Image Enhancement

- It is the process of manipulating an image so that the result is more suitable than the original for a specific application.
- The method used for enhancing is problem specific.
- For example the method quite useful for enhancing X-ray images may not be best approach for enhancing satellite images taken in infrared band.
- Two ways to enhance an image
  - Intensity transformation
  - Filtering

- The spatial domain processes can be denoted by the expression

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

- $f(x,y)$ is the input image,
- $g(x,y)$ is the output image,
- T is an operator on f defined over a $n*n$ neighborhood of point (x,y).
- The smallest neighborhood is of size 1*1.

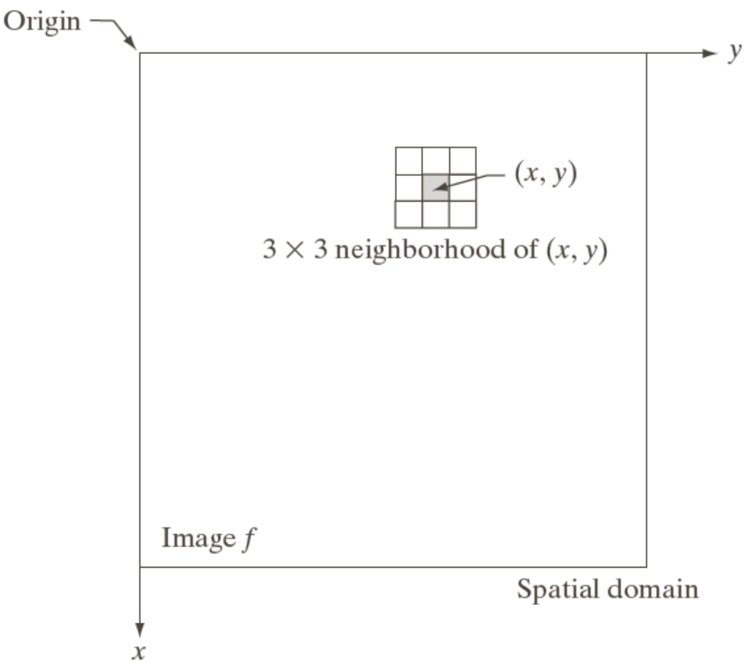


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

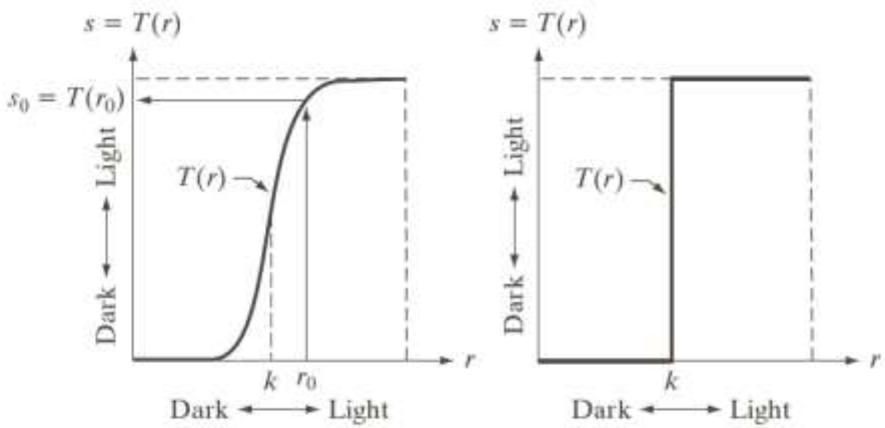
- In this case g depends only on the value of f at a single point (x,y).
- T becomes an intensity transformation function of the form

  $s = T(r)$

  where s and r variables denoting intensity of g and f at any point (x,y)

Contrast Streching and Thresholding

- If T(r) has the form in fig 3.2(a), would produce an image of higher contrast than the original.
- The function darkens the intensity levels below k and brightens the intensity levels above k
- Function T(r) in fig.3.2(b) will produce a binary image.
- This function is called thresholding function.



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Contrast enhancement

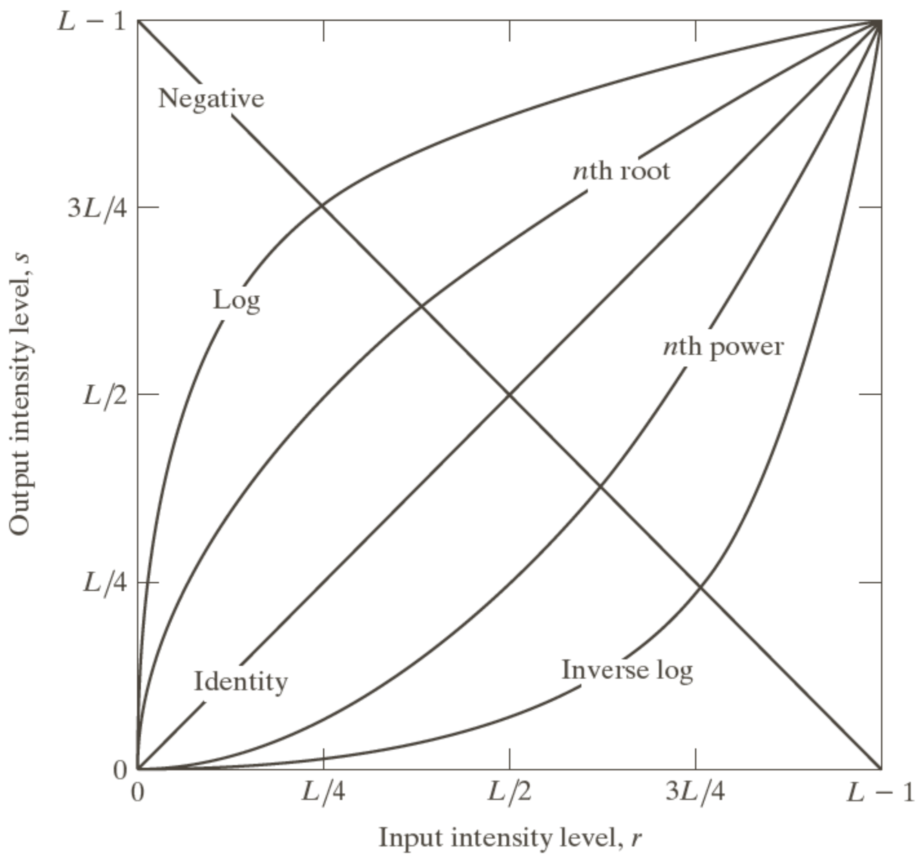
Thresholding

Image Negative

- The negative tranformation of an image with intensity levels in the range [0, L-1] is obtained using the expression

  s = L - 1- r

Negative transformation is used for enhancing white or grey detail embedded in dark regions of an image



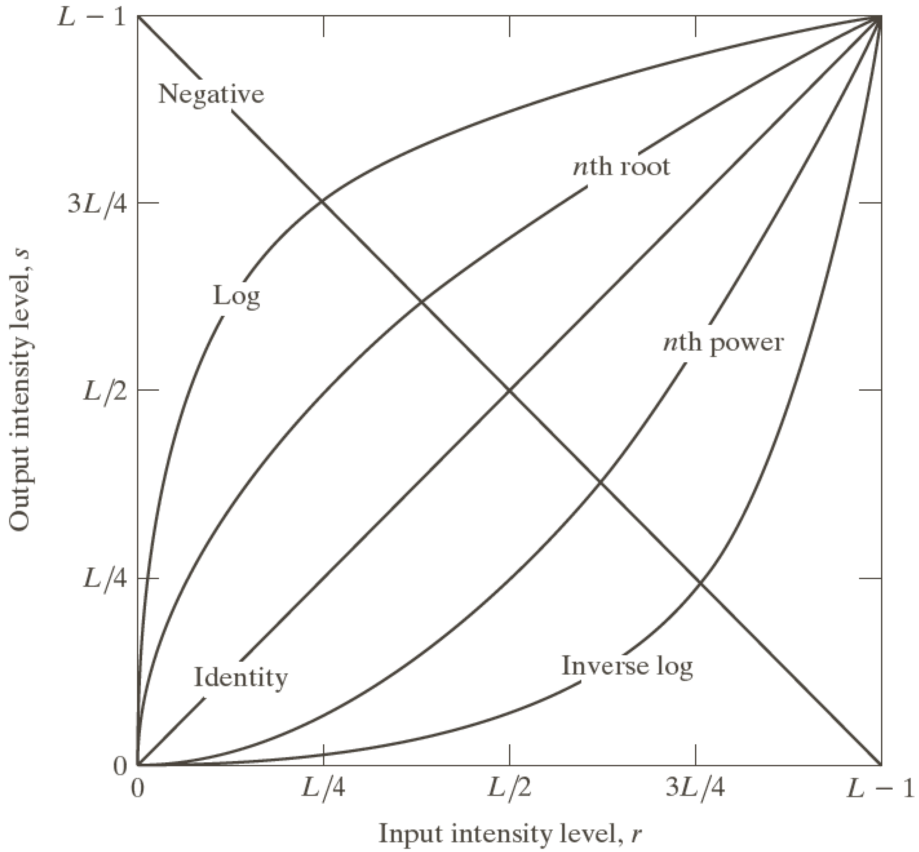
Negative image

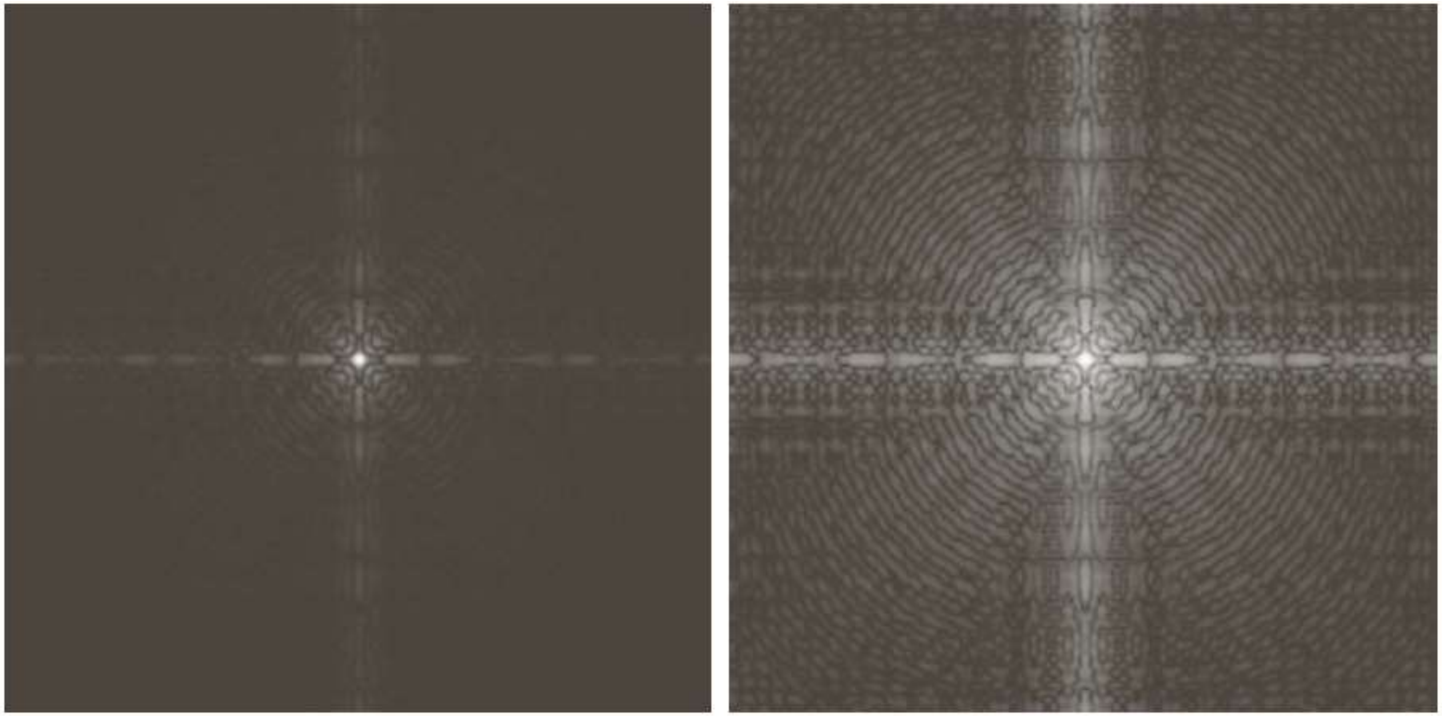
Log transformation

- The log transformation of an image is given as

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

- It maps a narrow range of low intensity values in the input into a wider range of output levels.
- Used to expand values of dark pixels in an image while compressing higher level values.
- The opposite is true of the inverse log tranformation





a b

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

Power-law or Gamma Transformation

- Gamma transformation of an image is given as

  $S = c\, r^{\gamma}$

- As with log transformations, power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- Curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$.
- When *$c = \gamma = 1$, equation reduces to the identity transformation.*

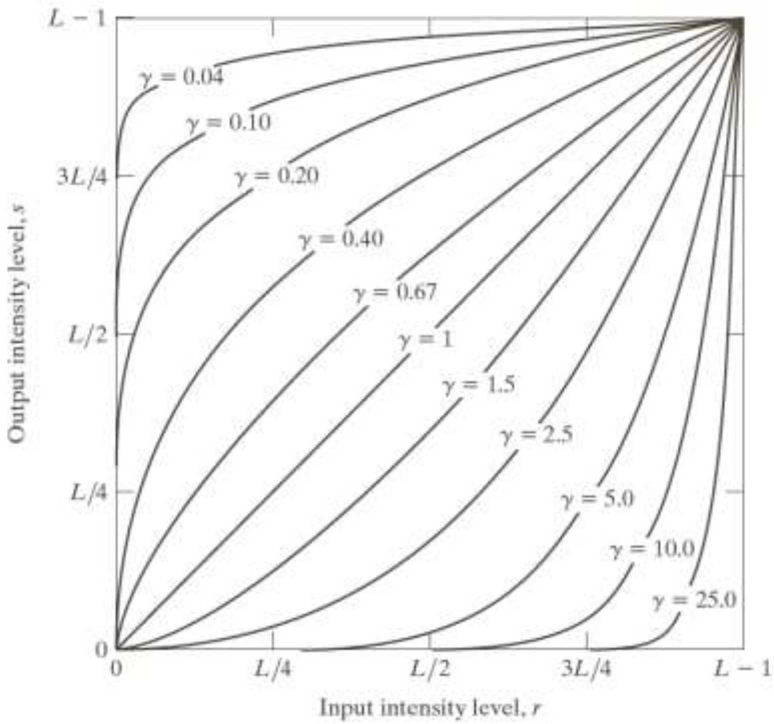
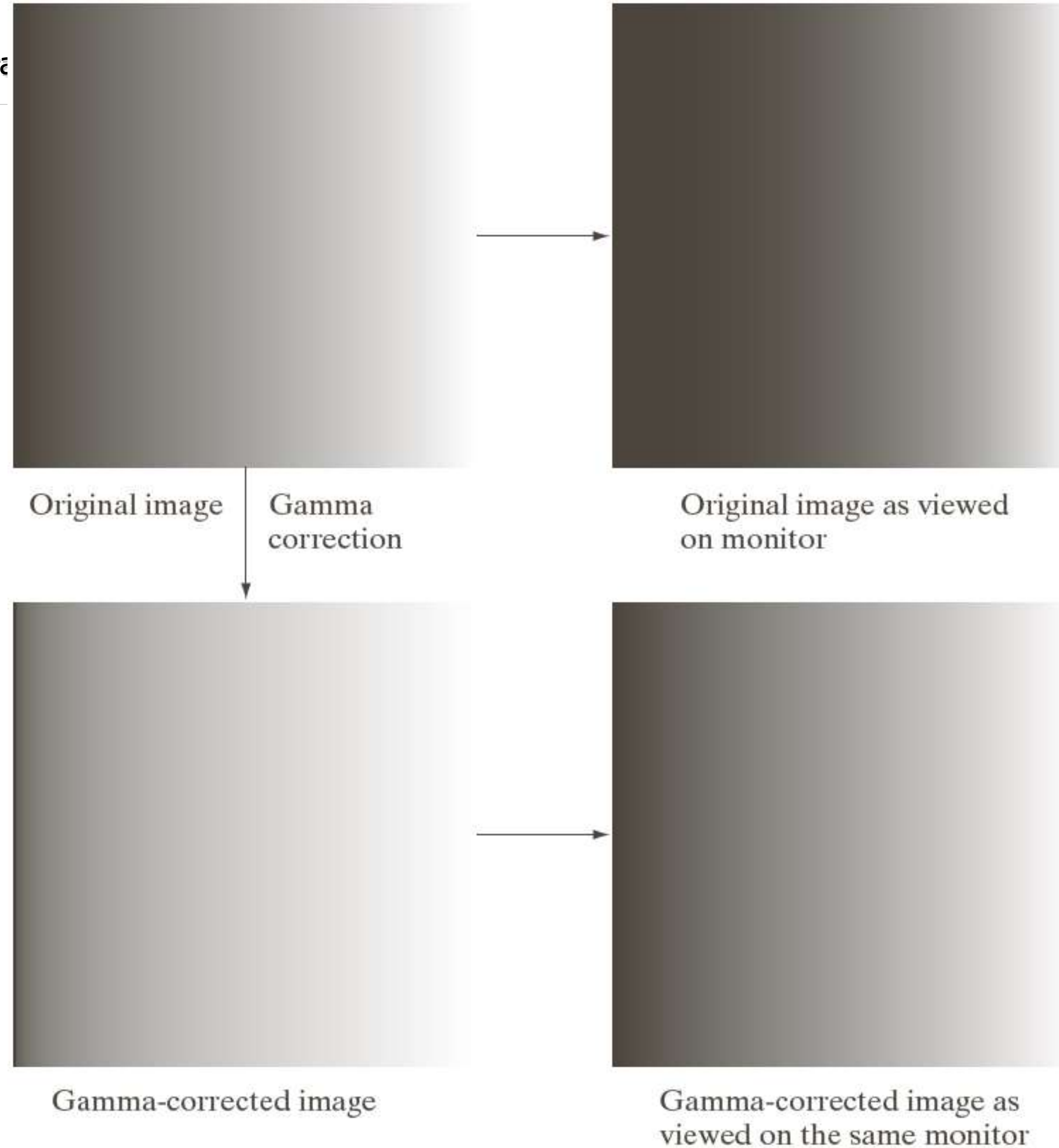
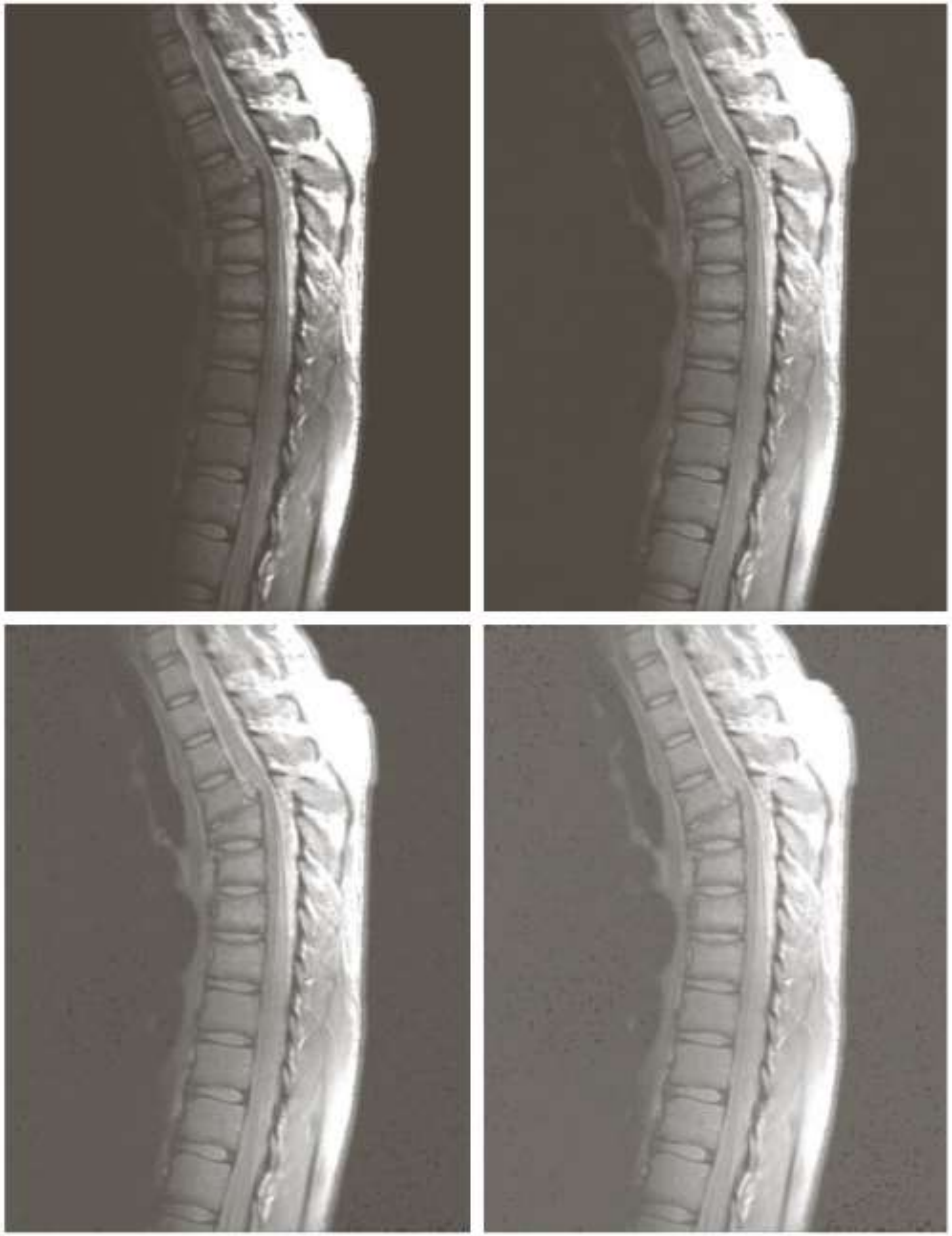


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

- 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.
- As the curve for γ = 2 5 . in Fig. 3.6 shows, such display systems would tend to produce images that are darker than intended.
- In this case, gamma correction consists of using the transformation *s = r^1/2.5 = r^0.4* to preprocess the image before inputting it into the monitor.



with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).



a b
c d

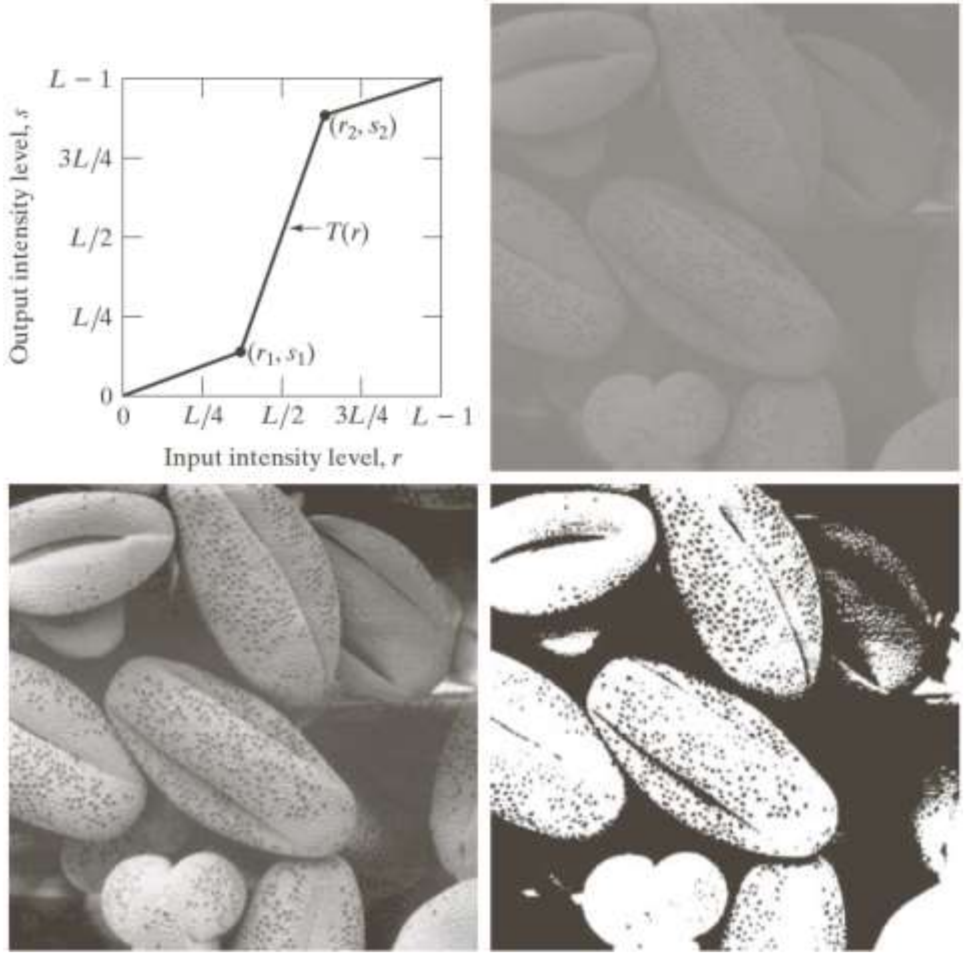
FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt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

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.)

Piecewise- linear transformations: contrast stretching

- It is the process that expands the range of intensity levels in an image
- So that it spans the full intensity range of the recording medium or display device

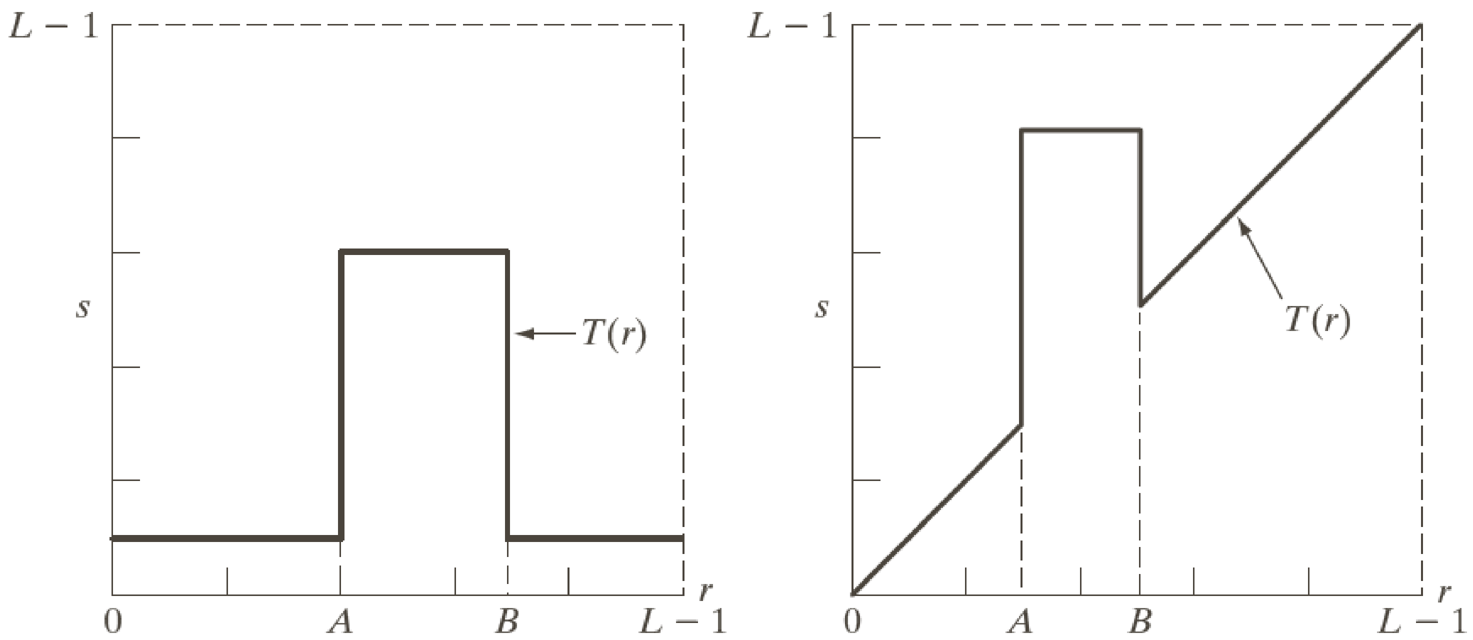


a b
c d

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.)

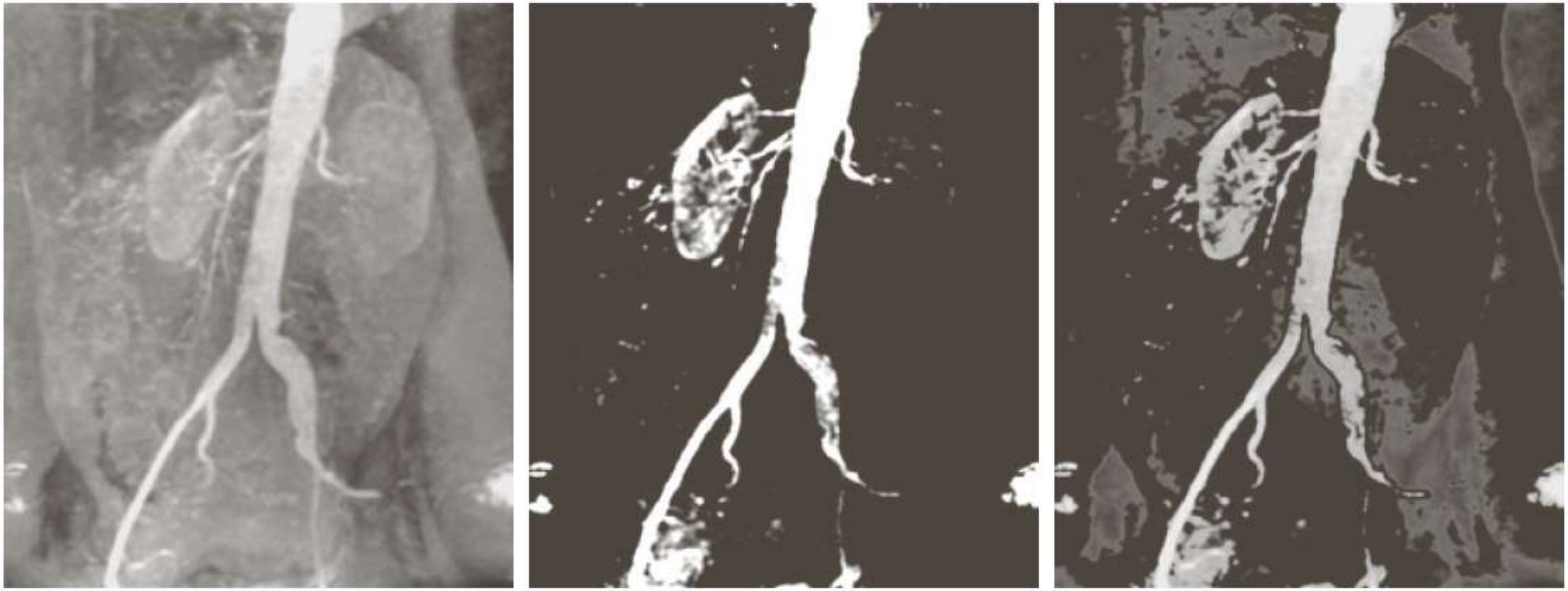
Piecewise- linear transformations :Intensity level slicing

- Is used for highlighting a specific range of intensities in an image.



a b

FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.



a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Piecewise- linear transformations: Bit-plane slicing

- An image may be considered as being composed of eight 1-bit planes.
- Plane 1 containing lower order bit of all pixels in the image and plane 8 all the higher order bits.
- Decomposing an image into its planes is useful for
  - Image compression
  - To determine the adequacy of the number of bits used to quantize the image.

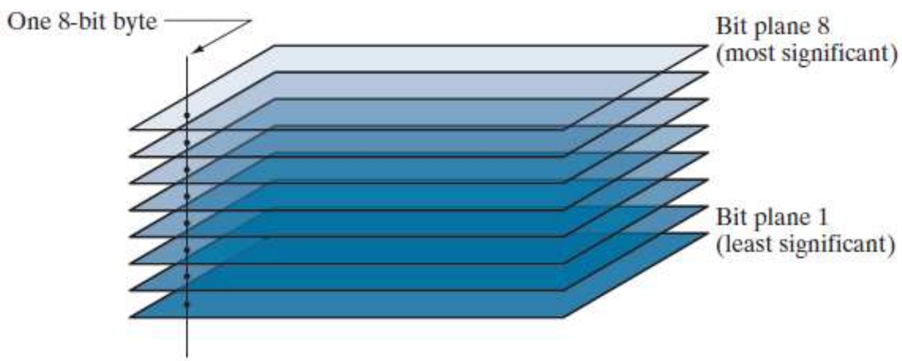


FIGURE 3.13 Bit-plane representation of an 8-bit image.

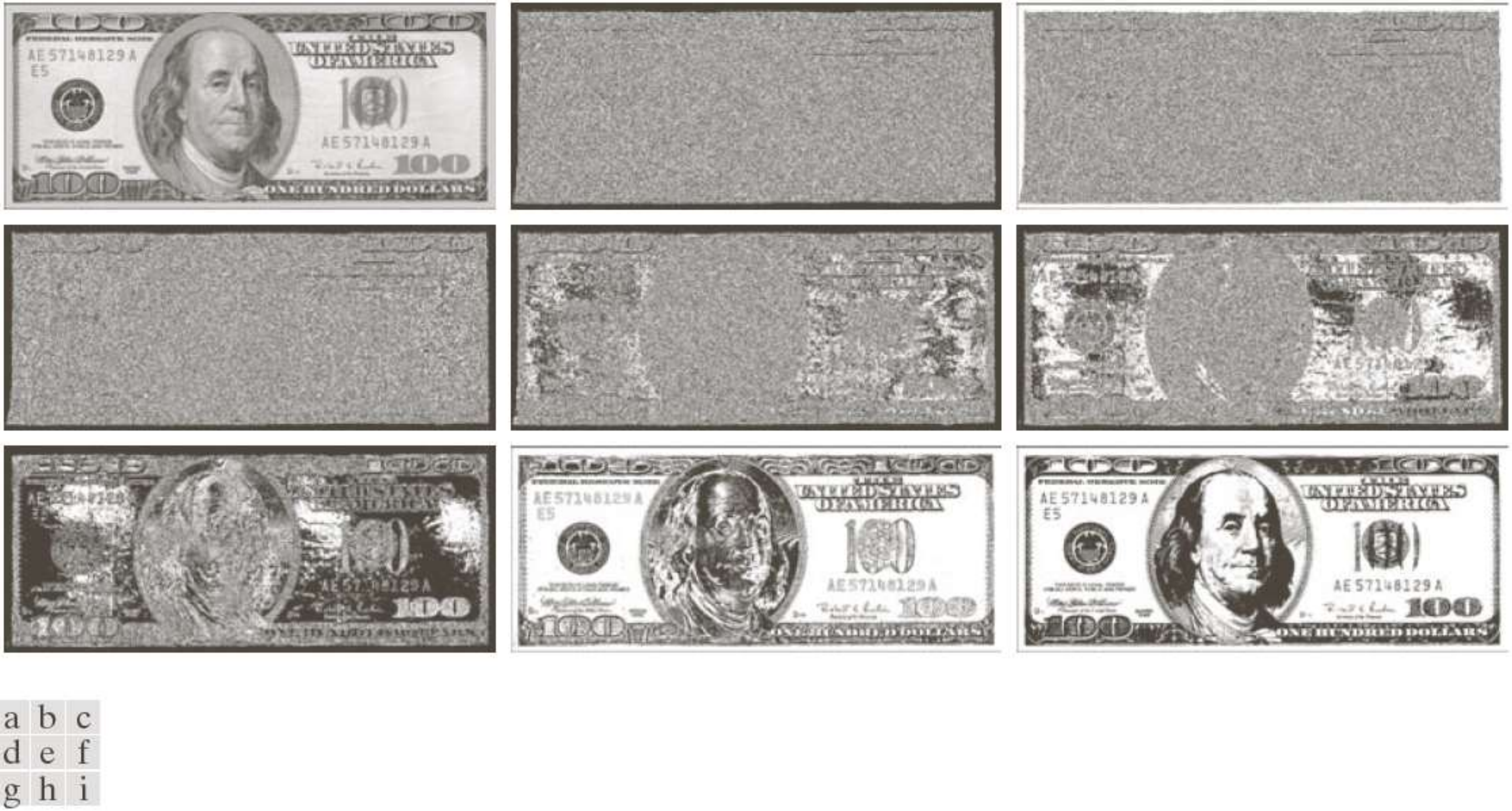


FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

Histogram Processing

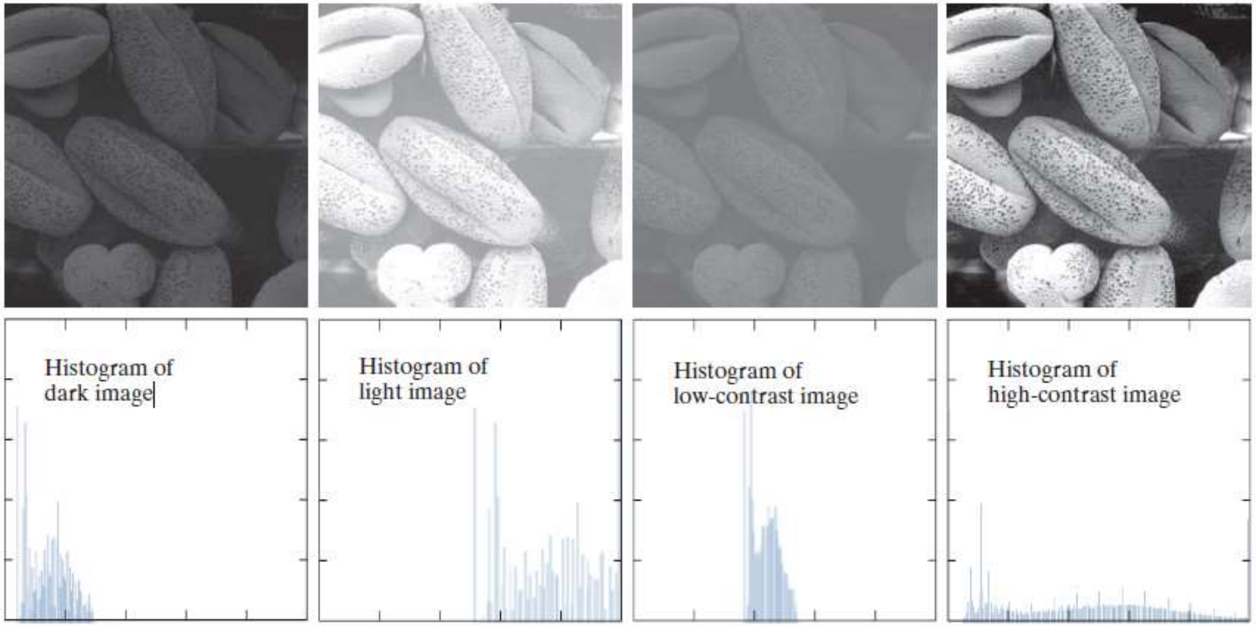
- Histogram of a digital image with intensity levels in the image [0,L-1] is a discrete function

$$h(r_k) = n_k$$

- r_k is the kth intensity value.
- n_k is the number of pixels in the image with intensity r_k
- A normalized histogram of an image is given by

$$p(r_k) = n_k / MN$$

for k = 0,1,2,........., L-1



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 $p(r_k)$.

- Let the variable *r denote the intensities* of an image to be processed.
- we focus attention on transformations (intensity mappings) of the form

$$s = T(r) \quad 0 \le r \le L - 1$$

- We assume that

(a) $T(r)$ is a monotonic† increasing function in the interval $0 \le r \le L-1$; and
(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

In some formulations to be discussed shortly, we use the inverse transformation

$$r = T^{-1}(s) \qquad 0 \le s \le L-1 \qquad (3\text{-}9)$$

in which case we change condition (a) to:

(a′) $T(r)$ is a *strictly* monotonic increasing function in the interval $0 \le r \le L-1$.

Histogram Equalization

- Assuming initially continuous intensity values, let the variable *r denote the intensities* of an input image to be processed.
- The variable ‘r’ may be viewed as a random variable in the interval [0,L − 1].
- Let T(r) is a transformation function that produce an output intensity value, *s*, for a given intensity value *r* in the input image.
- Let $p_r(r)$ and $p_s(s)$ denote the PDFs of intensity values r and s in two different images.
- A fundamental result from probability theory is that if $p_r(r)$ and T(r) are known, and T(r) is continuous and differentiable over the range of values of interest, then the PDF of the transformed (mapped) variable s can be obtained as

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \qquad (3\text{-}10)$$

- Thus, the PDF of the output intensity variable, *s, is determined by the* PDF of the input intensities and the transformation function used.

- A transformation function of particular importance in image processing is

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

- We use Eq. (3-10) to find the $p_s(s)$ *corresponding to the above transformation*
- We know from Leibniz's rule in calculus that the derivative of a definite integral with respect to its upper limit is the integrand evaluated at the limit. That is,

$$\begin{aligned}\frac{ds}{dr} &= \frac{dT(r)}{dr}\\ &= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] \qquad (3\text{-}12)\\ &= (L-1)p_r(r)\end{aligned}$$

Substituting this result for dr/ds in Eq. (3-10), and noting that all probability values are positive, gives the result

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| \\ &= \frac{1}{L-1} \qquad 0 \le s \le L-1 \end{aligned} \tag{3-13}$$

We recognize the form of $p_s(s)$ in the last line of this equation as a *uniform* probability density function. Thus, performing the intensity transformation in Eq. (3-11) yields a random variable, s, characterized by a uniform PDF. What is important is that $p_s(s)$ in Eq. (3-13) will *always* be uniform, *independently* of the form of $p_r(r)$. Figure 3.18 and the following example illustrate these concepts.

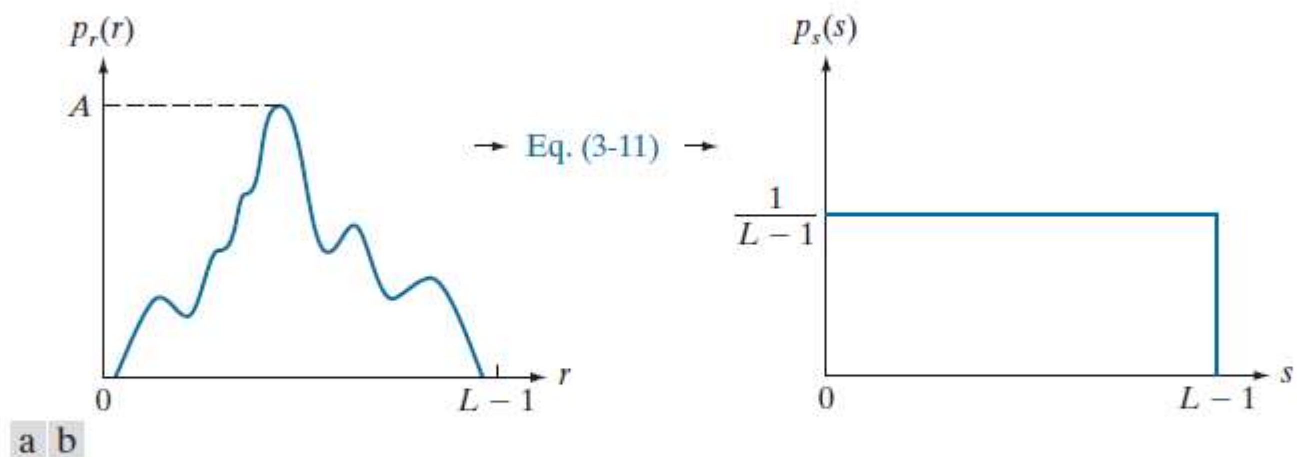


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying Eq. (3-11) to the input PDF. The resulting PDF is always uniform, independently of the shape of the input.

EXAMPLE 3.4: Illustration of Eqs. (3-11) and (3-13).

Suppose that the (continuous) intensity values in an image have the PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

From Eq. (3-11)

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

Suppose that we form a new image with intensities, s, obtained using this transformation; that is, the s values are formed by squaring the corresponding intensity values of the input image, then dividing them by $L-1$. We can verify that the PDF of the intensities in the new image, $p_s(s)$, is uniform by substituting $p_r(r)$ into Eq. (3-13), and using the fact that $s = r^2/(L-1)$; that is,

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|$$

$$= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| = \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1}$$

The last step follows because r is nonnegative and $L > 1$. As expected, the result is a uniform PDF.

For discrete values, we work with probabilities and summations instead of probability density functions and integrals (but the requirement of monotonicity stated earlier still applies). Recall that the probability of occurrence of intensity level r_k in a digital image is approximated by

$$p_r(r_k) = \frac{n_k}{MN} \tag{3-14}$$

where MN is the total number of pixels in the image, and n_k denotes the number of pixels that have intensity r_k. As noted in the beginning of this section, $p_r(r_k)$, with $r_k \in [0, L-1]$, is commonly referred to as a normalized image histogram.

The discrete form of the transformation in Eq. (3-11) is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0,1,2,\ldots,L-1 \tag{3-15}$$

where, as before, L is the number of possible intensity levels in the image (e.g., 256 for an 8-bit image). Thus, a processed (output) image is obtained by using Eq. (3-15) to map each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image, This is called a *histogram equalization* or *histogram linearization* transformation. It is not difficult to show (see Problem 3.9) that this transformation satisfies conditions (a) and (b) stated previously in this section.

EXAMPLE 3.5: Illustration of the mechanics of histogram equalization.

It will be helpful to work through a simple example. Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution in Table 3.1, where the intensity levels are integers in the range $[0, L-1] = [0, 7]$. The histogram of this image is sketched in Fig. 3.19(a). Values of the histogram equalization transformation function are obtained using Eq. (3-15). For instance,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

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

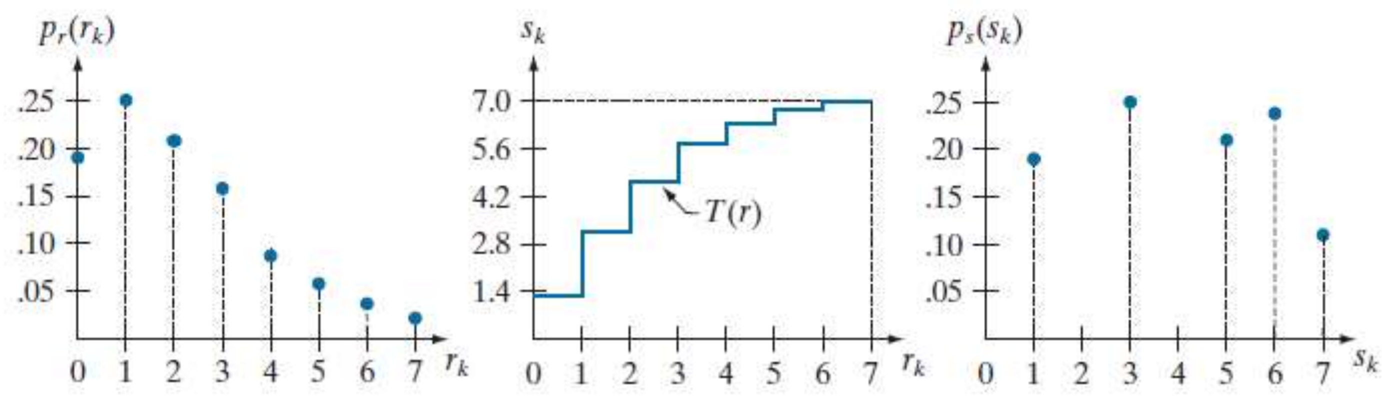
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Similarly, $s_1 = T(r_1) = 3.08$, $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, and $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

At this point, the s values are fractional because they were generated by summing probability values, so we round them to their nearest integer values in the range $[0, 7]$:

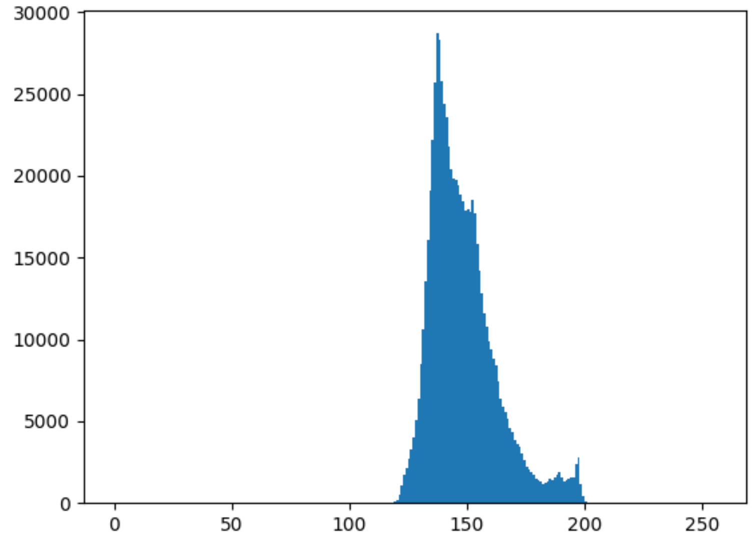
$$\begin{array}{llll} s_0 = 1.33 \rightarrow 1 & s_2 = 4.55 \rightarrow 5 & s_4 = 6.23 \rightarrow 6 & s_6 = 6.86 \rightarrow 7 \\ s_1 = 3.08 \rightarrow 3 & s_3 = 5.67 \rightarrow 6 & s_5 = 6.65 \rightarrow 7 & s_7 = 7.00 \rightarrow 7 \end{array}$$

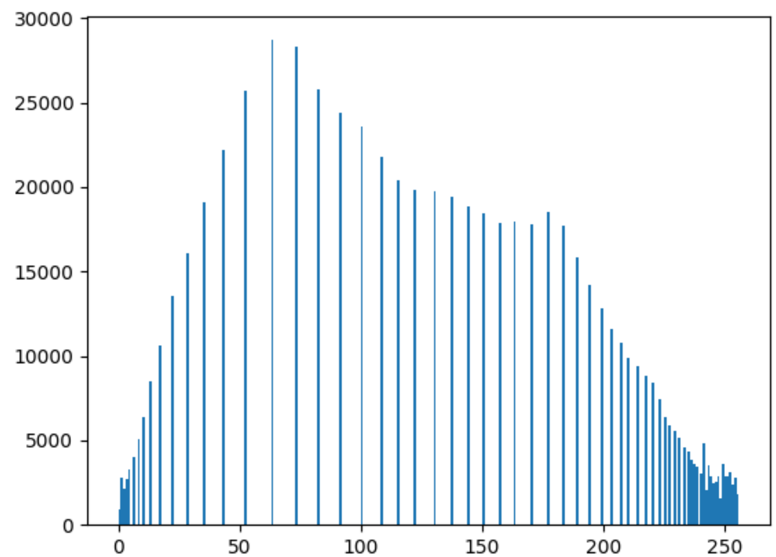
These are the values of the equalized histogram. Observe that the transformation yielded only five distinct intensity levels. Because $r_0 = 0$ was mapped to $s_0 = 1$, there are 790 pixels in the histogram equalized image with this value (see Table 3.1). Also, there are 1023 pixels with a value of $s_1 = 3$ and 850 pixels with a value of $s_2 = 5$. However, both r_3 and r_4 were mapped to the same value, 6, so there are $(656 + 329) = 985$ pixels in the equalized image with this value. Similarly, there are $(245 + 122 + 81) = 448$ pixels with a value of 7 in the histogram equalized image. Dividing these numbers by $MN = 4096$ yielded the equalized histogram in Fig. 3.19(c).

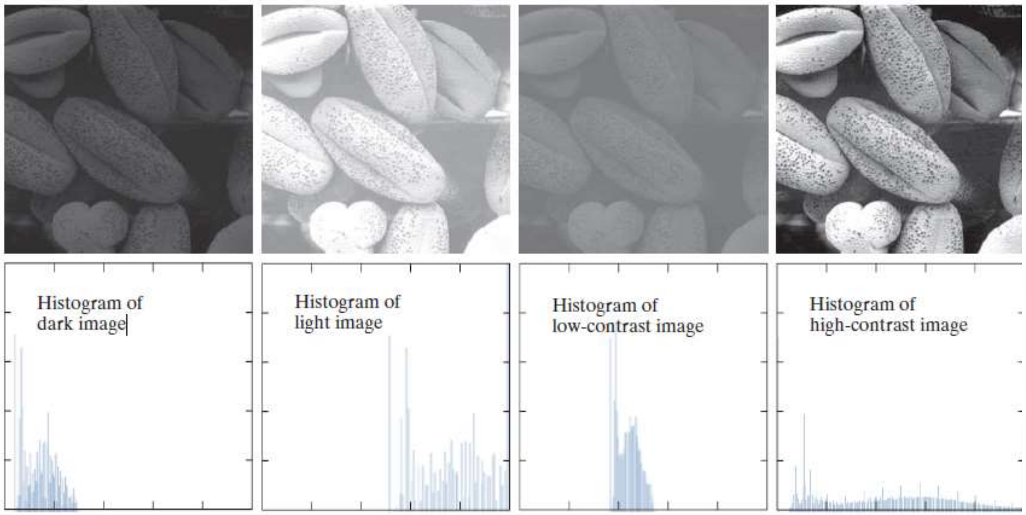


a b c

FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.


30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
250


30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
250



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 $p(r_k)$.

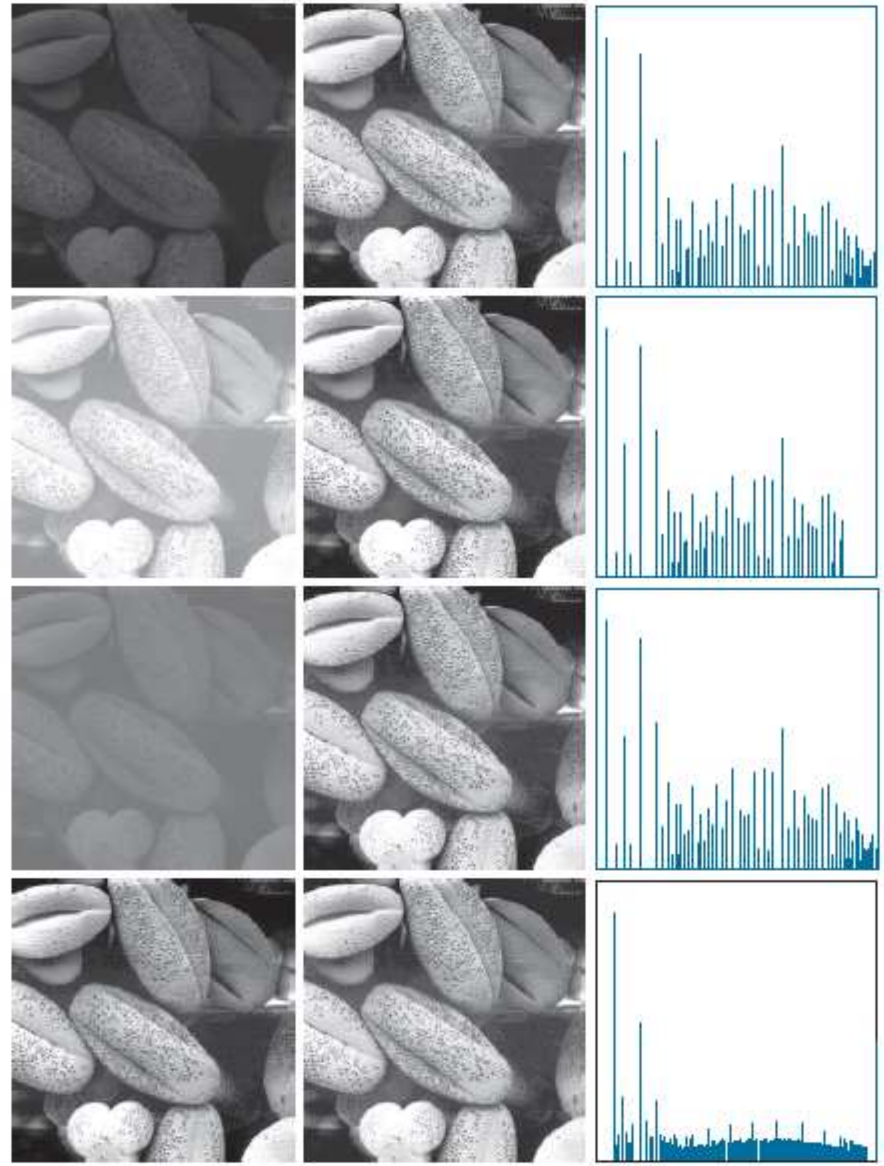


FIGURE 3.20 Left column: Images from Fig. 3.16. Center column: Corresponding histogram-equalized images. Right column: histograms of the images in the center column (compare with the histograms in Fig. 3.16).

- The intensity levels of the equalized image span a wider range of the intensity scale.
- The net result is contrast enhancement.
- The enhancement is fully automatic.
- Based on only information extracted from the given image.
- No external parameters are specified

- Given the histogram of a hypothetical 3-bit image of size 64*64, compute its histogram equalized image.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

- For the given hypothetical 2-bit image of size 5*5 compute its histogram equalized image.

0	0	1	1	2
1	2	3	0	1
3	3	2	2	0
2	3	1	0	0
1	1	3	2	2

- Compute the normalized histogram for the given hypothetical 2-bit image of size 5*5

0	0	1	1	2
1	2	3	0	1
3	3	2	2	0
2	3	1	0	0
1	1	3	2	2

Histogram Matching (Specification) of an image

- The method to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Given an input image, a specified histogram, $p_z(z_i)$, $i = 0,1,2,\ldots,L-1$, and recalling that the s_k's are the values resulting from Eq. (3-20), we may summarize the procedure for discrete histogram specification as follows:

1. Compute the histogram, $p_r(r)$, of the input image, and use it in Eq. (3-20) to map the intensities in the input image to the intensities in the histogram-equalized image. Round the resulting values, s_k, to the integer range $[0, L-1]$.
2. Compute all values of function $G(z_q)$ using the Eq. (3-21) for $q = 0,1,2,\ldots,L-1$, where $p_z(z_i)$ are the values of the specified histogram. Round the values of G to integers in the range $[0, L-1]$. Store the rounded values of G in a lookup table.
3. For every value of s_k, $k = 0,1,2,\ldots,L-1$, use the stored values of G from Step 2 to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k. Store these mappings from s to z. When more than one value of z_q gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.
4. Form the histogram-specified image by mapping every equalized pixel with value s_k to the corresponding pixel with value z_q in the histogram-specified image, using the mappings found in Step 3.

- Consider hypothetical 3-bit image of size 64*64 whose histogram is given below figure 1. Transform this histogram to the specified histogram given below in figure 2

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

z_q	Specified $p_z(z_q)$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15

The first step is to obtain the histogram-equalized values, which we did in Example 3.5:

$$s_0 = 1;\ s_1 = 3;\ s_2 = 5;\ s_3 = 6;\ s_4 = 6;\ s_5 = 7;\ s_6 = 7;\ s_7 = 7$$

In the next step, we compute the values of $G(z_q)$ using the values of $p_z(z_q)$ from Table 3.2 in Eq. (3-21):

$$G(z_0) = 0.00 \quad G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$
$$G(z_1) = 0.00 \quad G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

As in Example 3.5, these fractional values are rounded to integers in the range $[0,7]$:

$$G(z_0) = 0.00 \rightarrow 0 \qquad G(z_4) = 2.45 \rightarrow 2$$
$$G(z_1) = 0.00 \rightarrow 0 \qquad G(z_5) = 4.55 \rightarrow 5$$
$$G(z_2) = 0.00 \rightarrow 0 \qquad G(z_6) = 5.95 \rightarrow 6$$
$$G(z_3) = 1.05 \rightarrow 1 \qquad G(z_7) = 7.00 \rightarrow 7$$

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

TABLE 3.3
Rounded values of the transformation function $G(z_q)$.

Step-3

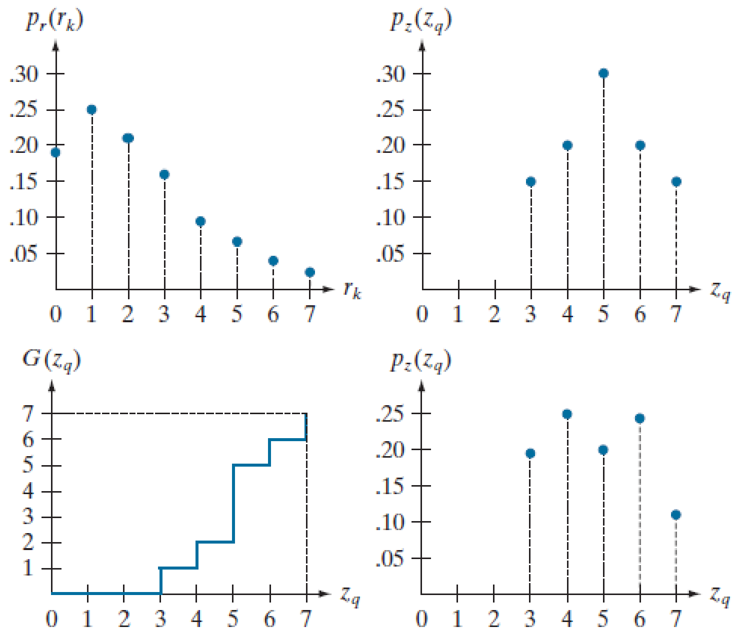
3. For every value of s_k, $k = 0, 1, 2, \ldots, L-1$, use the stored values of G from Step 2 to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k. Store these mappings from s to z. When more than one value of z_q gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

TABLE 3.4 Mapping of values s_k into corresponding values z_q.

Step-4

In the final step of the procedure, we use the mappings in Table 3.4 to map every pixel in the histogram equalized image into a corresponding pixel in the newly created histogram-specified image. The values of the resulting histogram are listed in the third column of Table 3.2, and the histogram is shown in Fig. 3.22(d). The values of $p_z(z_q)$ were obtained using the same procedure as in Example 3.5. For instance, we see in Table 3.4 that $s_k = 1$ maps to $z_q = 3$, and there are 790 pixels in the histogram-equalized image with a value of 1. Therefore, $p_z(z_3) = 790/4096 = 0.19$.



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 $p_z(z_q)$	Actual $p_z(z_q)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

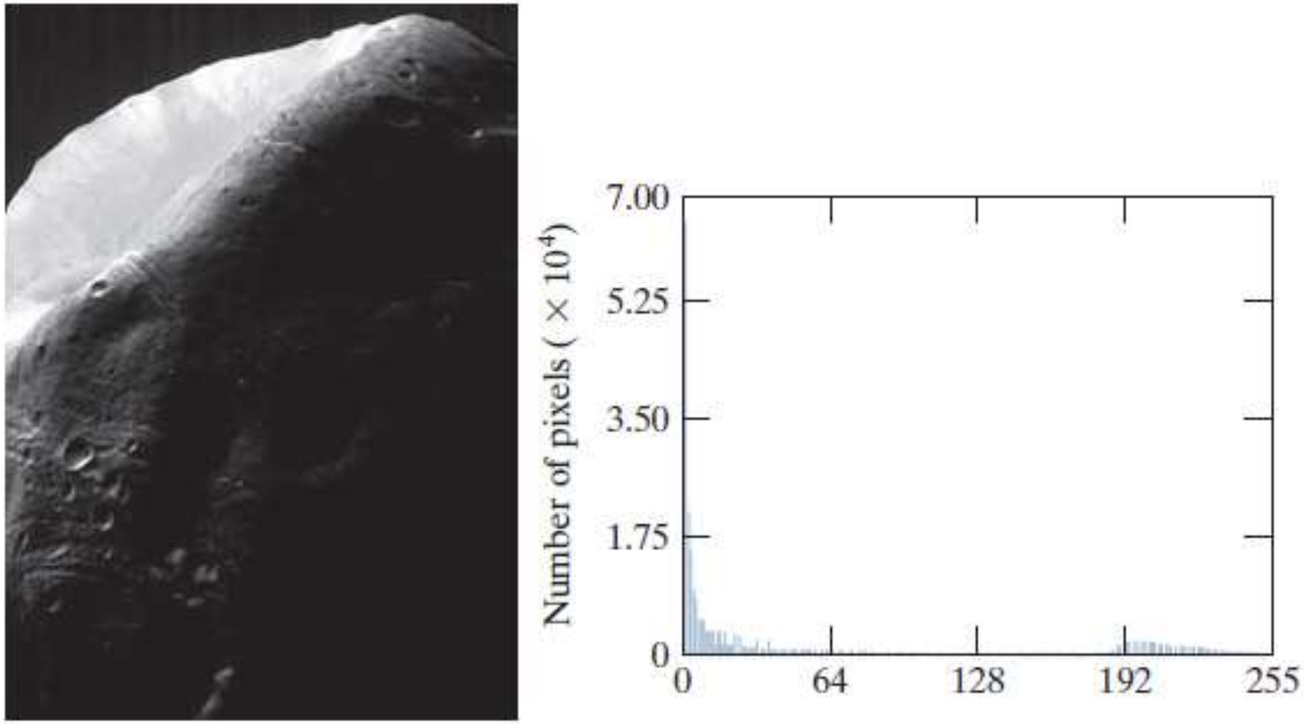
TABLE 3.2 Specified and actual histograms (the values in the third column are computed in Example 3.7).

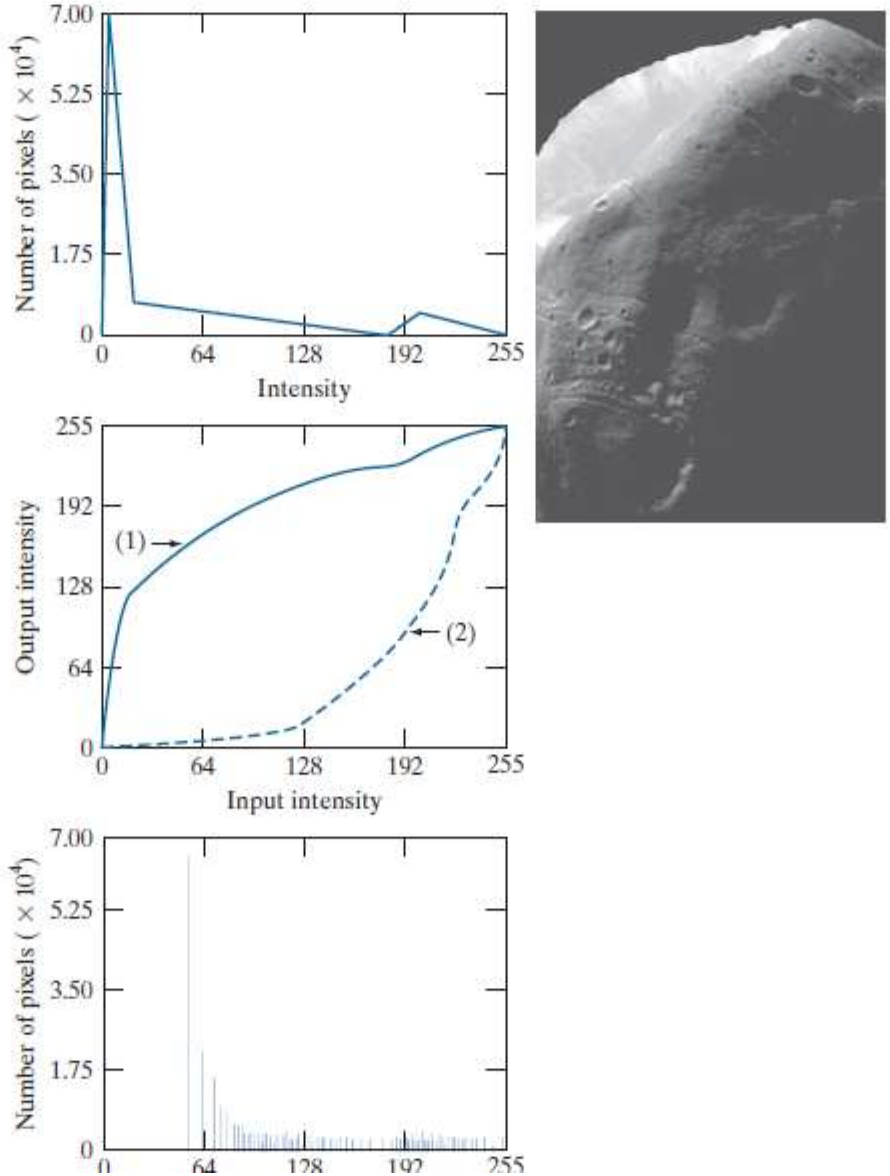
- Although the final result in Fig. 3.22(d) does not match the specified histogram exactly, the general trend of moving the intensities toward the high end of the intensity scale definitely was achieved.
- As mentioned earlier, obtaining the histogram-equalized image as an intermediate step is useful for explaining the procedure, but this is not necessary.
- Instead, we could list the mappings from the r's to the s's and from the s's to the z's in a three-column table.
- Then, we would use those mappings to map the original pixels directly into the pixels of the histogram-specified image.

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

a b

FIGURE 3.23
(a) An image, and
(b) its histogram.





a c
b
d

FIGURE 3.25 Histogram specification. (a) Specified histogram. (b) Transformation $G(z_q)$, labeled (1), and $G^{-1}(s_k)$, labeled (2). (c) Result of histogram specification. (d) Histogram of image (c).

Local Histogram Processing

- The histogram processing methods discussed thus far are *global*, in the sense that pixels are modified by a transformation function based on the intensity distribution of an entire image.
- This global approach is suitable for overall enhancement, but generally fails when the objective is to enhance details over small areas in an image.
- This is because the number of pixels in small areas have negligible influence on the computation of global transformations.
- The solution is to devise transformation functions based on the intensity distribution of pixel neighborhoods.
- The histogram processing techniques previously described can be adapted to local enhancement.

Local Histogram Processing

- The procedure is to define a neighborhood and move its center from pixel to pixel in a horizontal or vertical direction.
- At each location, the histogram of the points in the neighborhood is computed, and either a histogram equalization or histogram specification transformation function is obtained.
- This function is used to map the intensity of the pixel centered in the neighborhood.
- The center of the neighborhood is then moved to an adjacent pixel location and the procedure is repeated.

Local Histogram Processing

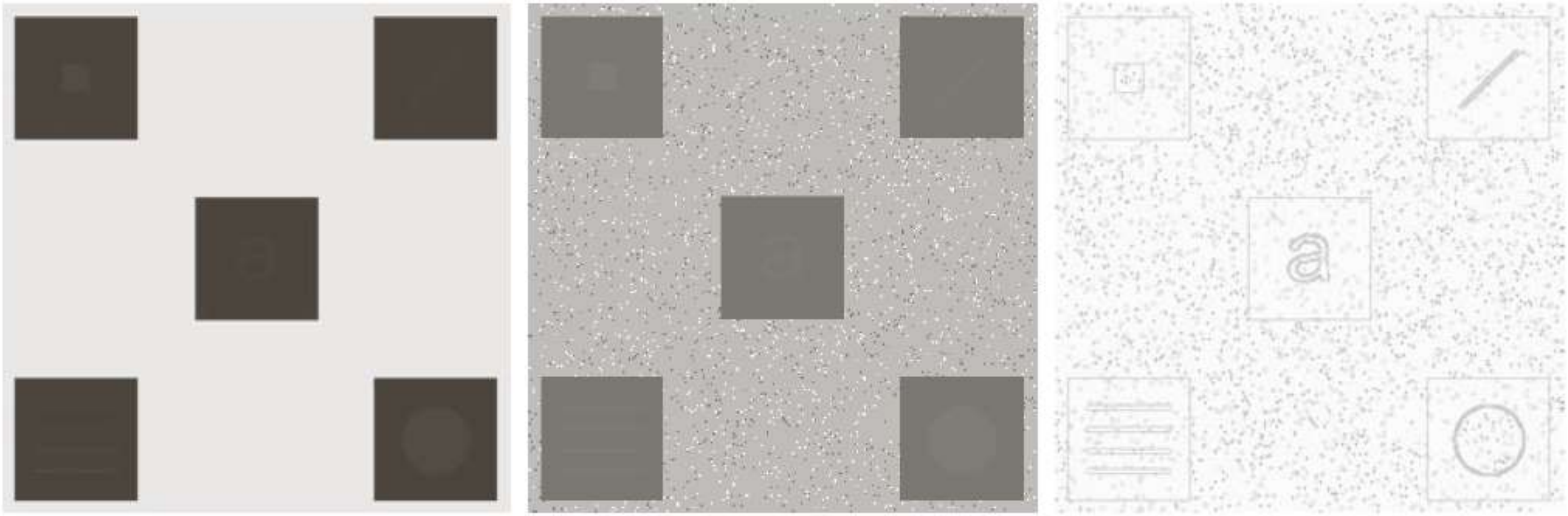


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.