# DIGITAL IMAGE PROCESSING

LECTURE-2

SPATIAL FILTERING

SYED ZUBAIR ALI

# DIGITAL IMAGE PROCESSING

## *Spatial Filtering*

- *A filter that passes low frequencies is called a lowpass filter.*
- *The net effect produced by a lowpass filter is to blur (smooth) an image.*
- *A similar smoothing can be accomplished directly on the image itself by using spatial filters (also called spatial masks, kernels, templates, and windows).*
- *They can be used also for nonlinear filtering, something we cannot do in the frequency domain.*

# DIGITAL IMAGE PROCESSING

## *The Mechanics of Spatial Filtering*

*Spatial filter consists of*

*(1) a neighborhood, (typically a small rectangle)*

*(2) a predefined operation that is performed on the image pixels encompassed by the neighborhood.*

*(3) Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighborhood, and whose value is the result of the filtering operation.†*

*A processed (filtered) image is generated as the center of the filter visits each pixel in the input image.*

*If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter.*

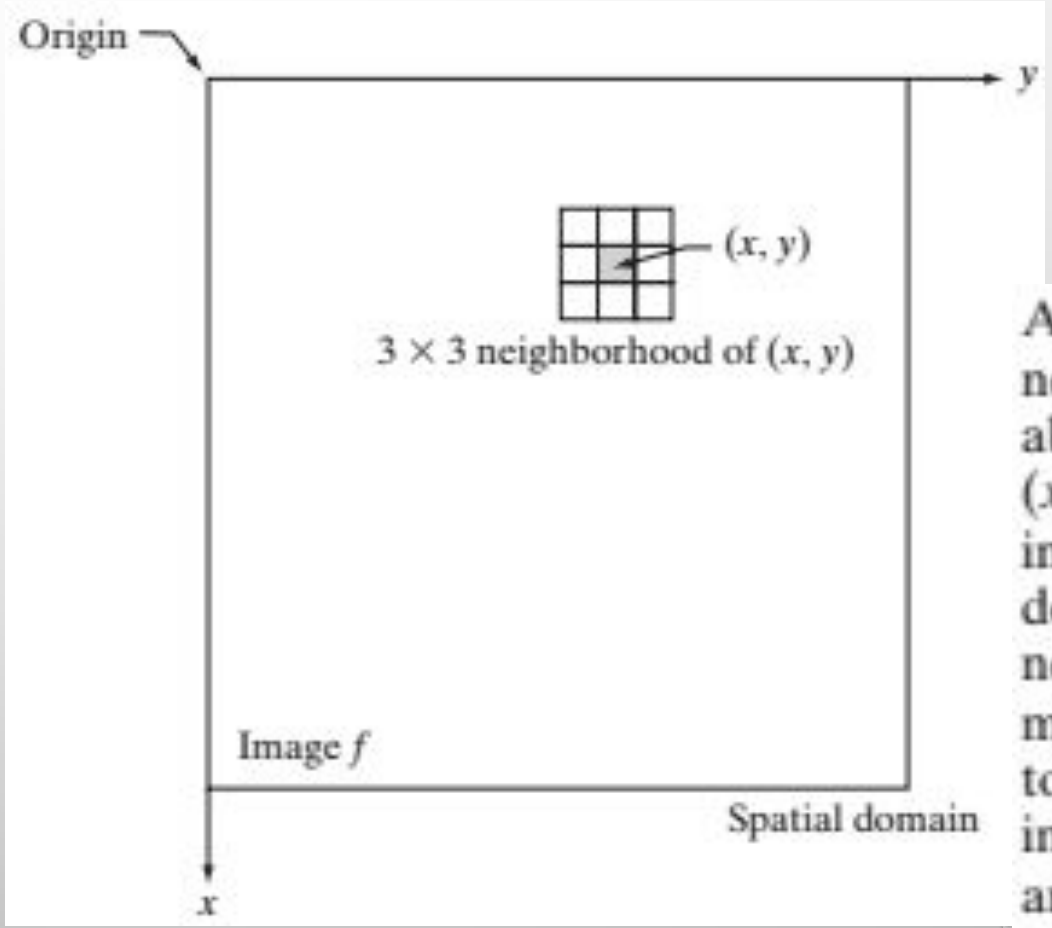
# DIGITAL IMAGE PROCESSING

## *The Mechanics of Spatial Filtering*

*The spatial domain processes can be denoted by the expression*

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

*where $f(x, y)$ is the input image, is the output image, and T is an operator on f defined over a neighborhood of point (x, y). The point (x, y) shown is an arbitrary location in the image, and the small region shown containing the point is a neighborhood of (x, y), is rectangular, centered on (x, y), and much smaller in size than the image.*




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.

# DIGITAL IMAGE PROCESSING
## *Enhancement*

***Enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application.***

# DIGITAL IMAGE PROCESSING

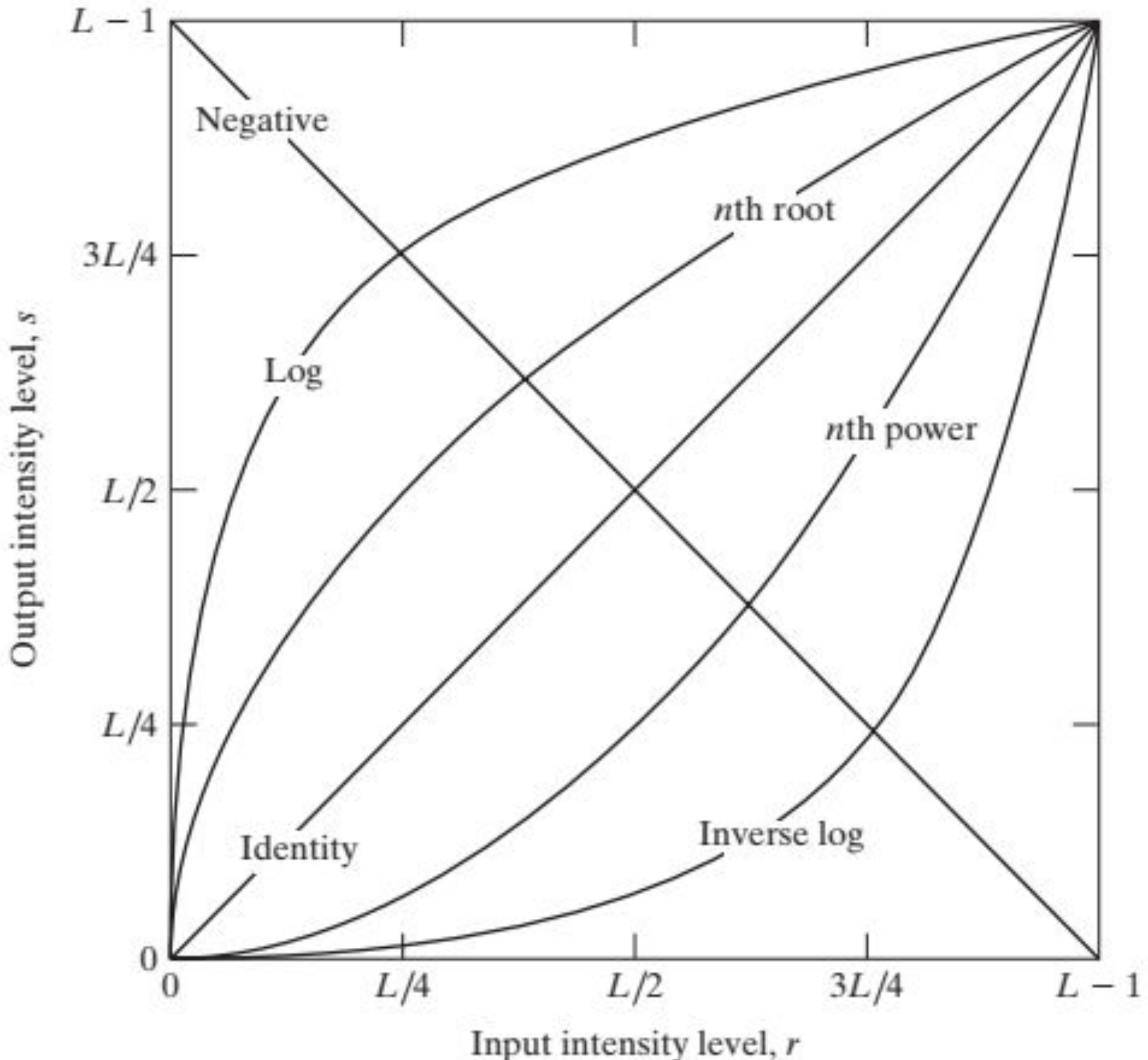
## Image Negatives

The negative of an image with intensity levels in the range is obtained by using the negative transformation

$$s = L - 1 - r$$

*Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image.*

## Log Transformations




Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

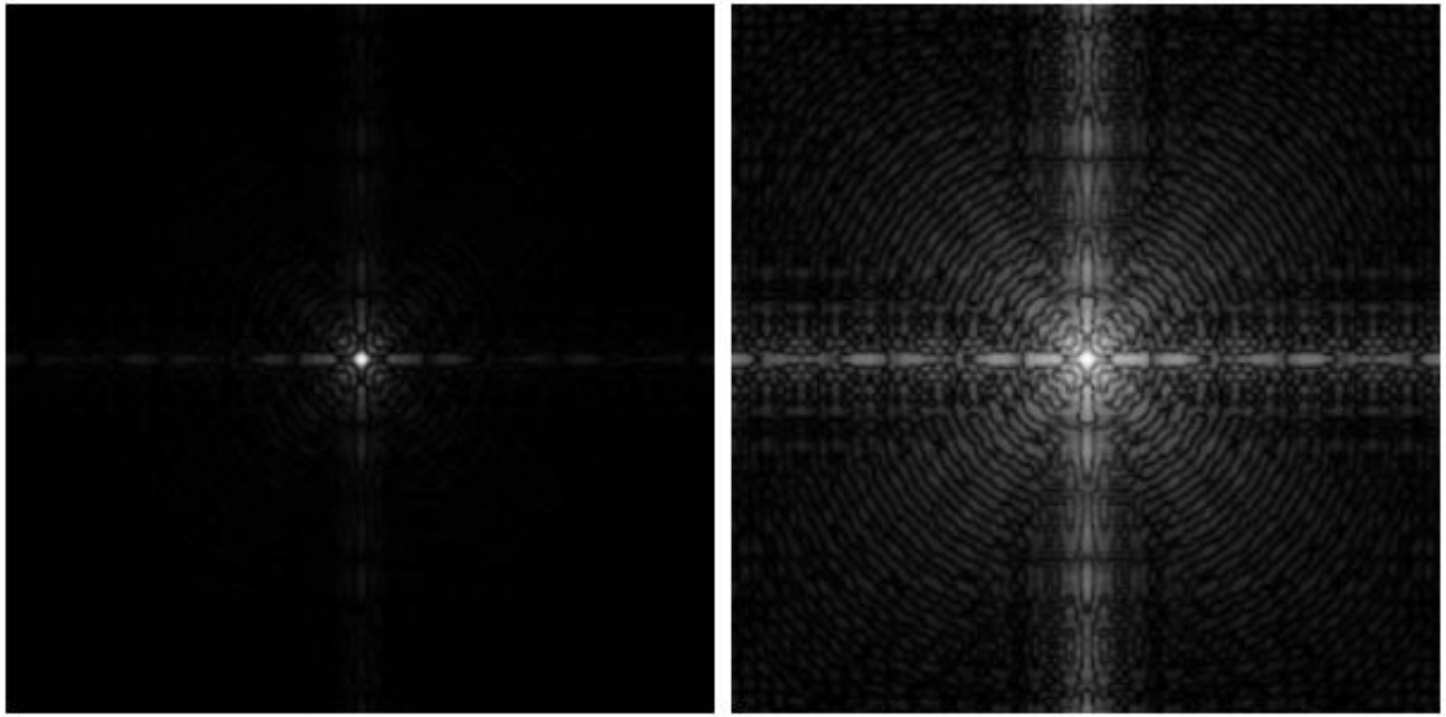
# DIGITAL IMAGE PROCESSING

## Log Transformations

*The general form of the log transformation*
$s = c\log(1 + r)$

(a) Fourier
spectrum.
(b) Result of
applying the log
transformation i

Intensity Transformations and Spatial Filtering

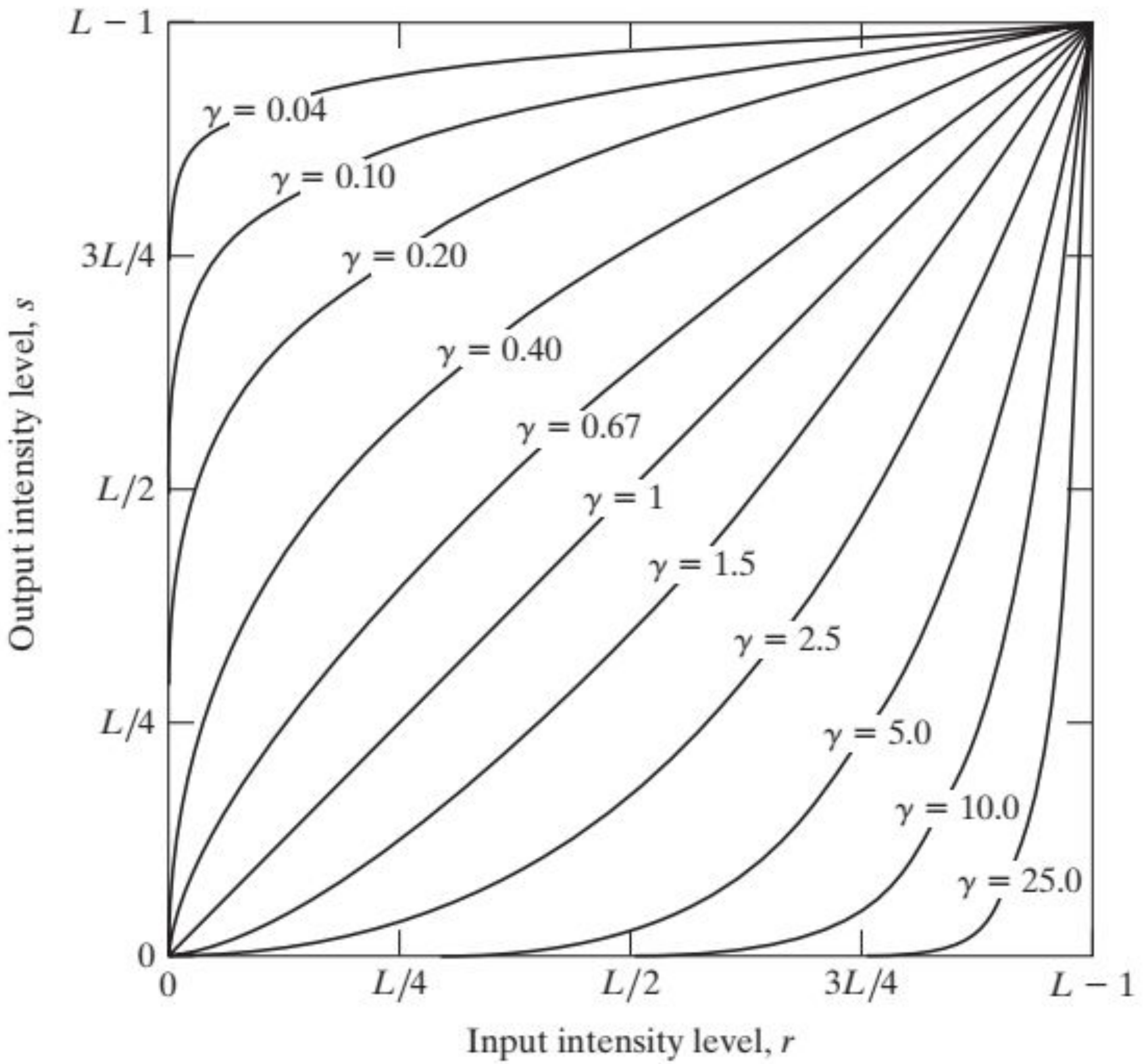


# DIGITAL IMAGE PROCESSING

## Power-Law (Gamma) Transformations

$s = cr^{\gamma}$

where $c$ and $\gamma$ are positive constants. Sometimes the equation is written as $s = (c + \varepsilon)^{\gamma}$ to account for an offset (that is, a measurable output when the input is zero).



# DIGITAL IMAGE PROCESSING

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.)

# DIGITAL IMAGE PROCESSING


**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.)

# Contrast stretching

- One of the simplest piecewise linear functions is a contrast-stretching transformation.
- Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even the wrong setting of a lens aperture during image acquisition.
- ***Contrast stretching*** is a 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.

## Piecewise-Linear Transformation Functions

a b
c d

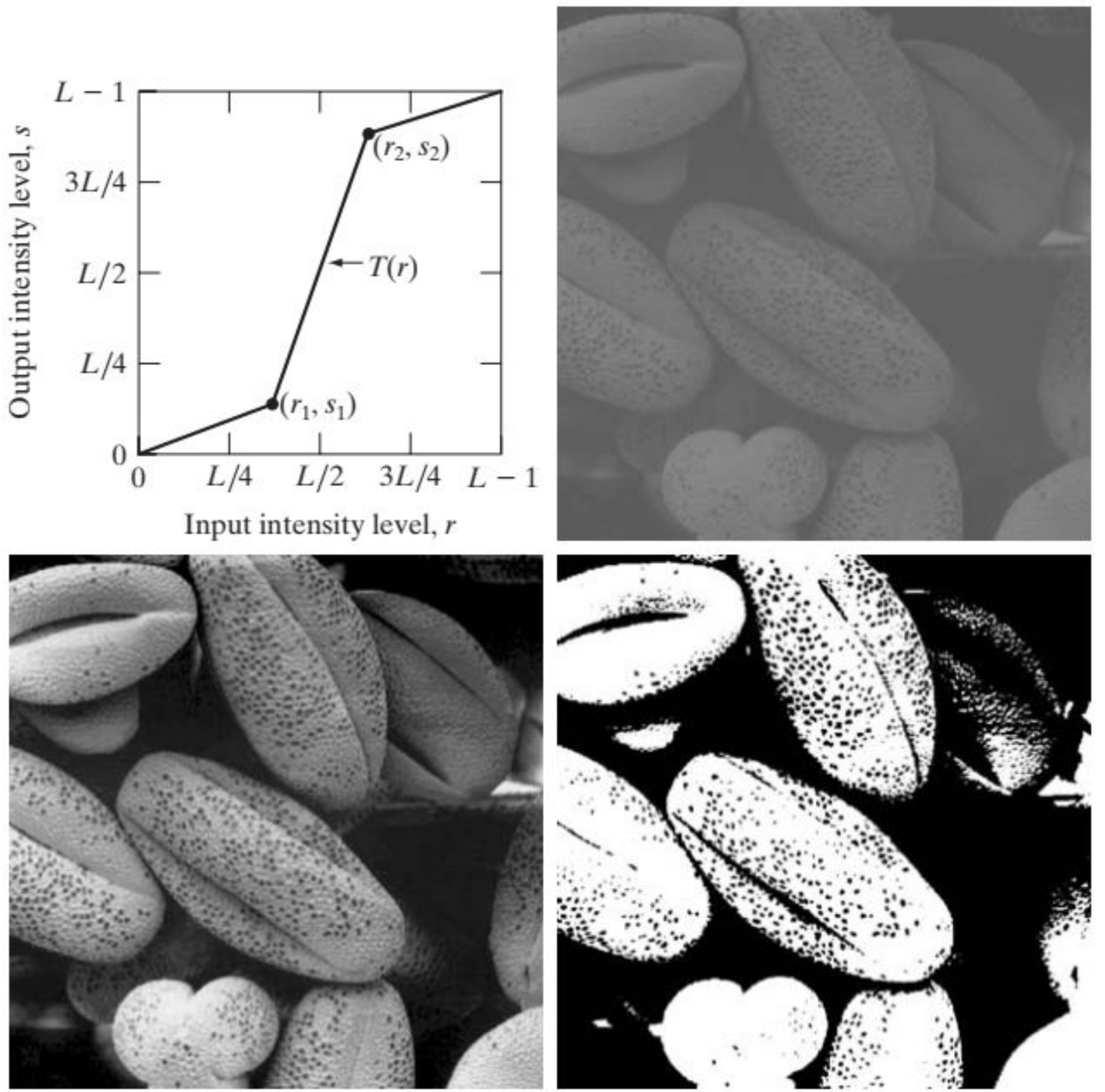
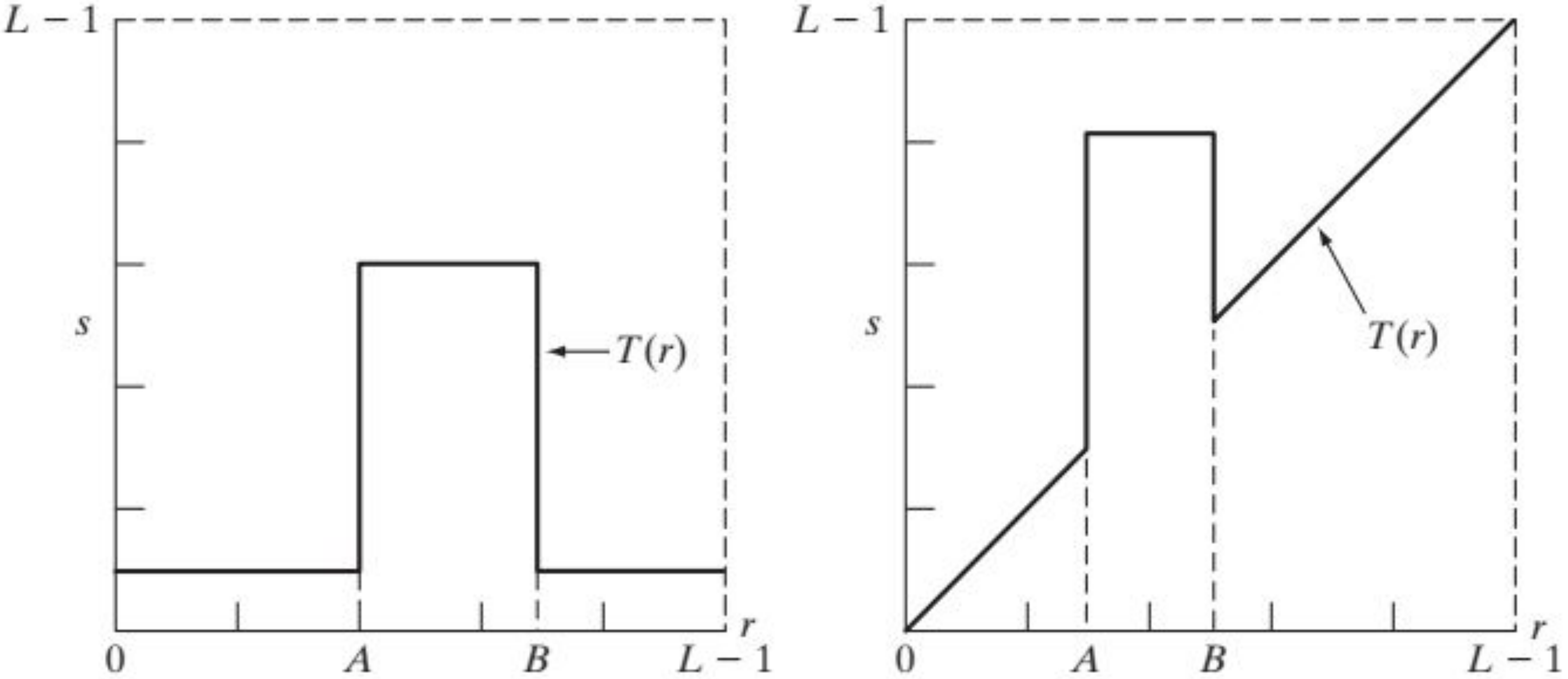



**FIGURE 3.10** Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

# Intensity-level slicing Piecewise-Linear Transformation Functions

Highlighting a specific range of intensities in an image often is of interest. Applications include enhancing features such as masses of water in satellite imagery
and enhancing flaws in X-ray images. The process, often called intensity-level slicing.
This transformation, shown in Fig. 3.11(a), produces a binary image. The second approach, based on the transformation in Fig. 3.11(b), brightens (or darkens) the desired range of intensities but leaves all other intensity levels in the image unchanged.




a b

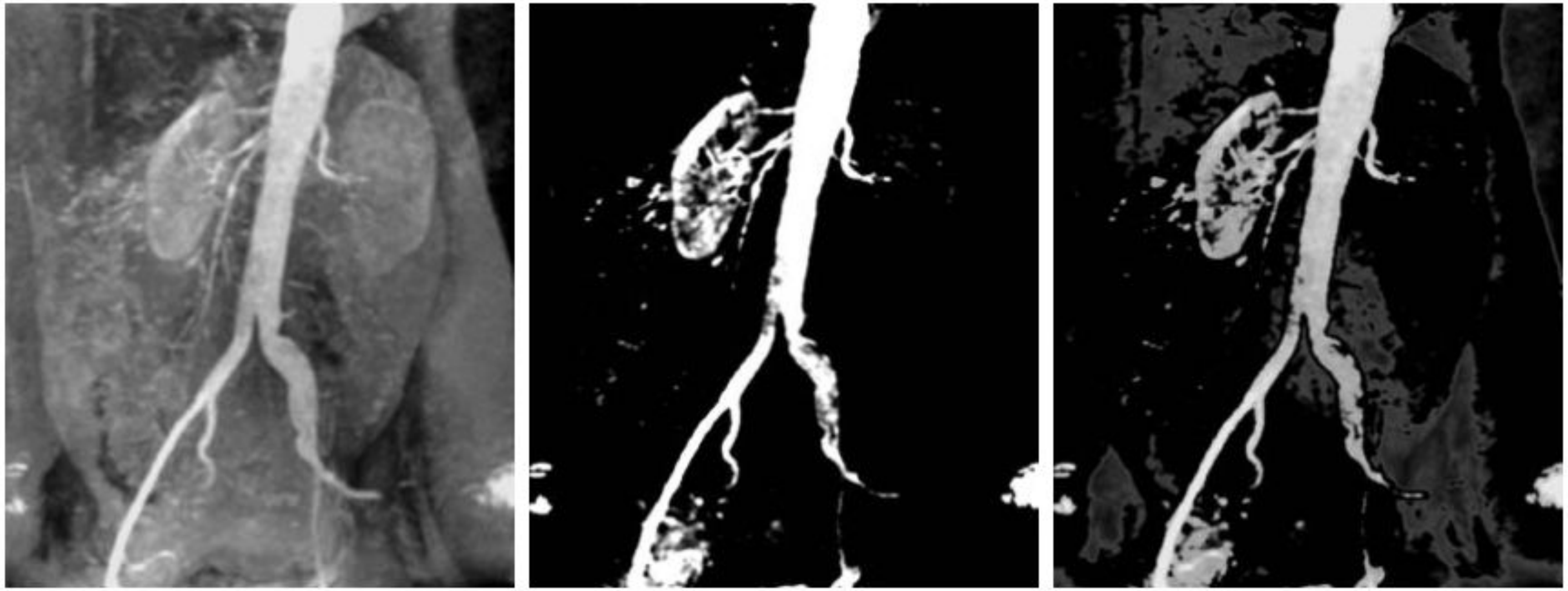
**FIGURE 3.11** (a) This transformation highlights intensity range [$A$, $B$] and reduces all other intensities to a lower level. (b) This transformation highlights range [$A$, $B$] and preserves all other intensity levels.

## Bit-plane slicing

# DIGITAL IMAGE PROCESSING

### Piecewise-Linear Transformation Functions

Pixels are digital numbers composed of bits. For example, the intensity of each pixel in a 256-level gray-scale image is composed of 8 bits (i.e., one byte). Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.



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

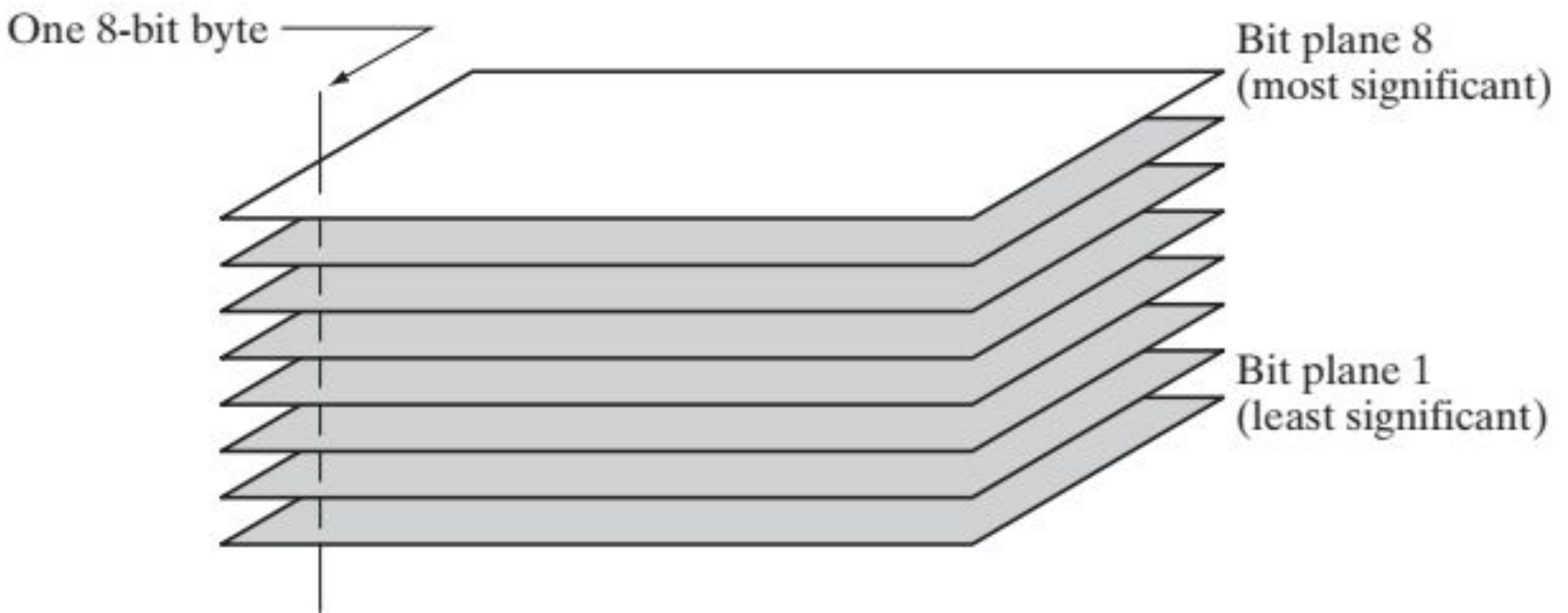
**Bit-plane slicing**

# DIGITAL IMAGE PROCESSING

**Piecewise-Linear Transformation Functions**

*Fig. 3.13 illustrates, an 8-bit image may be considered as being composed of eight 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane 8 all the highest-order bits.*

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



## Bit-plane slicing

# DIGITAL IMAGE PROCESSING

### Piecewise-Linear Transformation Functions

Figure 3.14(a) shows an 8-bit gray-scale image and Figs. 3.14(b) through (i) are its eight 1-bit planes, with Fig. 3.14(b) corresponding to the lowest-order bit.

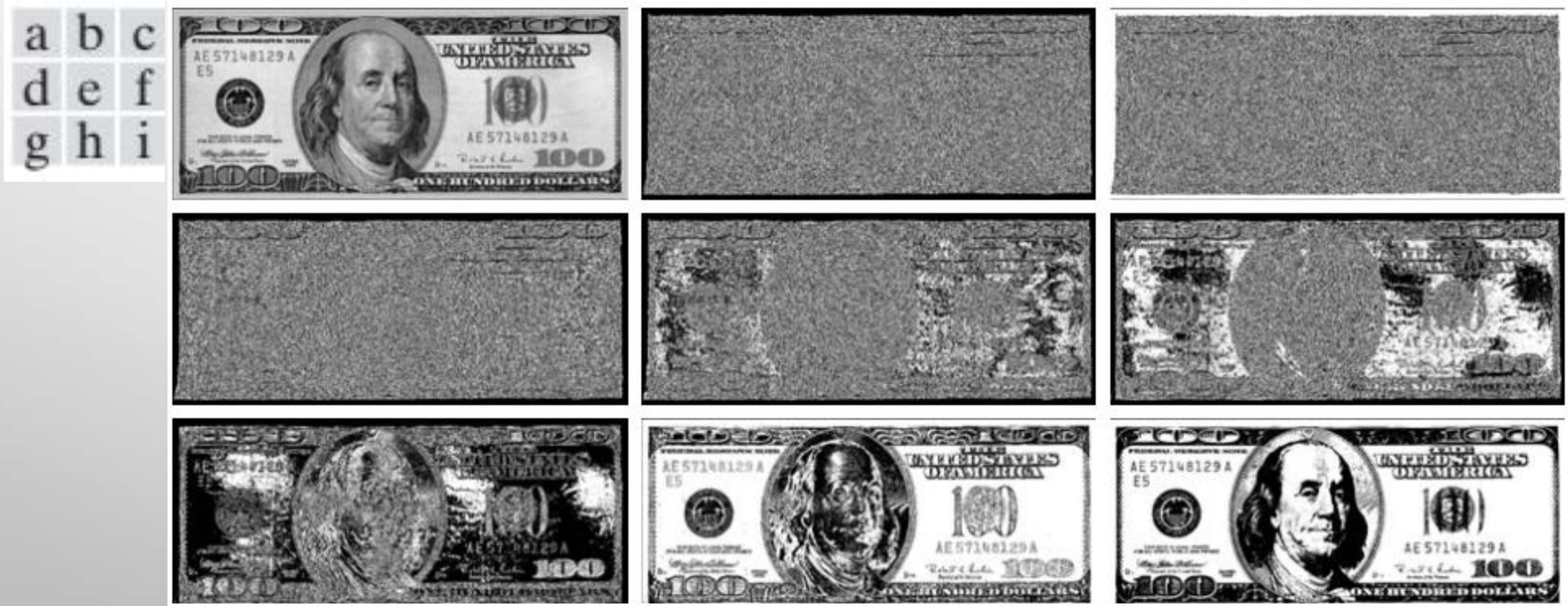



**FIGURE 3.14** (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

**Bit-plane slicing**

# DIGITAL IMAGE PROCESSING

## Piecewise-Linear Transformation Functions

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).

Using more planes in the reconstruction would not contribute significantly to the appearance of this image. Thus, we conclude that storing the four highest-order bit planes would allow us to reconstruct the original image in acceptable detail. Storing these four planes instead of the original image requires 50% less storage (ignoring memory architecture issues).

# DIGITAL IMAGE PROCESSING

## Histogram Processing

The *histogram* of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where $r_k$ is the $k$th intensity value and $n_k$ is the number of pixels in the image with intensity $r_k$. It is common practice to normalize a histogram by dividing each of its components by the total number of pixels in the image, denoted by the product $MN$, where, as usual, $M$ and $N$ are the row and column dimensions of the image. Thus, a normalized histogram is given by $p(r_k) = r_k/MN$, for $k = 0, 1, 2, \dots, L-1$. Loosely speaking, $p(r_k)$ is an estimate of the probability of occurrence of intensity level $r_k$ in an image. The sum of all components of a normalized histogram is equal to 1.

# DIGITAL IMAGE PROCESSING

## Histogram Processing

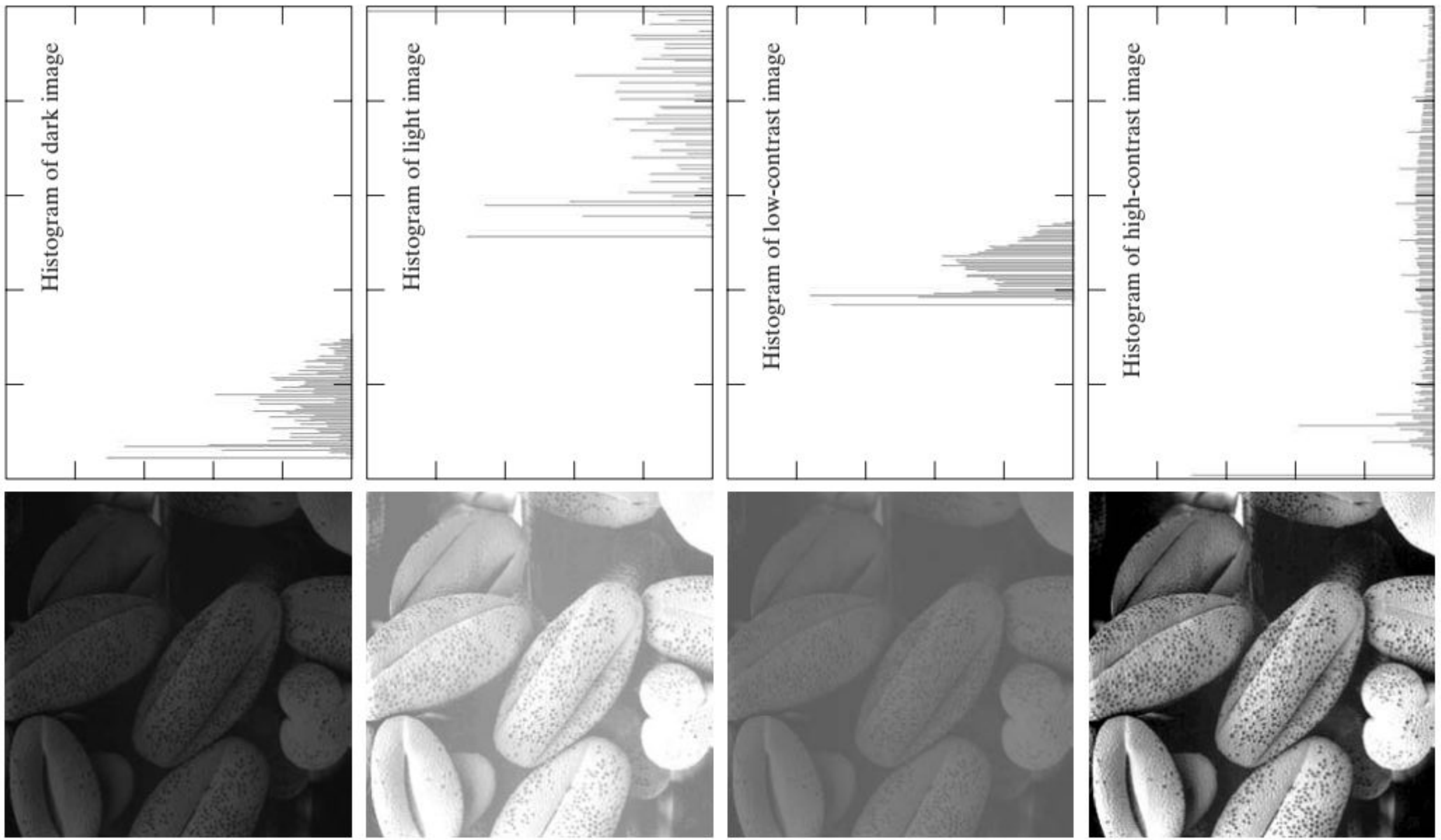



**FIGURE 3.16** Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

# Histogram Processing



earlier, the probability of occurrence of intensity level $r_k$ in a digital image is approximated by

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, 2, \ldots, L - 1 \tag{3.3-7}$$

where $MN$ is the total number of pixels in the image, $n_k$ is the number of pixels that have intensity $r_k$, and $L$ is the number of possible intensity levels in the image (e.g., 256 for an 8-bit image). As noted in the beginning of this section, a plot of $p_r(r_k)$ versus $r_k$ is commonly referred to as a *histogram*.

The discrete form of the transformation in Eq.

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

*The transformation (mapping) in this equation is called a* ***histogram equalization*** *or* ***histogram linearization*** *transformation.*

**EXAMPLE 3.5:**
A simple illustration of histogram equalization.

# DIGITAL IMAGE PROCESSING

## Histogram Processing

Suppose that a 3-bit image ($L = 8$) of size $64 \times 64$ pixels ($MN = 4096$) has the intensity distribution shown in Table 3.1, where the intensity levels are integers in the range $[0, L - 1] = [0, 7]$.

The histogram of our hypothetical image is sketched in Fig. 3.19(a). Values of the histogram equalization transformation function are obtained using Eq. (3.3-8). For instance,

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

Similarly,

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

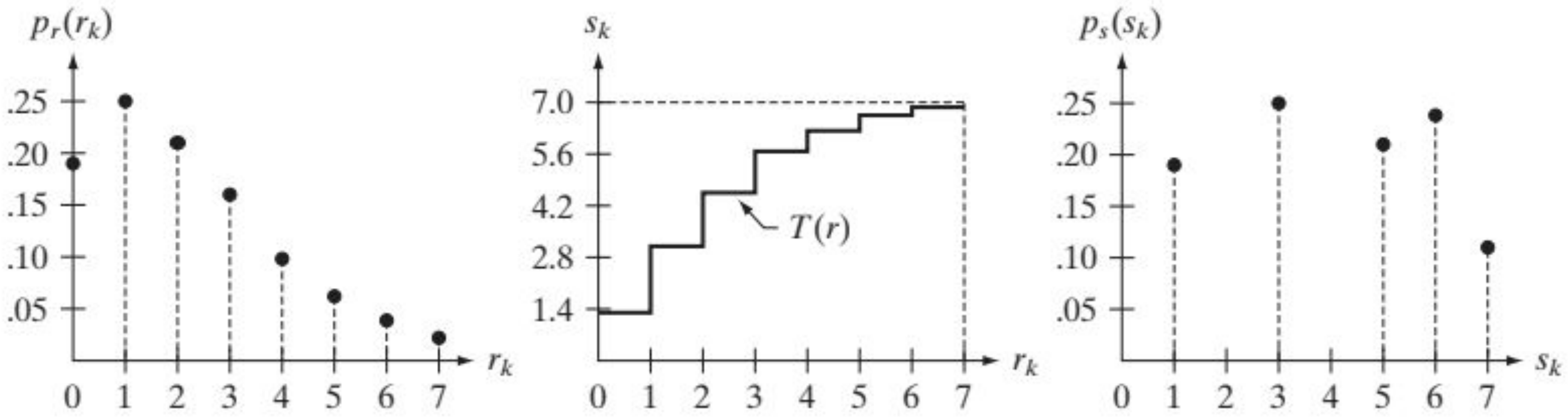
and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

**TABLE 3.1**
Intensity distribution and histogram values for a 3-bit, $64 \times 64$ 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 |

# DIGITAL IMAGE PROCESSING

**EXAMPLE 3.5:** A simple illustration of histogram equalization.




a b c

**FIGURE 3.19** Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

At this point, the $s$ values still have fractions because they were generated by summing probability values, so we round them to the nearest integer:

$$s_0 = 1.33 \rightarrow 1 \qquad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \qquad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \qquad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \qquad s_7 = 7.00 \rightarrow 7$$

**EXAMPLE 3.5:** A simple illustration of histogram equalization.

# DIGITAL IMAGE PROCESSING

## Histogram Processing

These are the values of the equalized histogram. Observe that there are 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 in this image 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

**TABLE 3.1** Intensity distribution and histogram values for a 3-bit, $64 \times 64$ 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 |

Histogram Processing

# DIGITAL IMAGE PROCESSING

Quiz Question: Histogram Equalization of a 3-bit Image

Consider a 3-bit digital image with dimensions $64 \times 64$ (4096 pixels) and the following intensity distribution:

| Intensity Level $r_k$ | Number of Pixels $n_k$ | Probability $p_r(r_k)$ |
|---|---|---|
| $r_0 = 0$ | 820 | 0.20 |
| $r_1 = 1$ | 980 | 0.24 |
| $r_2 = 2$ | 760 | 0.19 |
| $r_3 = 3$ | 630 | 0.15 |
| $r_4 = 4$ | 345 | 0.08 |
| $r_5 = 5$ | 245 | 0.06 |
| $r_6 = 6$ | 180 | 0.04 |
| $r_7 = 7$ | 136 | 0.03 |

## Steps:

1. **Calculate the cumulative distribution function (CDF) for each intensity level** using the formula:

$$T(r_k) = (L - 1) \times \sum_{j=0}^{k} p_r(r_j)$$

where $L = 8$ (the number of intensity levels for a 3-bit image)

# DIGITAL IMAGE PROCESSING

2. **Compute the transformed (equalized) intensity levels $s_k$** based on the CDF and round them to the nearest integer.
3. **Map the original intensity levels $r_k$ to the new intensity levels $s_k$** and determine the number of pixels for each equalized intensity level.
4. **Sketch the original and equalized histograms.**

# DIGITAL IMAGE PROCESSING

1. **CDF Calculation:**

   For $r_0 = 0$, the CDF is calculated as follows:

   $$T(r_0) = (L - 1) \times p_r(r_0) = 7 \times 0.20 = 1.4 \Rightarrow s_0 = 1$$

   Similarly, calculate for other intensity levels:

   $$T(r_1) = 7 \times (0.20 + 0.24) = 3.08 \Rightarrow s_1 = 3$$

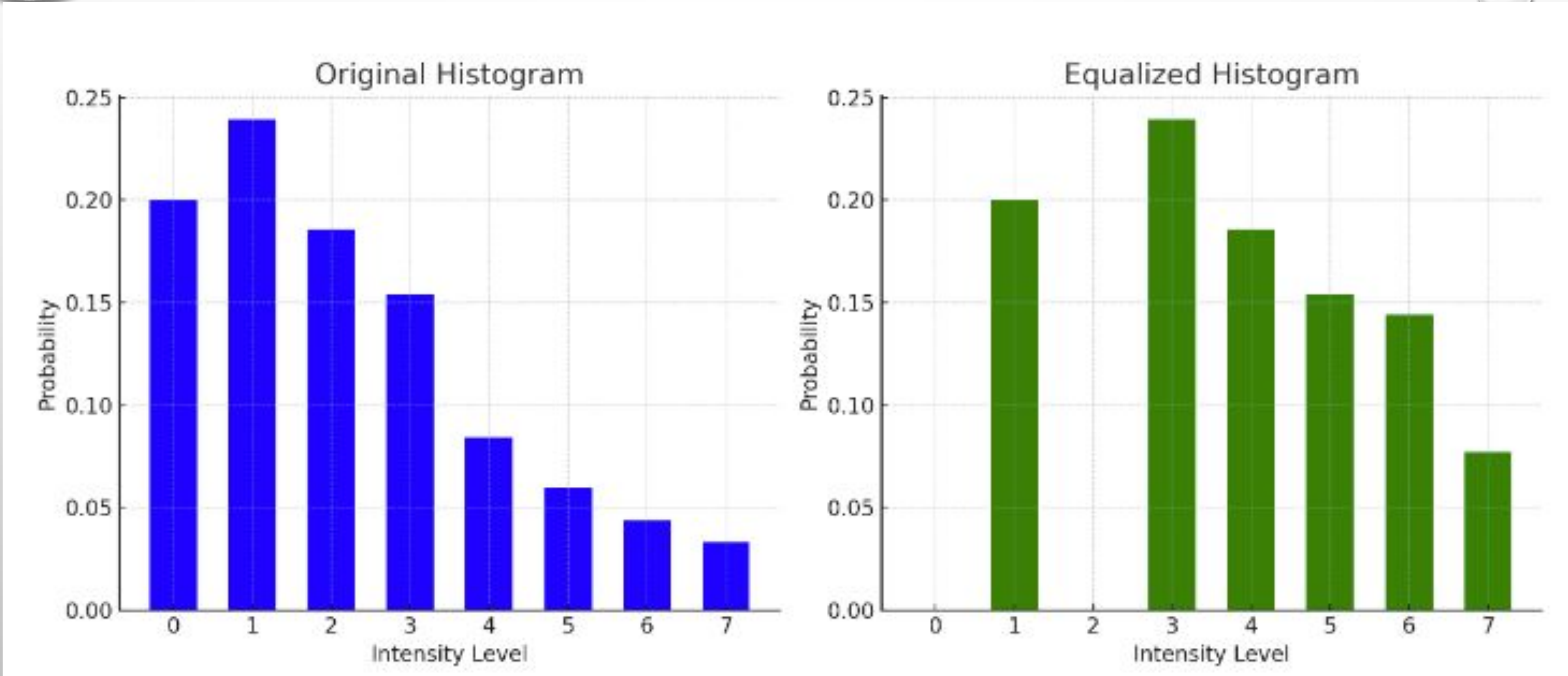
   $$T(r_2) = 7 \times (0.20 + 0.24 + 0.19) = 4.41 \Rightarrow s_2 = 4$$

   $$T(r_3) = 7 \times (0.20 + 0.24 + 0.19 + 0.15) = 5.46 \Rightarrow s_3 = 5$$

   (Continue for other values.)

# Histogram Processing

# DIGITAL IMAGE PROCESSING



# Histogram Processing

# DIGITAL IMAGE PROCESSING

## Step 2: Cumulative Distribution Function (CDF) and Equalized Intensity Levels

Using the CDF calculation formula, we found the following values for the equalized intensity levels:

$$
\begin{aligned}
s_0 &= 7 \times 0.200 = 1.4 \approx 1, \\
s_1 &= 7 \times 0.439 = 3.08 \approx 3, \\
s_2 &= 7 \times 0.625 = 4.41 \approx 4, \\
s_3 &= 7 \times 0.779 = 5.46 \approx 5, \\
s_4 &= 7 \times 0.863 = 6.04 \approx 6, \\
s_5 &= 7 \times 0.923 = 6.46 \approx 6, \\
s_6 &= 7 \times 0.967 = 6.77 \approx 7, \\
s_7 &= 7 \times 1.000 = 7.00 \approx 7.
\end{aligned}
$$

***Step 3: Mapping and Histogram Plotting***

*We mapped the original intensity levels $r_k$ to their new equalized values $S_k$, and then calculated the number of pixels for each equalized intensity level.*

- ***Original Histogram****: The original histogram is displayed in the left graph, showing the probability distribution of intensity levels in the image.*
- ***Equalized Histogram****: The equalized histogram (right graph) shows the transformation effect, where the intensity levels are spread out over a wider range, resulting in contrast enhancement.*

*In the equalized histogram, the intensity levels are distributed more evenly, improving image contrast.*

# THANK YOU