Chapter 3
Image Enhancement in the Spatial Domain - 1

3.0 Image enhancement in two domains

1. Spatial domain

   Spatial domain refers to the image itself. Based on direct manipulation of pixels in an image.

2. Frequency domain

   Based on modifying the Fourier transform of an image.

3.1 Spatial domain

- Spatial domain processes will be denoted by

$$g(x, y) = T\left[f(x, y) \right]$$

  - where *f*(*x*, *y*) is the input image,
    g(*x*, *y*) is the processed image,
    T is an operator on *f*, defined over some neighborhood of (*x*, *y*).
  - The size of "some neighborhood" is usually odd numbers.

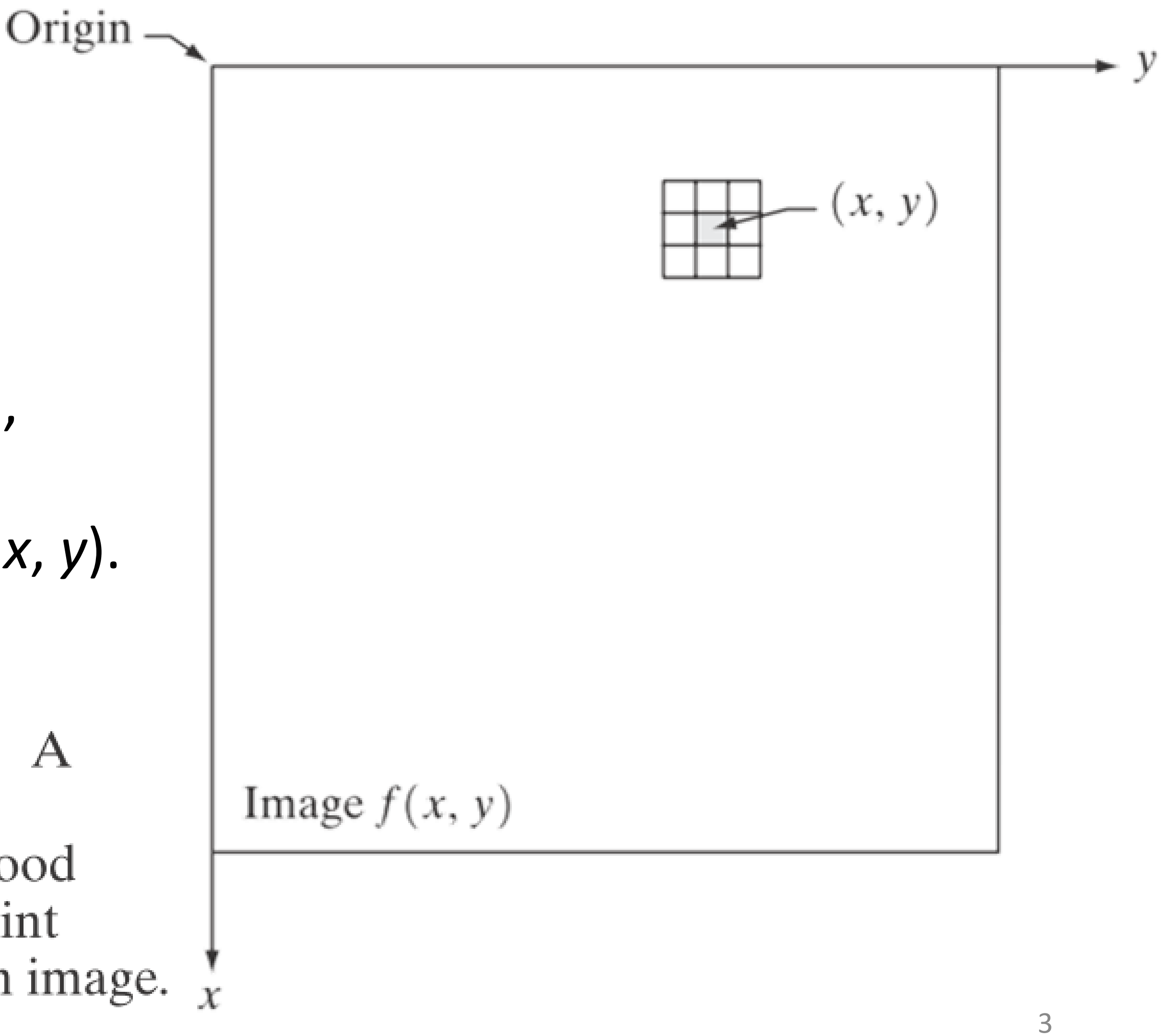


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

3.1 Spatial domain

- The simplest form of gray level transformation function *T* is when the neighborhood is of size 1*1.

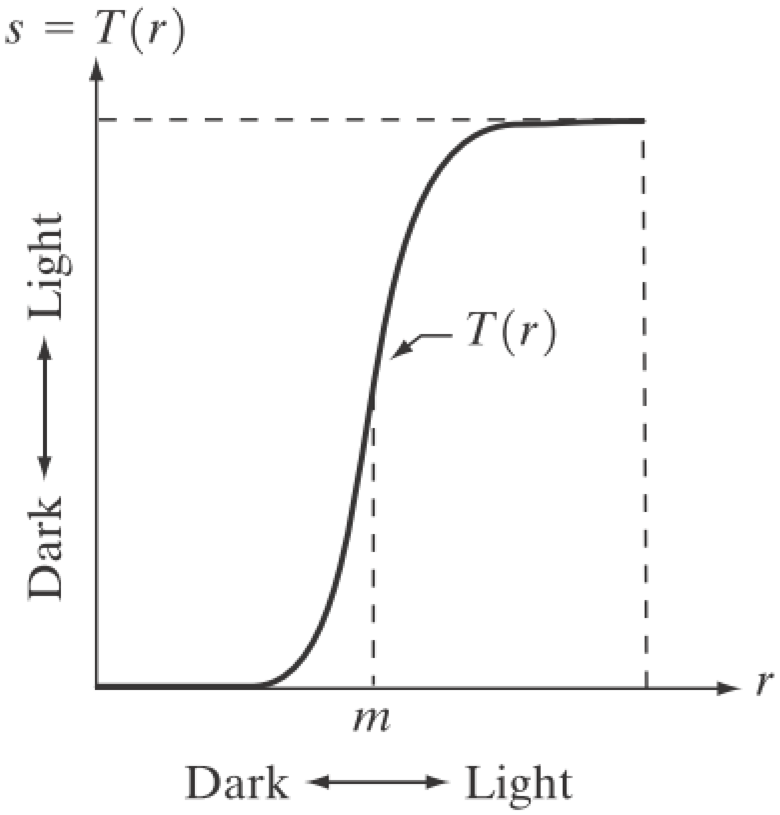
$$g(x,y) = T\left[f(x,y)\right] = c \cdot f(x,y)$$

  - where *c* is a scaling factor.

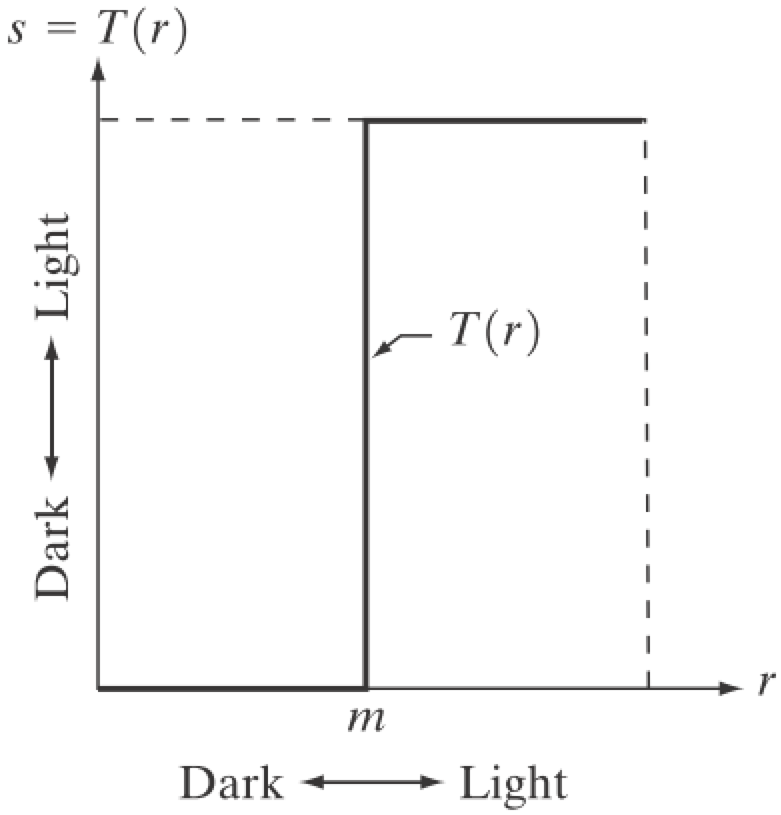
  - *T* becomes a gray-level (also called an intensity or mapping) transformation function

$$s = T(r)$$

3.1 Spatial domain



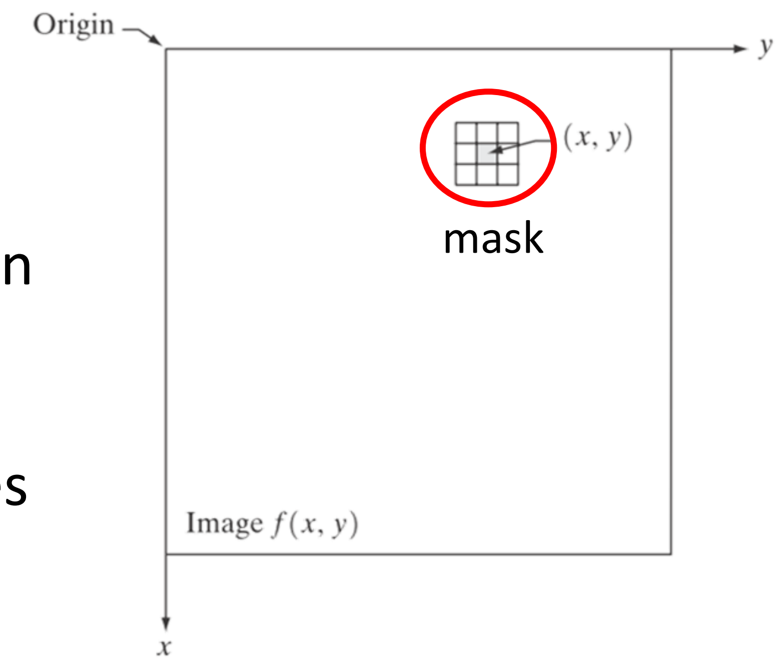
Contrast Stretching



Thresholding

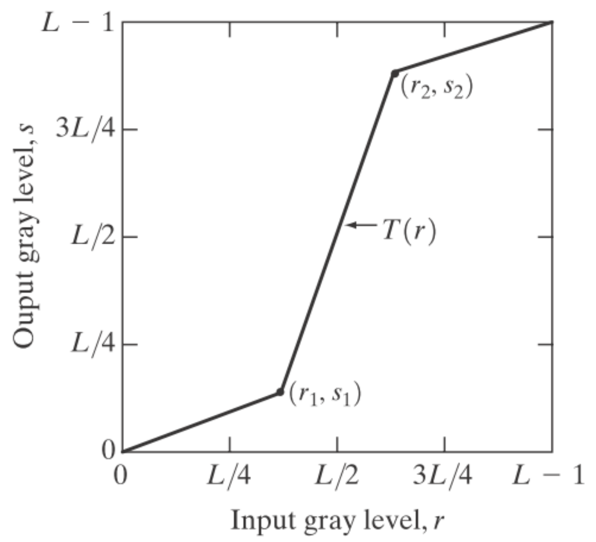
3.1 Spatial domain

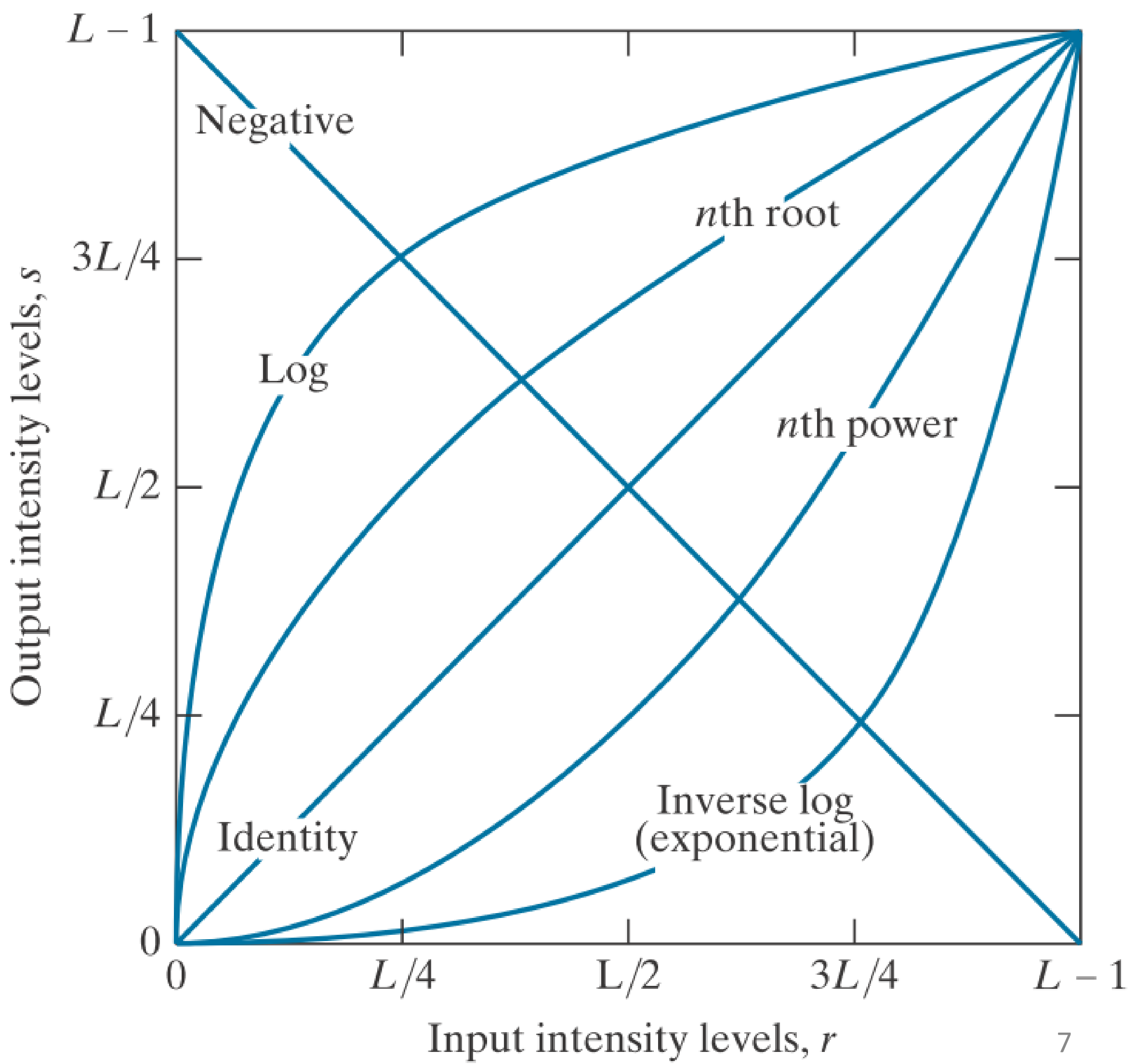
- Larger Neighborhood
  - Larger neighborhood allow considerably more flexibility
  - One of the principal approaches in this formulation is based on the use of masks (filters, kernels, templates, windows)
  - Enhancement techniques based on this techniques are referred to as mask processing or filtering



3.2 Some Basic Gray Level Transformations

- Image negatives
- Log transformation
- Power-Law transformation
- Piecewise-linear transformation

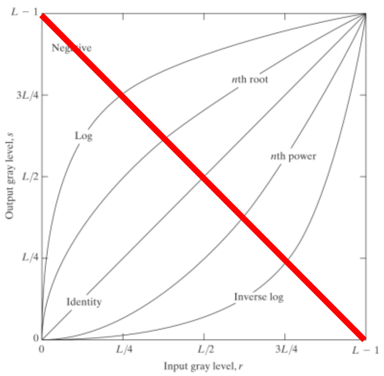


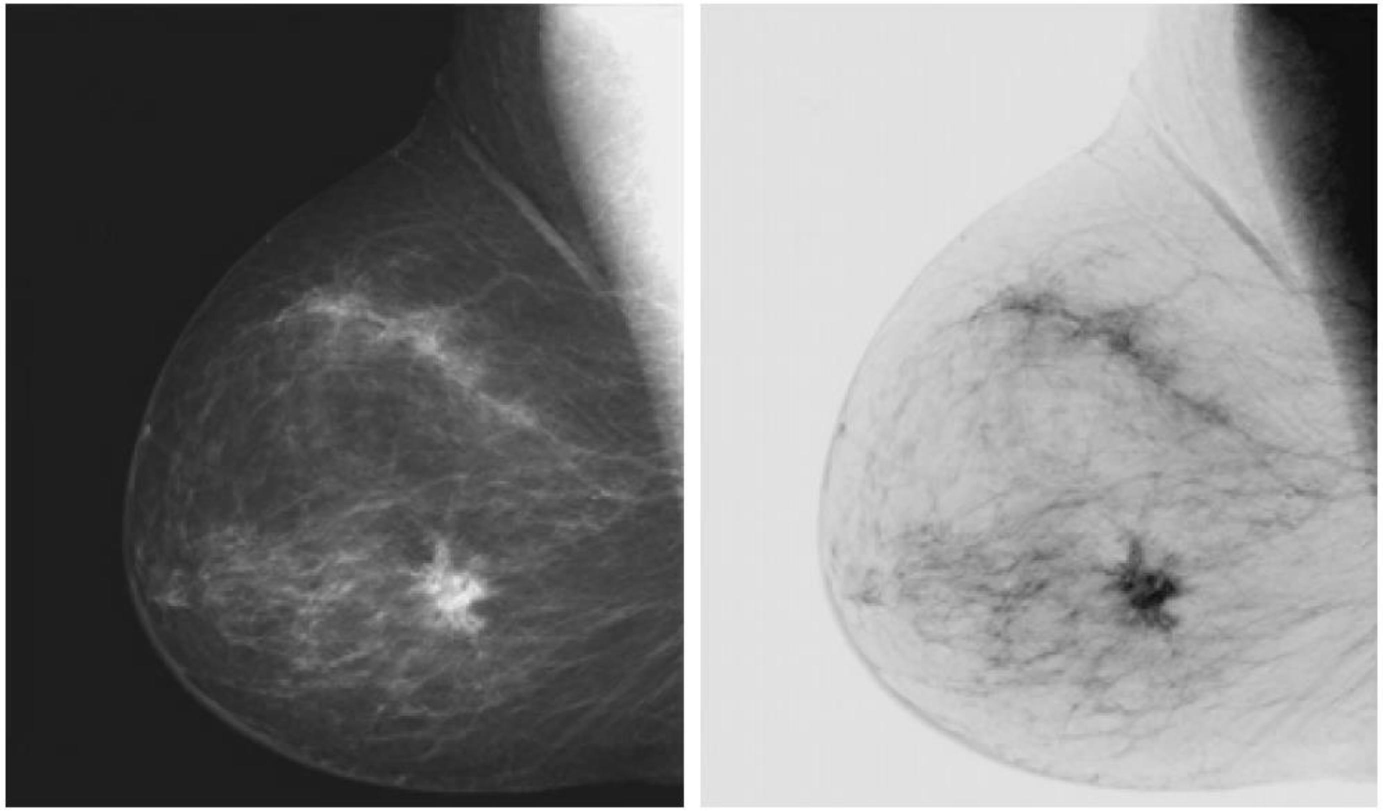


3.2 Some Basic Gray Level Transformations

- Image negatives
  - The negative of an image with gray levels in the range [0,*L*-1] is obtained by the following expression

$$s = L - 1 - r$$

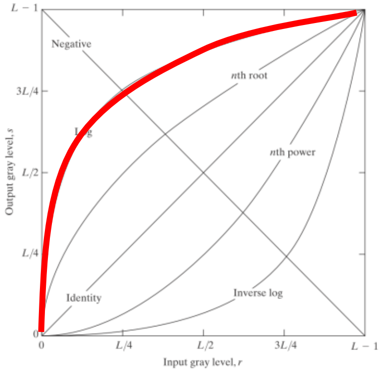




a b

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.)

3.2 Some Basic Gray Level Transformations

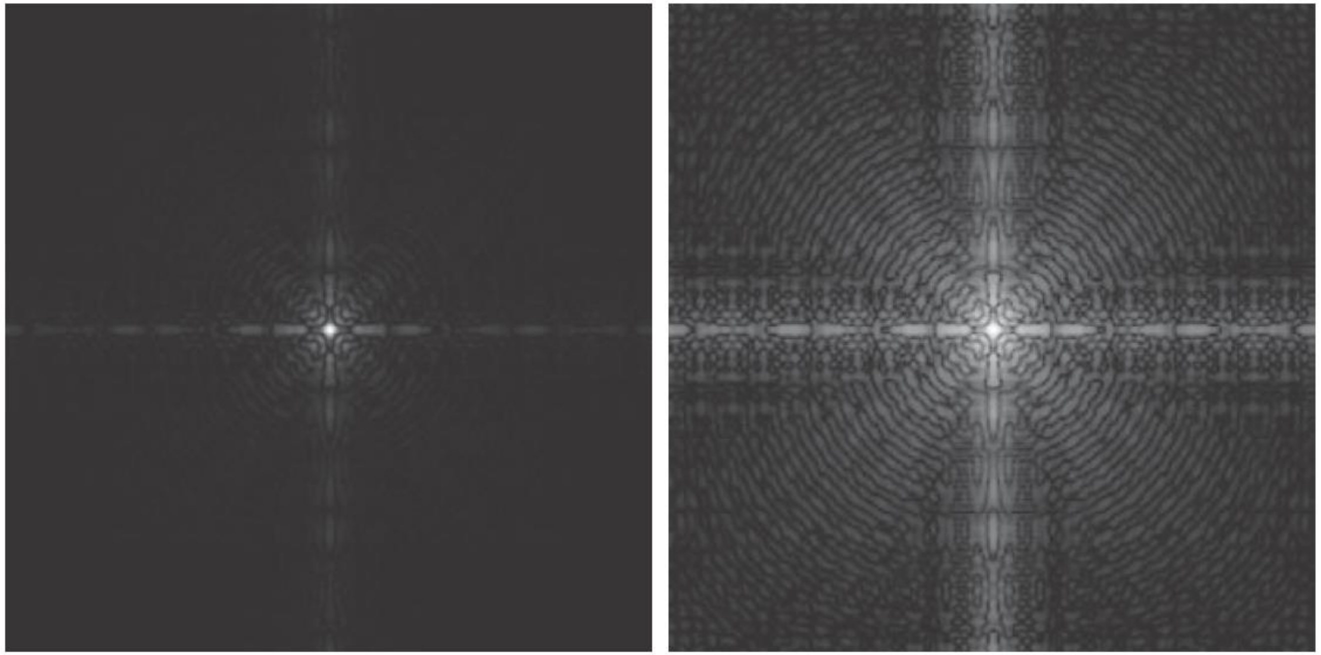


- Log transformation
  - The general form of log transformation is

    $$s = c\log(1+r)$$
  - where *c* is a constant, and *r* is assumed that r≥0
  - It compresses the dynamic range of images with large variations on pixel values

a b

FIGURE 3.5 (a) Fourier spectrum displayed as a grayscale image. (b) Result of applying the log transformation in Eq. (3-4) with $c = 1$. Both images are scaled to the range [0, 255].



[0, 1.5*10^6]

[0, 6.2]

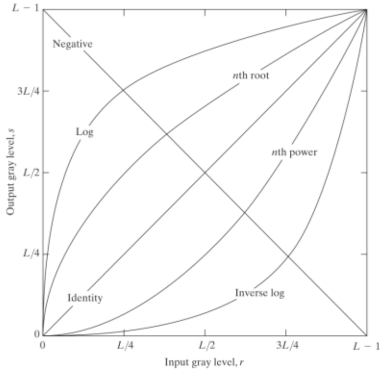
γ < 1

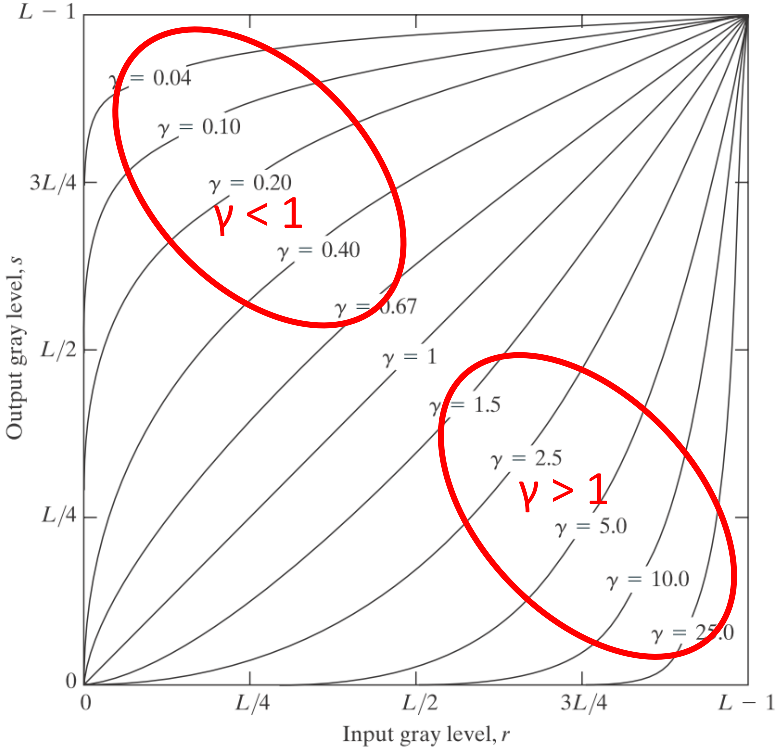
3.2 Some Basic Gray Level Transformations

- Power-Law transformation
  - The basic form of power-low transformation is

    $s = cr^{\gamma}$ or $s = c\left(r+\varepsilon\right)^{\gamma}$ a measurable output when the input is zero

  - Where *c*, γ and ε are positive constants.





3.2 Some Basic Gray Level Transformations

- Power-Law transformation

$\gamma = 0.6$ $\gamma = 0.4$ $\gamma = 0.3$

3.2 Some Basic Gray Level Transformations

- Power-Law transformation

γ = 3

γ = 4

γ = 5

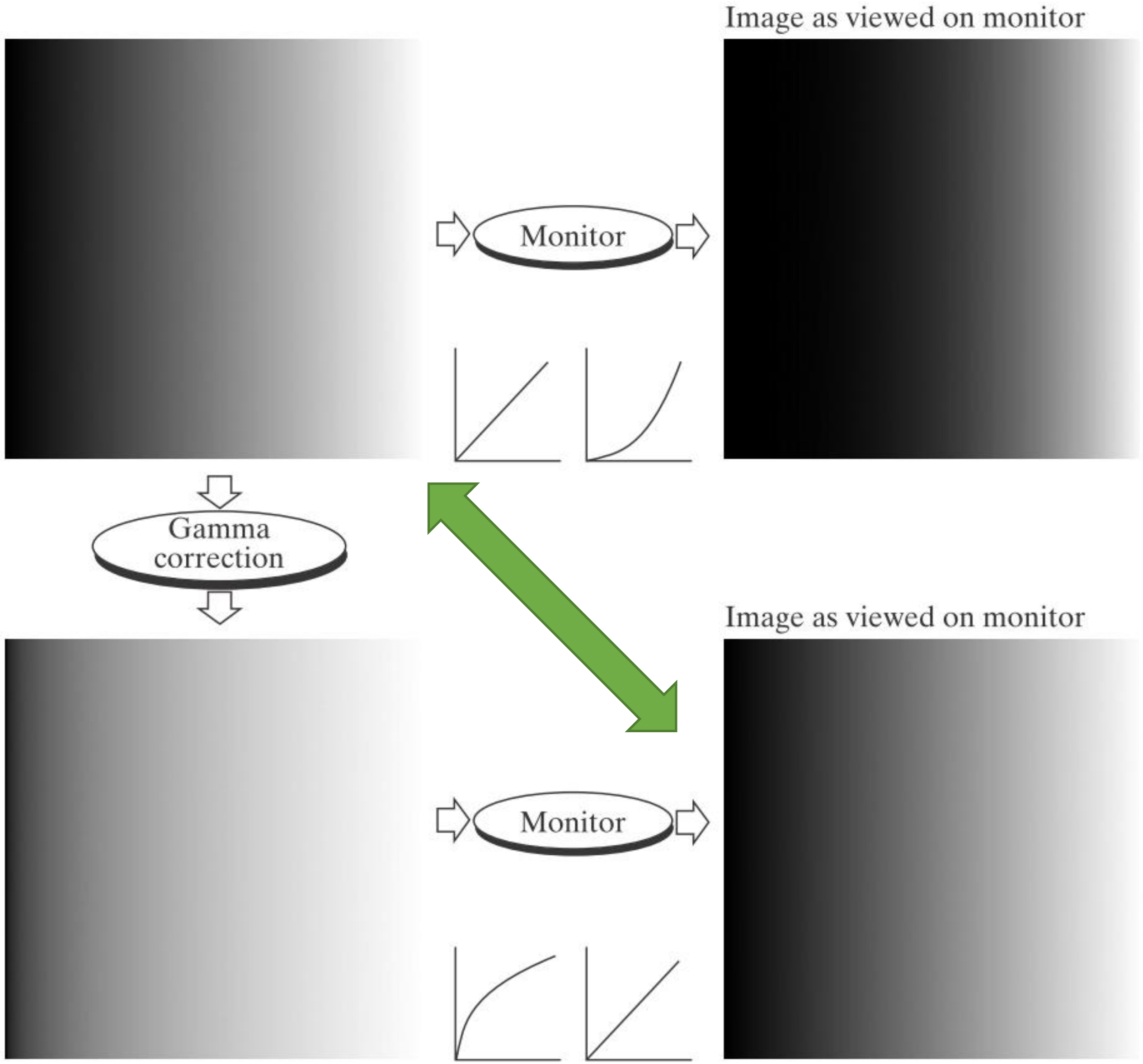
3.2 Some Basic Gray Level Transformations

- Gamma correction
  - A variety of devices used for image capture, printing, and display respond according to a power law.
  - The process to correct the power-law response phenomena is called gamma correction.
  - Ex: CRT
    - intensity-to-voltage response is a power function. (γ=1.8 to 2.5)
    - => Gamma correction is performed by inverse power function:

$$s = r^{1/2.5} = r^{0.4}$$

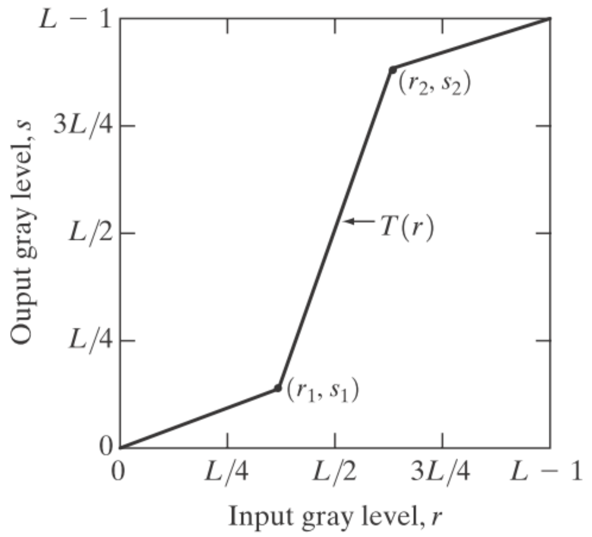
3.2 Some Basic Gray Level Transformations

- Gamma correction



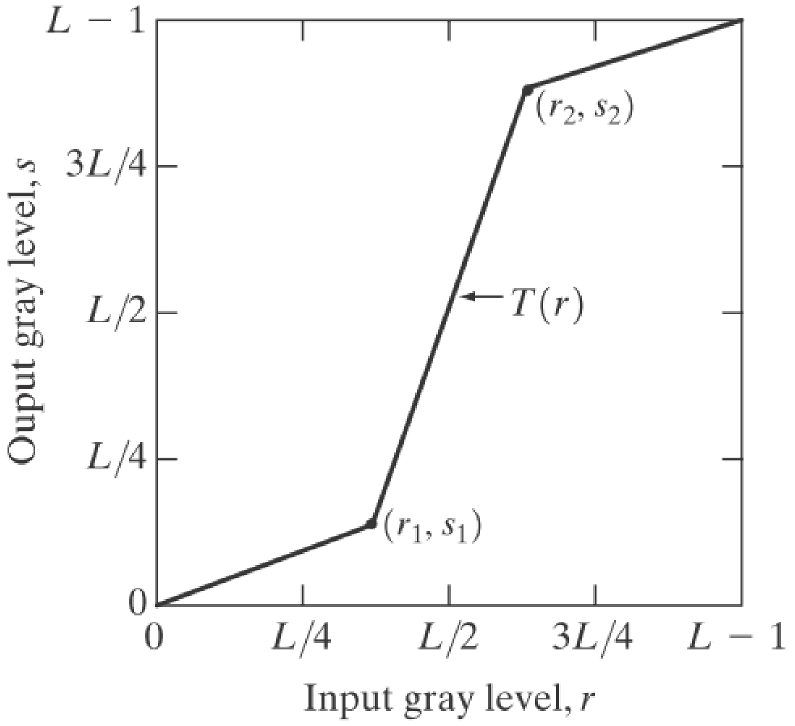
3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing



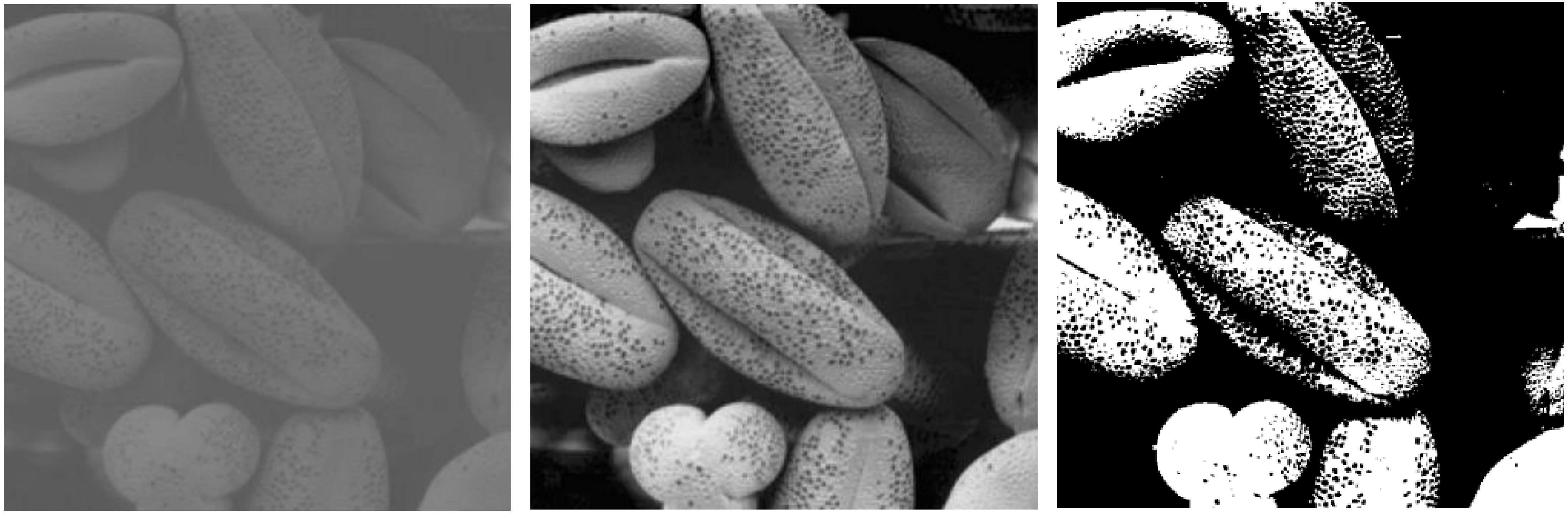
3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Contrast stretching: increasing the dynamic range of the gray-level
  - (r_1, s_1) and (r_2, s_2) control the transformation
    - $r_1 = s_1$ and $r_2 = s_2$: linear transformation with no gray-level changes
    - $r_1 = r_2$, $s_1 = 0$ and $s_2 = L$-1: thresholding function
    - Intermediate values of (r_1, s_1) and (r_2, s_2) :various degrees of spread in the gray-levels



3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Contrast stretching:



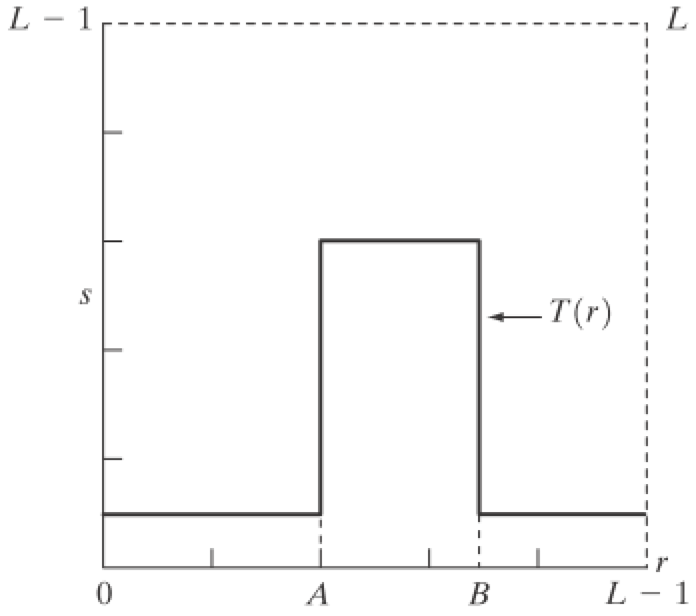
$(r_1, s_1) = (r_{min}, 0)$
$(r_2, s_2) = (r_{max}, L\text{-}1)$

$r_1 = r_2$ = mean gray value

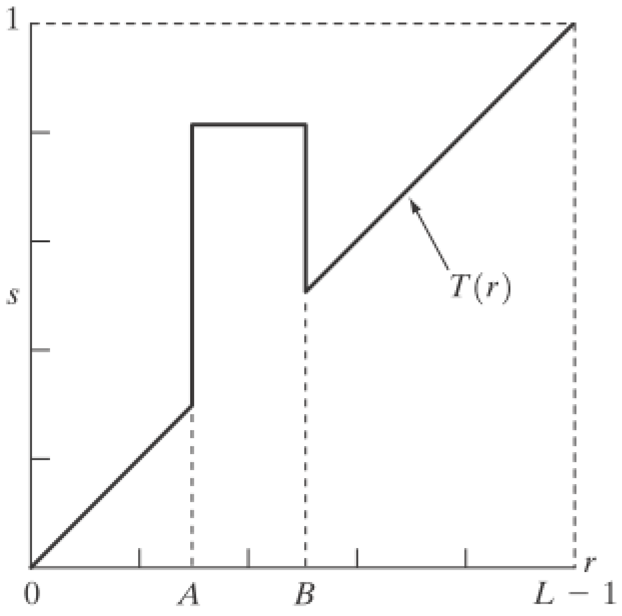
r_{min} and r_{max} are the maximal and minimal grey values of the image

3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Gray-level slicing: Highlighting a specific range of gray levels in an image.



Highlight range [A,B] and reduce all others to a constant level

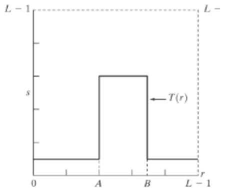


Highlight range [A,B] but preserve all other levels

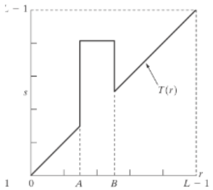
3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Gray-level slicing: Highlighting a specific range of gray levels in an image.

Before transformation

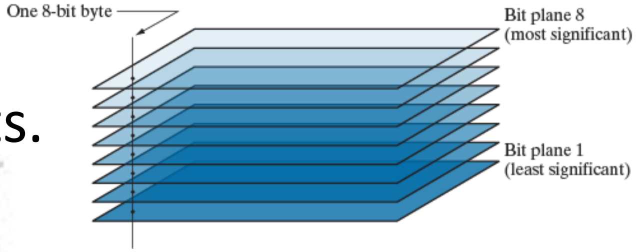


After transformation



3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Bit-plane slicing: highlighting the contributions of specific bits.



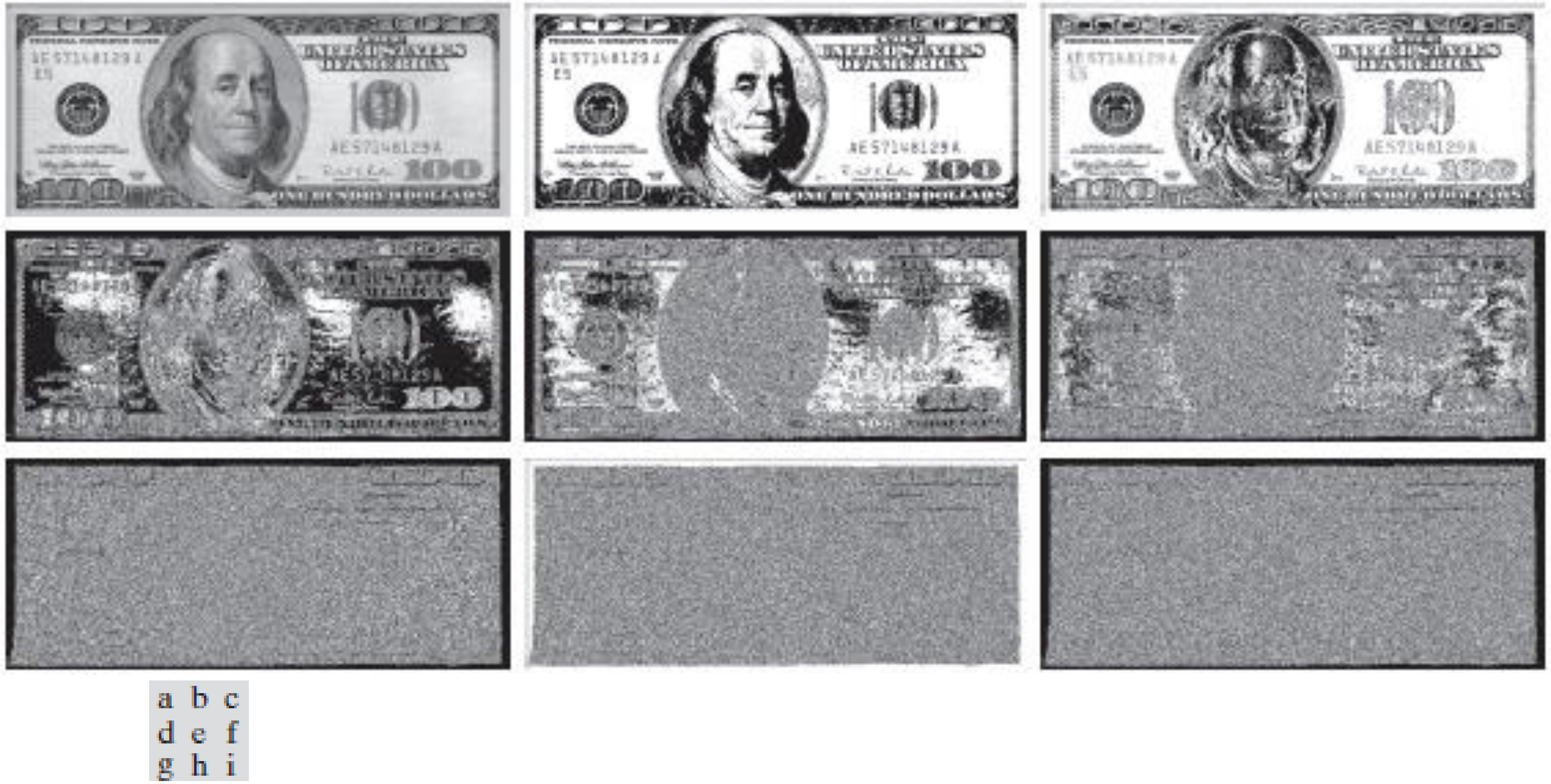


FIGURE 3.14 (a) An 8-bit gray-scale image of size 550×1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

3.2 Some Basic Gray Level Transformations

- Piecewise-linear transformation
  - Bit-plane slicing: highlighting the contributions of specific bits.

a b c **FIGURE 3.15** Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5.

3.3 Histogram Processing

- The histogram of a digital image with gray levels in the range [0, *L*-1] is a discrete function.

$$h(r_k) = n_k$$

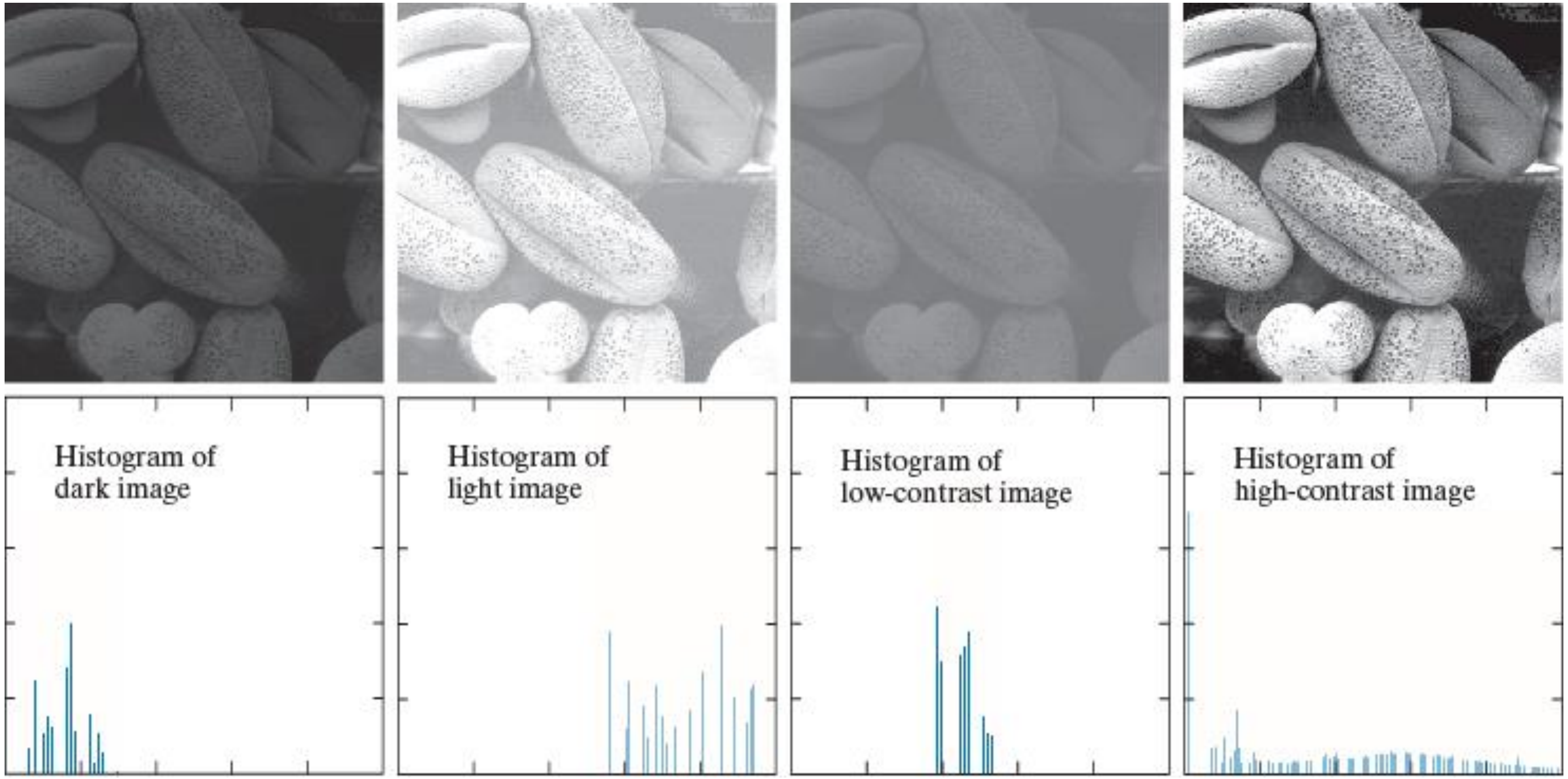
  - where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k.

- A normalized histogram is given by

$$h_n(r_k) = \frac{n_k}{N}, \text{ and } \sum_{k=0}^{L-1} h_n(r_k) = 1$$

  - where N is the total pixel count in the image.

3.3 Histogram Processing



a b c d

FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

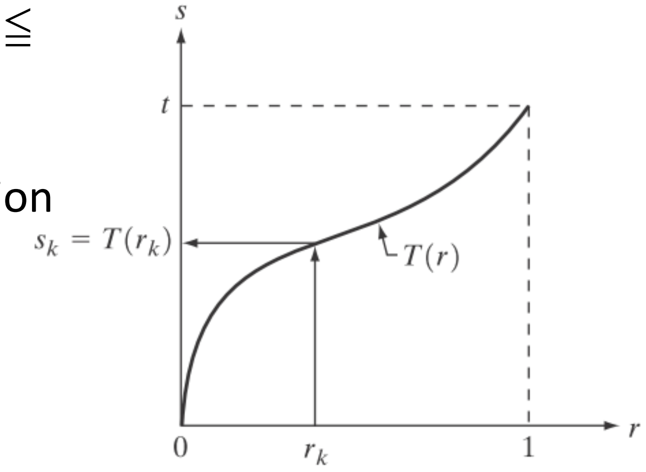
3.3 Histogram Processing

- Histogram Equalization:
  - Let the variable *r* represent the gray levels of the image to be enhanced.
  - *r* is normalized to [0, 1], where 0 is black, and 1 is white.
  - For any *r* satisfying the aforementioned conditions, we focus attention on transformations of the form

$$s = T(r),\ 0 \le r \le 1$$

- The transformation function *T*(*r*) satisfies the following conditions:
  a) *T*(*r*) is single-valued and monotonically increasing in the interval $0 \leqq r \leqq 1$; and
  b) $0 \leqq T(r) \leqq 1$ for all *r.*

  The requirement in *a*) is needed to guarantee that the inverse transformation will exist.



- The inverse transformation from *s* back to *r* is denoted

$$r = T^{-1}(s),\ 0 \le s \le 1$$

3.3 Histogram Processing

- Histogram Equalization:
  - The gray levels in an image may be viewed as random variables in the interval [0, 1]. Let $p_r(r)$ and $p_s(s)$ denote the probability density functions (PDF) of random variables r and s, respectively.
  - A basic result from an elementary probability theory is that, if $p_r(r)$ and $T(r)$ are known and satisfies condition a), the probability density function p_s(s) of the transformed variable s can be obtained using a rather simple formula:

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \quad .....(3.3\text{-}3)$$

3.3 Histogram Processing

- Histogram Equalization:
  - A transformation function of particular importance in image processing has the form

$$s = T(r) = \underline{\int_0^r p_r(w)dw} \quad \text{.....(3.3-4)}$$

Cumulative Probability Function (CPF) of random variable *r*, which satisfies both condition a) and b).

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = p_r(r) \quad \text{.....(3.3-5) (Leibniz's rule)}$$

substituting this result for *dr*/*ds* into Eq. (3.3-3),

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1, \text{ for } 0 \le s \le 1 \quad \text{.....(3.3-6)}$$

- The form of $p_s(s)$ given in Eq. (3.3-6) as a uniform probability density function.

3.3 Histogram Processing

- Histogram Equalization:
  - For discrete values we deal with probabilities and summations instead of probability density functions and integrals. The probability of occurrence of gray level r_k in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n}, k = 0, 1, 2, \ldots, L-1 \quad \text{.....(3.3-7)}$$

  - Where n is the total number of pixels in the image, n_k is the number of pixels that have gray level r_k, and L is the total number of possible gray levels in the image.

3.3 Histogram Processing

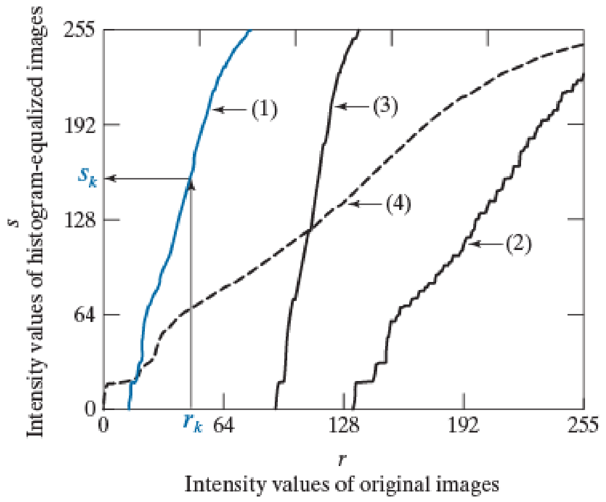
- Histogram Equalization:
  - The discrete version of the transformation function given in Eq. (3.3-4) is

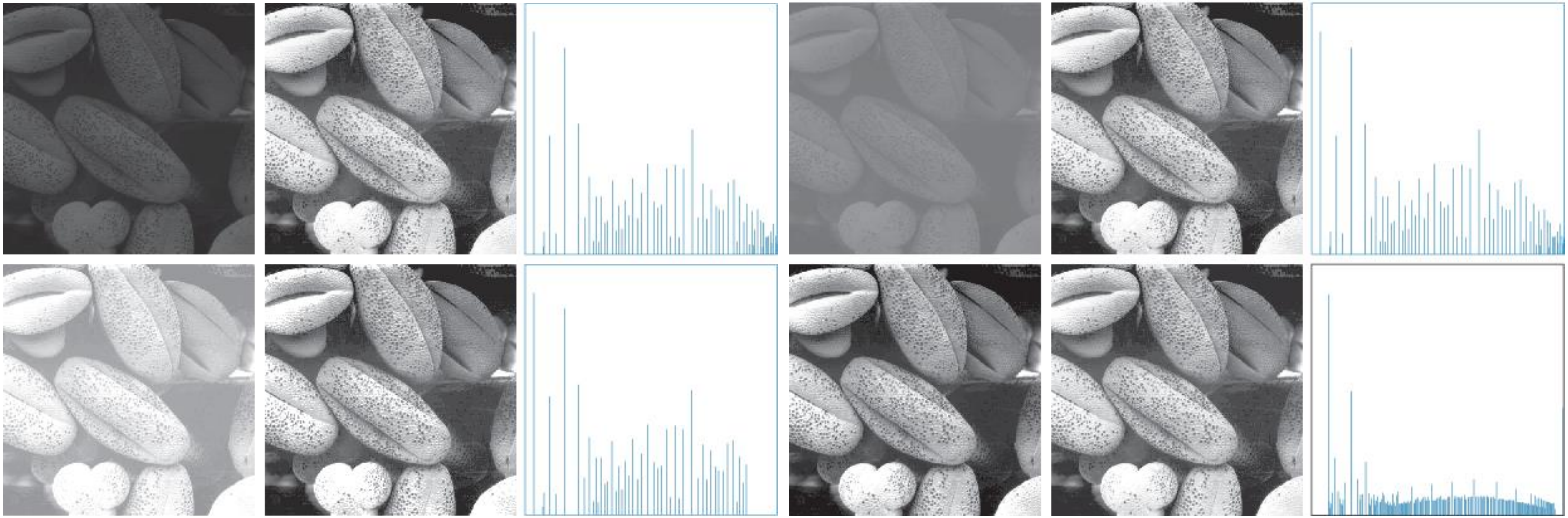
$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, k = 0, 1, 2, \ldots, L-1 \quad \text{.....(3.3-8)}$$

  - As indicated earlier, a plot of $p_r(r_k)$ versus r_k is called a histogram. The transformation (mapping) given in Eq. (3.3-8) is called histogram equalization or histogram linearization.

3.3 Histogram Processing

- Histogram Equalization:





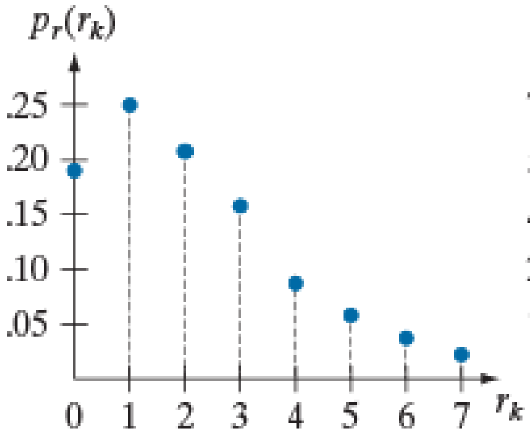
3.3 Histogram Processing

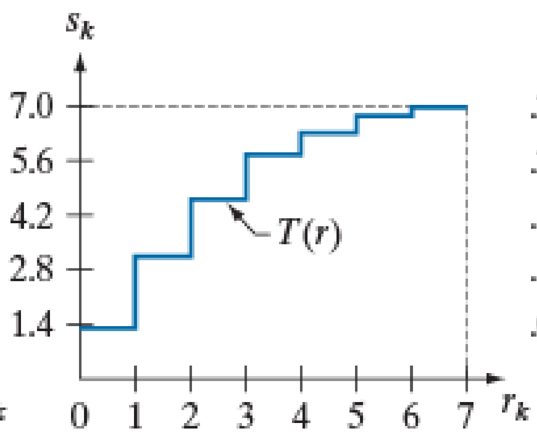
- Histogram Equalization: 3-bit image example
  - MN=4096, L=8

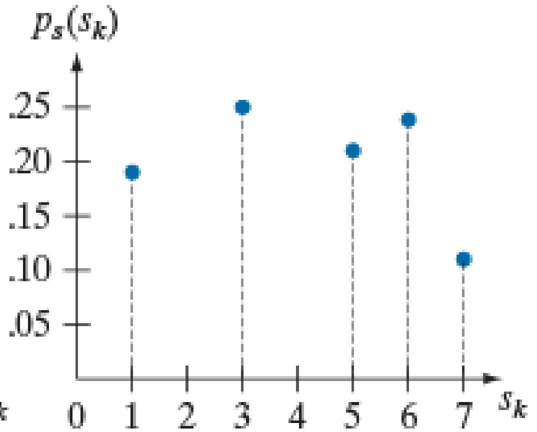
TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64 × 64 digital image.

r_k	n_k	$p_r(r_k) = n_k/MN$	s_k	s_k*(L-1)
$r_0 = 0$	790	0.19	0.19	1.33=1
$r_1 = 1$	1023	0.25	0.44	3.08=3
$r_2 = 2$	850	0.21	0.65	4.55=5
$r_3 = 3$	656	0.16	0.81	5.67=6
$r_4 = 4$	329	0.08	0.89	6.23=6
$r_5 = 5$	245	0.06	0.95	6.65=7
$r_6 = 6$	122	0.03	0.98	6.86=7
$r_7 = 7$	81	0.02	1.00	7.00=7

r_k	s_k
0	1
1	3
2	5
3	6
4	6
5	7
6	7
7	7







3.3 Histogram Processing

TABLE 10.2-1. Histogram Modification Transfer Functions

Output Probability Density Model		Transfer Function[a]
Uniform	$p_g(g) = \dfrac{1}{g_{max} - g_{min}} \quad g_{min} \le g \le g_{max}$	$g = (g_{max} - g_{min})P_f(f) + g_{min}$
Exponential	$p_g(g) = \alpha \exp\{-\alpha(g - g_{min})\} \quad g \le g_{min}$	$g = g_{min} - \dfrac{1}{\alpha}\ln\{1 - P_f(f)\}$
Rayleigh	$p_g(g) = \dfrac{g - g_{min}}{\alpha^2}\exp\left\{-\dfrac{(g - g_{min})^2}{2\alpha^2}\right\} \quad g \ge g_{min}$	$g = g_{min} + \left[2\alpha^2 \ln\left\{\dfrac{1}{1 - P_f(f)}\right\}\right]^{1/2}$
Hyperbolic (Cube root)	$p_g(g) = \dfrac{1}{3}\dfrac{g^{-2/3}}{g_{max}^{1/3} - g_{min}^{1/3}}$	$g = \left[g_{max}^{1/3} - g_{min}^{1/3}[P_f(f)] + g_{max}^{1/3}\right]^3$
Hyperbolic (Logarithmic)	$p_g(g) = \dfrac{1}{g[\ln\{g_{max}\} - \ln\{g_{min}\}]}$	$g = g_{min}\left(\dfrac{g_{max}}{g_{min}}\right)^{P_f(f)}$

[a]The cumulative probability distribution $P_f(f)$, of the input image is approximated by its cumulative histogram:

$$p_f(f) \approx \sum_{m=0}^{j} H_F(m)$$

3.3 Histogram Processing

- Histogram Matching (Specification):
  - When automatic enhancement is desired, histogram equalization is a good approach because the results from this technique are predictable and the method is simple to implement.
  - There are applications in which attempting to base enhancement on a uniform histogram is not the best approach.
  - It is useful sometimes to be able to specify the shape of the histogram.
  - The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - let $p_r(r)$ and $p_z(z)$ denote their corresponding continuous probability density functions. In this notation, r and z denote the gray levels of the input and output (processed) images, respectively.
  - $p_r(r)$ is estimated from the given input image, while $p_z(z)$ is the specified probability density function that we wish the output image to have.

$$s = T(r) = \int_0^r p_r(w)dw$$ .....(3.3-10) which is the same as (3.3.4)

  - we define a random variable z with the property:

$$G(z) = \int_0^z p_z(t)dt = s$$ .....(3.3-11)

3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - From $s = T(r) = \int_0^r p_r(w)dw$ (3.3.10) and $G(z) = \int_0^z p_z(t)dt = s$ (3.3.11):

    $z = G^{-1}(s) = G^{-1}\left[T(r)\right]$ .....(3.3-12)

  - The transformation $T(r)$ can be obtained from Eq. (3.3-10) once $p_r(r)$ has been estimated from the input image.
  - The transformation function $G(z)$ can be obtained using Eq. (3.3-11) because $p_z(z)$ is given.

3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - Assume G^{-1} exists and that it satisfies conditions a) and b) in the previous section.
  - An image with a specified probability density function can be obtained from an input image by using the following procedure:
    - 1) Obtain the transformation function $T(r)$ using Eq. (3.3-10).
    - 2) Use Eq. (3.3-11) to obtain the transformation function $G(z)$.
    - 3) Obtain the inverse transformation function G^{-1} .
    - 4) Obtain the output image by applying Eq. (3.3-12) to all the pixels in the input image.
  - The result of this procedure will be an image whose gray levels, z, have the specified probability density function $p_z(z)$.

3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - The discrete formulation of Eq. (3.3-10) is given by Eq. (3.3-8), which we repeat here for convenience:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, k = 0,1,2,\ldots, L-1 \quad \text{.....(3.3-13)}$$

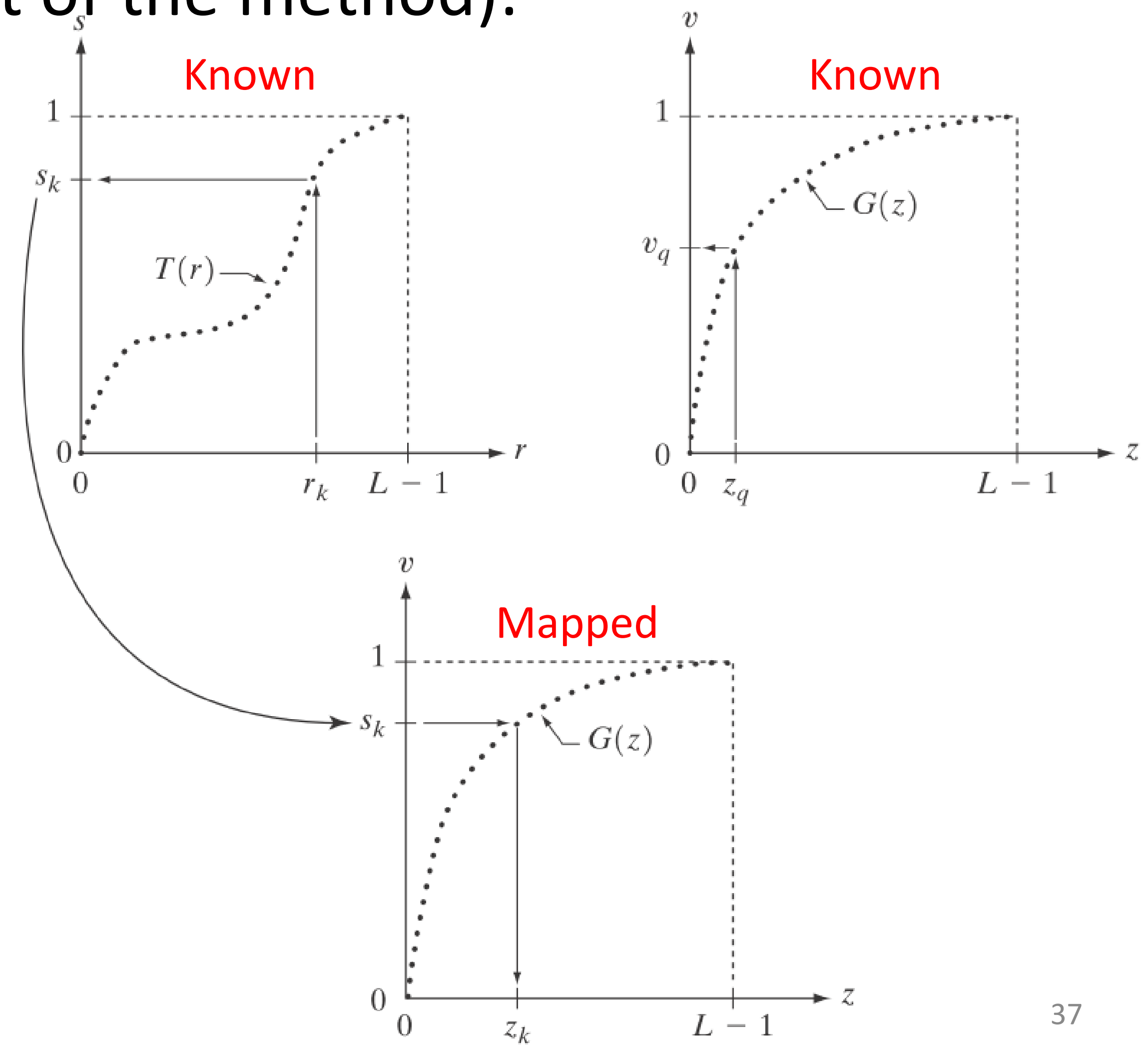
where n is the total number of pixels in the image, n_j is the number of pixels with gray level r_j, and L is the number of discrete gray levels.

- Similarly, the discrete formulation of $G(z) = \int_0^z p_z(t)dt = s$ (3.3-11) is obtained:

$$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k, k = 0,1,2,\ldots, L-1 \quad \text{.....(3.3-14)}$$

3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - Finally, the discrete version of
    $$z = G^{-1}(s) = G^{-1}\left[T(r)\right] \quad (3.3\text{-}12)$$
    is given by:
    $$z_k = G^{-1}\left[T(r_k)\right] = G^{-1}(s_k) \quad .....(3.3\text{-}16)$$



3.3 Histogram Processing

- Histogram Matching (Development of the method):
  - The procedure we have just developed for histogram matching may be summarized as follows:
    1. Obtain the histogram of the given image.
    2. Use Eq. (3.3-13) to precompute a mapped level s_k for each level r_k. LUT(r_k)=s_k
    3. Obtain the transformation function G from the given $P_z(z)$ using Eq. (3.3-14).
    4. Precompute z_k for each value of s_k using the iterative scheme defined in connection with Eq. (3.3-16). LUT(s_k)=z_k
    5. For each pixel in the original image, if the value of that pixel is r_k, map this value to its corresponding level s_k; then map level s_k into the final level z_k by using the precomputed values from Steps (2) and (4) for these mappings.

3.3 Histogram Processing

- Histogram Matching: 3-bit image as example
  - MN=4096, L=8

a b
c d

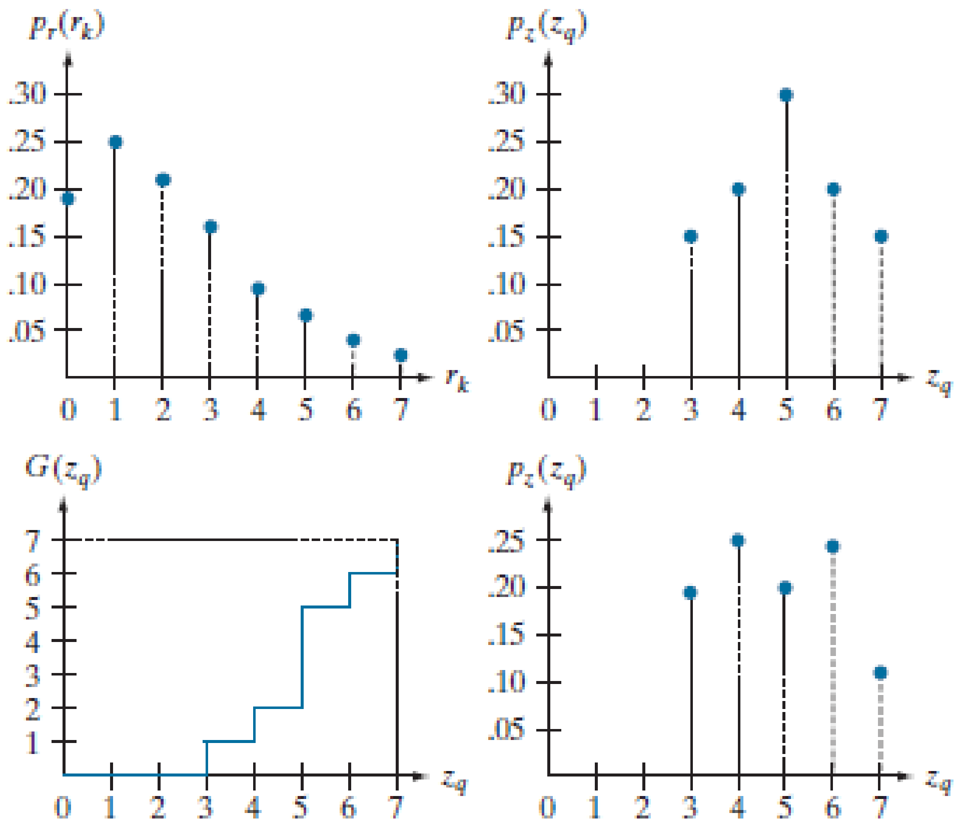


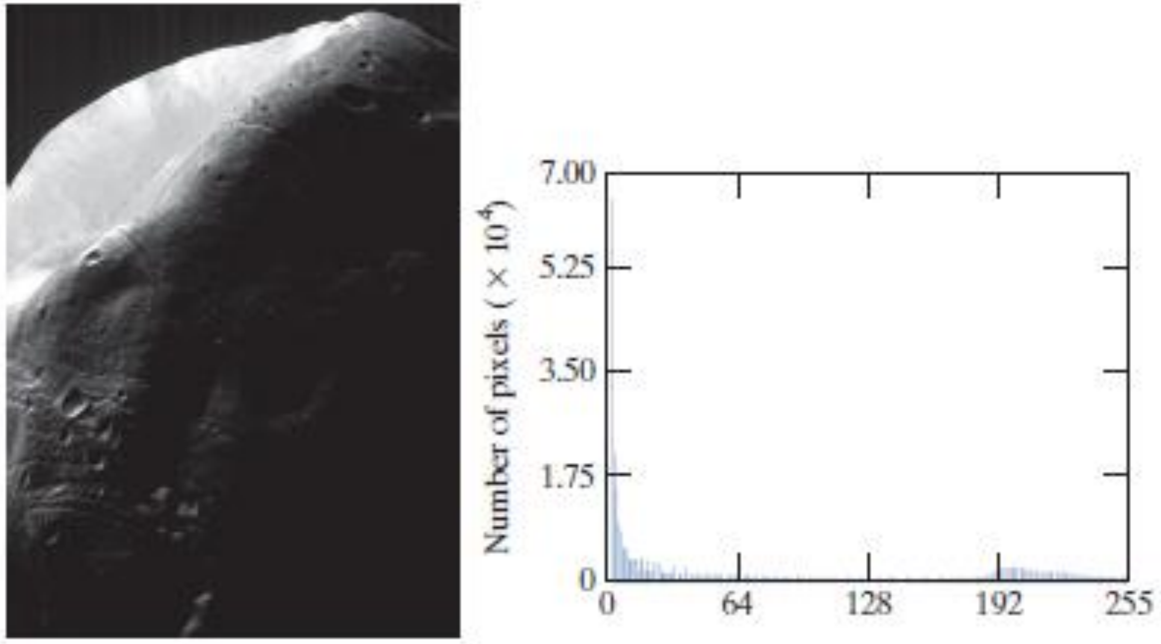
FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of histogram specification. Compare the histograms in (b) and (d).

z_q	Specified $p_z(z_q)$	$G(z_k)$	$G(z_k)*(L\text{-}1)$
$z_0 = 0$	0.00	0.0	0.0=0
$z_1 = 1$	0.00	0.0	0.0=0
$z_2 = 2$	0.00	0.0	0.0=0
$z_3 = 3$	0.15	0.15	1.05=1
$z_4 = 4$	0.20	0.35	2.45=2
$z_5 = 5$	0.30	0.65	4.55=5
$z_6 = 6$	0.20	0.85	5.95=6
$z_7 = 7$	0.15	1.00	7.00=7

r_k	s_k	z_k	Actual $p_z(z_q)$
0	1	3	0.00
1	3	4	0.00
2	5	5	0.00
3	6	6	0.19
4	6	6	0.25
5	7	7	0.21
6	7	7	0.24
7	7	7	0.11

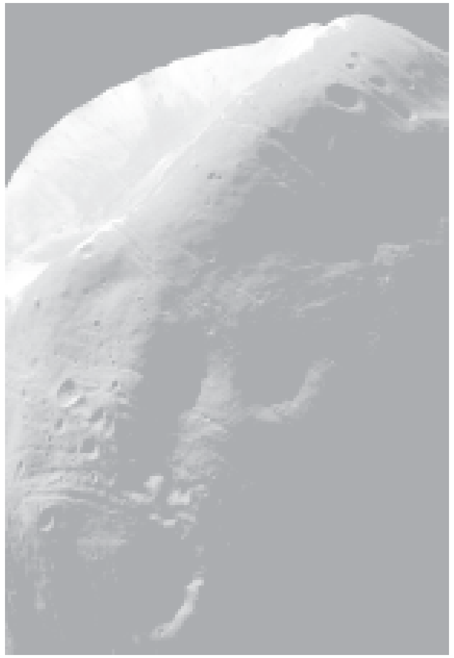
3.3 Histogram Processing

- Histogram Matching:

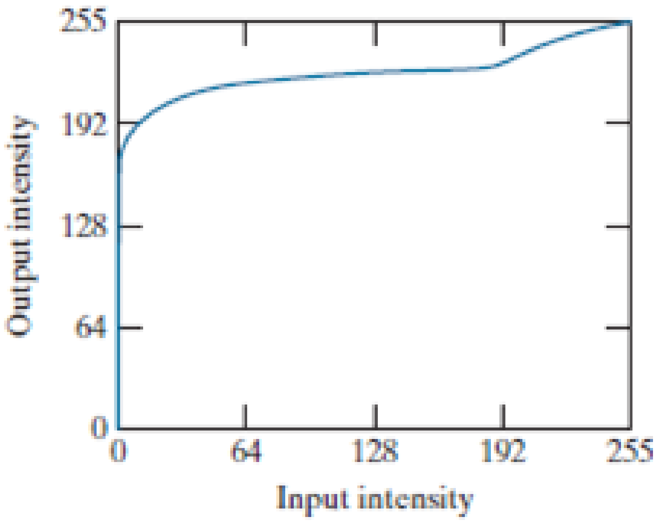


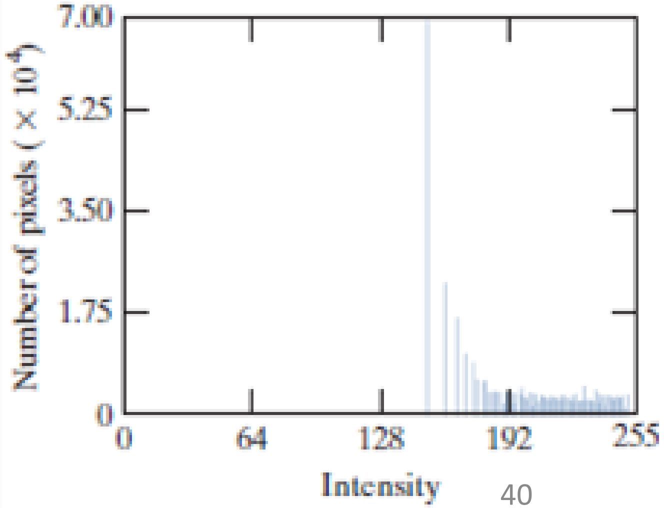
a b

FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)



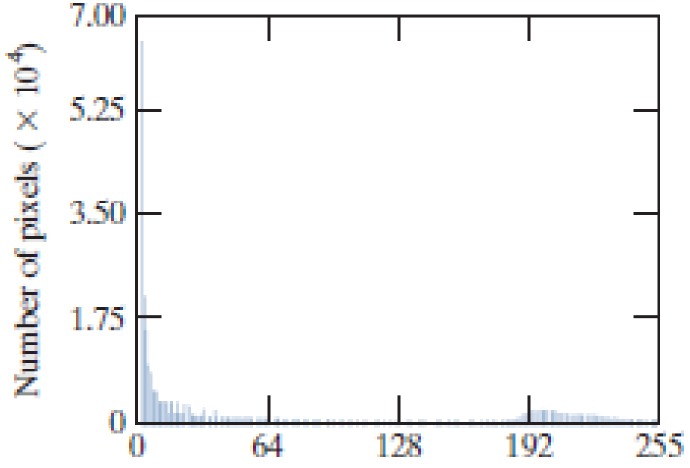
Histogram Equalization

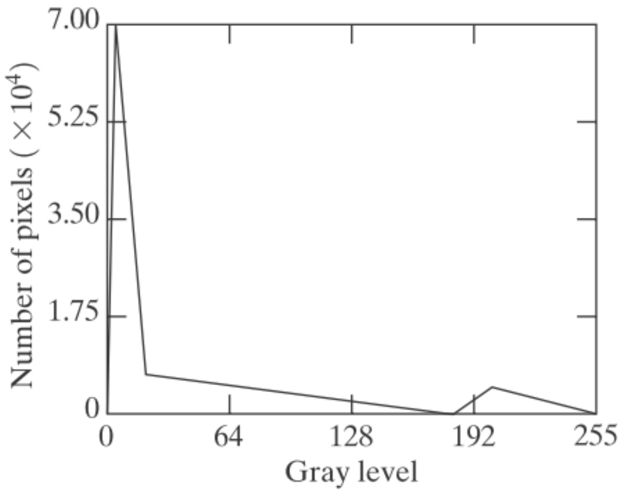


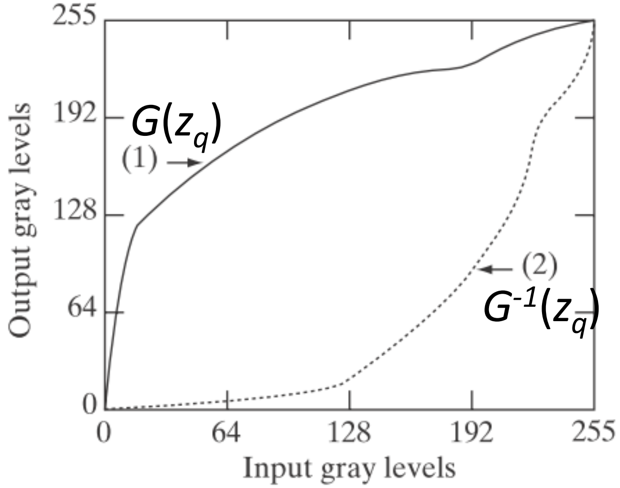


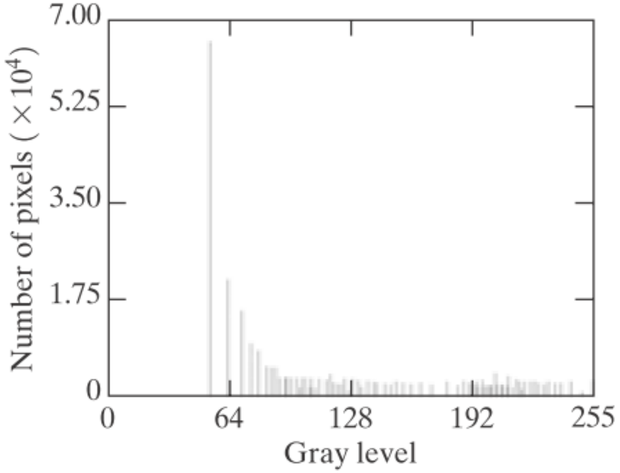
3.3 Histogram Processing

- Histogram Matching:

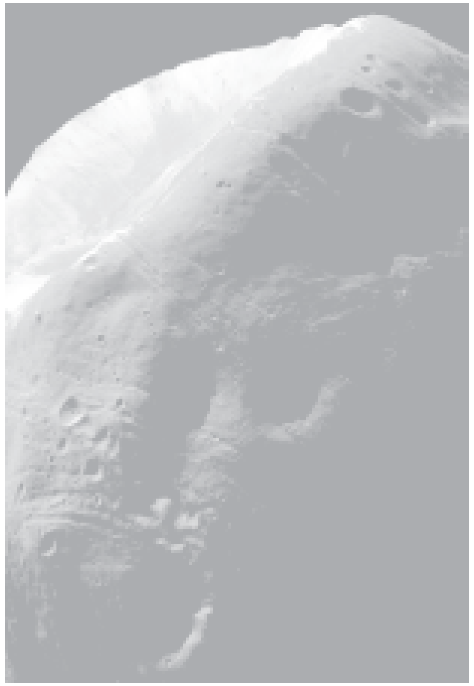








3.3 Histogram Processing



Histogram Equalization

Histogram Matching

3.3 Histogram Processing

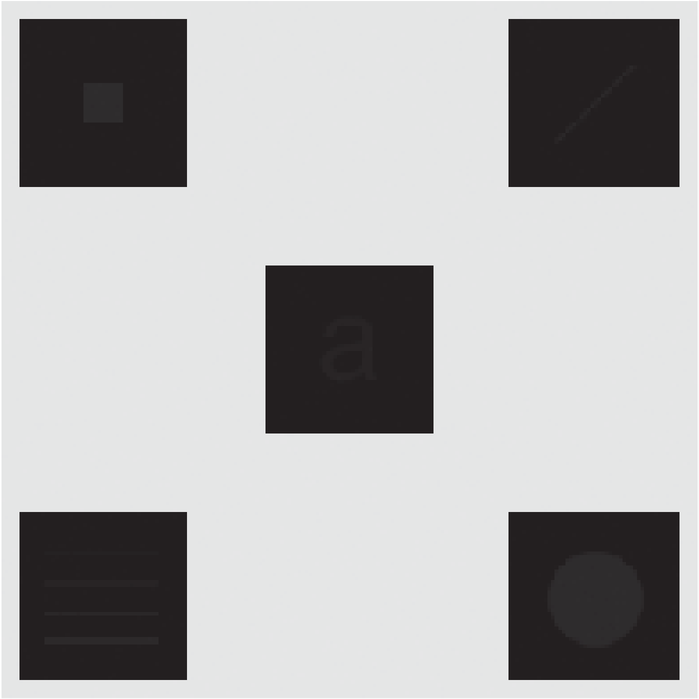
- Local Enhancement:
  - The histogram processing methods discussed in the previous two sections are global.
    - Pixels of an entire image are modified by one transformation function.
  - There are cases in which it is necessary to enhance details over small areas in an image.
    - The solution is to devise transformation functions based on the gray-level distribution—or other properties—in the neighborhood of every pixel in the image.

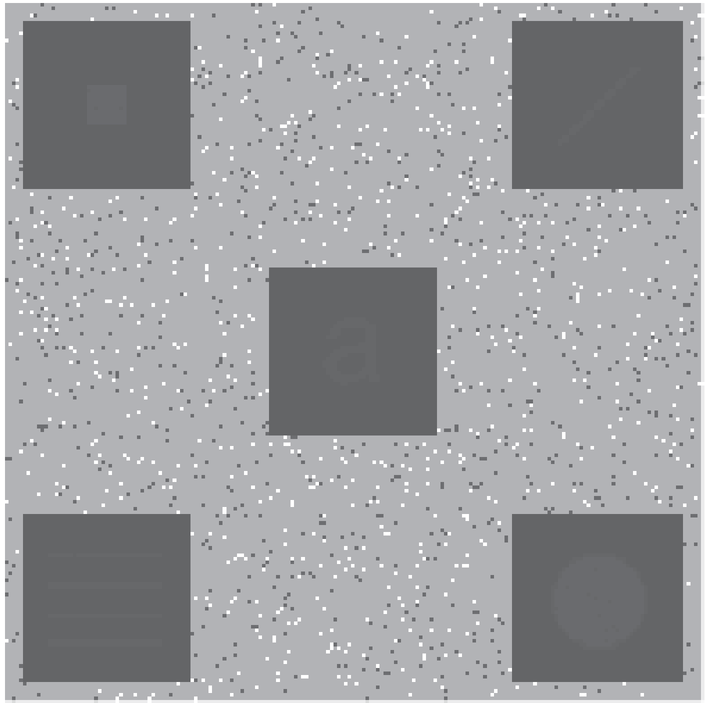
3.3 Histogram Processing

- Local Enhancement:
  - The histogram processing techniques previously described are easily adaptable to local enhancement.
    1. Define a square or rectangular neighborhood and move the center to the pixel.
    2. At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is performed.
    3. This function is finally used to map the gray level of the pixel centered in the neighborhood.
    4. The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.

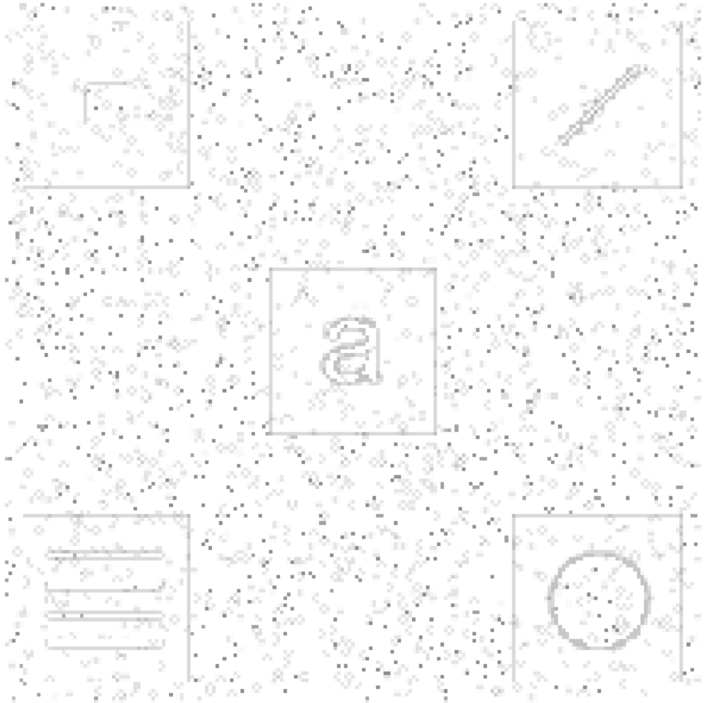
3.3 Histogram Processing

- Local Enhancement:





Global histogram equalization.



Local histogram equalization using a 3x3 neighborhood.

3.3 Histogram Processing

- Use of Histogram Statistics for Image Enhancement:
  - Let *r* denote a discrete random variable representing discrete gray-levels in the range [0, *L*-1], and let $p(r_i)$ denote the normalized histogram component corresponding to the i^{th} value of *r*. The n^{th} moment of *r* about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i) \quad .....(3.3\text{-}18)$$

    - where *m* is the mean value of *r* (its average gray level):

$$m = \sum_{i=0}^{L-1} r_i \, p(r_i) \quad .....(3.3\text{-}19)$$

  - It follows from Eqs. (3.3-18) and (3.3-19) that $\mu_0 = 1$ and $\mu_1 = 0$. The second moment is:

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i) \quad .....(3.3\text{-}20) \rightarrow \text{Variance of } r$$

3.3 Histogram Processing

- Use of Histogram Statistics for Image Enhancement:
  - we are interested primarily in the mean, which is a measure of average gray level in an image, and the variance (or standard deviation), which is a measure of average contrast.
    - The global mean and variance are measured over an entire image and are useful primarily for cross adjustments of overall intensity and contrast.
    - The local mean and variance are used as the basis for making changes that depend on image characteristics in a predefined region about each pixel in the image. ←Much powerful!!!

3.3 Histogram Processing

- Use of Histogram Statistics for Image Enhancement:
  - Let (x, y) be the coordinates of a pixel in an image and let S_{xy} denote a neighborhood (sub-image / patch) of specified size, centered at (x, y).
  - From Eq. (3.3-19) the mean value of the pixels in S_{xy} can be computed using the expression

$$m_{S_{xy}} = \sum_{(s,t)\in S_{xy}} r_{s,t}\, p(r_{s,t}) \quad \text{.....(3.3-21)}$$

  - where $r_{s,t}$ is the gray level at coordinates (s, t) in the neighborhood, and $p(r_{s,t})$ is the neighborhood normalized histogram component corresponding to that value of gray level.
  - The gray-level variance of the pixels in region S_{xy} is given by

$$\sigma^2_{S_{xy}} = \sum_{(s,t)\in S_{xy}} \left[r_{s,t} - m_{S_{xy}} \right]^2 p(r_{s,t}) \quad \text{.....(3.3-22)}$$

3.3 Histogram Processing

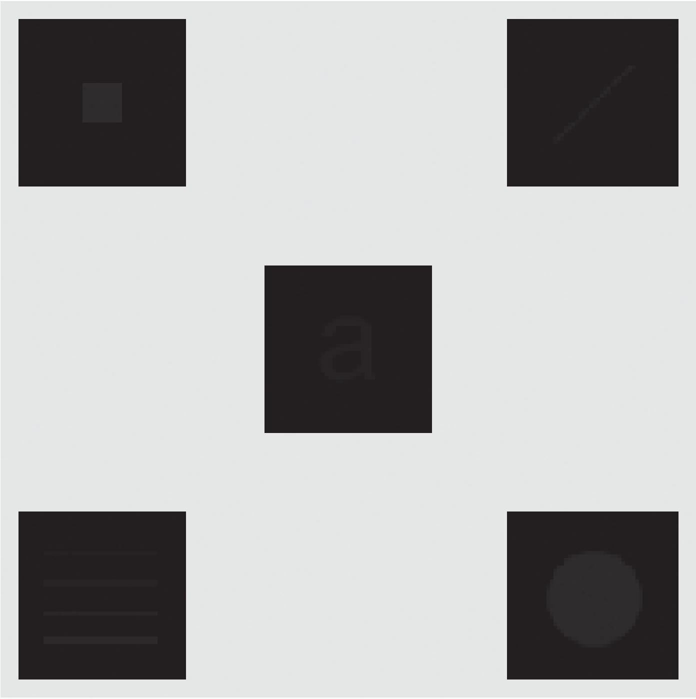
- Use of Histogram Statistics for Image Enhancement:
  - An important aspect of image processing using the local mean and variance :
    - The flexibility
    - The statistical measures have a close, predictable correspondence with image appearance.

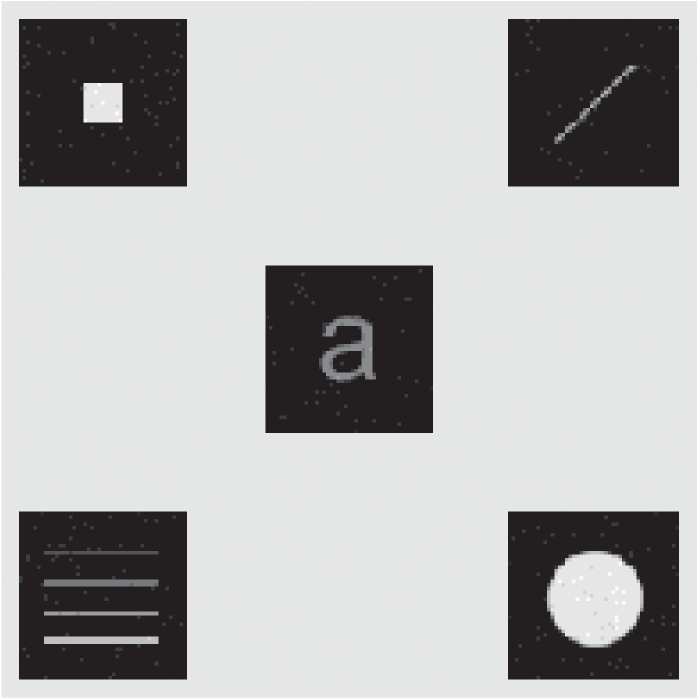
$$g(x,y)=\begin{cases} Cf(x.y) & \text{if } k_0 m_G \le m_{s_{x,y}} \le k_1 m_G \text{ AND } k_2\sigma_G \le \sigma_{s_{x,y}} \le k_3\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

where m_G and σ_G are the global mean and standard deviation.

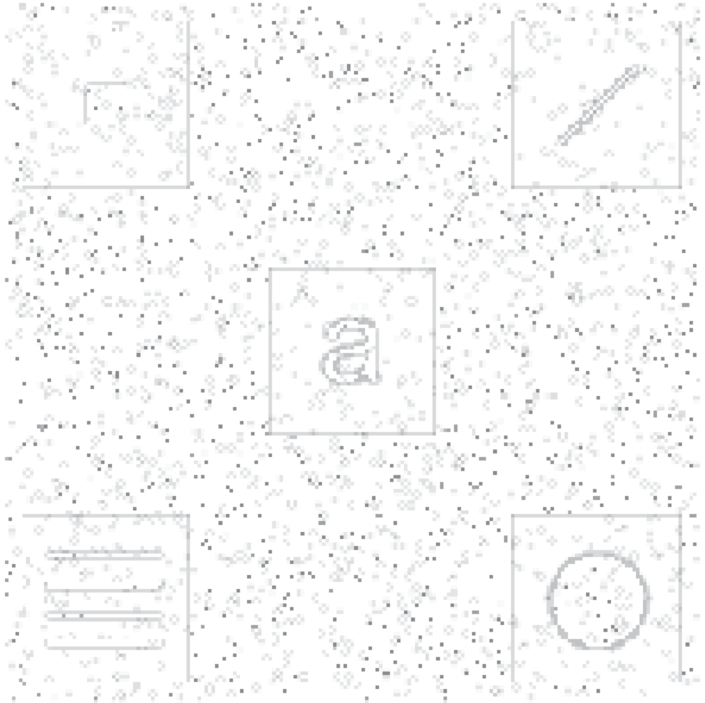
3.3 Histogram Processing

- Use of Histogram Statistics for Image Enhancement:





Enhanced by Local histogram statistic.

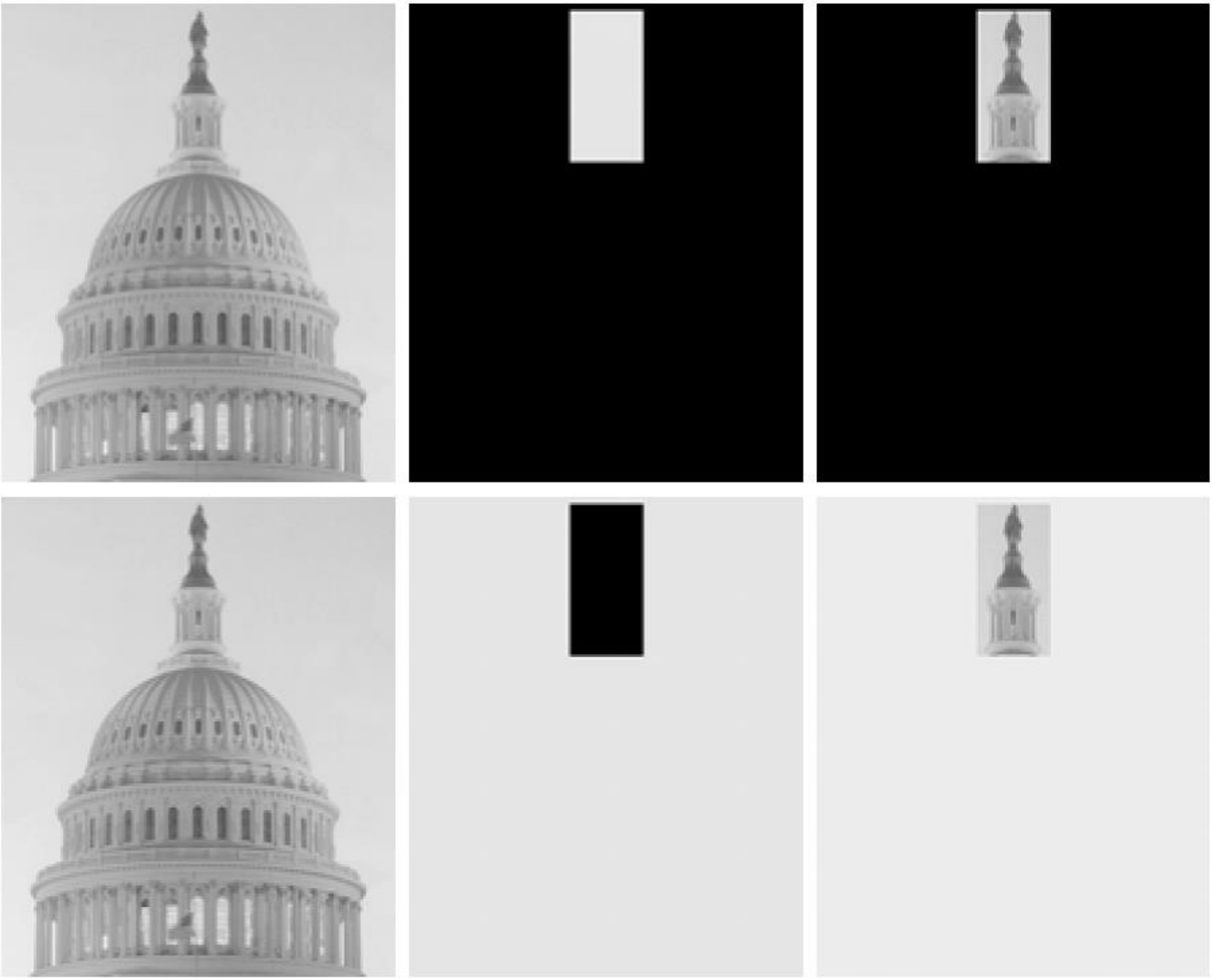


Local histogram equalization using a 3x3 neighborhood.

3.4 Enhancement Using Arithmetic/Logic Operations

- Arithmetic/logic operations involving images are performed on a pixel-by-pixel basis between two or more images.
  - This excludes the logic operation NOT, which is performed on a single image.
- The AND, OR, and NOT logic operators are functionally complete. Any other logic operator can be implemented by using only these three basic functions.

3.4 Enhancement Using Arithmetic/Logic Operations



AND operation

OR operation

Source 1 Source 2 Result

3.4 Enhancement Using Arithmetic/Logic Operations

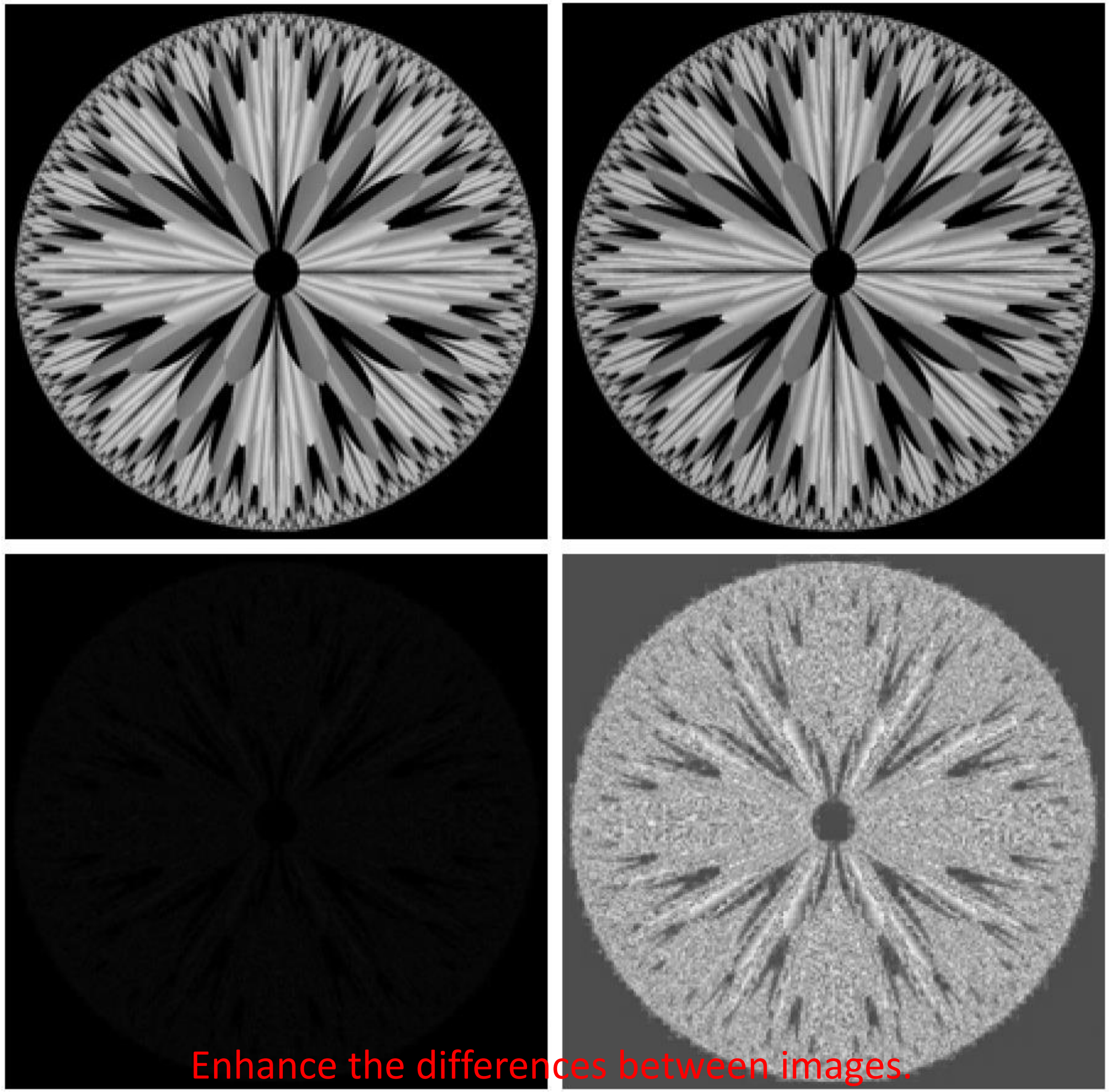
- Image Subtraction:
  - The difference between two images $f(x, y)$ and $h(x, y)$, expressed as

    $$g(x,y) = f(x,y) - h(x,y) \quad .....(3.4\text{-}1)$$

    is obtained by computing the difference between all pairs of corresponding pixels from f and h.

3.4 Enhancement Using Arithmetic/Logic Operations

- Image Subtraction:



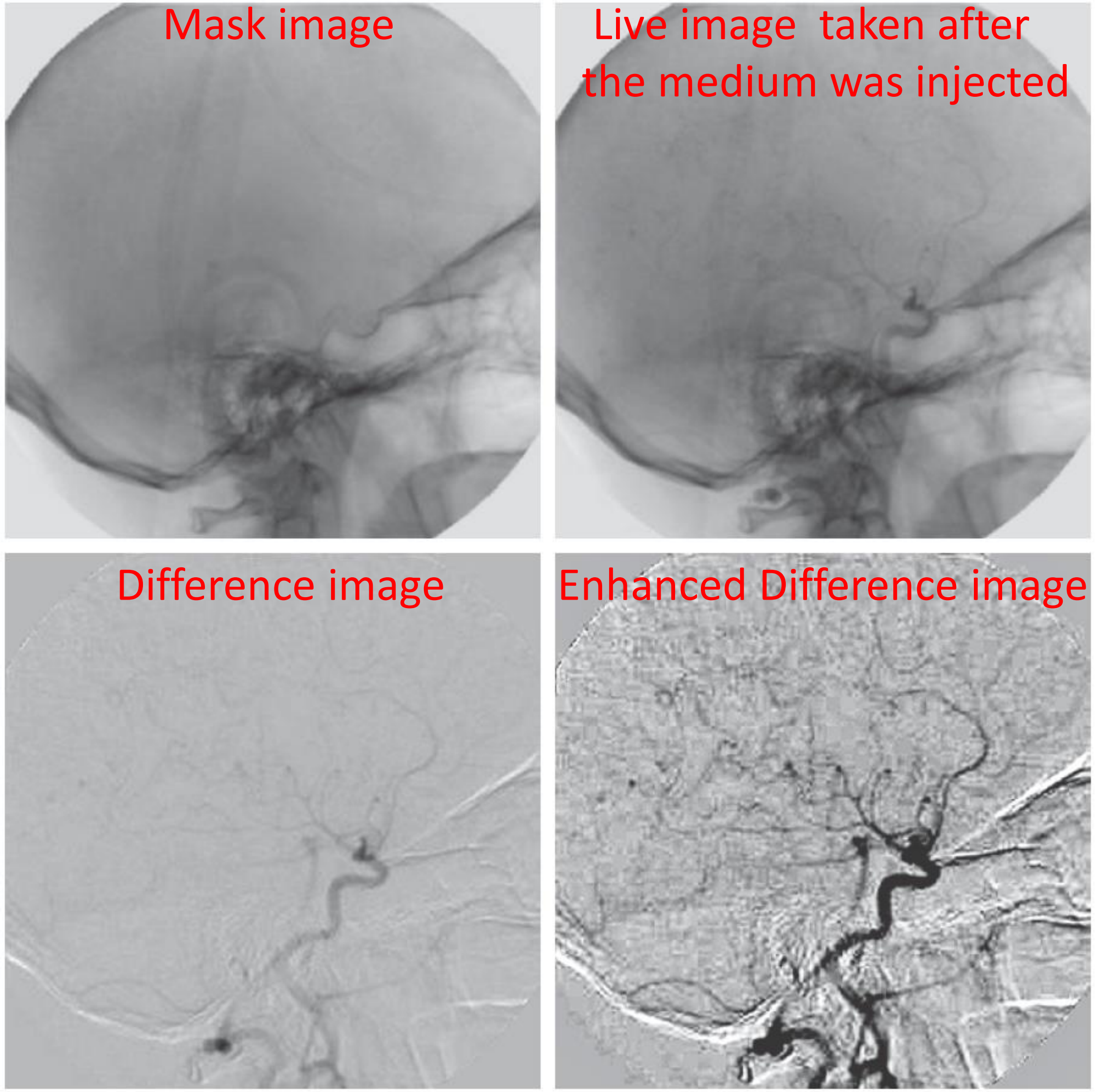
a b
c d

FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

Enhance the differences between images.

3.4 Enhancement Using Arithmetic/Logic Operations

- Image Subtraction:



3.4 Enhancement Using Arithmetic/Logic Operations

- Image Subtraction:
  - Data ranges from 0~255 to -255~255 after subtraction.
    1. (Data+255)/2
    2. 255*(Data – Min.)/(Max.-Min.)

3.4 Enhancement Using Arithmetic/Logic Operations

- Image Averaging:
  - Consider a noisy image $g(x, y)$ formed by the addition of noise eta $\eta(x, y)$ to an original image $f(x, y)$:

    $$g(x,y) = f(x,y) + \eta(x,y) \quad .....(3.4\text{-}2)$$

    - where the assumption is that at every pair of coordinates (x, y) the noise is uncorrelated* and has zero average value.
  - If an image $\bar{g}(x,y)$ is formed by averaging K different noisy images

    $$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

    then it follows that $E\{\bar{g}(x,y)\} = f(x,y)$ and $\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$
    - where $E\{\bar{g}(x,y)\}$ is the expected value of $\bar{g}$, and $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{\eta(x,y)}$ are the variance of $\bar{g}$ and η.

*:The covariance of two random variables x_i and x_j is defined as $E[(x_i-m_i)(x_j-m_j)]$.
If the variables are uncorrelated, their covariance is 0.

3.4 Enhancement Using Arithmetic/Logic Operations

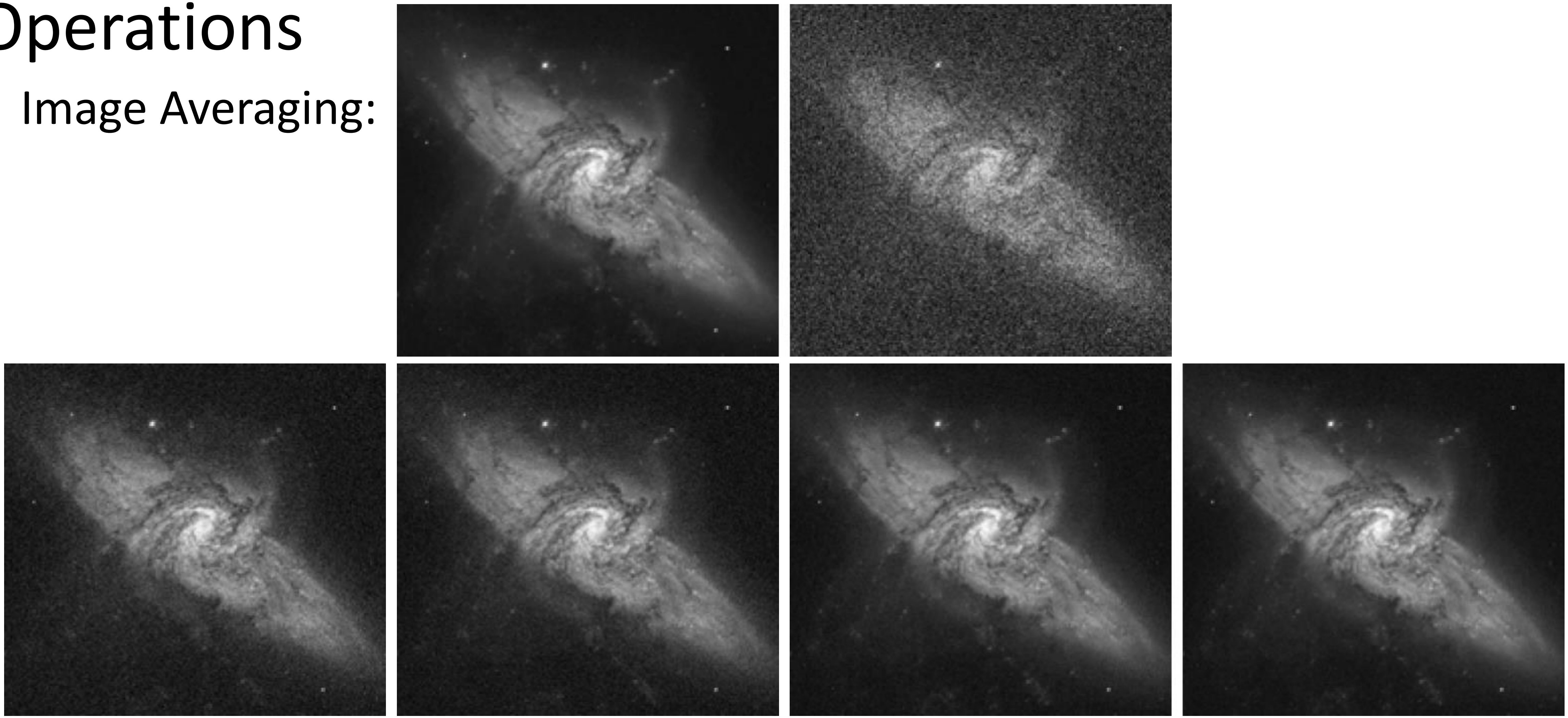
- Image Averaging:
  - The standard deviation at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)} \quad .....(3.4\text{-}6)$$

  - As *K* increases, Eq. (3.4-6) indicate that the variability (noise) of the pixel values at each location (*x*, *y*) decreases.

3.4 Enhancement Using Arithmetic/Logic Operations

- Image Averaging:

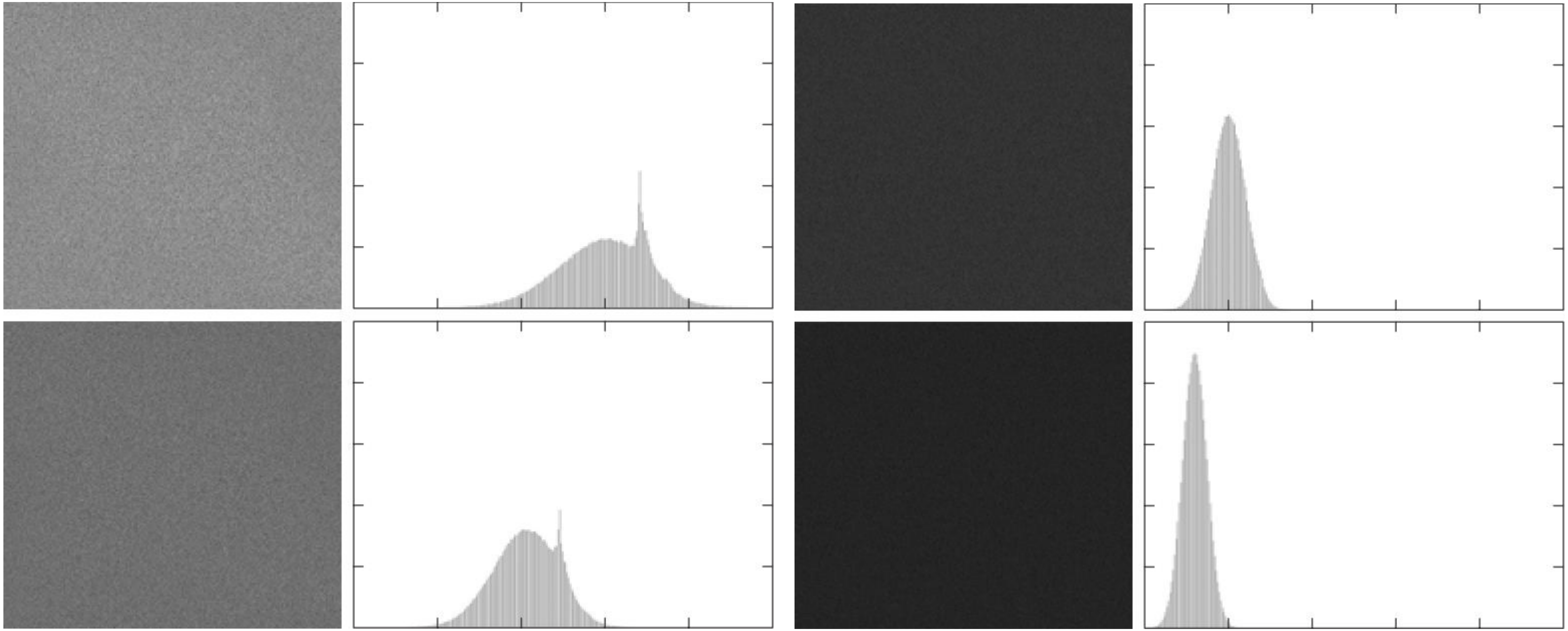


a b
c d e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

3.4 Enhancement Using Arithmetic/Logic Operations

- Image Averaging:



Difference images of Figure 3.30