Image Restoration

Image restoration attempts to recover the original image from a degraded version of the original image by using a model for the degradation.

Image Restoration

Image restoration attempts to recover the original image from a degraded version of the original image by using a model for the degradation.

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

where $g(x,y)$ is the degraded image
$h(x,y)$ is a degradation function
$f(x,y)$ is the original input image
$n(x,y)$ is additive noise

Image Restoration

Image restoration attempts to recover the original image from a degraded version of the original image by using a model for the degradation.

space $$g(x,y) = h(x,y) * f(x,y) + n(x,y)$$

where $g(x,y)$ is the degraded image

$h(x,y)$ is a degradation function

$f(x,y)$ is the original input image

$n(x,y)$ is additive noise

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

Noise Models

Noise properties can depend on pixel location as well as the image gray level $f(x,y)$.

Noise Models

Noise properties can depend on pixel location as well as the image gray level $f(x,y)$.

A common assumption is that the noise is independent and identically distributed (i.i.d.) meaning that the noise at any pixel is independent of the noise at any other pixel and the additive noise process at every pixel has the same probability density function (pdf)

Gaussian noise

pdf $\quad p(z) = \dfrac{1}{\sqrt{2\pi}\,\sigma} e^{-(z-\mu)^2/2\sigma^2}$

μ = mean

σ^2 = variance

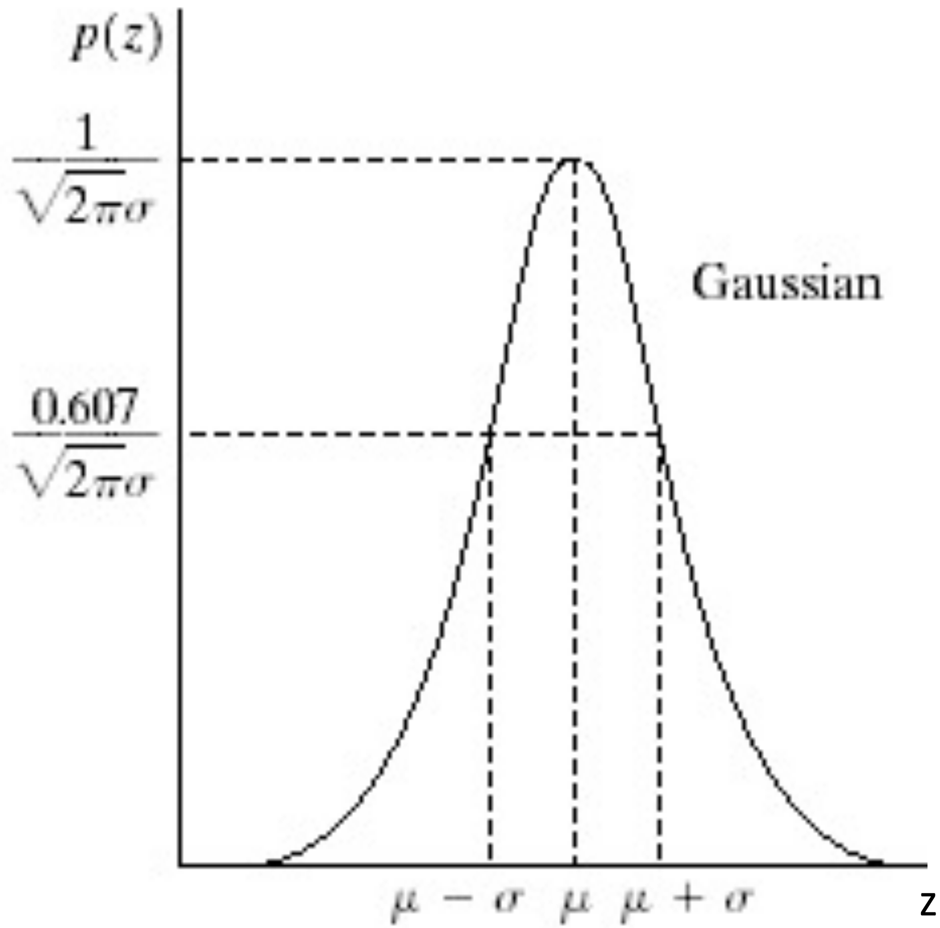


FIGURE 5.2

Rayleigh noise

pdf

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

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

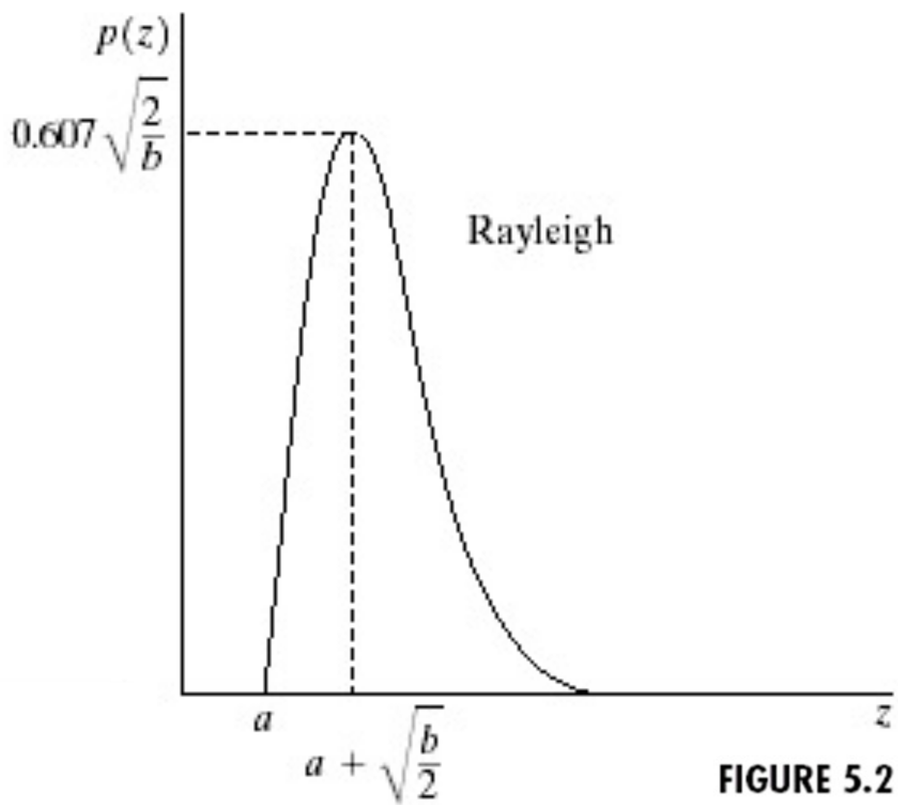


FIGURE 5.2

Gamma noise

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

$a > 0$, b is positive integer

$\mu = \dfrac{b}{a}$ $\sigma^2 = \dfrac{b}{a^2}$

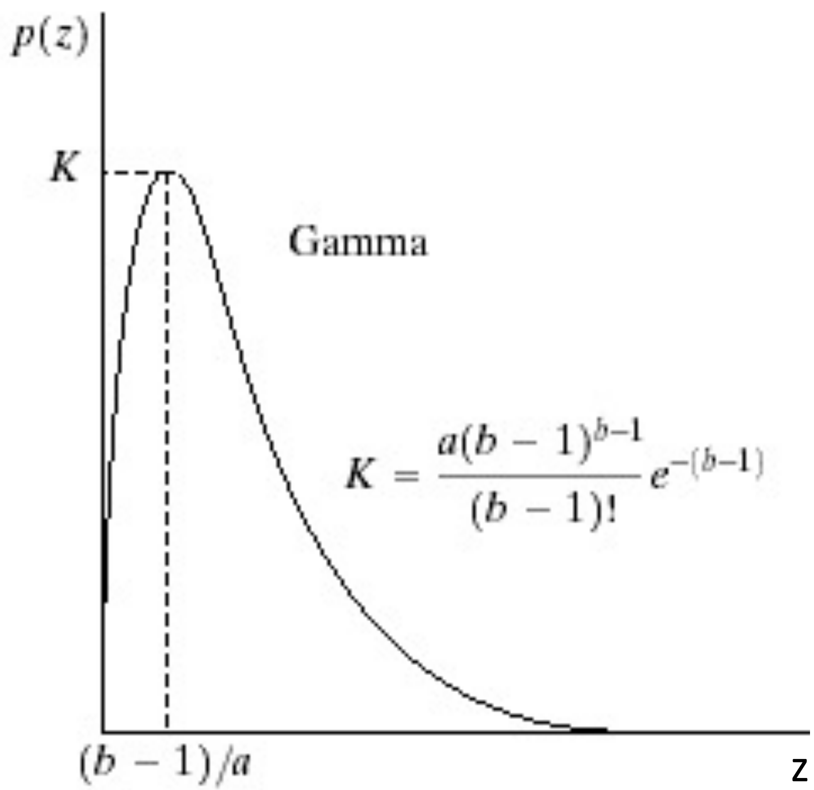


FIGURE 5.2

If b=1, then gamma pdf is exponential pdf

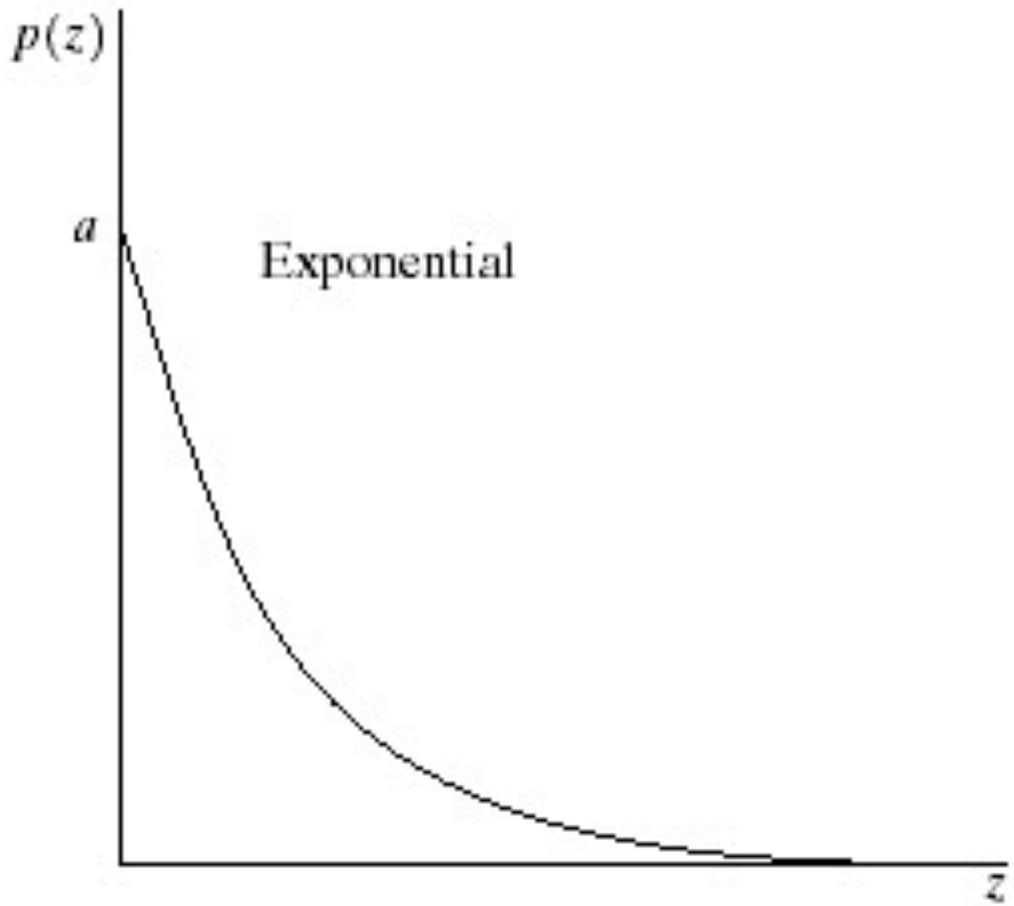


FIGURE 5.2

Uniform noise

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

$$\mu = \frac{a+b}{2} \qquad \sigma^2 = \frac{(b-a)^2}{12}$$

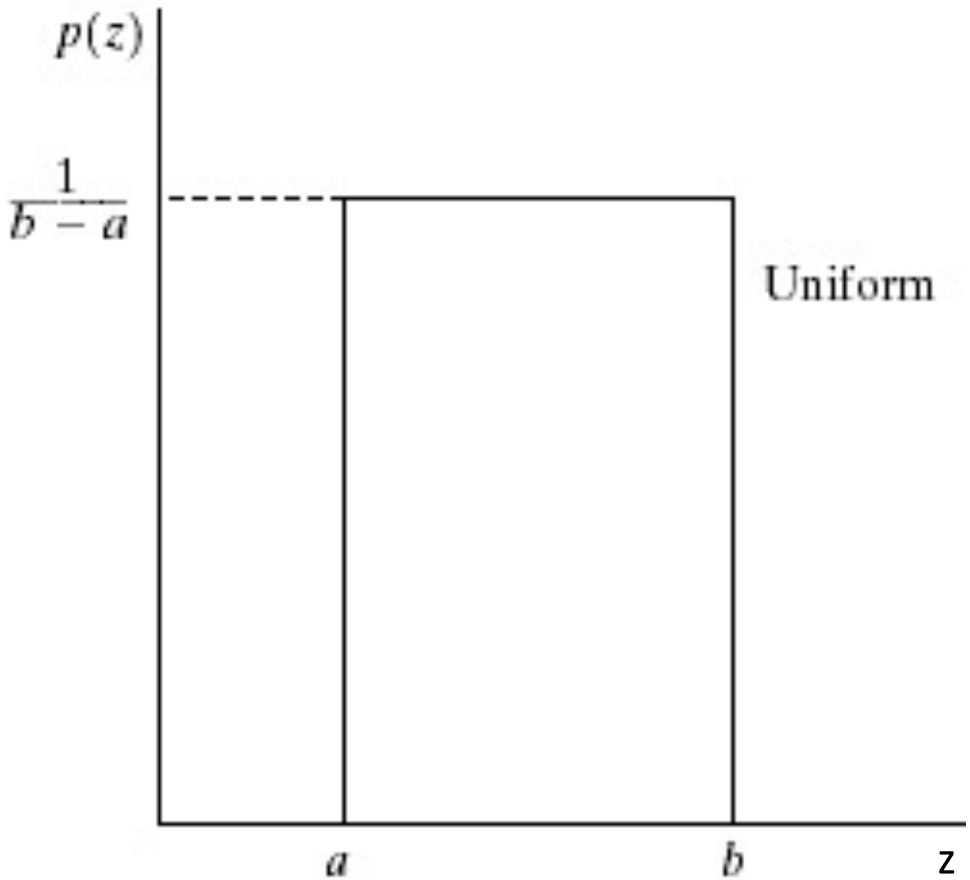


FIGURE 5.2

Impulse (salt-and-pepper) noise

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

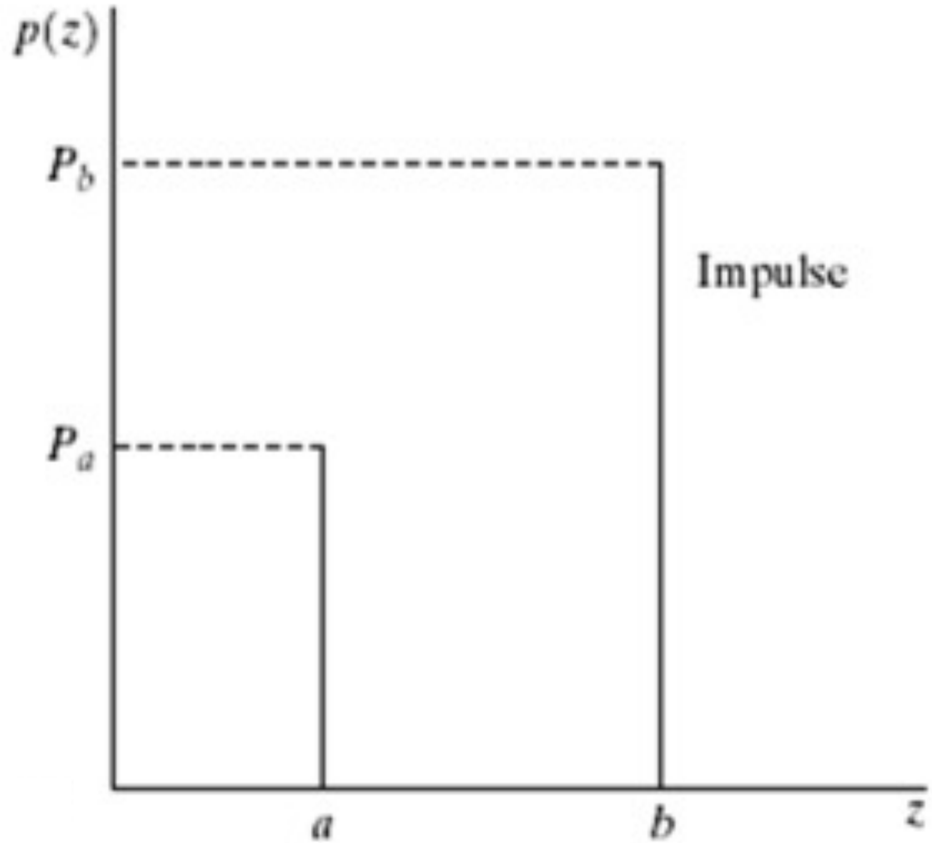


FIGURE 5.2

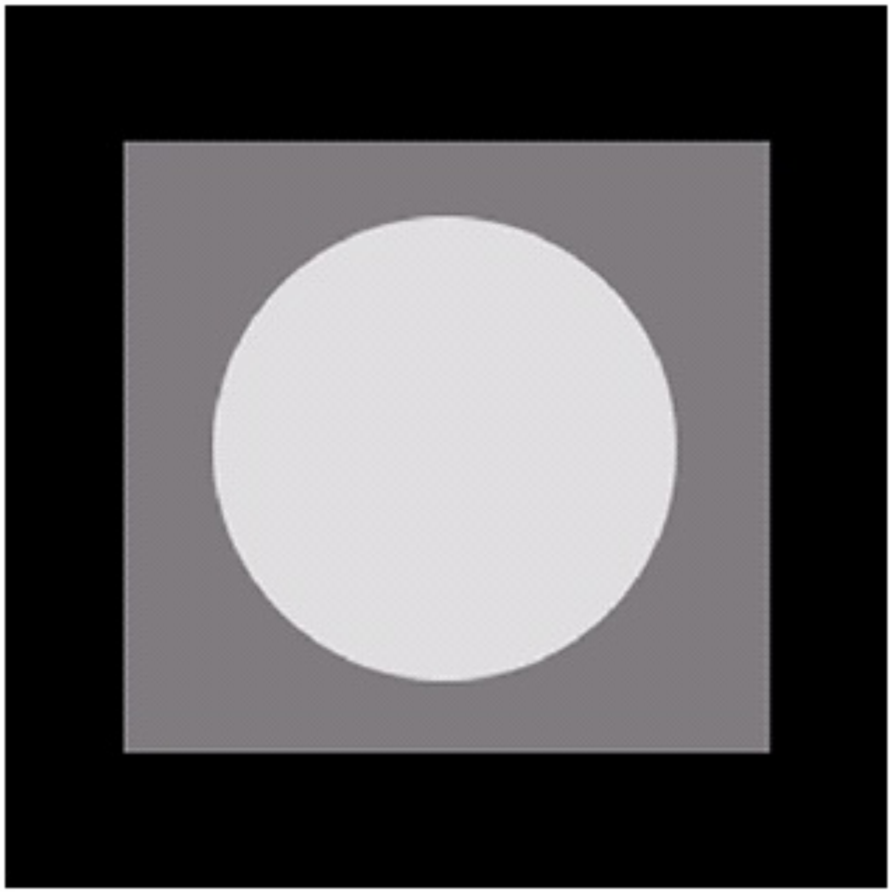


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

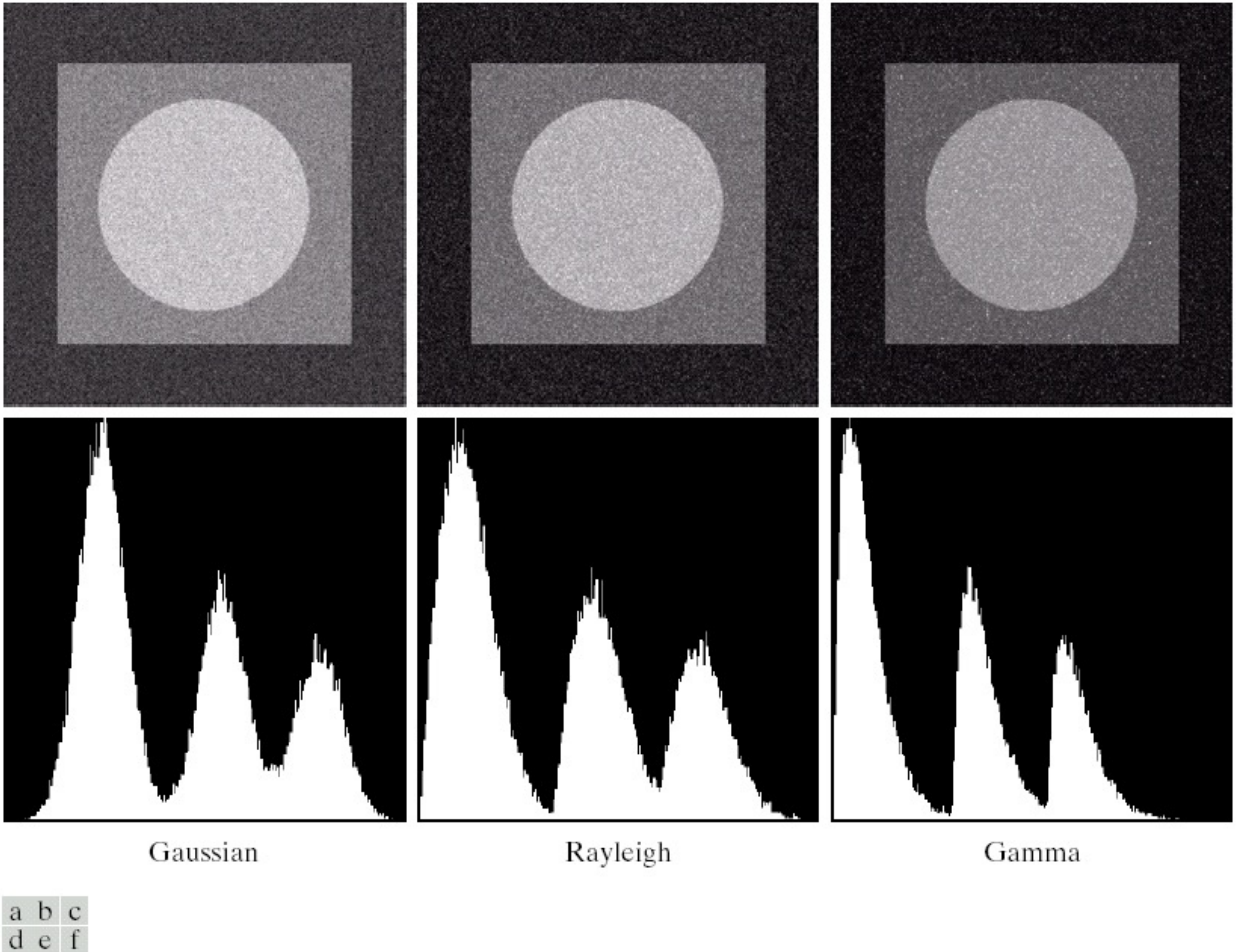


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

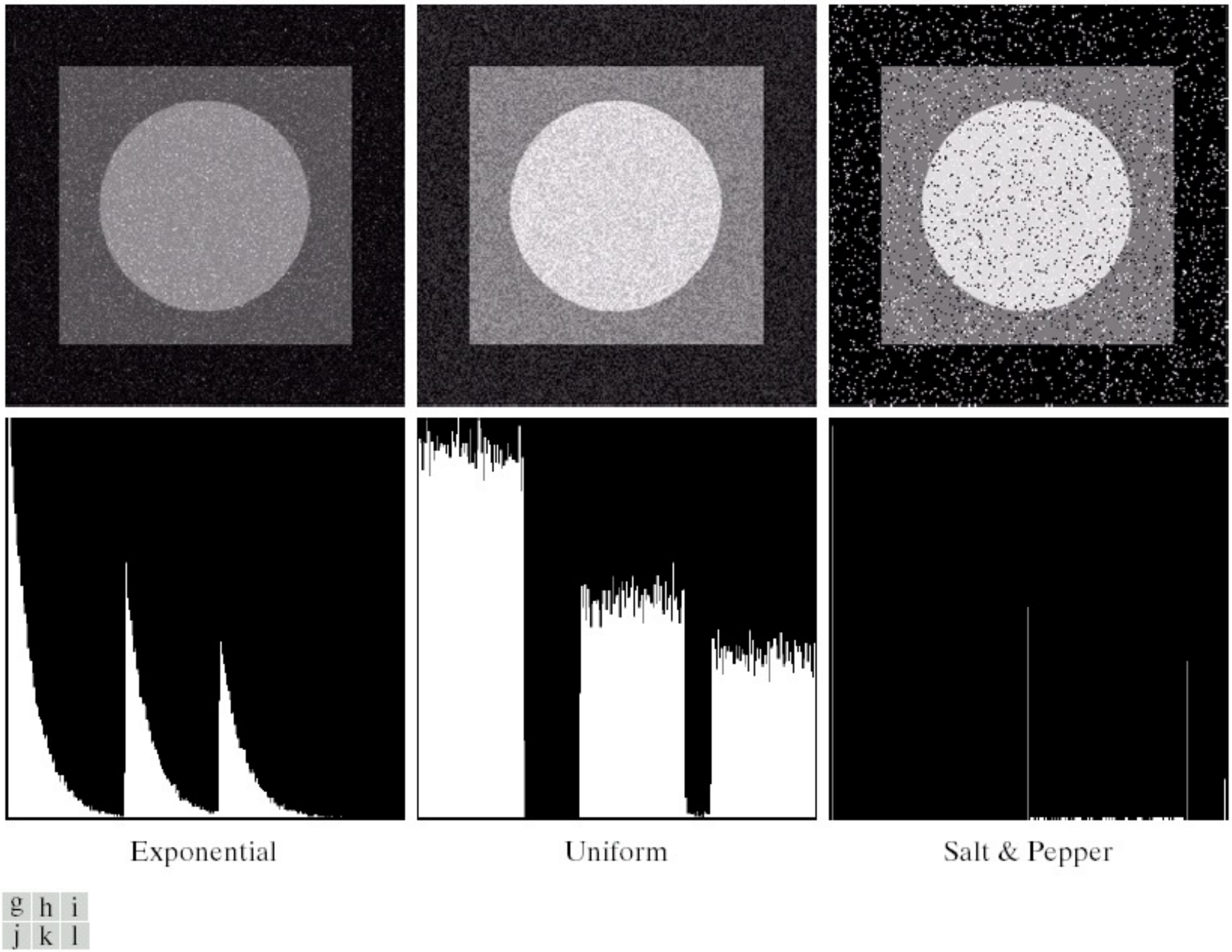


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

Periodic noise

The power in periodic noise is concentrated at specific spatial frequencies

a
b

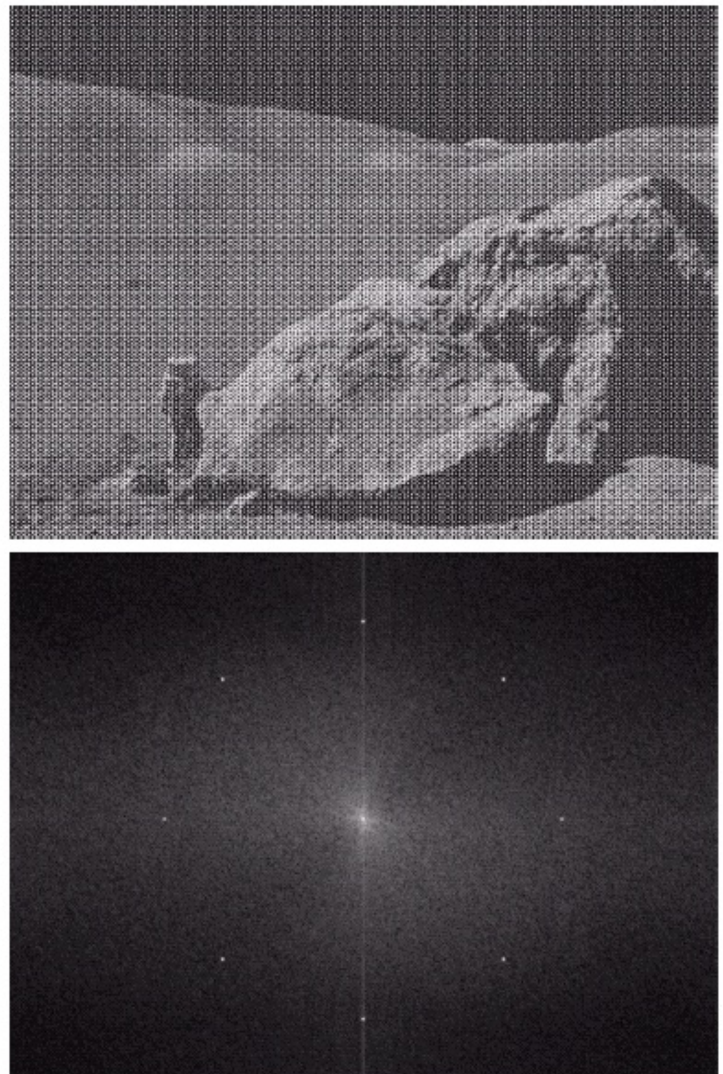
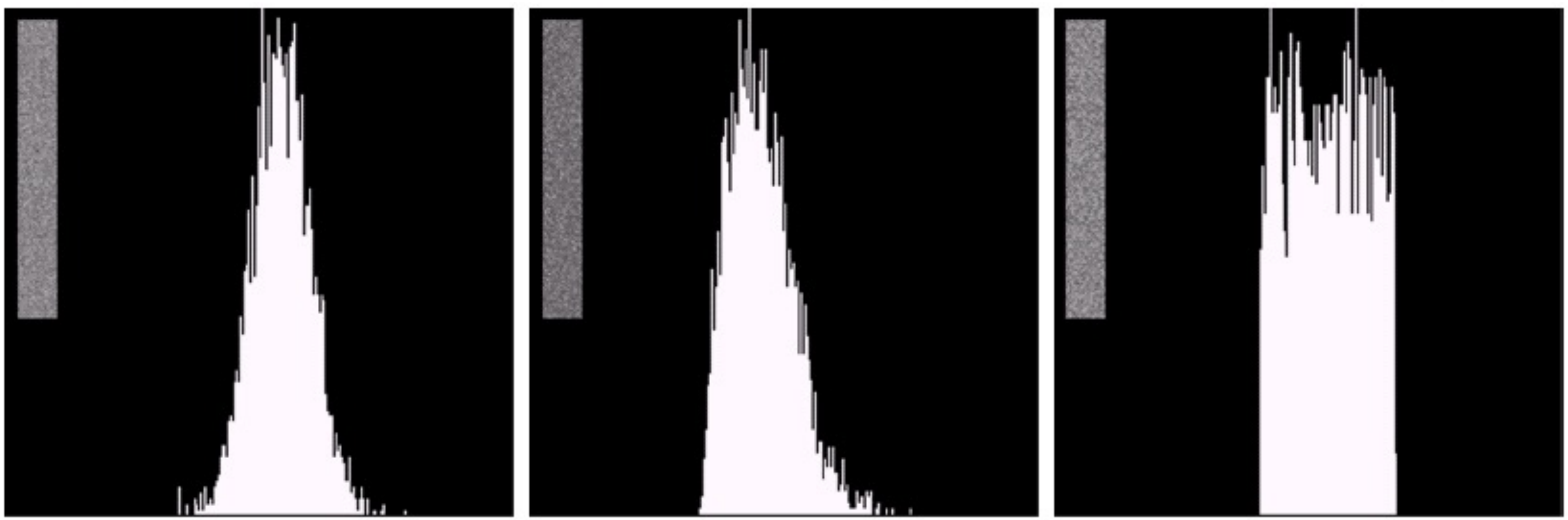


FIGURE 5.5
(a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Estimation of Noise Parameters

$$g(x,y) = f(x,y) + n(x,y)$$

If a uniform scene $(f(x,y) = \text{constant})$ can be constructed, then noise parameters can be estimated from the image $g(x,y)$ because the image histogram will resemble a mean shifted version of the noise pdf.



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.

Noise is often assumed to be zero-mean when estimating noise parameters since it can be difficult to separate the mean of the noise from the mean of $f(x,y)$

Noise is often assumed to be zero-mean when estimating noise parameters since it can be difficult to separate the mean of the noise from the mean of $f(x,y)$

Estimating Noise Variance

$$g(x,y) = f(x,y) + n(x,y)$$

If $f(x,y)$ is a constant then $VAR[g(x,y)] = VAR[n(x,y)]$

Noise is often assumed to be zero-mean when estimating noise parameters since it can be difficult to separate the mean of the noise from the mean of $f(x,y)$

Estimating Noise Variance

$$g(x,y) = f(x,y) + n(x,y)$$

If $f(x,y)$ is a constant then $VAR[g(x,y)] = VAR[n(x,y)]$

Estimate of $VAR[g(x,y)]$ (sample variance) $= \dfrac{1}{N} \sum_{(x,y) \in S} (g(x,y) - \mu)^2$

where $\mu = \dfrac{1}{N} \sum_{(x,y) \in S} g(x,y)$ and S is a region of N pixels in $g(x,y)$

If noise properties do not depend on the pixel gray levels, then we can use a dark image to create a constant $f(x,y)$. Otherwise it can be difficult to construct a scene with illumination and surface variation that is significantly smaller than the variation due to noise.

If noise properties do not depend on the pixel gray levels, then we can use a dark image to create a constant $f(x,y)$. Otherwise it can be difficult to construct a scene with illumination and surface variation that is significantly smaller than the variation due to noise.

How can we estimate noise variance if the scene has pixel to pixel variation that is comparable to the noise level?

Take two images of the same stationary scene

$$g_1(x,y) = f(x,y) + n_1(x,y)$$
$$g_2(x,y) = f(x,y) + n_2(x,y)$$

Take two images of the same stationary scene

$$g_1(x,y) = f(x,y) + n_1(x,y)$$
$$g_2(x,y) = f(x,y) + n_2(x,y)$$

$$\Delta g(x,y) = g_1(x,y) - g_2(x,y) = n_1(x,y) - n_2(x,y)$$

Take two images of the same stationary scene

$$g_1(x,y) = f(x,y) + n_1(x,y)$$
$$g_2(x,y) = f(x,y) + n_2(x,y)$$

$$\Delta g(x,y) = g_1(x,y) - g_2(x,y) = n_1(x,y) - n_2(x,y)$$

If $n_1(x,y)$ and $n_2(x,y)$ are independent with the same variance σ^2, then

$$VAR\left[n_1(x,y) - n_2(x,y)\right] = 2\sigma^2$$

Take two images of the same stationary scene

$$g_1(x,y) = f(x,y) + n_1(x,y)$$
$$g_2(x,y) = f(x,y) + n_2(x,y)$$

$$\Delta g(x,y) = g_1(x,y) - g_2(x,y) = n_1(x,y) - n_2(x,y)$$

If $n_1(x,y)$ and $n_2(x,y)$ are independent with the same variance σ^2, then

$$VAR\left[n_1(x,y) - n_2(x,y)\right] = 2\sigma^2$$

If we estimate the variance of the $\Delta g(x,y)$ image over a region S, we get an estimate of $2\sigma^2$ that is not affected by spatial variation in $f(x,y)$.

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Noise is zero-mean and n_1 and n_2 have sample variance of 0.5. Let σ_n^2 be noise variance.

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Noise is zero-mean and n_1 and n_2 have sample variance of 0.5. Let σ_n^2 be noise variance.

$$g_1(x,y) = f(x,y) + n_1(x,y) = \begin{matrix} 20 & 24 \\ 20 & 24 \end{matrix}$$

$$g_2(x,y) = f(x,y) + n_2(x,y) = \begin{matrix} 19 & 23 \\ 21 & 25 \end{matrix}$$

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $\quad n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $\quad n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Noise is zero-mean and n_1 and n_2 have sample variance of 0.5. Let σ_n^2 be noise variance.

$$g_1(x,y) = f(x,y) + n_1(x,y) = \begin{matrix} 20 & 24 \\ 20 & 24 \end{matrix}$$

$$g_2(x,y) = f(x,y) + n_2(x,y) = \begin{matrix} 19 & 23 \\ 21 & 25 \end{matrix}$$

$$\Delta g(x,y) = \begin{matrix} 1 & 1 \\ -1 & -1 \end{matrix}$$

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Noise is zero-mean and n_1 and n_2 have sample variance of 0.5. Let σ_n^2 be noise variance.

$$g_1(x,y) = f(x,y) + n_1(x,y) = \begin{matrix} 20 & 24 \\ 20 & 24 \end{matrix}$$

$$g_2(x,y) = f(x,y) + n_2(x,y) = \begin{matrix} 19 & 23 \\ 21 & 25 \end{matrix}$$

$$\Delta g(x,y) = \begin{matrix} 1 & 1 \\ -1 & -1 \end{matrix}$$

$$\sigma_{\Delta g}^2 = 1 \approx 2\sigma_n^2 \longrightarrow \sigma_n^2 \approx 0.5$$

Ex $f(x,y) = \begin{matrix} 20 & 23 \\ 21 & 24 \end{matrix}$ $n_1(x,y) = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ $n_2(x,y) = \begin{matrix} -1 & 0 \\ 0 & 1 \end{matrix}$

Noise is zero-mean and n_1 and n_2 have sample variance of 0.5. Let σ_n^2 be noise variance.

$$g_1(x,y) = f(x,y) + n_1(x,y) = \begin{matrix} 20 & 24 \\ 20 & 24 \end{matrix}$$

$$g_2(x,y) = f(x,y) + n_2(x,y) = \begin{matrix} 19 & 23 \\ 21 & 25 \end{matrix}$$

$$\Delta g(x,y) = \begin{matrix} 1 & 1 \\ -1 & -1 \end{matrix}$$

$$\sigma_{\Delta g}^2 = 1 \approx 2\sigma_n^2 \longrightarrow \sigma_n^2 \approx 0.5$$

sample means and variances

$\mu_f = 22$ $\sigma_f^2 = 2.5$

$\mu_{g_1} = 22$ $\sigma_{g_1}^2 = 4$

$\mu_{g_2} = 22$ $\sigma_{g_2}^2 = 5$

Example summary: variance of g $(\sigma_{g_1}^2, \sigma_{g_2}^2)$ overestimates σ_n^2 because variance of g is influenced by variance of f. $\sigma_{\Delta g}^2/2$ is a better estimate of σ_n^2 because it removes the effect of variation in f.

Spatial Filtering for Noise Removal

Consider only noise degradation $g(x,y) = f(x,y) + n(x,y)$

Spatial Filtering for Noise Removal

Consider only noise degradation $g(x,y) = f(x,y) + n(x,y)$

Arithmetic Mean Filter (AMF)

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

S_{xy} is image window centered at (x,y) with m rows and n columns.

Geometric Mean Filter (GMF)

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

Nonlinear

Geometric Mean Filter (GMF)

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

GMF smooths and tends to darken more than the AMF in regions with gray level variability

Geometric Mean Filter (GMF)

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

Nonlinear

GMF smooths and tends to darken more than the AMF in regions with gray level variability

Ex

pixel gray levels		arithmetic mean	geometric mean
I	I	I	I
$I-1$	$I+1$	I	$\sqrt{I^2-1}$
$I-2$	$I+2$	I	$\sqrt{I^2-4}$

Ex Salt noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	190	100	100
100	100	100	100	100
100	100	100	100	100

Ex Salt noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	190	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	107	107	107	100
100	107	107	107	100
100	107	107	107	100
100	100	100	100	100

Ex Salt noise reduction using 3x3 GMF

Input						GMF Output				
100	100	100	100	100		100	100	100	100	100
100	100	100	100	100		100	107	107	107	100
100	100	190	100	100		100	107	107	107	100
100	100	100	100	100		100	107	107	107	100
100	100	100	100	100		100	100	100	100	100

For any window containing 190, we get $(100^8 \cdot 190)^{1/9} = 107$

AMF gives 110.

(Ex) Salt noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	190	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	107	107	107	100
100	107	107	107	100
100	107	107	107	100
100	100	100	100	100

For any window containing 190, we get $(100^8 \cdot 190)^{1/9} = 107$

AMF gives 110.

(Ex) Pepper noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	10	100	100
100	100	100	100	100
100	100	100	100	100

Ex Salt noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	190	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	107	107	107	100
100	107	107	107	100
100	107	107	107	100
100	100	100	100	100

For any window containing 190, we get $(100^8 \cdot 190)^{1/9} = 107$

AMF gives 110.

Ex Pepper noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	10	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	77	77	77	100
100	77	77	77	100
100	77	77	77	100
100	100	100	100	100

Ex Salt noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	190	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	107	107	107	100
100	107	107	107	100
100	107	107	107	100
100	100	100	100	100

For any window containing 190, we get $(100^8 \cdot 190)^{1/9} = 107$

AMF gives 110.

Ex Pepper noise reduction using 3x3 GMF

Input

100	100	100	100	100
100	100	100	100	100
100	100	10	100	100
100	100	100	100	100
100	100	100	100	100

GMF Output

100	100	100	100	100
100	77	77	77	100
100	77	77	77	100
100	77	77	77	100
100	100	100	100	100

For any window containing 10, we get $(100^8 \cdot 10)^{1/9} = 77$

AMF gives 90.

Ex Edge blur due to 3x3 GMF

Input

100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

Ex Edge blur due to 3x3 GMF

Input

100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

GMF Output

100	126	159	200
100	126	159	200
100	126	159	200
100	126	159	200

$$(100^6 200^3)^{1/9} = 126$$

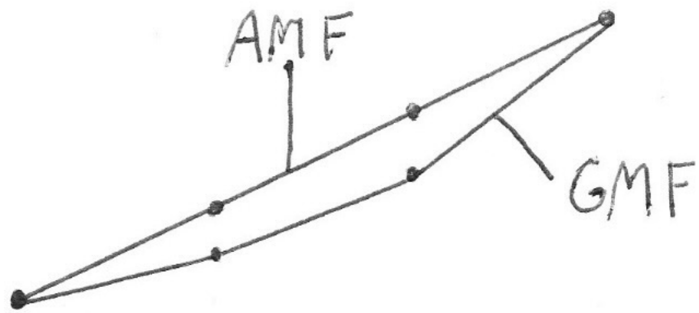
$$(100^3 200^6)^{1/9} = 159$$

AMF gives 100 133 167 200

Ex Edge blur due to 3x3 GMF

Input			
100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

GMF Output			
100	126	159	200
100	126	159	200
100	126	159	200
100	126	159	200



$$(100^6 200^3)^{1/9} = 126$$

$$(100^3 200^6)^{1/9} = 159$$

AMF gives 100 133 167 200

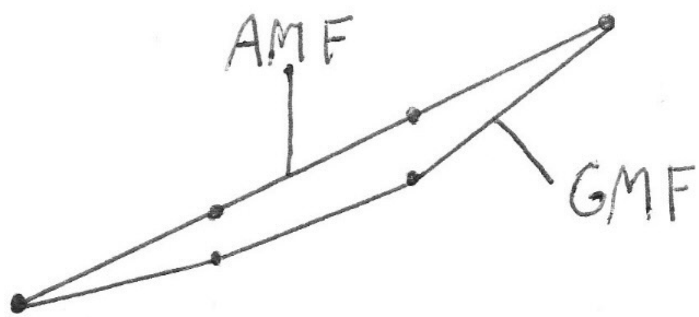
Ⓔⓧ Edge blur due to 3x3 GMF

Input

100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

GMF Output

100	126	159	200
100	126	159	200
100	126	159	200
100	126	159	200



$$(100^6 200^3)^{1/9} = 126$$

$$(100^3 200^6)^{1/9} = 159$$

AMF gives 100 133 167 200

GMF gives sharper edge than AMF.

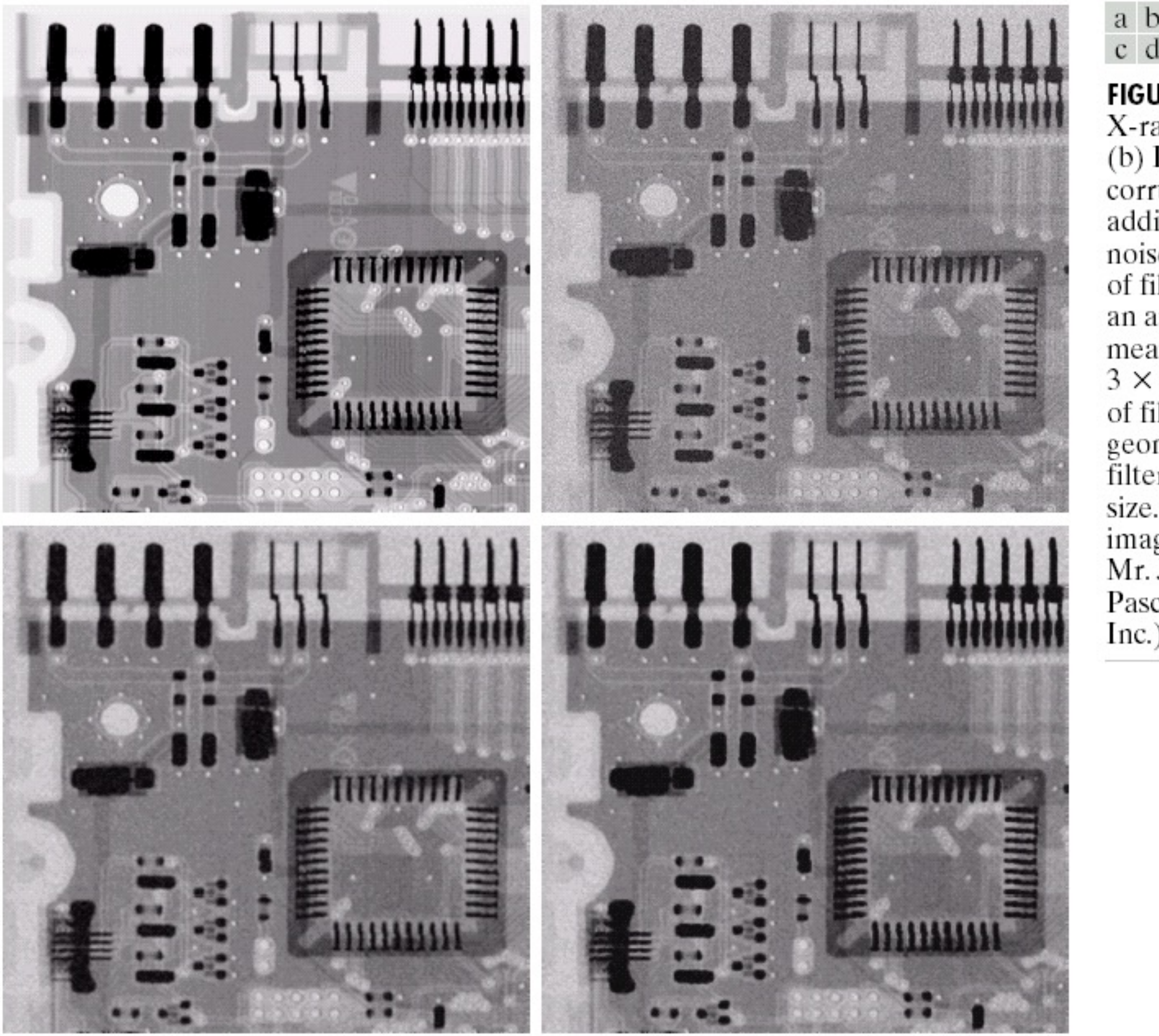


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Harmonic Mean Filter (HMF)

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

Nonlinear

HMF smooths and tends to darken more than the AMF in regions with gray level variability.

HMF is unstable for small gray level values.

Ex

pixel gray levels		arithmetic mean	harmonic mean
I	I	I	I
$I-1$	$I+1$	I	$I-\frac{1}{I}$
$I-2$	$I+2$	I	$I-\frac{4}{I}$

Ex Salt noise reduction using 3x3 HMF

For any window containing 190, we get $9/\left[8(0.01) + \frac{1}{190}\right]$

$= 106$

GMF gives 107.

Ex Salt noise reduction using 3x3 HMF

For any window containing 190, we get $9/\left[8(0.01) + \frac{1}{190}\right]$

$= 106$

GMF gives 107.

Ex Pepper noise reduction using 3x3 HMF

For any window containing 10, we get $9/\left[8(0.01) + \frac{1}{10}\right]$

$= 50$

GMF gives 77.

(Ex) Salt noise reduction using 3x3 HMF

For any window containing 190, we get $9/\left[8(0.01) + \frac{1}{190}\right]$

$= 106$

GMF gives 107.

(Ex) Pepper noise reduction using 3x3 HMF

For any window containing 10, we get $9/\left[8(0.01) + \frac{1}{10}\right]$

$= 50$

GMF gives 77.

HMF is good for removing salt noise, poor for removing pepper noise

Ex Edge blur due to 3x3 HMF

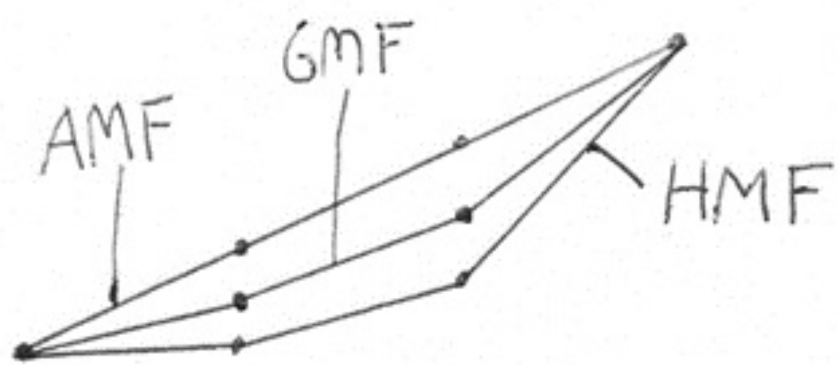
$9/\left[6(0.01)+3(0.005)\right]=120$ GMF gives 126

$9/\left[3(0.01)+6(0.005)\right]=150$ GMF gives 159

Ex Edge blur due to 3×3 HMF

$9/[6(0.01) + 3(0.005)] = 120$ GMF gives 126

$9/[3(0.01) + 6(0.005)] = 150$ GMF gives 159



Contraharmonic Mean Filter (CMF)

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

Contraharmonic Mean Filter (CMF)

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

$Q = 0 \longrightarrow$ CMF becomes AMF

$Q = -1 \longrightarrow$ CMF becomes HMF

For $Q>0$, CMF smooths and tends to brighten more than the AMF in regions with gray level variability

For $Q>0$, CMF smooths and tends to brighten more than the AMF in regions with gray level variability

Ex $Q=1$ $$\hat{f}(x,y) = \frac{\displaystyle\sum_{(s,t)\in S_{xy}} g(s,t)^2}{\displaystyle\sum_{(s,t)\in S_{xy}} g(s,t)}$$

For $Q>0$, CMF smooths and tends to brighten more than the AMF in regions with gray level variability

Ex $Q=1$ $\quad \hat{f}(x,y) = \dfrac{\sum_{(s,t)\in S_{xy}} g(s,t)^2}{\sum_{(s,t)\in S_{xy}} g(s,t)}$

pixel gray levels		arithmetic mean	contraharmonic mean
I	I	I	I
$I-1$	$I+1$	I	$I+\frac{1}{I}$
$I-2$	$I+2$	I	$I+\frac{4}{I}$

Ex Salt noise reduction using 3x3 CMF $(Q = 1)$

For any window containing 190, we get $[8(100)^2 + 190^2]/990 = 117$

AMF gives 110.

(Ex) Salt noise reduction using 3x3 CMF ($Q = 1$)

For any window containing 190, we get $[8(100)^2 + 190^2]/990 = 117$

AMF gives 110.

(Ex) Pepper noise reduction using 3x3 CMF ($Q = 1$)

For any window containing 10, we get $[8(100)^2 + 10^2]/810 = 99$

AMF gives 90.

(Ex) Salt noise reduction using 3x3 CMF ($Q = 1$)

For any window containing 190, we get $[8(100)^2 + 190^2]/990 = 117$

AMF gives 110.

(Ex) Pepper noise reduction using 3x3 CMF ($Q = 1$)

For any window containing 10, we get $[8(100)^2 + 10^2]/810 = 99$

AMF gives 90.

(Ex) Edge blur due to 3x3 CMF ($Q = 1$)

$[6(100)^2 + 3(200)^2]/1200 = 150$ AMF gives 133

$[3(100)^2 + 6(200)^2]/1500 = 180$ AMF gives 167

(Ex) Salt noise reduction using 3x3 CMF $(Q=1)$

For any window containing 190, we get $[8(100)^2 + 190^2]/990 = 117$

AMF gives 110.

(Ex) Pepper noise reduction using 3x3 CMF $(Q=1)$

For any window containing 10, we get $[8(100)^2 + 10^2]/810 = 99$

AMF gives 90.

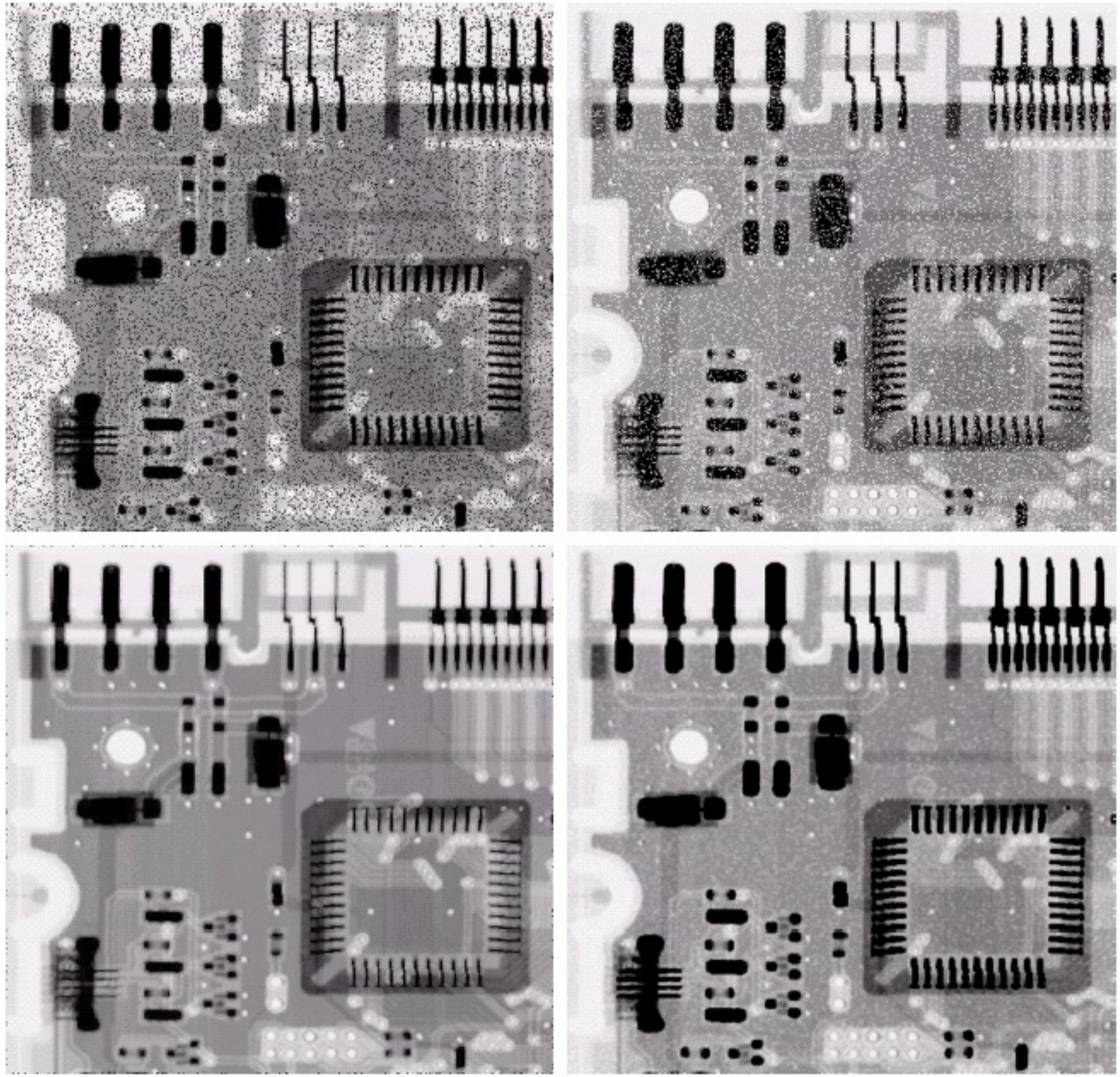
(Ex) Edge blur due to 3x3 CMF $(Q=1)$

$[6(100)^2 + 3(200)^2]/1200 = 150$ AMF gives 133

$[3(100)^2 + 6(200)^2]/1500 = 180$ AMF gives 167

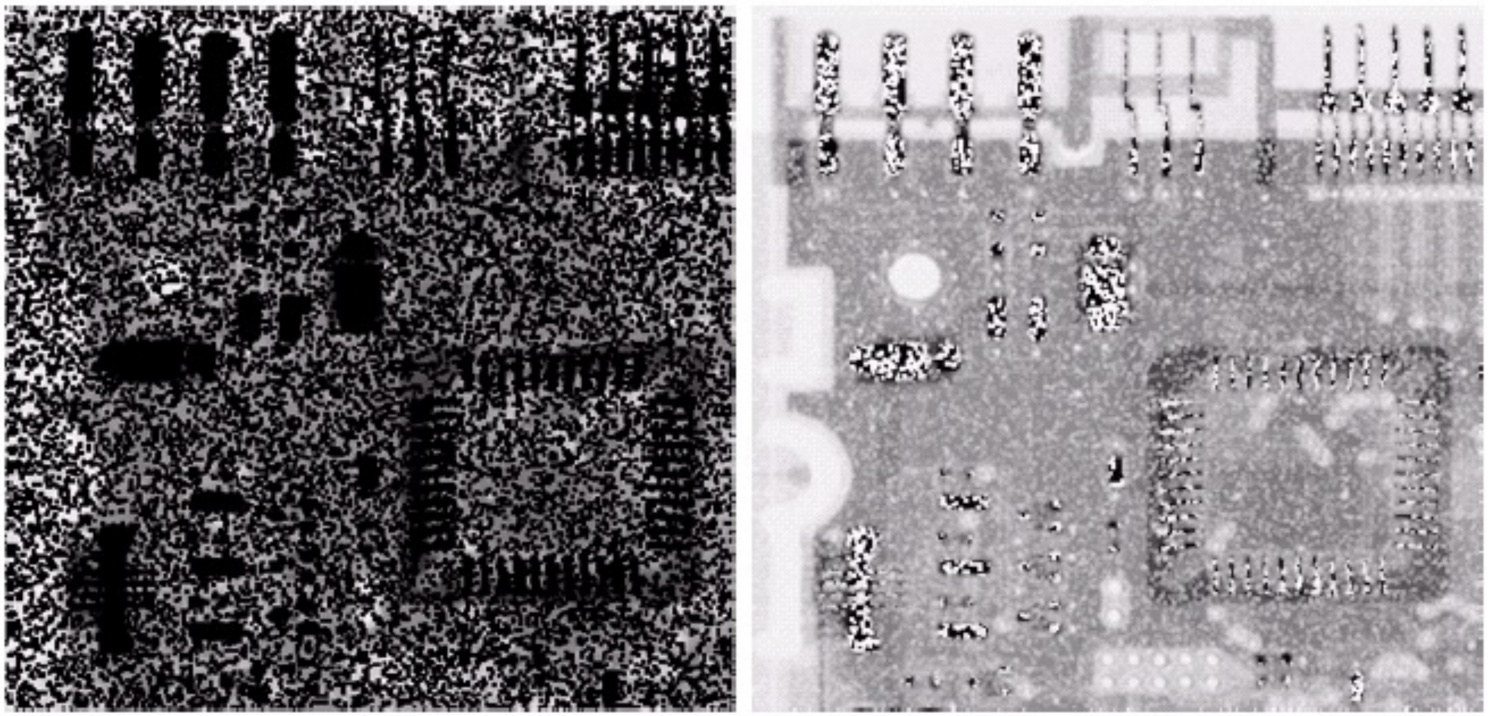
$Q>0$ CMF is good for removing pepper noise,

poor for removing salt noise



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 $Q = 1.5$.

Adaptive Filters

The behavior of an adaptive filter at (x,y) depends on the characteristics of the pixels in S_{xy}.

Adaptive Filters

The behavior of an adaptive filter at (x,y) depends on the characteristics of the pixels in S_{xy}.

Adaptive Noise Reduction Filter (ANRF)

$$g(x,y) = f(x,y) + n(x,y)$$

S_{xy} = rectangular window

σ_n^2 = variance of noise $n(x,y)$ (assumed Known)

m_L = mean of pixel values in S_{xy}

σ_L^2 = variance of pixel values in S_{xy}

Adaptive Filters

The behavior of an adaptive filter at (x,y) depends on the characteristics of the pixels in S_{xy}.

Adaptive Noise Reduction Filter (ANRF)

$$g(x,y) = f(x,y) + n(x,y)$$

S_{xy} = rectangular window

σ_n^2 = variance of noise $n(x,y)$ (assumed Known)

m_L = mean of pixel values in S_{xy}

σ_L^2 = variance of pixel values in S_{xy}

Goal for adaptive noise reduction filter is to smooth noise but not smooth edges.

Filter properties

1) If $\sigma_n^2 = 0$, then $\hat{f}(x,y) = g(x,y)$

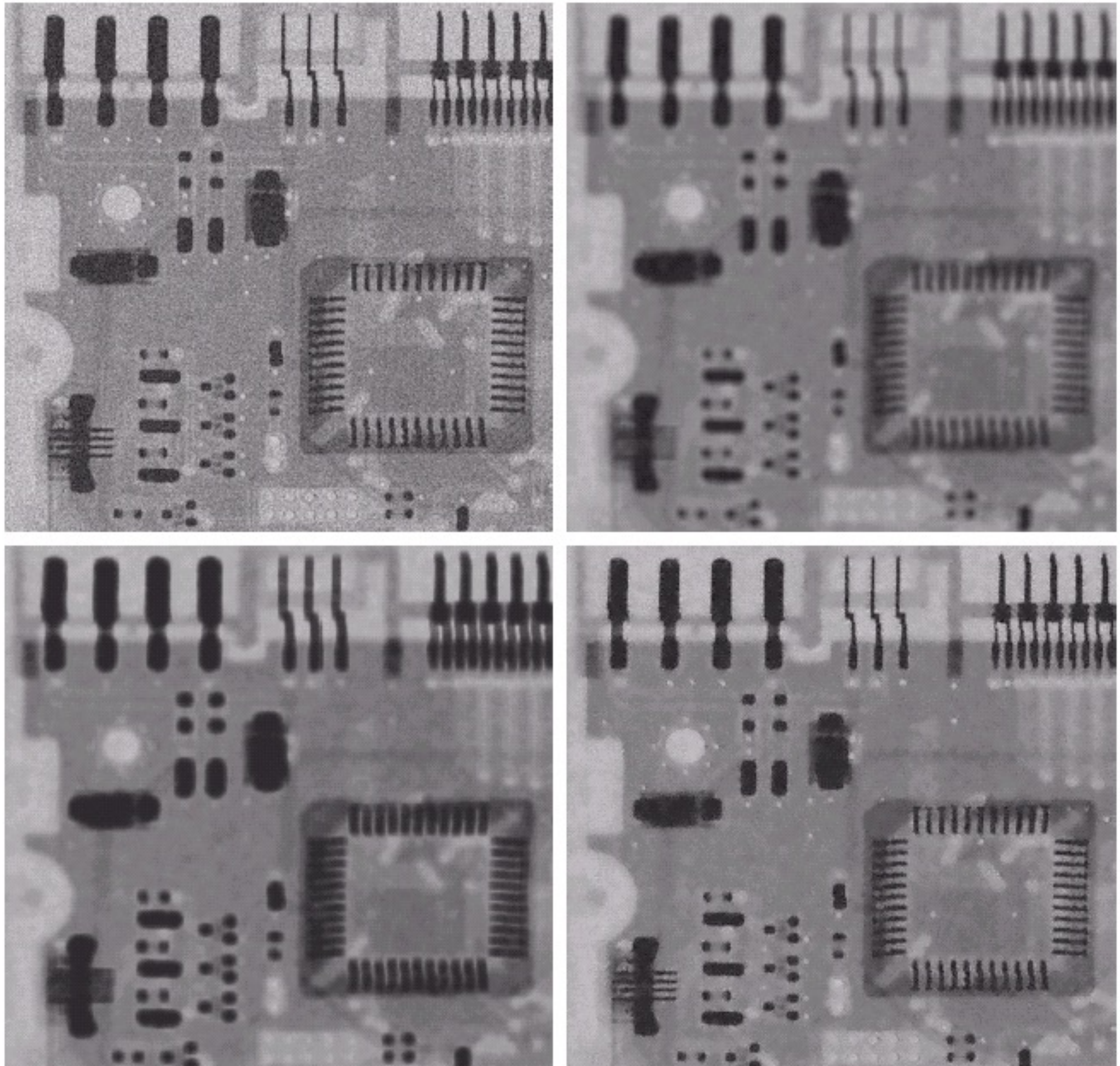
2) If S_{xy} contains an edge, then $\sigma_L^2 > \sigma_n^2$

Don't smooth, $\hat{f}(x,y) = g(x,y)$

3) If S_{xy} does not contain an edge, then $\sigma_L^2 \approx \sigma_n^2$

Smooth, $\hat{f}(x,y) = m_L$

ANRF $\quad \hat{f}(x,y) = g(x,y) - \dfrac{\sigma_n^2}{\sigma_L^2}\left(g(x,y) - m_L\right)$



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

Goal for adaptive median filter is to remove impulse noise from images with a large fraction of impulse noise pixels.

Adaptive Median Filter

Goal for adaptive median filter is to remove impulse noise from images with a large fraction of impulse noise pixels.

Problem with standard median filter is that if half of the pixels in S_{xy} have the same impulse noise value, then the median filter will output this impulse noise value.

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

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

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

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

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

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

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

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

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

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

Adaptive Median Filter

A: if $z_{min} < z_{med} < z_{max}$ then goto B

else
- increase size of S_{xy}
- if size $\le S_{max}$ then goto A
  else $\hat{f}(x,y) = z_{xy}$

B: if $z_{min} < z_{xy} < z_{max}$ then $\hat{f}(x,y) = z_{xy}$

else $\hat{f}(x,y) = z_{med}$

Summary: Increase window size until median cannot be impulse. If pixel itself cannot be impulse, then return pixel value. If pixel value might be impulse, then return median

Summary: Increase window size until median cannot be impulse. If pixel itself cannot be impulse, then return pixel value. If pixel value might be impulse, then return median

Ex Impulse noise

49	50	50	49	50
50	255	255	50	255
50	50	50	255	50
50	255	255	50	255
50	49	50	51	50

Summary: Increase window size until median cannot be impulse. If pixel itself cannot be impulse, then return pixel value. If pixel value might be impulse, then return median

Ex Impulse noise

49	50	50	49	50
50	255	255	50	255
50	50	50	255	50
50	255	255	50	255
50	49	50	51	50

3x3 median filter applied to center pixel $\rightarrow \hat{f}(x,y) = 255$

3x3 adaptive median filter applied to center pixel

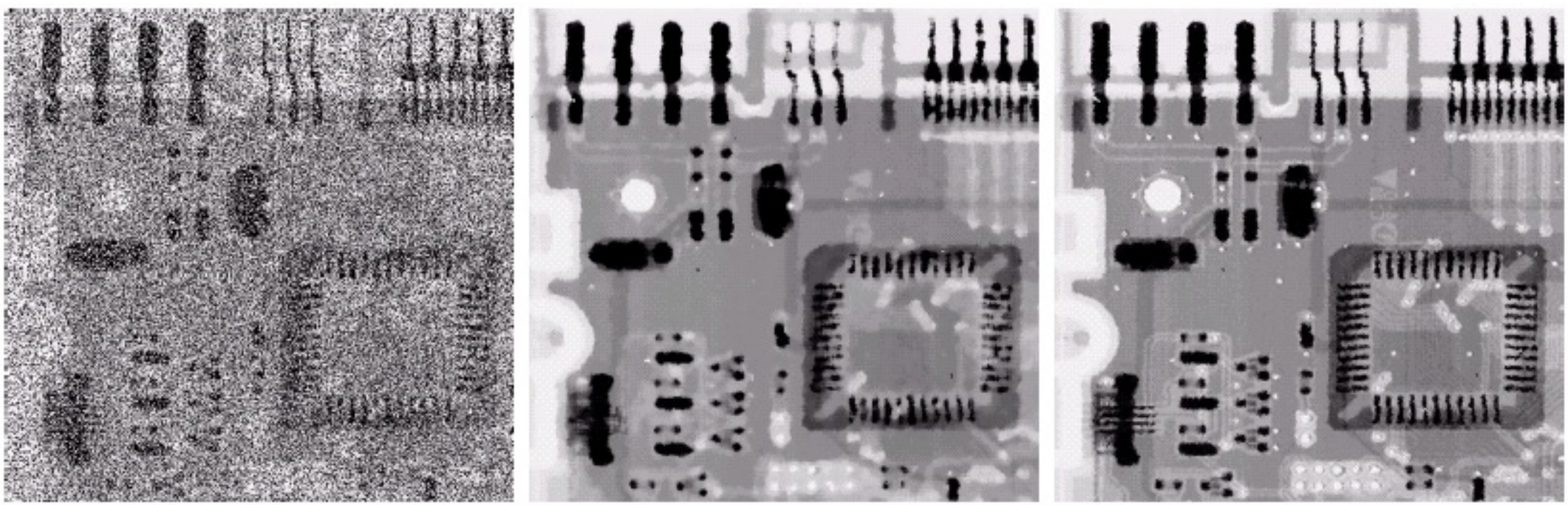
$\rightarrow$ increase size to 5x5 $\rightarrow \hat{f}(x,y) = 50$

Ex Thin line

0	1	51	2	0
2	0	50	1	2
1	2	52	0	1

3x3 median filter applied to center pixel $\rightarrow \hat{f}(x,y)=2$

3x3 adaptive median filter applied to center pixel $\rightarrow \hat{f}(x,y)=50$



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 $S_{max} = 7$.