Phục hồi ảnh

Image Restoration

Phục hồi ảnh

- Most images obtained by optical, electronic, or electro-optic means is likely to be degraded.
- The degradation can be due to camera misfocus, relative motion between camera and object, noise in electronic sensors, atmospheric turbulence, etc.
- The goal of image restoration is to obtain a relatively “clean” image from the degraded observation.
- It involves techniques like filtering, noise reduction etc.

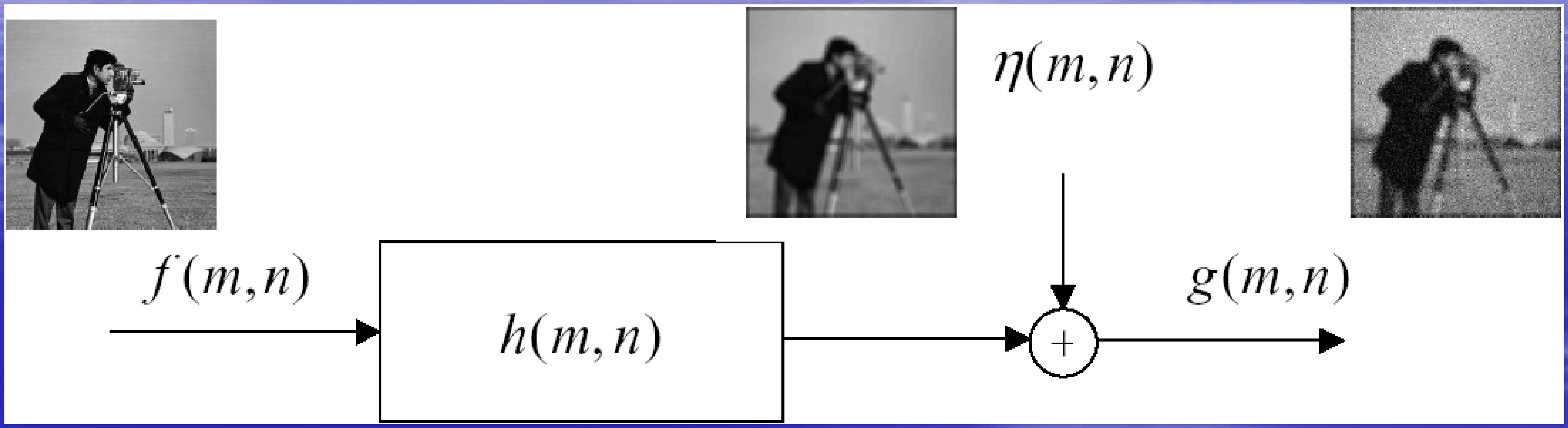
Phục hồi – Tăng cường

- **Restoration**:
  - A process that attempts to reconstruct or recover an image that has been degraded by using some prior knowledge of the degradation phenomenon.
  - Involves modeling the degradation process and applying the inverse process to recover the original image.
  - A criterion for "goodness" is required that will recover the image in an optimal fashion with respect to that criterion.
  - Ex. Removal of blur by applying a deblurring function.

Phục hồi – Tăng cường

- **Enhancement**:
  - Manipulating an image in order to take advantage of the psychophysics of the human visual system.
  - Techniques are usually “heuristic.”
  - Ex. Contrast stretching, histogram equalization.

(Linear) Degradation Model



$$g(m,n) = f(m,n)*h(m,n) + \eta(m,n)$$

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

f(m,n) : Degradation free image

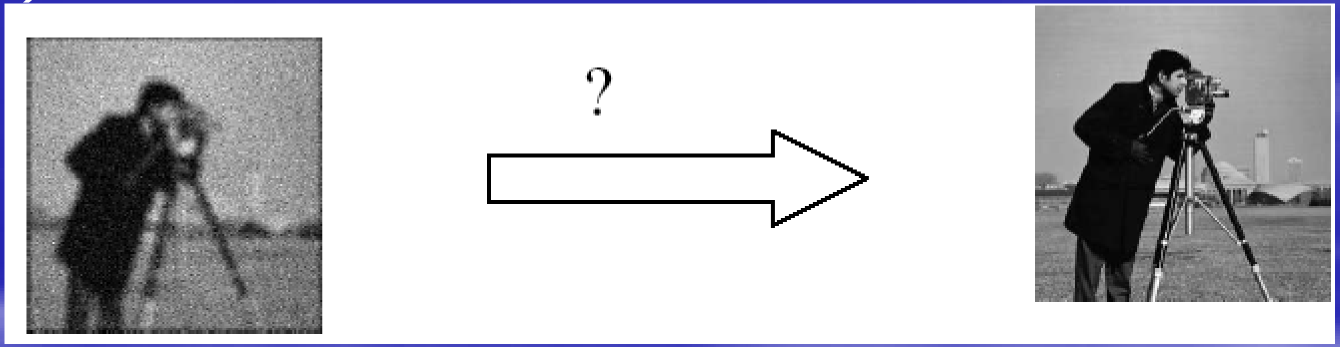
g(m,n) : Observed image

h(m,n) : PSS of blur degradation

η*(m,n)* : Additive Noise

(Linear) Degradation Model

Problem: Given an observed image *g(m,n) , to recover the* original image *f(m,n) , using knowledge about the blur* function *h(m,n)* and the characteristics of the noise $\eta(m,n)$?



- We need to find an image *^f (m,n)* , such that the error *f (m,n) - ^f (m,n)* is "small."

Mô hình nhiễu

- With the exception of periodic interference, we will assume that noise values are uncorrelated from pixel to pixel and with the (uncorrupted) image pixel values.
- These assumptions are usually met in practice and simplify the analysis.
- With these assumptions in hand, we need to only describe the statistical properties of noise; i.e., its probability density function (PDF).

Nhiễu Gauss

- Mathematically speaking, it is the most tractable noise model.
- Therefore, it is often used in practice, even in situations where they are not well justified from physical principles.
- The pdf of a Gaussian random variable *z* is given by:

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

  where *z* represents (noise) gray value, m is the mean, and s is its standard deviation. The squared standard deviation σ^2 is usually referred to as variance
- For a Gaussian pdf, approximately 70% of the values are within one standard deviation of the mean and 95% of the values are within two standard deviations of the mean.

Nhiễu Rayleigh

- The pdf of a Rayleigh noise is given by:

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

- The mean and variance are given by:

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

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

- This noise is “one-sided” and the density function is skewed.

Nhiễu Erlang(Gama)

- The pdf of Erlang noise is given by:

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

  where, *a* > 0, *b* is an integer and "!" represents factorial.
- The mean and variance are given by:

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

- This noise is "one-sided" and the density function is skewed.

Nhiễu dạng hàm mũ

- The pdf of exponential noise is given by:

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

  where, *a* > 0.
- The mean and variance are given by:

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

- This is a special case of Erlang density with *b*=1.

Uniform noise

- The pdf of uniform noise is given by:

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

where, *a* > 0, *b* is an integer and "!" represents factorial.

- The mean and variance are given by:

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

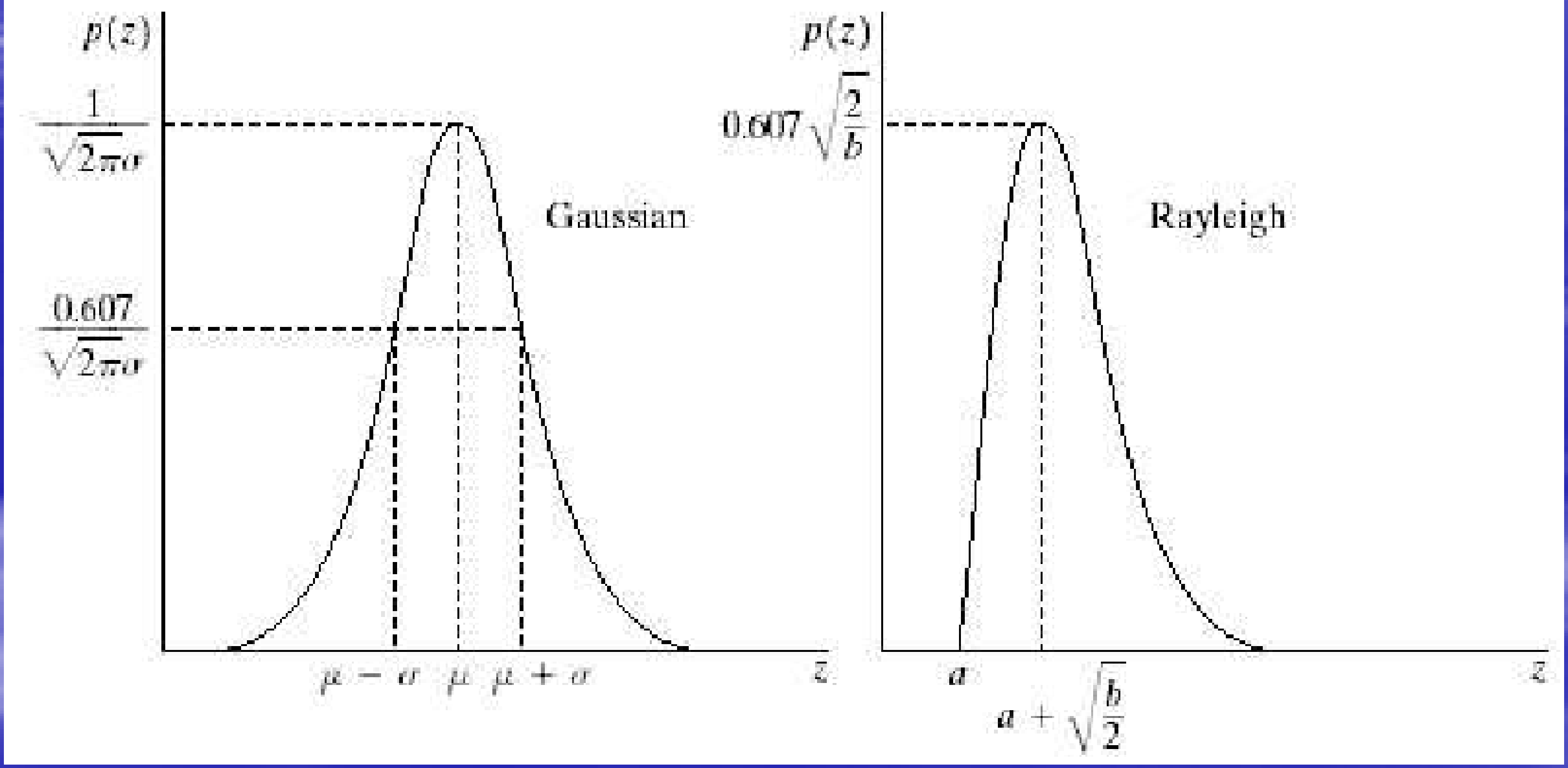
Impulse (salt-and-pepper) noise

- The pdf of (bipolar) impulse noise is given by:

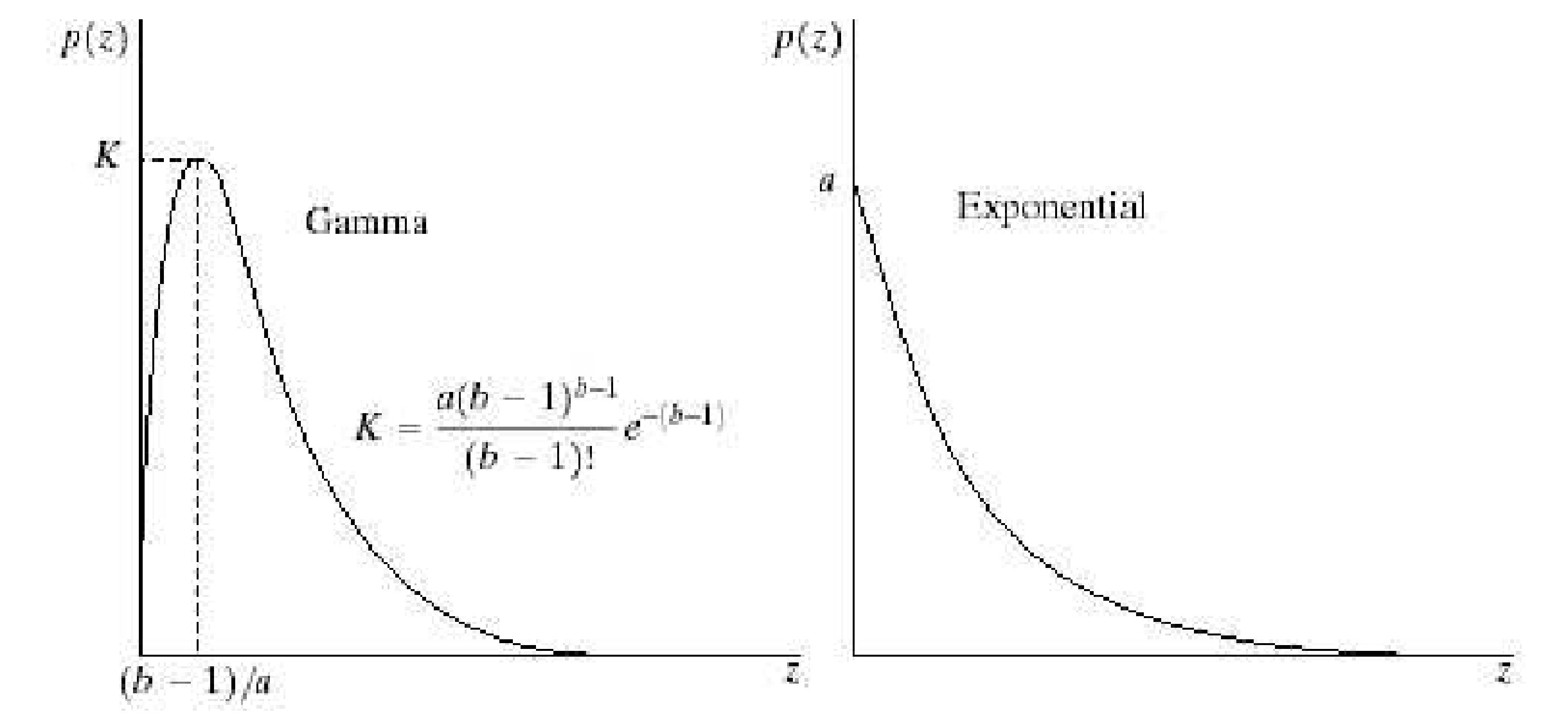
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b = 1 - P_a & \text{for } z = b \\ 0 & \text{otherwise} \end{cases},$$

where, *a* > 0, *b* is an integer and “!” represents factorial.

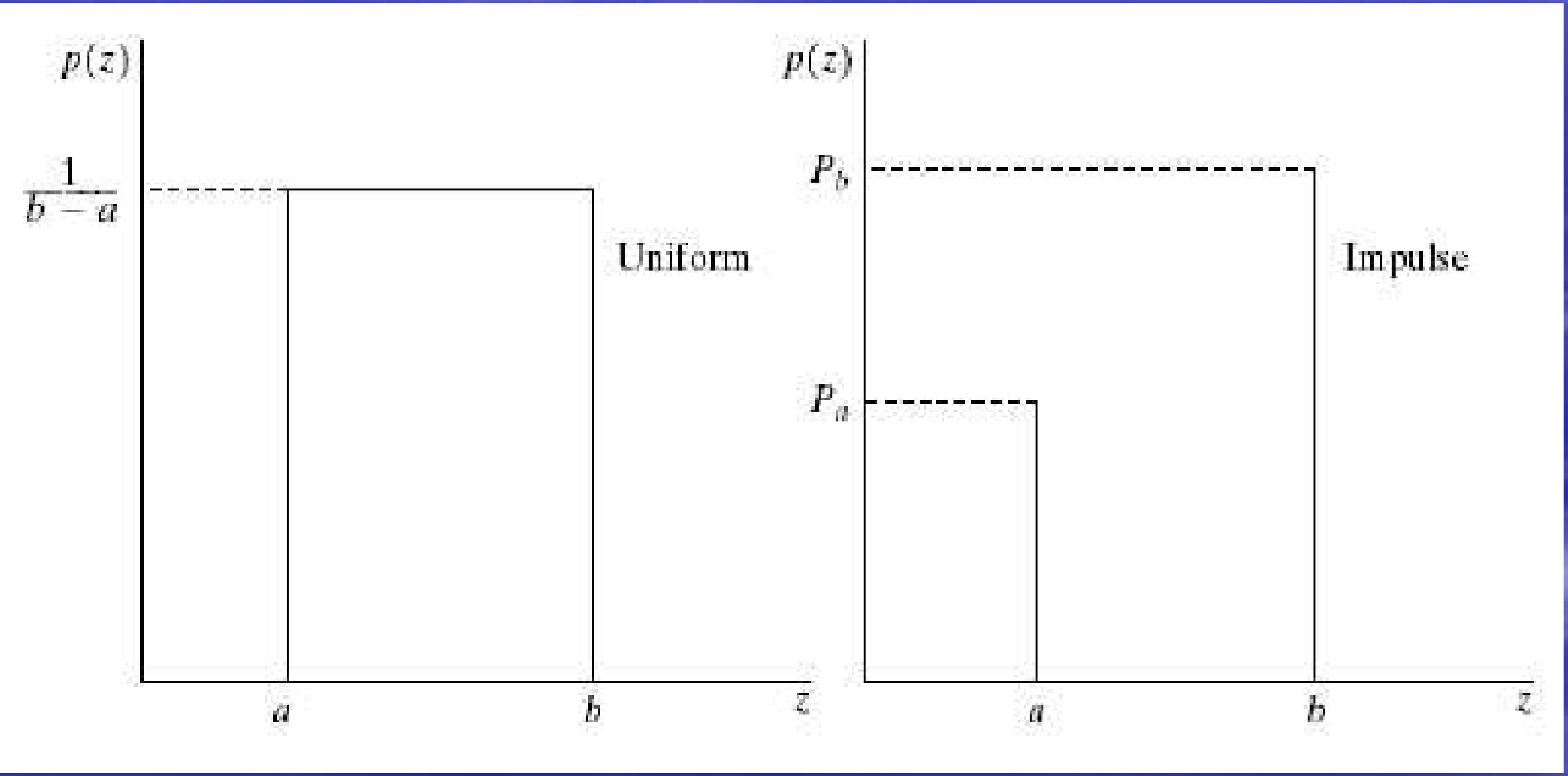
Plot of density function of different noise models



Plot of density function of different noise models



Plot of density function of different noise models



Test pattern and illustration of the effect of different types of noise

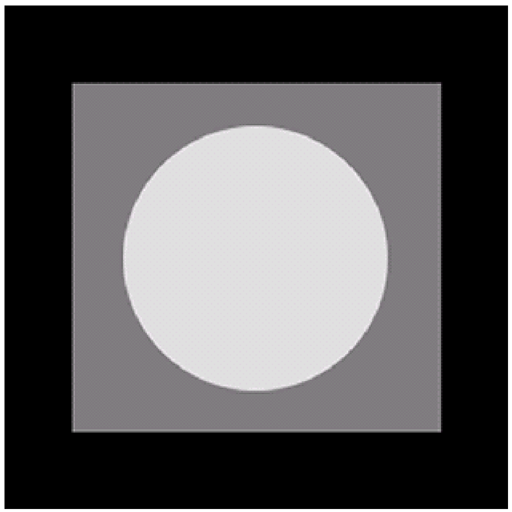


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

Test pattern and illustration of the effect of different types of noise

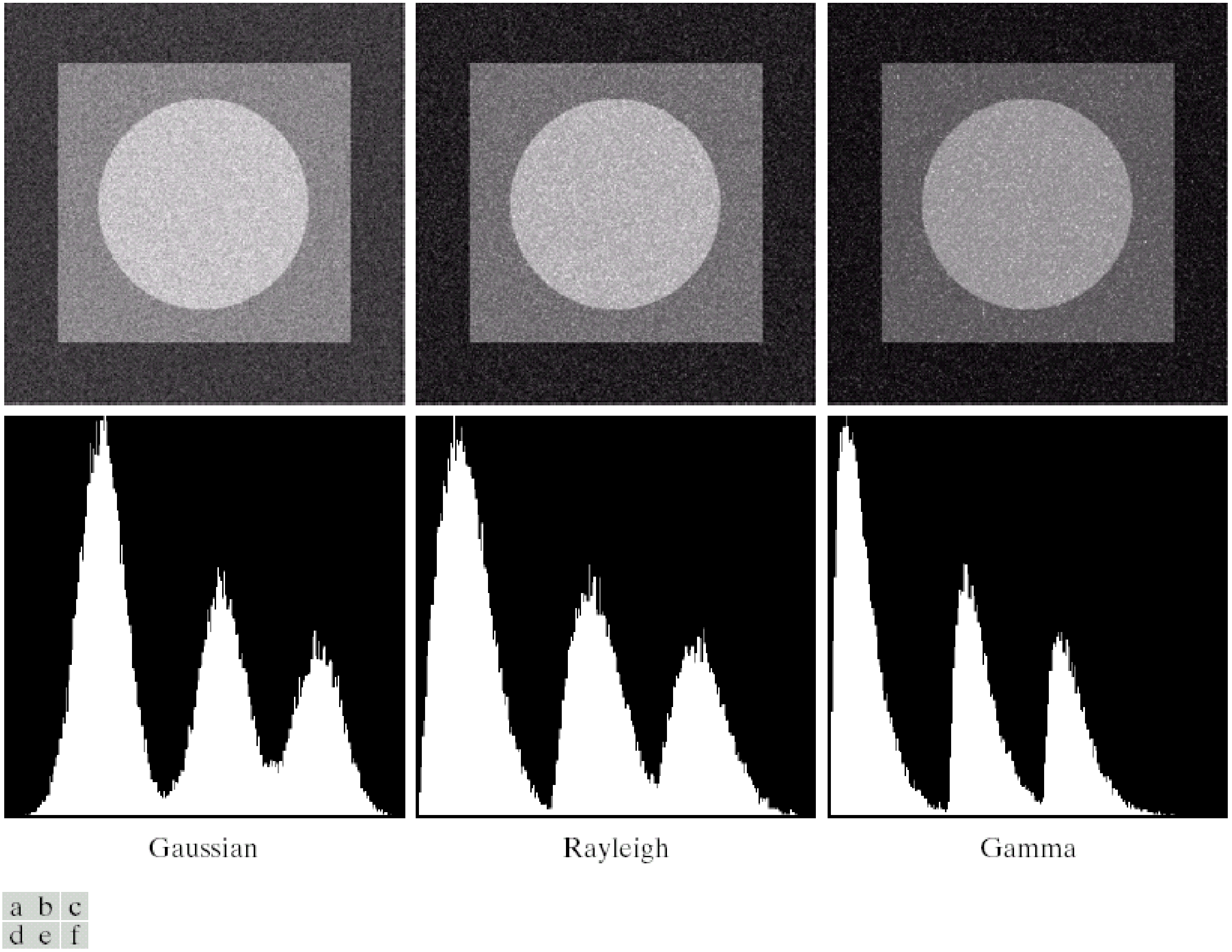


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Test pattern and illustration of the effect of different types of noise

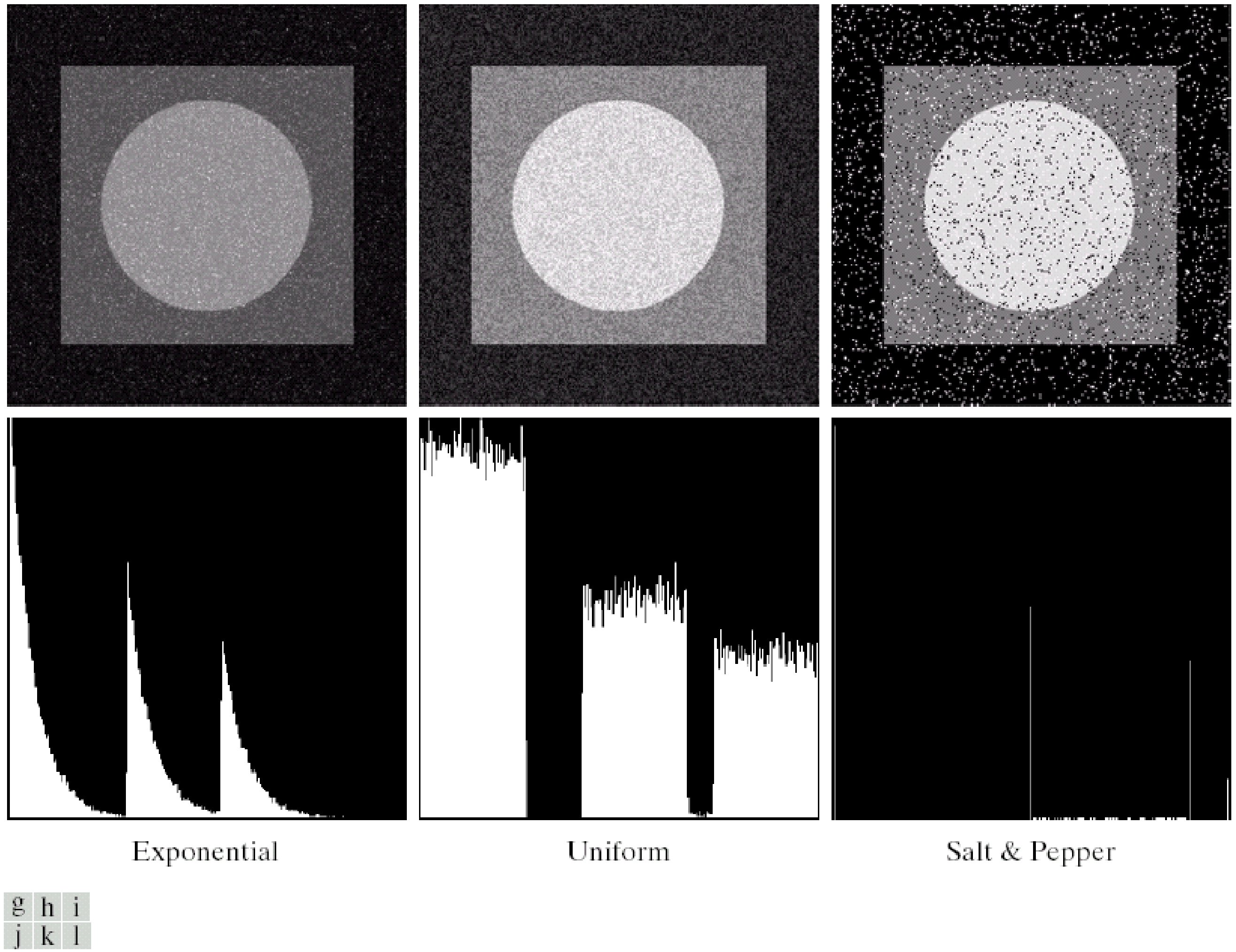


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

Estimation of noise parameters

- The noise pdf is usually available from sensor specifications. Sometimes, the form of the pdf is knowm from physical modeling.
- The pdf (or parameters of the pdf) are also often estimated from the image.
- Typically, if feasible, a flat uniformly illuminated surface is imaged using the imaging system. The histogram of the resulting image is usually a good indicator of the noide pdf.
- If that is not possible, we can usually choose a small patch of an image that is relatively uniform and compute a histogram of the image over that region.

Estimation of noise parameters

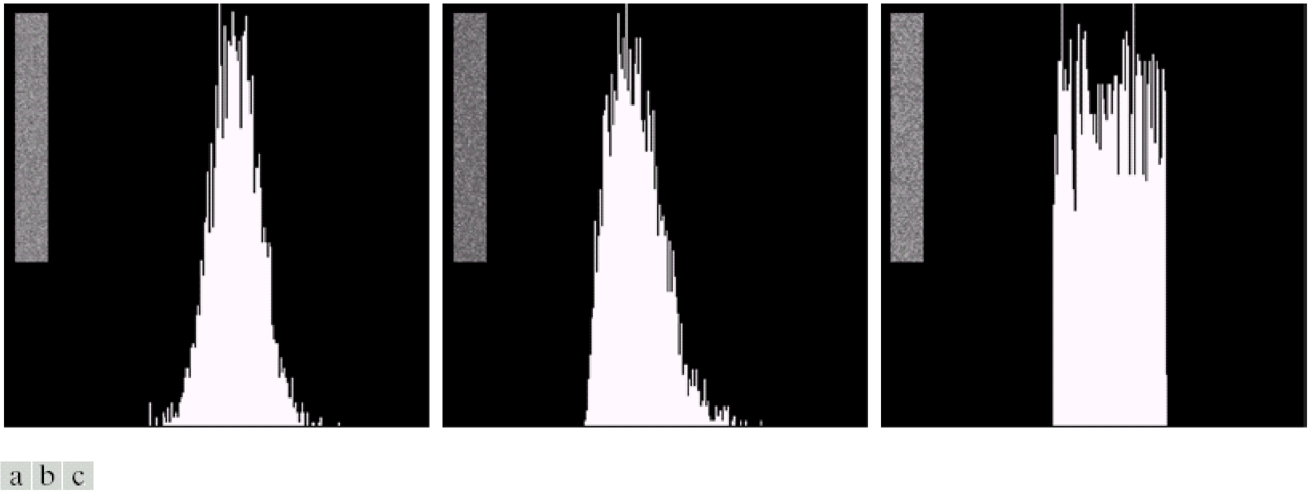


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Estimation of noise parameters

- Using the hitogram, we can estimate the noise mean and variance as follows:

$$\mu = \sum_{i \in S} z_i p(z_i)$$

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

  where z_i is the grayvalue of pixel *i* in S, and $p(z_i)$ is the histogram value.
- The shape of the histogram identifies the closest pdf match.
- The mean and variance are used to solve for the parameters *a* and *b* in the density function.

Restoration in the presence of only noise

- In this case, the degradation equation becomes:

$$g(m,n) = f(m,n) + \eta(m,n)$$

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

- Spatial filtering is usually the best method to restore images corrupted purely by noise. The process is similar to that of image enhancement.

Mean filter

Arithmetic mean

- Let S_{ab} be a rectangular window of size $a \times b$. The arithmetic mean filter computes the average value of the pixels in $g(m,n)$ over the window S_{ab}.

$$\hat{f}(m,n) = \frac{1}{ab} \sum_{S_{ab}} g(s,t)$$

- This operation can be thought of as a convolution with a uniform rectangular mask of size $a \times b$, each of whose values is 1/ab.
- This smoothes out variations and noise is reduced.

Mean filter

Geometric mean

- The geometric mean filter computes the geometric mean of the pixels in *g(m,n)* over the window S_{ab}.

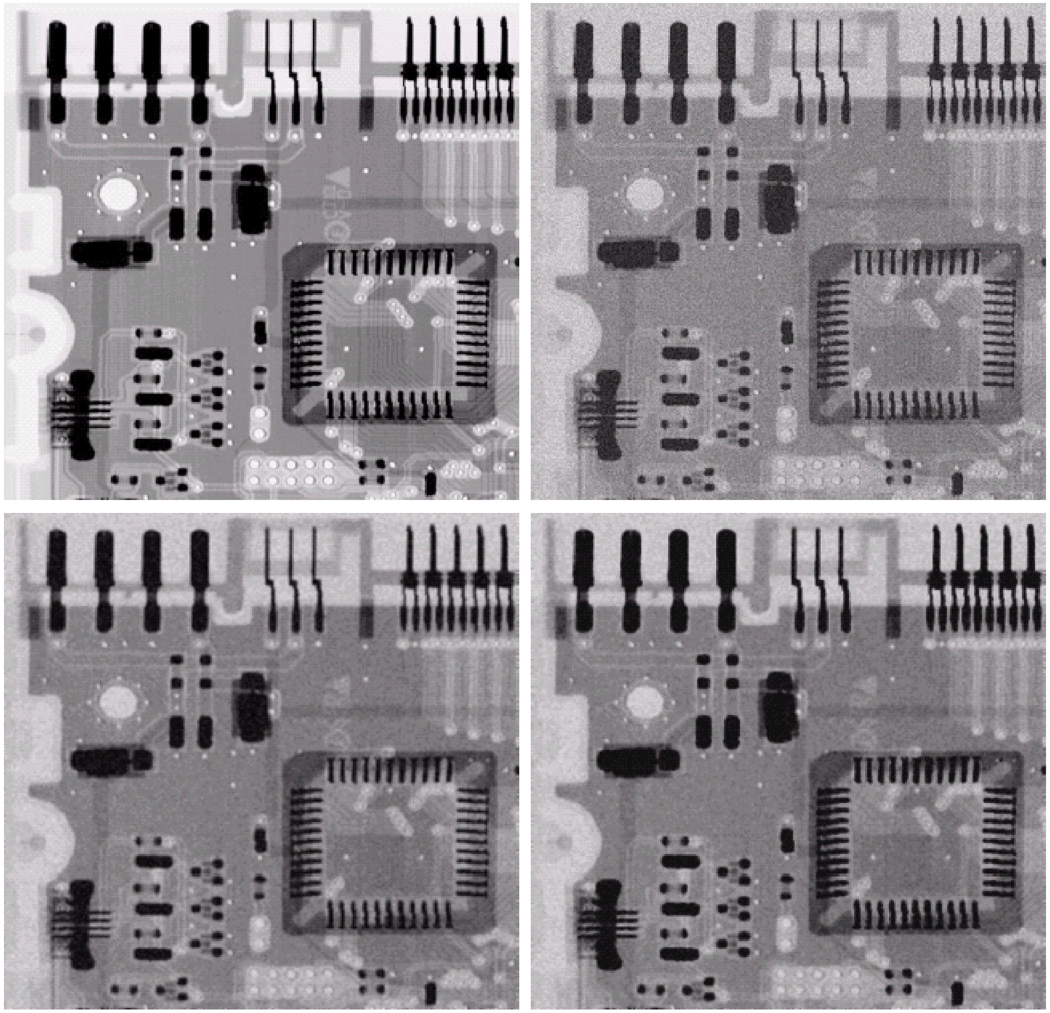
$$\hat{f}(m,n) = \left[\prod_{S_{ab}} g(s,t) \right]^{1/mn}$$

- This usually results in similar results as the arithmetic mean filter, with possibly less loss of image detail.

Example

- Image corrupted by additive Gaussian noise, mean 0 and variance 400.
- Note that the geometric mean filter has resulted in less blurred edges.

Example: additive Gaussian noise



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.)

Mean filter

Harmonic mean

- The harmonic mean filter computes the harmonic mean of the pixels in *g(m,n)* over the window S_{ab}.

$$\hat{f}(m,n) = \frac{ab}{\sum_{S_{ab}} \frac{1}{g(s,t)}}$$

- This works well for salt noise, but fails for pepper noise. It also works well with Gaussian noise.

Mean filter

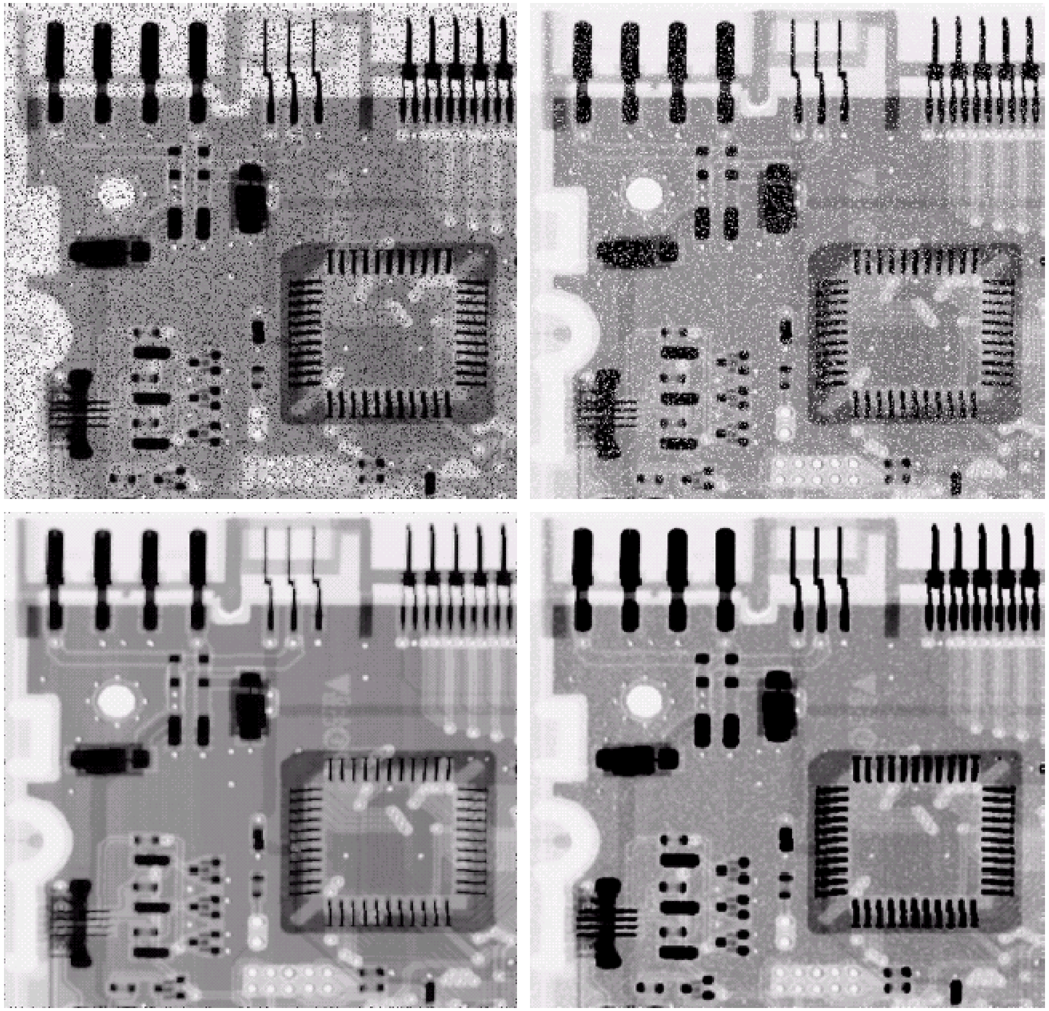
Contraharmonic mean

- The contraharmonic mean filter is given by the expression:

$$\hat{f}(m,n) = \frac{\sum_{S_{ab}} g(s,t)^{Q+1}}{\sum_{S_{ab}} g(s,t)^{Q}}$$

  where Q is called order of the filter.
- This yields the arithmetic mean filter for Q=0 and the harmonic mean filter for Q=-1.
- For positive values of Q, it reduces pepper noise and for negative values of Q, it reduces salt noise. It cannot do both simultaneously.

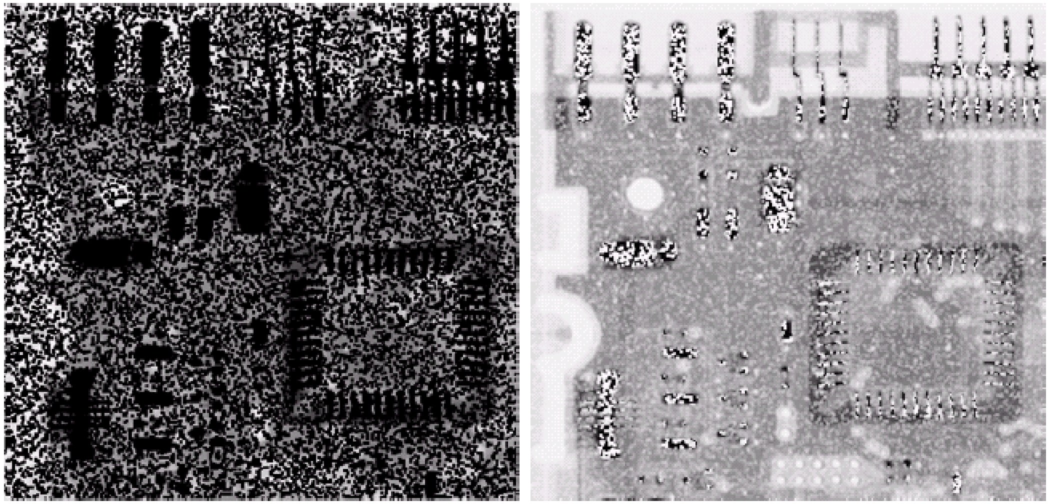
Example: salt/pepper noise



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 $Q = -1.5$.

Example: Effect of choosing the wrong sign for Q



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 Statistic filters

- Order statistic filters are obtained by first ordering (or ranking) the pixel values in a window *ab S* around a given pixel.

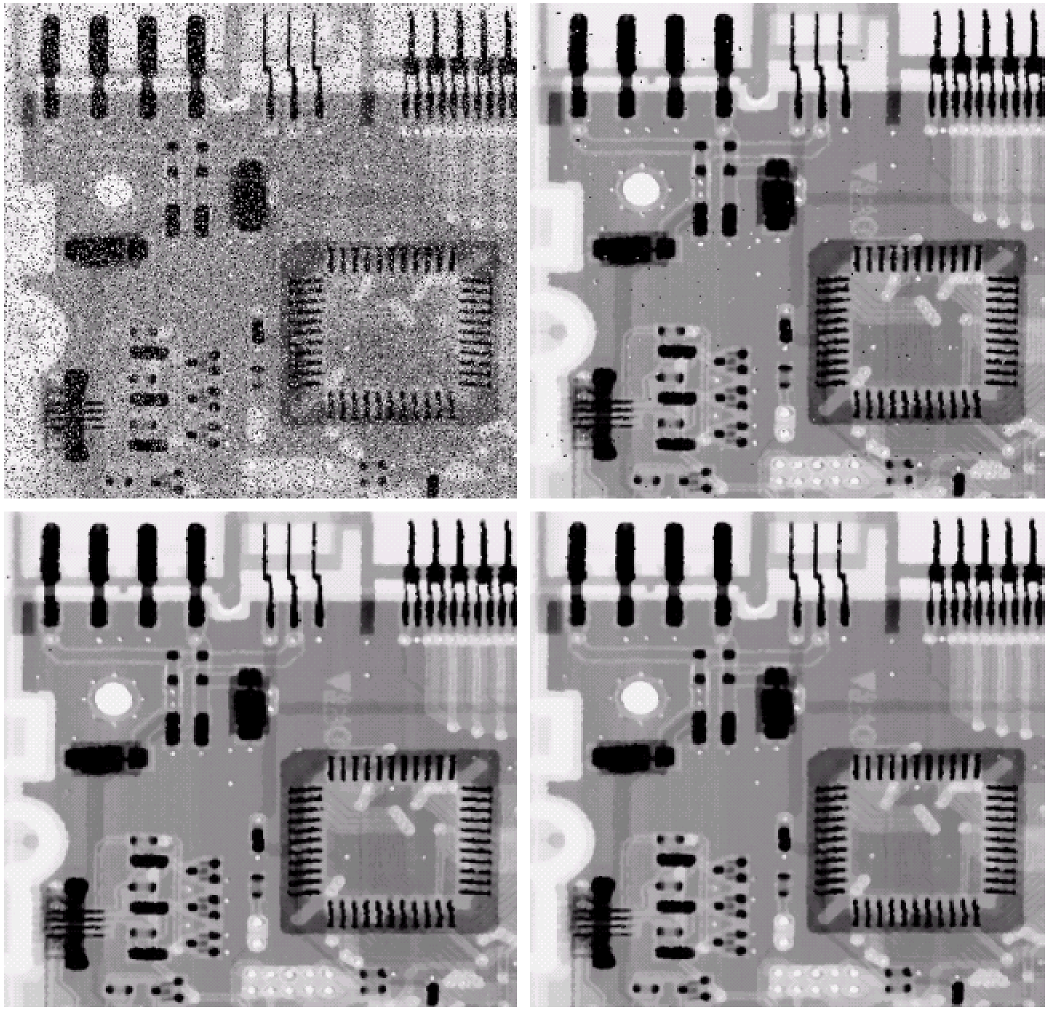
Median Filter

- It replaces the values of a pixel by the median of the gravalues in a neighborhood *ab S* of the pixel.

$$\hat{f}(m,n) = \underset{(s,t)\in S_{ab}}{\text{median}}\{g(s,t)\}$$

- Median filters are particularly suited for impulsive noise. They often result in much less loss of sharp edges in the original image.

Example of median filter



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.

Order Statistic filters

Max and Min Filter

- The Max filter replaces the values of a pixel by the maximum of the gravalues in a neighborhood S_{ab} of the pixel.

$$\hat{f}(m,n) = \max_{(s,t)\in S_{ab}} \{g(s,t)\}$$

- It is used to reduce pepper noise and to find the bright spots in an image.

Order Statistic filters

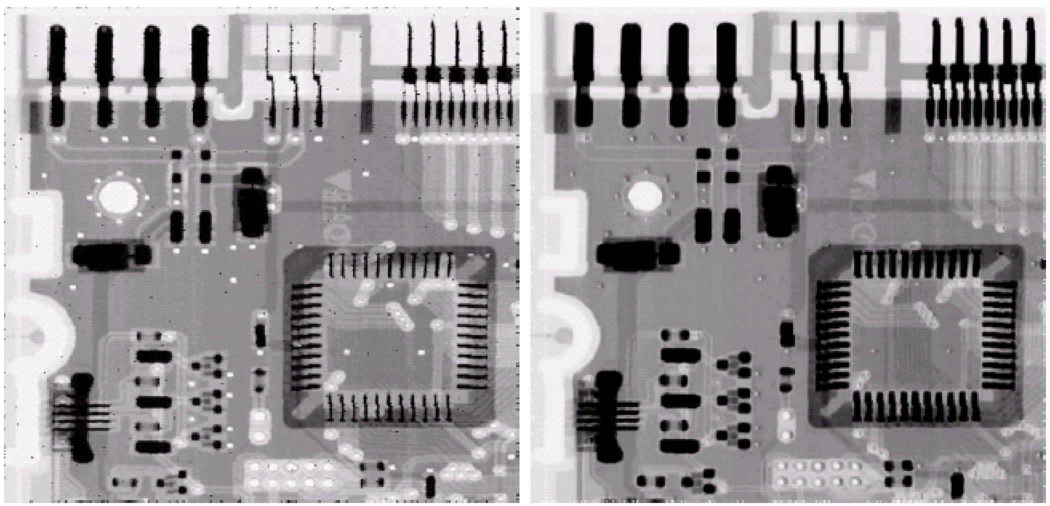
Max and Min Filter

- The Min filter replaces the values of a pixel by the minimum of the gravalues in a neighborhood S_{ab} of the pixel.

$$\hat{f}(m,n) = \min_{(s,t)\in S_{ab}} \{g(s,t)\}$$

- It is used to reduce salt noise and to find the dark spots in an image.
- Usually, the max and min filters are used in conjunction.

Example: Max and 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.

Order Statistic filters

Midpoint Filter

- The Midpoint filter replaces the values of a pixel by the midpoint (average) of the maximum and minimum of the gravalues in a neighborhood S_{ab} of the pixel.

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

- It combines order statistic with averaging.

Order Statistic filters

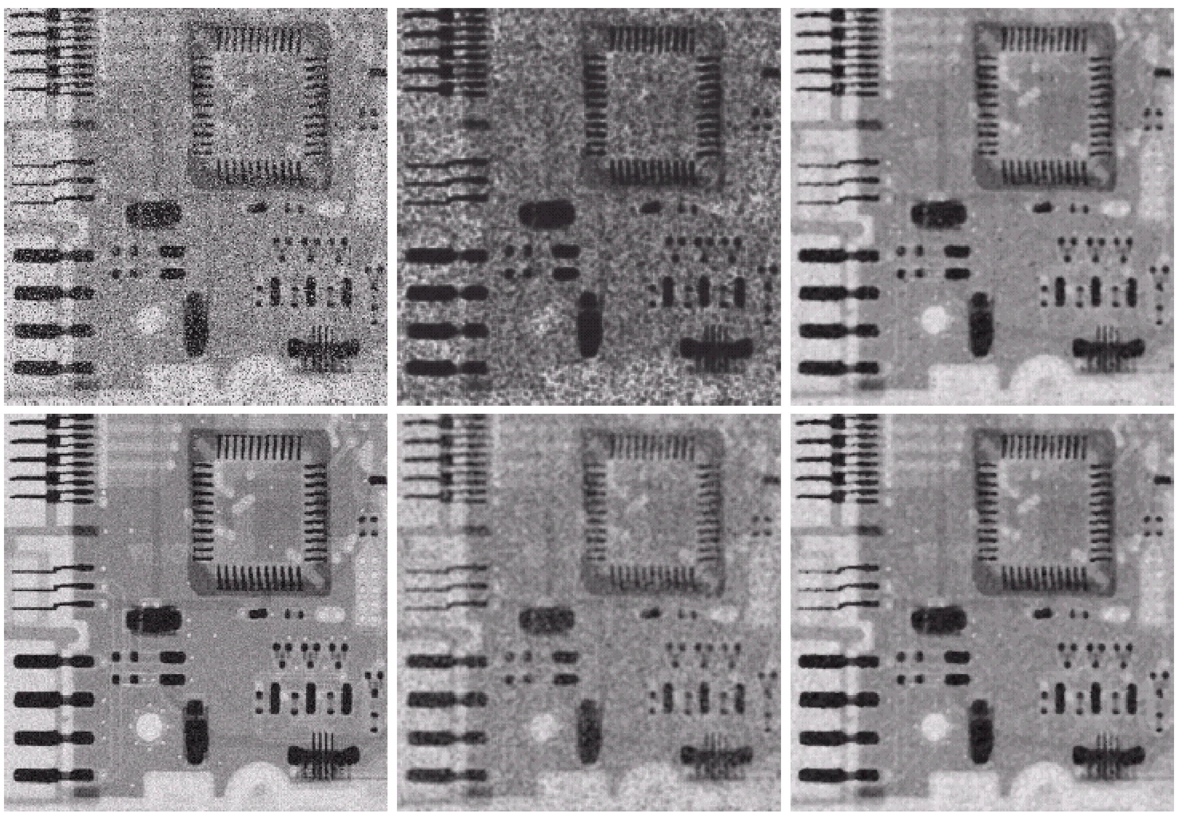
Alpha-trimmed mean Filter

- From the pixel values in a neighborhood S_{ab} of the pixel, we first delete (trim) the d/2 lowest and d/2 highest values. We then compute the arithmetic mean of the remaining (*ab*-*d*) values:

$$\hat{f}(m,n) = \frac{1}{ab-d} \sum_{(s,t)\in S_{ab}} g_r(s,t)$$

- When d=0, we get the regular arithmetic mean filter, wheread when $d = (ab - 1)/2$, we get the median filter.
- This filter is useful when there is multiple types of noise (for example: salt-and-pepper noise in addition to Gaussian noise).
- This filter also combines order statistic with averaging.

Example: **Alpha-trimmed mean Filter**



b	d	f
a	c	e

(a) Image corrupted by additive uniform noise; (b) Image additionally corrupted by additive salt-pepper noise. Image in (b) filtered by 5x5: (c) arithmetic mean fileter; (d) geometric mean fileter;(e) median mean fileter; (f) alpha-trimed mean fileter with d=5

Adaptive local noise reduction filter

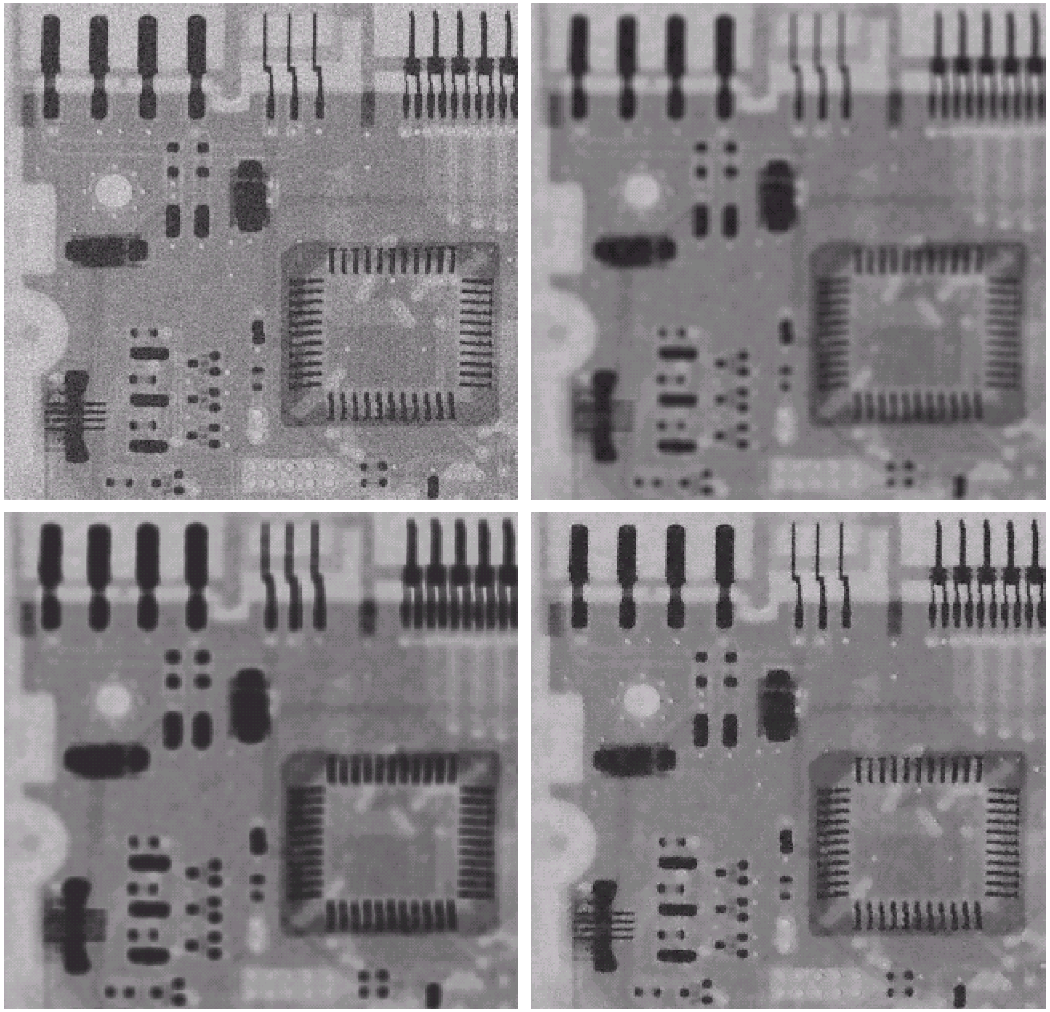
- Filter operation is not uniform at all pixel locations but depends on the local characteristics (local mean, local variance) of the observed image.
- Consider an observed image $g(m,n)$ and an $a \times b$ window S_{ab} . Let σ_η^2 be the noise variance and $m_L(m,n)$, $\sigma_L^2(m,n)$ be the local mean and variance of $g(m,n)$ over an $a \times b$ window around (m,n).
- The adaptive filter is given by:

$$\hat{f}(m,n) = g(m,n) - \frac{\sigma_\eta^2}{\sigma_L^2(m,n)}\left(g(m,n) - m_L(m,n)\right)$$

Adaptive local noise reduction filter

- Usually, we need to be careful about the possibility of $\sigma_L^2(m,n) < \sigma_\eta^2$, in which case, we could potentially get a negative output gray value.
- This filter does the following:
  - If $\sigma_\eta^2 = 0$ (or is small), the filter simply returns the value of $g(m,n)$.
  - If the local variance $\sigma_L^2(m,n)$ is high relative to the noise variance σ_η^2, the filter returns a value close to $g(m,n)$. This usually corresponds to a location associated with edges in the image.
  - If the two variances are roughly equal, the filter does a simple averaging over window S_{ab}.

Example: Adaptive local noise reduction filter

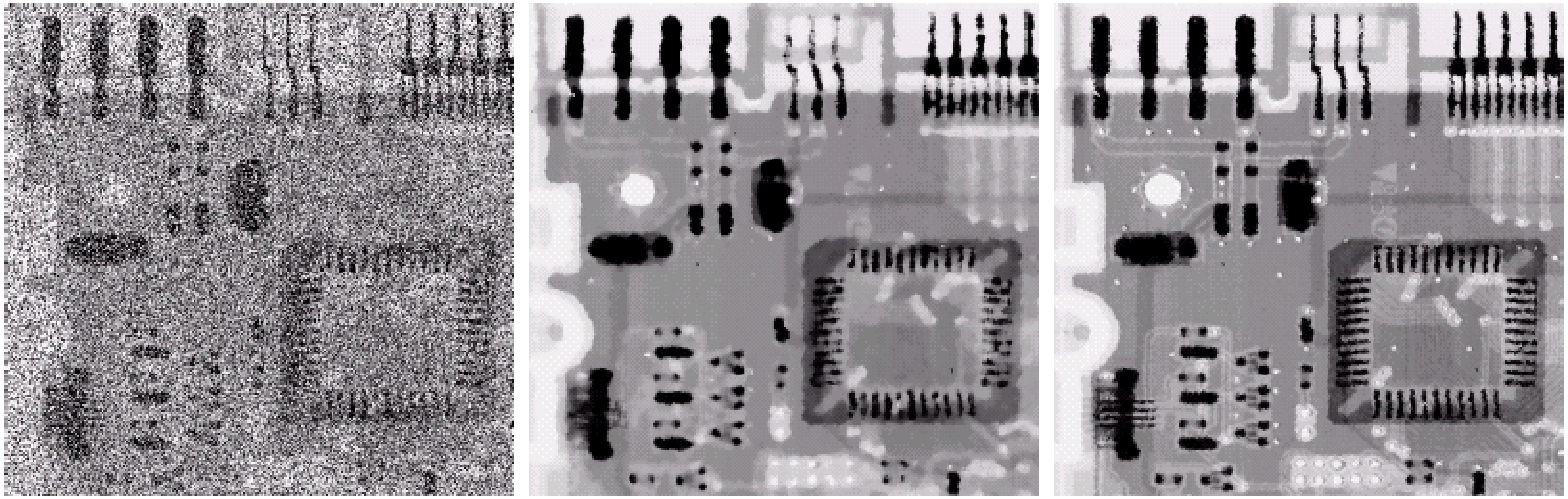


a b
c d

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

- Read from textbook Gonzalez (page 241-243).



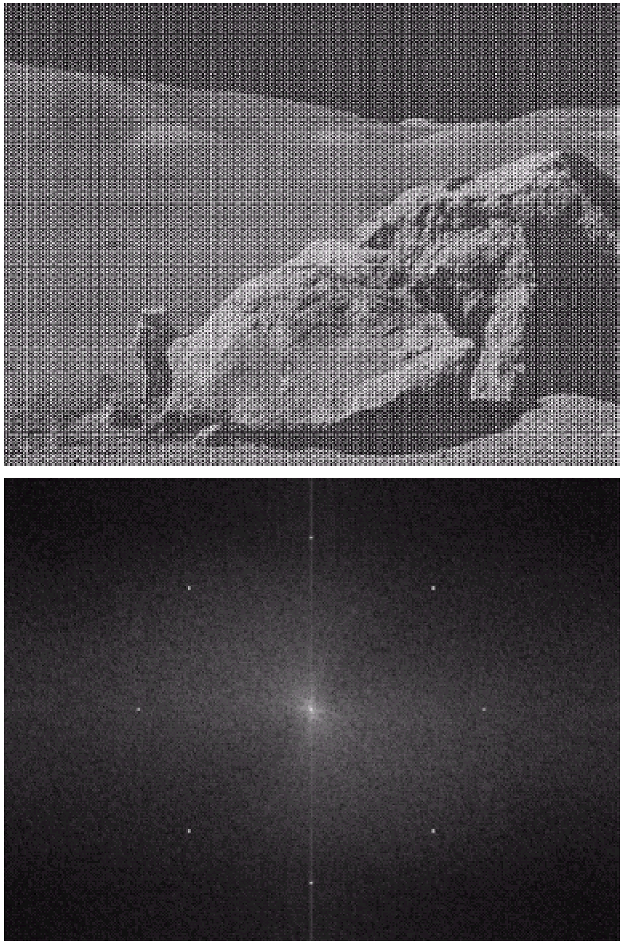
a b c

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 $S_{max} = 7$.

Periodic Interference/Noise

- Periodic noise or interference occurs in images due to electrical or electromechanical interference during image acquisition.
- It is an example of spatially dependent noise.
- This type of noise can be very effectively removed using frequency domain filtering. Recall that the spectrum of a pure sinusoid would be a simple impulse at the appropriate frequency location.

Periodic Interference/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.)

Bandreject filters

- Bandreject filters remove (or attenuate) a band of frequencies, around some frequency, say D_0 .
- An ideal bandreject filter is given by:

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

  where $D(u,v) = \sqrt{u^2 + v^2}$
- *W* is usually referred to the width of the (stop) band and D_0 as the center frequency.

Bandreject filters

- A Butterworth bandreject filter of order *n* is given by

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

- A Gaussian bandreject filter is given by

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

Bandreject filters

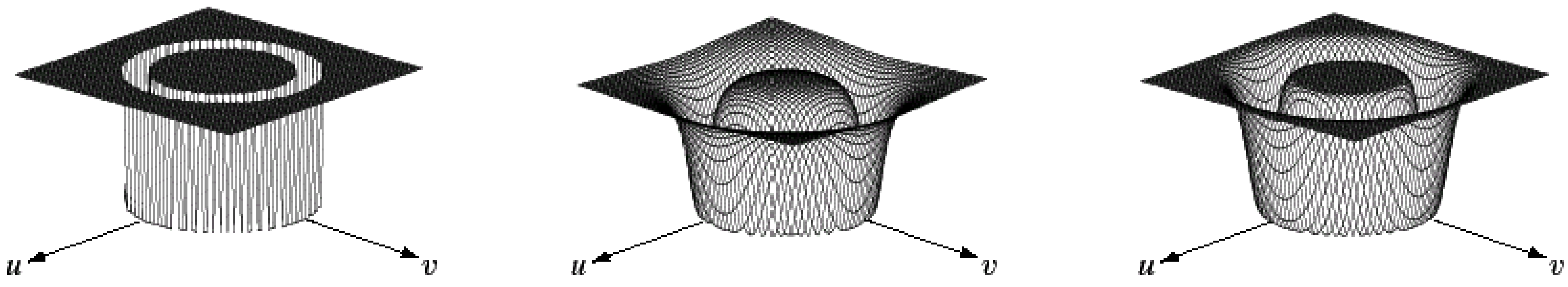


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Example: Bandreject filters

- Bandreject filters are ideally suited for filtering out periodic interference.
- Recall that the Fourier transform of a pure sine or cosine function is just a pair of impulses.
- Therefore the interference is “localized” in the spectral domain and one can easily identify this region and filter it out.

Example: Bandreject filters

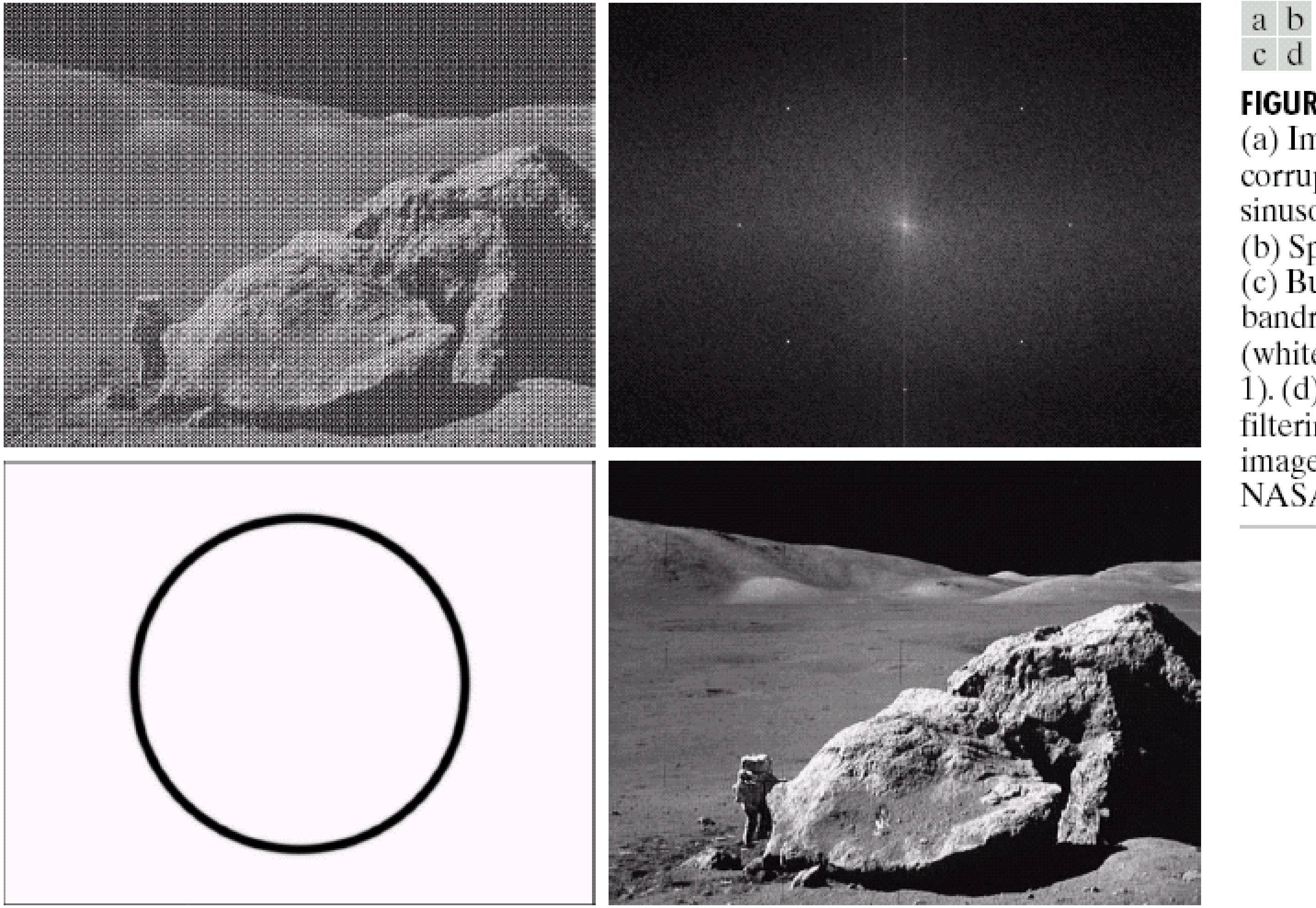


FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Bandpass filters

- Bandpass filters are the exact opposite of bandreject filters. They pass a band of frequencies, around some frequency, say D_0 (rejecting the rest).
- One can write:

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

- Bandpass filter is usually used to isolate components of an image that correspond to a band of frequencies.
- It can also be used to isolate noise interference, so that more detailed analysis of the interference can be performed, independent of the image.

Bandpass filters

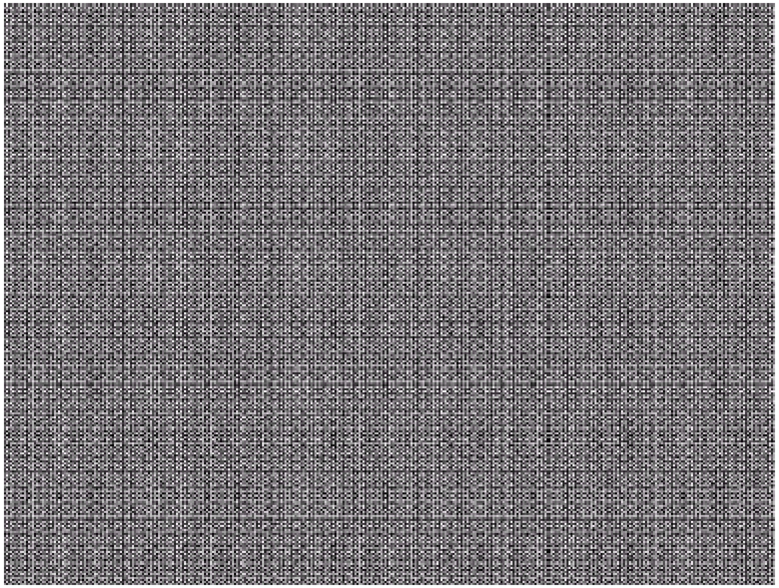


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

Notch filter

- It is a kind of bandreject/bandpass filter that rejects/passes a very narrow set of frequencies, around a center frequency.
- Due to symmetry considerations, the notches must occur in symmetric pairs about the origin of the frequency plane.
- The transfer function of an ideal notch-reject filter of radius D_0 with center frequency (u_0,v_0) is given by:

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

where

$$D_1(u,v)=\left[\left(u-M/2-u_0\right)^2+\left(v-N/2-v_0\right)^2\right]^{0.5}$$

and

$$D_2(u,v)=\left[\left(u-M/2+u_0\right)^2+\left(v-N/2+v_0\right)^2\right]^{0.5}$$

Notch filter

- The transfer function of a Butterworth notch reject filter of order *n* is given by

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

- A Gaussian notch reject filter is given by

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

- A notch pass filter can be obtained from a notch reject filter using:

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

Illustration of transfer function of notch filters

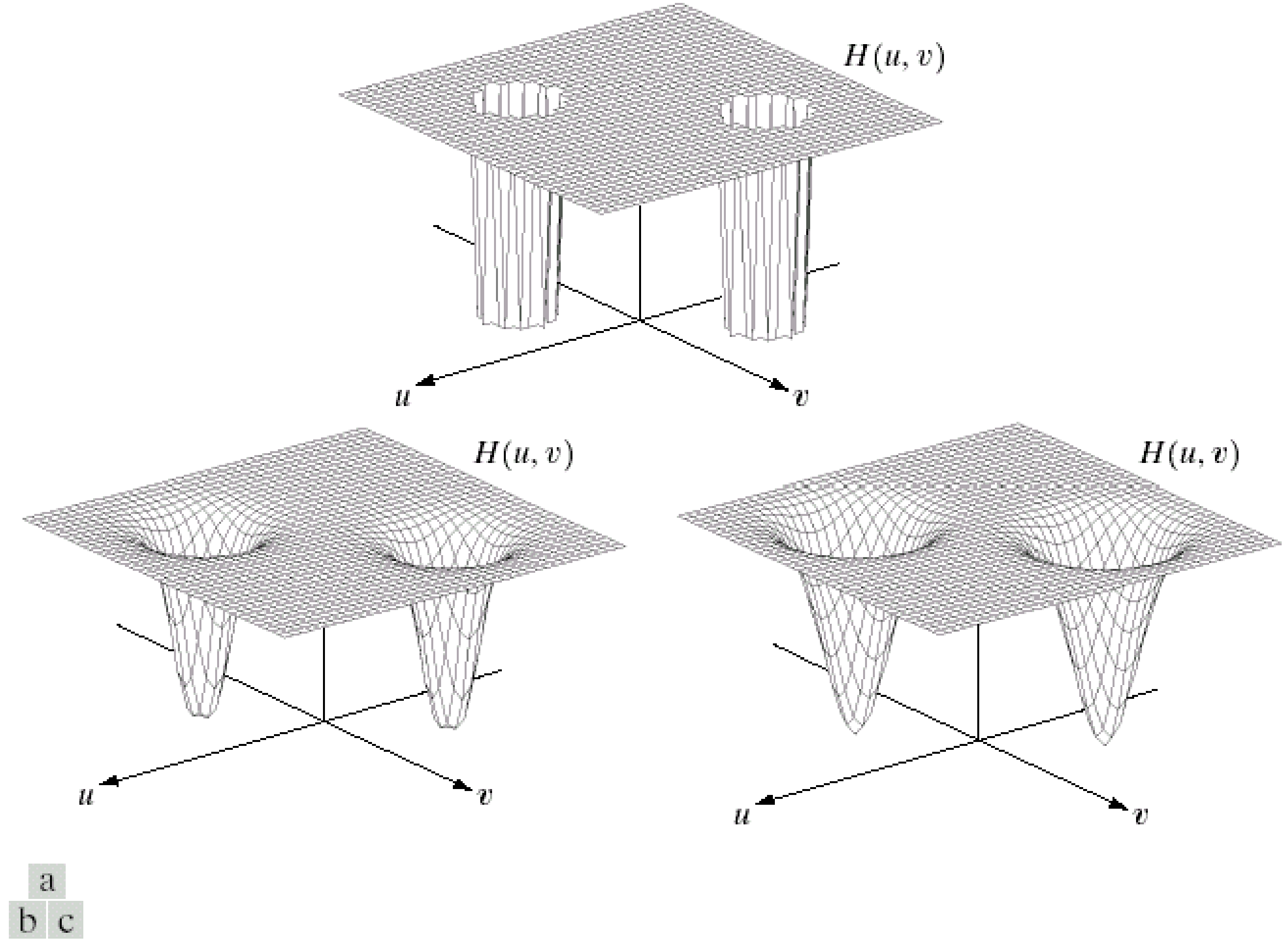
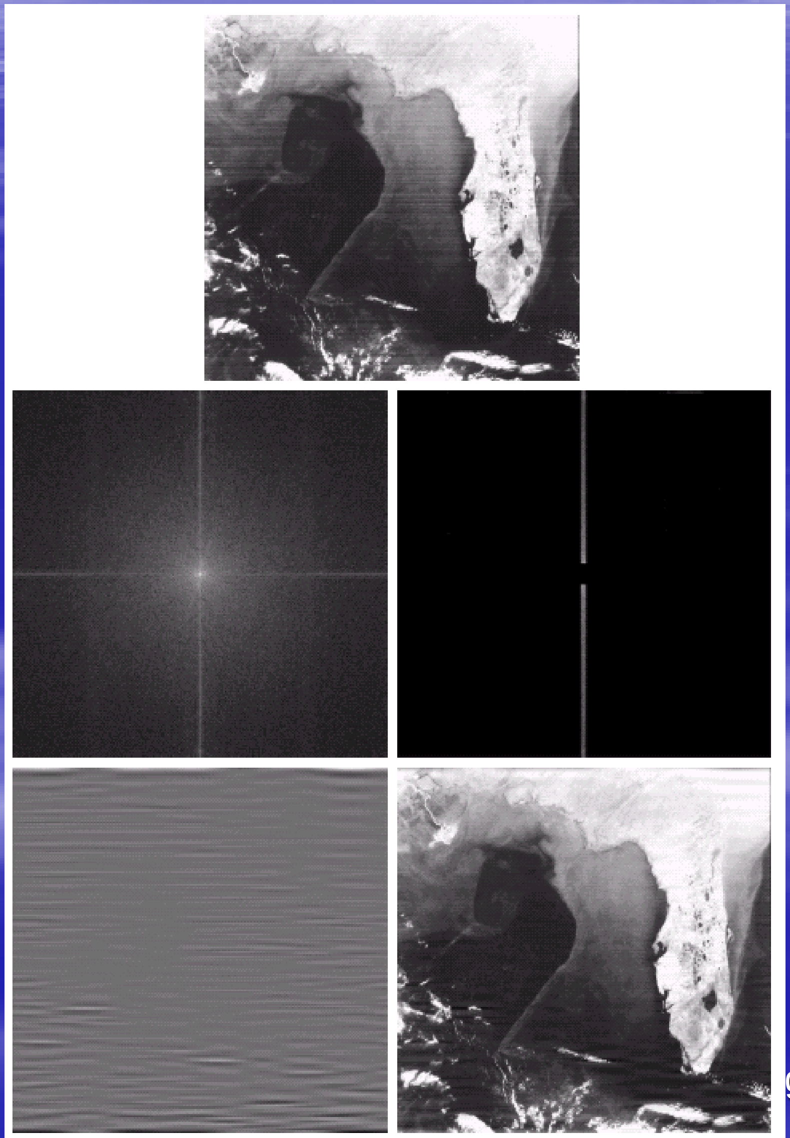


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Example: · Image corrupted by periodic horizontal scan lines.



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.)

Optimum Notch Filtering

- When interference patterns are more complicated, the preceding filters tend to reject more image information in an attempt to filter out the noise.
- In this case, we first filter out the noise interference using a notch pass filter:

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

$$\eta(m,n) = \mathscr{F}^{-1}\{N(u,v)\}$$

- The image $\eta(m,n)$ yields a rough estimate of the interference pattern.

Optimum Notch Filtering

- We can then subtract off a weighted portion of $\eta(m,n)$ from the image $g(m,n)$ to obtain our restored image:

$$\hat{f}(m,n) = g(m,n) - w(m,n)\eta(m,n)$$

- It is possible to design the weighting function or modulation function $w(m,n)$ in an optimal fashion. See section 5.4.4 (page 251,252) of textbook for details.

Linear, position-invariant degradation

- We will now consider the general degradation equation (see page 254, 255 of text for a derivation of this equation):

$$g(m,n) = h(m,n)*f(m,n)+\eta(m,n)$$

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

- This consists of a "blurring" function $h(m,n)$, in addition the random noise component $\eta(m,n)$.

Linear, position-invariant degradation

- The blurring function $h(m,n)$ is usually referred to as a point-spread function (PSF) and represents the observed image corresponding to imaging an impulse or point source of light.
- In this case, we need to have a good knowledge of the PSF $h(m,n)$, in addition to knowledge of the noise statistics. This can be done in practice using one of the following methods:

Using Image observation

Using Image observation

- Identify portions of the observed image (subimage) that are relatively noise-free and which corresponds to some simple structures.
- We can then obtain

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

  where $G_s(u,v)$ is the spectrum of the observed subimage, $\hat{F}_s(u,v)$ is our estimate of the spectrum of the original image (based on the simple structure that the subimage represents).
- Based on the characteristic of the function $H_s(u,v)$, once can rescale to obtain the overall PSF $H(u,v)$.

Experimentation

- If feasible, image a known object, usually a point source of light, using the given imaging equipment and setup.
- If A is the intensity of light source and $G(u,v)$ is the observed spectrum, we have

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

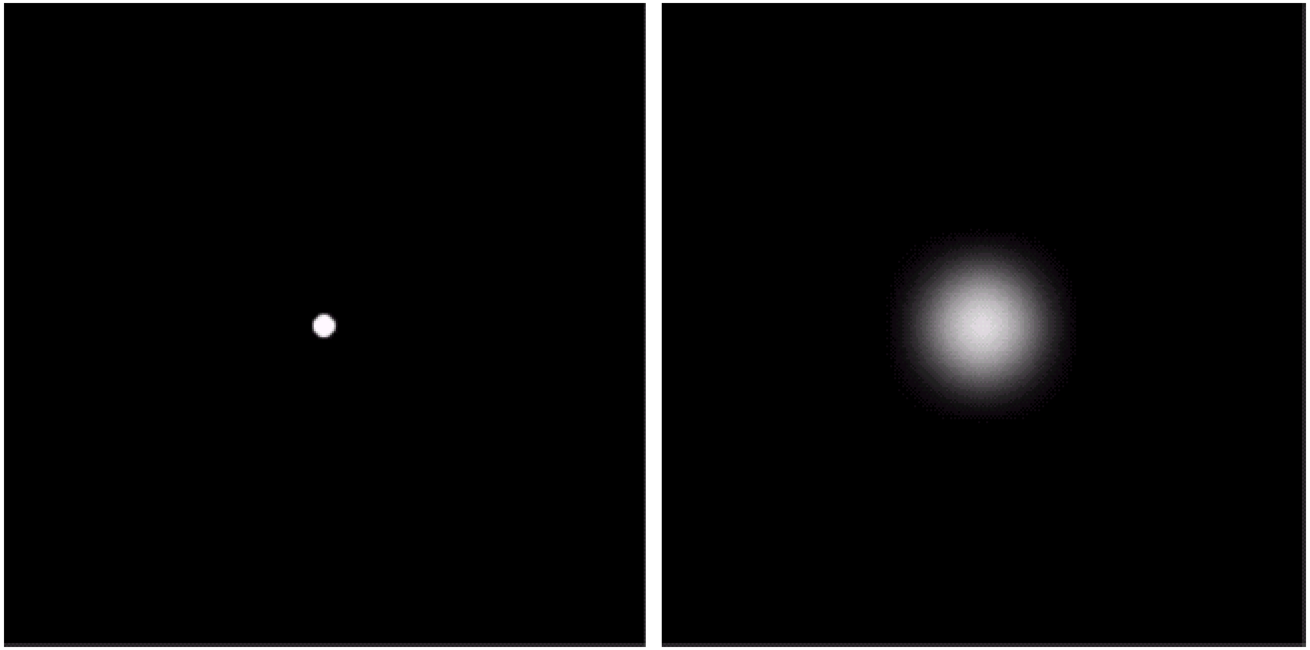


FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

Modeling

- A physical model is often used to obtain the PSF.
- Blurring due to atmospheric turbulence can be modeled by the transfer function:

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

  where *k* is a constant that depends on the nature of the turbulence.
- Note that this is similar to a Gaussian lowpass filter.
- Gaussian lowpass filter is also often used to model mild uniform blurring.

Modeling

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.)

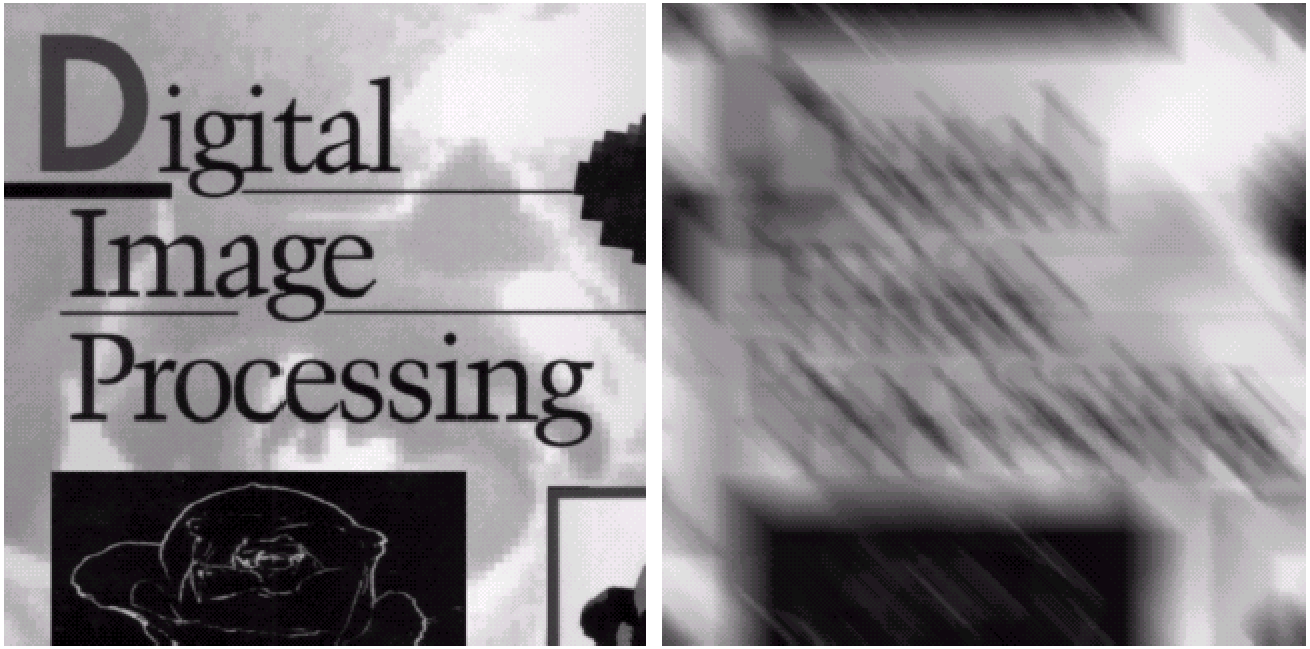
Modeling

- Precise mathematical modeling of the blurring process is sometime used. For example, blurring due to uniform motion is modeled as:

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

where *T* is the duration of exposure and *a* and *b* are the displacements in the *x*- and *y*-directions, respectively, during this time *T*.

Example



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 $T = 1$.

Inverse Filter

- The simplest approach to restoration is direct inverse filtering. This is obtained as follows:

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

$$\Rightarrow \hat{F}(u,v) = R(u,v)G(u,v), \quad u,v = 0,1,\cdots,N-1,$$

where

$$R(u,v) = \frac{1}{H(u,v)}$$

Inverse Filter

- We can rewrite this in the spatial domain as follows:

$$\hat{f}(m,n) = g(m,n) * r(m,n) = \mathrm{IDFT}\left\{\frac{G(u,v)}{H(u,v)}\right\}$$

- In practice, we actually use a slightly modified filter:

$$R(u,v) = \begin{cases} \dfrac{1}{H(u,v)}, & |H(u,v)| > \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

where ε is a small value. This avoids numerical problems when $|H(u,v)|$ is small.

Inverse Filter

- The inverse filter works fine provided there is no noise. This is illustrated in the following example.
- Let us now analyze the performance of the inverse filter in the presence of noise. Indeed, in this case:

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

which gives

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

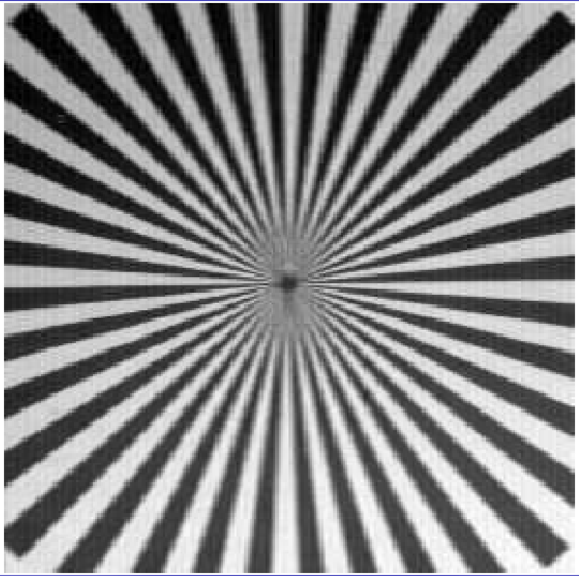
Inverse Filter

- Hence noise actually gets amplified at frequencies where $|H(u,v)|$ is zero or very small. In fact, the contribution from the noise term dominates at these frequencies.
- As illustrated by an example, the inverse filter fails miserably in the presence of noise. It is therefore, seldom used in practice, in the presence of noise.

Inverse Filtering example (no noise)

$$h = \frac{1}{N^2}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{N\times N}$$

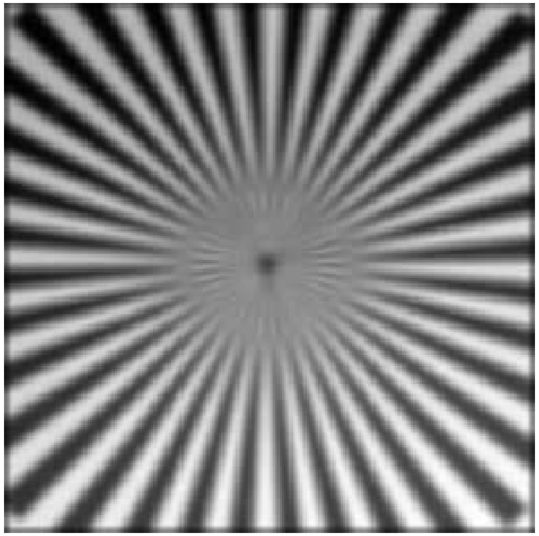
$$\varepsilon = 0.001$$



$f(m,n)$

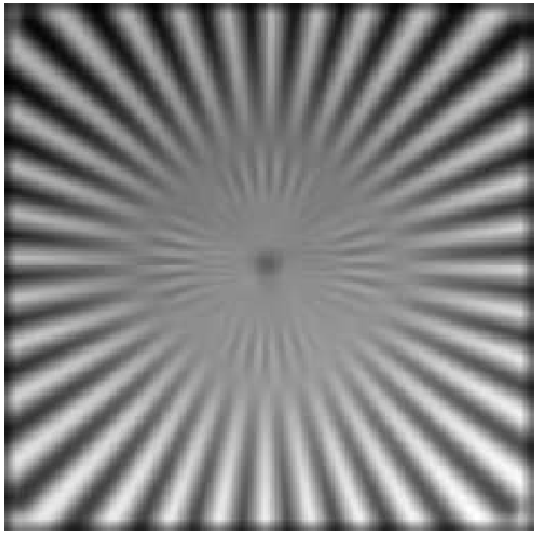
Inverse Filtering example (no noise)

$N = 7$



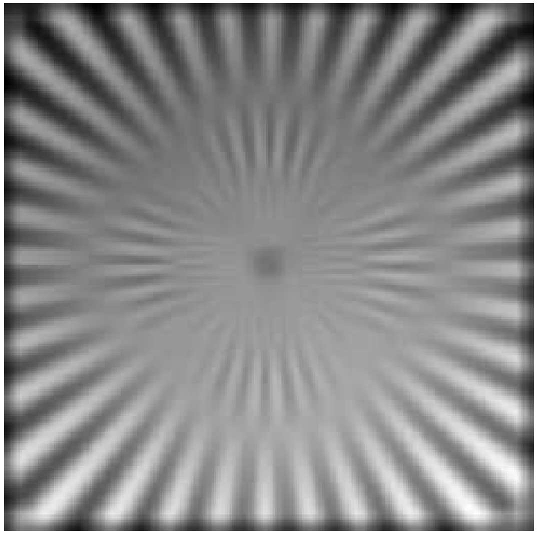
$\text{MSE} = 0.014$

$N = 11$



$\text{MSE} = 0.03$

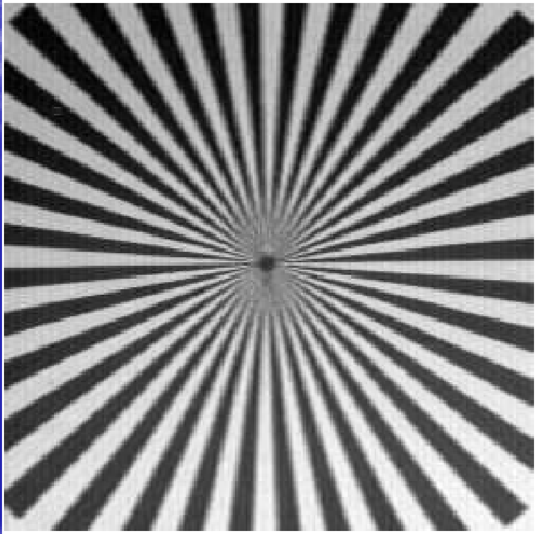
$N = 15$



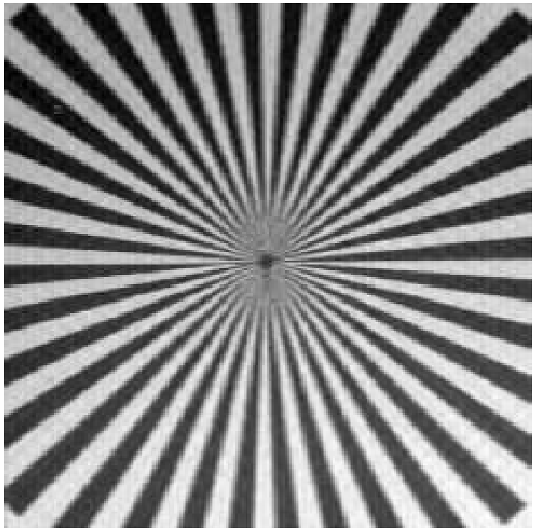
$\text{MSE} = 0.05$

g(m,n)

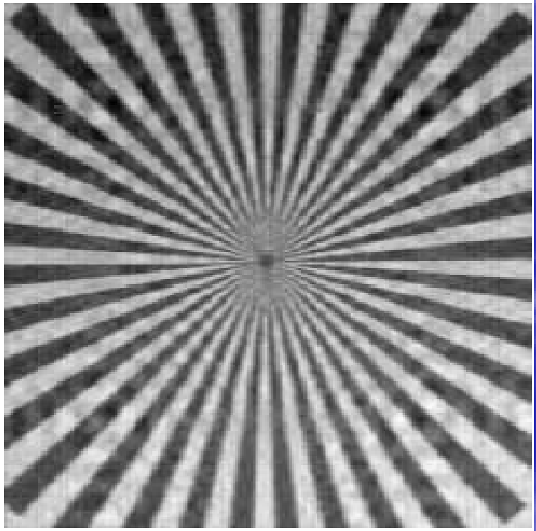
Inverse Filtering example (no noise)



$\mathrm{MSE} = 3.6\times10^{-5}$



$\mathrm{MSE} = 2.3\times10^{-4}$



$\mathrm{MSE} = 0.0029$

$$\hat{f}(m,n)$$

Inverse Filtering example (no noise)

$$H(u,v) = \frac{1}{1 + \left[\frac{\sqrt{u^2 + v^2}}{r_0} \right]^2}$$

$$\varepsilon = 0.001$$

f(m,n)

Inverse Filtering example (no noise)

$g(m,n)$

MSE = 0.02

$r_0 = 11$

$\hat{f}(m,n)$

MSE = 0.008

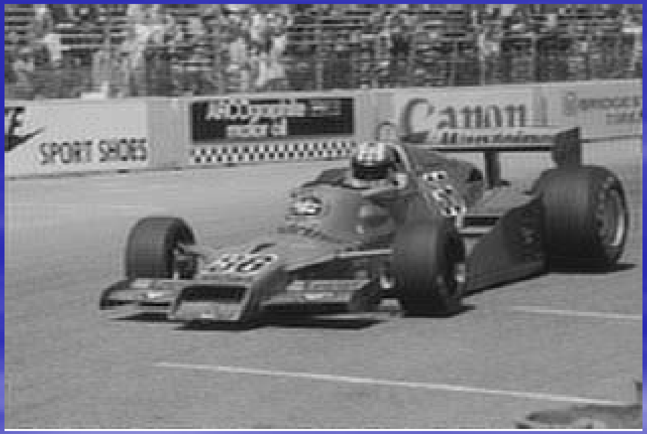
Inverse Filtering example (no noise)

$g(m,n)$

MSE = 0.017

$r_0 = 15$

$\hat{f}(m,n)$



MSE = 0.005

Inverse Filtering example (no noise)

$g(m,n)$

$\hat{f}(m,n)$

$r_0 = 23$

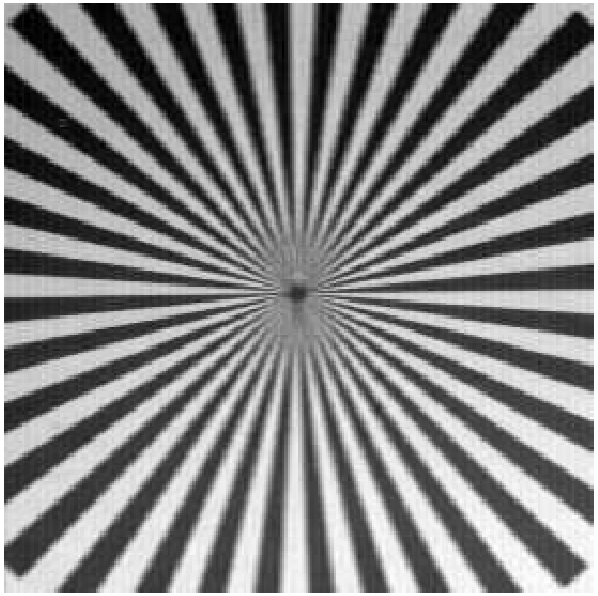
MSE = 0.013

MSE = 0.0016

Inverse Filtering example (with noise)

$$h=\frac{1}{25}\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix}_{5\times5}$$

$$\varepsilon=0.01$$

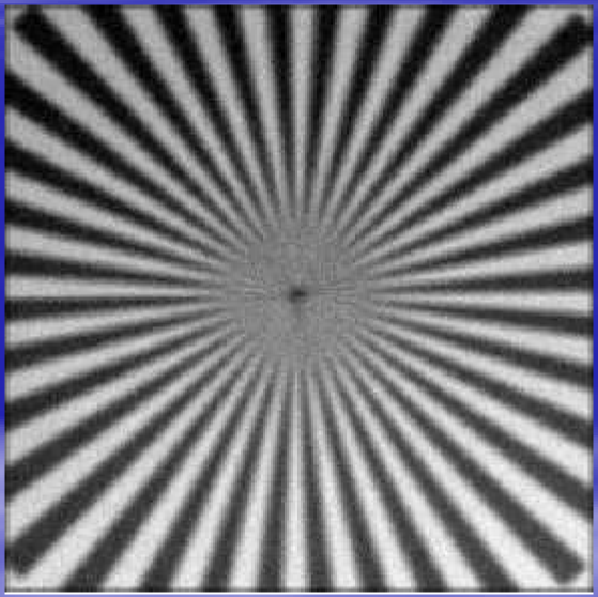
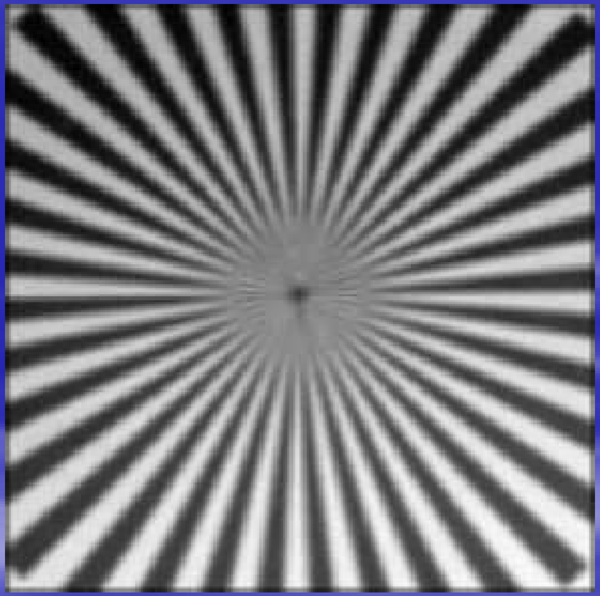
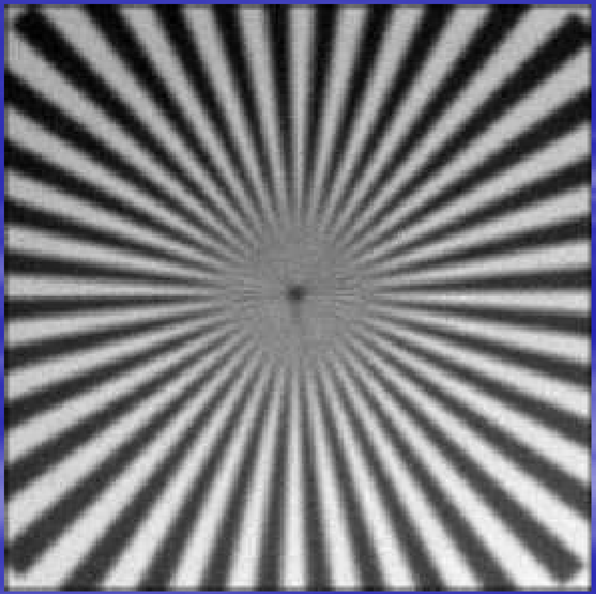


f(m,n)

We will add the Zero-mean Gaussian noise with variance σ^2

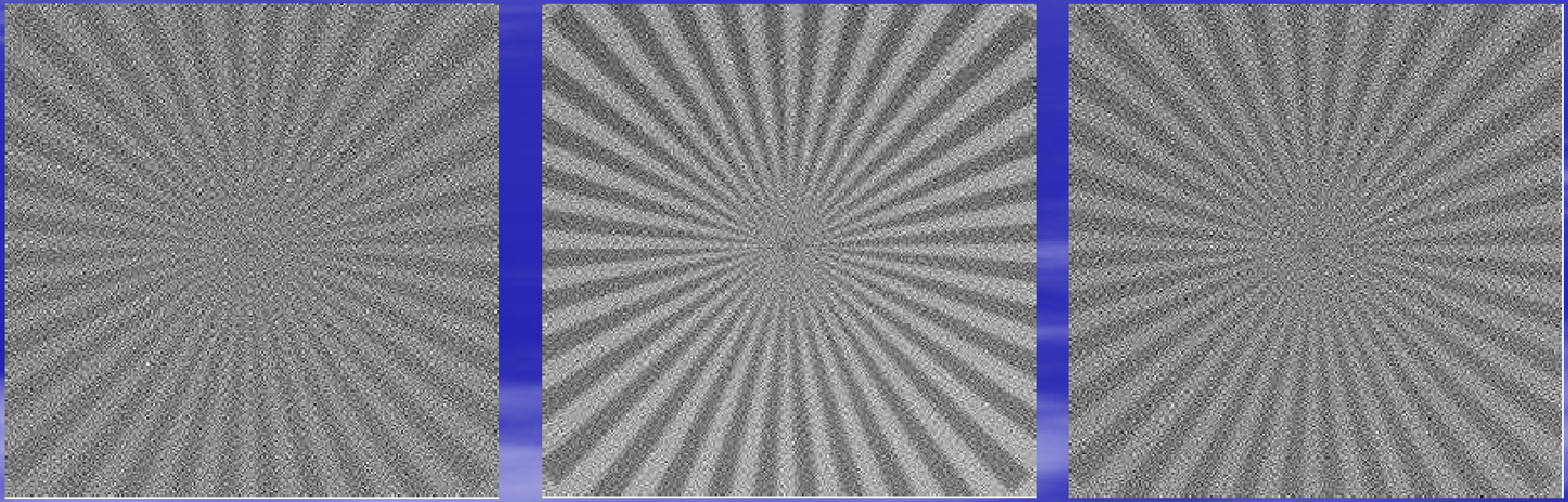
Inverse Filtering example (with noise)

g(m,n)

$\sigma = 0.03$	$\sigma = 0.01$	$\sigma = 0.02$
		
MSE = 0.008	MSE = 0.007	MSE = 0.0075

Inverse Filtering example (with noise)

$$\hat{f}(m,n)$$



MSE = 0.09 | MSE = 0.09 | MSE = 0.047