上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Institute of Media, Information, and Network

生命科学技术学院
School of Life Sciences and Biotechnology

# 图像恢复与重构


**Hongkai Xiong**
**熊红凯**
**http://min.sjtu.edu.cn**

**Department of Electronic Engineering**
**Shanghai Jiao Tong University**


**14 February. 2020**

# Topical today

- Image Restoration and Reconstruction

# Compare with Image enhancement

- Both of their goal is to improve an image in some predefined sense.
- Although there are areas of overlap, image enhancement is largely a subjective process, while image restoration is for the most part an objective process.

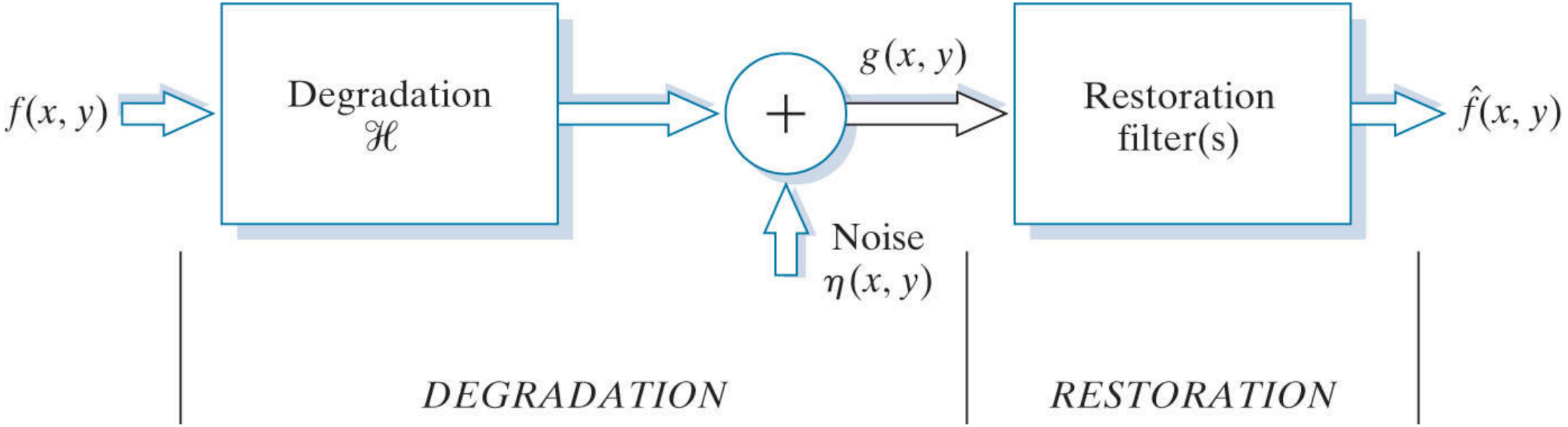
# Why we restoration or reconstruction

- Medical signals easy to be corrupted, e.g. Low-dose CT imaging introduces Poisson noise.

- Restoration attempts to recover an image that has been degraded by using a priori knowledge of the degradation phenomenon. Thus, restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image.

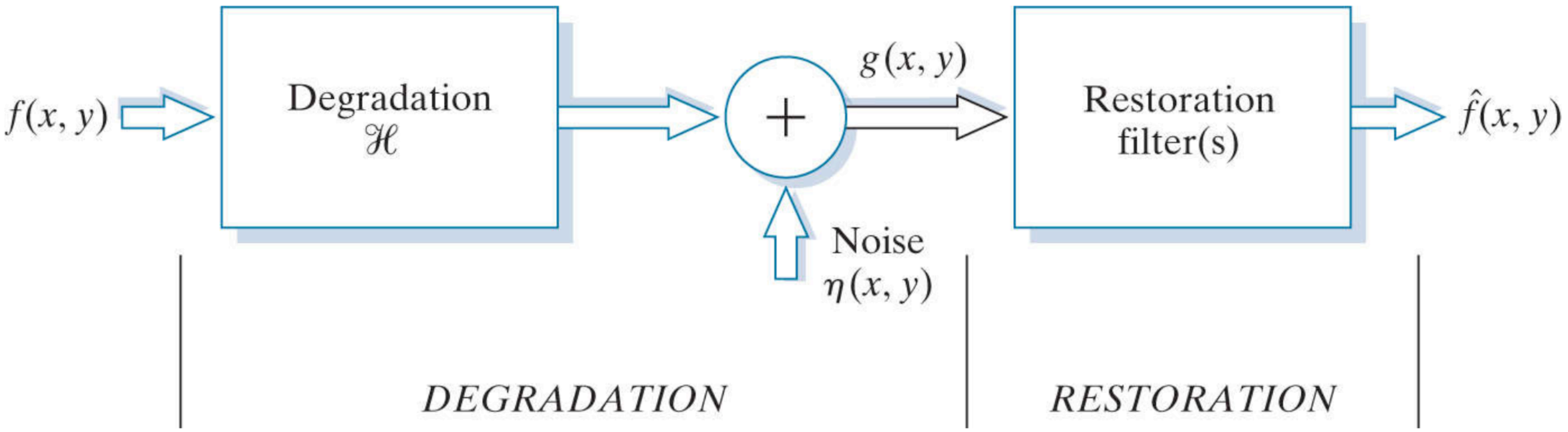
- reconstruction from projections, and their application to computed tomography (CT), one of the most important commercial applications of image processing, especially in health care.

# Model of the Image Degradation/Prestoration Process




We model image degradation as an operator H that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y) (see figure above). Given g(x, y), some knowledge about H, and some knowledge about the additive noise term $\eta(x, y)$, the objective of restoration is to obtain an estimate $\hat{f}(x, y)$ of the original image. We want the estimate to be as close as possible to the original image and, in general, the more we know about H and , the closer $\hat{f}(x, y)$ will be to f(x, y).

# Model of the Image Degradation/Prestoration Process




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

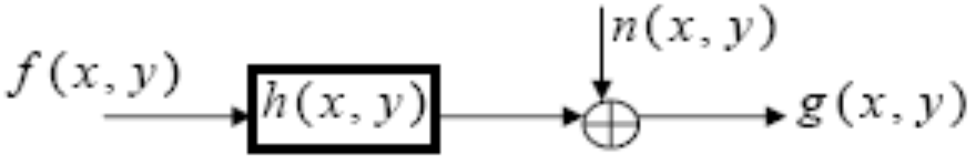
$$g(x,y) = (h \star f)(x,y) + \eta(x,y)$$

- In the frequency domain

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

# Model of the Image Degradation/Prestoration Process

- In the spatial domain




- In the frequency domain


- Our purpose is to recover $f(x, y)$ from the noise image $g(x, y)$, which is almost the same as to remove noise $\eta(x, y)$ from the $g(x, y)$ if we don’t consider the impact of $h(x, y)$.
- To remove noise efficiently, it is better to know the noise model first;
- To build a model for an unknow noise image, it is better to know all the existing and widely noise models

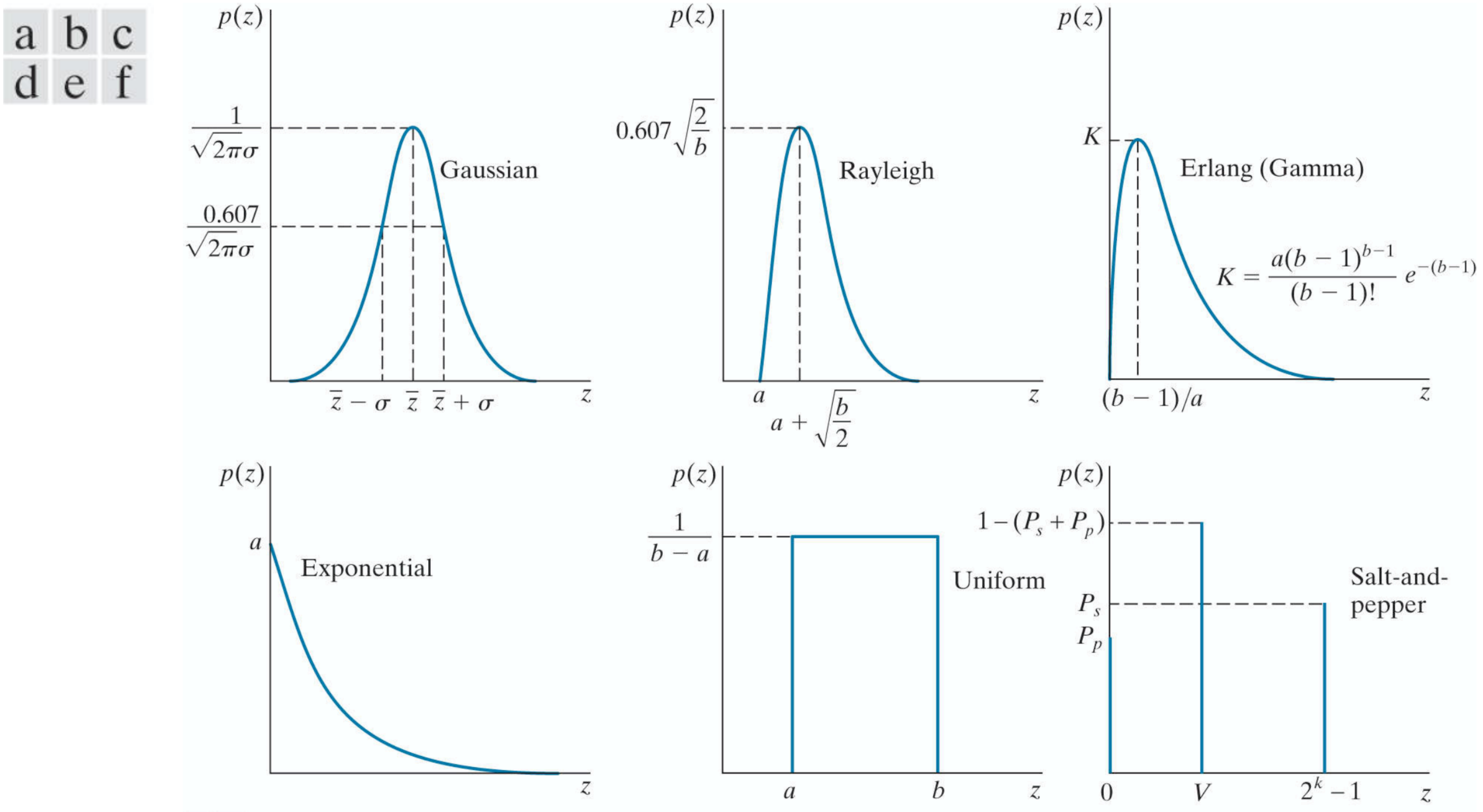
# Noise Models

- The principal sources of noise in digital images arise during image acquisition and/or transmission. The performance of imaging sensors is affected by a variety of environmental factors during image acquisition, and by the quality of the sensing elements themselves

- In the view of removing noise, it is essential to understand the spatial and frequency properties of noise. For example, when the Fourier spectrum of noise is constant, the noise is called white noise.

# Noise Models

- Some Important Noise Probability Density Functions



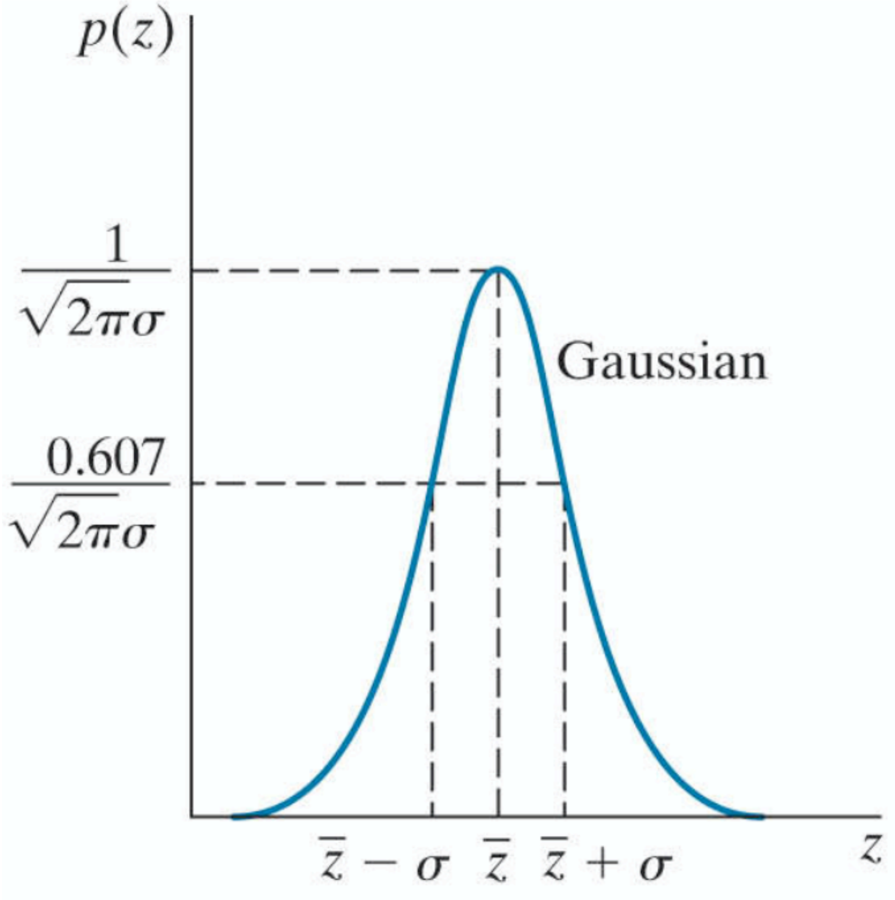
# Noise Models

- Gaussian Noise

The PDF of a Gaussian random variable z:

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

where z represents intensity, $\bar{z}$ is the mean (average) value of z, and $\sigma$ is its standard deviation. Figure right shows a plot of this function. The probability that values of z are in the range $\bar{z} \pm \sigma$ is approximately 0.68; the probability is about 0.95 that the values of z are in the range $\bar{z} \pm 2\sigma$.




The probability density functions of Gaussian noise

# Noise Models

- Rayleigh Noise

The PDF of Rayleigh noise is given by:

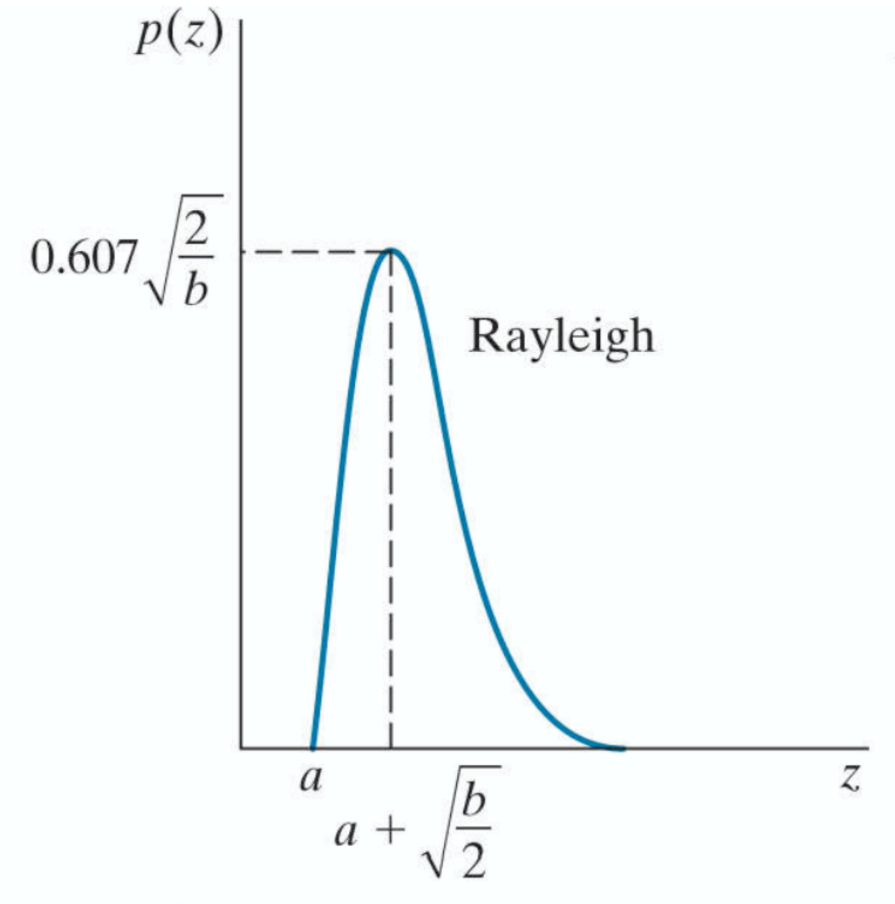
$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$$

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

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

and

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




The probability density functions of Rayleigh noise

# Noise Models

- Erlang(Gamma) Noise

The PDF of Erlang noise is:

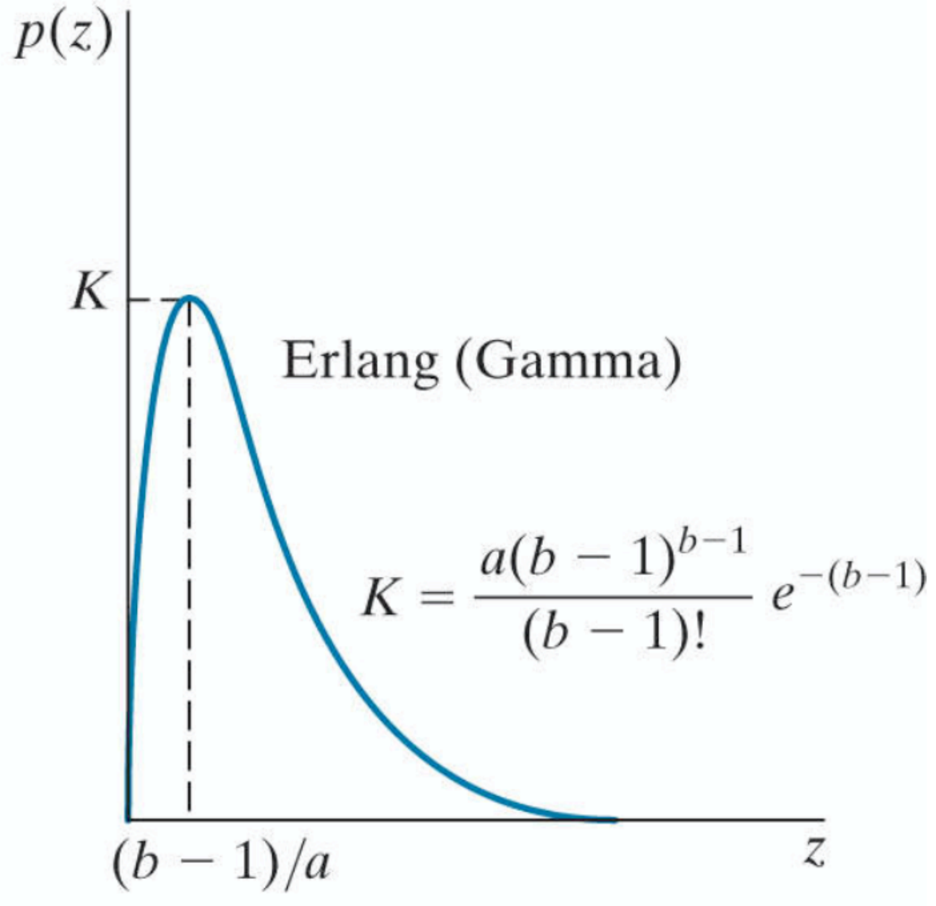
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Where the parameters are such that $a > b$, b is a positive integer, and "!" indicates factorial. The mean and variance of z are :

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

and

$$\sigma^2 = \frac{b}{a^2}$$




The probability density functions of Erlang noise

# Noise Models

- Exponential Noise

The PDF of Erlang noise is:

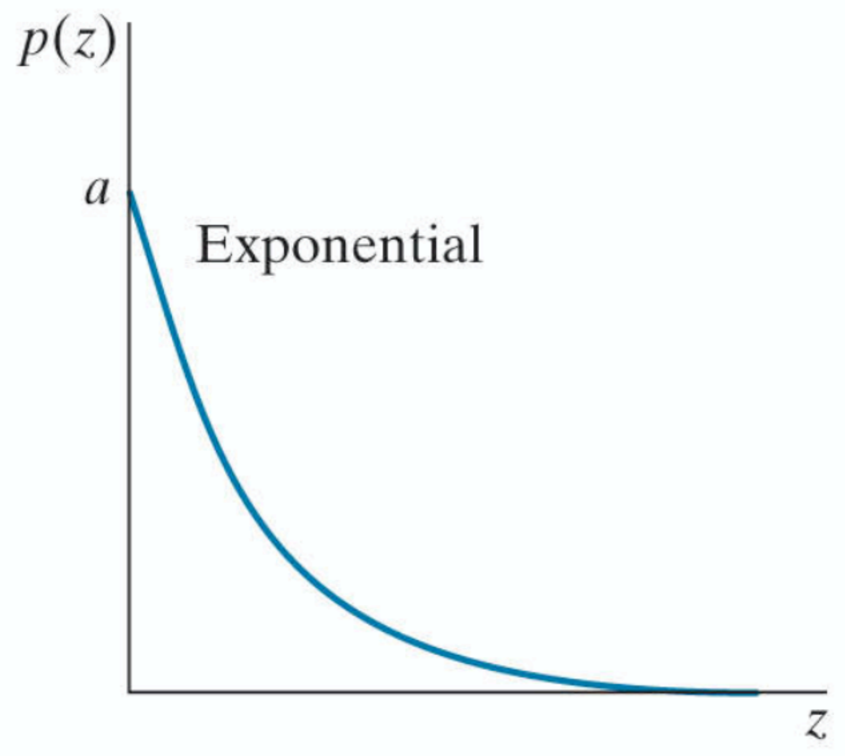
$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

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

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

and

$$\sigma^2 = \frac{1}{a^2}$$




The probability density functions of Exponential noise

# Noise Models

- Uniform Noise

The PDF of uniform noise is:

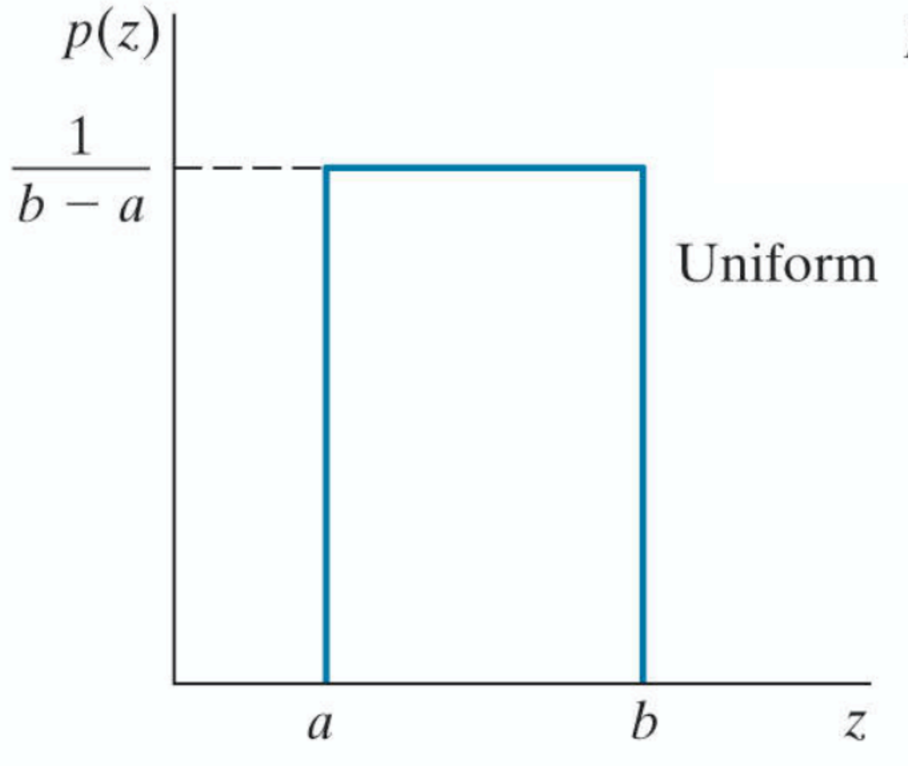
$$p(z) = \begin{cases} \frac{1}{b-a} & a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of z are:

$$\bar{z} = \frac{a+b}{2}$$

and

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




The probability density functions of Uniform noise

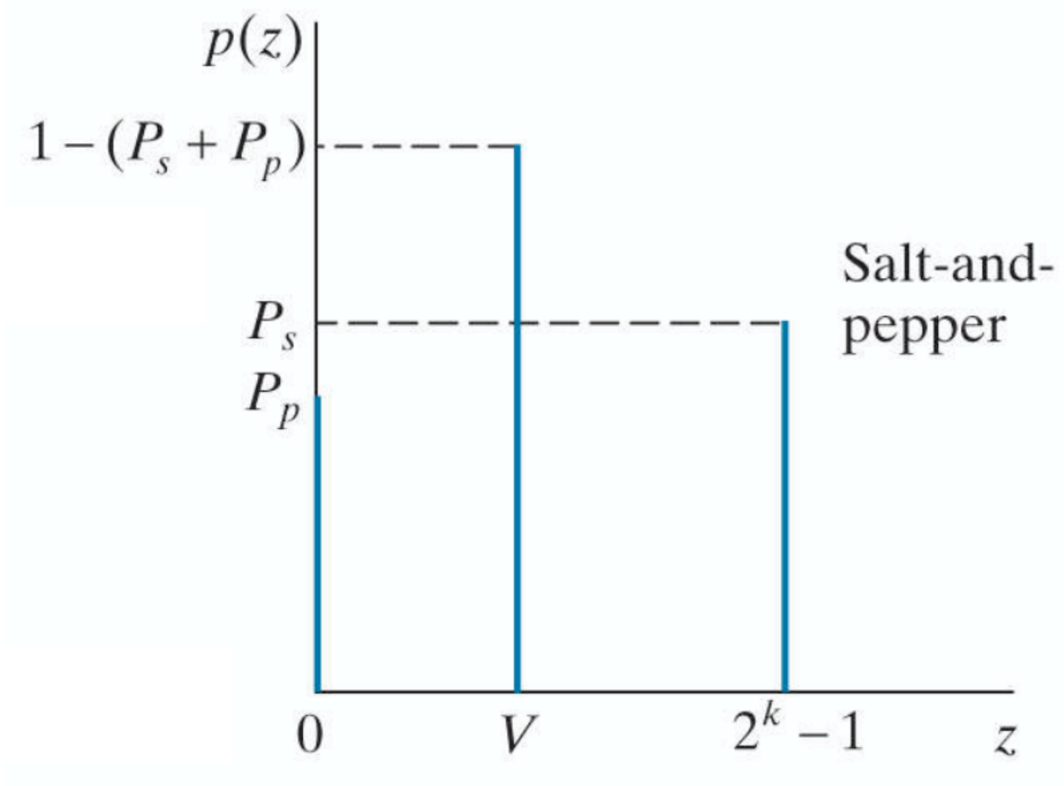
# Noise Models

- Salt-and-Pepper Noise

The PDF of salt-and-pepper noise is:

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

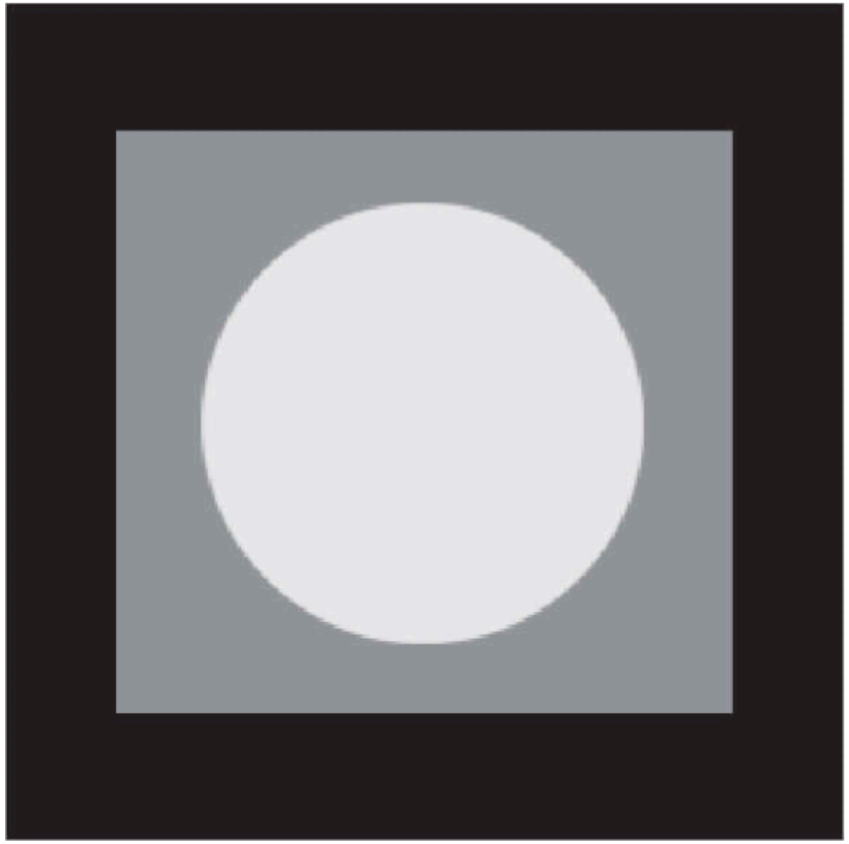
where k represents the number of bits used to represent the intensity values in a digital image. The V is any integer value in the range $0 < V < 2^k - 1$.




The probability density functions of Salt-and-pepper noise

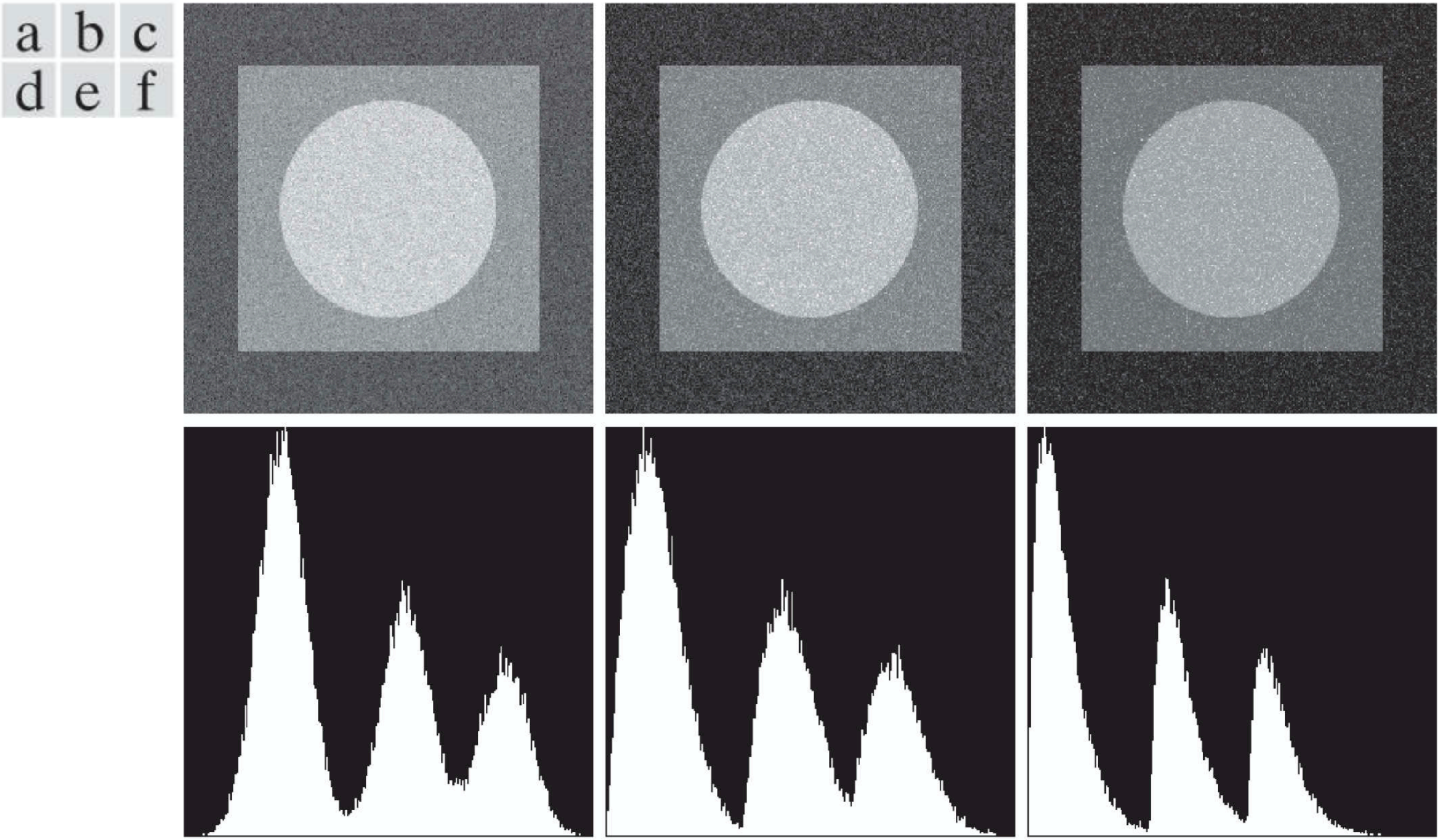
# Noise images and their histograms

- Test pattern used to illustrate the characteristics of the PDFs above



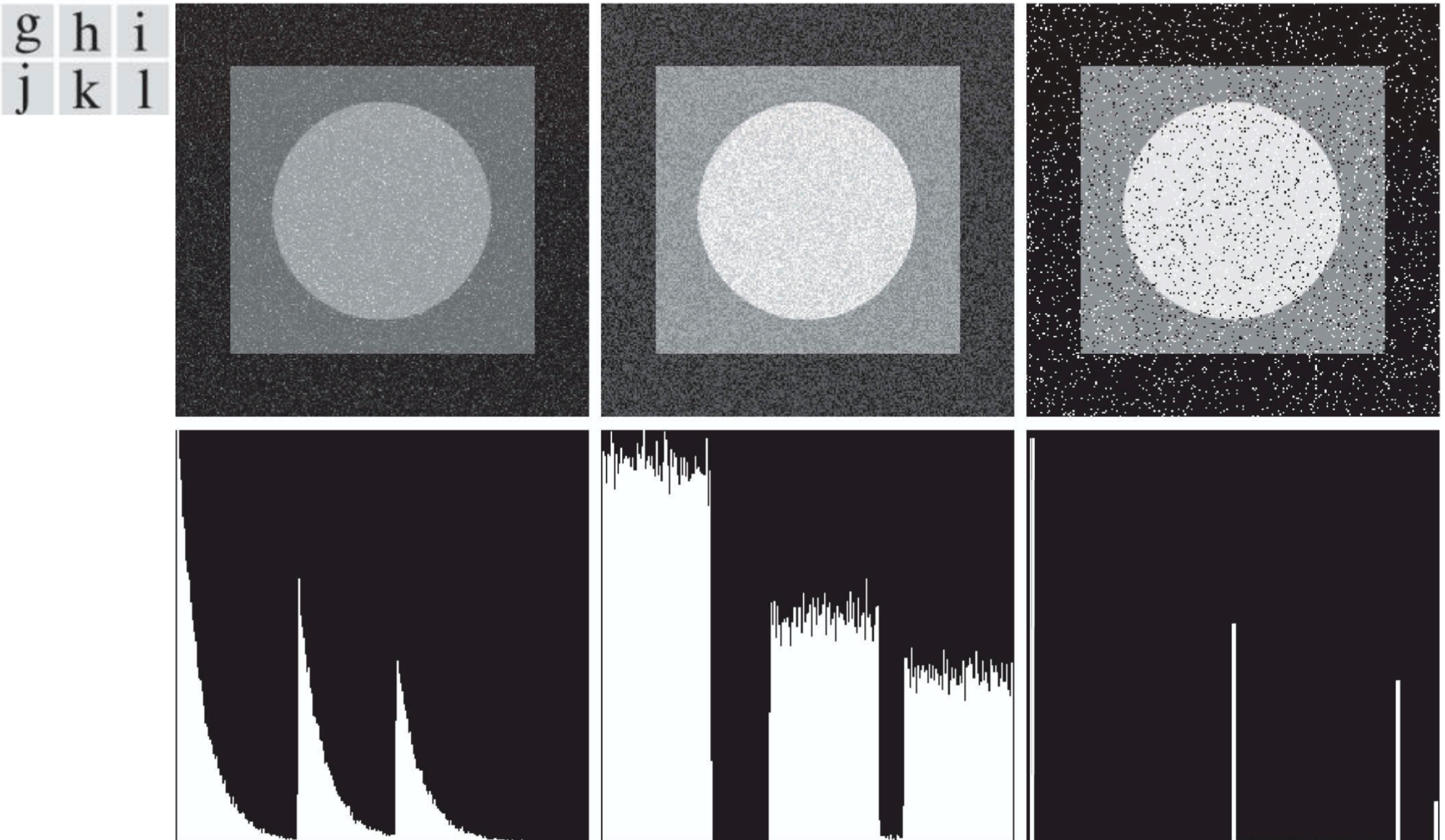
# Noise images and their histograms

- Images and histograms resulting from adding Gaussian, Rayleigh, and the Erlanga noise to the test pattern.



# Noise images and their histograms

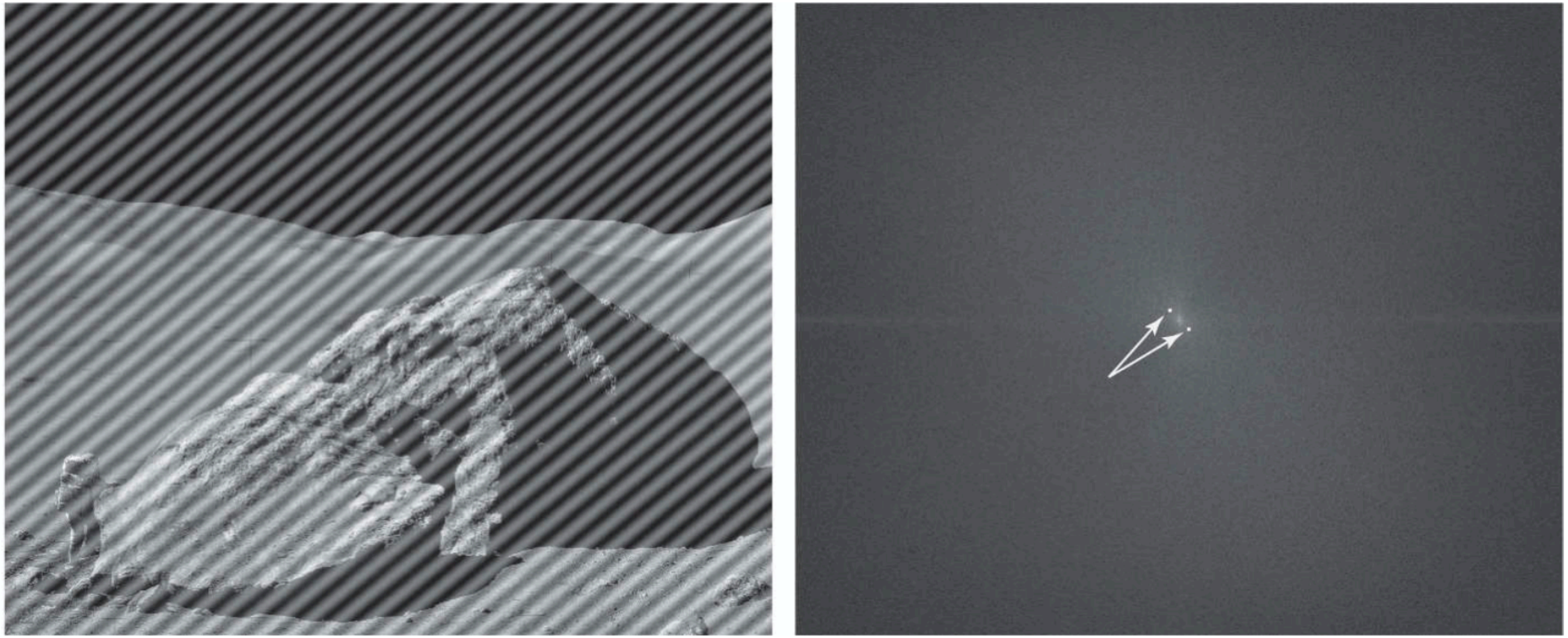
- Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the test pattern.



# Periodic Noise

- Periodic noise in images typically arises from electrical or electromechanical interference during image acquisition. This is the only type of spatially dependent noise we will consider in this section. (a) Image corrupted by additive sinusoidal noise. (b) Spectrum showing two conjugate impulses caused by the sine wave.

a b



# Estimating Noise Parameters

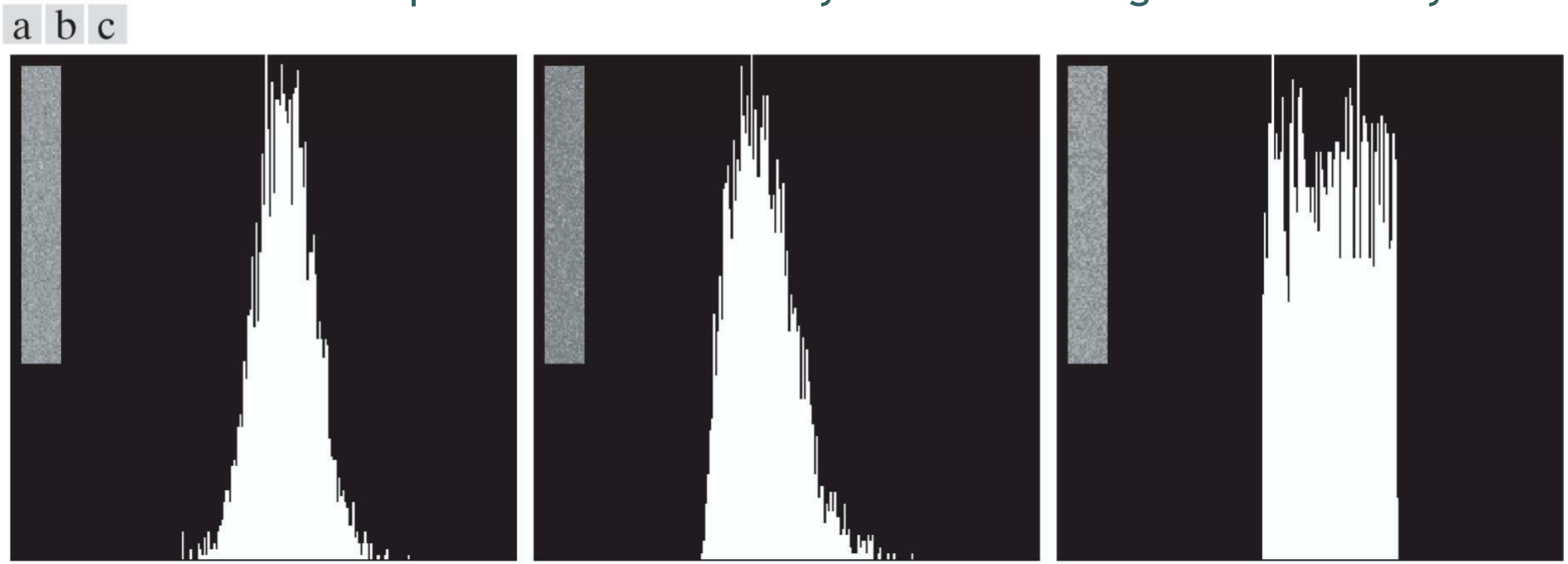
- The parameters of periodic noise typically are estimated by inspection of the Fourier spectrum. Periodic noise tends to produce frequency spikes that often can be detected even by visual analysis. Another approach is to attempt to infer the periodicity of noise components directly from the image, but this is possible only in simplistic cases. Automated analysis is possible in situations in which the noise spikes are either exceptionally pronounced, or when knowledge is available about the general location of the frequency components of the interference.

- The parameters of noise PDFs may be known partially from sensor specifications, but it is often necessary to estimate them for a particular imaging arrangement. If the imaging system is available, one simple way to study the characteristics of system noise is to capture a set of “flat” images. For example, in the case of an optical sensor, this is as simple as imaging a solid gray board that is illuminated uniformly. The resulting images typically are good indicators of system noise.

# Estimating Noise Parameters

- When only images already generated by a sensor are available, it is often possible to estimate the parameters of the PDF from small patches of reasonably constant background intensity.

The histograms shown were calculated using image data from these small strips.




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

# Restoration in the presence of Noise Only Spatial Filtering

- When an image is degraded only by additive noise, we have:

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

and

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

The noise terms generally are unknown, so subtracting them from g(x, y) [G(u, v)] to obtain f(x, y) [F(u, v)] typically is not an option. In the case of periodic noise, sometimes it is possible to estimate N(u, v) from the spectrum of G(u, v), as noted in Section 5.2 . In this case N(u, v) can be subtracted from G(u, v) to obtain an estimate of the original image, but this type of knowledge is the exception, rather than the rule.

Spatial filtering is the method of choice for estimating f(x, y) [i.e., denoising image g(x, y)] in situations when only additive random noise is present. Spatial filtering was discussed in detail before.

# Mean Filters

- Arithmetic Mean Filter

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

- Geometric Mean Filter

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

- Harmonic Mean Filter

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

Let $S_{xy}$ represent the set of coordinates in a rectangular subimage window (neighborhood) of size $m \times n$ , centered on point (x, y). $\hat{f}$ is the restored image. The r and c are the row and column coordinates of the pixels contained in the neighborhood $S_{xy}$.

# Mean Filters

- Contraharmonic Mean Filter

The contraharmonic mean filter yields a restored image based on the expression:

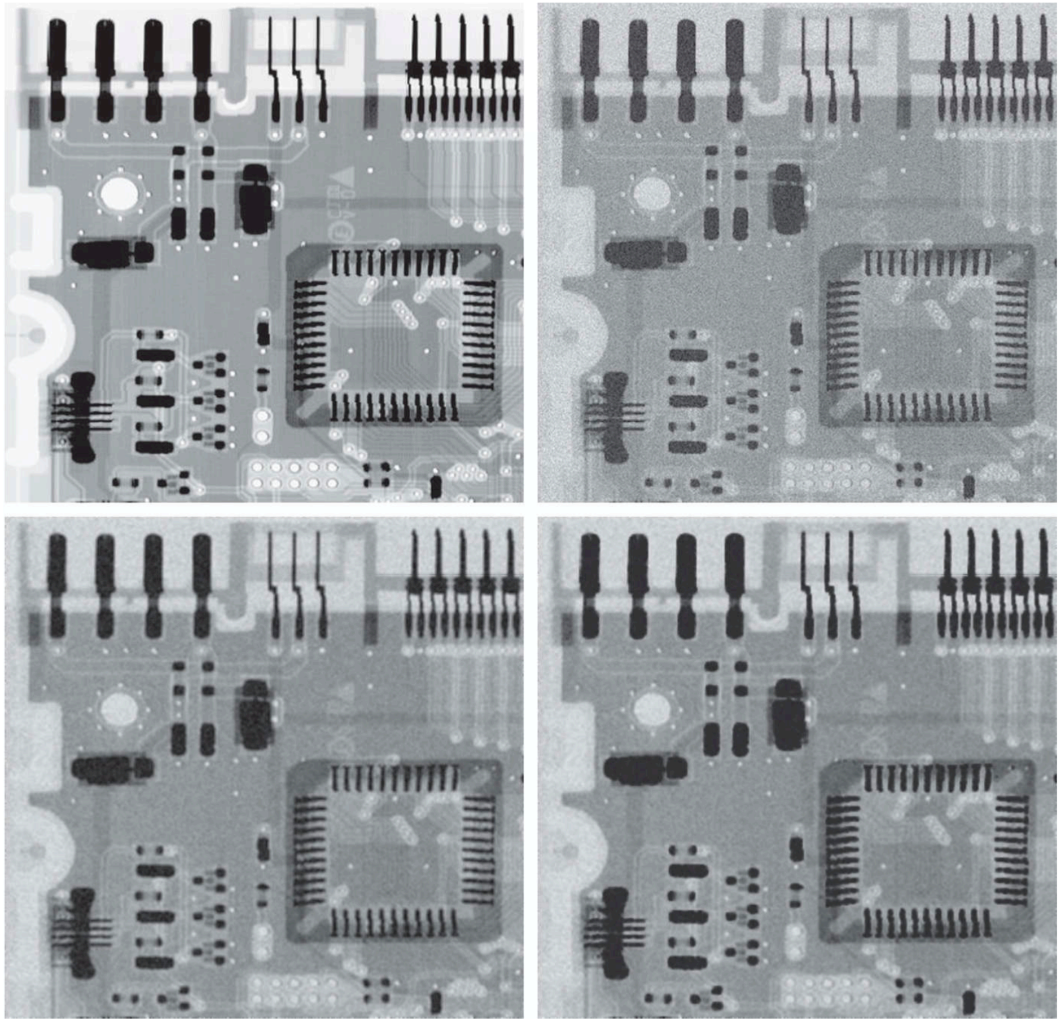
$$\hat{f}(x,y) = \frac{\sum_{(r,c) \in S_{xy}} g(r,c)^{Q+1}}{\sum_{(r,c) \in S_{xy}} g(r,c)^{Q}}$$

where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q, the filter eliminates pepper noise. For negative values of Q, it eliminates salt noise. It cannot do both simultaneously. Note that the contraharmonic filter reduces to the arithmetic mean filter if Q = 0, and to the harmonic mean filter if Q = -1.

# Example:Image denoising using spatial mean filters

(a) X-ray image of circuit board. (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.

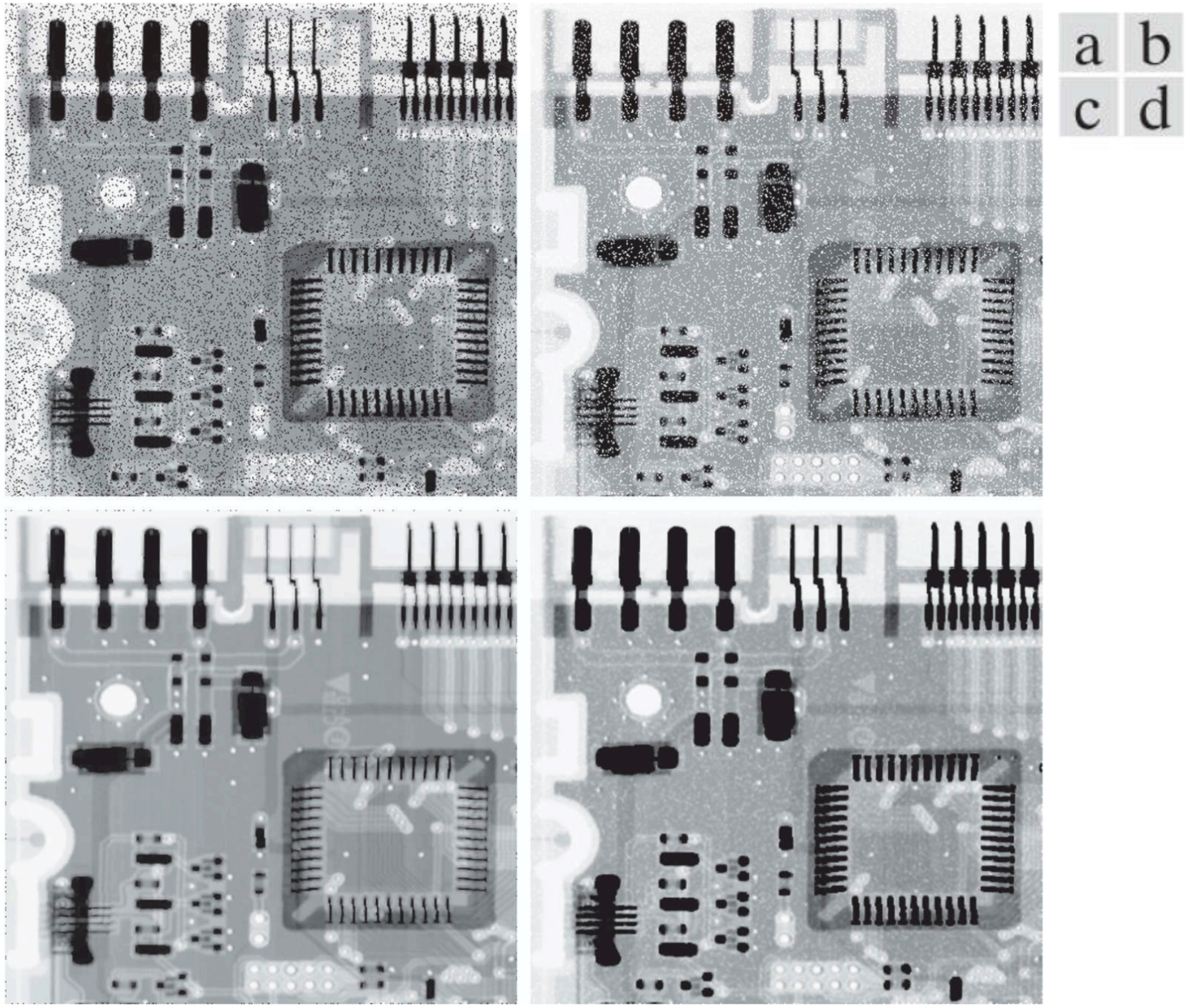
(a) shows an 8-bit X-ray image of a circuit board, and (b) shows the same image, but corrupted with additive Gaussian noise of zero mean and variance of 400. For this type of image, this is a significant level of noise. (c) and (d) show, respectively, the result of filtering the noisy image with an arithmetic mean filter of size 3 × 3 and a geometric mean filter of the same size.



a b
c d

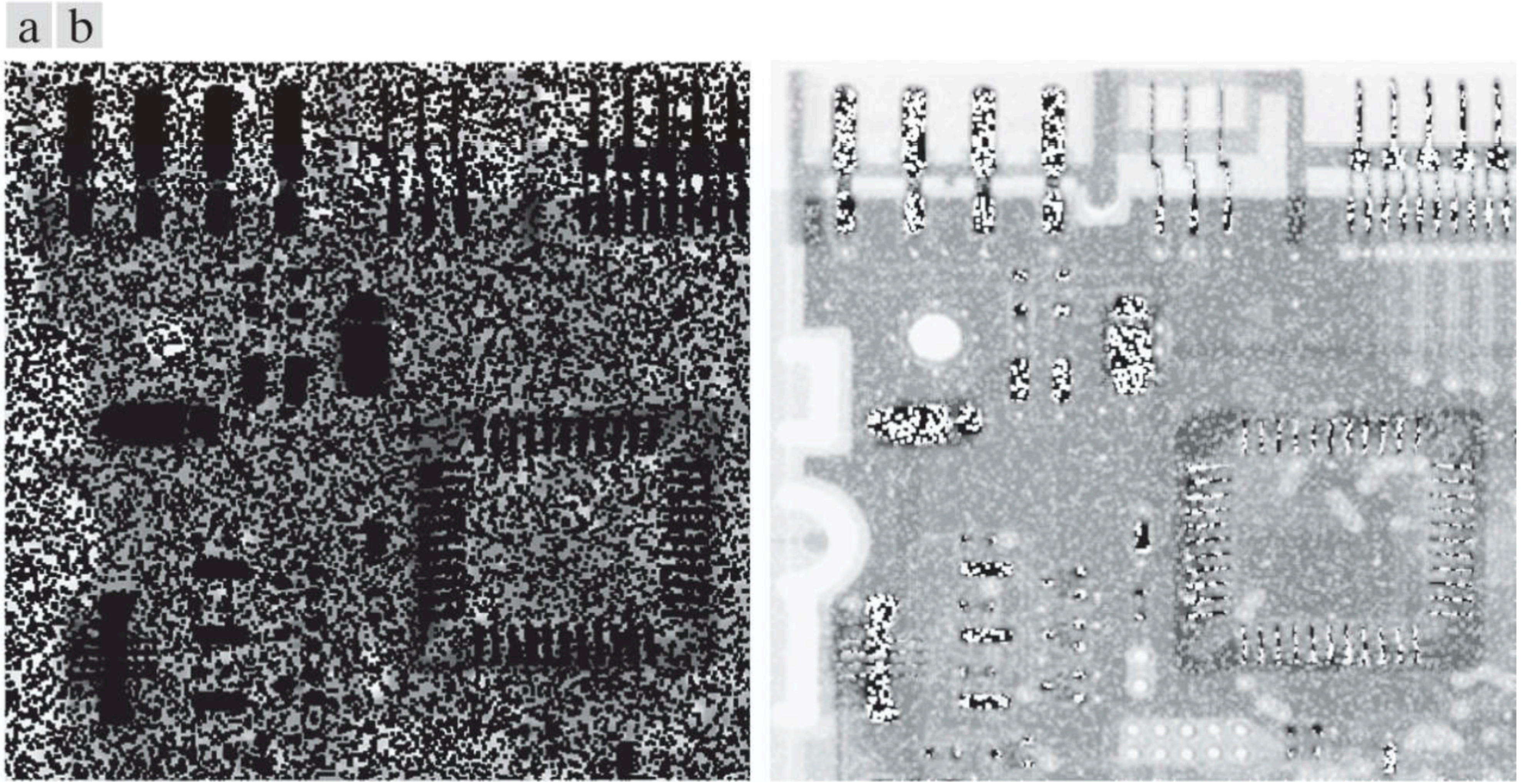
# Example:Image denoising using spatial mean filters

(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 = 1.5. (d) Result of filtering (b) with = − 1.5.



# Example:Image denoising using spatial mean filters

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 = − 1.5. (b) Result of filtering Fig. 5.8(b) using = 1.5.



# 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

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

# Order-Statistic Filters

- Alpha-Trimmed Mean Filter

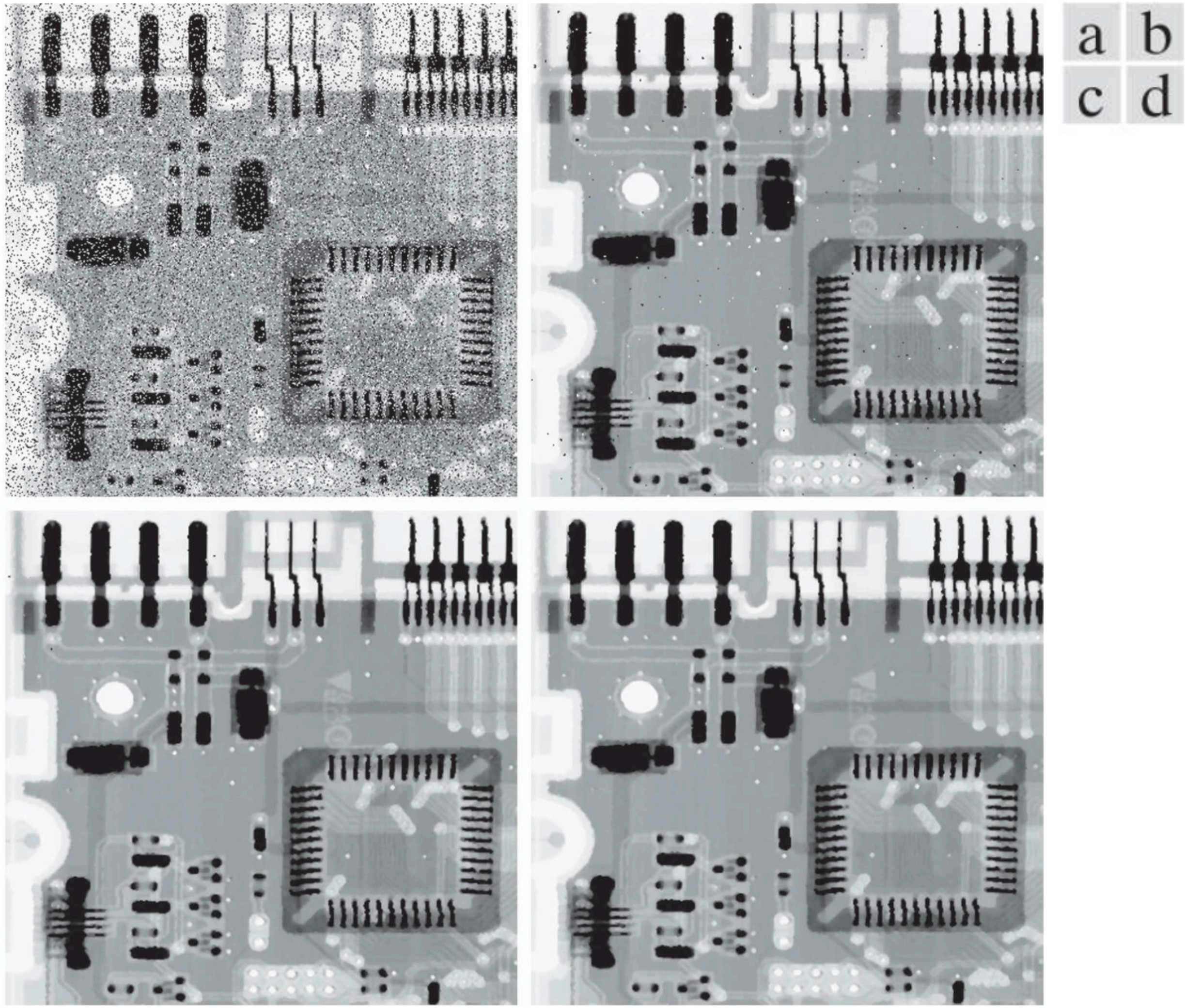
Suppose that we delete the d/2 lowest and the d/2 highest intensity values of g(r, c) in the neighborhood . Let $g_R(r,c)$ represent the remaining mn − d pixels in $S_{xy}$. A filter formed by averaging these remaining pixels is called an alpha-trimmed mean filter. The form of this filter is

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

where the value of d can range from 0 to mn-1. When d = 0 the alpha-trimmed filter reduces to the arithmetic mean filter discussed earlier. If we choose d = mn − 1, the filter becomes a median filter. For other values of d, the alpha-trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

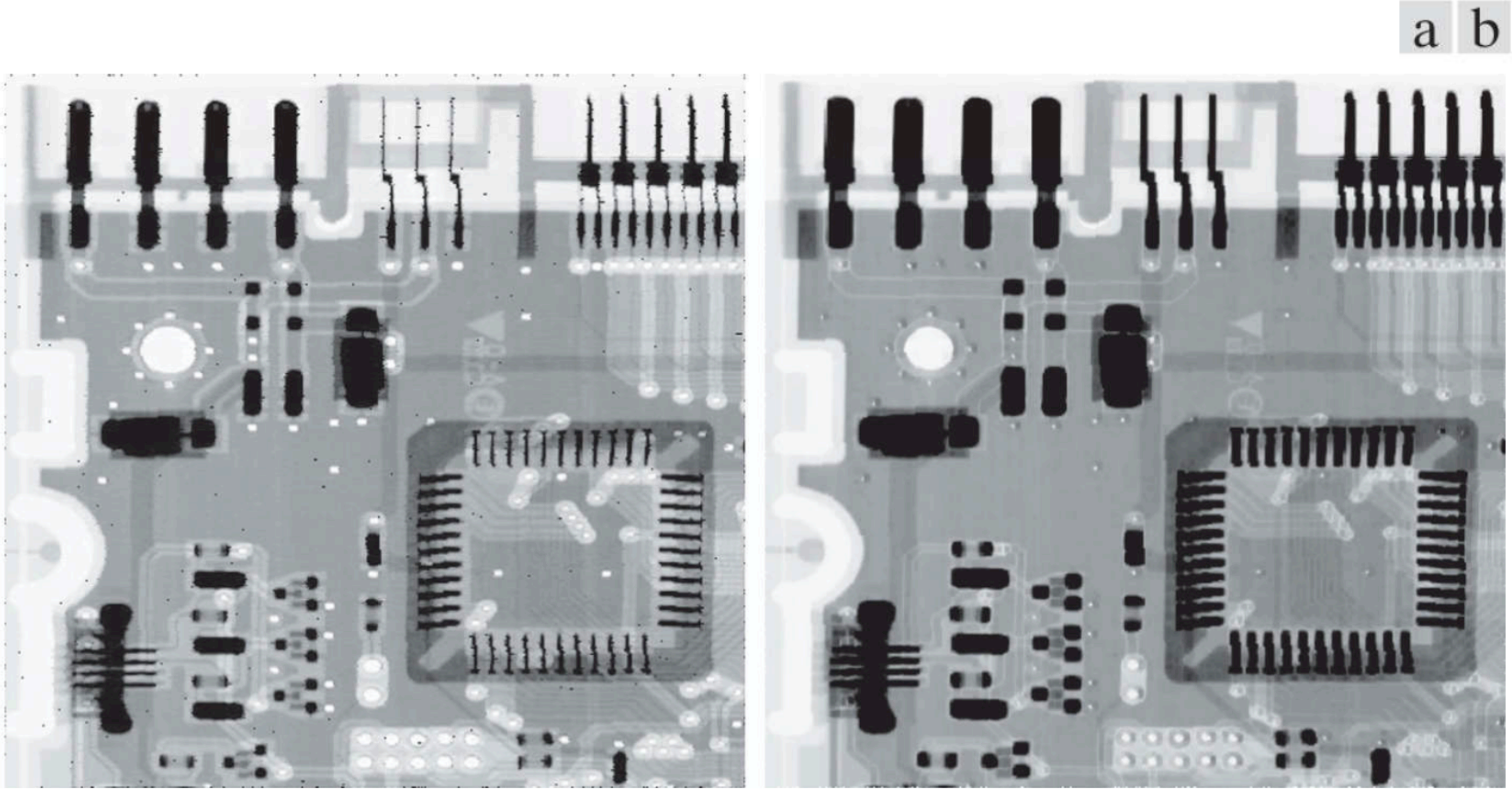
# Examples:Image denoising using order-statistic filters

(a) Image corrupted by salt-and-pepper noise with probabilities = = 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.



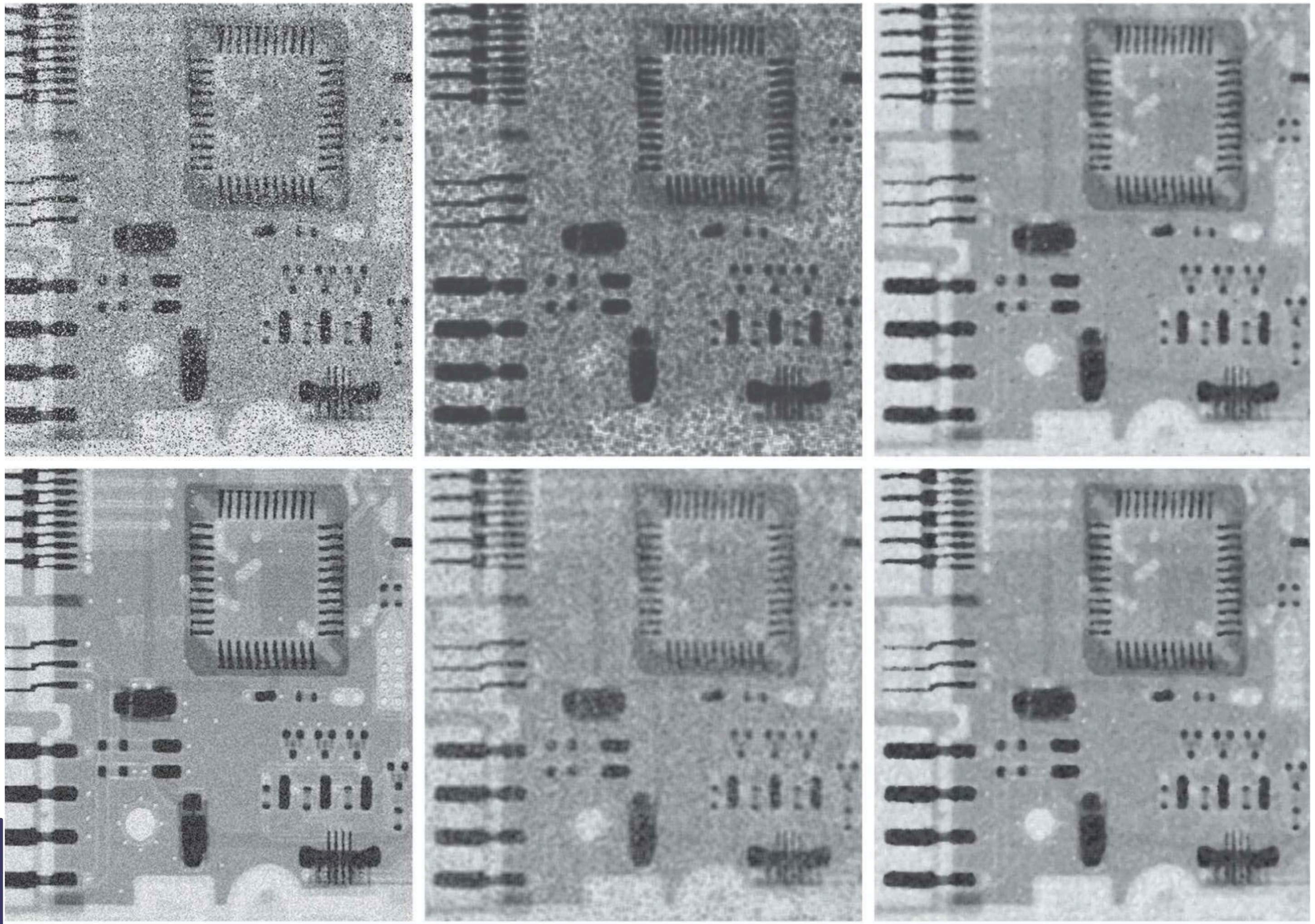
# Examples:Image denoising using order-statistic filters

(a) Result of filtering image adding pepper noise with a max filter of size 3 × 3. (b) Result of filtering image adding salt noise with a min filter of the same size.



# Examples:Image denoising using order-statistic filters

(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 = 6.



| | | |
|---|---|---|
| b | d | f |
| a | c | e |

# Adaptive Filters

- Once selected, the filters discussed thus far are applied to an image without regard for how image characteristics vary from one point to another. In this section, we take a look at two adaptive filters whose behavior changes based on statistical characteristics of the image inside the filter region defined by the m×n rectangular neighborhood $S_{xy}$. As the following discussion shows, adaptive filters are capable of performance superior to that of the filters discussed thus far. The price paid for improved filtering power is an increase in filter complexity. Keep in mind that we still are dealing with the case in which the degraded image is equal to the original image plus noise. No other types of degradations are being considered yet.

# Adaptive,Local Noise Reduction Filter

- Our filter is to operate on a neighborhood, $S_{xy}$, centered on coordinates (x, y). The response of the filter at (x, y) is to be based on the following quantities: g(x, y), the value of the noisy image at (x, y); $\sigma_\eta^2$, the variance of the noise; $\bar{z}_{S_{xy}}$, the local average intensity of the pixels in $S_{xy}$; and $\sigma_{S_{xy}}^2$, the local variance of the intensities of pixels in $S_{xy}$. We want the behavior of the filter to be as follows:

1. If $\sigma_\eta^2$ is zero, the filter should return simply the value of g at (x, y). This is the trivial, zero-noise case in which g is equal to f at (x, y).
2. If the local variance $\sigma_{S_{xy}}^2$ is high relative to $\sigma_\eta^2$, the filter should return a value close to g at (x, y). A high local variance typically is associated with edges, and these should be preserved.
3. If the two variances are equal, we want the filter to return the arithmetic mean value of the pixels in . This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced by averaging.

# Adaptive,Local Noise Reduction Filter

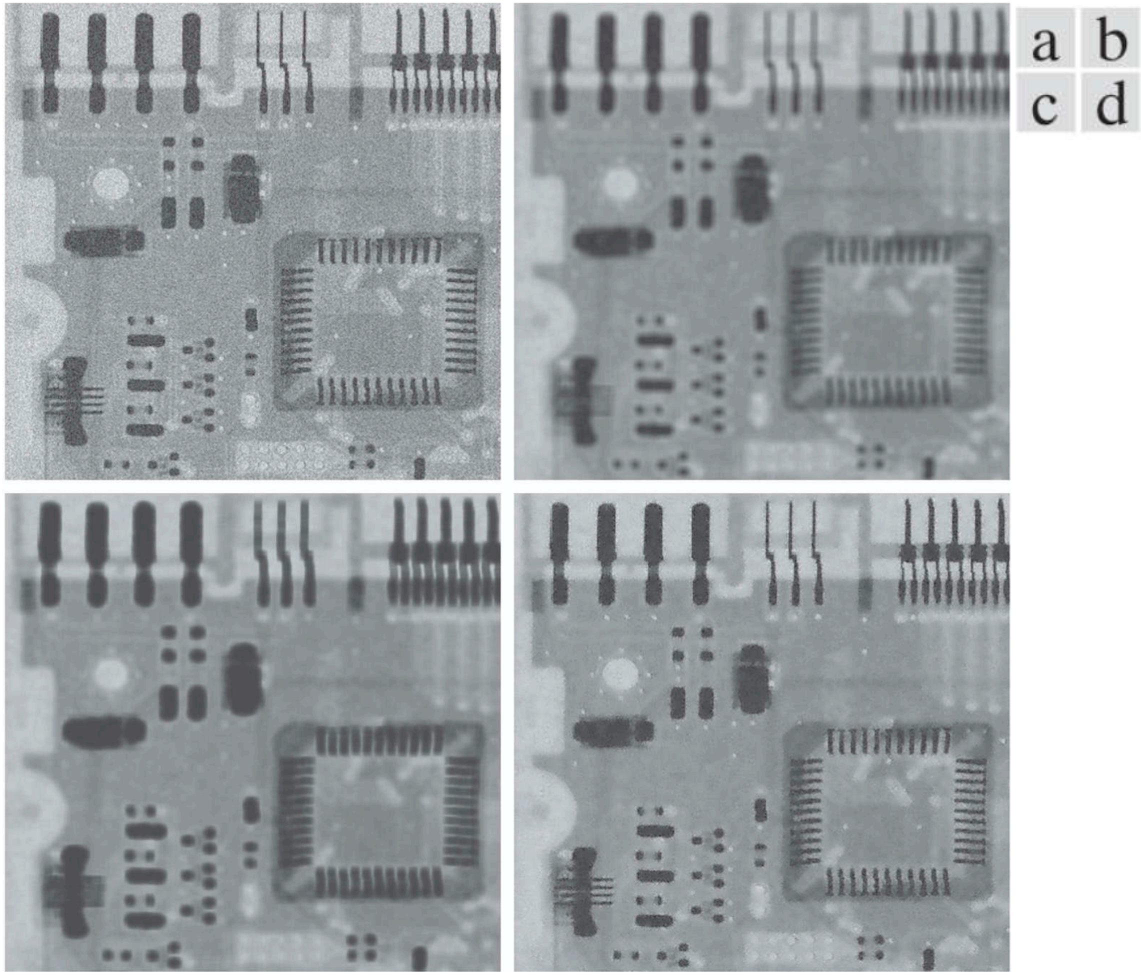
- An adaptive expression for obtaining $\hat{f}(x, y)$ based on these assumptions may be written as:

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{S_{xy}}^2}\left[g(x,y) - \bar{z}_{S_{xy}}\right]$$

The only quantity that needs to be known a priori is , the variance of the noise corrupting image f(x, y). This is a constant that can be estimated

# Example:Image denoising using adaptive,local noise-reduction filtering

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



# Adaptive Median Filter

- We use fellow notation:

$$
\begin{aligned}
z_{\min} &= \text{minimum intensity value in } S_{xy} \\
z_{\max} &= \text{maximum intensity value in } S_{xy} \\
z_{\text{med}} &= \text{median of intensity values in } S_{xy} \\
z_{xy} &= \text{intensity at coordinates } (x, y) \\
S_{\max} &= \text{maximum allowed size of } S_{xy}
\end{aligned}
$$

# Adaptive Median Filter

The adaptive median-filtering algorithm uses two processing levels, denoted level A and level B , at each point (x, y):

Level $A$: If $z_{\text{min}} < z_{\text{med}} < z_{\text{max}}$, go to Level $B$

Else, increase the size of $S_{xy}$

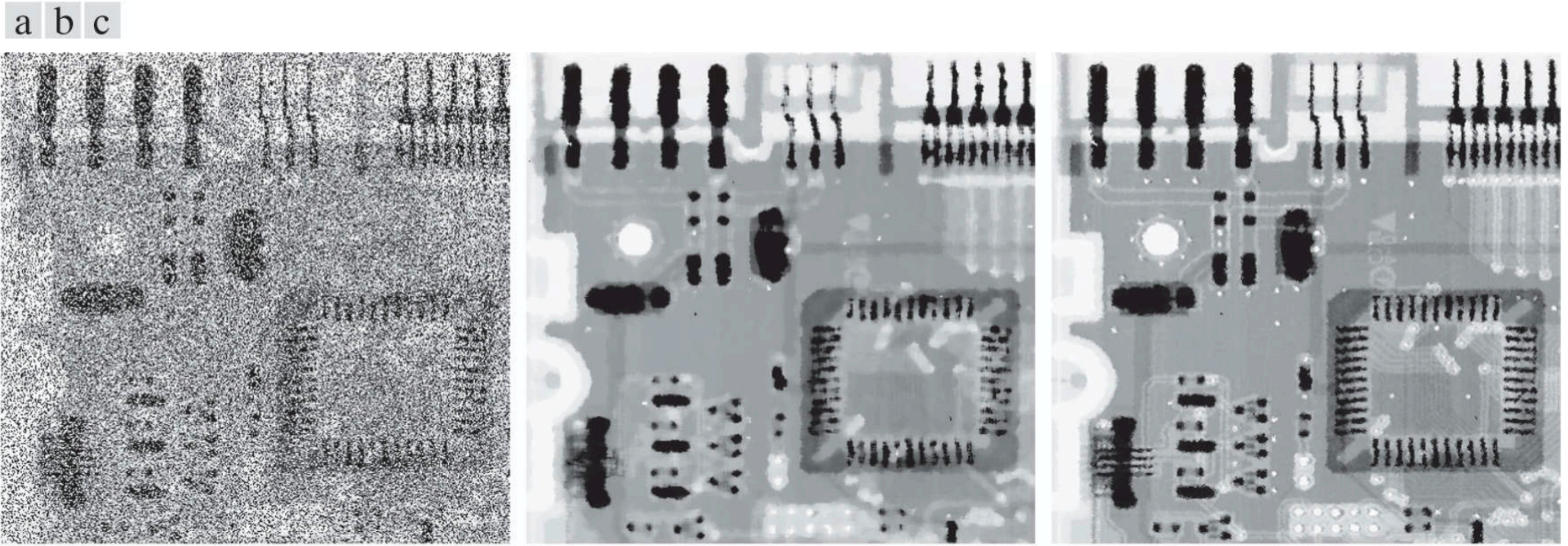
If $S_{xy} \leq S_{\text{max}}$, repeat level $A$

Else, output $z_{\text{med}}$ .

Level $B$: If $z_{\text{min}} < z_{xy} < z_{\text{max}}$, output $z_{xy}$

Else output $z_{\text{med}}$ .

# Example:Image denoising using adaptive median filtering



(a) Image corrupted by salt-and-pepper noise with probabilities = = 0.25. (b) Result of filtering with a 7 $\times$ 7 median filter. (c) Result of adaptive median filtering with = 7.

# Periodic Noise Reduction Using Frequency Domain Filtering

- Periodic noise can be analyzed and filtered quite effectively using frequency domain techniques. The basic idea is that periodic noise appears as concentrated bursts of energy in the Fourier transform, at locations corresponding to the frequencies of the periodic interference.

# More on Notch Filtering

- Notch reject filter transfer functions are constructed as products of high-pass filter transfer functions whose centers have been translated to the centers of the notches. The general form of a notch filter transfer function is :

$$H_{\mathrm{NR}}(u,v) = \prod_{k=1}^{Q} H_k(u,v)H_{-k}(u,v)$$

where $H_k(u,v)$ and $H_{-k}(u,v)$ are high-pass filter transfer functions whose centers are at $(u_k, v_k)$ and $(-u_k, -v_k)$, respectively. These centers are specified with respect to the center of the frequency rectangle, [floor( M/2), floor(N/2)], where, as usual, M and N are the number of rows and columns in the input image.

# More on Notch Filtering

- The distance computations for the filter transfer functions are given by:

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

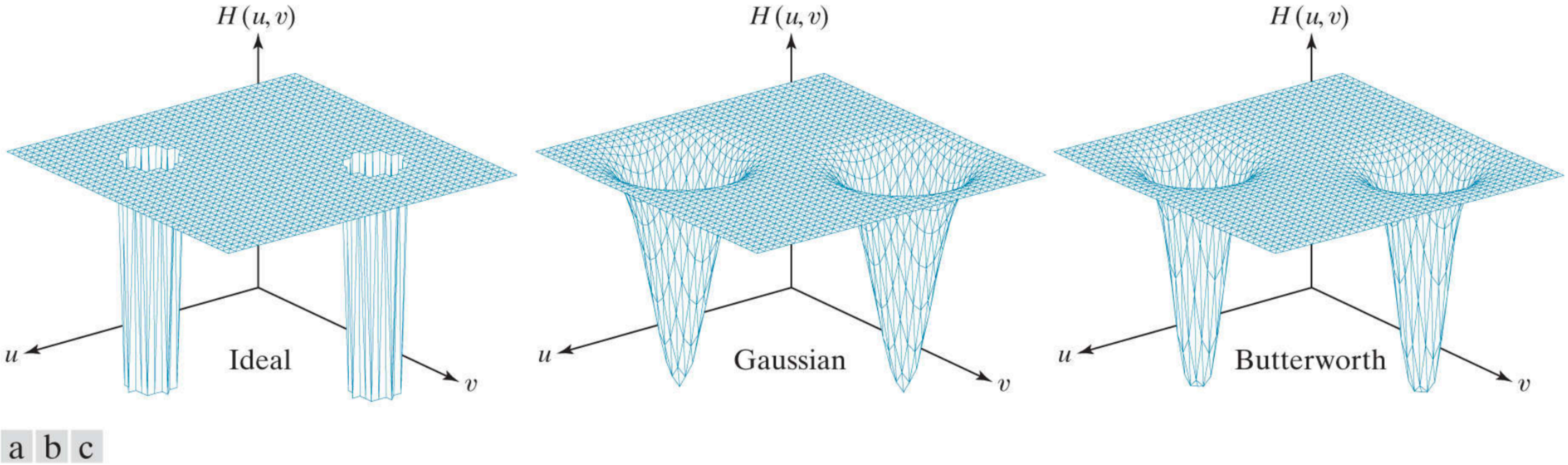
and

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

For example, the following is a Butterworth notch reject filter transfer function of order n with three notch pairs:

$$H_{\mathrm{NR}}(u,v) = \prod_{k=1}^{3} \left[\frac{1}{1 + [D_{0k}/D_k(u,v)]^n}\right]\left[\frac{1}{1 + [D_{0k}/D_{-k}(u,v)]^n}\right]$$

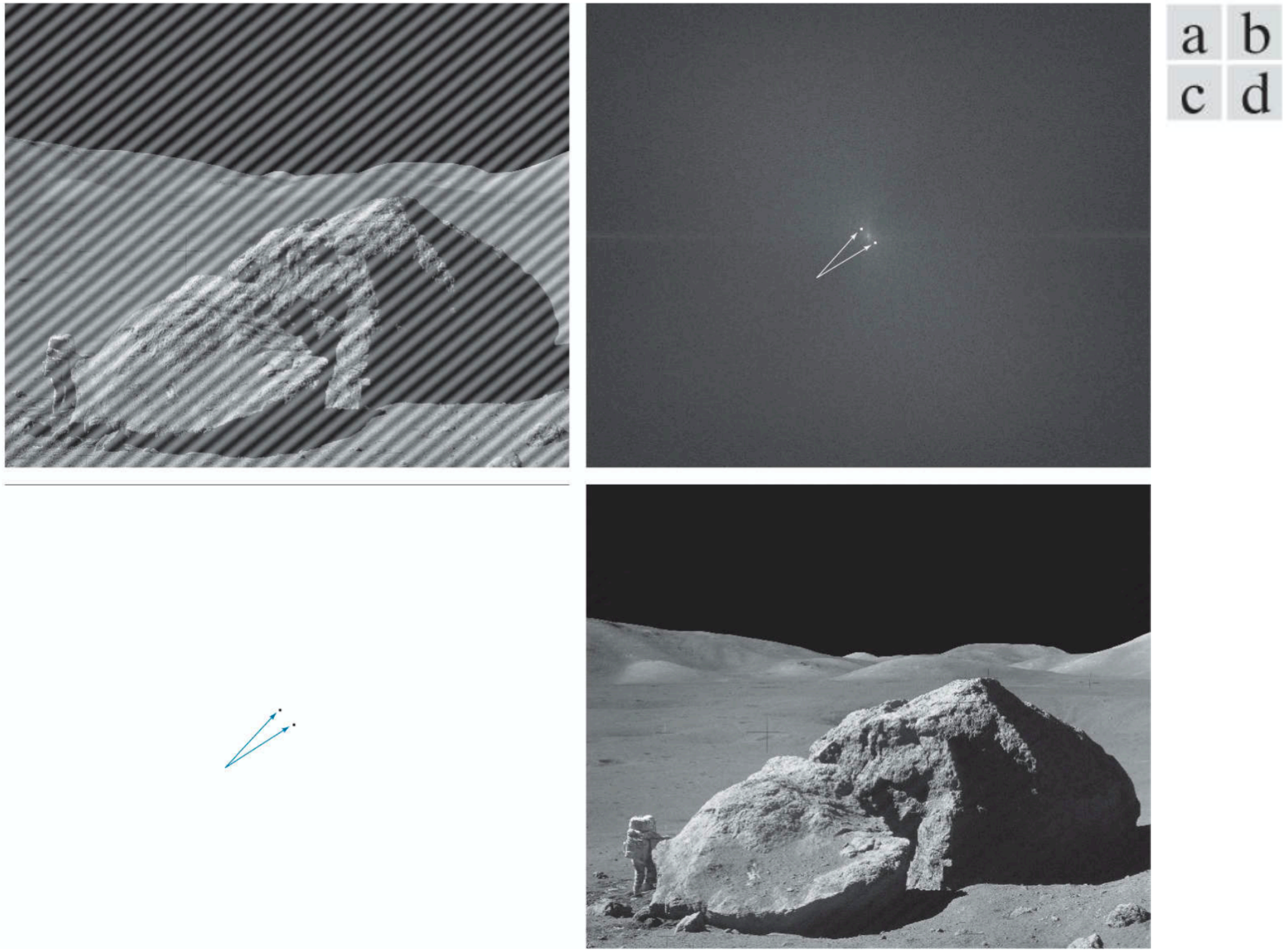
# More on Notch Filtering




Perspective plots of (a) ideal, (b) Gaussian, and (c) Butterworth notch reject filter transfer functions.

# Example:Image denoising (interference reduction) using notch filtering

(a) Image corrupted by sinusoidal interference. (b) Spectrum showing the bursts of energy caused by the interference. (The bursts were enlarged for display purposes.) (c) Notch filter (the radius of the circles is 2 pixels) used to eliminate the energy bursts. (The thin borders are not part of the data.) (d) Result of notch reject filtering.



# Linear, Position-Invariant Degradations

- The expression:

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

which is called the superposition (or Fredholm) integral of the first kind. This expression is a fundamental result that is at the core of linear system theory. It states that if the response of H to an impulse is known, the response to any input $f(\alpha,\beta)$ can be calculated using equation avove. In other words, a linear system H is characterized completely by its impulse response.

- If H is position invariant, then:

$$\mathcal{H}[\delta(x-\alpha,y-\beta)] = h(x-\alpha,y-\beta)$$

# Linear, Position-Invariant Degradations

- Then we have:

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

- In the presence of additive noise, if H is position invariant, the expression of the linear degradation model becomes:

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

# Inverse Filtering

- The material in this section is our first step in studying restoration of images degraded by a degradation function H, which is given, or is obtained by a method such as those discussed in the previous section. The simplest approach to restoration is direct inverse filtering, where we compute an estimate, $\hat{F}(u,v)$, of the transform of the original image by dividing the transform of the degraded image, G(u, v), by the degradation transfer function:

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

# Inverse Filtering

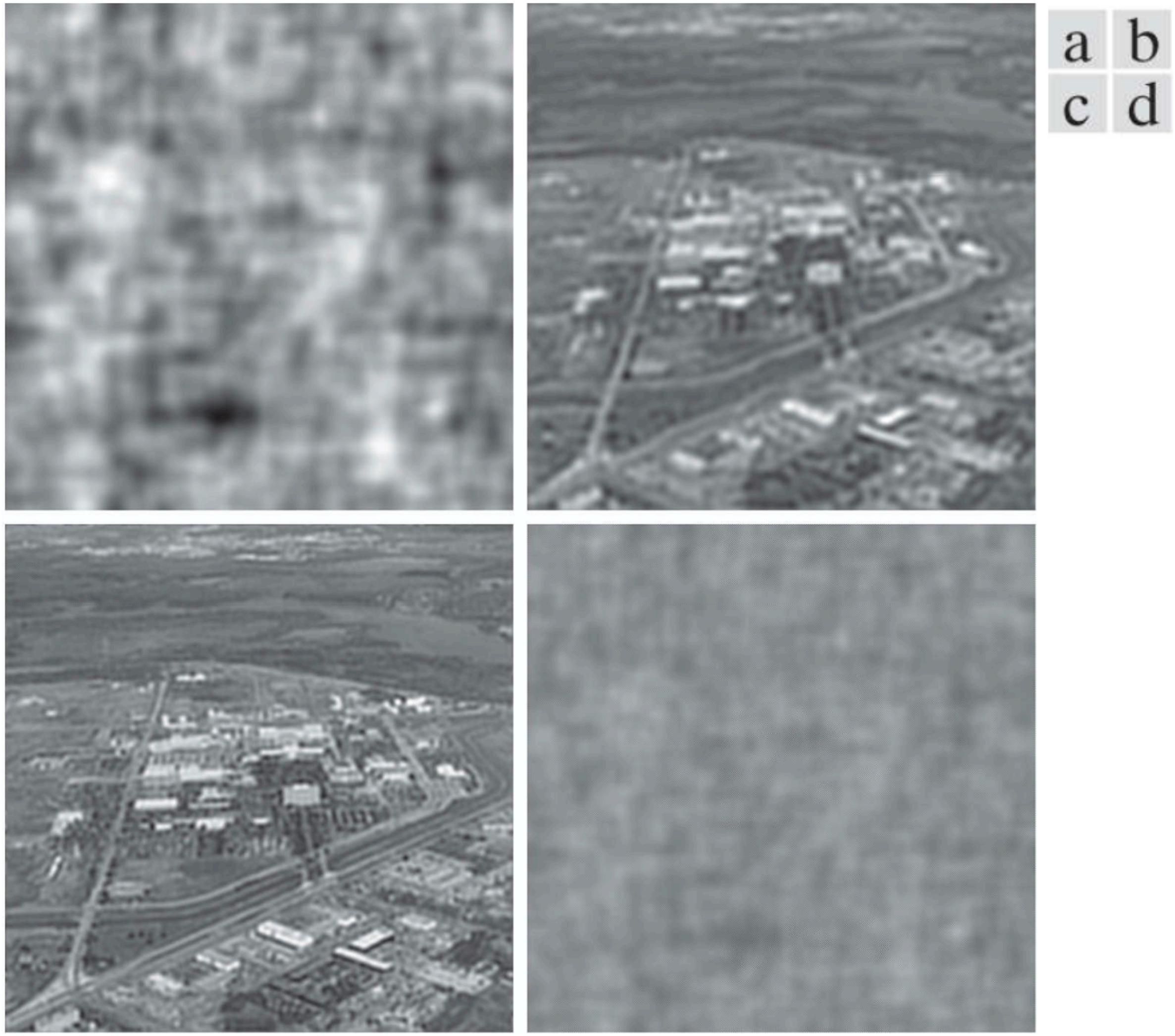
- The division is elementwise. Substituting the G(u, v) yields:

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

This is an interesting expression. It tells us that, even if we know the degradation function, we cannot recover the undegraded image [the inverse Fourier transform of F(u, v)] exactly because N(u, v) is not known. There is more bad news. If the degradation function has zero or very small values, then the ratio N(u, v)/H(u, v) could easily dominate the term F(u, v). In fact, this is frequently the case, as you will see shortly.

# Example:Image deblurring by inverse filtering

Restoring image using inverse filtering. (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40. (c) Result with H cut off outside a radius of 70. (d) Result with H cut off outside a radius of 85.



# Minimum Mean Square Error (Wiener) Filtering

- The expression of error is:

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

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

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

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

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

# Minimum Mean Square Error (Wiener) Filtering

- The terms in equation above are as follows:

1. $\hat{F}(u,v)$ = Fourier transform of the estimate of the undegraded image.
2. $G(u,v)$ = Fourier transform of the degraded image.
3. $H(u,v)$ = degradation transfer function (Fourier transform of the spatial degradation).
4. $H^*(u,v)$ = complex conjugate of $H(u,v)$.
5. $|H(u,v)|^2 = H^*(u,v)H(u,v)$.
6. $S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise [see Eq. (4-89) ]†

   † The term $|N(u,v)|^2$ also is referred to as the *autocorrelation* of the noise. This term comes from the correlation theorem (first line of entry 7 in Table 4.4 ). When the two functions are the same, correlation becomes *autocorrelation* and the right side of that entry becomes $H^*(u,v)H(u,v)$, which is equal to $|H(u,v)|^2$. Similar comments apply to $|F(u,v)|^2$, which is the autocorrelation of the image. We will discuss correlation in more detail in Chapter 12 .

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

# Example: Comparison of deblurring by inverse and Wiener filtering


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.

# Constrained Least Squares Filtering

- One way to reduce the effects of noise sensitivity, is to base optimality of restoration on a measure of smoothness, such as the second derivative of an image (our old friend, the Laplacian). To be meaningful, the restoration must be constrained by the parameters of the problems at hand. Thus, what is desired is to find the minimum of a criterion function, C, defined as:

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[ \nabla^2 f(x,y) \right]^2$$

subject to the constraint

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\eta\|^2$$

# Constrained Least Squares Filtering

- The frequency domain solution to this optimization problem is given by the expression:

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

where $\gamma$ is a parameter that must be adjusted so that the constraint above is satisfied, and $P(u,v)$ is the Fourier transform of the function:

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

We recognize this function as a Laplacian kernel. Note that the expression reduces to inverse filtering if $\gamma = 0$.

# Example: Comparison of deblurring by Wiener and constrained least squares filtering.

a b c


Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results above, respectively.

# Thank You!