Module-2
Digital Image Processing

Image Transformation

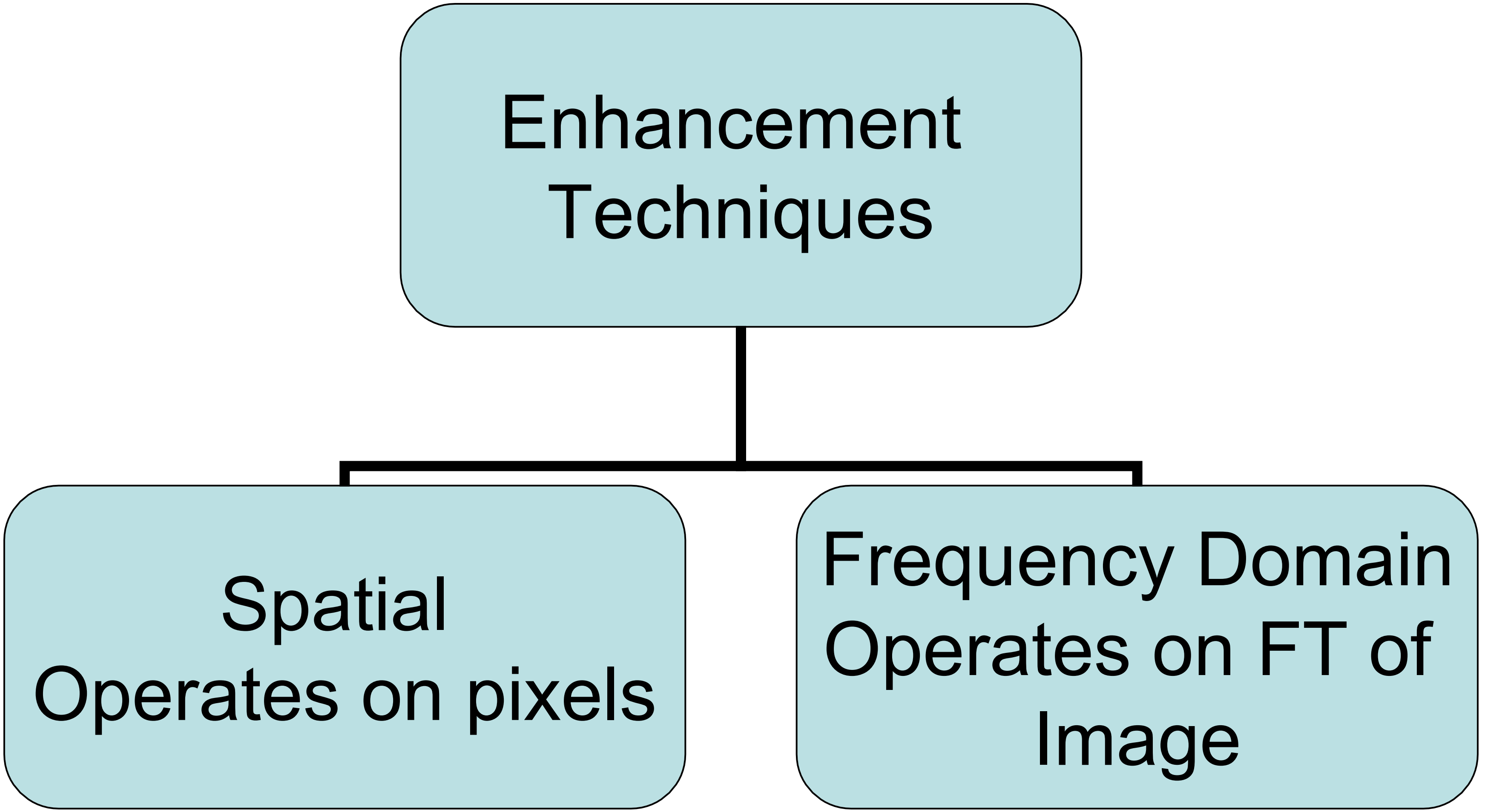

Enhancement Techniques
Spatial
Operates on pixels
Frequency Domain
Operates on FT of Image

Image Enhancement Definition

- **Image Enhancement:** is the process that improves the quality of the image for a specific application

Image Enhancement Methods

- **Spatial Domain Methods (Image Plane)**

  Techniques are based on direct manipulation of pixels in an image

- **Frequency Domain Methods**

  Techniques are based on modifying the Fourier transform of the image.

- **Combination Methods**

  There are some enhancement techniques based on various combinations of methods from the first two categories

Spatial Domain Methods

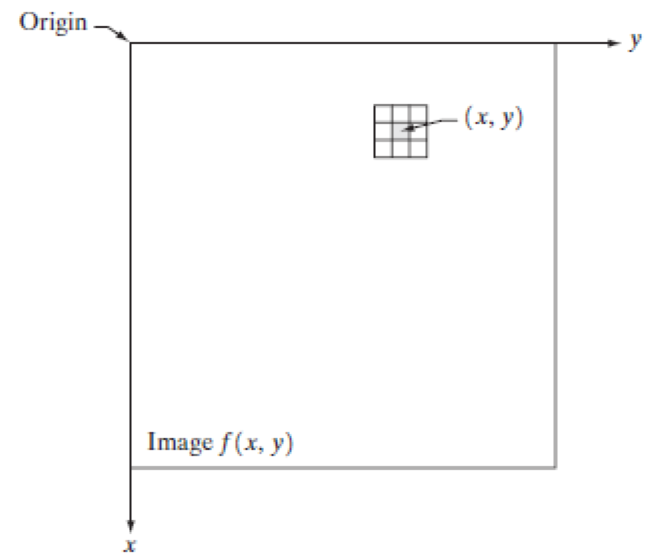
- As indicated previously, the term *spatial domain* refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

  g(x,y) = ***T*** [f(x,y)]

  Where f(x,y) in the input image, g(x,y) is the processed image and ***T*** is as operator on f, defined over some neighborhood of (x,y)

- In addition, ***T*** can operate on a set of input images.

FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.



- The simplest form of ***T***, is when the neighborhood of size 1X1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

$$s = T\,(r)$$

Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

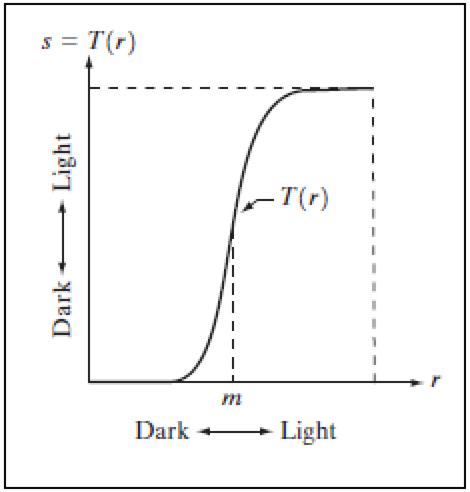
Examples of Enhancement Techniques

- **Contrast Stretching**:

  If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

  Produce higher contrast than the original image, by:

  - Darkening the levels below m in the original image
  - Brightening the levels above m in the original image

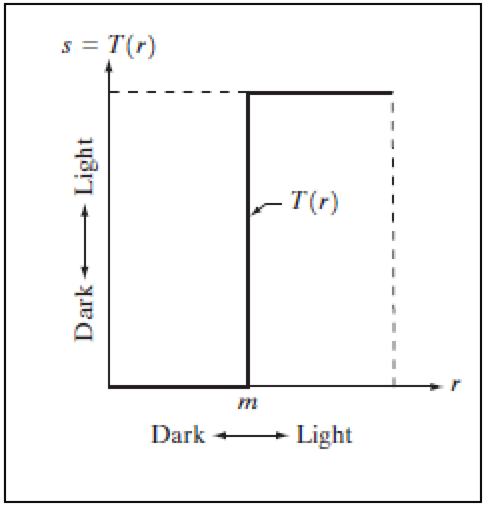


  So, Contrast Stretching: is a simple image enhancement technique that improves the contrast in an image by 'stretching' the range of intensity values it contains to span a desired range of values. Typically, it uses a linear function

Examples of Enhancement Techniques

- **Thresholding**

  Is a limited case of contrast stretching, it produces a two-level (binary) image.



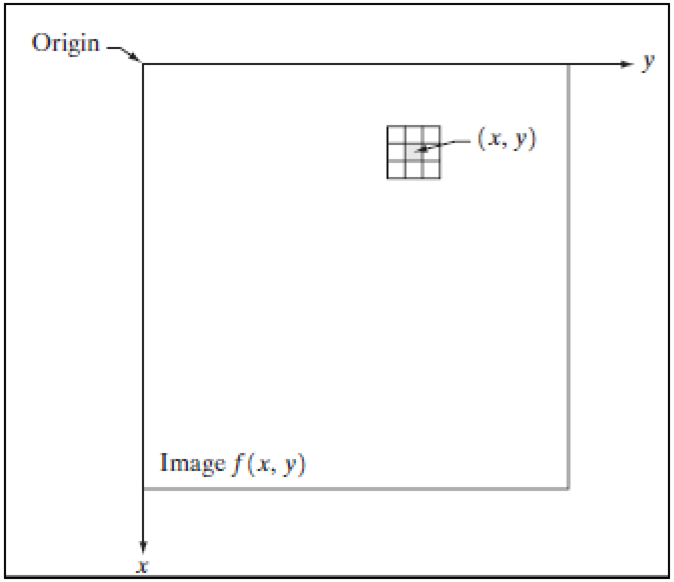
Some fairly simple, yet powerful, processing approaches can be formulated with grey-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as *point processing*.

Examples of Enhancement Techniques

Larger neighborhoods allow considerable more flexibility. The general approach is to use a function of the values of *f* in a predefined neighborhood of (x,y) to determine the value of *g* at (x,y).

One of the principal approaches in this formulation is based on the use of so-called *masks* (also referred to as *filters*)

So, a mask/filter: is a small (say 3X3) 2-D array, such as the one shown in the figure, in which the values of the mask coefficients determine the nature of the process, such as *image sharpening*. Enhancement techniques based on this type of approach often are referred to as *mask processing* or *filtering*.



Some Basic Intensity (Gray-level) Transformation Functions

- Grey-level transformation functions (also called, intensity functions), are considered the simplest of all image enhancement techniques.
- The value of pixels, before an after processing, will be denoted by *r* and *s*, respectively. These values are related by the expression of the form:

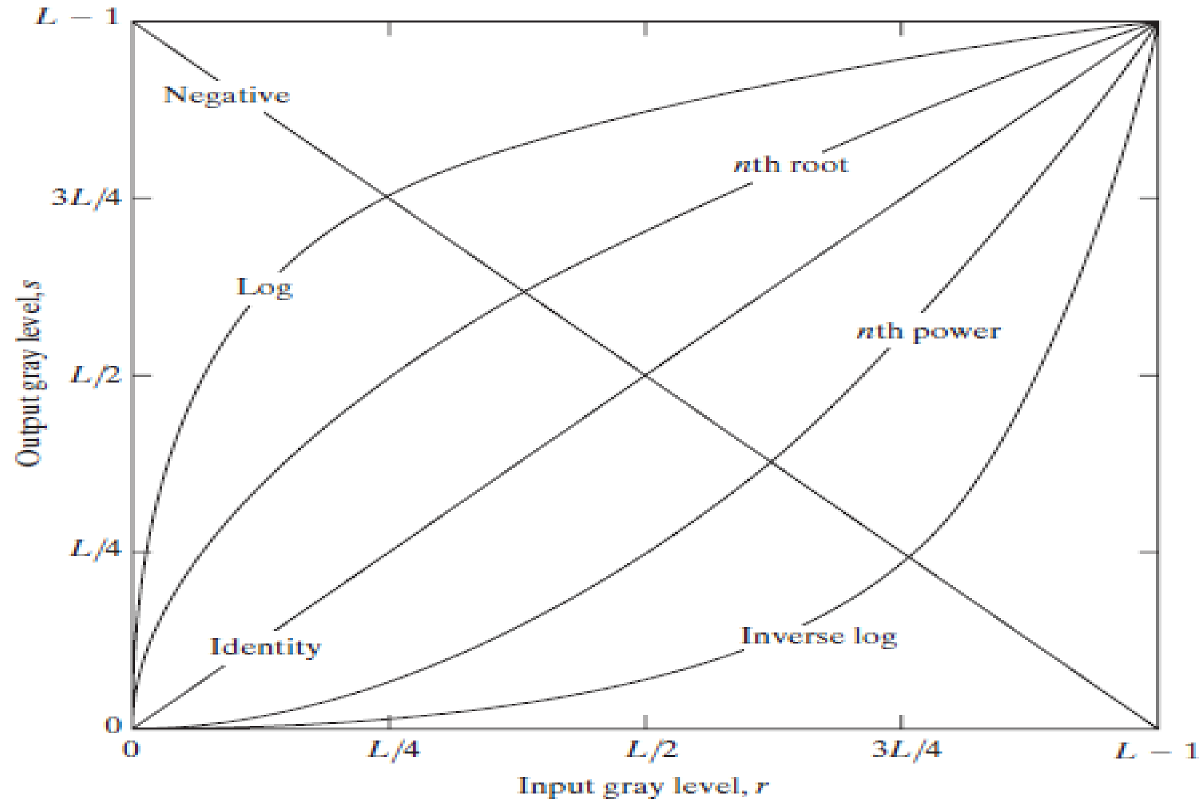
  $$\mathbf{s = \mathit{T}\,(r)}$$

  where T is a transformation that maps a pixel value r into a pixel value s.

Some Basic Intensity (Gray-level) Transformation Functions

Consider the following figure, which shows three basic types of functions used frequently for image enhancement:

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Some Basic Intensity (Gray-level) Transformation Functions

- The three basic types of functions used frequently for image enhancement:
  - Linear Functions:
    - Negative Transformation
    - Identity Transformation
  - Logarithmic Functions:
    - Log Transformation
    - Inverse-log Transformation
  - Power-Law Functions:
    - n^{th} power transformation
    - n^{th} root transformation

Linear Functions

- **Identity Function**
  - Output intensities are identical to input intensities
  - This function doesn’t have an effect on an image, it was included in the graph only for completeness
  - Its expression:

    $s = r$

Linear Functions

- **Image Negatives (Negative Transformation)**
  - The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

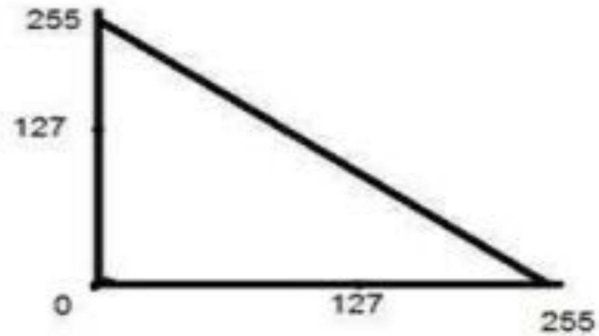
    $$s = L - 1 - r$$

    Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative.

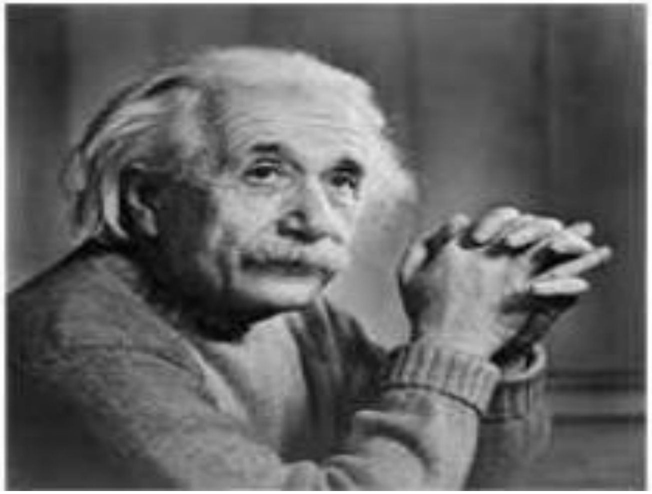
  - The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

NEGATIVE TRANSFORMATION EXAMPLE

Graph representation

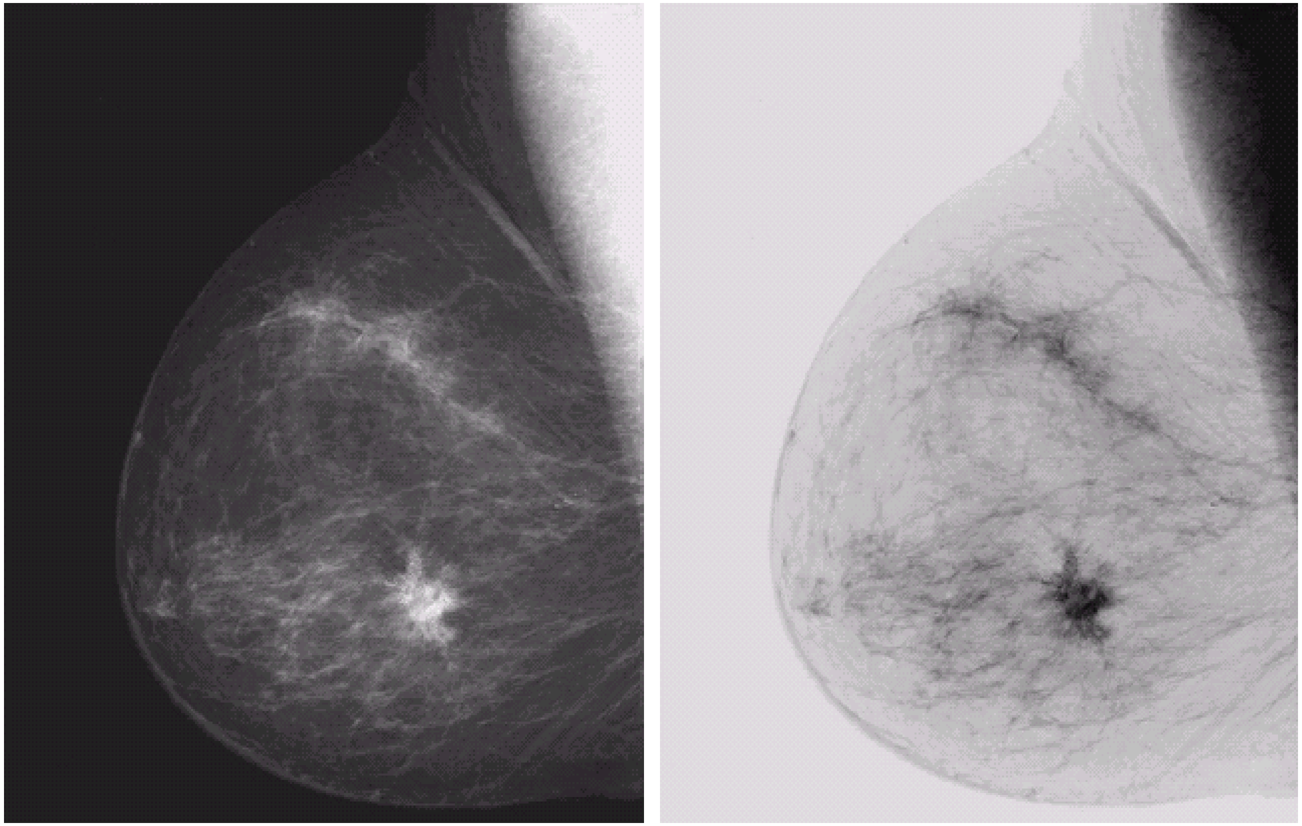


Input image



Output image

Image Negative



a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Image Negative: $s = L - 1 - r$

Logarithmic Transformations

- **Log Transformation**

  The general form of the log transformation:

  $$s = c \log (1+r)$$

  Where c is a constant, and $r \geq 0$

  – Log curve maps a narrow range of low gray-level values in the input image into a wider range of the output levels.

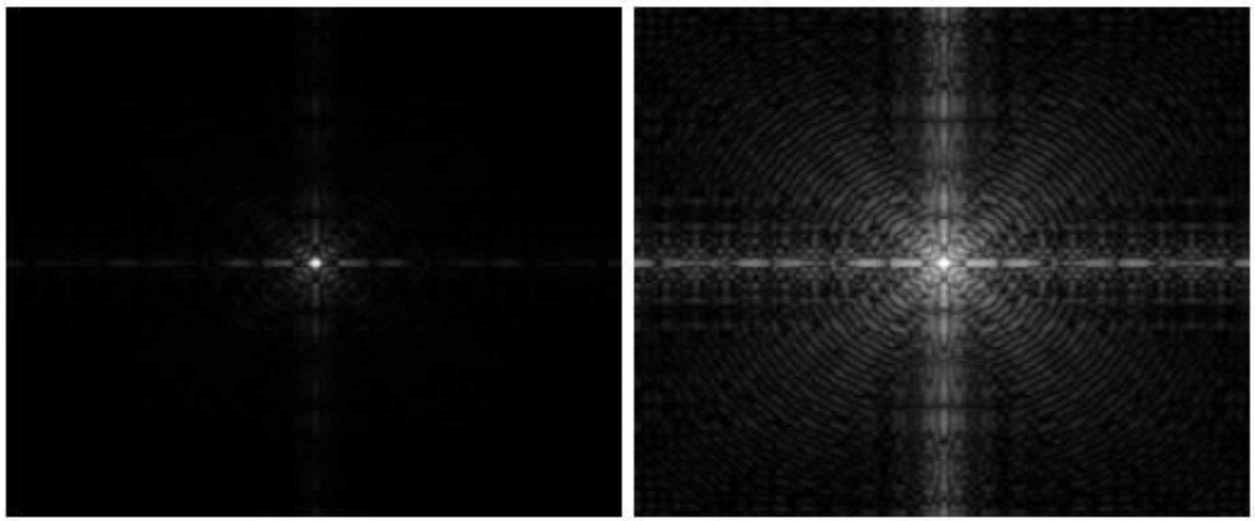
  – Used to expand the values of dark pixels in an image while compressing the higher-level values.

  – It compresses the dynamic range of images with large variations in pixel values.

Logarithmic Transformations

a b

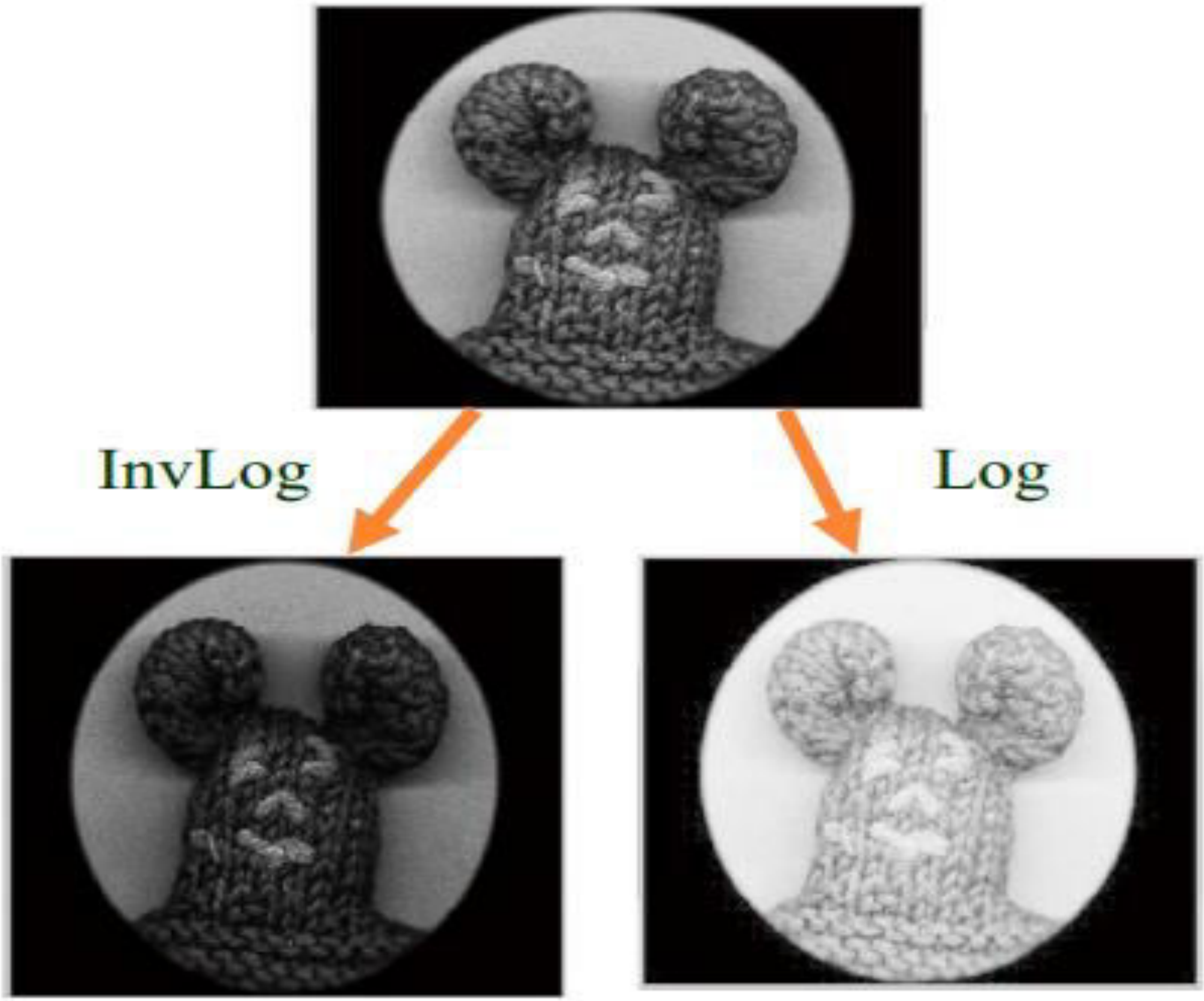
FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Logarithmic Transformations

- **Inverse Logarithm Transformation**
  - Do opposite to the log transformations
  - Used to expand the values of high pixels in an image while compressing the darker-level values.

Log transformation example



Power-Law Transformations

- Power-law transformations have the basic form of:

  $$s = c.r^{\gamma}$$

  Where c and γ are positive constants

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)

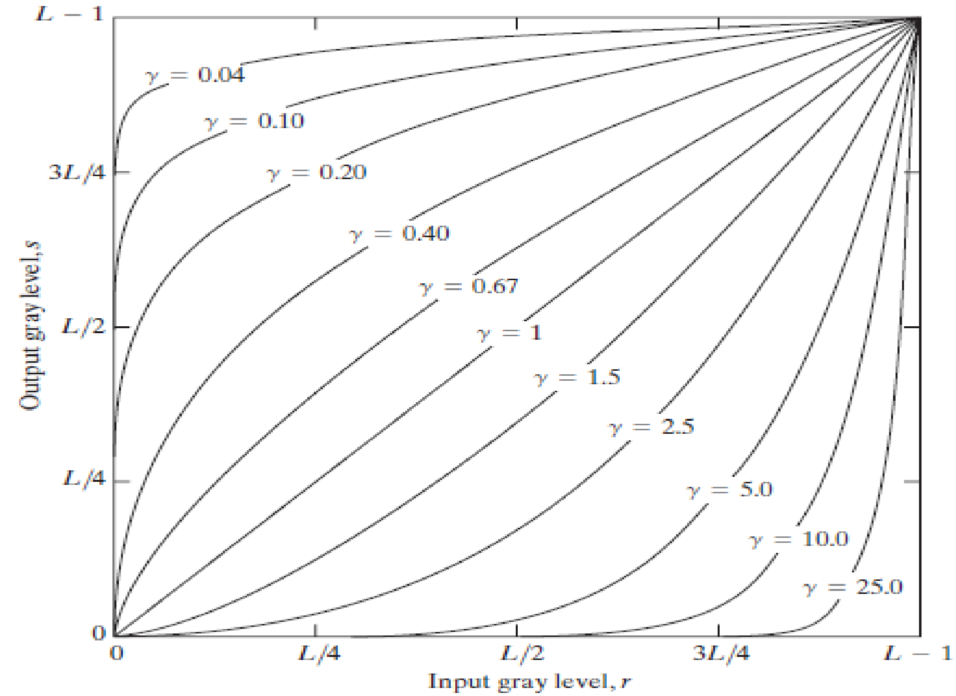


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

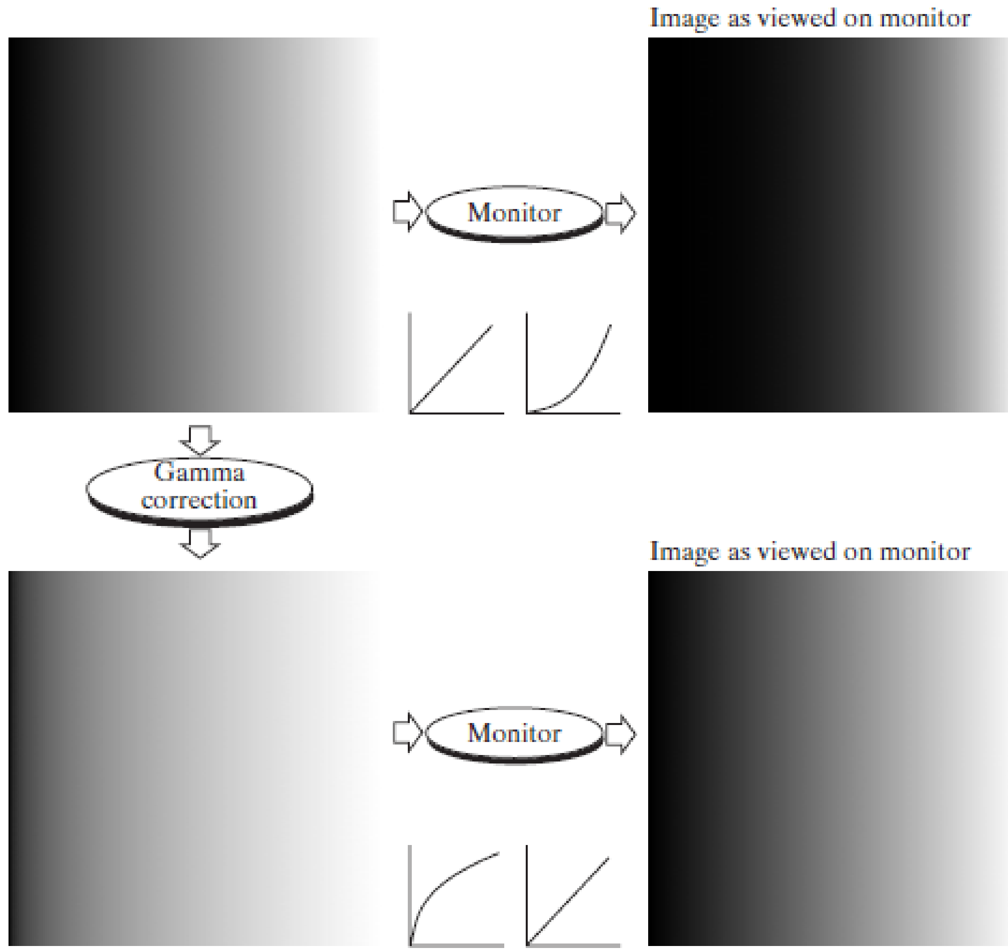
Power-Law Transformations

- Variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power-law response phenomena is called ***gamma correction**.*

  For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple gray-scale linear wedge input into a CRT monitor. As expected, the output of the monitor appears darker than the input, as shown in Fig. 3.7(b). Gamma correction

  in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation. The result is shown in Fig. 3.7(c).When input into the same monitor, this gamma-corrected input produces an output that is close in appearance to the original image, as shown in Fig. 3.7(d).

Power-Law Transformation

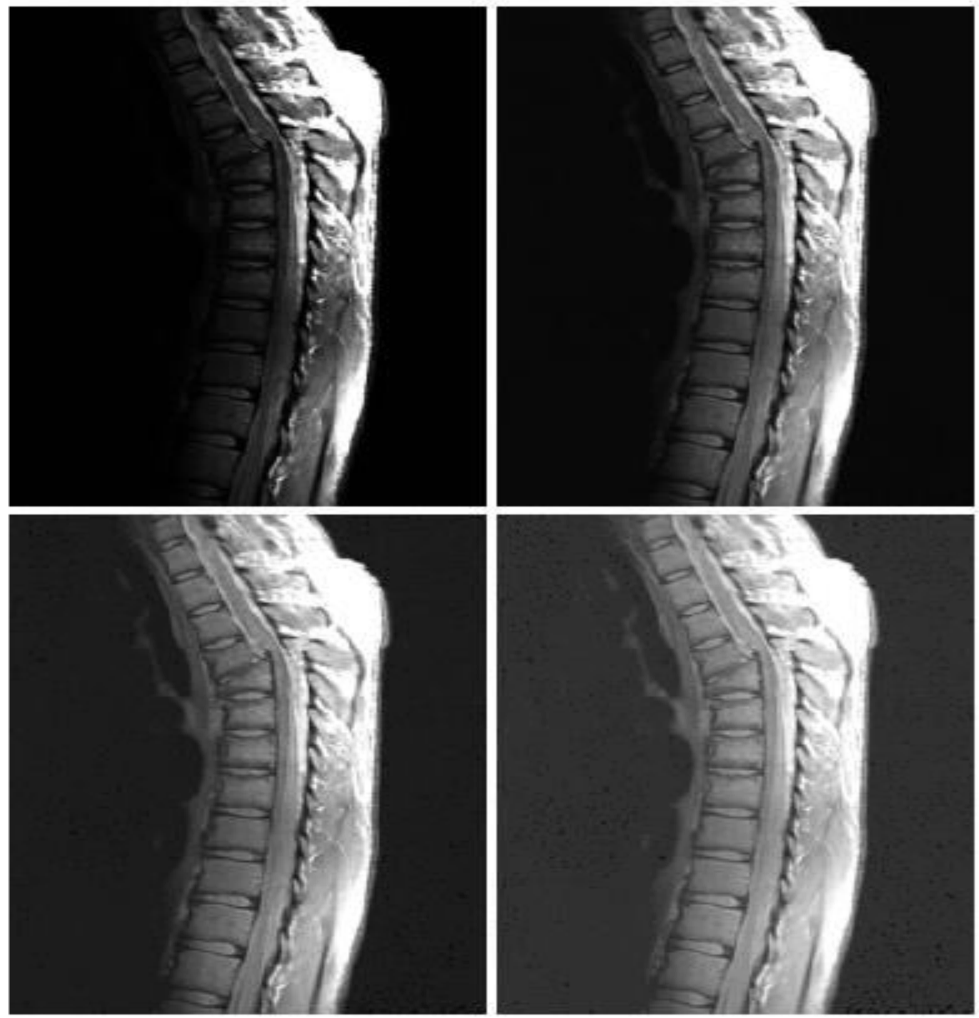


a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power-Law Transformation

- In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation. See figure 3.8



a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Power-Law Transformation

- Another illustration of Power-law transformation

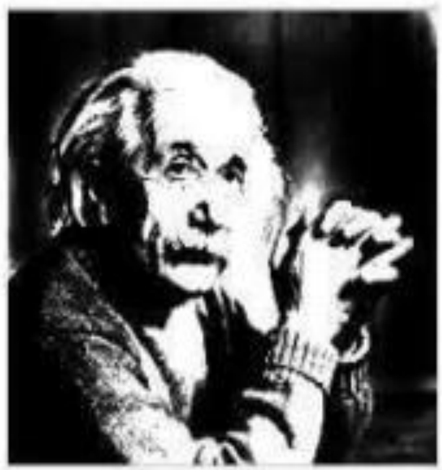
a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

POWER LAW TRANSFORMATION EXAMPLE

Gamma=10

Gamma=8



Gamma=6

Piecewise-Linear Transformation Functions

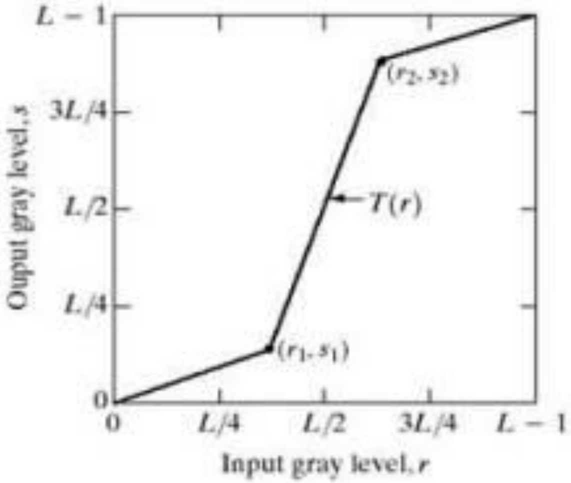
- **Principle Advantage:** Some important transformations can be formulated only as a piecewise function.
- **Principle Disadvantage:** Their specification requires more user input that previous transformations
- **Types of Piecewise transformations are:**
  – Contrast Stretching
  – Gray-level Slicing
  – Bit-plane slicing

Contrast Stretching

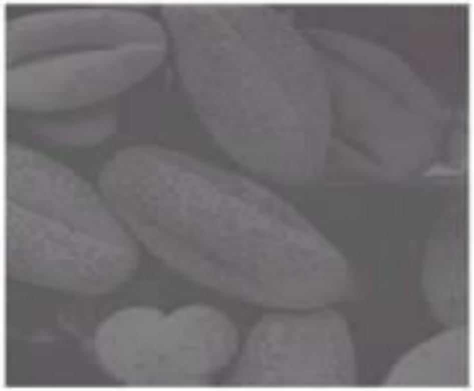
- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

CONTRAST STRETCHING EXAMPLE

Transformation function

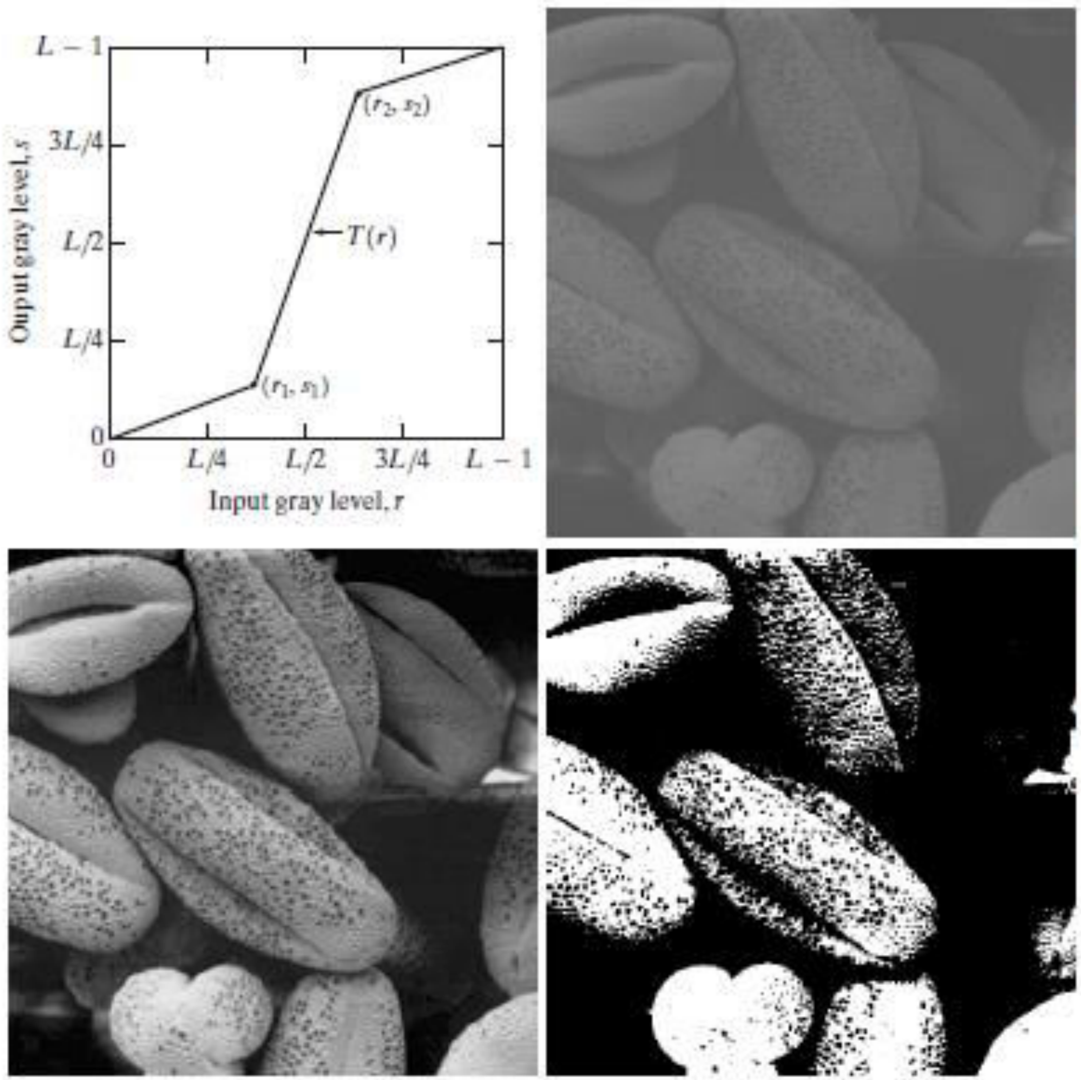


Low contrast image



Contrast stretching image

Contrast Stretching



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast Stretching

- Figure 3.10(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.
- If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels.
- If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a *thresholding function* that creates a binary image. As shown previously in slide 7.
- Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, r1 ≤ r2 and s1 ≤ s2 is assumed, so the function is always increasing.

Contrast Stretching

- Figure 3.10(b) shows an 8-bit image with low contrast.

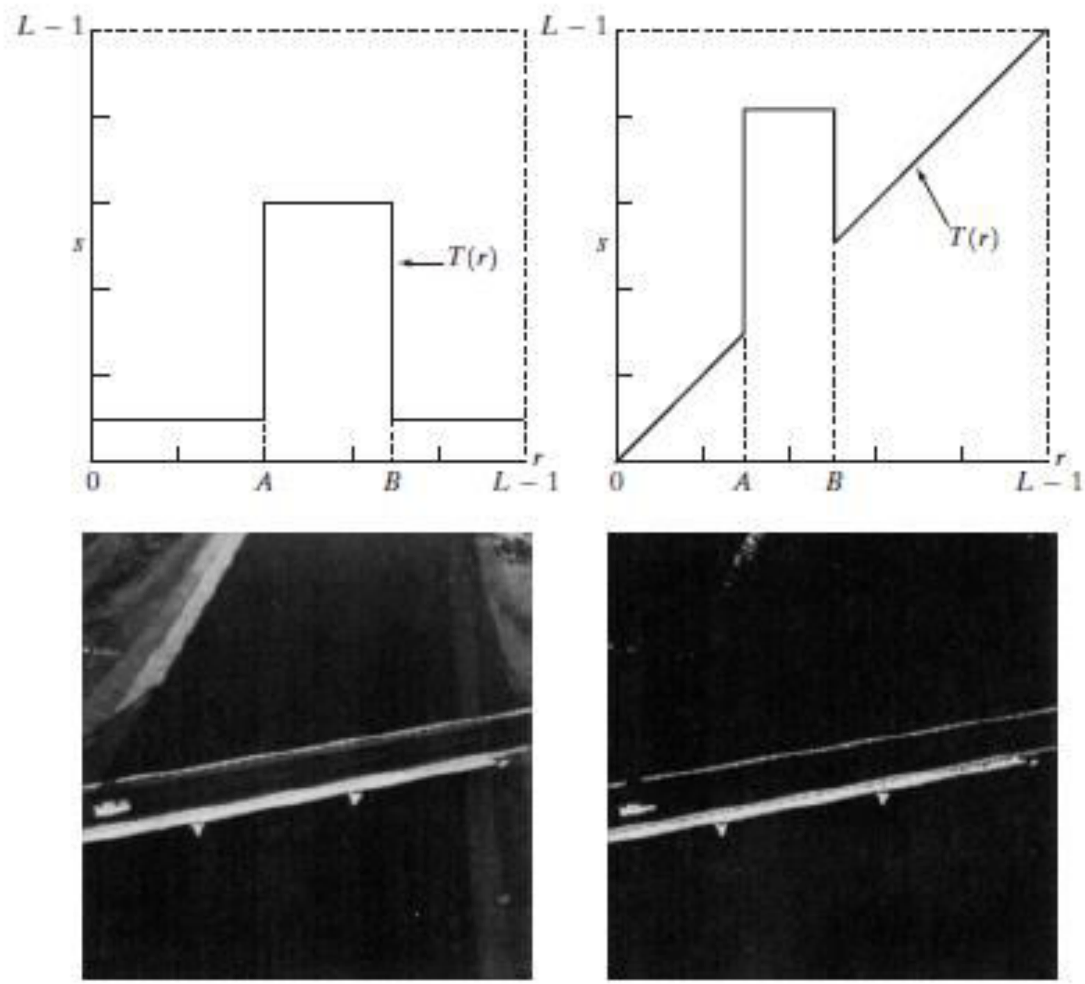
- Fig. 3.10(c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

- Finally, Fig. 3.10(d) shows the result of using the *thresholding function* defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig 3.11 (a), produces a binary image.
  - The second approach, based on the transformation shown in Fig 3.11 (b), brightens the desired range of gray levels but preserves gray levels unchanged.
  - Fig 3.11 (c) shows a gray scale image, and fig 3.11 (d) shows the result of using the transformation in Fig 3.11 (a).

Gray-level Slicing

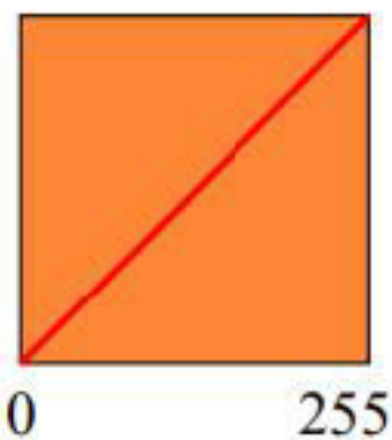


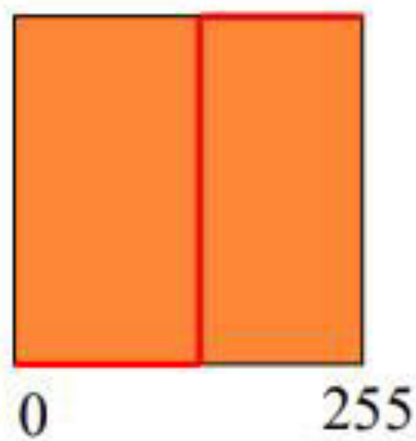
a b
c d

FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Input image

Output image





Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

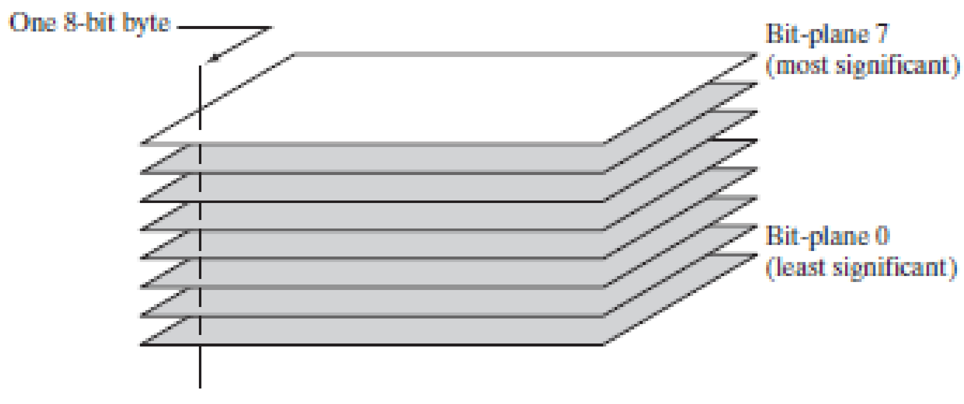


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

BIT PLANE SLICING EXAMPLE

Original image

Bit plane 7

Bit plane 6

Bit plane 4

Bit plane 1

1. Which of the following expression is used to denote spatial domain process?

a) $g(x,y)=T[f(x,y)]$
b) $f(x+y)=T[g(x+y)]$
c) $g(xy)=T[f(xy)]$
d) $g(x-y)=T[f(x-y)]$

1. Which of the following expression is used to denote spatial domain process?

a) $g(x,y)=T[f(x,y)]$

b) $f(x+y)=T[g(x+y)]$

c) $g(xy)=T[f(xy)]$

d) $g(x-y)=T[f(x-y)]$

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

Digital Image Processing

MODULE-3

Image Restoration

What is Image Restoration:

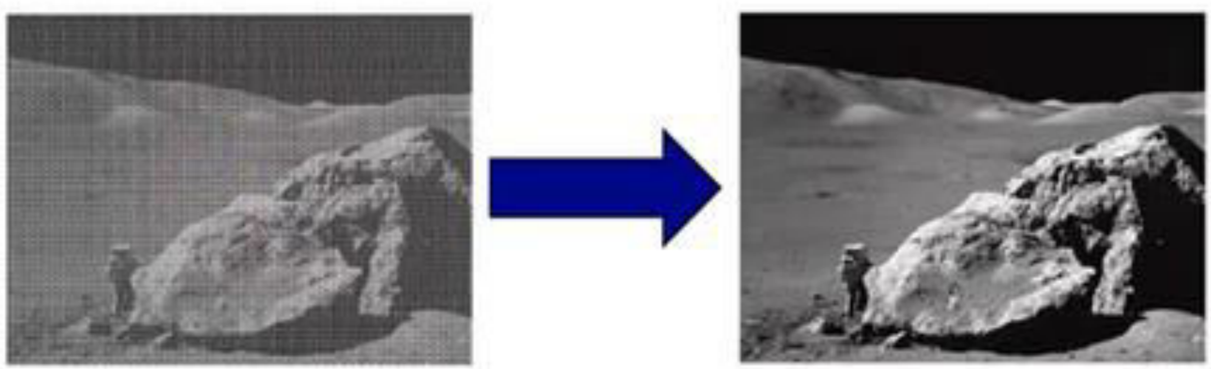
Image restoration aim to improve an image in some predefined sense.

What about image enhancement?

Image enhancement also improves an image by applying filters.

What is Image Restoration?

- Image restoration attempts to restore/reconstruct/ recover images that have been degraded.
  - It identifies the degradation process and attempt to reverse it
  - It is similar to image enhancement, but more objective



Difference:

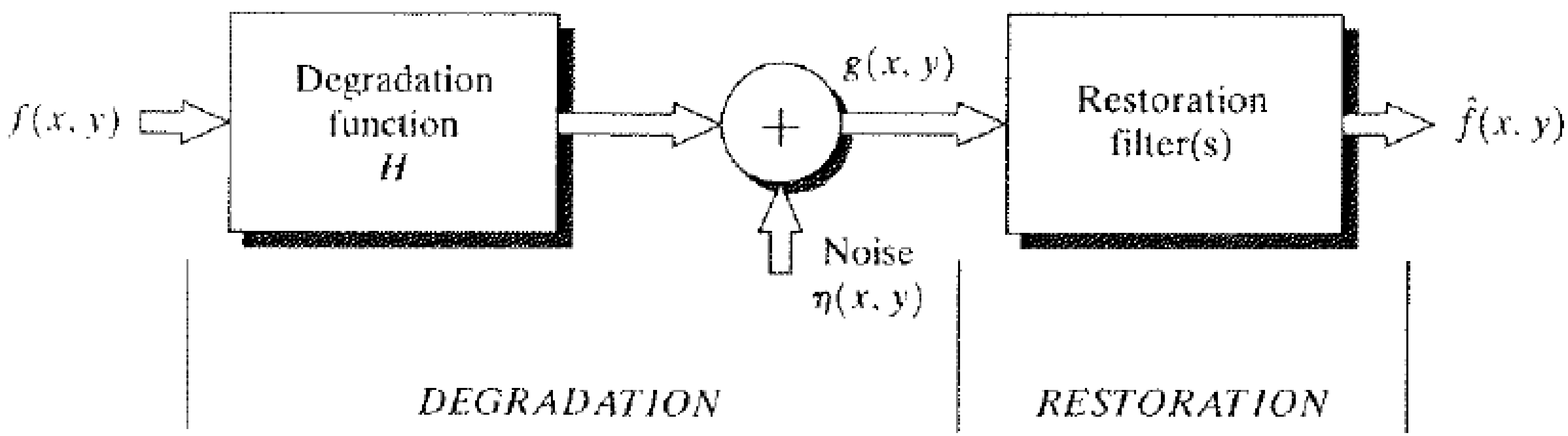
Image Enhancement	---	Subjective process
Image Restoration	---	Objective Process

- ❑ Restoration tries to recover / restore degraded image by using a prior knowledge of the degradation phenomenon.

- ❑ Restoration techniques focuses on:

1. Modeling the degradation
2. Applying inverse process in order to recover the original image.

Model of the Image Degradation / Restoration Process



- Degradation function along with some additive noise operates on f(x, y) to produce degraded image g(x, y)

- Given g(x, y), some knowledge about the degradation function H and additive noise η(x, y), objective of restoration is to obtain estimate f'(x, y) of the original image.

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

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

- h(x, y) = Spatial representation of H
- * indicates convolution

- Since convolution in Spatial domain = multiplication in Frequency Domain

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

- We Assume that H is identity operator

- We deal only with degradation due to Noise

Noise Models: Noise in digital image arises during

1. Image Acquisition
2. Transmission

During Image Acquisition

- Environmental conditions (Light Levels)
- Quality of sensing element

During Transmission

- Interference during transmission

Spatial Noise Descriptor

- Statistical behavior of the gray level values in the noise component.
- Can be considered as random variables
- Characterized by Probability Density Functions (PDFs)

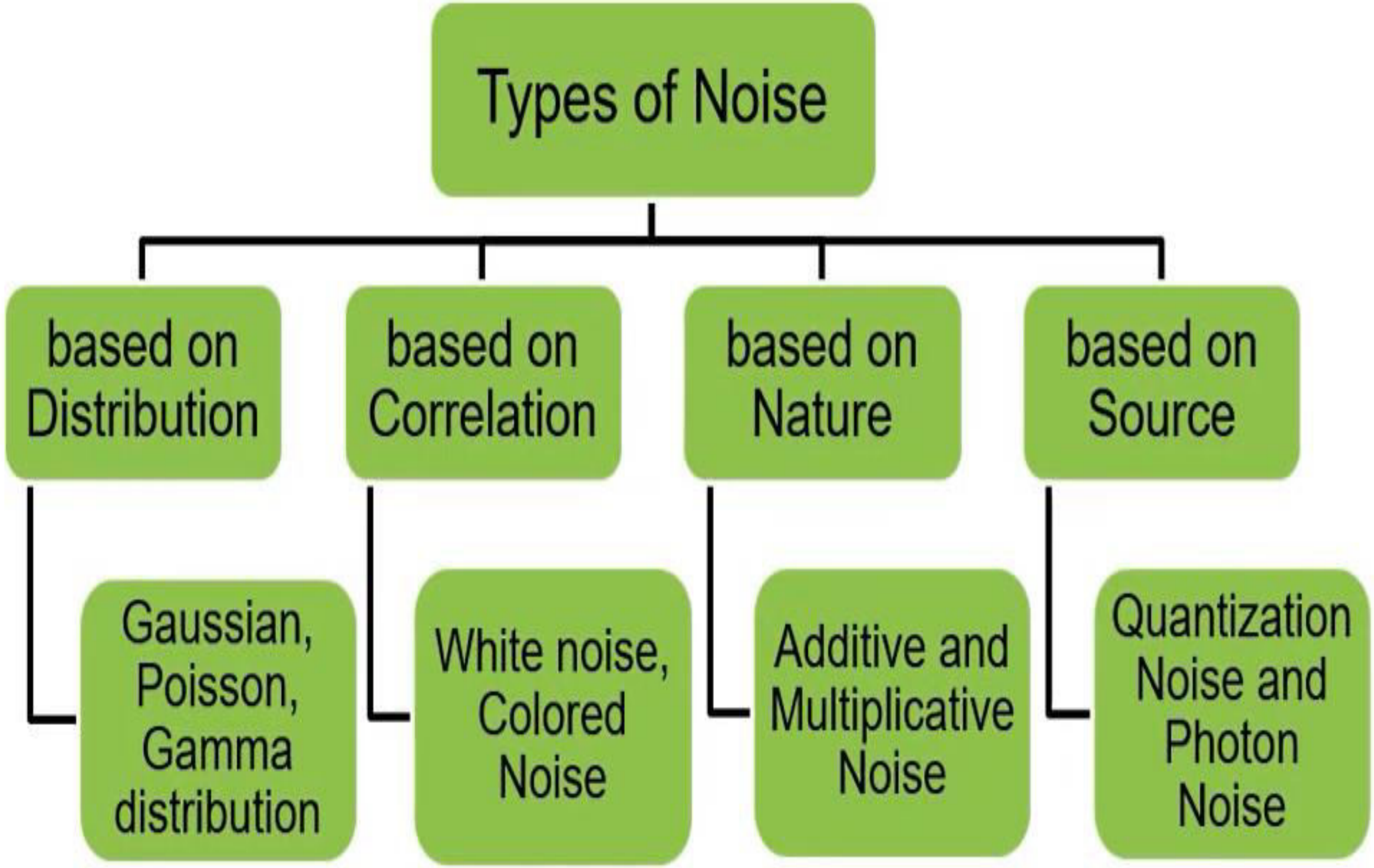
Noise

- Noise is signal similar to image.
- It's an unwanted signal which distorts the original pixel intensity values, thus, degrading the quality of image
- Noise Model Assumption
  - Independent of Spatial Coordinates
  - Un-correlated with the image, i.e., no relation between the pixel values and noise components

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

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

Some of the types of frequent occurred noise that are encountered in Image Processing can be categorized into four categories.



Some Noise Probability Density Functions (PDFs):

- Uniform Noise
- Gaussian Noise
- Rayleigh Noise
- Erlang (Gamma) Noise
- Exponential Noise
- Impulse (Salt & Pepper Noise)
- Periodic Noise

Noise Modelling

Types of Noise based on Distribution

- Noise is a fluctuation in pixel values and it is characterized by random variable
- A random variable probability distribution is an equation that links the values of the statistical result with its probability of occurrence
- Categorization of noise based on probability distribution is very popular

1. Gaussian Noise

- Random noise that enters a system can be modelled as a Gaussian or normal distribution
- Gaussian noise affects both dark and light areas of image

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 gray level
- μ is the **mean of average** values of z
- σ is the standard deviation.
- The standard deviation squared, σ^2, is known as **variance** of z.

Noise Modelling

2. Impulse Noise

- It is also known as Shot Noise, Salt and Pepper Noise, and Binary Noise
- It occurs mostly because of sensor and memory problem because of which pixels are assigned incorrect maximum values

PDF of impulse noise

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

$z \rightarrow$ a & $b \rightarrow$ grey levels

$P_a = P_b = 0$ Unipolar Noise

3. Poisson Noise

- This type of noise manifests as a random structure or texture in images
- It is very common in X-ray images

PDF

$$P(z) = \frac{(np)^z}{z!} e^{-np}$$

$n \rightarrow$ total no. of pixels

$p \rightarrow$ Ratio of noise pixels to the total no. of pixels

Noise Modelling

4. Exponential Noise

- This type of noise occurs mostly due to the illumination problems
- ✓ It is present in laser imaging

PDF

$$P(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

✓ Mean $\rightarrow \frac{1}{a}$

✓ Variance $\rightarrow \frac{1}{a^2}$

5. Gamma Noise

- This type of noise also occurs mostly due to the illumination problems

PDF

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

* Mean $\rightarrow \frac{b}{a}$

* Variance $\rightarrow \frac{b}{a^2}$

Noise Modelling

6. Rayleigh Noise

- This type of noise is mostly present in range images
- Range images are mostly used in remote sensing applications

PDF

$$P(z) = \begin{cases} \frac{2}{b}(z-a)\, e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{otherwise} \end{cases}$$

* Mean $\rightarrow a + \sqrt{\frac{\pi b}{4}}$
* Variance $\rightarrow \frac{b(4-\pi)}{4}$

7. Uniform Noise

- It is also very popular noise occurs in images where different values of noise are equally probable
- It occurs because of Quantization noise

$a \rightarrow$ min grey value
$b \rightarrow$ max gray value

PDF

$$P(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq f(x,y) \leq b \\ 0 & \text{otherwise} \end{cases}$$

* Mean $\rightarrow a+b/2$
* Variance $\rightarrow (b-a)^2/12$

Noise Modelling

Types of Noise based on Correlation

- Statistical dependence among pixels is known as correlation
- If a pixel is independent of its neighbouring pixels, it is known as uncorrelated pixel otherwise it is known as correlated pixel

- Uncorrelated Noise is known as White Noise
- Mathematically for white noise, the noise power spectrum or power spectral density remains constant with frequency

- Characterization of colored noise is quite difficult because its origin is mostly unknown
- One popular colored noise is Pink Noise
- Its power spectrum is not constant, rather it is proportional to reciprocal of frequency
- This is also known as 1/f or Flicker Noise

Noise Modelling

Types of Noise based on Nature

Based on the nature of its presence, noise can be modelled in two ways

1. Additive Noise

- In this case an image can be perceived as the image plus noise
- This is a linear problem

Perceived image,

$$g(x,y) = \underbrace{f(x,y)}_{\text{i/p image}} + \underbrace{n(x,y)}_{\text{noise}}$$

2. Multiplicative Noise

- It can be modelled as multiplicative process
- Speckle noise is most encountered multiplicative noise in image processing
- It is mostly present in medical images
- It can be modelled as pixel value multiplied by the random value

$$I = f(x,y) + [f(x,y) \times N_g]$$

N_g: Random noise having a zero mean gaussian PDF

Noise Modelling

Types of Noise based on Source

Noise based on source, commonly encountered in image processing are:

- Quantization Noise
- Photon Noise

- Quantization Noise occurs due to a difference between the actual and allocated values
- It is inherent in the quantization process

- Photon Noise occurs due to the statistical nature of electromagnetic waves
- Generation of photon is not constant because of statistical variation
- This causes variation in photon count which is known as photon noise
- It is present in many medical images

Graph representation for Uniform Noise Model

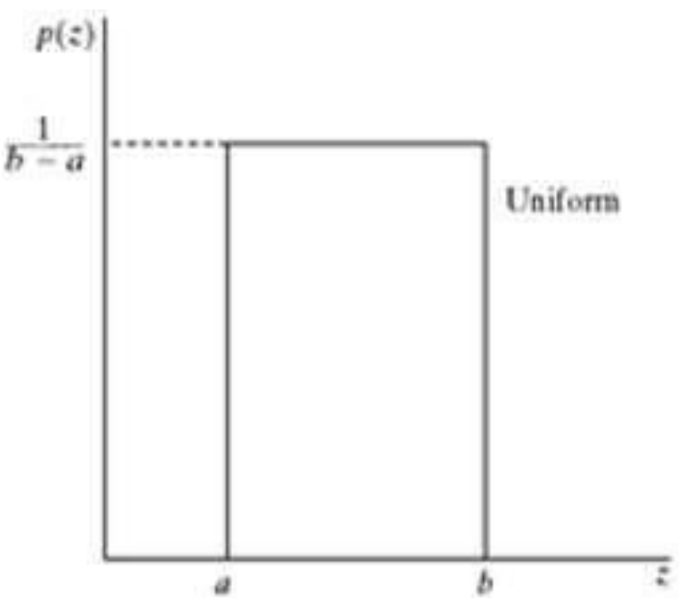
PDF of Uniform Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

$$\mu = \frac{a+b}{2}$$

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



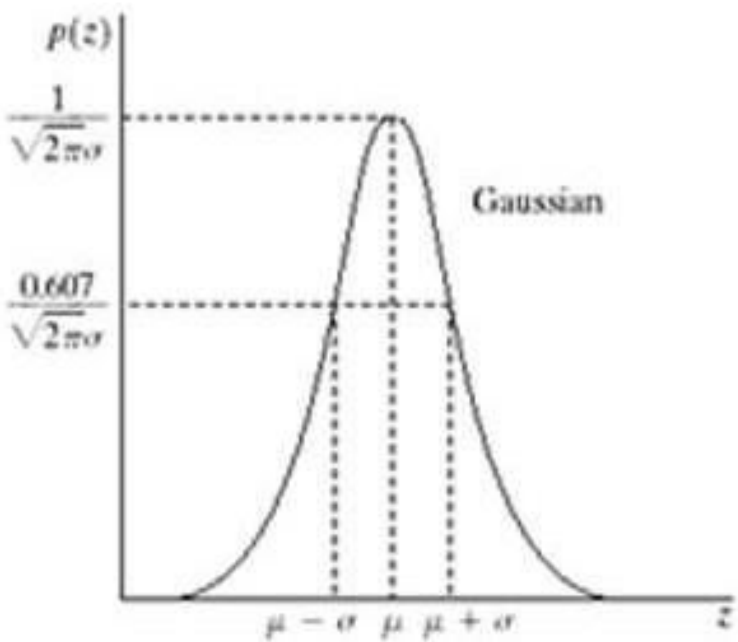
Plot of function

Graph representation for Gaussian / Normal Noise Model

1. Most frequently used.
2. PDF of Gaussian random variable z is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ → Standard Deviation of z
- σ^2 → Variance of z



Plot of function

When z is defined by this equation then

- About 70% of its values will be in the range [(μ - σ),(μ + σ)] and
- About 95% of its values will be in the range [(μ - 2σ),(μ + 2σ)]

Graph representation for Rayleigh Noise Model

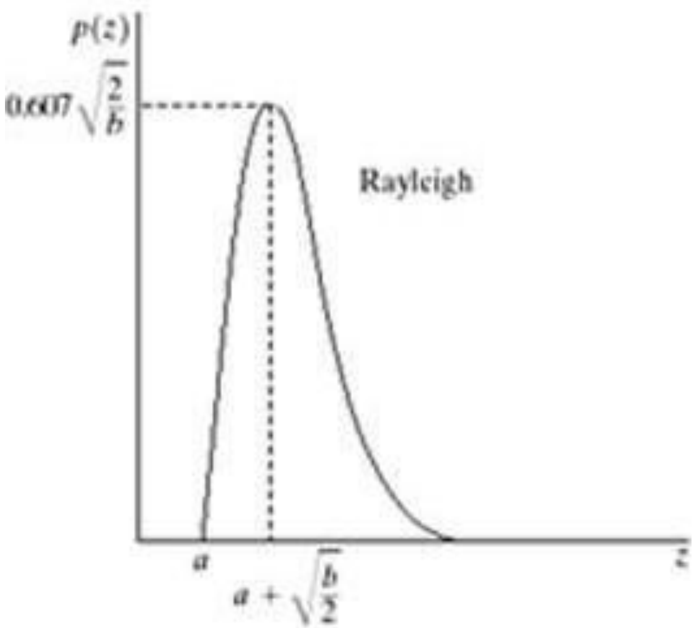
PDF of Rayleigh Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

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

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



Plot of function

- Basic shape of this density is skewed to the right.
- Quite useful for approximating skewed histograms.

Graph representation for Erlang (Gamma) Noise Model

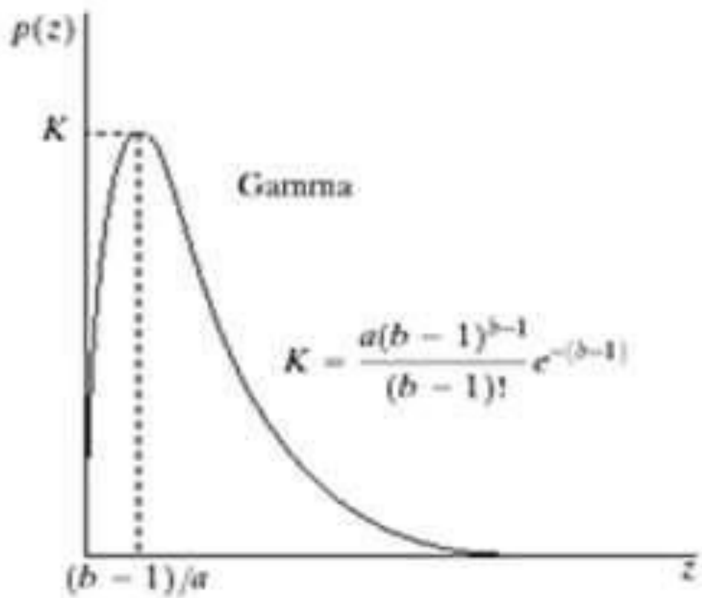
PDF of Erlang Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

- a > 0
- b = positive integer

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



Plot of function

- Above equation is also called Erlang Density

- If denominator is Gamma function $\Gamma(b)$ then it is called Gamma density

Graph representation for Exponential Noise Model

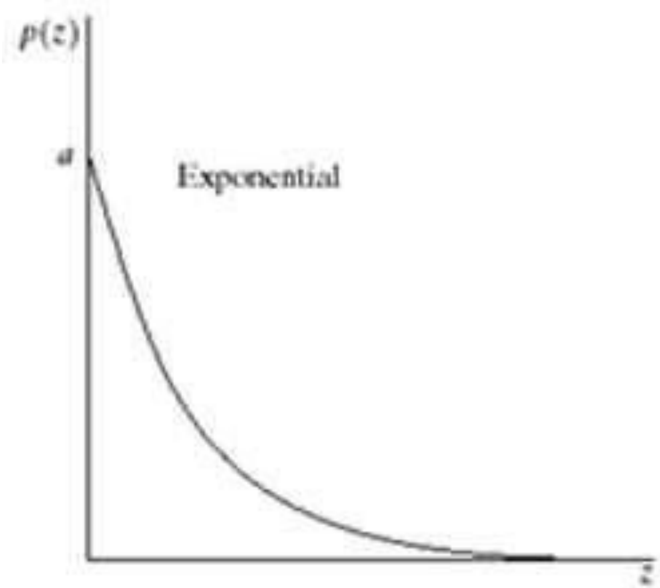
PDF of Exponential Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z
- a > 0

$$\mu = \frac{1}{a}$$

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



Plot of function

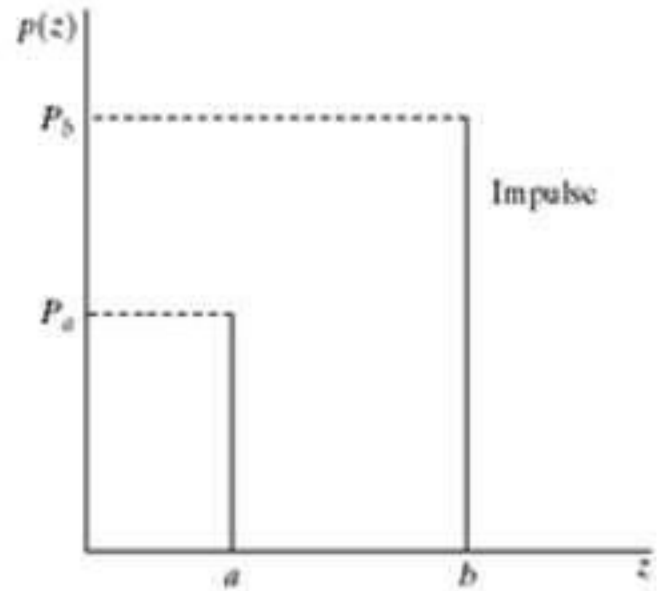
- Special case of Erlang Density Where b=1

Graph representation for Impulse (Salt & Pepper) Noise Model

PDF of Uniform Noise is given by:

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

- z → Gray level
- If b > a then b → light dot and a→ dark dot
- If either P_a or P_b = 0 → Unipolar Impulse Noise otherwise Bipolar Impulse Noise.



Plot of function

- If Neither probability is 0 and approximately equal then noise values will resemble salt & pepper granules randomly distributed over the image.
- Also referred as Shot and Spike Noise

Example

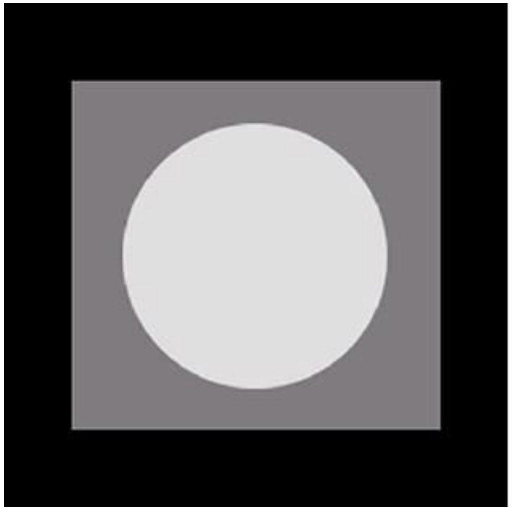
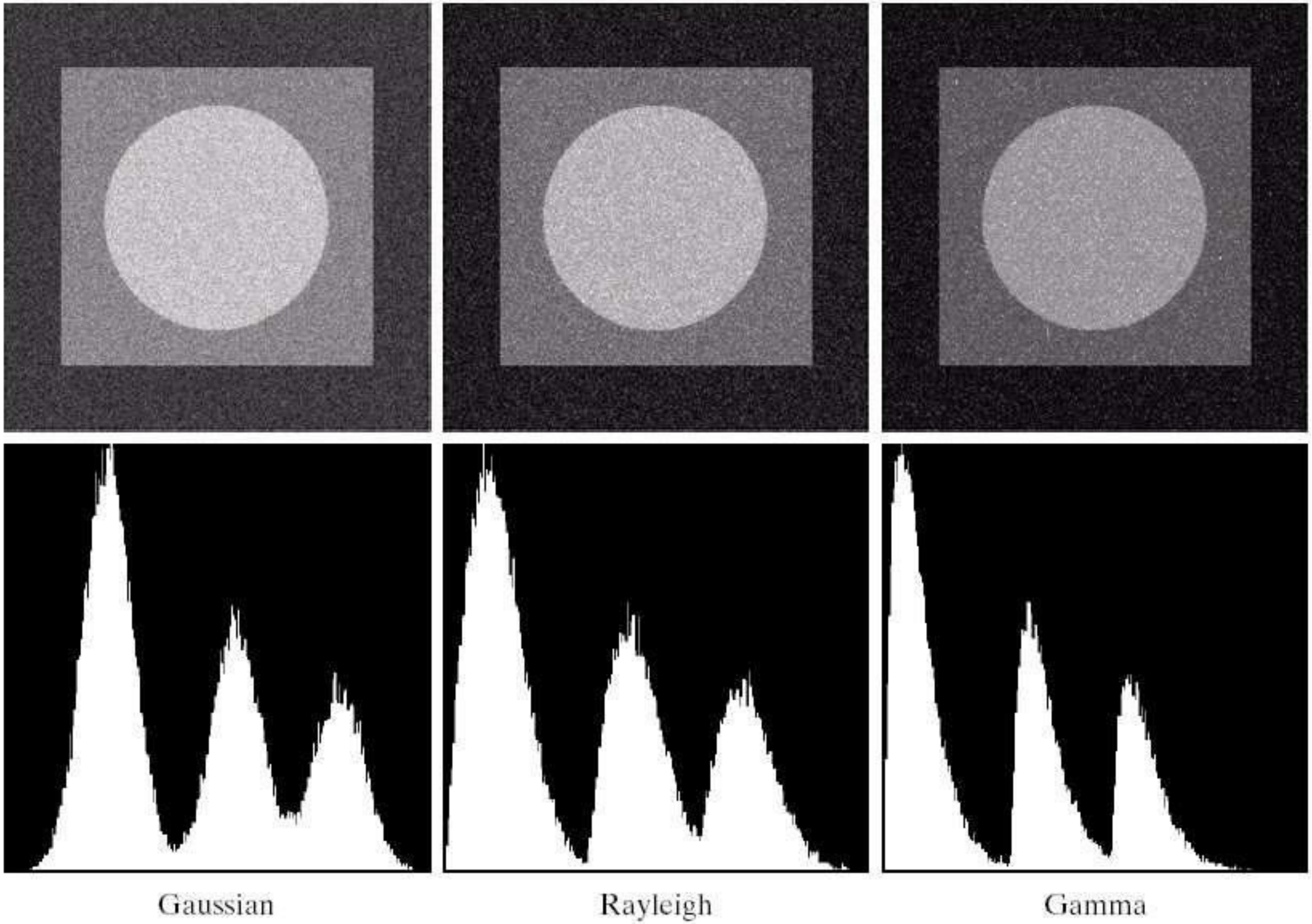


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

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.

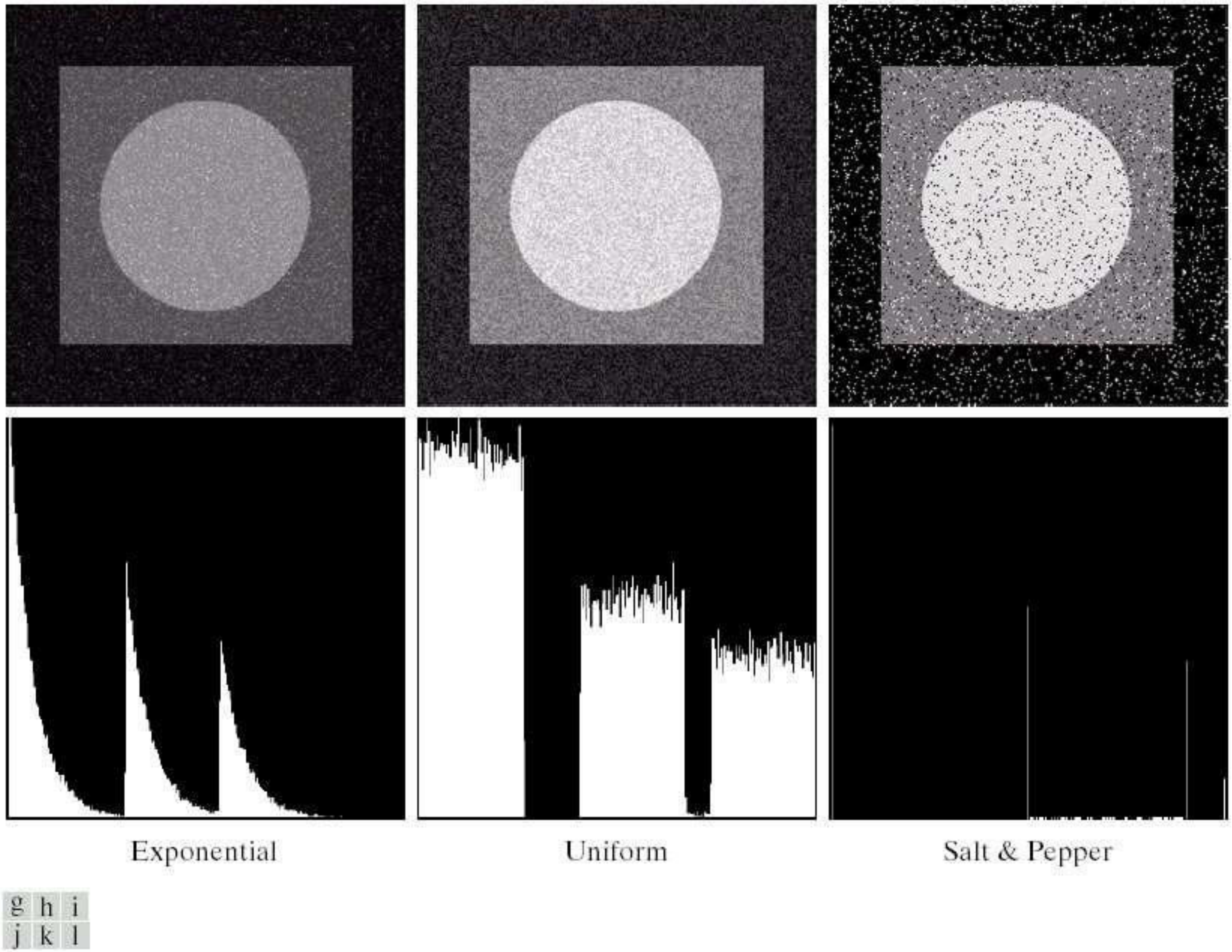
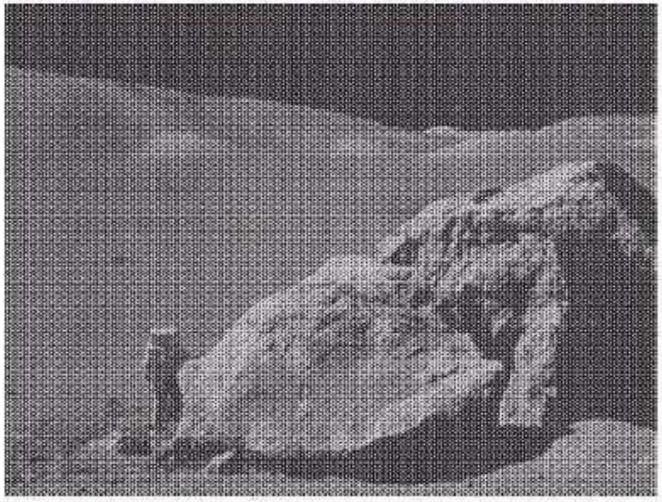


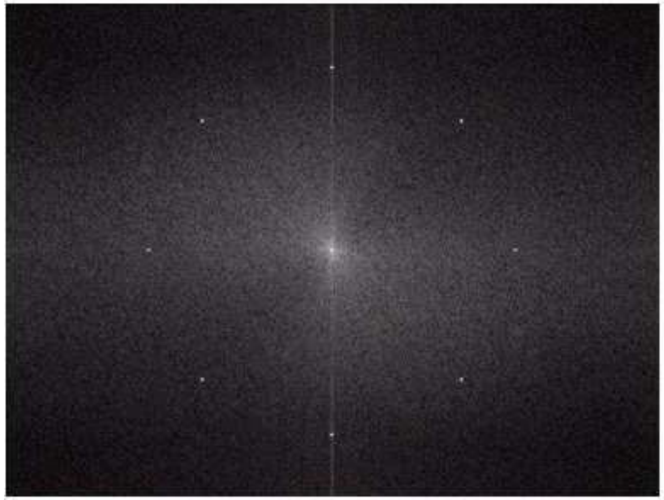
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

Periodic Noise

- Typically arises due to electrical / Electro-mechanical interference during image acquisition.
- Spatially dependent noise.
- Can be reduced significantly via Frequency Domain Filtering.
- Parameters can be estimated by inspecting the Frequency Spectrum of the image.
- Periodic noise tend to produce frequency spikes

Image corrupted by Sinusoidal noise





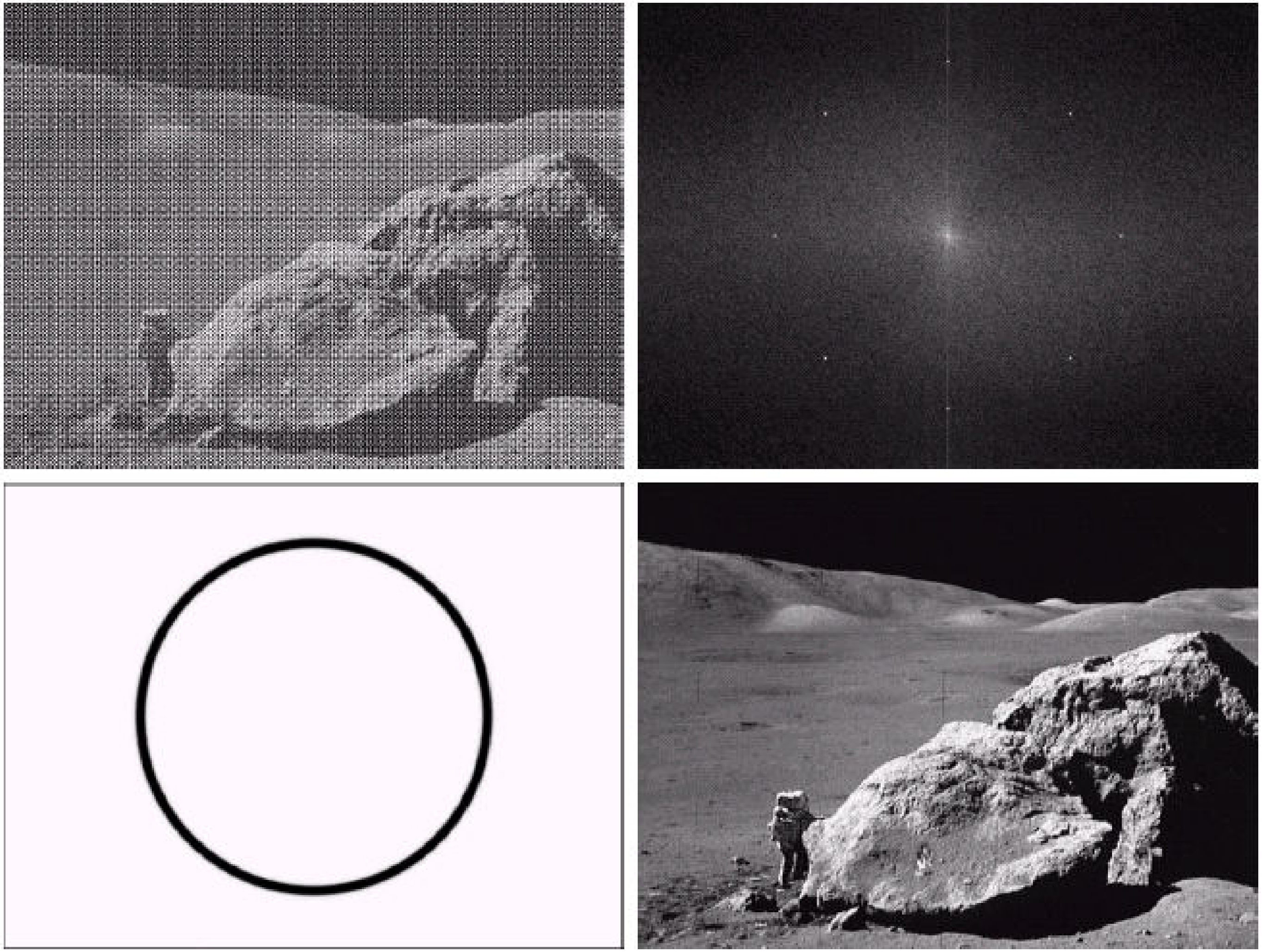
Spectrum (Each pair of conjugate impulses corresponds to one sine wave)

Periodic noise reduction using Frequency Domain Filtering

Periodic Noise Reduction by Frequency Domain Filtering

- Periodic noise is due to the electrical or electromechanical interference during image acquisition.
- Can be estimated through the inspection of the Fourier spectrum of the image.

Periodic Noise Reduction by Frequency Domain Filtering

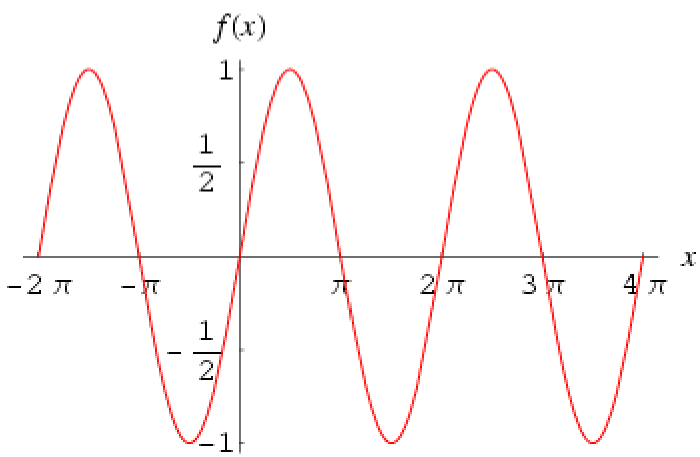


a b
c d

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 Function

- A function *f* is periodic with period *P* greater than zero if
  - $Af(x + P) = Af(x)$, where A denotes amplitude.



$f(x) = sinx$, $P = 2\pi$, frequency=$1/2\pi$, A=1.

$f(x) = A sin nx$, $P = 2\pi/n$, frequency=$n/2\pi$.

  - $n\uparrow$, frequency$\uparrow$.

Periodic noise reduction in frequency domain

- Pure sine wave
  - Appear as a pair of impulse (conjugate) in the frequency domain

$$\begin{cases} f(x,y) = A\sin(u_0 x + v_0 y) \\ F(u,v) = -j\dfrac{A}{2}\left[\delta(u - \dfrac{u_0}{2\pi}, v - \dfrac{v_0}{2\pi}) - \delta(u + \dfrac{u_0}{2\pi}, v + \dfrac{v_0}{2\pi})\right] \end{cases}$$

Periodic noise reduction using Frequency Domain Filtering

- Bandreject filters
- Bandpass filters
- Notch filters
- Optimum notch filtering

Periodic Noise Reduction by Frequency Domain Filtering

- Bandreject filters
  - Remove or attenuate a band of frequencies.

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

  - D_0 is the radius.
  - $D(u, v)$ is the distance from the origin, and
  - W is the width of the frequency band.

Butterworth and Gaussian Bandreject Filters

- Butterworth bandreject filter (order *n*)

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

- Gaussian band reject filter

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

Bandreject Filters

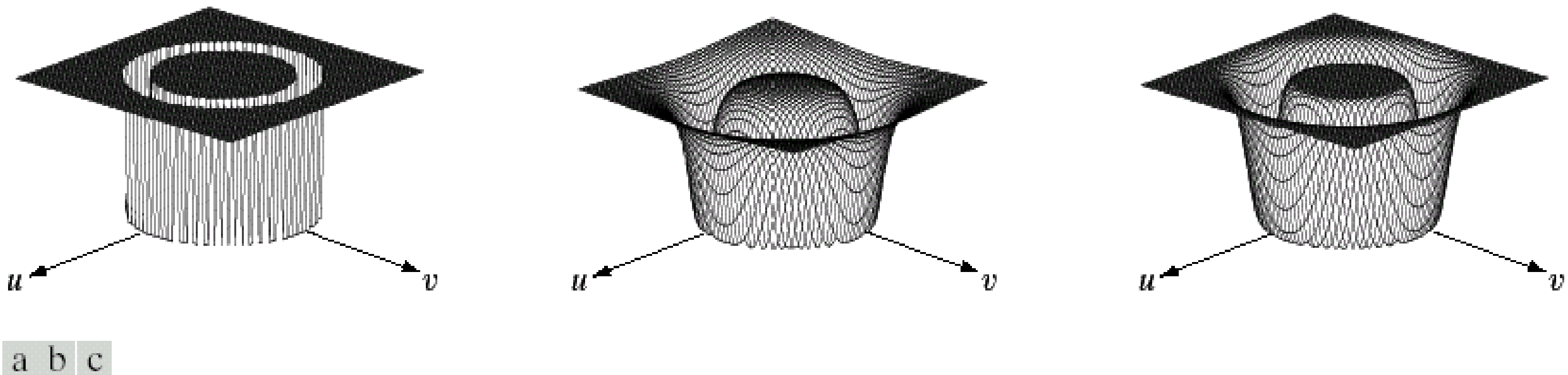


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandpass filter

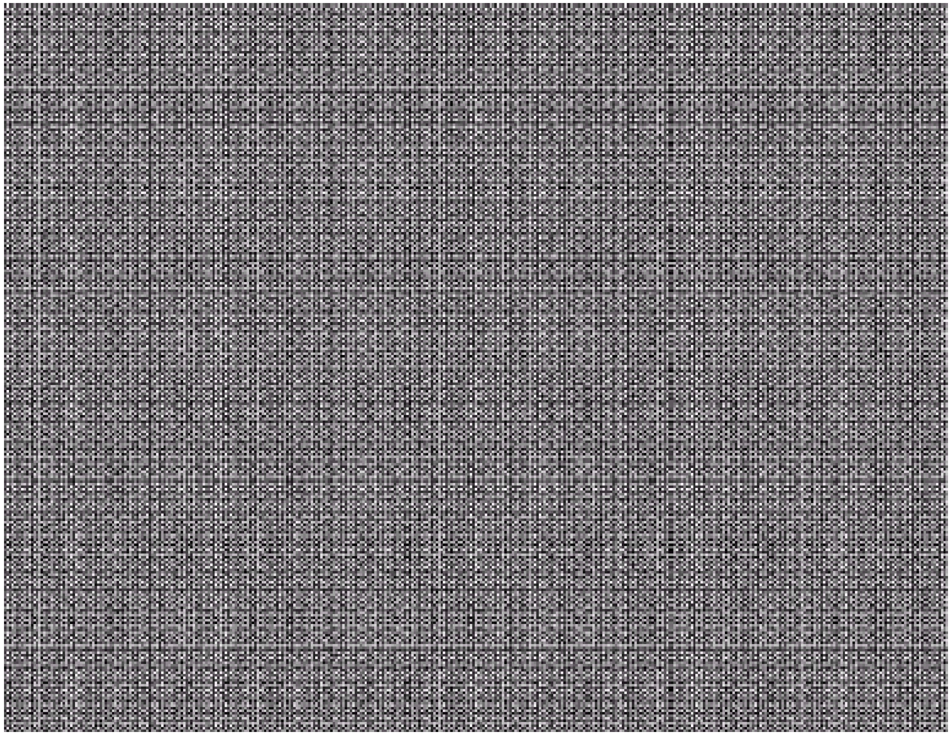
- Obtained form bandreject filter

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

- The goal of the bandpass filter is to isolate the noise pattern from the original image, which can help simplify the analysis of noise, reasonably independent of image content.

Result of The BandPass Filter

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



Notch filters

- Notch filter rejects (passes) frequencies in predefined neighborhoods about a center frequency.

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

where

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

$$D_2(u,v)=\left[(u-M/2+u_0)^2+(v-N/2+v_0)^2\right]^{1/2}$$

Notch filters

- Butterworth notch filter

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

- Gaussian notch filter

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

- Note that these notch filters will become highpass when $u_0=v_0=0$

Notch filters

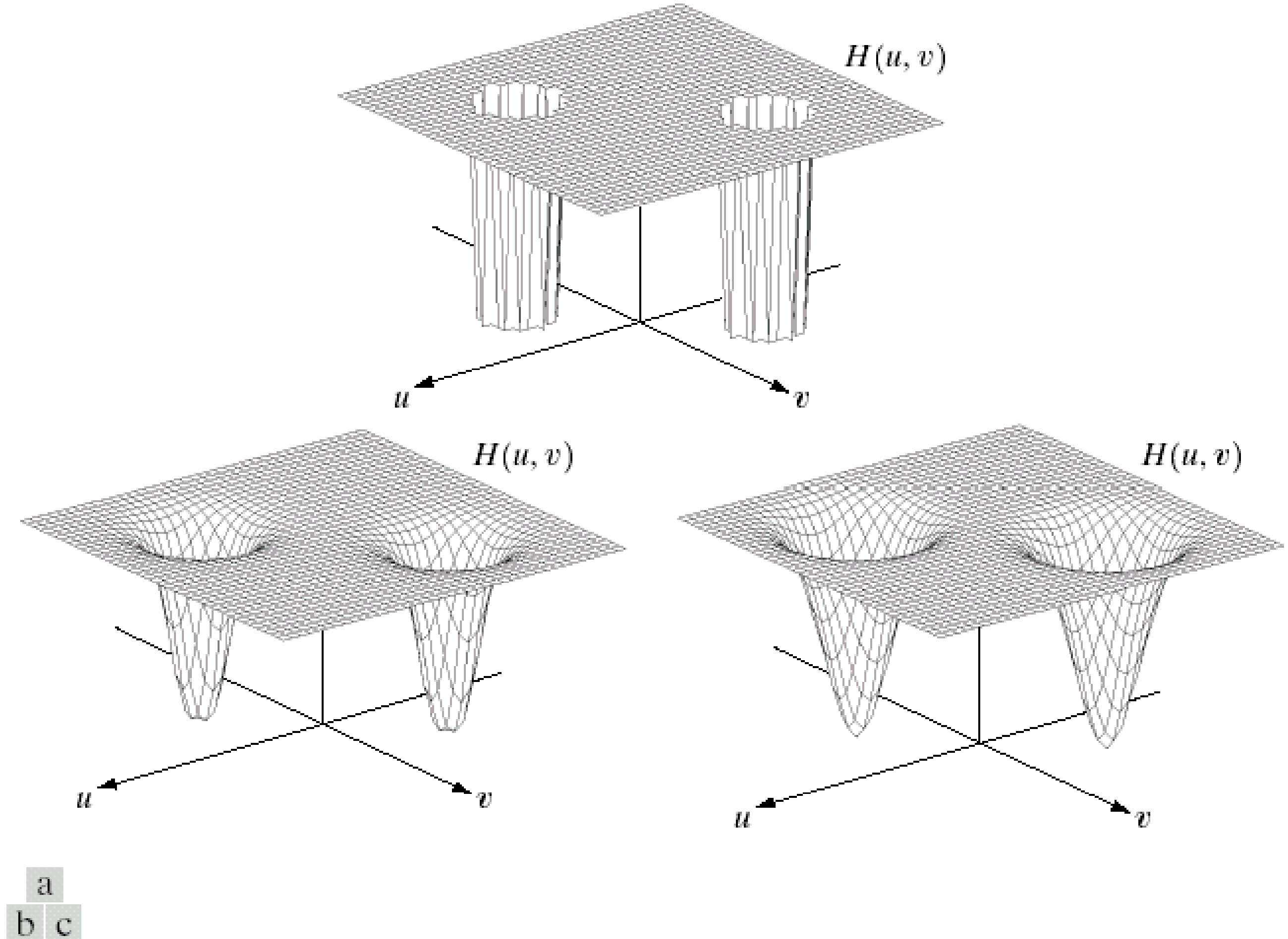
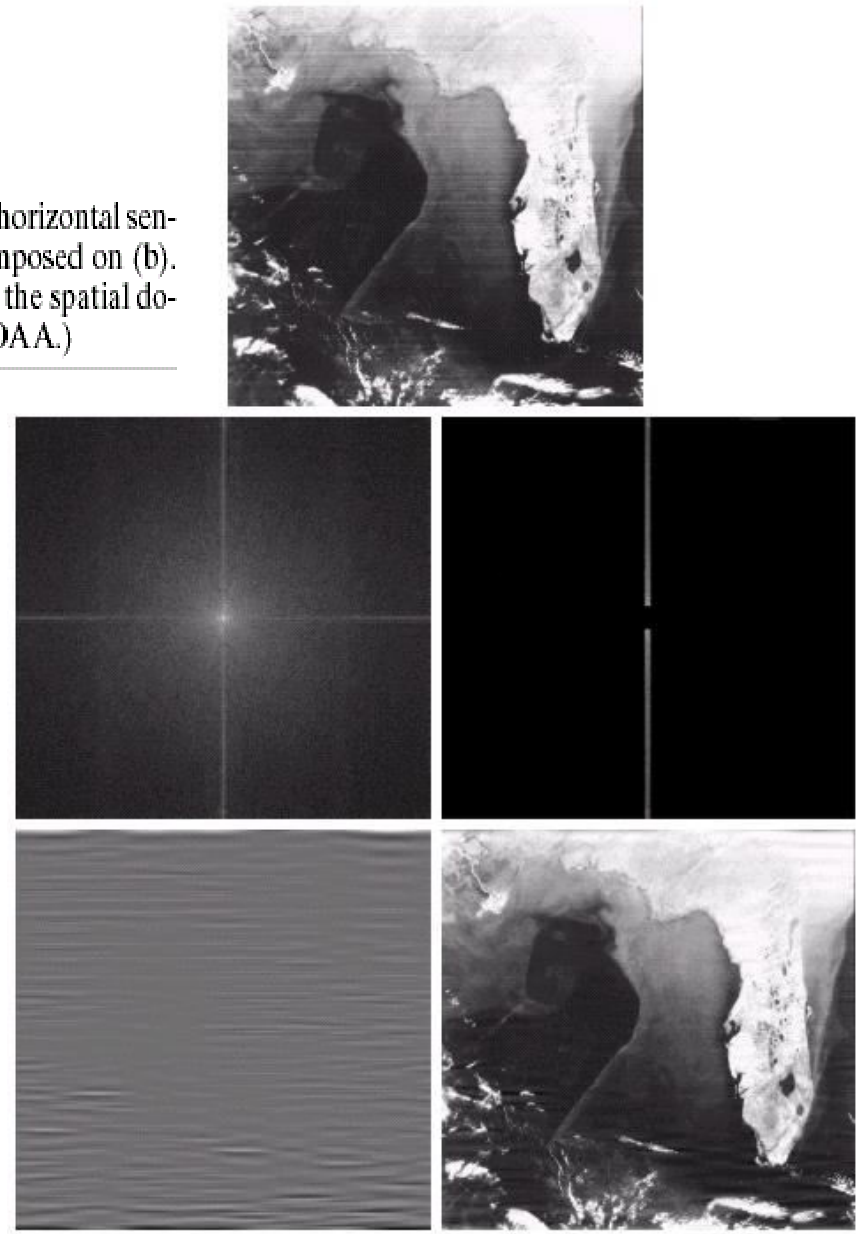


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

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.)

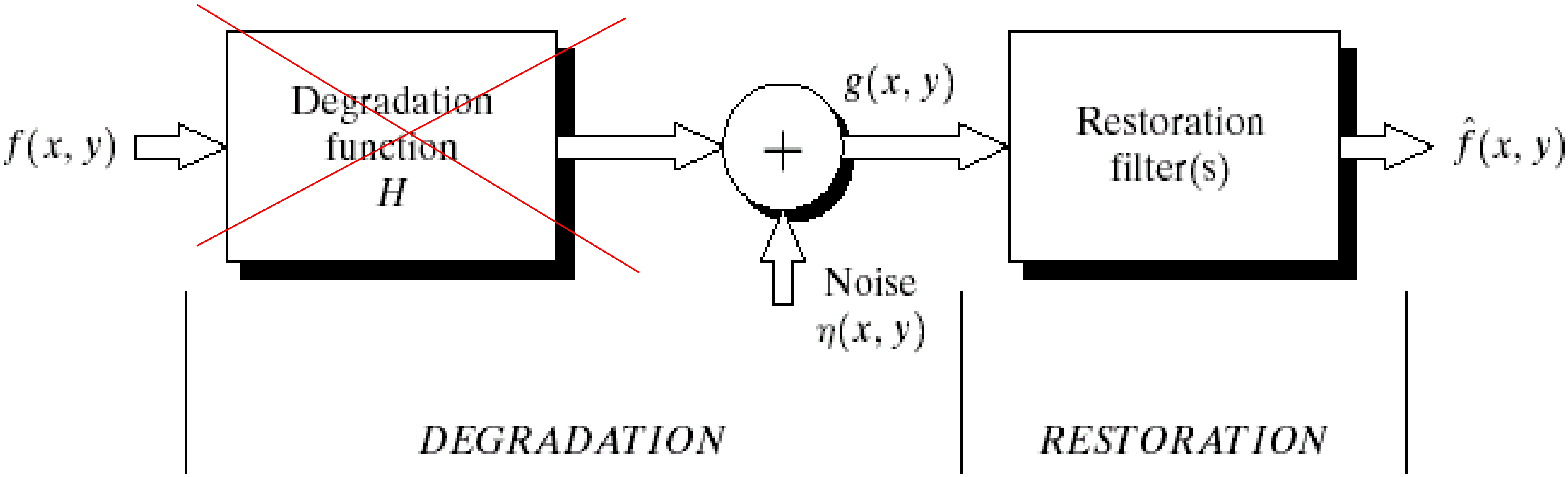


Use 1-D Notch pass filter to find the horizontal ripple noise

Restoration in the Presence of Noise Only using Spatial Filtering

- Mean filters
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contra-harmonic mean filter
- Order statistics filters
  - Median filter
  - Max and min filters
  - Mid-point filter
  - alpha-trimmed filters
- Adaptive filters
  - Adaptive local noise reduction filter
  - Adaptive median filter

Additive noise only



$$
\begin{cases} g(x,y)=f(x,y)+\eta(x,y) \\ G(u,v)=F(u,v)+N(u,v) \end{cases}
$$

Spatial filters for de-noising additive noise

- Skills similar to image enhancement
- Mean filters
- Order-statistics filters
- Adaptive filters

Mean filters

- Arithmetic mean

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

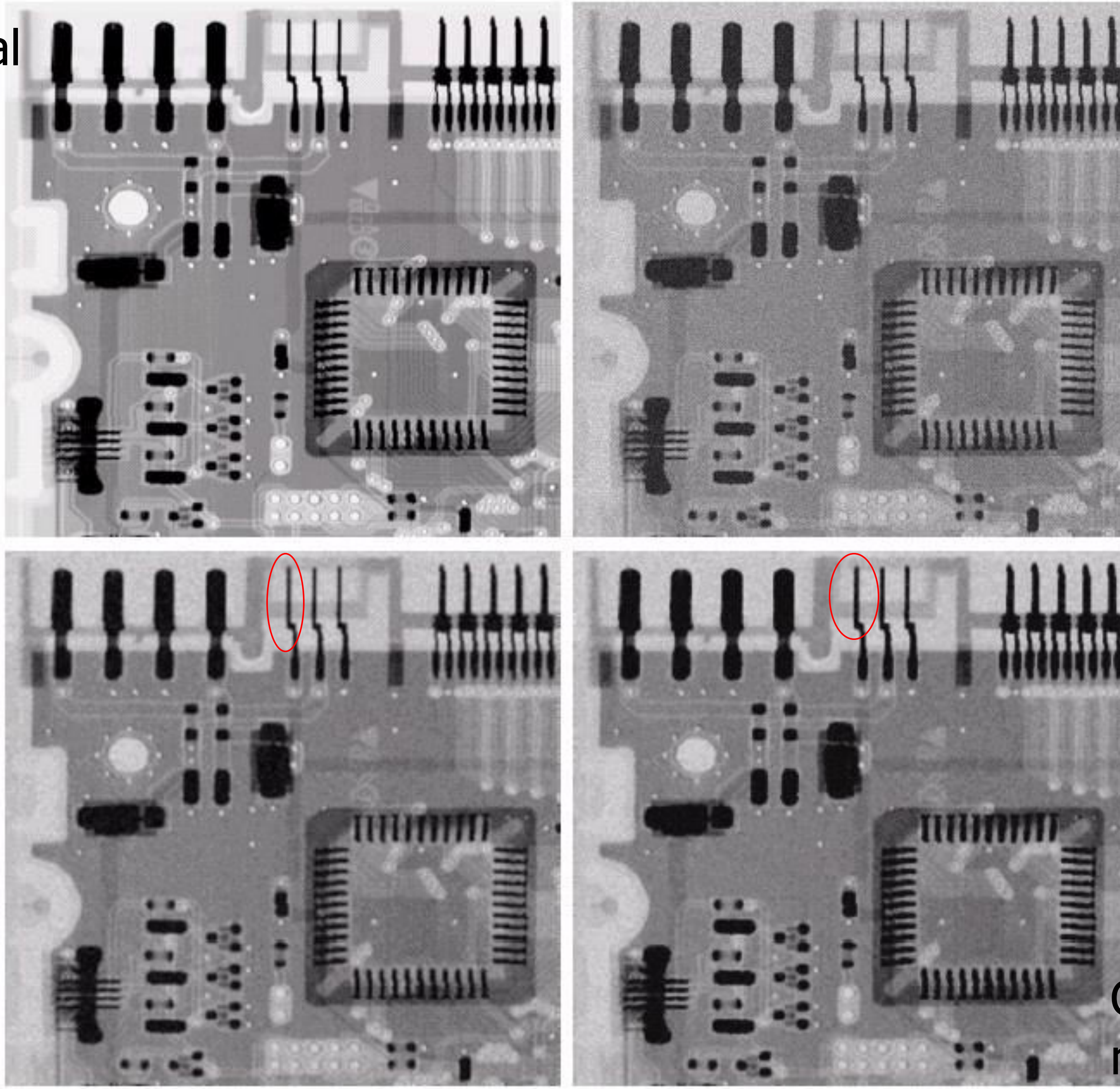
Window centered at (x,y)

- Geometric mean

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

original

Noisy Gaussian $\mu=0$ $\sigma=20$



Arith. mean

Geometric mean

Mean filters (cont.)

- Harmonic mean filter

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

- Contra-harmonic mean filter

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

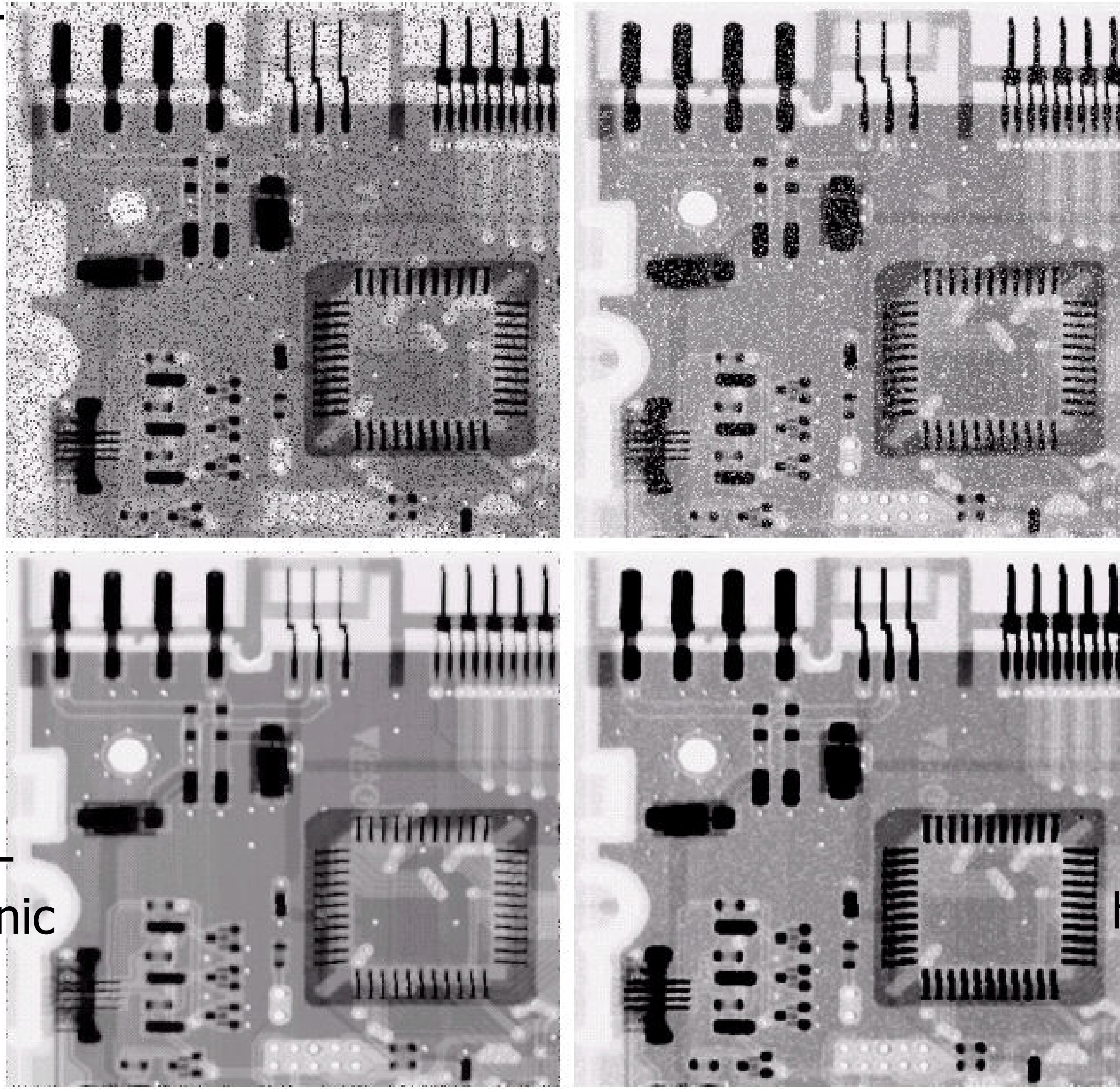
Q=-1, harmonic

Q=0, airth. mean

Q=+, ?

Pepper Noise

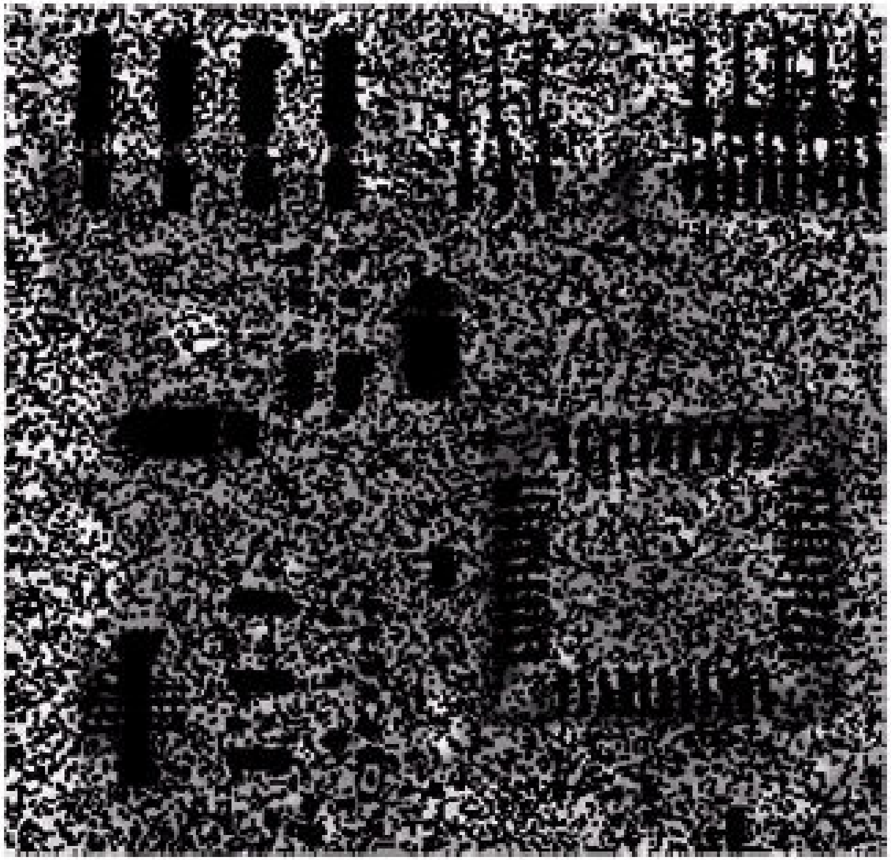
Salt Noise



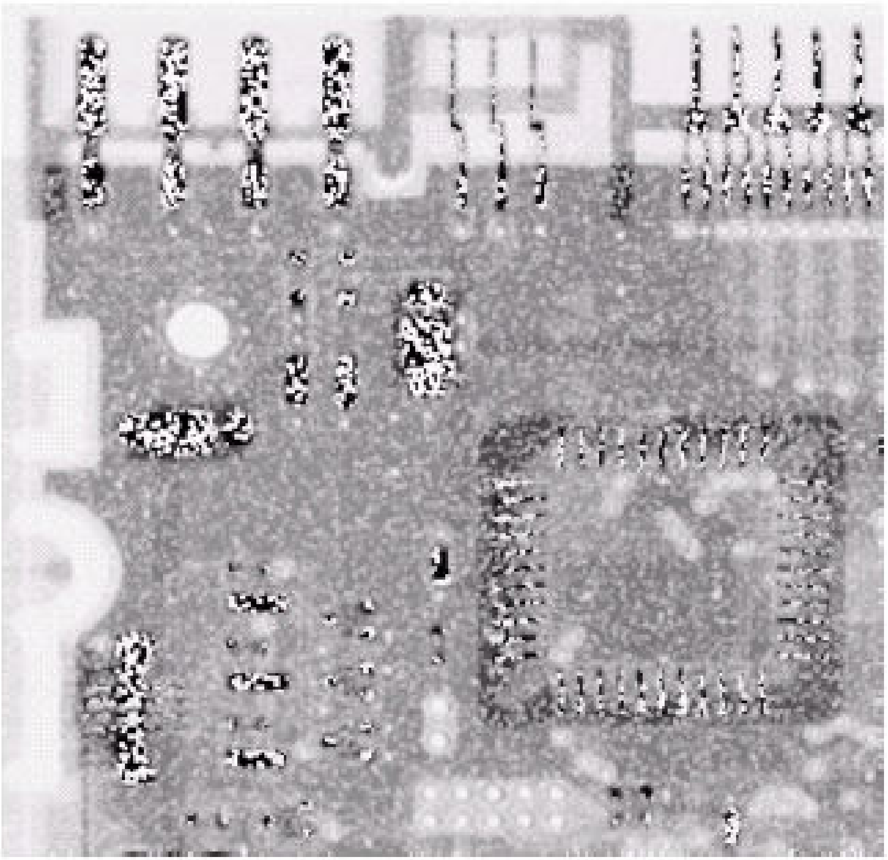
Contra-harmonic Q=1.5

Contra-harmonic Q=-1.5

Wrong sign in contra-harmonic filtering



Q=-1.5

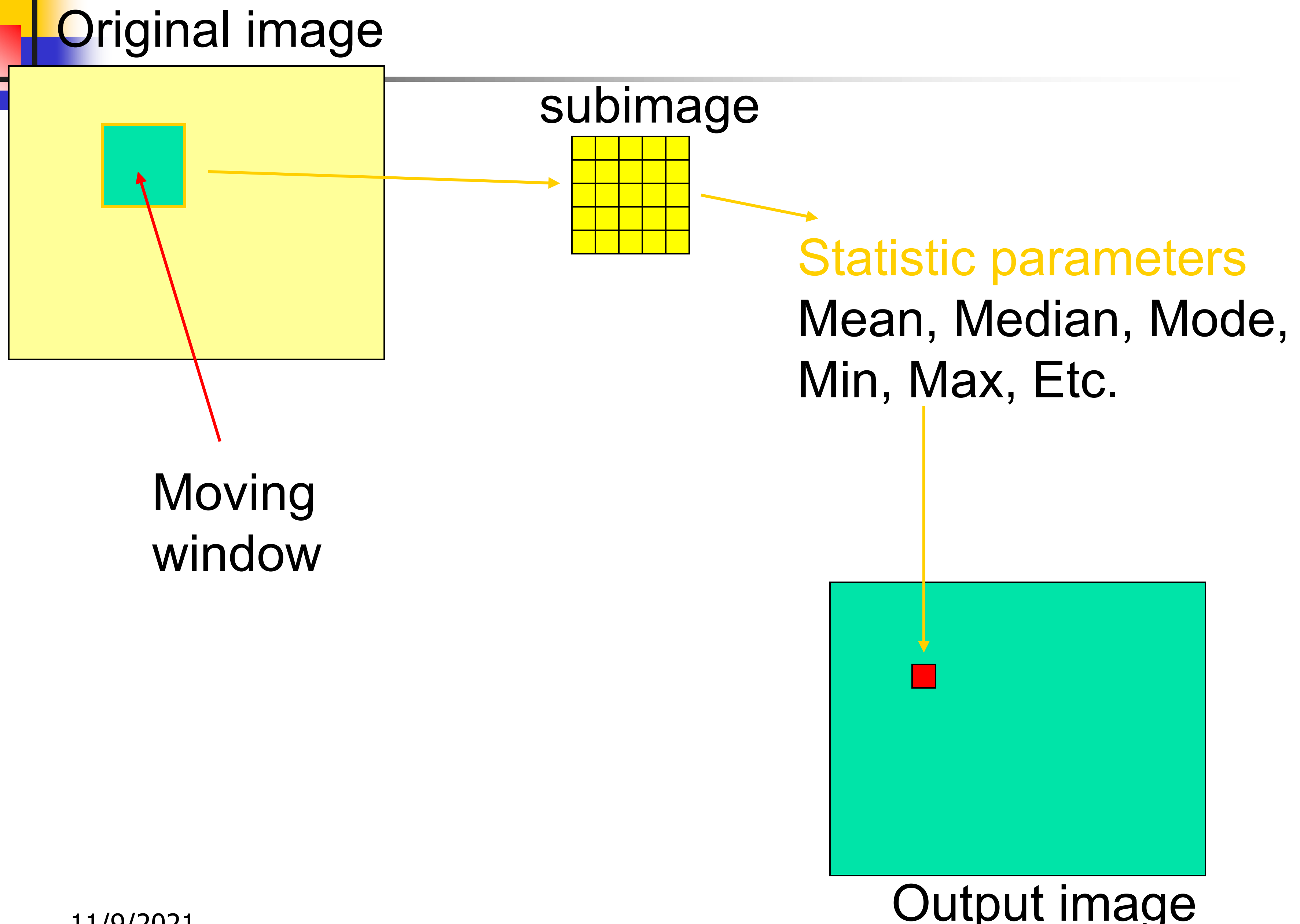


Q=1.5

Order-statistics filters

- Based on the ordering(ranking) of pixels
  - Suitable for unipolar or bipolar noise (salt and pepper noise)
- Median filters
- Max/min filters
- Midpoint filters
- Alpha-trimmed mean filters

Order-Statistic Filters



Order-statistics filters

- Median filter

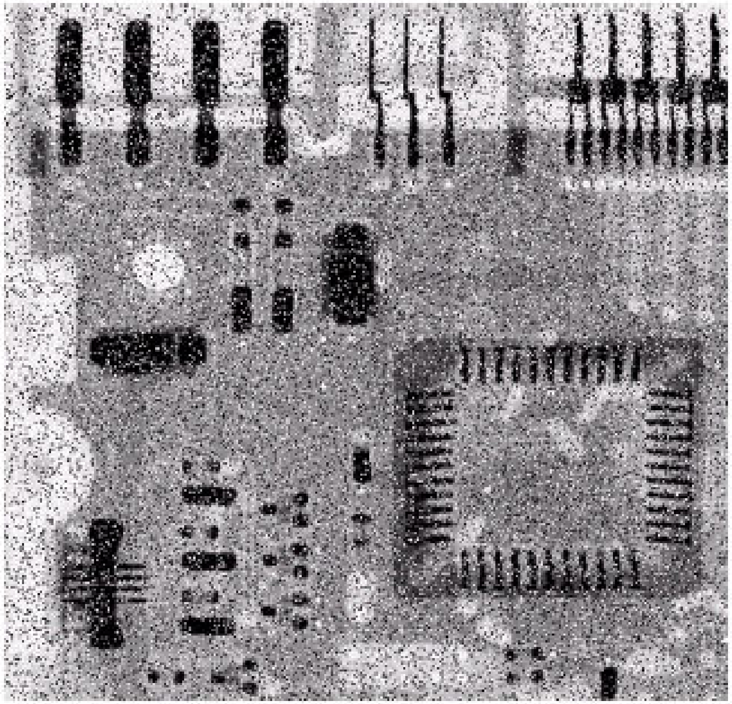
$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

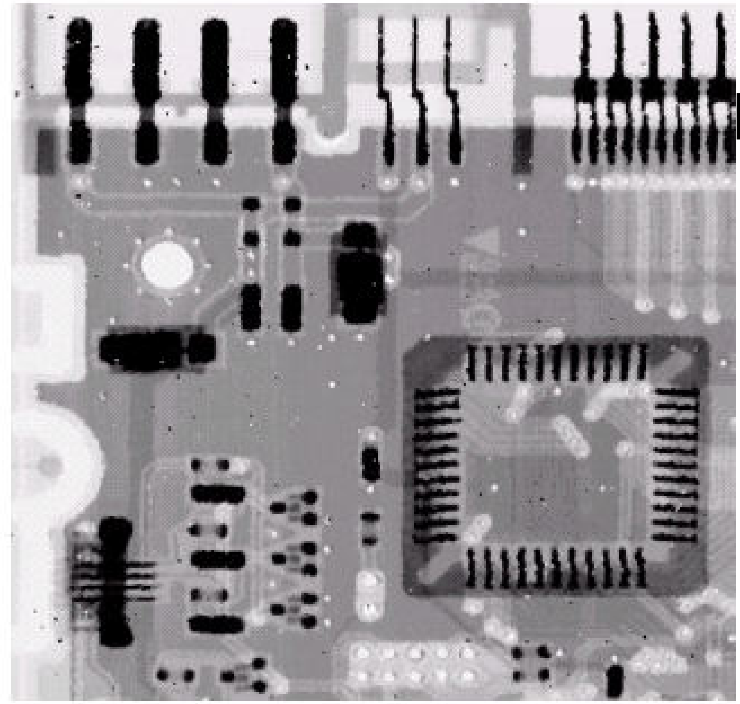
- Max/min filters

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

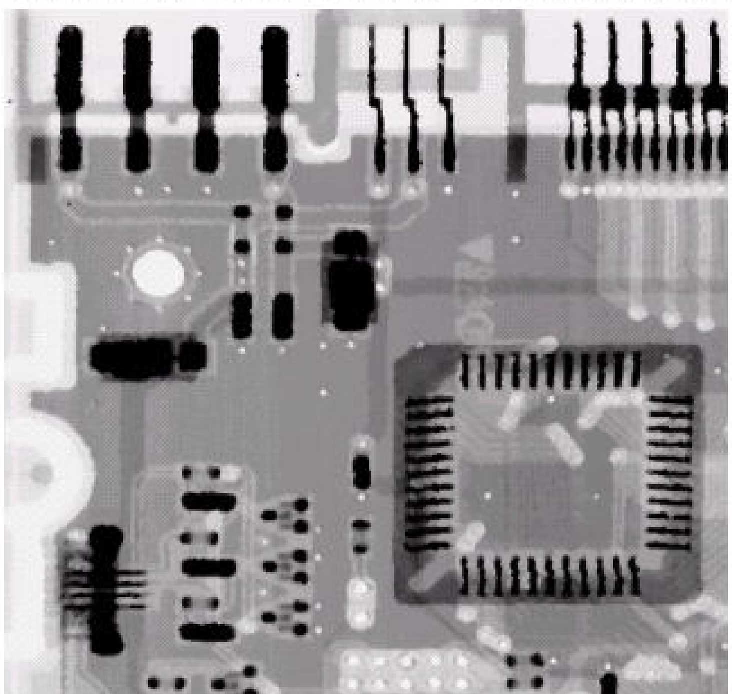
$$\hat{f}(x, y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

bipolar Noise
$P_a = 0.1$
$P_b = 0.1$

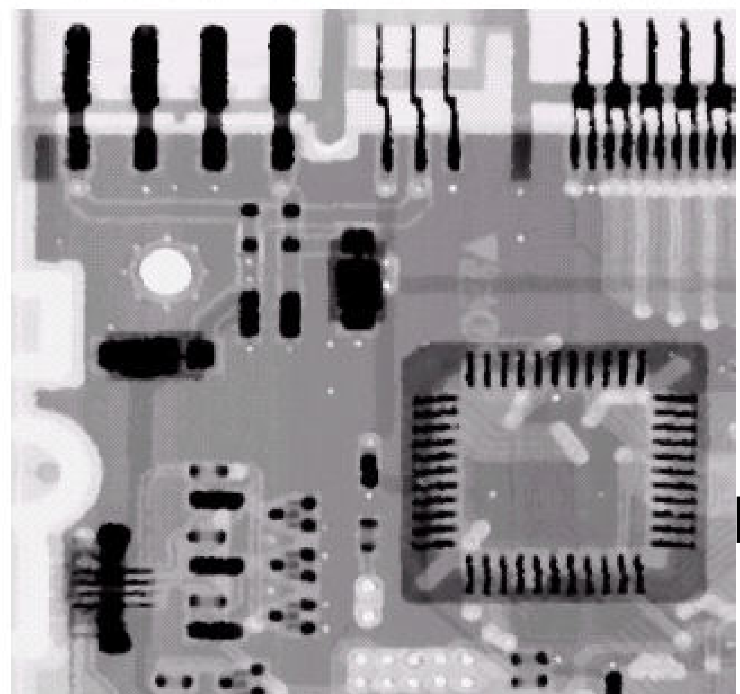




3x3 Median Filter Pass 1



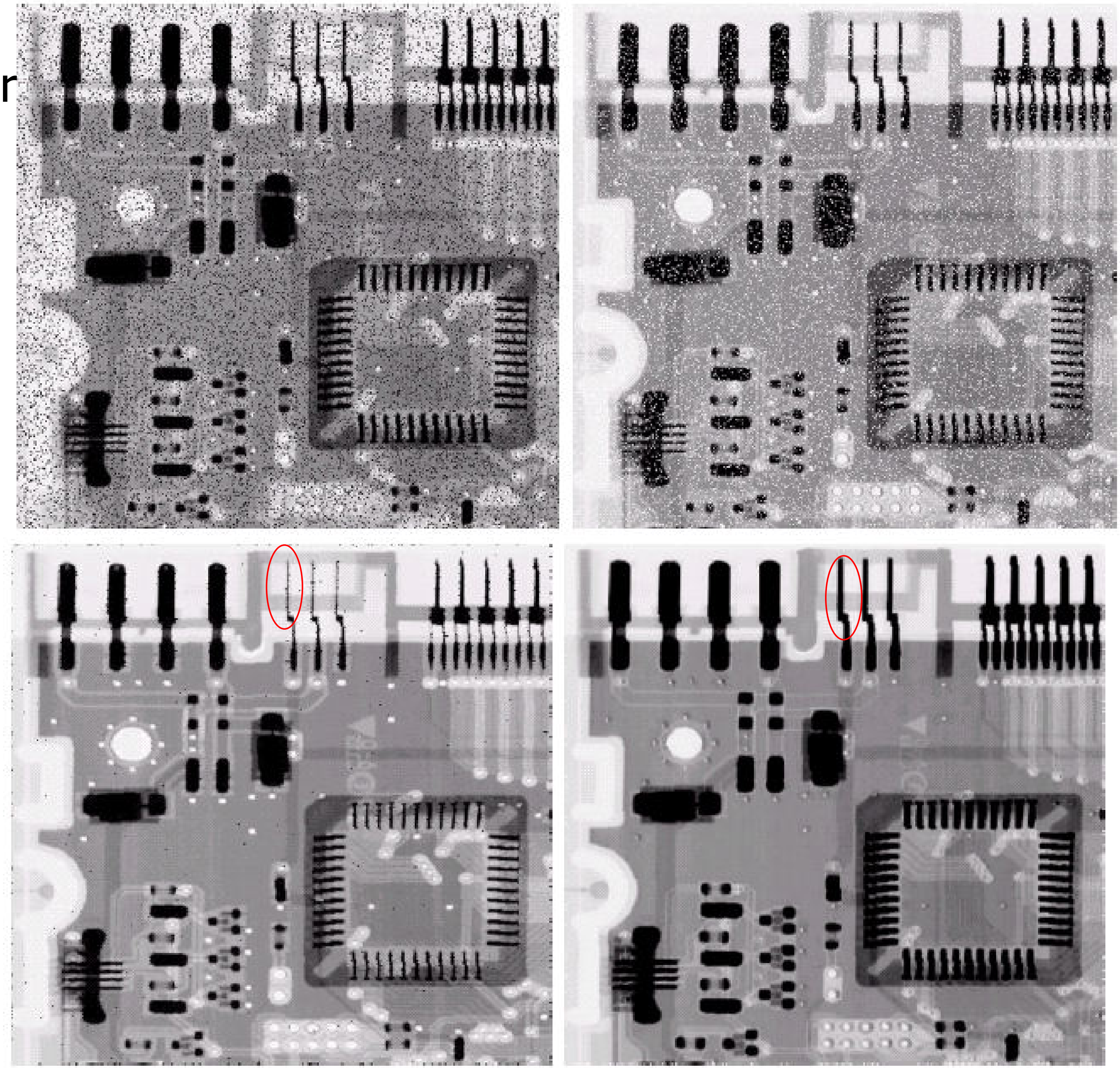
3x3 Median Filter Pass 2



3x3 Median Filter Pass 3

Pepper noise

Salt noise



Max filter

Min filter

Order-statistics filters (cont.)

- Midpoint filter

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

- Alpha-trimmed mean filter
  - Delete the d/2 lowest and d/2 highest gray-level pixels

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

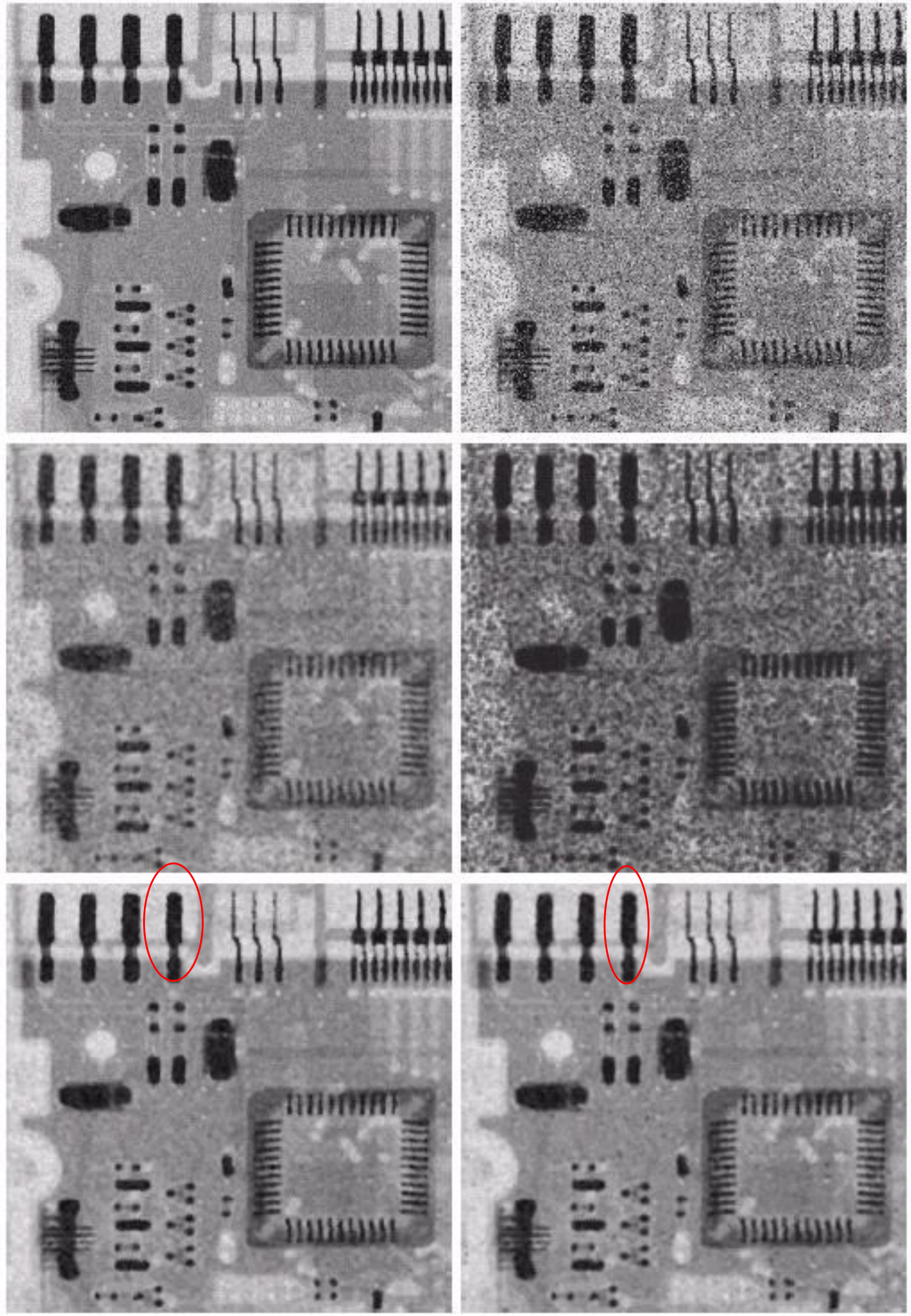
Middle (mn-d) pixels

Uniform noise

$\mu=0$

$\sigma^2=800$

Left +
Bipolar Noise
$P_a = 0.1$
$P_b = 0.1$



5x5
Arith. Mean
filter

5x5
Geometric
mean

5x5
Median
filter

5x5
Alpha-trim.
Filter
d=5

Adaptive filters

- Adapted to the behavior based on the statistical characteristics of the image inside the filter region S_{xy}
- Improved performance v.s increased complexity
- Example: Adaptive local noise reduction filter

Adaptive local noise reduction filter

- Simplest statistical measurement
  - Mean and variance
- Known parameters on local region S_{xy}
  - $g(x,y)$: noisy image pixel value
  - σ^2_η: noise variance (assume known a prior)
  - m_L : local mean
  - σ^2_L : local variance

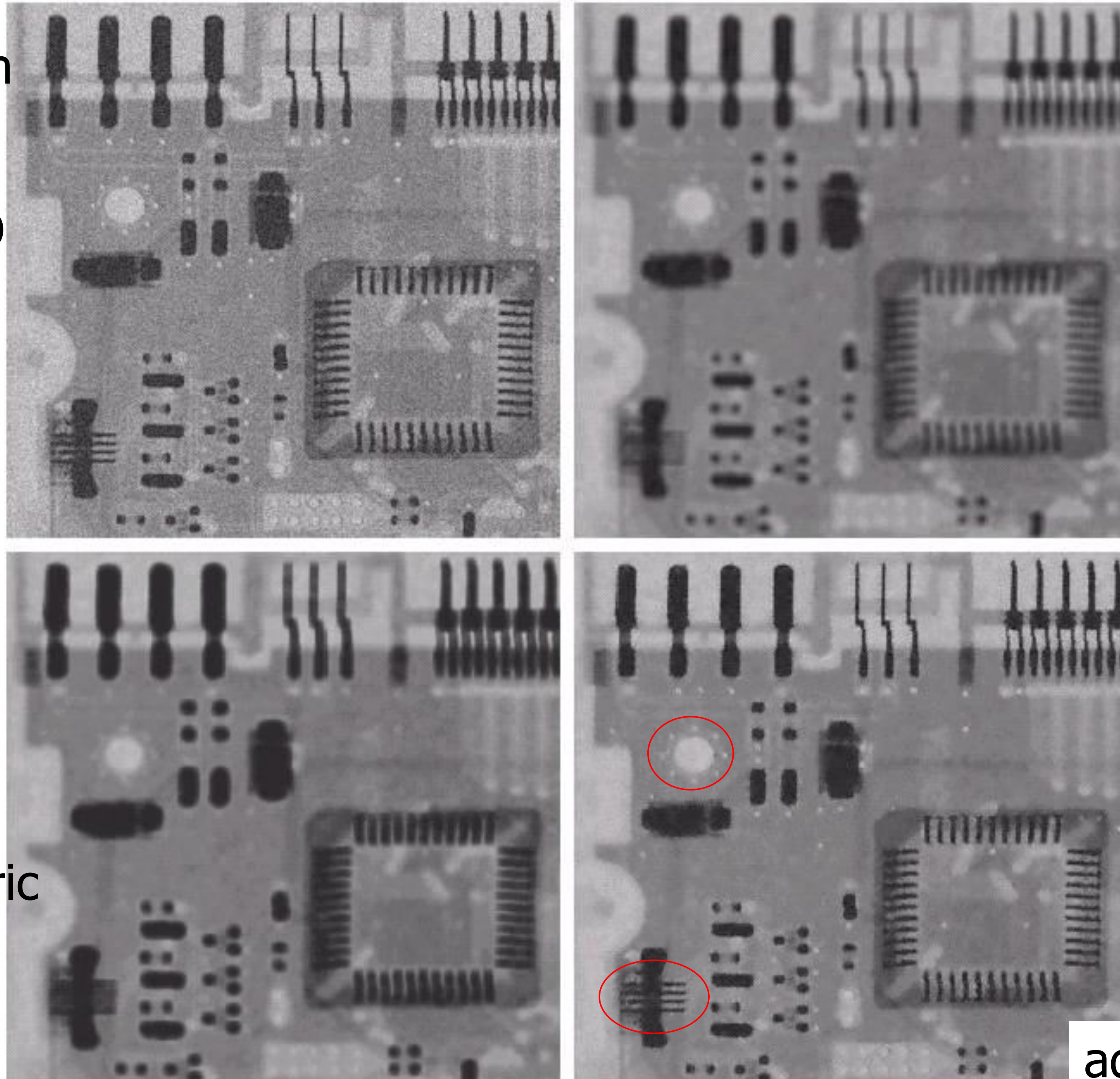
Adaptive local noise reduction filter (cont.)

- Analysis: we want to do
  - If σ^2_η is zero, return g(x,y)
  - If $\sigma^2_L > \sigma^2_\eta$, return value close to g(x,y)
  - If $\sigma^2_L = \sigma^2_\eta$, return the arithmetic mean m_L
- Formula

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

Gaussian noise
$\mu=0$
$\sigma^2=1000$

Arith. mean 7x7



Geometric mean 7x7

adaptive

Inverse filtering

- With the estimated degradation function H(u,v)

Unknown noise

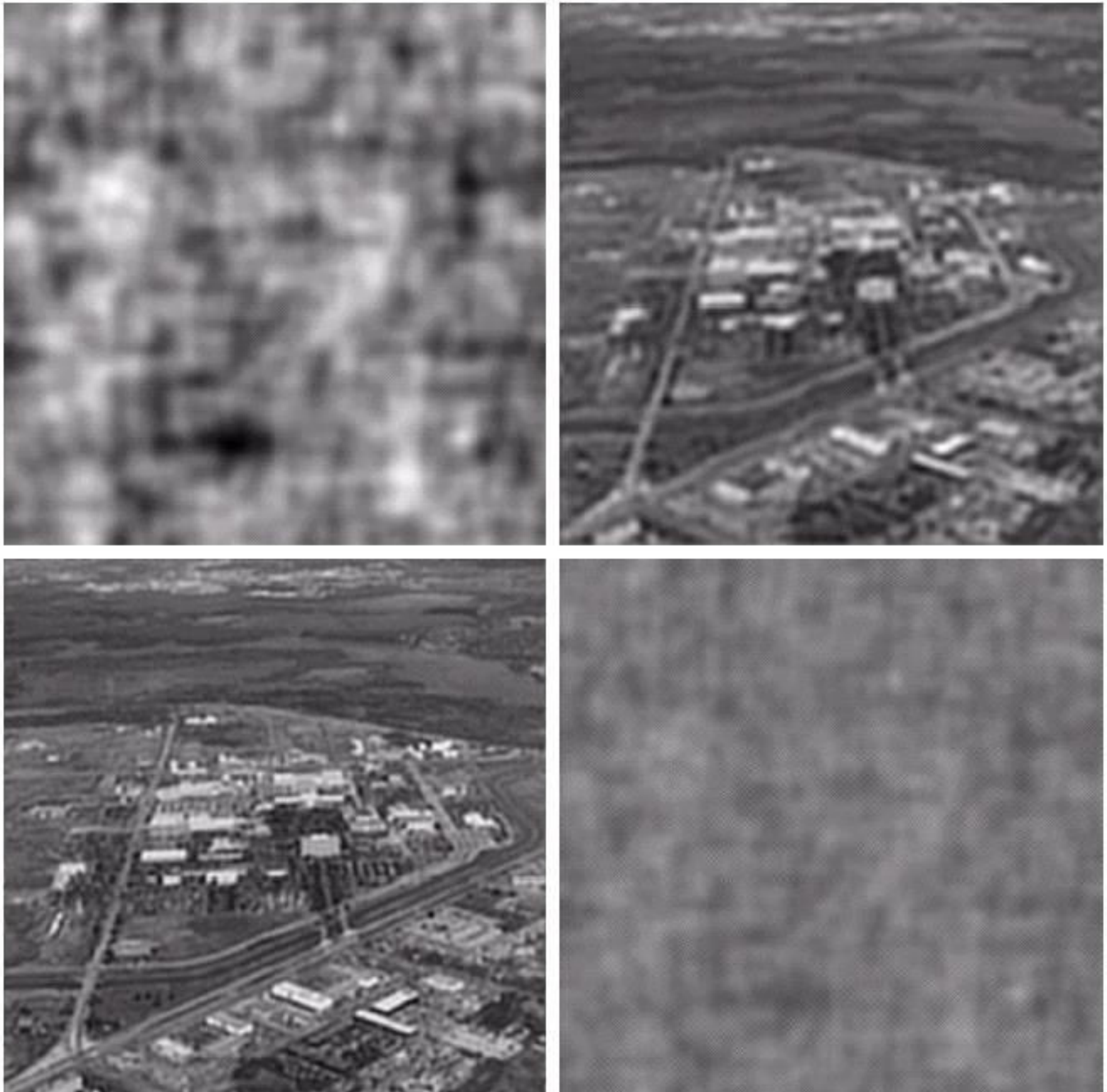
G(u,v)=F(u,v)H(u,v)+N(u,v)

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

Estimate of original image

Problem: 0 or small values

Sol: limit the frequency around the origin



Full inverse filter for k=0.0025

Cut Outside 40%

Cut Outside 70%

Cut Outside 85%