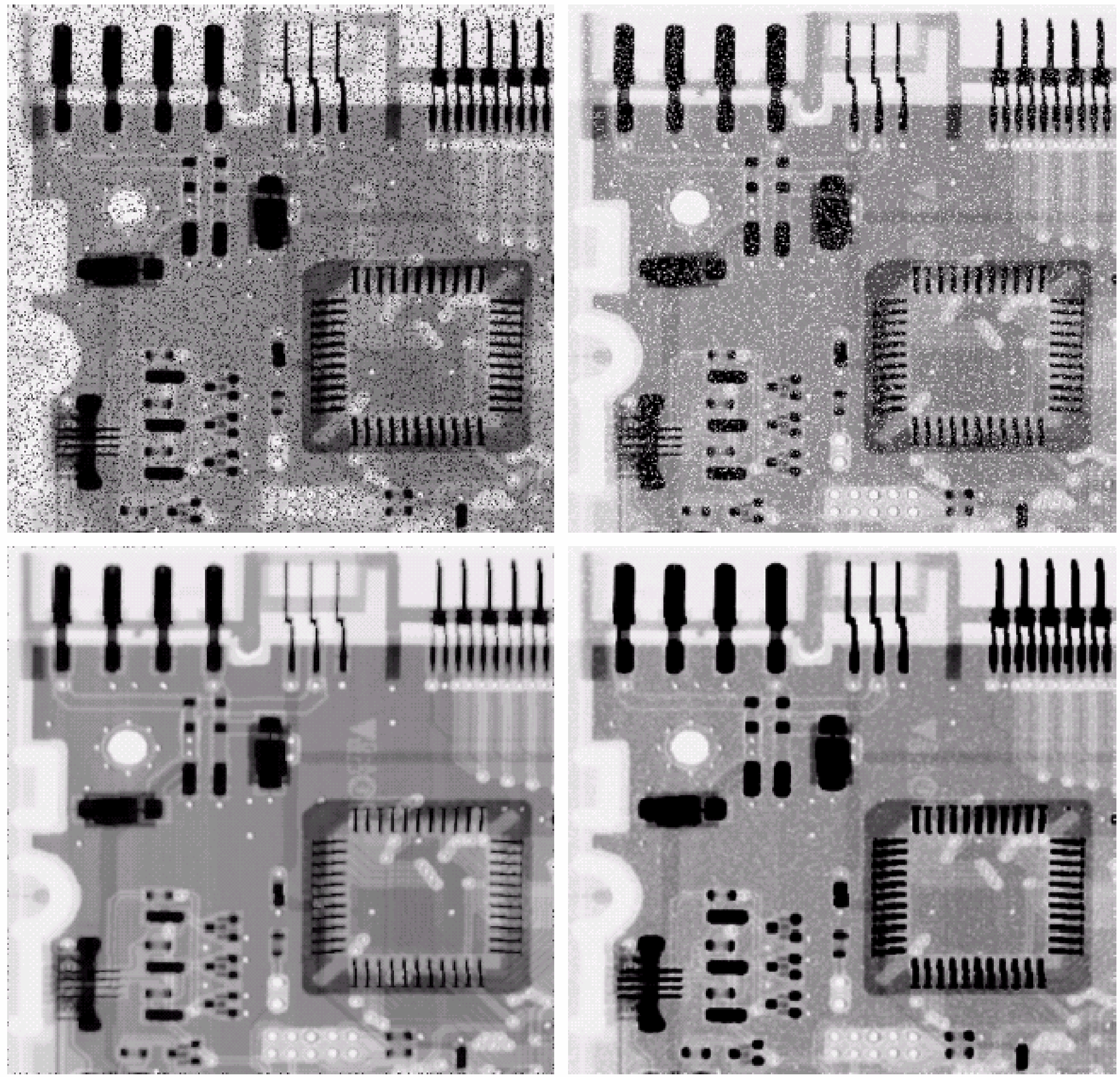
5.3 Restoration using spatial filter



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.)

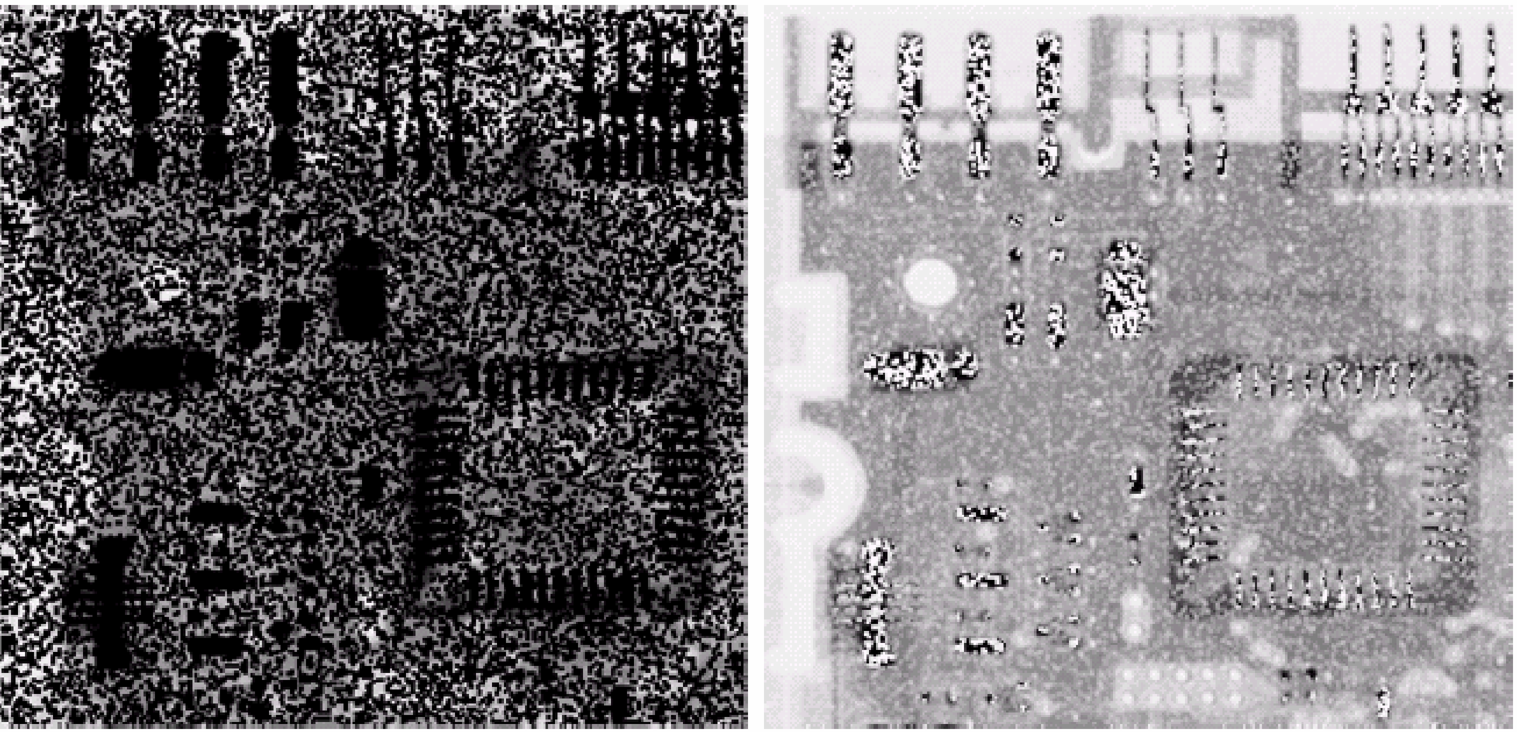
5.3 Restoration using spatial filter



a b
c d

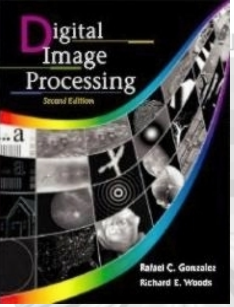
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

5.3 Restoration using spatial filter



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 $Q = 1.5$.



5.3 Restoration using spatial filter

- ***Order-Statistics filters***

Median filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Max filter: find the brightest points to reduce the pepper noise

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

Min filter: find the darkest point to reduce the salt noise

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

Midpoint filter

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

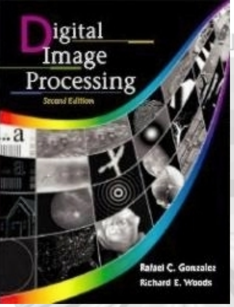
5.3 Restoration using spatial filter

- ***Alpha-trimmed mean filter***

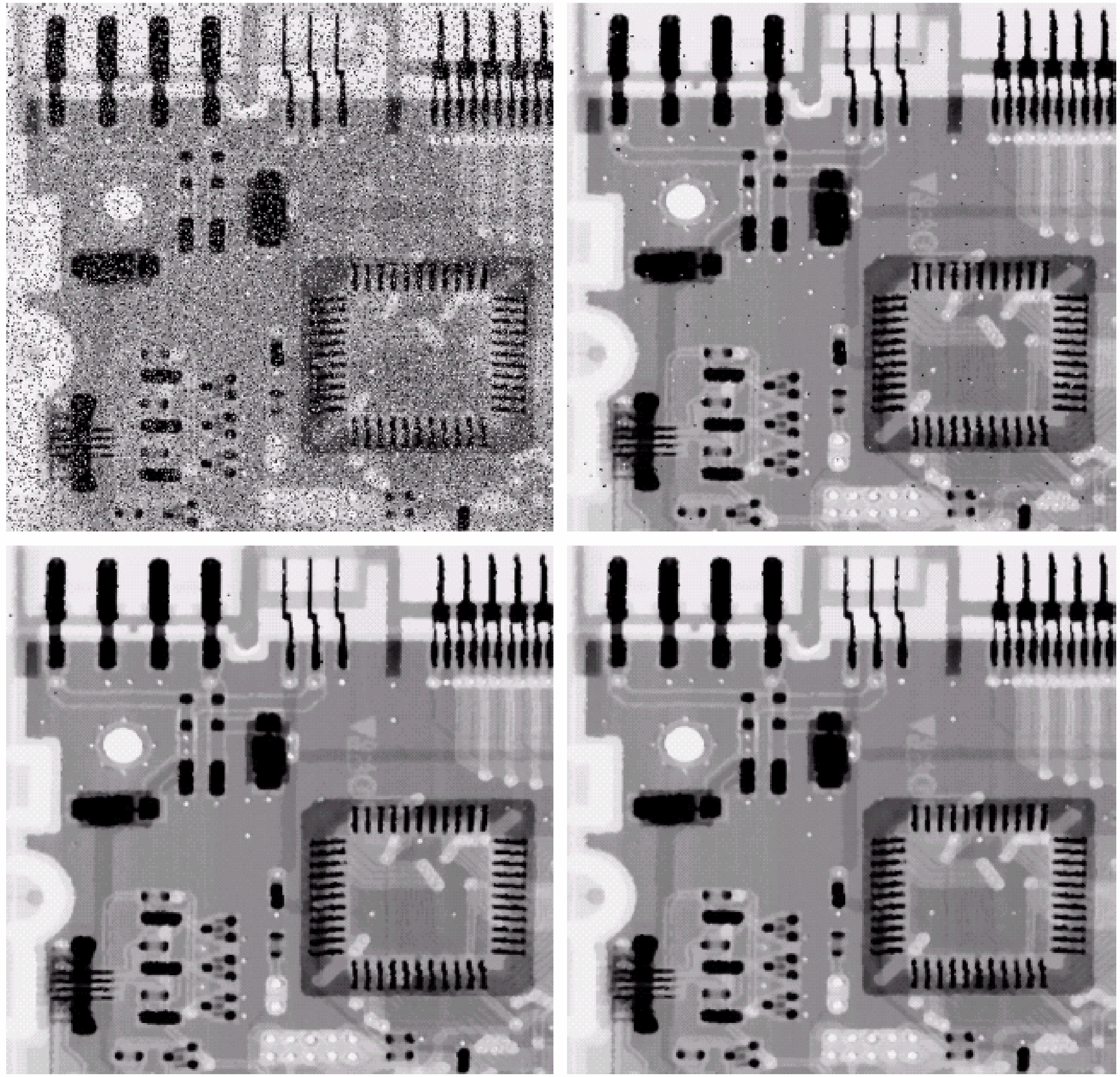
  Suppose delete d/2 lowest and d/2 highest gray-level value in the neighborhood of *(s, t)* and *average the rest.*

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

where $d=0 \sim mn\text{-}1$



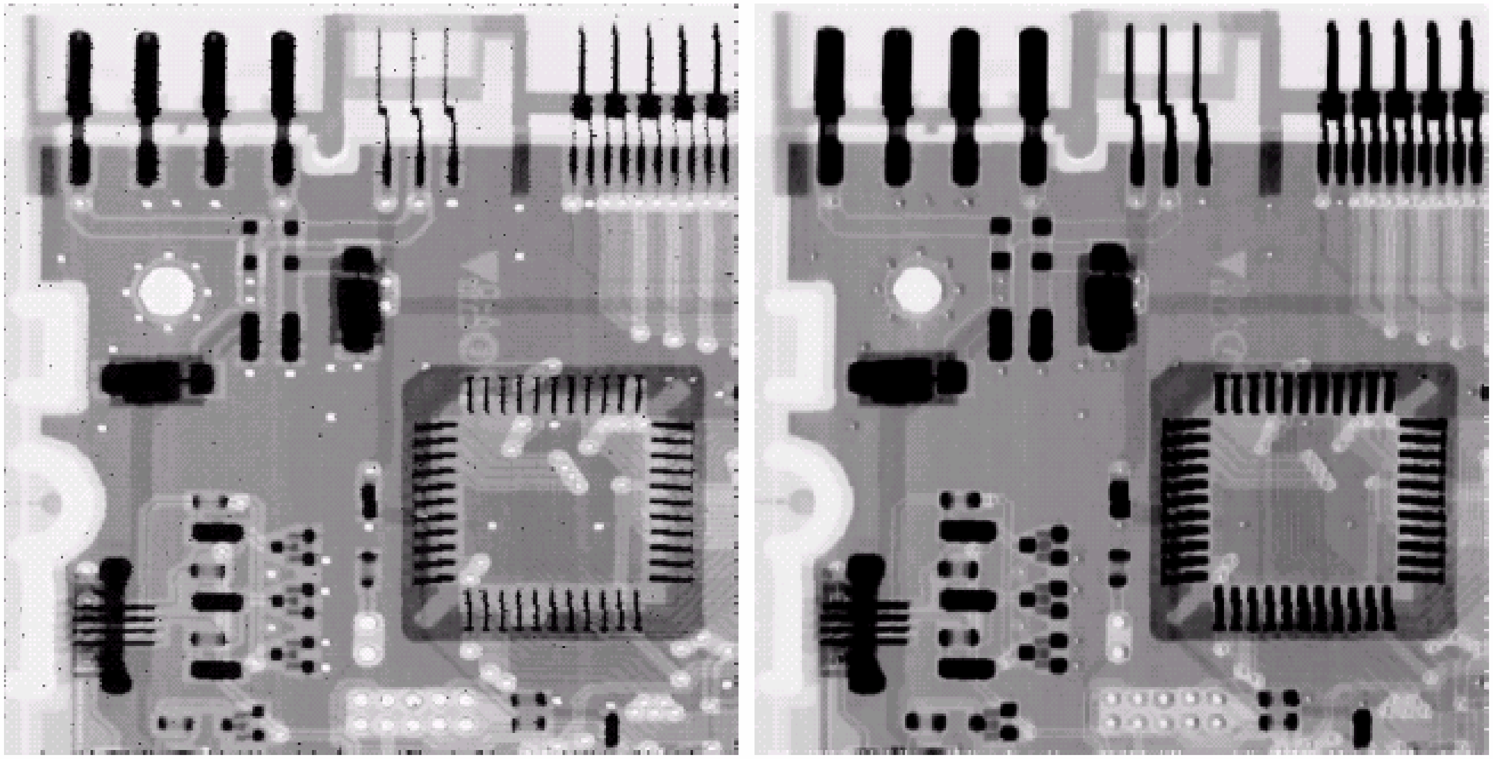
5.3 Restoration using spatial filter



a	b
c	d

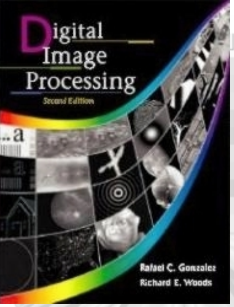
FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

5.3 Restoration using spatial filter

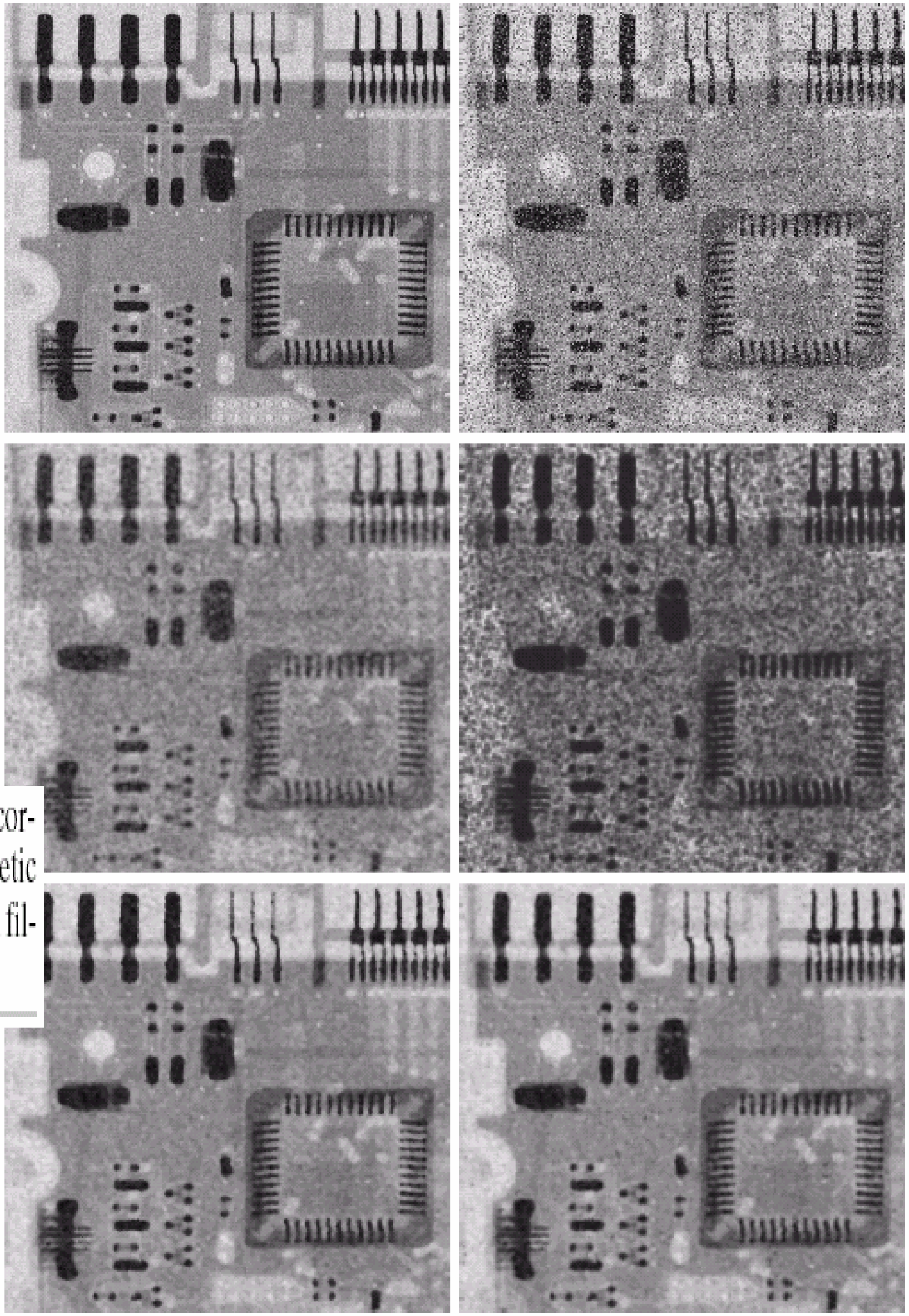


a b

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.

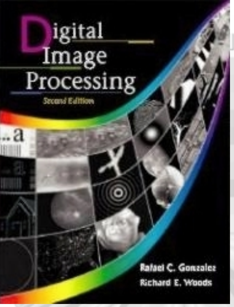


5.3 Restoration using spatial filter



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 in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.



5.3 Restoration using spatial filter *-Adaptive local noise reduction filter*

Measurements for local region S_{xy} centered at (x, y)

(1) noisy image at (x, y): $g(x, y)$ with variance $\sigma^2{}_{\eta}$

(2) The local mean m_L in S_{xy}

(3) The local variance $\sigma^2{}_L$

(4) The local variance of noise is high $\sigma^2{}_L \geq \sigma^2{}_{\eta}$

Conditions: Assume ***additive noise***

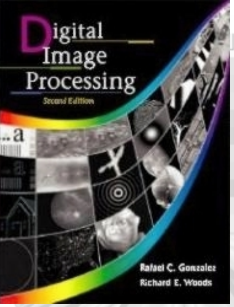
(a) Zero-noise case: If $\sigma^2{}_{\eta} = 0$ then $f(x, y) = g(x, y)$

(b) Edges: If $\sigma^2{}_L > \sigma^2{}_{\eta}$ then $f(x, y) \approx g(x, y)$

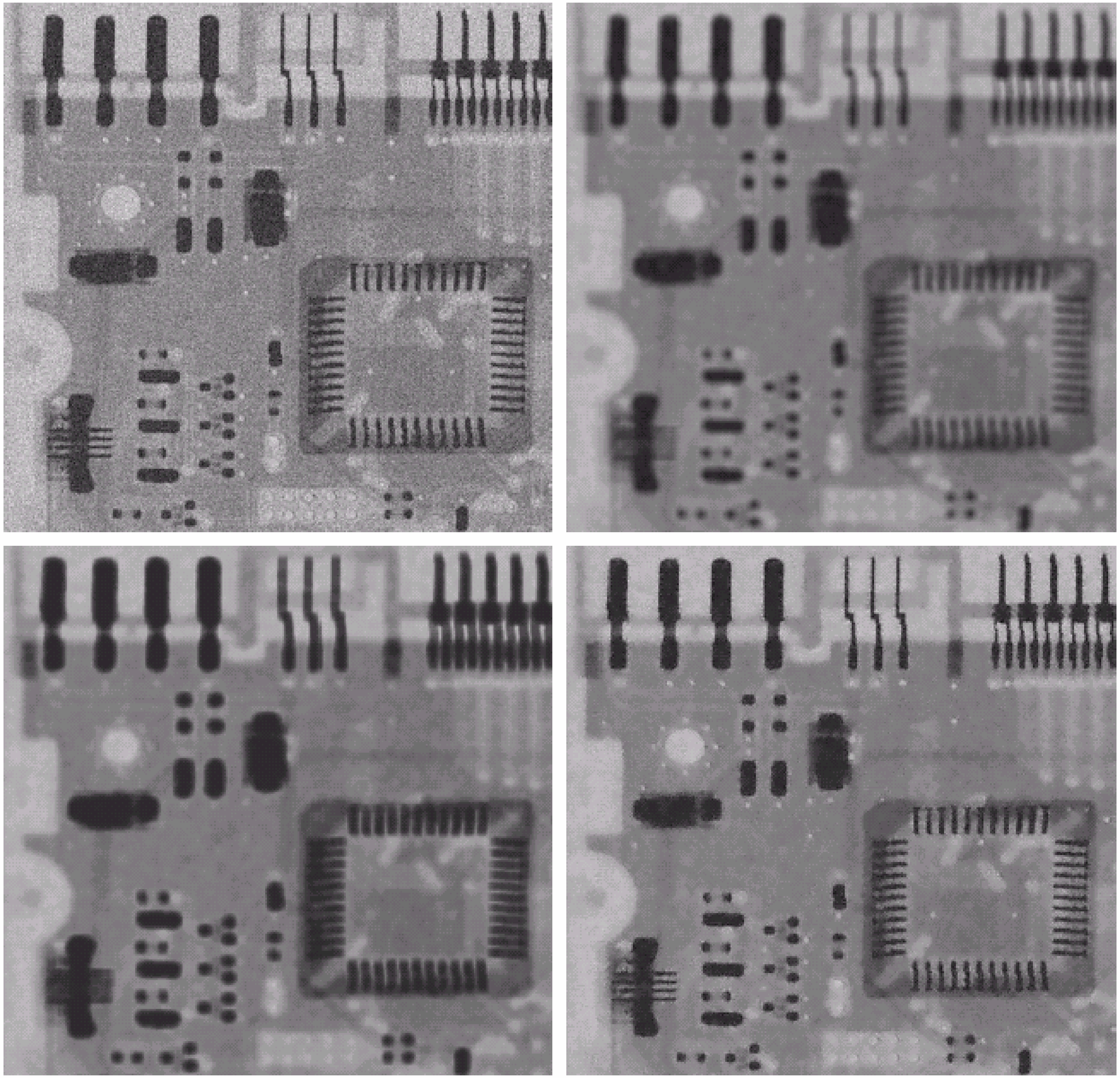
(c) If $\sigma^2{}_L = \sigma^2{}_{\eta}$ then $f(x,y) = m_L$, the arithmetic mean

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

In practice, $\sigma^2{}_{\eta}$ is unknown, so a test is built before filtering



5.3 Restoration using spatial filter



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.

5.3 Restoration using spatial filter

- ***Adaptive median filter*** can handle impulse noise with larger probability (P_a and P_b are large). Increasing(change) window size during operation
- Notations:

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

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

  z_{med}=median gray-level value in S_{xy}

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

  S_{max}=maximum allowed size of S_{xy}

5.3 Restoration using spatial filter

- *The **adaptive median filter** algorithm works in two levels: A and B*
- *Level A:* $A1 = z_{med} - z_{min}$,

  $A2 = z_{med} - z_{max}$

  If A1>0 AND A2<0 goto level B

  else increase the window size

  If window size $\leq S_{max}$ *repeat level A*

  else output z_{xy}

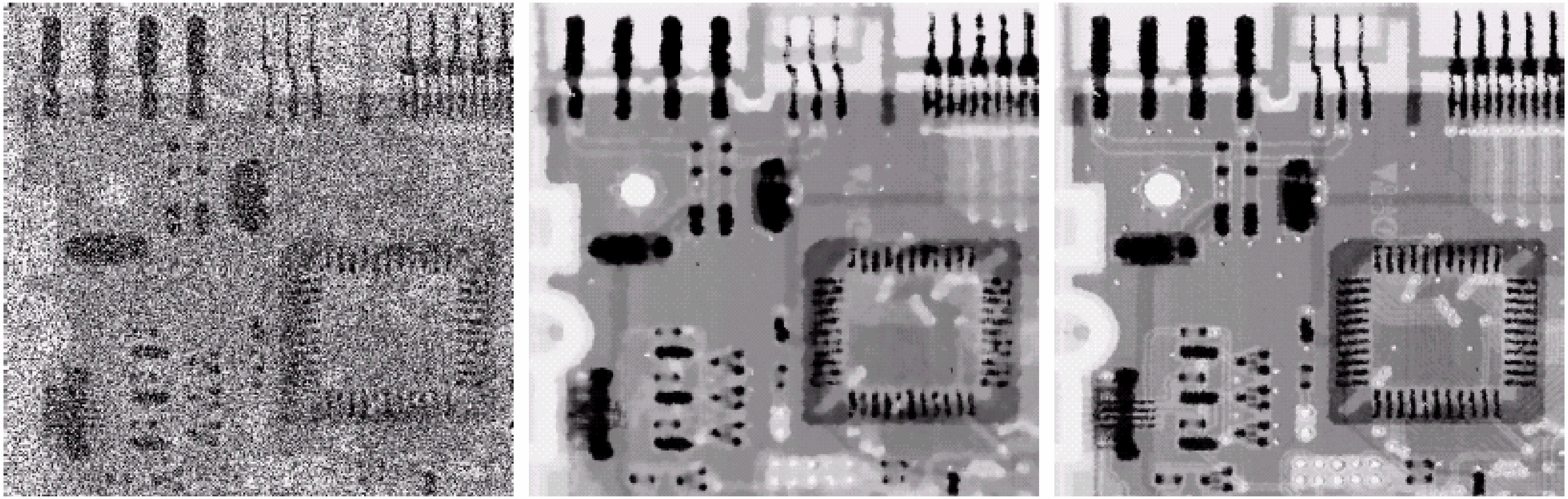
  Level B: $B1 = z_{xy} - z_{min}$,

  $B2 = z_{xy} - z_{max}$

  If B1>0 AND B2<0, output z_{xy}

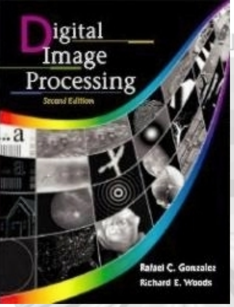
  Else output z_{med}

5 .3 Restoration using spatial filter



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 $S_{max} = 7$.



5.4 Restoration using frequency-domain filter

5.4.1 band-reject filter

- Ideal band reject filter

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

- Butterworth band reject filter

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

5.4.1 Restoration using frequency-domain filter

- Gaussian band reject filter

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

5.4 Restoration using frequency-domain filter

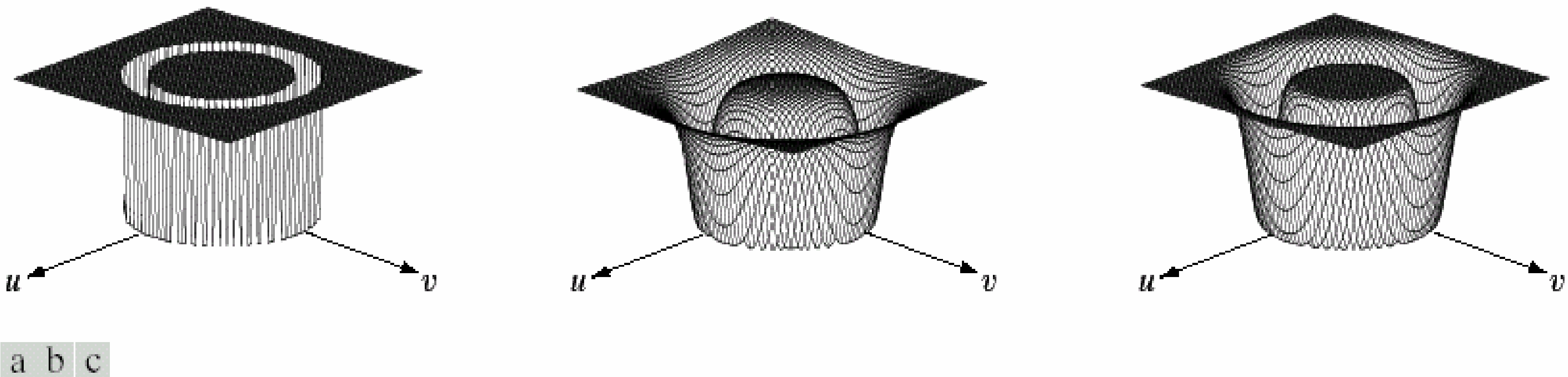
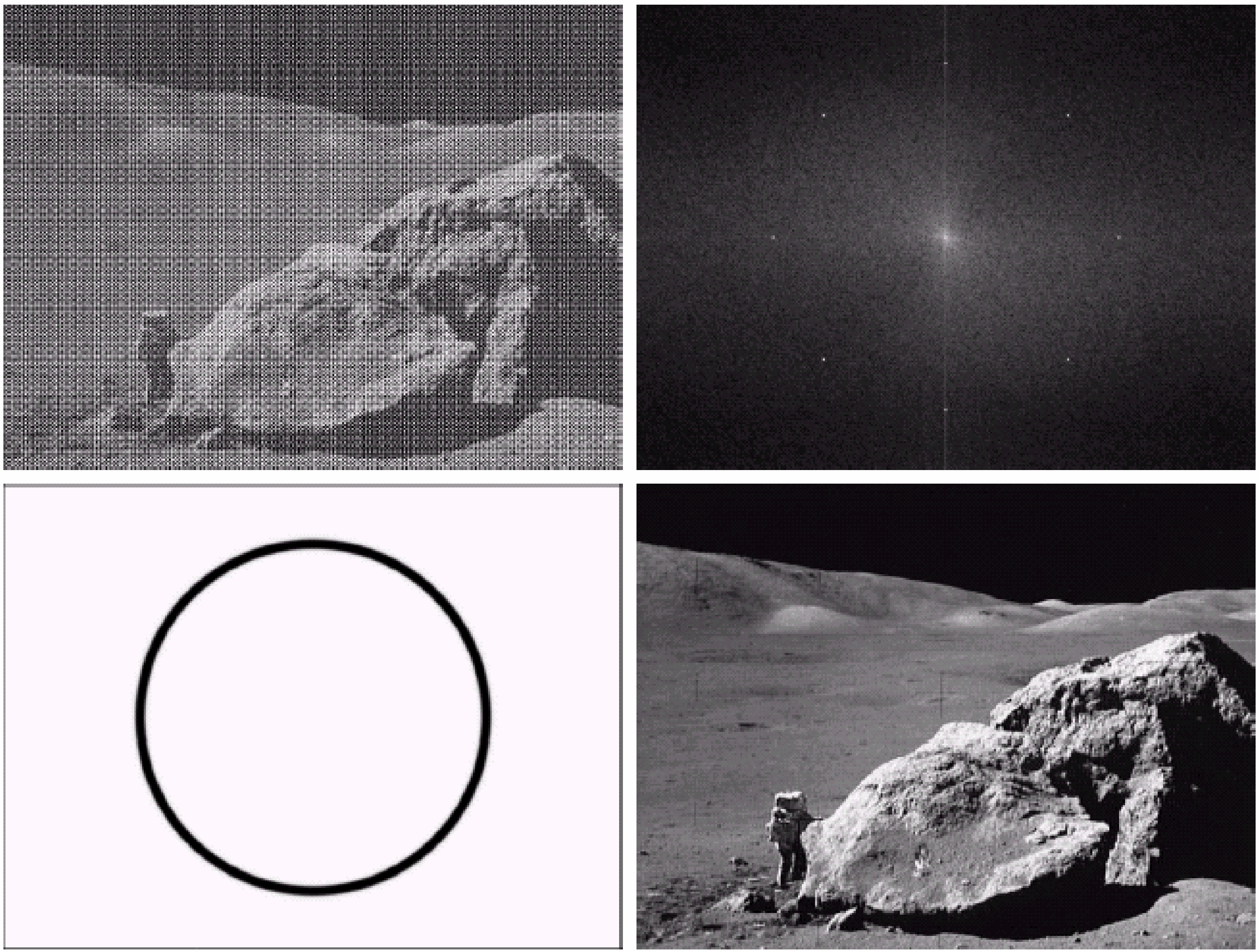


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

5.4 Restoration using frequency-domain filter



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.)

5.4.2 Bandpass filter

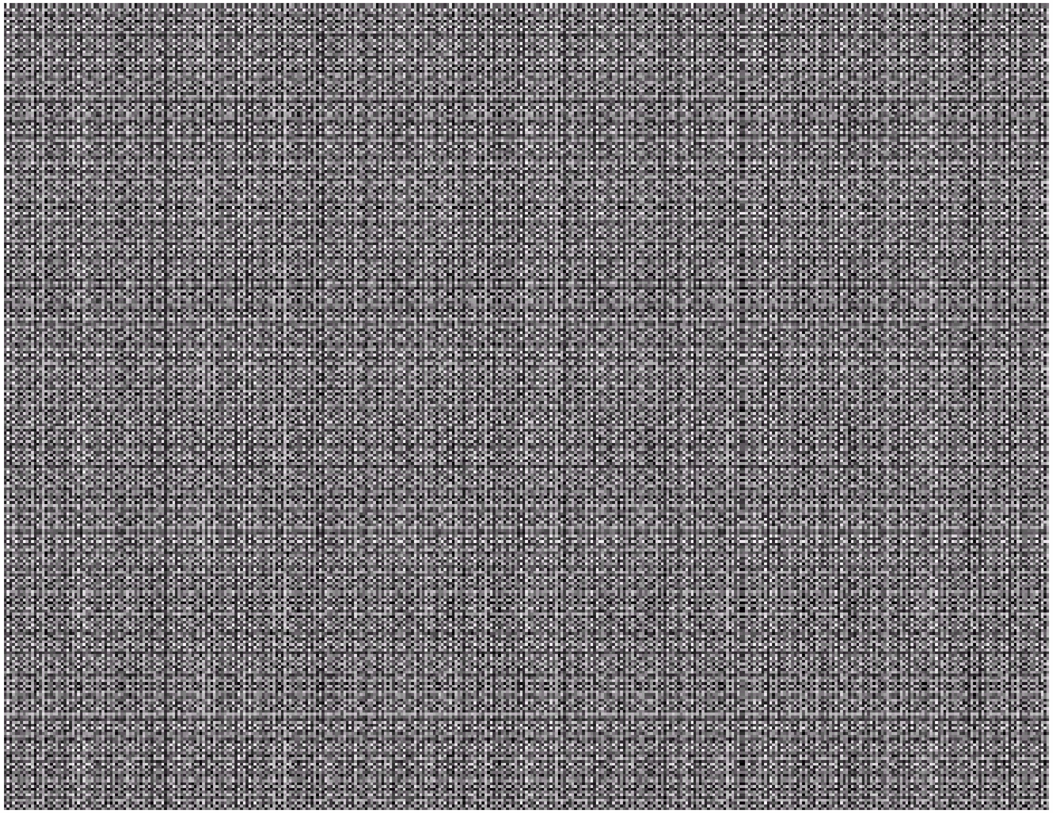
- Obtained form band reject filter

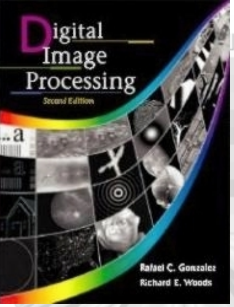
$$H_{bp}(u, v)=1-H_{br}(u, v)$$

- To isolated an image of certain frequency band.

5.4 Restoration using frequency-domain filter
Bandpass filter

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





5.4.3 Notch filter

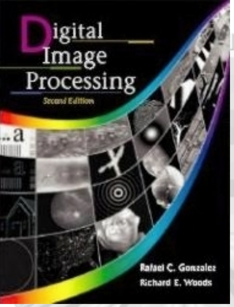
- Notch filter rejects (passes) frequencies in predefined neighborhoods about a center frequency. For notch rejects filter as

$$H(u,v)=\begin{cases}0 & \text{if } D_1(u,v)\le D_0 \quad \text{or} \quad D_2(u,v)\le D_0\\ 1 & \text{otherwise}\end{cases}$$

where

$$D_1(u,v)=\left[(u-M/2-u_0)^2+(v-N/2-v_0)^2\right]^{1/2}$$

$$D_2(u,v)=\left[(u-M/2+u_0)^2+(v-N/2+v_0)^2\right]^{1/2}$$



5.4 Restoration using frequency-domain filter -Notch filter

- Butterworth notch reject filter

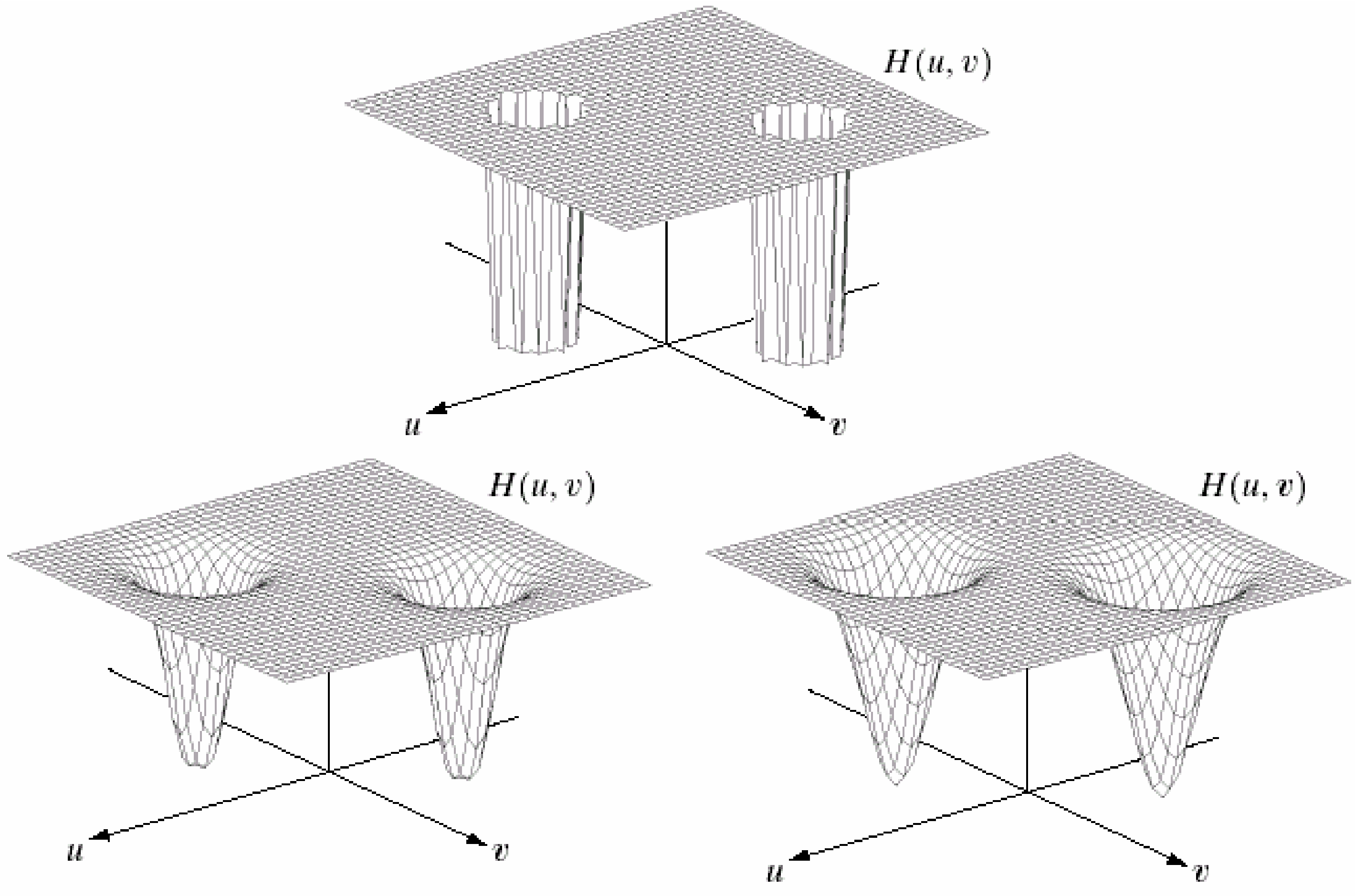
$$H(u,v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u,v) D_2(u,v)} \right]^n}$$

- Gaussian notch reject filter

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v) D_2(u,v)}{D_0^2} \right]}$$

- $H_{np}(u, v) = 1 - H_{nr}(u, v)$
- It becomes a low-pass filter when $u_0 = v_0 = 0$

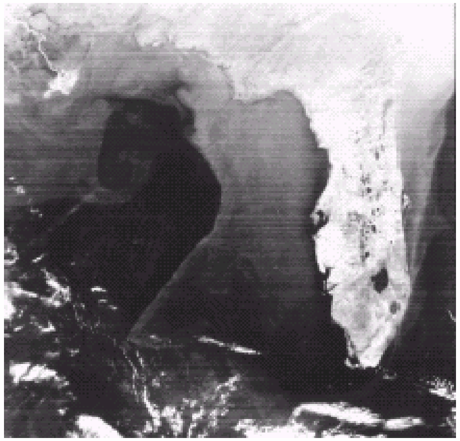
5.4 Restoration using frequency-domain filter
Notch filter



a
b c

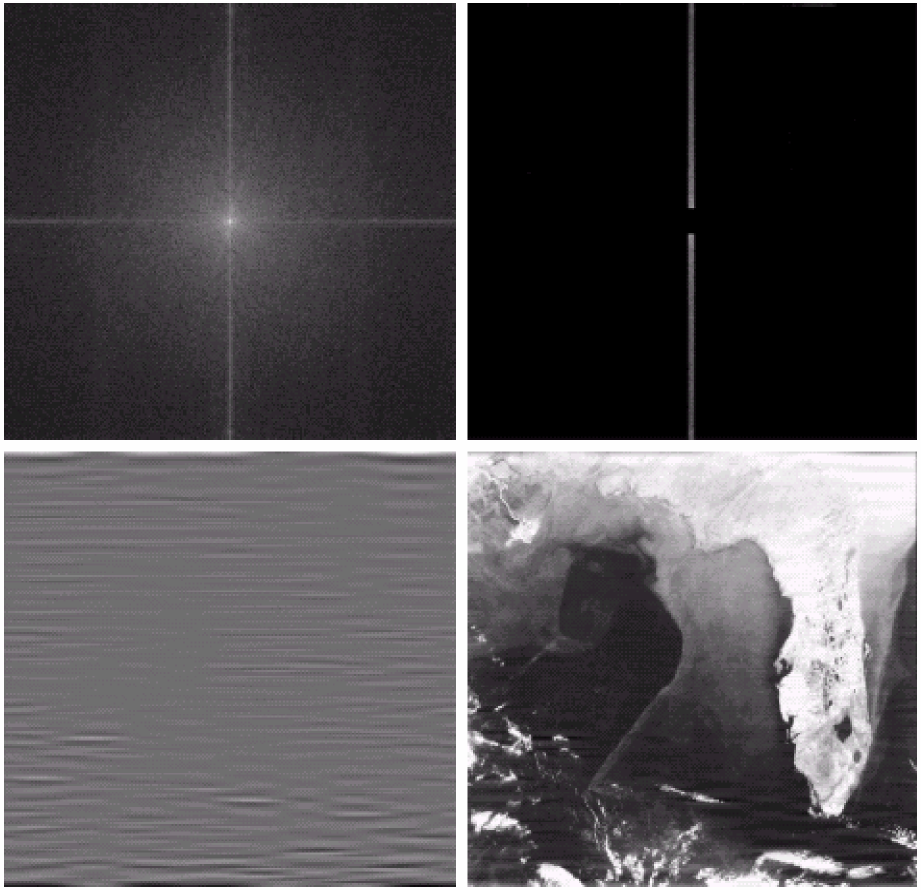
FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

5.4 Restoration using frequency-domain filter Notch filter



a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)



Use 1-D Notch pass filter to find the horizontal ripple noise

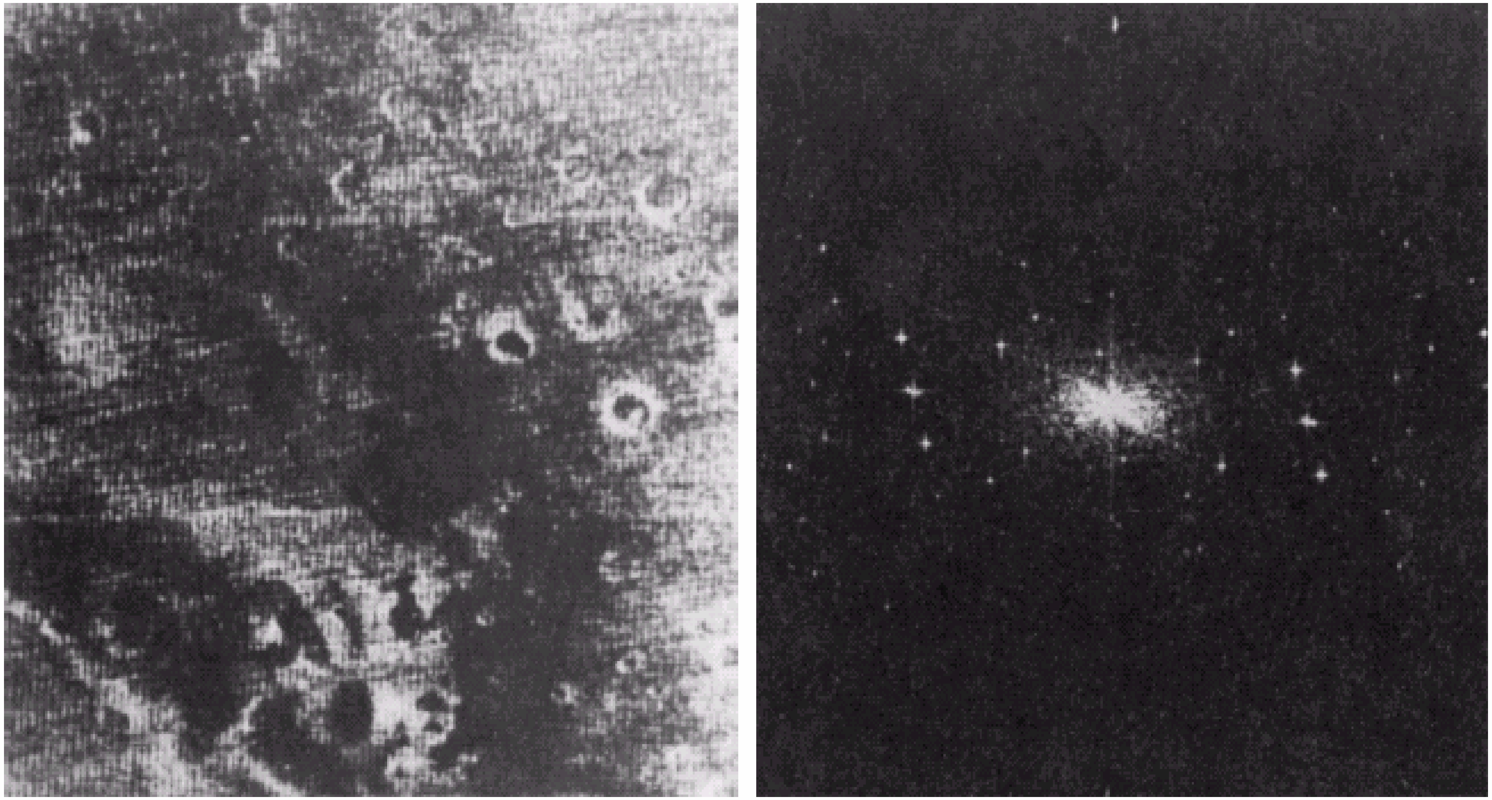
5.4.4 Optimal Notch filter

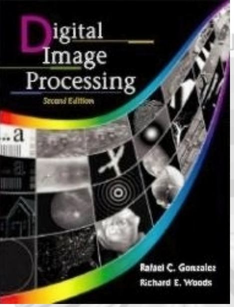
- Clearly defined interference is not common.
- Images from electro-optical scanner are corrupted by periodic degradation.
- Several interference components are present.
- Place a **notch pass filter** $H(u, v)$ at the location of each spike, *i.e.*, $N(u, v)=H(u, v)G(u, v)$, where $G(u, v)$ is the corrupted image.
- $\eta(x, y)=\mathcal{F}^{-1}\{H(u, v)G(u, v)\}$.
- The effect of components not present in the estimate of $\eta(x, y)$ can be minimized.

5.4 Restoration using frequency-domain filter

a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6.* (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)





5.4.4 Optimal Notch filter

- The restored image is $\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y)$
- The weighting function $w(x, y)$ is found so that the variance of is minimized for a selected neighborhood.
- One way is to select $w(x, y)$ so that the variance of the estimate $\hat{f}(x,y)$ is minimized over a small local neighborhood of size $(2a+1, 2b+1)$.

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{t=-b}^{b} \sum_{s=-a}^{a} \left[\hat{f}(x+s, y+t) - \bar{\hat{f}}(x,y) \right]^2$$

- $\partial \sigma^2(x, y) / \partial w(x, y) = 0 \rightarrow$ to select $w(x, y)$

5.4.4 Restoration using frequency-domain filter

$a = b = 15$

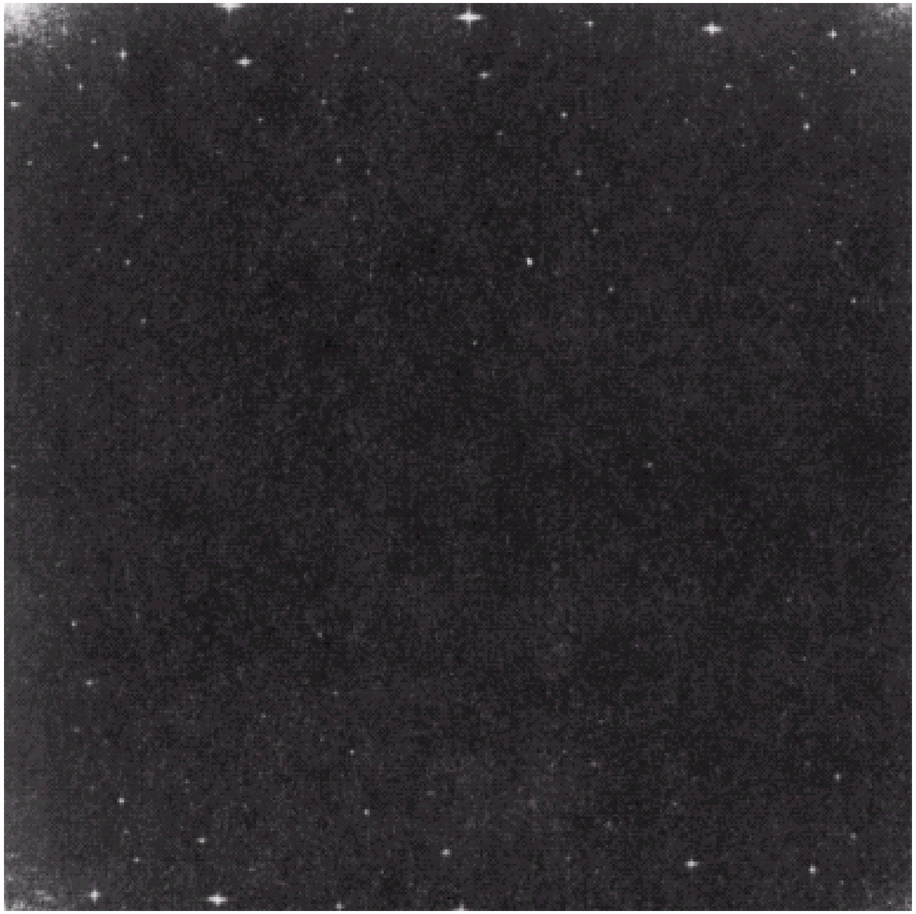
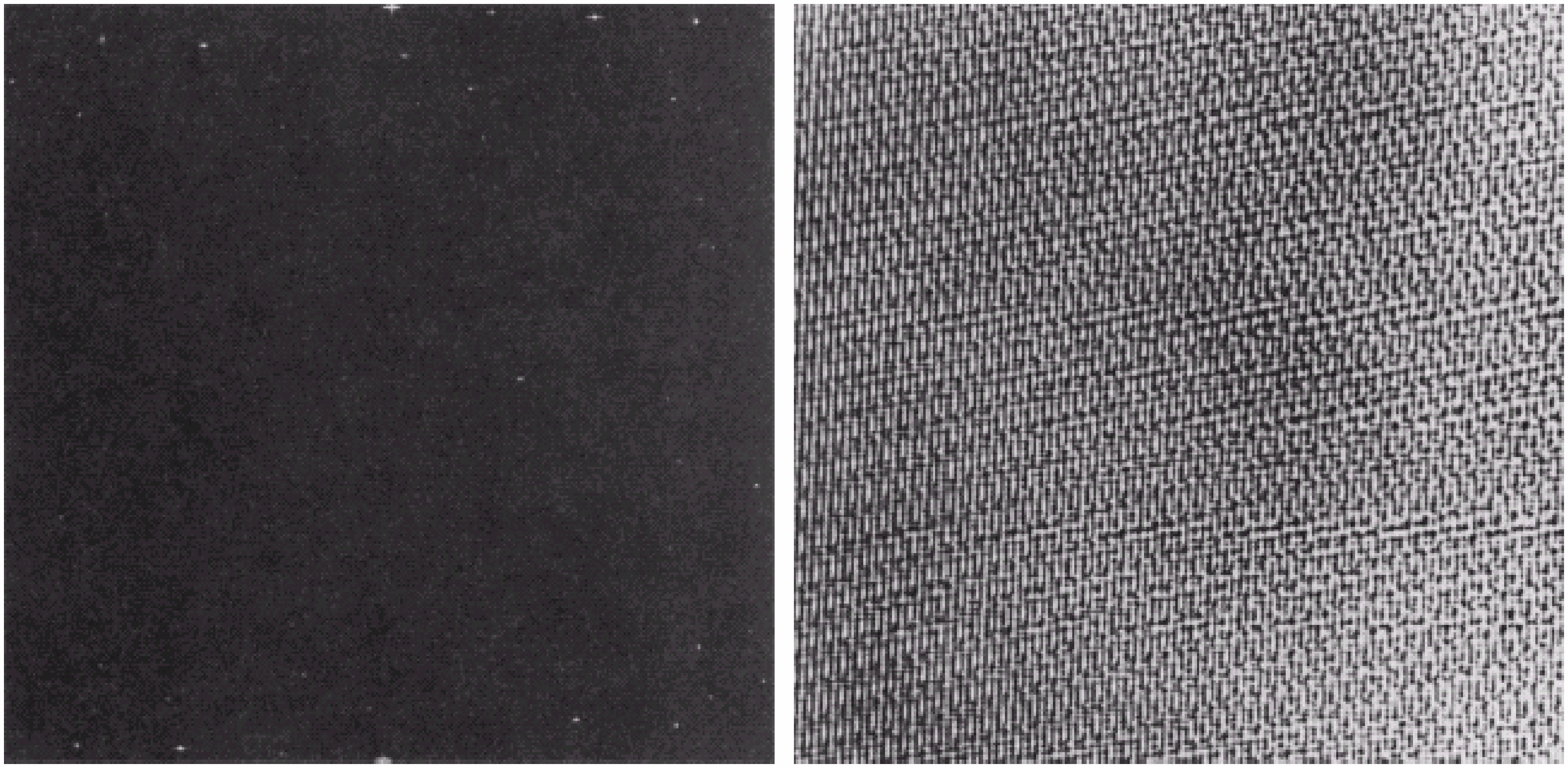


FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)

5.4 Restoration using frequency-domain filter



a b

FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

5.4 Restoration using frequency-domain filter

$$\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y)$$

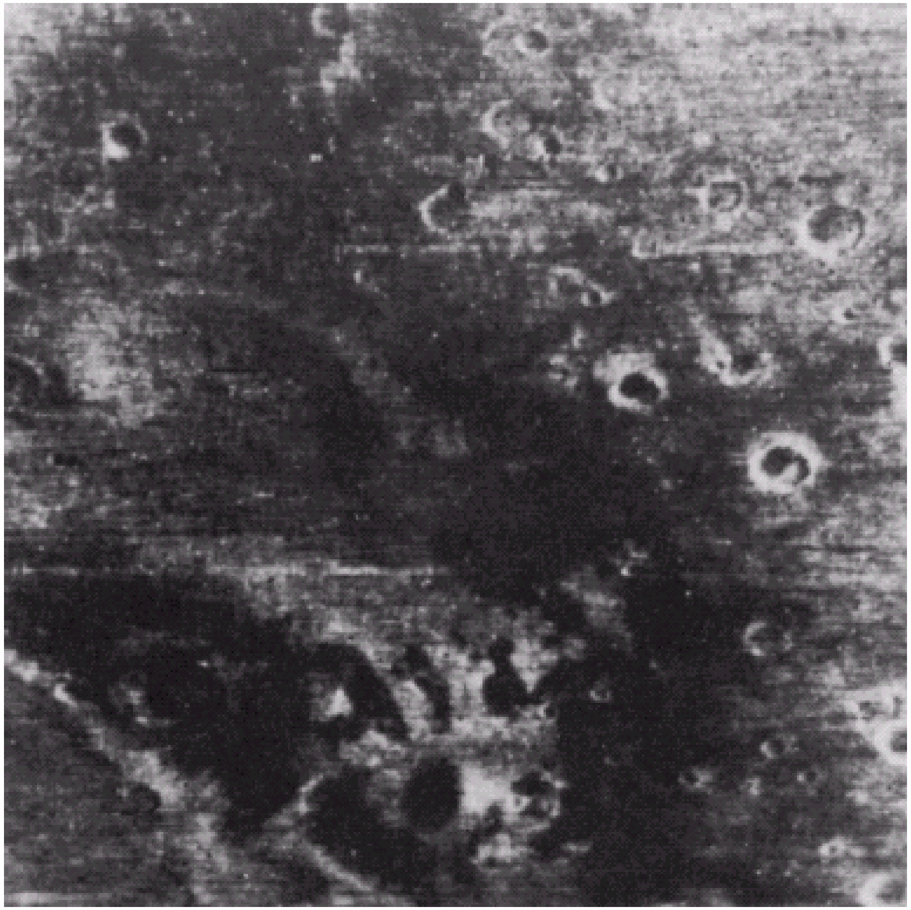
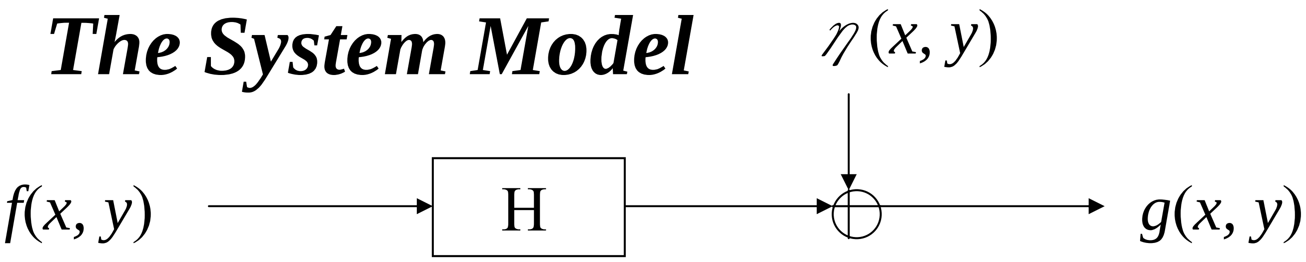


FIGURE 5.23 Processed image. (Courtesy of NASA.)

5.5 Linear Position Invariant Degradation

- ***The System Model***



g(x, y): the degraded image

f(x, y) : the original image

η(x, y): additive noise

H: System function which is a linear operator

5.5 Linear Position Invariant Degradation

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

Assume H is linear and $\eta(x, y)=0$

If $g(x, y)=H[f(x, y)]$ is position invariant then

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

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

$$=\iint f(\alpha,\beta)h(x-\alpha,y-\beta)d\alpha d\beta+\eta(x, y)$$

$$=h(x, y)*f(x, y)+ \eta(x, y)$$

5.6 Estimating the Degradation Function

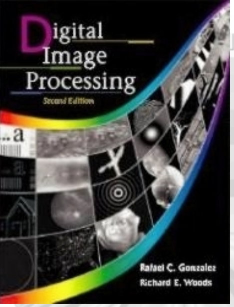
- To estimate the degradation function
  - by Observation
  - by Experimentation
  - by Mathematical modeling

(Blind deconvolution)

5.6 Estimating the Degradation Function

- ***By observation***: construct an unblurred image of some ***strong signal content***.
- Let the observed image is $g_s(x, y)$
- The constructed image is $\tilde{f}_s(x, y)$
- Assume the noise is negligible
- Find the degradation function $H(u, v)$ which is similar to

$$H_s(u,v) = \frac{G_s(u,v)}{\tilde{F}_s(u,v)}$$

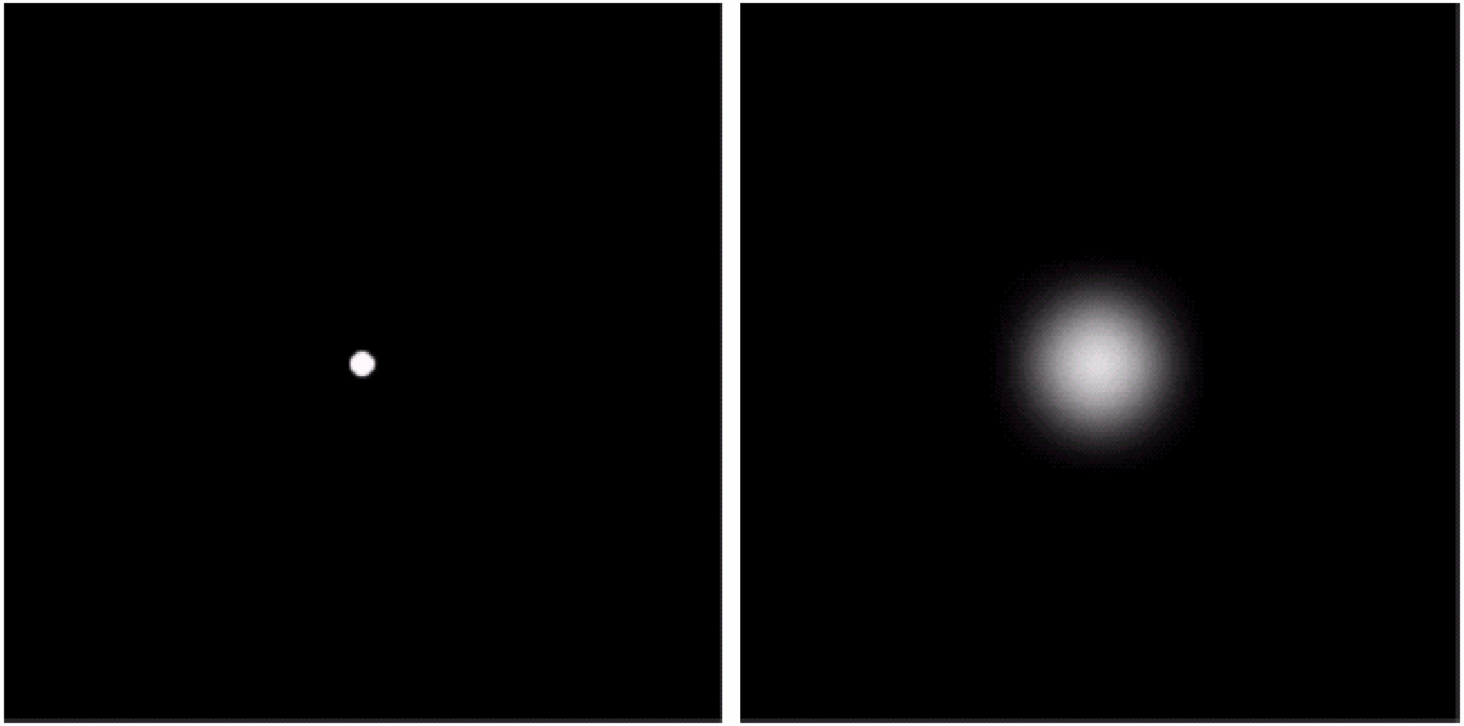


5.6 Estimating the Degradation Function

- ***By experimentation***: to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system setting.
- The Four Transform of an impulse is a constant A
- Let $g(x, y)$ be the observed image.
- Find the degradation function as

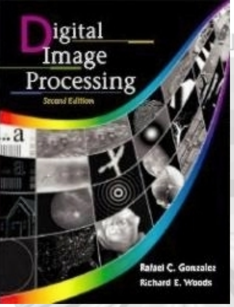
$$H(u, v)=G(u, v)/A$$

5.6 Estimating the Degradation Function



a b

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



5.6 Estimating the Degradation Function -by modeling

- ***By Modeling***: through experience
- The physical characteristic of turbulence as

$$H(u, v)=e^{-k(u^2+v^2)^{5/6}}$$

- Where k is a constant which depend on the nature of turbulence.
- It is similar to the Gaussian Low pass.

k=0.0025 (severe turbulence)

k=0.001

k=0.00025 (low turbulence)

5.6 Estimating the Degradation Function

a b
c d

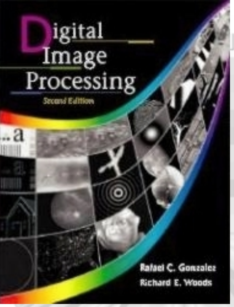
FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

5.6 Estimating the Degradation Function -by modeling

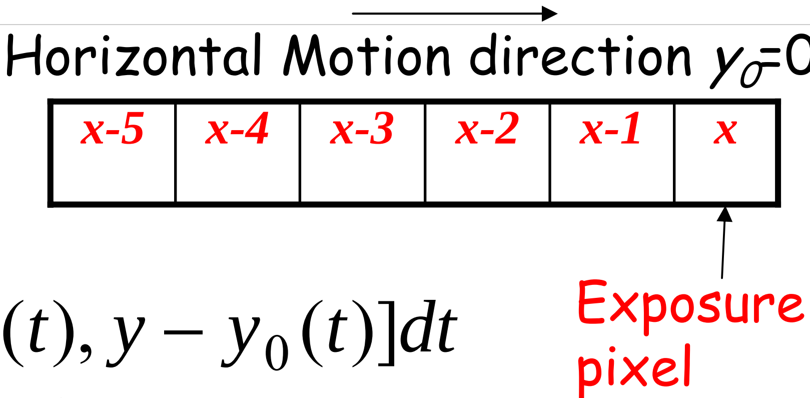
- ***Through mathematical model***

  Image $f(x, y)$ is blurred by ***uniform motion***.

  - $x_0(t)$ and $y_0(t)$ are the time varying component in x, y directions.
  - The total exposure at any point of the film is obtained by integrating the instantaneous exposure over the time interval during which the shutter is opened.



5.6 Estimating the Degradation Function -by modeling



- Assume ***T*** is ***duration of exposure***, the blurred image $g(x, y)$ is

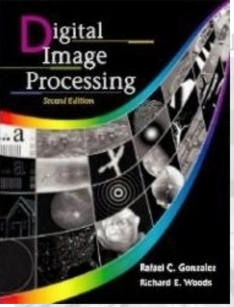
$$g(x,y)=\int_0^T f[x-x_0(t),y-y_0(t)]dt$$

$x_0(t)=at/T$, T=constant
a=velocity

- Fourier Transform of $g(x, y)$ is

$$G(u,v)=\iint g(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

$$=\iint\left[\int_0^T f[x-x_0(t),y-y_0(t)]dt\right]exp[-j2\pi(ux+vy)]dxdy$$



5.6 Estimating the Degradation Function -by modeling

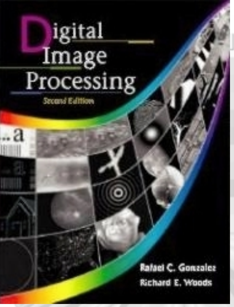
- Reverse the order of integration

$$G(u,v) == \int_0^T \left[\int \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] \exp[-j2\pi(ux + vy)]dxdy \right] dt$$

- The term inside the outer brackets is Fourier transform of the displacement
- Therefore

$$G(u,v) = \int_0^T F(u,v) \exp[-j2\pi(ux_0(t) + vy_0(t))]dt$$

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



5.6 Estimating the Degradation Function -by modeling

- Assume uniform motion in x direction only, *i.e.*, $x_0(t)=at/T$, $y_0(t)=0$, a is *velocity*, T=*exposure duration*
- Simplify $H(u,v)=\int_0^T \exp[-j2\pi(ux_0(t)+vy_0(t))]dt$ as

$$H(u,v)=\int_0^T \exp[-j2\pi ux_0(t)]dt$$

$$=\int_0^T \exp[-j2\pi uat/T)]dt=\frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

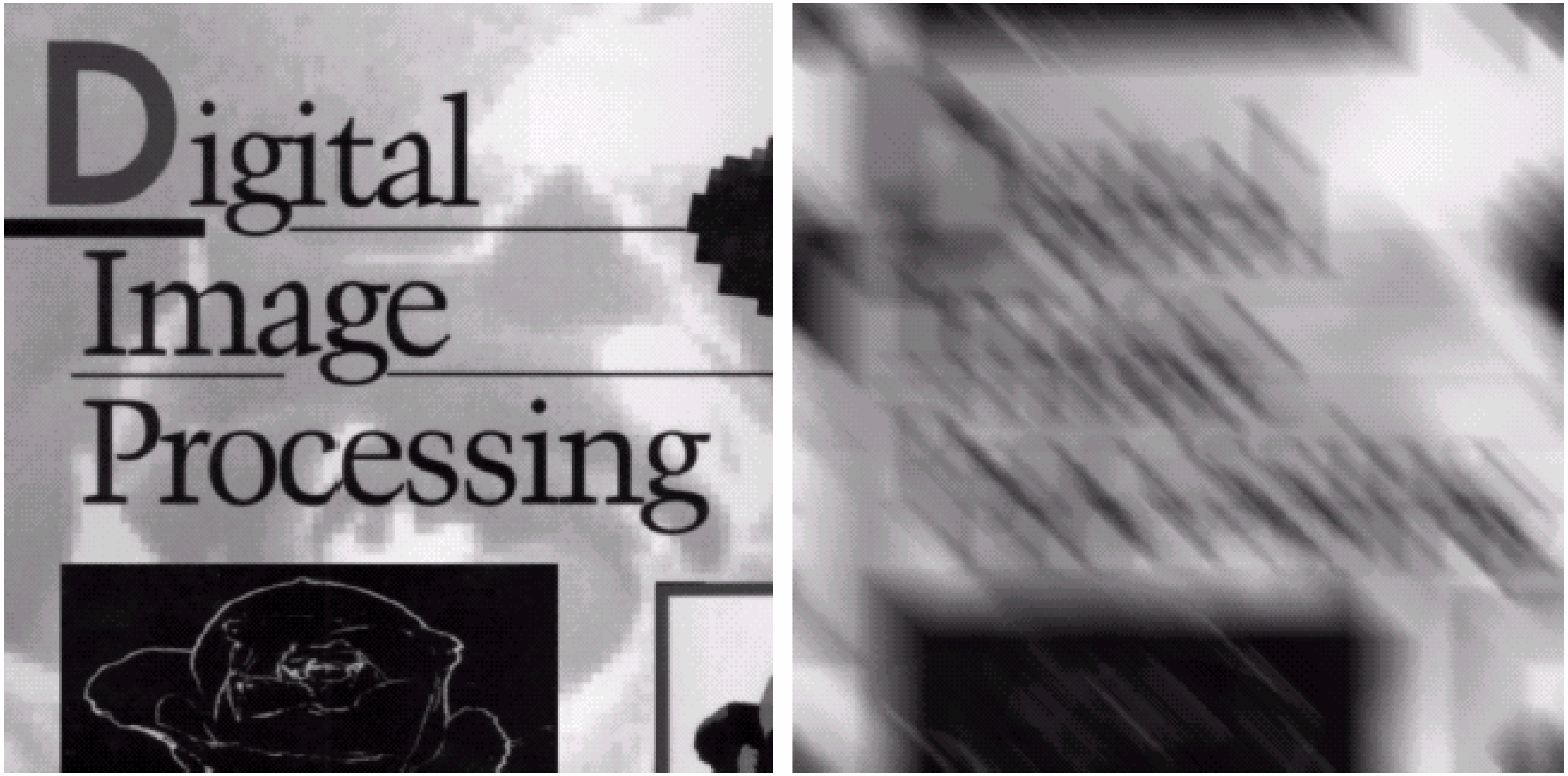
- ***Problem***: when $u=n/a$, $H(u, v)=0$

5.6 Estimating the Degradation Function -by modeling

- If we allow y-component movement, with the motion given by $y_0 = bt/T$ then the degradation function is

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

5.6 Estimating the Degradation Function -by modeling



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 $T = 1$.

5.6 Estimating the Degradation Function -by spatial modeling

- Assume *horizontal motion only* (y_0=0), then the blurred image is

$$g(x)=\int_0^T f[x-x_0(t)]dt=\int_0^T f(x-at/T)dt$$

where $0 \le x \le L$

a=displacement /unit time

- Substitute $\tau = x - at/T$

we have $g(x)=\int_{x-a}^{x} f(\tau)d\tau$

- Differentiation with respect to *x*

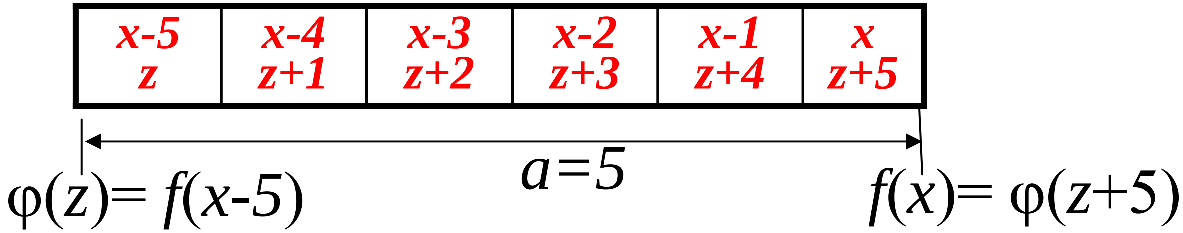
$$g'(x)=f(x)-f(x-a) \quad \text{or} \quad f(x)=g'(x)+f(x-a)$$

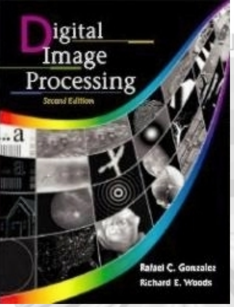
5.6 Estimating the Degradation Function -by modeling

- Let $L=Ka$, where K is integer, then $x=z+ma$ where $0 \leq z \leq a$, $m=Integer[x/a]$, and $m=0,1,\ldots K-1$.
- Substitute x with $z+ma$

$$f(z+ma)=g'(z+ma)+f(z+(m-1)a)$$

- Denote $\phi(z)$ as the portion of the scene moves into the range $0 \leq z \leq a$ during exposure, *i.e.*, $\phi(z)=f(z-a)$





5.6 Estimating the Degradation Function -by modeling

- For $m=0$

$$f(z) = g'(z) + f(z-a) = g'(z) + \phi(z)$$

- For $m=1$

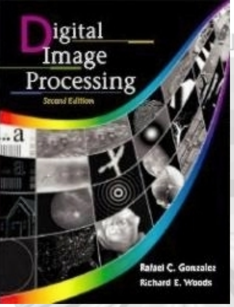
$$f(z+a) = g'(z+a) + f(z) = g'(z+a) + g'(z) + \phi(z)$$

- For $m=2$

$$f(z+2a) = g'(z+2a) + g'(z+a) + g'(z) + \phi(z)$$

- Finally we have

$$f(z+ma) = \sum_{k=0}^{m} g'(z+ka) + \phi(z)$$



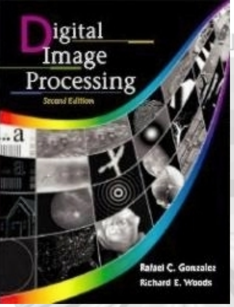
5.6 Estimating the Degradation Function -by modeling

- For $x=z+ma$ and $0 \leq x \leq L$, we have

$$f(x)=\sum_{k=0}^{m} g'(x-ka)+\phi(x-ma)$$

$g(x)$ is known, we may estimate $\phi(x)$ to find $f(x)$.

- Since $0 \leq x\text{-}ma \leq a$, so $\phi(x\text{-}ma)$ is repeated K times during the evaluation of $f(x)$ for $0 \leq x \leq L$
- Define $$\hat{f}(x)=\sum_{j=0}^{m} g'(z-ja)$$
- So we have $\phi(x-ma)=f(x)-\hat{f}(x)$



5.6 Estimating the Degradation Function -by modeling

- For $ka \leq x \leq (k+1)a$, and adding the results for $k=1,\ldots,K-1$

$$K\phi(x) = \sum_{k=0}^{K-1} f(x+ka) - \sum_{k=0}^{K-1} \hat{f}(x+ka)$$

where $m=0$ and $0 \leq x \leq a$

- For large value of K (small a)

$$\phi(x) = \frac{1}{K}\sum_{k=0}^{K-1} f(x+ka) - \frac{1}{K}\sum_{k=0}^{K-1} \hat{f}(x+ka) \approx A - \frac{1}{K}\sum_{k=0}^{K-1} \hat{f}(x+ka)$$

where constant A is the average of f

5.6 Estimating the Degradation Function -by modeling

- For $\phi(x\text{-}ma)$

$$\phi(x-ma) \approx A - \frac{1}{K}\sum_{k=0}^{K-1} \hat{f}(x+ka-ma)$$

$$\approx A - \frac{1}{K}\sum_{k=0}^{K-1}\sum_{j=0}^{k} g^{'}(x+ka-ma-ja)$$

- The restored image is

$$f(x) = \sum_{k=0}^{m} g^{'}(x-ka) + \phi(x-ma)$$

5.6 Estimating the Degradation Function -by modeling

(a) (b)

Figure 5.4 (a) Image blurred by uniform linear motion; (b) image restored by using Eq. (5.4-36). (From Sondhi [1972].)

5.7 Inverse filtering

- With degraded image: $\mathbf{g} = \mathbf{H} \cdot \mathbf{f} + \mathbf{n}$
- Our goal is to find $\hat{\mathbf{f}}$ such that $\mathbf{H}\hat{\mathbf{f}}$ approximate **g** in a least square sense

$$\mathbf{n} = \mathbf{g} - \mathbf{H} \cdot \hat{\mathbf{f}}$$

$$\|\mathbf{n}\|^2 = \left\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\right\|^2 = (\mathbf{g} - \mathbf{H}\hat{\mathbf{f}})^T (\mathbf{g} - \mathbf{H}\hat{\mathbf{f}})$$

5.7 Inverse filtering

- Minimizing $\|\mathbf{n}\|^2 = J(\hat{\mathbf{f}})$ by using $\partial J(\hat{\mathbf{f}})/\partial\hat{\mathbf{f}} = 0$

$$\partial J(\hat{\mathbf{f}})/\partial\hat{\mathbf{f}} = -2\mathbf{H}^T(\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}) = 0$$

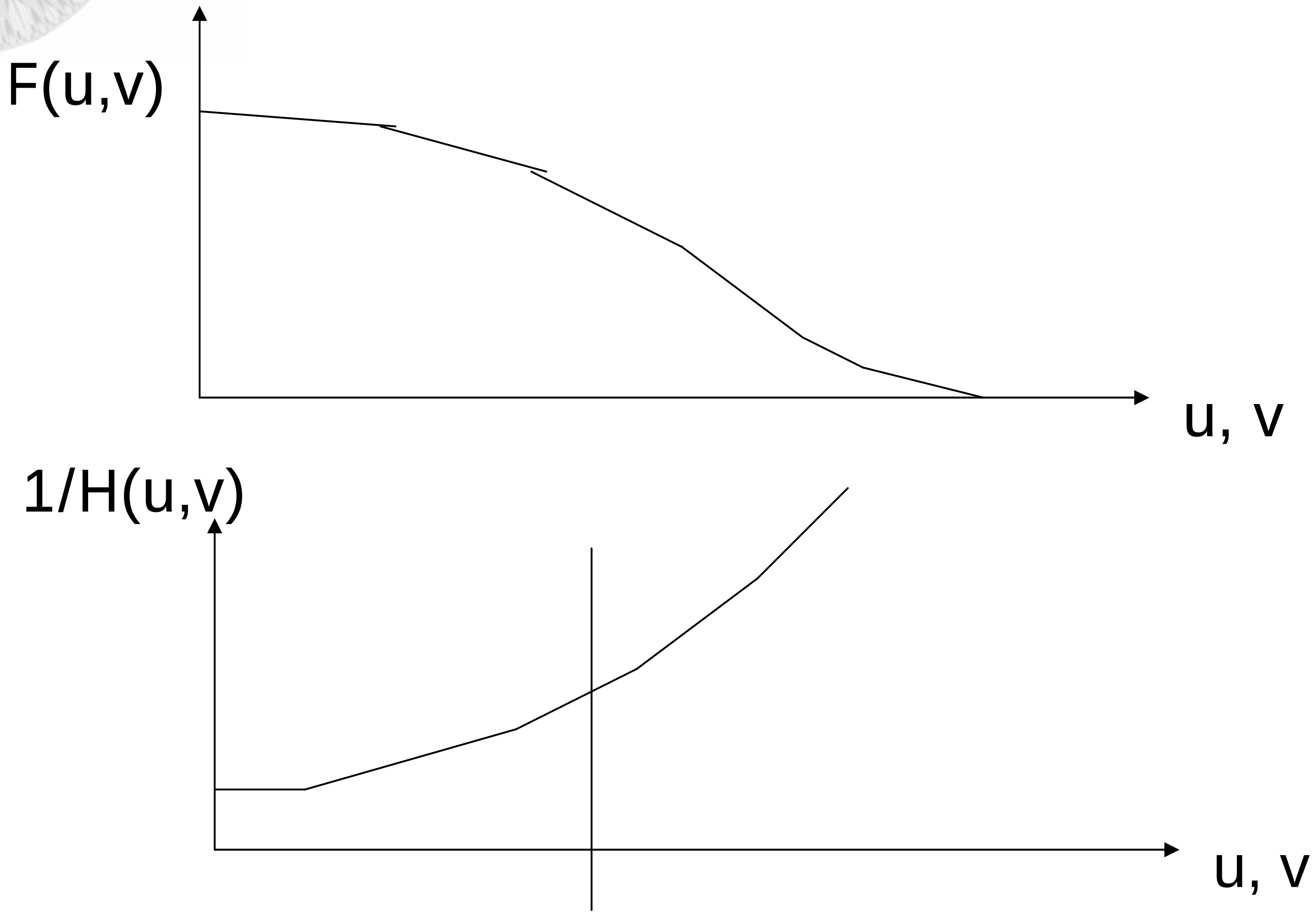
$$\hat{\mathbf{f}} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{g} = \mathbf{H}^{-1}(\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{g} = \mathbf{H}^{-1}\mathbf{g}$$

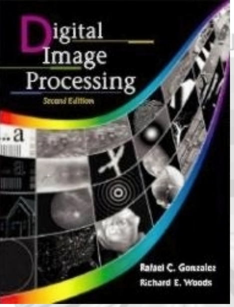
- Or we may have $\hat{\mathbf{F}}(u,v) = \dfrac{\mathbf{G}(u,v)}{\mathbf{H}(u,v)}$

5.7 Inverse filtering

- ***Without noise*** $$\mathbf{G}(u,v) = \mathbf{H}(u,v)\mathbf{F}(u,v)$$
- We have $$\hat{\mathbf{F}}(u,v) = \frac{\mathbf{G}(u,v)}{\mathbf{H}(u,v)} = \mathbf{F}(u,v)$$
- ***With noise*** $$\mathbf{G}(u,v) = \mathbf{H}(u,v)\mathbf{F}(u,v) + \mathbf{N}(u,v)$$
- We have $$\hat{\mathbf{F}}(u,v) = \mathbf{F}(u,v) + \frac{\mathbf{N}(u,v)}{\mathbf{H}(u,v)}$$
- *Restored image* $$\hat{f}(x,y) = F^{-1}\{\hat{\mathbf{F}}(u,v)\}$$

5.7 Inverse filtering





5.7 Inverse filtering

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.

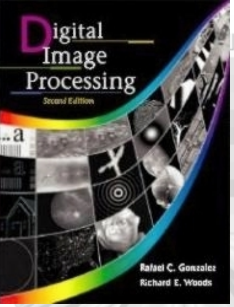
Degradation function:

$$H(u,.v)=e^{-k(u^2+v^2)^{5/6}}$$

$K=0.0025,\ M=N=480$

5.8 Minimum Mean Square Error (Wiener) Filtering

- Incorporates both the degradation function and statistical characteristics of noise into restoration
- Considering both the image and noise as random processes, and find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized
- The error measure is $e^2 = E\{(f - \hat{f}\,)^2\}$



5.8 Minimum Mean Square Error (Wiener) Filtering

- Minimizing the error we have

$$E\{[f(x,y)-\hat{f}(x,y)]g(x',y')\}=0$$

- For image coordinate pair (x, y) and (x', y'), we assume the restoration filter is $h_R(x, y)$, and have

$$E\{f(x,y)g(x,y)\}=\int\int E\{g(x,y)g(x',y')\}h_R(x-\alpha,y-\beta)d\alpha d\beta$$

- Assume the ideal image and observed image are jointly stationary, the expectation term can be expressed as covariance function as

$$K_{fg}(x-x',y-y')=\int\int_{-\infty}^{\infty}K_{gg}(\alpha-x',\beta-y')h_R(x-\alpha,y-\beta)d\alpha d\beta$$

5.8 Minimum Mean Square Error (Wiener) Filtering

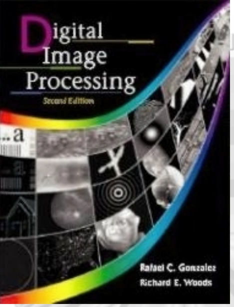
- Take Fourier Transform we have

$$H_R(u, v)=W_{fg}(u, v)W_{gg}^{-1}(u, v)$$

with additive noise

$$W_{fg}(u, v)=H^*(u, v)S_f(u, v)$$

$$W_{gg}(u, v)= |H(u, v)|^2 S_f (u, v)+S_\eta(u, v)$$



5.8 Minimum Mean Square Error (Wiener) Filtering

Assume the image and noise are uncorrelated

$$\hat{F}(u,v) = \left[\frac{H^*(u,v) S_f(u,v)}{S_f(u,v)\left|H(u,v)\right|^2 + S_\eta(u,v)} \right] G(u,v)$$

$$= \left[\frac{H^*(u,v)}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

$$= \left[\frac{1}{H(u,v)} \frac{\left|H(u,v)\right|^2}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

5.8 Minimum Mean Square Error (Wiener) Filtering

- $H(u, v)$: the degradation function
- $S_\eta(u, v) = |N(u, v)|^2$ = power spectrum of the noise
- $S_f(u, v) = |F(u, v)|^2$ = power spectrum of the undegraded function
- If noise is zero, then it becomes an inverse filter.
- For a white noise $|N(u, v)|^2$ = constant then

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

5.8 Minimum Mean Square Error (Wiener) Filtering

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

5.8 Minimum Mean Square Error (Wiener) Filtering

a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

5.9 Constraint Least Squares Filtering

- ***Difficulty for Wiener filter***: the power spectra of the undegraded image and the noise must be known.
- Only the mean and variance of the noise are required
- Given a noisy image in vector form $\mathbf{g} = \mathbf{H} \cdot \mathbf{f} + \mathbf{n}$

  with $g(x, y)$ has a size M×N, the matrix **H** has dimension MN × MN and **H** is highly sensitive to noise.
- Base optimality of restoration on a measure of smoothness, such as the 2nd derivative of an image, *i.e.*, $$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x, y) \right]^2$$
- Find the minimum of C subject to the constraint

$$\left\| \mathbf{g} - \mathbf{H}\hat{\mathbf{f}} \right\|^2 = \left\| \boldsymbol{\eta} \right\|^2$$

5.9 Constraint Least Squares Filtering

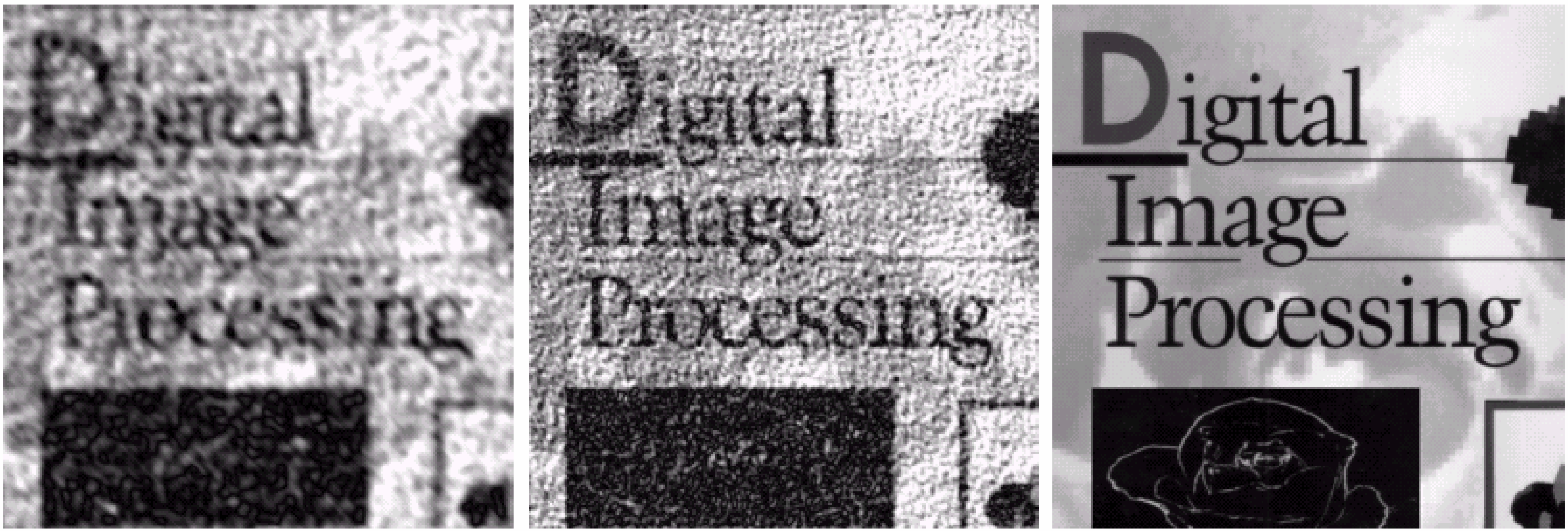
- The frequency domain solution to this optimization problem is

$$F(u,v) = \left[\frac{H^*(u,v)}{|U(u,v)|^2 + \gamma |P(u,v)|^2} \right] G(u,v)$$

- P(u,v) is Fourier transform of p(*x*, *y*)

$$p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

5.9 Constraint Least Squares Filtering



a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

5.9 Constraint Least Squares Filtering

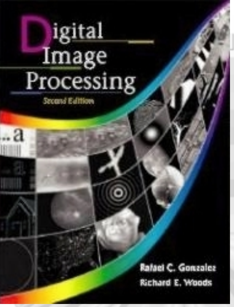
- To compute γ by iteration as follows.
- Define the residue vector **r**: $\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}$
- $\hat{F}(u,v)$ is a function of γ , so is **r**,
- $\phi(\gamma)=\mathbf{r}^T\mathbf{r}=\|\mathbf{r}\|^2$ is a monotonically increasing function of γ
- Adjust γ so that $\|\mathbf{r}\|^2 = \|\eta\|^2 \pm a$

  where a is a accuracy factor
- If $\|\mathbf{r}\|^2 = \|\eta\|^2$ then the best solution is found.

5.9 Constraint Least Squares Filtering

- Because $\phi(\gamma)$ is monotonic, finding γ is not difficult.

1. Specify an initial γ
2. Compute $\|\mathbf{r}\|^2$
3. if $\|\mathbf{r}\|^2 = \|\eta\|^2 \pm a$ is satisfied Stop
4. if $\|\mathbf{r}\|^2 < \|\eta\|^2 - a$ increase γ and go to step 2.
5. if $\|\mathbf{r}\|^2 > \|\eta\|^2 - a$ decrease γ and go to step 2.



5.9 Constraint Least Squares Filtering

- To compute the $\|\mathbf{r}\|^2$ and $\|\boldsymbol{\eta}\|^2$
- The vector from can also be rewritten as

$$R(u,v) = G(u,v) - H(u,v)\hat{F}(u,v)$$

- Compute the Inverse Fourier transform of $R(u, v)$

$$\|\mathbf{r}\|^2 = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r^2(x,y)$$

- Consider the variance of the noise over the entire image, using the sample-average method

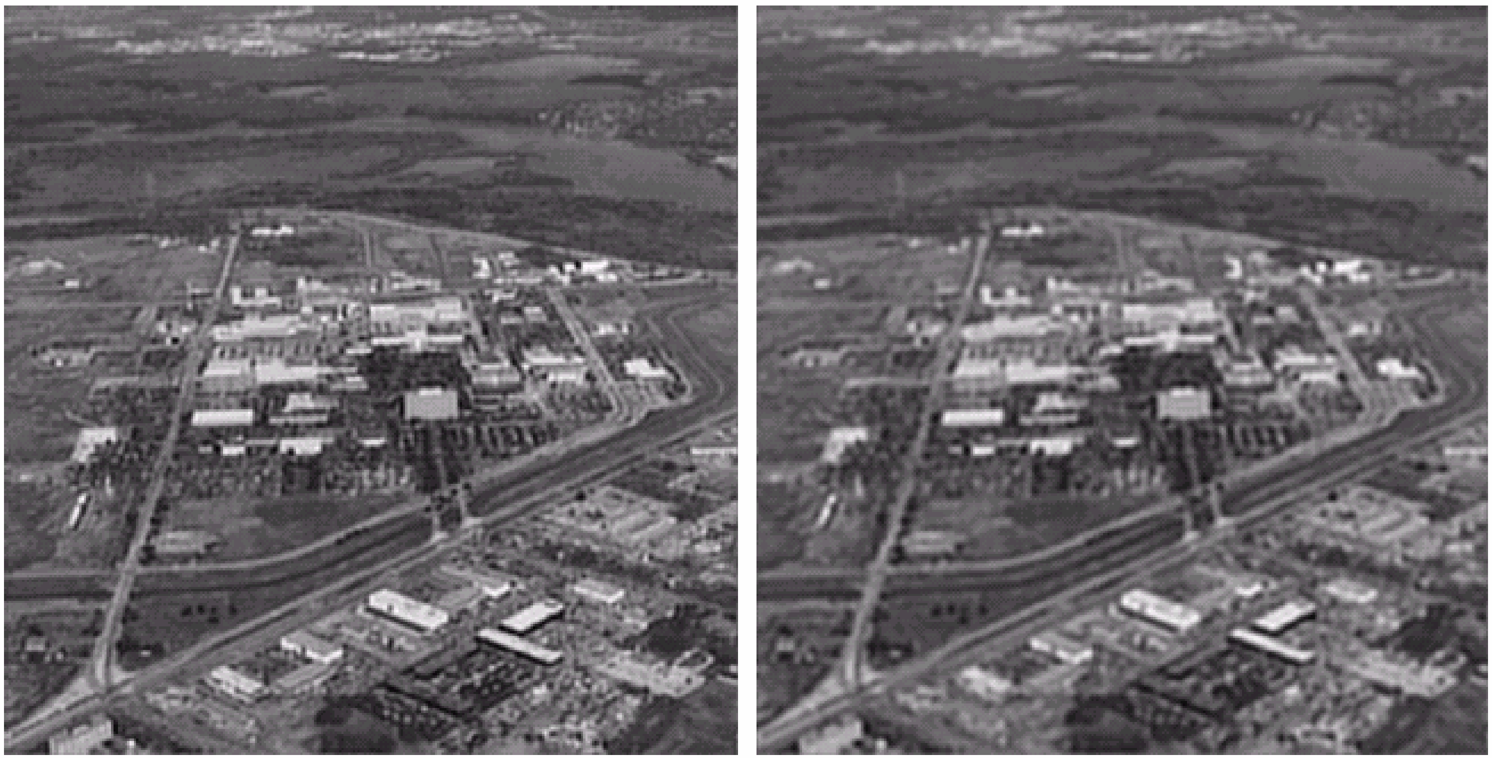
$$\sigma_\eta^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\eta(x,y) - m_\eta\right] \qquad m_\eta = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\eta(x,y)$$

- So we have $\|\boldsymbol{\eta}\|^2 = MN[\sigma^2_\eta - m_\eta]$

5.9 Constraint Least Squares Filtering

a b

FIGURE 5.31 (a) Iteratively determined constrained least squares restoration of Fig. 5.16(b), using correct noise parameters. (b) Result obtained with wrong noise parameters.



5.10 Geometric Mean Filter

Generalized form of Wiener filter:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha}\left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta\left[\frac{S_\eta(u,v)}{S_f(u,v)}\right]}\right]^{1-\alpha} G(u,v)$$

α, β are positive real constant

$\alpha=1$ reduces to inverse filter

$\alpha=0$ becomes parametric Wiener filter

$\alpha=0$, $\beta=1$ standard Wiener filter

$\alpha=1/2$, $\beta=1$ spectrum equalization filter

5.11 Geometric Transformation

- It is also called ***Rubber sheet transform***, which consists of two operations:

1) ***Spatial transformation***: ***rearrangement*** of the pixels (***locations***) on the image.
2) ***Gray-level interpolation***: ***assignment*** of ***gray levels*** to pixels in the spatially transformed image.

5.11 Geometric Transformation -Geometric Distortion

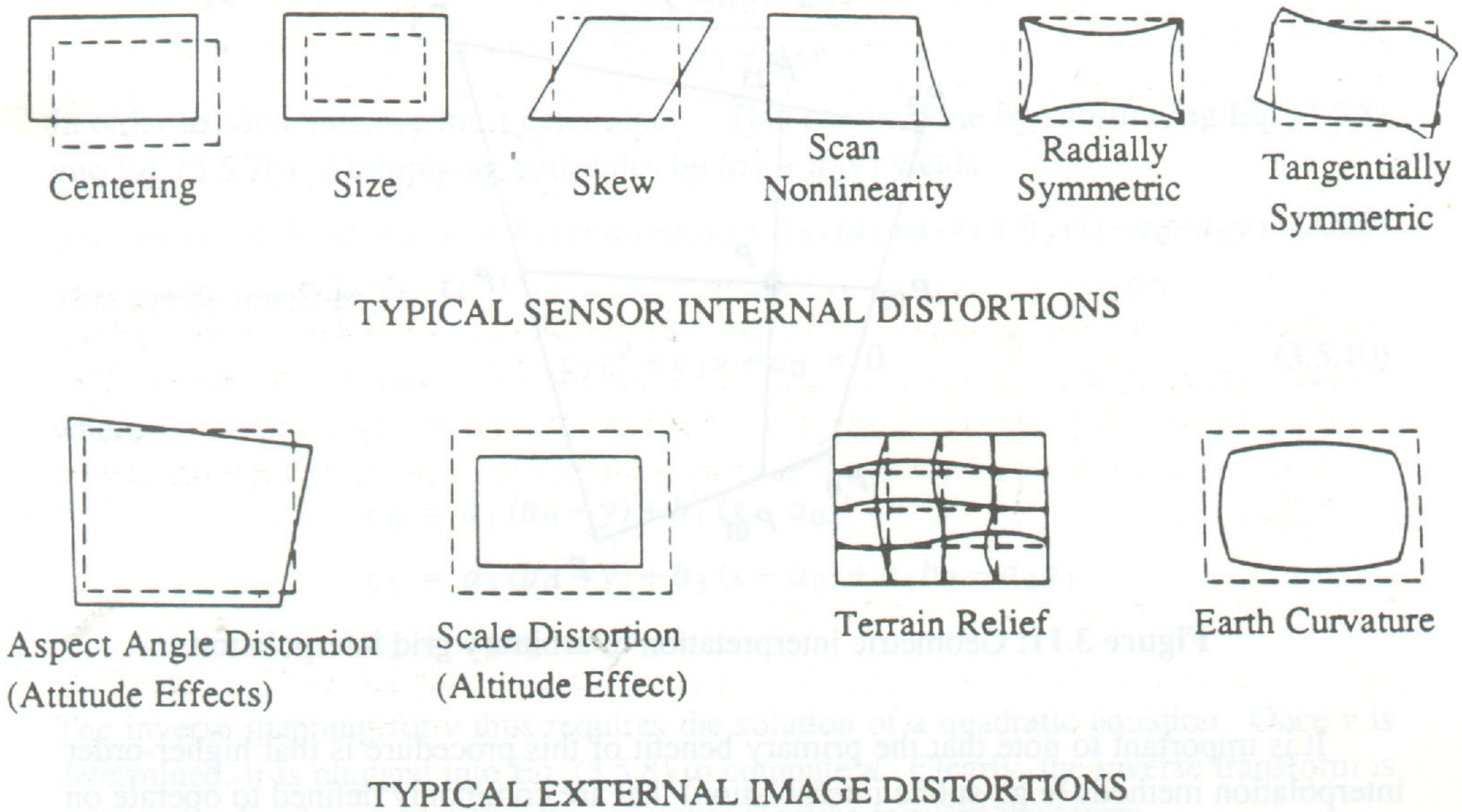


Figure 3.12: Common geometric image distortions.

5.11 Geometric Transformation
Spatial transformation

Mapping function *T* : maps a point from screen space to a point in texture space

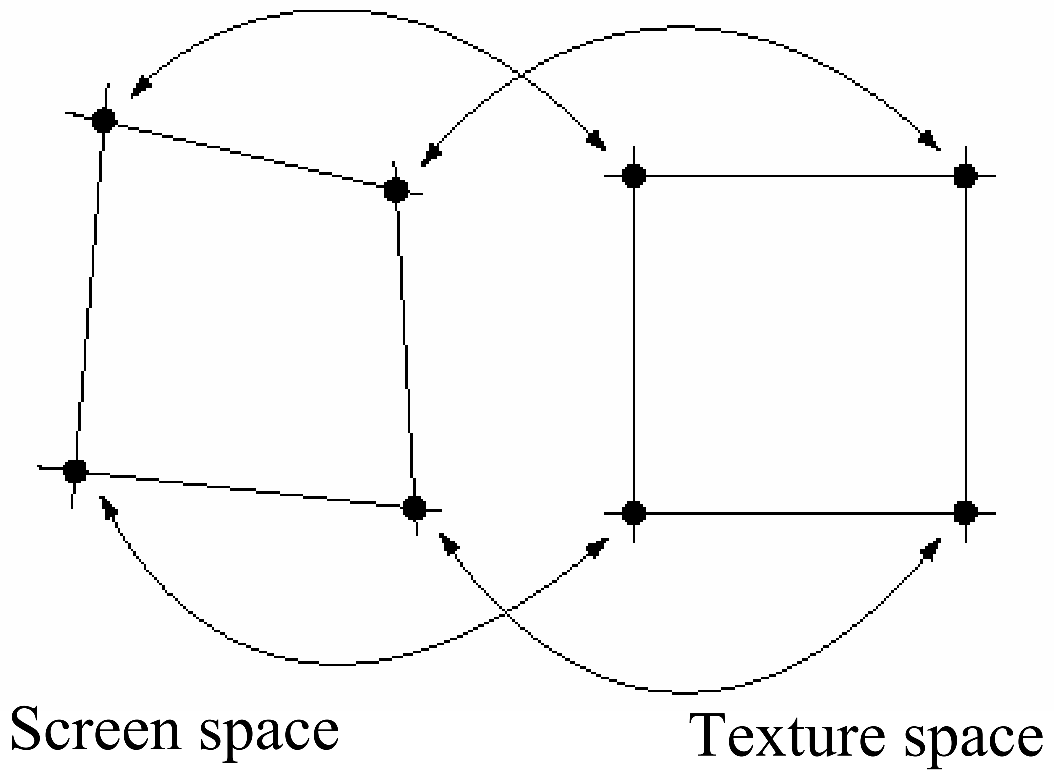
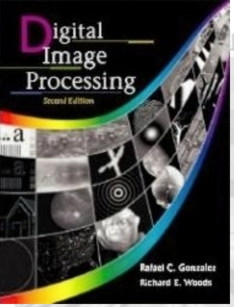


FIGURE 5.32 Corresponding tiepoints in two image segments.



5.11 Geometric Transformation

Spatial transformation

- The transformation T may be expressed as

$$x' = r(x, y) = c_1 x + c_2 y + c_3 xy + c_4$$

$$y' = s(x, y) = c_5 x + c_6 y + c_7 xy + c_8$$

Find the ***tiepoints***, which are subset of pixels whose locations in the distorted image (in screen space) and the corrected images (in texture space) are known precisely.

5.11 Geometric Transformation- *Spatial transformation*

- General Transformation:

$$\begin{bmatrix} x' & y' & w' \end{bmatrix} = \begin{bmatrix} u & v & w \end{bmatrix} T_1$$

where

$$T_1 = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}$$

5.11 Geometric Transformation- *Spatial transformation*

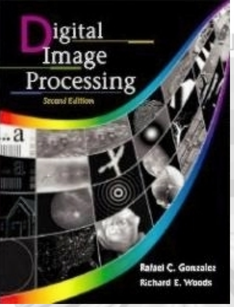
- T_1 is said to be one-to-one if:

(a) The inverse transformation of T_1 always exists.

(b) With T_1, one point in screen space produces only one point in texture space

(c) Through T_1^{-1}, each point in texture space can find the corresponding point in screen space.

=> that is the determinate of T_1 is nonzero.



5.11 Geometric Transformation- *Spatial transformation*

- The 3×3 transformation matrix can be best understood by partitioning it into four separate sections.
- Rotation: $T_2 = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$
- Translation: $\begin{bmatrix} S_{31} & S_{32} \end{bmatrix}$
- Perspective transformation: $\begin{bmatrix} S_{13} & S_{23} \end{bmatrix}^T$
- Scaling: S_{33}

5.11 Geometric Transformation- *Spatial transformation*

- The general representation of an ***Affine transformation*** is :

$$[x \quad y \quad 1] = [u \quad v \quad 1]\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}$$

It is a planar mapping, that can be used to map an input triangle to any arbitrary triangle at the output

5.11 Geometric Transformation-
Spatial transformation

Affine Transformation

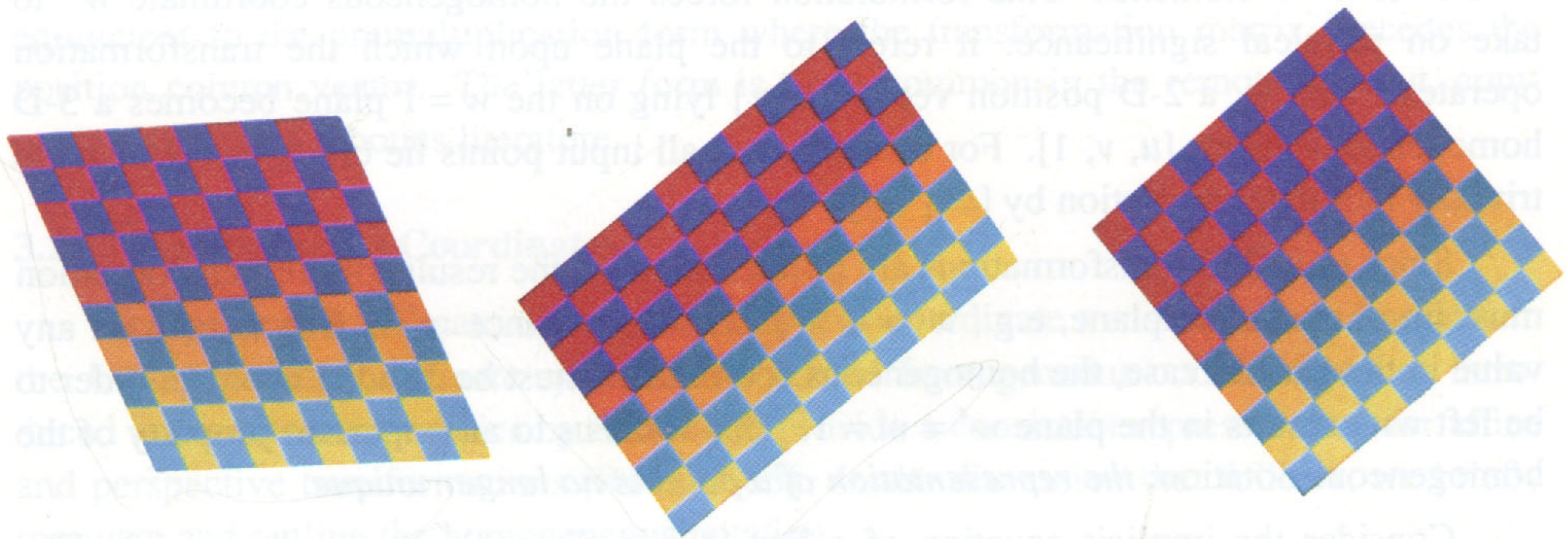


Figure 3.5: Affine warps.

5.11 Geometric Transformation-
Spatial transformation

■ Perspective Transformation

$$[x', y', w'] = [u, v, w]\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

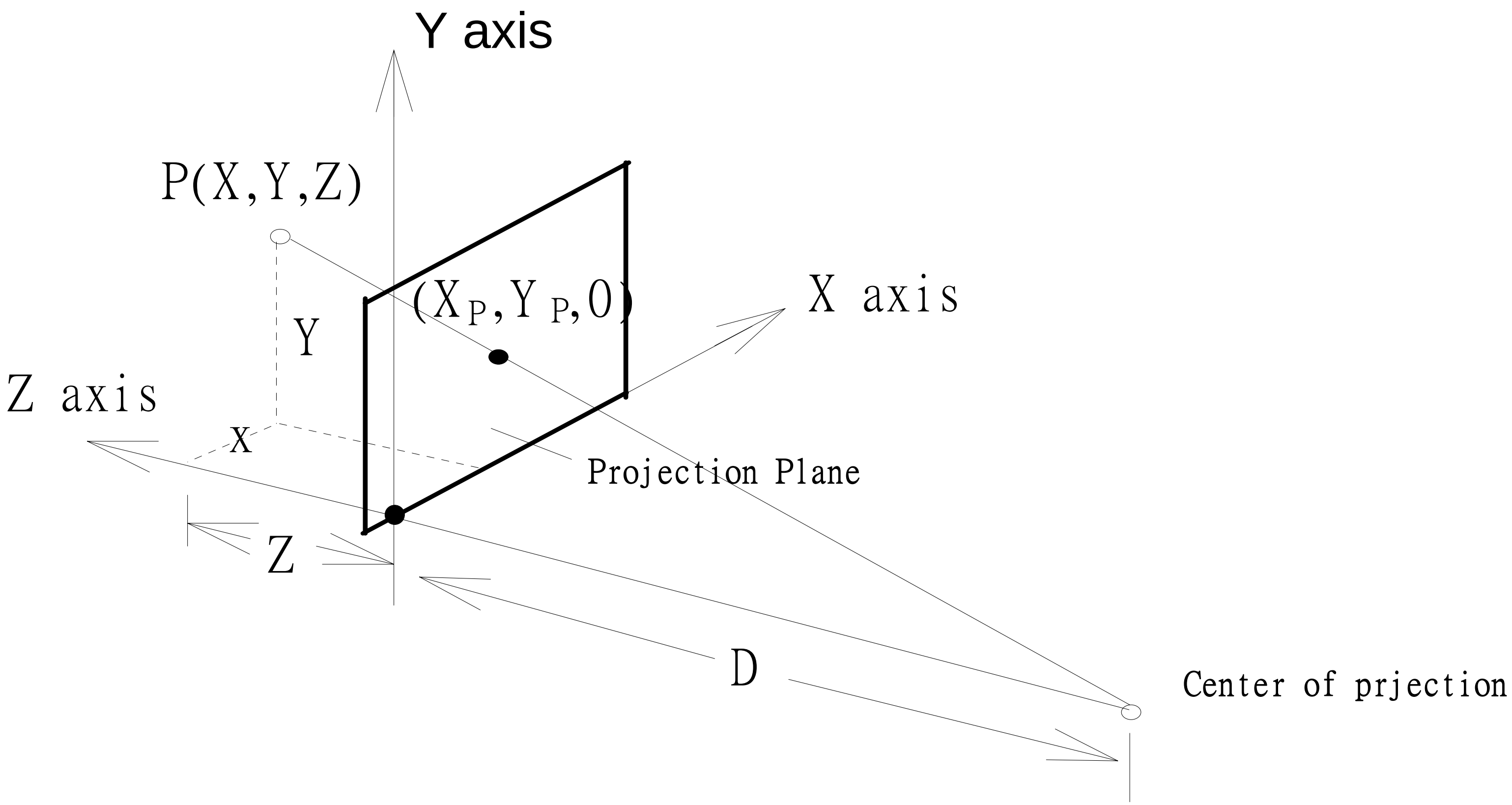
and let $w=1$

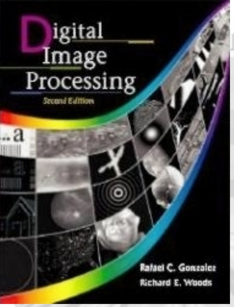
$$x = x'/w' = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}} \quad y = y'/w' = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}}$$

If $a_{13} = a_{23} = 0$ then it becomes Affine Transformation

5.11 Geometric Transformation-
Spatial transformation

- Perspective projection





5.11 Geometric Transformation- *Spatial transformation*

- Any position along the projection line is (α, β, γ) is

$$\alpha = X - X\delta$$

$$\beta = Y - Y\delta$$

$$\gamma = Z - (Z + D)\delta$$

where $\delta = Z/(Z+D)$, and $0 < \delta < 1$

$$X_p = X \cdot \left(\frac{1}{\frac{Z}{D} + 1}\right) \qquad Y_p = Y \cdot \left(\frac{1}{\frac{Z}{D} + 1}\right)$$

5.11 Geometric Transformation-
Spatial transformation

- Perspective Transformation

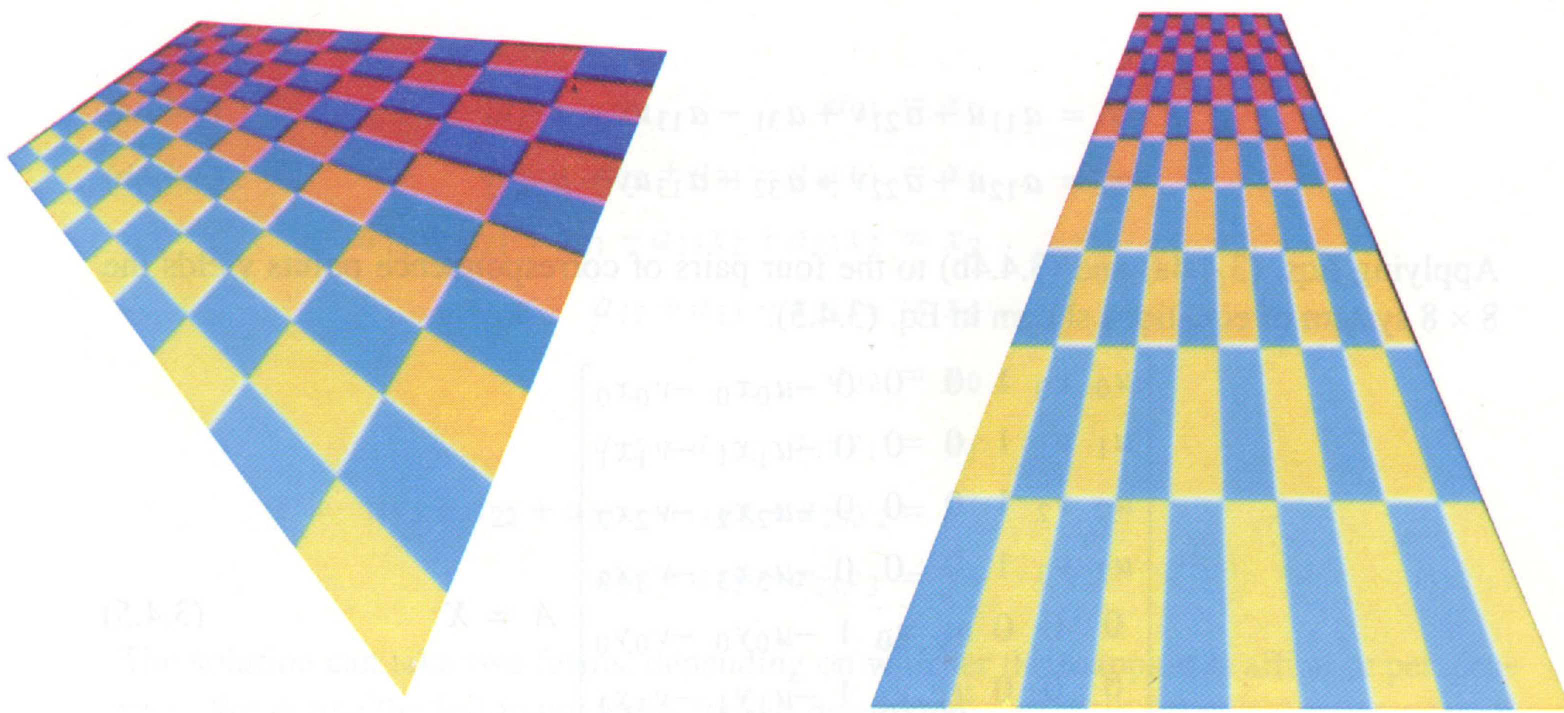


Figure 3.6: Perspective warps.

5.11 Geometric Transformation- *Spatial transformation*

- ***Bilinear transformation*** is defined as

$$T : W \rightarrow Z \qquad Z = W \cdot T$$

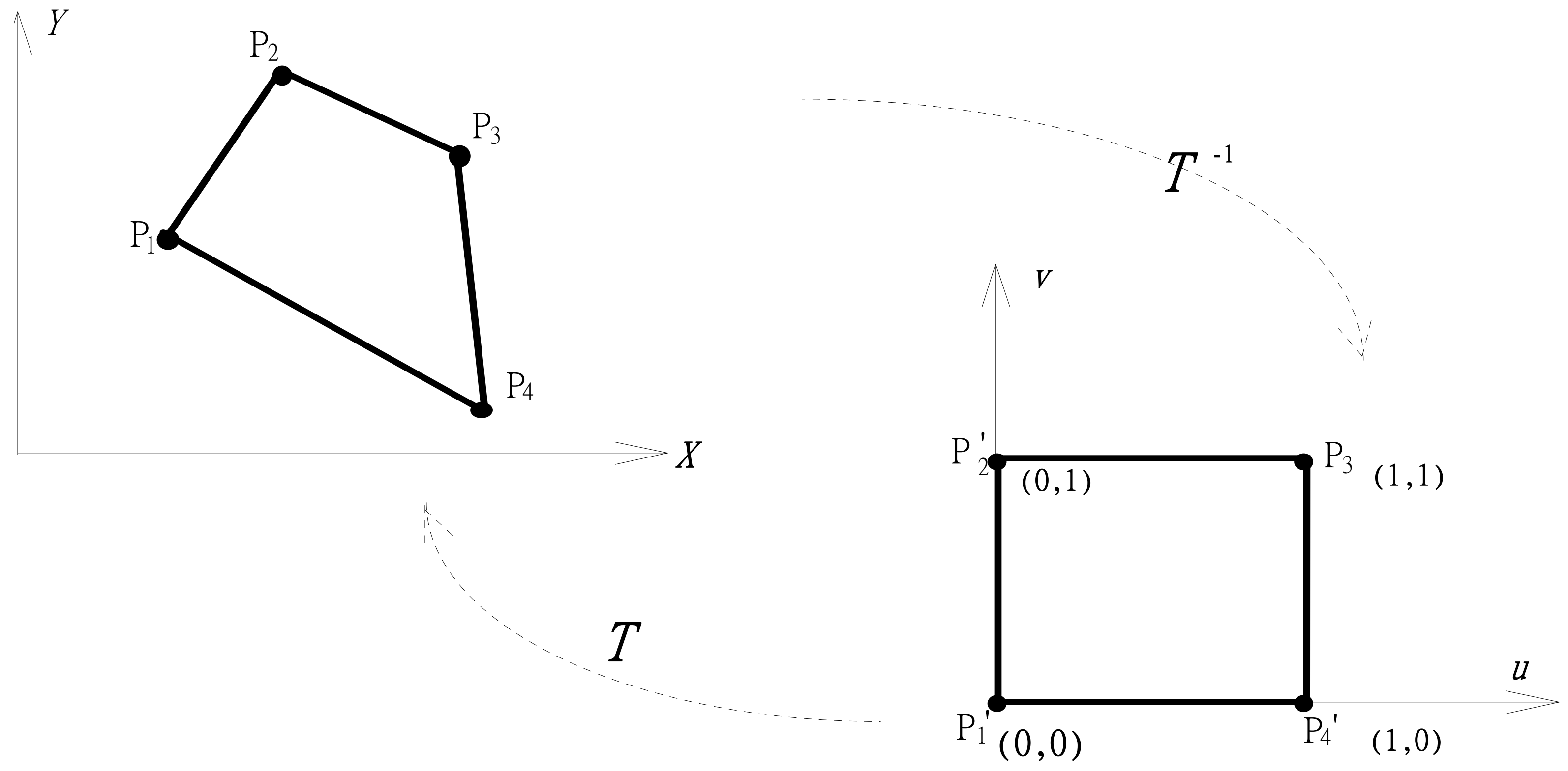
where W=[*u*, *v*, 1] and Z=[*x*, *y*, 1]

$$T = \begin{bmatrix} a_1 & b_1 + b_3 v & 0 \\ a_2 + a_3 u & b_2 & 0 \\ a_0 & b_0 & 1 \end{bmatrix}$$

and $a_0, a_1, a_2, a_3, b_0, b_1, b_2$, and b_3 are constant.

5.11 Geometric Transformation-
Spatial transformation

Bilinear distortion for reference quadrangle and points.



5.11 Geometric Transformation- *gray-level interpolation*

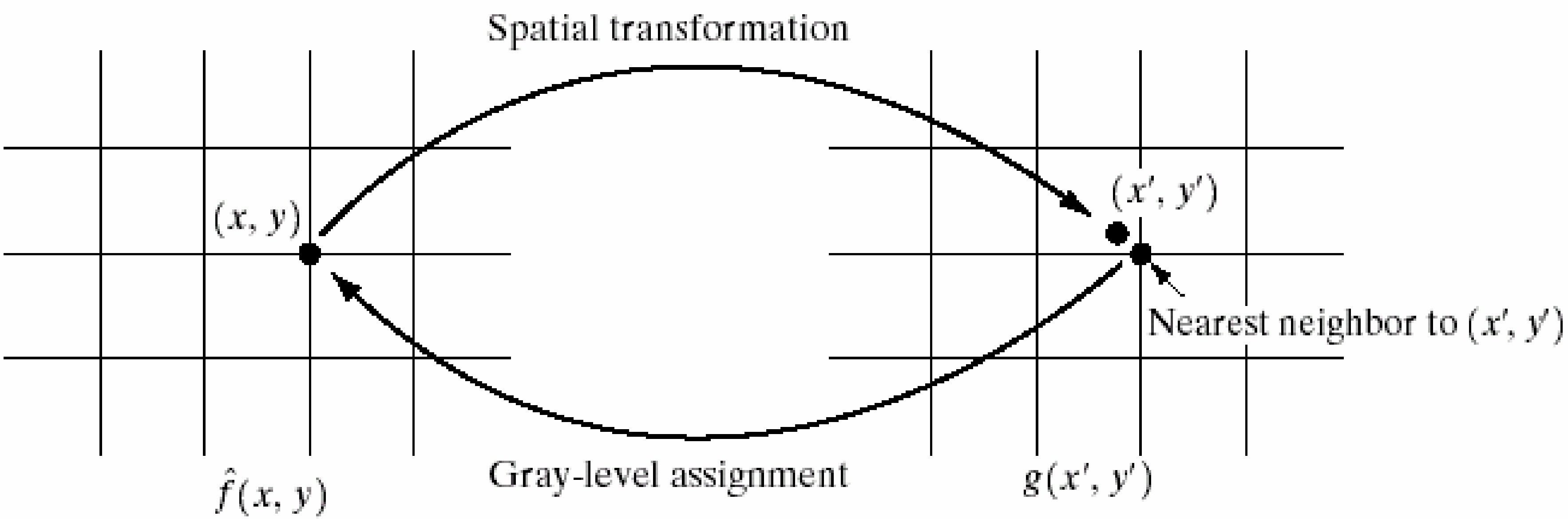
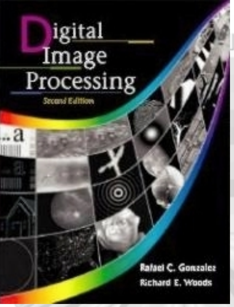


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.



5.11 Geometric Transformation- *gray-level interpolation*

- **Zero-order interpolation**: the simplest one.
- **Bilinear interpolation**: using four neighbors to do the interpolation for the gray level of non-integer point at (x', y'):

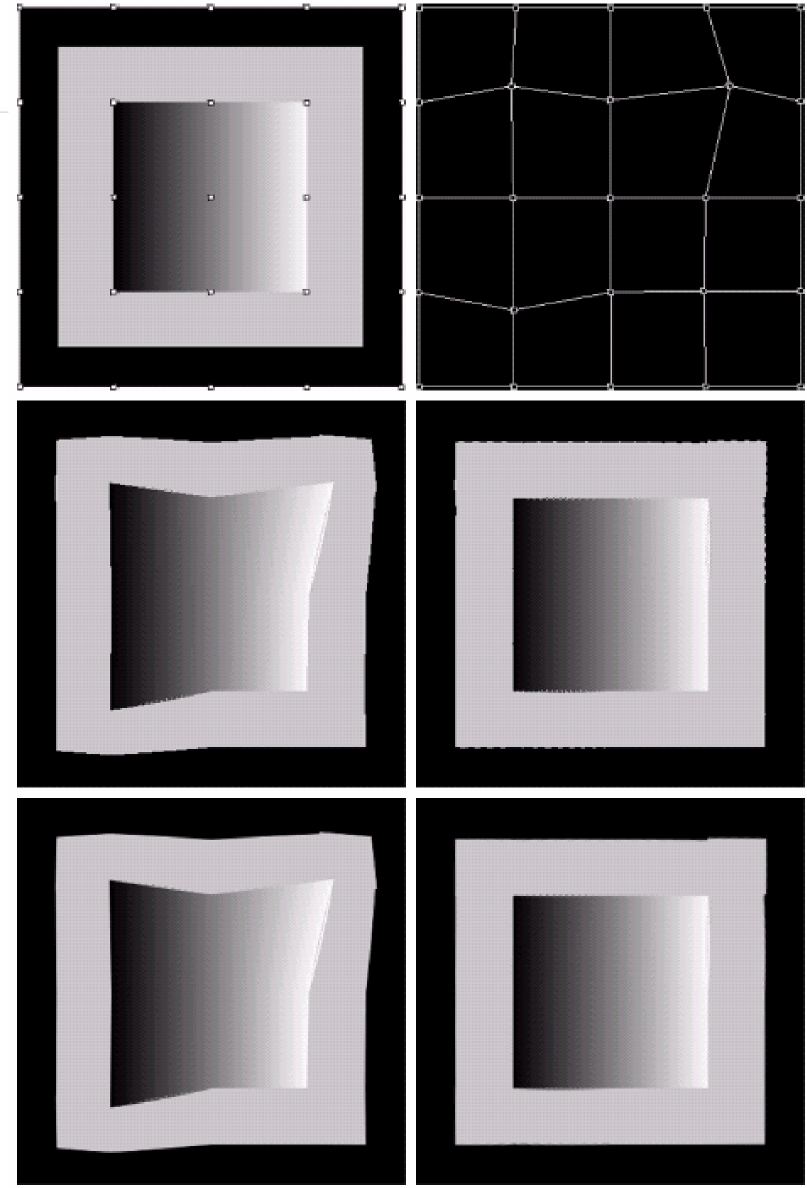
$$v(x', y') = a + bx' + cy' + dx'y'$$

- The four neighbors located at (x_i, y_i), with known gray level v_{ii} $i=1$ and 2, then we have four different equations as

$$v_{ii} = a + bx_i + cy_i + d\, x_i\, y_i$$

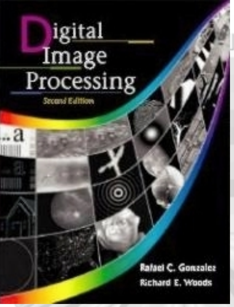
- Solving the above four equations for the four parameters, *a*, *b*, *c*, and *d*.

5.11 Geometric Transformation



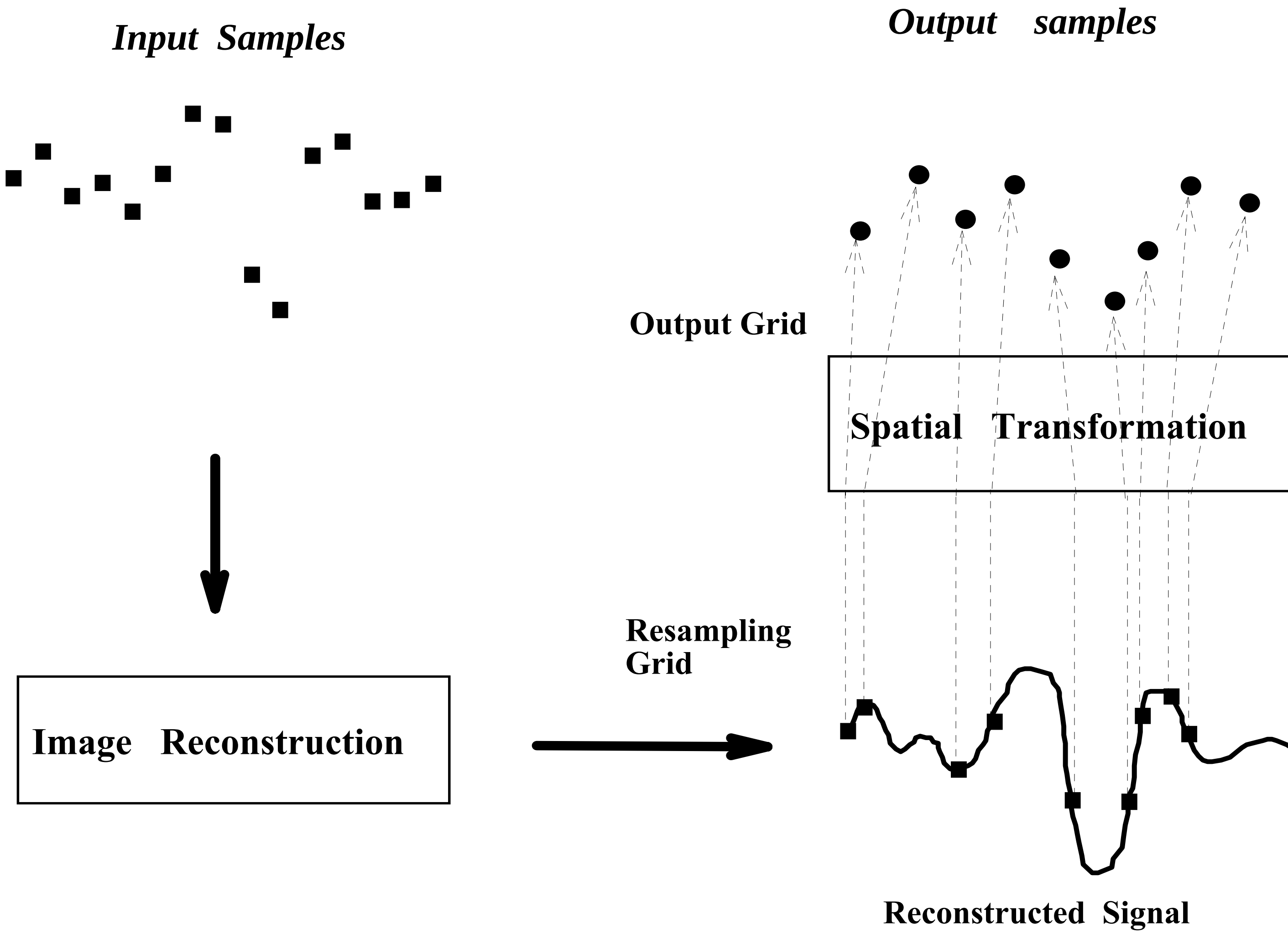
a b
c d
e f

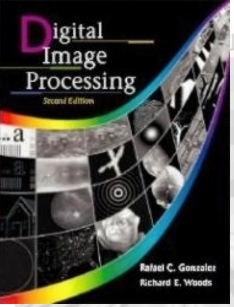
FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.



5.11 Geometric Transformation- *gray-level interpolation*

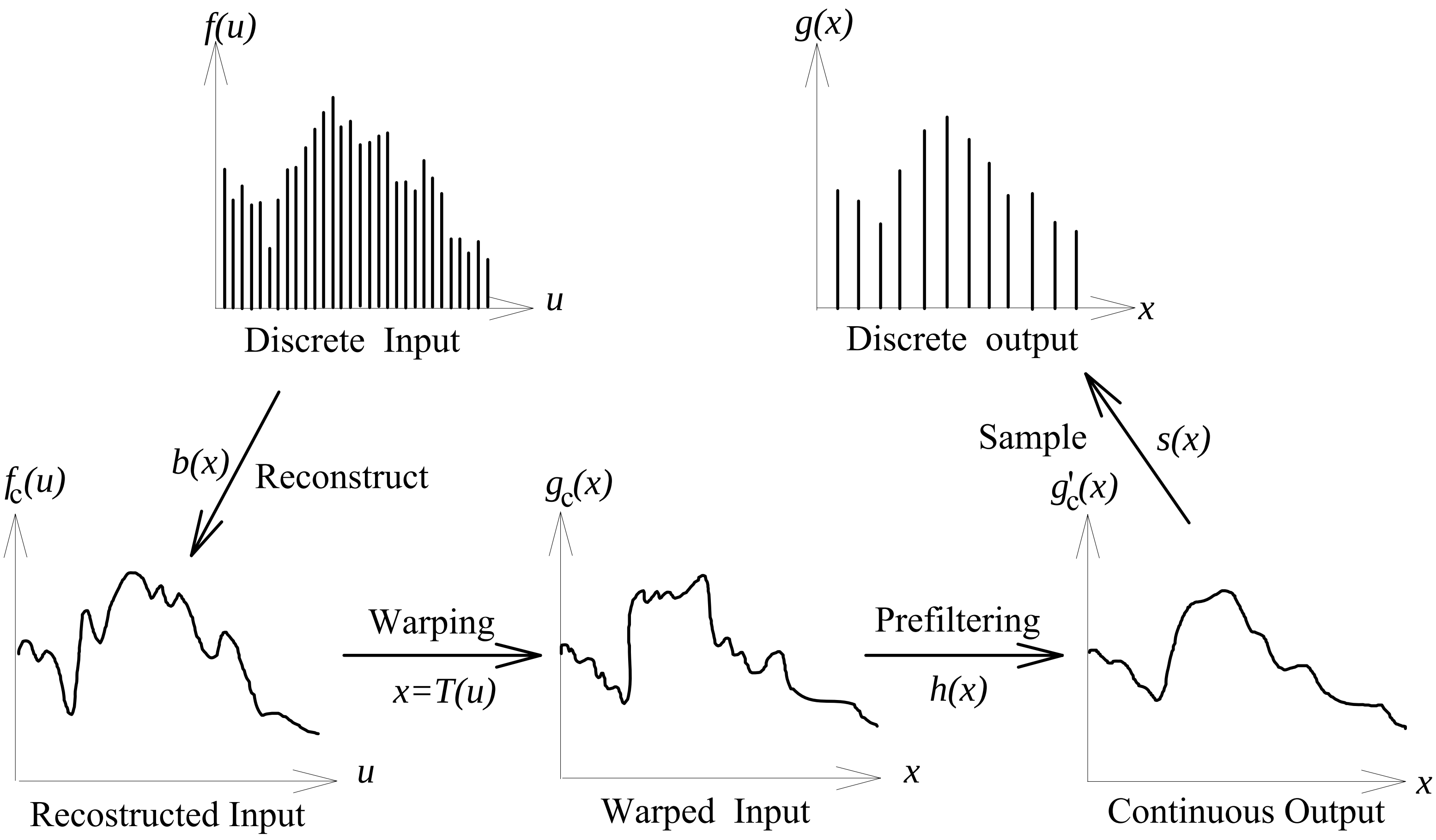
- resampling





5.11 Geometric Transformation- *gray-level interpolation*

- *resampling*



5.11 Geometric Transformation- *gray-level interpolation*

Ideal warping resampling

$$g(x) = g_c'(x) = g_c(x) * h(x) \qquad \text{for } x \in Z$$

$$= \int f_c(T^{-1}(t))h(x-t)dt$$

$$= \int \left[\sum_{j \in Z} f(j) b(T^{-1}(t) - j) \right] h(x-t)dt$$

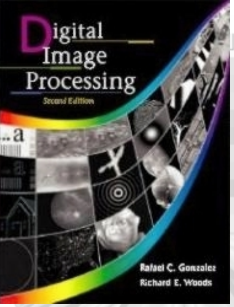
$$= \sum_{j \in Z} f(j) \psi(x, j)$$

where $\psi(x, j) = \int b(T^{-1}(t) - j)h(x-t)dt$

5.11 Geometric Transformation- *gray-level interpolation*

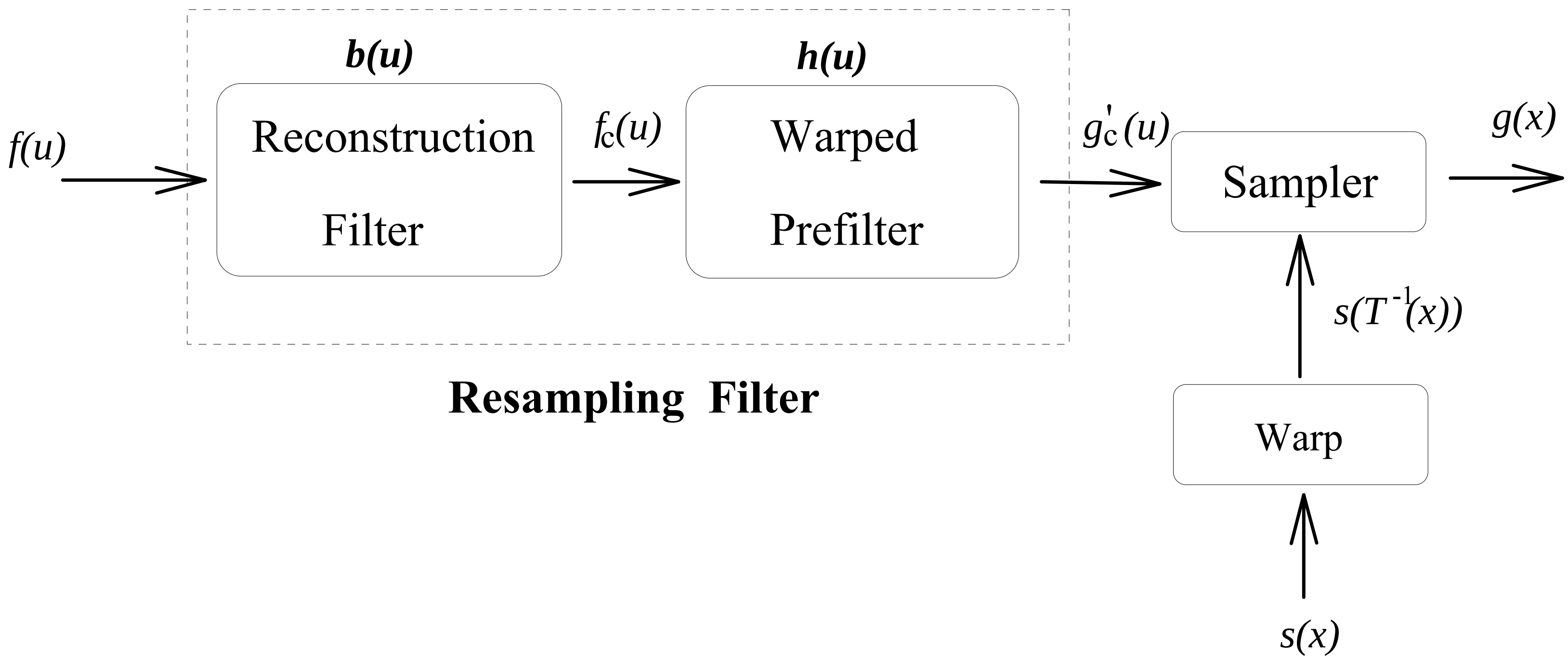
Table : Elements of ideal resampling.

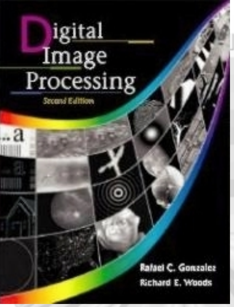
Stage	Mathematical definition
Discrete Input	$f(u), \qquad u \in Z$
Reconstructed Input	$f_c(u) = f(u) * b(u) = \sum_{j \in Z} f(j) b(u-j)$
Warped Signal	$g_c(x) = f_c(T^{-1}(x))$
Continuous Output	$g'_c(x) = g_c(x) * h(x) = \int g_c(t) h(x-t) dt$
Discrete Output	$g(x) = g'_c(x) s(x)$



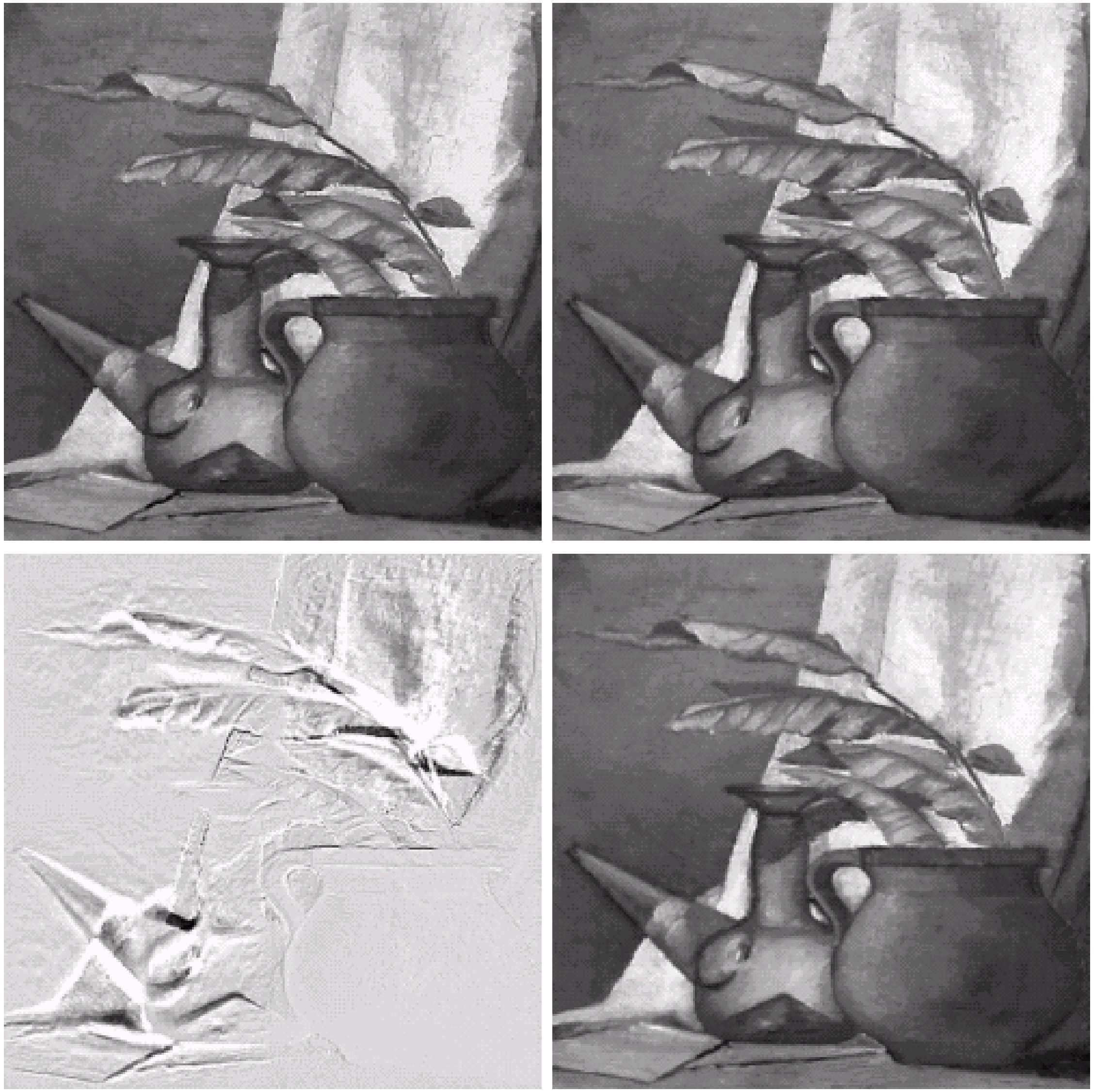
5.11 Geometric Transformation- *gray-level interpolation*

- ***Cubic convolution interpolation***:using much larger number of neighbors (*i.e*., 16) to obtain a smoother interpolation.
- ***Ideal Interpolation***:





5.11 Geometric Transformation



a b
c d

FIGURE 5.35 (a) An image before geometric distortion. (b) Image geometrically distorted using the same parameters as in Fig. 5.34(e). (c) Difference between (a) and (b). (d) Geometrically restored image.