Digital Image Processing CS390S

Feng Jiang
2018 Spring

METROPOLITAN STATE UNIVERSITY OF DENVER

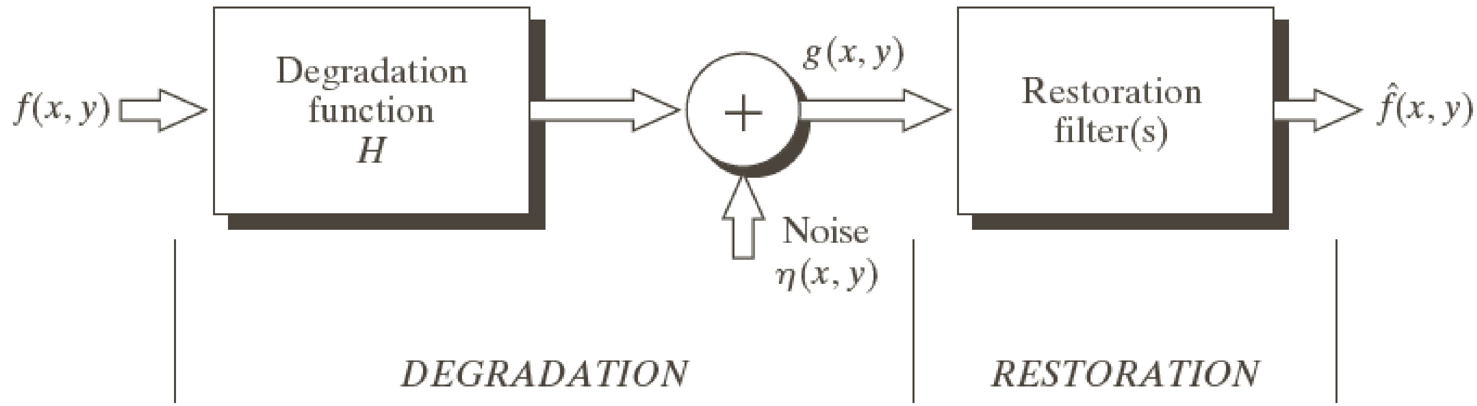
Image Restoration

- Image restoration: recover an image that has been degraded by using a prior knowledge of the degradation phenomenon.

- Model the degradation and applying the **inverse** process in order to recover the original image.

A Model of Image Degradation/Restoration Process

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



- Degradation
  - Degradation function H
  - Additive noise $\eta(x, y)$

METROPOLITAN STATE UNIVERSITY™ OF DENVER

A Model of Image Degradation/Restoration Process

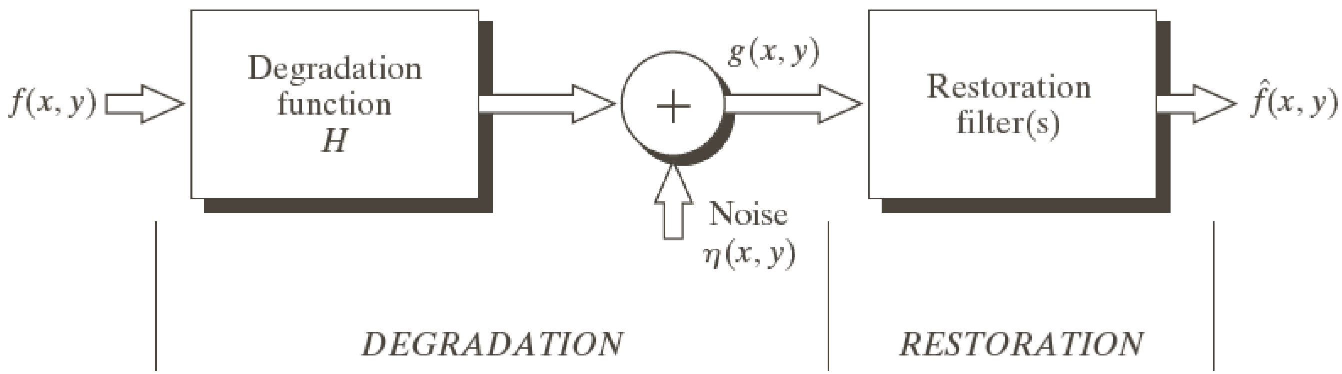


FIGURE 5.1 A model of the image degradation/ restoration process.

If H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

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

METROPOLITAN STATE UNIVERSITY™ OF DENVER

A Model of Image Degradation/Restoration Process

The model of the degraded image is given in the frequency domain by

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

Noise Sources

- The principal sources of noise in digital images arise during **image acquisition and/or transmission**

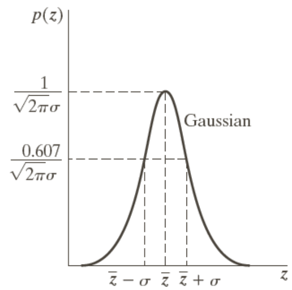
✓Image acquisition

e.g., light levels, sensor temperature, etc.

✓Transmission/environment

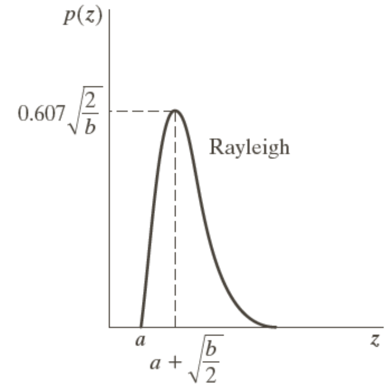
e.g., lightning or other atmospheric disturbance in wireless network

Noise Models (1)



- **Gaussian noise**
  - Mean, variance; sum of the independent noises
  - Probability of values follows Gaussian model
  - Electronic circuit noise, sensor noise due to poor illumination and/or high temperature
- **White noise**
  - **https://en.wikipedia.org/wiki/White_noise**
  - In discrete time, white noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variance;
  - Being uncorrelated in time does not restrict the values a signal can take. Any distribution of values is possible (although it must have zero DC component). Even a binary signal which can only take on the values 1 or –1 will be white if the sequence is statistically uncorrelated. Noise having a continuous distribution, such as a normal distribution, can of course be white.
  - It is often incorrectly assumed that Gaussian noise (i.e., noise with a Gaussian amplitude distribution – see normal distribution) necessarily refers to white noise, yet neither property implies the other.
  - Gaussian white noise is a good approximation of many real-world situations and generates mathematically tractable models.

Noise Models (2)



- **Rayleigh noise**

  Range imaging

➢ The intention of HDRI is to accurately represent the wide range of intensity levels found in real scenes ranging from direct **sunlight to shadows**

➢ HDRI, also called HDR (High Dynamic Range) is a feature commonly found in high-end graphics and imaging software

http://en.wikipedia.org/wiki/High_dynamic_range_imaging

http://www.webopedia.com/TERM/H/High_Dynamic_Range_Imaging.html

Range Imaging: Examples (1)

http://en.wikipedia.org/wiki/High_dynamic_range_imaging

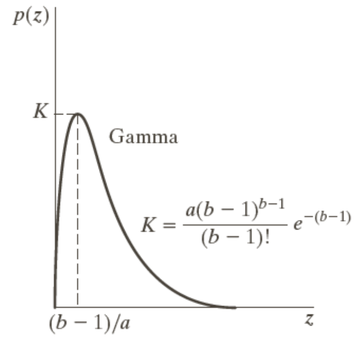
Tower Bridge in Sacramento, CA

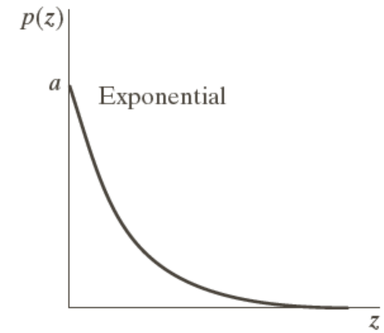
Range Imaging: Examples (2)

http://en.wikipedia.org/wiki/High_dynamic_range_imaging

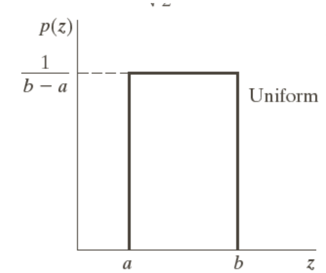
Sydney Harbour Bridge
HDRi produces greater detail and fewer shadows

Noise Models (3)

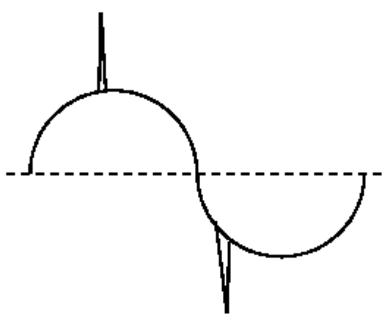




- **Erlang (gamma) noise**: Laser imaging
- **Exponential noise**: Laser imaging
- **Uniform noise**: Least descriptive; **quantization noise**



- **Impulse noise**: quick transients, such as faulty switching



Impulse Noise

METROPOLITAN STATE UNIVERSITY OF DENVER

Gaussian Noise (1)

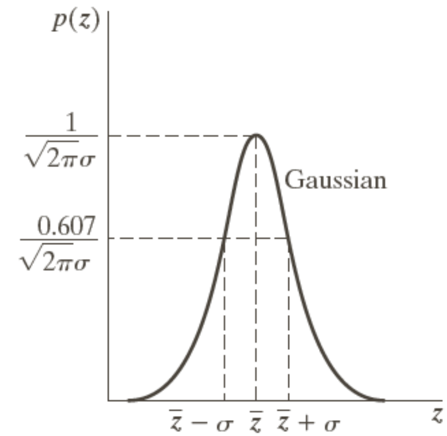
The PDF of Gaussian random variable, z, is given by

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

where, z represents intensity

$\bar{z}$ is the mean (average) value of z

σ is the standard deviation



METROPOLITAN STATE UNIVERSITY OF DENVER

Gaussian Noise (2)

The PDF of Gaussian random variable, z, is given by

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

- 70% of its values will be in the range

$$\left[(\mu-\sigma),(\mu+\sigma)\right]$$

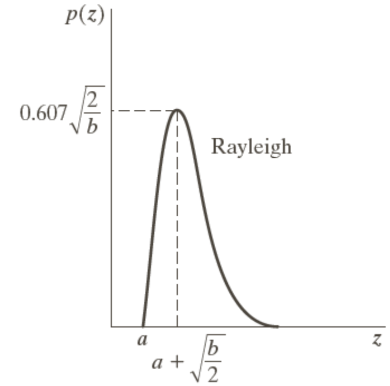
- 95% of its values will be in the range

$$\left[(\mu-2\sigma),(\mu+2\sigma)\right]$$

Rayleigh Noise

The PDF of Rayleigh noise is given by

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

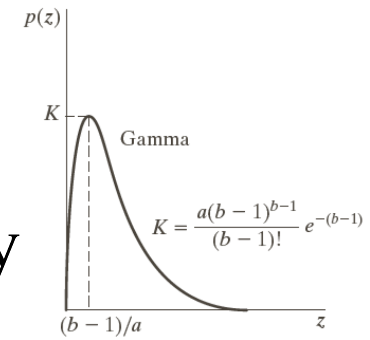


The mean and variance of this density are given by

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

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

Erlang (Gamma) Noise



The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

$$\bar{z} = b / a$$

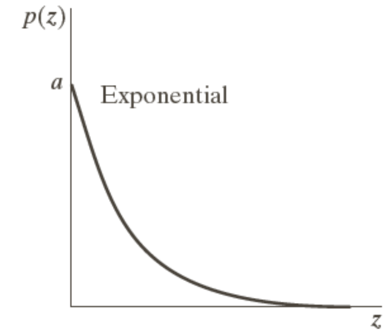
$$\sigma^2 = b / a^2$$

METROPOLITAN STATE UNIVERSITY™ OF DENVER

Exponential Noise

The PDF of exponential noise is given by

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



The mean and variance of this density are given by

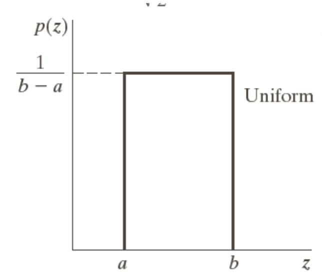
$$\bar{z} = 1/a$$

$$\sigma^2 = 1/a^2$$

Uniform Noise

The PDF of uniform noise is given by

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



The mean and variance of this density are given by

$$\bar{z}=(a+b)/2$$

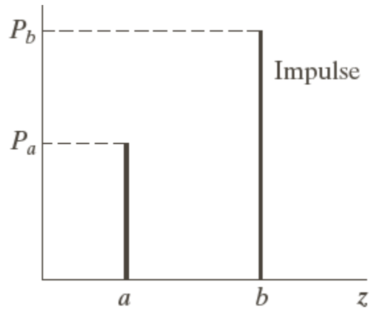
$$\sigma^2=(b-a)^2/12$$

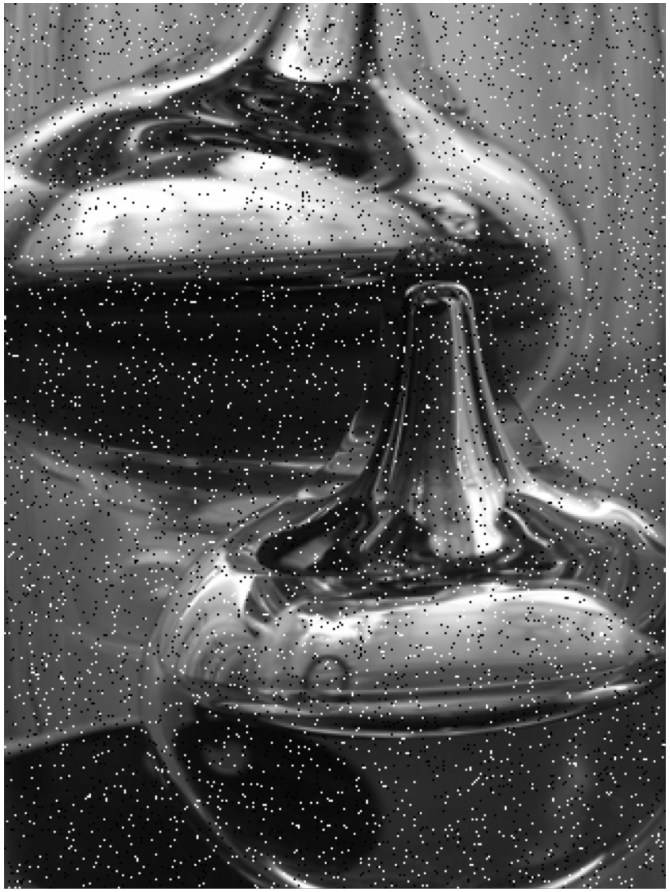
Impulse (Salt-and-Pepper) Noise

The PDF of (bipolar) impulse noise is given by

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

if $b > a$, gray-level b will appear as a light dot, while level a will appear like a dark dot.





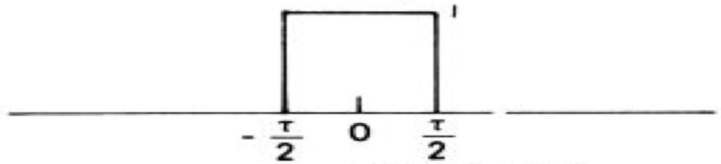
Day 12

- Review “frequency transform and filtering”
- Nosie reduction / image restoration

Time Function

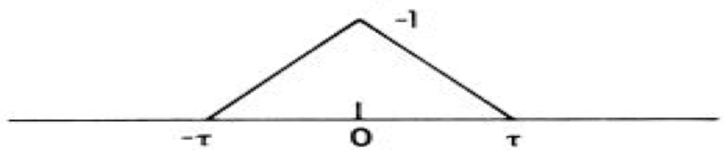
Boxcar

$$G(t) = \begin{cases} 1, & |t| < \tau/2 \\ 0, & |t| > \tau/2 \end{cases}$$



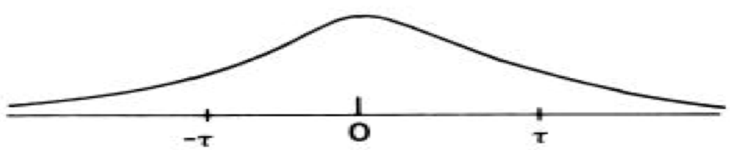
Triangle

$$G(t) = \begin{cases} 1-|t|/\tau, & |t| < \tau \\ 0, & |t| > \tau \end{cases}$$



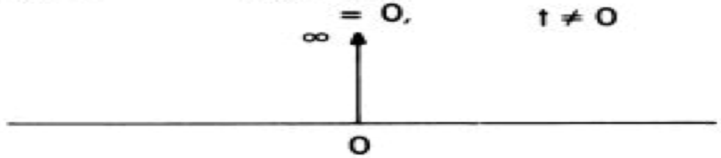
Gaussian

$$G(t) = e^{-\frac{1}{2}t^2}$$



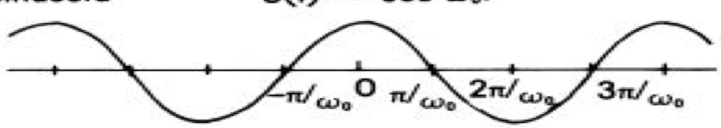
Impulse

$$G(t) = \delta(t)$$
$$= 0, \quad t \neq 0$$



Sinusoid

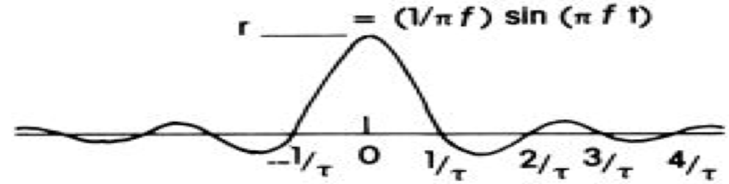
$$G(t) = \cos \omega_0 t$$



Frequency Function

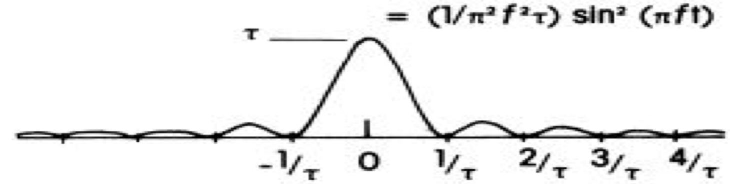
Sinc

$$S(f) = \tau \operatorname{sinc}(f\tau)$$
$$= (1/\pi f) \sin(\pi f t)$$



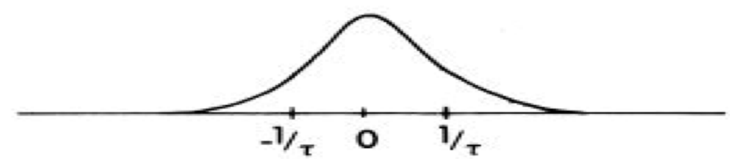
Sinc²

$$S(f) = \tau \operatorname{sinc}^2(ft)$$
$$= (1/\pi^2 f^2 \tau) \sin^2(\pi f t)$$



Gaussian

$$S(f) = \tau(2\pi)^{1/2} e^{-(\pi f \tau)^2}$$

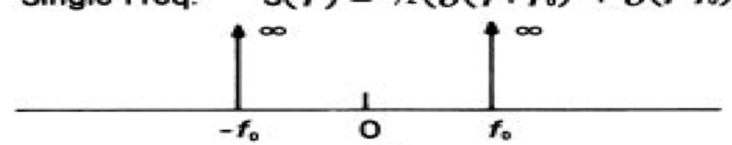


DC Shift

$$S(f) = 1$$

Single Freq.

$$S(f) = \tfrac{1}{2}(\delta(f+f_0) + \delta(f-f_0))$$



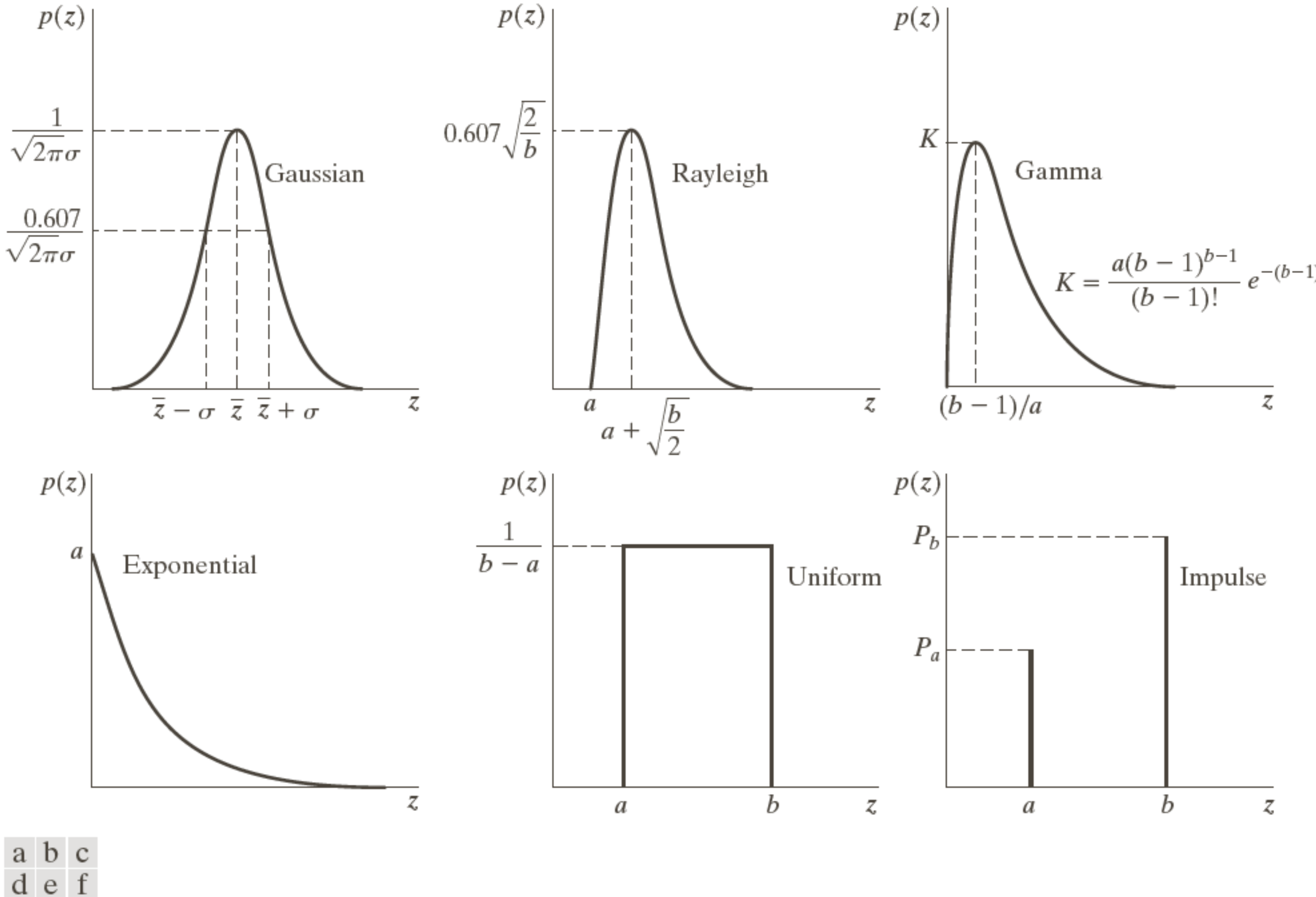


FIGURE 5.2 Some important probability density functions.

clockwise direction: exponential noise, gamma noise, gaussian noise, rayleigh noise, salt and pepper noise, uniform noise.

clockwise direction: exponential noise, gamma noise, gaussian noise, rayleigh noise, salt and pepper noise, uniform noise.

Examples of Noise: Original Image

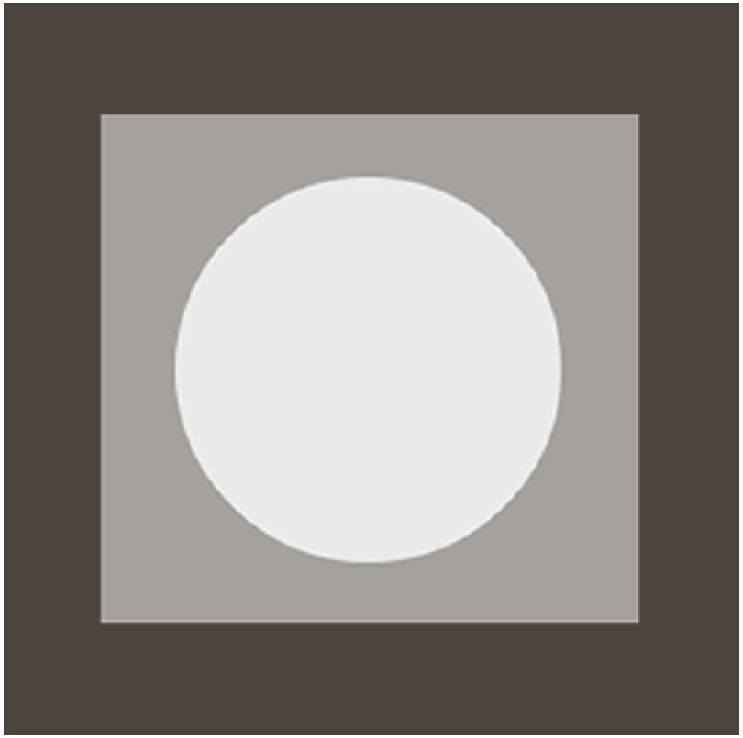


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

Examples of Noise: Noisy Images(1)

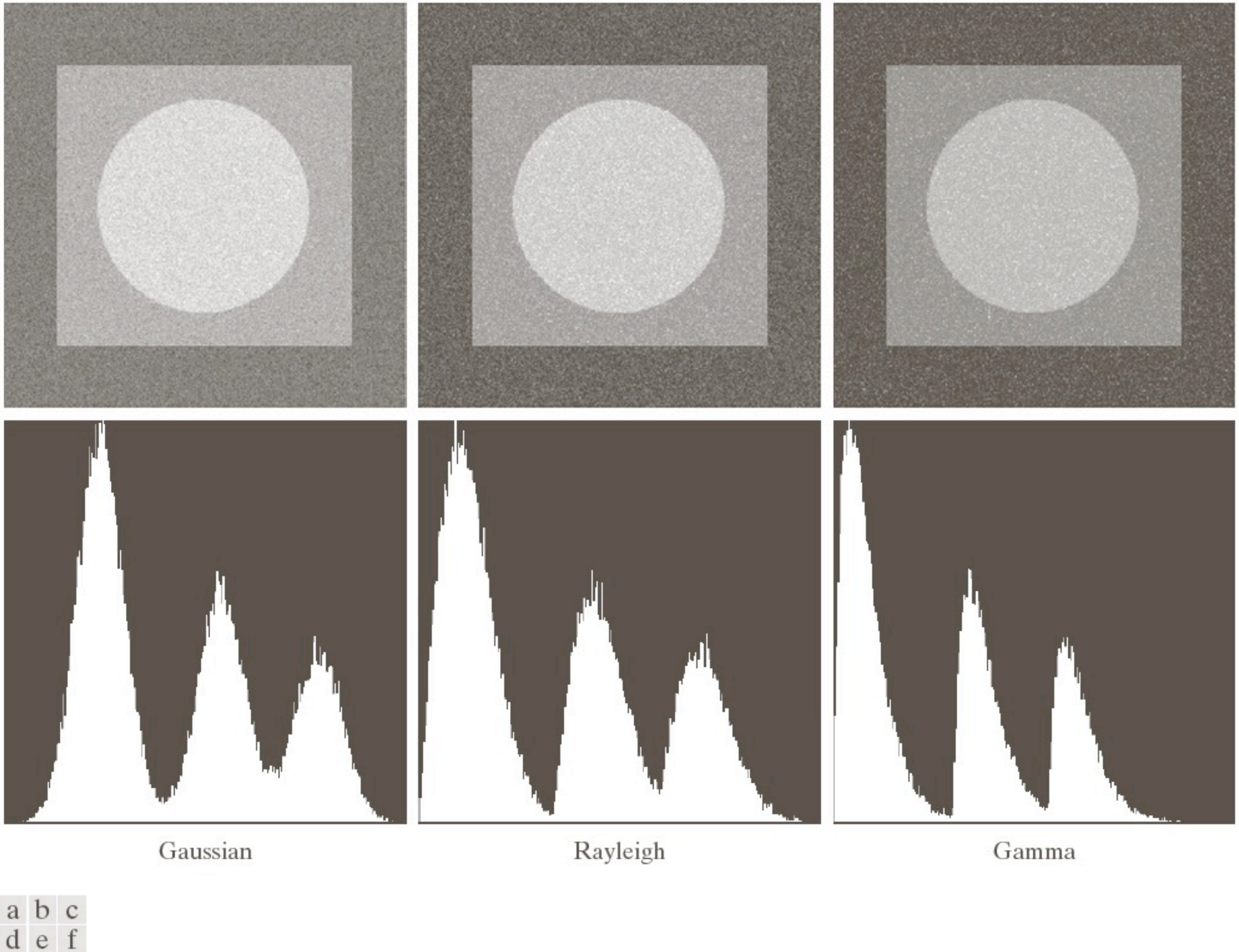


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Examples of Noise: Noisy Images(2)

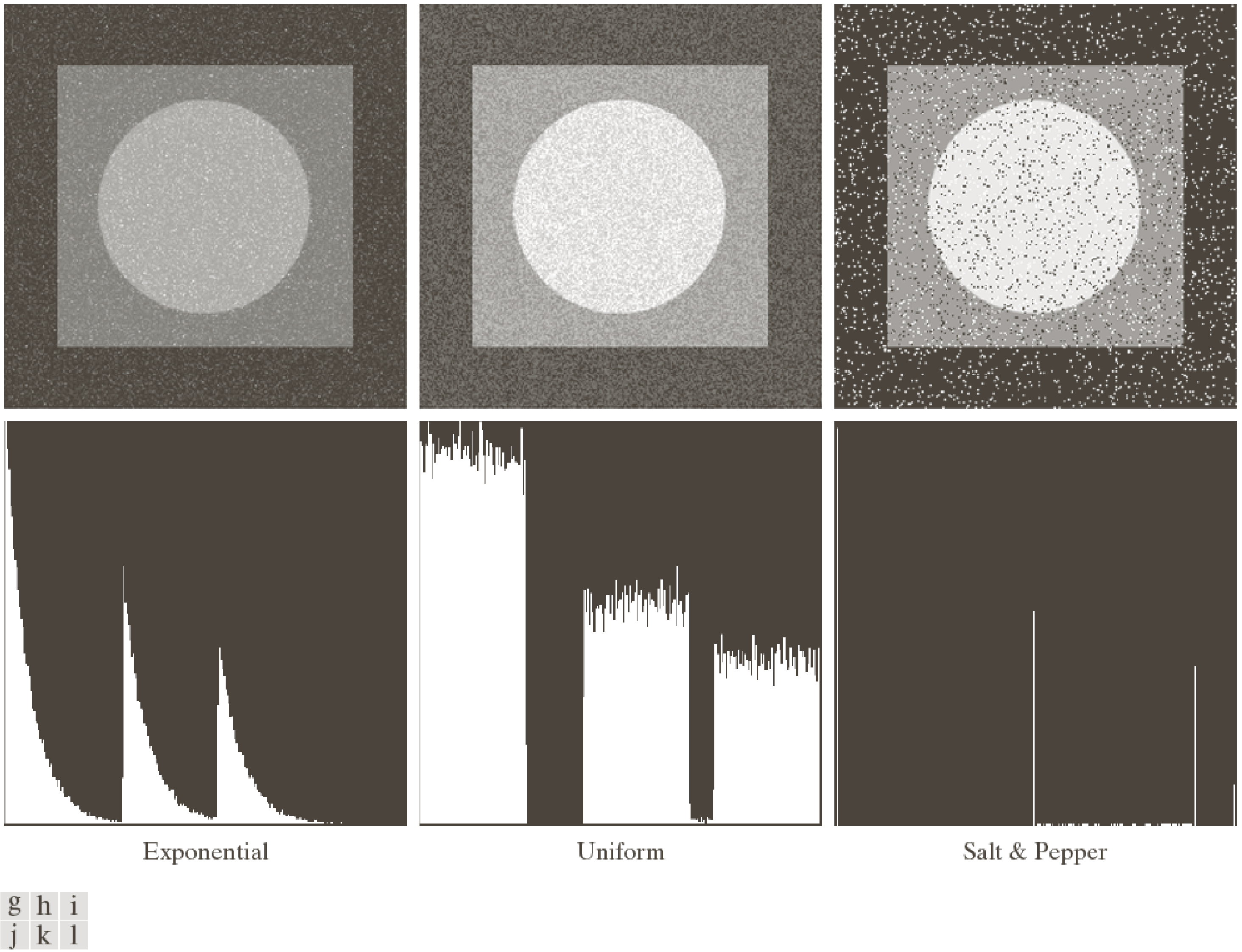


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Estimation of Noise Parameters

The shape of the histogram identifies the closest PDF match

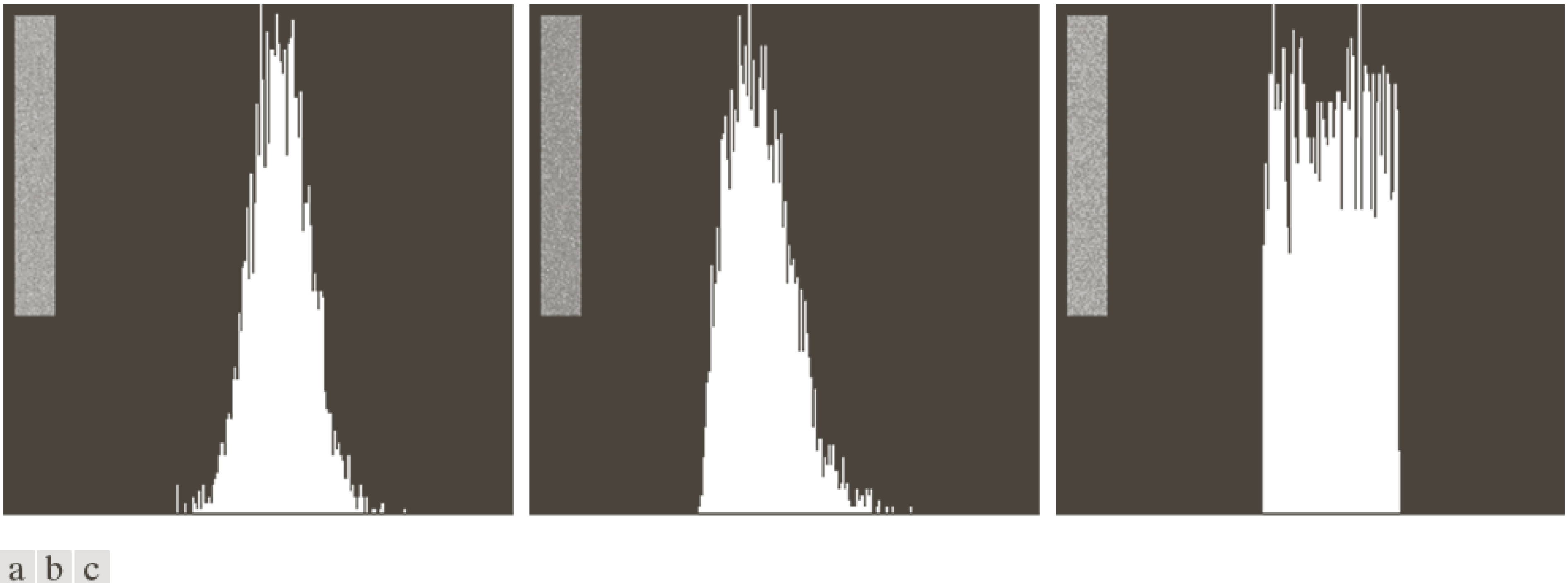


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Restoration in the Presence of Noise Only – Spatial Filtering

Noise model without degradation

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

and

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

e.g. Periodic noise fixed frequency

Spatial Filtering: Mean Filters (1)

Let S_{xy} represent the set of coordinates in a rectangle subimage window of size $m \times n$, centered at (x, y).

Arithmetic mean filter

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

Spatial Filtering: Mean Filters (2)

Geometric mean filter

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

Generally, a geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process

Spatial Filtering: Mean Filters (2)

Geomean()

Syntax

```
m = geomean(x)
geomean(X,dim)
```

Description

`m = geomean(x)` calculates the geometric mean of a sample. For vectors, `geomean(x)` is the geometric mean of the elements in x. For matrices, `geomean(X)` is a row vector containing the geometric means of each column. For N-dimensional arrays, `geomean` operates along the first nonsingleton dimension of X.

`geomean(X,dim)` takes the geometric mean along the dimension `dim` of X.

The geometric mean is

$$m = \left[\prod_{i=1}^{n} x_i\right]^{\frac{1}{n}}$$

The arithmetic mean is greater than or equal to the geometric mean.

a=rand(5,1)

a = 0.8147 0.9058 0.1270 0.9134 0.6324

geomean(a)

ans = 0.5581

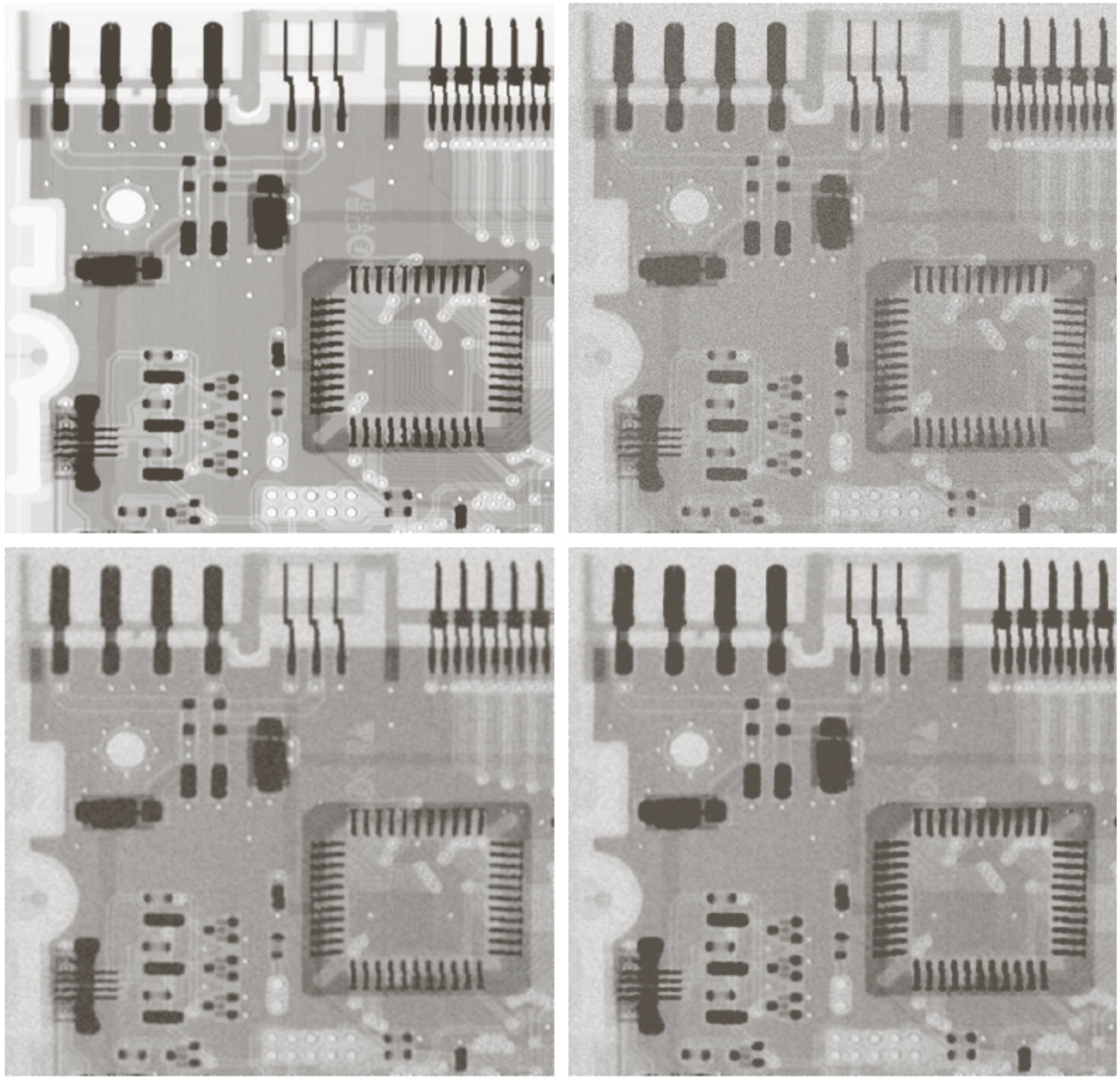
mean(a)

ans = 0.6786

Spatial Filtering: Example (1)

a	b
c	d

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

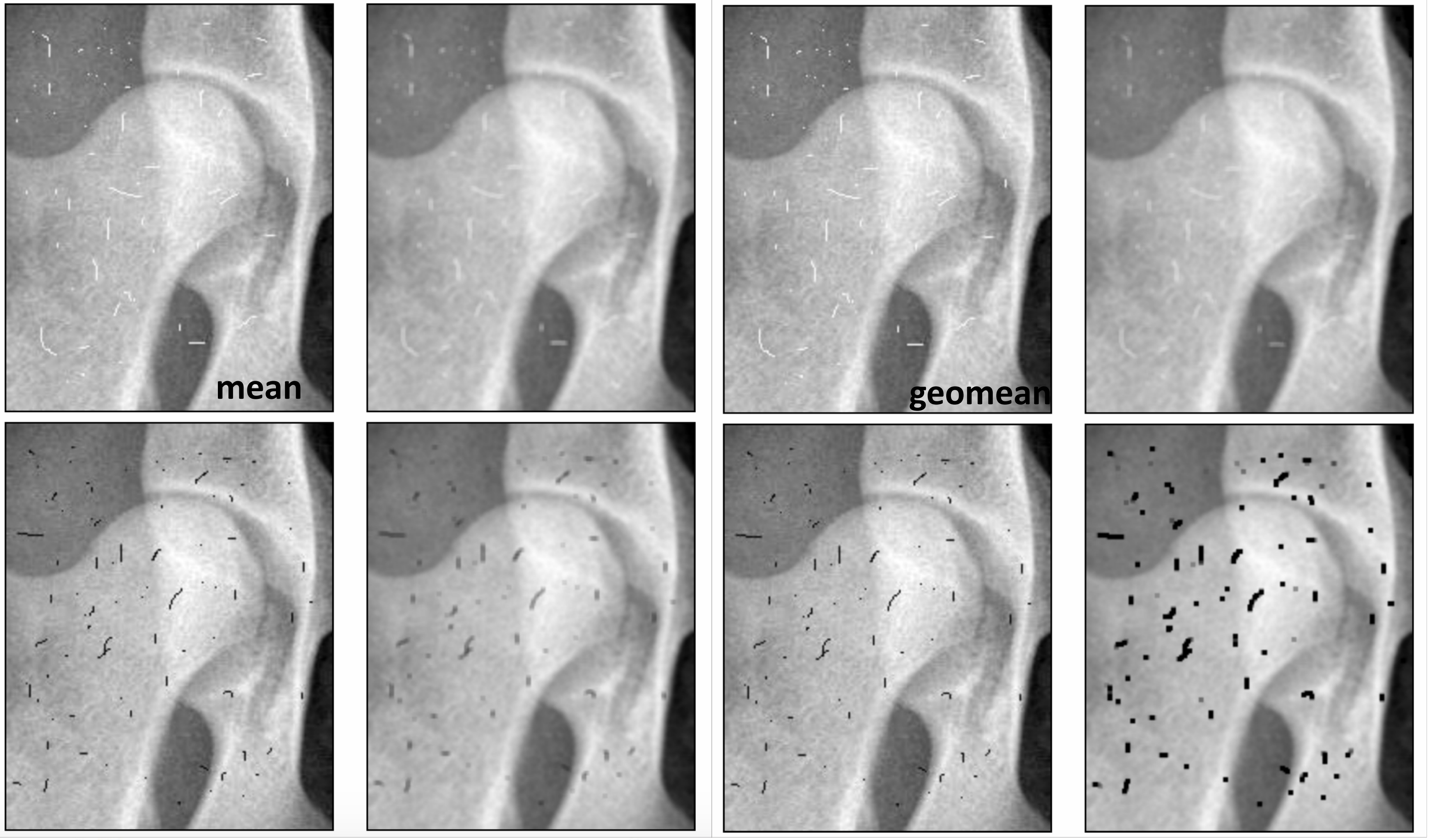


The **geometric mean** filter is better at removing Gaussian type noise and preserving edge features than the **arithmetic mean** filter.

Spatial Filtering: Example (2)

- https://www.digimizer.com/manual/m-image-filtermean.php
- https://www.digimizer.com/manual/m-image-filtergeomean.php

Spatial Filtering: Example (2)



Spatial Filtering: Mean Filters (3)

$$H_n = \frac{\sum_{i=1}^{n} x_i f_i}{\sum_{i=1}^{n} f_i} = \frac{1}{\frac{\frac{1}{x_1}m_1 + \frac{1}{x_2}m_2 + \cdots + \frac{1}{x_n}m_n}{m_1 + m_2 + \cdots + m_n}} = \frac{1}{\frac{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}}{n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Harmonic mean filter

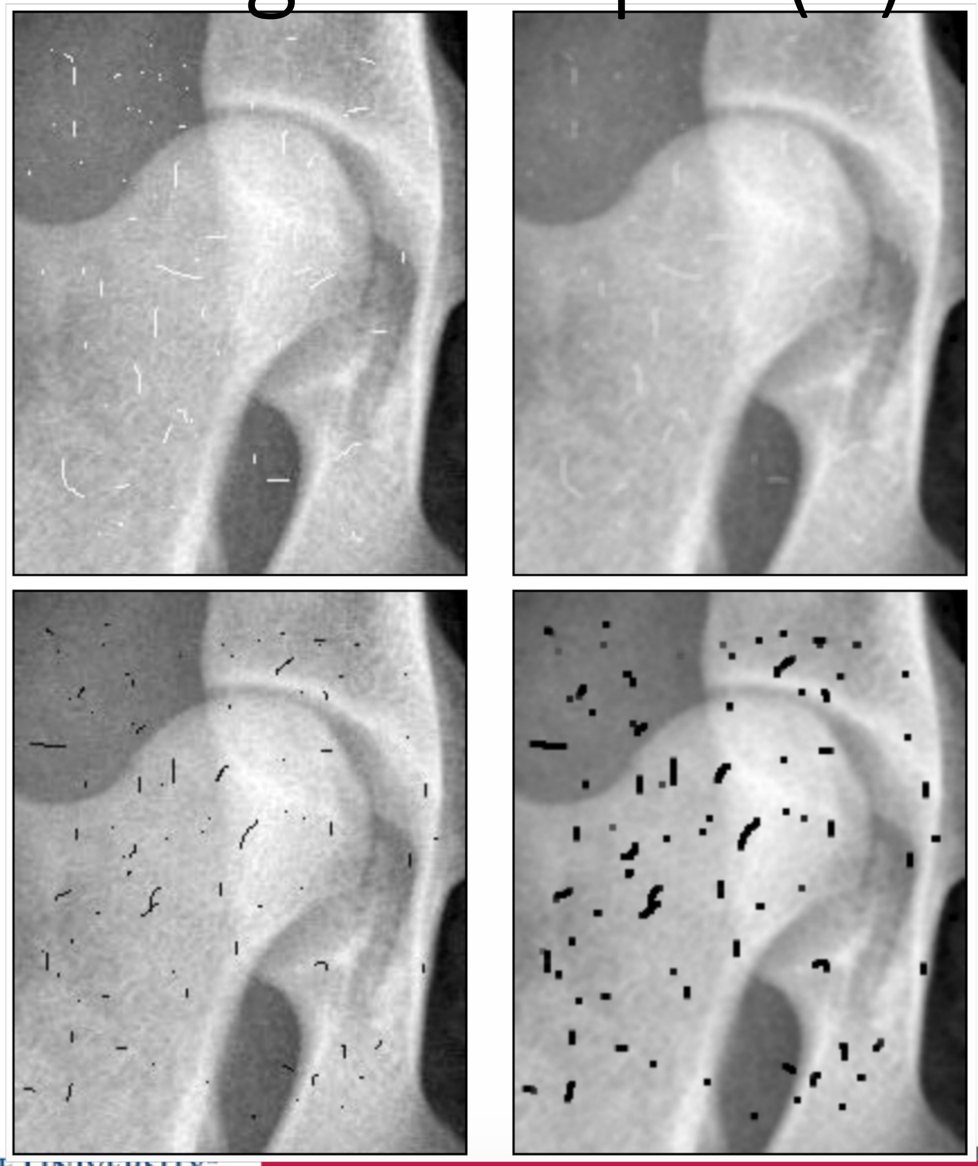
$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

"reciprocal of averaging of reciprocal"

It works **well for salt noise**, but fails for pepper noise.

It does well also with other types of noise like Gaussian noise.

Spatial Filtering: Example (2)



METROPOLITAN STATE UNIVERSITY OF DENVER

Spatial Filtering: Order-Statistic Filters (1)

Median filter

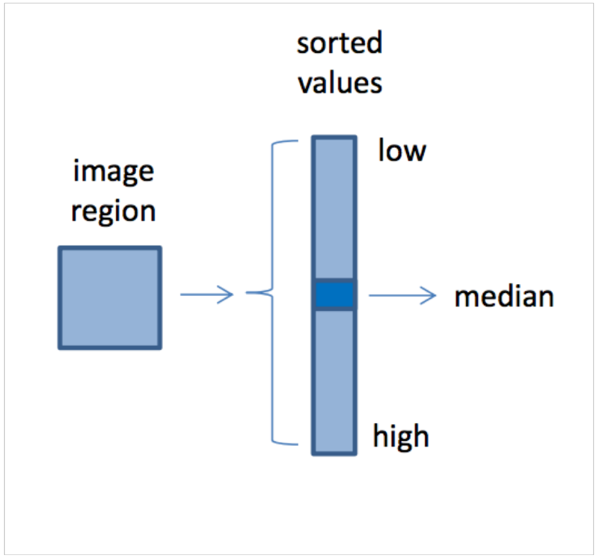
$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Max filter

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

Min filter

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

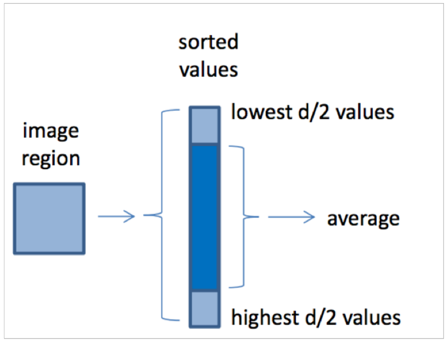


Spatial Filtering: Order-Statistic Filters (2)

Midpoint filter

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

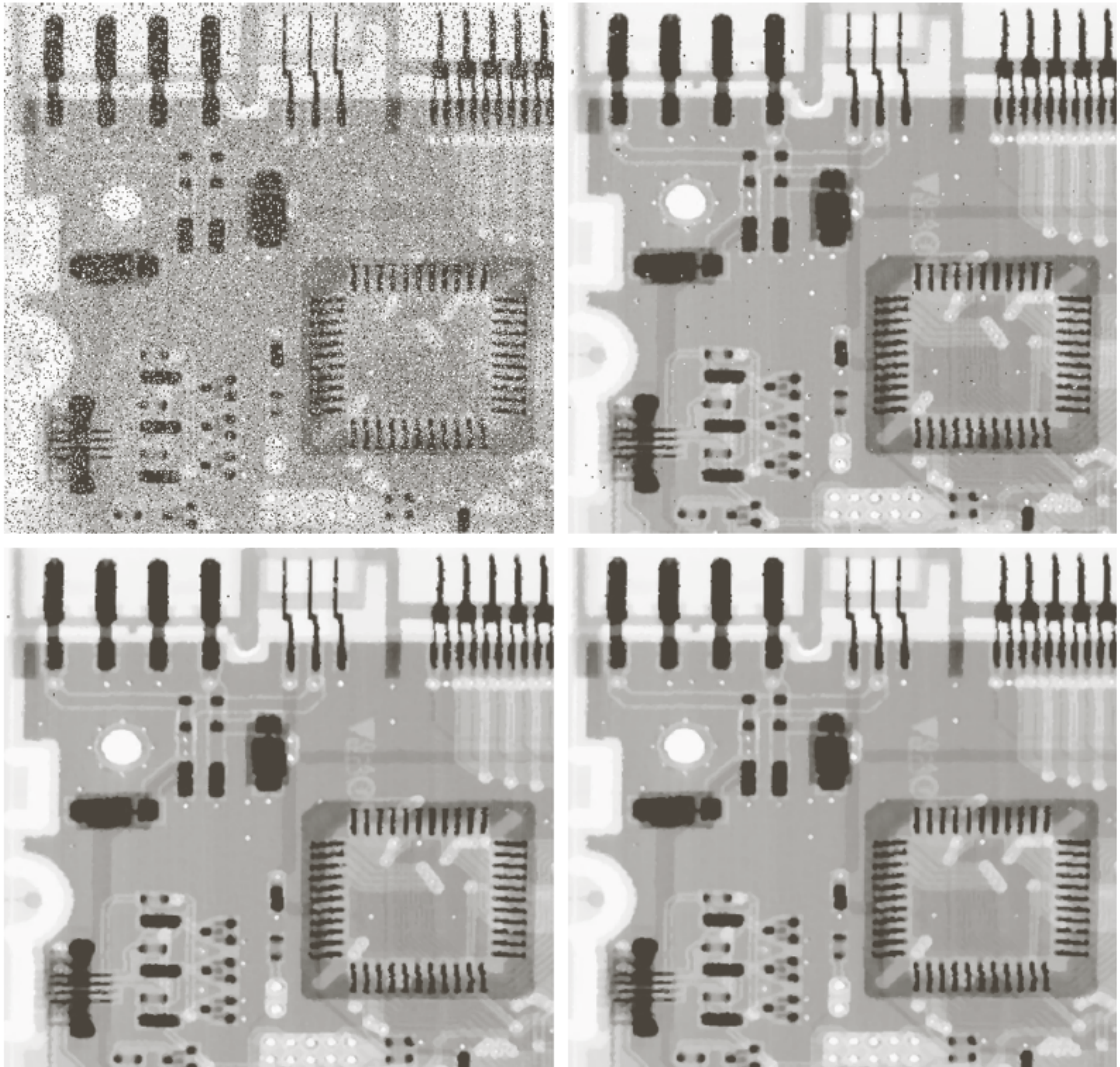
Spatial Filtering: Order-Statistic Filters (3)



Alpha-trimmed mean filter

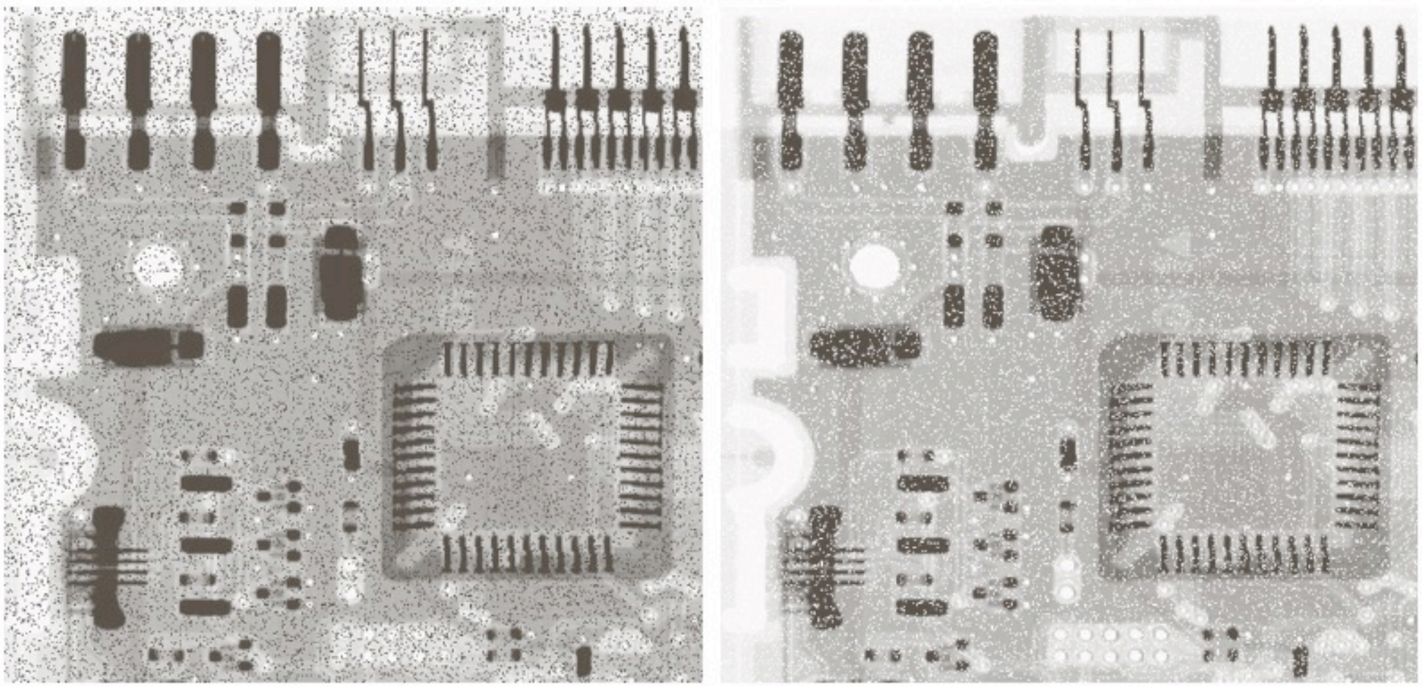
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} \{g_r(s,t)\}$$

We delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(s,t)$ in the neighborhood S_{xy}. Let $g_r(s,t)$ represent the remaining $mn - d$ pixels.



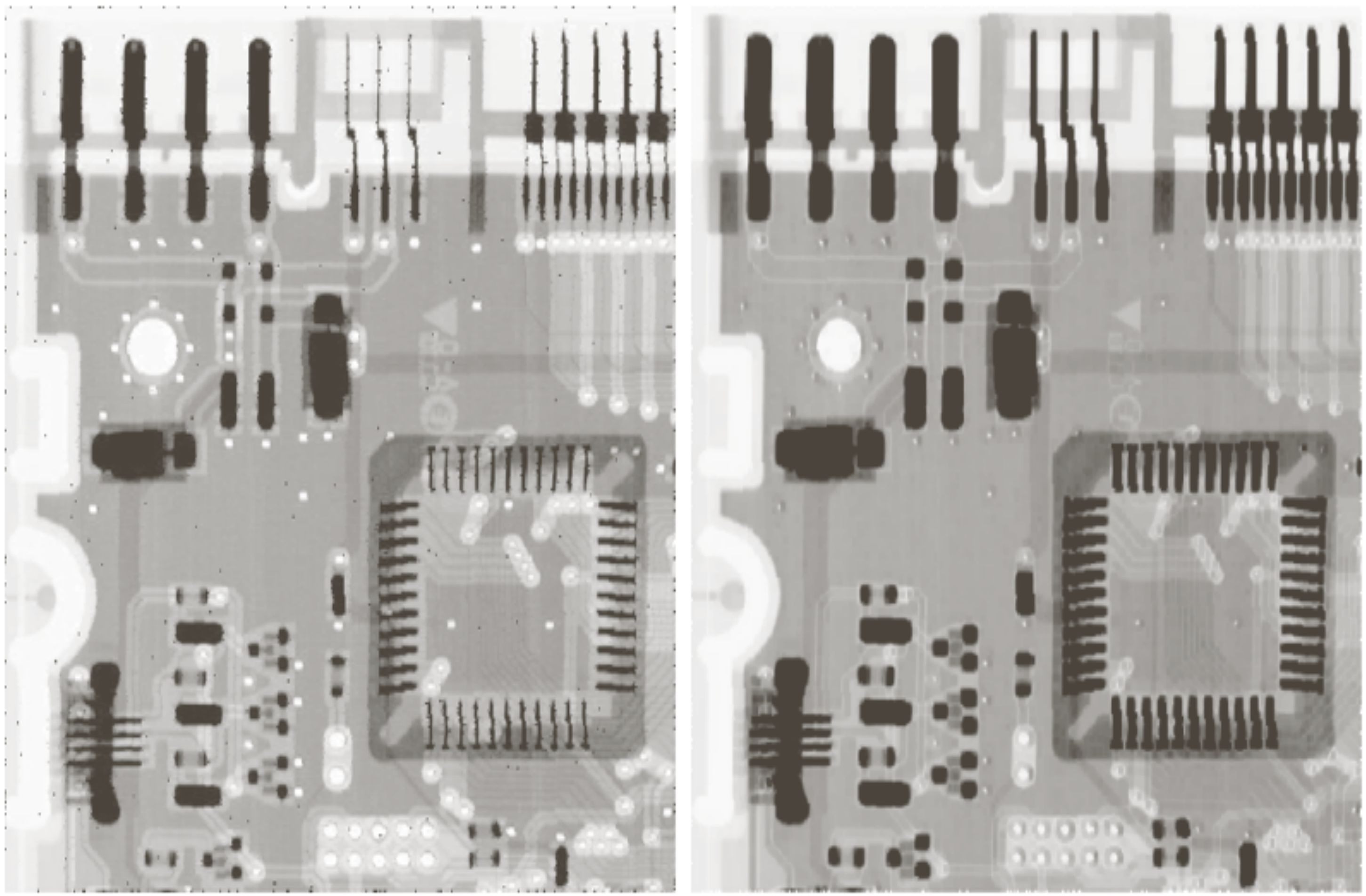
a	b
c	d

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



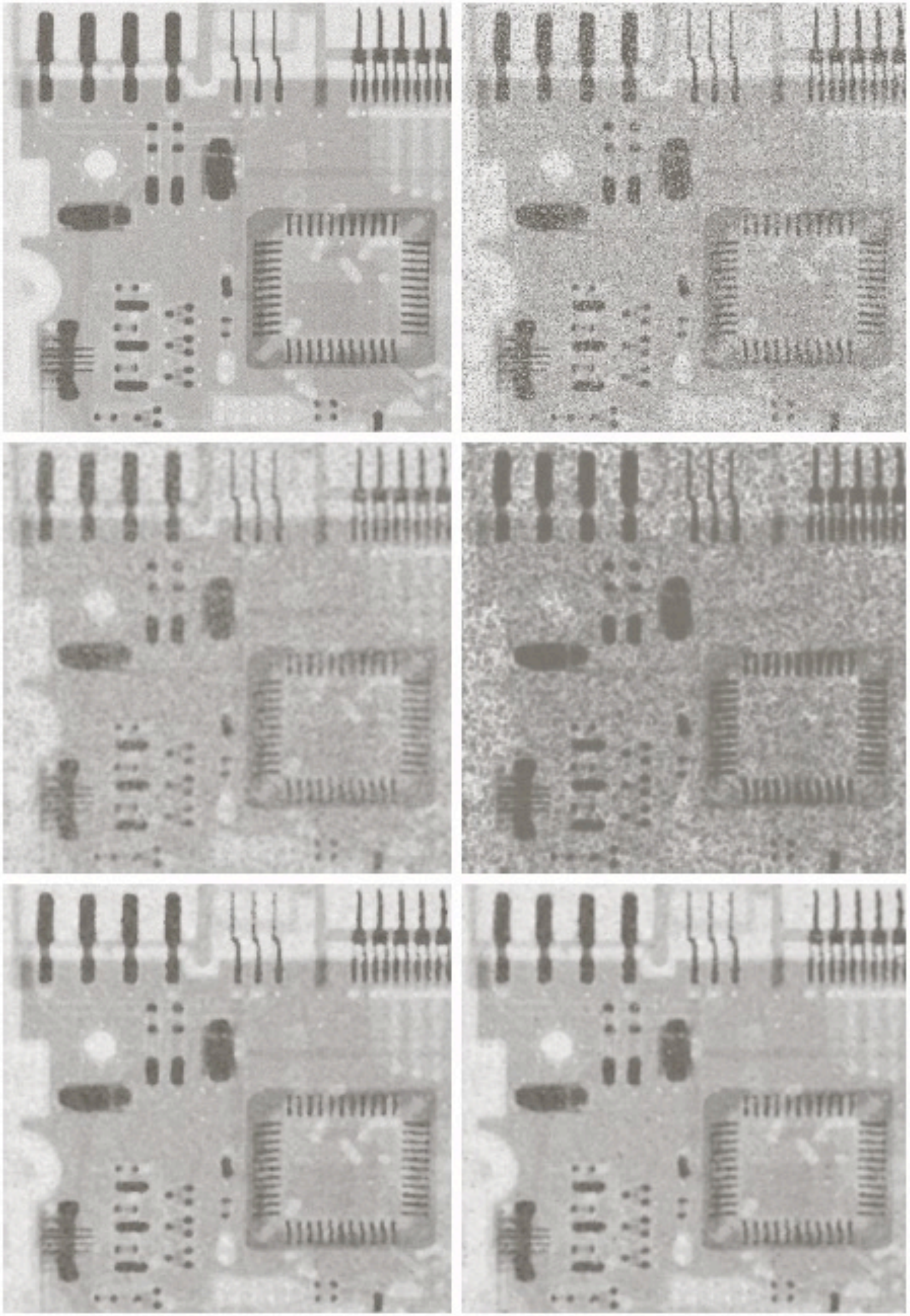
a	b
c	d

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 contra-harmonic filter of order 1.5.



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.



a b
c d
e f

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

Spatial Filtering: Adaptive Filters (1)

Adaptive filters

The behavior **changes based on statistical characteristics of the image** inside the filter region defined by the mxn rectangular window.

The performance is superior to that of the filters discussed

Adaptive Filters:

Adaptive, Local Noise Reduction Filters (1)

S_{xy}: local region

The response of the filter at the center point (x,y) of S_{xy} is based on four quantities:

(a) $g(x, y)$, the value of the noisy image at (x, y);

(b) σ_η^2, the variance of the noise corrupting $f(x, y)$ to form $g(x, y)$;

(c) m_L, the local mean of the pixels in S_{xy};

(d) σ_L^2, the local variance of the pixels in S_{xy}.

Adaptive Filters:

Adaptive, Local Noise Reduction Filters (2)

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

The behavior of the filter:

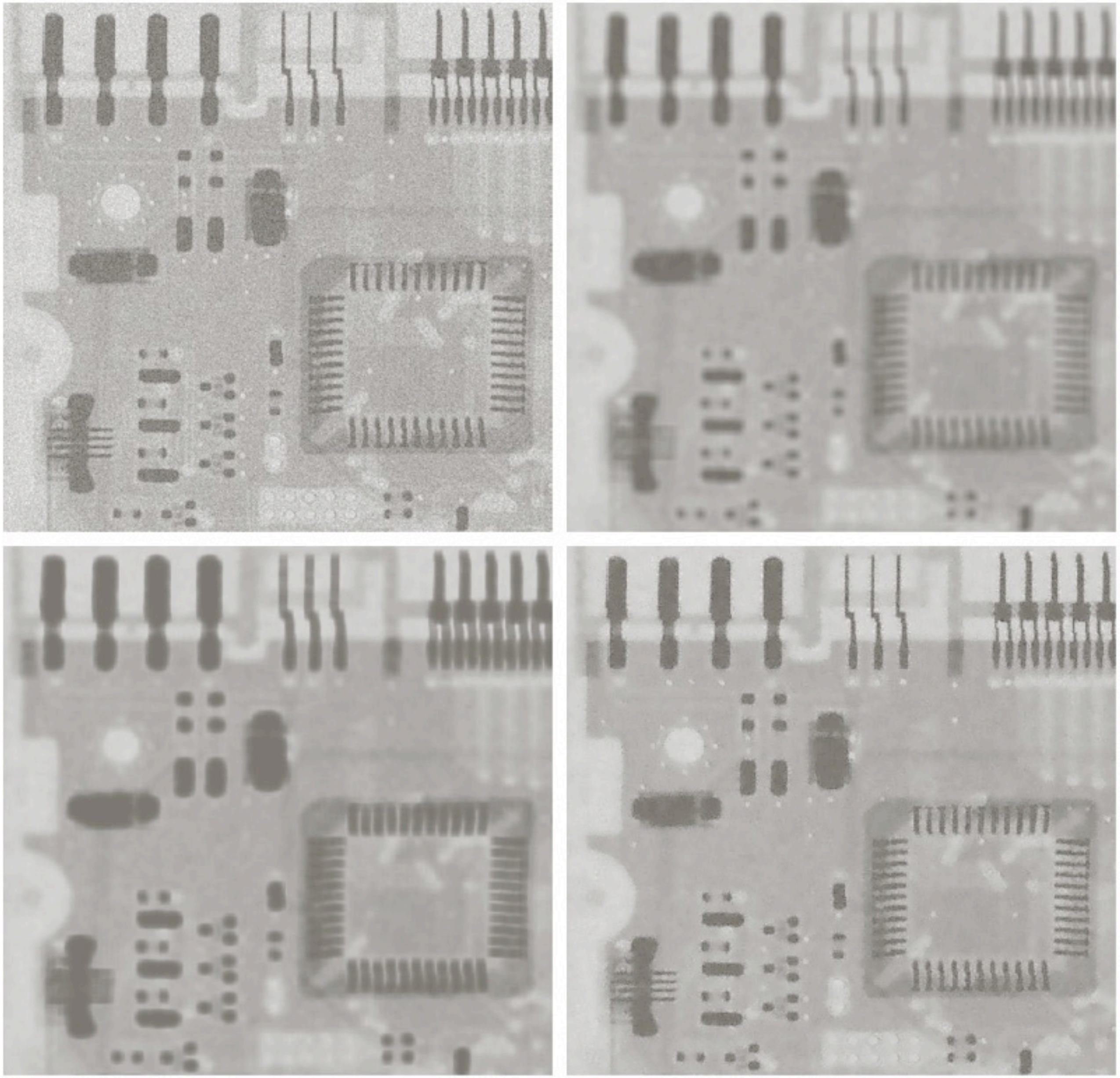
(a) if σ_η^2 is zero, the filter should return simply the value of $g(x, y)$.

(b) if the local variance is high relative to σ_η^2, the filter should return a value close to $g(x, y)$;

(c) if the two variances are equal, the filter returns the arithmetic mean value of the pixels in S_{xy}.

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 Filters:

Adaptive Median Filters (1)

The notation:

$$z_{\min} = \text{minimum intensity value in } S_{xy}$$

$$z_{\max} = \text{maximum intensity value in } S_{xy}$$

$$z_{\text{med}} = \text{median intensity value in } S_{xy}$$

$$z_{xy} = \text{ intensity value at coordinates } (x, y)$$

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

Adaptive Filters:

Adaptive Median Filters (1)

The adaptive median-filtering works in two stages:

Stage A:

$A1 = z_{\text{med}} - z_{\min}; \quad A2 = z_{\text{med}} - z_{\max}$

if A1>0 and A2<0, go to stage B

Else increase the window size

if window size $\leq S_{\max}$, repeat stage A; Else output z_{med}

Stage B:

$B1 = z_{xy} - z_{\min}; \quad B2 = z_{xy} - z_{\max}$

if B1>0 and B2<0, output z_{xy}; Else output z_{med}

Adaptive Filters:

Adaptive Median Filters (1)

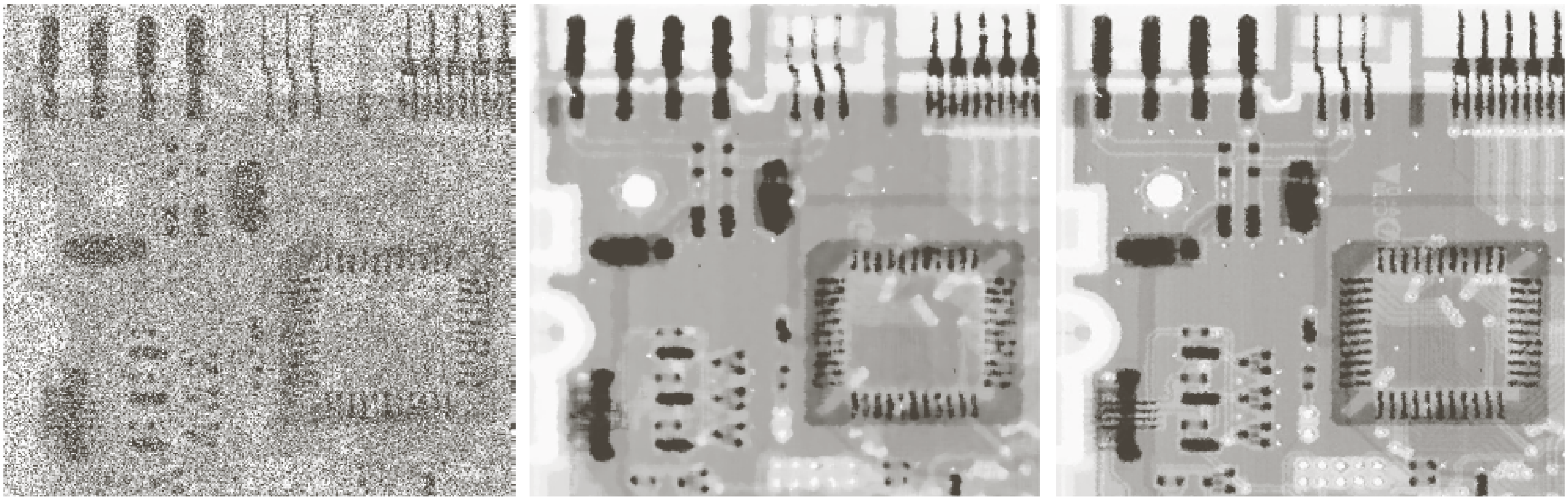


FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Periodic Noise

- Periodic noise in an image arises typically from **electrica**l or electro**mechanical** interference during image acquisition.

- It is a type of spatially dependent noise

- Periodic noise can be reduced significantly via frequency domain filtering

METROPOLITAN STATE UNIVERSITY OF DENVER

Periodic Noise Reduction by Frequency Domain Filtering

The basic idea

Periodic noise appears as **concentrated bursts of energy** in the Fourier transform, at locations corresponding to the frequencies of the periodic interference

Approach

A **selective filter** is used to isolate the noise

Perspective Plots of Bandreject Filters

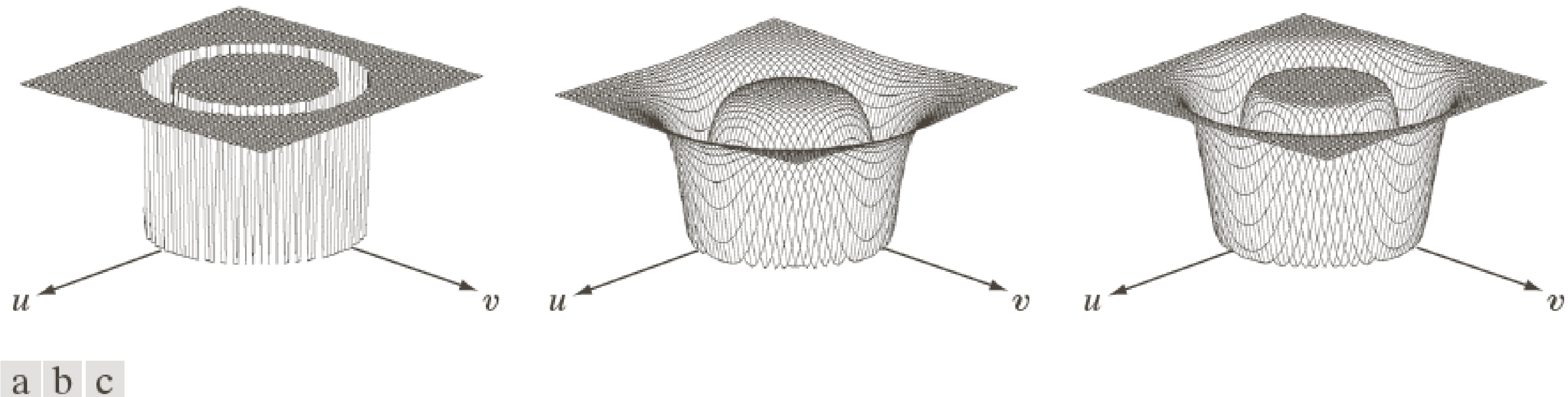
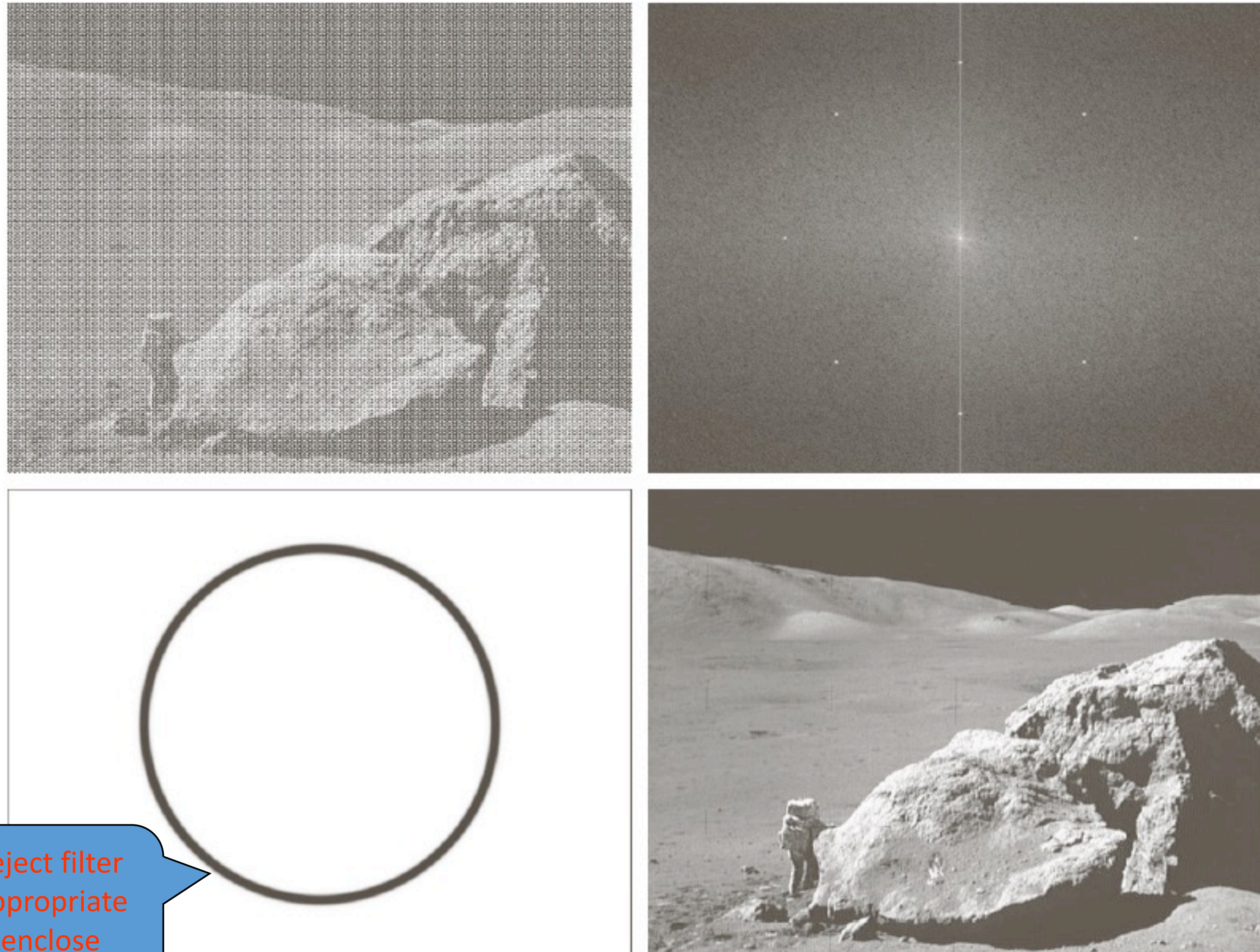


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

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.)



A Butterworth bandreject filter of order 4, with the appropriate radius and width to enclose completely the noise impulses

Perspective Plots of Notch Filters

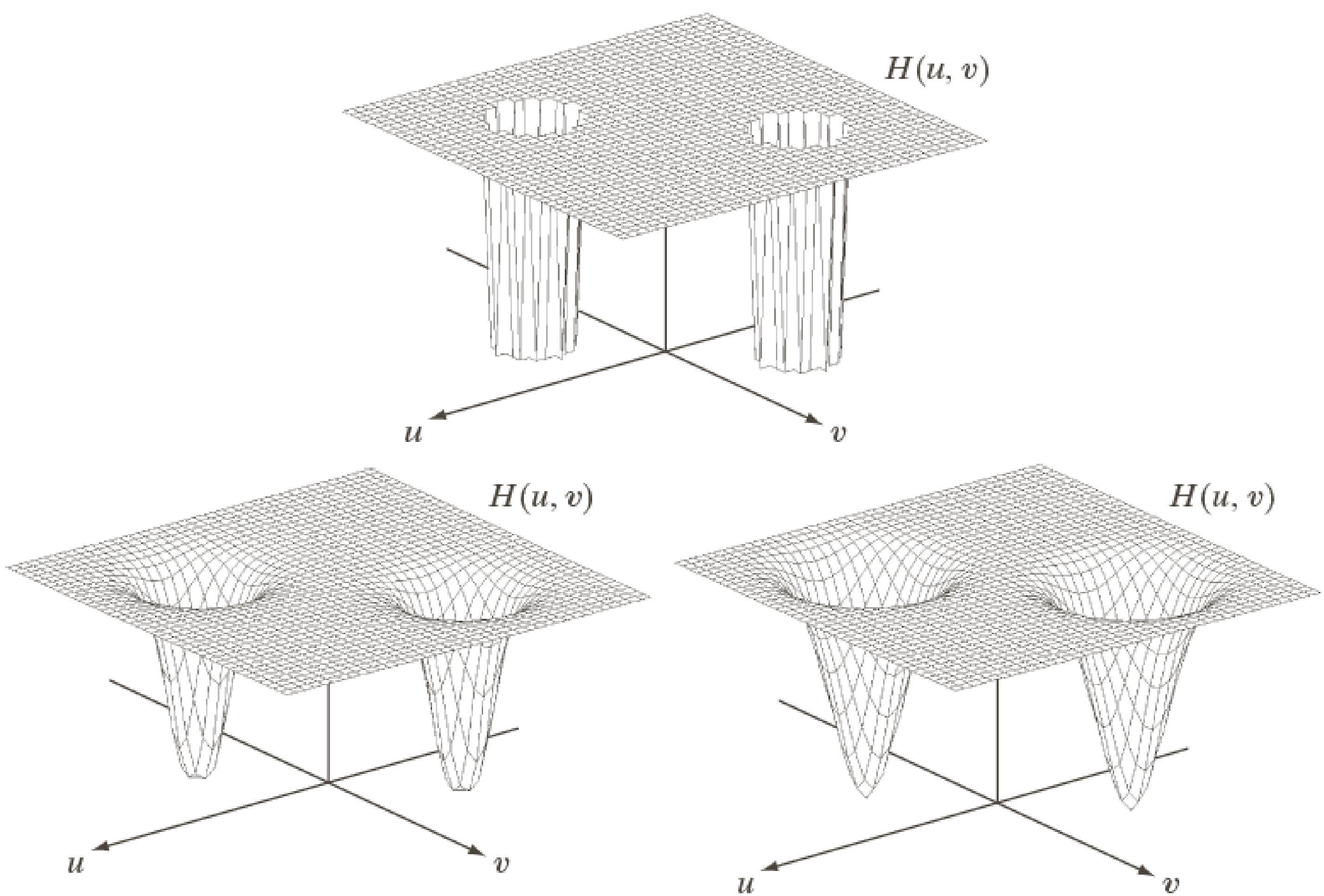
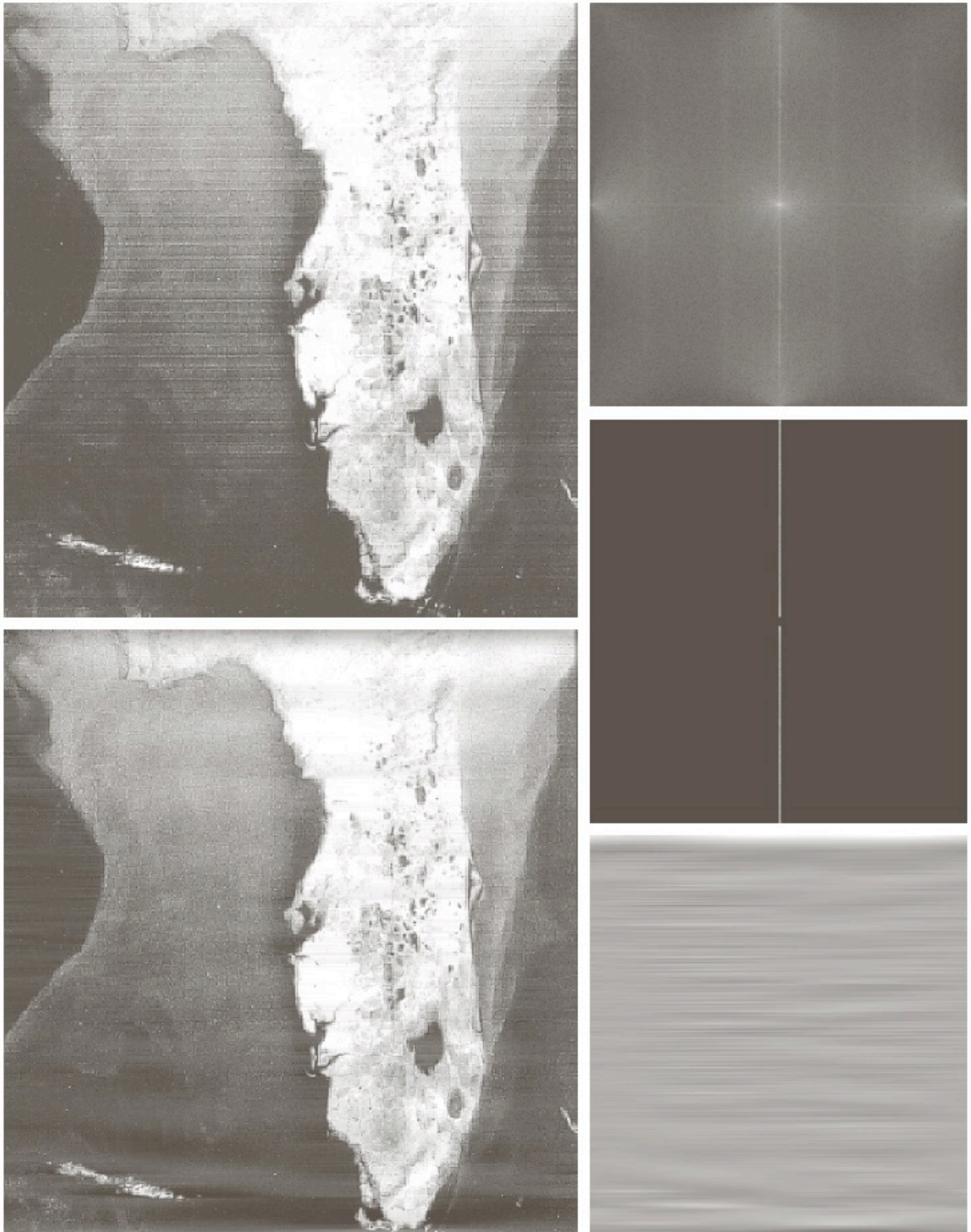


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



a b c e d

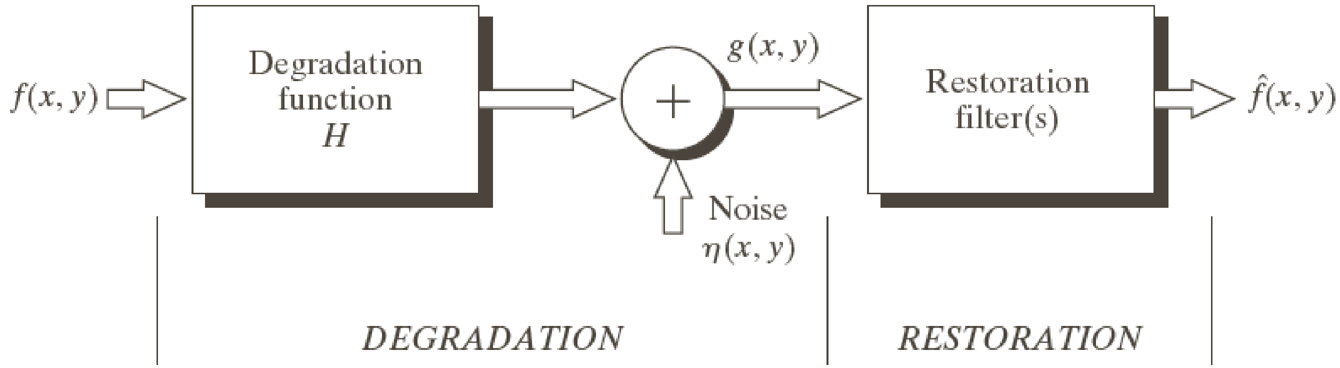
FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Exercise

- Select one of your own image with lots of edges
- Plot the frequency transform output
- Perform notch filtering in frequency and perform inverse transform
- Analyze your result
- You could define any information as “noise”
- Use fft2_show.m as an example code

Linear, Position-Invariant Degradations

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



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

Linear, Position-Invariant Degradations

H is linear

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

f_1 and f_2 are any two input images.

An operator having the input-output relationship

$g(x,y)=H\left[f(x,y)\right]$ is said to be position invariant

if

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

for any $f(x,y)$ and any α and β.

METROPOLITAN STATE UNIVERSITY™ OF DENVER

Linear, Position-Invariant Degradations

In the presence of additive noise,

if H is a linear operator and position invariant,

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

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

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

Estimating the Degradation Function

- Three principal ways to estimate the degradation function

1. Observation

2. Experimentation

3. Mathematical Modeling

Mathematical Modeling (1)

- Environmental conditions cause degradation

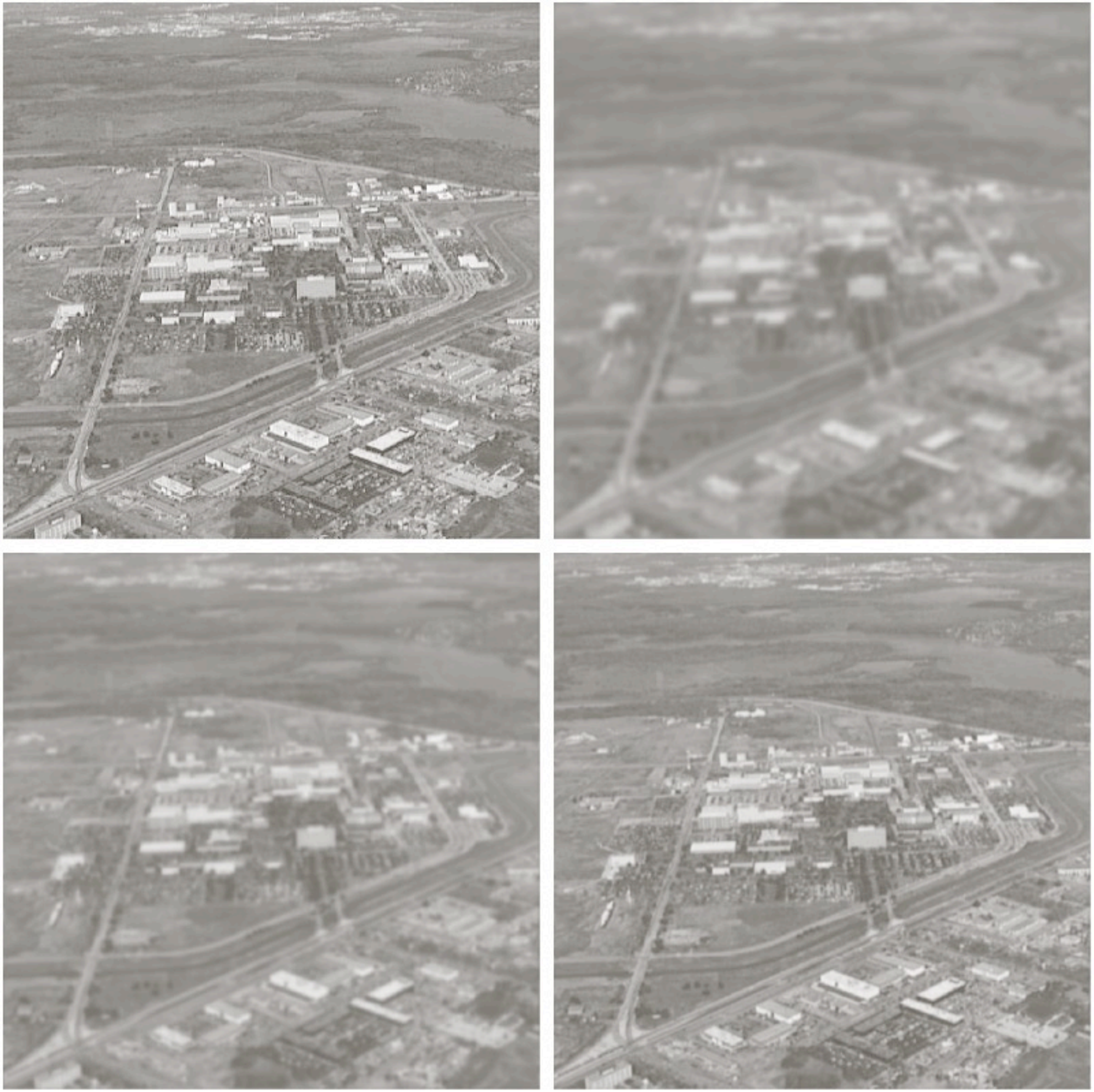
A model about atmospheric turbulence

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

k : a constant that depends on the nature of the turbulence

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.)



Mathematical Modeling (2)

- Derive a mathematical model from basic principles

E.g., An image blurred by uniform linear motion between the image and the sensor during image acquisition

Mathematical Modeling (3)

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

The optical imaging process is perfect. T is the duration of the exposure. The blurred image $g(x, y)$

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

Mathematical Modeling (4)

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

$$G(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy$$

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

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

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

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

Mathematical Modeling (4)

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

Suppose that the image undergoes uniform linear motion in the x-direction only, at a rate given by $x_0(t) = at/T$.

$$\begin{aligned} H(u,v) &= \int_0^T e^{-j2\pi ux_0(t)}dt \\ &= \int_0^T e^{-j2\pi uat/T}dt \\ &= \frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua} \end{aligned}$$

Mathematical Modeling (5)

Suppose that the image undergoes uniform linear motion in the x-direction and y-direction, at a rate given by

$x_0(t) = at / T$ and $y_0(t) = bt / T$

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

$$= \int_0^T e^{-j2\pi[ua+vb]t/T}dt$$

Inverse Filtering

An estimate of the transform of the original image

$$\hat{f}(x,y) = \frac{G(u,v)}{H(u,v)}$$

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

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

Inverse Filtering

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

1. We can't exactly recover the undegraded image because $N(u,v)$ is not known.

METROPOLITAN STATE UNIVERSITY OF DENVER

Minimum Mean Square Error (Wiener) Filtering

- **N. Wiener (1942)**

- **Objective**

  Find an estimate of the uncorrupted image such that the mean square error between them is minimized

$$e^2 = E\left\{(f - \hat{f})^2\right\}$$

METROPOLITAN STATE UNIVERSITY™ OF DENVER

Minimum Mean Square Error (Wiener) Filtering

The minimum of the error function is given in the frequency domain by the expression

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

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

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

Minimum Mean Square Error (Wiener) Filtering

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

$H(u,v)$: degradation function

$H^*(u,v)$: complex conjugate of $H(u,v)$

$|H(u,v)|^2 = H^*(u,v)H(u,v)$

$S_\eta(u,v) = |N(u,v)|^2 =$ power spectrum of the noise

$S_f(u,v) = |F(u,v)|^2 =$ power spectrum of the undegraded image

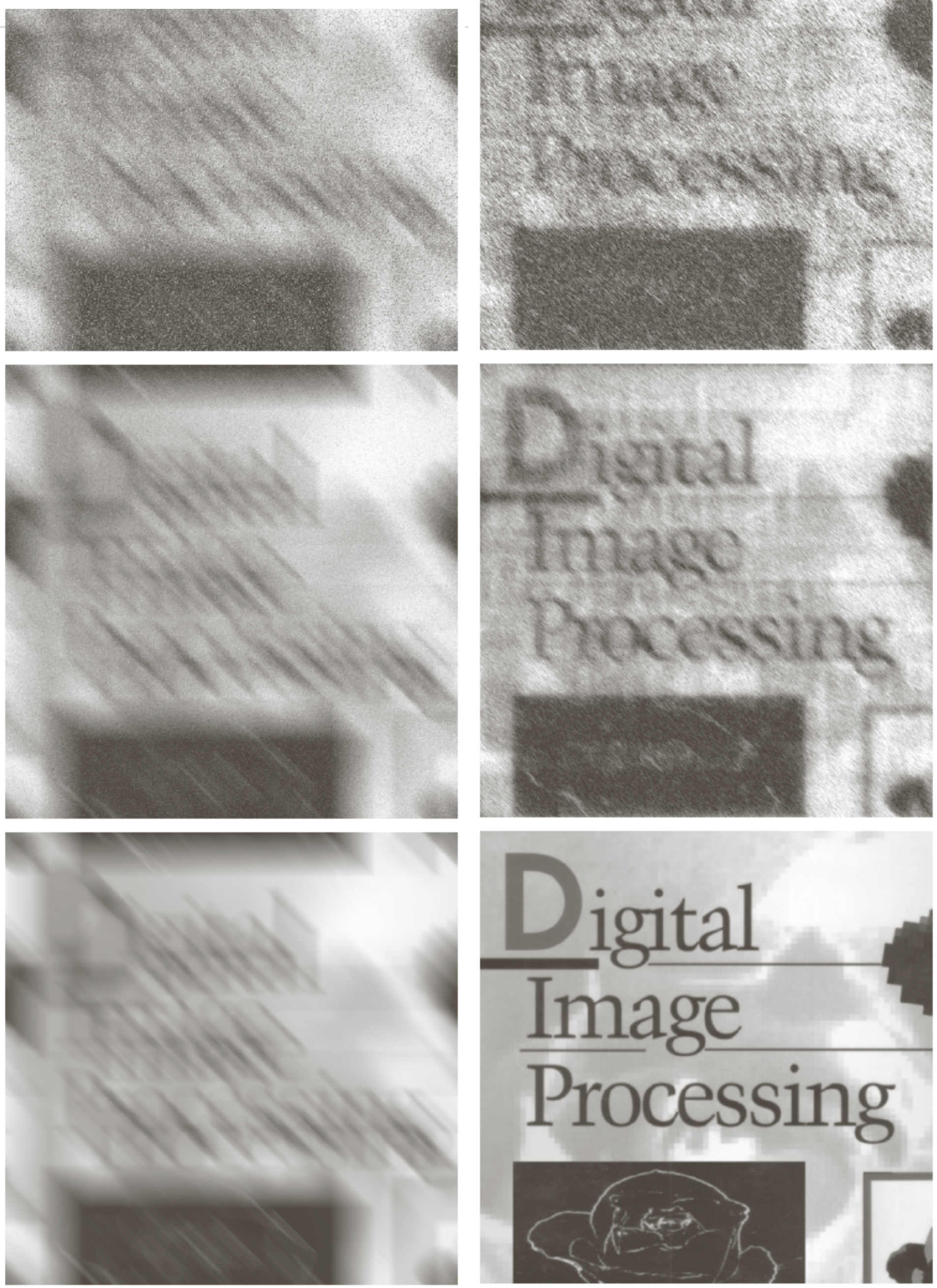
Minimum Mean Square Error (Wiener) Filtering

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

K is a specified constant. Generally, the value of K is chosen interactively to yield the best visual results.

Left: degraded image

Right: Wiener filtering



Geometric Mean Filter

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

$\alpha = 1$: inverse filter

$\alpha = 0$: parametric Wiener filter

$\alpha = 1/2$: geometric mean filter

Exercise

- Gaussian noise
- Uniform noise
- Salt & Pepper

- Averaging filter, LPF
- Averaging filter, LPF
- Median filter

Exercise

- Gaussian noise
- Uniform noise
- Rayleigh noise
- Exponential noise
- Salt & Pepper

- Randn()
- Rand()
- a + sqrt(-b*ln(1-U(0,1)))
- -(1/a)*ln(1-U(0,1))
- ?

- "Log" inn matlab is Natural logarithm

Syntax

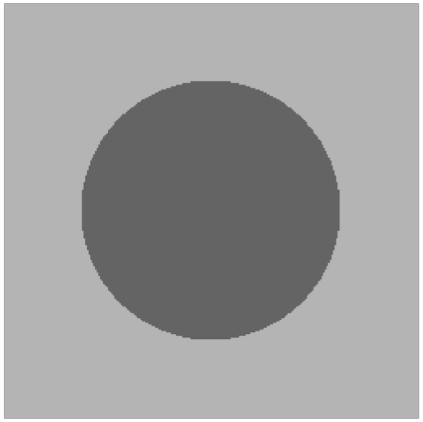
```
J = imnoise(I,type)
J = imnoise(I,type,parameters)
J = imnoise(I,'gaussian',M,V)
J = imnoise(I,'localvar',V)
J = imnoise(I,'localvar',image_intensity,var)
J = imnoise(I,'poisson')
J = imnoise(I,'salt & pepper',d)
J = imnoise(I,'speckle',v)
gpuarrayJ = imnoise(gpuarrayI, ___)
```

Description

`J = imnoise(I,type)` adds noise of a given type to the intensity image `I`. *type* is a string that specifies any of the following types of noise. Note that certain types of noise support additional parameters. See the related syntax.

Value	Description
'gaussian'	Gaussian white noise with constant mean and variance
'localvar'	Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson'	Poisson noise
'salt & pepper'	On and off pixels
'speckle'	Multiplicative noise

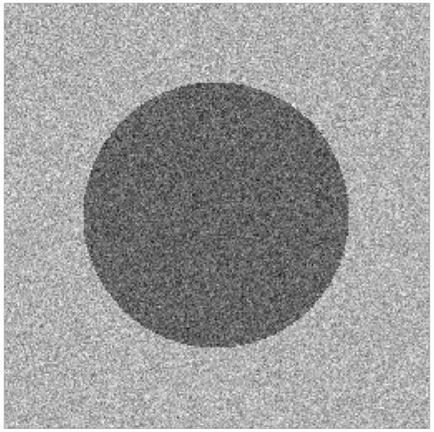
Exercise

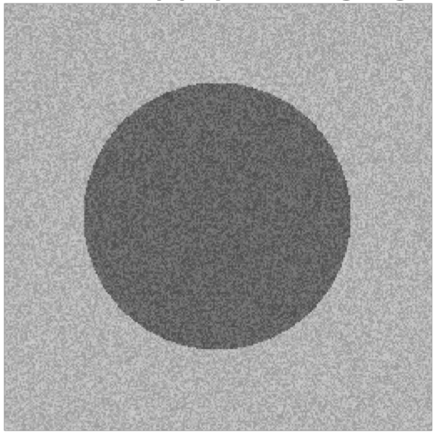


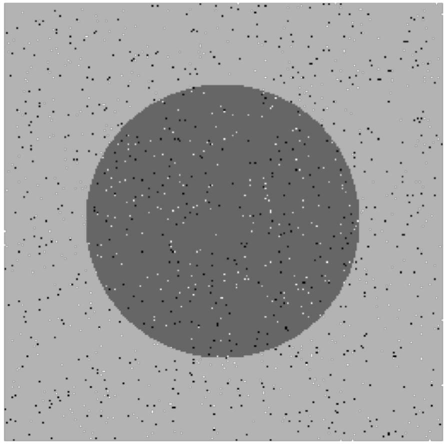
- 256 by 256 double type [0,1]
- sqrt((i-128)^2+(j-128)^2)<80 Center part set 0.4
- Other part set 0.7

- 1, Gaussian noise (0, 0.01)
- 2, uniform noise [-0.05   0.05]
- 3, Salt and Pepper Ps=Pp=0.02

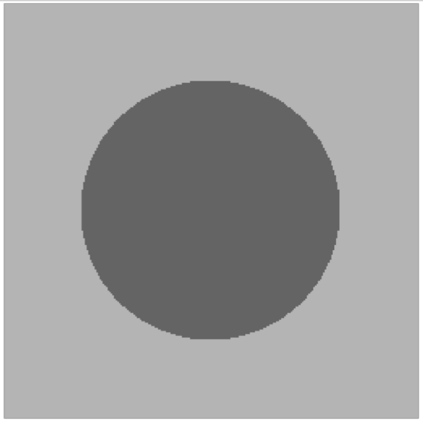
- Plot histograms
- Apply averaging and median filter

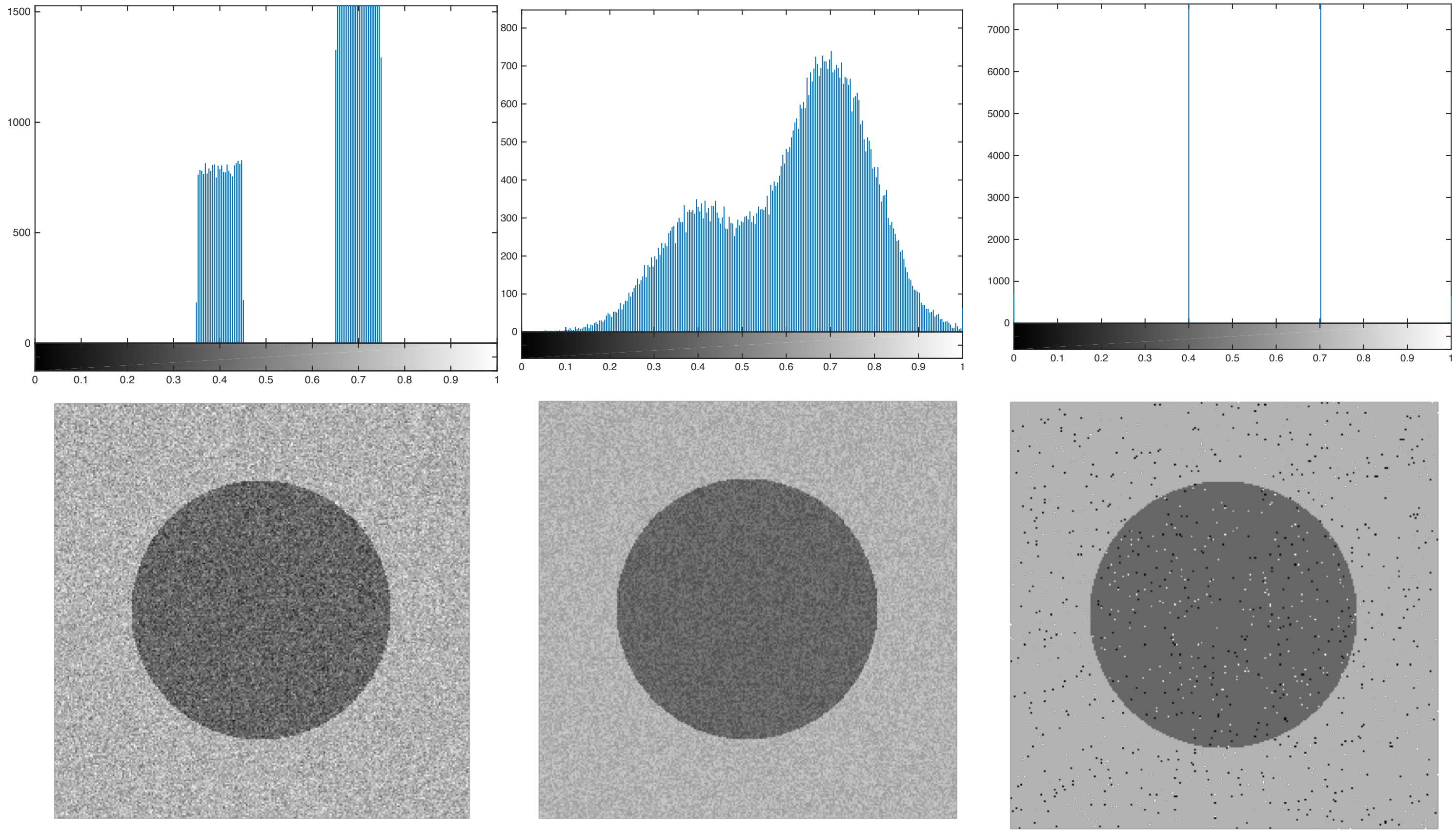






Exercise





1500
1000
500
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
800
700
600
500
400
300
200
100
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
7000
6000
5000
4000
3000
2000
1000
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1