Image Restoration

Dr. Badri Narayan Subudhi

Department of EE

IIT Jammu

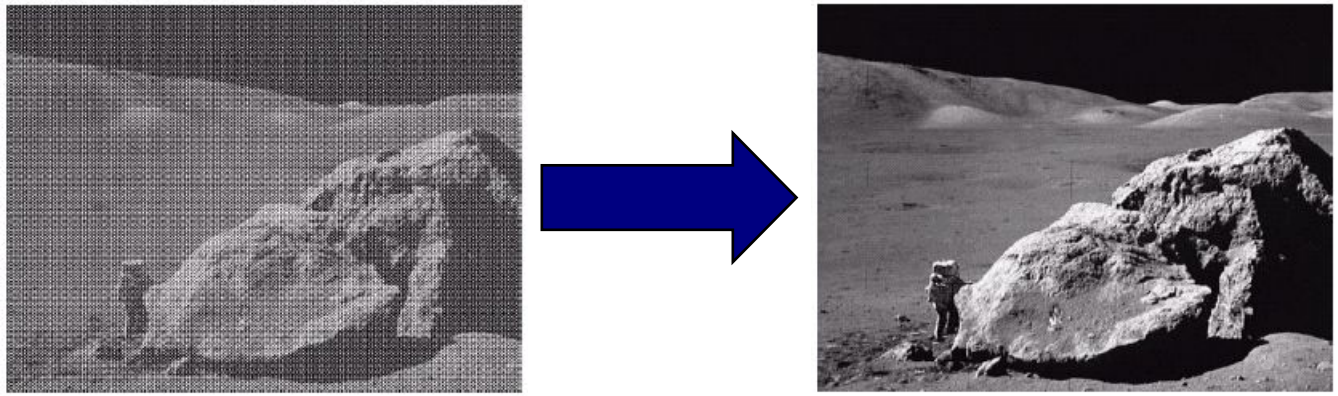
Image Restoration

- Image restoration vs. image enhancement
  - Enhancement:
    - largely a subjective process
    - Priori knowledge about the degradation is not a must (sometimes no degradation is involved)
    - Procedures are heuristic and take advantage of the psychophysical aspects of human visual system
  - Restoration:
    - more an objective process
    - Images are degraded
    - Tries to recover the images by using the knowledge about the degradation

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission

Noise Model

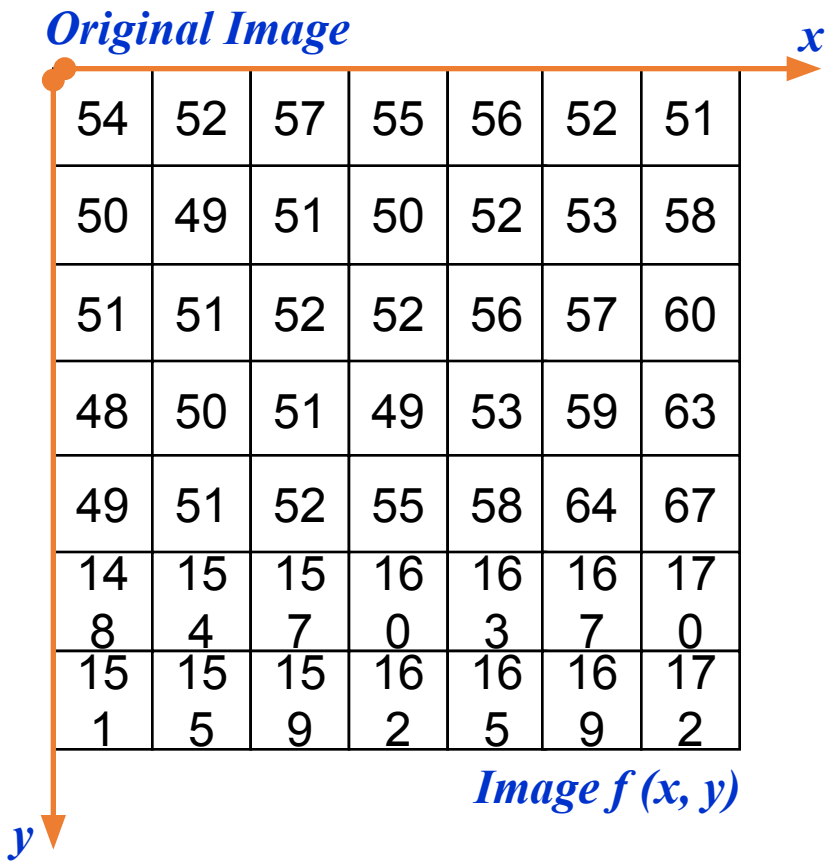
We can consider a noisy image to be modelled as follows:

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

where $f(x, y)$ is the original image pixel, $\eta(x, y)$ is the noise term and $g(x, y)$ is the resulting noisy pixel

If we can estimate the model the noise in an image is based on this will help us to figure out how to restore the image

Noise Corruption Example



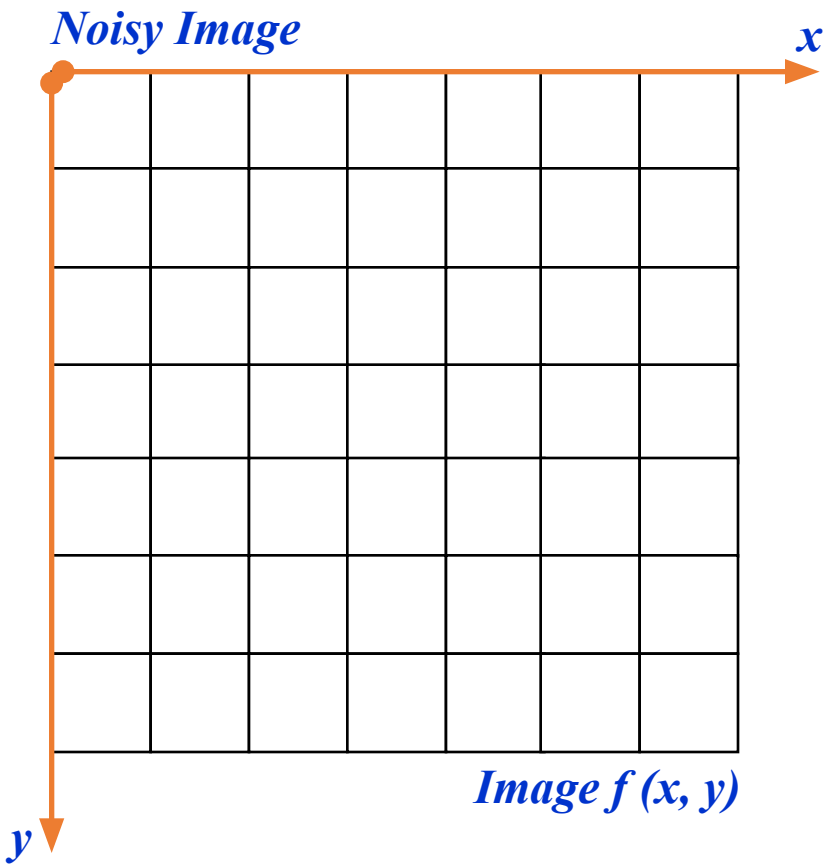


Image Restoration - Applications

Microscopy

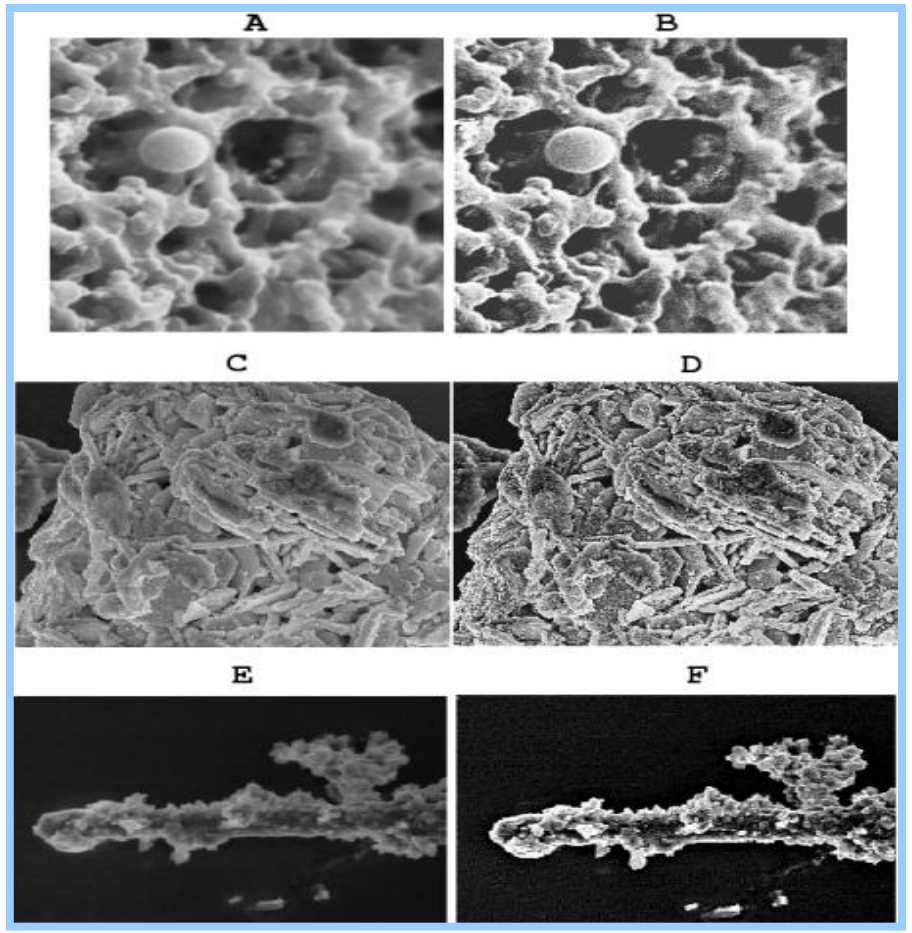
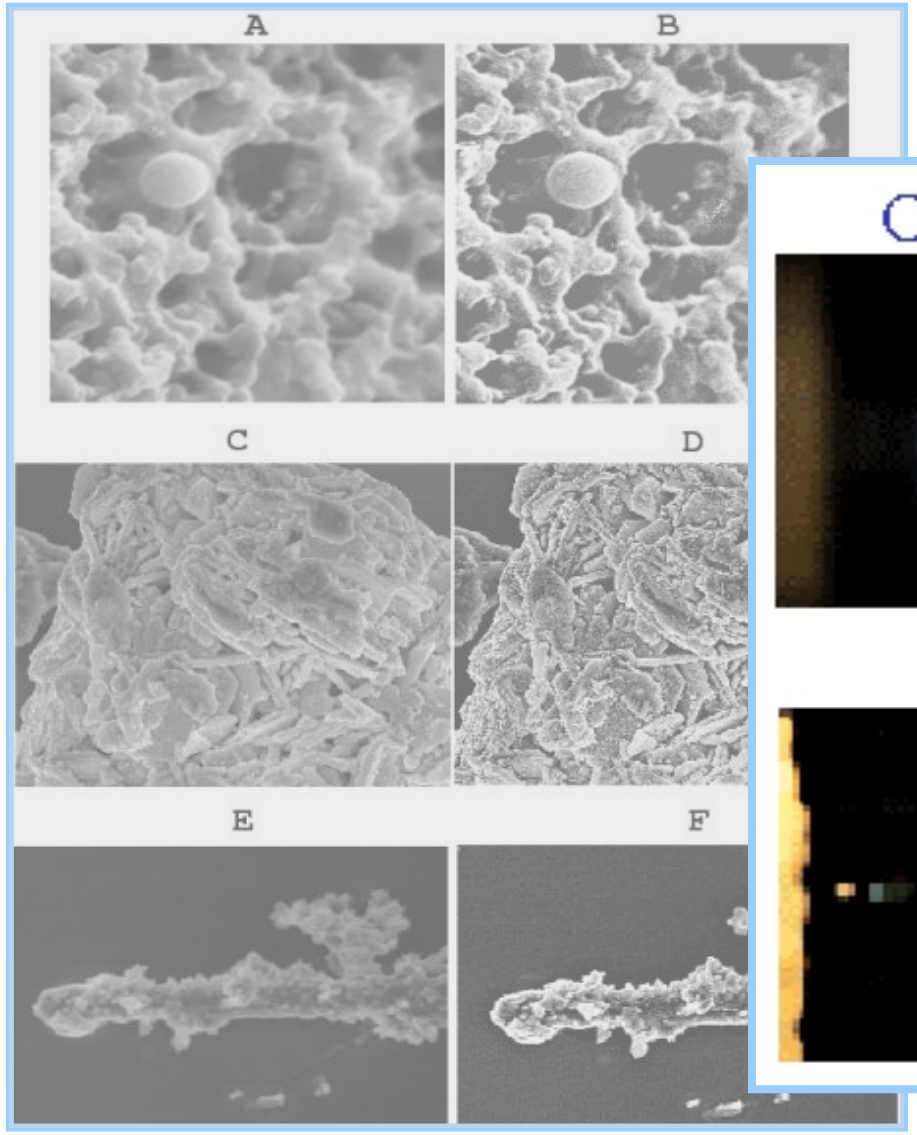


Image Restoration - Applications



Astronomy

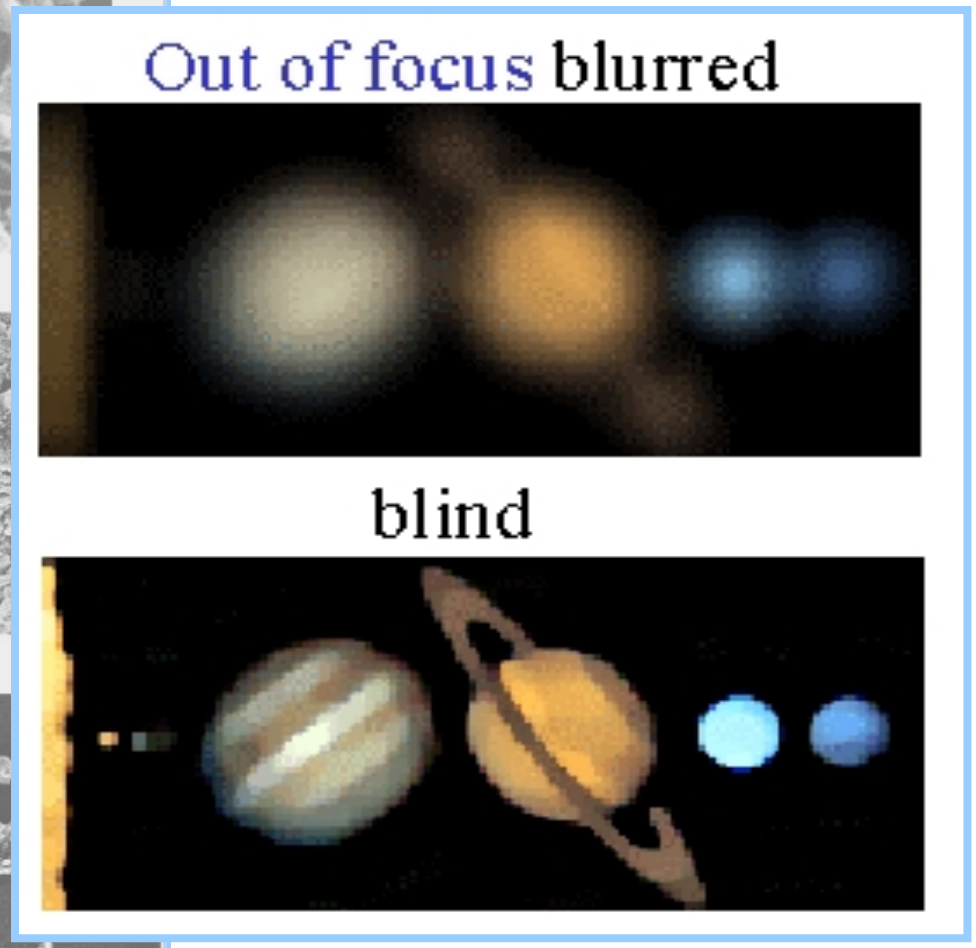
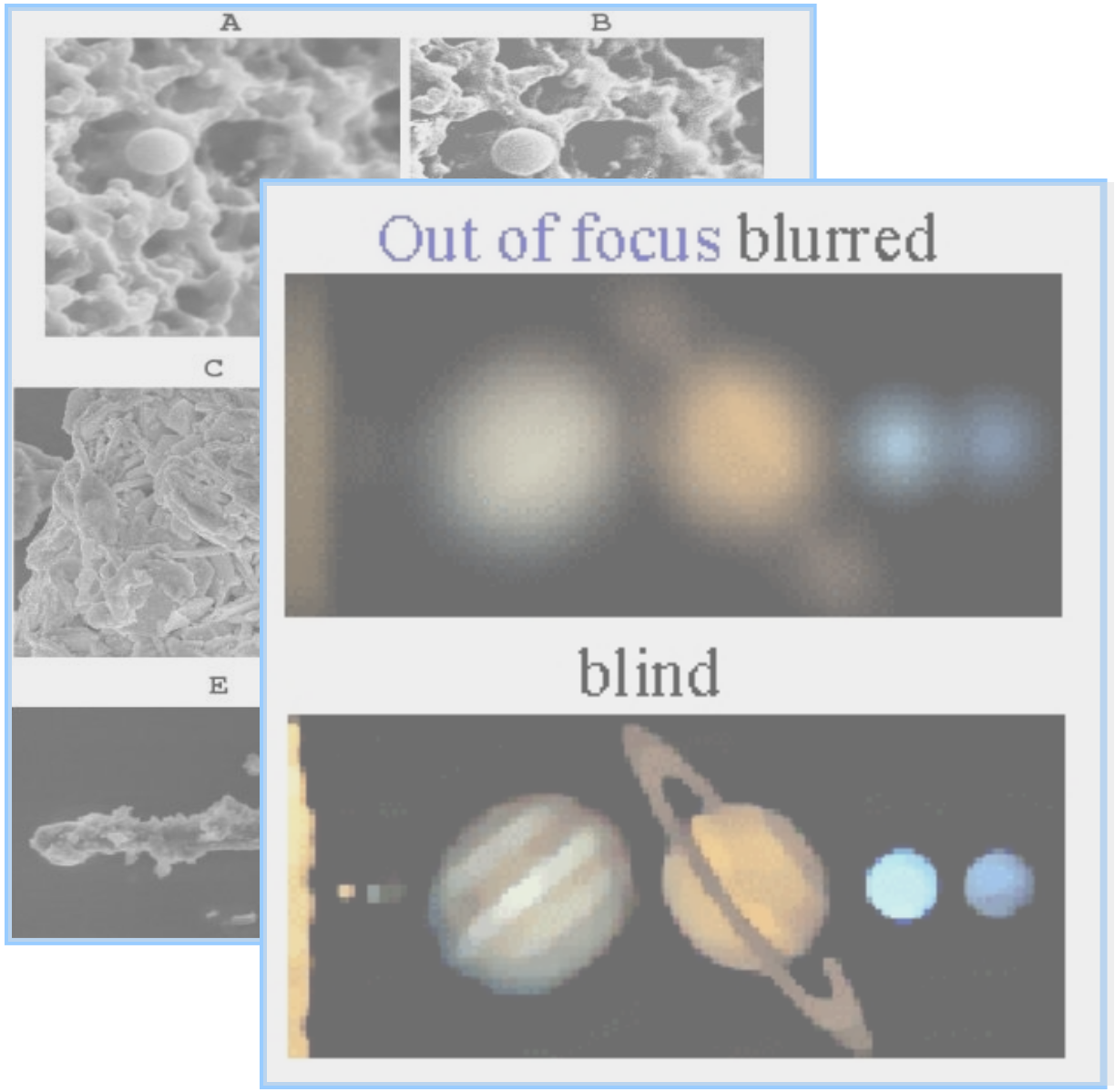
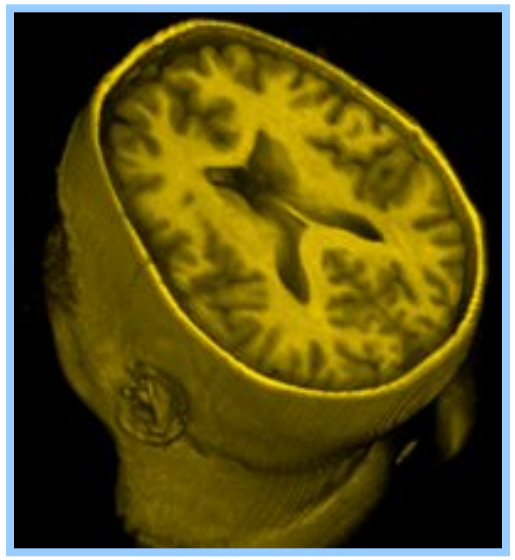


Image Restoration - Applications



Medical Imaging



Goal of Restoration : Improve an image in some predefined sense. i.e.

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

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

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

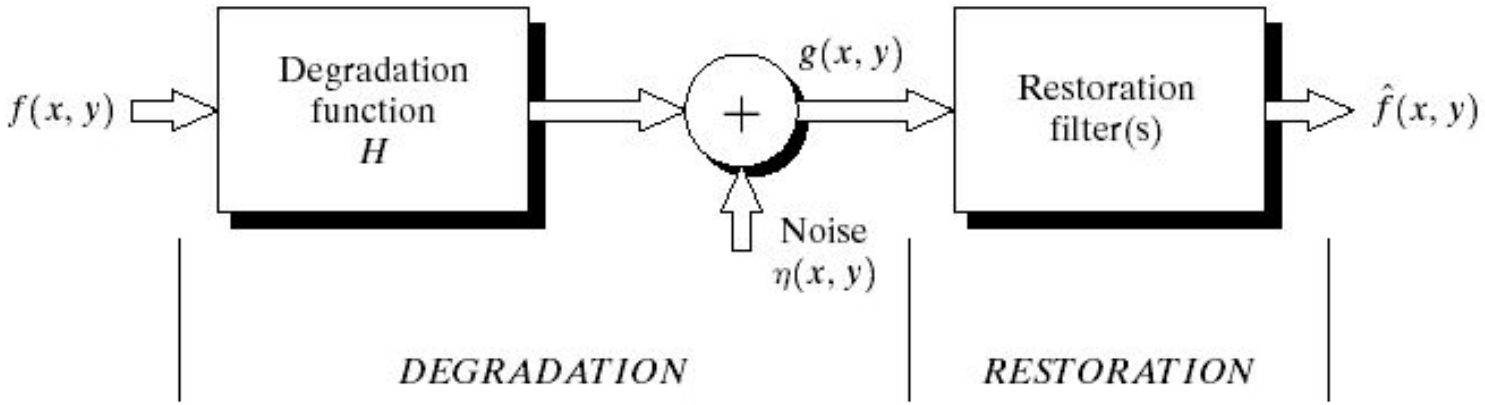
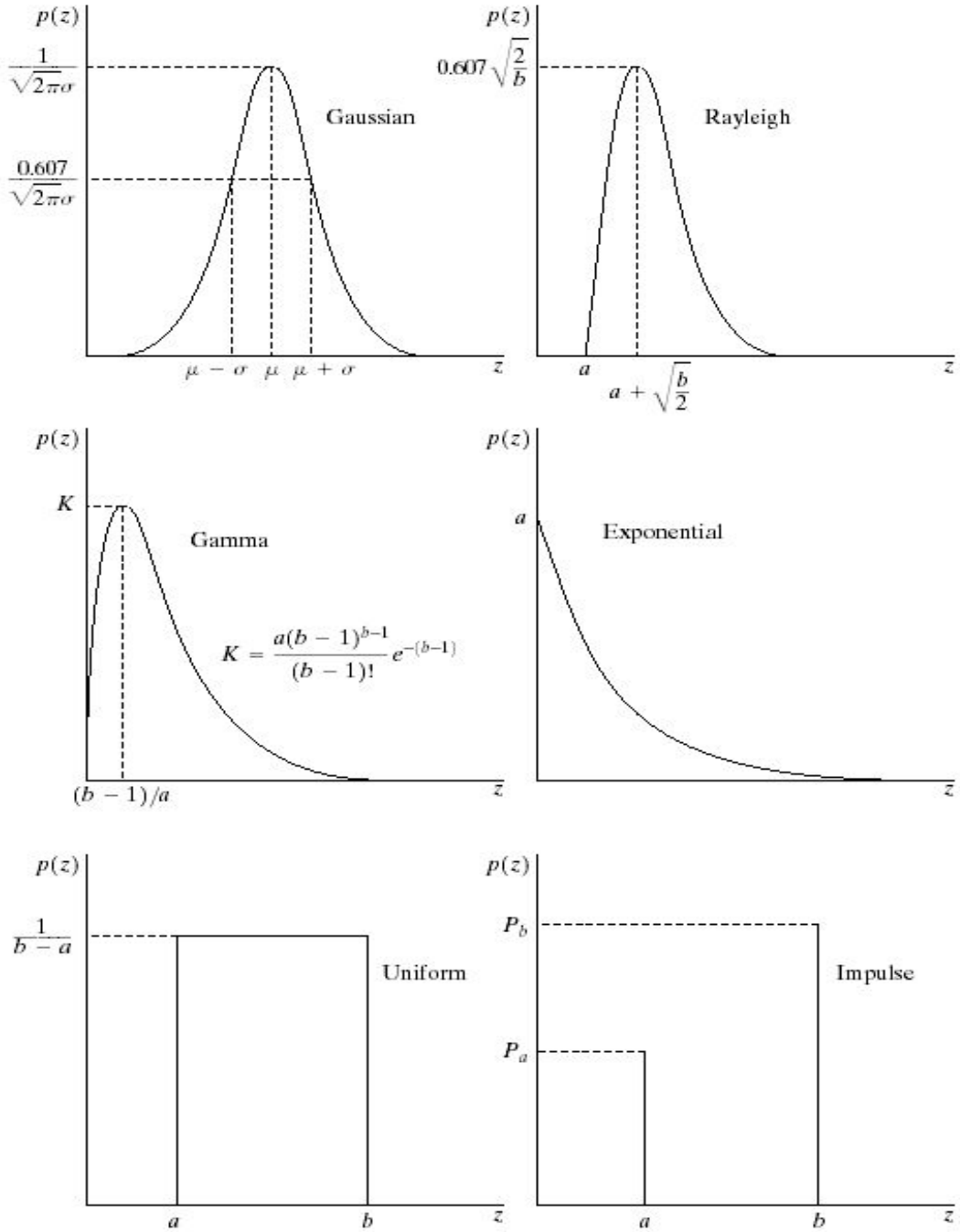


FIGURE 5.1 A model of the image degradation/ restoration process.



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

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

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

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

e) $p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$ Uniform Noise

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

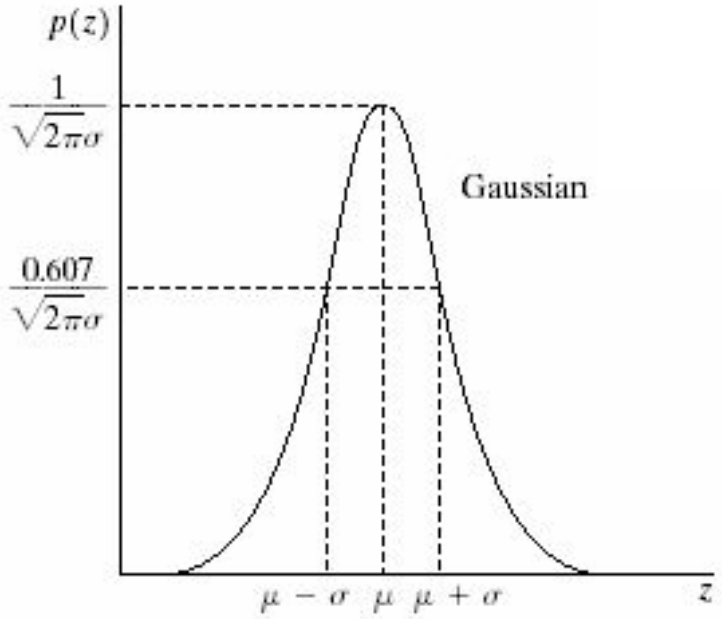
a b
c d
e f

FIGURE 5.2 Some important probability density functions.

Gaussian Noise

- Noise (image) can be classified according the distribution of the values of pixels (of the noise image) or its (normalized) histogram
- Gaussian noise is characterized by two parameters, μ (mean) and σ^2 (variance), by

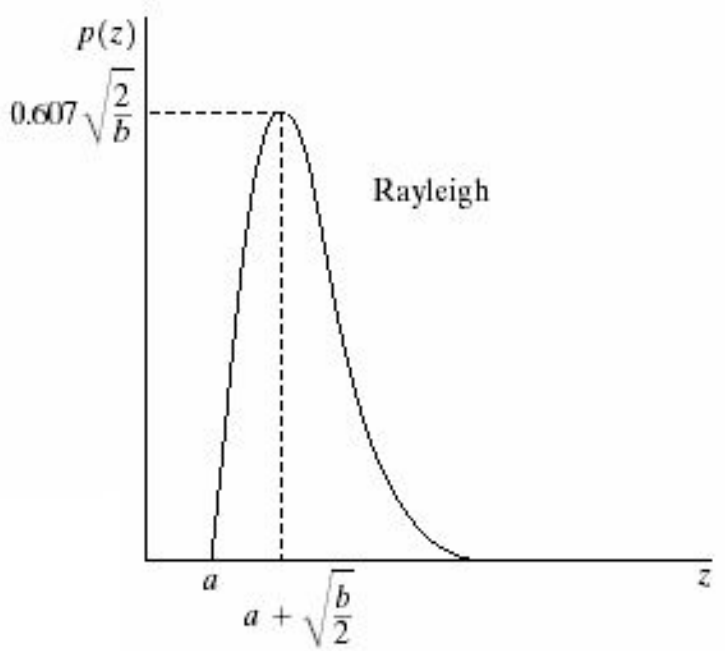
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$



- 70% values of z fall in the range [(μ-σ),(μ+σ)]
- 95% values of z fall in the range [(μ-2σ),(μ+2σ)]

- **Rayleigh** noise

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



- The mean and variance of this density are given by

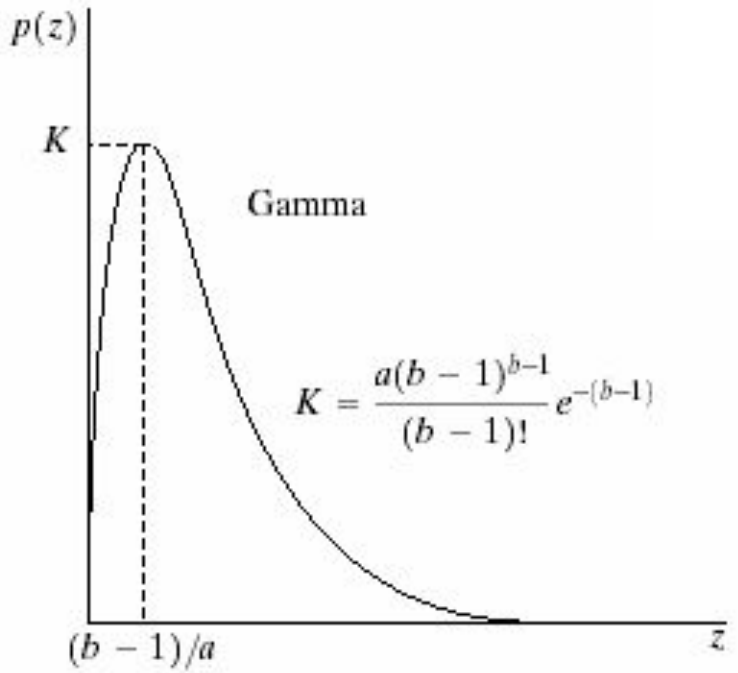
$$\mu = a + \sqrt{\pi b/4} \quad \text{and} \quad \sigma^2 = \frac{b(4-\pi)}{4}$$

- **Erlang (Gamma)** noise

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

  - The mean and variance of this density are given by

$$\mu = b/a \text{ and } \sigma^2 = \frac{b}{a^2}$$

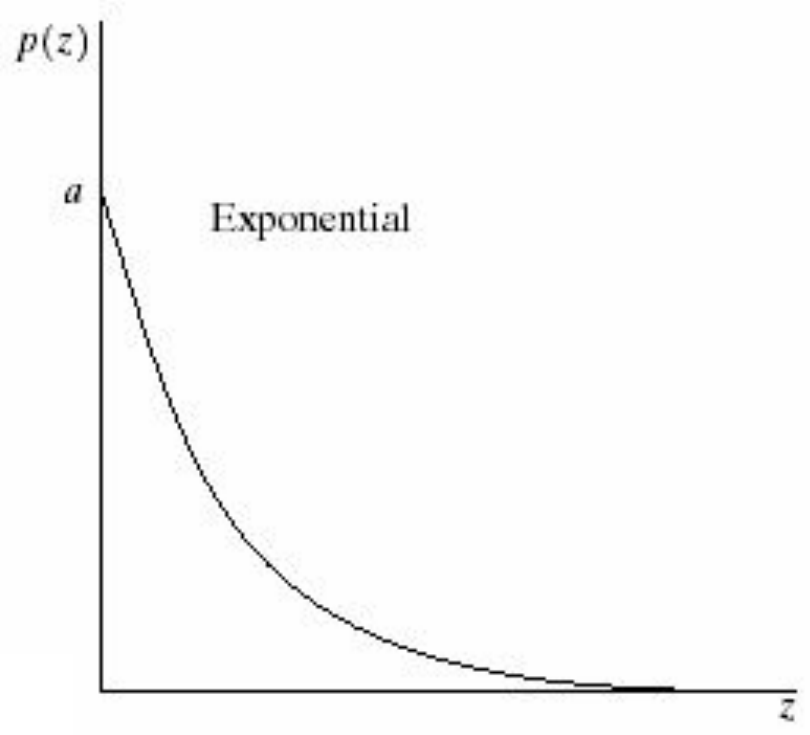


- **Exponential** noise

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

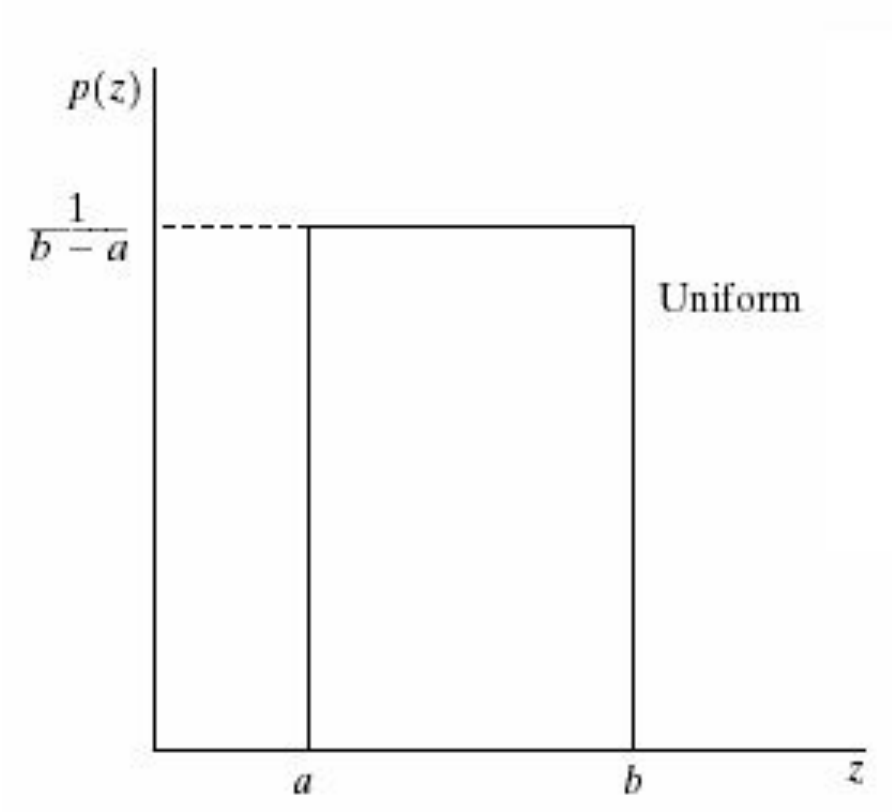
  - The mean and variance of this density are given by

$$\mu = 1/a \text{ and } \sigma^2 = \frac{1}{a^2}$$



- **Uniform** noise

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

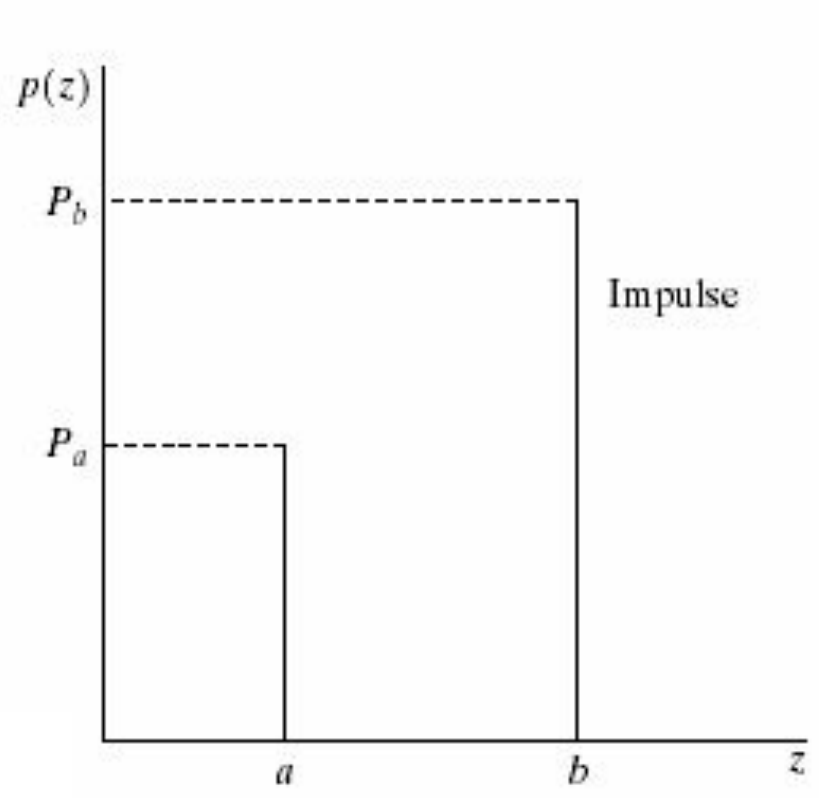


- The mean and variance of this density are given by

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

- **Impulse** (salt-and-pepper) noise

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



- If either Pa or Pb is zero, the impulse noise is called unipolar
- a and b usually are extreme values because impulse corruption is usually large compared with the strength of the image signal
- It is the only type of noise that can be distinguished from others visually

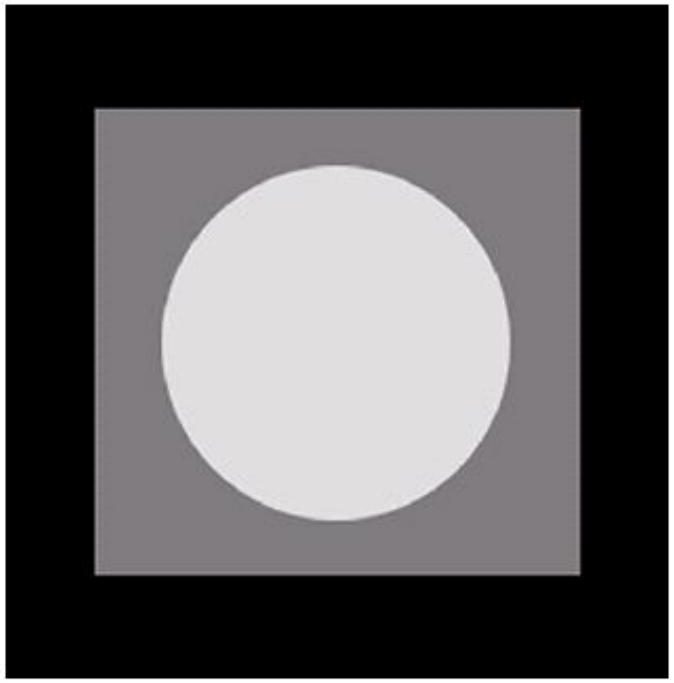
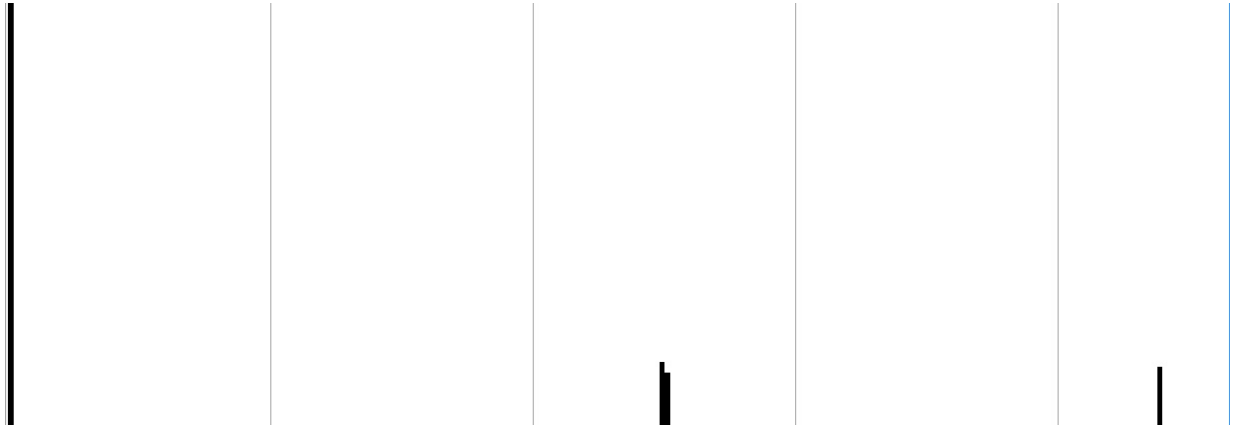
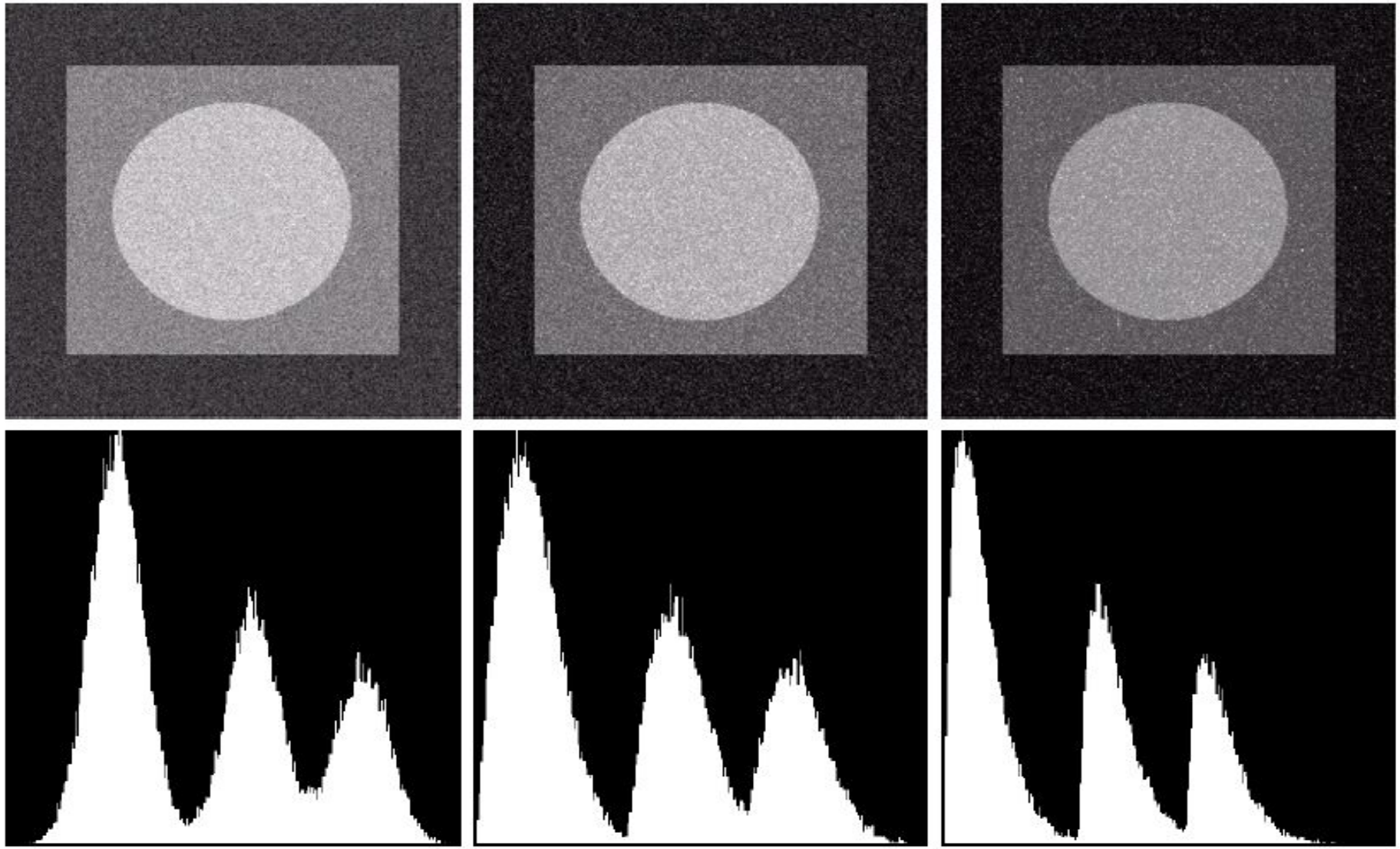


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

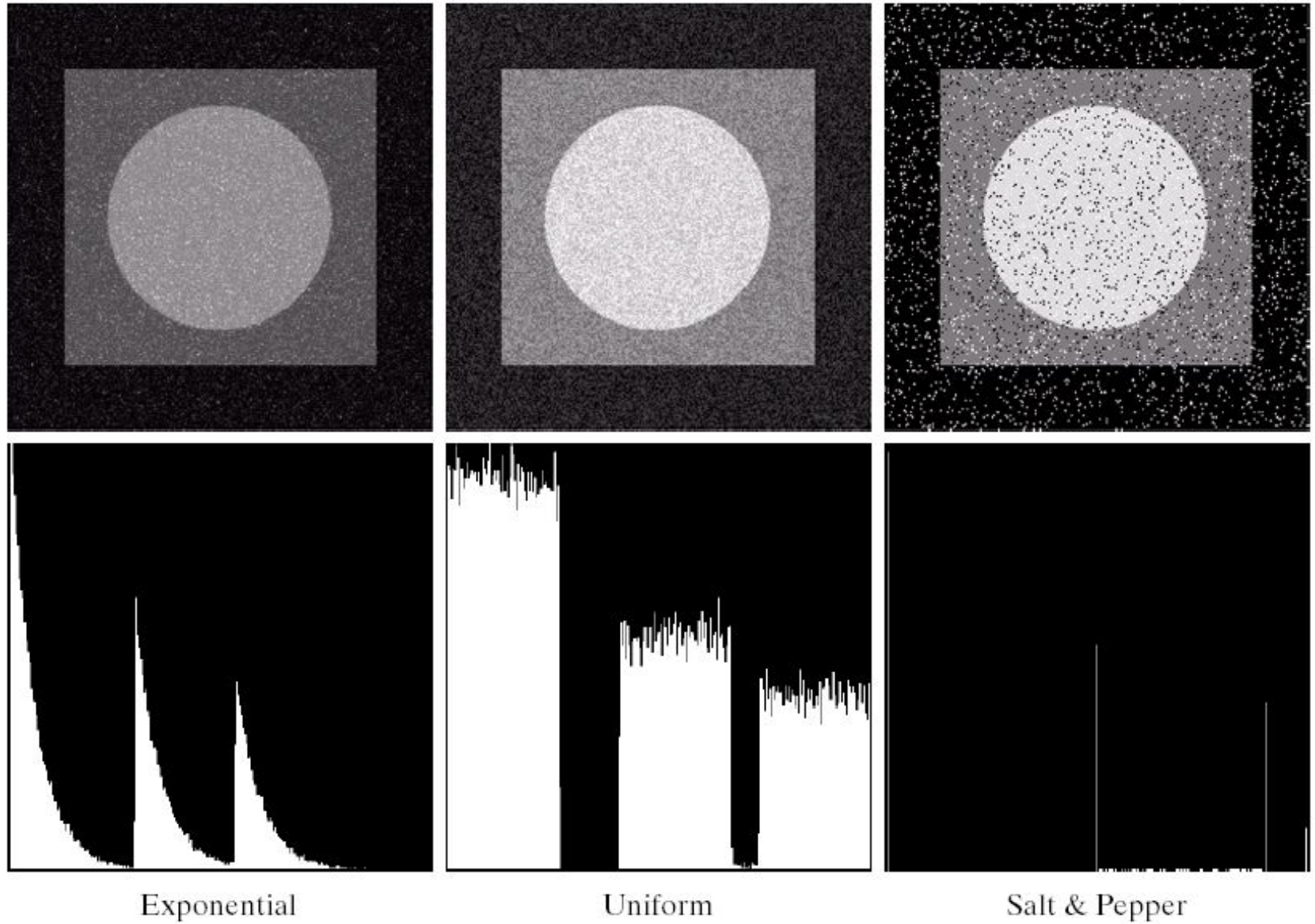


This test pattern is well-suited for illustrating the noise models, because it is composed of simple, constant areas that span the grey scale from black to white in only three increments. This facilitates visual analysis of the characteristics of the various noise components added to the image.



a b c
d e f

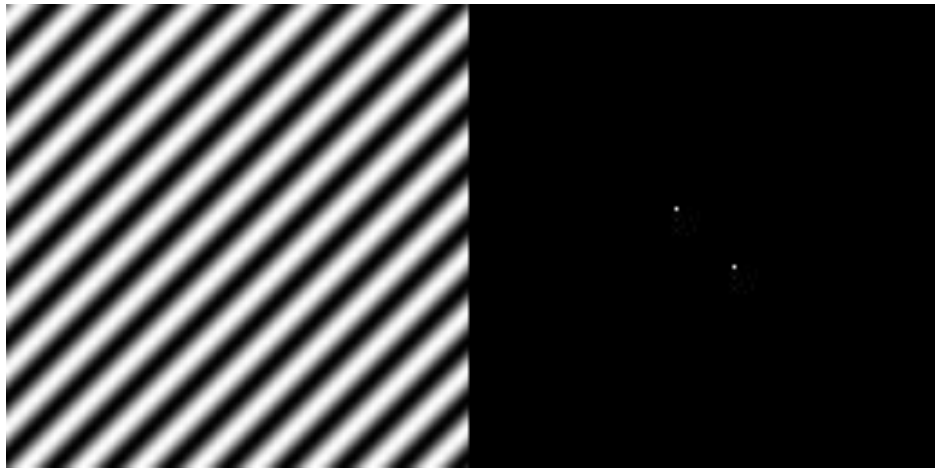
FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

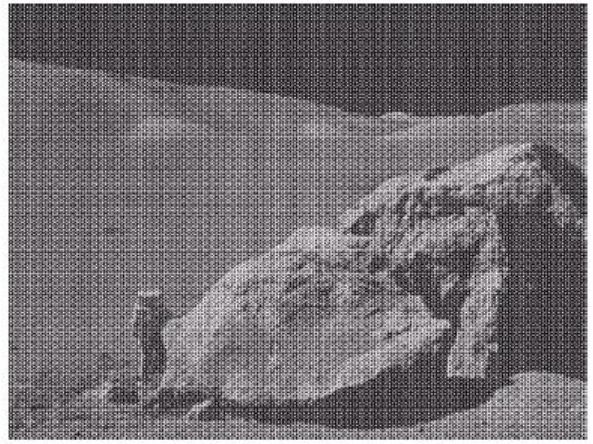


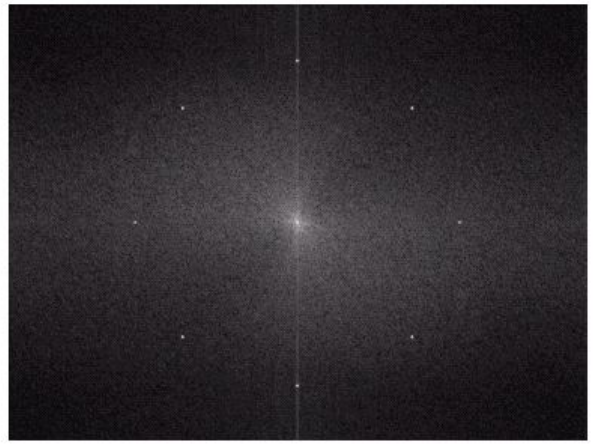
g h i
j k l

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

Periodic Noise



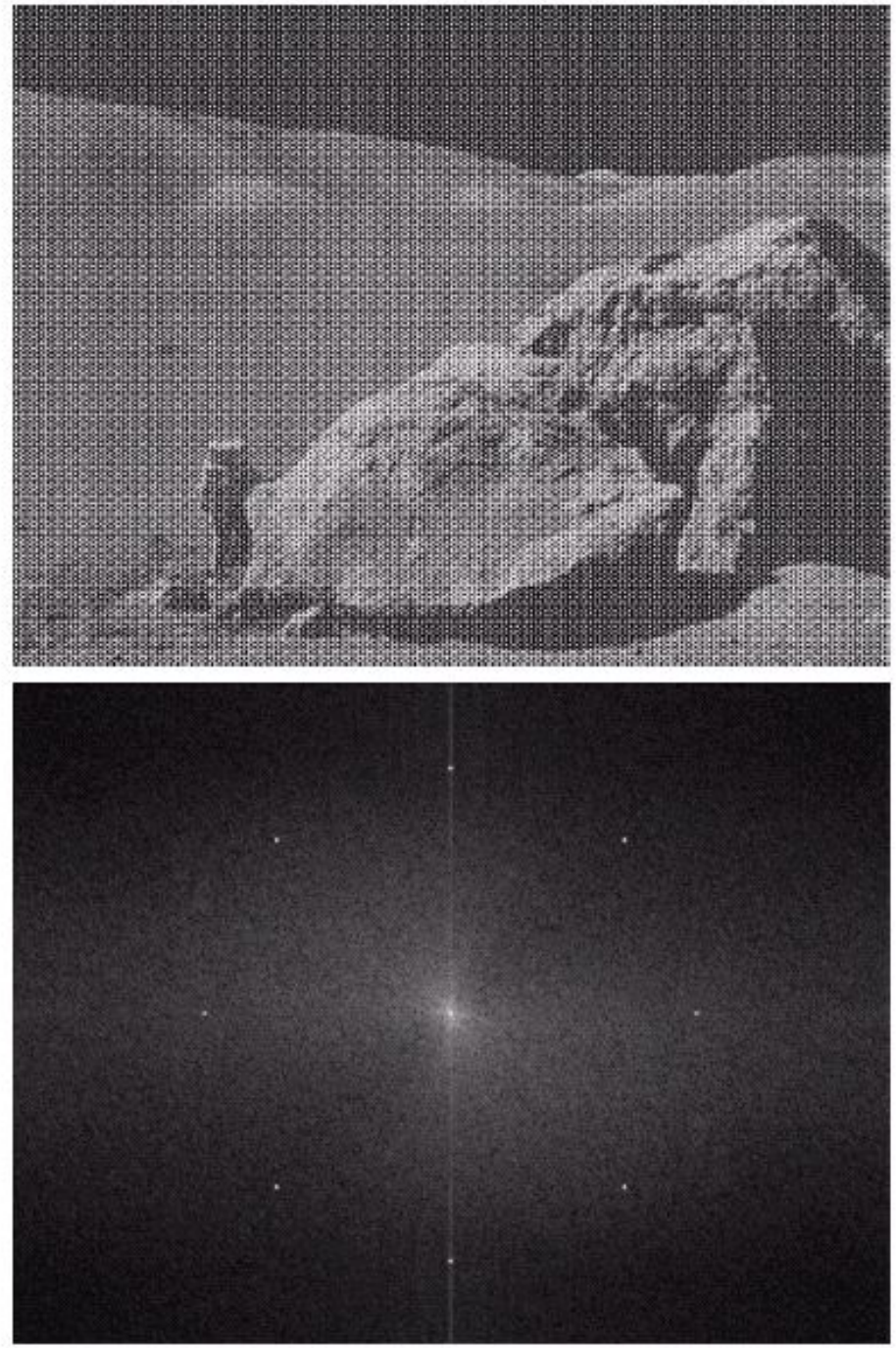


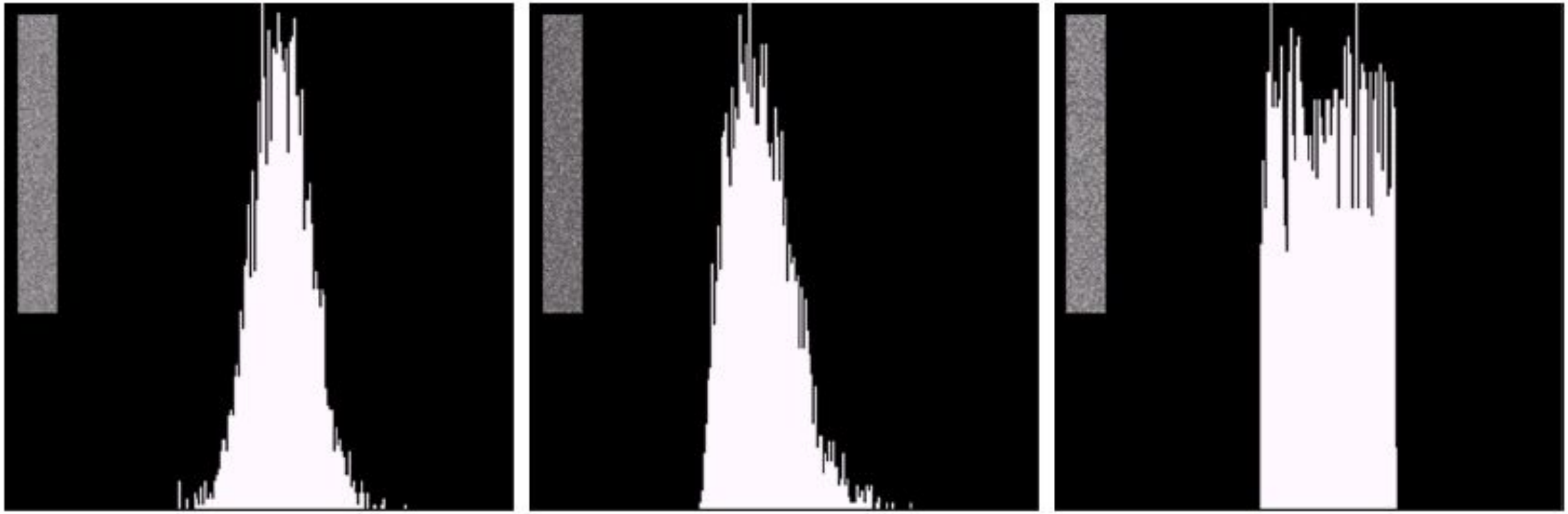


- Typically arises due to electrical or electromagnetic interference
- Gives rise to regular noise patterns in an image
- Frequency domain techniques in the Fourier domain are most effective at removing periodic noise

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.)





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.

Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise

The *arithmetic mean* filter is a very simple one and is calculated as follows:

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

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

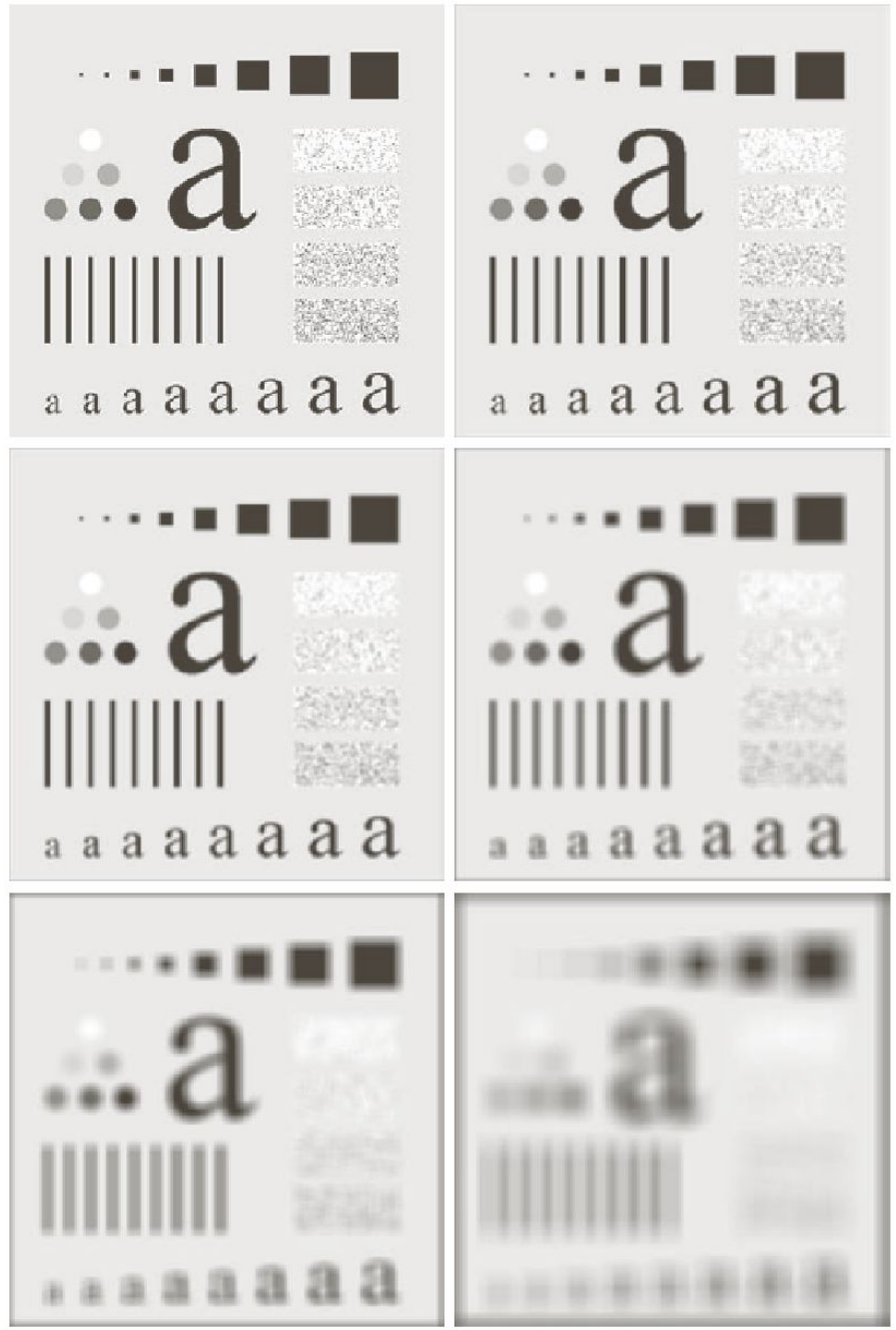
This is implemented as the simple smoothing filter

Blurs the image to remove noise

0	0	0	0	0	0	0	0	0
0	25	51	72	90	85	35	61	0
0	21	21	23	35	34	36	37	0
0	38	41	44	45	47	48	31	0
0	21	79	90	91	81	85	90	0
0	88	21	34	56	76	88	90	0
0	51	51	56	57	67	88	90	0
0	91	95	93	92	66	67	65	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	13	24	32	38	35	32	19	0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Other Means

There are different kinds of mean filters all of which exhibit slightly different behaviour:

- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

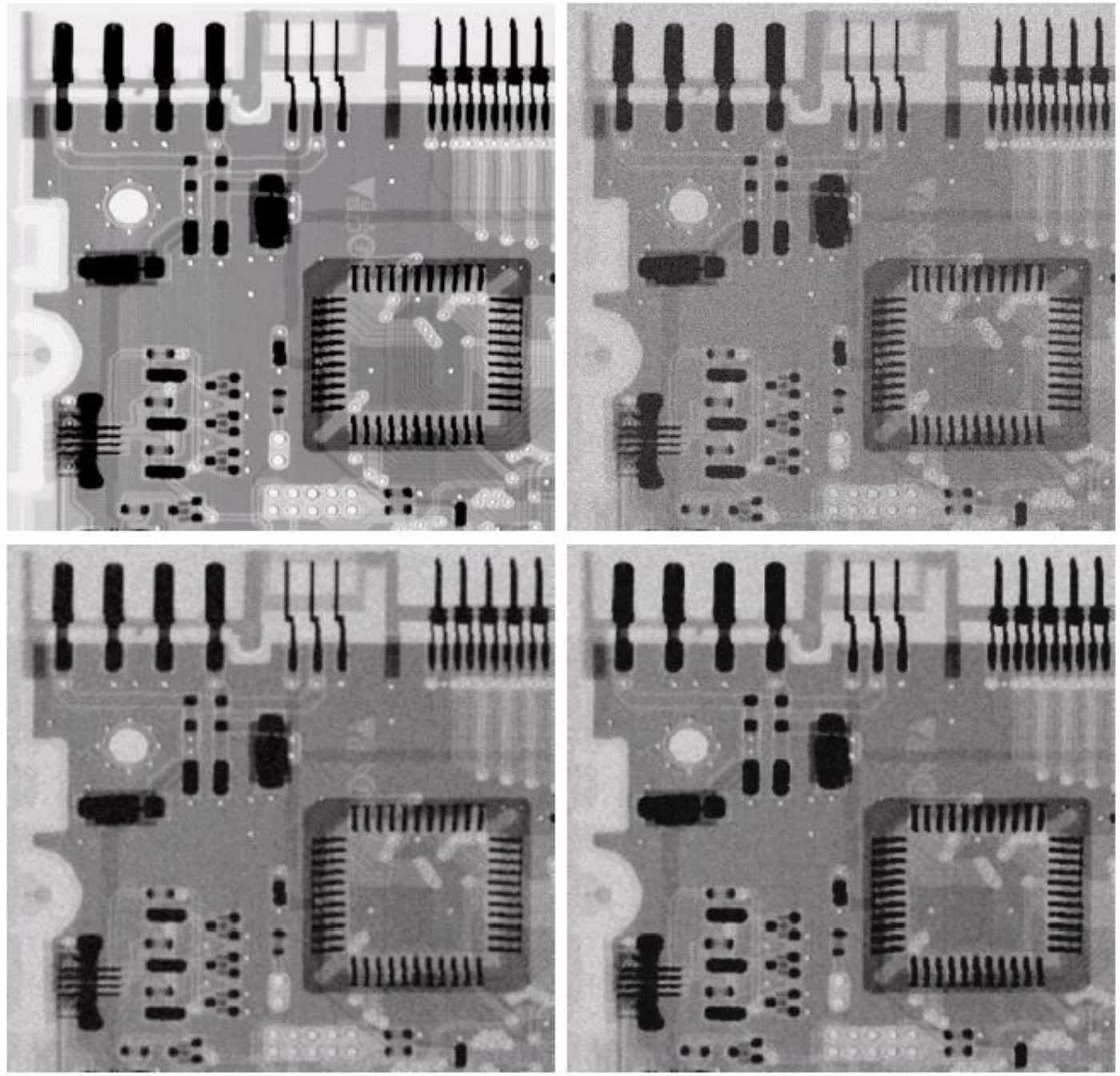
Other Means (cont…)

There are other variants on the mean which can give different performance

Geometric Mean:

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

Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail



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.)

Other Means (cont…)

Harmonic Mean:

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

Works well for salt noise, but fails for pepper noise

Also does well for other kinds of noise such as Gaussian noise

Other Means (cont…)

Contraharmonic Mean:

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

Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

a b
c d

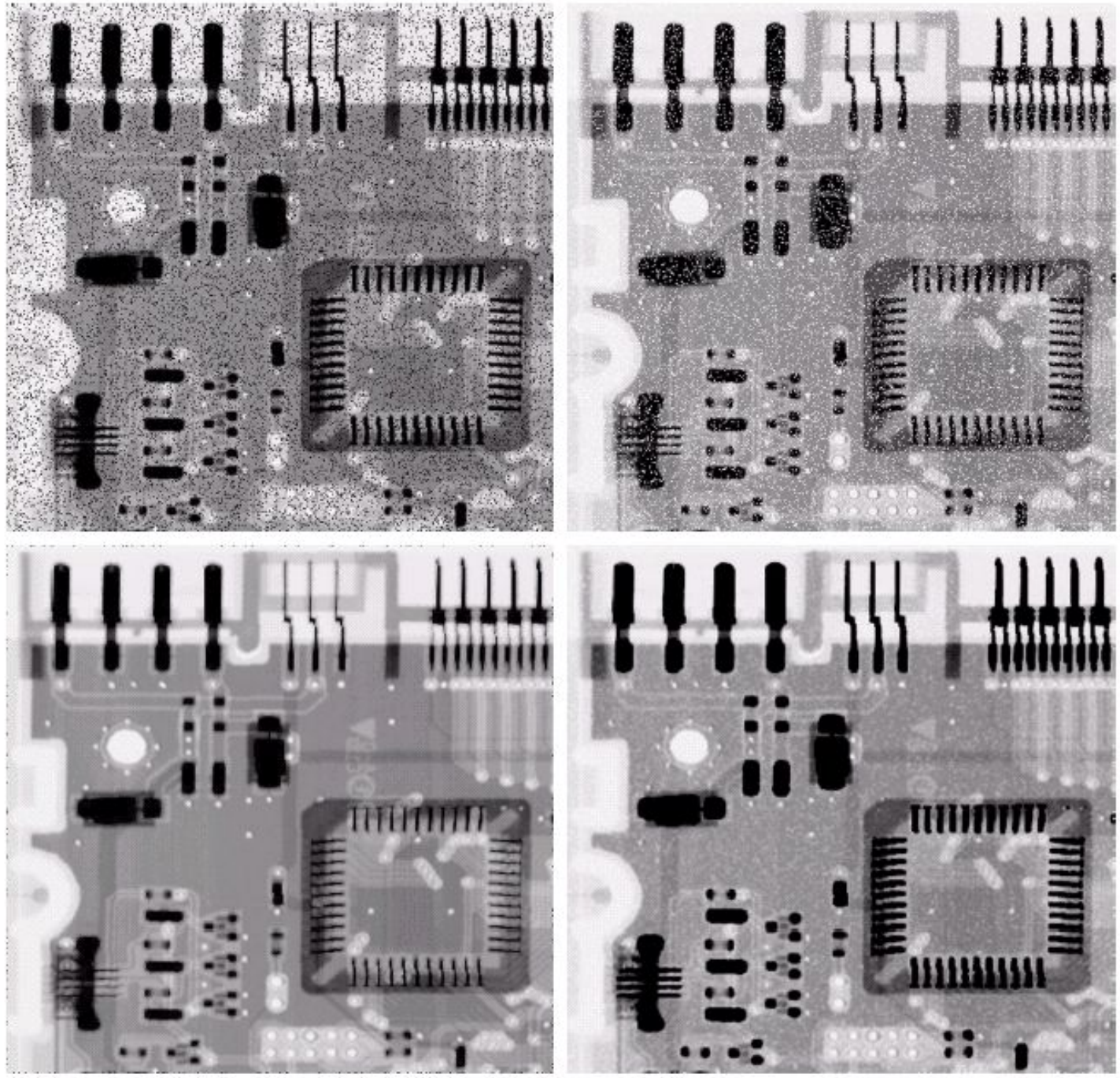
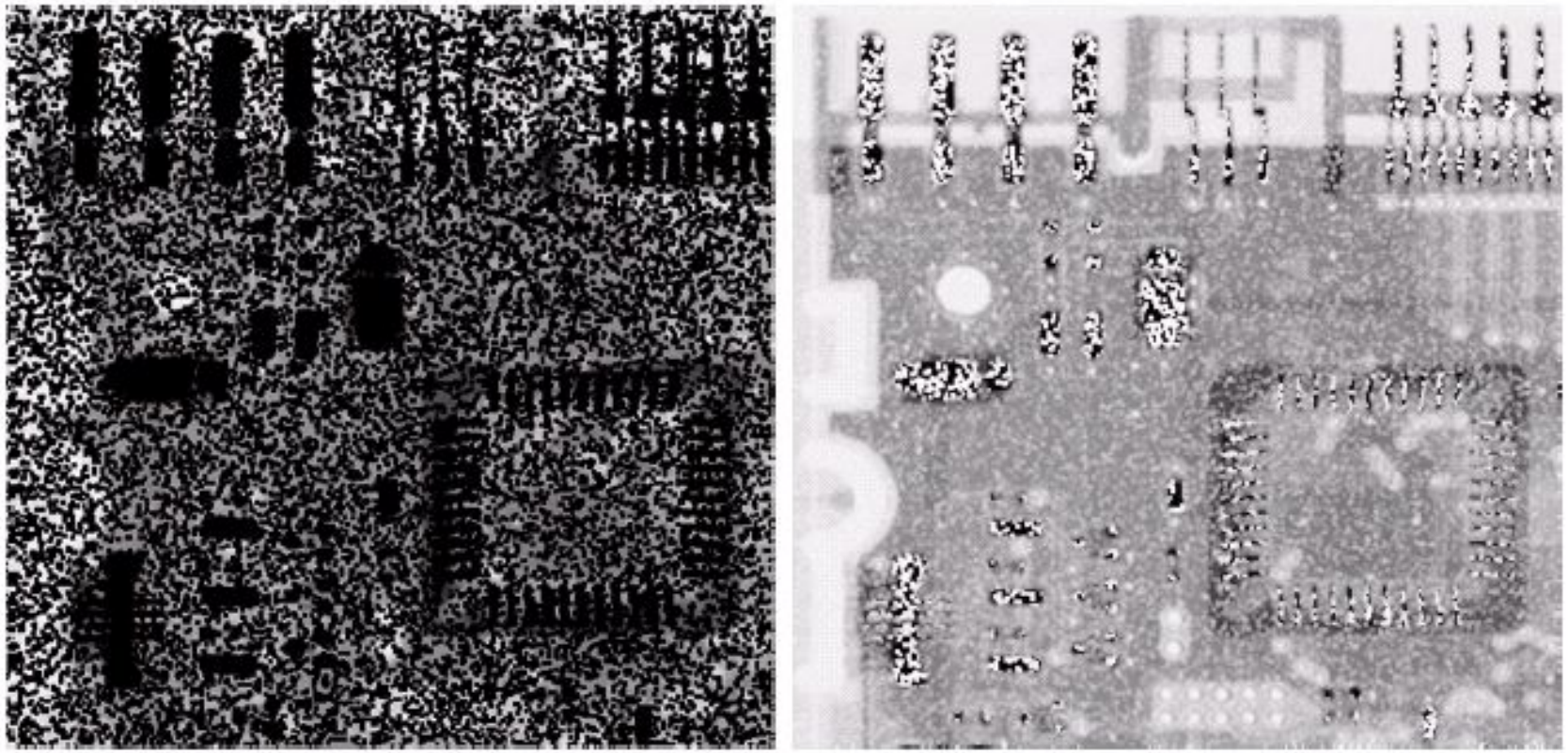


FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



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 $Q = 1.5$.

Order Statistics Filters

Spatial filters that are based on ordering the pixel values that make up the neighbourhood operated on by the filter

Useful spatial filters include

- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Median Filter

Median Filter:

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

Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters

Particularly good when salt and pepper noise is present

a b
c d

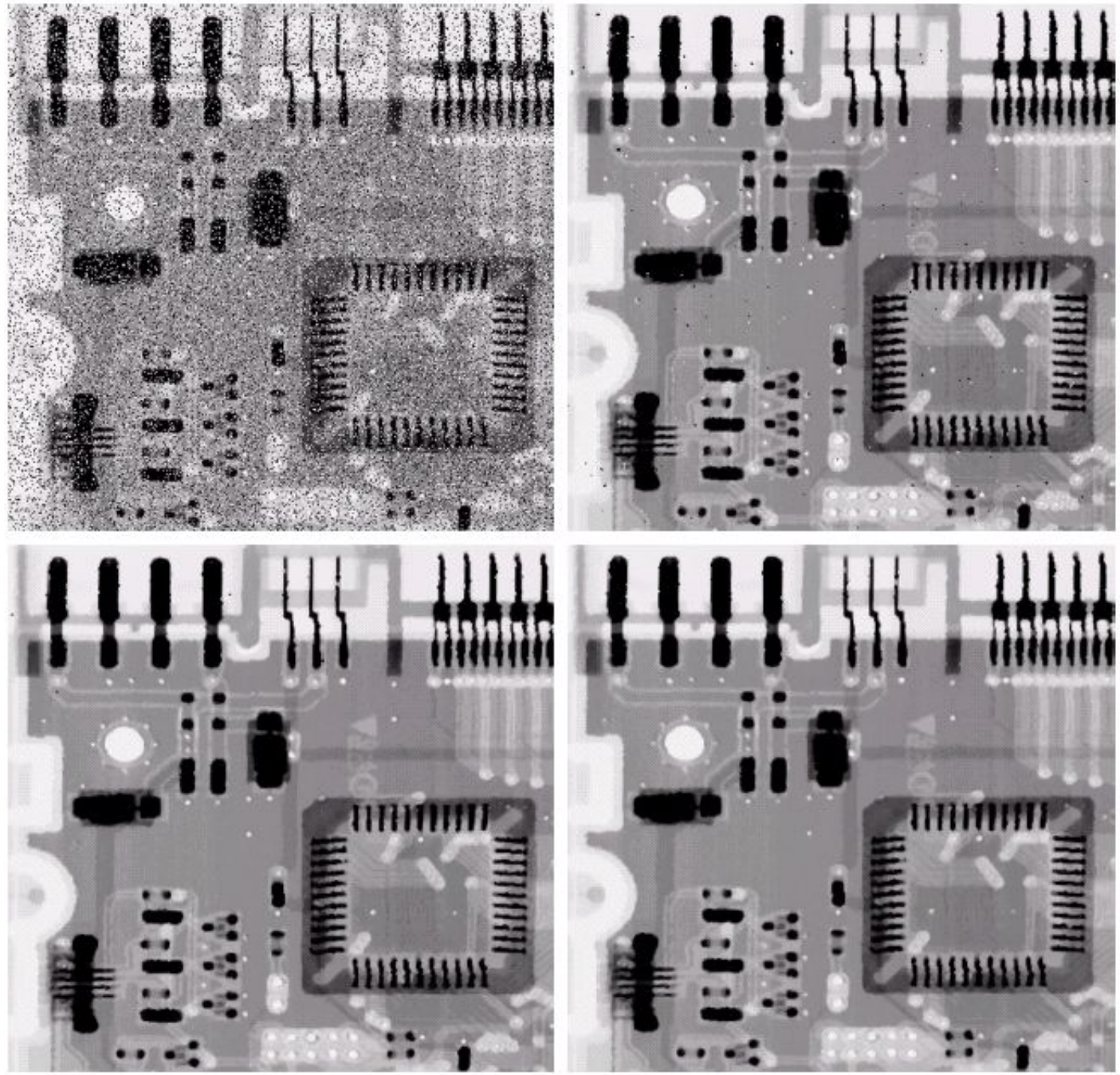


FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.
(b) Result of one pass with a median filter of size 3×3.
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.

Max and Min Filter

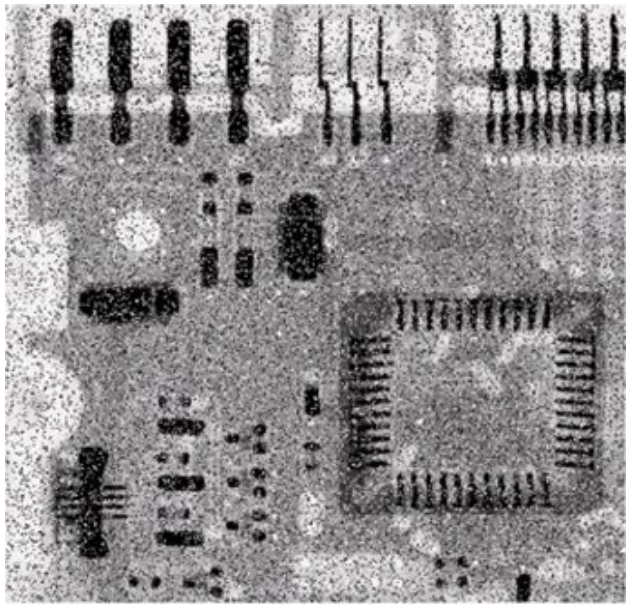
Max Filter:

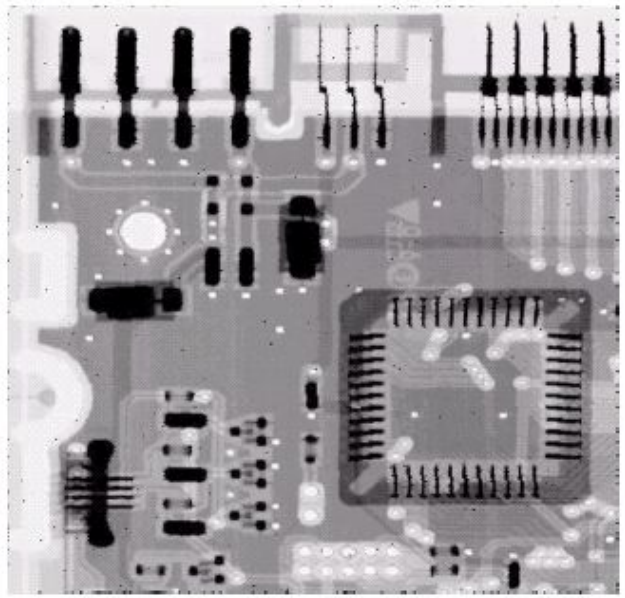
$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

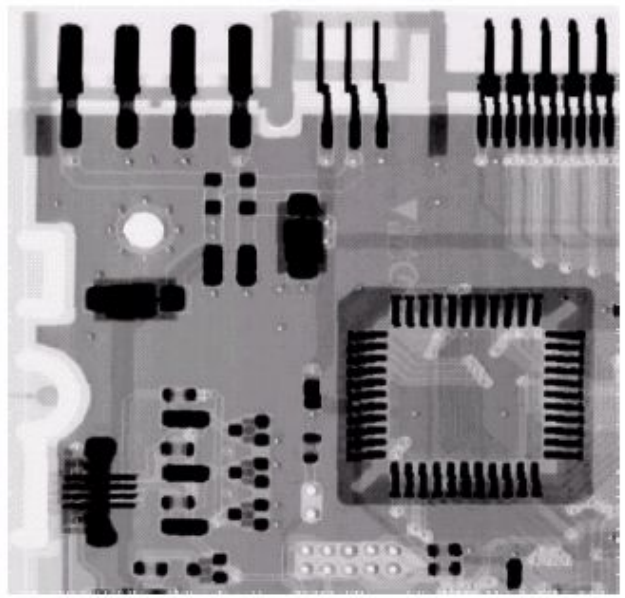
Min Filter:

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

Max filter is good for pepper noise and min is good for salt noise







a b

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

Midpoint Filter

Midpoint Filter:

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

Good for random Gaussian and uniform noise

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

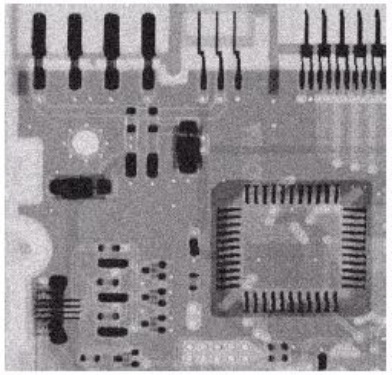
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

We can delete the $d/2$ lowest and $d/2$ highest grey levels

So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Noise Removal Examples (cont…)

Image Corrupted By Uniform Noise



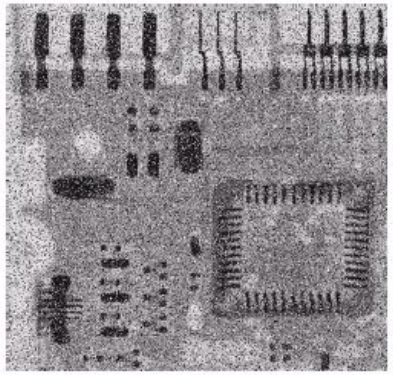
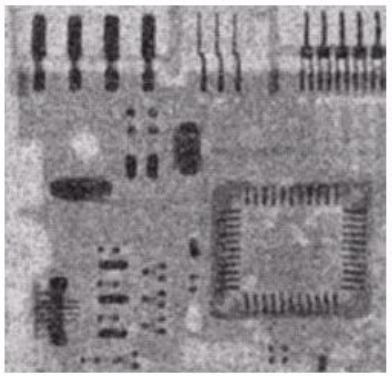
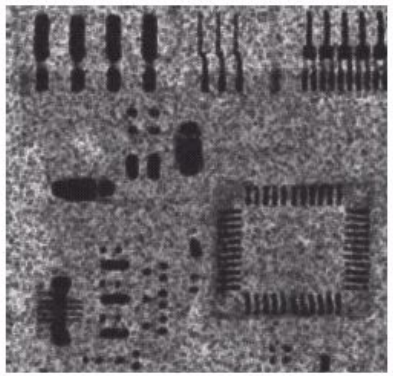


Image Further Corrupted By Salt and Pepper Noise

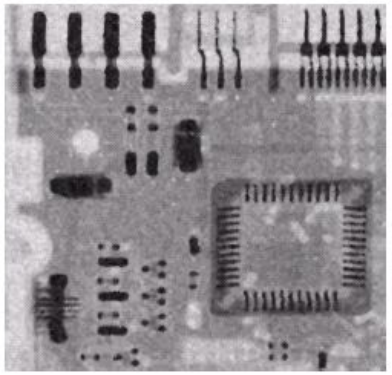
Filtered By 5*5 Arithmetic Mean Filter

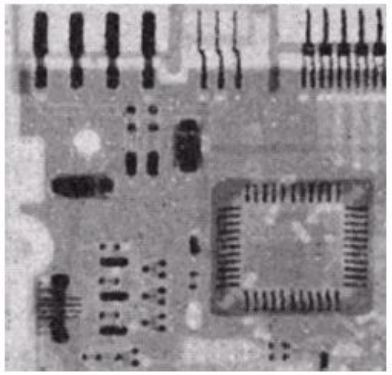




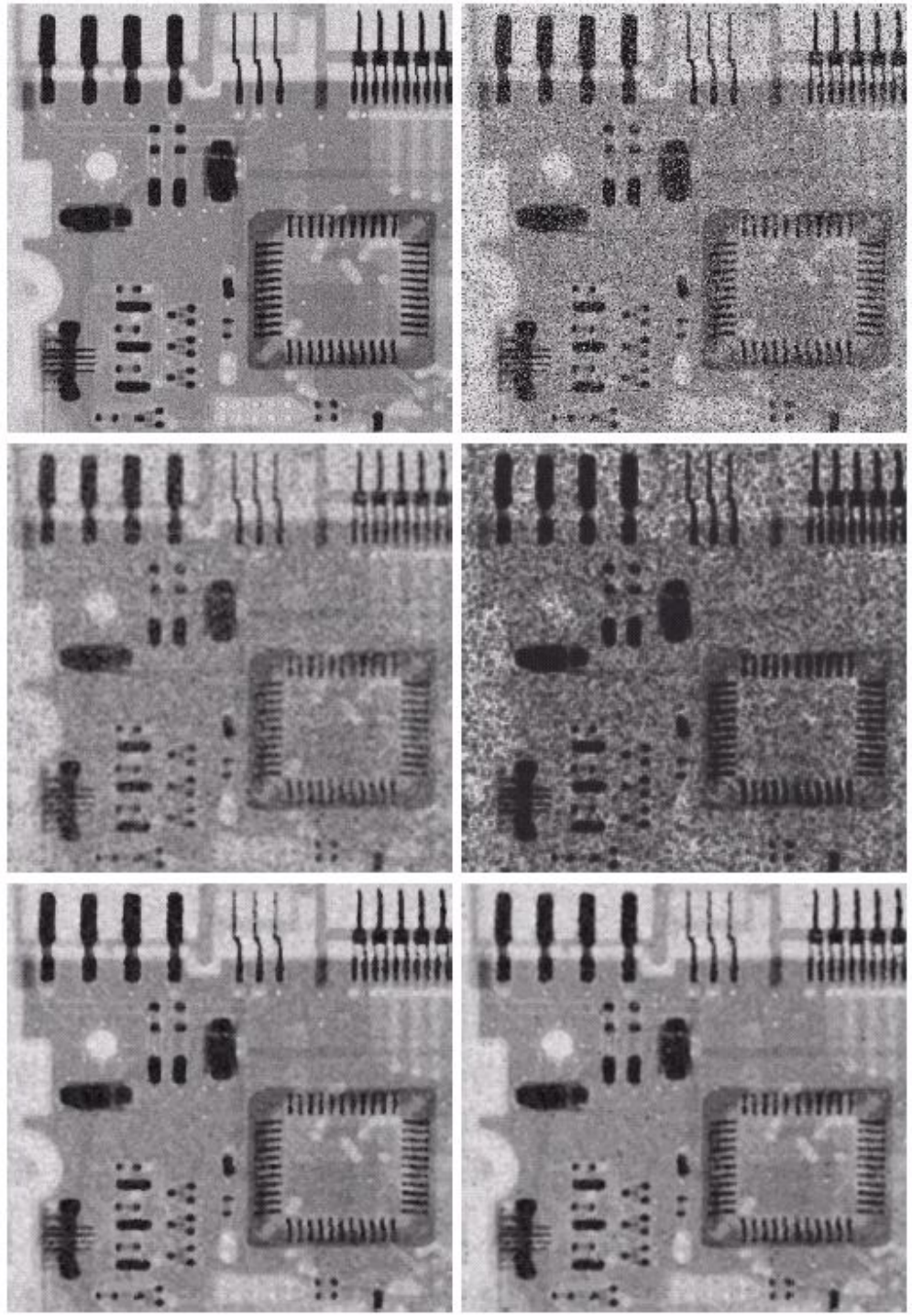
Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter





Filtered By 5*5 Alpha-Trimmed Mean 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 $d = 5$.

a b
c d

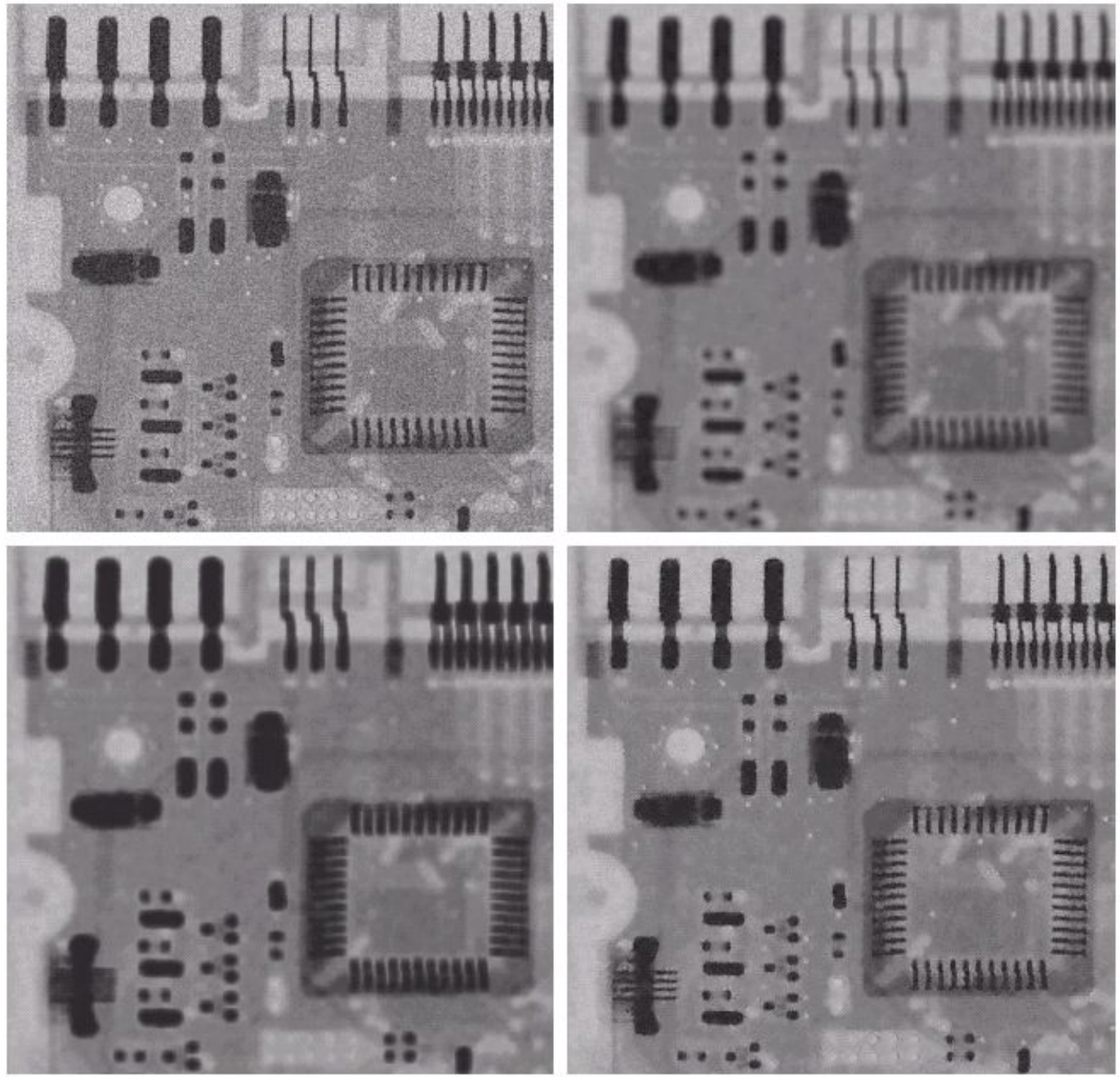
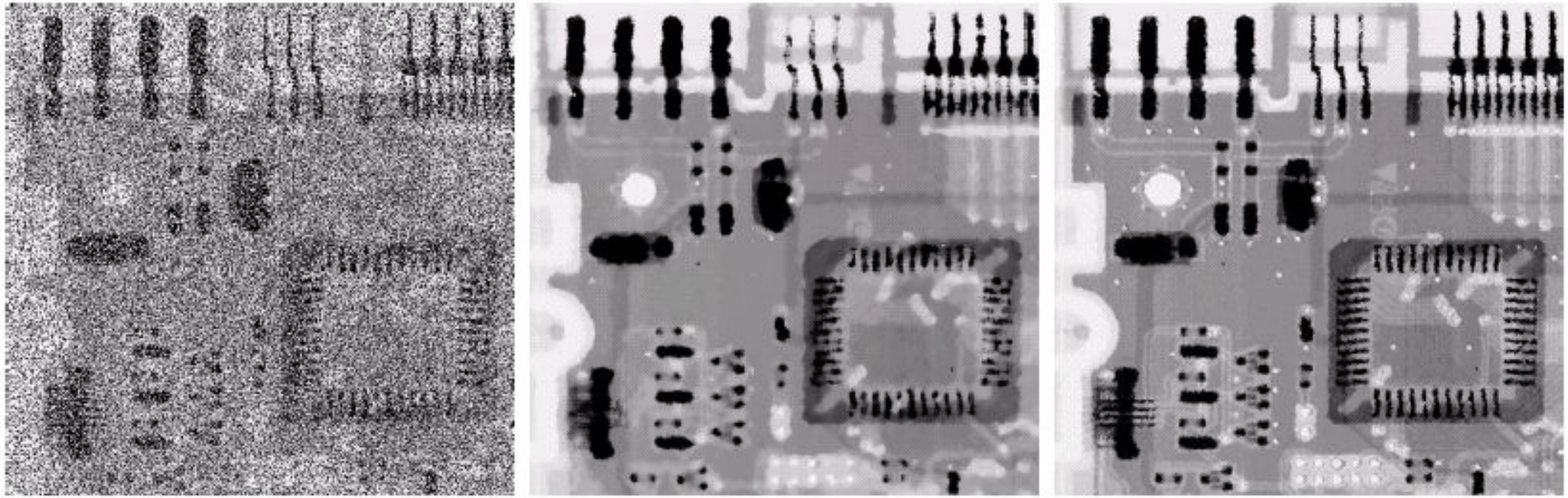


FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.



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 $S_{max} = 7$.

Adaptive Filters

The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another

The behaviour of **adaptive filters** changes depending on the characteristics of the image inside the filter region

We will take a look at the **adaptive median filter**

Adaptive Median Filtering

The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large

The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise

The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image

Adaptive Median Filtering (cont…)

Remember that filtering looks at each original pixel image in turn and generates a new filtered pixel

First examine the following notation:

- z_{min} = minimum grey level in S_{xy}
- z_{max} = maximum grey level in S_{xy}
- z_{med} = median of grey levels in S_{xy}
- z_{xy} = grey level at coordinates (x, y)
- S_{max} =maximum allowed size of S_{xy}

Adaptive Median Filtering (cont…)

Level A: $A1 = z_{med} - z_{min}$

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

If $A1 > 0$ and $A2 < 0$, Go to level B

Else increase the window size

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

Else output z_{med}

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

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

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

Else output z_{med}

Adaptive Median Filtering (cont…)

The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:

- Remove impulse noise
- Provide smoothing of other noise
- Reduce distortion

Adaptive Filtering Example

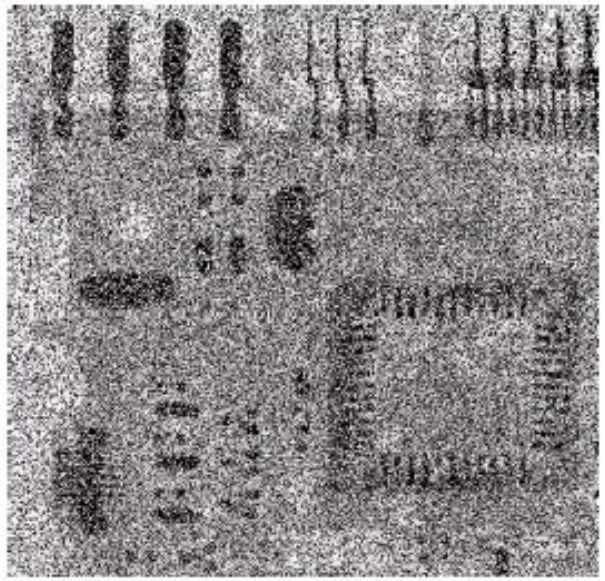
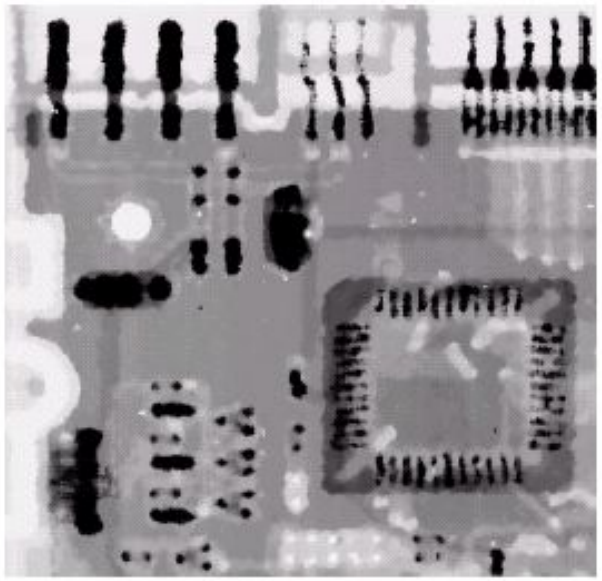
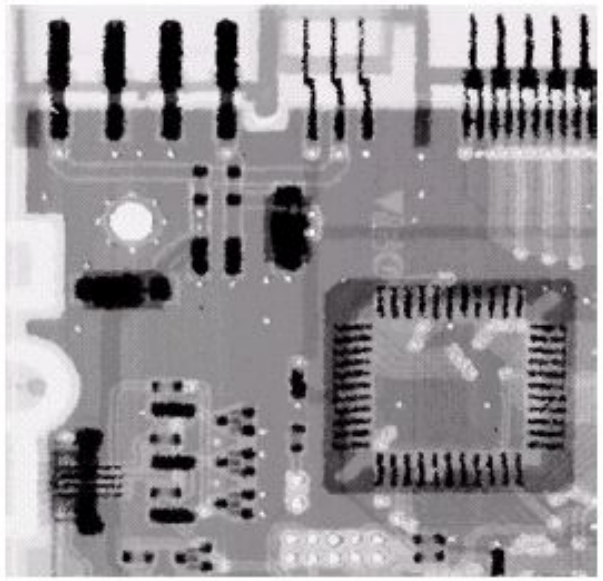


Image corrupted by salt and pepper noise with probabilities $P_a = P_b$=0.25



Result of filtering with a 7 * 7 median filter



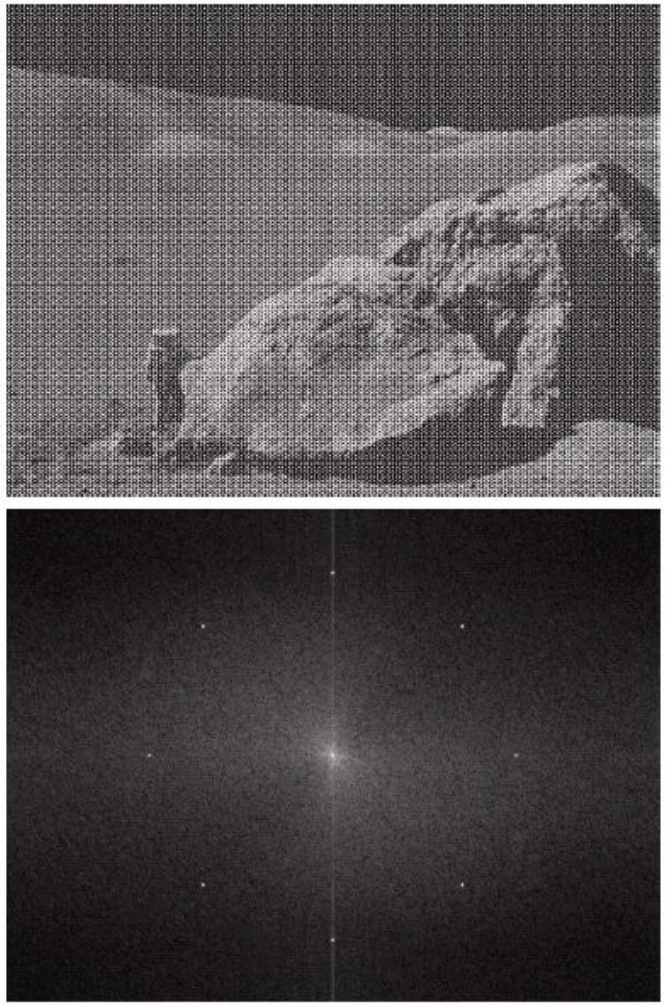
Result of adaptive median filtering with i = 7

Periodic Noise

Typically arises due to electrical or electromagnetic interference

Gives rise to regular noise patterns in an image

Frequency domain techniques in the Fourier domain are most effective at removing periodic noise



Periodic Noise Reduction by Frequency Domain Filtering

LPF & HPF □ Image Enhancement

1) Bandreject Filters

1-1) Ideal :

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

D(u,v): distance from the origin of the centered frequency rectangle

W: bandwidth

D_0: radial center

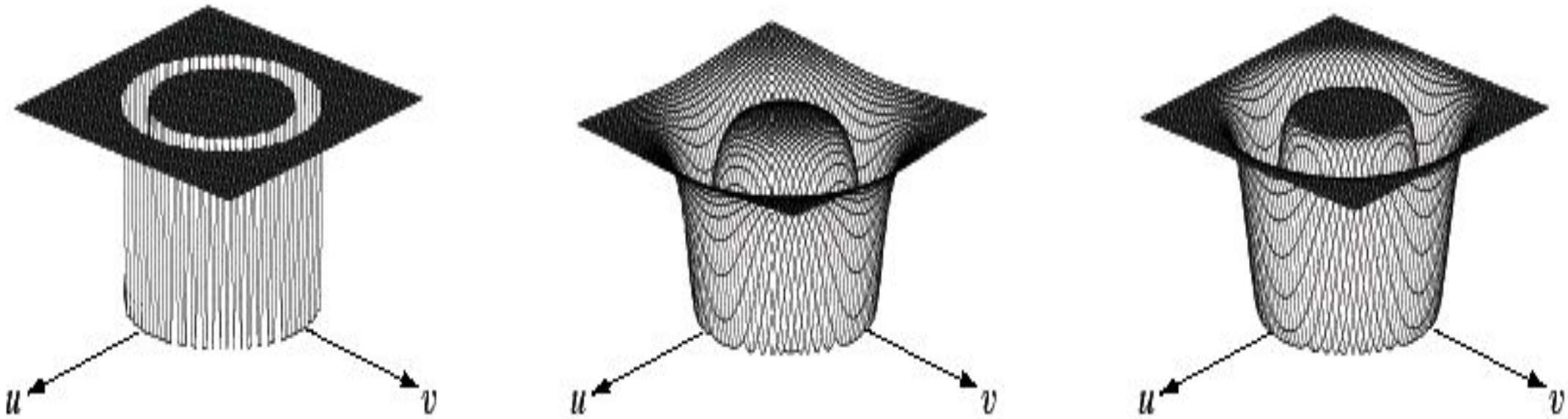
Periodic Noise Reduction by Frequency Domain Filtering

1-2) Butterworth of order n :

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

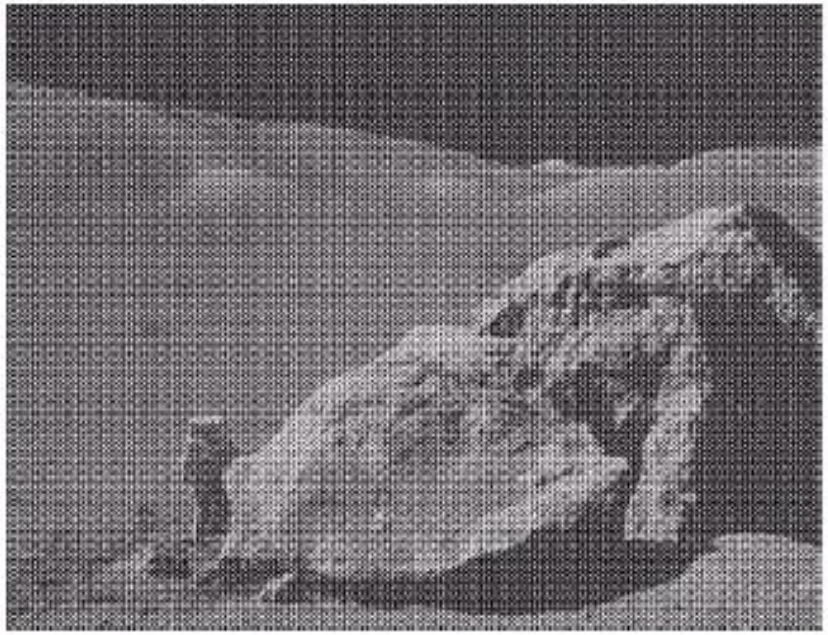
1-3) Gaussian :

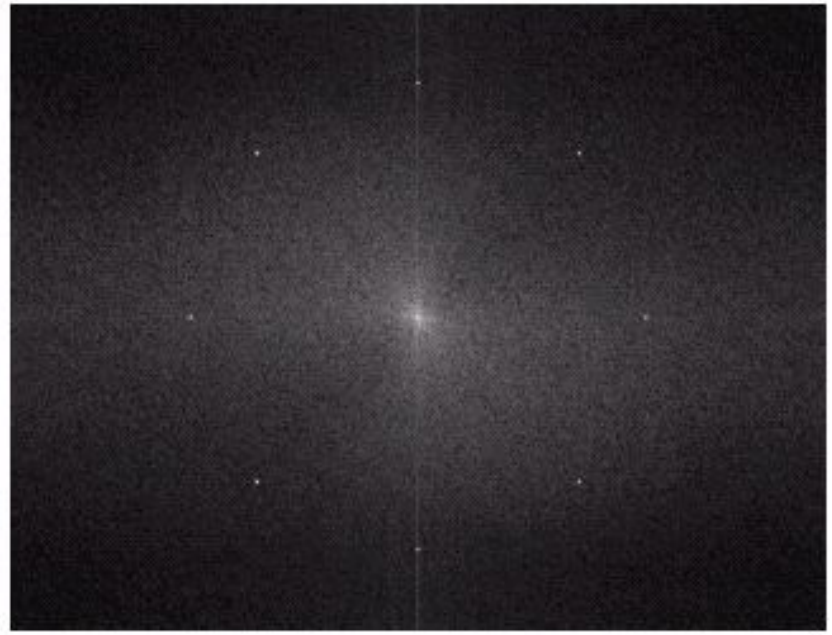
$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0{}^2}{D(u,v)W}\right]^2}$$

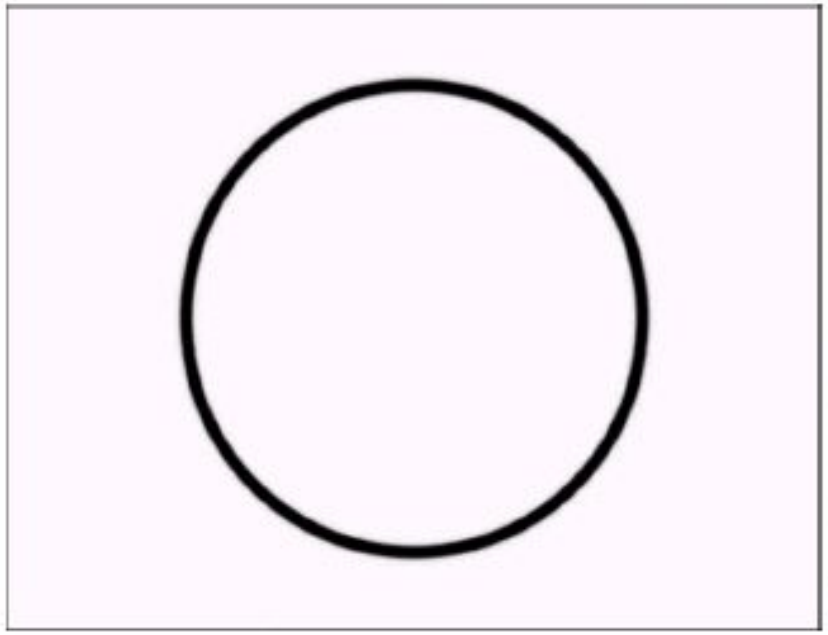


a b c

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.)

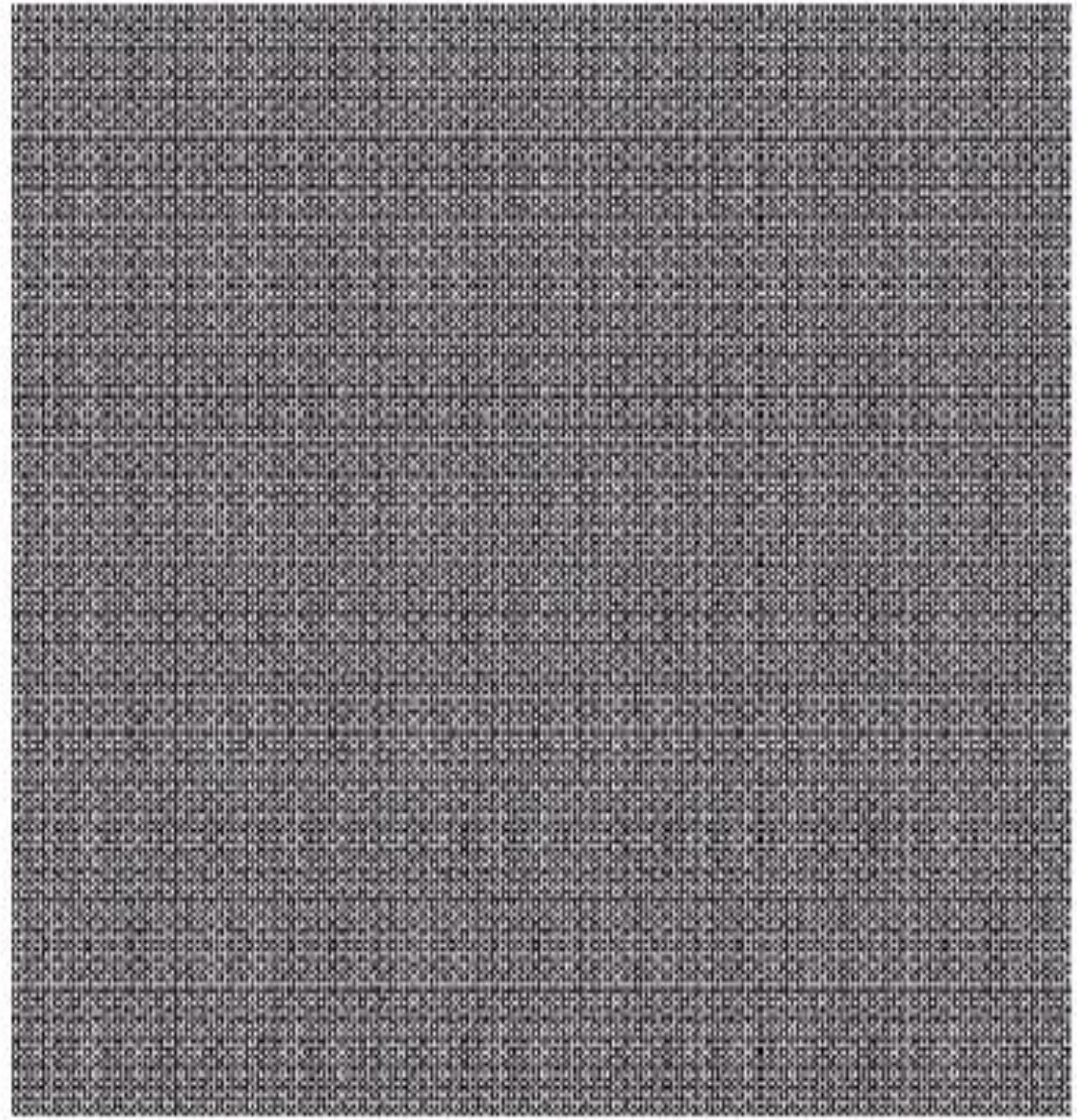
Periodic Noise Reduction by Frequency Domain Filtering

2) Bandreject Filters

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

It helps isolate the noise pattern.

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



Periodic Noise Reduction by Frequency Domain Filtering

3) Notch Filters

3-1) Ideal :

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

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

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

The center of the frequency rectangle has been shifted to the point (M/2,N/2)

Periodic Noise Reduction by Frequency Domain Filtering

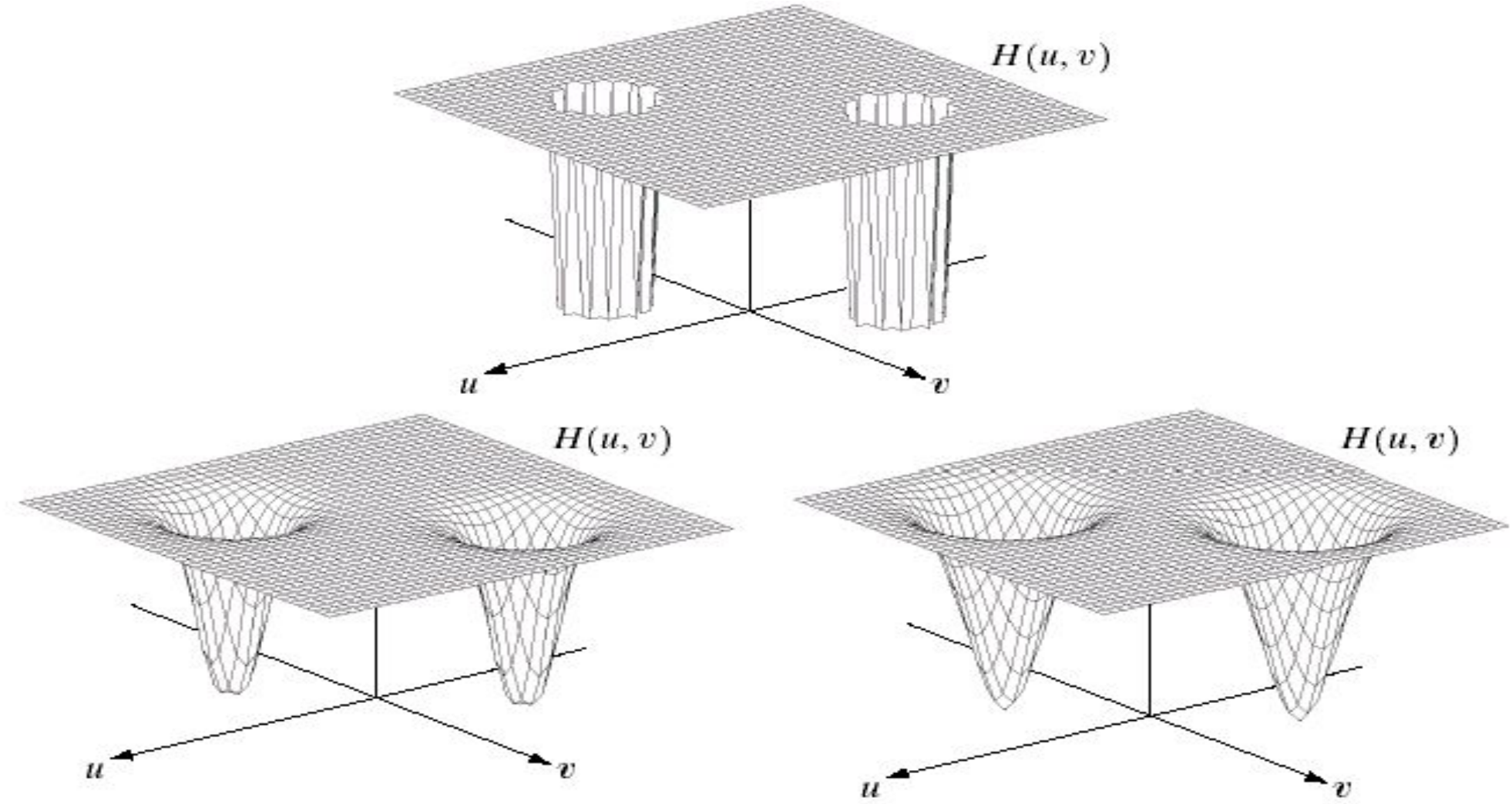
Notch Filters

3-2) Butterworth of order n :

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

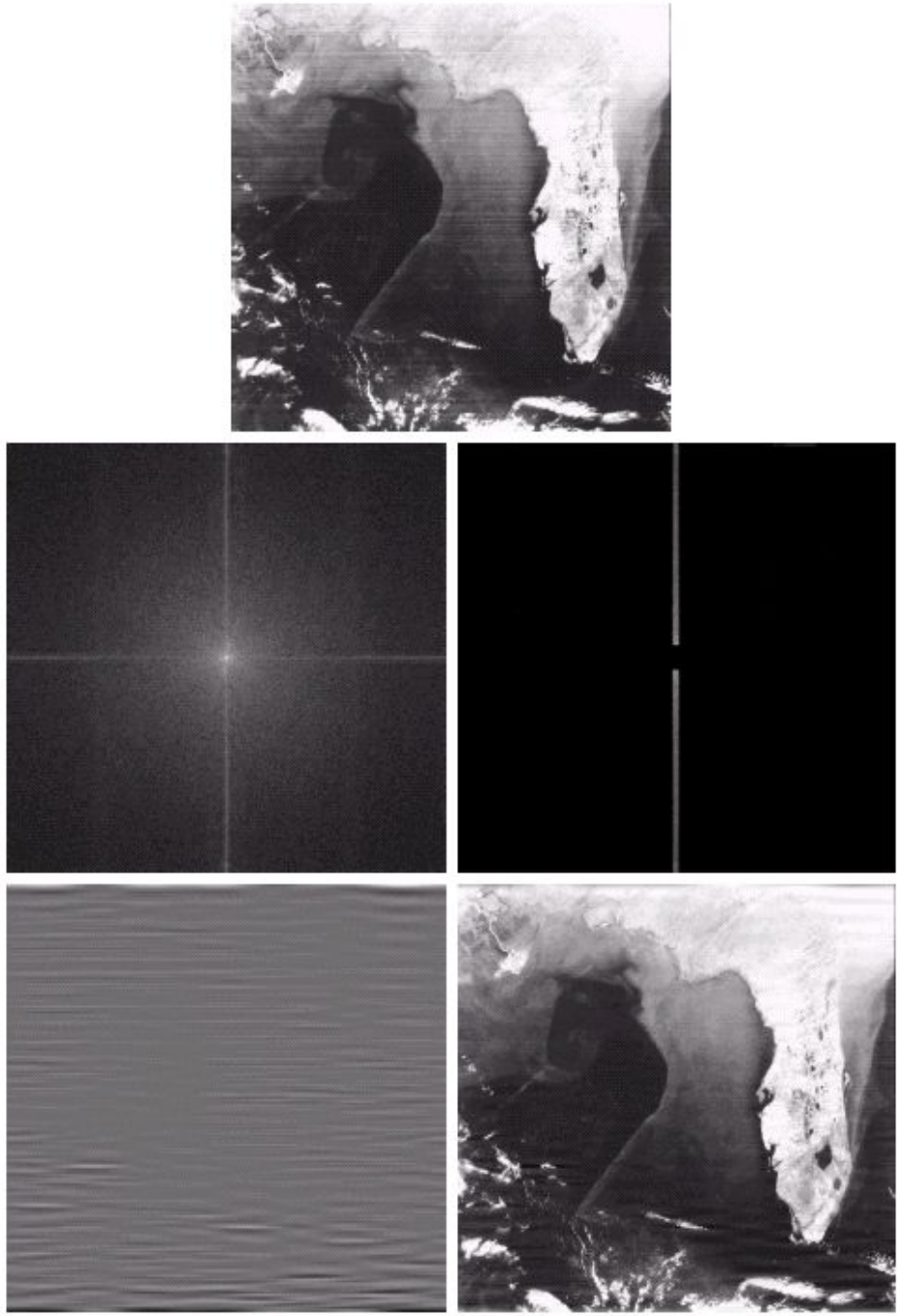
3-3) Gaussian :

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



a
b c

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



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.)

Periodic Noise Reduction by Frequency Domain Filtering

4) Optimum Notch Filters

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

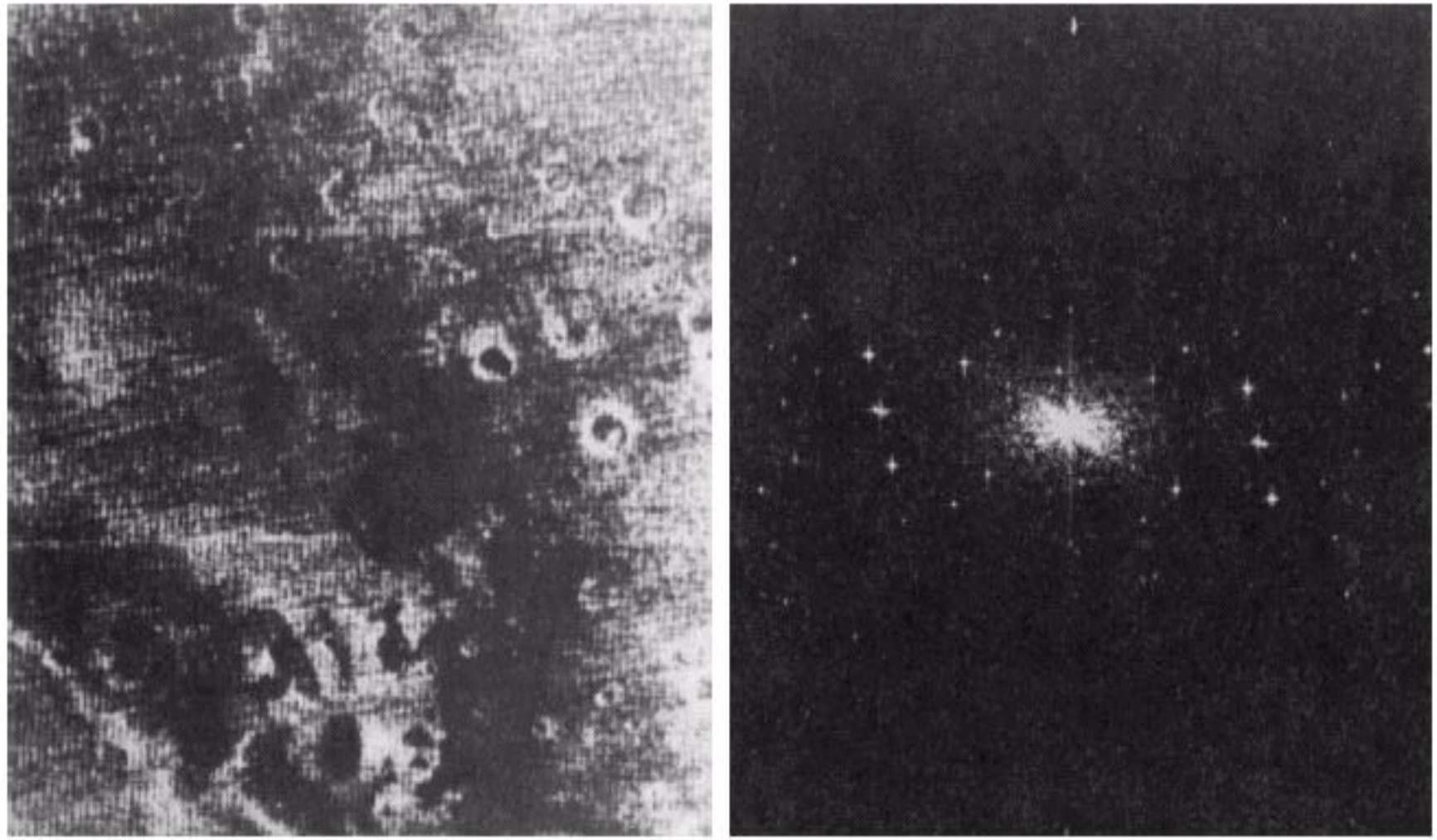
$$\eta(x,y) = F^{-1}\{H(u,v)G(u,v)\}$$

$$w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \overline{g}(x,y)\overline{\eta}(x,y)}{\overline{\eta^2}(x,y) - \overline{\eta}^2(x,y)}$$

To obtain w(x,y) the goal is to minimize the variance in the neighborhood of x,y in the image.

a b

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



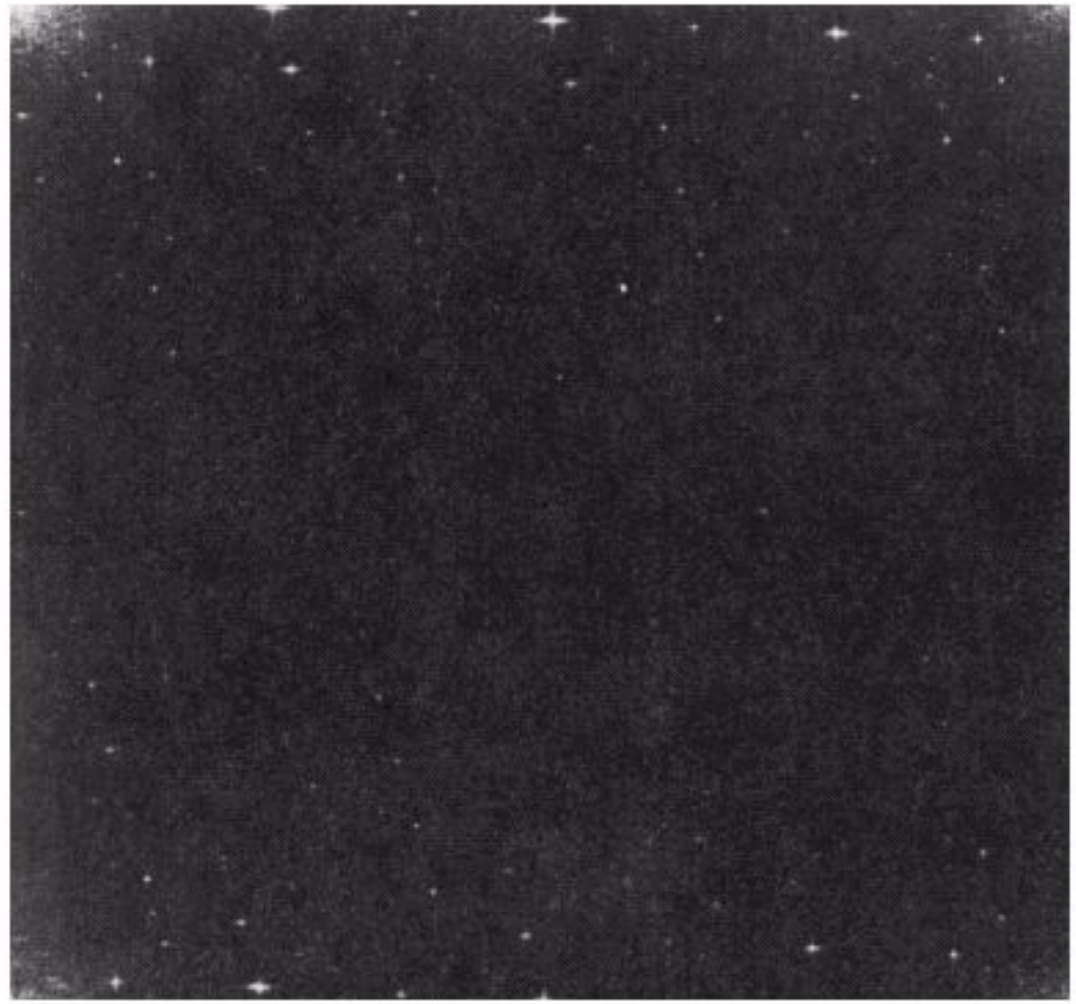
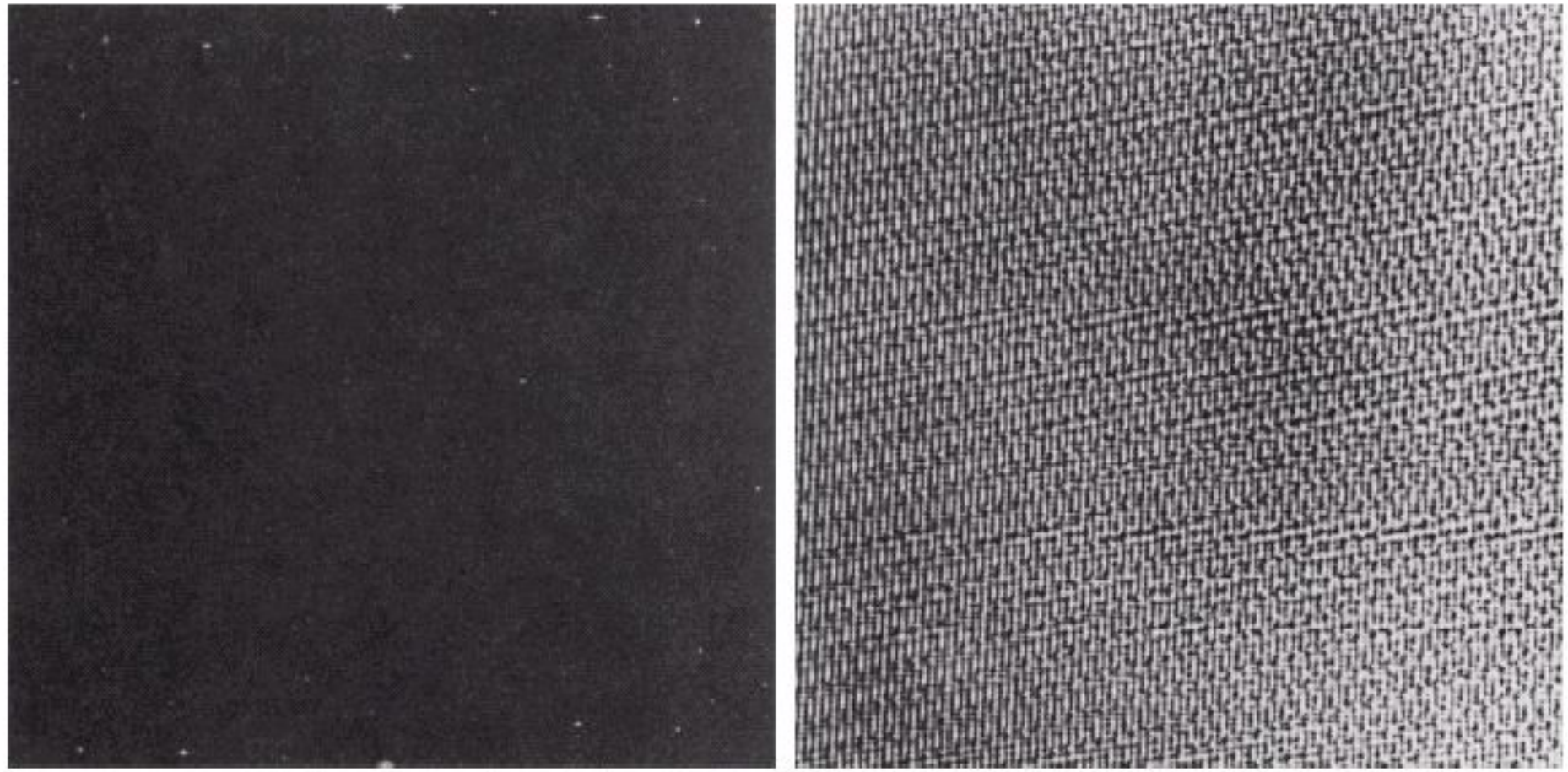


FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)



a b

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

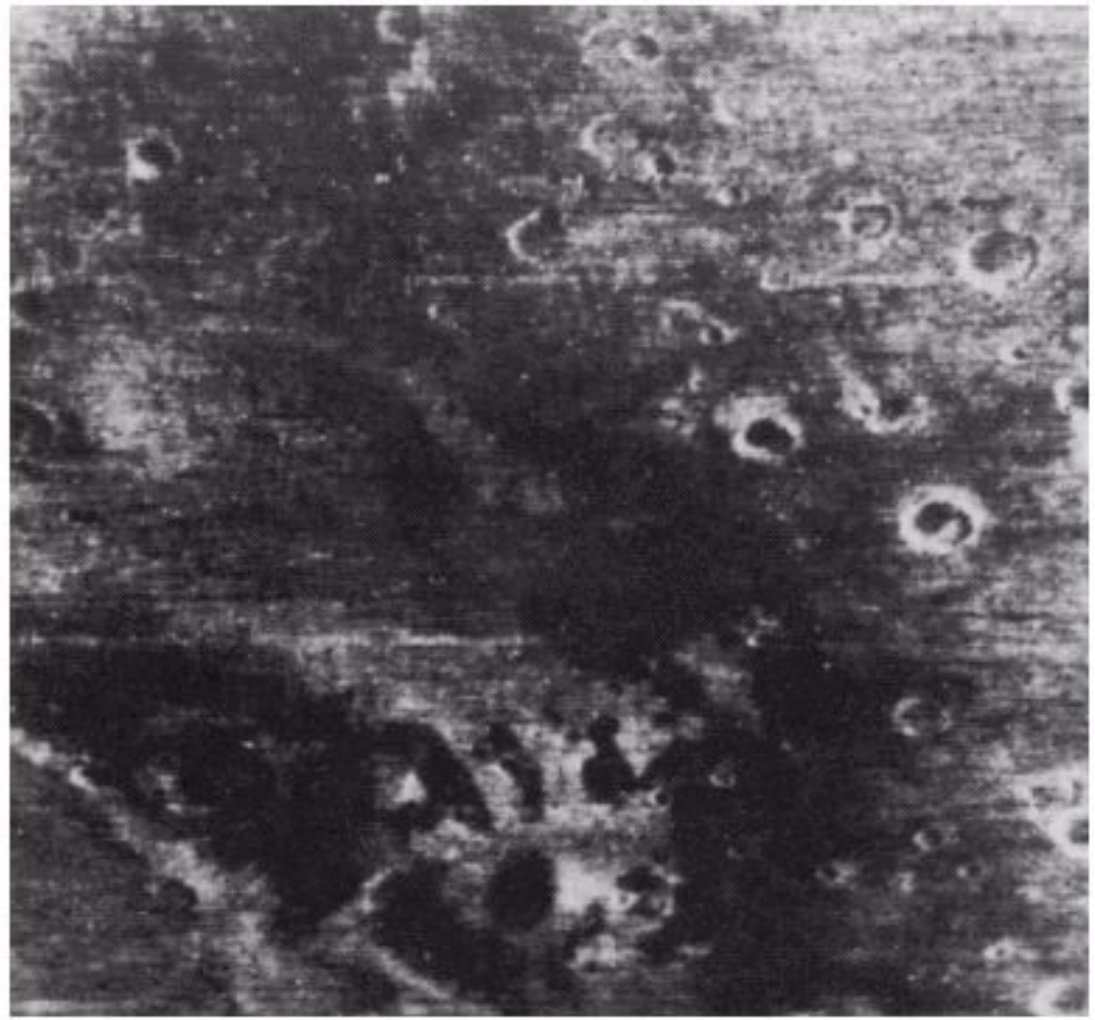


FIGURE 5.23 Processed image. (Courtesy of NASA.)

Linear, Position-Invariant Degradations

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

Estimation the Degradation Function

1) Estimation by Image Observation

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

Assuming that the effect of noise is negligible.

Linear, Position-Invariant Degradations

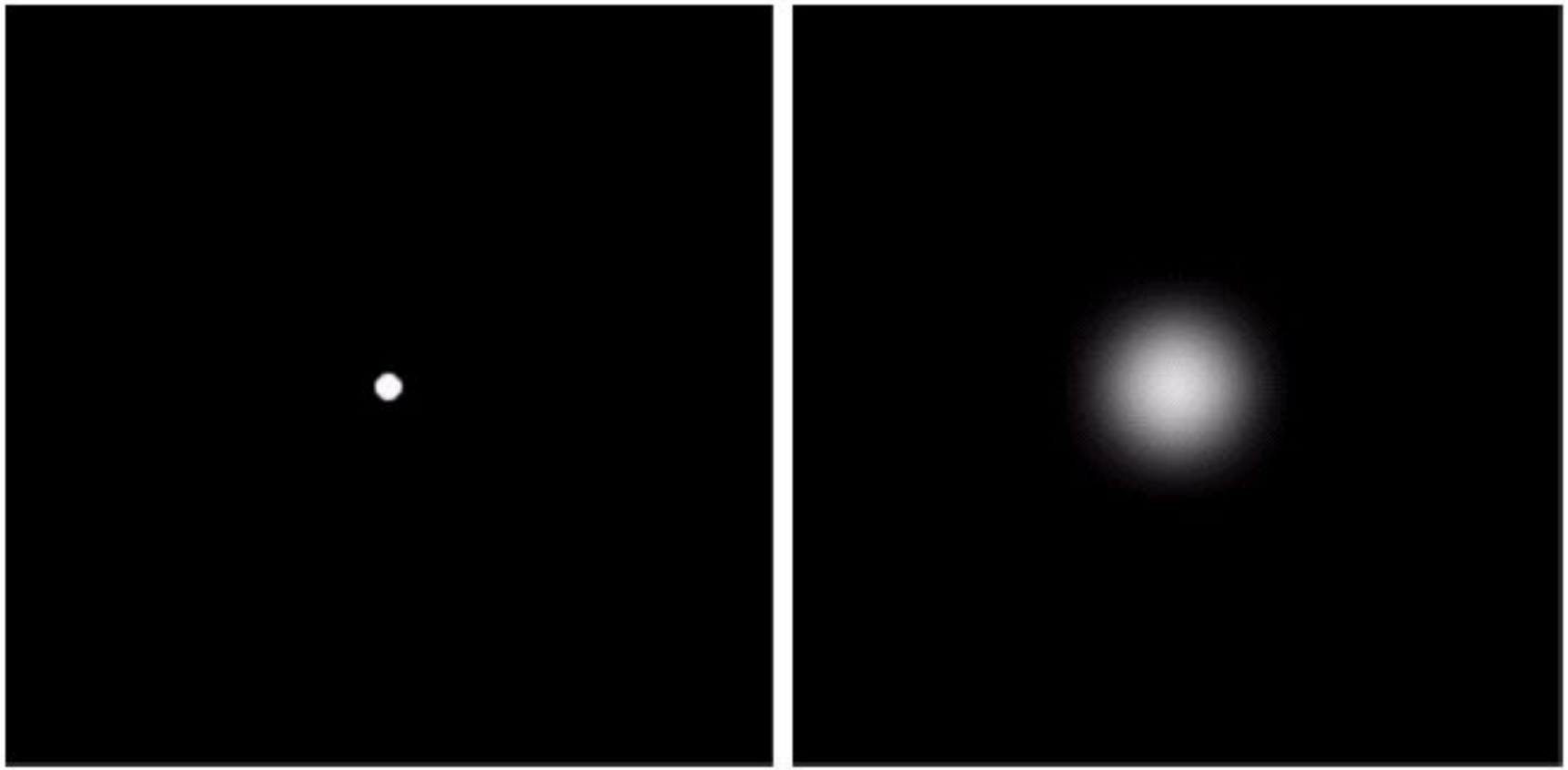
$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$
$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Estimation the Degradation Function

2) Estimation by Experimentation

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

A : impulse Fourier transform which is constant.



a b

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

Linear, Position-Invariant Degradations

Estimation the Degradation Function

3) Estimation by modeling

Turbulence model :

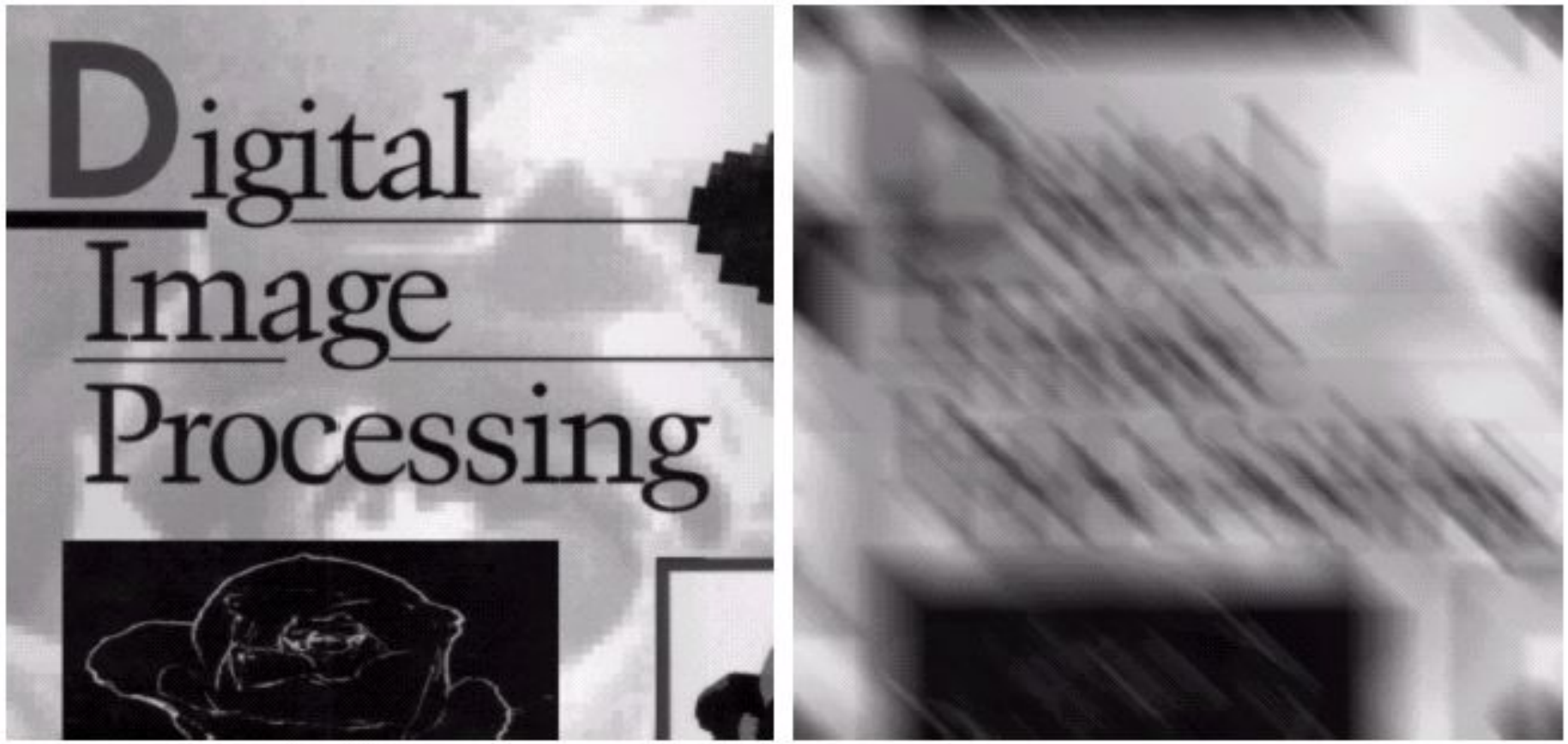
$$H(u,v) = e^{-K\left(u^2+v^2\right)^{5/6}}$$

Mathematical model :

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

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.)



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 $T = 1$.

How to get $\hat{F}(u,v)$ from degraded image G(u,v) :

1) Inverse Filtering

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

We should know N(u,v) to use this method.

We should use this method near origin because H(u,v) is near zero in other areas.

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.

How to get $\hat{F}(u,v)$ from degraded image G(u,v) :

2) Wiener (Minimum mean square Error) Filtering

$$e^2 = E\left\{\left(f - \hat{f}\right)^2\right\} \to 0$$

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

H(u,v) : degradation function

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

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

If $S_f(u,v)$ is not known :

$$e^2 = E\left\{\left(f - \hat{f}\right)^2\right\} \to 0$$

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

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.

a b c
d e f
g h i

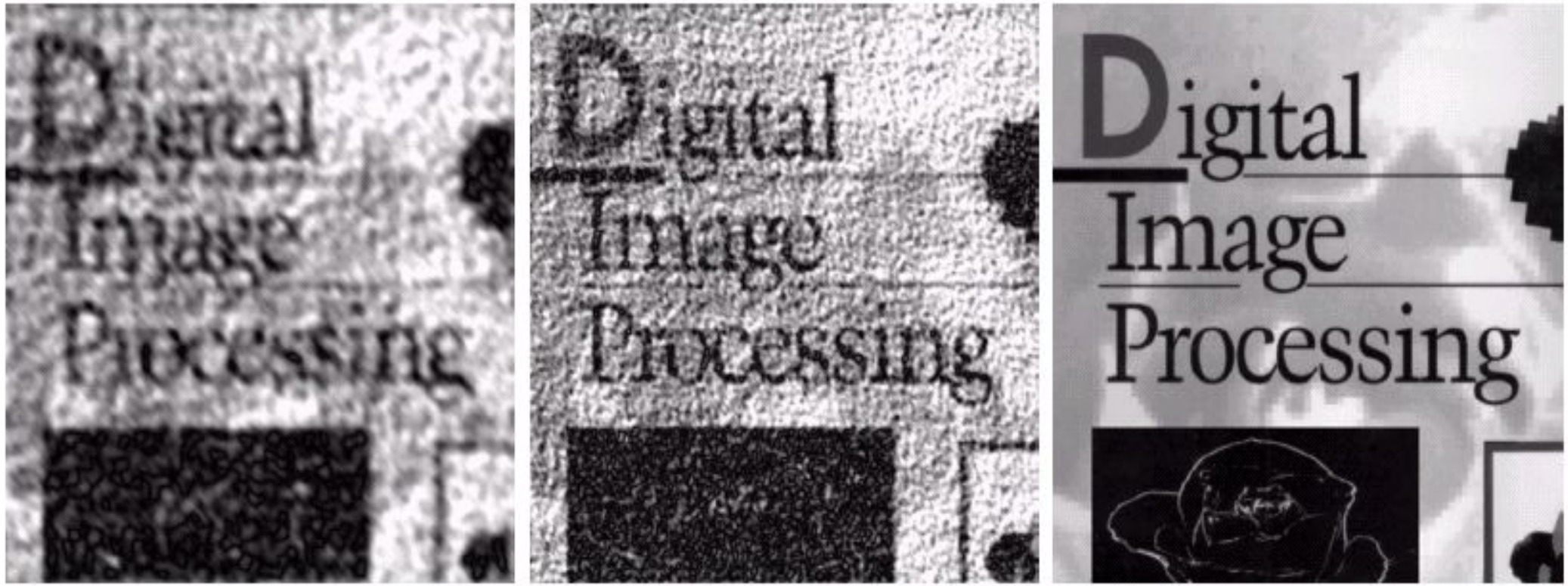
FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note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

How to get $\hat{F}(u,v)$ from degraded image G(u,v) :

3) Constrained Least Squares Filtering

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

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



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.

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.

Geometric Transformations : (rubber-sheet transformations)

1) Spatial Transformations

2) Gray-level Transformations

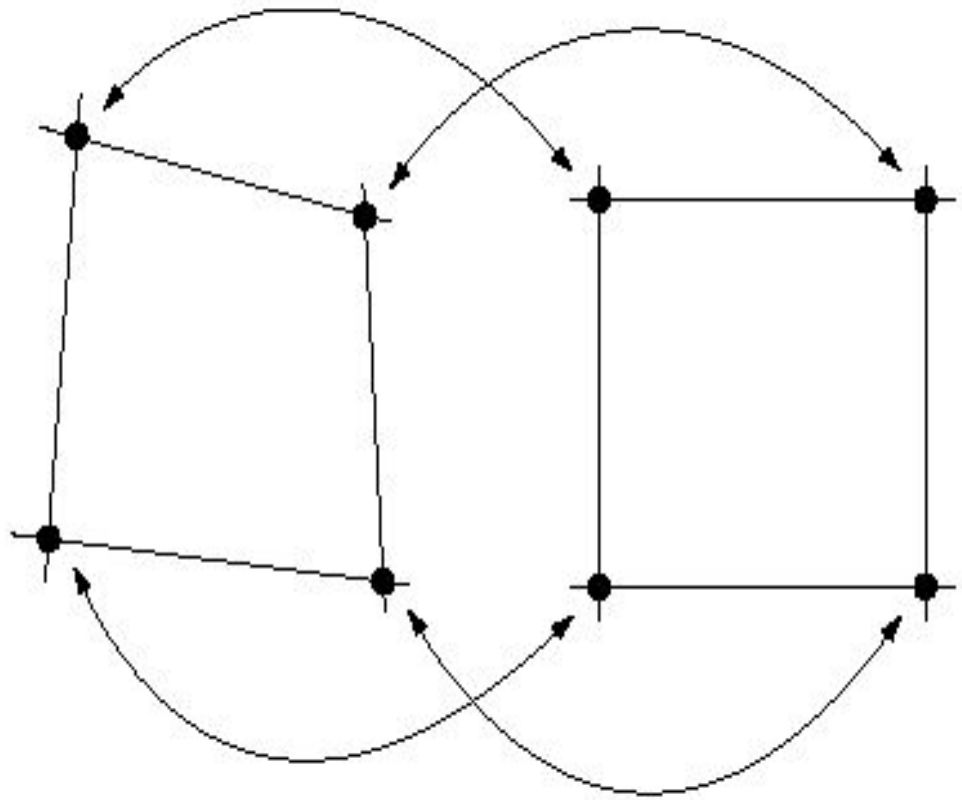


FIGURE 5.32
Corresponding tiepoints in two image segments.

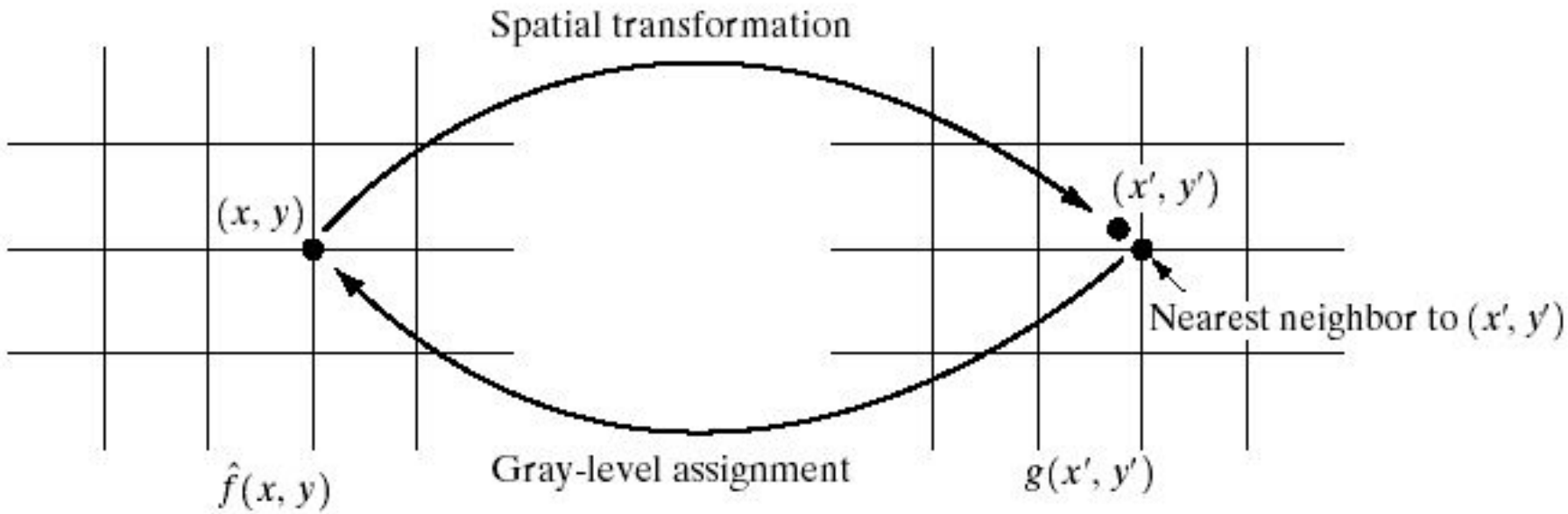
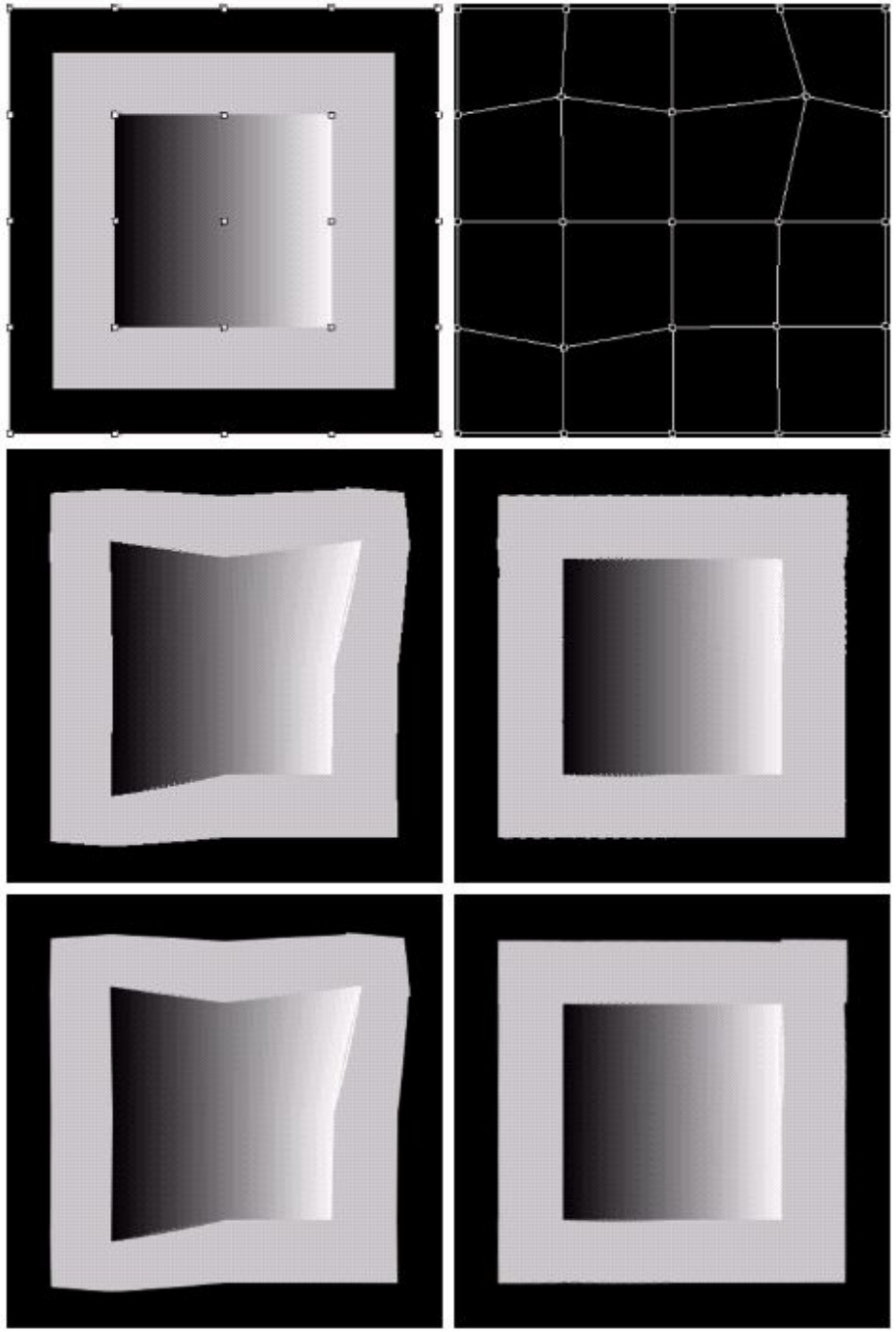
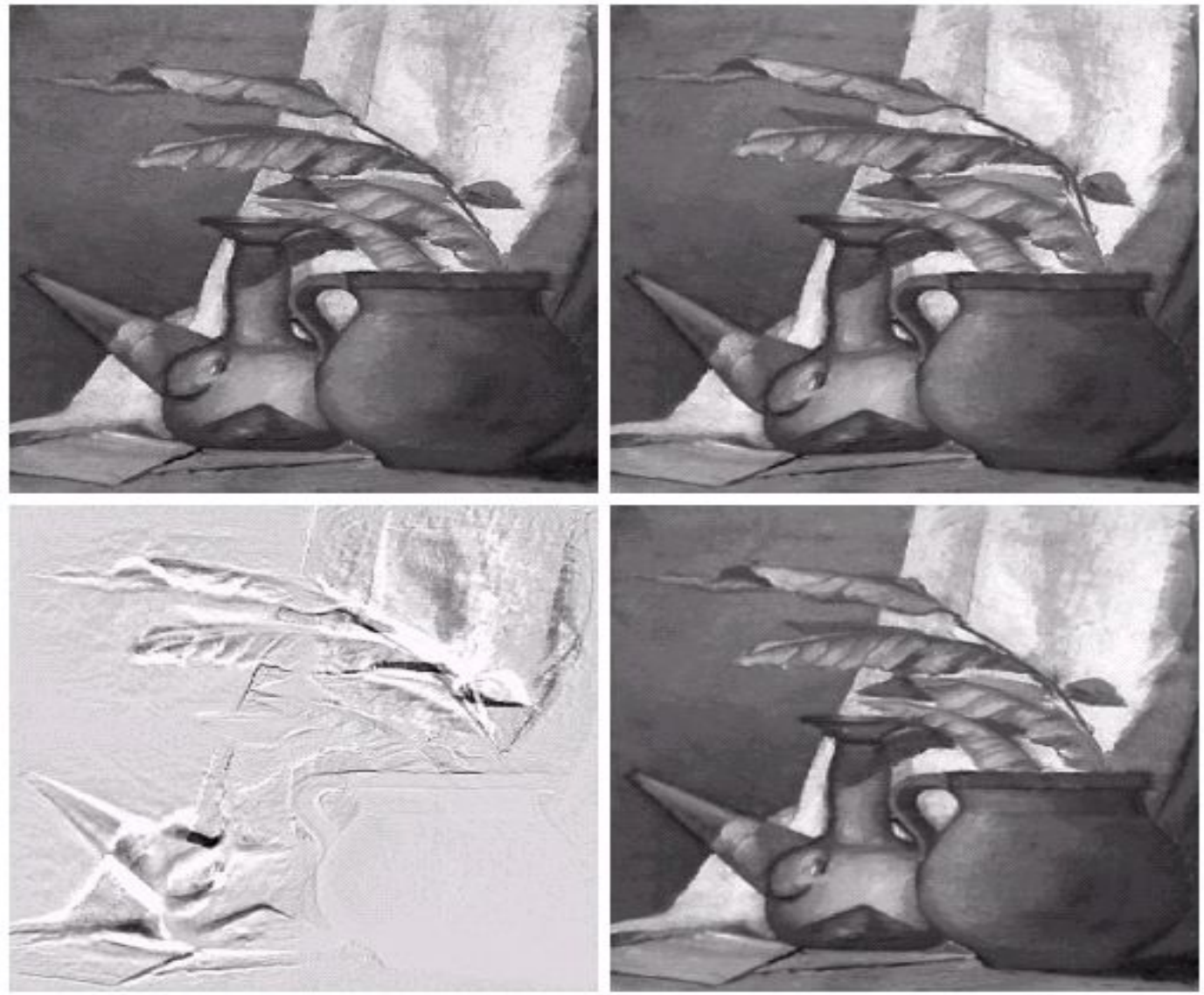


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.



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.



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.