Some terms

A Simple Image Formation Model

- Images by two-dimensional functions of the form f(x, y).
- The value or amplitude of f at spatial coordinates (x, y) gives the intensity (brightness) of the image at that point.
- As light is a form of energy, f(x,y) must be non zero and finite.

- The function f(x, y) may be characterized by two components:

(1)the amount of source illumination incident on the scene being viewed

(2)the amount of illumination reflected by the objects in the scene.

- These are called the *illumination and reflectance components and are denoted by i(x, y) and r(x, y), respectively.*

- The two functions combine as a product to form $f(x, y)$:

$$f(x, y) = i(x, y)\, r(x, y)$$

$r(x, y) = 0$ --- total absorption

1 --- total reflection

- The intensity of a monochrome image f at any coordinates (x, y) the *gray level (l) of the image at that point.*

That is, $l = f(x_0, y_0)$

L lies in the range $L_{min} \leq \ell \leq L_{max}$

In practice, $L_{min} = i_{min} r_{min}$ and $L_{max} = i_{max} r_{max}$.

GRAY SCALE

- The interval [L_{min} , L_{max}] is called the *gray scale.*

- *Common practice is to shift this interval numerically to the interval [0, L-1],*

- *where L = 0 is considered black and*

L = L-1 is considered white on the gray scale.

All intermediate values are shades of gray varying from black to white.

GRAY SCALE

- The interval [L_{min} , L_{max}] is called the*gray scale.*

- *Common practice is to shift this interval numerically to the interval [0, L-1],*

- *where L = 0 is considered black and*

L = L-1 is considered white on the gray scale.

All intermediate values are shades of gray varying from black to white.

Terminology

- Pixel
- Pixel is the smallest element of an image.
- Each pixel correspond to any one value.
- In an 8-bit gray scale image, the value of the pixel between 0 and 255.
- The value of a pixel at any point correspond to the intensity of the light photons striking at that point.
- Each pixel store a value proportional to the light intensity at that particular location.

Terminology

- Gray level
- The value of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level.

- Pixel value.(0)
- that each pixel can have only one value and each value denotes the intensity of light at that point of the image.
- The value 0 means absence of light. It means that 0 denotes dark, and it further means that when ever a pixel has a value of 0, it means at that point, black color would be formed.

Terminology

- Look at this image matrix
- 000
- 000
- 000
- Now this image matrix has all filled up with 0. All the pixels have a value of 0. If we were to calculate the total number of pixels form this matrix, this is how we are going to do it.
- Total no of pixels = total no. of rows X total no. of columns
- = 3 X 3
- = 9.

Terminology

- Bpp or bits per pixel denotes the number of bits per pixel.
- The number of different colors in an image is depends on the depth of color or bits per pixel.

Some terms

- its in mathematics:
- Its just like playing with binary bits.
- How many numbers can be represented by one bit.
- 0
- 1
- How many two bits combinations can be made.
- 00
- 01
- 10
- 11
- If we devise a formula for the calculation of total number of combinations that can be made from bit, it would be like this.
- Where bpp denotes bits per pixel. Put 1 in the formula you get 2, put 2 in the formula, you get 4. It grows exponentially.

Some terms

- Number of different colors:
- Now as we said it in the beginning, that the number of different colors depend on the number of bits per pixel.
- The table for some of the bits and their color is given below.
- Bits per pixel
- Number of colors 1 bpp2 colors
- 2 bpp 4 colors
- 3 bpp 8 colors
- 4 bpp 16 colors
- 5 bpp 32 colors
- 6 bpp 64 colors
- 7 bpp 128 colors
- 8 bpp 256 colors
- 10 bpp1024 colors16 bpp65536 colors24 bpp16777216 colors (16.7 million colors)32 bpp4294967296 colors (4294 million colors)

Some terms

- Shades
- You can easily notice the pattern of the exponentional growth. The famous gray scale image is of 8 bpp , means it has 256 different colors in it or 256 shades.

- Color images are usually of the 24 bpp format, or 16 bpp.
- We will see more about other color formats and image types in the tutorial of image types.
- 0 pixel value denotes black color.
- 0 pixel value always denotes black color. But there is no fixed value that denotes white color.
- White color:
- The value that denotes white color can be calculated as :

$$\text{White color} = (2)^{bpp} - 1$$

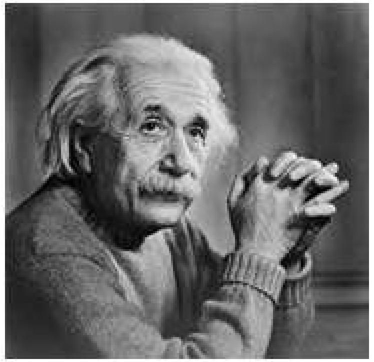
Image storage requirements

- After the discussion of bits per pixel, now we have every thing that we need to calculate a size of an image.
- Image size
- The size of an image depends upon three things.
- Number of rows
- Number of columns
- Number of bits per pixel
- The formula for calculating the size is given below.
- Size of an image = rows * cols * bpp

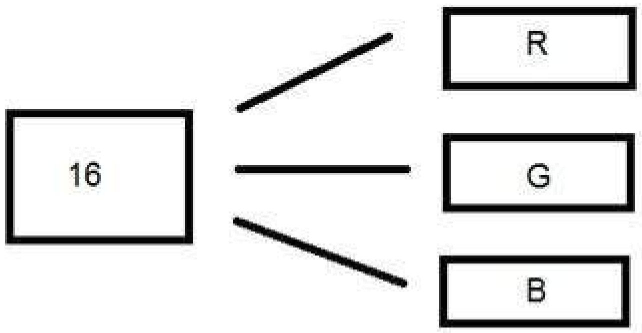
The binary image

- The binary image as it name states, contain only two pixel values.
- 0 and 1.
- Here 0 refers to black color and 1 refers to white color. It is also known as Monochrome.
- Black and white image:
- The resulting image that is formed hence consist of only black and white color and thus can also be called as Black and White image.
- No gray level
- One of the interesting this about this binary image that there is no gray level in it. Only two colors that are black and white are found in it.

- Format
- Binary images have a format of PBM (Portable bit map)
- 2, 3, 4,5, 6 bit color format
- The images with a color format of 2, 3, 4, 5 and 6 bit are not widely used today. They were used in old times for old TV displays, or monitor displays.
- But each of these colors have more then two gray levels, and hence has gray color unlike the binary image.
- In a 2 bit 4, in a 3 bit 8, in a 4 bit 16, in a 5 bit 32, in a 6 bit 64 different colors are present.
- 8 bit color format
- 8 bit color format is one of the most famous image format. It has 256 different shades of colors in it. It is commonly known as Grayscale image.
- The range of the colors in 8 bit vary from 0-255. Where 0 stands for black, and 255 stands for white, and 127 stands for gray color.
- This format was used initially by early models of the operating systems UNIX and the early color Macintoshes.
- A grayscale image of Einstein is shown below:
-

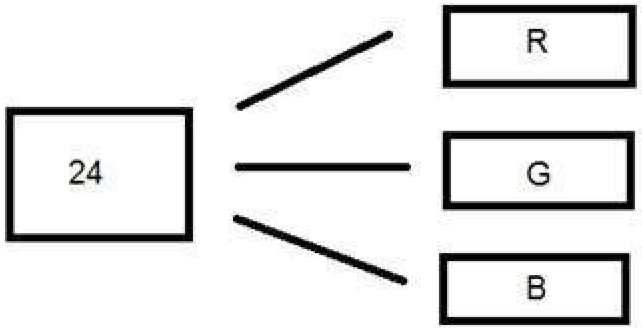


- 16 bit color format
- It is a color image format. It has 65,536 different colors in it. It is also known as High color format.
- It has been used by Microsoft in their systems that support more then 8 bit color format. Now in this 16 bit format and the next format we are going to discuss which is a 24 bit format are both color format.
- The distribution of color in a color image is not as simple as it was in grayscale image.
- A 16 bit format is actually divided into three further formats which are Red , Green and Blue. The famous (RGB) format.
- It is pictorially represented in the image below.


16
R
G
B

- 24 bit color format
- 24 bit color format also known as true color format. Like 16 bit color format, in a 24 bit color format, the 24 bits are again distributed in three different formats of Red, Green and Blue.

- 24 bit color format
- 24 bit color format also known as true color format. Like 16 bit color format, in a 24 bit color format, the 24 bits are again distributed in three different formats of Red, Green and Blue.


24
R
G
B

- Since 24 is equally divided on 8, so it has been distributed equally between three different color channels.
- Their distribution is like this.
- 8 bits for R, 8 bits for G, 8 bits for B.

- Behind a 24 bit image.
- Unlike a 8 bit gray scale image, which has one matrix behind it, a 24 bit image has three different matrices of R, G, B.

- Different color codes
- All the colors here are of the 24 bit format, that means each color has 8 bits of red, 8 bits of green, 8 bits of blue, in it. Or we can say each color has three different portions. You just have to change the quantity of these three portions to make any color.

- Binary color format
- Color:Black
- Image:
- Decimal Code:
- (0,0,0)

- Color:White
- Image:
- Decimal Code:
- (255,255,255)

- RGB color model:
- Color:Red
- Image:
- Decimal Code:
- (255,0,0)

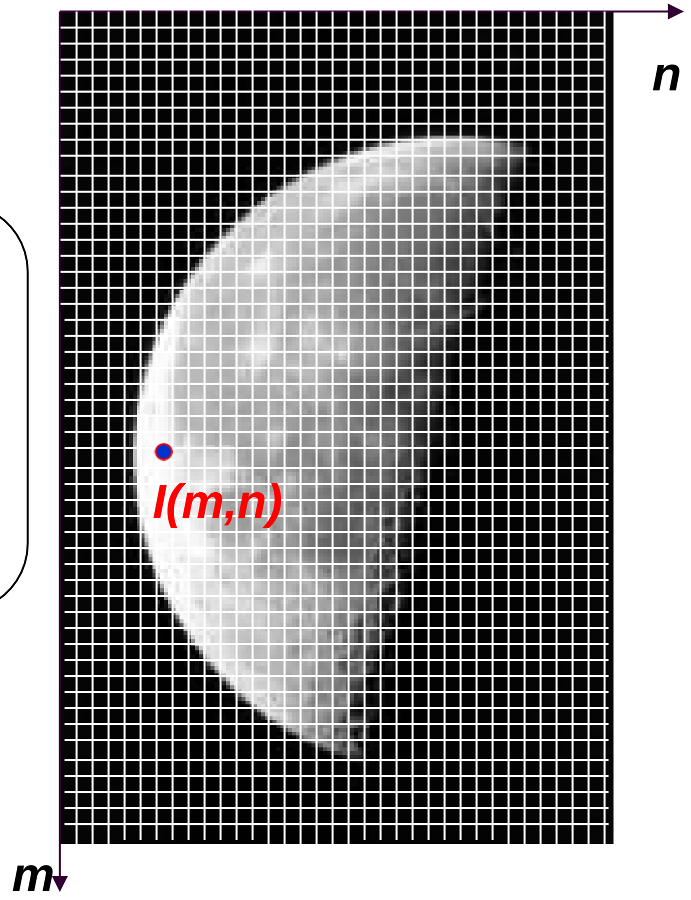
- Color:Green
- Image:
- green
- Decimal Code:
- (0,255,0)

- Color: Blue
- Image:
- Decimal Code:
- (0,0,255)

Digital Image Representation

- **A digital image can be represented as a two-dimensional matrix.**

$$
I(mn) = \begin{pmatrix}
i(1,1) & i(1,2) & \ldots\ldots. & i(1, n-1) \\
i(2,1) & i(2,2) & \ldots\ldots. & i(2, n-1) \\
. & & & . \\
. \; . & . \; . \; . \; . \; . \; . \; . & & . \; . \; . \\
. & & & . \\
i(m-1,1) & i(m-1,2) & \ldots & i(m-1, n-1)
\end{pmatrix}
$$



- **Each element is called a *pixel* (picture element).**
- **A color (RGB) image is represented by a 3-dimensional matrix I(mxnx3)**

Digital Image Size

- The size of a digital image is determines by its dimensions (M x N) multiplied by the number of bits b required to store the intensity levels ($L = 2^b$).

- image size = M x N x b (bits)

- Typical values of b are:
  - b = 1 black and white (binary) images.
  - b = 8 grayscale (256 gray levels), or indexed color images
  - b = 24 RGB color image.

Image Representation

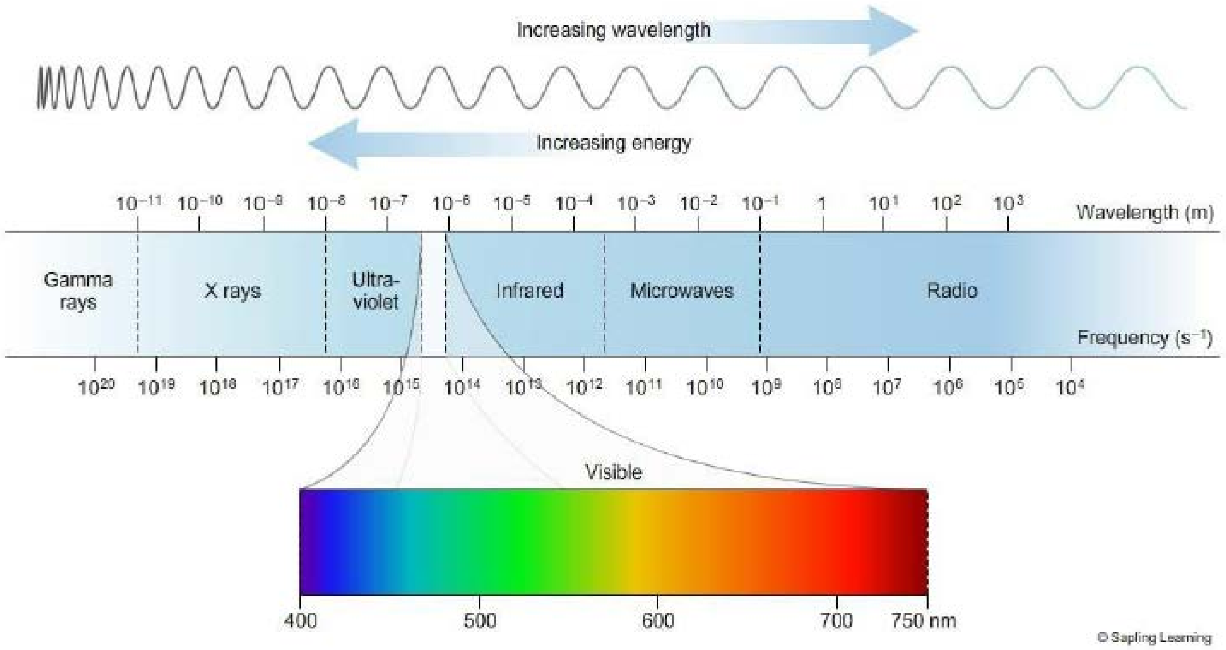
- Spatial discretization by grids
- Intensity discretization by quantization

Image Representation

$$
I = \begin{bmatrix}
f(0,0) & f(0,1) & f(0,2) \ldots\ldots\ldots & f(0,N-1) \\
f(1,0) & f(1,1) & f(1,2) \ldots\ldots\ldots & f(1,N-1) \\
f(2,0) & f(2,1) & f(2,2) \ldots\ldots\ldots & f(2,N-1) \\
. & . & . & . \\
. & . & . & . \\
. & . & . & . \\
. & . & . & . \\
f(M-1,0) & f(M,1) & f(M,2) .. & f(M-1,N-1)
\end{bmatrix}
$$

Image Size : 256x256, 512x512, 640x480, 1024x1024 etc

Quantization: 8 bits


Increasing wavelength
Increasing energy
Wavelength (m)
10^{-11} 10^{-10} 10^{-9} 10^{-8} 10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 1 10^{1} 10^{2} 10^{3}
Gamma rays
X rays
Ultra-violet
Infrared
Microwaves
Radio
Frequency (s^{-1})
10^{20} 10^{19} 10^{18} 10^{17} 10^{16} 10^{15} 10^{14} 10^{13} 10^{12} 10^{11} 10^{10} 10^{9} 10^{8} 10^{7} 10^{6} 10^{5} 10^{4}
Visible
400 500 600 700 750 nm
© Sapling Learning

The EM spectrum ranges from gamma to radio waves. The visible part of the EM spectrum ranges from violet to red. The part of the spectrum before violet end is called Ultraviolet, and that after red end is called the Infrared.

Light that is void of color is called monochromatic or achromatic light. Its only attribute is its intensity. Intensity specifies the amount of light.

Since the intensity of monochromatic light is perceived to vary from black to gray shades and finally to white, the monochromatic intensity is called **Gray level.**

The range of measured values of monochromatic light from black to white is called the **Gray scale,** and the monochromatic images are called **gray scale images.**

Chromatic light spans the EM spectrum from 0.43µm (violet end) to 0.79µm (red end).

The quantities that describe the quality of a chromatic light source are

1. **Frequency**
2. **Radiance** – it is the amount of energy that flows from the light source and is measured in watts.
3. **Luminance** – it is the amount of energy that an observer perceives and is measured in lumens.
4. **Brightness** – it is similar to intensity in achromatic notion, and cannot be measured.

Sampling

It is the process of digitizing the spatial coordinate values. It may be viewed as partitioning the X-Y plane into a grid of M rows and N columns with the coordinates of the center of each cell in the grid being a pair from the Cartesian product Z^2. So f(x,y) is a digital image if x,y ϵ Z^2. Each cell is called a picture element or pixel.

Quantization

It is the process of digitizing the amplitude or intensity values.

Let f(s,t) represent a continuous image function of two continuous variables s and t. to convert it to a digital image, 1.**Sampling** – we sample the continuous image into a 2-D array f(x,y) containing M rows and N columns, where (x,y) are discrete coordinates taking up integer values x=0, 1, ..., M-1 and y=0,1,....., N-1. So f(0,1) indicates the second sample along the first row. Here, 0 and 1 are not the values of physical coordinates when the image was sampled.

2. **Quantization –** the values of the above samples that span a continuous range of intensity values, must be converted to discrete quantities. This is done by dividing the entire continuous intensity scale into L discrete intervals, ranging from black to white, where black is represented by a value 0 and white by L-1. Depending on the proximity of a sample to one of these L levels, the continuous intensity levels are quantized. In addition to the number of discrete levels used, the accuracy achieved in quantization is highly dependent on the noise content of the sampled signal.

Digital Image Fundamentals

1. **Dynamic Range** - It is the ratio of maximum measurable intensity to the minimum detectable intensity level in the imaging system. It establishes the lowest and highest intensity levels that an image can have.
    - The upper limit is determined by **Saturation.** It is the highest value beyond which all intensity levels are clipped. In an image, the entire saturated area has a high constant intensity level.
    - The lower limit is determined by **noise.** Especially in the darker regions of an image, the noise masks the lowest detectable true intensity level.
2. **Contrast** – It is the difference in intensity between the highest and lowest intensity levels in an image. For example, a low contrast image would have a dull washed out gray look.
3. **Spatial resolution** – it is a measure of the smallest detectable detail in an image. It is measured as line pairs per unit distance, or dots or pixels per unit distance (Dots per Inch or DPI). For example, newspapers are printed with a resolution of 75DPI, whereas textbooks are printed with a resolution of 2400 DPI.
4. **Intensity Resolution** – It is a measure of the smallest detectable change in intensity level. It refers to the number of bits used to quantize intensity. For example, an image whose intensity is quantized into 256 levels is said to have 8-bits of intensity resolution.

Digital Image Fundamentals

1. **Dynamic Range** - It is the ratio of maximum measurable intensity to the minimum detectable intensity level in the imaging system. It establishes the lowest and highest intensity levels that an image can have.
    - The upper limit is determined by **Saturation.** It is the highest value beyond which all intensity levels are clipped. In an image, the entire saturated area has a high constant intensity level.
    - The lower limit is determined by **noise.** Especially in the darker regions of an image, the noise masks the lowest detectable true intensity level.
2. **Contrast** – It is the difference in intensity between the highest and lowest intensity levels in an image. For example, a low contrast image would have a dull washed out gray look.
3. **Spatial resolution** – it is a measure of the smallest detectable detail in an image. It is measured as line pairs per unit distance, or dots or pixels per unit distance (Dots per Inch or DPI). For example, newspapers are printed with a resolution of 75DPI, whereas textbooks are printed with a resolution of 2400 DPI.
4. **Intensity Resolution** – It is a measure of the smallest detectable change in intensity level. It refers to the number of bits used to quantize intensity. For example, an image whose intensity is quantized into 256 levels is said to have 8-bits of intensity resolution.

3. **Spatial resolution** – it is a measure of the smallest detectable detail in an image. It is measured as line pairs per unit distance, or dots or pixels per unit distance (Dots per Inch or DPI). For example, newspapers are printed with a resolution of 75DPI, whereas textbooks are printed with a resolution of 2400 DPI.
4. **Intensity Resolution** – It is a measure of the smallest detectable change in intensity level. It refers to the number of bits used to quantize intensity. For example, an image whose intensity is quantized into 256 levels is said to have 8-bits of intensity resolution.

- $L = 2^{b}$

Terminology

- Bpp or bits per pixel denotes the number of bits per pixel.
- The number of different colors in an image is depends on the depth of color or bits per pixel.

Digital Image Size

- The size of a digital image is determines by its dimensions (M x N) multiplied by the number of bits b required to store the intensity levels ($L = 2^b$).

- image size = M x N x b (bits)

- Typical values of b are:
  - b = 1 black and white (binary) images.
  - b = 8 grayscale (256 gray levels), or indexed color images
  - b = 24 RGB color image.

GRAY SCALE

- The interval [L_{min} , L_{max}] is called the*gray scale.*

- *Common practice is to shift this interval numerically to the interval [0, L-1],*

- *where L = 0 is considered black and*

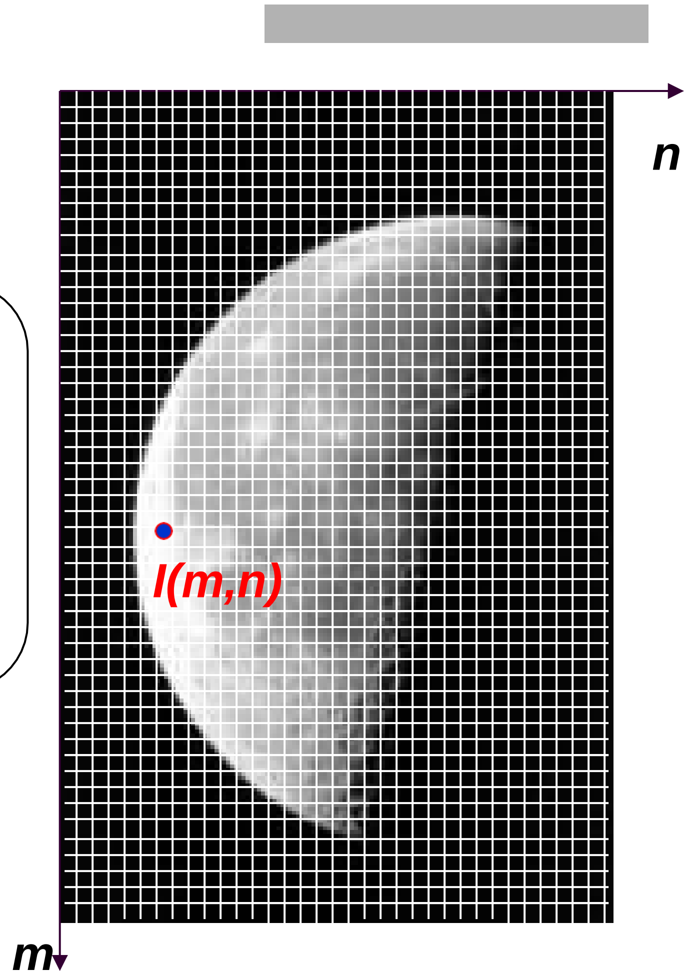
L = L-1 is considered white on the gray scale.

All intermediate values are shades of gray varying from black to white.

Digital Image Representation

- **A digital image can be represented as a two-dimensional matrix.**

$$I(mn) = \begin{pmatrix} i(1,1) & i(1,2) & \ldots\ldots. & i(1, n\text{-}1) \\ i(2,1) & i(2,2) & \ldots\ldots. & i(2, n\text{-}1) \\ . & & & . \\ . \; . & . \; . \; . \; . \; . \; . & & . \; . \; . \\ . & & & . \\ i(m\text{-}1,1) & i(m\text{-}1,2) & \ldots & i(m\text{-}1, n\text{-}1) \end{pmatrix}$$



- **Each element is called a *pixel* (picture element).**
- **A color (RGB) image is represented by a 3-dimensional matrix I(mxnx3)**

Digital Image Size

- The size of a digital image is determines by its dimensions (*M* x *N*) multiplied by the number of bits *b* required to store the intensity levels ($L = 2^b$).

- image size = *M* x *N* x *b* (bits)

- Typical values of b are:
  - b = 1 black and white (binary) images.
  - b = 8 grayscale (256 gray levels), or indexed color images
  - b = 24 RGB color image.

TABLE 2.1

Number of storage bits for various values of N and k.

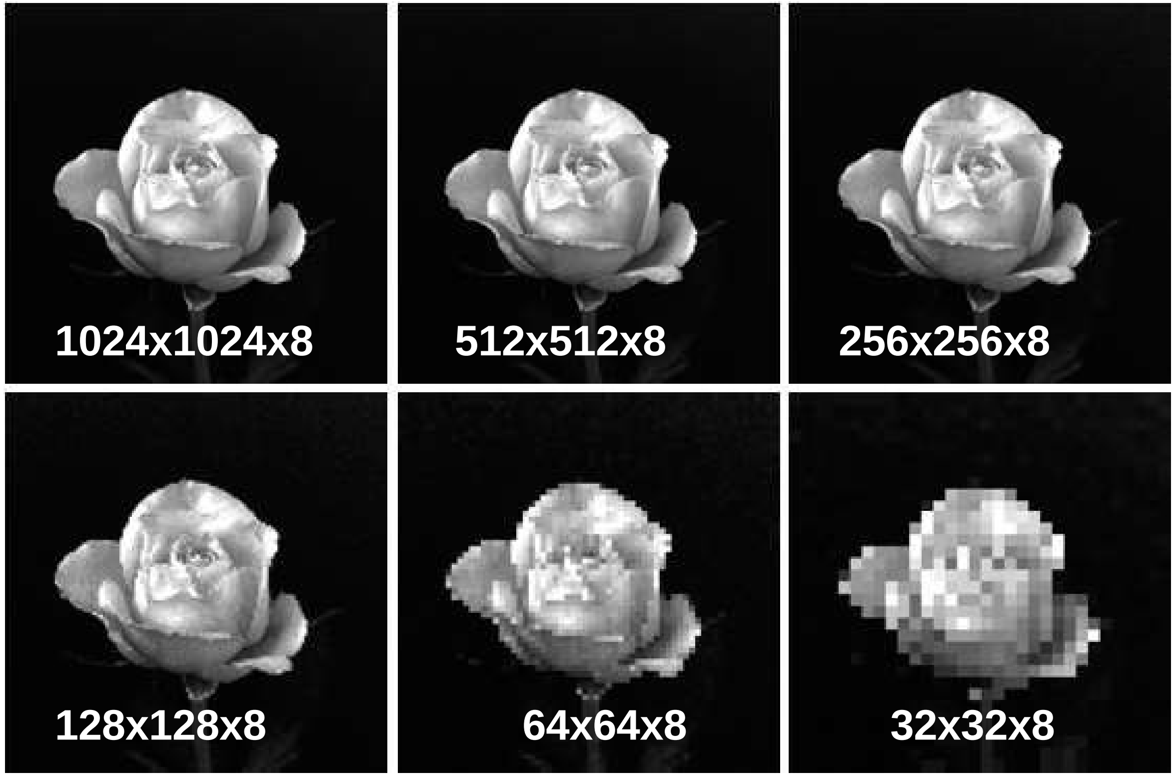
N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Digital Image size

- Digital image resolution is determined by the number of pixels (samples) in the image.

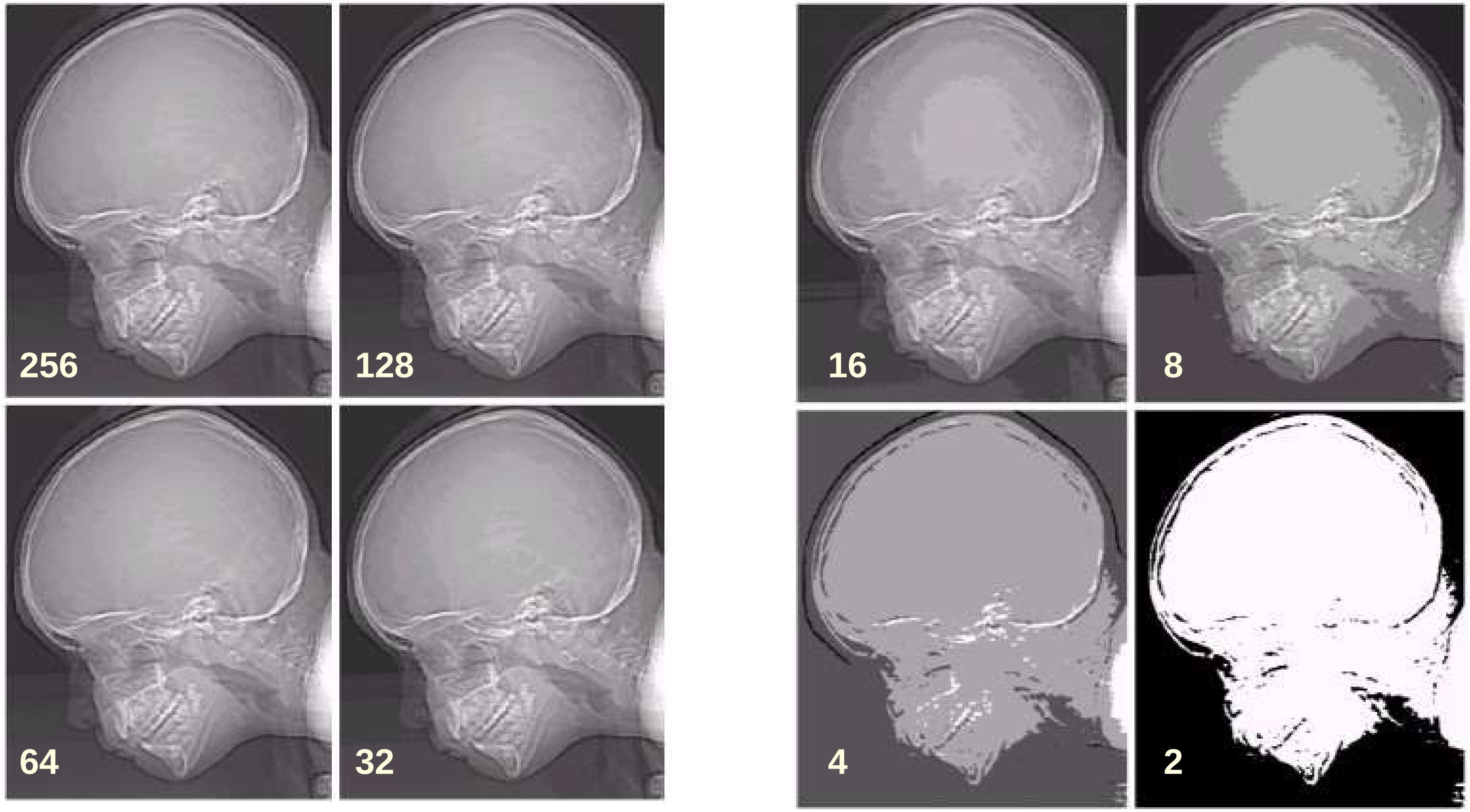
Digital Image Resolution

- Digital image with low resolution has low quality.



Digital Image Acquisition

- **Effect of Quantization**



Digital image Types

- RGB (Color) Images

Each pixel is a mixture of three values of Red, Geen, and Blue.

R,G,B = {0-255, 0-255, 0-255}

0 = Black

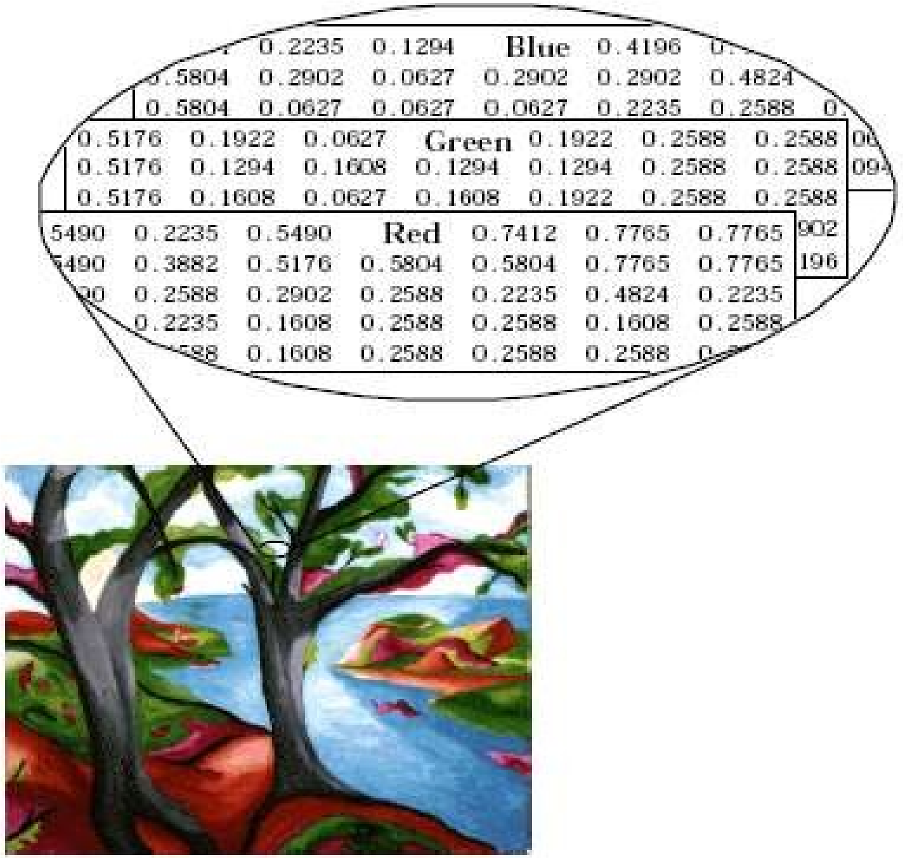
255 = White

In normalized values:

R,G,B = {0-1, 0-1, 0-1}

0 = Black

1 = White

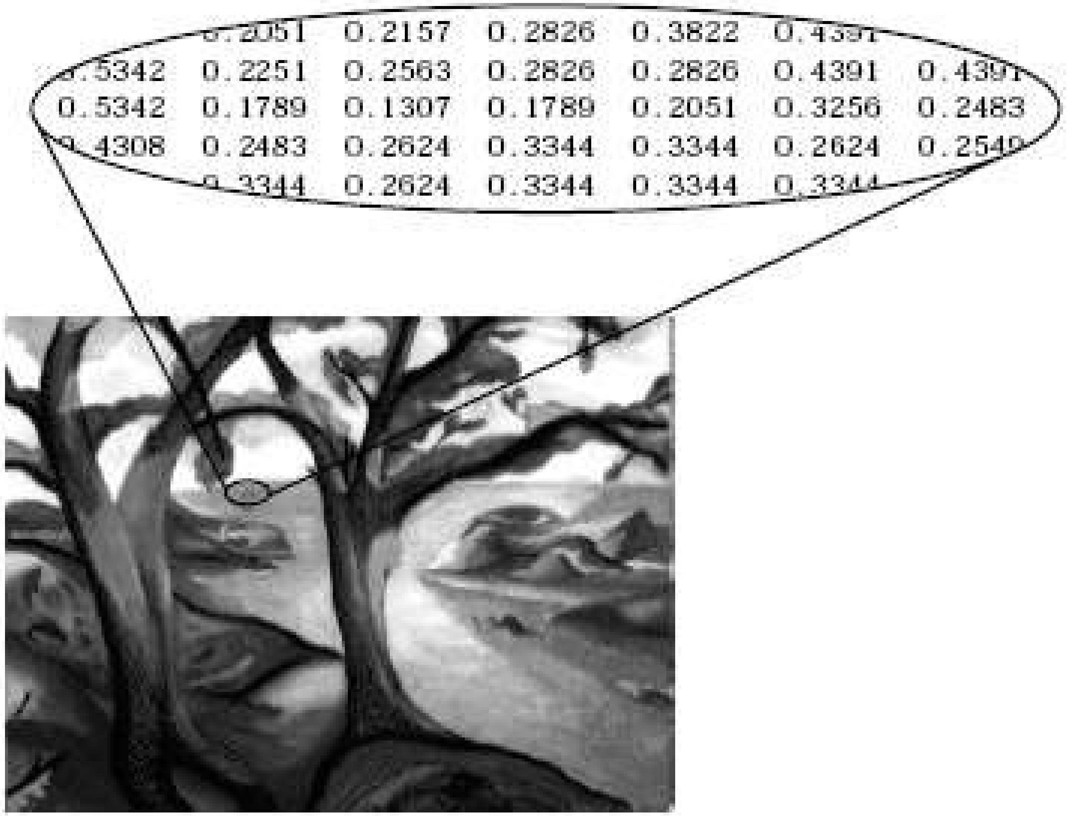


Digital image Types

- Grayscale (Intensity) Images

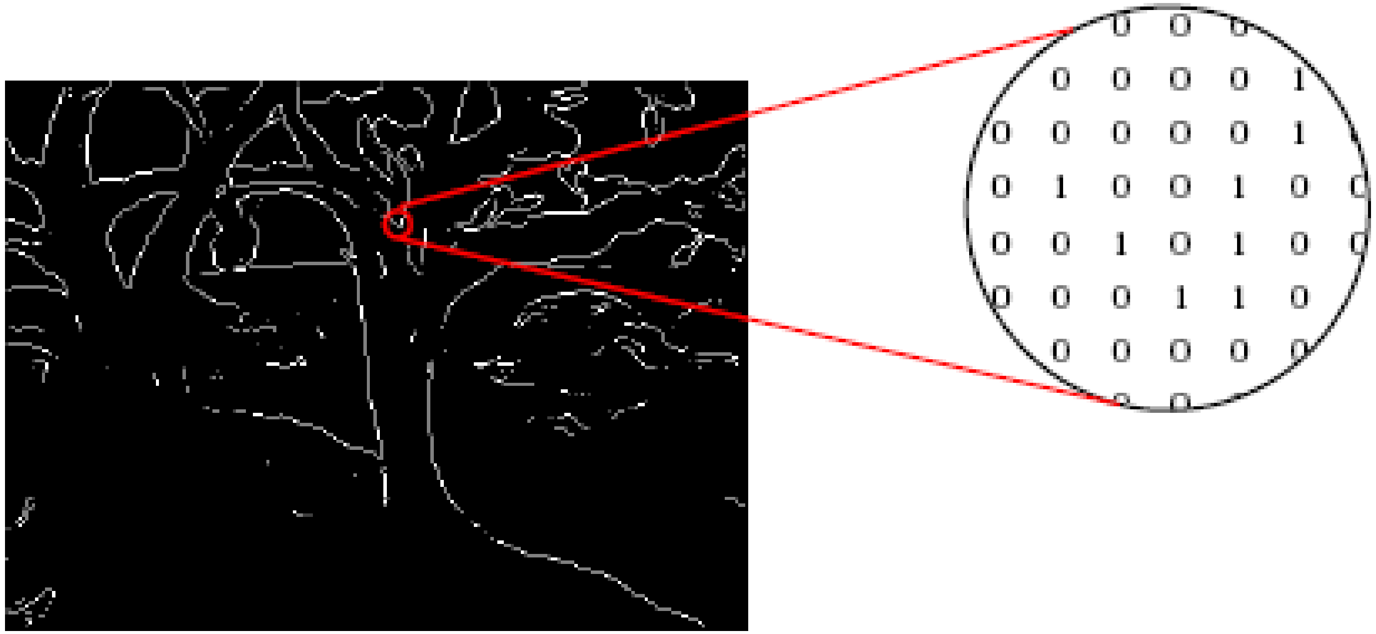
Each pixel is given a gray level value between 0 – 255 or between 0 – 1.

We need 8 bits to store a grayscale value.



Digital image Types

- Black and white (Binary) Images
  - Each pixel has one of two gray levels either black (0) or white (1).
  - We need 8 bits to store a grayscale value.



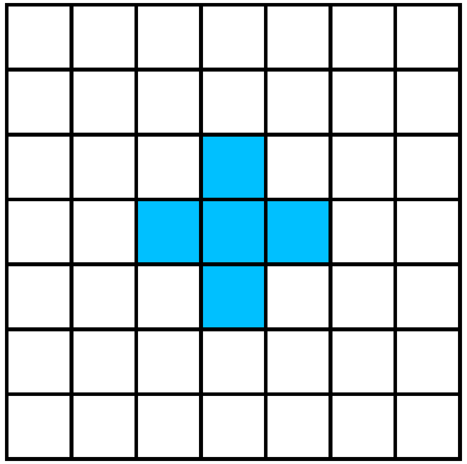
Types of Pixel Neighborhoods

- ***Image sampling*:**
  - *Rectangular sampling* - In most cases, images are sampled by laying a rectangular grid over an image.
  - *Hexagonal sampling* - An alternative sampling scheme is shown.

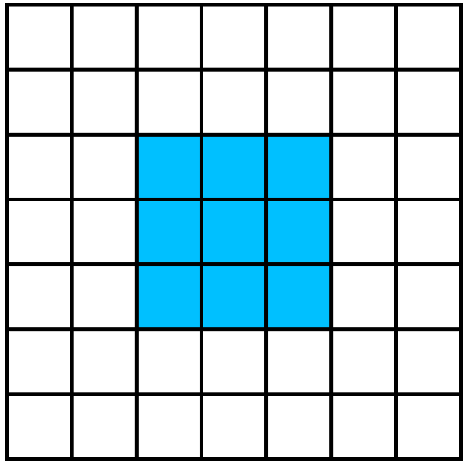
- ***Pixel Neighborhoods*:**
  - *4-connected* and *8-connected* neighborhood (Rectangular sampling)
  - *6-connected* neighborhood (Hexagonal sampling)

Types of Pixel Neighborhoods

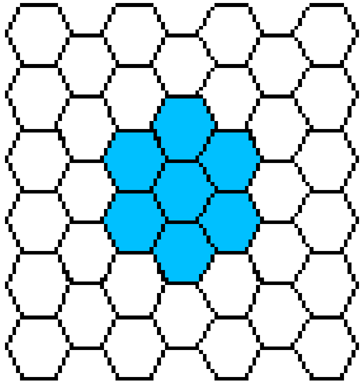
- **Basic Relationships Between Pixels**



4-connected



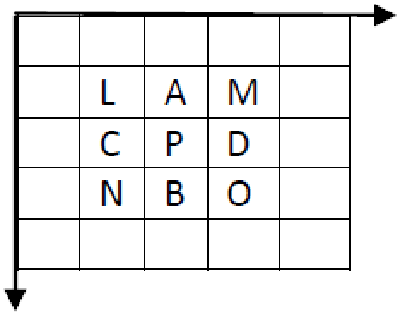
8-connected



6-connected

Basic Relationships between pixels

1. **Neighbor of a pixel** – we consider the following subset of pixels of an image



Let the position of pixel P be the coordinates (x,y). Then it has two horizontal neighbors and two vertical neighbors:

Horizontal neighbors : C : (x,y-1)

D : (x,y+1)

Vertical Neighbors : A : (x-1,y)

B : (x+1,y)

These horizontal and vertical neighbors are called the **4-nighbors** of P and the set is denoted by

$N_4(P) = \{A, B, C, D\} = \{(x-1,y), (x+1,y), (x,y-1), (x,y+1)\}$

If P is on the border of the image, some of the neighbors may not exist.

P also has four **diagonal neighbors:**

L : $(x-1,y-1)$

M : $(x-1,y+1)$

N : $(x+1,y-1)$

O : $(x+1,y+1)$

This set is denoted by $N_D(P) = \{L, M, N, O\}$.

All the above are together called the **8-neighbors** of P, and are denoted by $N_8(P)$. So

$$N_8(P) = N_4(P) \cup N_D(P)$$

Basic Relationships Between Pixels

- 1. **Neighbors of a Pixel :-**

A pixel p at coordinates (x, y) has four *horizontal and vertical neighbors whose coordinates are given by* (x+1, y), (x-1, y), (x, y+1), (x, y-1)

- This set of pixels, called the 4-*neighbors of p, is denoted by* N_4(p).

- Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

N_D(p) and N_8(p)

- The four *diagonal neighbors of p have coordinates*

  (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)

  and are denoted by N_D(p).

- These points, together with the 4-neighbors, arecalled the 8-*neighbors of p, denoted by* N_8(p).

- If some of the points in N_D(p) and N_8(p) fall outside the image if (x, y) is on the border of theimage.

Distance Measures

- If pixels p and q have coordinates (x, y) and (s, t), respectively.
  - The *Euclidean distance* between p and q
    - $D_e(p, q) = [(x - s)^2 + (y - t)^2]^{½}$
  - The D_4 (*City-block*) *distance* between p and q
    - $D_4(p, q) = |x - s| + |y - t|$
  - The D_8 (*Chessboard*) *distance* between p and q
    - $D_8(p, q) = \max(|x - s| , |y - t|)$

Distance Measures

- Results of D_4 and D_8 distances

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

- The pixels with $D_4=1$ *are the 4-neighbors*
- The pixels with $D_8=1$ *are the 8-neighbors*

Adjacency

- To establish whether two pixels are connected, it must be determined if they are neighbors and
- if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal).

- For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors,

- but they are said to be connected only if they have the same value

- Let *V be the set of gray-level values used todefine connectivity. In a binary image, V={1} for the connectivity of pixels with value 1.*

- *In a grayscale image, for connectivity of pixels with a range of intensity values of say 32, 64 V typically contains more elements.*

- *For example,*

- *In the adjacency of pixels with a range of possible gray-level values 0 to 255,*
- *set V could be any subset of these 256 values.We consider three types of adjacency:*

- **We consider three types of adjacency:**

(a) 4-*adjacency.*

Two pixels p and q with values from V are 4-adjacent if q is in the set N_4(p).

(b) 8-*adjacency.*

Two pixels p and q with values from V are 8-adjacent if q is **in the set N_8(p).**

(c) m-adjacency (mixed adjacency).

$N_4(p) \cap N_4(q)$

(d) Two pixels p and q with values from V are m-adjacent if

- **(i) q is in N_4**(p), *or*

- ***(ii) q is in N_D***(p) *and the set whose values are from V.*

- A *path from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates*

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

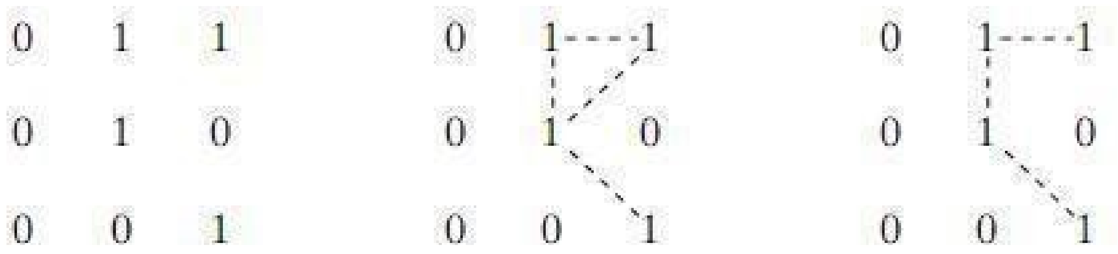
- where $(x_{б'}\ y_0) = (x, y)$ and $(x_{Ŷ'}\ y_n) = (s, t)$,

$(x_{i'}\ y_i)$ and $(x_{i-б'}\ y_{i-1})$ pixels and are adjacent for б ч i ч Ŷ. IŶ this Đase, Ŷ is the *length of the path.*

- *If* $(x_{б'}\ y_0) = (x_{Ŷ'}\ y_n)$ the path is a *closed path*.

- Two pixels p and q are said to be *connected in S if there exists a path between them consisting entirely of pixels in S.*

- *For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S.*



a b c

FIGURE (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency.

Path - a path from pixel p with coordinates (x,y) to pixel q with coordinates (s,t) is a sequence of distinct pixels with coordinates (x_0, y_0), (x_1, y_1), ... (x_n, y_n) where pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are

adjacent for $1 \leq i \leq n$, and where $(x_0, y_0) = (x,y)$ and $(x_n, y_n) = (s,t)$.

- Here, n is the **length of the path.**
- If (x_0, y_0) and (x_n, y_n) are the same, then the path is a **closed path.**
- Depending on the type of adjacency, we can have **4-path, 8-path** or **m-path.**

0	1(A)	1(B)
0	1(C)	0
0	0	1(D)

For example, in the above arrangement of pixels, the paths are as follows:

4-path : B A C

8-paths: (i) B A C, (ii) B A C D, (iii)B C D

m-path: B A C D

Connectivity – let S represent a subset of pixels in an image. Then, two pixels p and q are said to be **connected** in S, if there exists a path between them consisting entirely of pixels in S. For example, in the previous figure, if all the 9 pixels form the subset S, then B and C (or B and D in case of 8 and m-paths) are said to be connected to each other. On the other hand, if S consists of pixels of rightmost column, then B and D are not connected to each other.

For any pixel p in S, the set of pixels that are connected to it in S is called a **connected component** of S. for example, if all the 9 pixels in the above figure form S, the pixels A, B, C and D form a connected component.

Region – let R be a subset of pixels of an image. R is called a region of the image, if R is a connected set.

Mathematical Tools used in Digital Image Processing

1. **Array versus Matrix operations** – Array operations in images are carried out on a pixel – by – pixel basis. For example, raising an image to a power means that each individual pixel is raised to that power. There are also situations in which operations between images are carried out using matrix theory.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

The *array product* of these two images is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

On the other hand, the *matrix product* is given by

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Arithmetic operations – these operations are array operations and are denoted by

$$S(x,y) = f(x,y) + g(x,y)$$

$$D(x,y) = f(x,y) - g(x,y)$$

$$P(x,y) = f(x,y) * g(x,y)$$

$$V(x,y) = f(x,y) \div g(x,y)$$

These operations are performed between corresponding pixel pairs in f and g for x = 0, 1, 2, … M-1 and y = 0, 1, 2, …, N-1, where all the images are of size M X N.

For example,

<u>Set Operations</u> – for gray scale images, set operations are array operations. The **union** and **intersection** operations between two images are defined as the maximum and minimum of corresponding pixel pairs respectively. The **complement** operation on an image is defined as the pairwise differences between a constant and the intensity of every pixel in the image.

Let set A represent a gray scale image whose elements are the triplets (x, y, z) where (x,y) is the location of the pixel and z is its intensity. Then,

Union :

$$A \cup B = \left\{ \max_{z}(a,b) | a \in A, b \in B \right\}$$

Intersection:

$$A \cap B = \left\{ \min_{z}(a,b) | a \in A, b \in B \right\}$$

Complement:

$$A^C = \{(x,y,K-z) | (x,y,z) \in A\}$$

Logical Operations – when dealing with binary images, the 1 – valued pixels can be thought of as foreground and 0 – valued pixels as background. Now, considering two regions A and B composed of foreground pixels,

The **OR** operation of these two sets is the set of coordinates belonging either to A or to B or to both.

The **AND** operation is the set of elements that are common to both A and B.

The **NOT** operation on set A is the set of elements not in A.

The logical operations are performed on two regions of same image, which can be irregular and of different sizes.

The set operations involve complete images.

if we consider a set of K noisy images of a particular scene $\{g_i(x,y)\}$, then, in order to obtain an image with less noise, averaging operation can be done as follows:

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

This averaging operation can be used in the field of astronomy where imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis.

Single-Pixel Operations

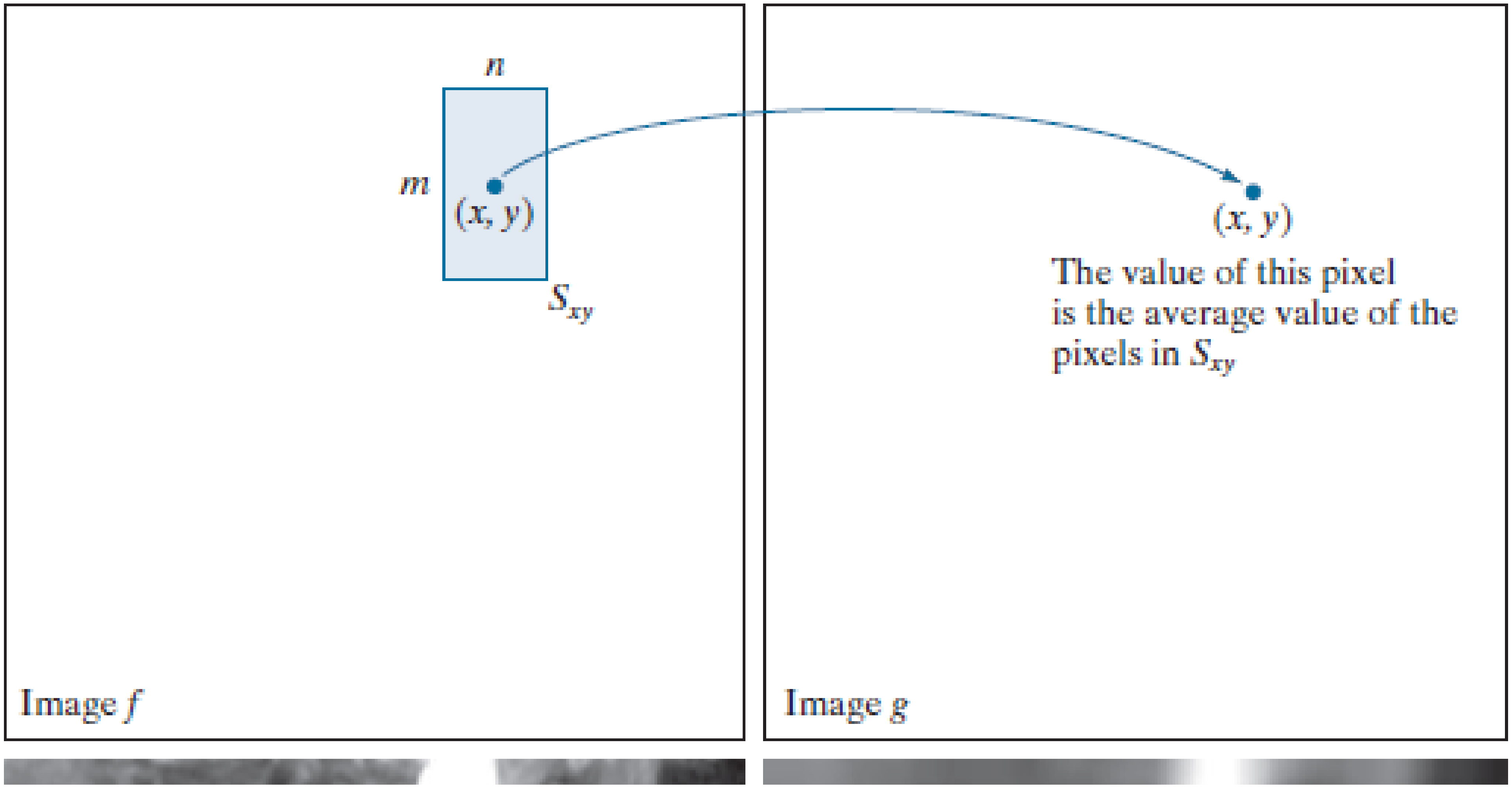
The simplest operation we perform on a digital image is to alter the intensity of its pixels individually using a transformation function, T, of the form:

$$s = T(z) \tag{2-42}$$

Neighborhood Operations

Let S_{xy} denote the set of coordinates of a neighborhood (see Section 2.5 regarding neighborhoods) centered on an arbitrary point (x, y) in an image, f. Neighborhood processing generates a corresponding pixel at the same coordinates in an output (processed) image, g, such that the value of that pixel is determined by a specified operation on the neighborhood of pixels in the input image with coordinates in the set S_{xy}. For example, suppose that the specified operation is to compute the average value of the pixels in a rectangular neighborhood of size $m \times n$ centered on (x, y). The coordinates of pixels in this region are the elements of set S_{xy}. Figures 2.39(a) and (b) illustrate the process. We can express this averaging operation as

$$g(x, y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} f(r, c) \tag{2-43}$$



n
m
(x, y)
S_{xy}
(x, y)
The value of this pixel is the average value of the pixels in S_{xy}
Image f
Image g

Neighborhood based arithmetic/Logic :

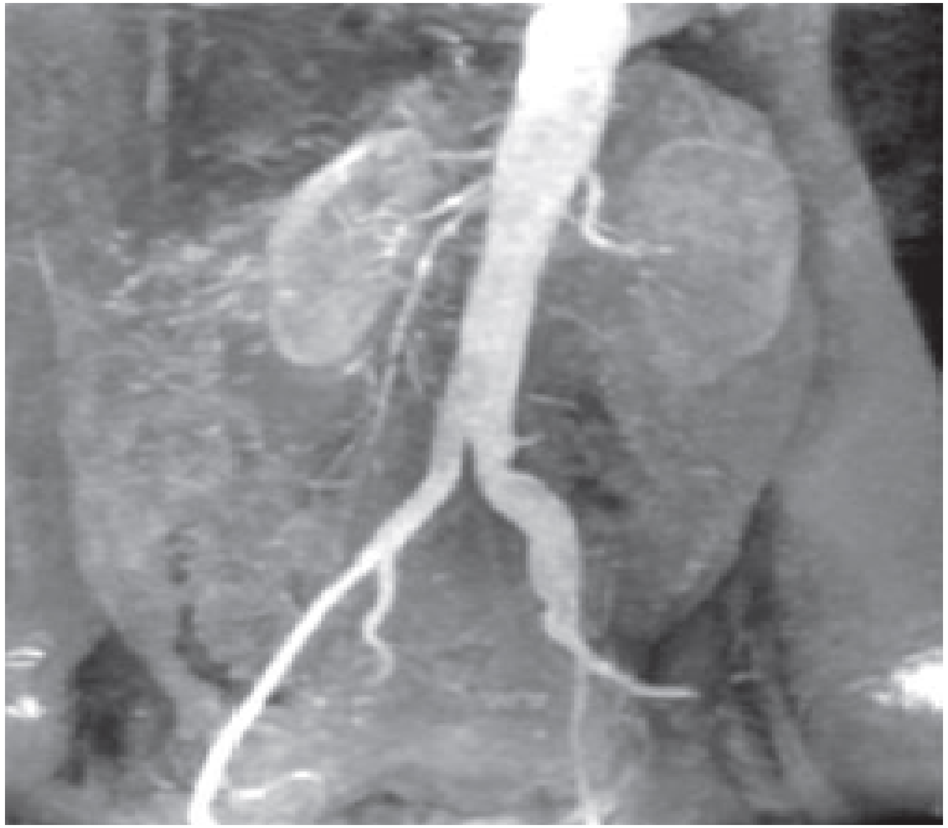
Value assigned to a pixel at position 'e' is a function of its neighbors and a set of window functions.

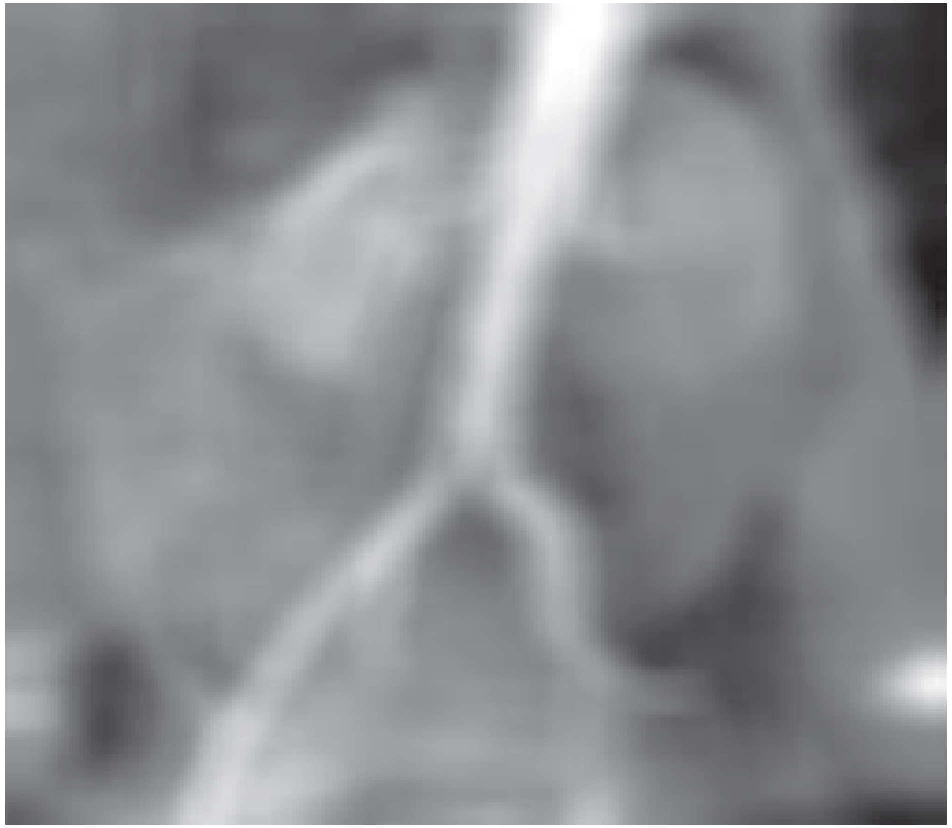
		:		
	a	b	c	
...	d	e	f	..
	g	h	i	
		:		

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$p = (w_1 a + w_2 b + w_3 c + w_4 d + w_5 e + w_6 f + w_7 g + w_8 h + w_9 i)$$

$$= \sum w_i f_i$$





Geometric Transformations

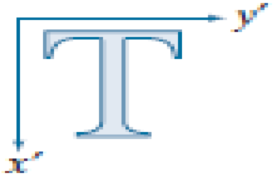
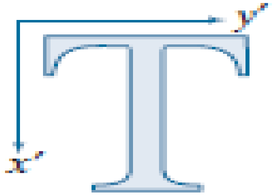
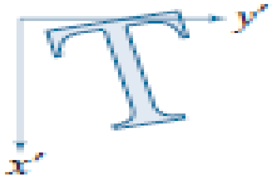
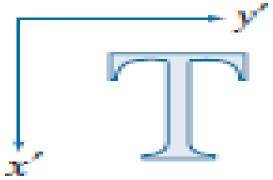
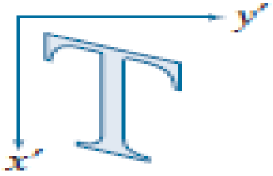
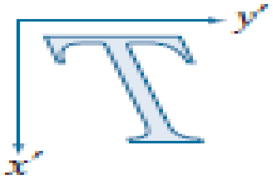
We use geometric transformations modify the spatial arrangement of pixels in an image. These transformations are called *rubber-sheet transformations* because they may be viewed as analogous to "printing" an image on a rubber sheet, then stretching or shrinking the sheet according to a predefined set of rules. Geometric transformations of digital images consist of two basic operations:

1. Spatial transformation of coordinates.
2. Intensity interpolation that assigns intensity values to the spatially transformed pixels.

The transformation of coordinates may be expressed as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathrm{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2-44}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2-45}$$

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to −1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

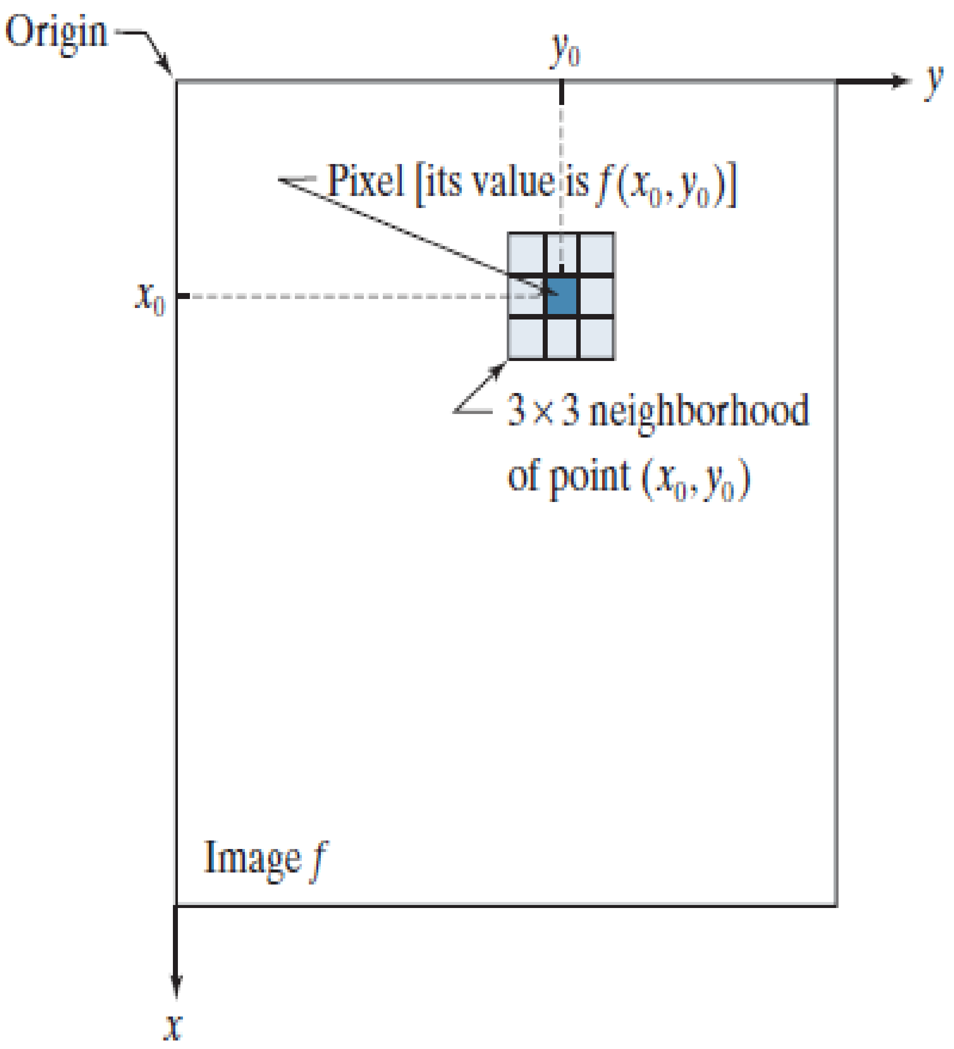
THE BASICS OF INTENSITY TRANSFORMATIONS AND SPATIAL FILTERING

The spatial domain processes we discuss in this chapter are based on the expression

$$g(x,y) = T\left[f(x,y)\right] \tag{3-1}$$

where $f(x,y)$ is an input image, $g(x,y)$ is the output image, and T is an operator on f defined over a neighborhood of point (x,y). The operator can be applied to the pix-

FIGURE 3.1 A 3×3 neighborhood about a point (x_0, y_0) in an image. The neighborhood is moved from pixel to pixel in the image to generate an output image. Recall from Chapter 2 that the value of a pixel at location (x_0, y_0) is $f(x_0, y_0)$, the value of the image at that location.



The smallest possible neighborhood is of size 1×1. In this case, g depends only on the value of f at a single point (x, y) and T in Eq. (3-1) becomes an *intensity* (also called a *gray-level*, or *mapping*) *transformation function* of the form

$$s = T(r) \tag{3-2}$$

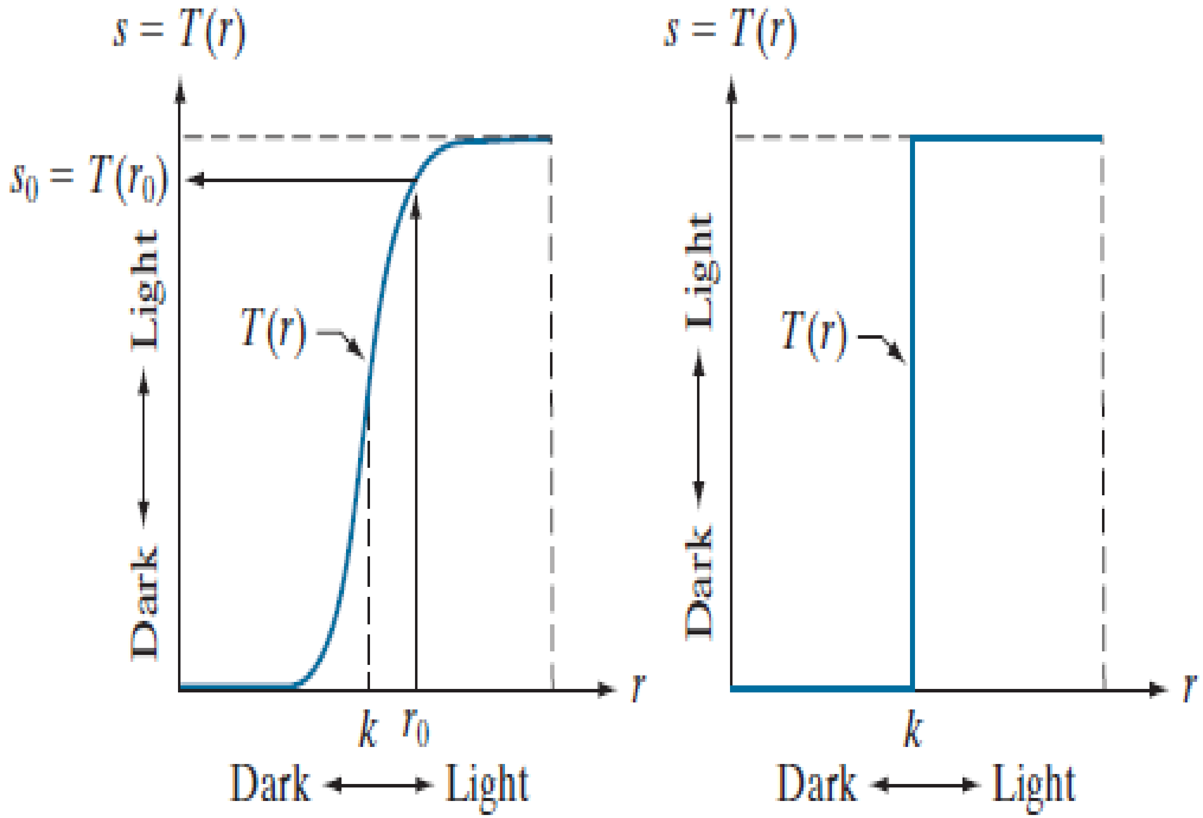


FIGURE 3.2 Intensity transformation functions. (a) Contrast stretching function. (b) Thresholding function.

3.2 SOME BASIC INTENSITY TRANSFORMATION FUNCTIONS

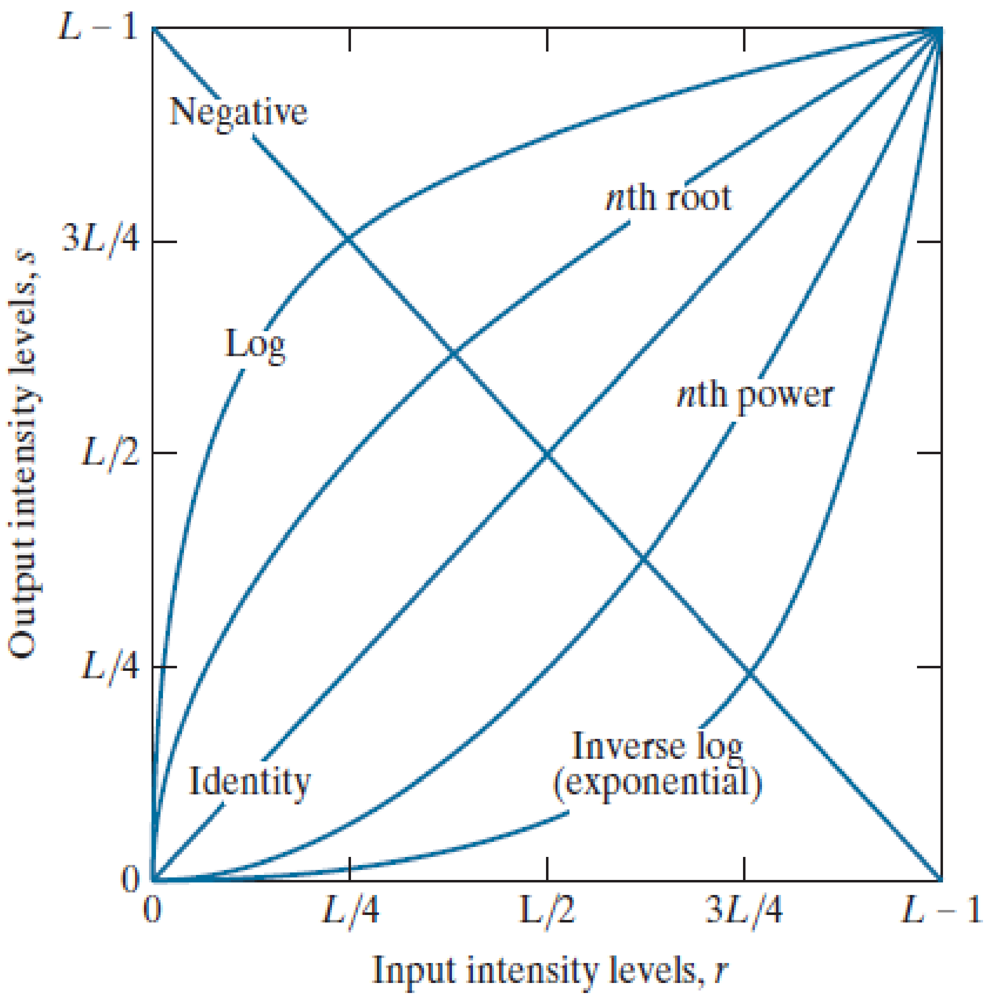
Intensity transformations are among the simplest of all image processing techniques. As noted in the previous section, we denote the values of pixels, before and after processing, by r and s, respectively. These values are related by a transformation T, as given in Eq. (3-2), that maps a pixel value r into a pixel value s. Because we deal with digital quantities, values of an intensity transformation function typically are stored in a table, and the mappings from r to s are implemented via table lookups. For an 8-bit image, a lookup table containing the values of T will have 256 entries.

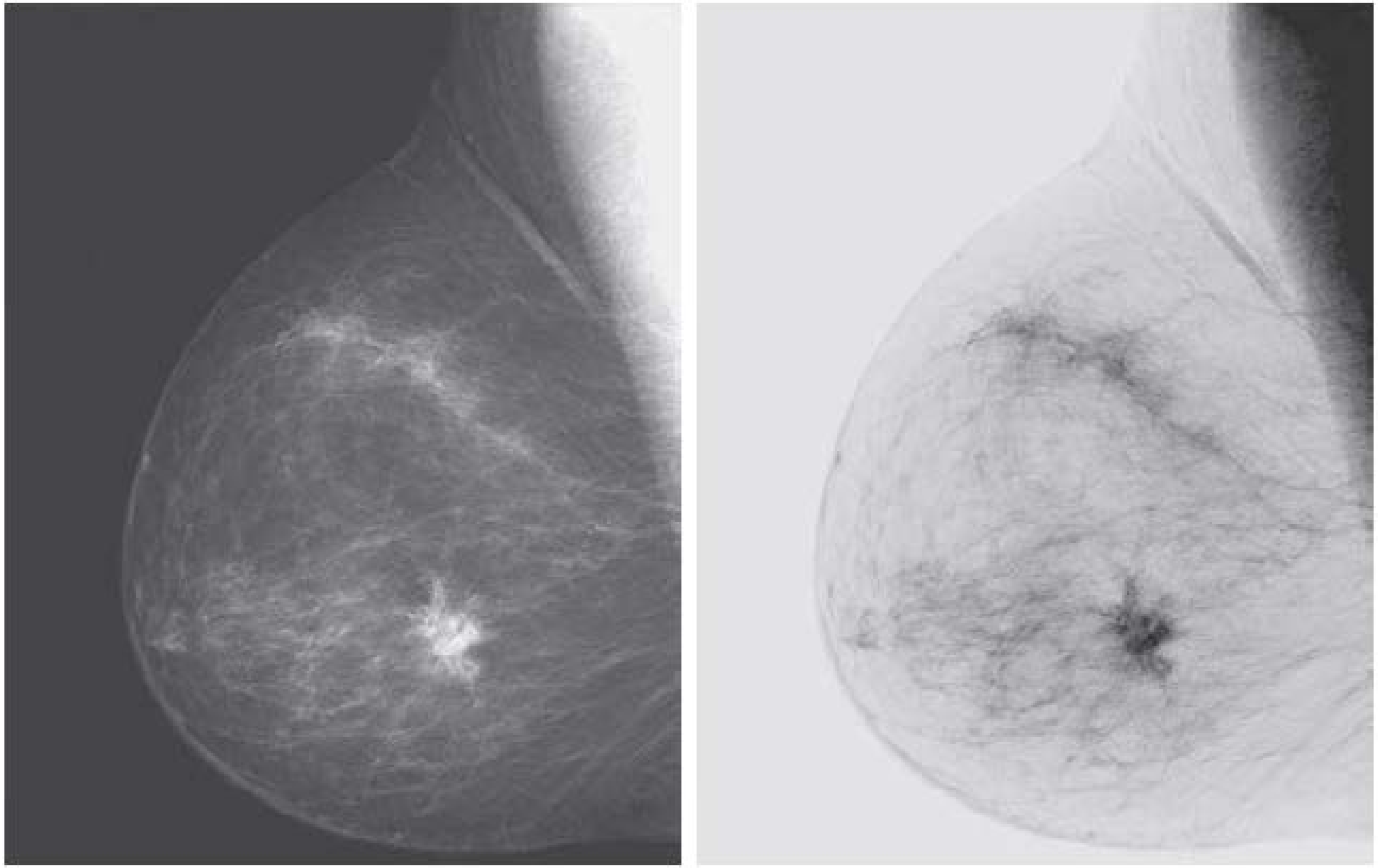
IMAGE NEGATIVES

The negative of an image with intensity levels in the range $[0, L-1]$ is obtained by using the negative transformation function shown in Fig. 3.3, which has the form:

$$s = L - 1 - r \tag{3-3}$$

FIGURE 3.3 Some basic intensity transformation functions. Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.





a b

FIGURE 3.4 (a) A digital mammogram. (b) Negative image obtained using Eq. (3-3). (Image (a) Courtesy of General Electric Medical Systems.)

LOG TRANSFORMATIONS

The general form of the log transformation in Fig. 3.3 is

$$s = c \log(1 + r) \tag{3-4}$$

where c is a constant and it is assumed that $r \geq 0$. The shape of the log curve in Fig. 3.3 shows that this transformation maps a narrow range of low intensity values in the input into a wider range of output levels. For example, note how input levels in the

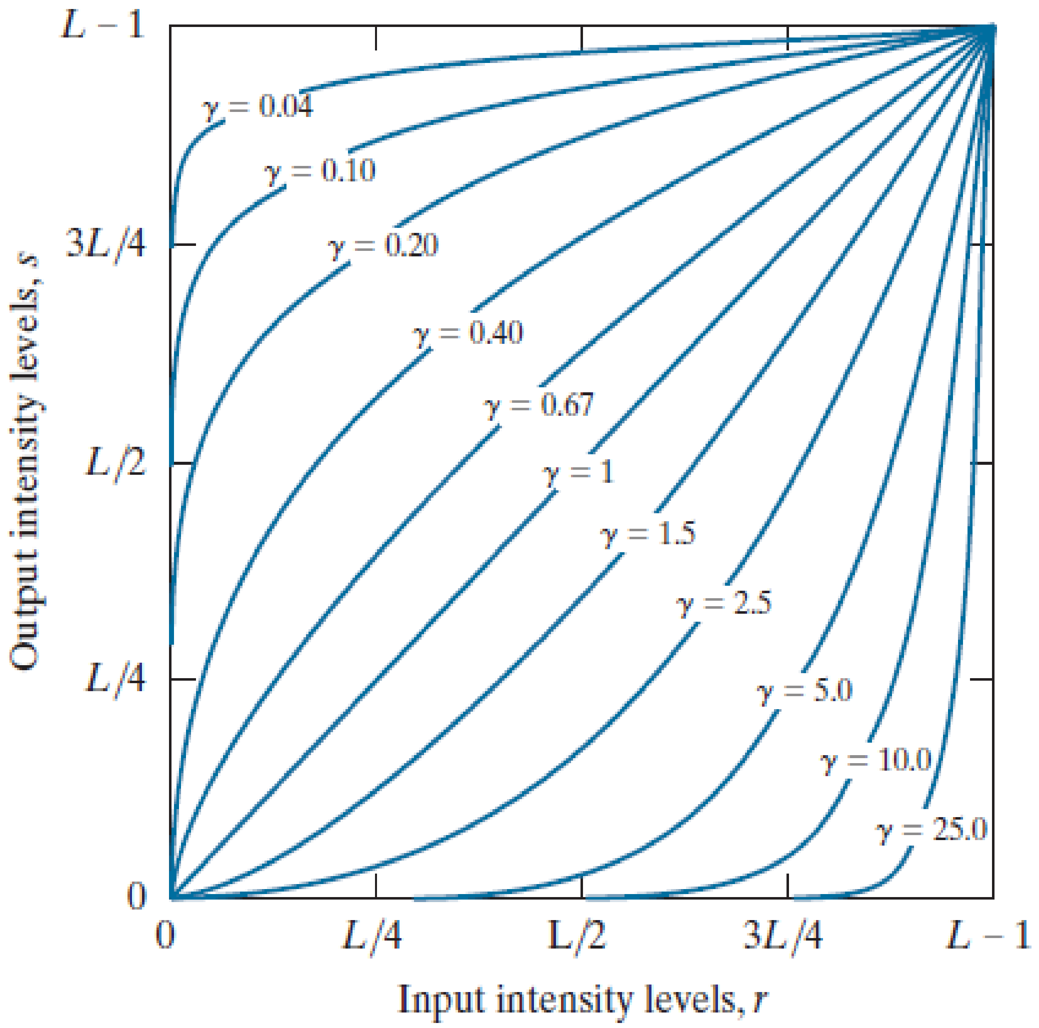
POWER-LAW (GAMMA) TRANSFORMATIONS

Power-law transformations have the form

$$s = cr^{\gamma} \tag{3-5}$$

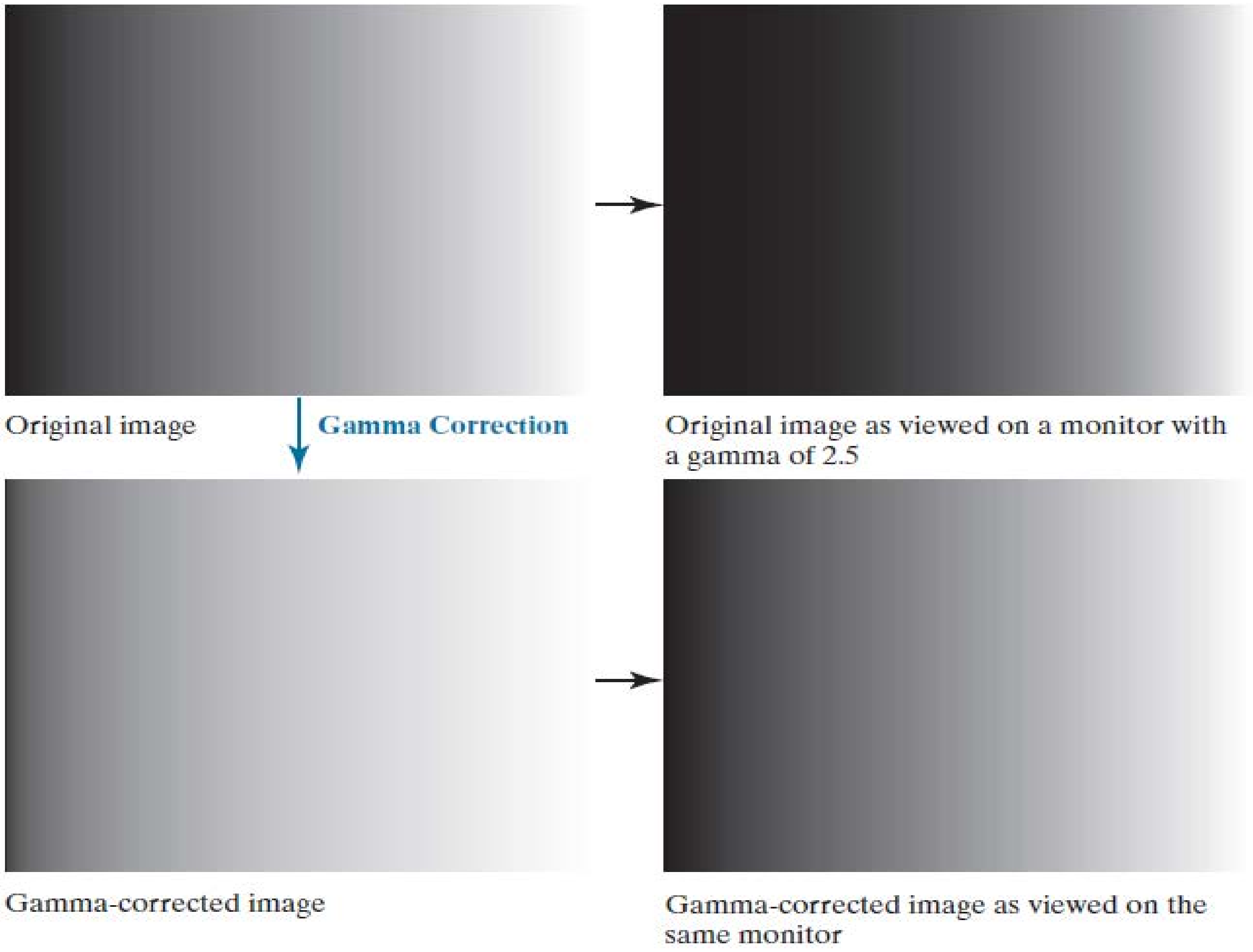
where c and γ are positive constants. Sometimes Eq. (3-5) is written as $s = c(r + \varepsilon)^{\gamma}$ to account for offsets (that is, a measurable output when the input is zero). However,

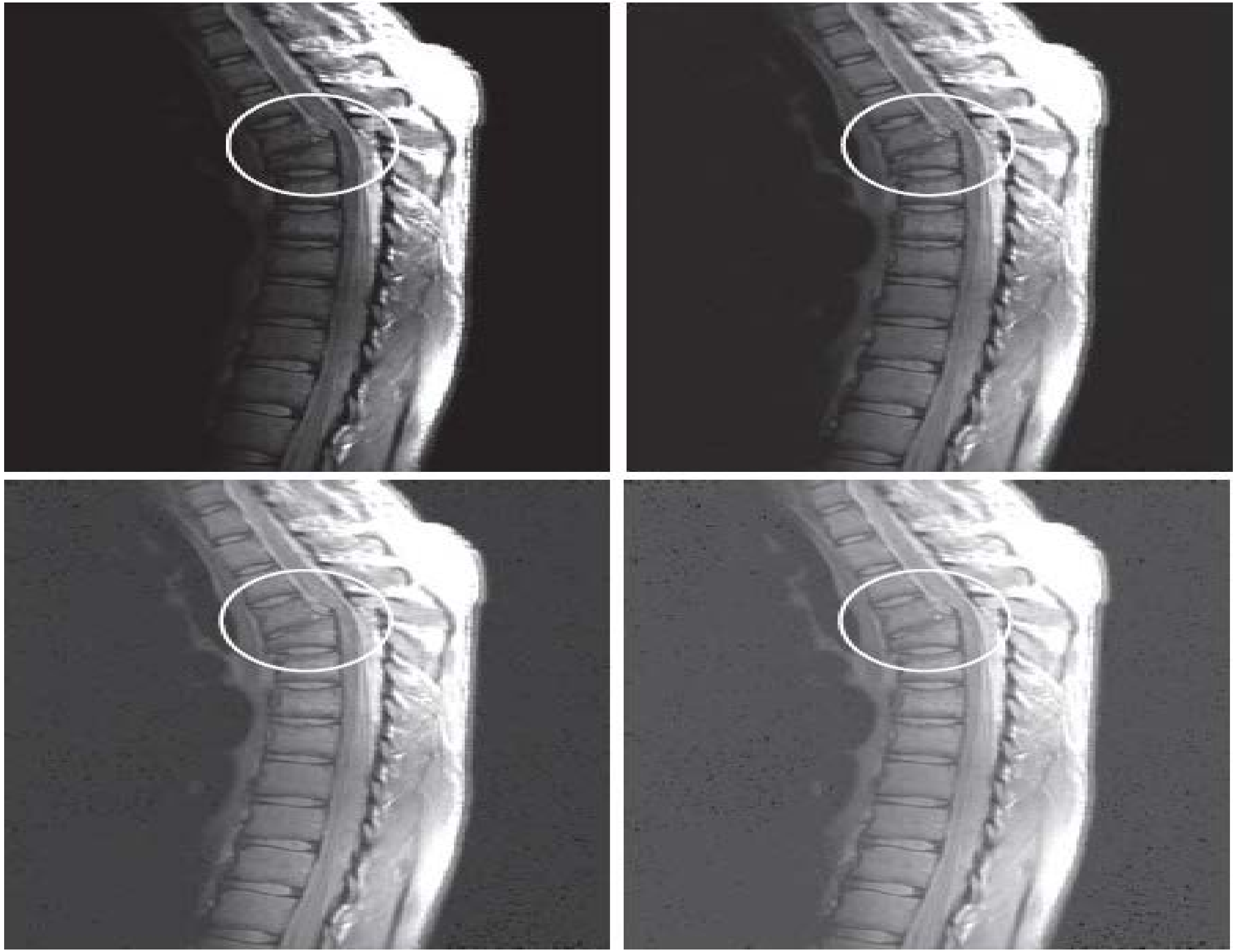
FIGURE 3.6 Plots of the gamma equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.



a b
c d

FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated 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 (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the transformation in Eq. (3-5) 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. (3-5) with $\gamma = 3.0, 4.0,$ and 5.0, respectively. ($c = 1$ in all cases.) (Original image courtesy of NASA.)

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

An approach complementary to the methods discussed in the previous three sections is to use piecewise linear functions. The advantage of these functions over those discussed thus far is that the form of piecewise functions can be arbitrarily complex. In fact, as you will see shortly, a practical implementation of some important transformations can be formulated only as piecewise linear functions. The main disadvantage of these functions is that their specification requires considerable user input.

Contrast Stretching

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* expands the range of intensity levels in an image so that it spans the ideal full intensity range of the recording medium or display device.

a b
c d

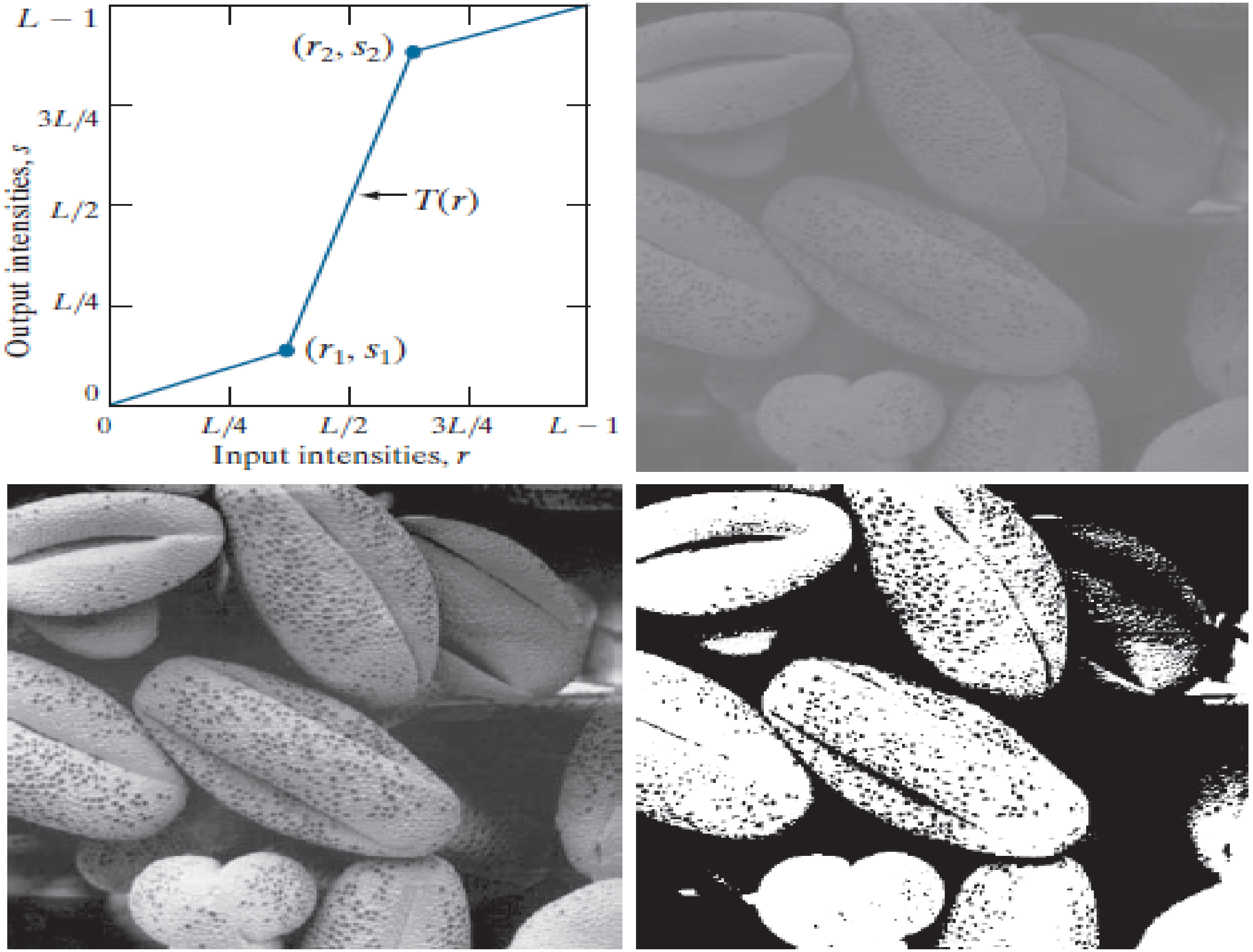


FIGURE 3.10 Contrast stretching. (a) Piecewise linear transformation function. (b) A low-contrast electron microscope image of pollen, magnified 700 times. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

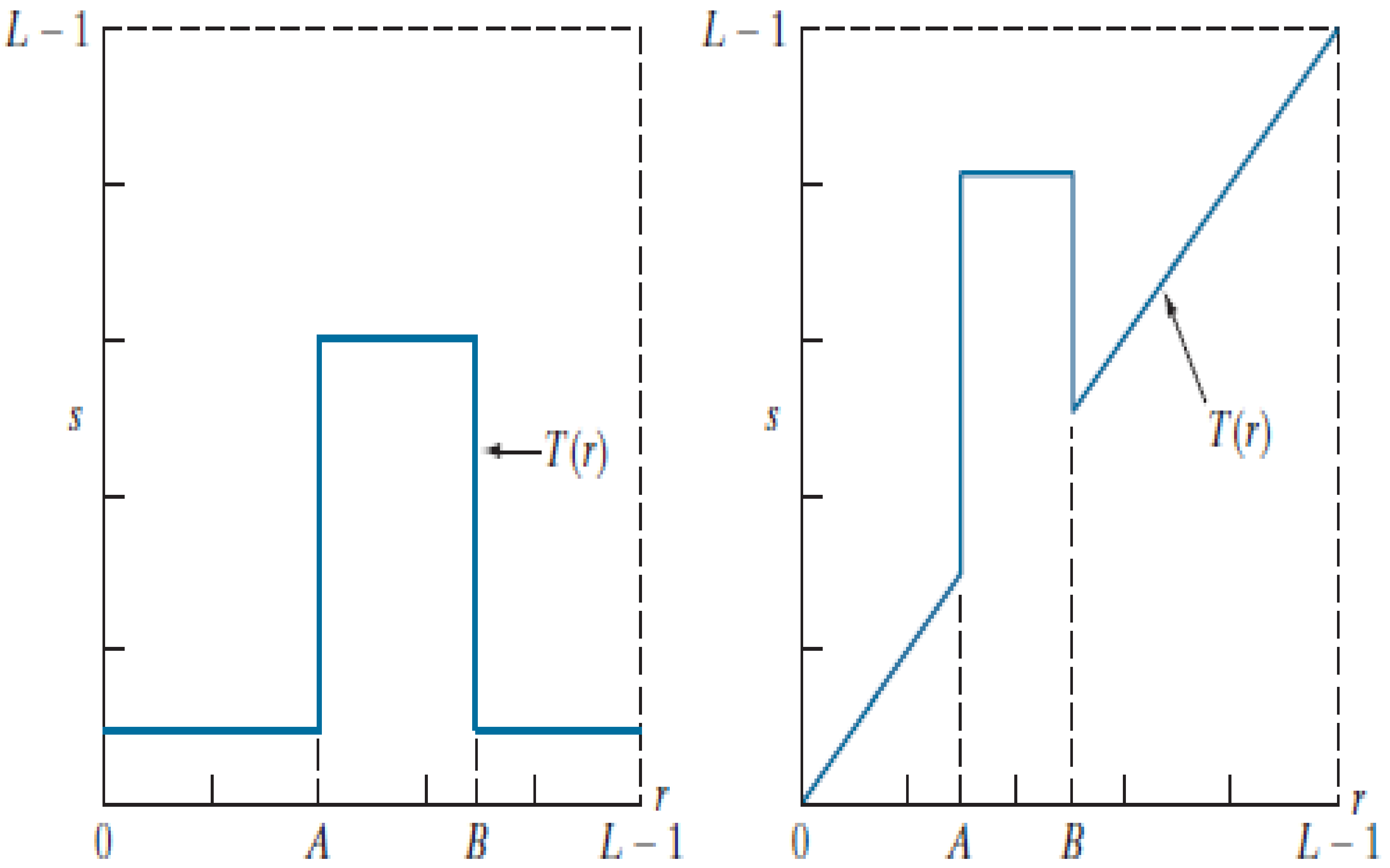


FIGURE 3.11 (a) This transformation function highlights range $[A, B]$ and reduces all other intensities to a lower level. (b) This function highlights range $[A, B]$ and leaves other intensities unchanged.