Chapter 5
Image Restoration and Reconstruction

General Degradation Model
Noise Models and Properties
Noise Parameter Estimation
Spatial Filtering for Noise Suppression

Enhancement or Restoration?

- The course focus has so far been on – *image enhancement*
- Contrast stretching, histogram equalization, range compression = techniques for subjective improvement of the perceived image quality
- *Image restoration* is mostly an objective process, aiming at the recovery of the original image under the assumption of known degradation and with respect to a specific criterion
- Degradations during image acquisition and reproduction are not considered in this course

General Degradation Model

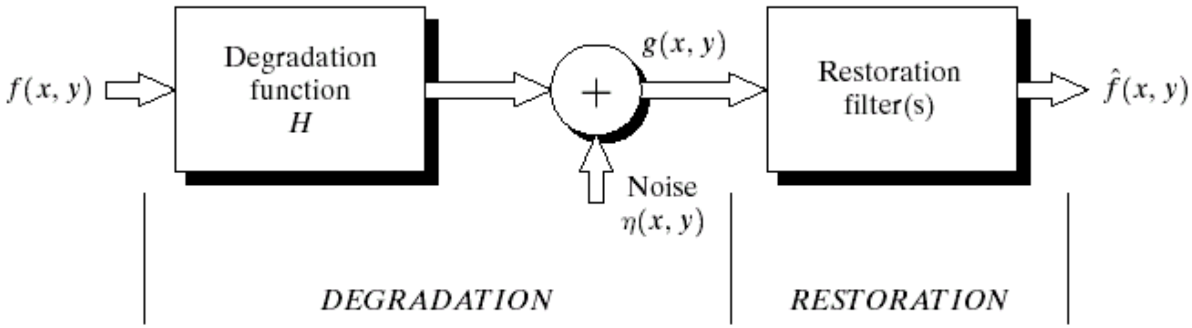


FIGURE 5.1 A model of the image degradation/restoration process.

<u>Assumption</u>: *H* is linear and position-invariant

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

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

<u>Goal</u>: design restoration filter(s) s.t. $\hat{f}(x,y)$ is close to $f(x,y)$

Properties of Noise Models

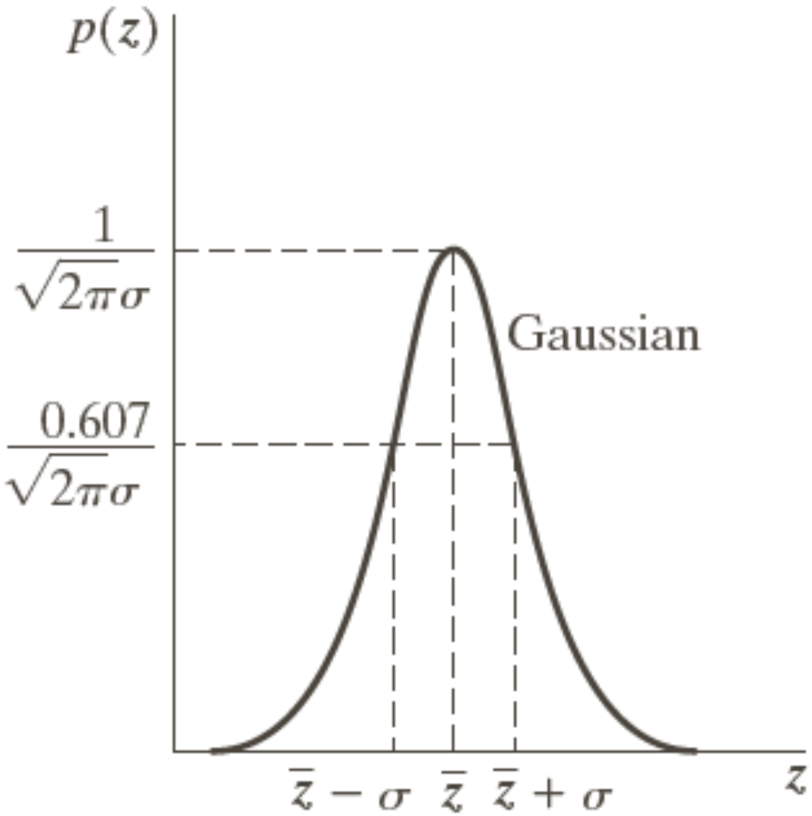
- Assumptions:
  - noise is independent of spatial coordinates OR periodic
  - noise is not correlated with the image

- Spatially independent noise can be treated as a random variable described by a probability density function (PDF)

- The choice of a filter depends on the assumed PDF

- The PDF assumption also allows for noise parameters to be estimated from the image

Gaussian Noise

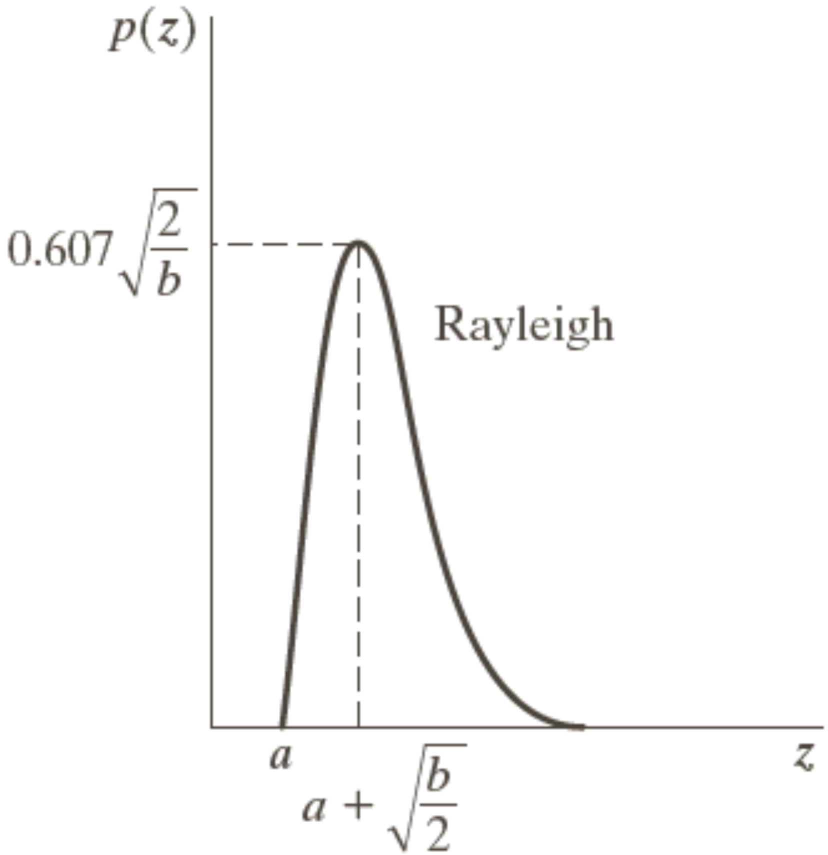


$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

μ – mean (average) grey level
σ – standard deviation of z

- Caused by electronic circuit noise, sensor noise
- Commonly used because of convenient properties

Rayleigh Noise



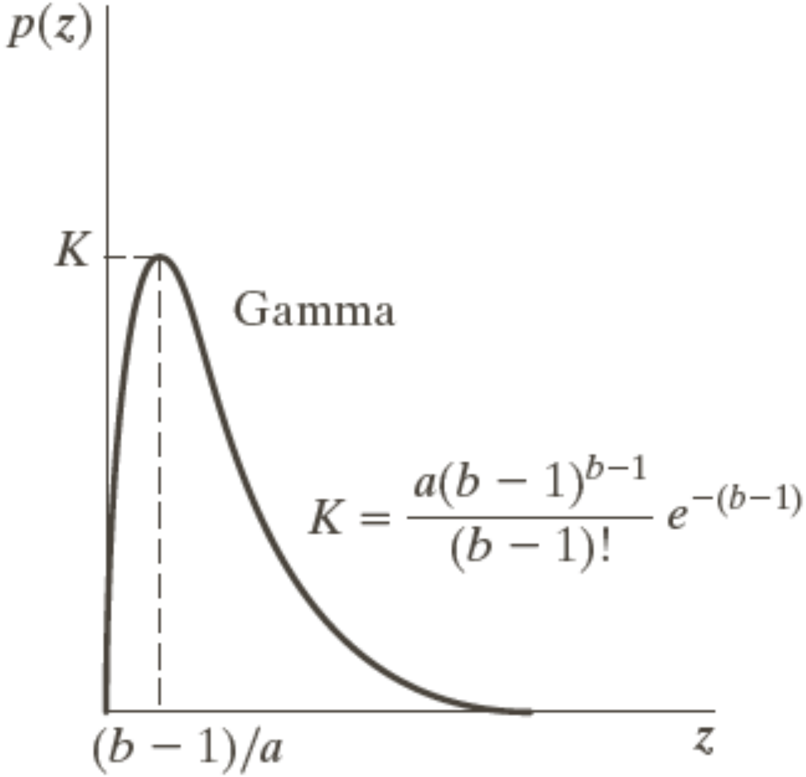
$$p(z) = \begin{cases} \frac{2}{b} e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

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

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

- Magnitude of a vector with 2 i.i.d. Gaussian components
- Used in range imaging

Gamma Noise



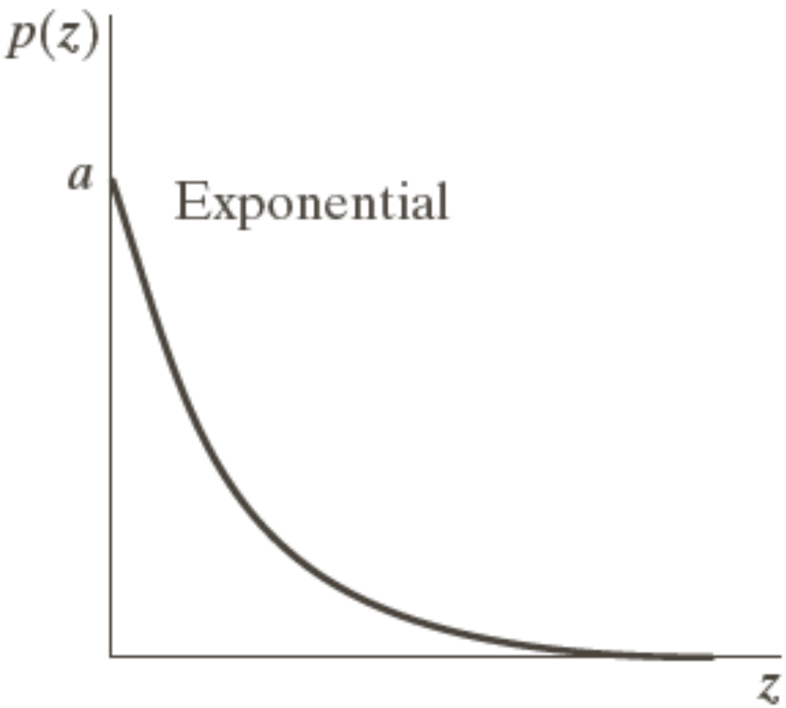
$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$$\mu = b/a$$

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

- Also known as Erlang noise
- Occurs in laser imaging

Exponential Noise



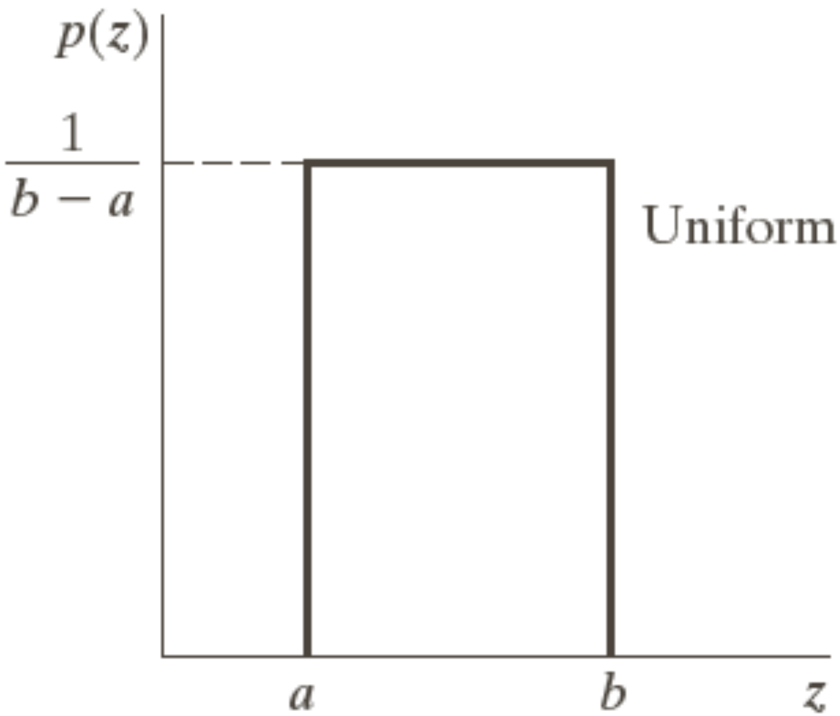
$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

$$\mu = 1/a$$

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

- Special case of gamma noise
- Occurs in laser imaging

Uniform Noise



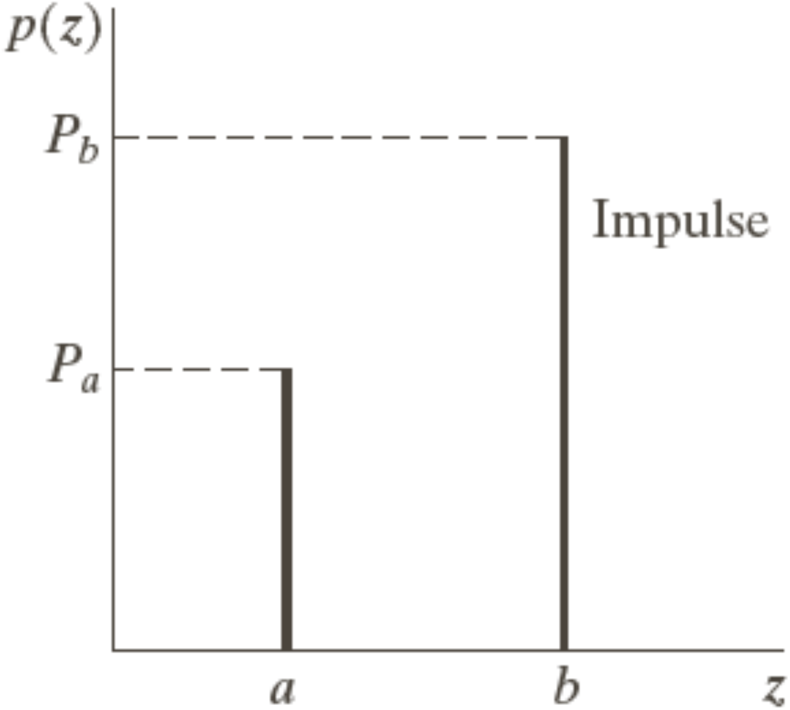
$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

$$\mu = (a+b)/2$$

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

- Occurs during quantization
- Useful for stochastic simulations

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 a or b are zero, *unipolar* noise, otherwise *bipolar*
- Typically of large scale, appears as scattered bright and dark dots

Noise Effect Examples

Consider the following input image:

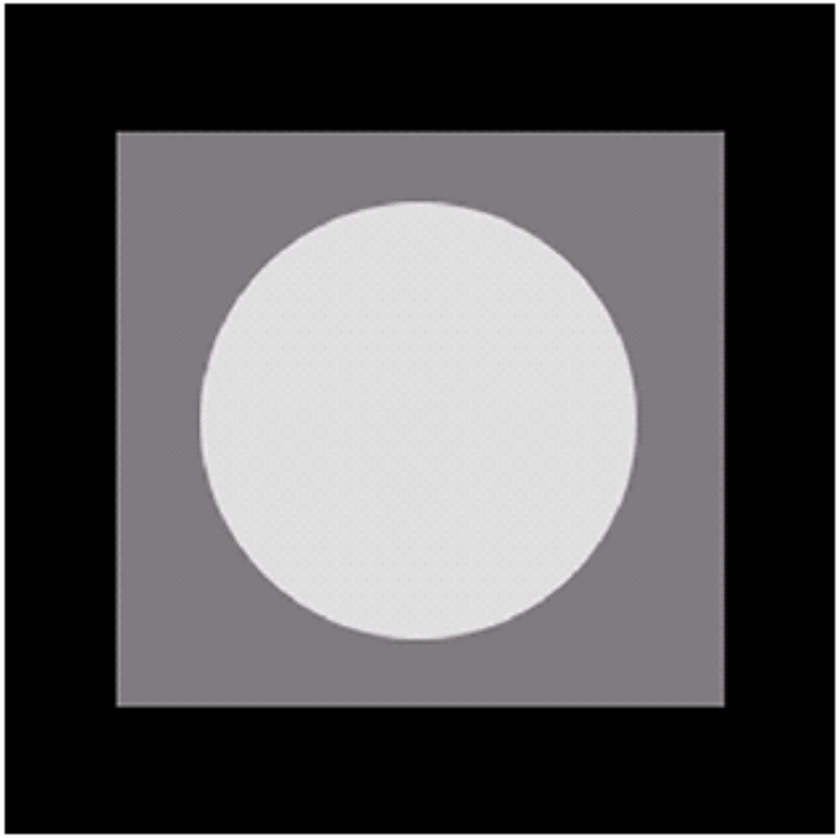
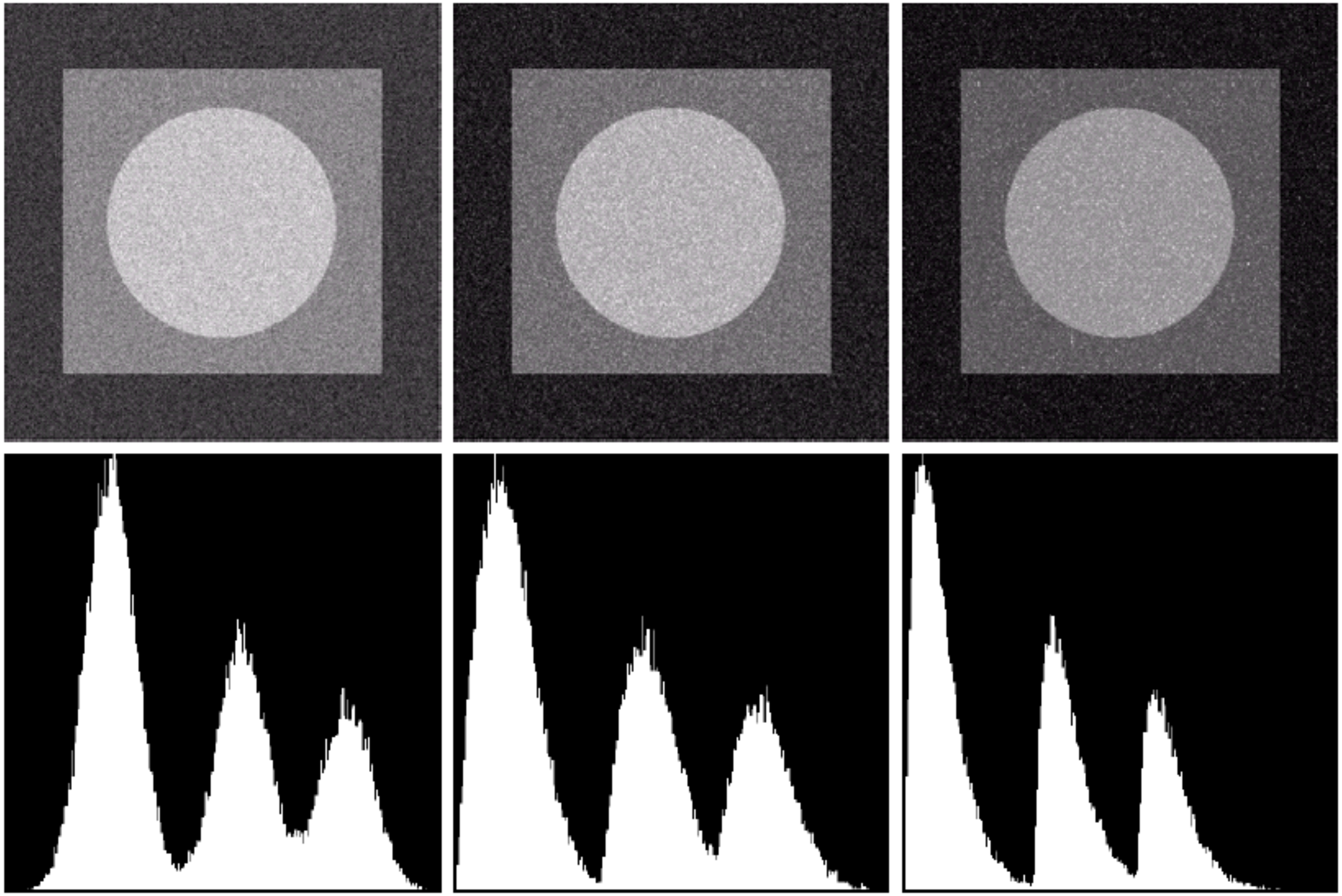


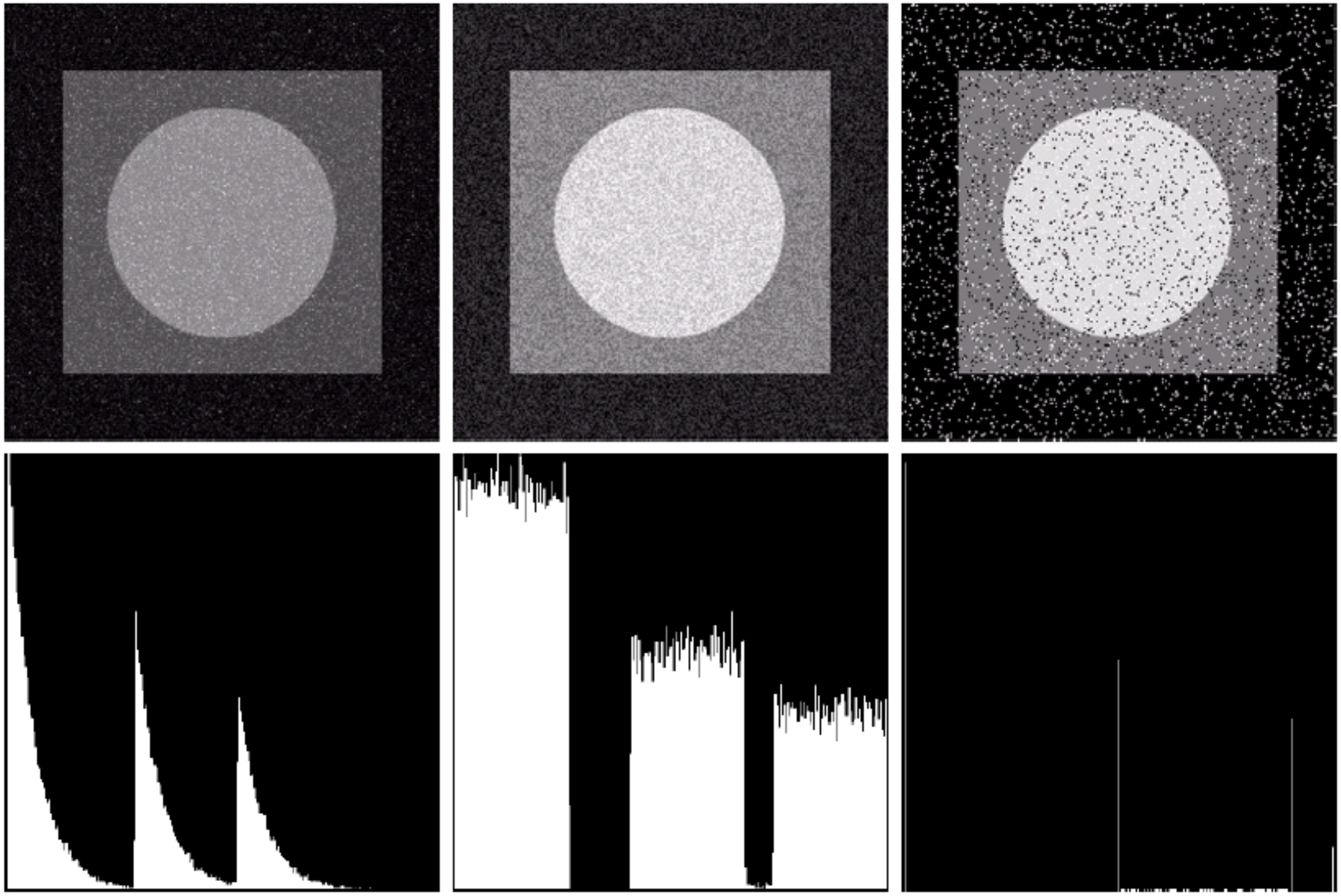
FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

Noise Effect Examples



Gaussian Rayleigh Gamma

Noise Effect Examples



Exponential Uniform Salt & Pepper

Periodic Noise

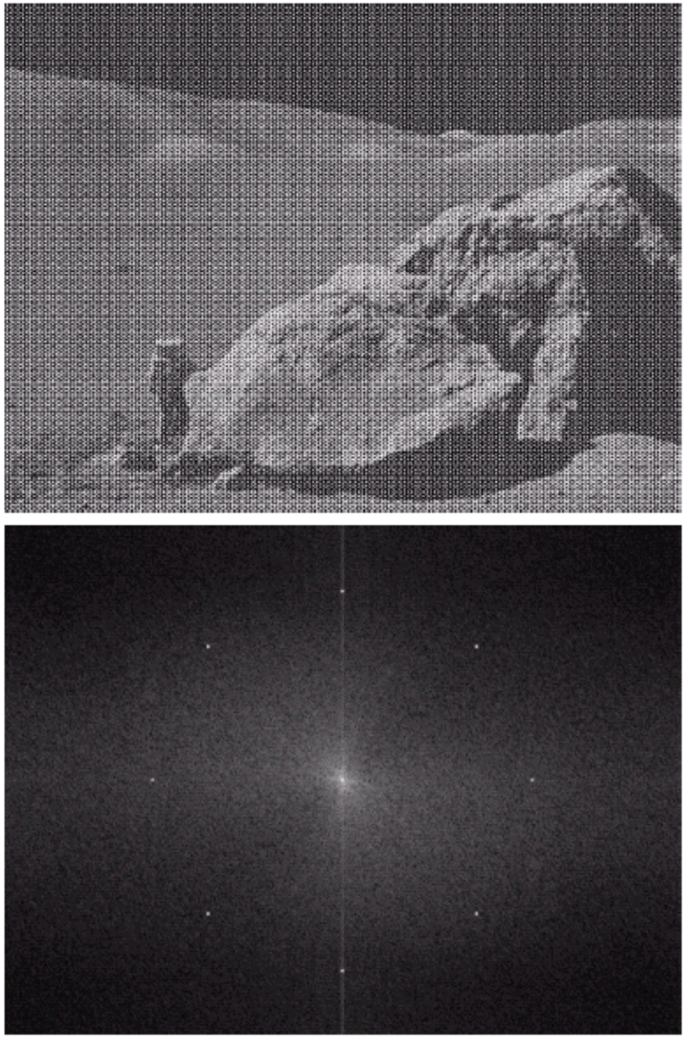
- Typically arises from electrical or electromechanical interference during image acquisition

- Can be effectively reduced by frequency domain filtering

Periodic Noise Example

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.)



the image is severely corrupted by spatial sinusoidal noise of various frequencies

note the pairs of conjugate impulses corresponding to different sine waves

Estimation of Noise Parameters

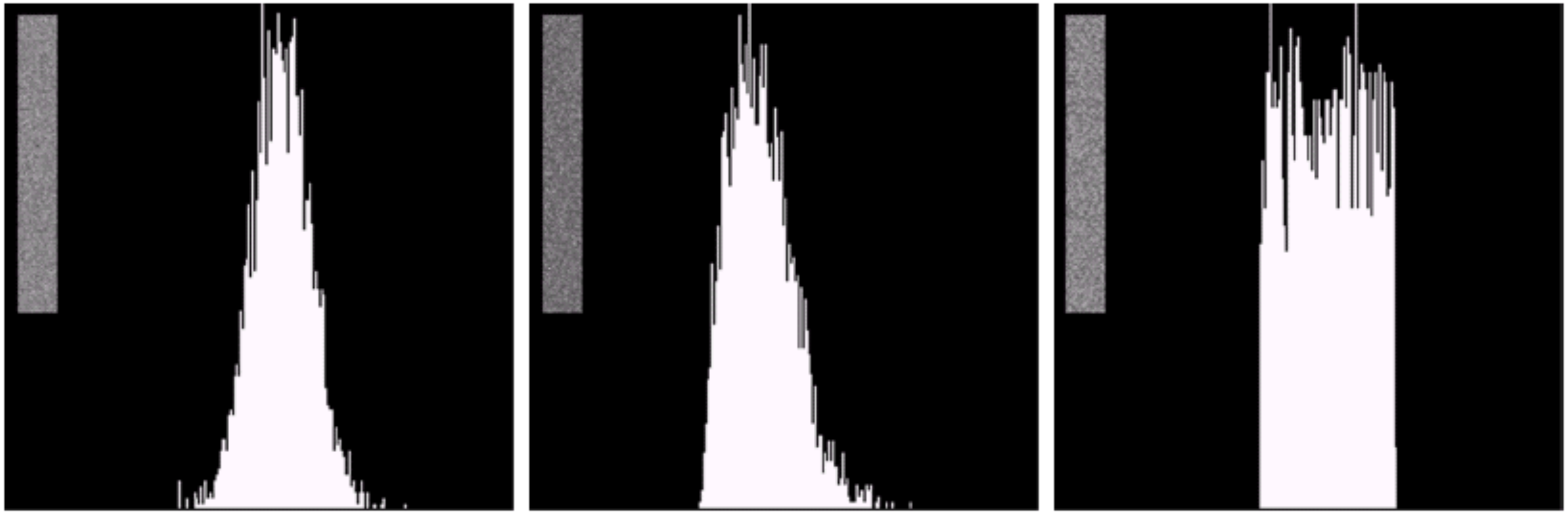
- Periodic noise:
  - inspection in the frequency domain
  - directly from the image (very simple cases)
  - automated analysis (if approx. frequency range known)
- Non-periodic noise:
  - from known sensor specifications
  - capturing “flat” environment images
  - parameter estimation from areas with reasonably constant background intensity, e.g. for the image patch S:

$$\mu = \sum_{z_i \in S} z_i p(z_i) \quad \text{and} \quad \sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

  The PDF parameters can then be estimated via provided formulas.

  - Impulse noise probability can be estimated from a “midgray” area

Example of Noise Parameter Estimation



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.

Restoration in the Presence of Noise Only

- Assume that the only degradation present in the image is noise:

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

- If noise is periodic, it may be possible to directly estimate $N(u,v)$ and subtract it
- Usually spatial filtering is a method of choice

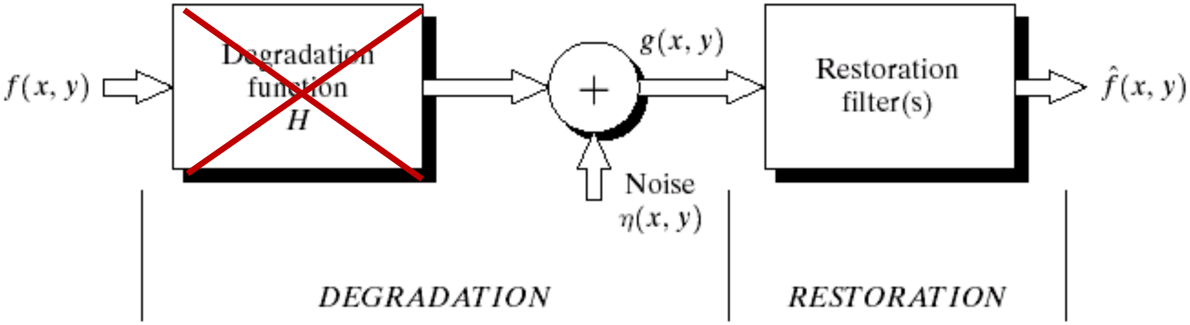


FIGURE 5.1 A model of the image degradation/restoration process.

Mean Filters

1. Arithmetic mean filter

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

- S_{xy} represents the set of coordinates in a rectangular subimage window of size $m \times n$, centered at (x, y)
- Can be implemented with a $m \times n$ linear filter with all coefficients equal to $1/mn$
- Reduces local variations, blurs the image, resulting in noise suppression

Mean Filters

2. Geometric mean filter

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

- Calculates the product of subimage window pixels, raised to the power of $1/mn$

- Achieves comparable smoothing to the arithmetic mean filter, with less details lost

Arithmetic and Geometric Mean Filter Examples

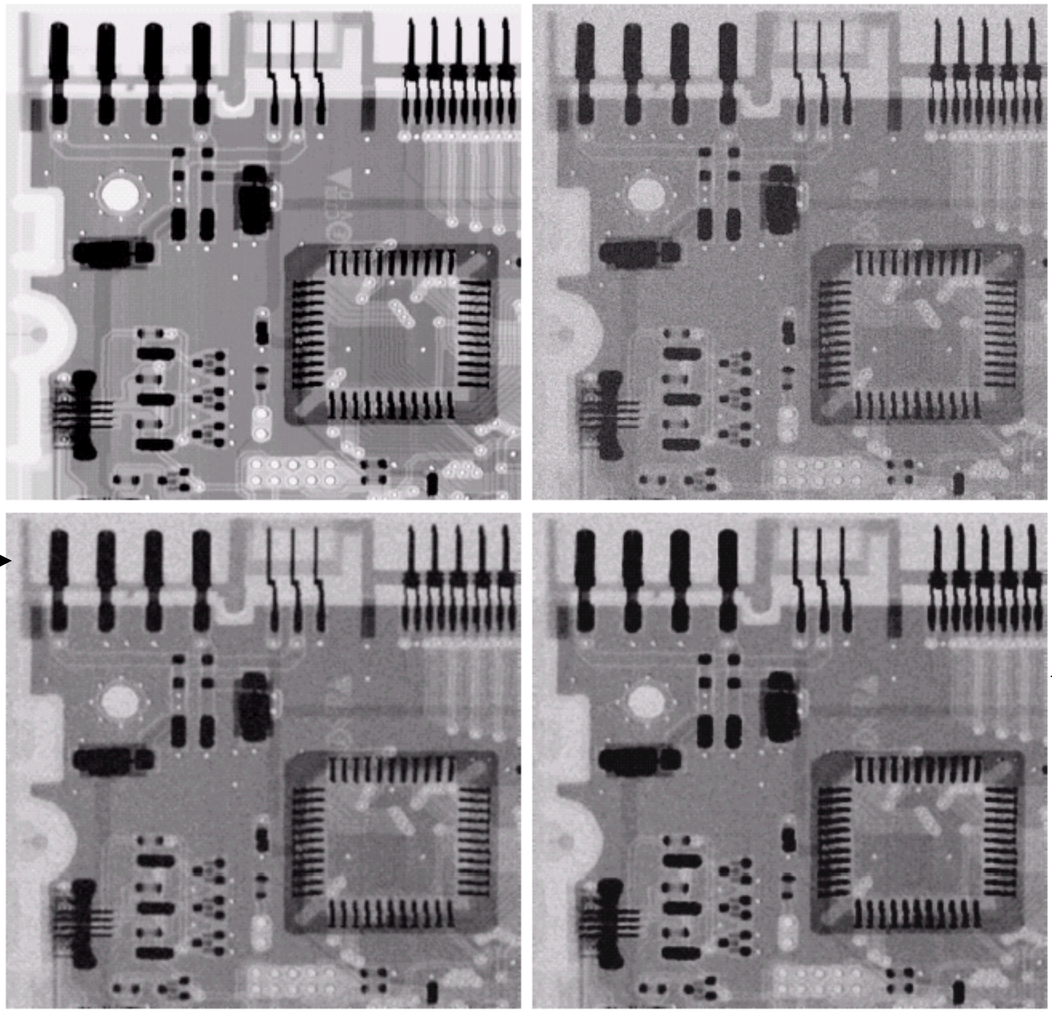


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Arithmetic Mean: noise is reduced at the expense of blurring

Geometric Mean: noise is also removed, but less blurring

Mean Filters

3. Harmonic mean filter

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

- Suitable for e.g. Gaussian noise, salt noise
- Spectacularly fails for pepper noise

Mean Filters

4. Contraharmonic mean filter

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

where Q is the *order* of the filter.

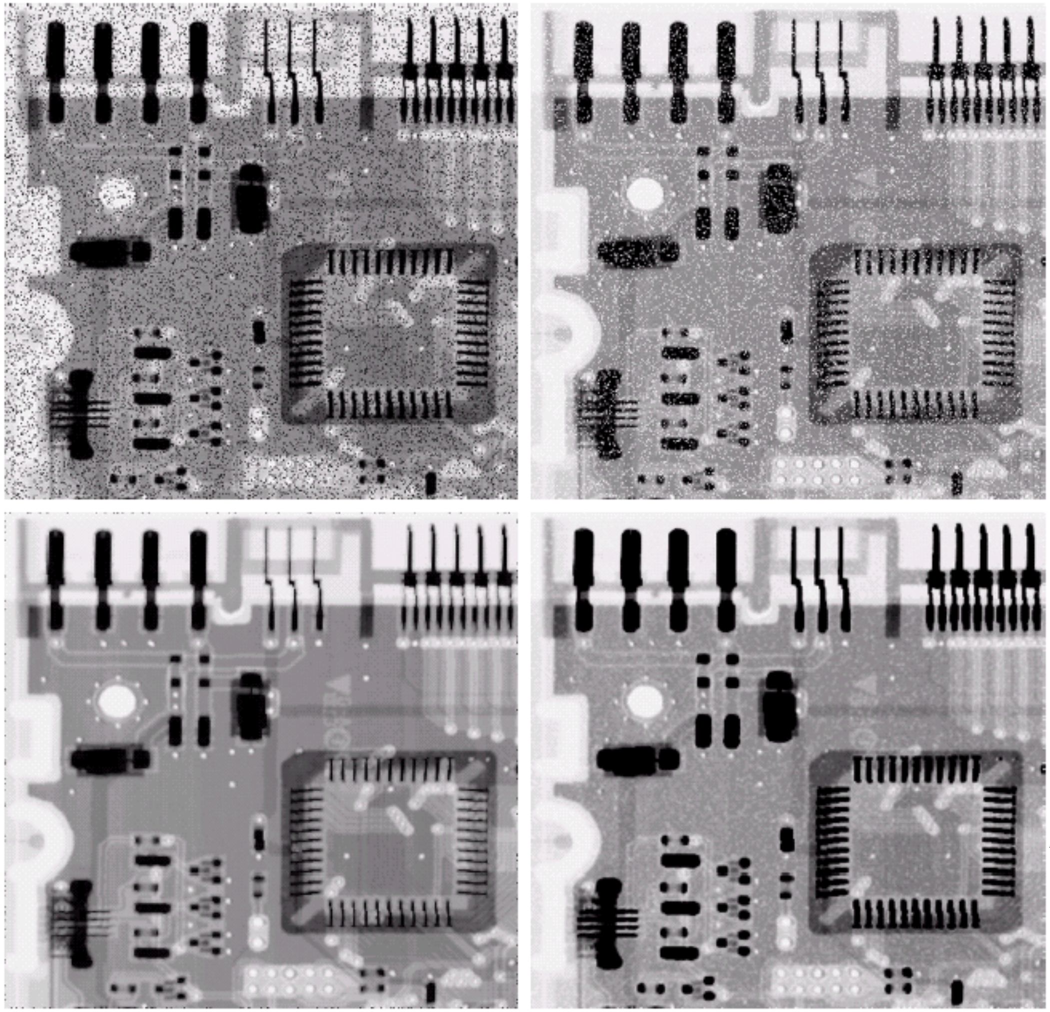
- Works well for salt-and-pepper noise
  - $Q > 0$: eliminates pepper noise
  - $Q < 0$: eliminates salt noise
- Reduces to arithmetic mean filter if $Q = 0$
- Reduces to harmonic mean filter if $Q = -1$

Contraharmonic Filtering Examples

Pepper noisy image Salt noisy image

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 $Q = -1.5$.



Q>0: filter did a good job in cleaning the background at the expense of some blur in dark areas

the opposite is true for Q<0

Contraharmonic Filtering Examples

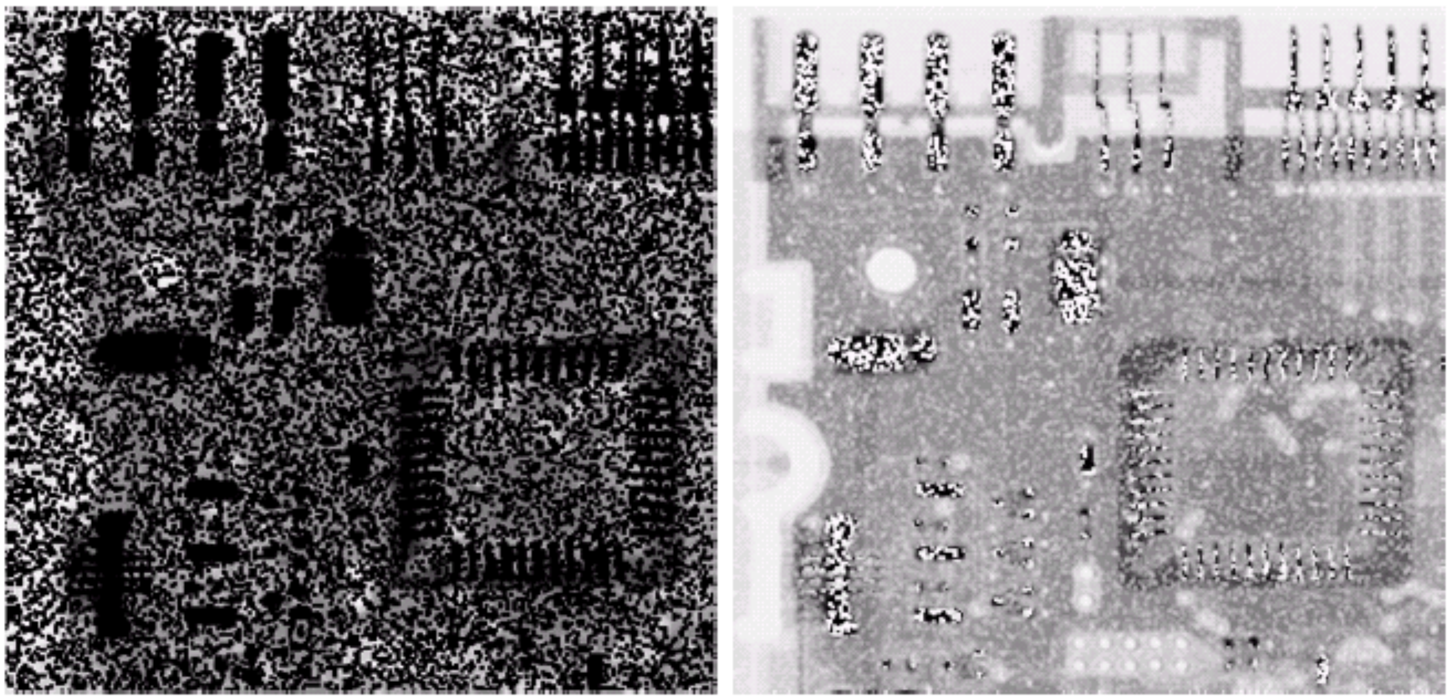


FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

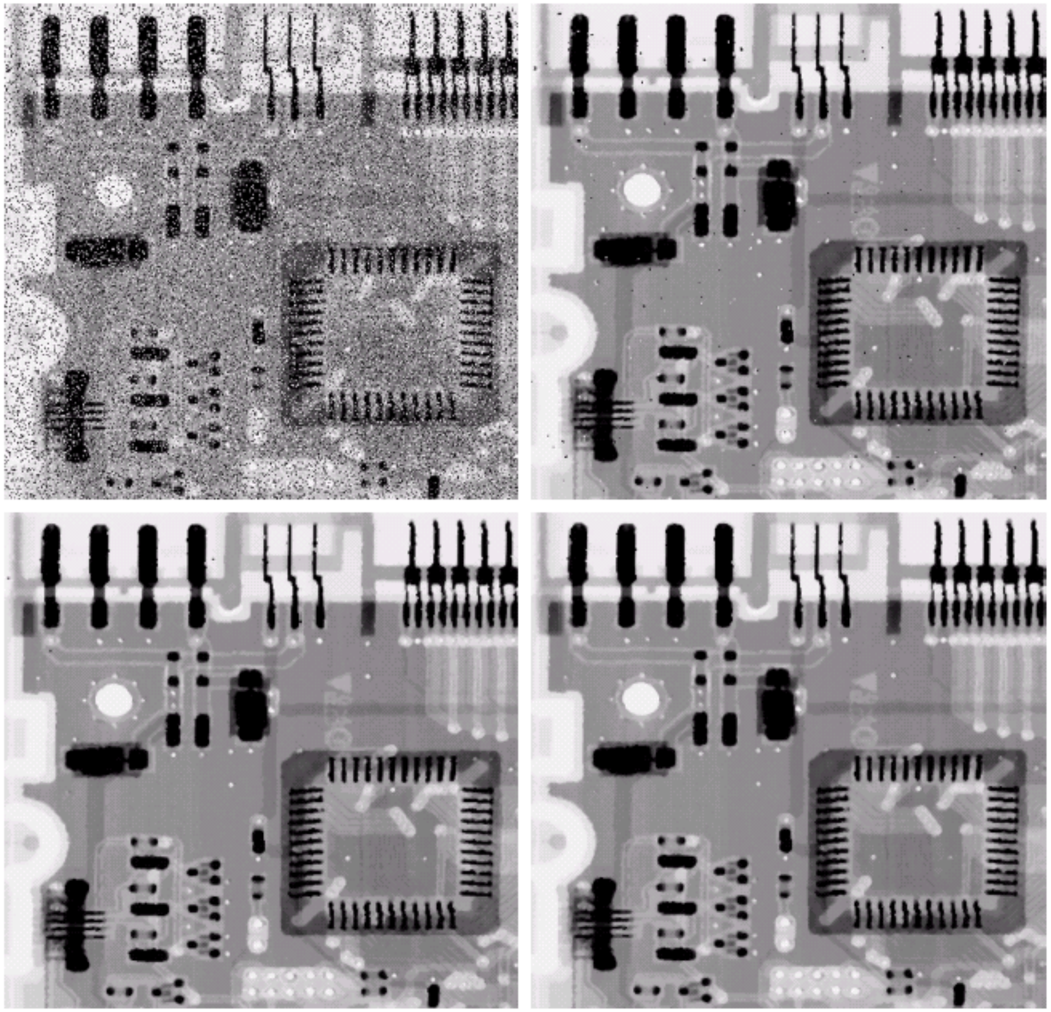
Order-Statistic Filters

1. Median filter

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

- Outputs the median value from the given pixel's neighborhood
- Can efficiently reduce noise (especially impulse noise) with significantly less blurring
- Multiple passes reduce noise further, but introduce extra blurring

Median Filter Example



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.

1^{st} pass result

2^{nd} pass result

3^{rd} pass result

Order-Statistic Filters

2. Max and Min filters

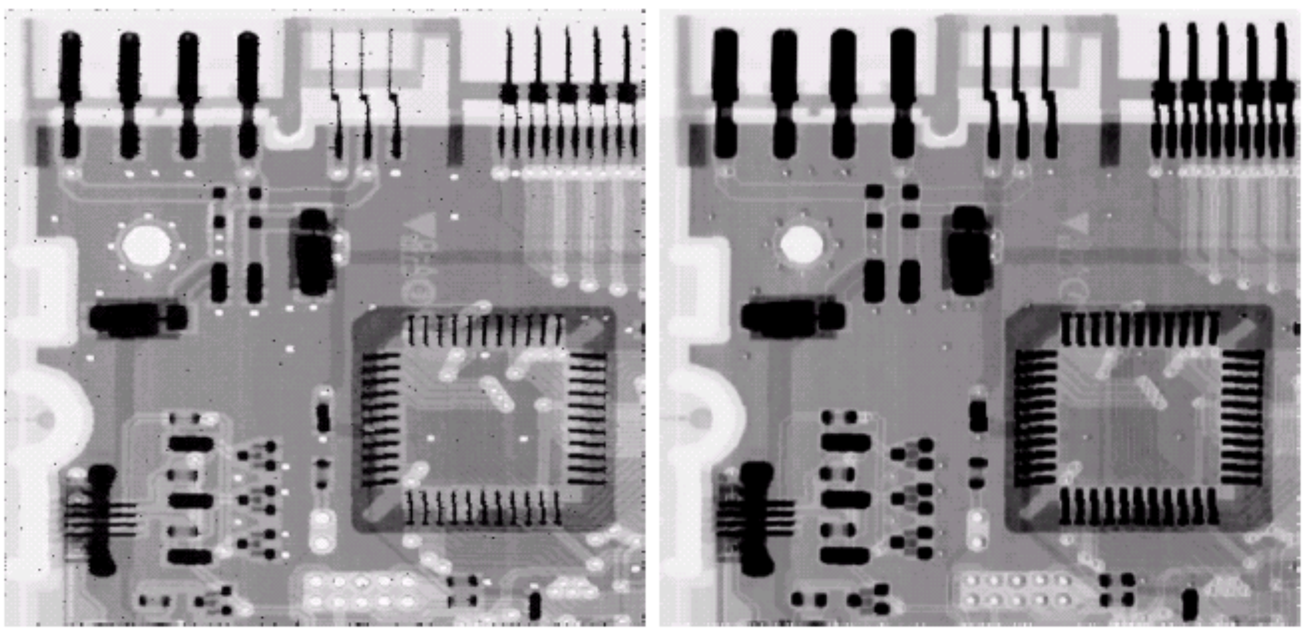
$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\} \quad \text{and} \quad \hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

- Useful for finding the brightest or darkest pixels in the image
- Can suppress pepper and salt noise, respectively

Max and Min Filter Examples

result of filtering the image corrupted by pepper noise with a max filter

result of filtering the image corrupted by salt noise with a min filter



a b

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

Note the thinning of edges of dark objects

Note the increase of the areas of dark objects

Order-Statistic Filters

3. Midpoint filter

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

- Combination of order statistics and averaging
- Works best for randomly distributed noise, such as Gaussian and Uniform noise.

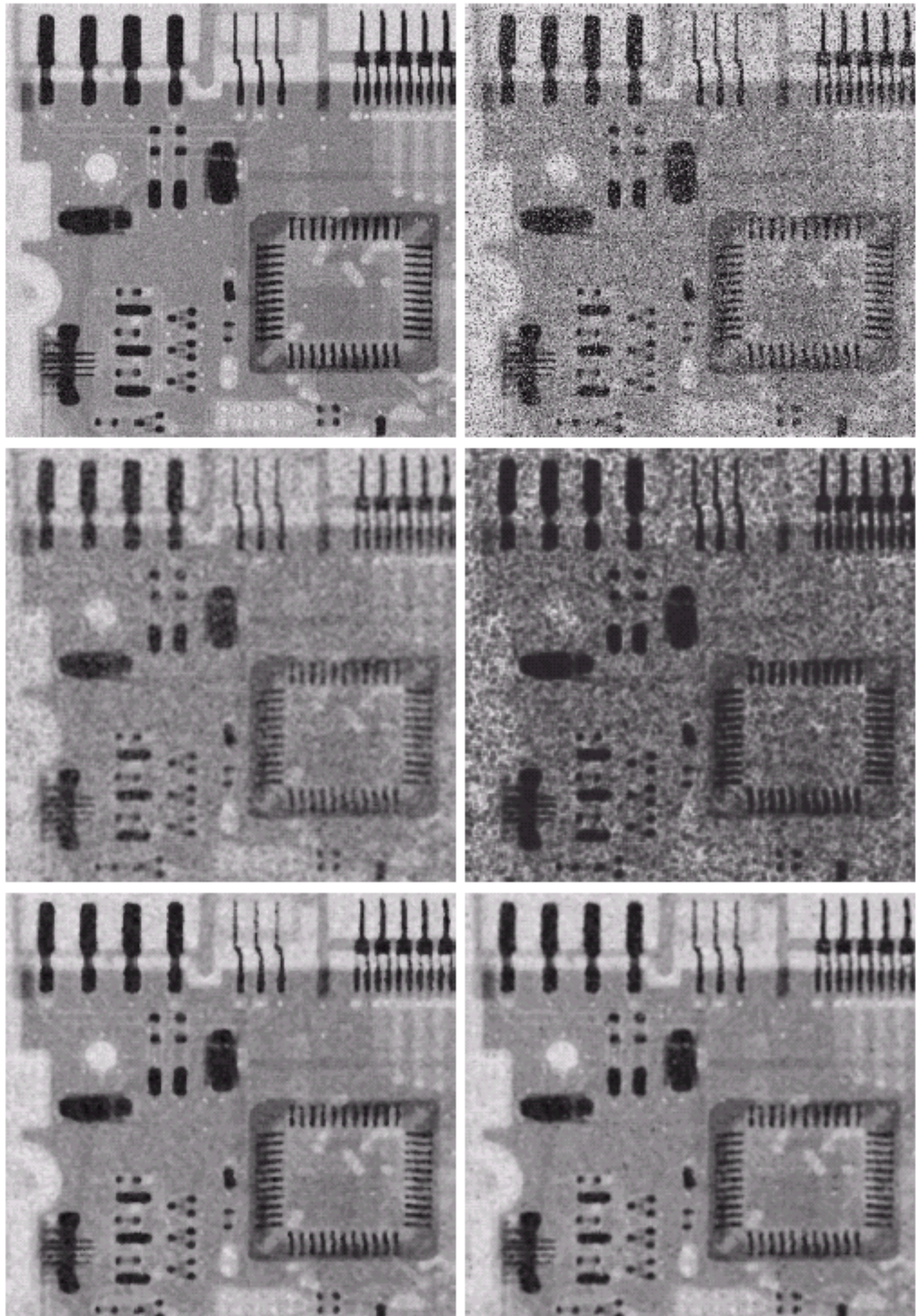
Order-Statistic Filters

4. Alpha-trimmed mean filter

- Remove $d/2$ lowest and $d/2$ highest intensity values from the neighborhood S_{xy} and let $g_r(s, t)$ represent the remaining pixels
- Averaging them yields a following filter:

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

- Useful for combination of noises, such as Gaussian with salt-and-pepper noises



Uniform noise

Additive uniform + salt-and-pepper noise

5x5 arithmetic mean filter

5x5 geometric mean filter

5x5 median filter

5x5 alpha-trimmed mean filter, $d = 5$

Adaptive Filtering

- Better filtering can be achieved, if the filter operates adaptively based on the contents of the window S_{xy}
- We can utilize the following terms:
  - $g(x, y)$ – the noisy image intensity
  - σ_η^2 – the variance of the noise
  - μ_L – the local mean within S_{xy}
  - σ_L^2 – the local variance within S_{xy}
- We want to achieve the following specifications:
  - if σ_η^2 is zero, the filter returns $g(x, y)$ (as there is no noise)
  - if σ_L^2 is large compared to σ_η^2, the window contains strong edges and the filter should closely preserve $g(x, y)$
  - if $\sigma_L^2 = \sigma_\eta^2$, arithmetic mean filter should be used

Adaptive Filtering

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

- This *adaptive filter* fits the given specifications
- We need to know or estimate the global noise variance σ_η^2
- Attention: $\sigma_L^2 \geq \sigma_\eta^2$ is assumed, but can be violated in practice, resulting in negative values!
- Two ways to address the problem:
  - Set the ratio to 1 if $\sigma_\eta^2 > \sigma_L^2$, then the filter becomes nonlinear
  - Allow negative values and rescale the final output, this will result in a loss in the dynamic range.

Adaptive Filtering Example

a b
c d

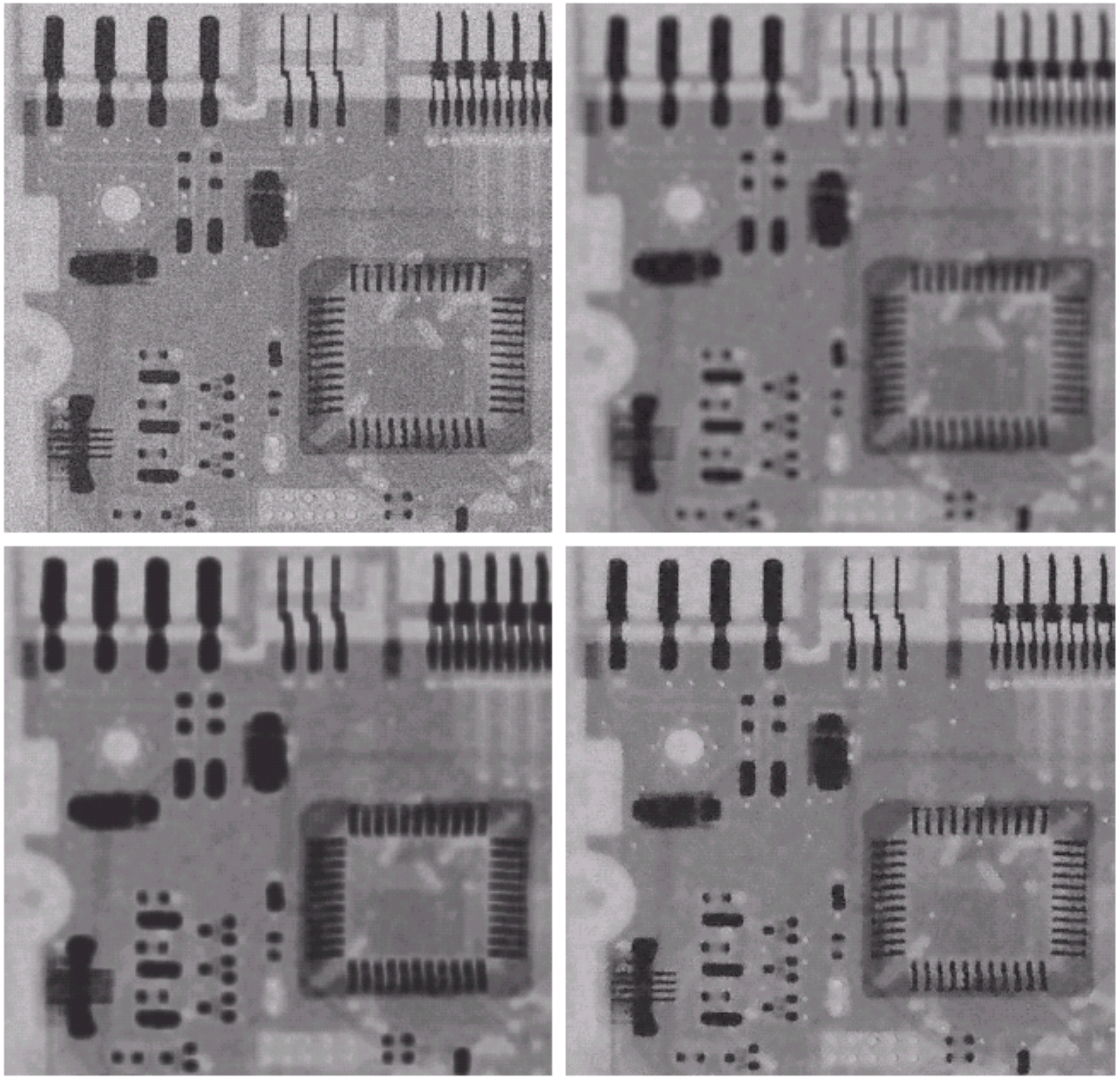


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

Adaptive Median Filtering

Consider the following notation:

$$z_{\min} = \text{minimum gray level value in } S_{xy}$$
$$z_{\max} = \text{maximum gray level value in } S_{xy}$$
$$z_{\text{med}} = \text{median of gray levels in } S_{xy}$$
$$z_{xy} = \text{gray level at coordinates } (x, y)$$
$$S_{\max} = \text{maximum allowed size of } S_{xy}.$$

The adaptive median filtering algorithm works in two levels, denoted level A and level B, as follows:

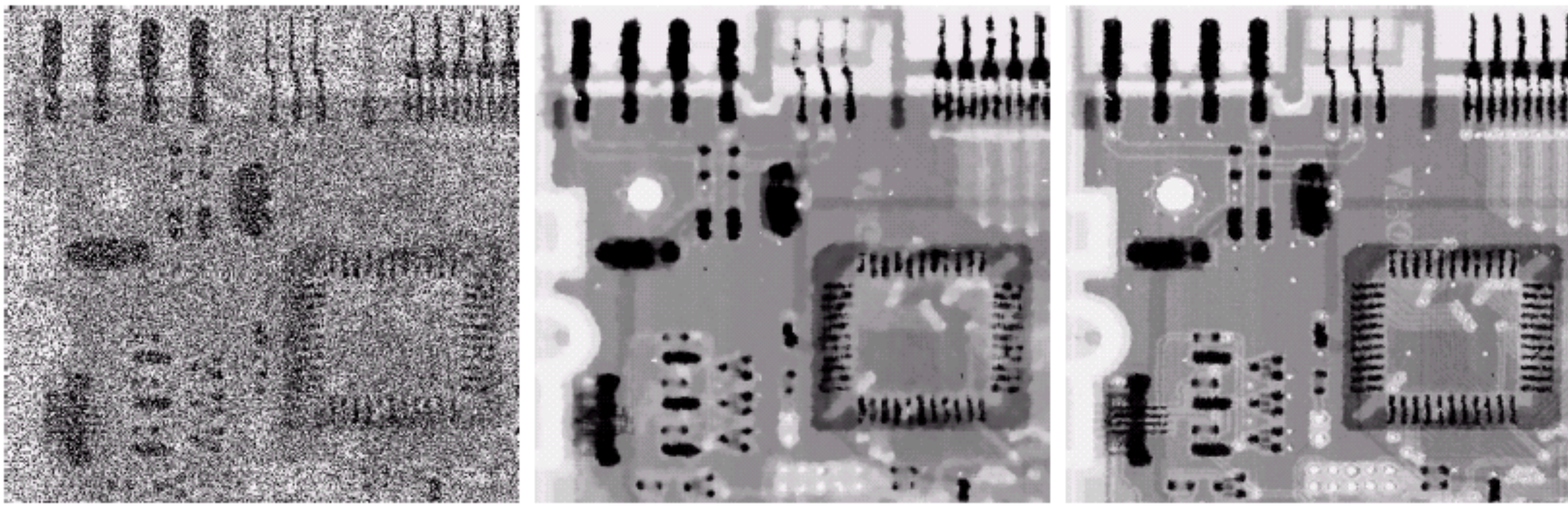
Level A:
$A1 = z_{\text{med}} - z_{\min}$
$A2 = z_{\text{med}} - z_{\max}$
If $A1 > 0$ AND $A2 < 0$, Go to level B
Else increase the window size
If window size $\leq S_{\max}$ repeat level A
Else output z_{xy}.

$z_{\min} < z_{\text{med}} < z_{\max}$

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}.

$z_{\min} < z_{xy} < z_{\max}$

Adaptive Median Filtering



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 $S_{max} = 7$.