# Fundamentals

Delivered by

**Dr. Subarna Chatterjee**
subarna.cs.et@msruas.ac.in

# Text Book

## a. Essential Reading

1. Gonzalez and Woods. (2009) Digital Image Processing, 3rd Edition, Prentice Hall
2. Chanda Bhabatosh and Majumder Dwijesh. (2009) Digital Image Processing and Analysis, PHI Willis J.
3. M. C. Bishop, 2006, Pattern Recognition and Machine Learning, Springer
4. S. Theodoridis, K. Koutroumbas, 2008, Pattern Recognition, Academic Press.

## b. Recommended Reading

1. Bernd Jahne. (2005) Digital Image Processing, Springer, 6th Edition
2. Stephane Maarchand Maillet and Yazid M. Sharaiha. (2000) Binary Digital Image Processing: A Discrete Approach, Academic Press
3. R. Szeliski, 2010, Computer Vision: Algorithms and Application, Springer-Verlag Inc.
4. D. A. Forsyth, J. Ponce, 2003, Computer Vision: A Modern Approach, Pearson Education

## c. Online Tutorial

1. https://nptel.ac.in/courses/117/105/117105135/

# Digital Image Fundamentals

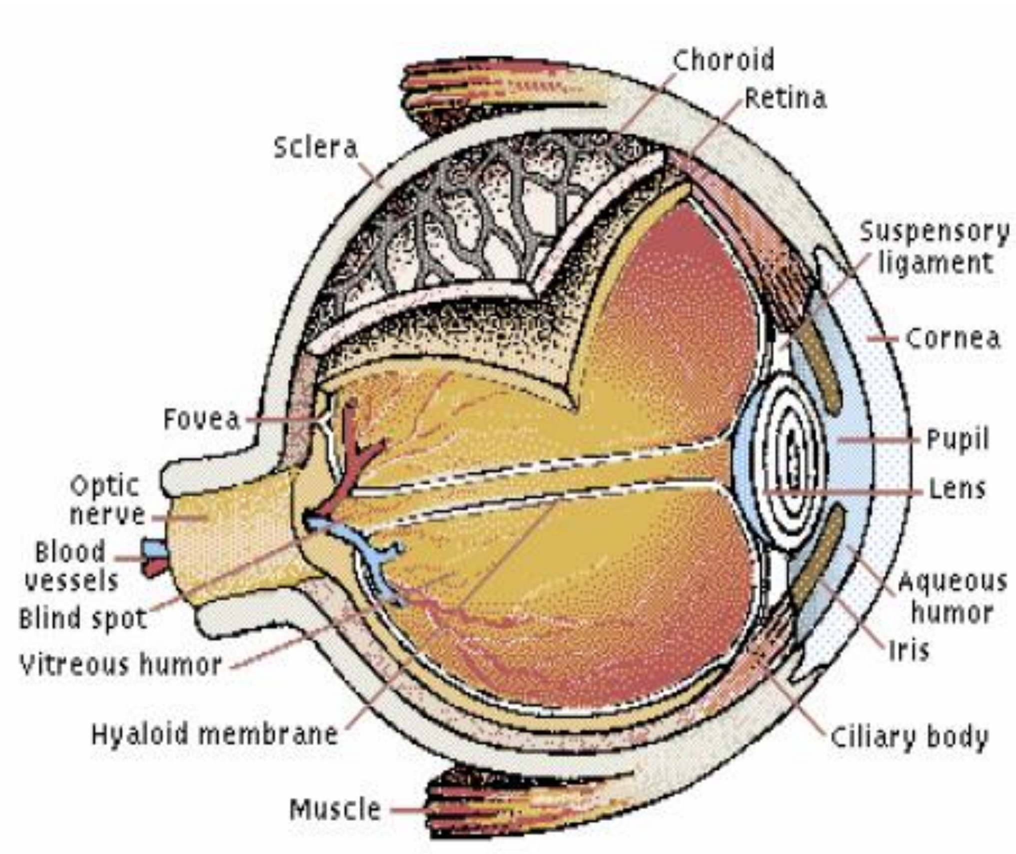
- Human visual system
- A simple image model
- Sampling and quantization

# Human Visual System

- Brightness adaptation
- Brightness discrimination
- Weber ratio
- Mach band pattern
- Simultaneous contrast

# Human Visual System

- Elements of visual perception




The amount of light entering the eye is controlled by the pupil, which dilates and contracts accordingly. The cornea and lens, whose shape is adjusted by the ciliary body, focus the light on the retina, where receptors convert it into nerve signals that pass to the brain.

# Human Visual System

- Elements of visual perception
  - **Cones**
    - 6 – 7 million in each eye
    - Photopic or bright-light vision
    - Highly sensitive to color
  - **Rods**
    - 75 – 150 million
    - Not involved in color vision
    - Sensitive to low level of illumination (scotopic or dim-light vision)
  - An object appears brightly colored in daylight will be seen colorless in moonlight (why)

# Visual Perception

- Image formation in the eye(Pinhole Camera Model)
  - Distance between center of lens and retina (focal length) vary between 14-17 mm.
  - Image length h = 17(mm) x (15/100)

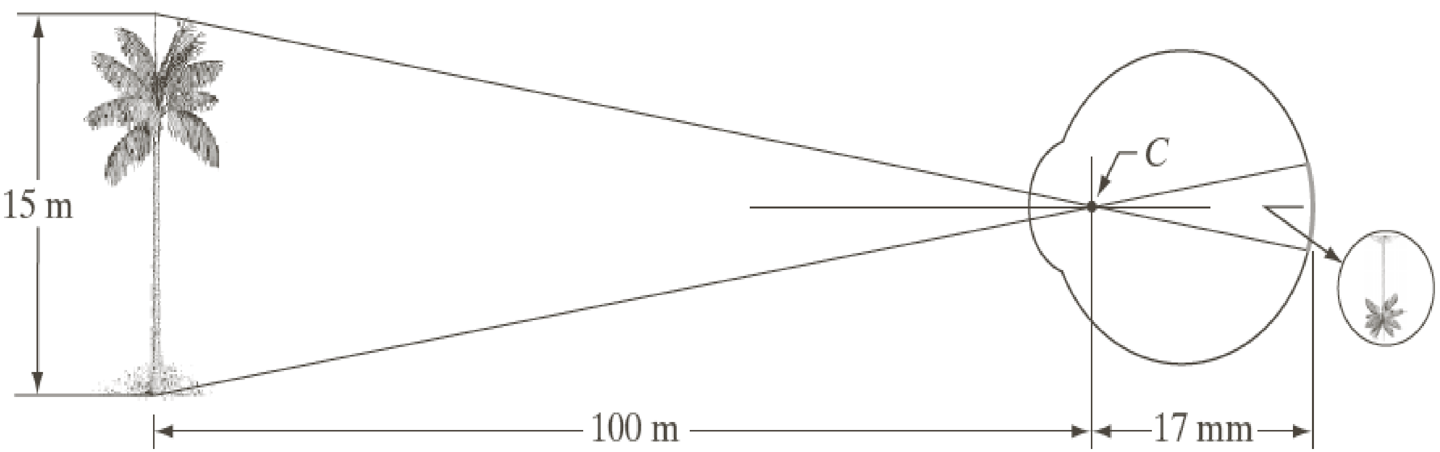


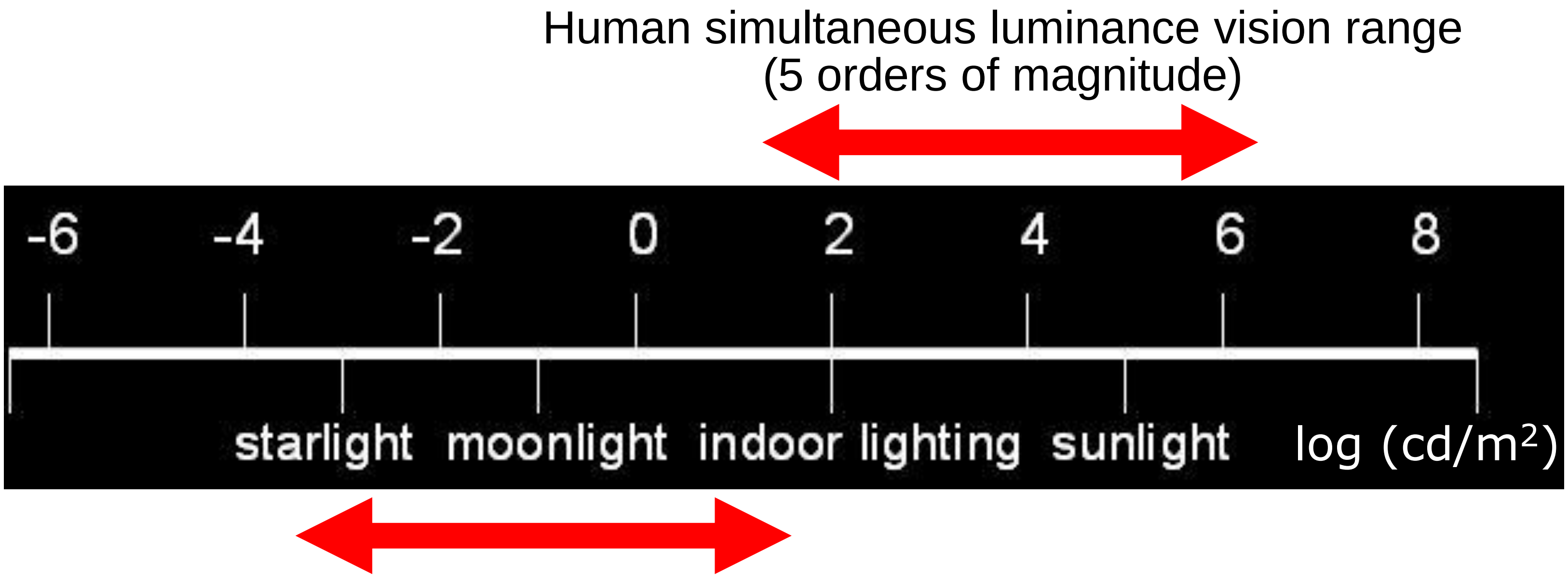

**FIGURE 2.3** Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

15 / 100 = h / 17
⇒h = 2.55 mm

focal length (min. refractive power)

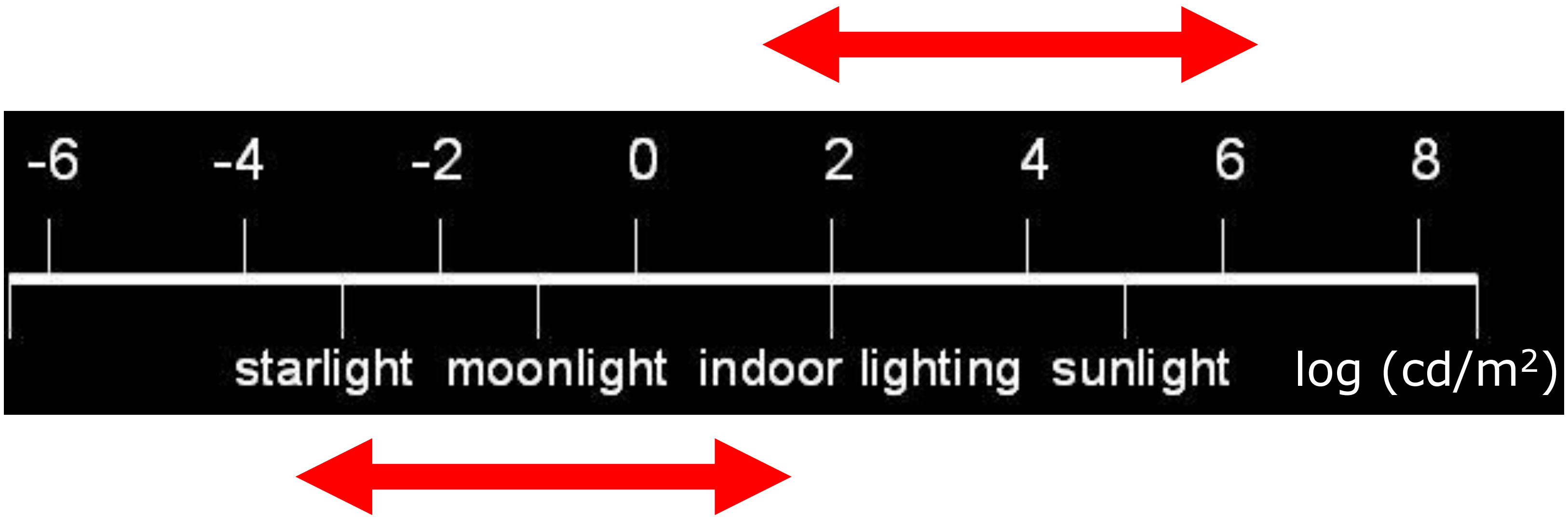
# Human Visual System

- Light is just a particular part of the electromagnetic spectrum that can be sensed by the human eye
- The electromagnetic spectrum is split up according to the wavelengths of different forms of energy



# Human Visual System

Human simultaneous luminance vision range
(5 orders of magnitude)



# Cont..

- Brightness adaptation in the Human Eye
  - HVS can adapt to light intensity range on the order of $10^{10}$
  - human eye can adapt over 10 orders of magnitude!
    - 6 orders in phototopic vision (cones)
    - accomplished by brightness adaptation (changes in the overall sensitivity)
    - much smaller range for each brightness adaptation level $B_a$
  - Subjective brightness is a log function of the light intensity incident on the eye
  - brightness discrimination
    - poor at low levels of illumination
    - better with increasing illumination

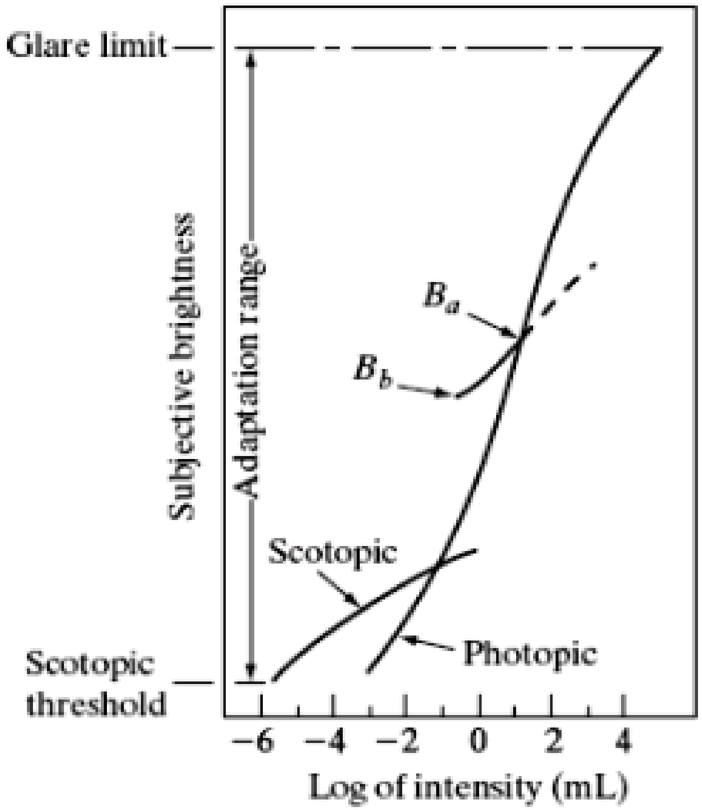


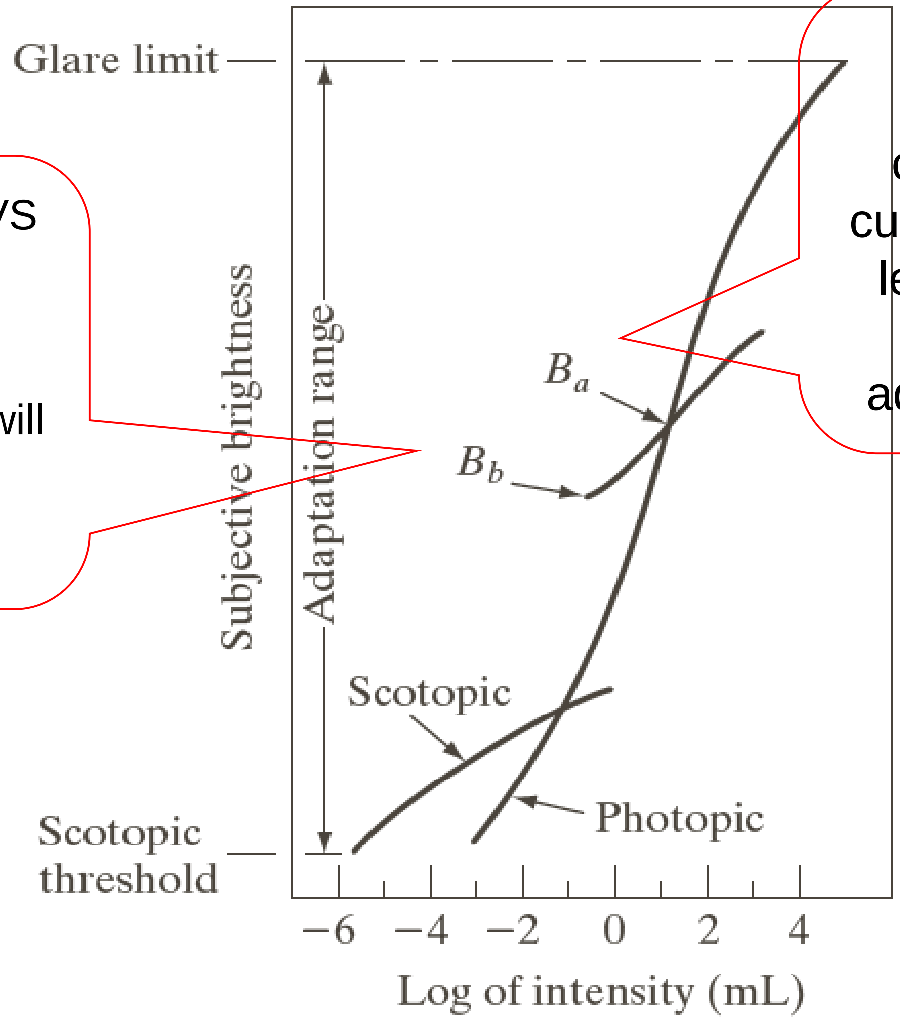

**FIGURE 2.4** Range of subjective brightness sensations showing a particular adaptation level.

# Human Visual System

- **Brightness adaptation**
  - The HVS cannot operate on such range (10 orders of magnitude) simultaneously
  - It accomplishes this through (brightness) adaptation
  - The total intensity level the HVS can discriminate simultaneously is rather small in comparison (about 4 orders of magnitude)

# Human Visual System

- Brightness adaptation




Sensitivity of the HVS for the given adaptation level

Anything below Bb will be perceived as indistinguishable blacks

For a given observation condition, the current sensitivity level is call the brightness adaptation level

# Cont..

- Brightness discrimination

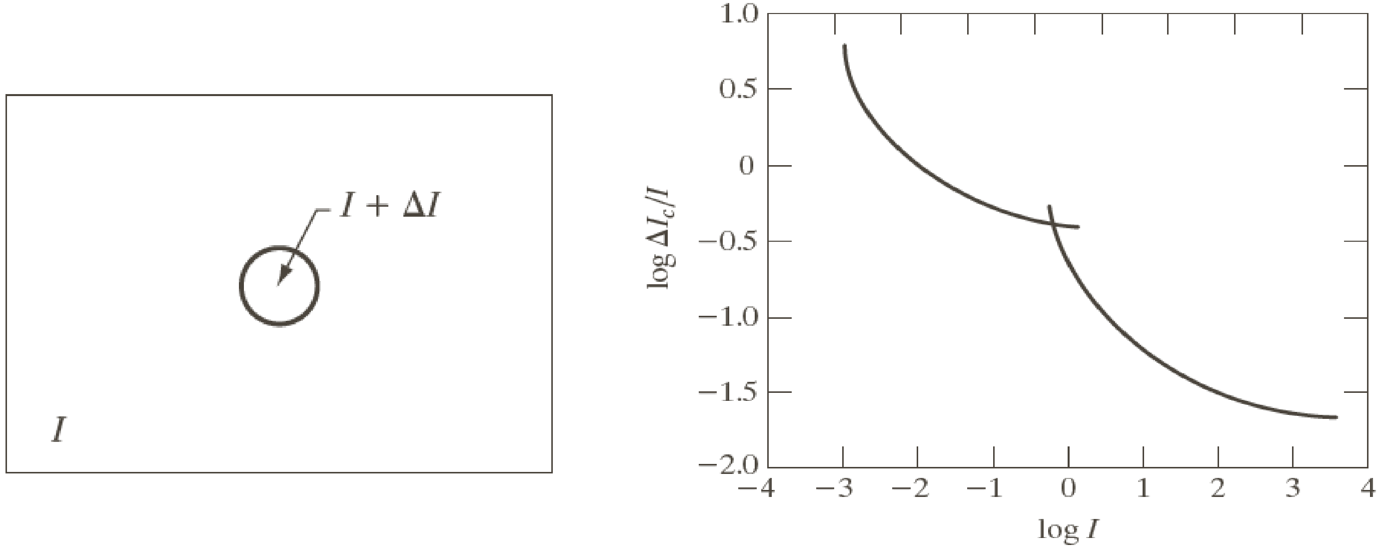
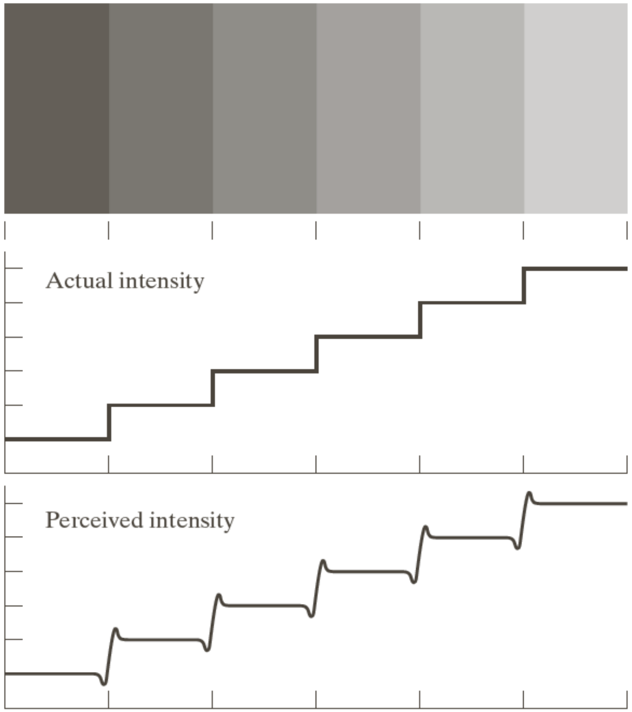



**FIGURE 2.6** Typical Weber ratio as a function of intensity.

- Perceivable changes at a given adaptation level
  - The **ratio** of increment of illumination to background of illumination is called as. **weber ratio**.(ie) Δi/i.
    - If the **ratio** (Δi/i) is small, then small percentage of change in intensity is needed. (i.e) good brightness adaptation.
    - If the ratio (Δi/i) is large , then large percentage of change in intensity is needed (i.e) poor brightness adaptation.

# Visual Perception

- Perceived brightness is not a simple function of intensity
  - Mach band pattern
    - stripes appear darker near a more intense stripe (and vice versa)
    - caused by inhibitory neural connections




a
b
c

**FIGURE 2.7** Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

# Visual Perception

- Perceived brightness is not a simple function of intensity
  - Mach band pattern
    - stripes appear darker near a more intense stripe (and vice versa)
    - caused by inhibitory neural connections
  - simultaneous contrast
    - a regions perceived brightness depends on the intensity in the neighbor hood

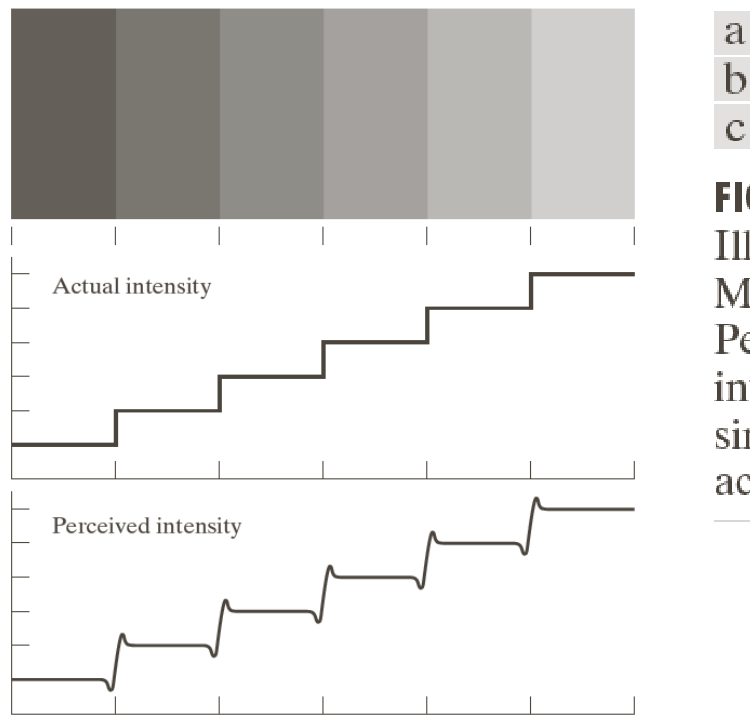
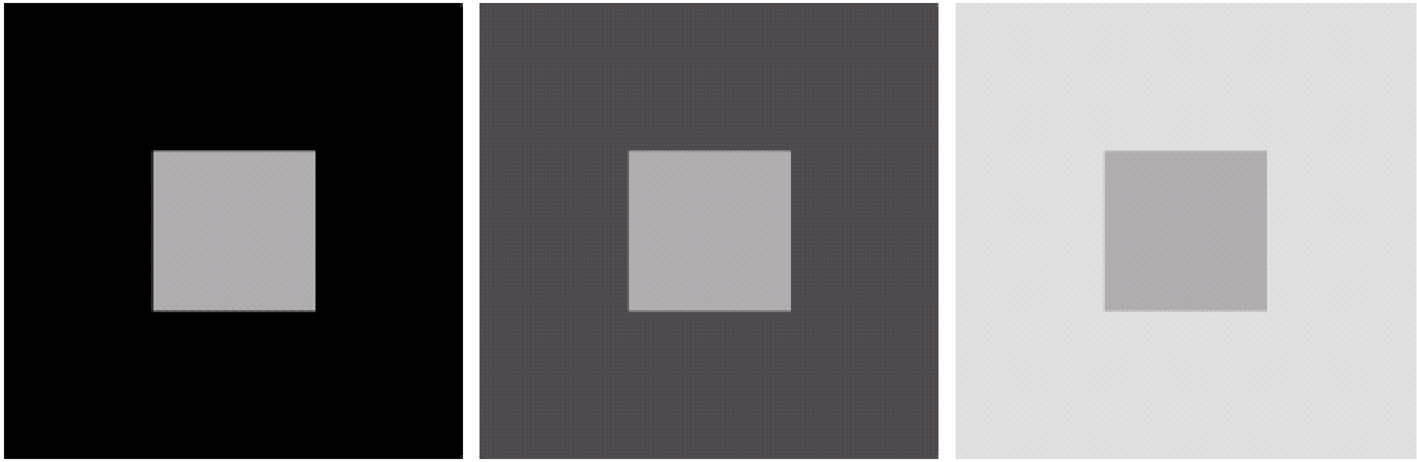



**FIGURE 2.7** Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

# Human Visual System

- Perceived brightness is not a simple function of intensity
  - Simultaneous contrast



a b c

**FIGURE 2.8** Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

# Sensation vs Perception

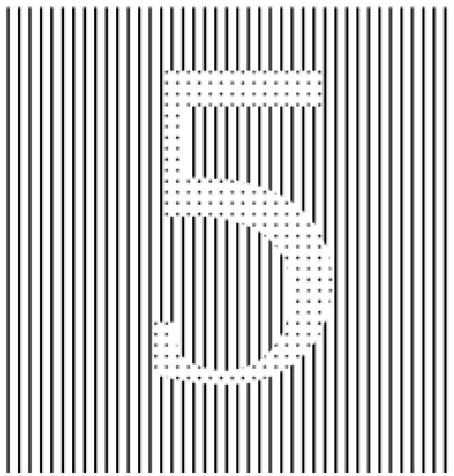
- Sensation
  - operation of basic sensory systems
  - result of physical stimuli and low-level processes
- Perception
  - involve higher-level processes in the percipient
  - memories
  - Expectations
  - Emotions
  - state of fatigue or alertness

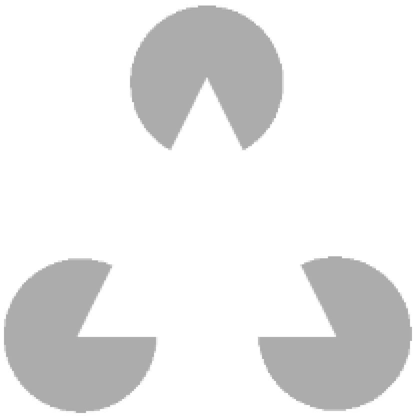
# Cont..

- Image Formation in the Human Eye
  - perceived brightness is not a simple function of intensity!
    - Mach bands
      - stripes appear darker near a more intense stripe (and vice versa)
      - caused by inhibitory neural connections
    - simultaneous contrast
      - a regions perceived brightness depends on the intensity in the neighborhood
  - optical illusions

# Optical Illusions

- Optical Illusions
  - the eye / brain fills in non existing information
  - perceives geometrical properties of an object wrongly

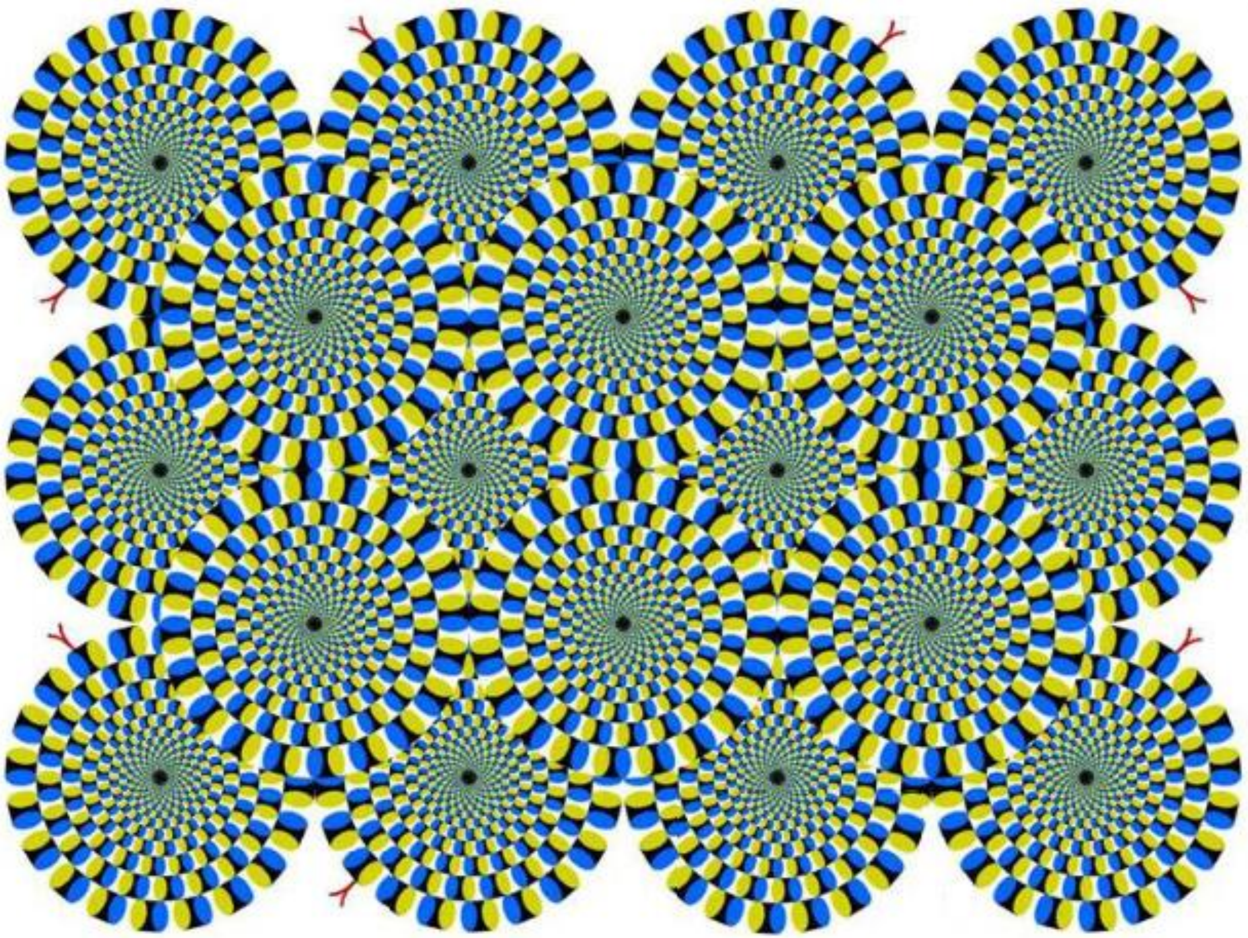




  - characteristic of the human visual system and not yet fully understood ...

# Optical Illusions

- movement created only in the brain



# Cont..

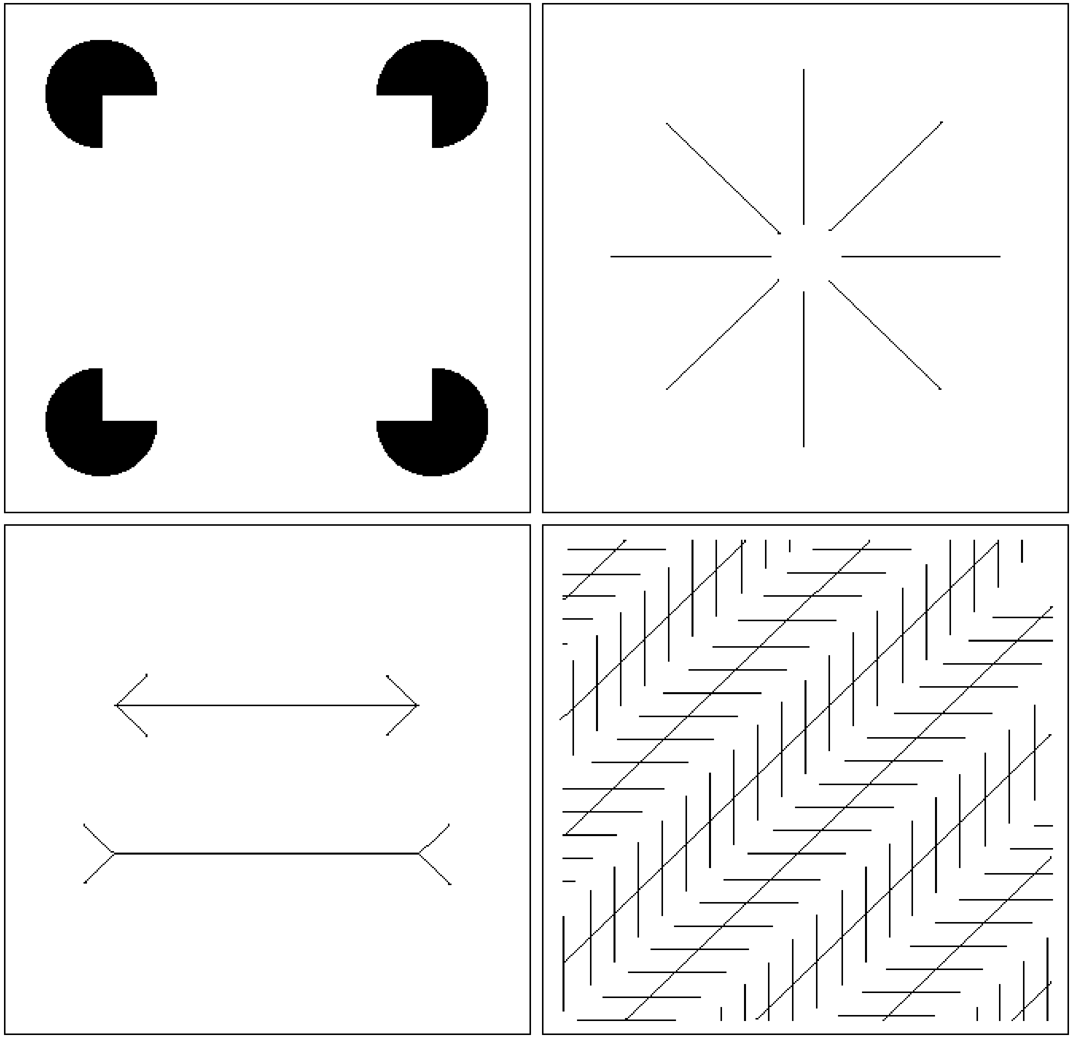
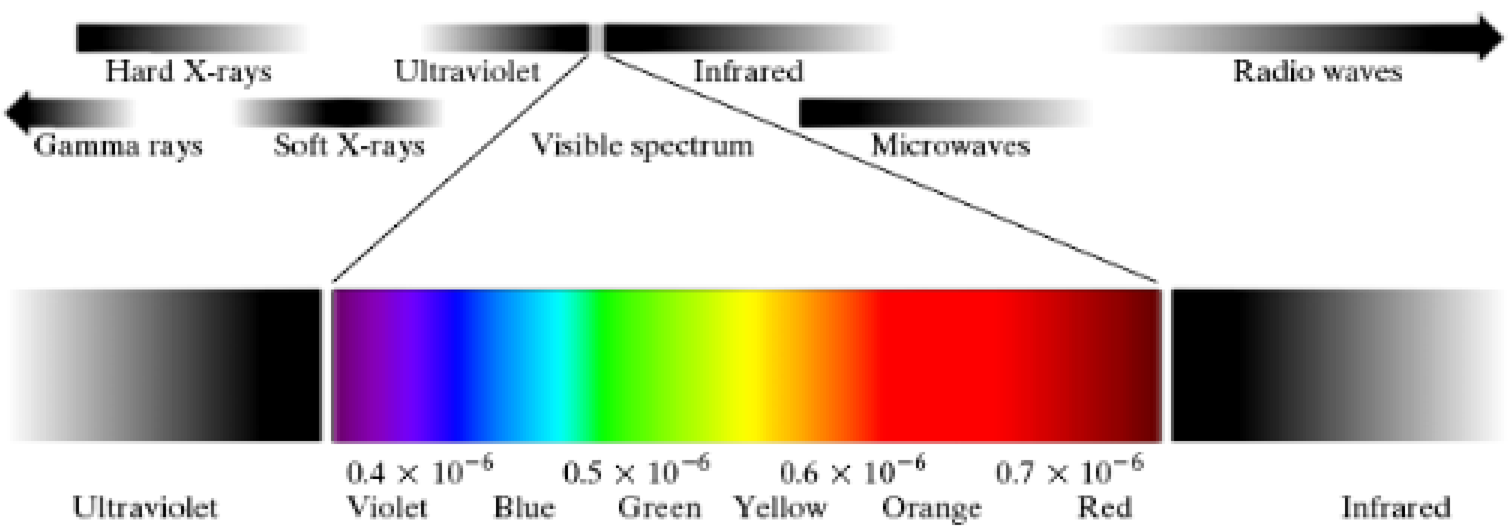



**FIGURE 2.9** Some well-known optical illusions.

- In Fig. 2.9(a), the outline of a square is seen clearly, despite the fact that no lines defining such a figure are part of the image. The same effect, this time with a circle, can be seen in Fig. 2.9(b); note how just a few lines are sufficient to give the illusion of a complete circle. The two horizontal line segments in Fig.2.9(c) are of the same length, but one appears shorter than the other. Finally, all lines in Fig.2.9(d)that are oriented at 45°are equidistant and parallel. Yet the crosshatching creates the illusion that those lines are far from being parallel. Optical illusions are a characteristic of the human visual system that is not fully understood.

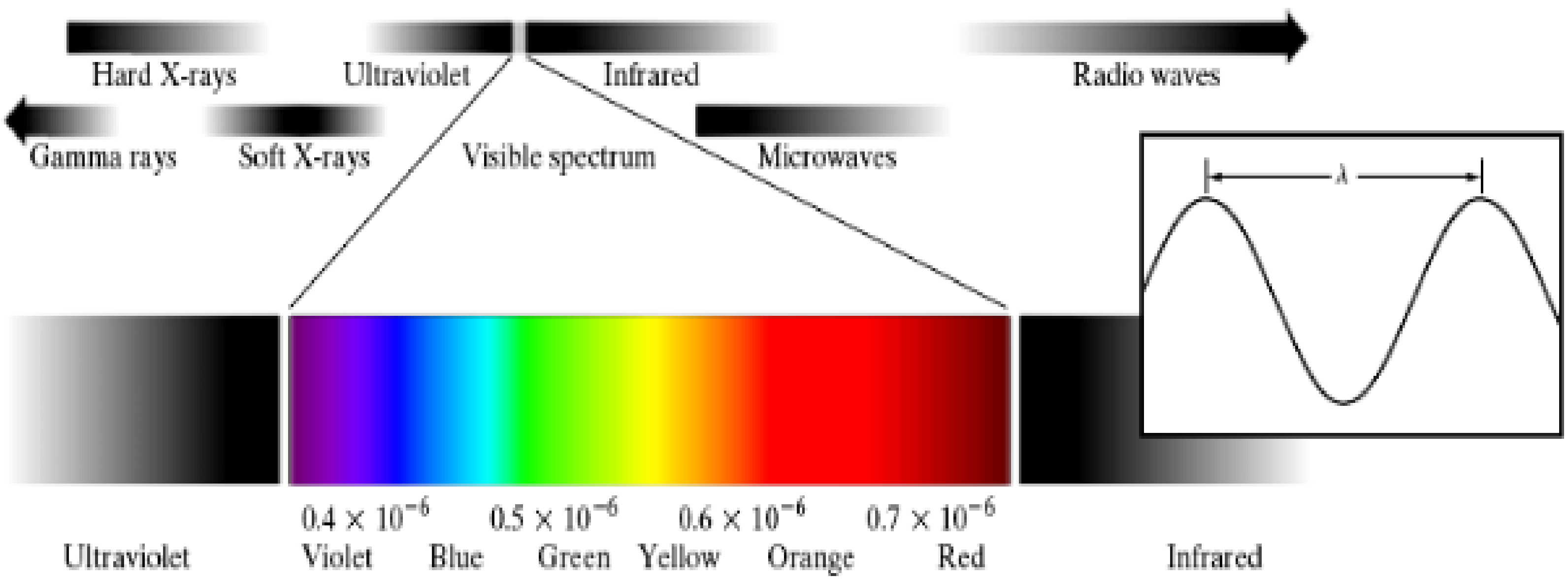
# Electromagnetic Spectrum

- The Electromagnetic Spectrum
  - we perceive only a small range of colours of the electromagnetic spectrum (~ 430nm –790nm)
    - gamma rays, X rays, ultraviolet light, visible spectrum, infrared, microwaves, radio waves, ...



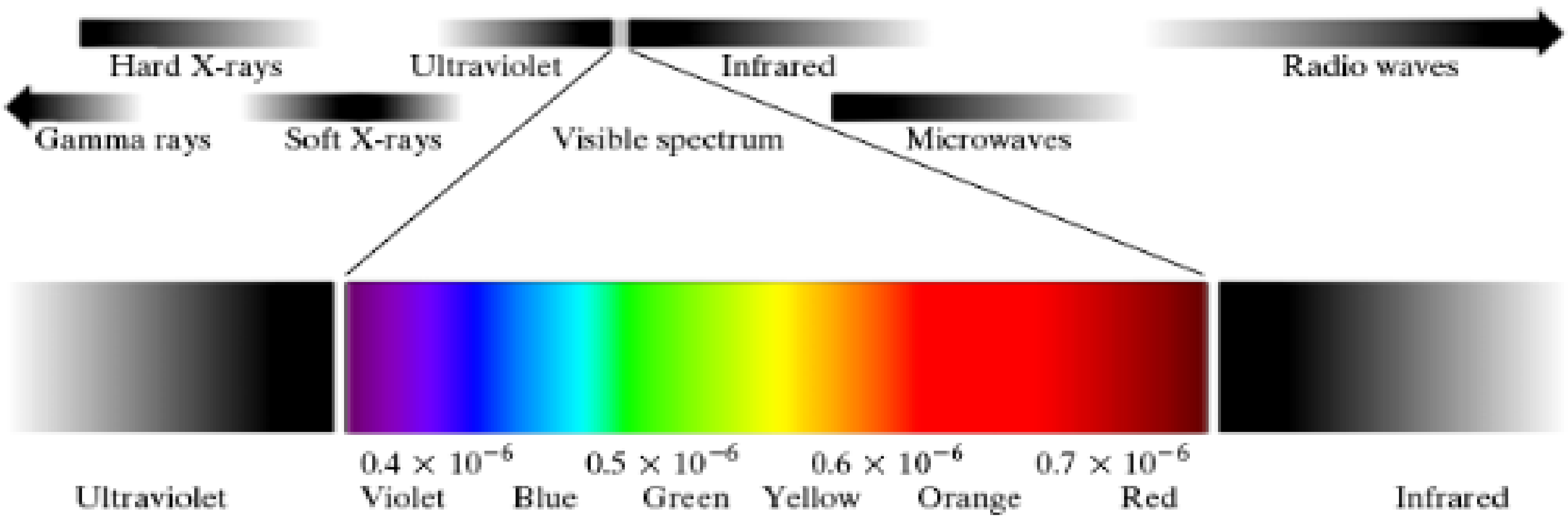
# Cont..

– Fundamental equations
  - Relation between wavelength (λ) and frequency (ν): $\lambda = {}^{c}/_{v}$
  - relation between energy(E) and frequency (ν): $E = hv$

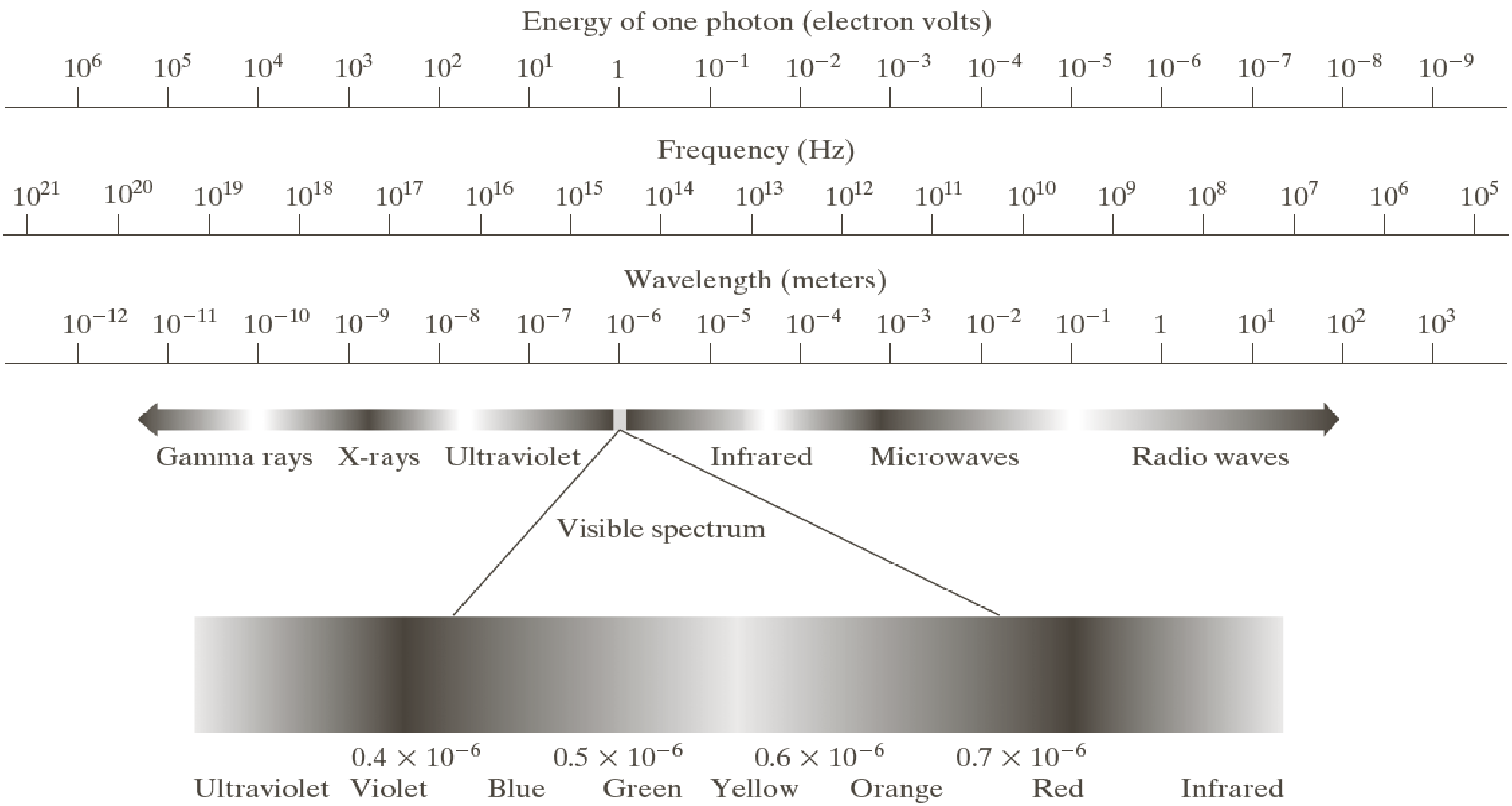


# Cont..

- The Electromagnetic Spectrum
  - we perceive only a small range of colours of the electromagnetic spectrum (~ 430nm –790nm)
  - objects are perceived by the light they reflect
    - achromatic light: all wavelengths are reflected equally
    - chromatic light: some wavelengths are reflected predominantly



# Light and EM Spectrum




$$c = \lambda \nu \qquad E = h\nu, \quad h: \text{ Planck's constant.}$$

***λ = wavelength, v = frequency, c = speed of light = 3 x 10$^8$ m/sec, E = energy***

# Light and EM Spectrum

- The colors that humans perceive in an object are determined by the nature of the light reflected from the object.

e.g. green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelength

# Light and EM Spectrum

- Monochromatic light: void of color

**Intensity** is the only attribute, from black to white

Monochromatic images are referred to as **gray-scale** images

- Chromatic light bands: 0.43 to 0.79 um

The quality of a chromatic light source:

**Radiance**: total amount of energy

**Luminance (lm)**: the amount of energy an observer perceives from a light source

**Brightness**: a subjective descriptor of light perception that is impossible to measure. It embodies the achromatic notion of intensity and one of the key factors in describing color sensation.

# Some Typical Ranges of illumination

- **Illumination**

  **Lumen** — A unit of light flow or luminous flux

  **Lumen per square meter ($lm/m^2$)** — The metric unit of measure for illuminance of a surface

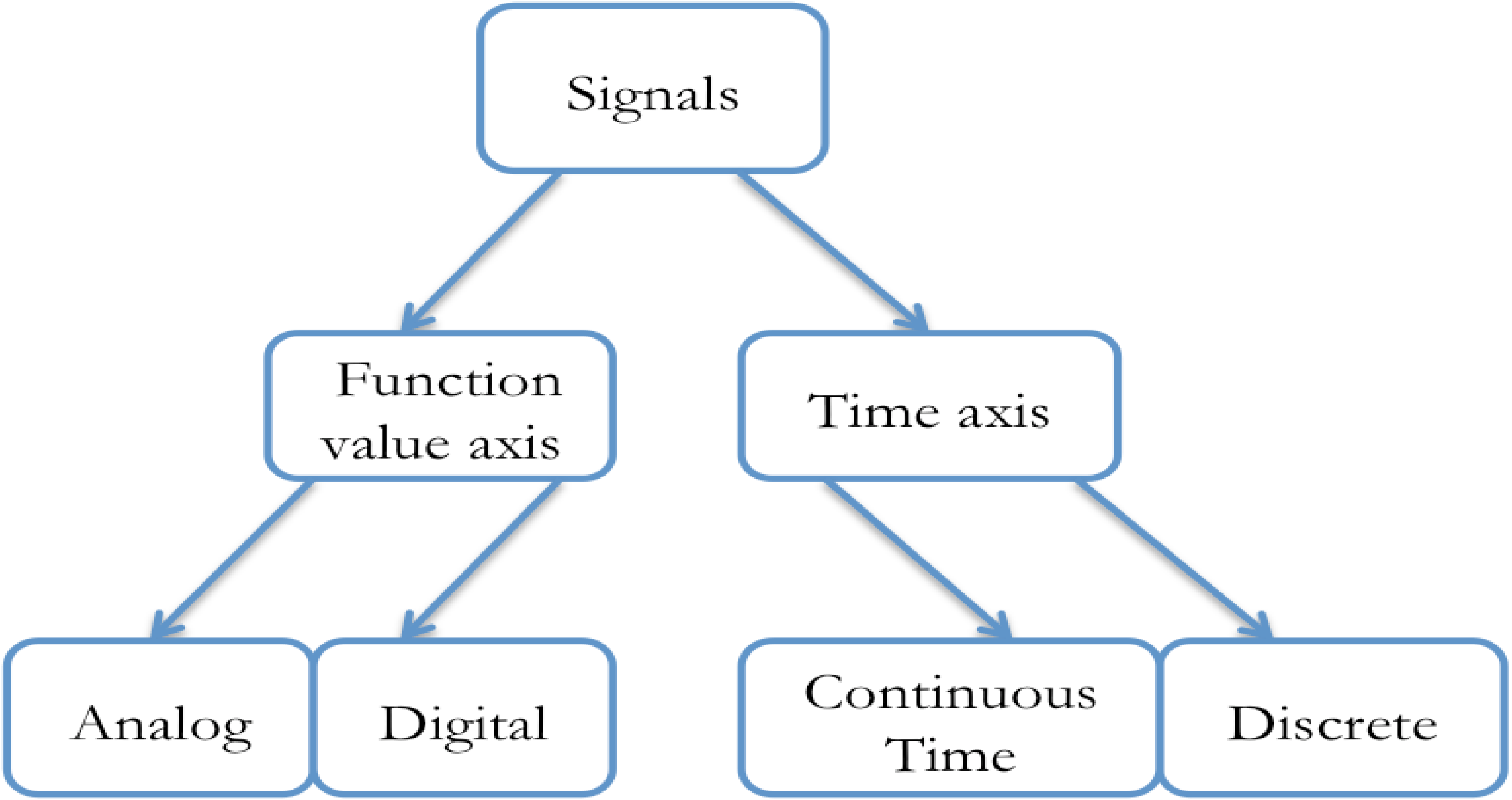
  – On a clear day, the sun may produce in excess of 90,000 $lm/m^2$ of illumination on the surface of the Earth

  – On a cloudy day, the sun may produce less than 10,000 $lm/m^2$ of illumination on the surface of the Earth

  – On a clear evening, the moon yields about 0.1 $lm/m^2$ of illumination

  – The typical illumination level in a commercial office is about 1000 $lm/m^2$

# Classification of Signals



# Classification of Signals

- **Analog Signals** (Continuous-Time Signals): Signals that are continuous in both the dependant and independent variable (e.g., amplitude and time). Most environmental signals are continuous-time signals.

- **Discrete Sequences** (Discrete-Time Signals): Signals that are continuous in the dependant variable (e.g., amplitude) but discrete in the independent variable (e.g., time). They are typically associated with sampling of continuous-time signals.

- **Digital Signals**: Signals that are discrete in both the dependant and independent variable (e.g., amplitude and time) are digital signals. These are created by quantizing and sampling continuous-time signals or as data signals (e.g., stock market price fluctuations).

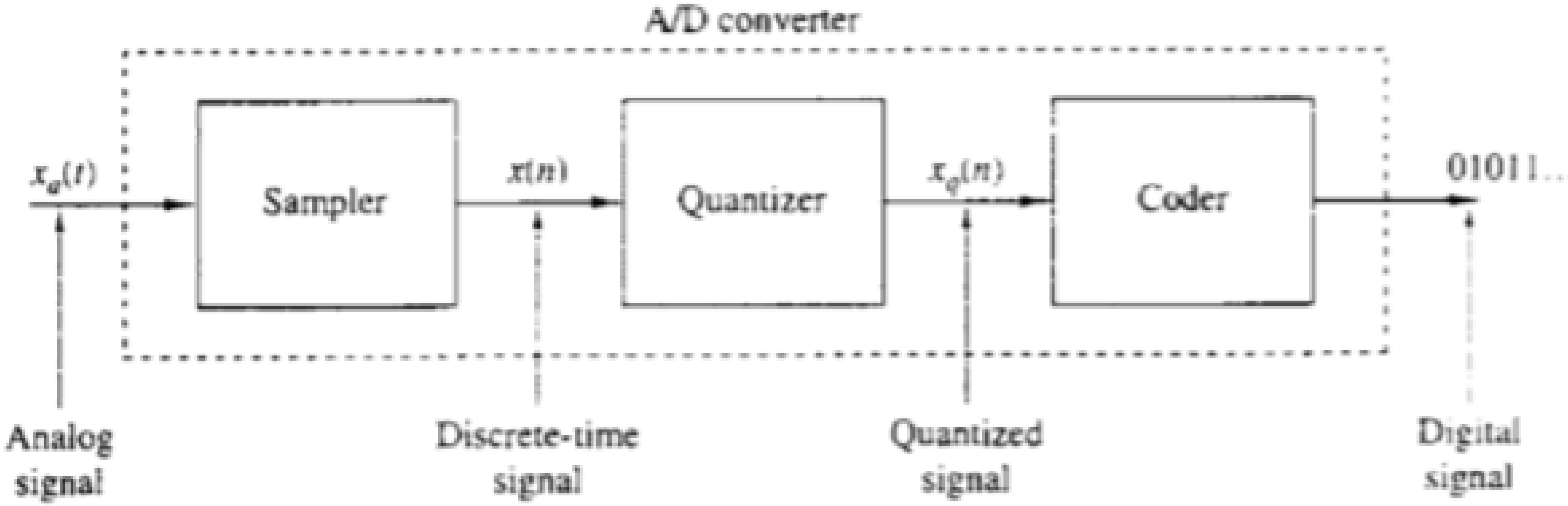
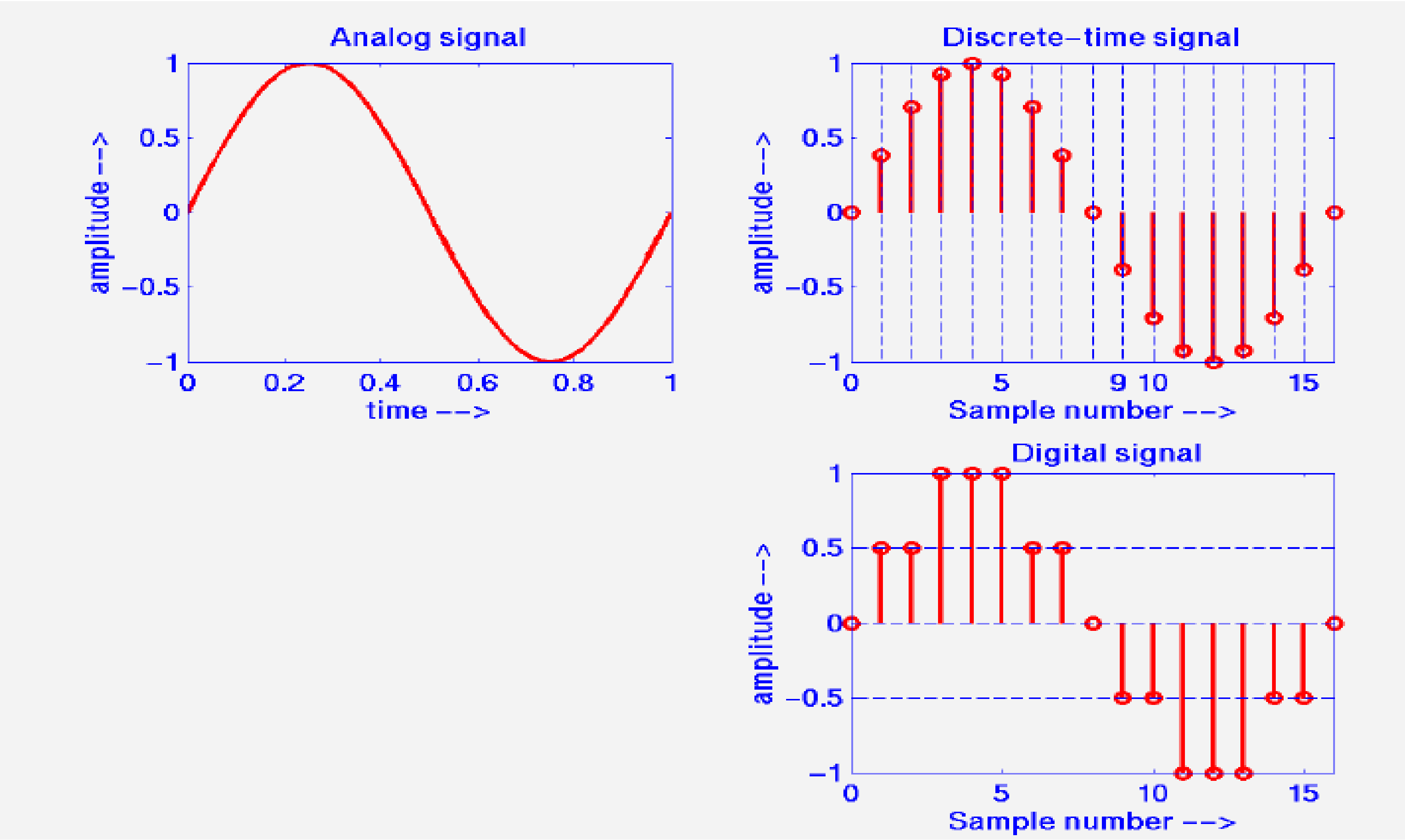
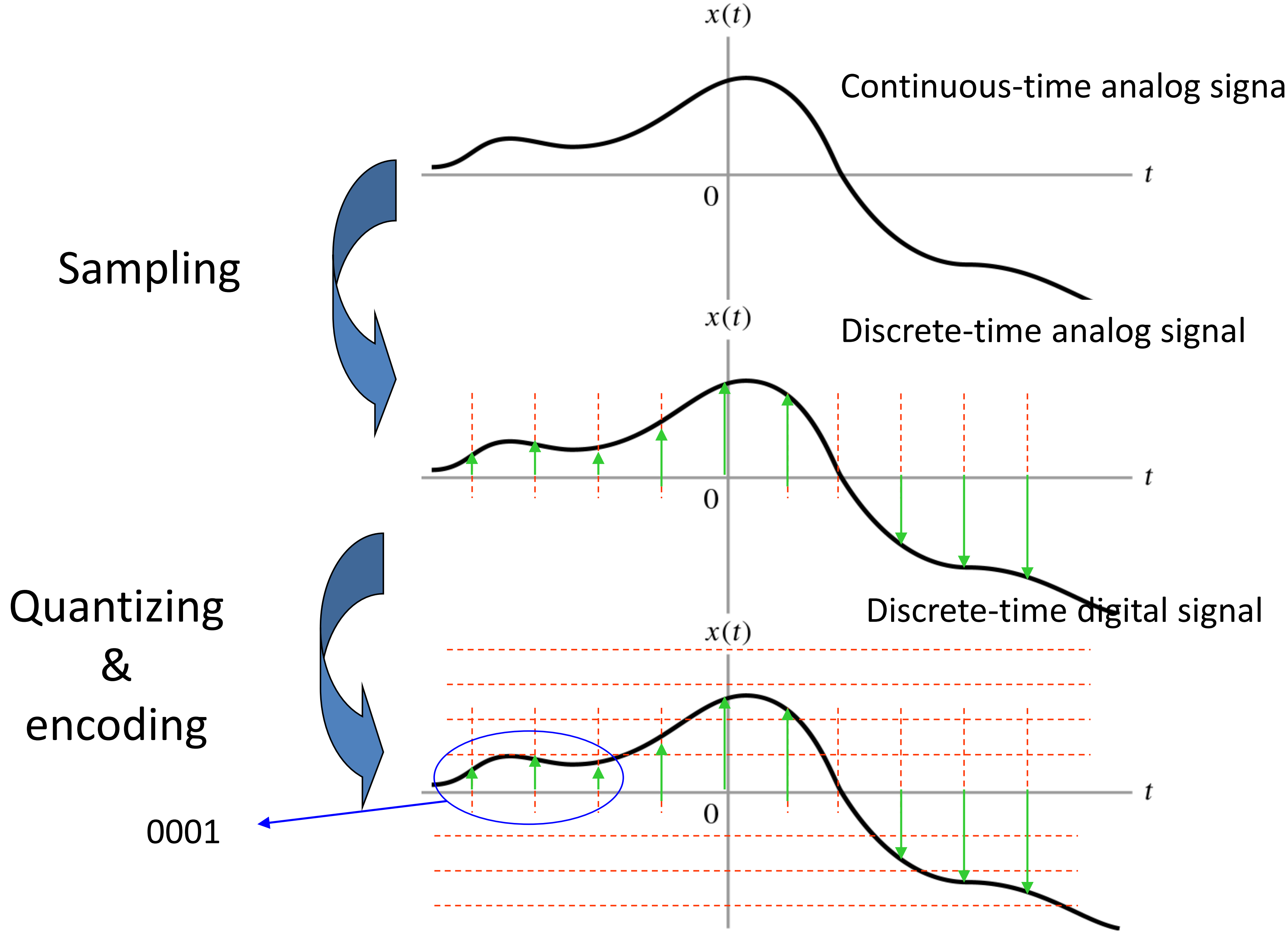



**Figure 1.14** Basic parts of an analog-to-digital (A/D) converter.

# Graphical Representation





Sampling
Quantizing
&
encoding
0001
x(t)
t
0
Continuous-time analog signal
Discrete-time analog signal
Discrete-time digital signal

# Sampling

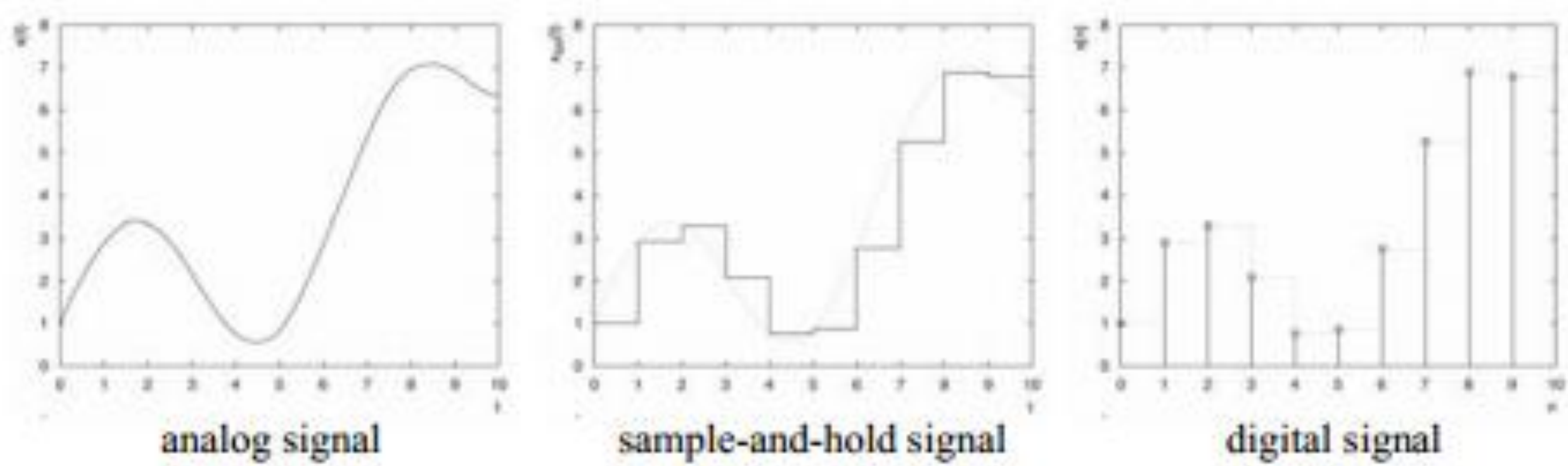
- Sampling period (sampling interval) – the time in seconds between samples
- Sampling frequency (sampling rate) – the number of samples per second, measured in Hertz (Hz)

$$f_s = \frac{1}{T_s}$$

where $f_s$ denote the sampling frequency, $T_s$ denote the sampling period. $\frac{f_s}{2} > F_{max} \rightarrow f_s > 2F_{max}$

- **Theorem:** If the highest frequency($F_{max}$) contained in an analog signal $x_a(t)$ is $F_{max} = B$ and the signal is sampled at a rate $f_s > 2F_{max} = 2B$ then $x_a(t)$ can be exactly recovered from its sample values.

# Sampling



# Sampling

- Digitization of the spatial coordinates, sample (x, y) at discrete values of (0, 0), (0, 1), ….

- f(x, y) is 2-D array

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

# Quantization

- Digitization of the light intensity function

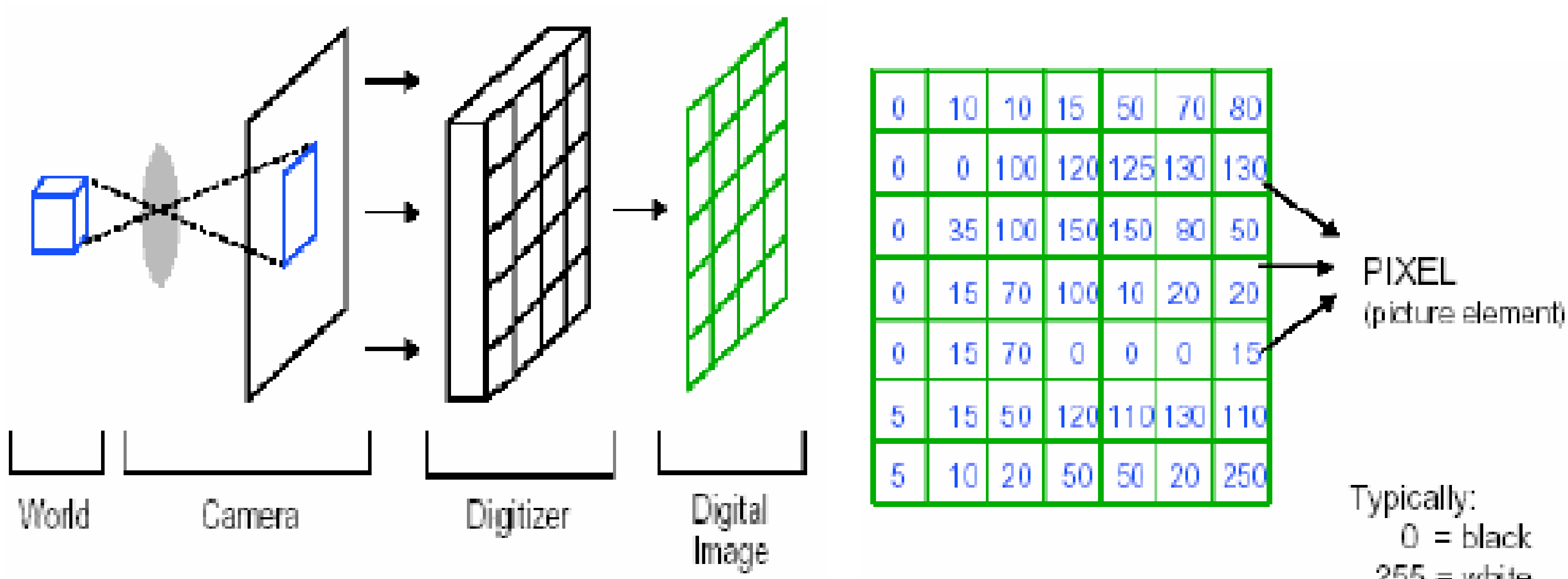
- Each *f(i,j)* is called a pixel

- The magnitude of *f(i,j)* is represented digitally with a fixed number of bits - quantization

# Need for Digitization

- Why image digitization is necessary
- What is meant by signal bandwidth
- Select the sampling frequency of a given signal
- Explain image reconstruction from sampled values

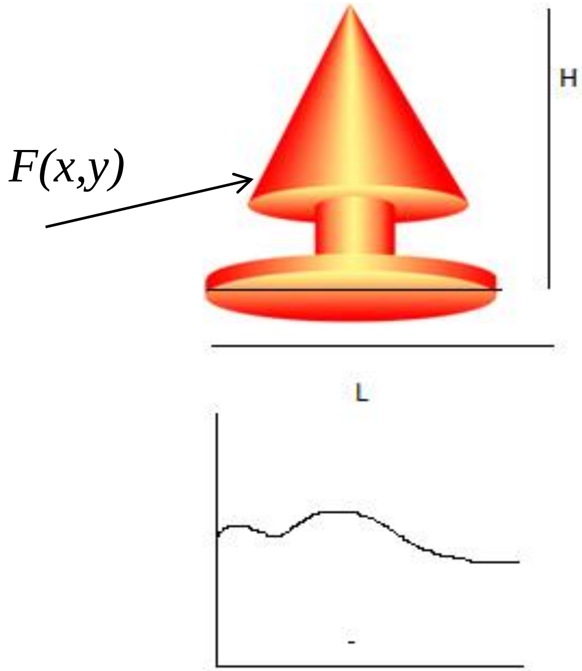
# Digital Image



# Image Digitization

- Why do we need Digitization
- What is Digitization
- How to digitize an image

# Image Digitization




$$f(x, y) = I(x, y)R(x, y)$$

$$0 \le x \le L$$

$$0 \le y \le H$$

Point f(x,y) can also vary from 0 to infinity but practically its not possible hence its considered to vary from $I_{min}$ to $I_{max}$

$$I_{min} \le f(x, y) \le I_{max}$$

f(x,y) = point in the image
R(x,y) = reflectance at the point (x,y) and varies from 0-1
I(x,y) = intensity at the (x,y) and varies from 0- Infinity

# Image Digitization – Sampling & Quantization

- The most basic requirement for computer processing of images is availability of image in digital form.
- Digitization of an image is done by sampling and quantization.
- The digitized image can then be processed by computer.
- To display, the processed digital image is converted back to analog form using DAC.
- The common method of image sampling is to scan the image row by row and sample each row.

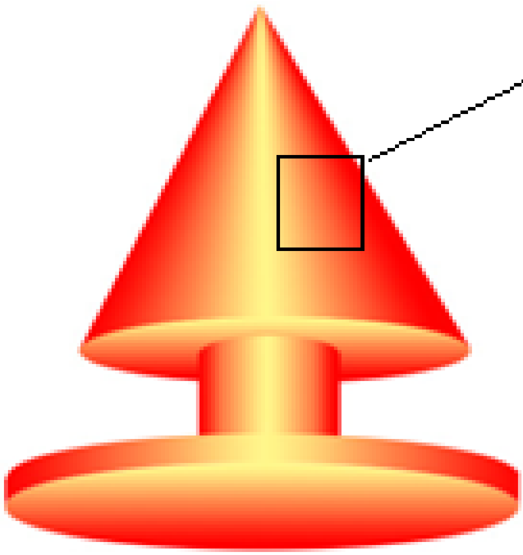
# Theory of Real numbers

- Between any 2 points, there are infinite no. of points
- An image should be represented by infinite no. of points
- Each such image may contain one of the infinitely many possible intensity/ colour values needing infinite no. of bits
- Practically not possible in any Digital computer
- Naturally a way out is to be found….!!
- Take discrete values – Sampling and Quanitzation
- Image is represented in a matrix (M x N) form as , where each element assumes a finite value

$$f(x,y)=\begin{bmatrix} f(0,0)\ f(0,1)\ f(0,2)\ldots\ldots\ldots f(0,N-1) \\ f(1,0)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots f(1,N-1) \\ \\ \\ \\ f(M-1,0)\ldots\ldots\ldots\ldots\ldots\ldots f(M-1,N-1) \end{bmatrix}$$

Each point represents intensity at a point
We have to consider finite values for digitization

# Sampling and Quantization



| | | | |
|---|---|---|---|
| 189 | 184 | 186 | 190 |
| 183 | 185 | 190 | 165 |
| 182 | 179 | 187 | 197 |
| 180 | 175 | 186 | 180 |

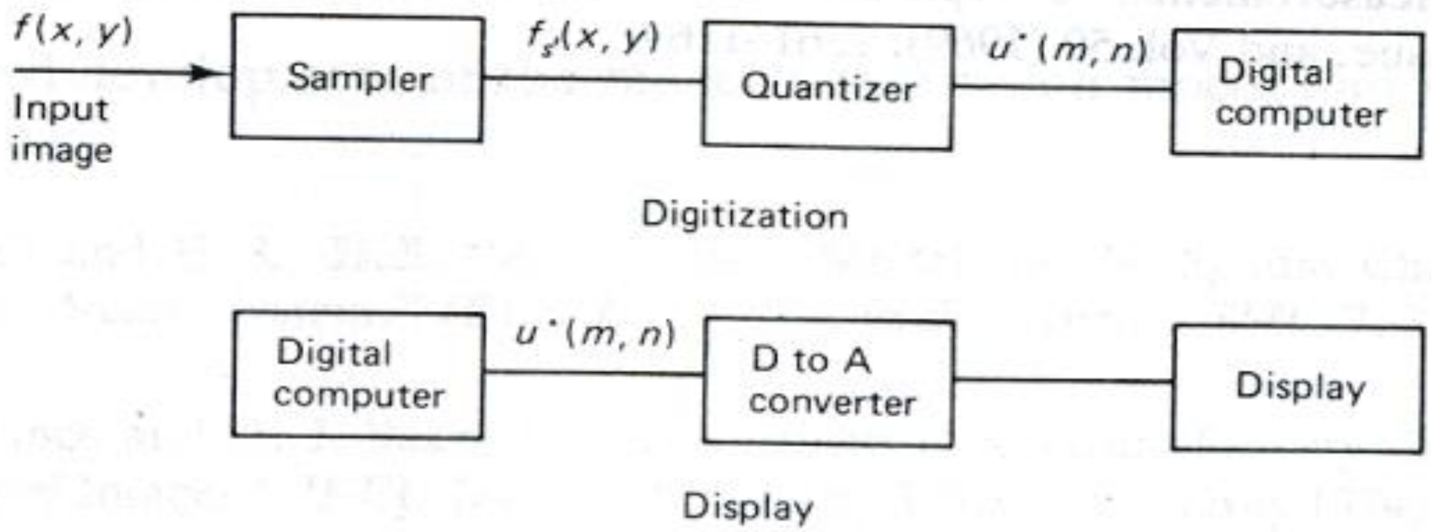
Image is represented by a matrix and each element is an integer value

Representation of an image as a Matrix is called Sampling

Each Matrix element represented by a finite values is called Quantization

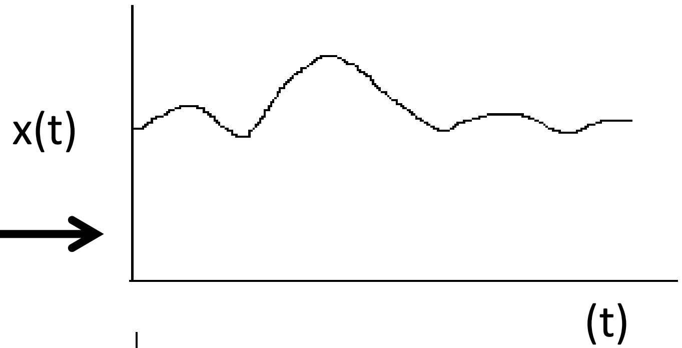
# Image Digitization – Sampling & Quantization

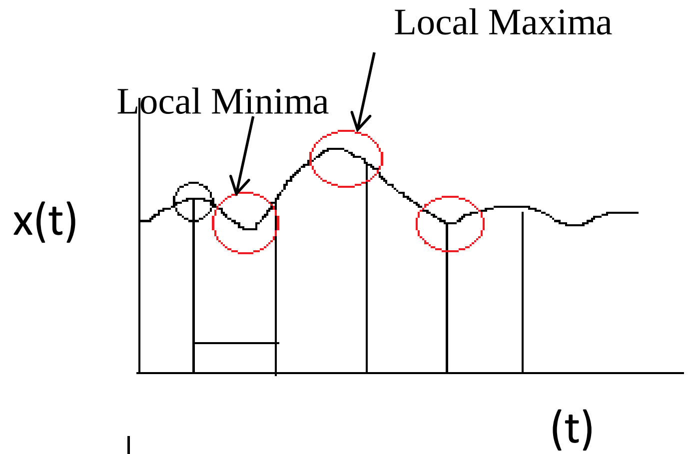
The figure explains the digitization of an image and reconverting back to analog form:

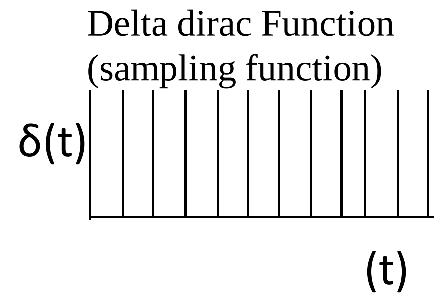


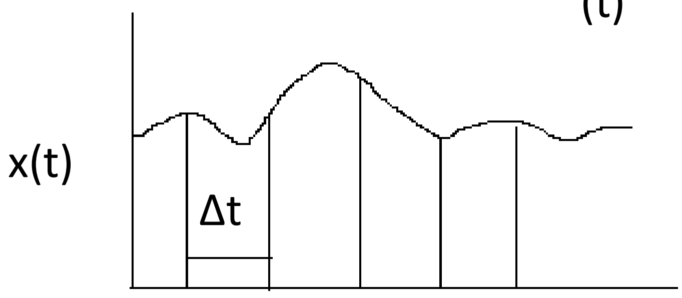

Sampling, quantization and display of images.

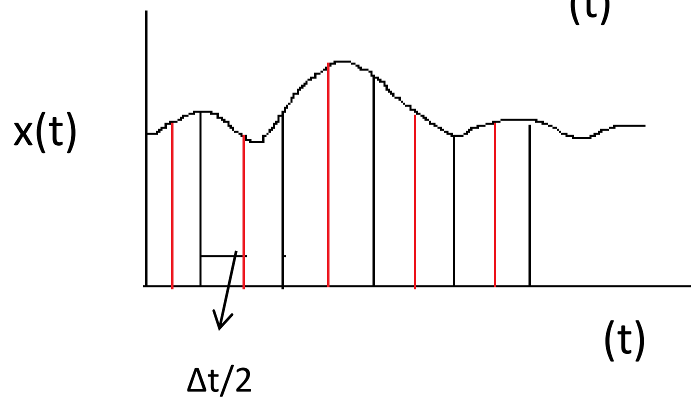
# Sampling



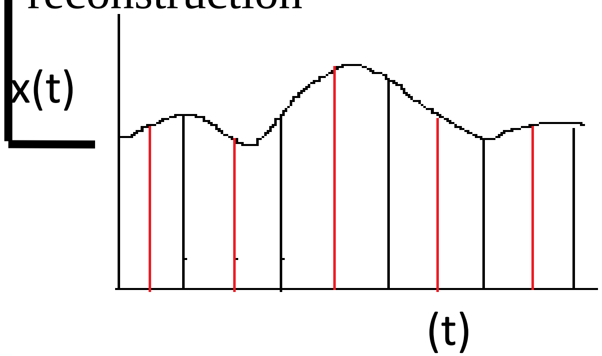




$$comb(t, \Delta t) = \sum_{m=-\infty}^{\infty} \partial(t - m\Delta t)$$

$$\partial(t) = \begin{cases} 1; t = 0 \\ 0; all\ other\ values\ of\ t \end{cases}$$

reconstruction




$$Xs(t) = X(t).comb(t, \Delta t)$$
$$= \sum_{m=-\infty}^{\infty} X(m\Delta t)\partial(t - m\Delta t)$$

# Mathematics Behind Sampling

$$Given\ discrete\ \ signal\ \ x(n),\ Signal\ in\ frequency\ domain\ is\ given\ by$$

$$F(x(n)) = X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\Pi}{N}\right)nk}$$

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\Pi}{N}\right)nk}$$

# Convolution

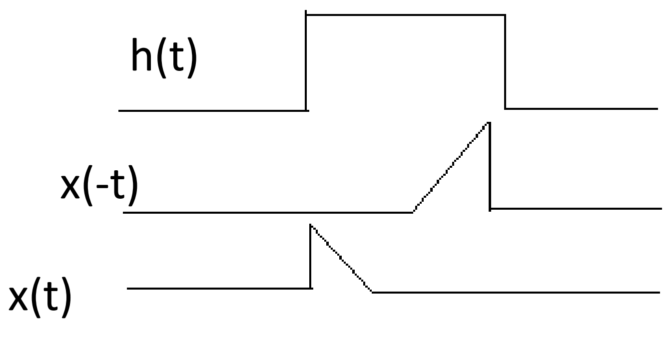
$$We\ know\ that$$

$$Xs(t) = X(t).comb(t, \Delta t)$$

$$given\ 2\ signals\ x(t)\ and\ h(t)$$

$$Convolution\ is\ given\ as$$

$$x(t) * h(t) = \int_{-\infty}^{\infty} h(T)x(t-T)dT$$




Convolution in time domain in multiplication in frequency domain

$$Time\ \text{domain}\ Convolution\ is\ given\ as$$

$$x(t) * h(t) = \int_{-\infty}^{\infty} h(T)x(t-T)dT$$

$$Fourier\ \text{domain convolution is given as}$$

$$\text{F}(\text{h}(\text{t}) * \text{x}(\text{t})) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} h(T)x(t-T)dT\right]e^{-jwt}dt$$

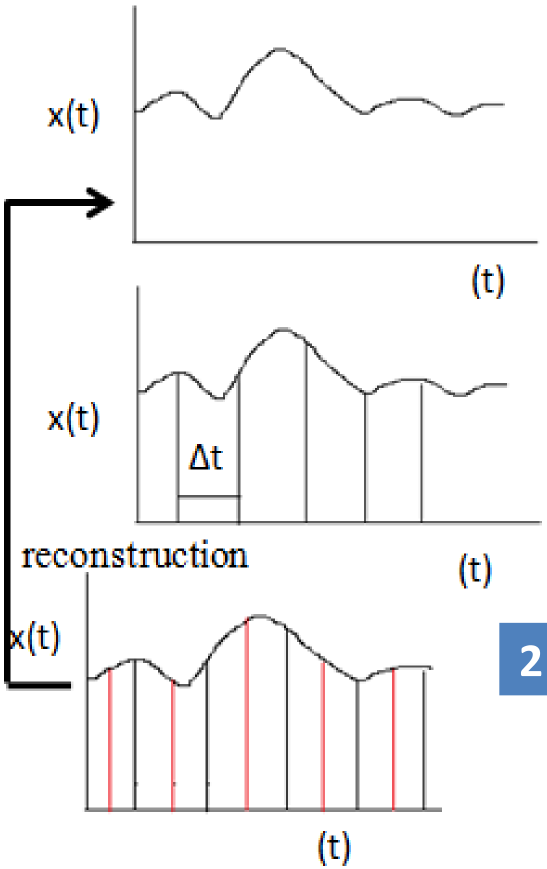
$$= \int_{-\infty}^{\infty} h(T)[\int_{-\infty}^{\infty} x(t-T)e^{-jw(t-T)}dt]e^{-jwT}dT$$

$$= \int_{-\infty}^{\infty} h(T)X(w)e^{-jwT}dT$$

$$= X(w)\int_{-\infty}^{\infty} h(T)e^{-jwT}dT$$

$$= X(w)H(w)$$

# Convolution – Concept




Convolution in time domain in multiplication in frequency domain

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | h(n) |

| | | | |
|---|---|---|---|
| 2 | 3 | 2 | x(n) or x(-n) |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 2 | 0 | 0 | 2 | 3 | 2 | 0 | 0 | 2 | 3 | 2 |

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m)$$

# Discrete-Time Convolution

* LTI system – Linear Time-Invariant

convolution sum

$$y[n] = \sum_{i=-\infty}^{\infty} x[i]\,h[n-i]$$

Example: $x[n] = \{\underline{4}, 1, 2, 5\}$ ↑ $n=0$

$h[n] = \{1, \underline{2}, -1\}$

Find $y[n] = x[n] * h[n]$ ← convolution operator

$y[n] = \{4, \underline{9}, 0, 8, 8, -5\}$

$$\begin{array}{rrrrrrl}
 & 1 & \underline{2} & -1 & & & \\
\times & \underline{4} & 1 & 2 & 5 & & \\
\hline
4 & 8 & -4 & & & & \leftarrow x[0]\cdot h[n] \quad i=0 \\
 & 1 & 2 & -1 & & & \leftarrow x[1]\cdot h[n-1],\ i=1 \\
 & & 2 & 4 & -2 & & \\
+ & & & 5 & 10 & -5 & \\
\hline
4 & \underline{9} & 0 & 8 & 8 & -5 &
\end{array}$$

6:17 / 6:30

# Detailed Solution

$h[n] = \{1, \underline{2}, 0, -3\}$

(d) $x_4[n] = \{2, 1, \underline{-1}, -2, -3\}$

$$
\begin{array}{c|cccccccc|l}
 & & & & \overset{n=0}{\downarrow} & & & & & \\
 & & & 1 & 2 & 0 & -3 & & & \\
\times & & 2 & 1 & -1 & -2 & -3 & & & \\
\hline
 & & & -1 & -2 & 0 & 3 & & & x[0]h[n] \\
 & & 1 & 2 & 0 & -3 & & & & x[-1]h[n+1] \\
 & 2 & 4 & 0 & -6 & & & & & x[-2]h[n+2] \\
 & & & & -2 & -4 & 0 & 6 & & x[1]h[n-1] \\
+ & & & & & -3 & -6 & 0 & 9 & +\,x[2]h[n-2] \\
\hline
 & 2 & 5 & 1 & \underline{-10} & -10 & -3 & 6 & 9 & \sum_{k=-2}^{2} x[k]h[n-k]
\end{array}
$$

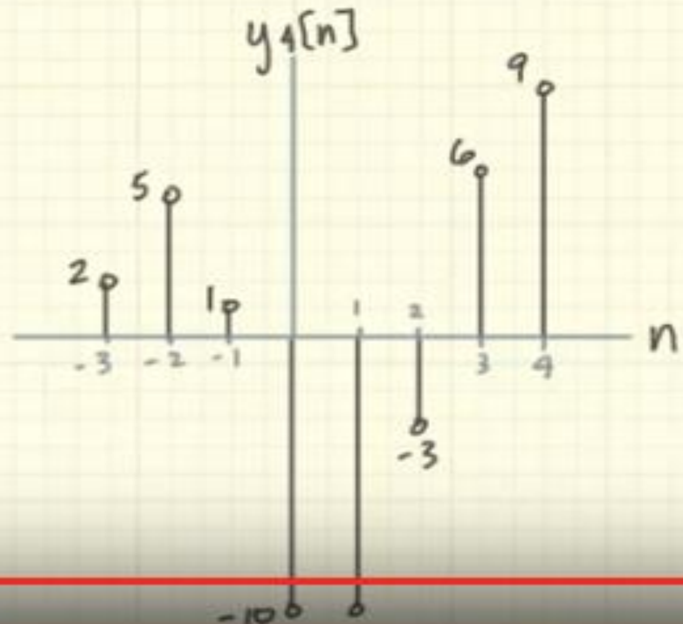
ROSE-HULMAN INSTITUTE OF TECHNOLOGY

# Detailed Solution

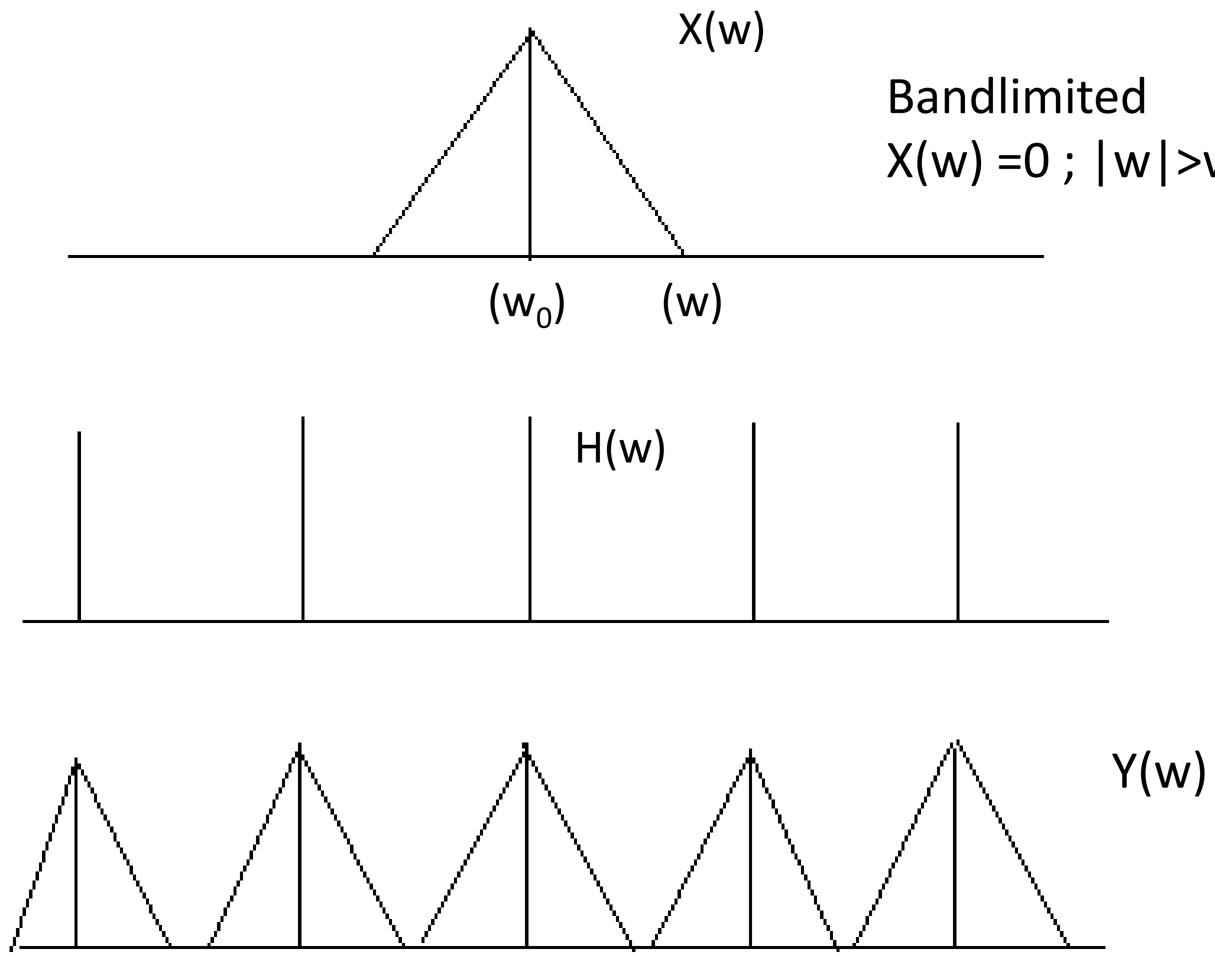
$$h[n] = \{1, \underline{2}, 0, -3\}$$

(d) $x_4[n] = \{2, 1, \underline{-1}, -2, -3\}$

$$y_4[n] = \{2, 5, 1, \underline{-10}, -10, -3, 6, 9\}$$



# Convolution – Concept

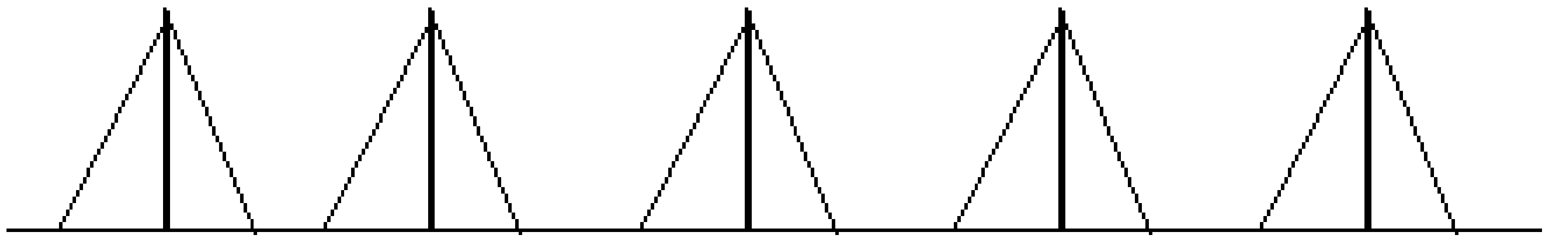



Bandlimited

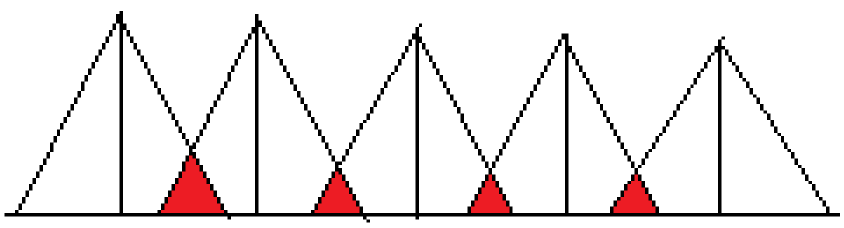
$X(w) = 0 \; ; \; |w| > w_0$

$f_s > 2w_o$

# Aliasing and Anti-aliasing

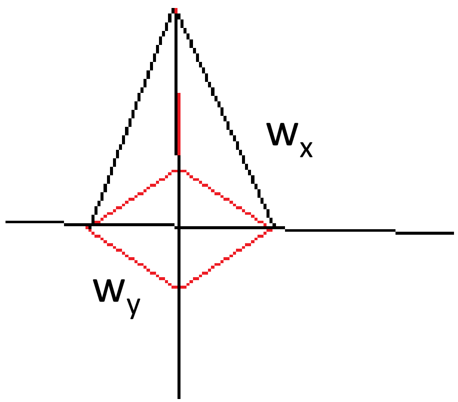


$f_s < 2w_o$

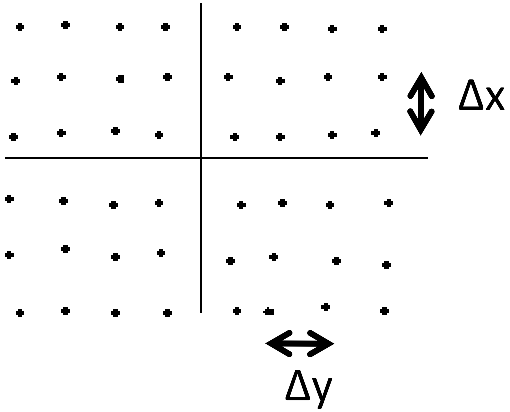


If sampling rate is not large enough (not larger than 2B Hz), then interference among adjacent bands will occur, and this results in the phenomenon of aliasing.

# 2D Sampling



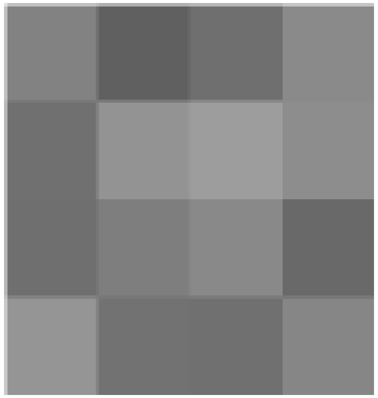

Bandlimited
$F(w_x,w_y) > 0$ ;
$|w_x| > w_0$
$|w_y| > w_0$

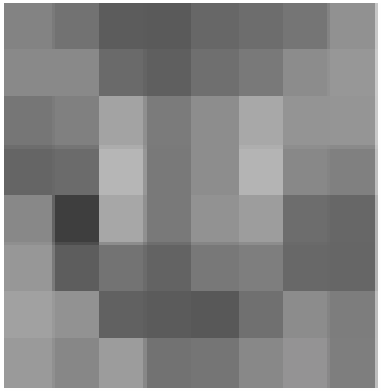


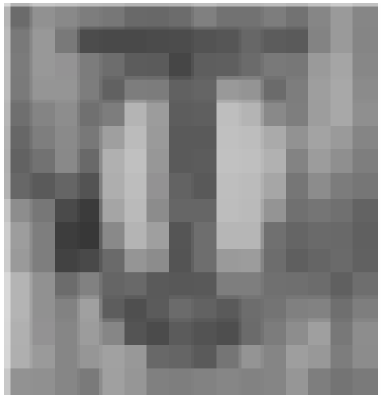

$$Fs(x,y) = F(x,y).comb(x,y;\Delta x,\Delta y)$$

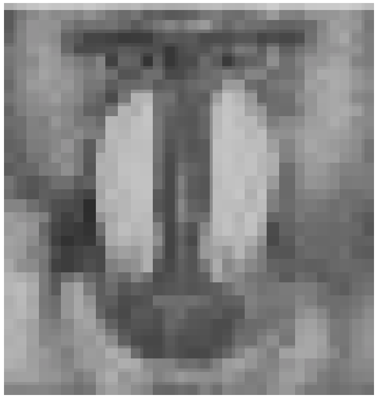
$$= \sum \sum_{m,n=-\infty}^{\infty} f(m\Delta x, n\Delta y)\partial(t - m\Delta x, y - n\Delta y)$$

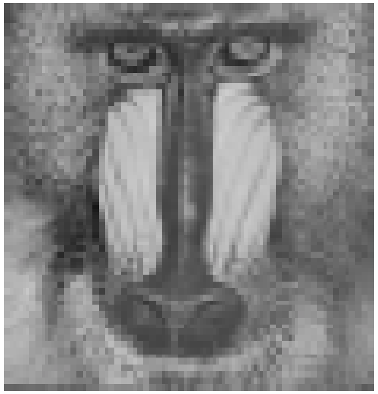
# Image Sampling

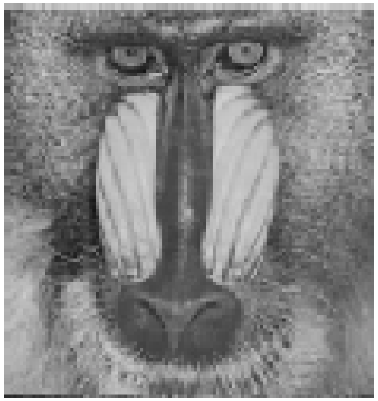

4x4=16


8x8=64

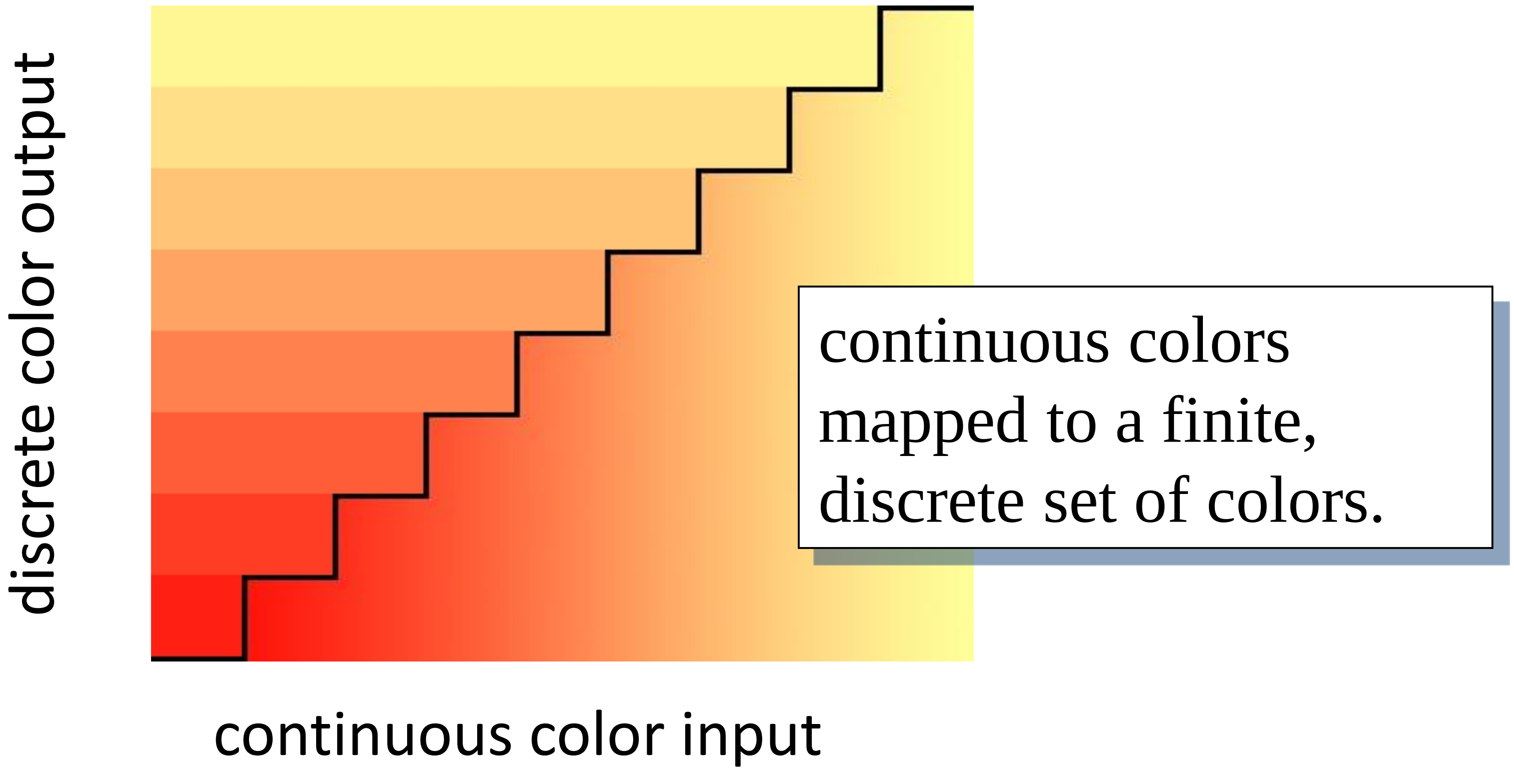

16x16=256


32x32=1024

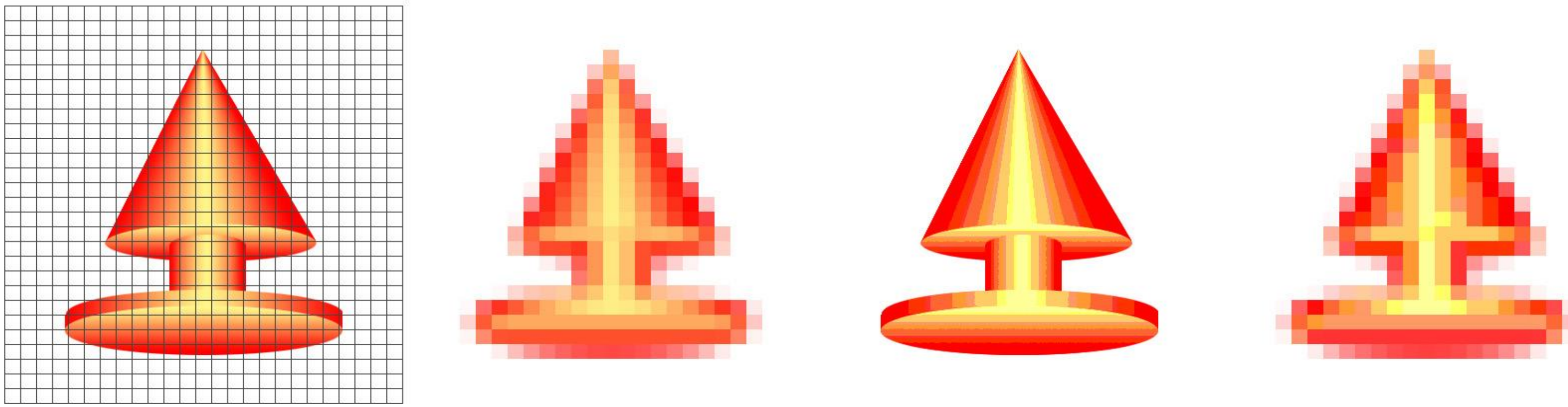

64x64=4096


128x128=16384

# Digital Quantization



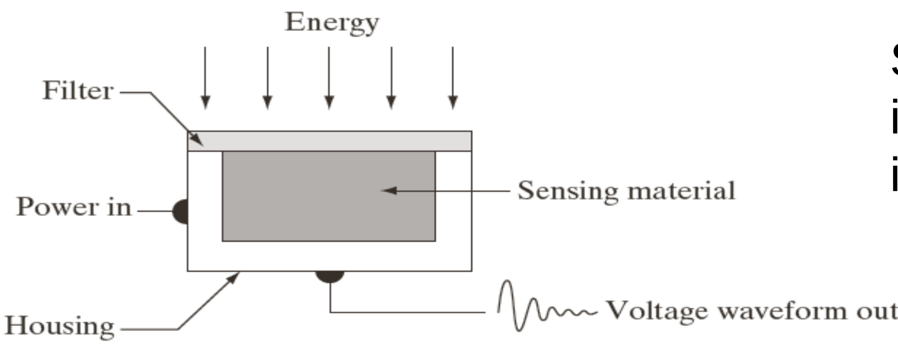
# Sampling and Quantization

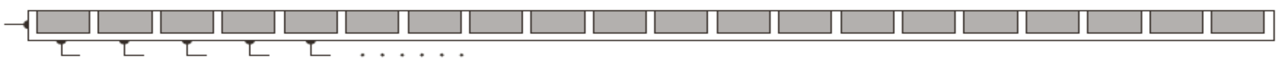


real image | sampled | quantized | sampled & quantized

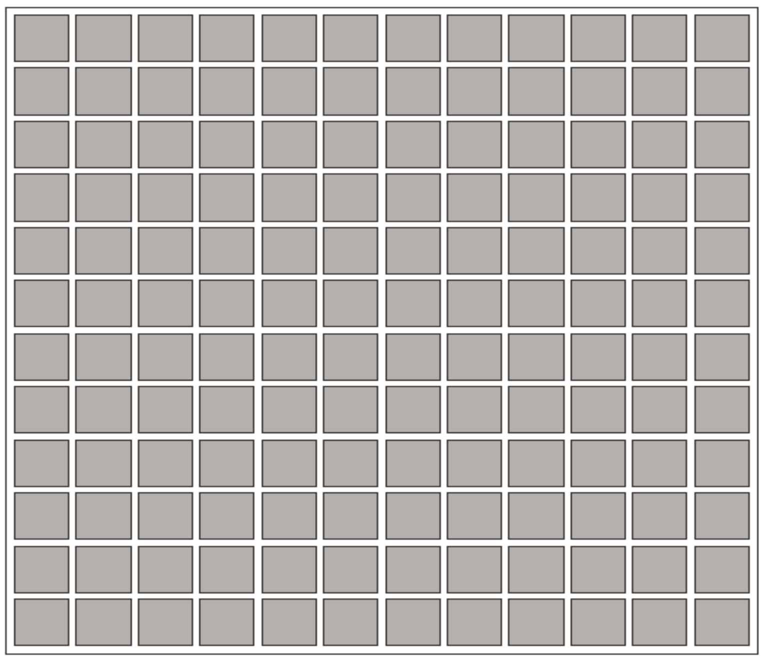
# Image Sensor

Sensors transform the incoming energy into voltage and the output of the sensor is digitized.



Transform illumination energy into digital images



a
b
c

**FIGURE 2.12**
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

# Image Acquisition Using a Single Sensor

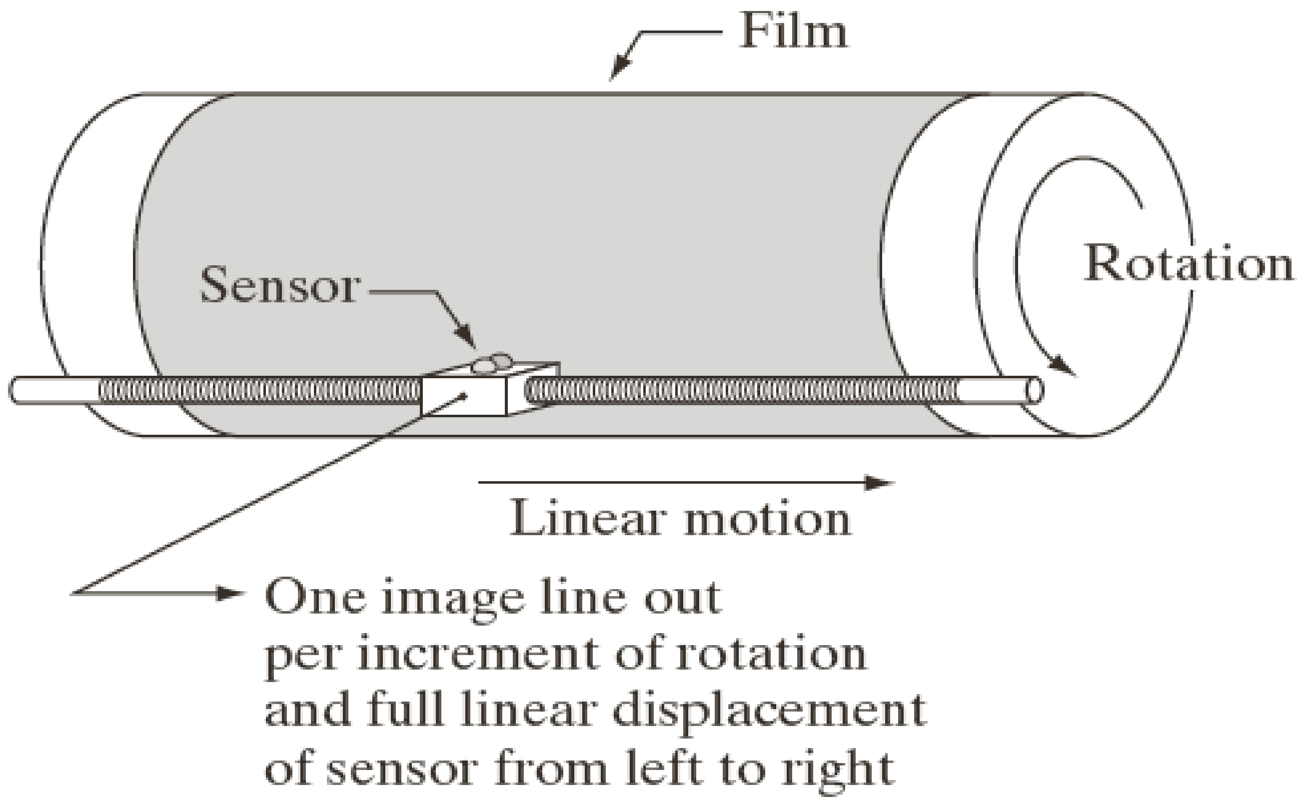
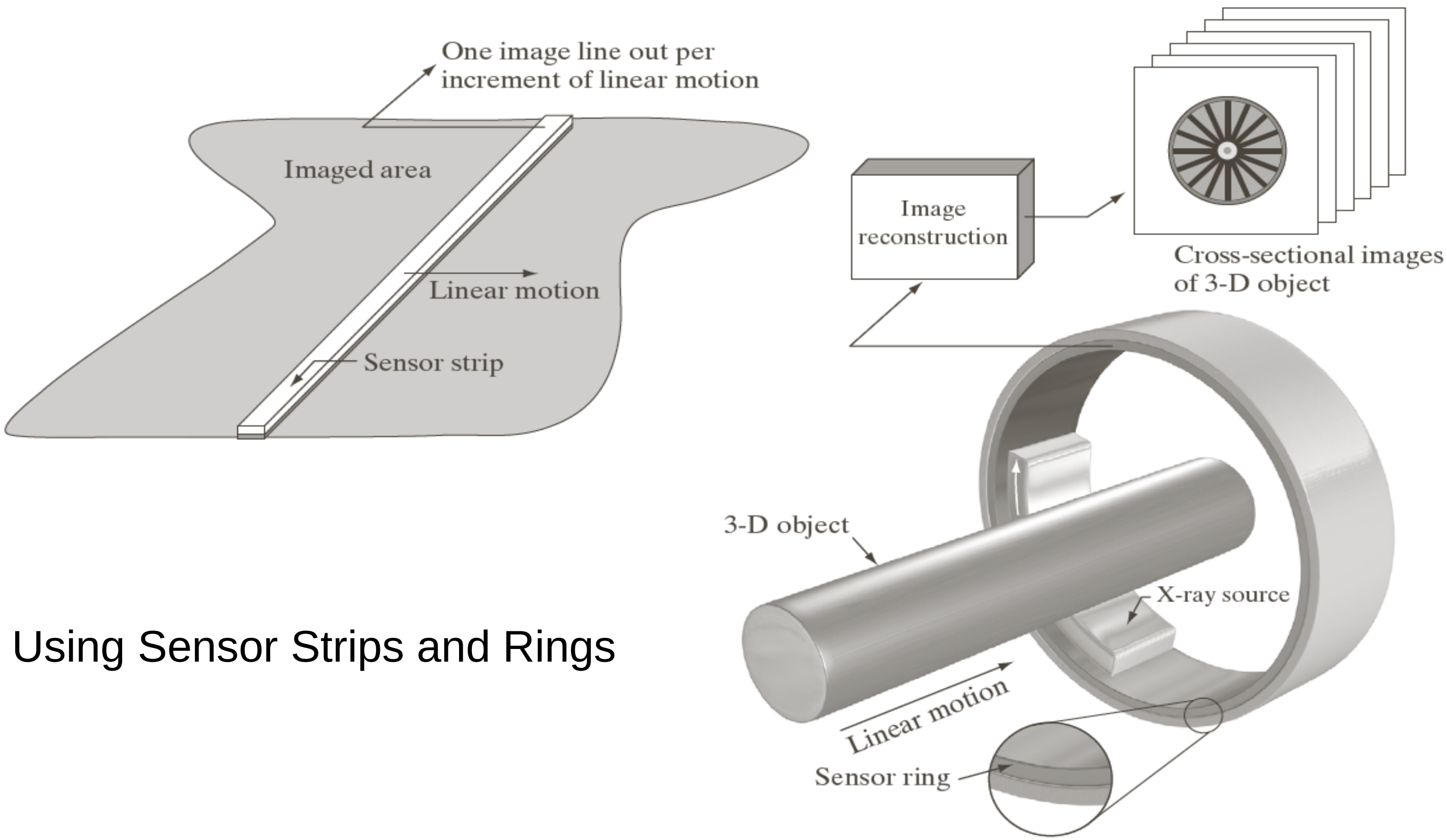



**FIGURE 2.13** Combining a single sensor with motion to generate a 2-D image.

# Image Acquisition Using Sensor Strips




Using Sensor Strips and Rings

a b

**FIGURE 2.14** (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

# Image Acquisition

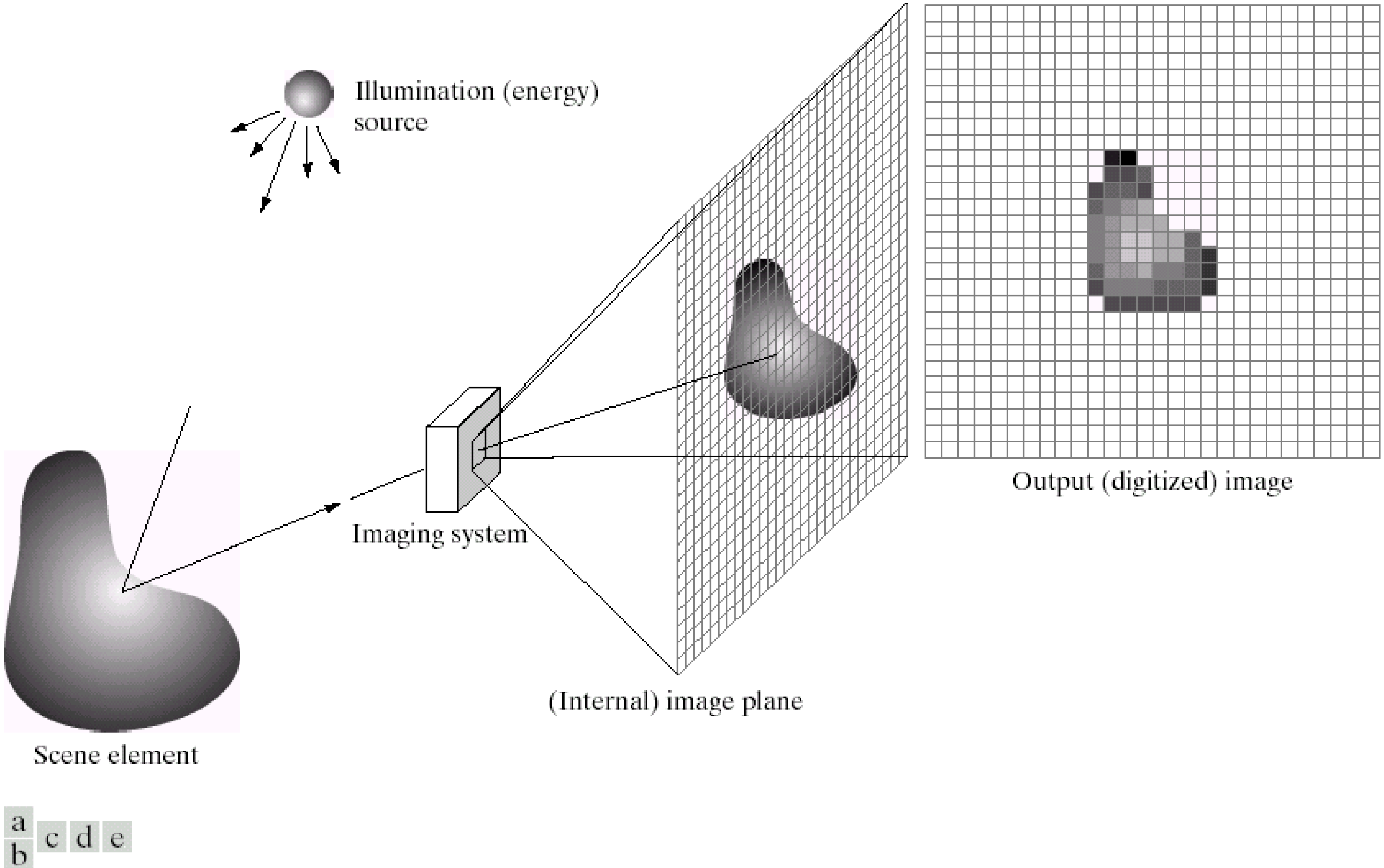



**FIGURE 2.15** An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

# Image Acquisition

- Images are typically generated by *illuminating* a *scene* and absorbing the energy reflected by the objects in that scene

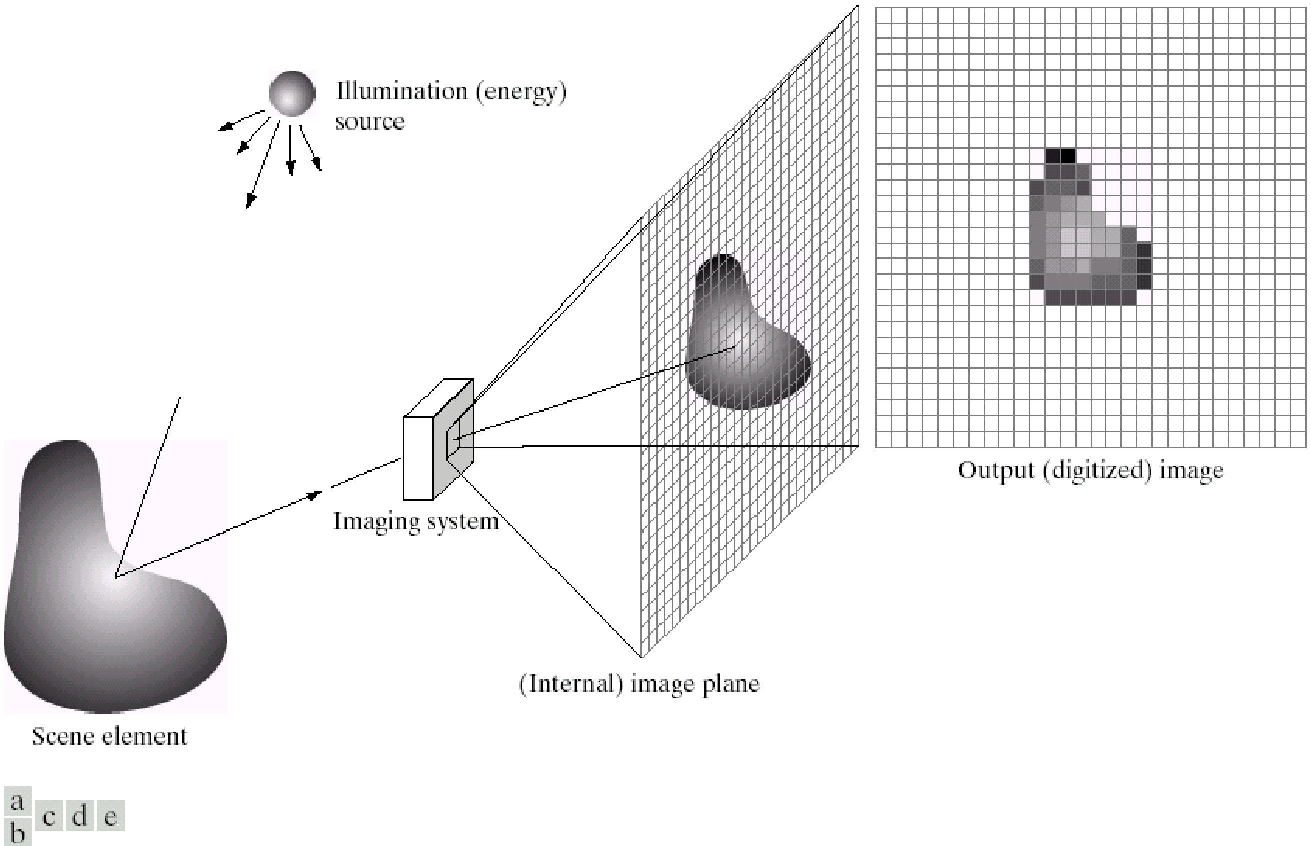



**FIGURE 2.15** An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

- Typical notions of illumination and scene can be way off:
  - X-rays of a skeleton
  - Ultrasound of an unborn baby
  - Electro-microscopic images of molecules

# A simple image formation model

- Image as two-dimensional(2D) light-intensity function
  - $f(x,y) = i(x,y)\,r(x,y)$ $\quad 0 \le f(x,y) < \infty$
    - the amount of source illumination incident on the scene being viewed
    - the amount of illumination reflected by the objects in the scene.
  - $i(x,y)$ – illumination component at point (x, y)
  - $r(x,y)$ – reflectance component at point (x, y)
  - $f(x,y)$- intensity at point $(x, y)$

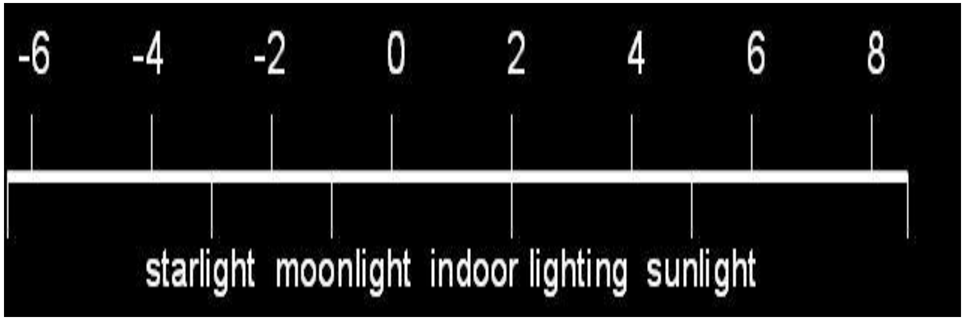
where $0 \le i(x,y) < \infty$ and $0 \le r(x,y) \le 1$

# Image Formation and Image Sampling

- Image Formation Model
  - illumination i(x,y) from a source
  - reflectivity r(x,y) = reflection / absorption in the scene f(x,y) = i(x,y) r(x,y)
- Image Sampling
  - digital image can be seen as a 2D function f(x,y)
    - x and y are the spatial coordinates
    - f(x,y) is the grey level or intensity at position (x,y)
  - a digital image must be sampled (digitized)
    - in space (x,y): image sampling
    - in amplitude f(x,y): grey-level quantization

# A simple image model

- ***l*(*x,y*) - illumination range**
  - 0.1 $lm/m^2$(full moon)
  - 1000 $lm/m^2$(office)
  - 10'000 $lm/m^2$(cloudy day)
  - 90'000 $lm/m^2$(sunny day)




- ***r*(*x,y*) – typical reflectance indixes**
  - black velvet (0.01)
  - stainless steel (0.65)
  - white paint (0.80)
  - silver plate (0.90)
  - snow (0.93)

# Image Sampling and Quantization

- A digital sensor can only measure a limited number of **samples** at a **discrete** set of energy levels
- Quantization is the process of converting a continuous **analogue** signal into a digital representation of this signal
- Remember that a digital image is always only an **approximation**of a real world scene

# Image Sampling and Quantization

- conversion of continuous input signal to a digital form

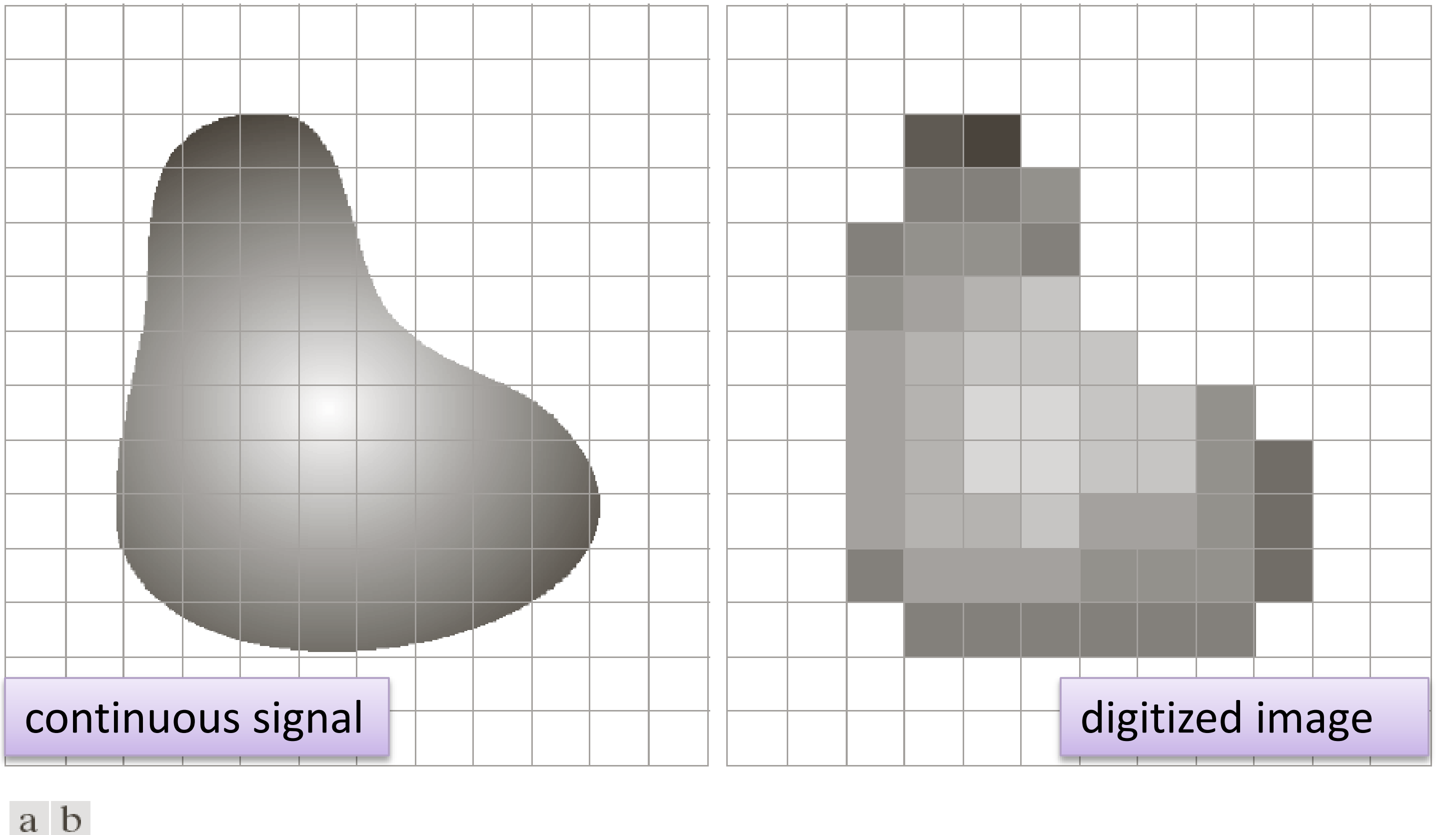



**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

# Cont..

- conversion of continuous input signal to a digital form

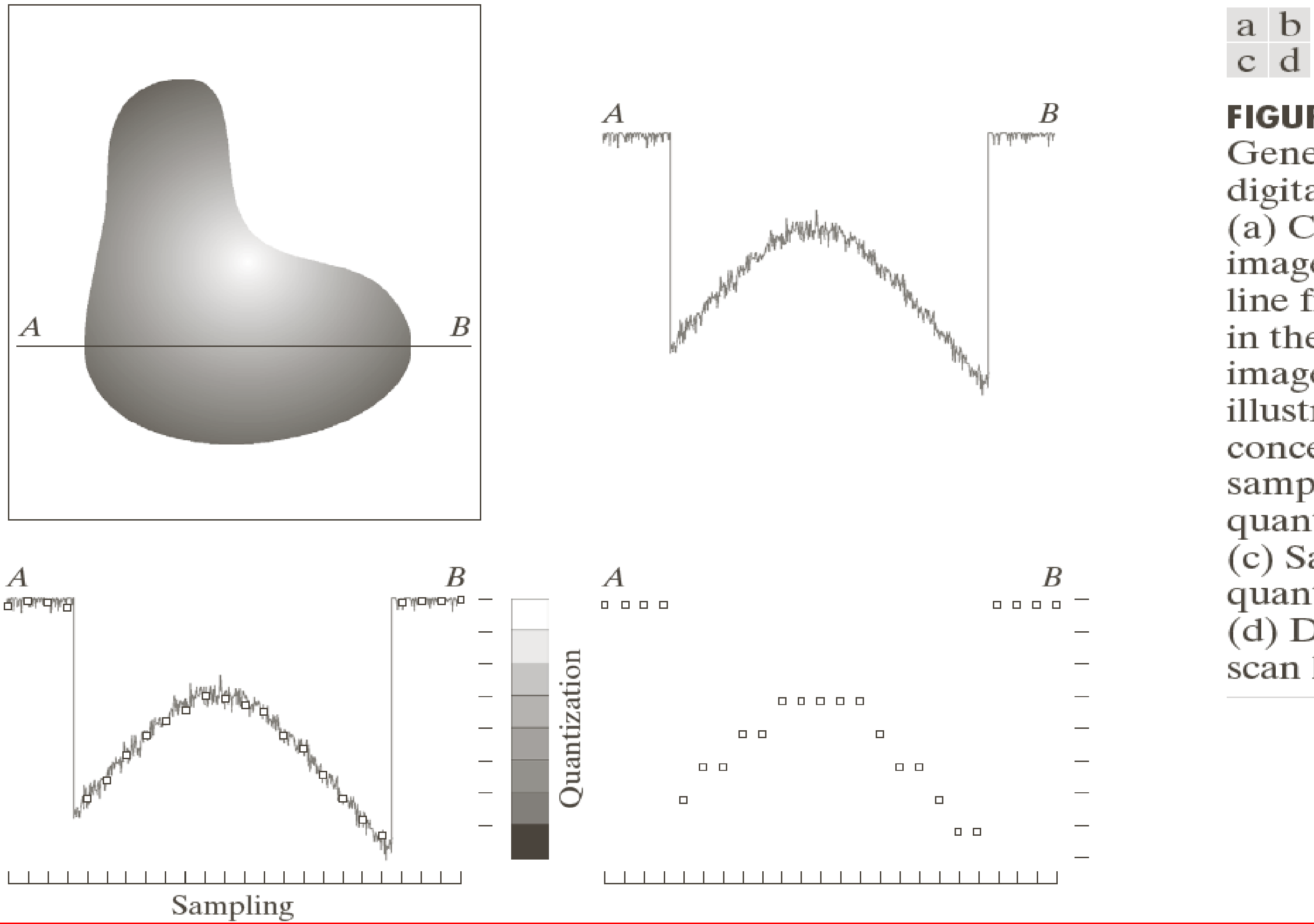



**FIGURE 2.16** Generating a digital image. (a) Continuous image. (b) A scan line from $A$ to $B$ in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

# Cont..

- conversion of continuous input signal to a digital form
- sample f(x,y) in both coordinates(sampling)

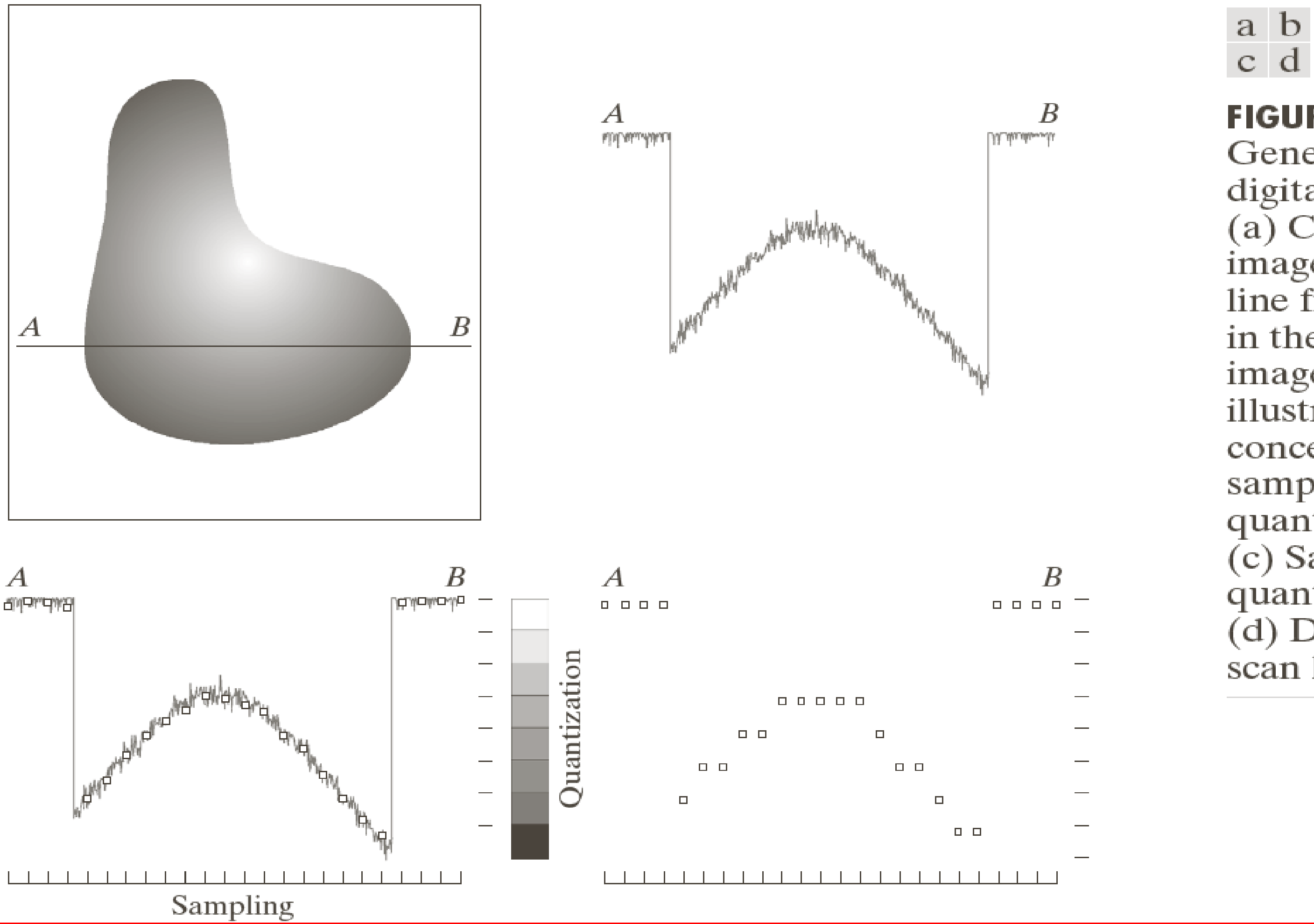



**FIGURE 2.16** Generating a digital image. (a) Continuous image. (b) A scan line from $A$ to $B$ in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

# Cont..

- conversion of continuous input signal to a digital form
- sample f(x,y) in both coordinates(sampling)

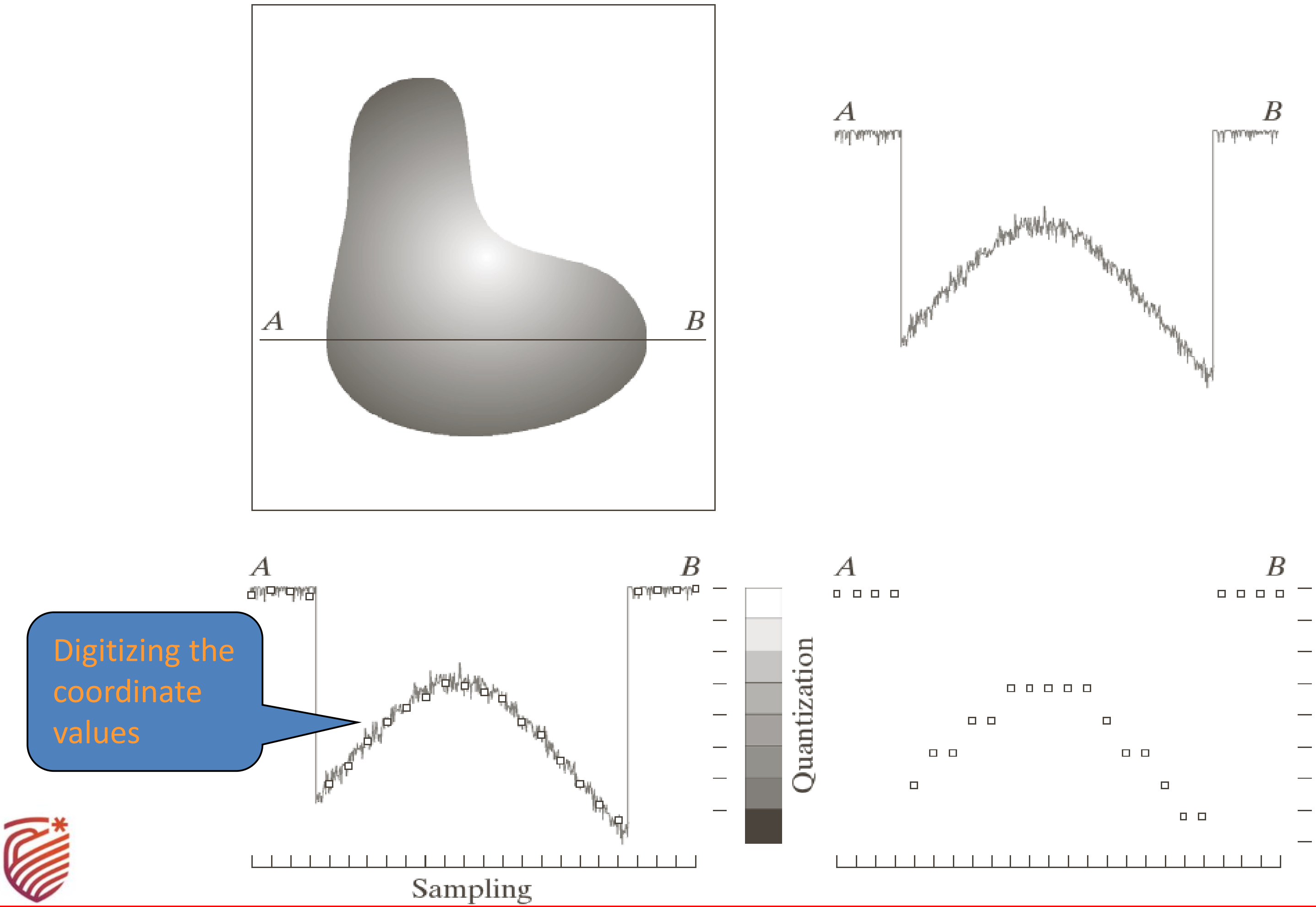



**FIGURE 2.16** Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

# Cont..

- conversion of continuous input signal to a digital form
- sample f(x,y) in both coordinates(sampling)
- sample f(x,y) in amplitude(quantization)

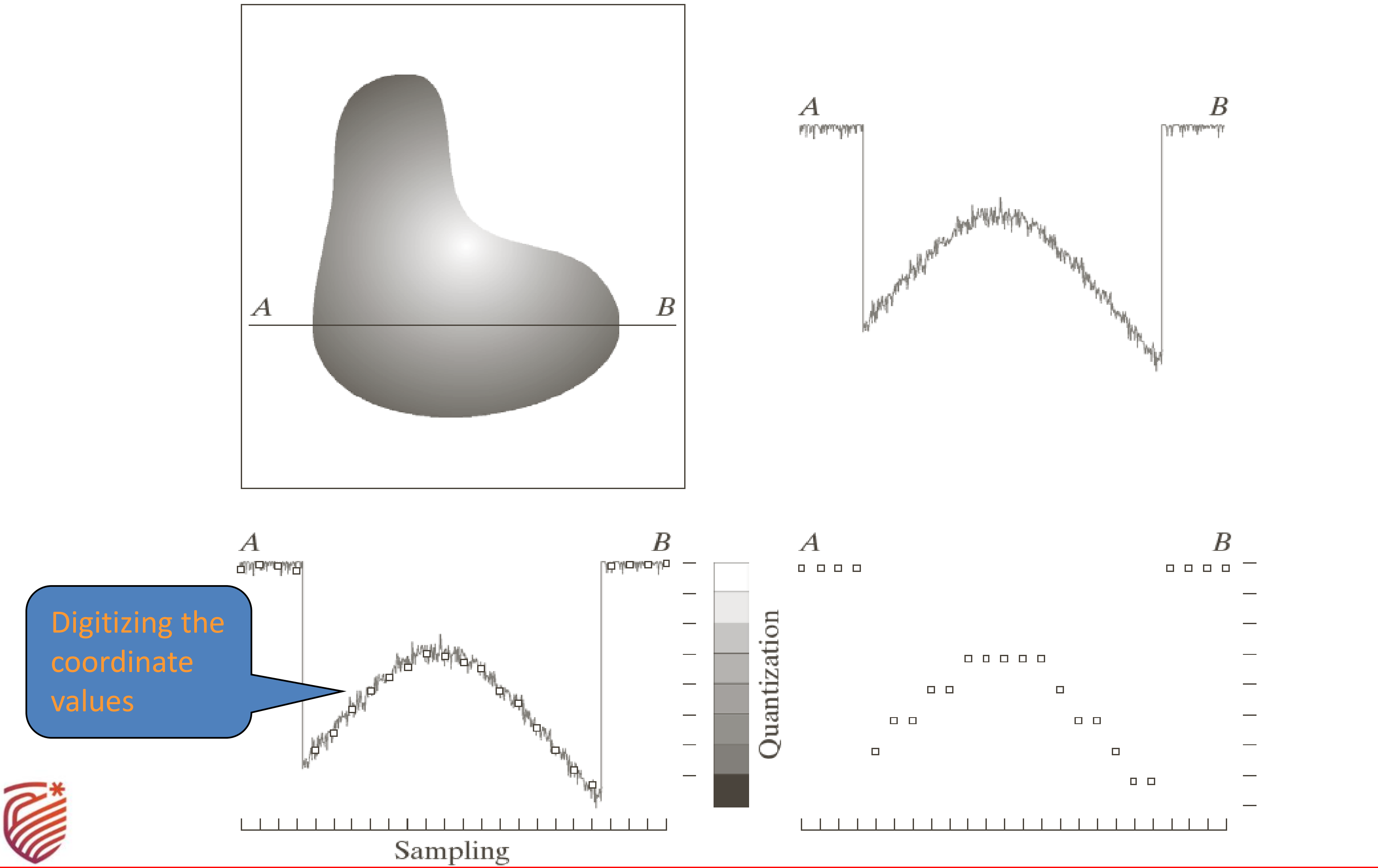



**FIGURE 2.16** Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

# Cont..

- conversion of continuous input signal to a digital form
- sample f(x,y) in both coordinates(sampling)
- sample f(x,y) in amplitude(quantization)

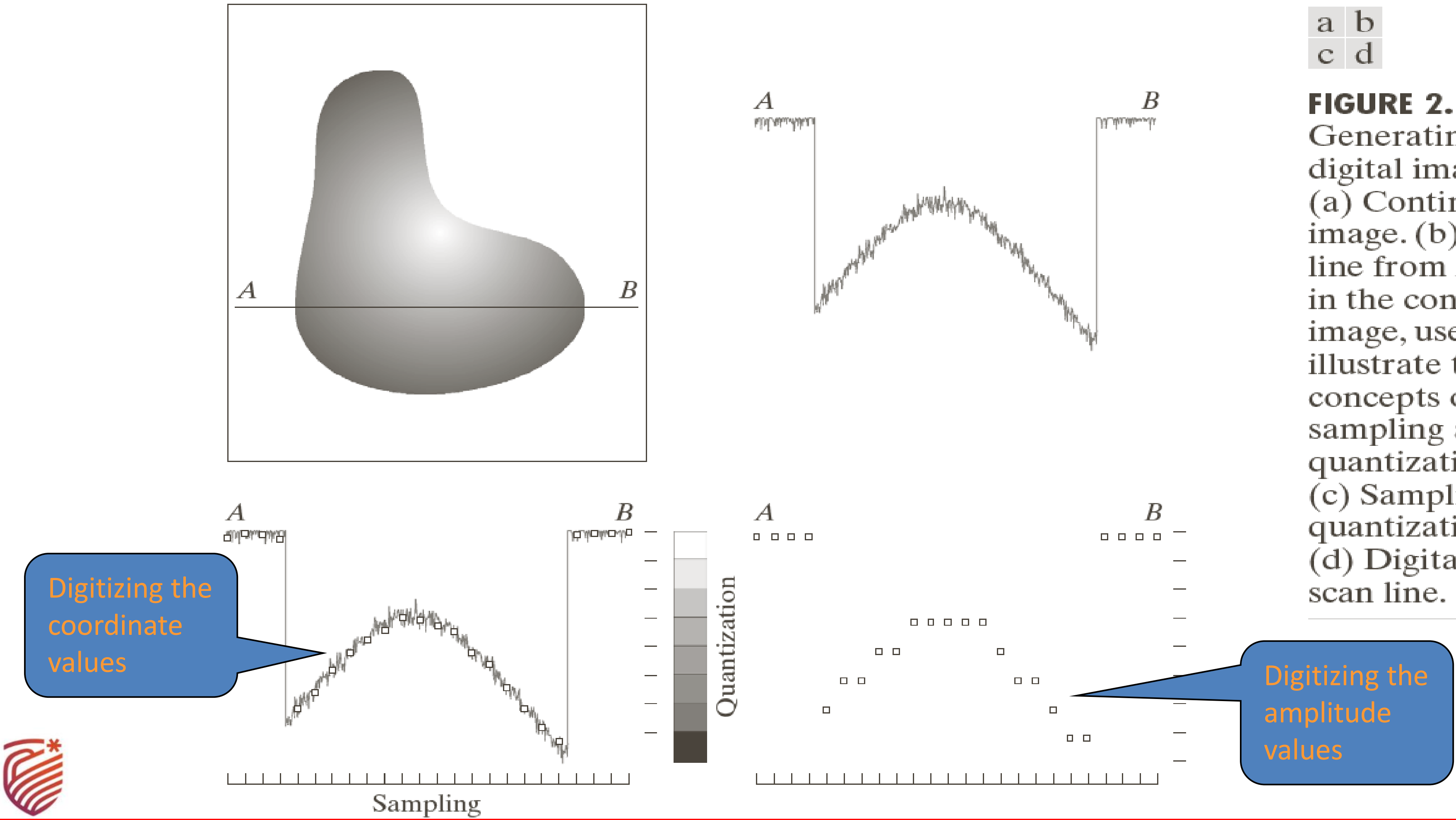



**FIGURE 2.16** Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

# Representing Digital Images

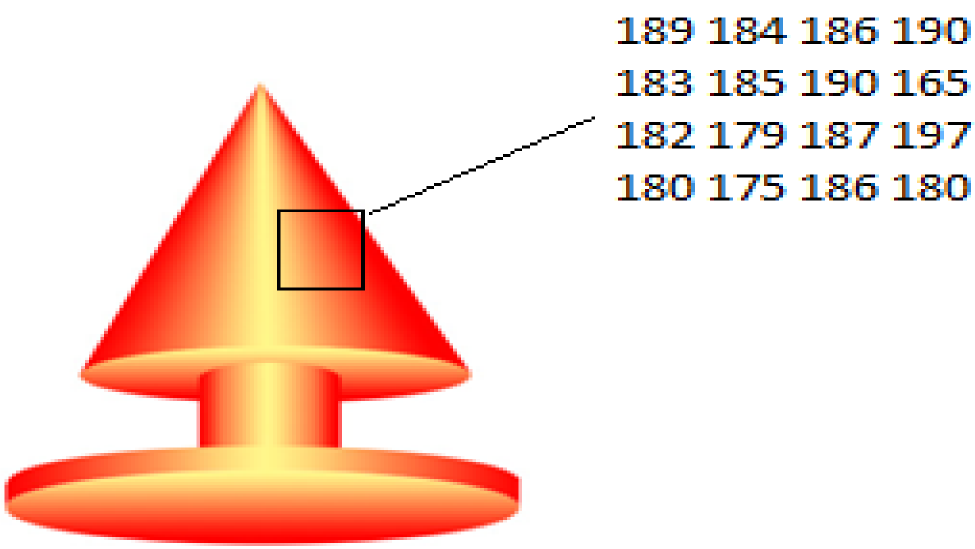
- The representation of an M×N numerical array as
  - f(x,y) as a matrix of real numbers

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

  - elements of the matrix are called pixels (2D)

# Representing Digital Images

- Digital Images
  - a finite set of digital values(picture elements = pixels)
    - each pixel is associated to a position in a 2D region
    - each pixel has a value




can be thought of as a matrix (raster image / raster map) of grey levels / intensity values

digital image

# Representing Digital Images

- Discrete intensity interval [0, L-1], $L=2^k$

- The number b of bits required to store a $M \times N$ digitized image

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

# Sampling and Quantization

- How many samples to take?
  - Number of pixels (samples) in the image
  - Nyquist rate
- How many gray-levels to store?
  - At a pixel position (sample), number of levels of color or intensity to be represented

# Sampling and Quantization

- How many samples to take?
  - The Nyquist Rate
  - Samples must be taken at a rate that is twice the frequency of the highest frequency component to be reconstructed.
  - Under-sampling: sampling at a rate that is too coarse, i.e., is below the Nyquist rate.
  - Aliasing: artefacts that result from under-sampling.

# Spatial and Intensity Resolution

- Spatial resolution
  - — A measure of the smallest discernible detail in an image
  - — stated with *line pairs per unit distance, dots (pixels) per unit distance, dots per inch (dpi)*
- Intensity resolution
  - — The smallest discernible change in intensity level
  - — stated with *8 bits, 12 bits, 16 bits, etc.*

# Spatial and Intensity Resolution

- How many samples to take?

| | |
|---|---|
| a | b |
| c | d |

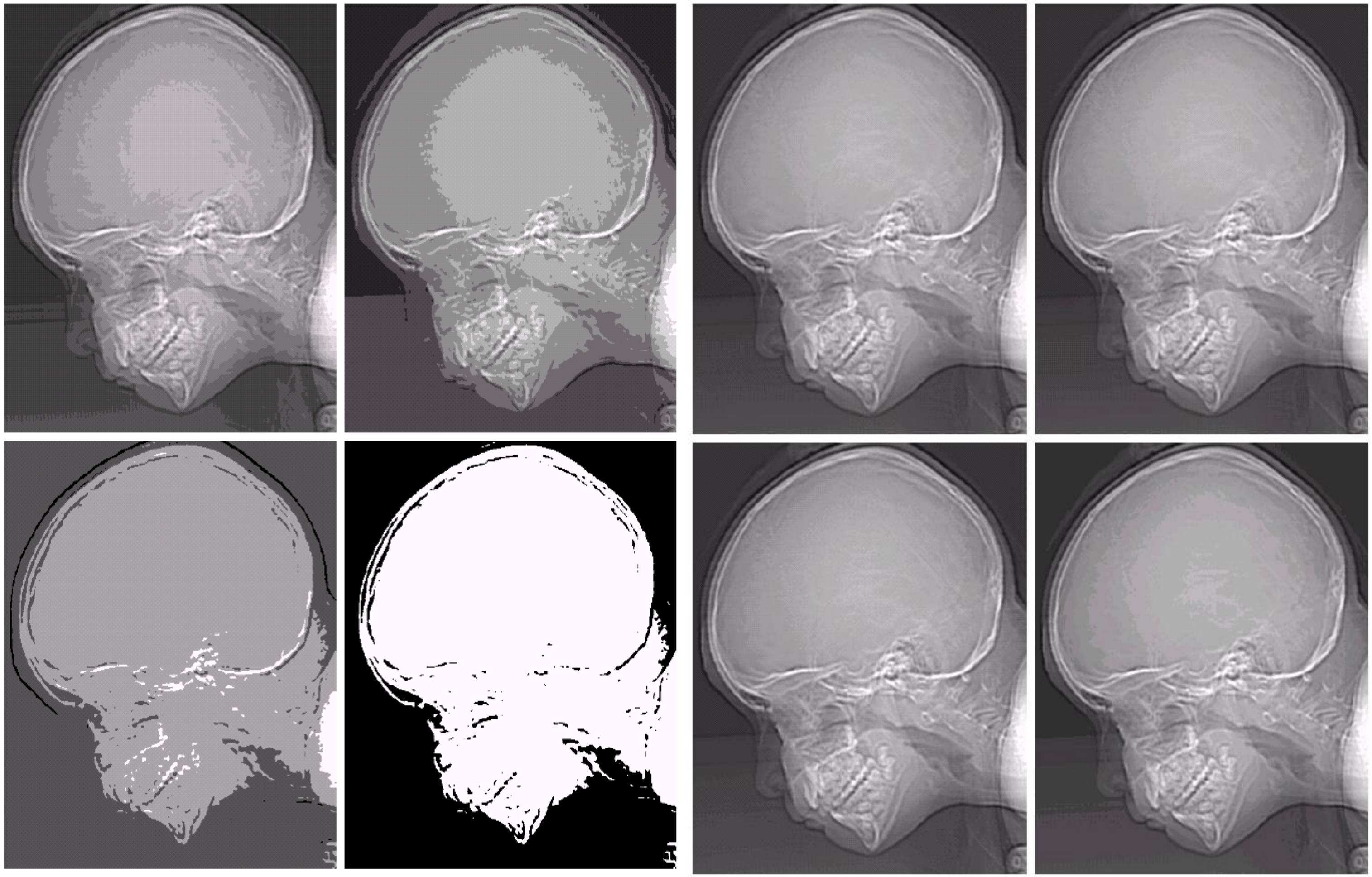
**FIGURE 2.20** Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

# Sampling and Quantization

- How many gray-levels to store?

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.)



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.

# Spatial and Intensity Resolution

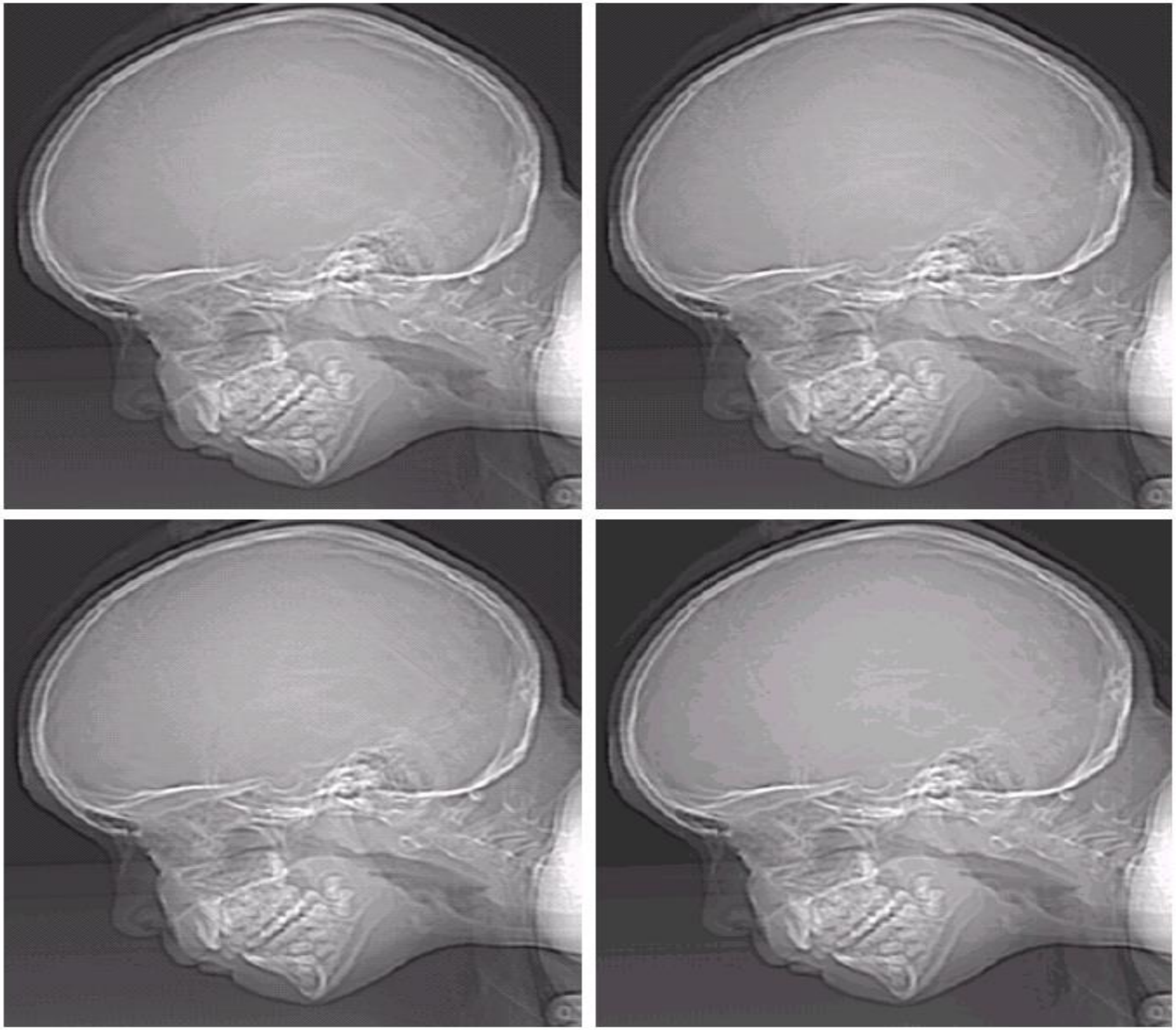
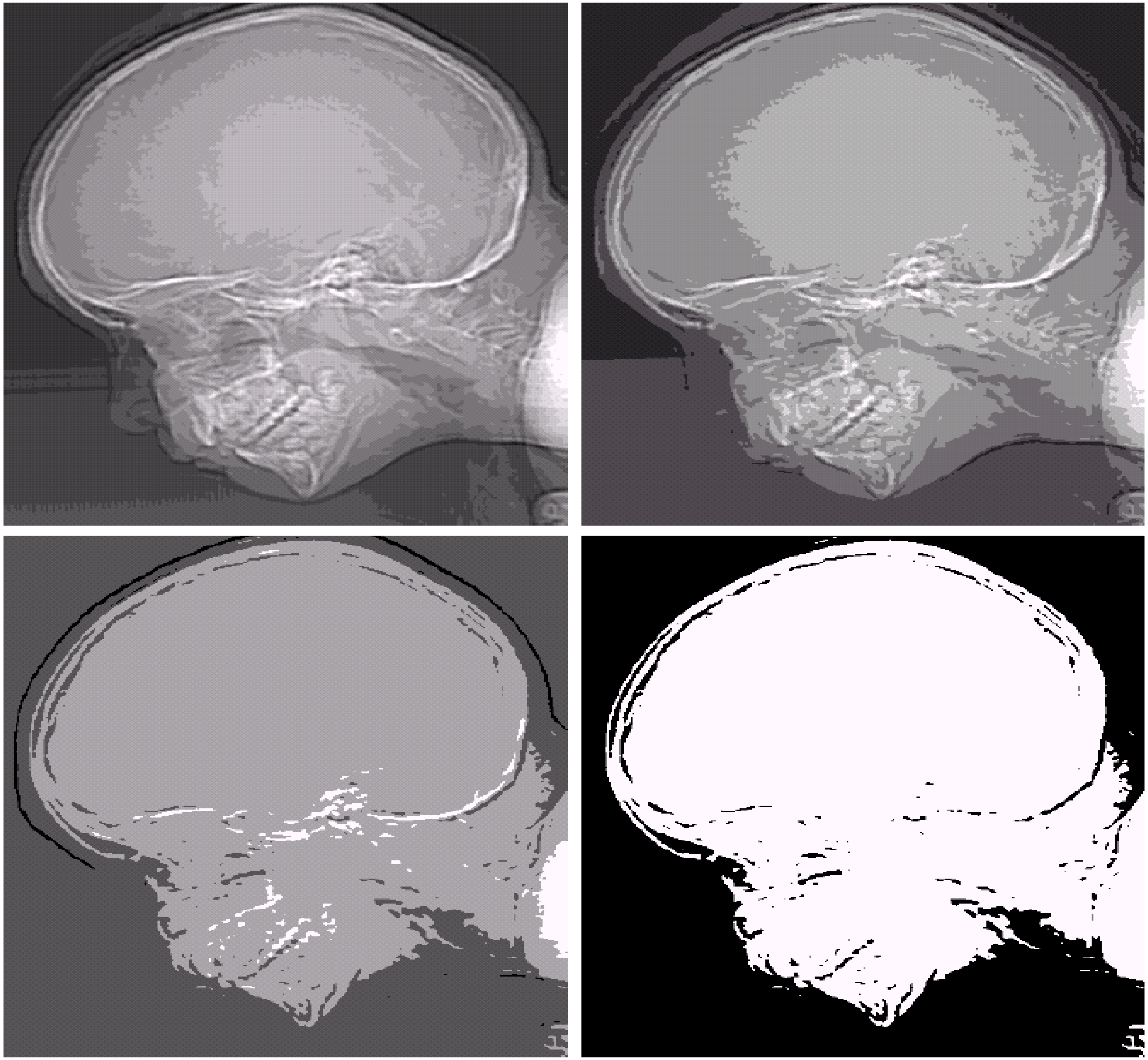



**FIGURE 2.21** (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

# Cont..

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.)

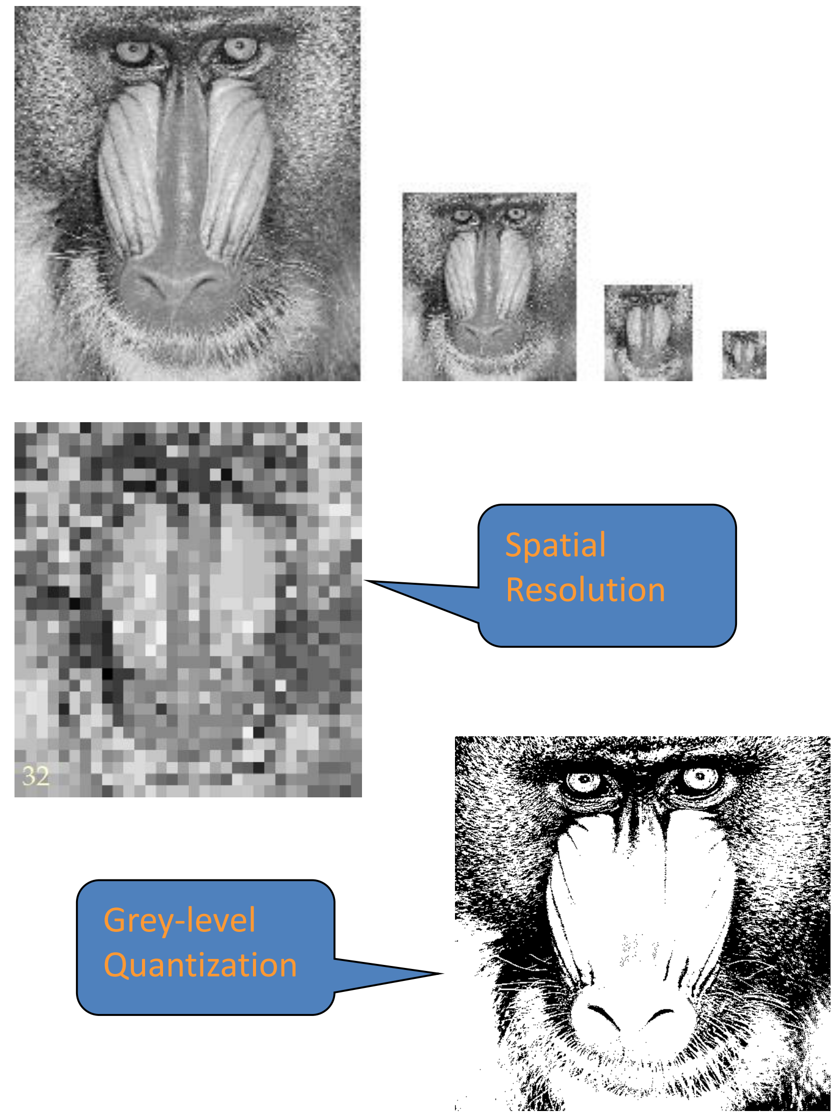


# Sampling and Quantization

- Non-uniform sampling
- Non-uniform quantization

# Cont..

- Image Sampling
  - uniform – same sampling frequency everywhere
  - adaptive – higher sampling frequency in areas with greater detail (not very common)
  - **spatial resolution**: smallest discernible detail in the image (line pairs per mm, for example)
- Image Quantization
  - Grey-level quantization



# Image Interpolation

- **Interpolation** — Process of using known data to estimate unknown values

*e.g.,* zooming, shrinking, rotating, and geometric correction

- **Interpolation** (sometimes called ***resampling***) — an imaging method to increase (or decrease) the number of pixels in a digital image.

Some digital cameras use interpolation to produce a larger image than the sensor captured or to create digital zoom

http://www.dpreview.com/learn/?/key=interpolation

# Cont..

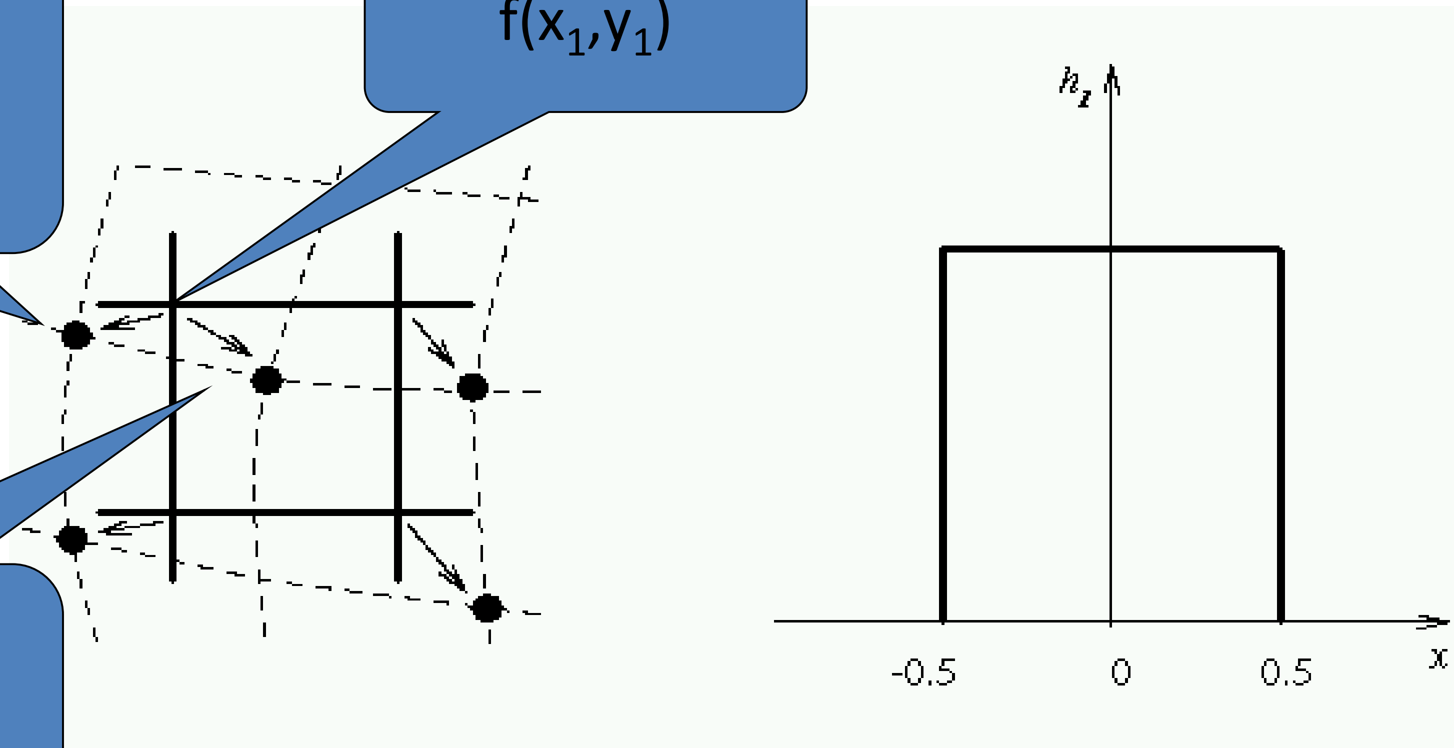
- ***It is the process of using known data to estimate values at unknown locations.***
- **Example:** Suppose that an image has to be enlarged 1.5 times.
- A simple way to visualize zooming is to create an imaginary grid with the same pixel spacing as the original, and then shrink it so that it fits exactly over the original image. Obviously, the pixel spacing in the shrunken grid will be less than the pixel spacing in the original image.
- To perform intensity-level assignment for any point in the overlay, we look for its closest pixel in the original image and assign the intensity of that pixel to the new pixel in the grid. When we are finished assigning intensities to all the points in the overlay grid, we expand it to the original specified size to obtain the zoomed image.

# Image Interpolation: Nearest Neighbor Interpolation

$f_1(x_2,y_2) =$
$f(round(x_2), round(y_2))$
$=f(x_1,y_1)$

$f(x_1,y_1)$

$f_1(x_3,y_3) =$
$f(round(x_3), round(y_3))$
$=f(x_1,y_1)$

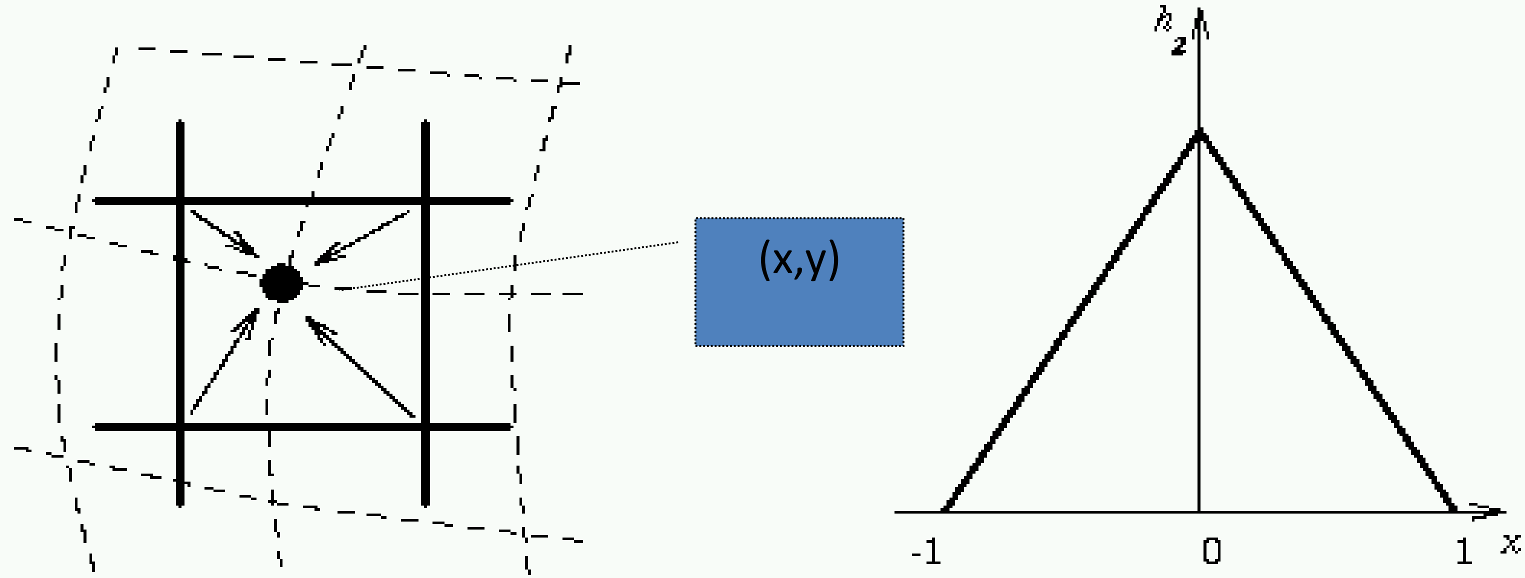


# Nearest Neighbor Interpolation

- It assigns to each new location the intensity of its nearest neighbor in the original image.
- This approach is simple but, it has the tendency to produce undesirable artifacts, such as severe distortion of straight edges

# Bilinear Interpolation

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




(1-a)(1-b)f(x,y) + a(1-b)f(x+1,y) + (1-a)bf(x,y+1) + abf(x+1,y+1)

$$f_2(x,y)$$

$$= (1-a)\cdot(1-b)\cdot f(l,k) + a\cdot(1-b)\cdot f(l+1,k)$$

$$+(1-a)\cdot b\cdot f(l,k+1) + a\cdot b\cdot f(l+1,k+1)$$

$$l = floor(x), k = floor(y), a = x-l, b = y-k.$$

# Bilinear Interpolation

- The 4 nearest neighbors to estimate the intensity at a given location.
- Let denote the coordinates of the location to which we want to assign an intensity value (think of it as a point of the grid described previously), and let denote that intensity value.
- For bilinear interpolation, the assigned value is obtained using the equation

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

- where the 4 coefficients are determined from the 4 equations in 4 unknowns that can be written using the 4 nearest neighbors of point.
- It gives much better results than nearest neighbor interpolation, with a modest increase in computational burden.

# Image Interpolation: Bi-cubic Interpolation

- The intensity value assigned to point (x,y) is obtained by the following equation

$$f_3(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

- The sixteen coefficients are determined by using the sixteen nearest neighbors.

http://en.wikipedia.org/wiki/Bicubic_interpolation

# Cont..

- The sixteen coefficients are determined from the sixteen equations in sixteen unknowns that can be written using the sixteen nearest neighbors of point .
- Observe that Eq.(2) reduces in form to Eq.(1) if the limits of both summations in the former equation are 0 to 1.
- Generally, bicubic interpolation does a better job of preserving fine detail than its bilinear counterpart.
- Bicubic interpolation is the standard used in commercial image editing programs, such as Adobe Photoshop and Corel Photo-paint.

# Examples: Interpolation

Original Image

# Examples: Interpolation

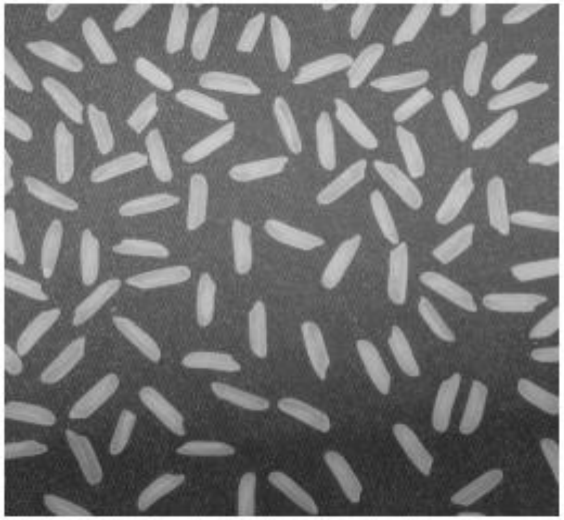
Nearest Neighbor Interpolation

# Examples: Interpolation

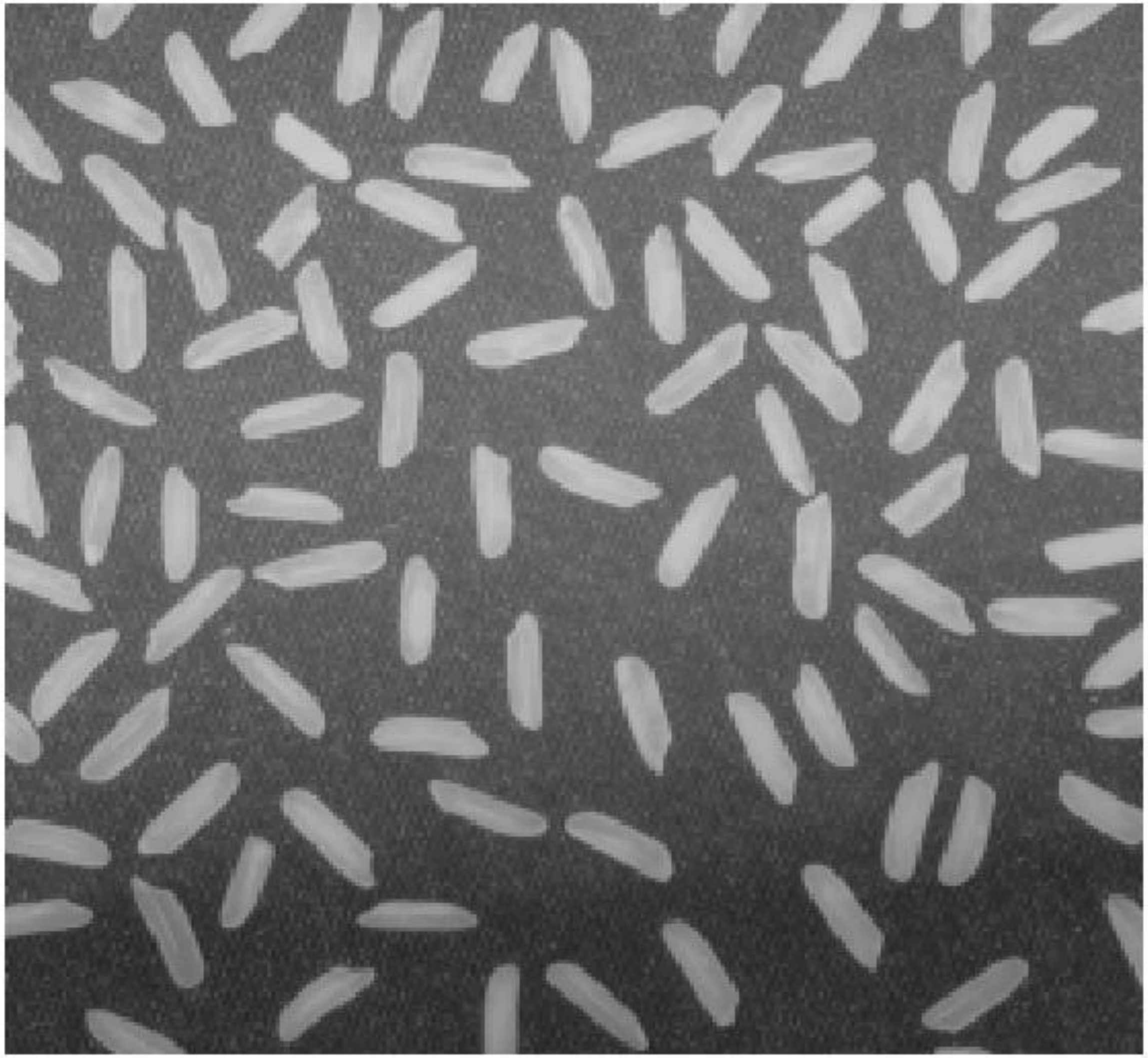
Bilinear Interpolation

# Examples: Interpolation

original image

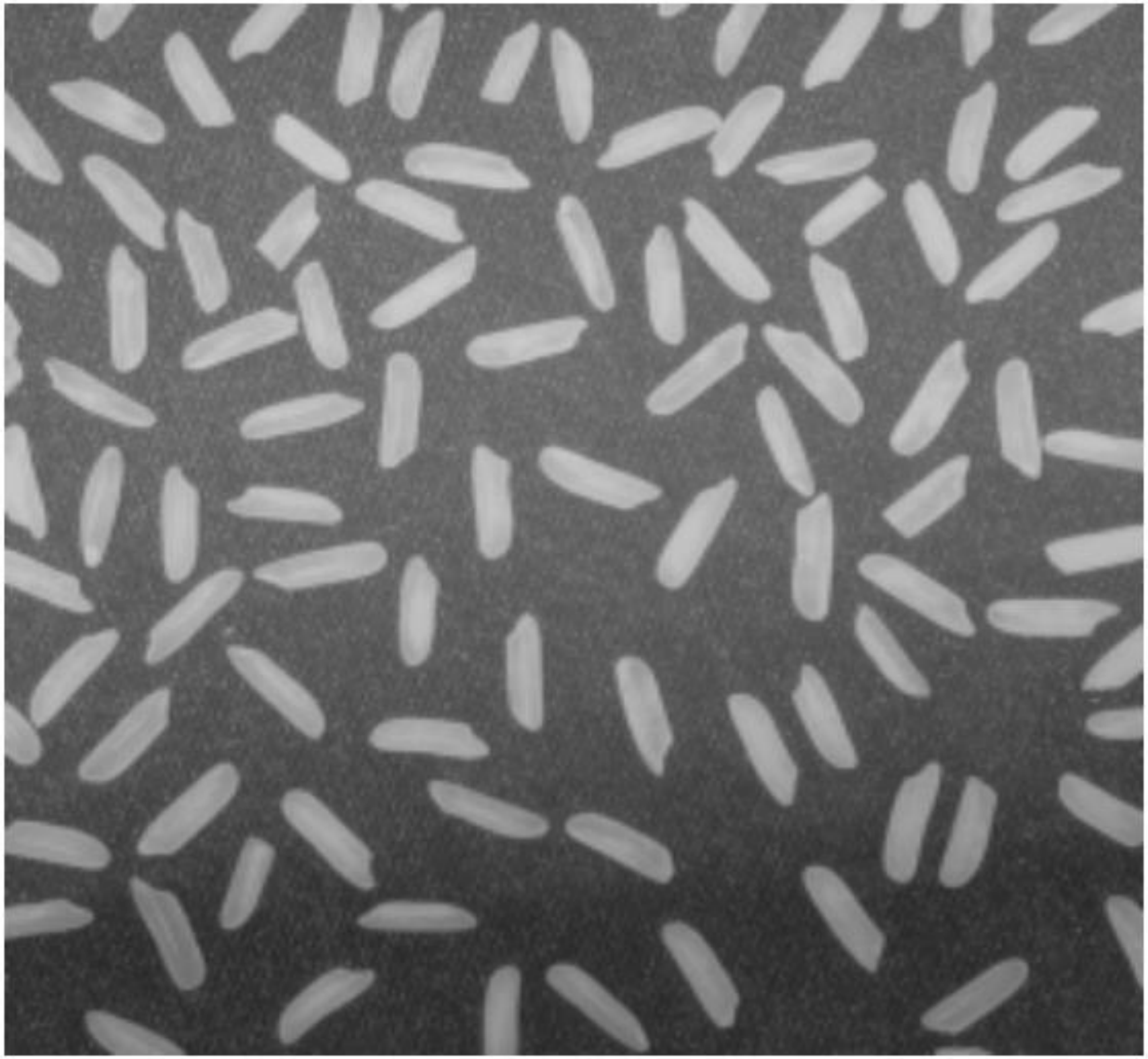


# Examples: Interpolation



# Cont..

bilinear



# Basic Relationships Between Pixels

- Neighborhood
- Adjacency
- Connectivity
- Paths
- Regions and boundaries

# Identify Individual Object

- Connected component labelling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify **connected** pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values.

# Identify Individual Object

- Connected Pixels
  - Two pixels p and q are said to be connected if there is a sequence of foreground (1) pixels

$$p_0, p_1, \ldots p_n$$

  - Such that

$$p_0 = p$$

$$p_n = q$$

$$p_i \text{ is a neighbor of } p_{i-1}$$

# Identify Individual Object

- Pixel Neighbours

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | p | 1 |
| 0 | 1 | 0 |

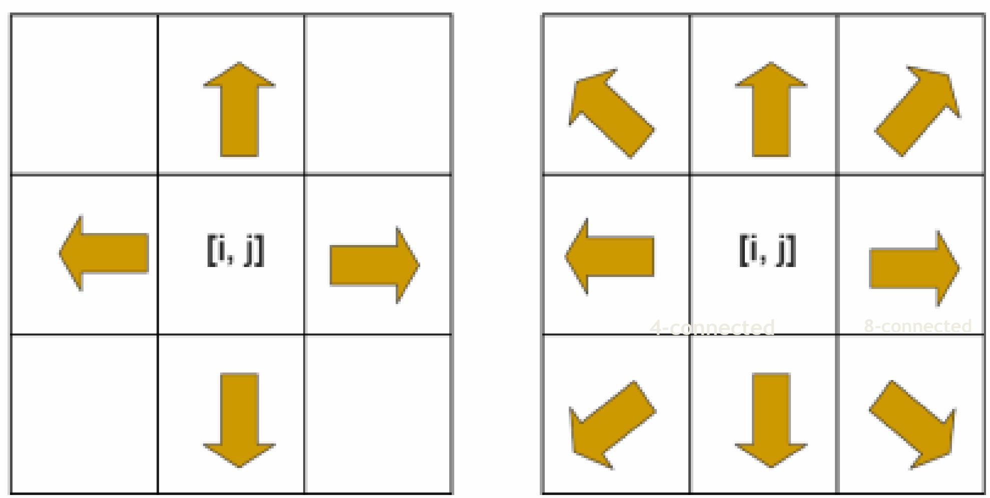
Four nearest neighbors

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | p | 1 |
| 1 | 1 | 1 |

Eight nearest neighbors

# Identify Individual Object

- 4- and 8-Connected Pixels
  - When only the four nearest neighbors are considered part of the neighborhood, then pixels p and q are said to be "4-connected"
  - When the 8 nearest neighbors are considered part of the neighborhood, then pixels p and q are said to be "8-connected"



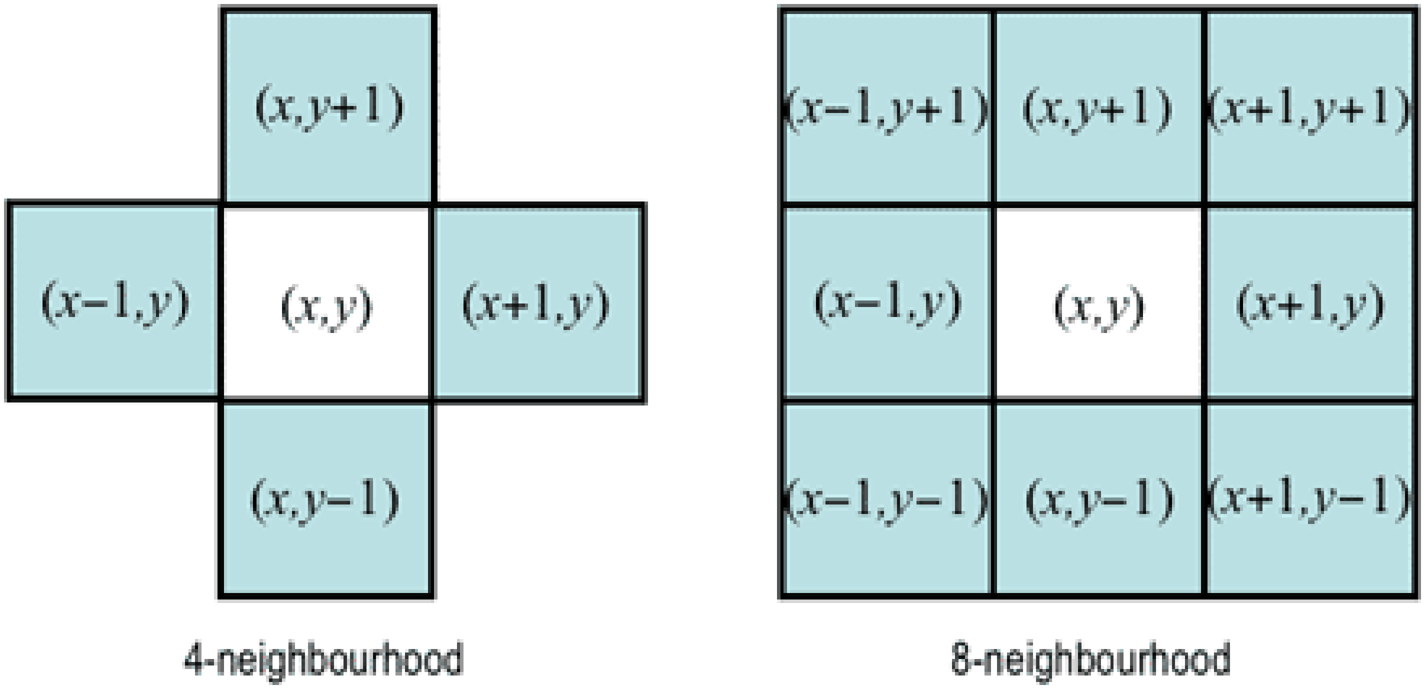
# Neighbors of a Pixel

- A pixel p at (x, y) has 2 horizontal and 2 vertical neighbors:
  - (x+1,y), (x-1,y), (x,y+1), (x,y-1)
  - This set of pixels is called the 4-neighbors of p: $N_4(p)$

- The 4 diagonal neighbors of p are: ($N_D(p)$)
  - (x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)

- $N_4(p) + N_D(p) \rightarrow N_8(p)$: the 8-neighbors of p

# Cont..


(x,y+1)
(x−1,y)
(x,y)
(x+1,y)
(x,y−1)
4-neighbourhood
(x−1,y+1)
(x,y+1)
(x+1,y+1)
(x−1,y)
(x,y)
(x+1,y)
(x−1,y−1)
(x,y−1)
(x+1,y−1)
8-neighbourhood

# Cont..

- **Neighbors** of a pixel p at coordinates (x,y)

➢ **4-neighbors of p**, denoted by $\mathbf{N_4(p)}$:
(x-1, y), (x+1, y), (x,y-1), and (x, y+1).

➢ **4 diagonal neighbors of p**, denoted by $\mathbf{N_D(p)}$:
(x-1, y-1), (x+1, y+1), (x+1,y-1), and (x-1, y+1).

➢ **8 neighbors of p**, denoted $\mathbf{N_8(p)}$
$N_8(p) = N_4(p) \cup N_D(p)$

# Cont..

- **Adjacency**

  Let V {1,2} be the set of intensity values

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

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

# Cont..

- **Adjacency**

  Let V be the set of intensity values

- **m-adjacency**: Two pixels p and q with values from V are m-adjacent if

  (i) q is in the set $N_4(p)$, or

  (ii) q is in the set $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

# Cont..

- **Path**

➢ A (digital) path (or curve) from pixel p with coordinates $(x_0, y_0)$ to pixel q with coordinates $(x_n, y_n)$ is a sequence of distinct pixels with coordinates

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

Where $(x_i, y_i)$ and $(x_{i-1}, y_{i-1})$ are adjacent for $1 \le i \le n$.

➢ Here *n* is the *length* of the path.

➢ If $(x_0, y_0) = (x_n, y_n)$, the path is ***closed*** path.

➢ We can define 4-, 8-, and m-paths based on the type of adjacency used.

# Examples: Adjacency and Path

**V = {1, 2}**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | | 0 | 1 | 1 | | 0 | 1 | 1 |
| 0 | 2 | 0 | | 0 | 2 | 0 | | 0 | 2 | 0 |
| 0 | 0 | 1 | | 0 | 0 | 1 | | 0 | 0 | 1 |

**4-adjacent**

# Examples: Adjacency and Path

**V = {1, 2}**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 2 | 0 |
| 0 | 0 | 1 |

**4-adjacent**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 2 | 0 |
| 0 | 0 | 1 |

**8-adjacent**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 2 | 0 |
| 0 | 0 | 1 |

# Examples: Adjacency and Path

**V = {1, 2}**

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{array}$$

**4-adjacent** **8-adjacent** **m-adjacent**

# Examples: Adjacency and Path

**V = {1, 2}**

$$\begin{matrix} 0_{1,1} & 1_{1,2} & 1_{1,3} \\ 0_{2,1} & 2_{2,2} & 0_{2,3} \\ 0_{3,1} & 0_{3,2} & 1_{3,3} \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{matrix}$$

**8-adjacent** **m-adjacent**

The 8-path from (1,3) to (3,3):

(i) (1,3), (1,2), (2,2), (3,3)

(ii) (1,3), (2,2), (3,3)

The m-path from (1,3) to (3,3):

(1,3), (1,2), (2,2), (3,3)

# Examples: Adjacency and Path

**V = {1, 2, 3}**

| | | |
|---|---|---|
| 0 | 1 | 3 |
| 3 | 2 | 3 |
| 0 | 0 | 0 |

| | | |
|---|---|---|
| 0 | 1 | 3 |
| 3 | 2 | 3 |
| 0 | 0 | 0 |

| | | |
|---|---|---|
| 0 | 1 | 3 |
| 3 | 2 | 3 |
| 0 | 0 | 0 |

$N_4(p) \cap N_4(q) = \emptyset$ ; $N_4(2) \cap N_4(3) = (1,3) \cap (1,3) \neq \emptyset$

# Examples: Adjacency and Path

**V = {1}**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

**4-adjacent**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

**8-adjacent**

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

**m-adjacent**

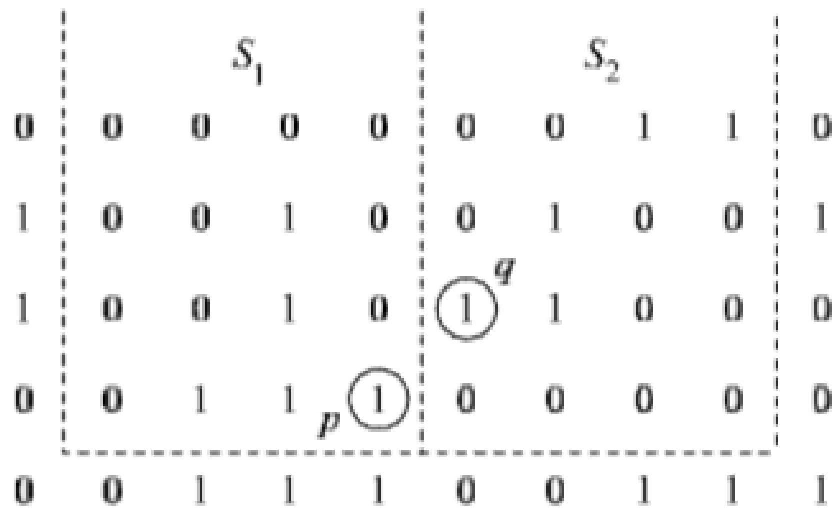
# Cont..

- **Example: Q2.11:** Consider the two image subsets, and shown in the following figure. For V={1} determine whether these two subsets are (a) 4-adjacent, (b) 8-adjacent, or (c) m-adjacent.

| | | $S_1$ | | | | $S_2$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

# Cont..

- **Example Q2.11:**
- Let p and q be as shown in Fig. Then,
- (a) $S_1$ and $S_2$ are not 4-connected because q is not in the set $N_4(p)$;
- (b) $S_1$ and $S_2$ are 8-connected because q is in the set $N_8(p)$
- (c) $S_1$ and $S_2$ are m-connected because (i) q is in $N_D(p)$, and (ii) the set $N_4(p) \cap N_4(q)$ is empty.



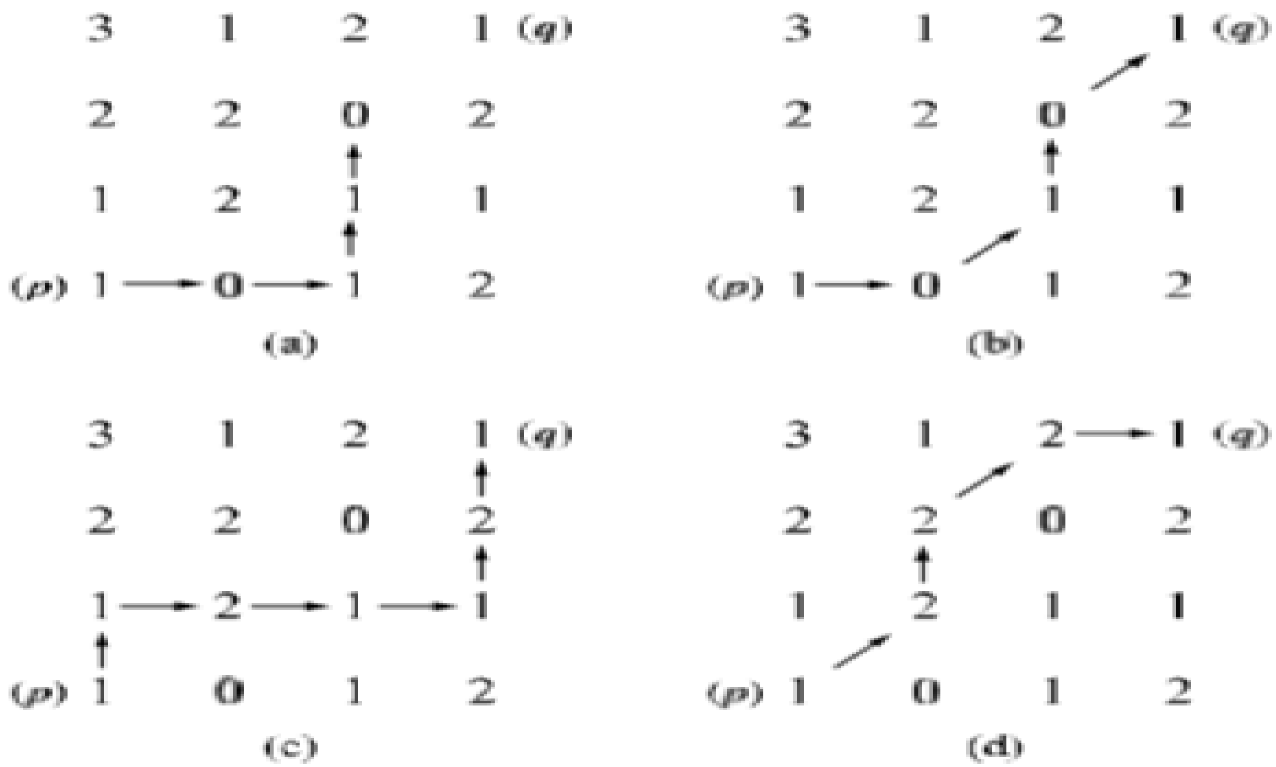
# Cont..

- **Example Q2.15:** Consider the image segment shown.
  a) Let V = {0,1} and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.
  b) Repeat for V = {1, 2}.

| | | | | |
|---|---|---|---|---|
| | 3 | 1 | 2 | 1 $(q)$ |
| | 2 | 2 | 0 | 2 |
| | 1 | 2 | 1 | 1 |
| $(p)$ | 1 | 0 | 1 | 2 |

# Cont..

- **Example Q2.15:** Consider the image segment shown.
  a) Let V = {0,1} and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.
  b) Repeat for V = {1, 2}.



# Cont..

- **Example Q2.15:**
- When V ={0,1}, 4-path does not exist between p and q because it is impossible to get from p to q by traveling along points that are both 4-adjacent and also have values from V . Figure (a)shows this condition unit is not possible to get to q.
- The shortest 8-path is shown in Fig. (b); its length is 4. In this case the length of shortest m and 8-paths is the same. Both of these shortest paths are unique in this case.
- (b) One possibility for the shortest 4-path when V ={1,2} is shown in Fig. P (c); its length is 6. It is easily verified that another 4-path of the same length exists between p and q.
- One possibility for the shortest 8-path (it is not unique) is shown in Fig. (d); its length is 4. The length of a shortest m-path similarly is 4.

# Path

- A ***digital path (or curve)*** *from pixel* ***p*** *with coordinate* ***(x, y)*** *to pixel* ***q*** *with* coordinate **(s, t)** is a sequence of ***distinct pixels*** *with coordinates* $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, where $(x_0, y_0) = (x,y)$, $(x_n, y_n) = (s,t)$
  - $(x_i, y_i)$ is adjacent pixel $(x_{i-1}, y_{i-1})$ for $1 \leq j \leq n$ ,
  - **n**- The ***length of the path.***
  - If $(x_0, y_0) = (x_n, y_n)$ : the path is ***closed path.***
  - We can define ***4- ,8- ,*** *or* ***m-paths*** *depending on the type of adjacency specified.*

# Basic Relationships Between Pixels

- **Connected in S**

  Let **S** represent a subset of pixels in an image. Two pixels p with coordinates $(x_0, y_0)$ and **q** with coordinates $(x_n, y_n)$ are said to be **connected** in **S** if there exists a path

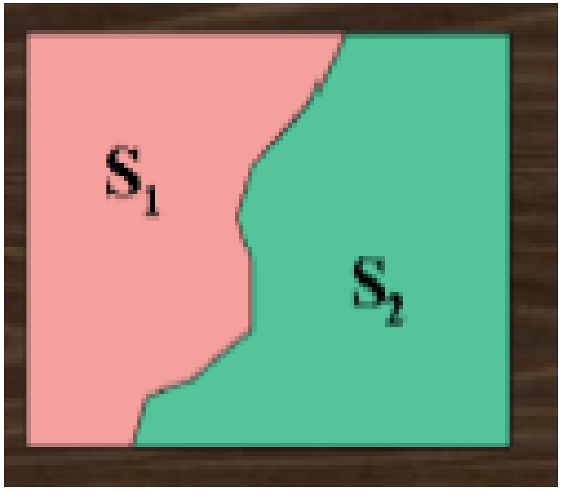
  $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

  Where $\forall i, 0 \le i \le n, (x_i, y_i) \in S$

# Connectivity

- 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.

- Two image subsets $S_1$ and $S_2$ are adjacent if some pixel in $S_1$ is adjacent to some pixel in $S_2$



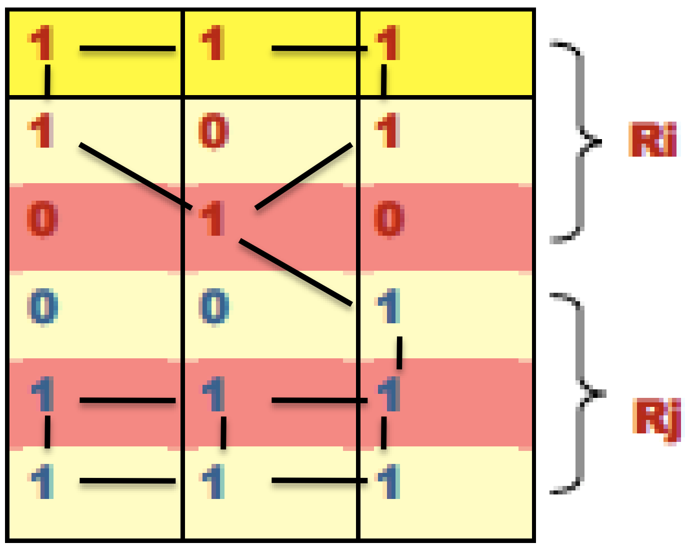
# Cont..

Let ***S*** represent a subset of pixels in an image

- For every pixel ***p*** in ***S***, the set of pixels in ***S*** that are connected to ***p*** is called a ***connected component*** of **S**.
- If S has only one connected component, then **S** is called ***Connected Set*.**
- We call **R** a **region** of the image if **R** is a connected set
- Two regions, $R_i$ and $R_j$ are said to be ***adjacent*** if their union forms a connected set.
- Regions that are not to be adjacent are said to be ***disjoint*.**

# Region

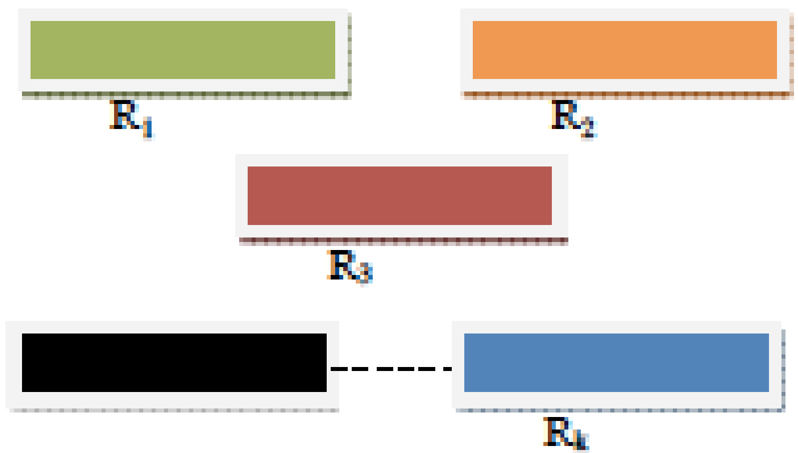
- Let R to be a subset of pixels in an image, we call a R a region of the image. If R is a ***connected set.***
- Region that are not **adjacent** are said to be **disjoint.**
- *Example:* ***the two regions (of 1s) in figure, are adjacent only if 8-adjacany is used.***



# Cont…

- ***4-path** between the two regions does not exist, (so their union in not a connected set).*

- **Boundary (border)** image contains K disjoint regions, $R_k$, k = 1, 2, …., k, none of which touches the image border.



# Cont..

- **Boundary (or border)**

- The ***boundary*** of the region R is the set of pixels in the region that have one or more neighbors that are not in R.
- If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns of the image.

- **Foreground and background**

- An image contains K disjoint regions, $R_k$, $k = 1, 2, \ldots, K$. Let $R_u$ denote the union of all the K regions, and let $(R_u)^c$ denote its complement.

All the points in $R_u$ is called **foreground**;

All the points in $(R_u)^c$ is called **background**.

# Question 1

- **In the following arrangement of pixels, are the two regions (of 1s) adjacent? (if 8-adjacency is used)**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Region 1

Region 2

# Question 2

- **In the following arrangement of pixels, are the two parts (of 1s) adjacent? (if 4-adjacency is used)**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Part 1

Part 2

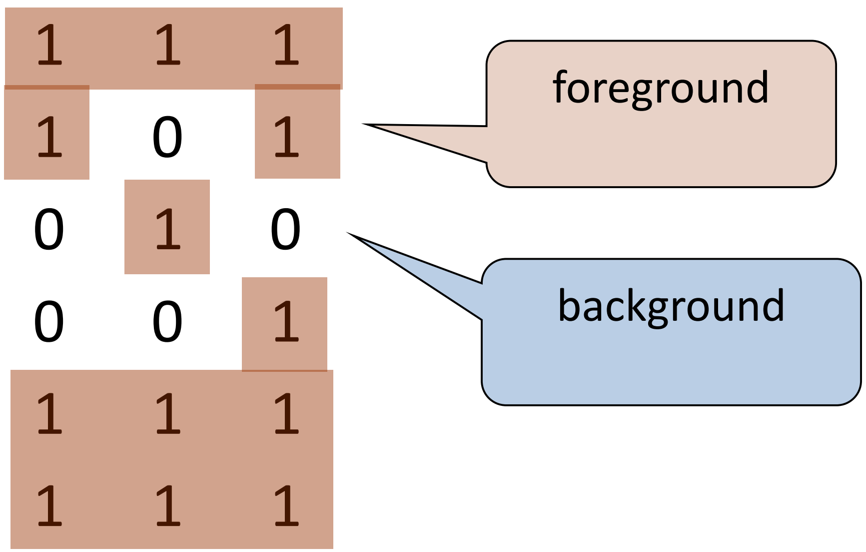
- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Region 1

Region 2

- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**



# Question 3

- **In the following arrangement of pixels, the circled point is part of the boundary of the 1-valued pixels if 8-adjacency is used, true or false?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Question 4

- **In the following arrangement of pixels, the circled point is part of the boundary of the 1-valued pixels if 4-adjacency is used, true or false?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Distance Measures

- Given pixels ***p, q*** and ***z*** with coordinates **(x, y), (s, t), (u, v)** respectively, the **distance function D** has following properties:

a. **$D(p, q) \geq 0$ [$D(p, q) = 0$, iff $p = q$]**

b. **$D(p, q) = D(q, p)$**

c. **$D(p, z) \leq D(p, q) + D(q, z)$**

# Distance Measures

The following are the different **Distance measures**:

a. **Euclidean Distance :**

$D_e(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}$

Example: The points contained in a disk of radius r centred at (x,y).

b. **$D_4$Distance or City Block Distance**:

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

Pixels having a D4 distance from (x,y)

| | | | | |
|---|---|---|---|---|
| | | 2 | | |
| | 2 | 1 | 2 | |
| 2 | 1 | 0 | 1 | 2 |
| | 2 | 1 | 2 | |
| | | 2 | | |

less than or equal to some value r from a Diamond centred (x,y)

Example: the pixels with D4=1 are the **4-nighbors of (x, y).**

c. **$D_8$ distance or Chess Board Distance**:

$D_8(p, q) = \max(|x-s|, |y-t|)$ square – centred at (x, y)

$D_8$ = 1 are 8-neighbors of (x,y)

Example: $D_8$ distance ≤ 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 |

# Cont..

d. **$D_m$ distance:**

- Is defined as the shortest m-path between the points
- The distance between pixels depends only on the values of pixels.
- **Example:** Consider the following arrangement of pixels and assume that P, $P_2$ and $P_4$ have value 1 and that $P_1$ and $P_3$ can have a value of 0 or 1.

| | | |
|---|---|---|
| | $P_3$ | $P_4$ |
| $P_1$ | $P_2$ | |
| $P$ | | |

Suppose, that we consider adjacency of pixels value 1 (i.e. v = {1})

a) *if $P_1$ and $P_3$ are 0: then* shortest $D_m$ distance = 2;

b) *if $P_1 = 1$ and $P_3 = 0$: : then* shortest $D_m$ distance = 3;

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3;

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4;

# Cont..

– Example: Consider the following arrangement of pixels and $P = P_2 = P_4 = 1$ and $P_1$ and $P_3$ = 0 or 1. Suppose, that we consider adjacency of pixels value 1 (i.e. v = {1})

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P_3$ | $P_4$ | $P_3$ | 1 | $P_3$ | 1 | $P_3$ | 1 | $P_3$ | 1 | |
| $P_1$ | $P_2$ | | $P_1$ | 1 | $P_1$ | 1 | $P_1$ | 1 | $P_1$ | 1 | |
| P | | | 1 | | 1 | | 1 | | 1 | | |
| | | | (a) | | (b) | | (c) | | (d) | | |

a) *if $P_1$ and $P_3$ are 0: then* shortest $D_m$ distance = 2

b) *if $P_1 = 1$ and $P_3 = 0$: then* $D_m$ distance = 3;

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3;

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4;

# Cont..

– Example: Consider the following arrangement of pixels and $P = P_2 = P_4 = 1$ and $P_1$ and $P_3$ = 0 or 1. Suppose, that we consider adjacency of pixels value 1 (i.e. v={1})

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P_3$ | $P_4$ | | $0_3$ | 1 | | $P_3$ | 1 | | $P_3$ | 1 | | $P_3$ | 1 |
| $P_1$ | $P_2$ | | $0_1$ | 1 | | $P_1$ | 1 | | $P_1$ | 1 | | $P_1$ | 1 | |
| P | | | 1 | | | 1 | | | 1 | | | 1 | | |
| | | | | (a) | | | (b) | | | (c) | | | (d) | |

a) *if $P_1$ and $P_3$ are 0: then* shortest m-path b/w P and $P_4$, $D_m$ distance = 2

b) *if $P_1 = 1$ and $P_3 = 0$: then* $D_m$ distance = 3;

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3;

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4;

# Cont..

– Example: Consider the following arrangement of pixels and $P = P_2 = P_4 = 1$ and $P_1$ and $P_3 = 0$ or 1. Suppose, that we consider adjacency of pixels value 1 (i.e. v={1})

| | | |
|---|---|---|
| | $P_3$ | $P_4$ |
| $P_1$ | $P_2$ | |
| $P$ | | |

| | | |
|---|---|---|
| | $0_3$ | 1 |
| $0_1$ | 1 | |
| 1 | | |

(a)

| | | |
|---|---|---|
| | $0_3$ | 1 |
| $1_1$ | 1 | |
| 1 | | |

(b)

| | | |
|---|---|---|
| | $P_3$ | 1 |
| $P_1$ | 1 | |
| 1 | | |

(c)

| | | |
|---|---|---|
| | $P_3$ | 1 |
| $P_1$ | 1 | |
| 1 | | |

(d)

a) *if $P_1$ and $P_3$ are 0: then* $D_m$ distance = 2

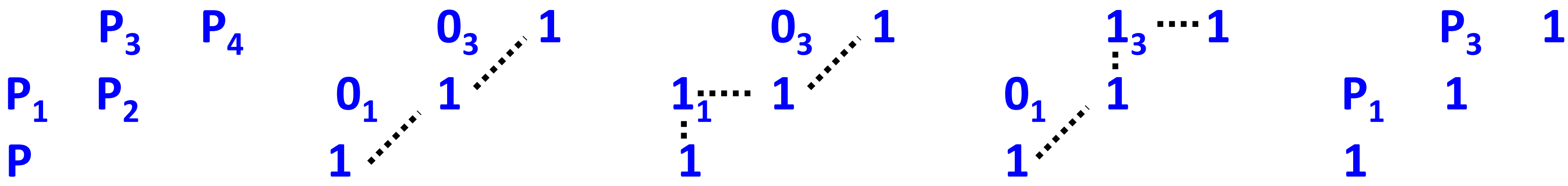
b) *if $P_1 = 1$ and $P_3 = 0$: then* P and $P_2$will not be m-adjacent. Shortest path $D_m$ distance = 3(Path goes through the sequence of points P $P_1P_2P_4$);

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3;

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4;

# Cont..

– Example: Consider the following arrangement of pixels and $P = P_2 = P_4 = 1$ and $P_1$ and $P_3 = 0$ or 1. Suppose, that we consider adjacency of pixels value 1 (i.e. v={1})




(a) (b) (c) (d)

a) *if $P_1$ and $P_3$ are 0: then* shortest $D_m$ distance = 2

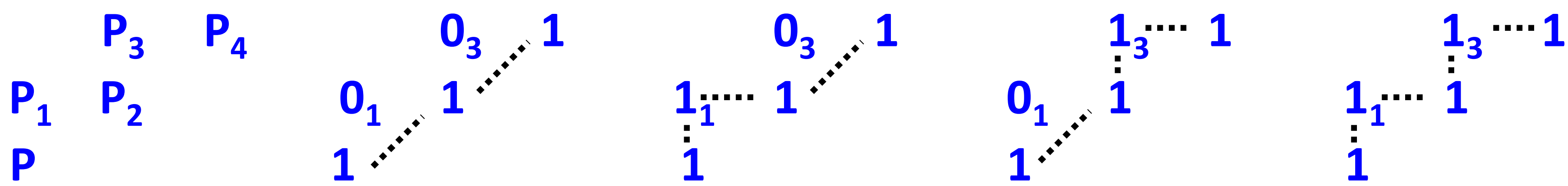
b) *if $P_1 = 1$ and $P_3 = 0$: then* shortest $D_m$ distance = 3;

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3(Path goes through the sequence of points $P\ P_2P_3P_4$);

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4;

# Cont..

– Example: Consider the following arrangement of pixels and $P = P_2 = P_4 = 1$ and $P_1$ and $P_3$ = 0 or 1. Suppose, that we consider adjacency of pixels value 1 (i.e. v={1})




(a) (b) (c) (d)

a) *if $P_1$ and $P_3$ are 0: then* shortest $D_m$ distance = 2

b) *if $P_1 = 1$ and $P_3 = 0$: then* shortest $D_m$ distance = 3;

c) *if $P_1 = 0$ ; and $P_3 = 1$: then* shortest $D_m$ distance = 3;

d) *if $P_1 = P_3 = 1$: then* shortest $D_m$ distance = 4 (Path goes through the sequence of points $P\ P_1P_2P_3P_4$);

# Question 5

- **In the following arrangement of pixels, what's the value of the chessboard distance between the circled two points?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Question 6

- **In the following arrangement of pixels, what's the value of the city-block distance between the circled two points?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Question 7

- **In the following arrangement of pixels, what's the value of the length of the m-path between the circled two points?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Question 8

- **In the following arrangement of pixels, what's the value of the length of the m-path between the circled two points?**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Introduction to Mathematical Operations in DIP

- **Array vs. Matrix Operation**
- An array operation involving one or more images is carried out on a pixel-by pixel basis.
- Images can be viewed equivalently as matrices.
- In many situations operations between images are carried out using matrix theory.
- It is for this reason that a clear distinction must be made between array and matrix operations.
- **Example:** Consider the following images-

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

# Introduction to Mathematical Operations in DIP

- **Array vs. Matrix Operation**
- The array product and the matrix product of these two images are given by

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

Array product operator

$$A \;.*\; B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

**Array product**

Matrix product operator

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

**Matrix product**

# Cont..

- **Linear vs. Nonlinear Operation:**
- Consider a general operator, H that produces an output image, g(x, y), for a given input image, f(x, y):

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

$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right]$$

$$= H\left[a_i f_i(x,y)\right] + H\left[a_j f_j(x,y)\right]$$

Additivity

$$= a_i H\left[f_i(x,y)\right] + a_j H\left[f_j(x,y)\right]$$

Homogeneity

$$= a_i g_i(x,y) + a_j g_j(x,y)$$

H is said to be a **linear operator;**

H is said to be a **nonlinear operator** if it does not meet the above qualification.

# Relationship between pixels (Contd..)

- Arithmetic/Logic Operations:
  - **Addition : *p + q***
  - **Subtraction: *p – q***
  - **Multiplication: *p*q***
  - **Division: *p/q***
  - **AND: *p AND q***
  - **OR : *p OR q***
  - **Complement: NOT(*q*)**

# Cont..

- Arithmetic operations (e.g. image subtraction pixel by pixel)
- Matrix and vector operations
- Linear (e.g. sum) and nonlinear operations (e.g. min and max)
- Set and logical operations
- Spatial and neighborhood operations (e.g. local average)
- Geometric spatial transformations (e.g. rotation)

# Arithmetic Operations

- Arithmetic operations between images are **array operations**. They are carried out between corresponding pixel pairs. The four arithmetic operations are denoted as
  - $s(x,y) = f(x,y) + g(x,y)$
  - $d(x,y) = f(x,y) - g(x,y)$
  - $p(x,y) = f(x,y) \times g(x,y)$
  - $v(x,y) = f(x,y) \div g(x,y)$
- The 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 M and N are the row and column sizes of the images
- $s, d, p$ and $v$ are images of size $M \times N$

# Addition of Noisy Images for Noise Reduction

- **Noiseless image:** $f(x, y)$
- **Noise:** $n\ (x, y)$ (at every pair of coordinates (x,y), the noise is uncorrelated and has zero average value)
- **Corrupted image:** $g(x, y)$

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

- Reducing the noise by adding a set of noisy images, $\{g_i(x, y)\}$

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

- This is a technique used frequently for image enhancement.

# Addition of Noisy Images for Noise Reduction

- Let the image is formed by averaging K different noisy images

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

then $E\{\bar{g}(x,y)\} = f(x,y)$ and $\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{n(x,y)}$ ---(2)

where $E\{\bar{g}(x,y)\}$ is the expected value of $\bar{g}$ and $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{n(x,y)}$ are the variances of $\bar{g}$ and $n$ respectively, all at coordinates $(x,y)$. The standard deviation (square root of the variance) at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)} \quad \text{-------(3)}$$

# Addition of Noisy Images for Noise Reduction

- As **K** increases, in Eqs.(2) and (3) indicate that the variability (as measured by the variance or the standard deviation) of the pixel values at each location ***(x, y)*** decreases.

- Because $E\{\bar{g}(x,y)\} = f(x,y)$ this means that $\bar{g}(x,y)$ approaches $f(x, y)$ as the number of noisy images used in the averaging process increases.

- In practice, the images $g_i(x, y)$ must be *registered* (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

# Addition of Noisy Images for Noise Reduction

- Let the image is formed by averaging K different noisy images

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

then

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

$$= E\left\{\frac{1}{K}\sum_{i=1}^{K}[f(x,y)+n_i(x,y)]\right\}$$

$$= f(x,y) + E\left\{\frac{1}{K}\sum_{i=1}^{K} n_i(x,y)\right\}$$

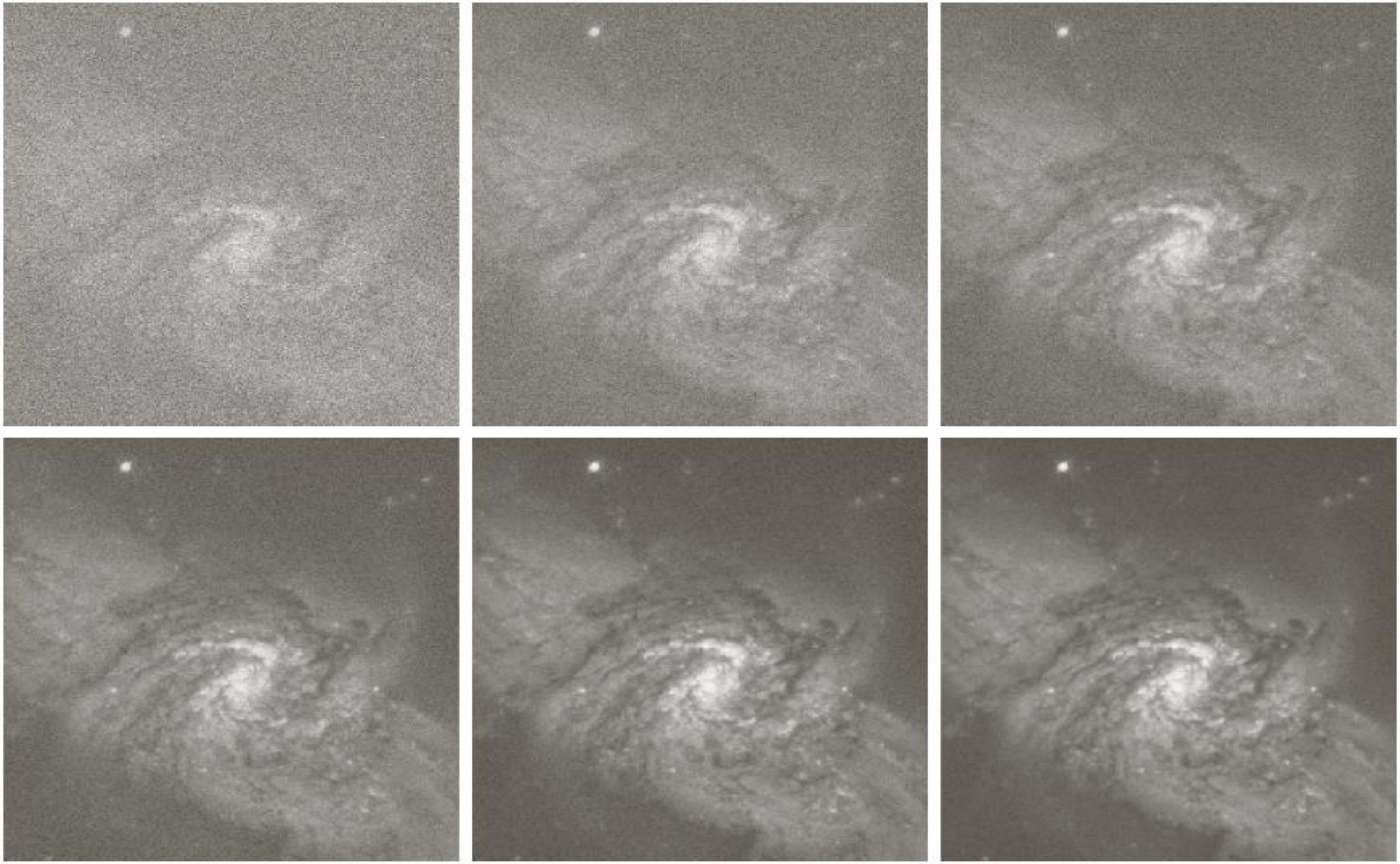
$$= f(x,y)$$

$$\sigma^2_{\bar{g}(x,y)} = \sigma^2_{\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)}$$

$$= \sigma^2_{\frac{1}{K}\sum_{i=1}^{K} n_i(x,y)} = \frac{1}{K}\sigma^2_{n(x,y)}$$

# Example: Addition of Noisy Images for Noise Reduction

- Application of image averaging is visible in the field of **astronomy**, where imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis.
- Figure 2.26(a) shows an 8-bit image in which corruption was simulated by adding to it Gaussian noise with zero mean and a standard deviation of 64 intensity levels. This image, typical of noisy images taken under low light conditions, is useless for all practical purposes.
- Addition is a discrete version of continuous integration. In astronomical observations, similar sensors for noise reduction by observing the **same scene** over long periods of time. Image averaging is then used to reduce the noise.



| | | |
|---|---|---|
| a | b | c |
| d | e | f |

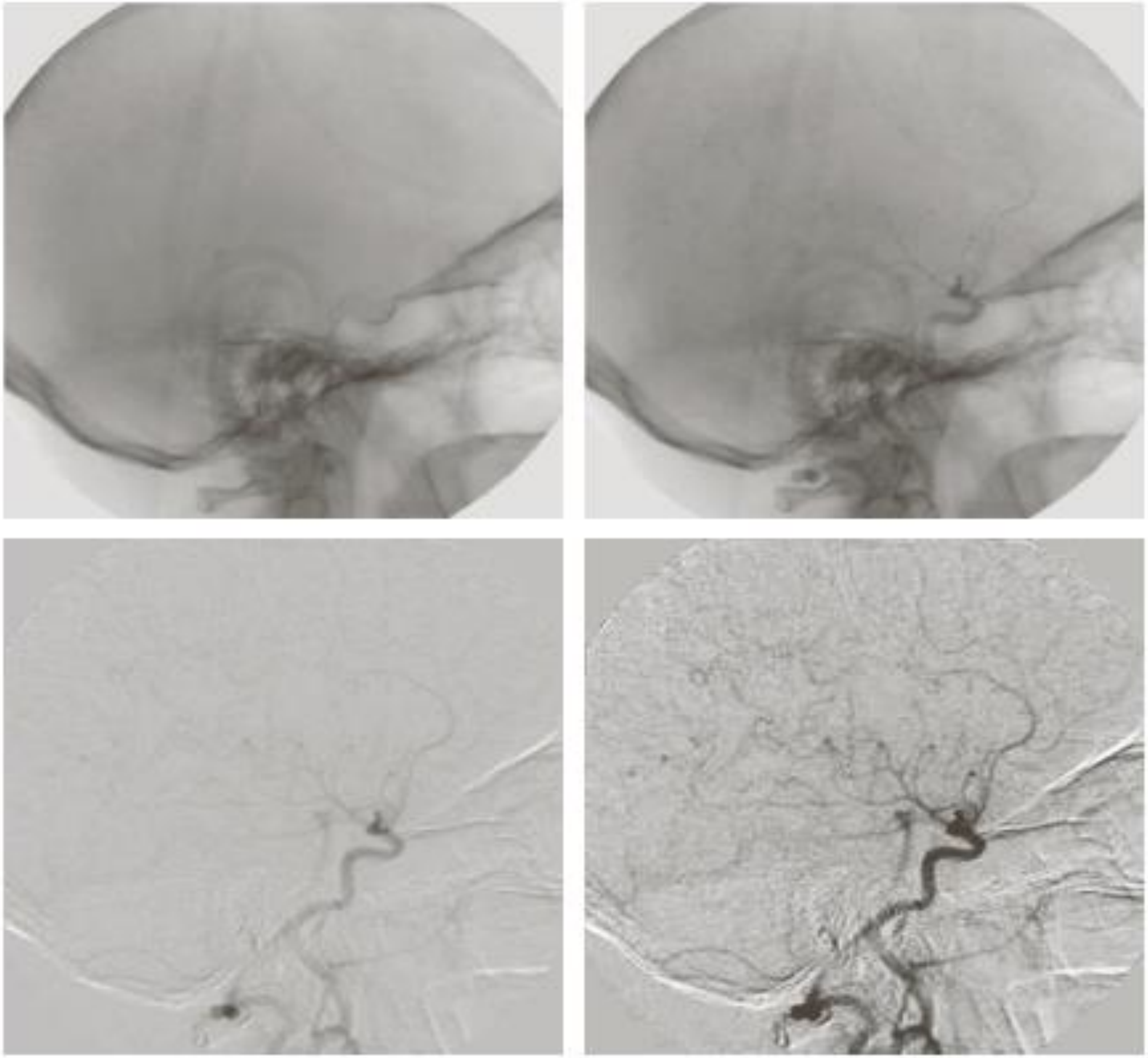
**FIGURE 2.26** (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

# Image Subtraction
## Example : Mask Mode Radiography

- A frequent application of image subtraction is in the enhancement of *differences between images.*
- **Mask** $h(x, y)$ **:** an X-ray image of a region of a patient's body
- **Live images** $f(x, y)$ **:** X-ray images captured at TV rates after injection of the contrast medium
- **Enhanced detail** $g(x,y)$

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

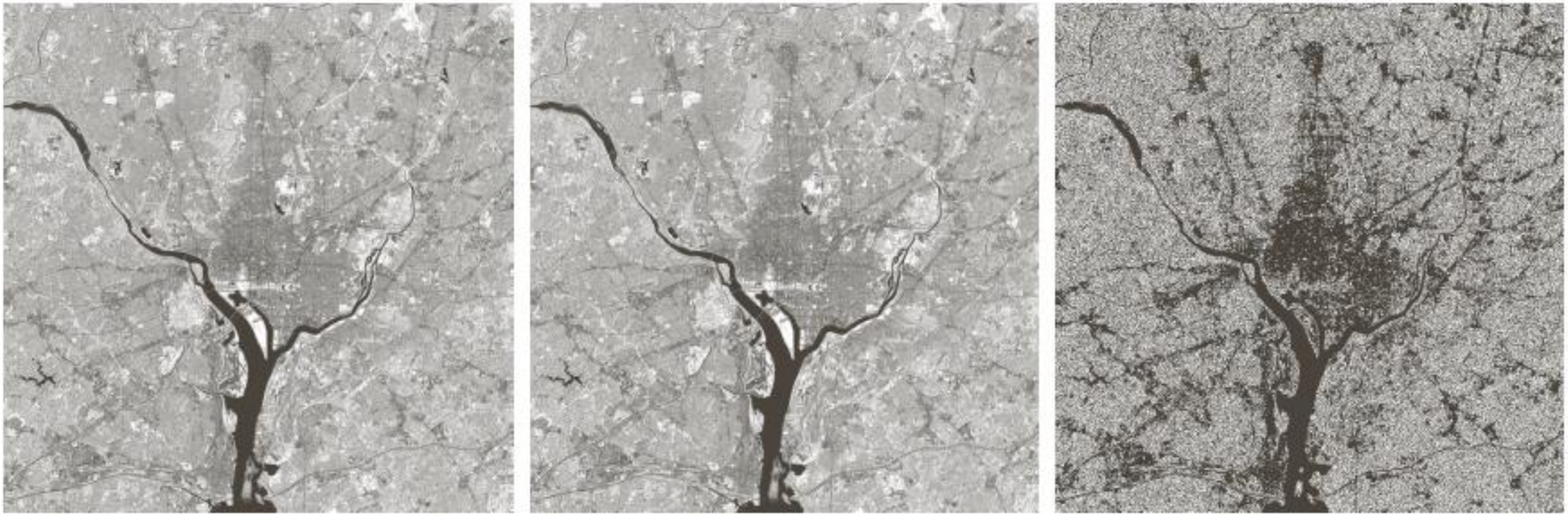
- **The procedure gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.**



| | |
|---|---|
| a | b |
| c | d |

**FIGURE 2.28** Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

# Example : Image Subtraction



a b c

**FIGURE 2.27** (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

- Visually, images (a) & (b) are indistinguishable. Fig. (c) clearly shows their differences.
- Black (0) values in this difference image indicate locations where there is no difference between the images in Figs. (a) and (b).

# Image Multiplication

- An important application of image multiplication (and division) is *shading correction.*
- Another common use of image multiplication is in *masking, also called region of interest (ROI), operations.*
- *The process, illustrated in Fig. 2.30, consists* simply of multiplying a given image by a mask image that has 1s in the ROI and 0s elsewhere.
- There can be more than one ROI in the mask image, and the shape of the ROI can be arbitrary, although rectangular shapes are used frequently for ease of implementation.

# Image Multiplication

a b c

**FIGURE 2.30** (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

# An Example of Image Multiplication

a b c

**FIGURE 2.29** Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

# Set and Logical Operations

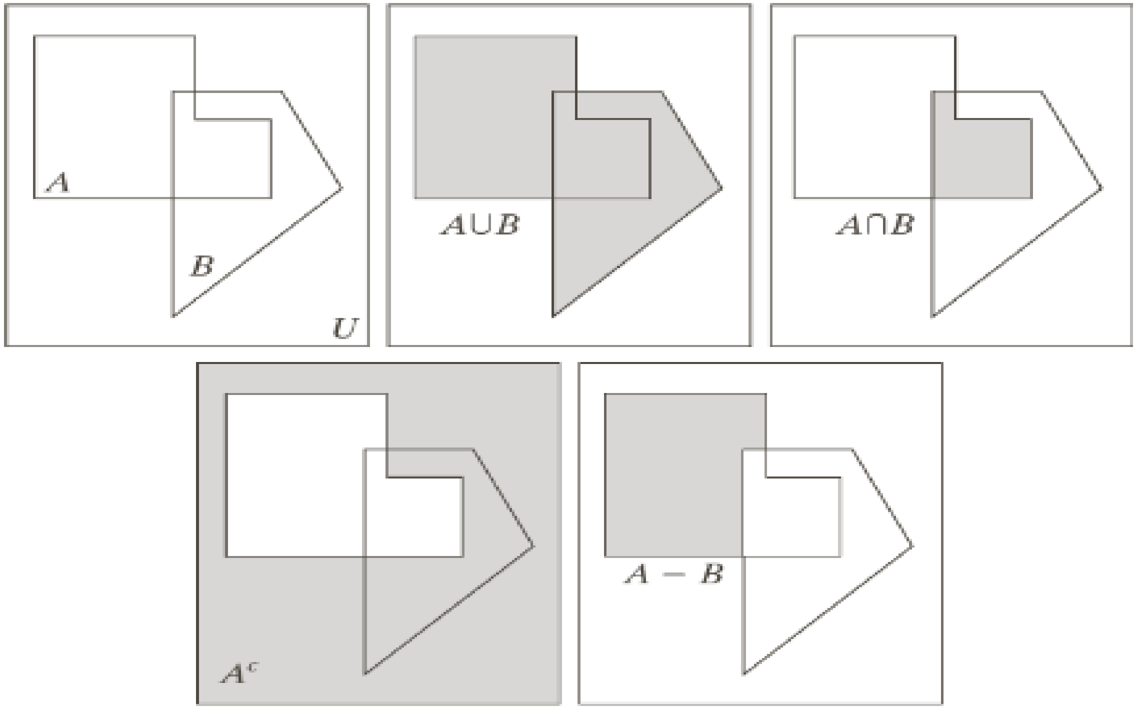



**FIGURE 2.31** (a) Two sets of coordinates, $A$ and $B$, in 2-D space. (b) The union of $A$ and $B$. (c) The intersection of $A$ and $B$. (d) The complement of $A$. (e) The difference between $A$ and $B$. In (b)–(e) the shaded areas represent the member of the set operation indicated.

$$A \cup B = \{w \mid w \in A \text{ OR } w \in B\}$$

$$A \cap B = \{w \mid w \in A \text{ AND } w \in B\}$$

$$A - B = \{w \mid w \in A,\ w \notin B\} = A \cap B^c$$

$$A^c = \{w \mid w \notin A\}$$

# Set and Logical Operations

- Let A be the elements of a gray-scale image
- The elements of A are triplets of the form (x, y, z), where x and y are **spatial coordinates** and z denotes the **intensity** at the point (x, y).

$$A = \{(x, y, z) \mid z = f(x, y)\}$$

- The complement of A is denoted $A^c$ which simply denotes the set of pixels of *A, whose intensities have been subtracted* from a constant *K.*

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

$$K = 2^k - 1; k \text{ is the number of intensity bits used to represent } z$$

# Set and Logical Operations

- Let A denote the 8-bit gray-scale image in Fig. 2.32(a), and suppose that we want to form the negative of *A using set* operations.

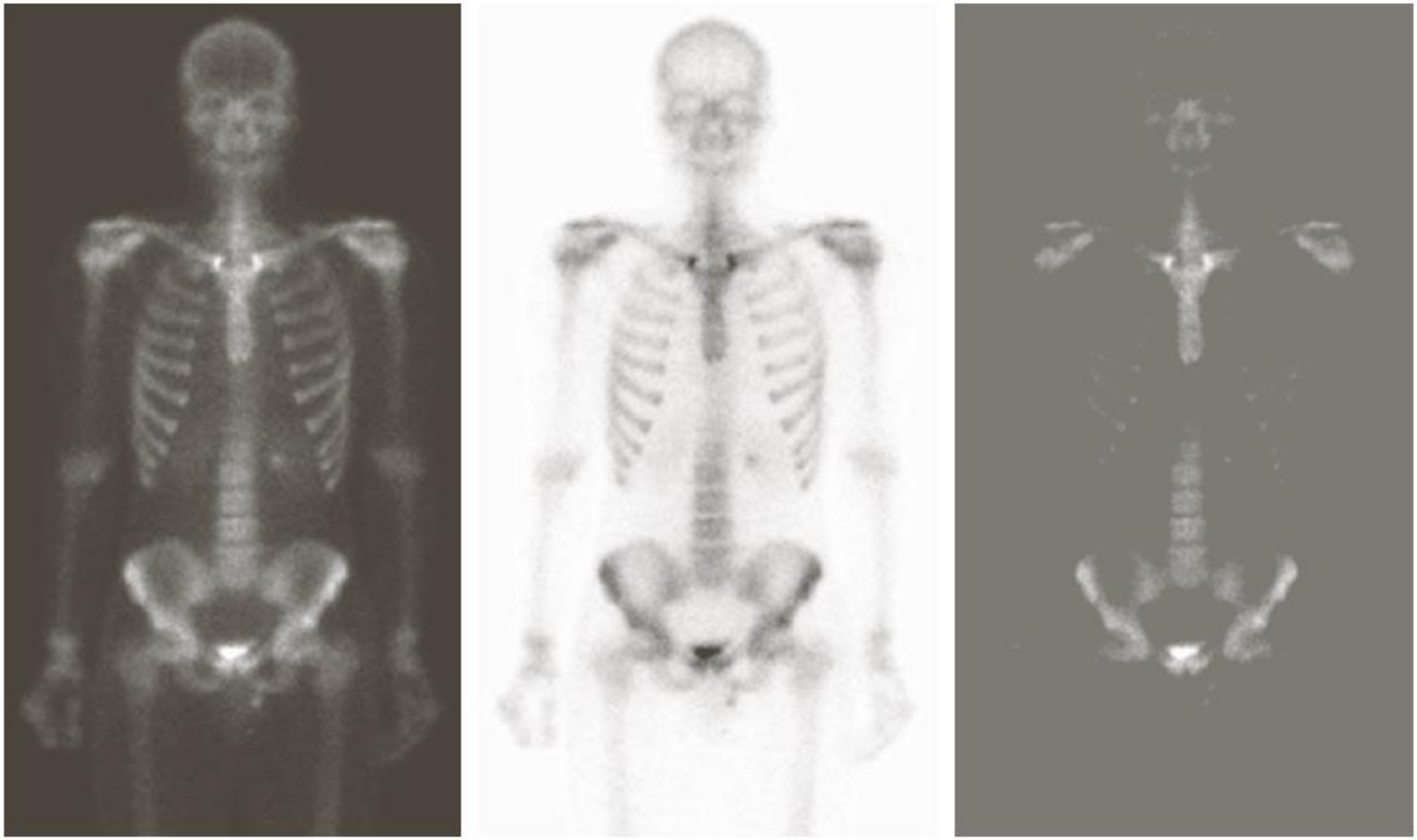
$$A_n = A^c = \left\{ (x, y, 255 - z) \,|\, (x, y, z) \in A \right\}$$

- Coordinates are carried over, so is an image $A_n$ of the same size as *A.*

- The union of two gray-scale images (sets) A and B is defined as the set

$$A \cup B = \{ \max_z (a, b) \,|\, a \in A, b \in B \}$$

# Cont..



a b c

**FIGURE 2.32** Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

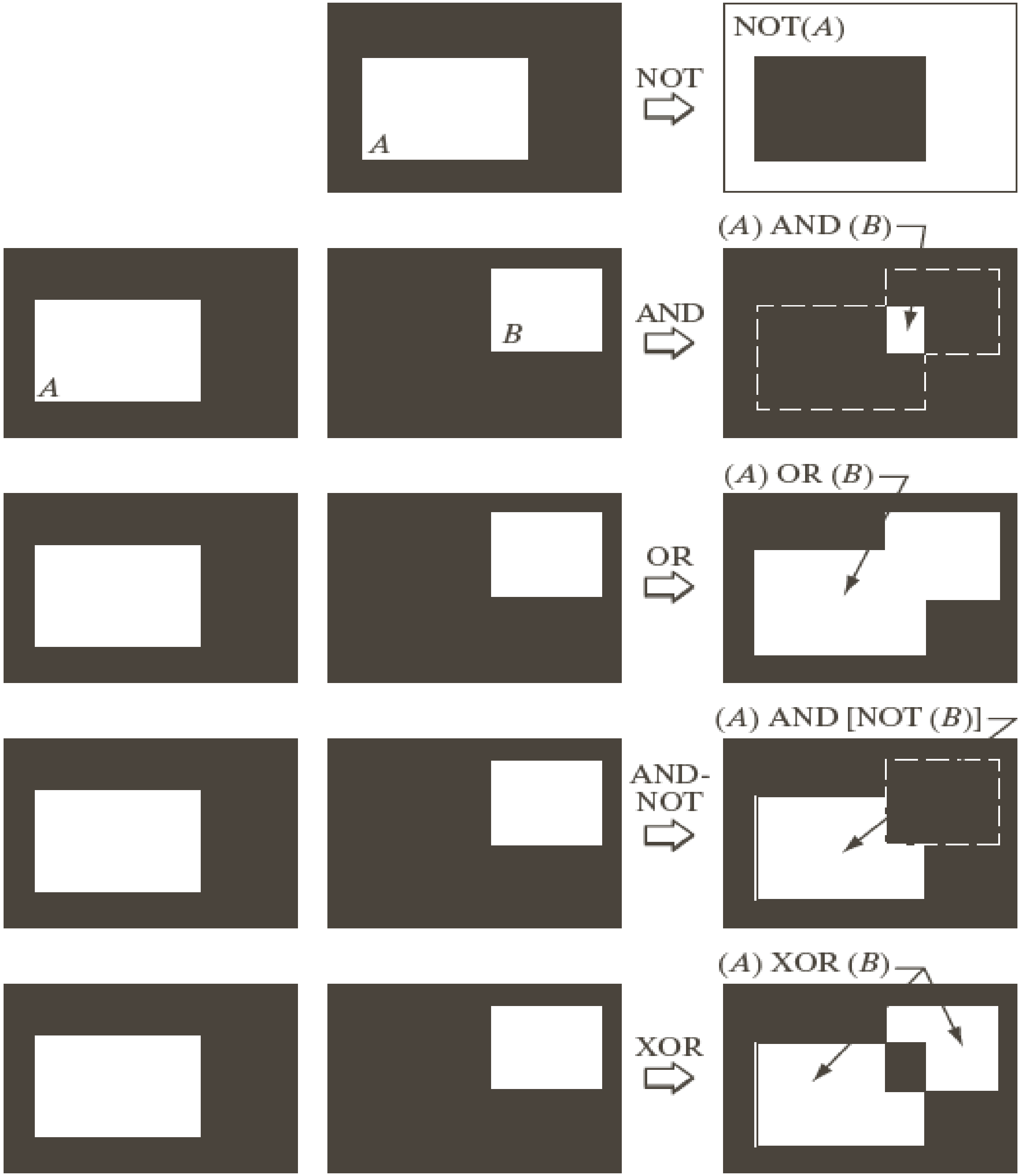



**FIGURE 2.33** Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

# Fuzzy Set

- The word "fuzzy" means "vagueness". Fuzziness occurs when the boundary of a piece of information is not clear-cut.

- Fuzzy sets have been introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set.

- Classical set theory allows the membership of the elements in the set in binary terms, a bivalent condition - an element either belongs or does not belong to the set.

  Fuzzy set theory permits the gradual assessment of the membership of elements in a set, described with the aid of a membership function valued in the real unit interval [0, 1].

# Cont..

- Suppose that we define any person as young of age 20 or younger. We see an immediate difficulty. A person whose age is 20 years and 1 sec would not be a member of the set of young people. This limitation arises regardless of the age threshold we use to classify a person as being young. What we need is more flexibility in what we mean by “young,” that is, we need a gradual transition from young to not young.

- The theory of fuzzy sets implements this concept by utilizing membership functions that are gradual between the limit values of 1 (definitely young) to 0 (definitely not young).Using fuzzy sets, we can make a statement such as a person being 50% young (in the middle of the transition between young and not young).In other words, age is an imprecise concept, and fuzzy logic provides the tools to deal with such concepts

# Cont..

- **Example:**

  Words like young, tall, good, or high are fuzzy.

  - There is no single quantitative value which defines the term young.
  - For some people, age 25 is young, and for others, age 35 is young.
  - The concept young has no clean boundary.
  - Age 1 is definitely young and age 100 is definitely not young;
  - Age 35 has some possibility of being young and usually depends on the context in which it is being considered.

# Spatial Operations

- Spatial operations are performed directly on the pixels of a given image. We classify spatial operations into three broad categories:
    1. single-pixel operations,
    2. neighborhood operations, and
    3. geometric spatial transformations.

**1. Single-pixel operations**

- Simplest operation we perform on a digital image is to alter the values of its individual pixels based on their intensity.
- Alter the values of an image's pixels based on the intensity.

# Cont..

- This type of process may be expressed as a transformation function, *T as* $$s = T(z)$$
- where *z is the intensity of a pixel in the original image and s is the (mapped)* intensity of the corresponding pixel in the processed image.

e.g.,

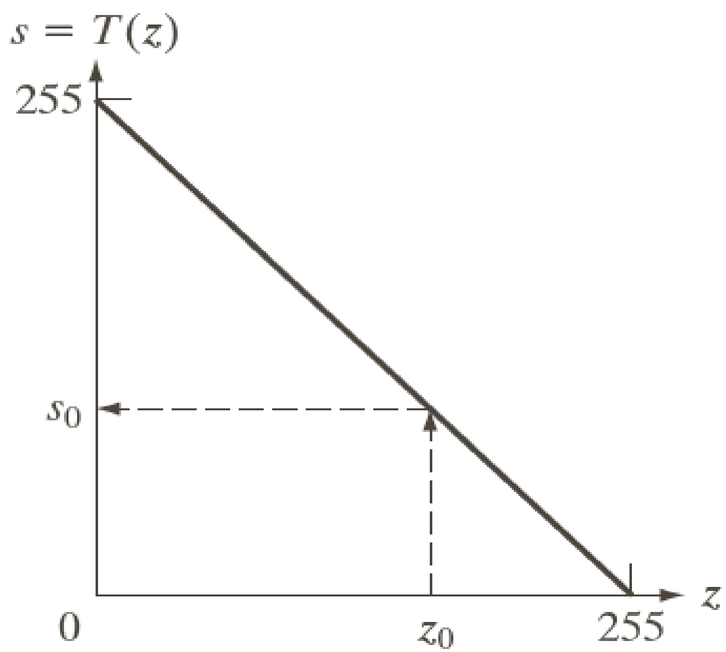



**FIGURE 2.34** Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value $z_0$ into its corresponding output value $s_0$.

# Cont..

## 2. Neighborhood operations

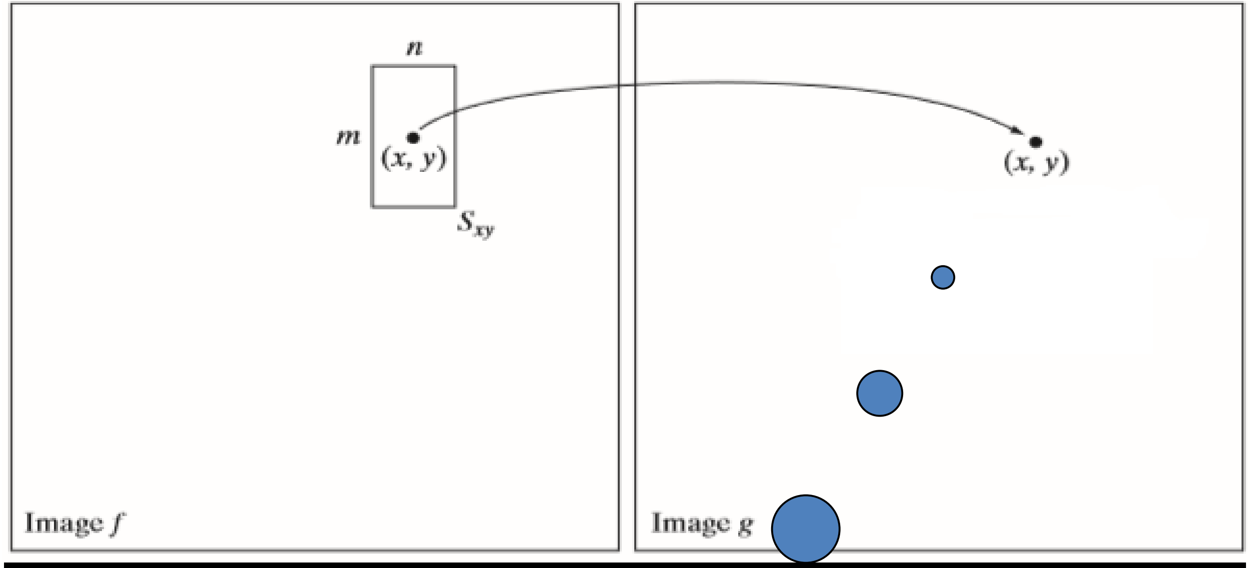
- Let $S_{xy}$ denote the set of coordinates of a neighborhood centered on an arbitrary point **(*x, y*)** *in an image **f**.*
- ***Neighborhood processing*** *generates a corresponding* pixel at the same coordinates in an output (processed) image ***g**, such* that the value of that pixel is determined by a specified operation involving the pixels in the input image with coordinates in $S_{xy}$
- **Example :** suppose that the specified operation is to compute the average value of the pixels in a rectangular neighborhood of size *m * n* centered on **(*x, y*).** *The locations of pixels* in this region constitute the set $S_{xy}$ Figures 2.35(a) and (b) illustrate the process. We can express this operation in equation form as

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

- **Neighborhood operations**

| a | b |
| --- | --- |
| c | d |

**FIGURE 2.35** Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790 × 686 pixels.

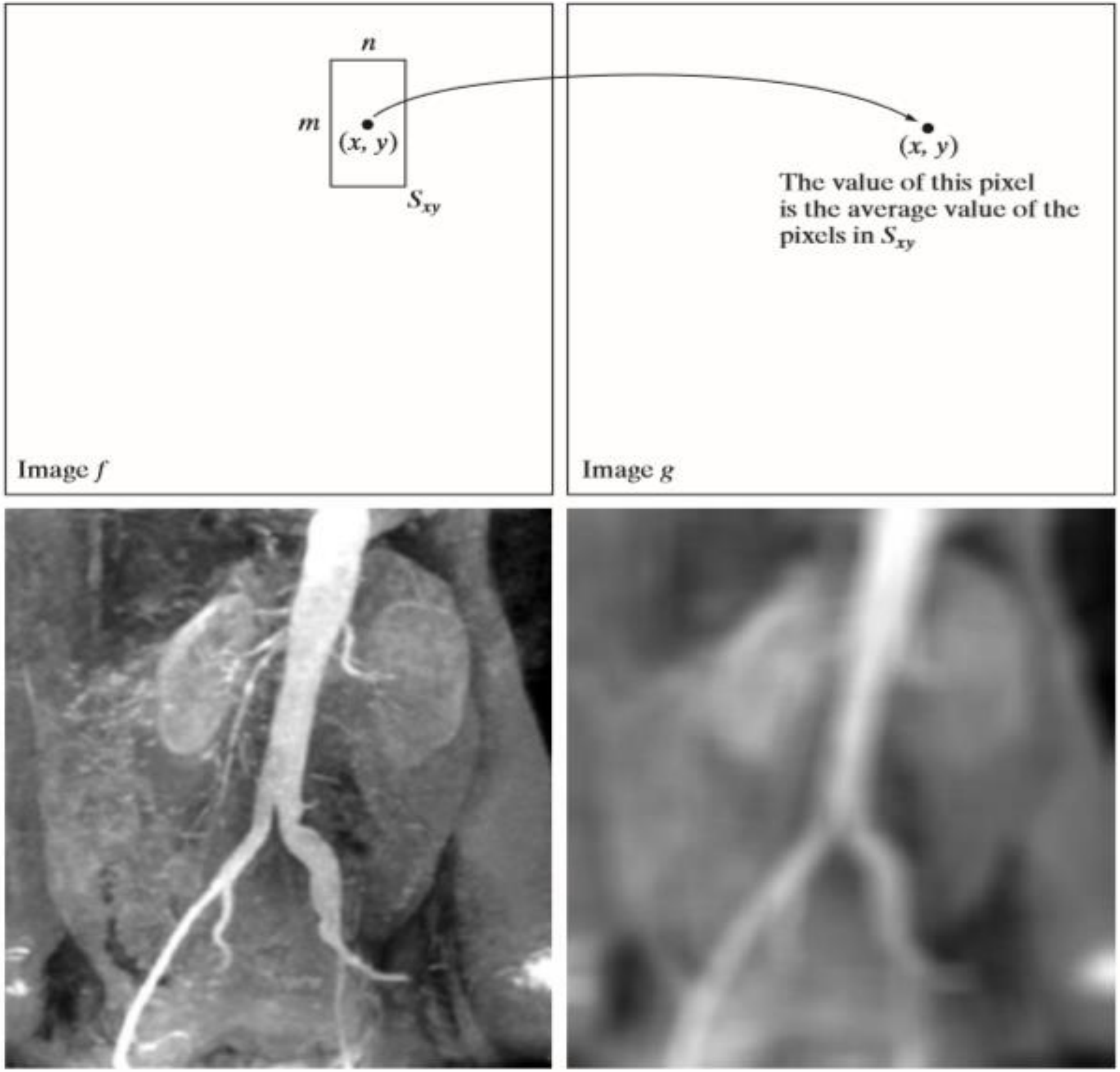



The value of this pixel is determined by a specified operation involving the pixels in the input image with coordinates in $S_{xy}$

- **Neighborhood operations**

a b
c d

**FIGURE 2.35** Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size $790 \times 686$ pixels.



# Geometric Spatial Transformations

- It modify the spatial relationship between pixels in an image.
- These transformations often are called rubber-sheet transformations because they may be viewed as analogous to "printing" an image on a sheet of rubber and then stretching the sheet according to a predefined set of rules.
- In terms of DIP, a GT consists of two basic operations:
  1. a spatial transformation of coordinates and
  2. intensity interpolation that assigns intensity values to the spatially transformed pixels.
- The transformation of coordinates may be expressed as

$$(x, y) = T\{(v, w)\}$$

- where are pixel coordinates in the original image and (x, y) are the corresponding pixel coordinates in the transformed image.

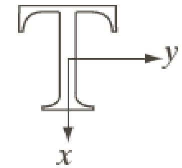
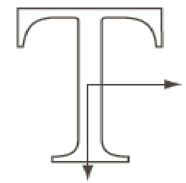
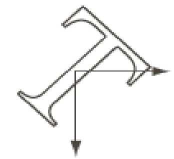
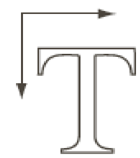
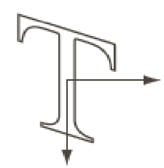
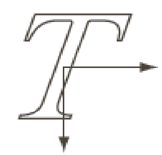
# Geometric Spatial Transformations

- **Example:** Transformation shrinks the original image to half its size in both spatial directions.
- One of the most commonly used spatial coordinate transformations is the affine transform(Wolberg [1990]), which has the general form
- **Affine transform:** A common geometric transformation is the affine transform.

$$[x \quad y \quad 1] = [v \quad w \quad 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

- It may translate, rotate, scale and sheer an image depending on the value of the elements of T
- To avoid empty pixels we implement the inverse mapping
- Interpolation is essential

**TABLE 2.2**
Affine transformations based on Eq. (2.6.–23).

| Transformation Name | Affine Matrix, T | Coordinate Equations | Example |
|---|---|---|---|
| Identity | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = w$ |  |
| Scaling | $\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = c_x v$<br>$y = c_y w$ |  |
| Rotation | $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v\cos\theta - w\sin\theta$<br>$y = v\cos\theta + w\sin\theta$ |  |
| Translation | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ | $x = v + t_x$<br>$y = w + t_y$ |  |
| Shear (vertical) | $\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v + s_v w$<br>$y = w$ |  |
| Shear (horizontal) | $\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = s_h v + w$ |  |

# Intensity Assignment

- **Forward Mapping**

$$(x, y) = T\{(v, w)\}$$

It's possible that two or more pixels can be transformed to the same location in the output image.

- **Inverse Mapping**

$$(v, w) = T^{-1}\{(x, y)\}$$

The nearest input pixels to determine the intensity of the output pixel value.

Inverse mappings are more efficient to implement than forward mappings.

# Example: Image Rotation and Intensity Interpolation

The effects and importance of interpolation in image transformations

a b c d

**FIGURE 2.36** (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

# Image Registration

- Estimate the transformation parameters between two images.
- Very important application of digital image processing.
  - Single and multimodal
  - Temporal evolution and quantitative analysis (medicine, satellite images)
- A basic approach is to use control points (user defined or automatically detected) and estimate the elements of the transformation matrix by solving a linear system.

# Image Registration

- Input and output images are available but the transformation function is unknown.
- **Goal:** estimate the transformation function and use it to register the two images.

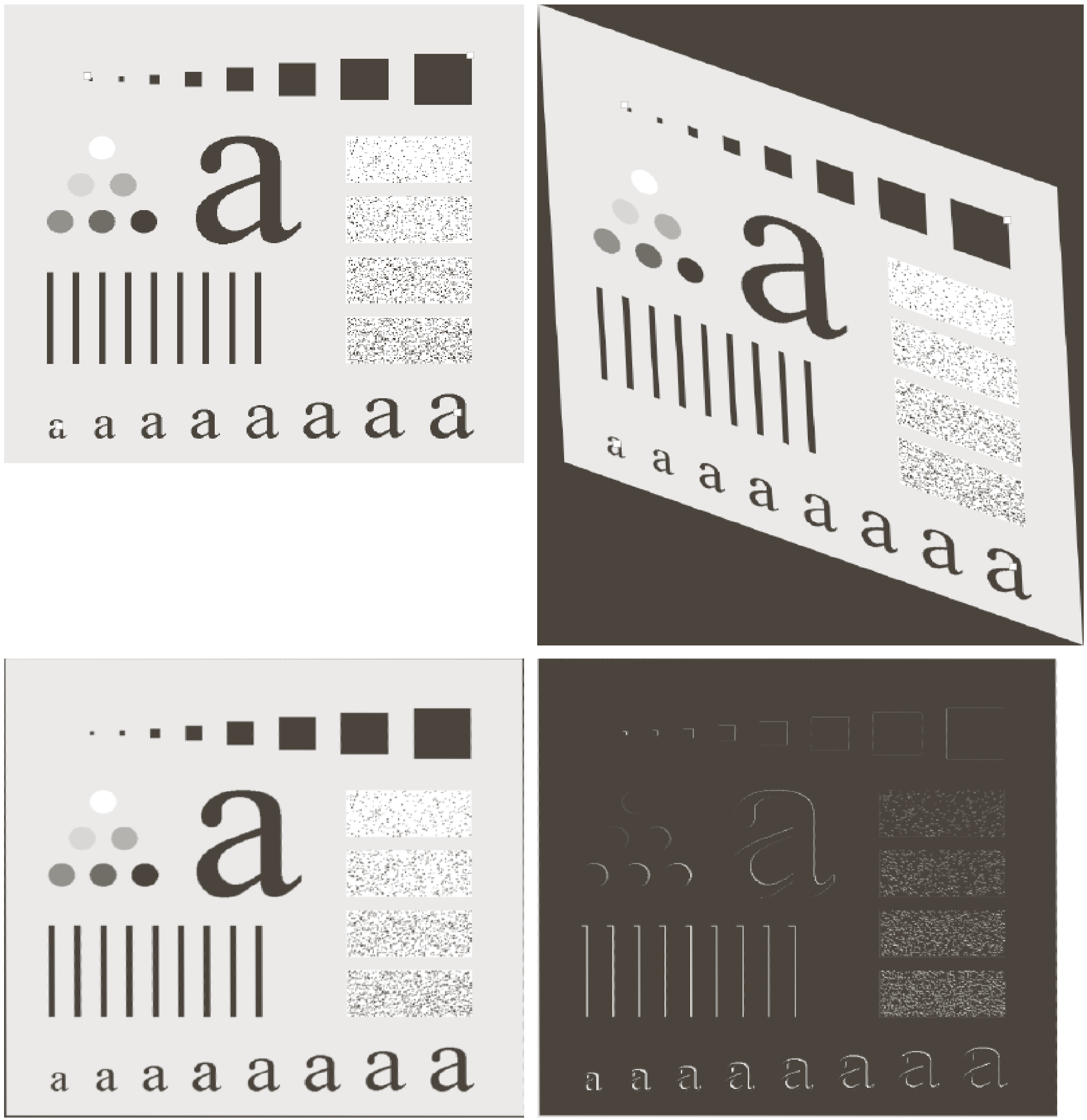
- One of the principal approaches for image registration is to use ***tie points*** (also called ***control points***)
- The corresponding points are known precisely in the input and output (**reference**) images.

# Cont..

- A simple model based on bilinear approximation:

$$\begin{cases} x = c_1 v + c_2 w + c_3 vw + c_4 \\ y = c_5 v + c_6 w + c_7 vw + c_8 \end{cases}$$

Where $(v, w)$ and $(x, y)$ are the coordinates of tie points in the input and reference images.



a b
c d

**FIGURE 2.37** Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

# Image Transform

- A particularly important class of 2-D linear transforms, denoted T(u, v)

$$T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)r(x,y,u,v)$$

where $f(x, y)$ is the input image,

$r(x, y, u, v)$ is the *forward transformation kernel*,

variables $u$ and $v$ are the transform variables,

$u$ = 0, 1, 2, ..., M-1 and $v$ = 0, 1, ..., N-1.

# Cont..

- Given T(u, v), the original image f(x, y) can be recovered using the inverse transformation of T(u, v).

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v)$$

where $s(x, y, u, v)$ is the *inverse transformation kernel*,

$x$ = 0, 1, 2, ..., M-1 and $y$ = 0, 1, ..., N-1.

# Cont..


**FIGURE 2.39**
General approach for operating in the linear transform domain.

# Example: Image De-noising by Using DCT

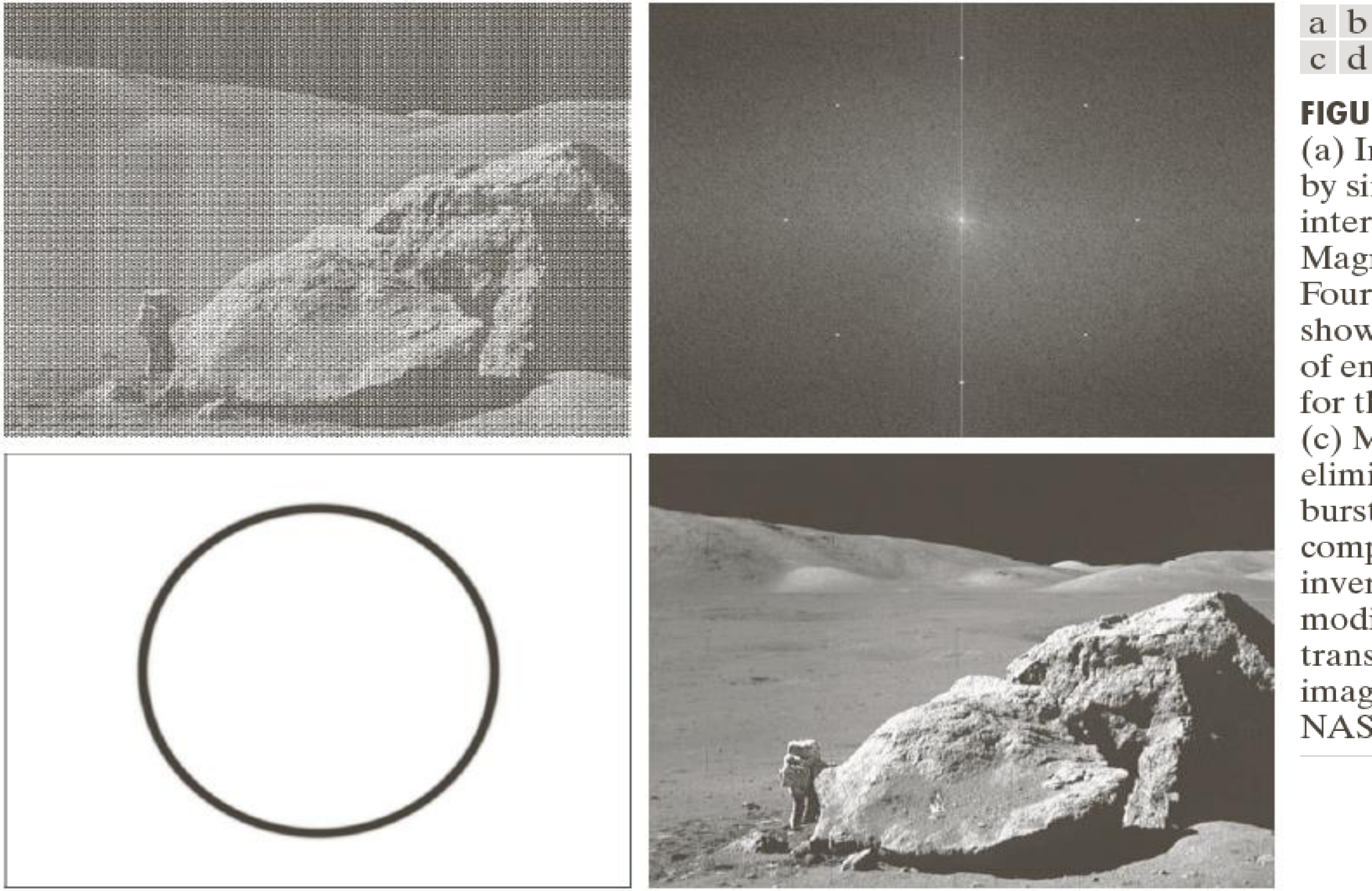



**FIGURE 2.40** (a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

# Forward Transform Kernel

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v)$$

The kernel $r(x,y,u,v)$ is said to be SEPERABLE if

$$r(x,y,u,v) = r_1(x,u) r_2(y,v)$$

In addition, the kernel is said to be SYMMETRIC if

$r_1(x,u)$ is functionally equal to $r_2(y,v)$, so that

$$r(x,y,u,v) = r_1(x,u) r_1(y,u)$$

# The Kernels for 2-D Fourier Transform

The *forward* kernel

$$r(x, y, u, v) = e^{-j2\pi(ux/M + vy/N)}$$

Where $j = \sqrt{-1}$

The *inverse* kernel

$$s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M + vy/N)}$$

# 2-D Fourier Transform

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) e^{j2\pi(ux/M + vy/N)}$$

# Probabilistic Methods

Let $z_i$, $i = 0, 1, 2, \ldots, L-1$, denote the values of all possible intensities in an $M \times N$ digital image. The probability, $p(z_k)$, of intensity level $z_k$ occurring in a given image is estimated as

$$p(z_k) = \frac{n_k}{MN},$$

where $n_k$ is the number of times that intensity $z_k$ occurs in the image.

$$\sum_{k=0}^{L-1} p(z_k) = 1$$

The mean (average) intensity is given by

$$m = \sum_{k=0}^{L-1} z_k \, p(z_k)$$

# Cont..

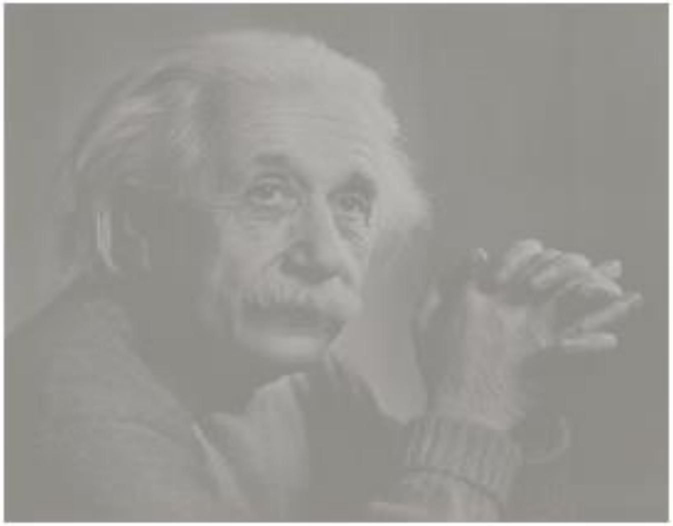
The variance of the intensities is given by

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 p(z_k)$$

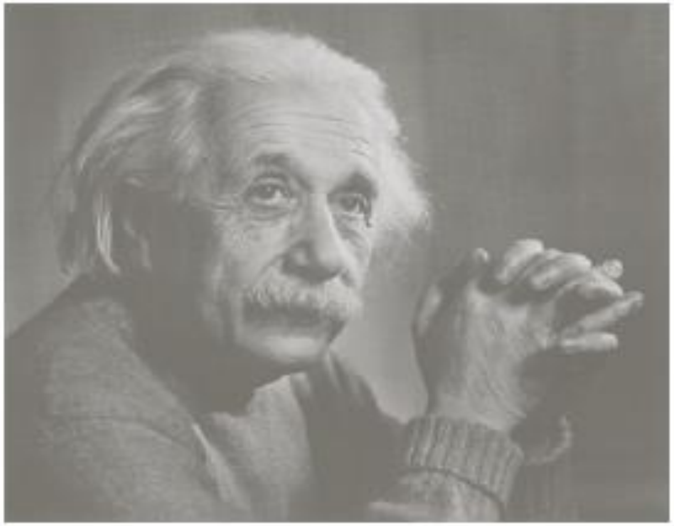
The $n^{\text{th}}$ moment of the intensity variable $z$ is

$$u_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n p(z_k)$$

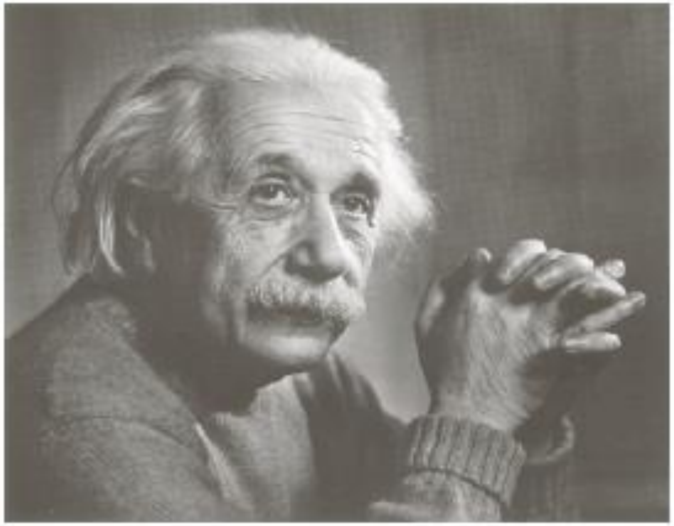
# Example: Comparison of Standard Deviation Values



$\sigma = 14.3$



$\sigma = 31.6$



$\sigma = 49.2$

# Simple intensity processing

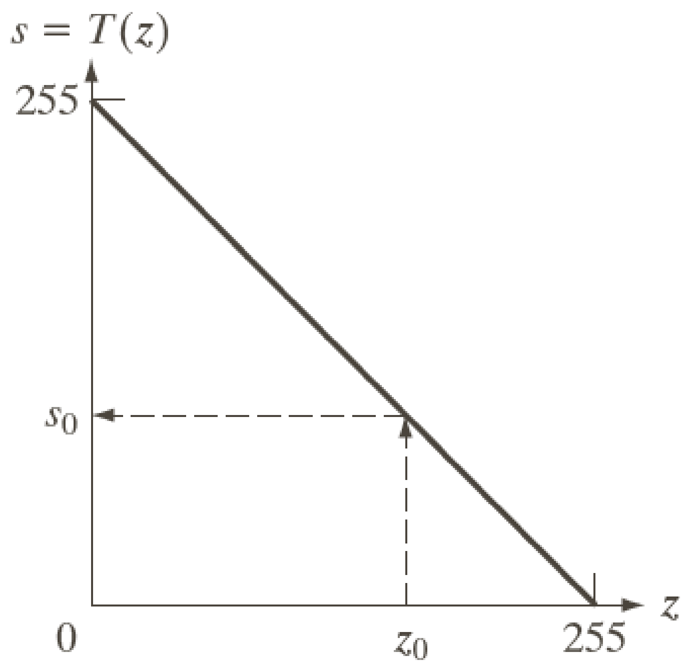



**FIGURE 2.34** Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value $z_0$ into its corresponding output value $s_0$.

# *Thank You*