CSL7320: Digital Image Analysis

Image Restoration and Reconstruction

Image Restoration and Reconstruction

http://fireoracleproductions.com/services__samp

Google image

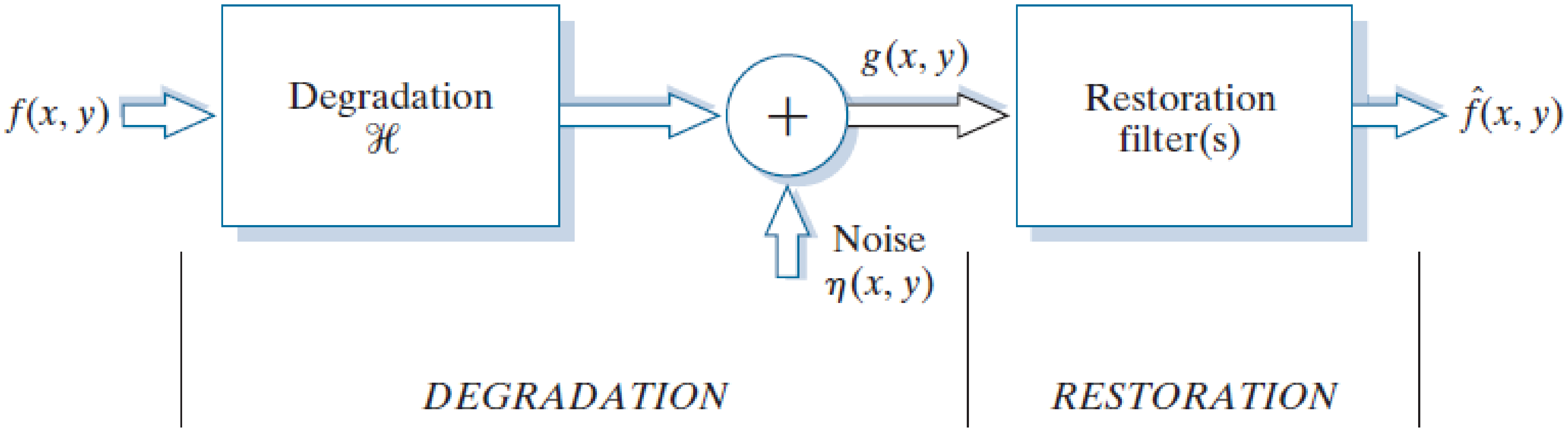
Image degradation due to

- noise in transmission
- imperfect image acquisition
    - environmental condition
    - quality of sensor

Slide credit: Yan Tong

Image Restoration and Reconstruction

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



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

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

Identity *H*(*u*,*v*) → degradation only comes from additive noise

Slide credit: Yan Tong

Image Restoration and Reconstruction

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

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

Noise models:

- Impulse noise: pepper and salt
- Continuous noise model:
  - Gaussian, Rayleigh, Gamma, Exponential, Uniform

Slide credit: Yan Tong

Properties of Noise

- **Spatial properties**
  - Spatially periodic noise
  - Spatially independent noise

Tt is uncorrelated with respect to the image itself (that is, there is no correlation between pixel values and the values of noise components)

- **Frequency properties**
  - White noise – noise containing all frequencies within a bandwidth

If Fourier spectrum of noise is constant, the noise is called *white noise*

Slide credit: Yan Tong

Noise Probability Density Functions

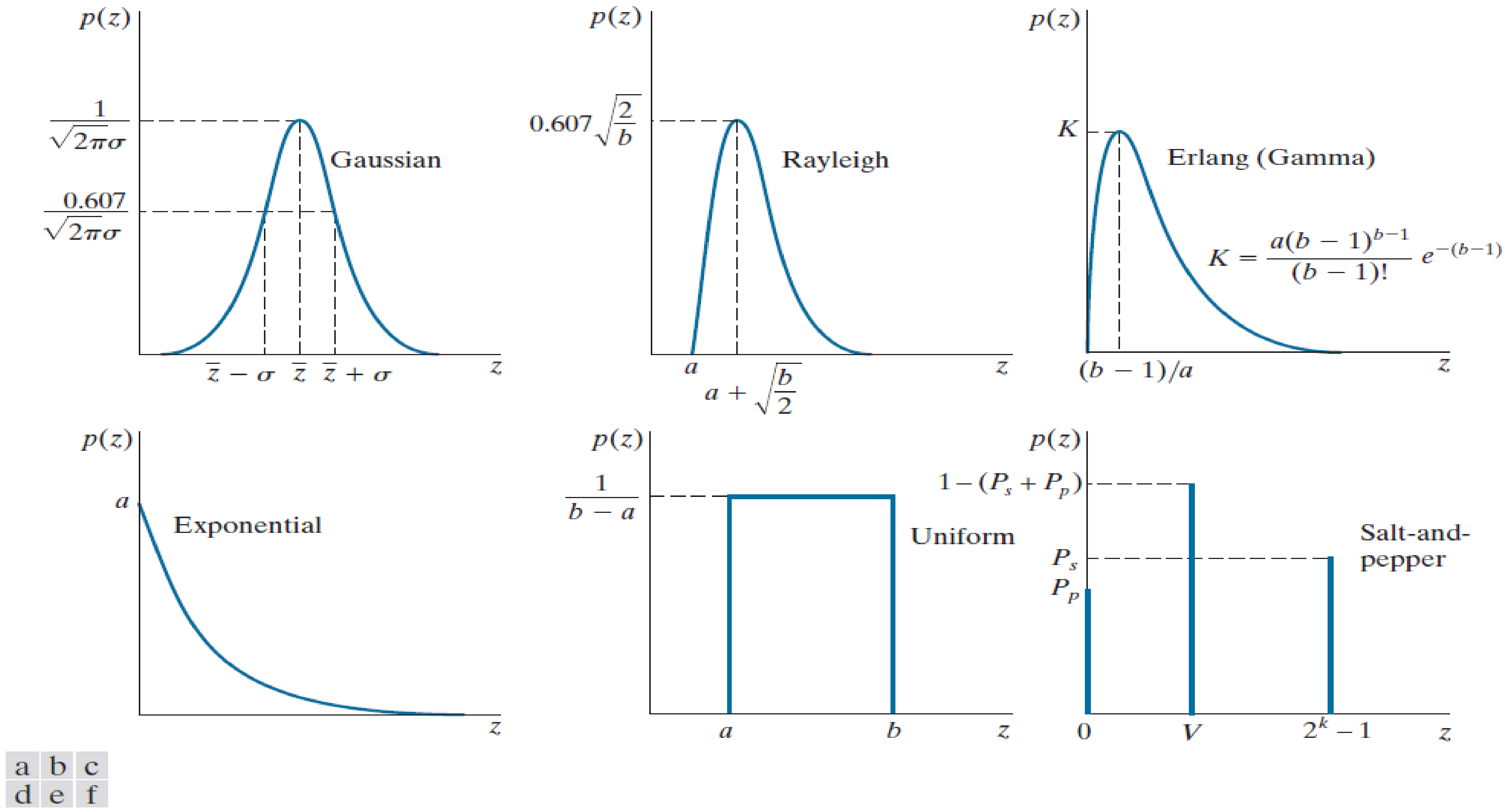


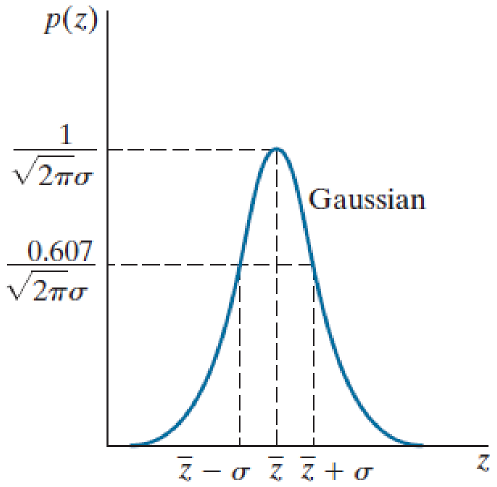
FIGURE 5.2 Some important probability density functions.

Noise Probability Density Functions

Gaussian Noise

The PDF of a *Gaussian* random variable, z, is defined by the following familiar expression:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}} \qquad -\infty < z < \infty \tag{5-3}$$



Rayleigh Noise

The PDF of *Rayleigh* noise is given by

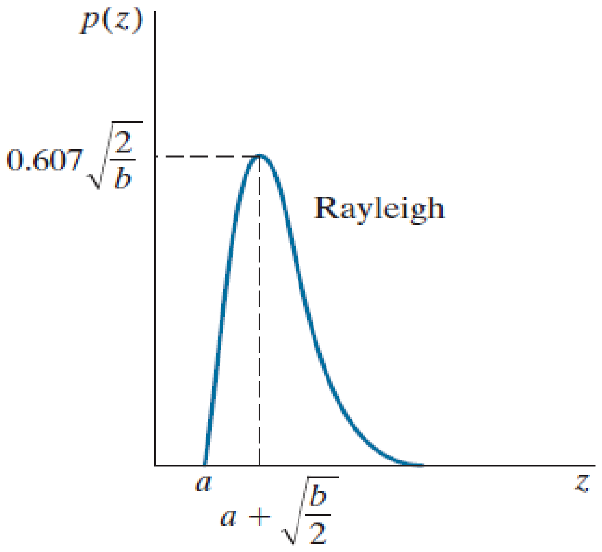
$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases} \tag{5-4}$$

The mean and variance of z when this random variable is characterized by a Rayleigh PDF are

$$\bar{z} = a + \sqrt{\pi b/4} \tag{5-5}$$

and

$$\sigma^2 = \frac{b(4-\pi)}{4} \tag{5-6}$$



Noise Probability Density Functions

Exponential Noise

The PDF of *exponential* noise is given by

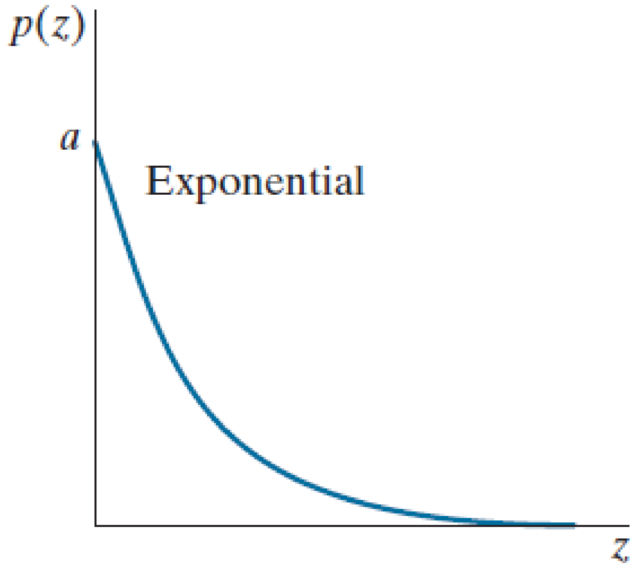
$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases} \tag{5-10}$$

where $a > 0$. The mean and variance of z are

$$\bar{z} = \frac{1}{a} \tag{5-11}$$

and

$$\sigma^2 = \frac{1}{a^2} \tag{5-12}$$



Noise Probability Density Functions

Uniform Noise

The PDF of *uniform* noise is

$$p(z) = \begin{cases} \dfrac{1}{b - a} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \qquad (5\text{-}13)$$

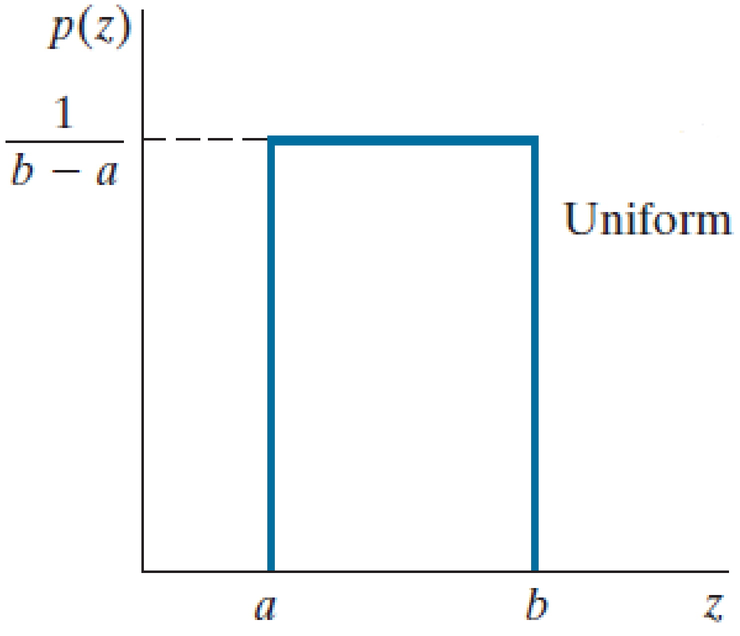
The mean and variance of z are

$$\bar{z} = \frac{a + b}{2} \qquad (5\text{-}14)$$

and

$$\sigma^2 = \frac{(b - a)^2}{12} \qquad (5\text{-}15)$$

Figure 5.2(e) shows a plot of the uniform density.



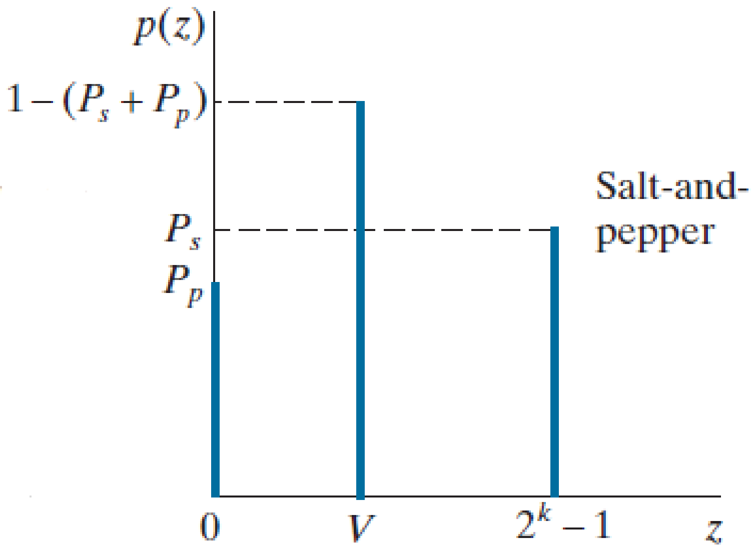
Noise Probability Density Functions

Salt-and-Pepper Noise

If k represents the number of bits used to represent the intensity values in a digital image, then the range of possible intensity values for that image is $[0, 2^k - 1]$ (e.g., $[0,255]$ for an 8-bit image). The PDF of *salt-and-pepper* noise is given by

$$p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1 - (P_s + P_p) & \text{for } z = V \end{cases} \tag{5-16}$$

where V is any integer value in the range $0 < V < 2^k - 1$.



Noisy Images and Their Histograms

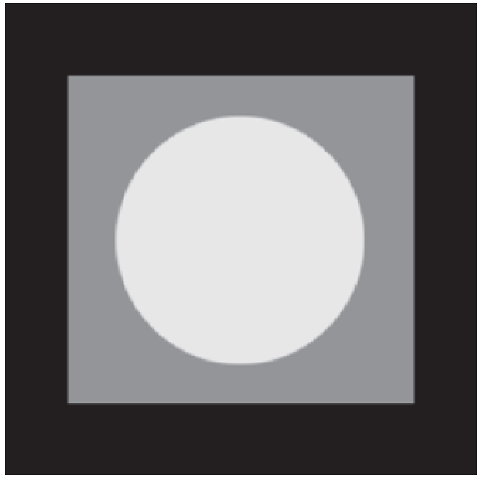


FIGURE 5.3 Test pattern used to illustrate the characteristics of the PDFs from Fig. 5.2.

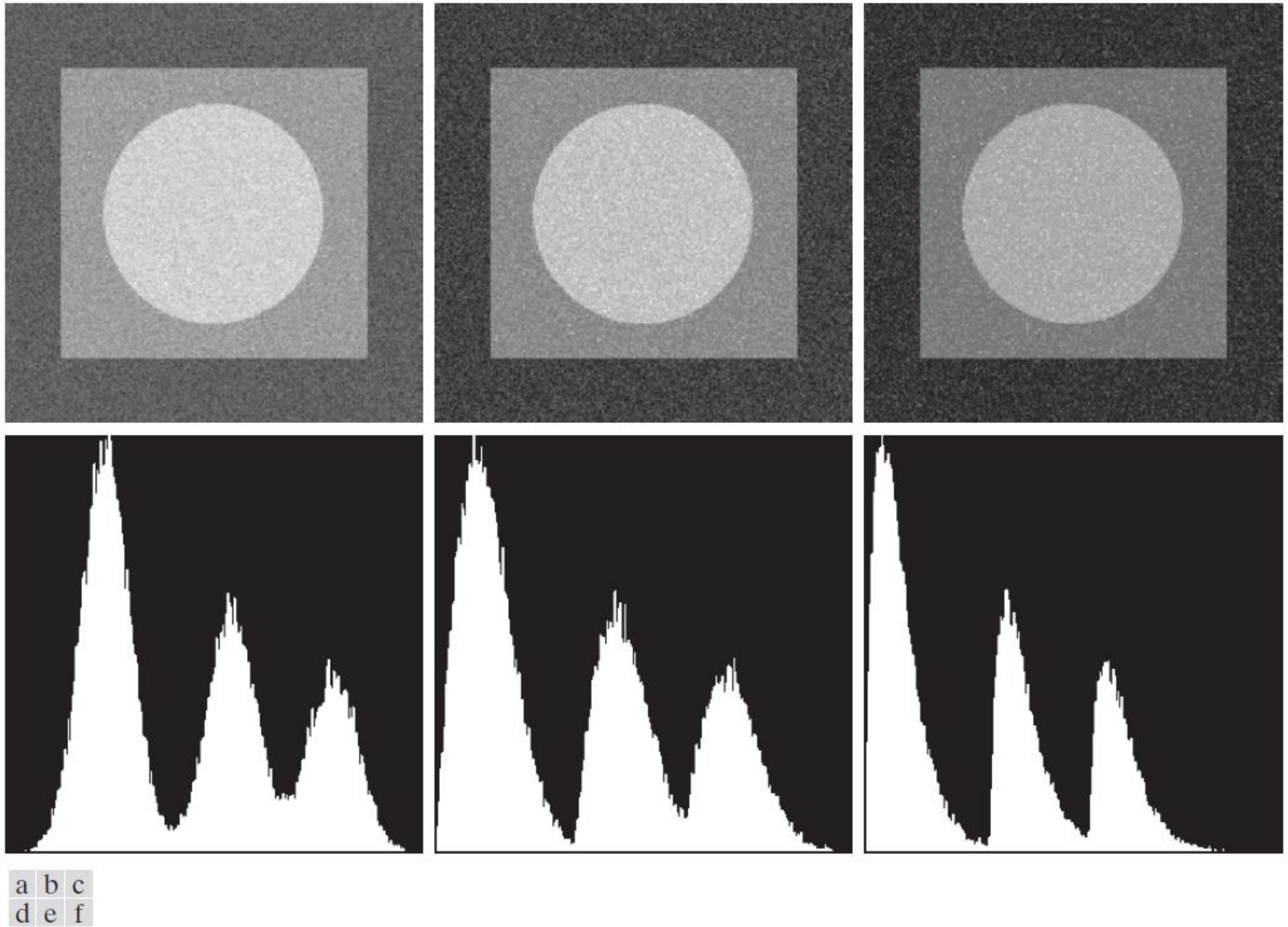


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and Erlanga noise to the image in Fig. 5.3.

Noisy Images and Their Histograms

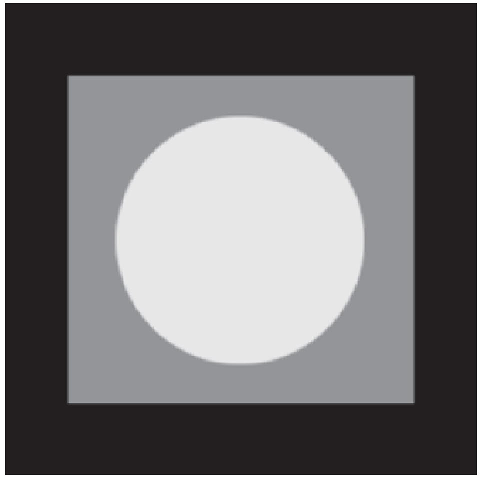
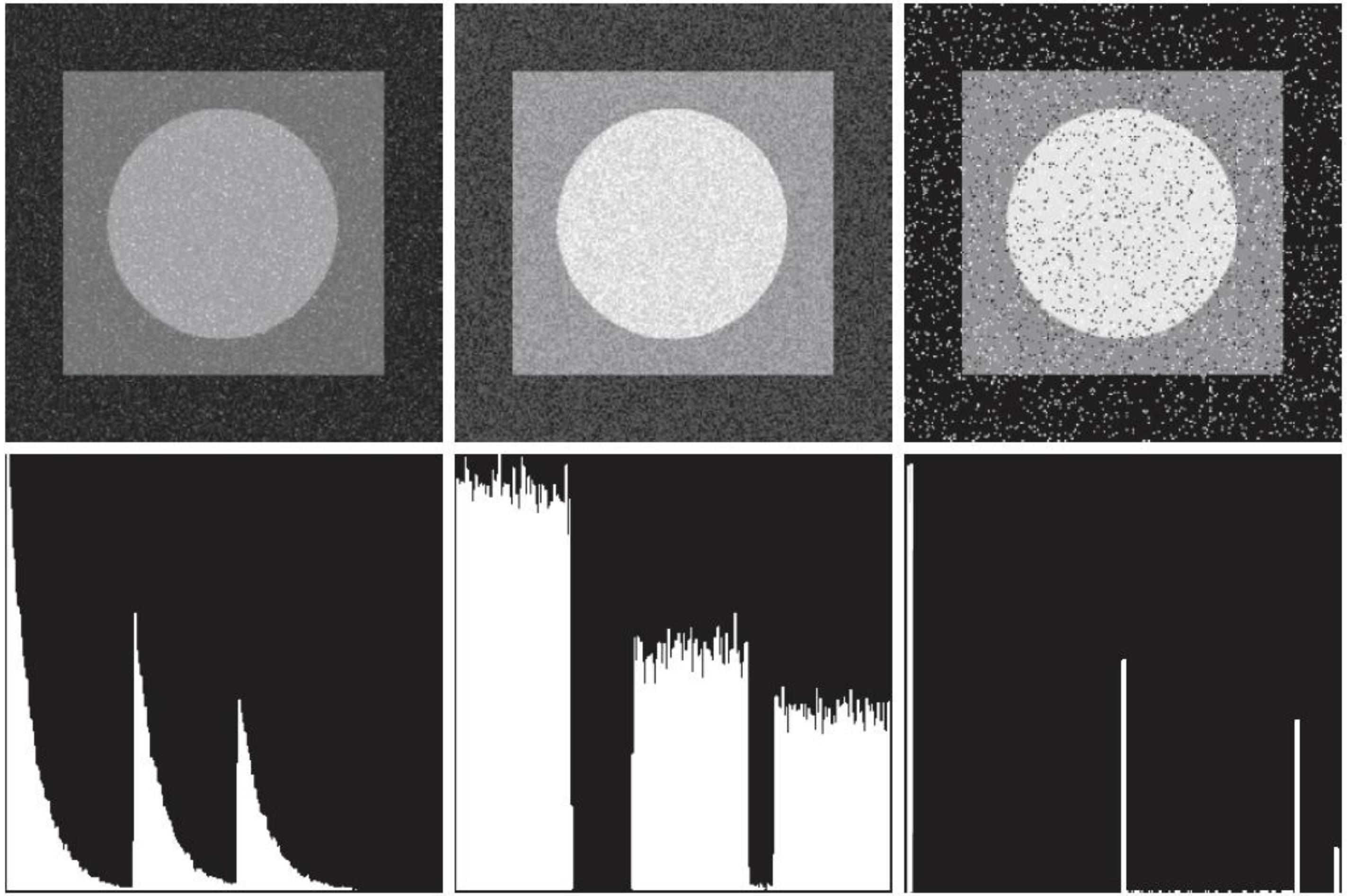


FIGURE 5.3 Test pattern used to illustrate the characteristics of the PDFs from Fig. 5.2.



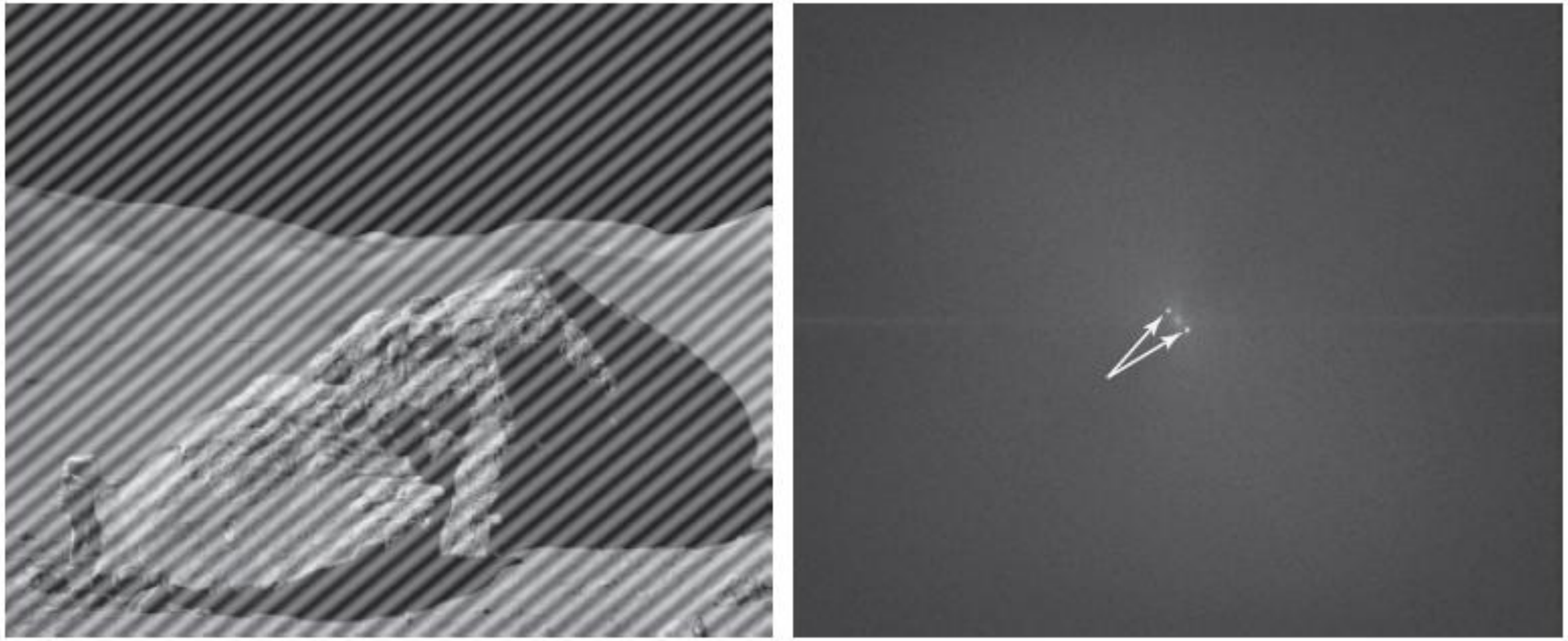
g	h	i
j	k	l

FIGURE 5.4 (*continued*) Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the image in Fig. 5.3. In the salt-and-pepper histogram, the peaks in the origin (zero intensity) and at the far end of the scale are shown displaced slightly so that they do not blend with the page background.

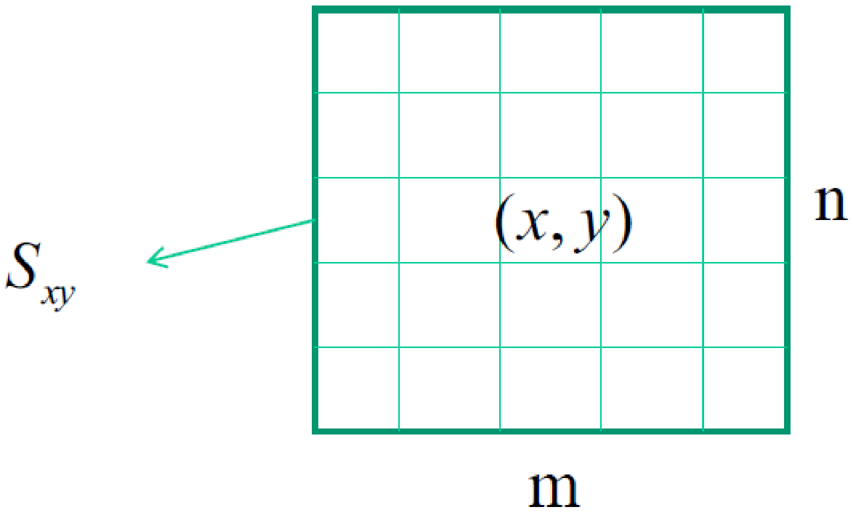
Noisy Images

a b

FIGURE 5.5 (a) Image corrupted by additive sinusoidal noise. (b) Spectrum showing two conjugate impulses caused by the sine wave. (Original image courtesy of NASA.)



Mean Filters for Continuous Noise Models



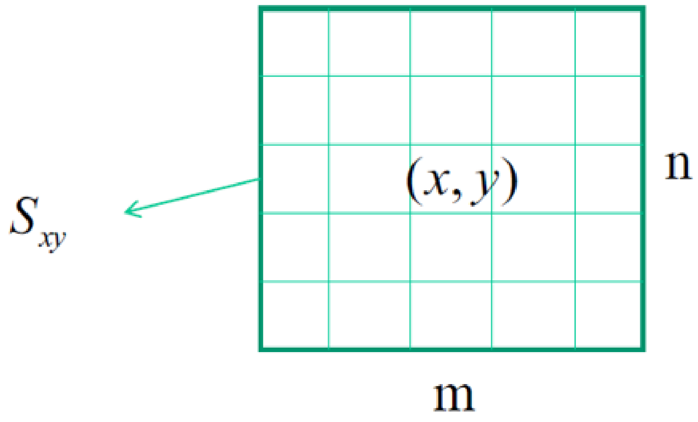
Arithmetic Mean Filter: a linear filter

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

Slide credit: Yan Tong

Non-linear Mean Filter: Geometric Mean Filter

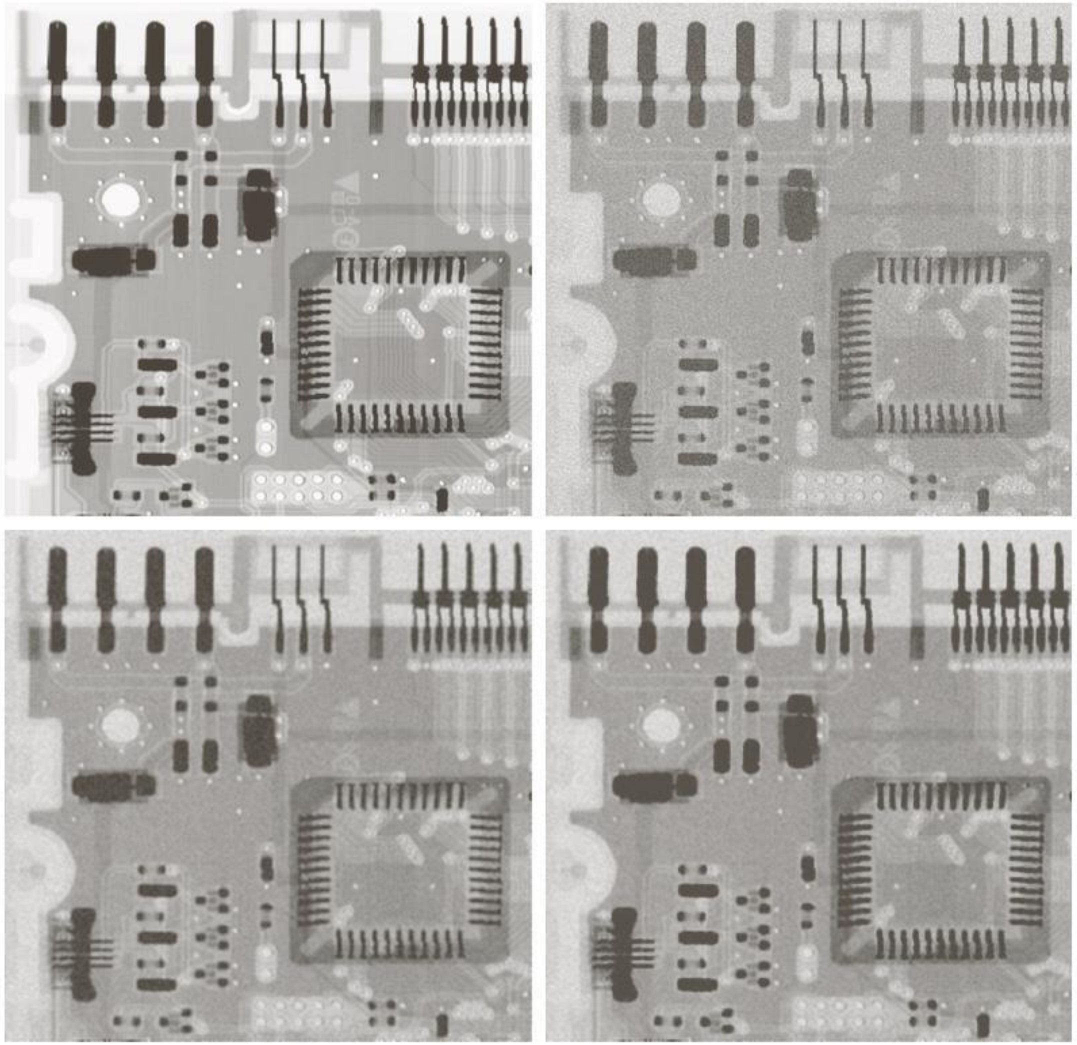
$$\hat{f}(x,y) = \left[\prod_{(r,c)\in S_{xy}} g(r,c) \right]^{\frac{1}{mn}}$$



- Removing the salt noise
- Fail in the pepper noise

Slide credit: Yan Tong

Example: Geometric Mean Filter



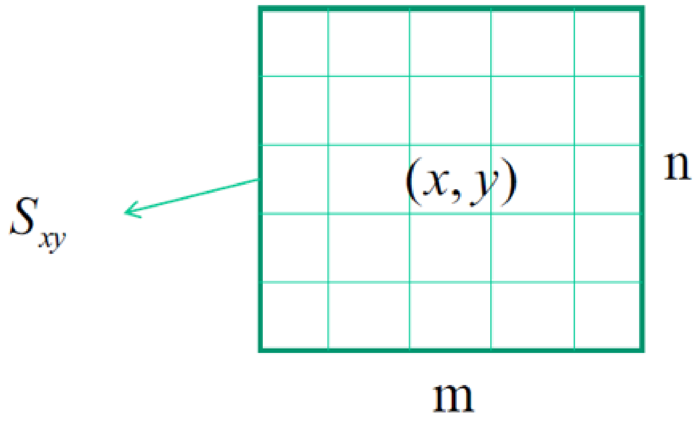
a b
c d

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.)

Slide credit: Yan Tong

Non-linear Mean Filter: Harmonic Mean Filter

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



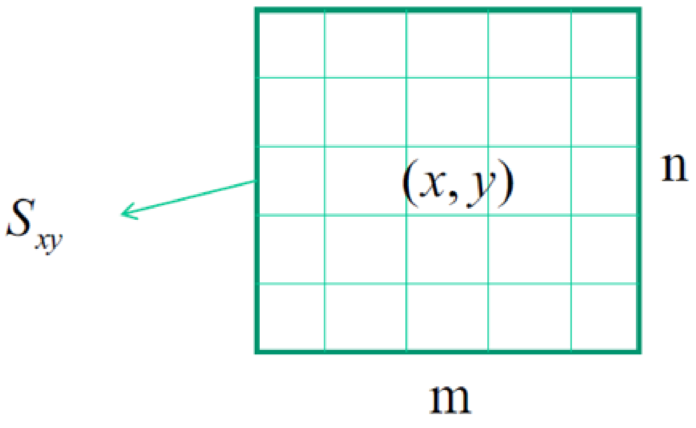
- Removing the salt noise
- Fail in the pepper noise

$$H(x_1, x_2, \ldots, x_n) = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Slide credit: Yan Tong

Non-linear Mean Filter: Contraharmonic Mean Filter

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



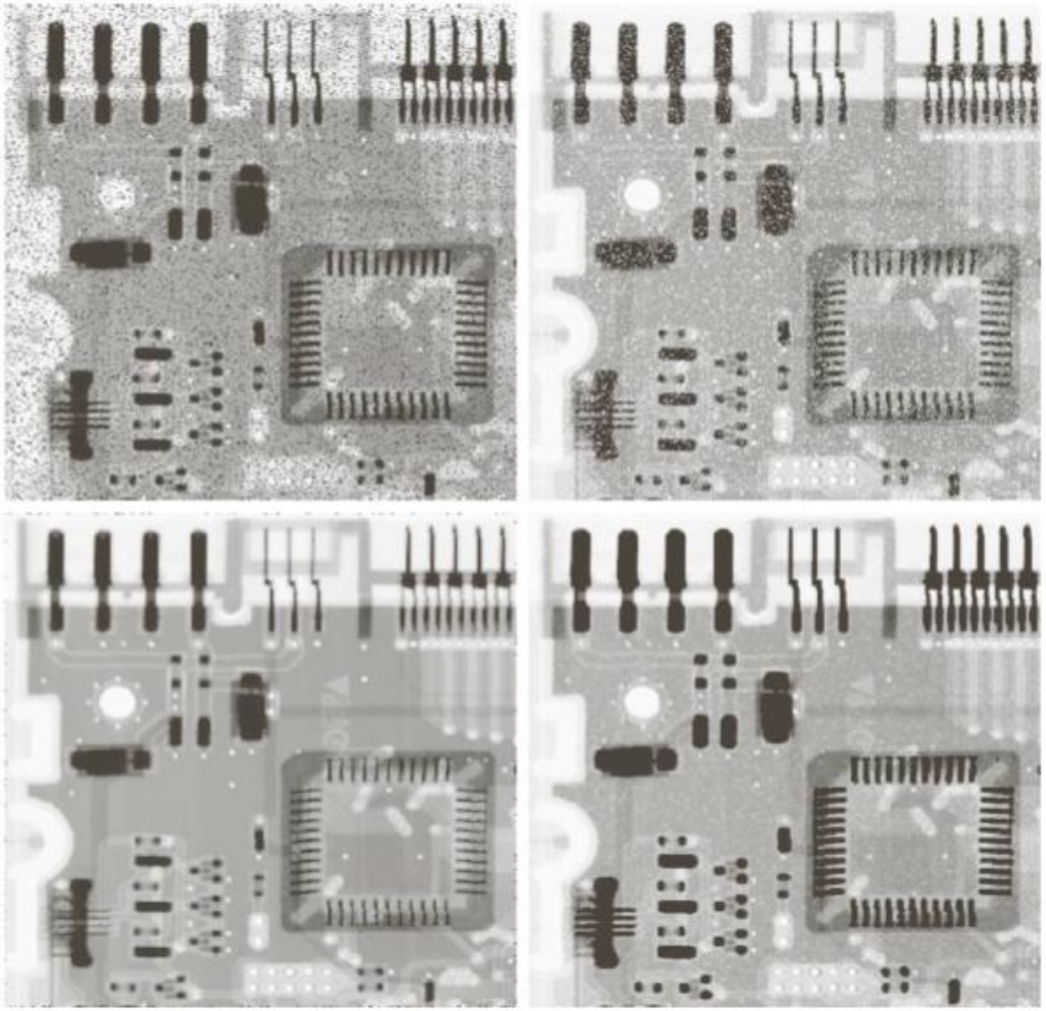
Q is the order of the filter

- positive Q removes pepper noise
- negative Q removes salt noise
- Special cases: Q=0, Q=-1

$$\mathrm{C}(x_1, x_2, \ldots, x_n) = \frac{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}{\frac{1}{n}\left(x_1 + x_2 + \cdots + x_n\right)}, = \frac{x_1^2 + x_2^2 + \cdots + x_n^2}{x_1 + x_2 + \cdots + x_n}.$$

Slide credit: Yan Tong

Example: Contraharmonic Mean Filter



a b
c d

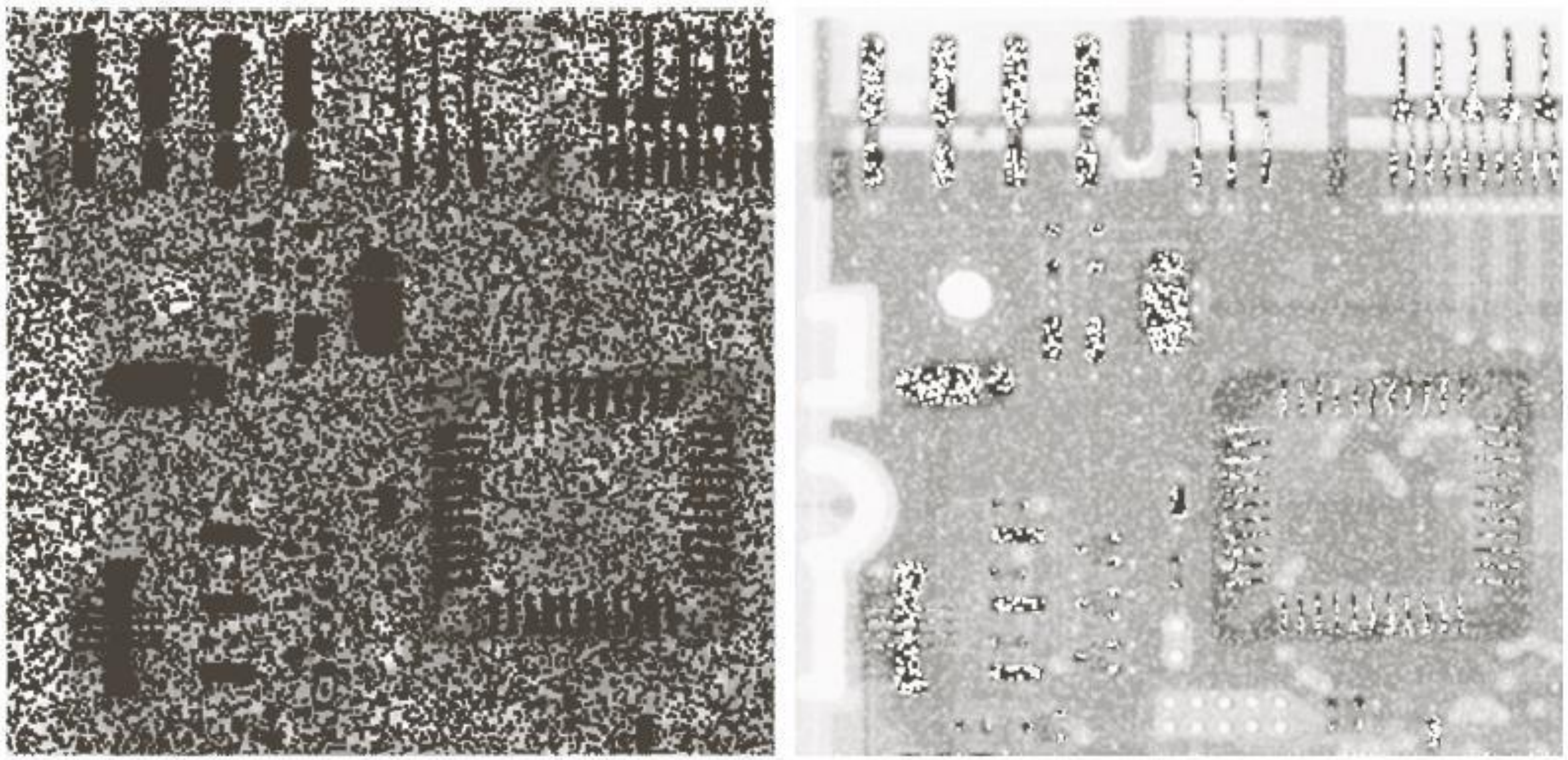
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Slide credit: Yan Tong

A Failed Case with Wrong Sign of Contraharmonic 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 $Q = 1.5$.



Slide credit: Yan Tong

Summary: Mean Filters for Continuous Noise Models

Arithmetic Mean Filter: a linear filter

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

- Work well for continuous noise

Slide credit: Yan Tong

Summary: Non-linear Mean Filters

Geometric Mean Filter

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

Harmonic Mean Filter

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

Work well for
- Continuous noise
- Salt noise

Fail for the pepper noise

Contraharmonic Mean Filter

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

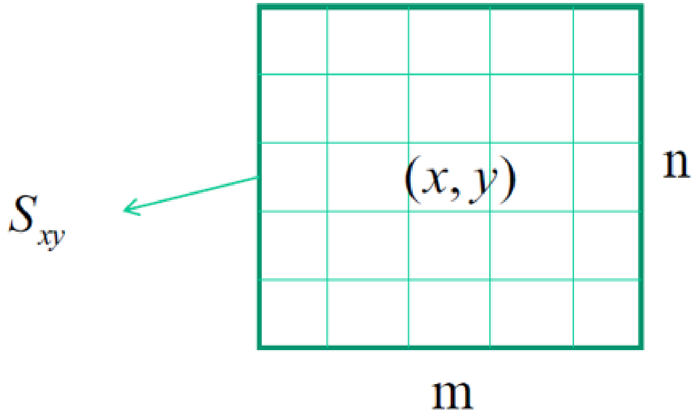
Q is the order of the filter
- positive Q removes pepper noise
- negative Q removes salt noise
- Special cases: Q=0, Q=-1

Slide credit: Yan Tong

Order-Statistic Filters

Median Filter

$$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}}\{g(r,c)\}$$



Max and Min Filters

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

Midpoint Filter

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

Alpha-Trimmed Mean Filter

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(r,c)\in S_{xy}} g_R(r,c)$$

Order-Statistic Filters -- Median Filter

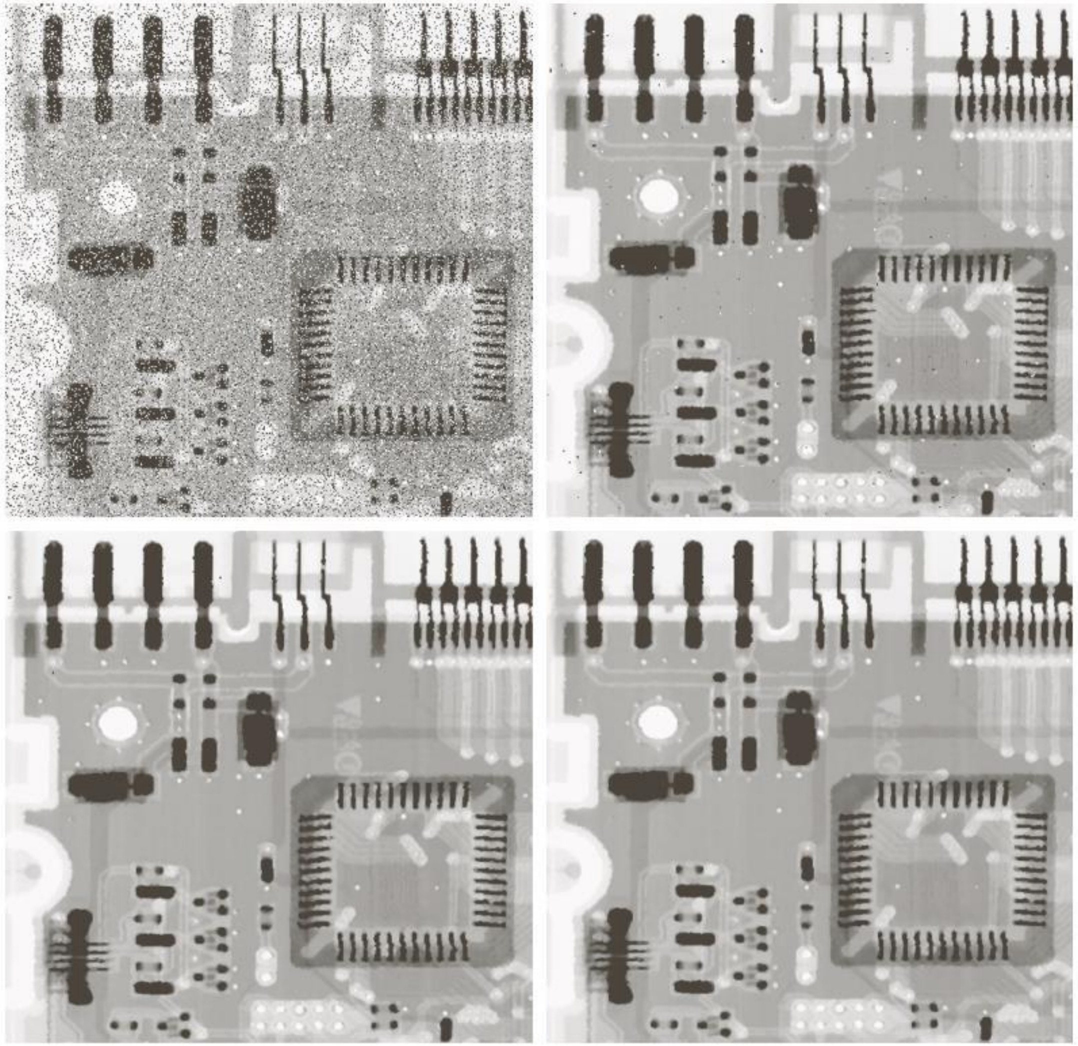
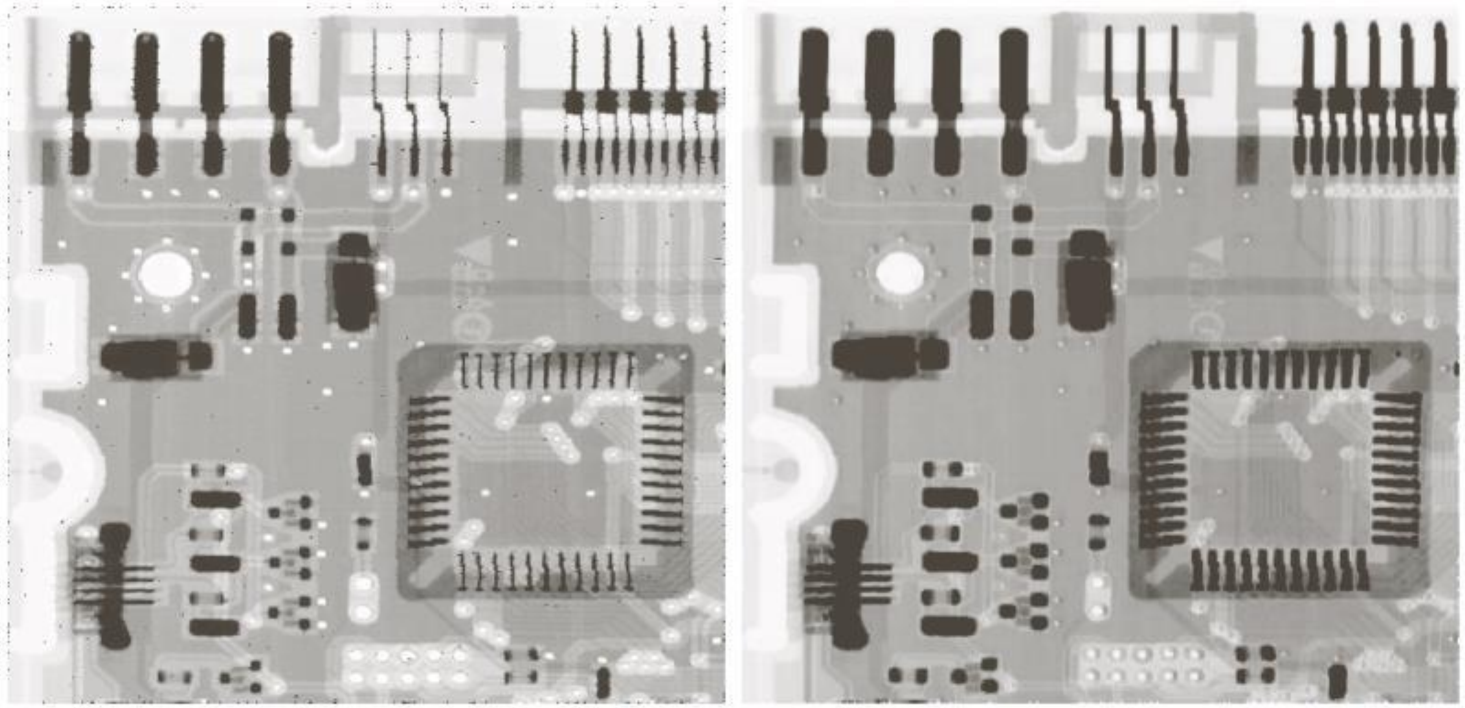


FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Repeating median filter will remove most of the noise while increase image blurring

Slide credit: Yan Tong

Order-Statistic Filters -- Max/Min Filters



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.

- Find the extreme points
- Remove the targeting impulse noise

Slide credit: Yan Tong

Order-Statistic Filters

a b
c d
e f

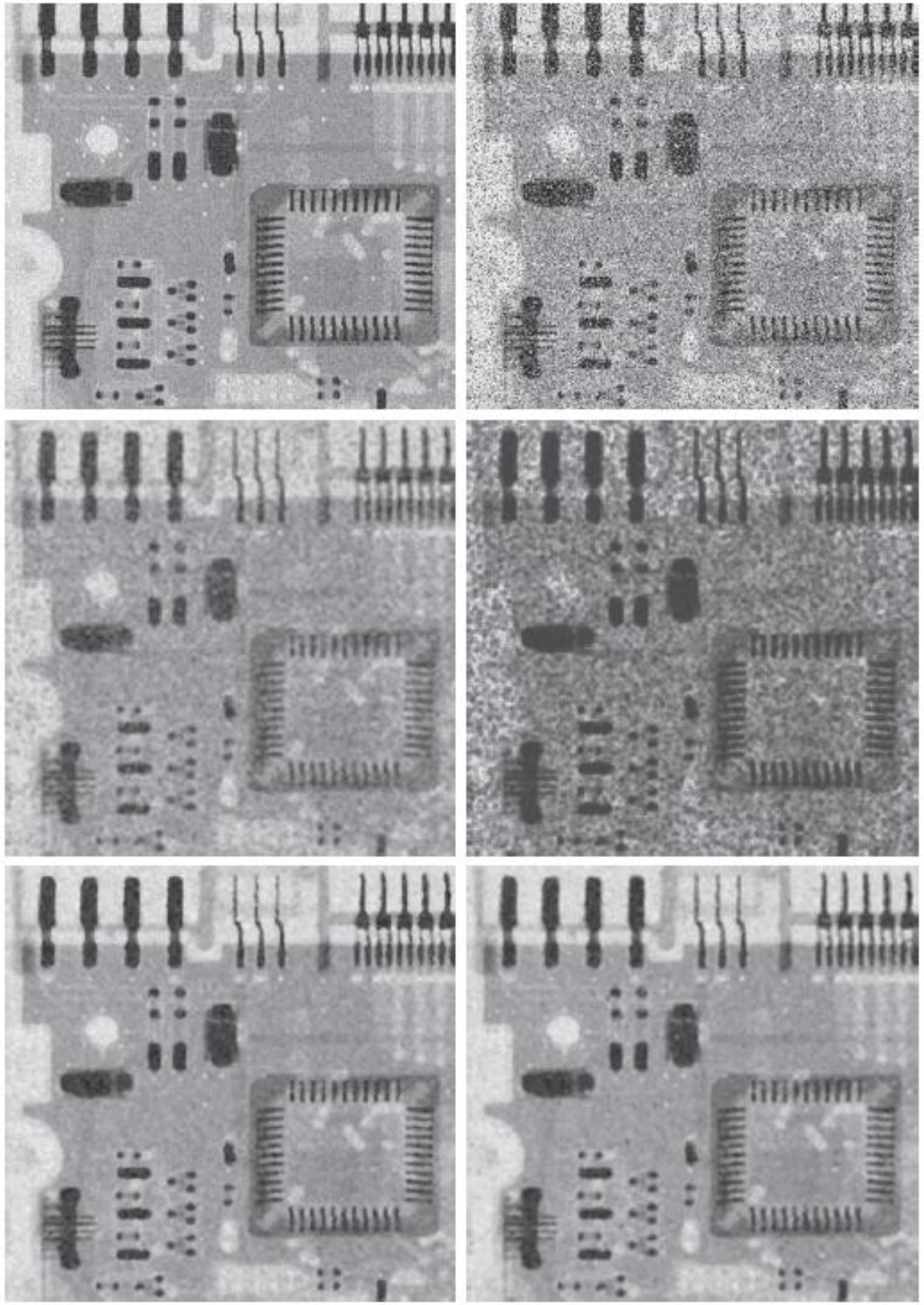


FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. (c)-(f) Image (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; (f) alpha-trimmed mean filter, with $d = 6$.

Slide credit: Yan Tong

Adaptive Median Filter

Stage A: check if the median value is an extreme value

$A1 = z_{\text{med}} - z_{\text{min}}$

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

If $A1 > 0$ AND $A2 < 0$, go to stage B

Else increase the window size

If window size $\leq S_{\text{max}}$ repeat stage A

Else output z_{med}

Goal 1: <u>remove salt-and-pepper noise with higher probability</u>

Goal 2: <u>smoothing the noise other than impulses</u>

Goal 3: <u>reduce distortion</u>

Stage B: check if the center pixel is an extreme value

$B1 = z_{xy} - z_{\text{min}}$

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

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

Else output z_{med}

Z_{min} = Minimum gray level value in S_{xy}.

Z_{max} = Maximum gray level value in S_{xy}

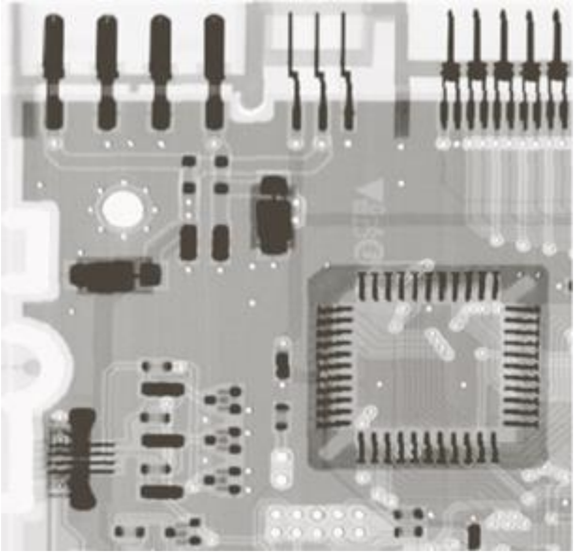
Z_{med} = Median of gray levels in S_{xy}

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

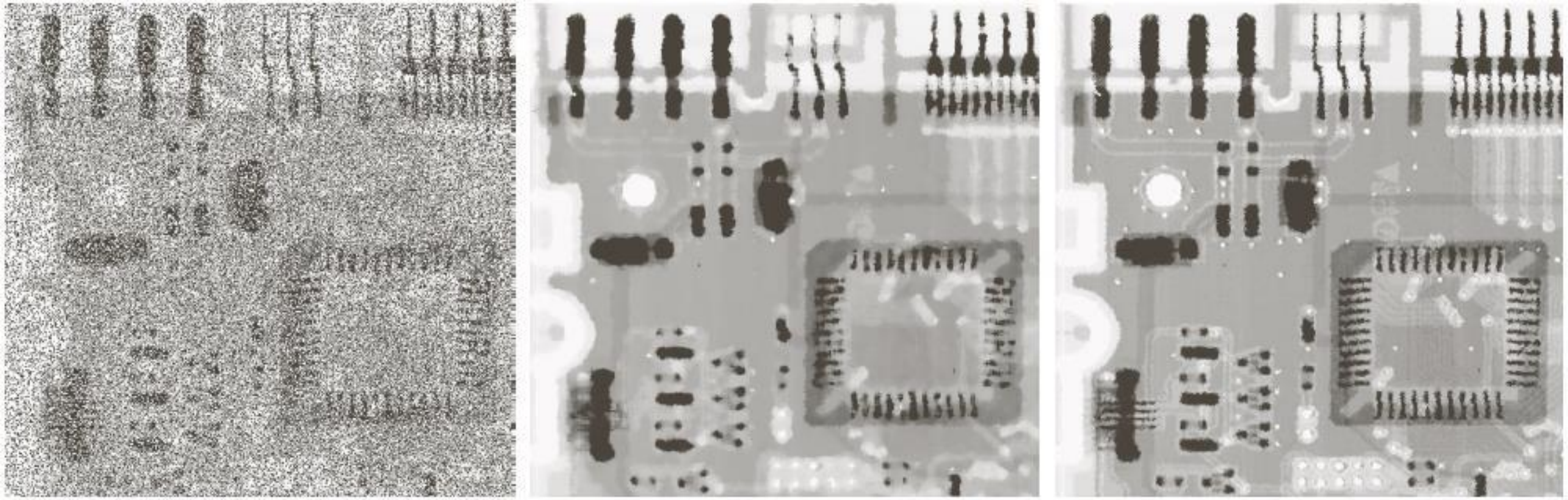
S_{max} = Maximum allowed size of S_{xy}

Slide credit: Yan Tong

Example: Adaptive Median Filter



Clean Image



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 $S_{\max} = 7$.

Slide credit: Yan Tong

Linear, Position-Invariant Degradations – Noise Free Case

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

Linearity $H[af_1(x,y)+bf_2(x,y)] = aH[f_1(x,y)]+bH[f_2(x,y)]$

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

$\Downarrow$

H does not depend on the location (x,y), only represented by the input and output

Slide credit: Yan Tong

Impulse Response for Linear H

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha,y-\beta)\,d\alpha\,d\beta$$

Sifting property

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

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

Impulse response

Slide credit: Yan Tong

Image Degradations

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

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

Slide credit: Yan Tong

Degradation VS Restoration

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

Note: a linear, position invariant degradation system with additive noise can be modeled as the convolution of the degradation function with the image plus the additive noise.

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

$$f(x,y) \underset{\text{Restoration (deconvolution)}}{\overset{\text{Degradation (convolution)}}{\rightleftarrows}} g(x,y)$$

Slide credit: Yan Tong

Degradation VS Restoration

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

$$\downarrow$$

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

Estimate the Degradation Function

- **Observation**
- **Experimentation**
- **Mathematical modeling**

Estimate Degradation Function - Observation

Assumptions:

- **The degradation function is linear and position-invariant**
- **No other knowledge about the degradation function**

Estimation by image observation:

- **Extract a subimage with strong signal** → Higher signal-to-noise ratio
- **Perform restoration on the subimage**

degradation function in the subimage ← $$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

$G_s(u,v)$ → Observed subimage

$\hat{F}_s(u,v)$ → Restored subimage

$$H_s(u,v) \Downarrow H(u,v)$$

Application: restoring old pictures

Slide credit: Yan Tong

Estimate Degradation Function - Experimentation

Assumptions:

- **A similar equipment is available**
- **Change the system setting can achieve similar degraded images**

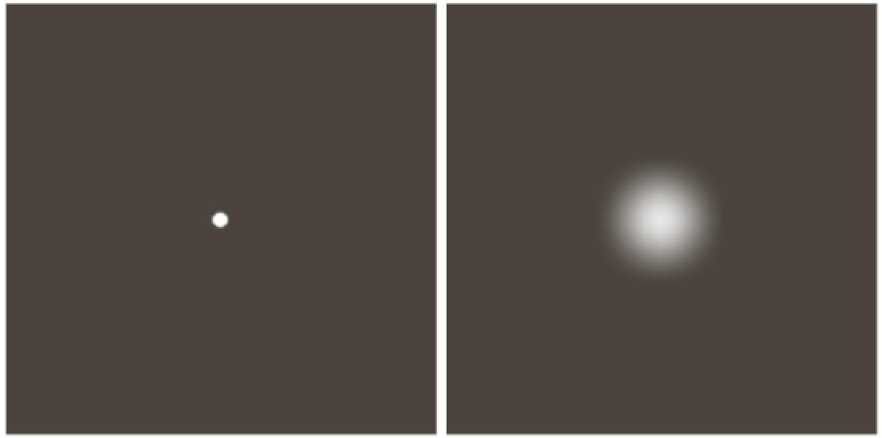
$$H(u,v) = \frac{G(u,v)}{A}$$

$G(u,v) \longrightarrow$ Observed image

$A \longrightarrow$ Impulse signal (A constant describing 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.



Slide credit: Yan Tong

Estimation by Modeling – Motion Blur

Constant velocity along x and y direction:

$$x_0(t) = \frac{at}{T} \qquad y_0(t) = \frac{bt}{T}$$

Suppose that an image $f(x, y)$ undergoes planar motion. These x_0 and y_0 are the time-varying components of motion in the *x*- and *y*-directions, respectively.

T is the duration of the exposure

The image has been displaced by a total distance *a* along x and b along y

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 $T = 1$.

Slide credit: Yan Tong

Estimation by Modeling – Motion Blur

An example of motion blur

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

Motion in both x and y direction during acquisition

Slide credit: Yan Tong

Estimation by Modeling – Motion Blur

An example of motion blur

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

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

Slide credit: Yan Tong

Estimation by Modeling – Motion Blur

Constant velocity along x and y direction:

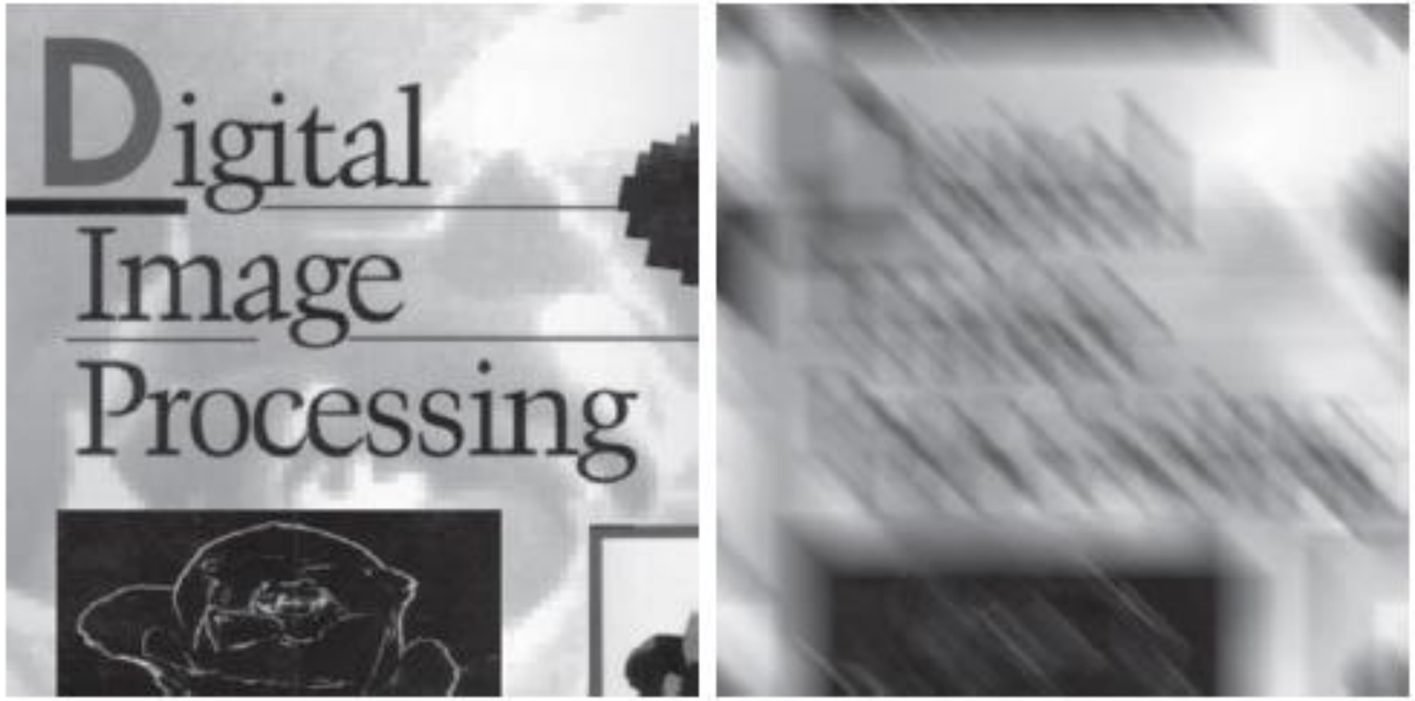
$$x_0(t) = at/T \qquad y_0(t) = bt/T$$

What is $H(u,v)$?

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

Slide credit: Yan Tong

Estimation by Modeling – Motion Blur



a b

FIGURE 5.26
(a) Original image. (b) Result of blurring using the function in Eq. (5-77) with $a = b = 0.1$ and $T = 1$.

Figure 5.26(b) is an image blurred by computing the Fourier transform of the image in Fig. 5.26(a), multiplying the transform by H(u,v) from Eq. (5-77), and taking the inverse transform. The images are of size 688 x 688 pixels, and we used $a = b = 0.1$ and $T = 1$ in Eq. (5-77).

Slide credit: Yan Tong

Image Restoration

Given the degradation system **H** and the input image **G**, recover the original image **F**

- Inverse filtering

Slide credit: Yan Tong

Inverse Filtering

Ideally:

$$G(u,v) = H(u,v)F(u,v)$$

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

In practice:

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

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

Limiting the analysis to frequencies near the origin (0,0)

Low pass filtering

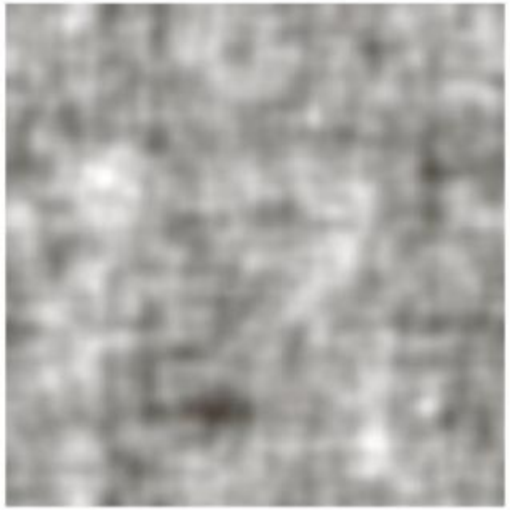
Slide credit: Yan Tong

An Example of Inverse Filtering

Original image

Degraded image

$$\frac{G(u,v)}{H(u,v)}$$



The reasons for this poor result are

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

That is, the degradation function used was

$$H(u,v) = e^{-k\left[(u + M/2)^2 + (v - N/2)^2\right]^{5/6}}$$

with $k = 0.0025$. The $M/2$ and $N/2$ constants are offset values; they center the function so that it will correspond with the centered Fourier transform

$M = N = 480$

Slide credit: Yan Tong

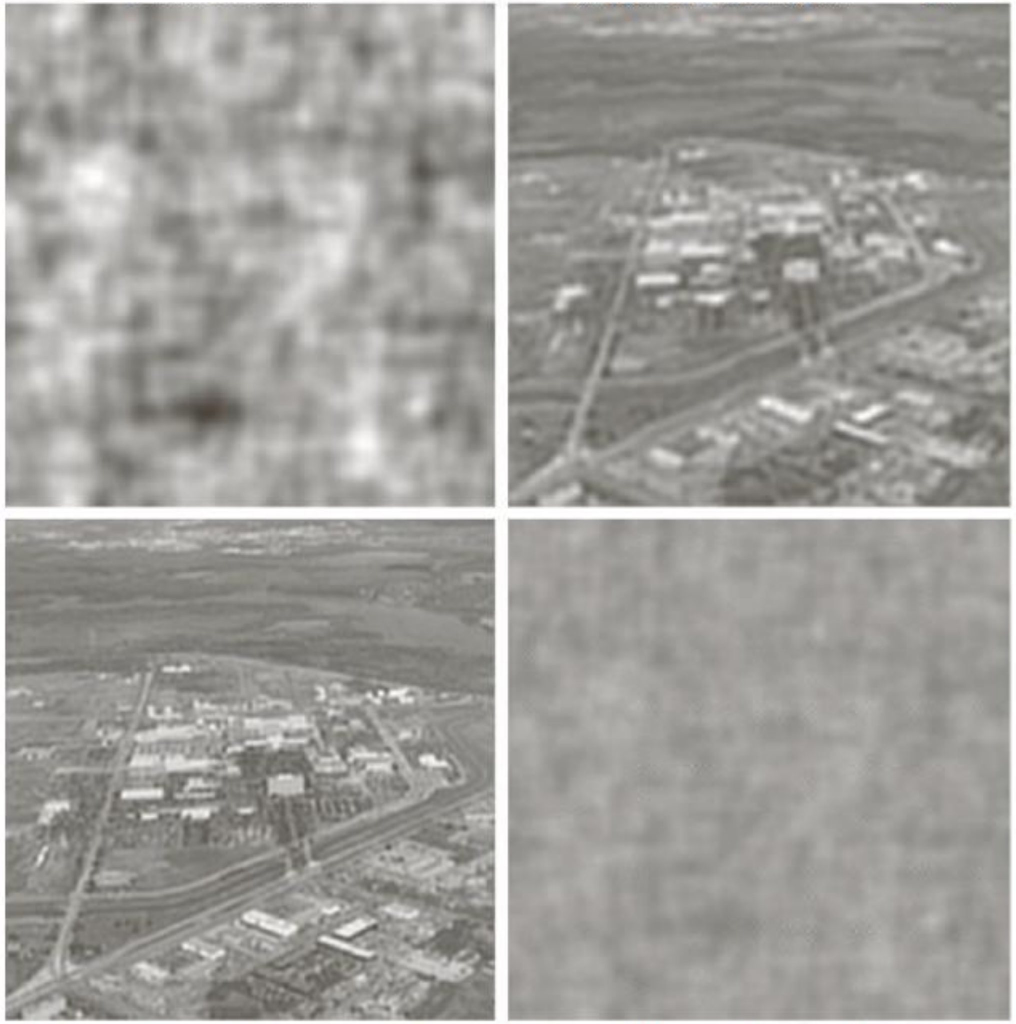
An Example of Inverse Filtering

Original image

Degraded image

$$\frac{G(u,v)}{H(u,v)}$$

Full filter **Cutoff radius = 40**



Cutoff radius = 70 **Cutoff radius = 85**

Cutting off values of the ratio G(u,v)/H(u,v) outside a radius of 40, 70, and 85, respectively

The cut off was implemented by applying to the ratio a Butterworth lowpass function of order 10.

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

n=10

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

D_0 is radius

Slide credit: Yan Tong