

MT390 (DIP): Tutorial 1

Introduction and Chapter 2: Digital Image Fundamentals

Course syllables:

Course code: MT390

Course name: Fundamentals of digital image processing

No. of credits: 3 Hours (Bi-weekly)

Course Textbook:

"Digital Image Processing" Global 4th Edition, by Rafael C. González, Richard E. Woods, 2018.

Online Course Support:

Individual: Ebook code/access.

General: <http://www.imageprocessingplace.com/>

Course assessments

Component	weight
1 TMA	20 %
Attendance/participation	5 %
In class quizzes	25 %
Final Exam	50 %
Total	100 %

Faculty of Computer Studies: Information Technology and Computing Programme**Course Code: MT390****Semester spring 2021-22****Course Title: Image Processing****Credit Hours: 3****"Digital Image Processing" Global 4th Edition, by Rafael C. González, Richard E. Woods, 2018**

Tut #	Course Chapter	Sections Covered	Remarks
1	Chapter 1: Introduction Chapter 2: Digital image Fundamentals	Introduction; Sec. 2.4—2.5	
2	Chapter 3: Intensity transformations and spatial filtering –Part I	Sec. 3.1—3.3	
3	Chapter 3: Intensity transformations and spatial filtering –Part II	Sec. 3.4 – 3.6, 3.8	
4	Chapter 4: Filtering in the frequency domain (Part I)	Sec. 4.4– 4.6	
5	Chapter 4: Filtering in the frequency domain (Part II)	Sec. 4.7– 4.9	
6	Chapter 8: Image compression Application: Image Classification End of material	Sec. 8.1, 8.2, 8.6, 8.9—8.11	TMA week: Submission dates to be specified by deanship
Final exam period			

**Note: Both Python (CV2, PyWavelet) or Matlab based solutions are acceptable for TMA.
Python and Matlab will be used for TMA assessments implementation only.**

Background

- DIP requires extensive **experimental** work.
- Main **objective** of this course is to bridge the gap between **Theory** and **Application**.
- **Theory:** The various mathematical concepts of DIP.
- **Application:** Implement DIP concepts using state-of-the-art software tools (MATLAB/Python)

Introduction:

What is an Image:

- An image may be defined as a **2-D function**, $f(x,y)$, where x and y are **spatial coordinates** and the amplitude of f is called the **intensity or gray level**.
- When x , y , and the amplitude values of f are all finite, discrete quantities, we call the image a ***digital image***



Figure 1.1

A digital picture produced in 1921 from a coded tape by a telegraph printer with special typefaces.



Figure 1.4

The first picture of the moon by a U.S. spacecraft. Ranger 7 took this image on July 31, 1964 at 9:09 A.M. EDT, about 17 minutes before impacting the lunar surface.



What is DIP (informal):

The field of Digital Image Processing (DIP) refers to **processing digital images by means of a computer.**



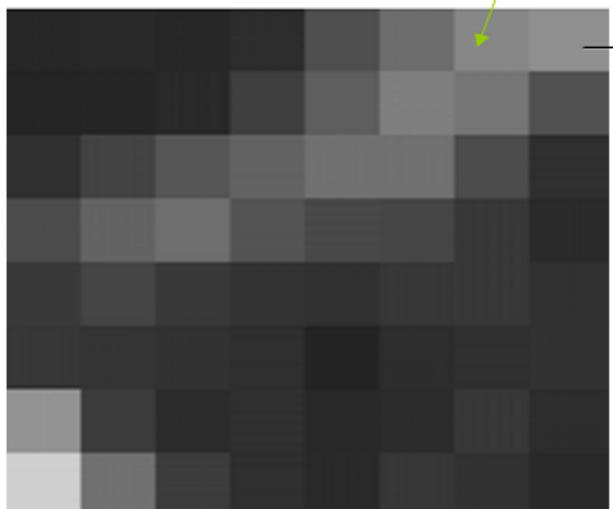
Examples:

What is digital image?



The image consists of finite number of pixels ($f(x,y)$)

Every pixel Is an intersection between a row and a column.



pixel

every pixel has intensity



Ex:

$$f(4,3)=123$$

Refers to a pixel existing on the intersection between row 4 with column 3, and its intensity is 123.

Sources/Types of Images:

- Visual Band (EM Spectrum)
- Gamma-Ray Imaging
- X-Ray Imaging
- Ultraviolet Band Imaging
- Ultrasound
- MRI
- Electron Microscopy
- Computer-generated images

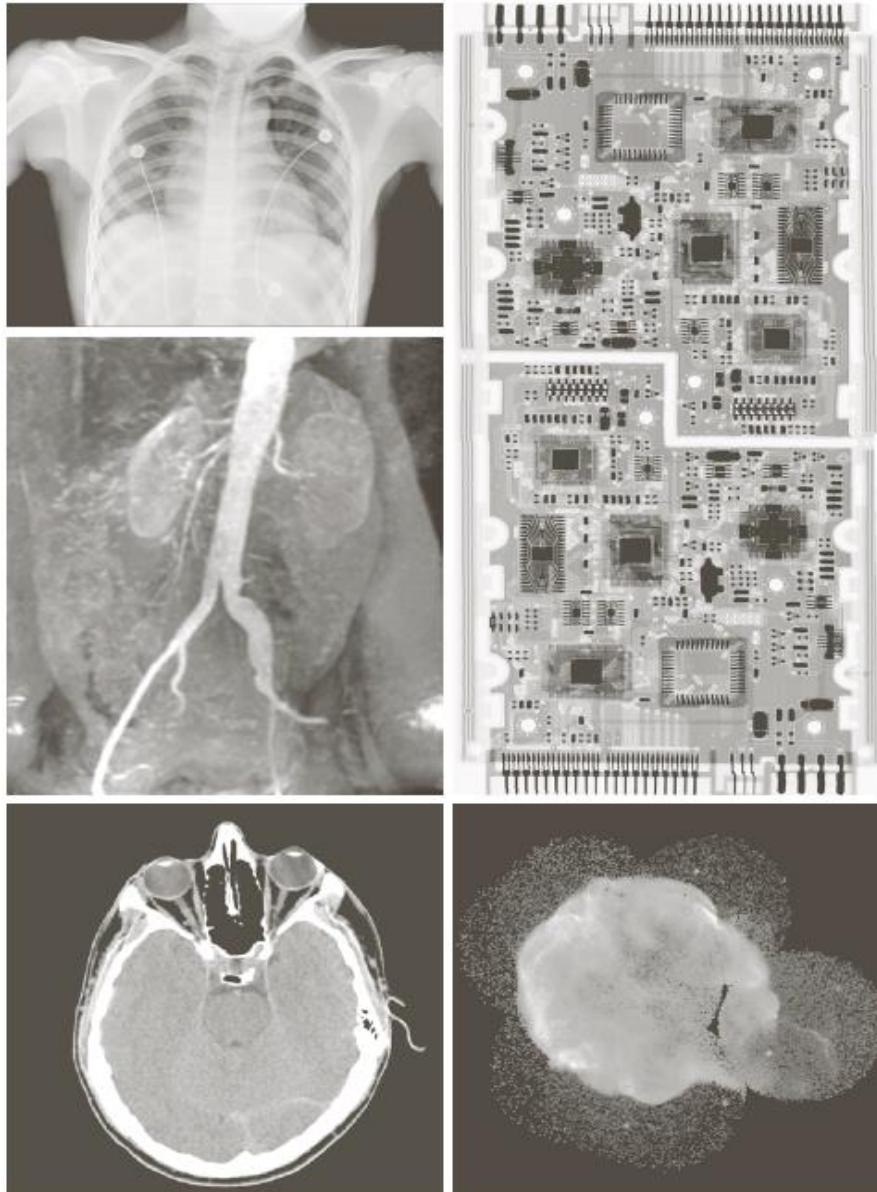
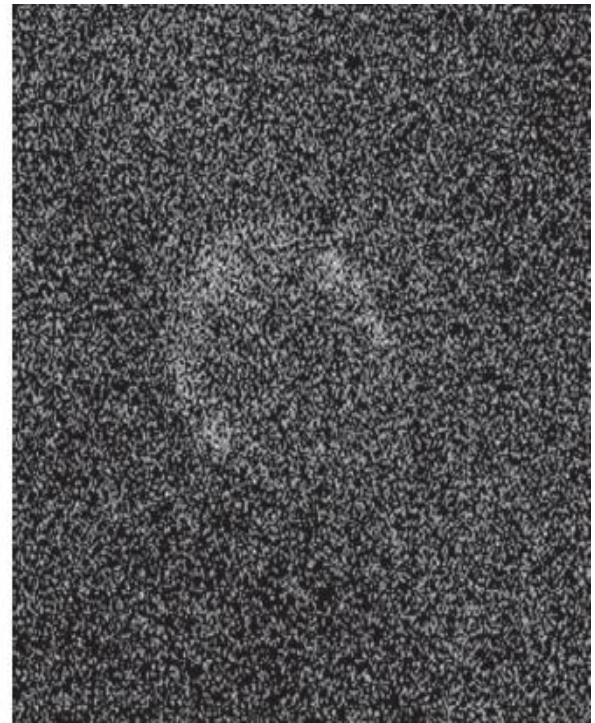
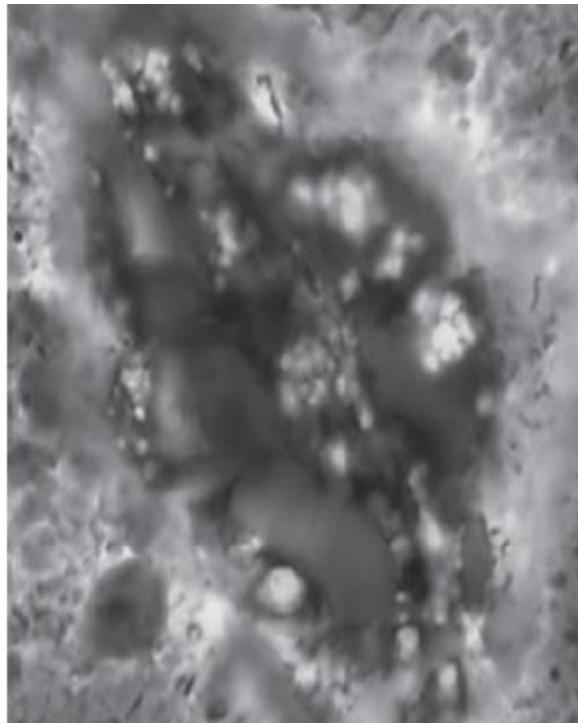
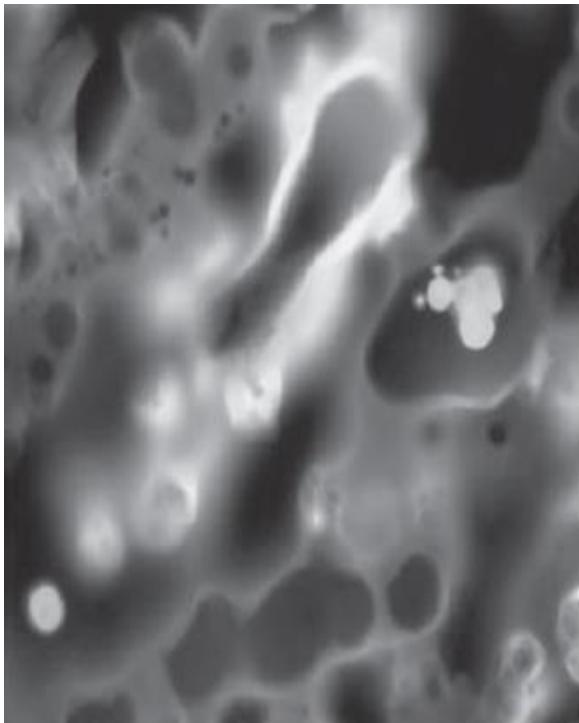


Figure 1.8

Examples of ultraviolet imaging. (a) Normal corn. (b) Corn infected by smut. (c) Cygnus Loop.

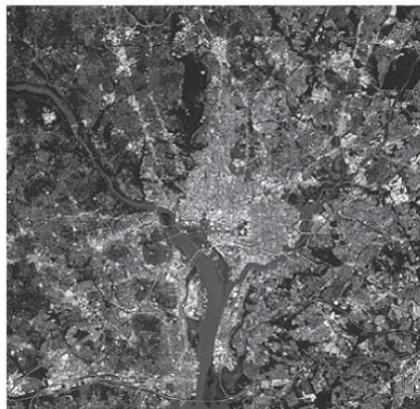


a b c

Figure 1.10

LANDSAT satellite images of the Washington, D.C. area.
The numbers refer to the thematic bands in Table 1.1.

1



2



3



4



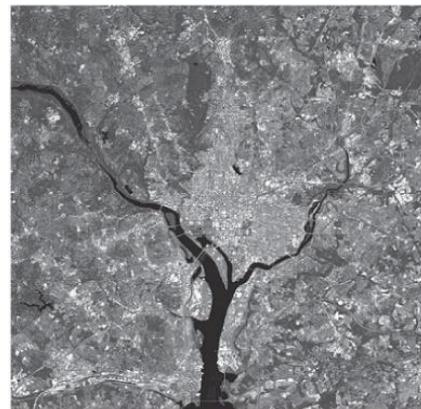
5



6



7



Images can be: binary, grayscale and color

- ▶ Binary images:

Binary images are images that have been quantized to two values, usually denoted 0 and 1, but often with pixel values 0 and 255, representing black and white as image below:



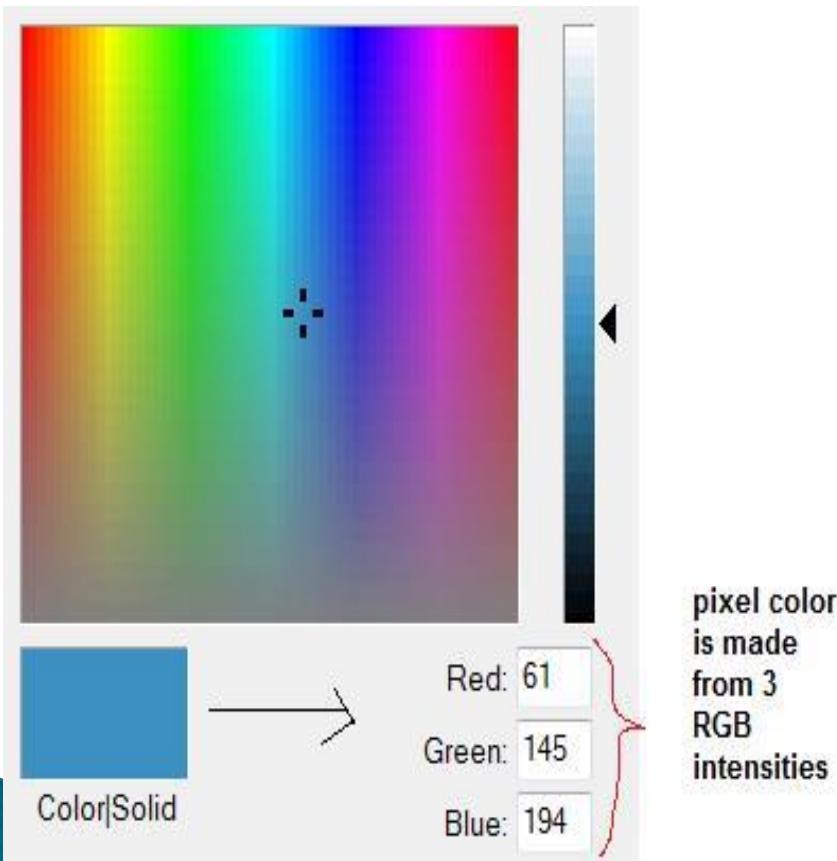
Grayscale images:

A grayscale (or gray level) image is simply one in which the only colors are shades of gray (0 - 255) as shown below:

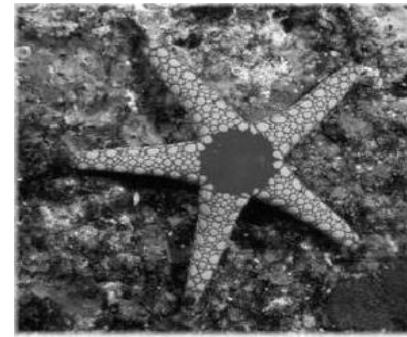


Color images:

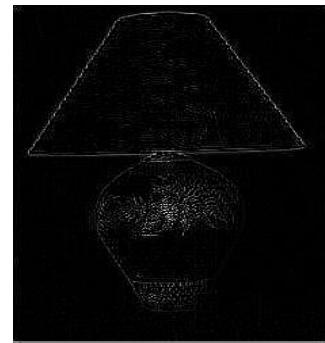
A color image contains pixels each of which holds three intensity values corresponding to the red, green, and blue or (RGB) as shown below:



Some Examples of DIP (Color to BW)



Some Examples of DIP (Edges)



Some Examples of DIP (Blurring)



More Formal Description of DIP:

DIP encompasses processes whose inputs and outputs are images, and in addition, includes processes that extract Attributes from images.

Low-Level Processes:

IP to reduce noise, contrast enhancement, image sharpening

Mid-Level Processes:

Segmentation, Classification (Using Attributes Extraction).

High-Level Processes:

“making sense” of an ensemble of recognized objects, image analysis (human vision type function/performance)

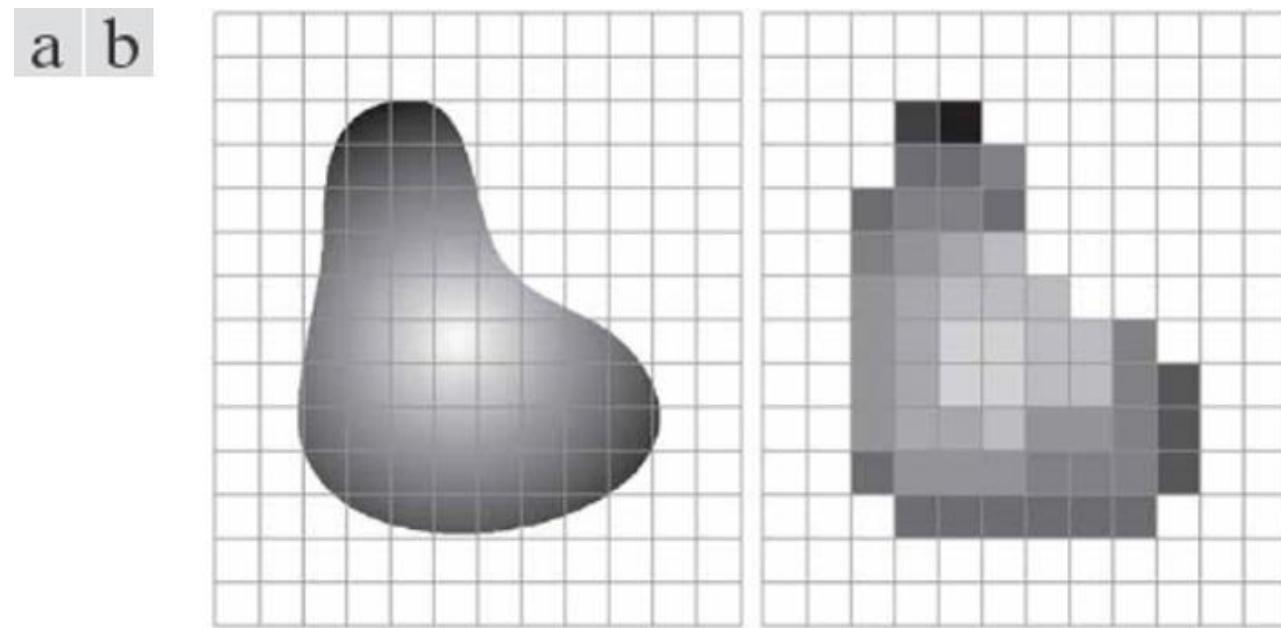
Chapter 2:

2.4 Image Sampling and Quantization

- Consider an **image** as a 2-D function $f(x,y)$
- The above image may be continuous in **coordinates** x, y and in **amplitude**.
- Digitizing the coordinate values is called **sampling**.
- *Digitizing the amplitude values is called **quantization**.*

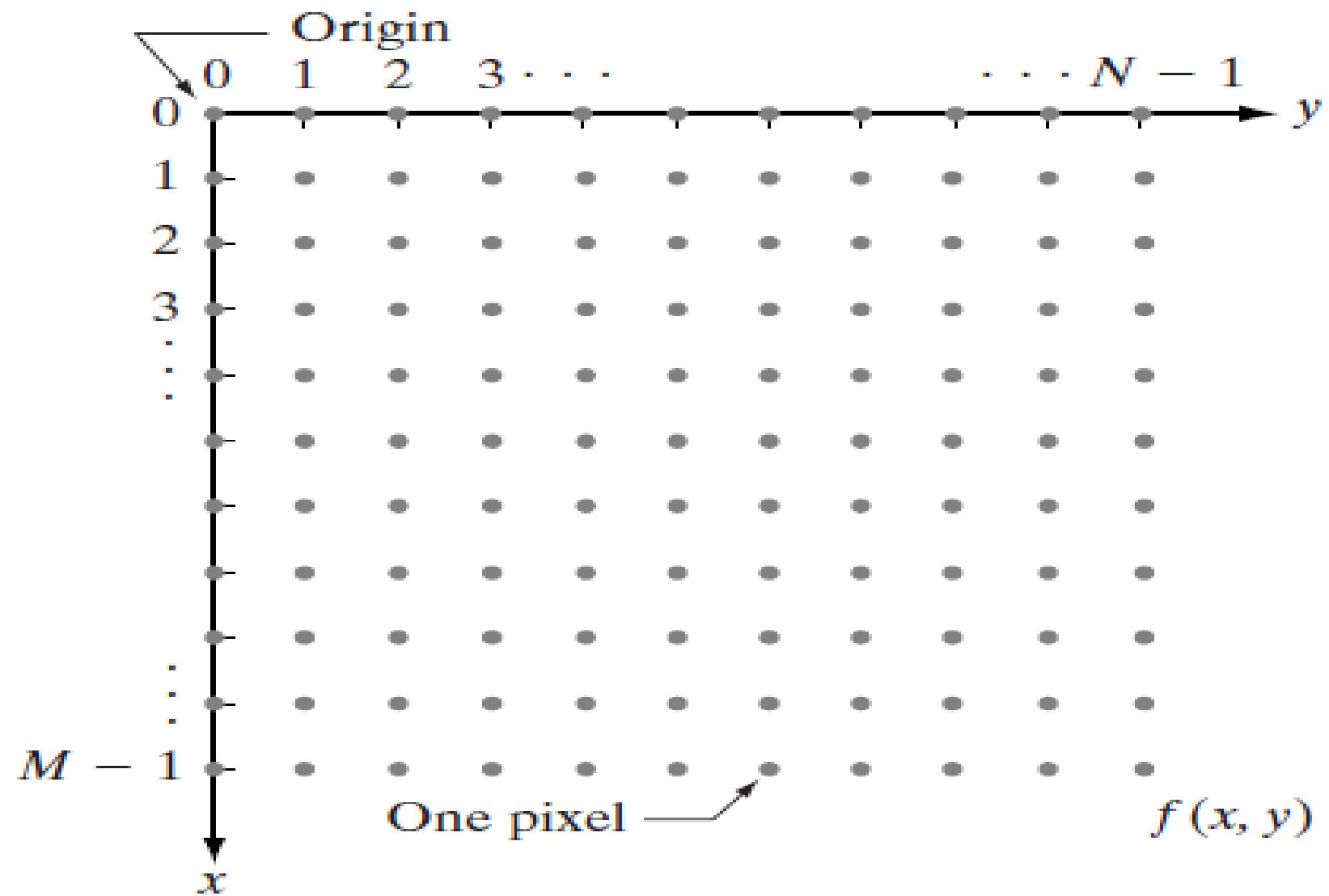
Figure 2.17

- (a) Continuous image projected onto a sensor array.
- (b) Result of image sampling and quantization.



2.4.Contd. Representing Digital Images

- A **digital** image $f(x,y)$ can be considered as **2-D array** of **M rows** and **N columns**.
- $x=0,1,2....M-1$ and $y=0,1,2.....N-1$.
- x, y are called **Spatial Coordinates**.
- The **values of f** at x, y are called **Intensities**.
- Symbol **L** is usually used to represent the Discrete Intensity Levels.
- Images can be **monochromatic (gray scale)** or **chromatic (colour)**.



Matrix form:

- ▶ MxN digital image can be represented in the following **compact** matrix form:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}.$$

- ▶ Each **element** of this matrix array is called an *image element, picture element, pixel, or pel*.

Size and Storage:

- ▶ The number of gray **levels** typically is an integer power of 2:

$$L = 2^k.$$

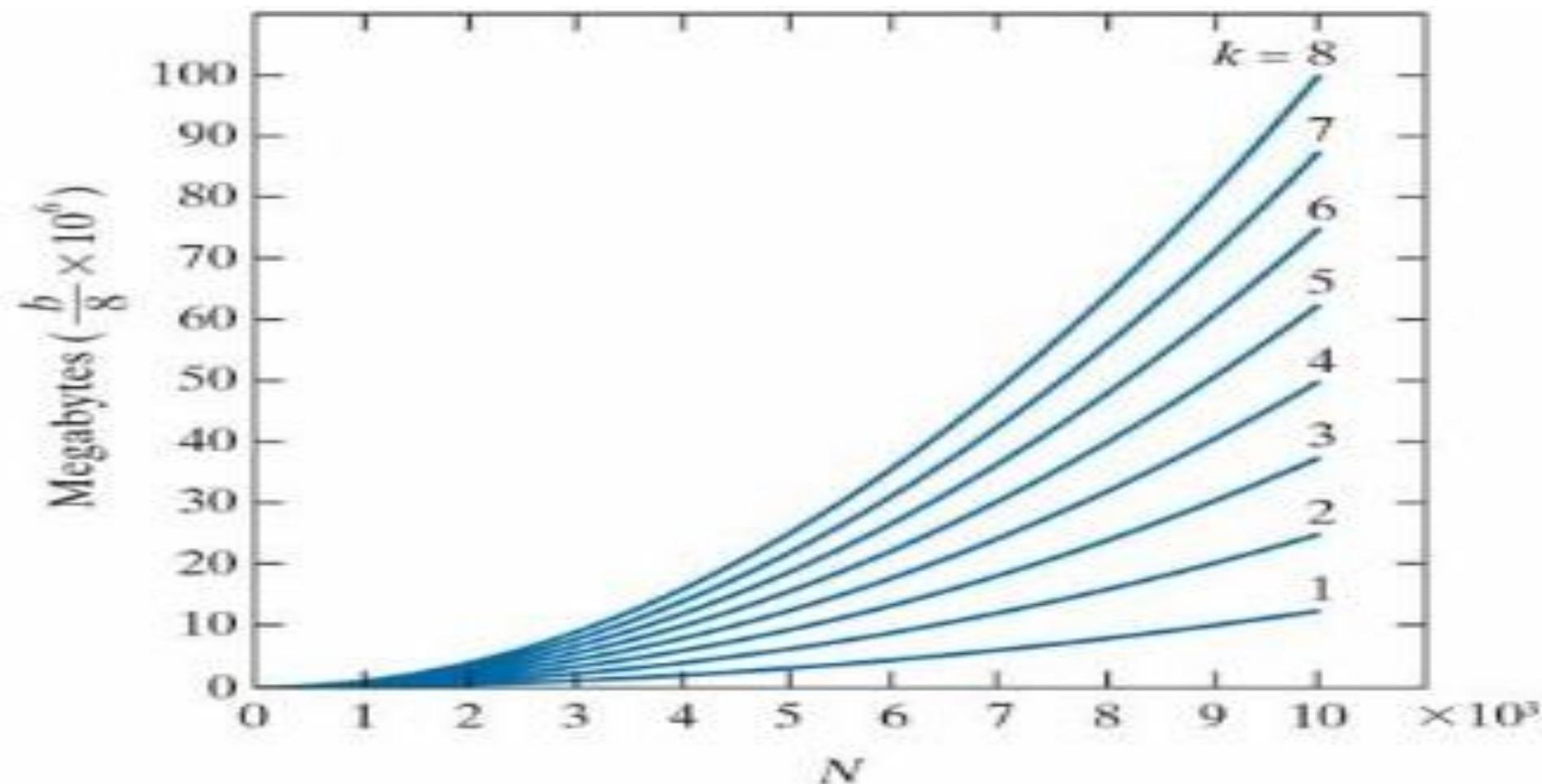
- ▶ The number, b , of **bits** required to **store** a digitized image is:

$$b = M \times N \times k.$$

- ▶ “**k-bit image**”: For example, an image with 256 possible gray-level values is called an 8-bit image.

Figure 2.21

Number of megabytes required to store images for various values of N and k .



Example:

A 3-bit image has how many intensity levels?

$$L=2^3=8$$

An 8-bit image has how many intensity levels?

$$L=2^8=256$$

EXAMPLE:

A 3-bit image of size 32x32 pixels requires
----- bits for storage.

$$B = M \times N \times k = 32 \times 32 \times 3 = 3072$$

2.4.Contd. Spatial and Gray-Level Resolution

- **Intuitively, Spatial Resolution (SR)** is a measure of the smallest discernable detail in an image.
- **Quantitatively:** It is line pairs per unit distance, **dots (pixels) per unit distance**.
- **Examples:** Newspapers are printed with a resolution of **75dpi**, magazines at **133dpi**, glossy brochures at **175dpi**.

EXAMPLE:

Which of following images has lower SR?

Image A of size 32x32

Image B of size 64x64

Image C of 8-bit

Intensity Resolution:

- **Intensity Resolution (IR)** refers to the smallest discernable change in intensity **level**.
- The number of Intensity **Levels** usually is an integer power of 2.
- The most **common** number is 8 bits (256 levels).

EXAMPLE:

Which of following images has lower IR?

Image A of size 32x32

Image B of size 64x64

Image C of 8-bit

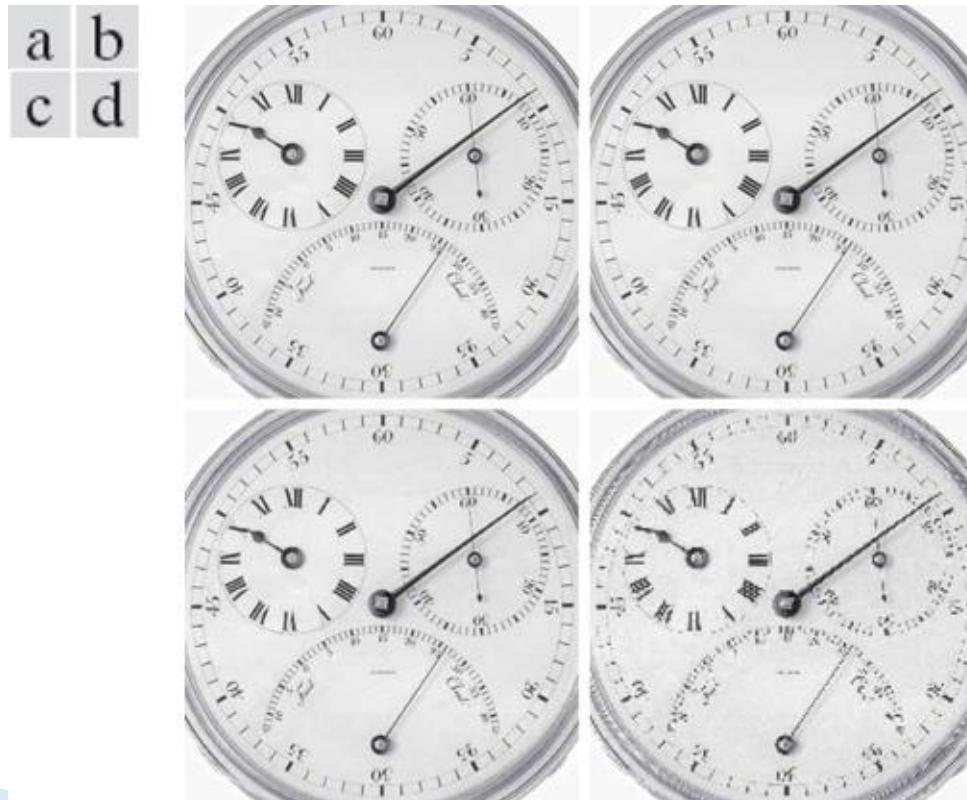
Image D of 3-bit

Example 2.2:

- Fig 2.23 shows the effects of **reducing SR** in an image.
- The images are shown in 930, 300, 150, and **72 dpi**.
- The **lower resolution** images are smaller than the original.
- The original is of **size 2136x2140 pixels**, but the 72 dpi image is an array of size 165x166.

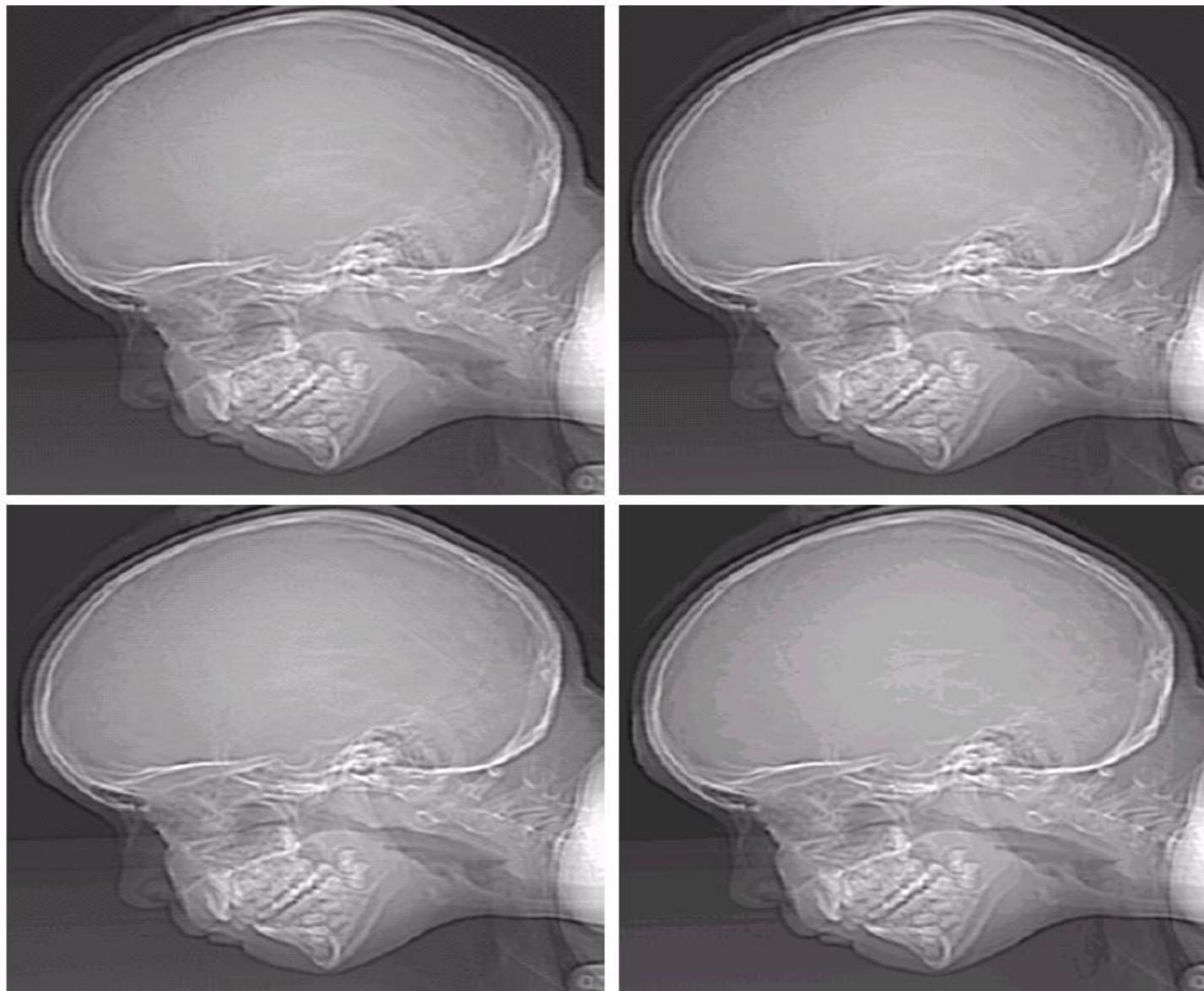
Figure 2.23

Effects of reducing spatial resolution. The images shown are at: (a) 930 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi.



Example 2.3:

- Effect of varying the **Intensity Levels**.
- Here we keep the number of samples constant and **reduce** the number of intensity levels from **256** to **2**.
- **Fig 2.21 (from 3rd edition of book)** shows this effect.



a b
c d

FIGURE 2.21
(a) 452×374 ,
256-level image.
(b)–(d) Image
displayed in 128,
64, and 32 gray
levels, while
keeping the
spatial resolution
constant.

e
f
g
h

FIGURE 2.21
(Continued)
(e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

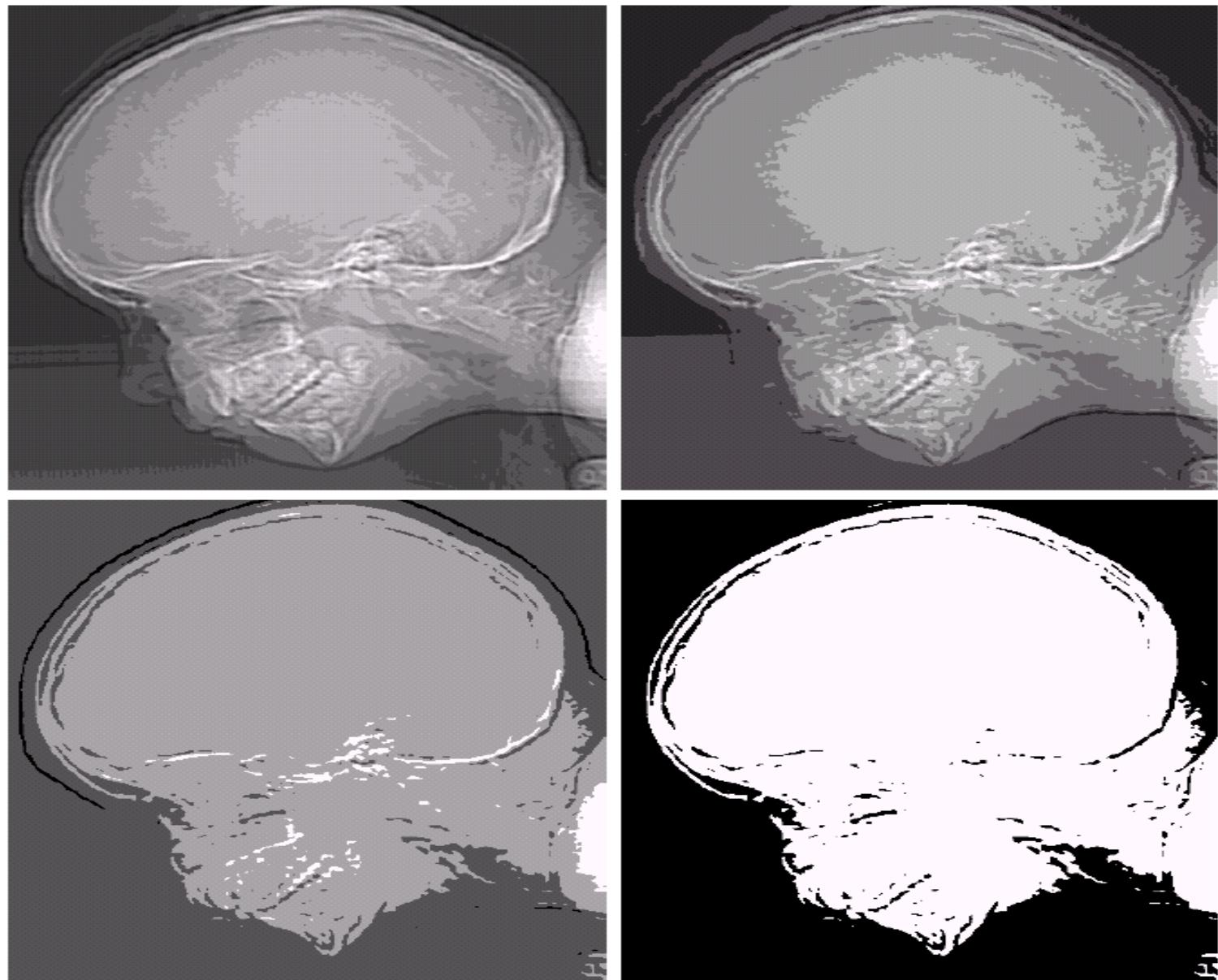


Figure 2.24 (1 of 2)

(a) 2022×1800 , 256-level image. (b)-(d) Image displayed in 128, 64, and 32 intensity levels, while keeping the image size constant.
(Original image courtesy of the National Cancer Institute.)

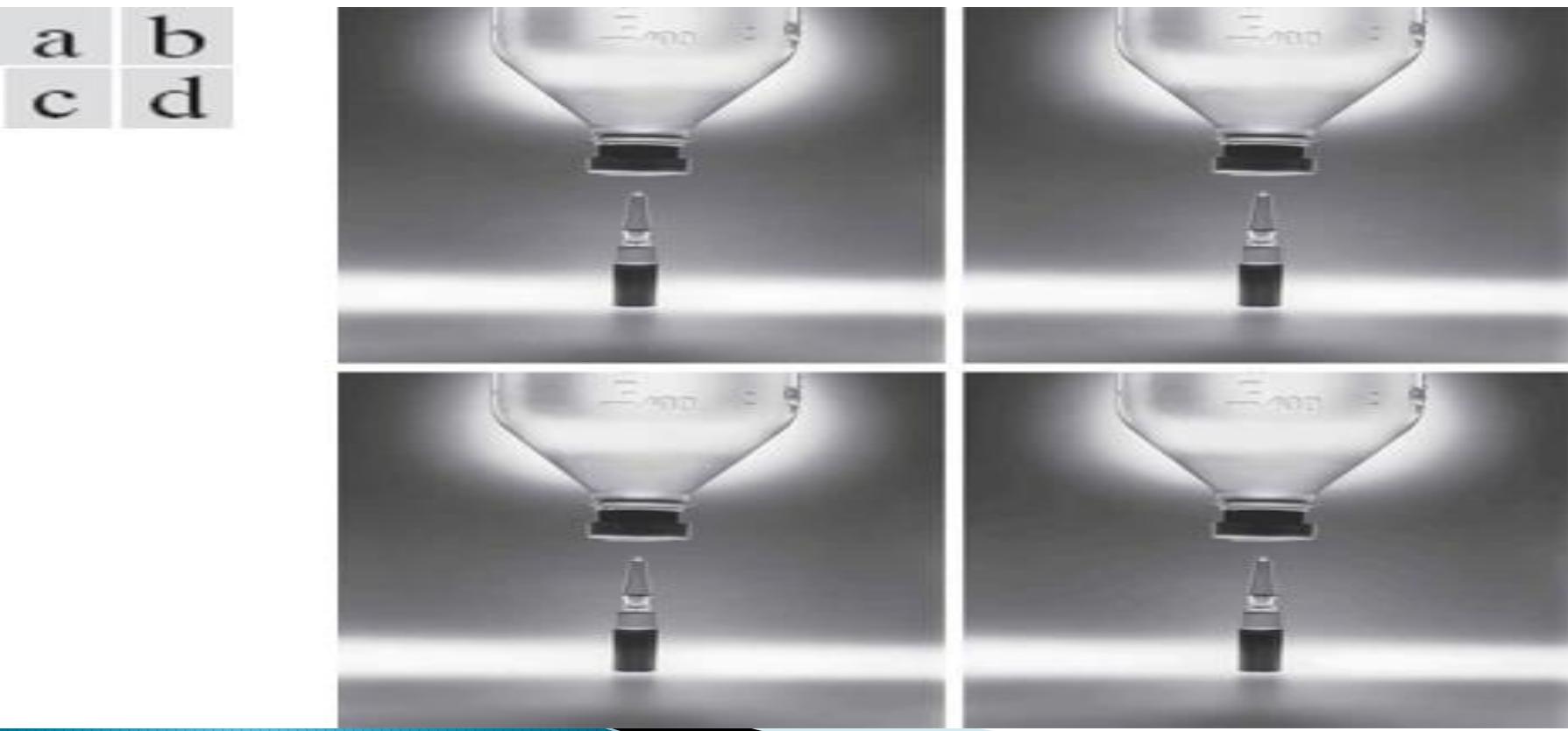
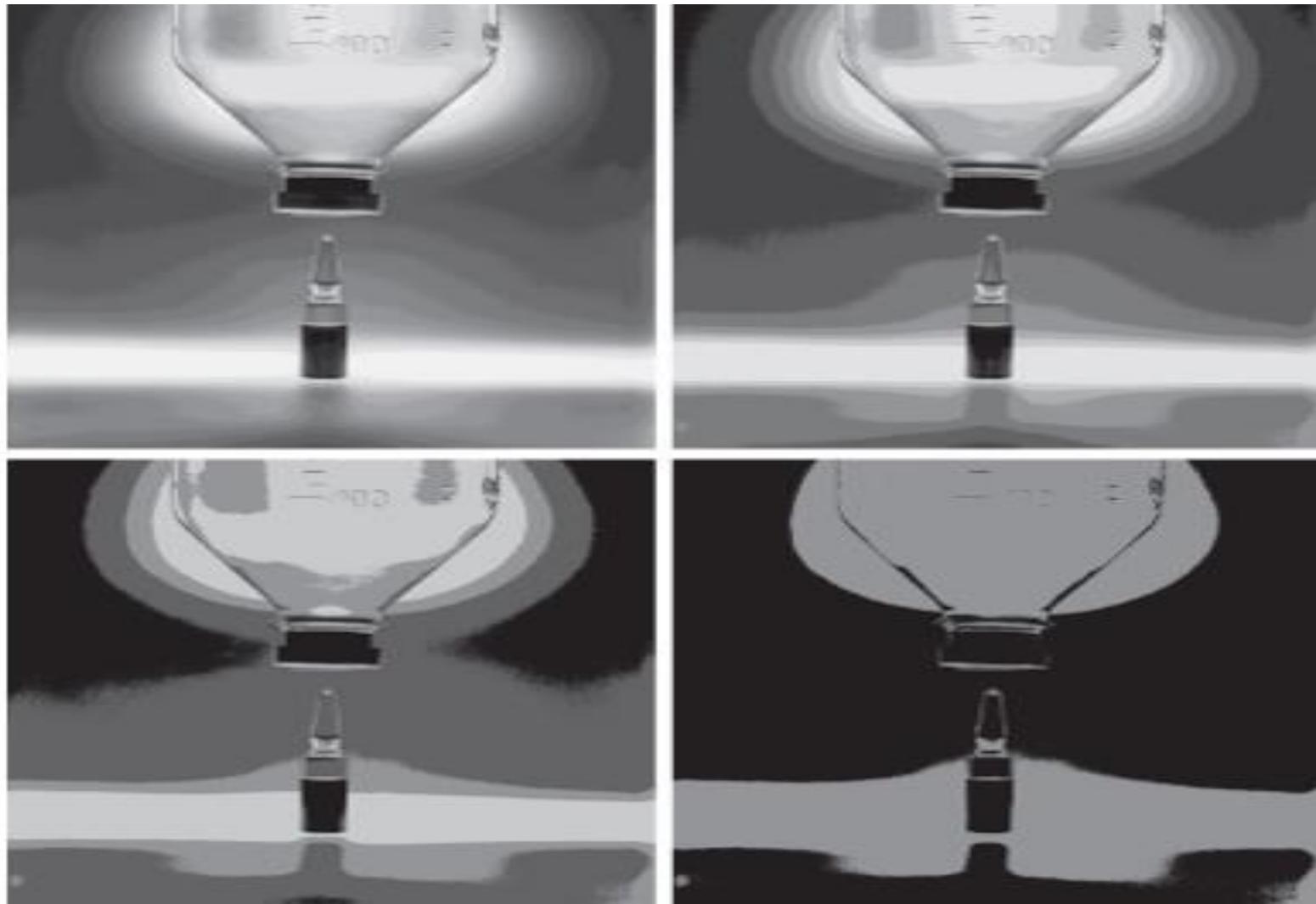


Figure 2.24 (2 of 2)

(Continued) (e)–(h) Image displayed in 16, 8, 4, and 2 intensity levels. (Original image courtesy of the National Cancer Institute.)

e f
g h



2.4.Contd. Image Interpolation

- Interpolation is the process of using known data to **estimate** values at **unknown** location
- It is **Used for** zooming, shrinking, rotating, geometric corrections.
- The first two are Image **resizing** tasks and come under image resampling methods.
- In this chapter we study Interpolation for **Image resizing** (zooming, shrinking).

Bilinear Interpolation:

- Here we use the **4 nearest neighbours** to estimate the intensity at a given location.
- Let (x,y) denote the **coordinates** of the location to which we want to assign an intensity value and let $v(x,y)$ denote that value, then:

$$v(x,y) = ax + by + cxy + d$$

- Here the 4 coefficients are determined from the 4 equations in 4 unknowns using the 4 nearest neighbours of point (x,y) .

Bicubic Interpolation:

- ▶ Involves 16 nearest neighbours of a point.

$$v(x,y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

- ▶ Here the 16 coefficients are determined from the 16 equations in 16 unknowns using the 16 nearest neighbours of point (x,y).

2.5 Some Basic Relationships Between Pixels

Neighbors of a Pixel

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

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

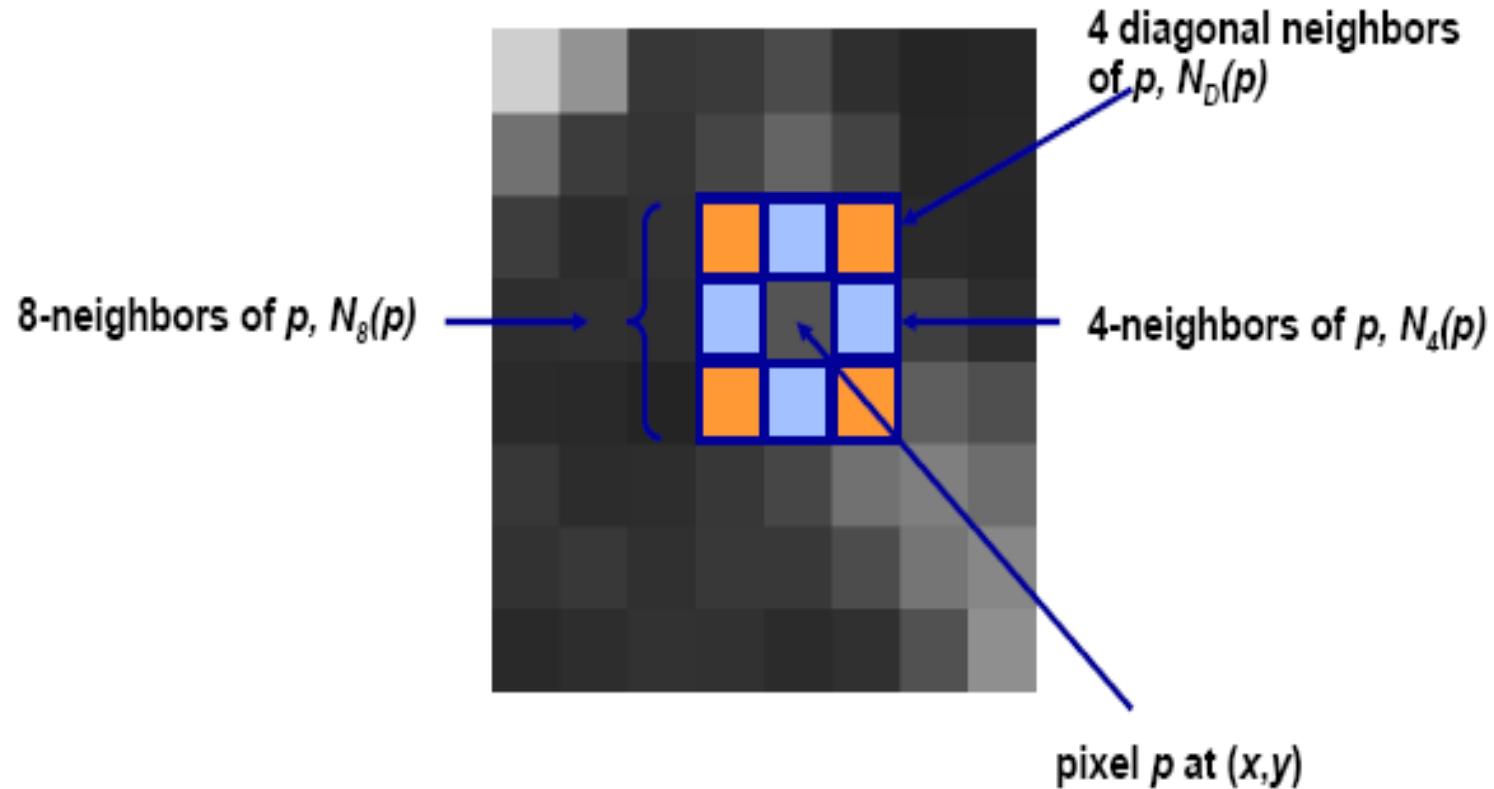
This set of pixels, called the *4-neighbors* of p , is denoted by $N_4(p)$. Each pixel is a unit distance from (x, y) , and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

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, are called the *8-neighbors* of p , denoted by $N_8(p)$. As before, some of the points in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of the image.

Example on pixel neighbors:



2.5 Contd. Adjacency, Connectivity, Regions, and Boundaries

- ▶ **Connectivity** between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as **regions** and **boundaries**.
- ▶ To establish if two pixels are connected, it must be determined if they are **neighbours** 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 maybe 4-neighbors, but they are said to be connected only if they have the same value.

- Let V be the set of intensity values used to define adjacency.
- In a **binary image**, $V=\{1\}$ if we are referring to adjacency of pixels with value 1.
- In a **gray-scale image**, the idea is the same, but set 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 3 types of adjacency as shown in next slide.

- (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). 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 $N_4(p) \cap N_4(q)$ has no pixels whose values are from V .

Examples of adjacencies

0 1 1

0 1 0

0 0 1

Arrangement of
pixels in a binary
image

0 1 1

0 1 0

0 0 1

4-adjacent pixels

0 1 1

0 1 0

0 0 1

8 - adjacent pixels

0 1 1

0 1 0

0 0 1

m - adjacent pixels

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

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

where

$$(x_0, y_0) = (x, y), (x_n, y_n) = (s, t), \text{ and pixels } (x_i, y_i) \text{ and } (x_{i-1}, y_{i-1})$$

are adjacent for

$$1 \leq i \leq n.$$

In this case, n is the *length of the path*.

Connected set:

- Let S represent a subset of pixels in an image.
- 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 .
- If it only has one connected component, then set S is called a **connected set**.

Region:

- Let R be a subset of pixels in an image.
- We call R a region of the image if R is a **connected** set.
- Two regions are said to be **adjacent** if their union forms a connected set.
- Regions that are **not adjacent** are said to be **disjoint**.
- We consider 4- and 8- adjacency when referring to regions.

Boundary:

- The boundary (also called border or contour) of a region R is the set of points that are **adjacent** to points in the **complement** of R .

Edge:

- Edges are formed from pixels with **derivative** values that exceed a preset threshold.
- There is a key **difference** between *edges* & boundaries.
- The boundary of a finite region forms a closed path and is thus a “**global**” concept.
- The idea of an edge is a “**local**” concept that is based on a measure of gray-level **discontinuity** at a point.
- It is **possible** to link edge points into edge segments, and sometimes these segments are linked in such a way that correspond to boundaries, but this is not always the case.
- The one **exception** in which edges and boundaries correspond is in **binary** images.
- For time being, it is helpful to think of edges as intensity **discontinuities** and boundaries as **closed paths**.

2.5Contd. Distance Measures

For pixels p , q , and z , with coordinates (x, y) , (s, t) , and (v, w) , respectively, D is a *distance function or metric* if

- (a) $D(p, q) \geq 0$ ($D(p, q) = 0$ iff $p = q$),
- (b) $D(p, q) = D(q, p)$, and
- (c) $D(p, z) \leq D(p, q) + D(q, z)$.

The *Euclidean distance* between p and q is defined as

$$D_e(p, q) = \left[(x - s)^2 + (y - t)^2 \right]^{\frac{1}{2}}.$$

For this distance measure, the pixels having a distance less than or equal to some value r from (x, y) are the points contained in a disk of radius r centered at (x, y) .

The D_4 distance (also called *city-block distance*) between p and q is defined as

$$D_4(p, q) = |x - s| + |y - t|.$$

In this case, the pixels having a D_4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y) . For example, the pixels with D_4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

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

The pixels with $D_4 = 1$ are the 4-neighbors of (x, y) .

The D_8 distance (also called *chessboard distance*) between p and q is defined as

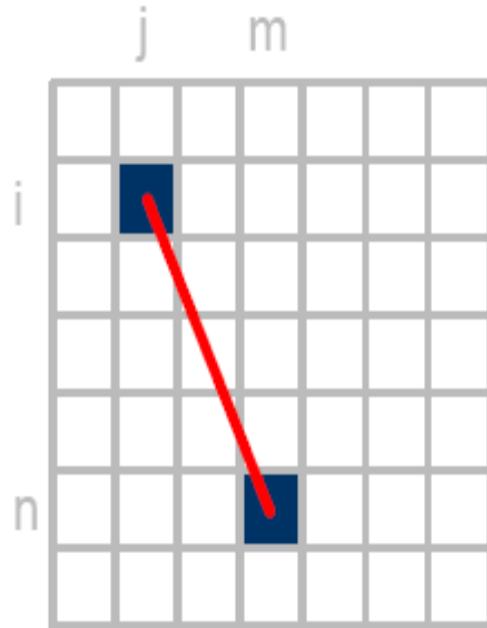
$$D_8(p, q) = \max(|x - s|, |y - t|).$$

In this case, the pixels with D_8 distance from (x, y) less than or equal to some value r form a square centered at (x, y) . For example, the pixels with D_8 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

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

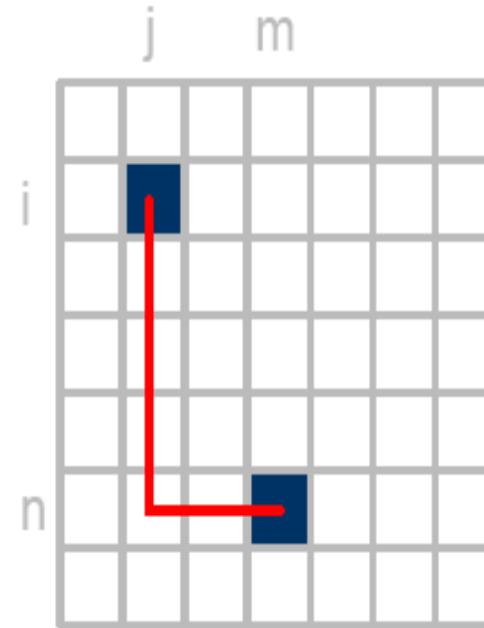
The pixels with $D_8 = 1$ are the 8-neighbors of (x, y) .

Examples of Distance measures:



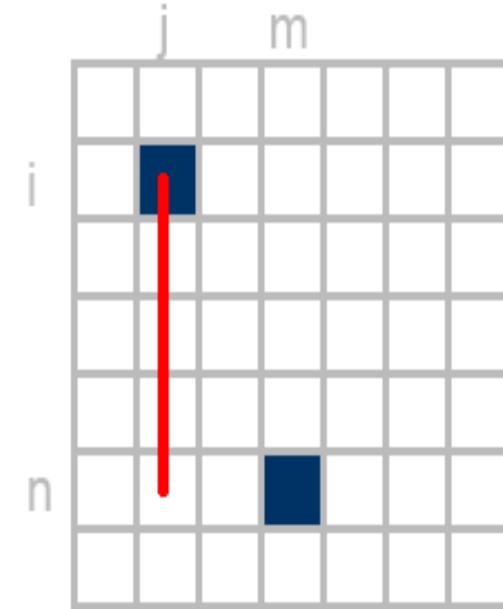
Euclidean Distance

$$= \sqrt{(i-n)^2 + (j-m)^2}$$



City Block Distance

$$= |i-n| + |j-m|$$



Chessboard Distance

$$= \max[|i-n|, |j-m|]$$

Course Software Package:

- Matlab will be used for this semester offer

[www.mathworks .com](http://www.mathworks.com)

- Matlab Tutorials:

- General:

<http://www.maths.dundee.ac.uk/~ftp/na-reports/MatlabNotes.pdf>

<http://users.ece.gatech.edu/bonnie/book/TUTORIAL/tutorial.pdf>

<http://www.mathworks.com/products/image/demos.jsp>

- DIP:

http://www.cs.otago.ac.nz/coursework/cosc451/Resources/matlab_ipt_tutorial.pdf

- ▶ More 3rd party mat lab and dip tutorials are at:
- ▶ http://www.imageprocessingplace.com/root_files_V3/tutorials.htm

Examples of MATLAB commands:

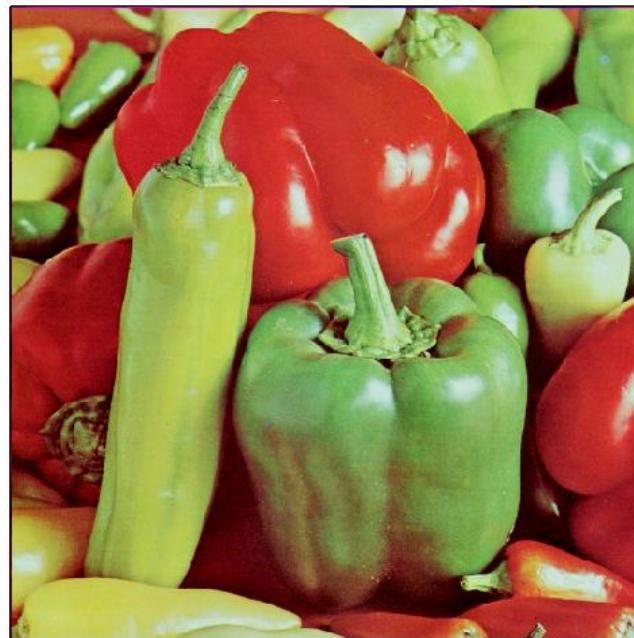
The screenshot shows the MATLAB Editor window with the title bar "C:\MATLAB6p5\work\tutorial.m". The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains various icons for file operations like Open, Save, and Print. A status bar at the bottom right shows "script" in the active tab, "Ln 3" (line 3), and "Col 22".

```
1 % Tutorial M-file
2 % Created by: Someone
3 % Created on: 9/11/03
4 % Last revised: 10/29/03
5
6 %load a color .bmp image and convert to grayscale
7 A = 'C:\MATLAB6p5\work\splash2.bmp'; %designate matrix A as the specified file
8 B = imread(A, 'bmp'); %matrix B loaded with bitmap file specified by A
9 figure(1), imshow(B); %show image B in figure window 1
10 C = rgb2gray(B); %convert color image to grayscale image
11 figure(2), imshow(C); %show image C in figure window 2
12
13 %load a color .jpg image and convert to grayscale
14 D = 'C:\MATLAB6p5\work\b747.jpg'; %designate matrix D as the specified file
15 E = imread(D, 'jpg'); %matrix E loaded with JPEG file specified by D
16 figure(3), imshow(E); %show image E in figure window 3
17 F = rgb2gray(E); %convert color image to grayscale
18 figure(4), imshow(F); %show image F in figure window 4
```

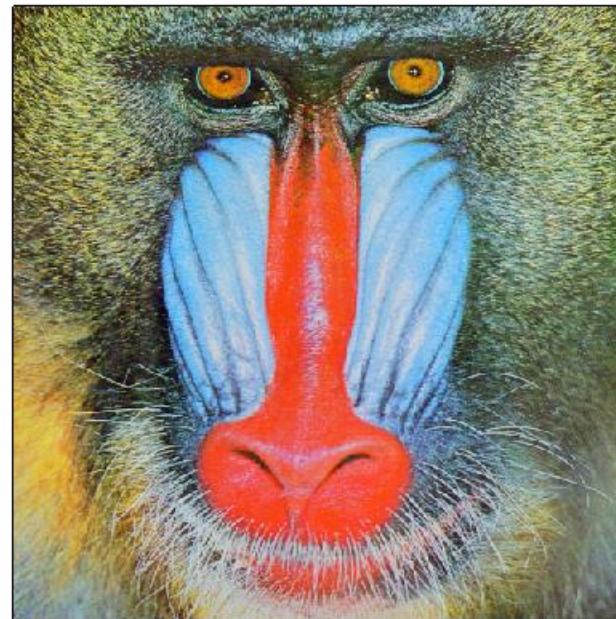
Matlab Practice Questions (ungraded)

1. **I/O and display:** Read the images peppers.png and baboon.png. Display them as colour images.
2. **Conversion:** Convert them to gray-scale images and display them.
3. **Processing:** Reduce their sizes to 200x200 using bilinear and bicubic interpolation methods available in Matlab.

Peppers.png



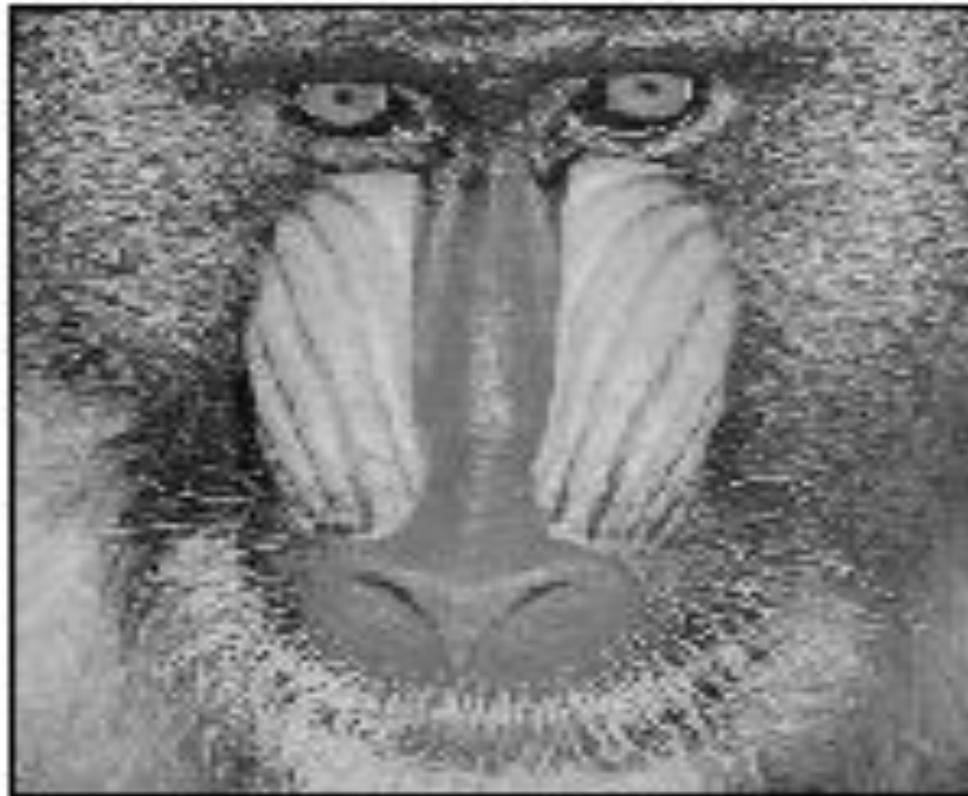
Baboon.png



200x200, bilinear



200x200. bicubic



Examples on MATLAB codes

Consider the given image of the flower.



- a) Load the flower image into Matlab and name it as A.
- b) Convert image A into another image B which is of type grayscale.
- c) Create a new image C from image B whose data type should be of type double.
- d) Convert C into a binary image D.
- e) Print all images in one figure.

Suggested codes for the solution

```
a. A=imread('flower.tif');  
b. B=rgb2gray(A);  
c. C=double(B);  
d. D=im2bw(uint8(C));  
e. figure(1)  
subplot(4,1,1), imshow(A,[])  
subplot(4,1,2), imshow(B,[])  
subplot(4,1,3), imshow(C,[])  
subplot(4,1,4), imshow(D,[])
```

