Digital Image Processing

Dr.-Ing. Tuan Do-Hong

Department of Telecommunications Engineering
Faculty of Electrical and Electronics Engineering
HoChiMinh City University of Technology

E-mail: do-hong@hcmut.edu.vn

Outline

1. Digital Image Fundamentals
2. Image Enhancement and Restoration
3. Image Compression
4. Morphological Image Processing
5. Image Segmentation
6. Image Representation and Description
7. Introduction to Object (Pattern) Recognition

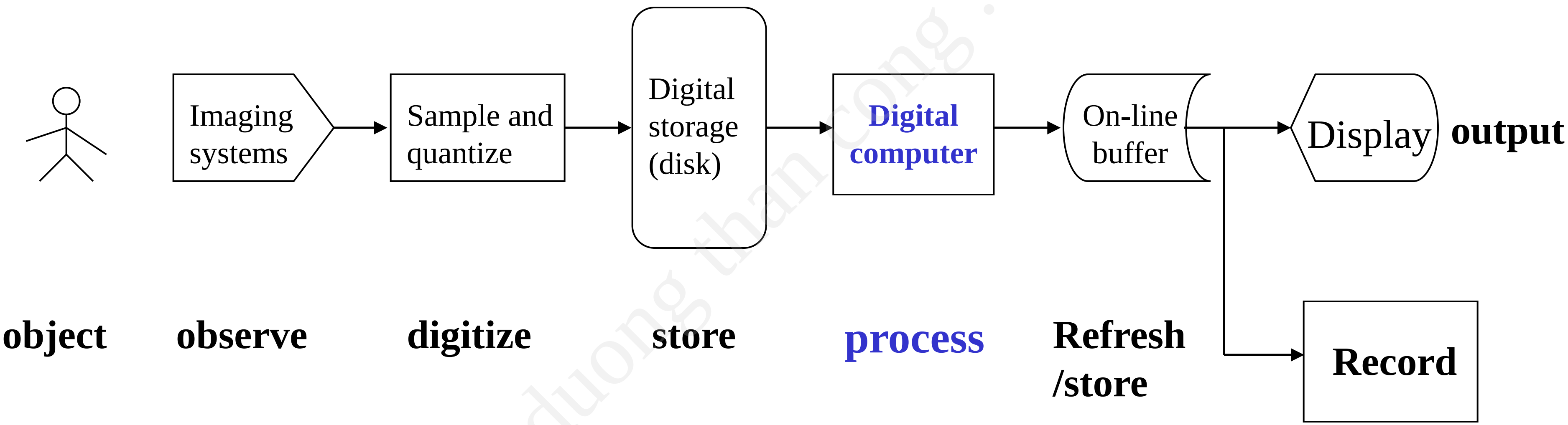
References

[1] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison Wesley, 1993.

Chapter 1: Digital Image Fundamentals

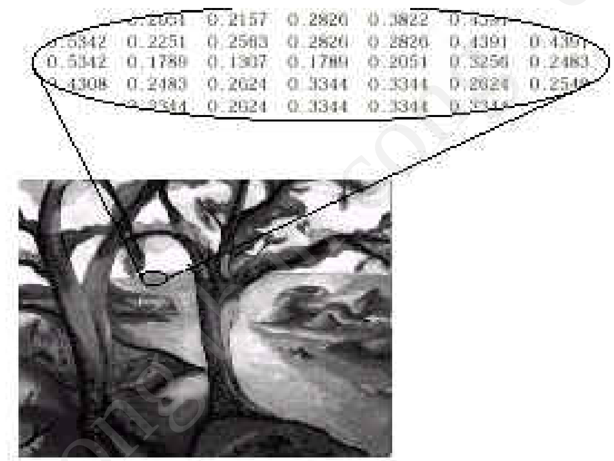
1. Digital Image

1. Analog to Digital (1)



1. Analog to Digital (2)

- Computers work with **numerical** (rather than **pictorial**) data.
- An image must be converted to numerical form before processing by computer.

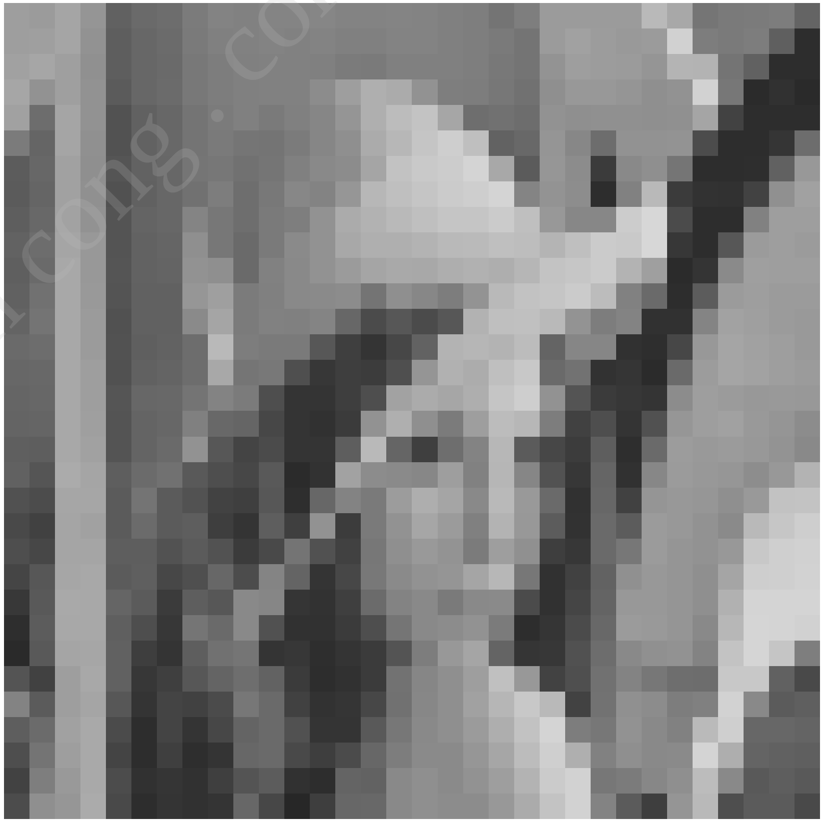


- Picture elements or pixels.
- Rectangular sampling grid.
- (Intensity) Brightness of the image.

1. Sampling (Digitization of Spatial Coordinates)

256×256

64×64



1. Quantization (Intensity Digitization) (1)

0	255	255	0	0
0	0	255	0	0
0	0	255	0	0
0	0	255	0	0
0	255	255	255	0

1. Quantization (2)

256×256, 256 gray-levels

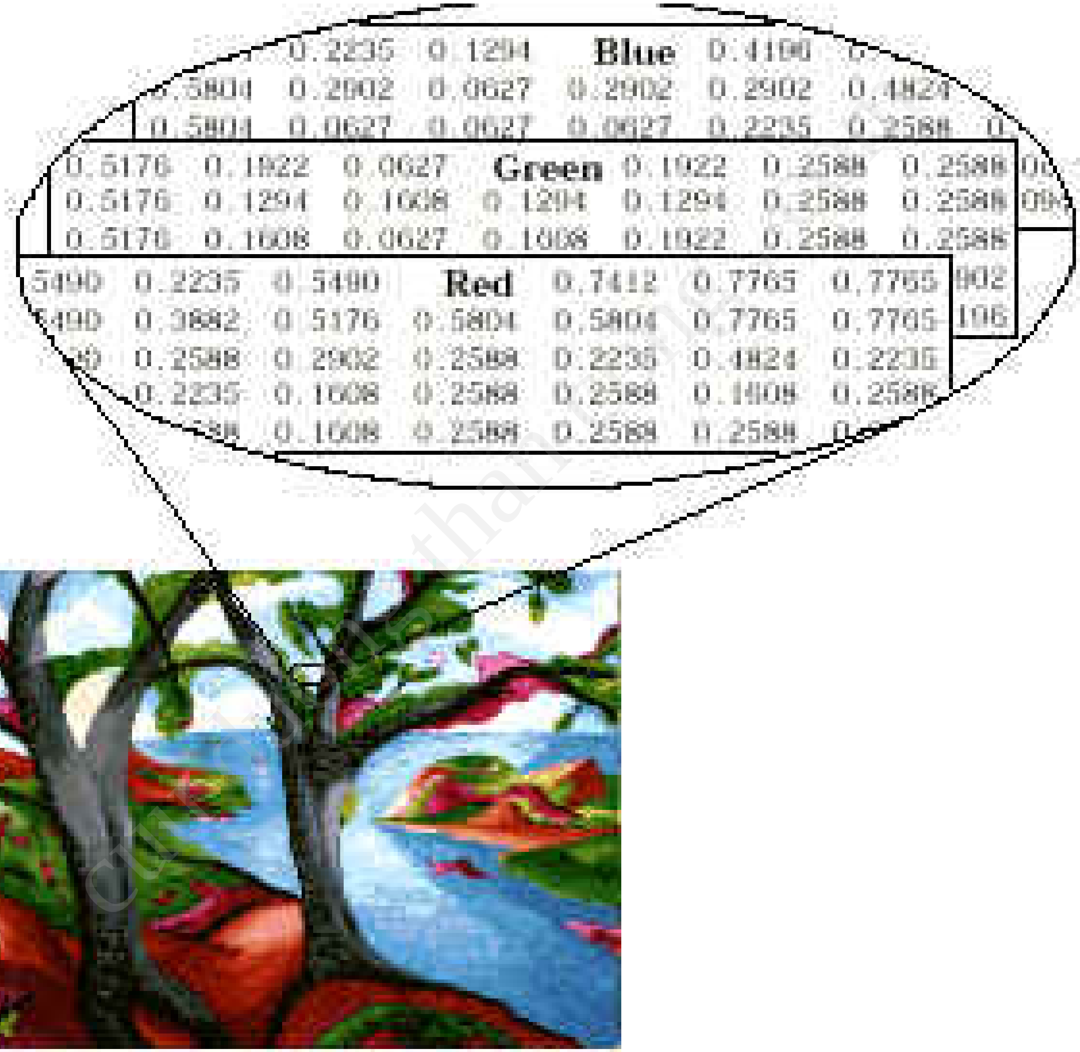
256×256, 32 gray-levels

1. Quantization (3)

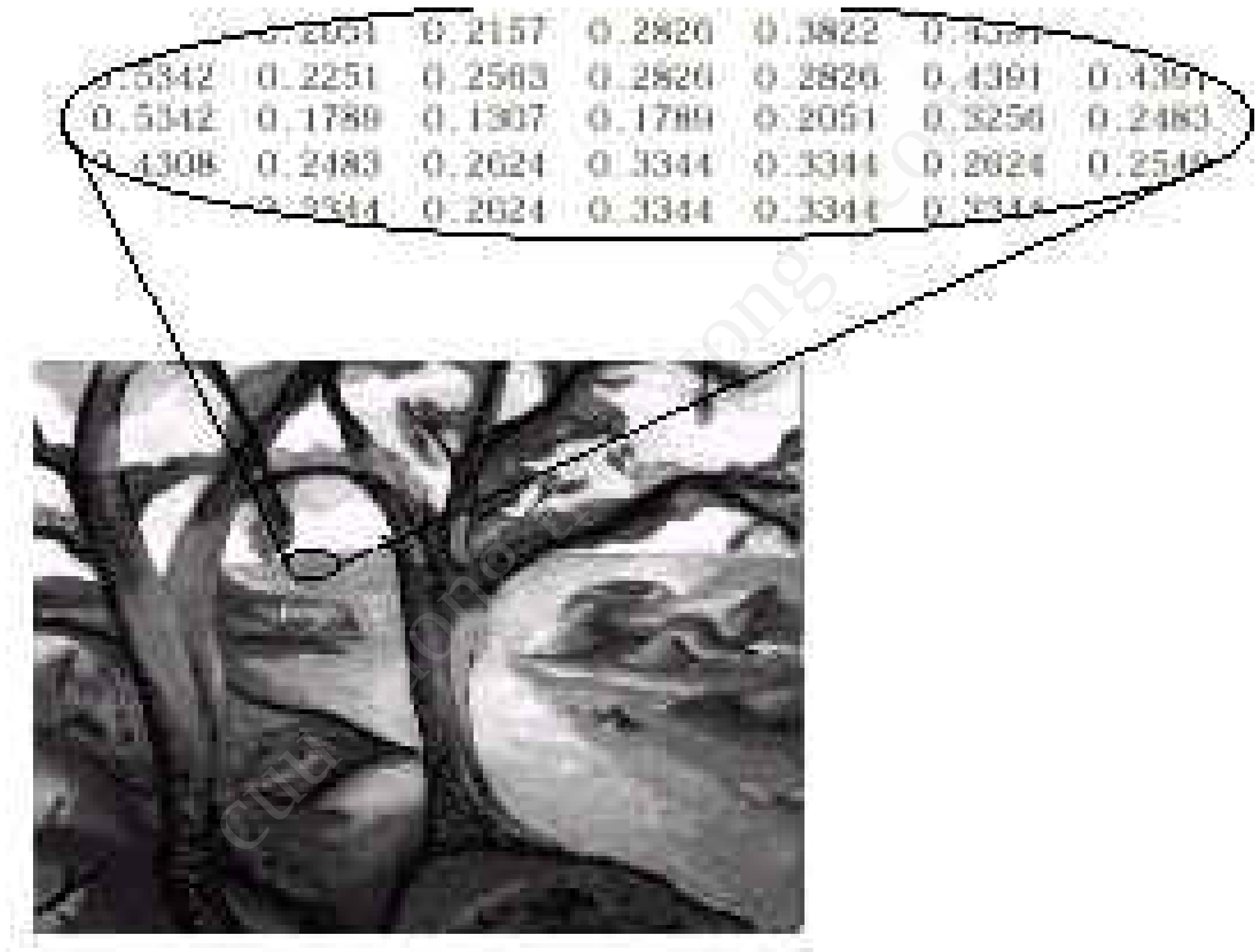
256×256, 256 gray-levels

256×256, 2 gray-levels

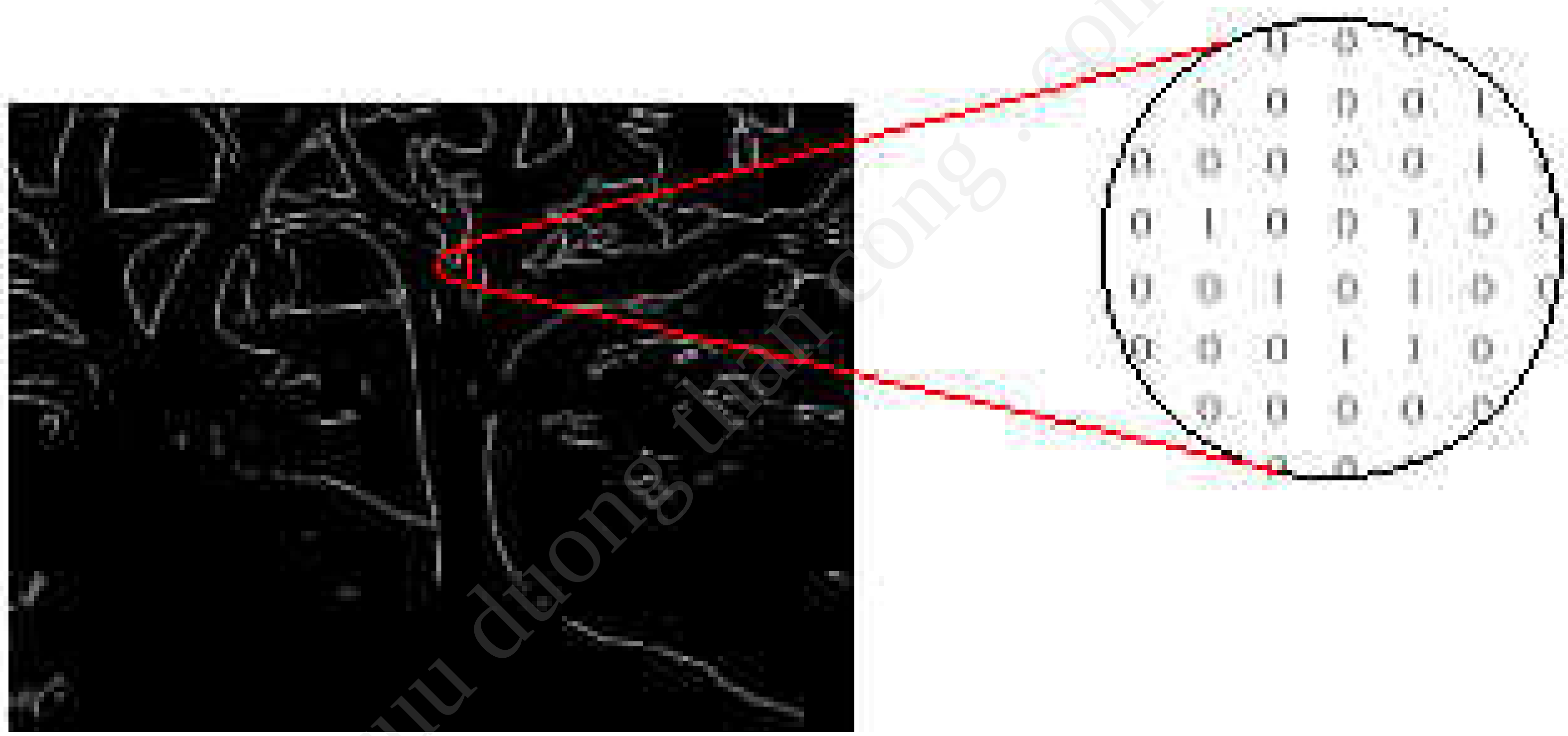
1. Colour Image



1. Gray-level Image



1. Binary Image



1. Digital Image Processing

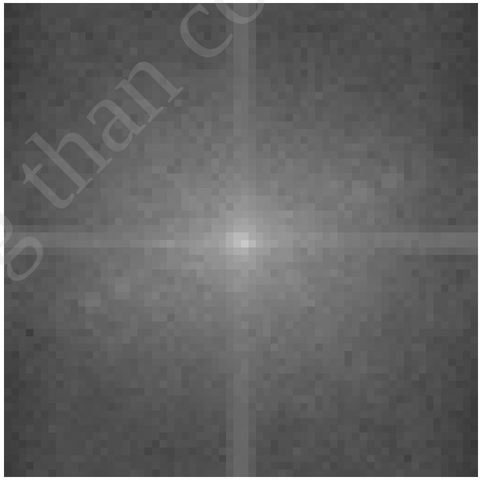
- **Manipulation of multidimensional signals**
  - Image (photo): $f(x, y)$
  - Video: $f(x, y, t)$
  - CT, MRI: $f(x, y, z, t)$
- **Coding/compression**
- **Enhancement, restoration, reconstruction**
- **Analysis, detection, recognition, understanding**
- **Visualization**

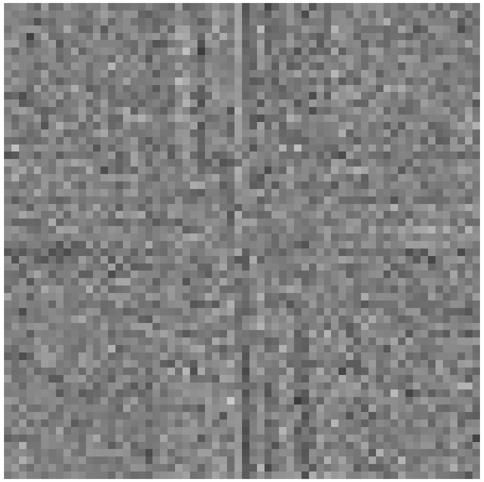
1. Image Transforms

Original Image

Fourier Transform

Magnitude **Phase**

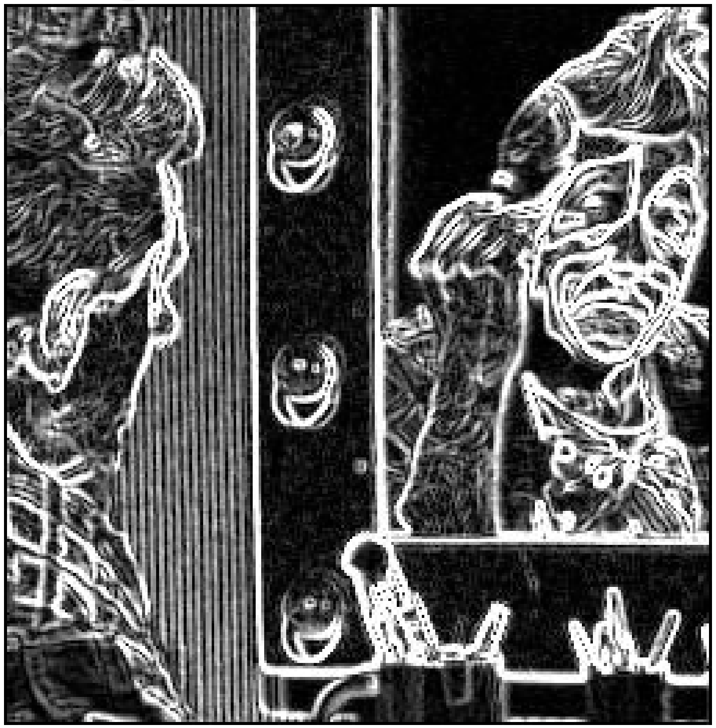




1. Image Enhancement

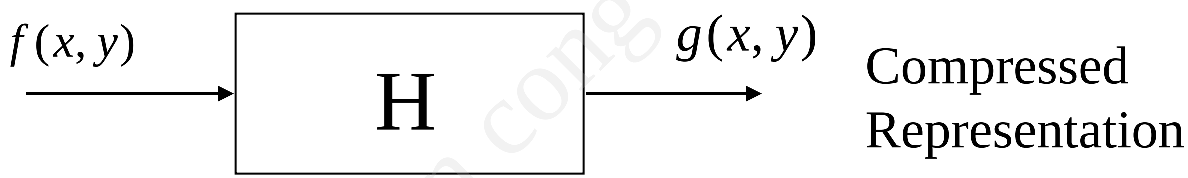
Original Image

High Pass Filtering

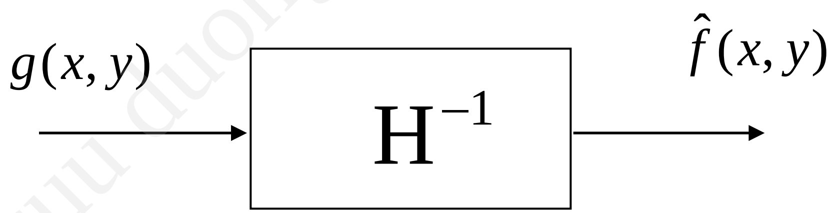


1. Image Compression

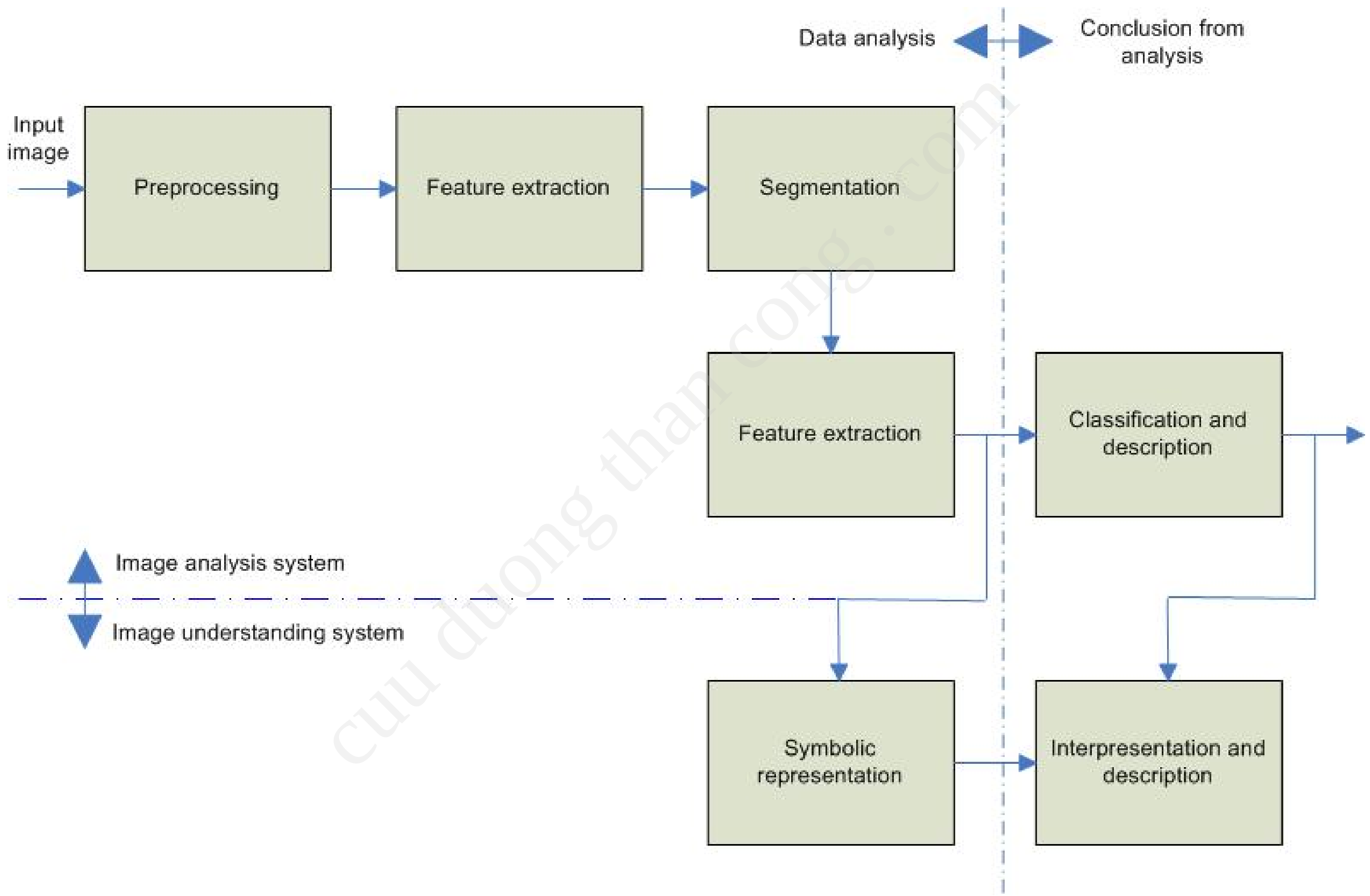
Encoder



Decoder



1. Image Analysis



1. Applications

- **Image processing** (medical image processing, satellite/astronomy image processing, radar image processing, etc.)

- **Image analysis - computer vision** (character/face/hand/gesture recognitions, content-based browsing and retrieval, medical image analysis, industrial automation, remote sensing, forensics, robotics, radar imaging, cartography, etc.)

- **Virtual reality** (applications in manufacturing, medicine, entertainment, etc.)

- **Multimedia communication/storage** (video processing, digital TV, video over networks, etc.)

1. Image Model

- **Image**: 2-D light-intensity function, $f(x,y)$: value of f at spatial coordinates (x,y) gives intensity of image at that point.

$$0 < f(x,y) < \infty \quad (2.1)$$

$$f(x,y) = i(x,y)r(x,y) \quad (2.2)$$

where $i(x,y)$: **illumination** (amount of source light incident on the scene), $r(x,y)$: **reflectance** (amount of light reflected by the objects in the scene).

$$0 < i(x,y) < \infty \quad (2.3)$$

$$0 < r(x,y) < 1$$

0 (total absorption), 1 (total reflectance)

- In digital image processing, f called **gray level** G, $L_{min} < G < L_{max}$.
  Interval $[L_{min}, L_{max}]$: **gray scale.**
  In practice, shifting gray scale to $[0, L]$, $G = 0$: black, $G = L$: white.

1. Image Sampling & Quantization (1)

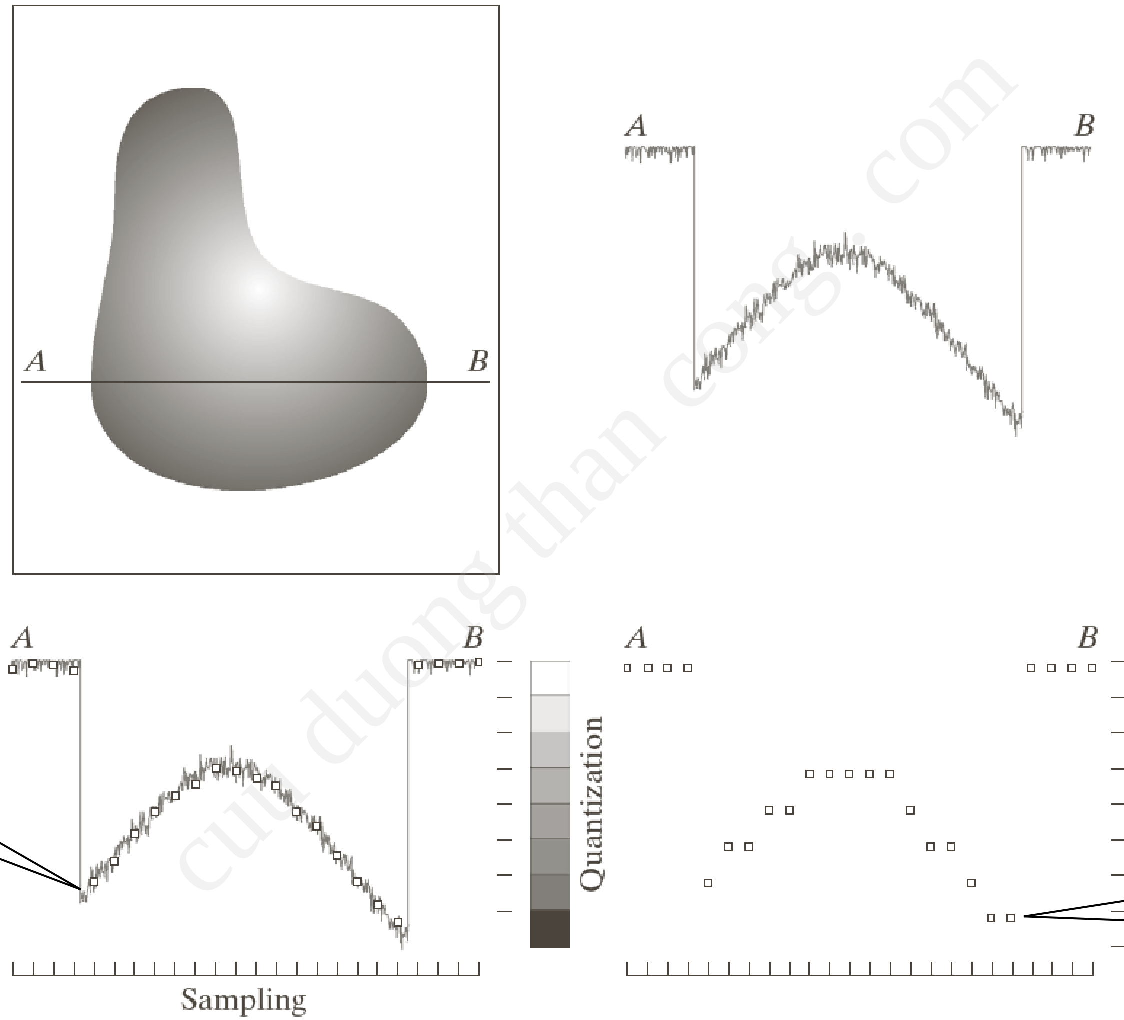
- $f(x,y)$ approximated by equally spaced samples in the form of $(N\times M)$-matrix:

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \vdots & \vdots & & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix} = \mathbf{A} \qquad (2.4)$$

- Matrix **A** is called **digital image**, each element of **A** is called **image element**, **pixel**, or **pel.**
- **Sampling:** partitioning *xy*-plane into grid, coordinates of center of each grid are (x,y); x,y: integer. **Quantization**: f is assigned a gray-level value G (real or integer numbers).
- In practice, $N = 2^n$, $M = 2^k$, $G = 2^m$. Total number of bits required to store a digital image: $N\times M\times m$
- **Resolution**: degree of discernible detail (how good approximation in (2.4)), depending on the number of samples (**spatial resolution**) and gray-levels (**intensity resolution**).

1. Image Sampling & Quantization (2)

Digitizing the coordinate values



Digitizing the amplitude values

a b
c d

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.

1. Image Sampling & Quantization (3)

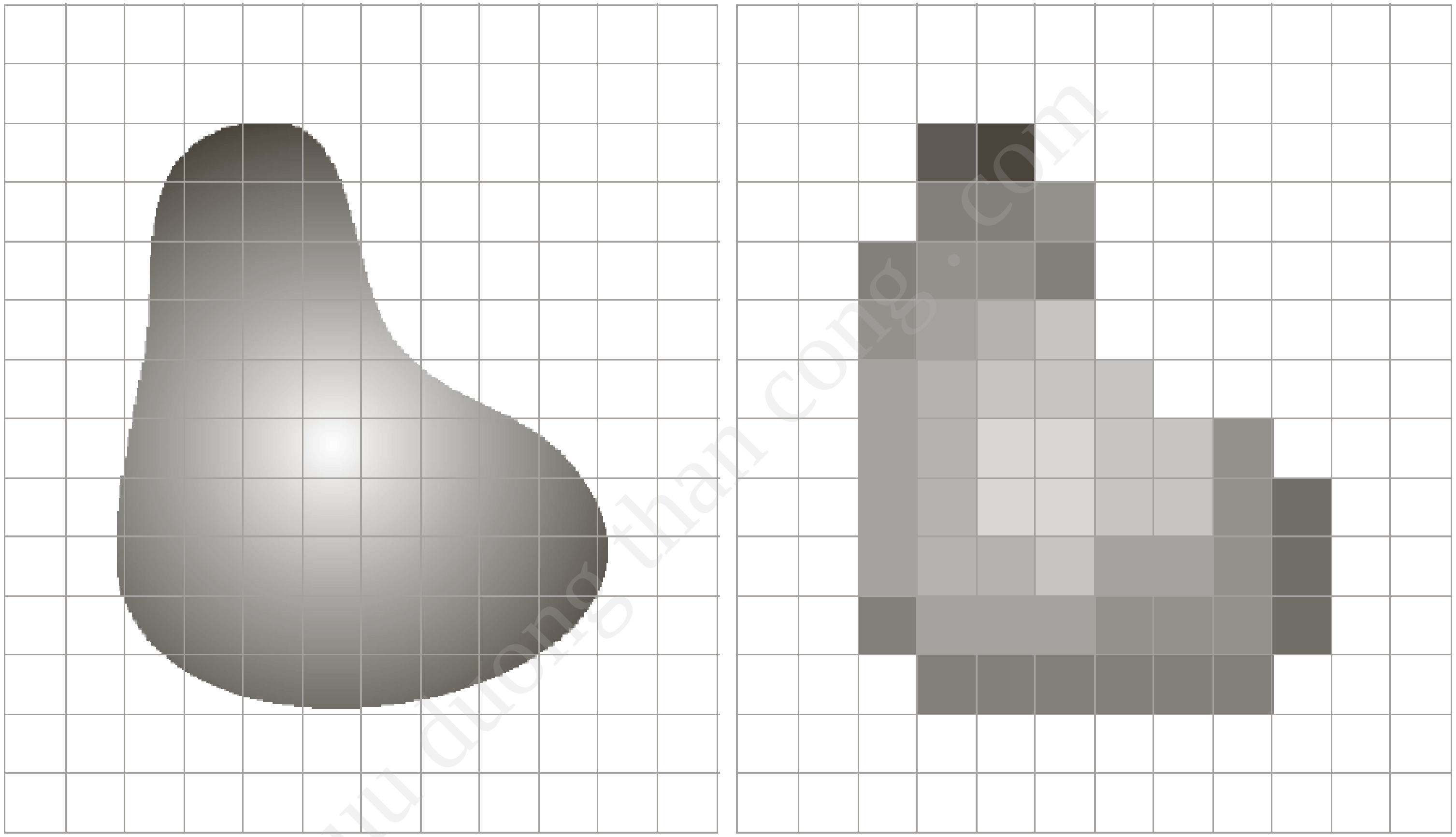
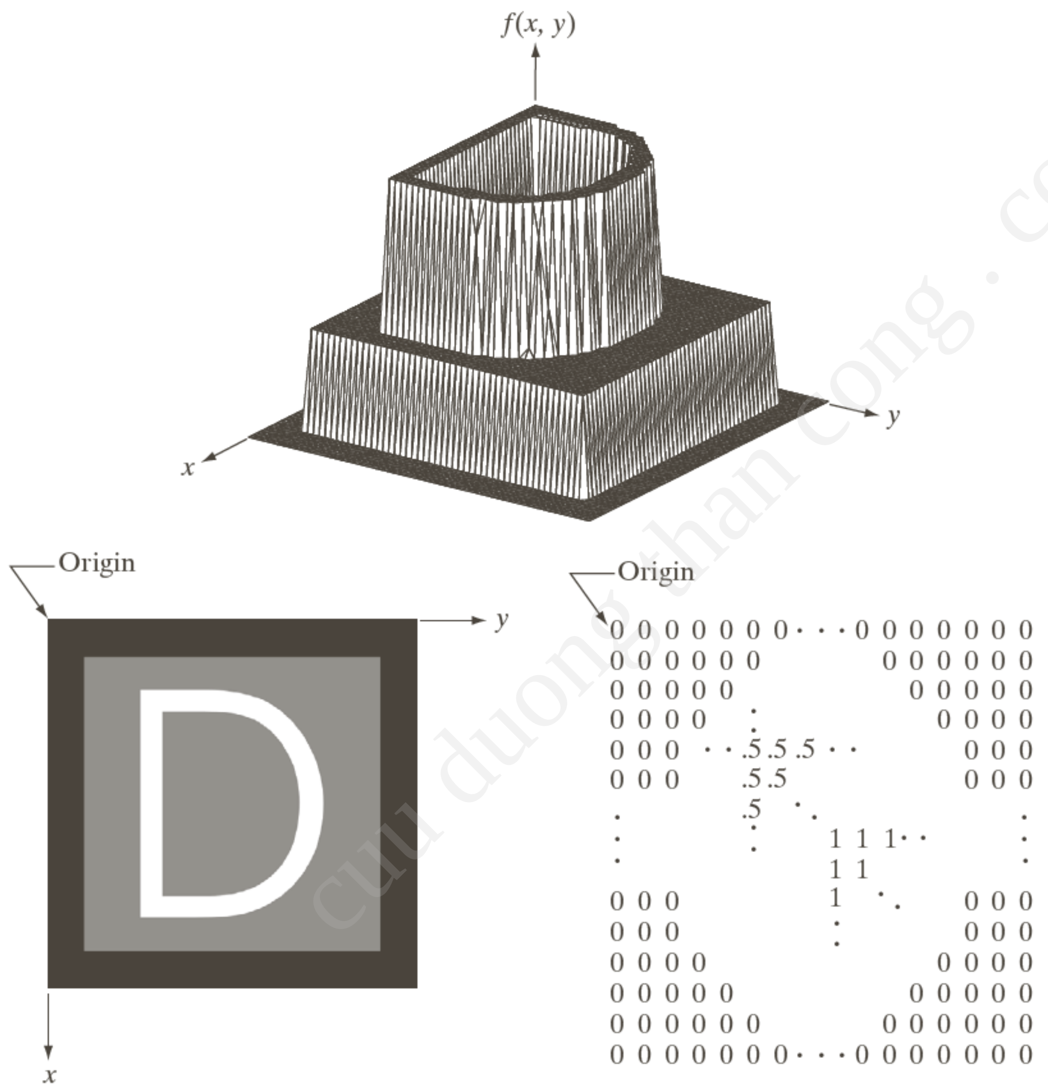


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

1. Image Sampling & Quantization (4)



a
b c

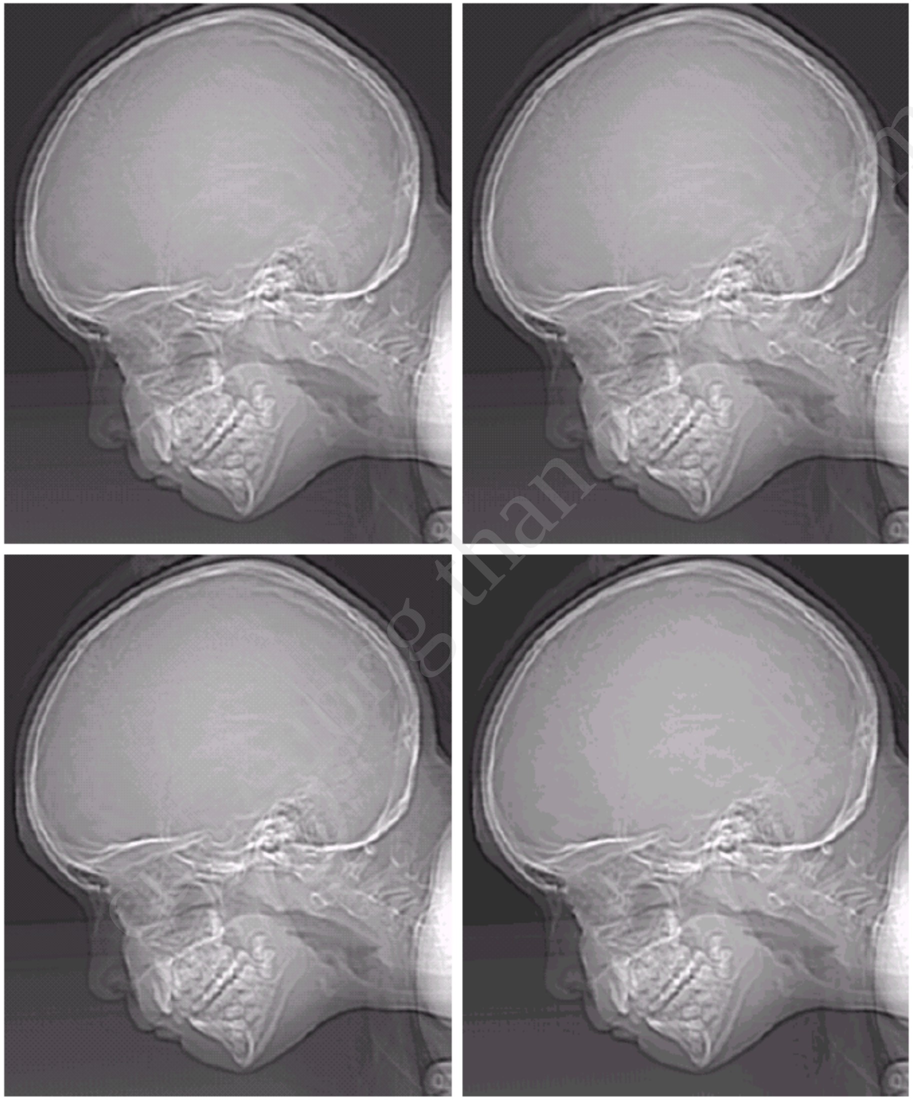
FIGURE 2.18
(a) Image plotted as a surface.
(b) Image displayed as a visual intensity array.
(c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

1. Image Sampling & Quantization (5)

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.

1. Image Sampling & Quantization (6)



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.

1. Image Sampling & Quantization (7)

e	f
g	h

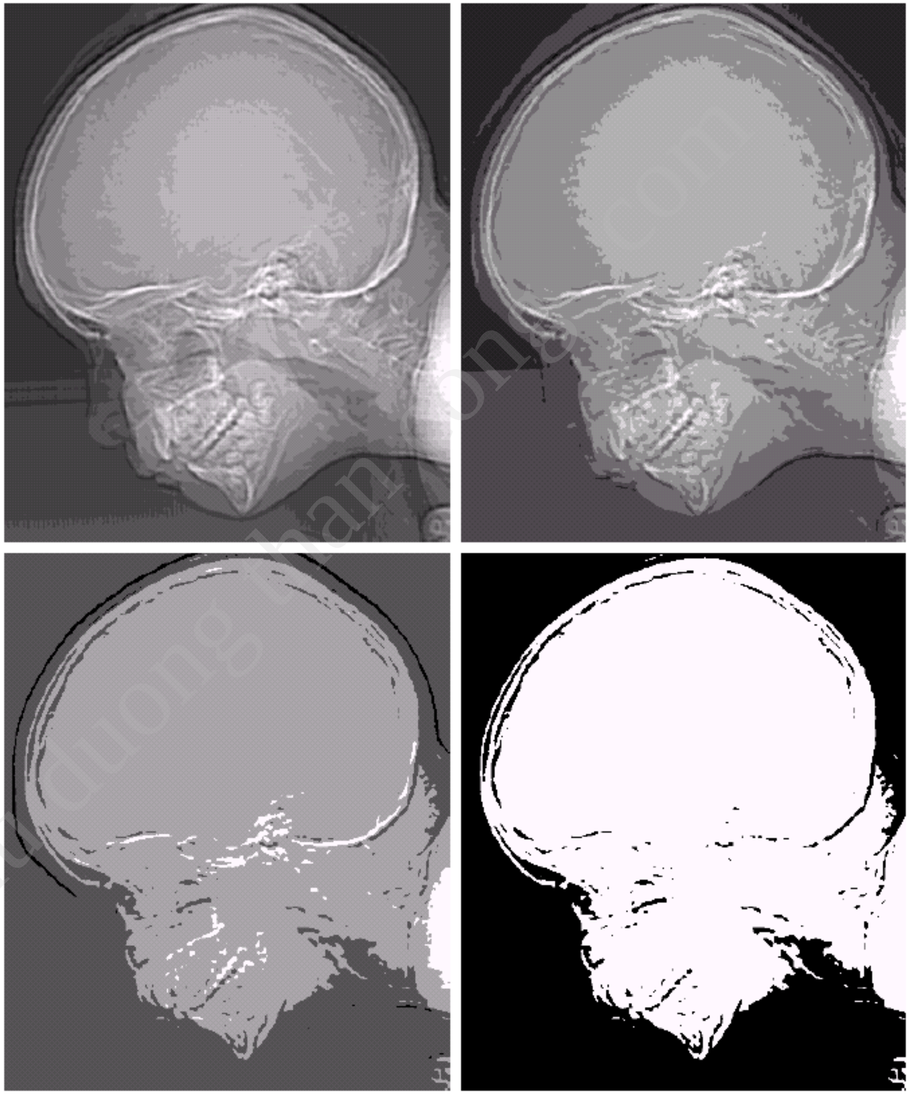


FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

1. Image Interpolation

- **Interpolation**: Process of using known data to estimate unknown values. E.g., zooming, shrinking, rotating, and geometric correction.

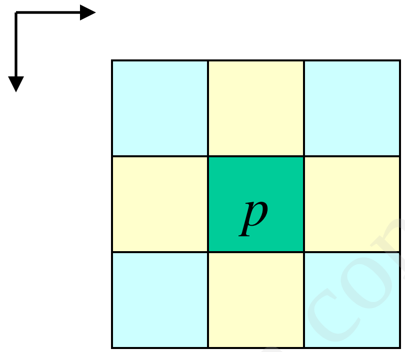
  Interpolation (sometimes called **resampling**): an imaging method to increase (or decrease) the number of pixels in a digital image.

  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)

1. Relationships Between Pixels (1)

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



- o *4-neighbors* of p, $N_4(p)$: neighbors at $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, $(x,y-1)$.
- o *4-diagonal neighbors* of p, $N_D(p)$: $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y+1)$, $(x-1,y-1)$.
- o *8-neighbors* of p, $N_8(p)$: $N_4(p)$ and $N_D(p)$.

1. Relationships Between Pixels (2)

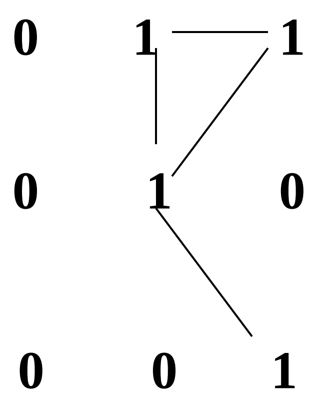
- **Connectivity:** Two pixels are connected if they are **adjacent** (e.g. 4-neighbors) and their gray levels satisfy specified criterion of similarity (e.g. they are equal).

  Let V set of gray levels used to define connectivity (e.g. V={1}: binary image; V={32,33,…,63,64}: gray-scale image).

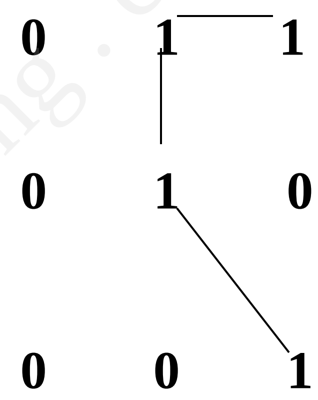
  - **4-connectivity** (**4-adjacent**): p and q are 4-connected if their values from V and q is in $N_4(p)$.
  - **8-connectivity** (**8-adjacent**): p and q are 8-connected if their values from V and q is in $N_8(p)$.
  - ***m*-connectivity** (***m*-adjacent**, mixed connectivity): p and q are m-connected if their values from V and
    - ✓ q is in $N_4(p)$, or
    - ✓ q is in $N_D(p)$ and set $S=N_4(p) \cap N_4(q)$ is empty (pixels $\in S$ are 4-neighbors of both p and q, and whose values are from V).

1. Relationships Between Pixels (3)

- *m*-connectivity introduced to eliminate ambiguity in path connections (resulted from allowing 8-connectivity)



8-neighbors of center pixel



m-neighbors of center pixel

1. Relationships Between Pixels (4)

❑ 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.

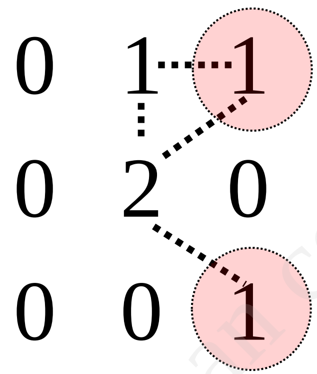
- If $(x_0, y_0) = (x_n, y_n)$, the path is **closed path**.

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

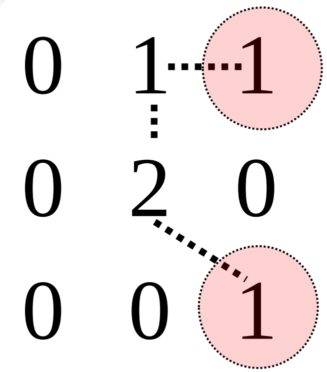
1. Relationships Between Pixels (5)

Examples: Connectivity and Path, $V = \{1, 2\}$

$0_{1,1}$	$1_{1,2}$	$1_{1,3}$
$0_{2,1}$	$2_{2,2}$	$0_{2,3}$
$0_{3,1}$	$0_{3,2}$	$1_{3,3}$



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)

1. Relationships Between Pixels (6)

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

  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.**

1. Relationships Between Pixels (7)

- **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**.

1. Relationships Between Pixels (8)

- **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. $D(p, q) \geq 0 \quad [D(p, q) = 0, \text{ iff } p = q]$

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

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

The following are the different distance measures:

a. **Euclidean distance**:

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

b. **City block distance**:

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

c. **Chess board distance**:

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

1. Relationships Between Pixels (9)

- **Array** vs. **Matrix operation**

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

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

1. Relationships Between Pixels (10)

- **Linear** vs. **Nonlinear operation**

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

$$\begin{aligned}
&H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] \\
&= H\left[a_i f_i(x,y)\right] + H\left[a_j f_j(x,y)\right] \\
&= a_i H\left[f_i(x,y)\right] + a_j H\left[f_j(x,y)\right] \\
&= a_i g_i(x,y) + a_j g_j(x,y)
\end{aligned}$$

Additivity

Homogeneity

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

1. Relationships Between Pixels (11)

❑ Arithmetic operation

- **Arithmetic operations** between 2 pixels *p* and *q*: **addition**, **subtraction**, **multiplication**, **division** (carry out pixel by pixel).
- **Mask (window) operation**:

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$p = \sum_{i=1}^{9} w_i p_i \rightarrow p_5$$

with proper selection of coefficients, the operation is used for noise reduction, region thinning, edge detection.

Example of average window:

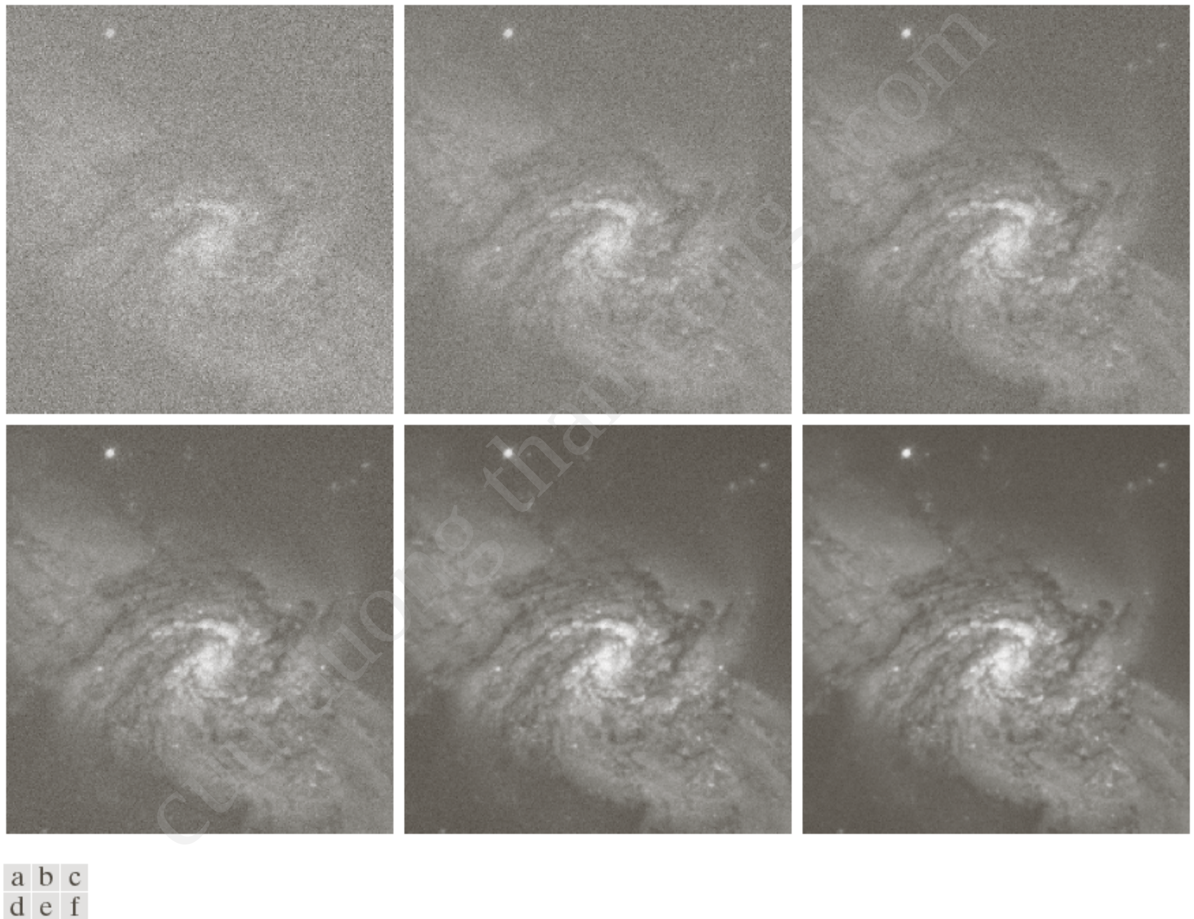


FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

1. Relationships Between Pixels (13)

- **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

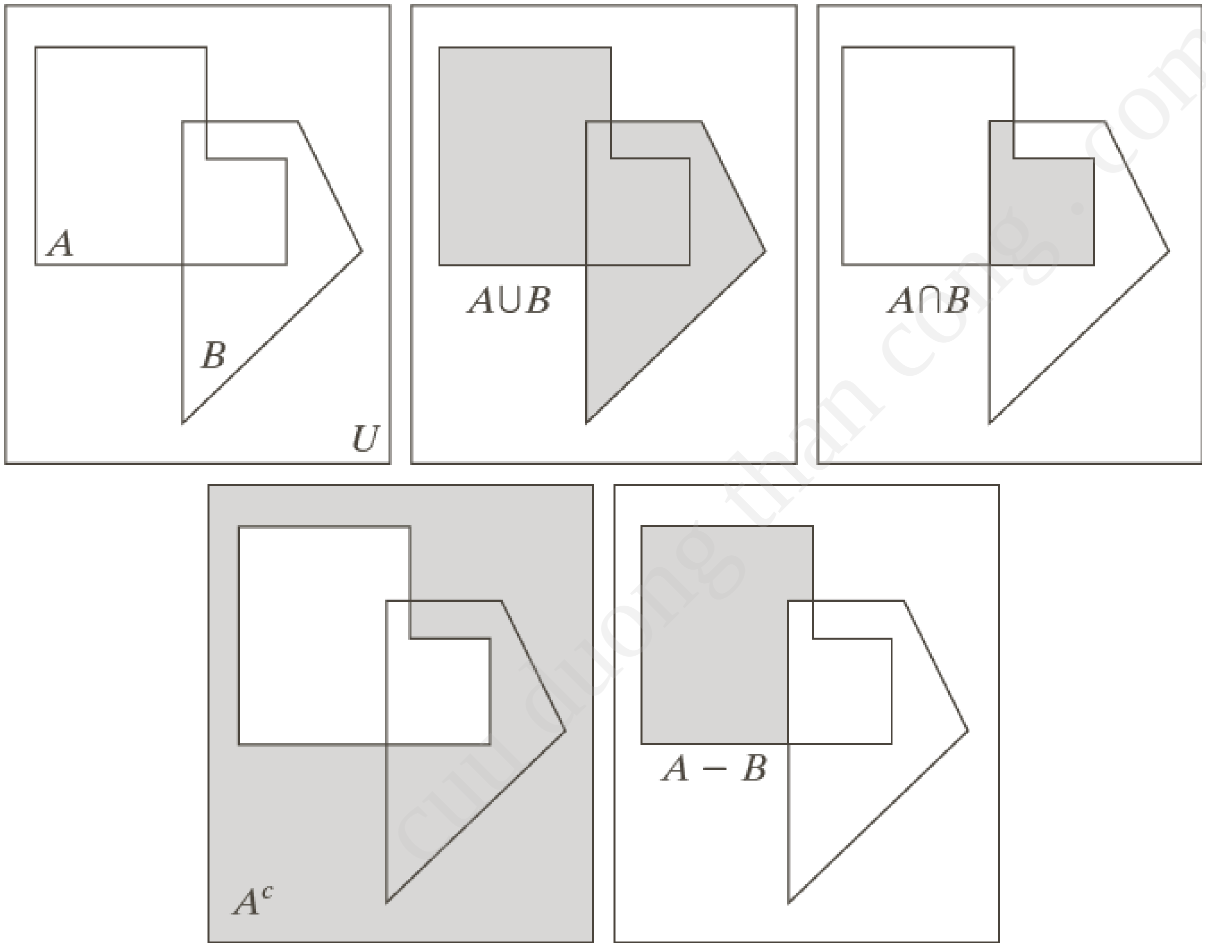
$$A^c = \{(x, y, K - z) \mid (x, y, z) \in A\}$$

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

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

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

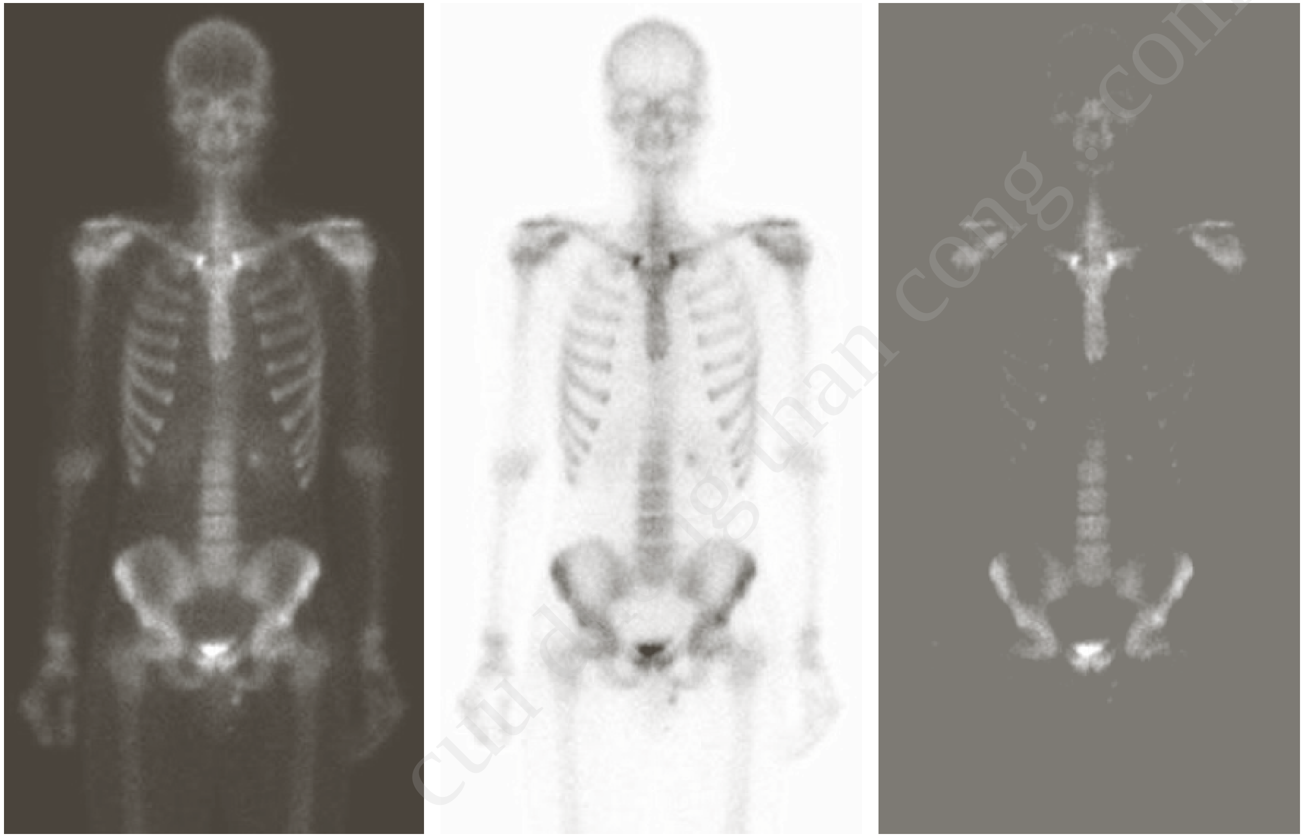
1. Relationships Between Pixels (14)



a b c
d e

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.

1. Relationships Between Pixels (15)



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.)

Logic operations: used for feature detection, shape analysis, binary image processing.

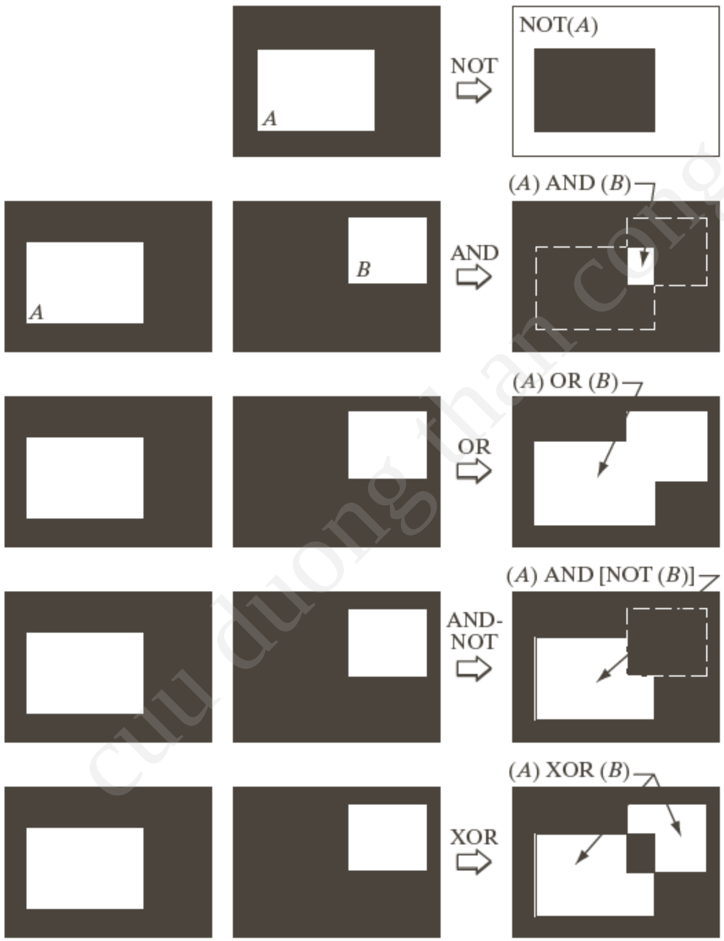


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

1. Imaging Geometry (1)

- **Translation:** A point (x, y, z) is shifted to new position at (x', y', z') by using displacements (x_0, y_0, z_0):

$$\begin{aligned} x' &= x + x_0 \\ y' &= y + y_0 \\ z' &= z + z_0 \end{aligned} \Rightarrow \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$\Rightarrow$ Matrix form: **v’=Tv.**

- **Scaling:** by factors S_x, S_y, S_z along the x, y, z axes given by

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Imaging Geometry (2)

- **Rotation:**
  - Rotation of a point about z-coordinate by an angle θ:

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Rotation of a point about x-coordinate by an angle α:

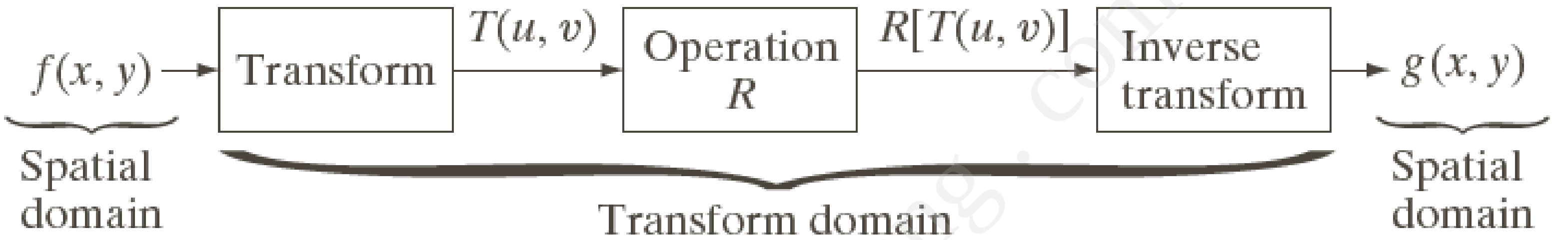
$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Imaging Geometry (3)

- Rotation of a point about *y*-coordinate by an angle β:

$$\mathbf{R}_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Image Transforms: General Form



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

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

1. Image Transforms: 2-D Fourier Transform (1)

- **Definition:**

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

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

- **Properties:**
  - <u>Separability:</u>

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

for u, v = 0, 1…., N-1.

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

for x, y = 0, 1…., N-1.

1. Image Transforms: 2-D Fourier Transform (2)

Advantage of separability property: F(u,v) or f(x,y) can be obtained in 2 steps by successive applications of the 1-D Fourier transform or its inverse:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v)e^{(-j2\pi ux/N)}$$

with

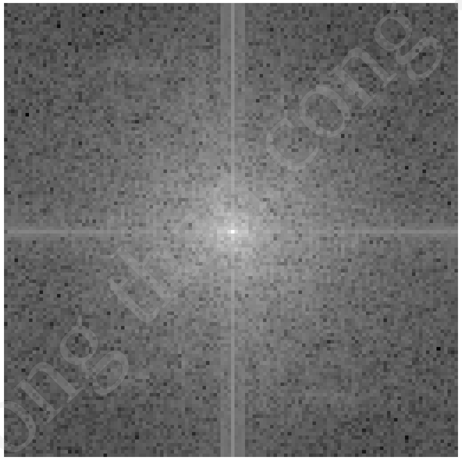
$$F(x,v) = N\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{(-j2\pi vy/N)}\right]$$

- Others: translation, periodicity and conjugate symmetry, rotation, scaling….

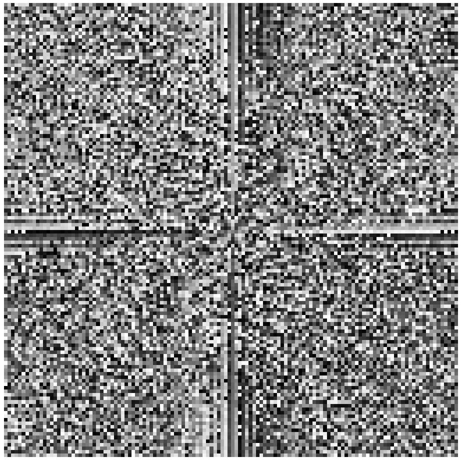
1. Image Transforms: 2-D Fourier Transform (3)

Examples of 2-D Fourier transform:

Original Image



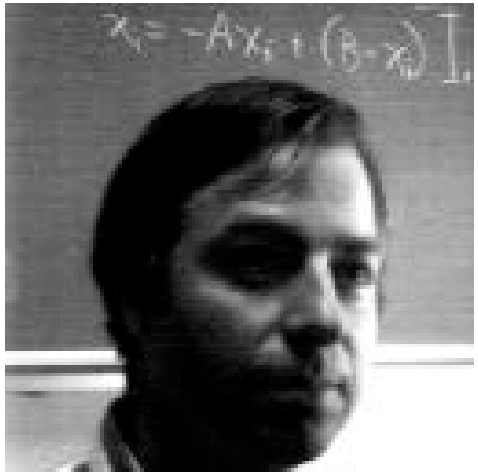
Magnitude of Fourier transform

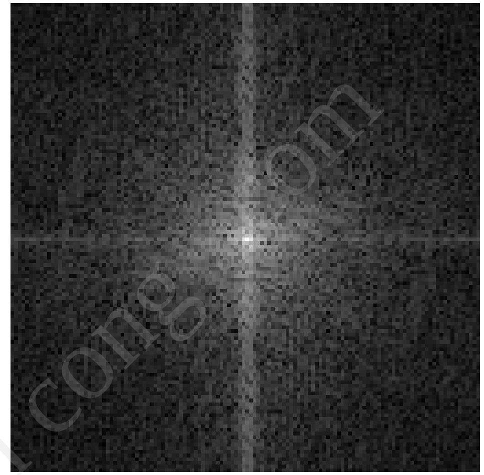


Phase of Fourier transform

1. Image Transforms: 2-D Fourier Transform (4)

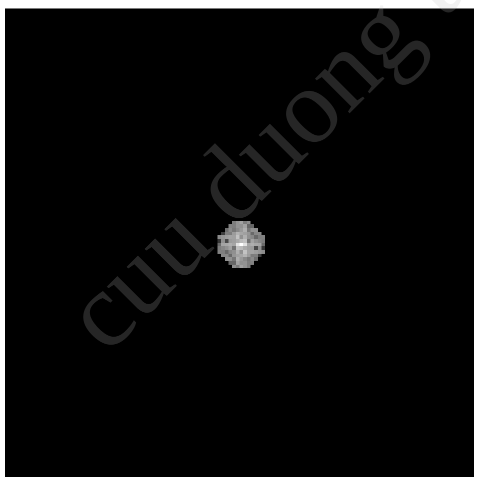
Original image

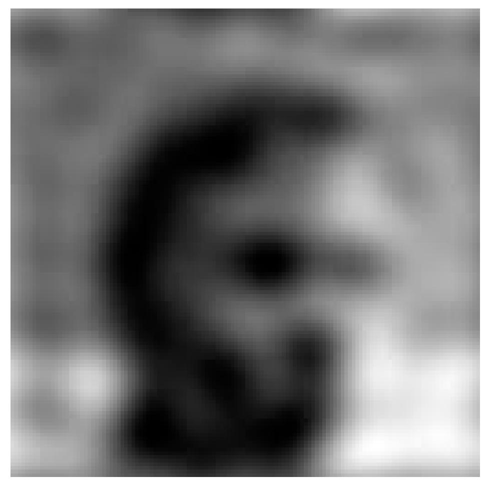




Magnitude of Fourier transform

Low pass filter used

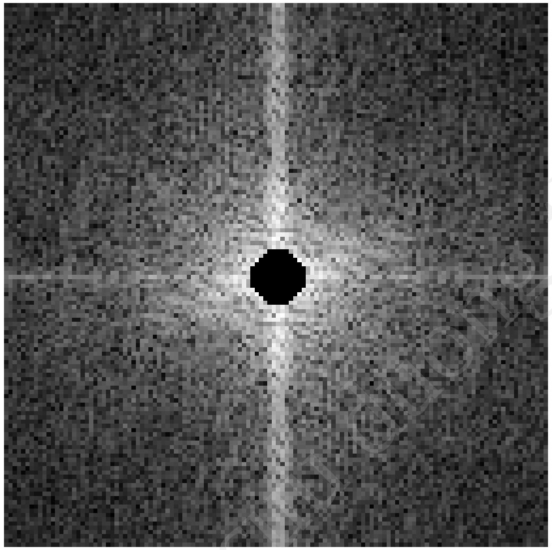




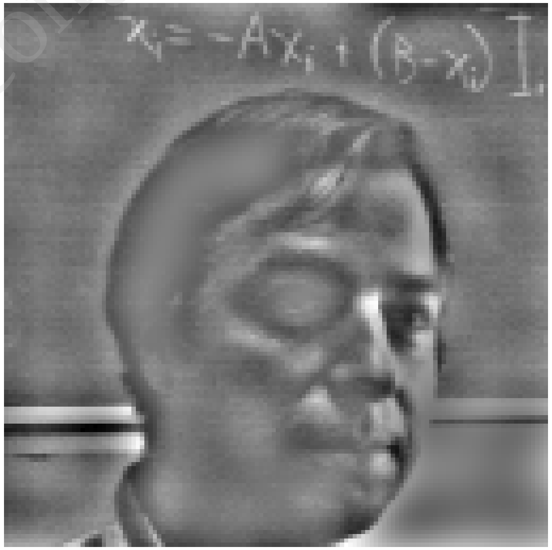
Reconstructed image

1. Image Transforms: 2-D Fourier Transform (5)

High pass filter used



Reconstructed image



1. Image Transforms: 2-D Fourier Transform (6)

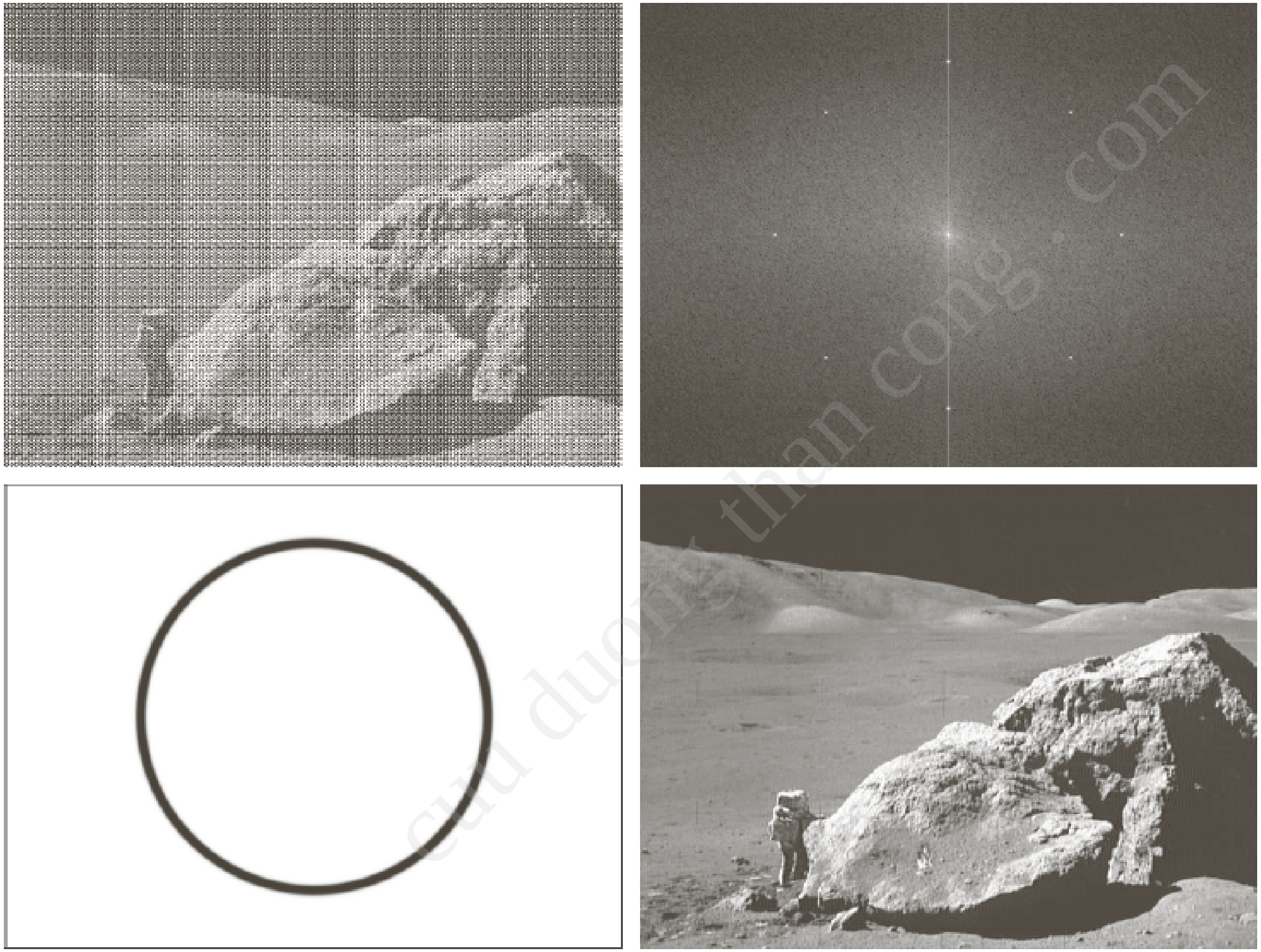


FIGURE 2.40
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy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

1. Image Transforms: 2-D DCT (1)

- **DCT (Discrete Cosine Transform):**

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

for u, v = 0, 1…., N-1, and

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

for x, y = 0, 1…., N-1.

where

$$\alpha(u)=\begin{cases}\sqrt{\dfrac{1}{N}}, & u=0\\ \sqrt{\dfrac{2}{N}}, & u=1,2,\ldots,N-1\end{cases}$$

1. Image Transforms: 2-D DCT (2)

- **2-D general transform**:

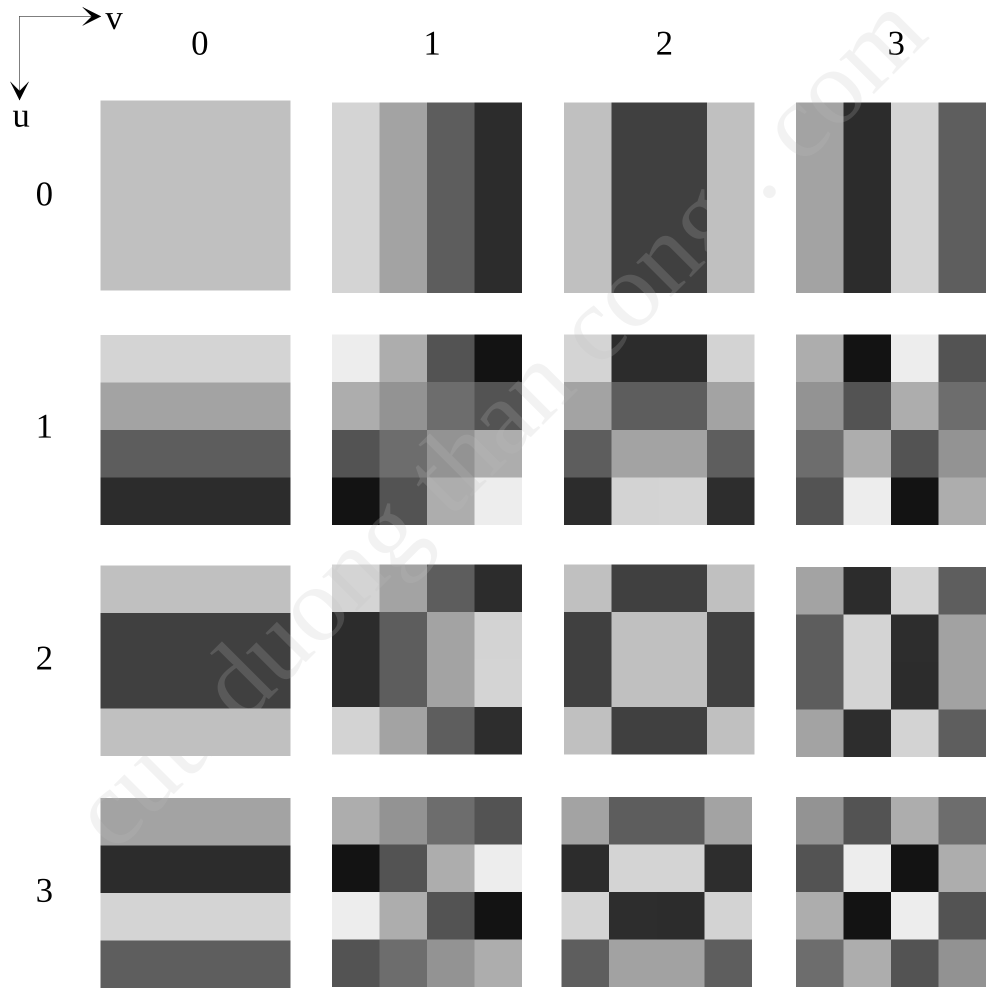
$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}T(x,y,u,v)\,f(x,y)$$

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

where $T(x,y,u,v)$ and $I(x,y,u,v)$ are forward and inverse transform **kernel** (**basis functions**), respectively. The basis function depends only on the indexes u, v, x, y, not on the values of image or its transform.

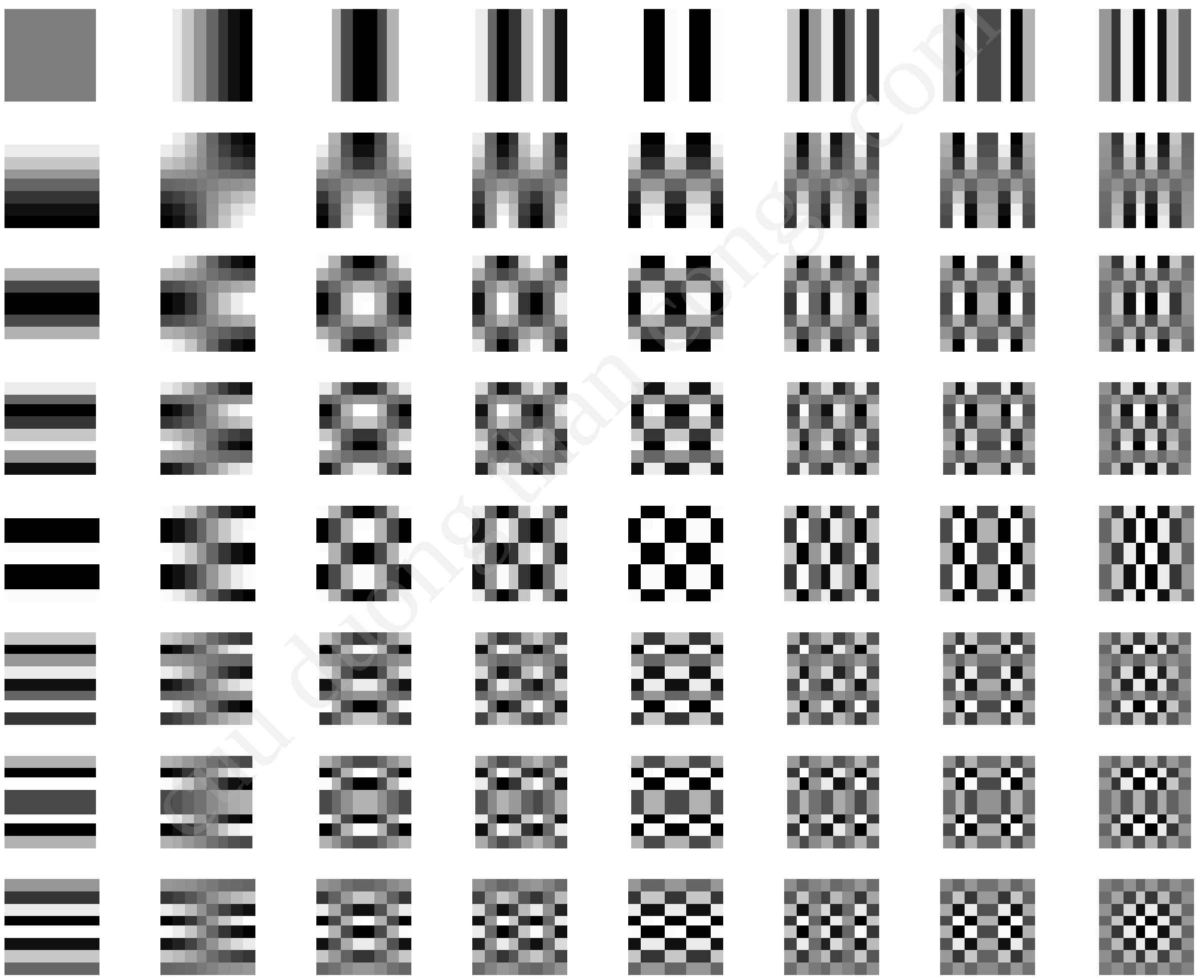
1. Image Transforms: 2-D DCT (3)

Example of basis functions of 2-D DCT, *N*=4:



1. Image Transforms: 2-D DCT (4)

Example of basis functions of 2-D DCT, *N*=8:



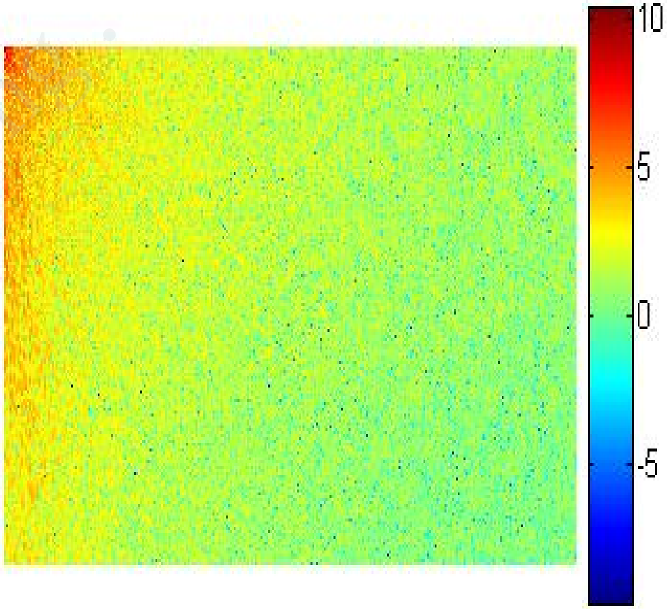
1. Image Transforms: 2-D DCT (5)

- DCT is a real transform.
- There are fast algorithms to compute the DCT similar to the FFT.
- DCT has **excellent energy compaction** properties.

- Original Lena image

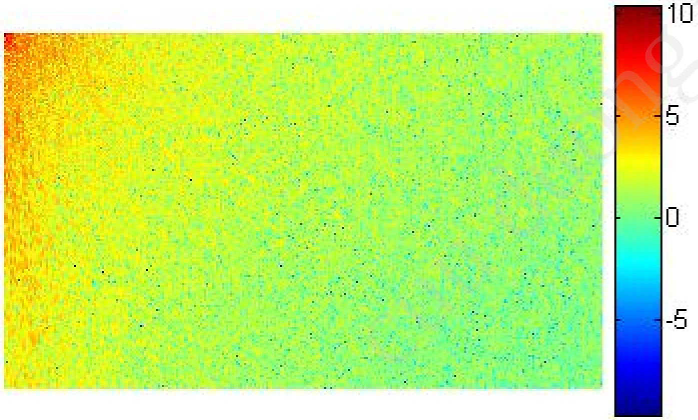
- 2D DCT

1. Image Transforms: 2-D DCT (6)

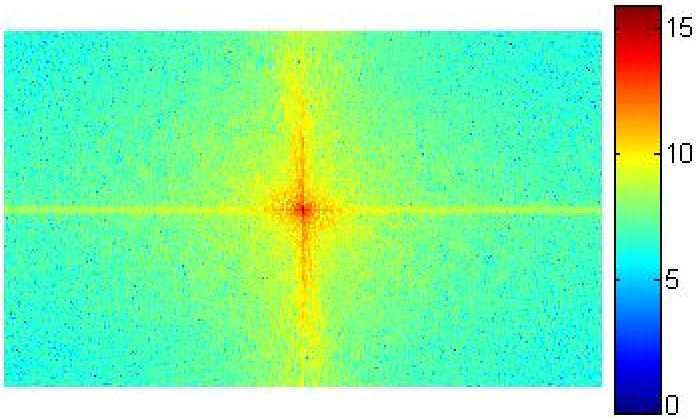


1. Image Transforms: 2-D DCT (7)

Original

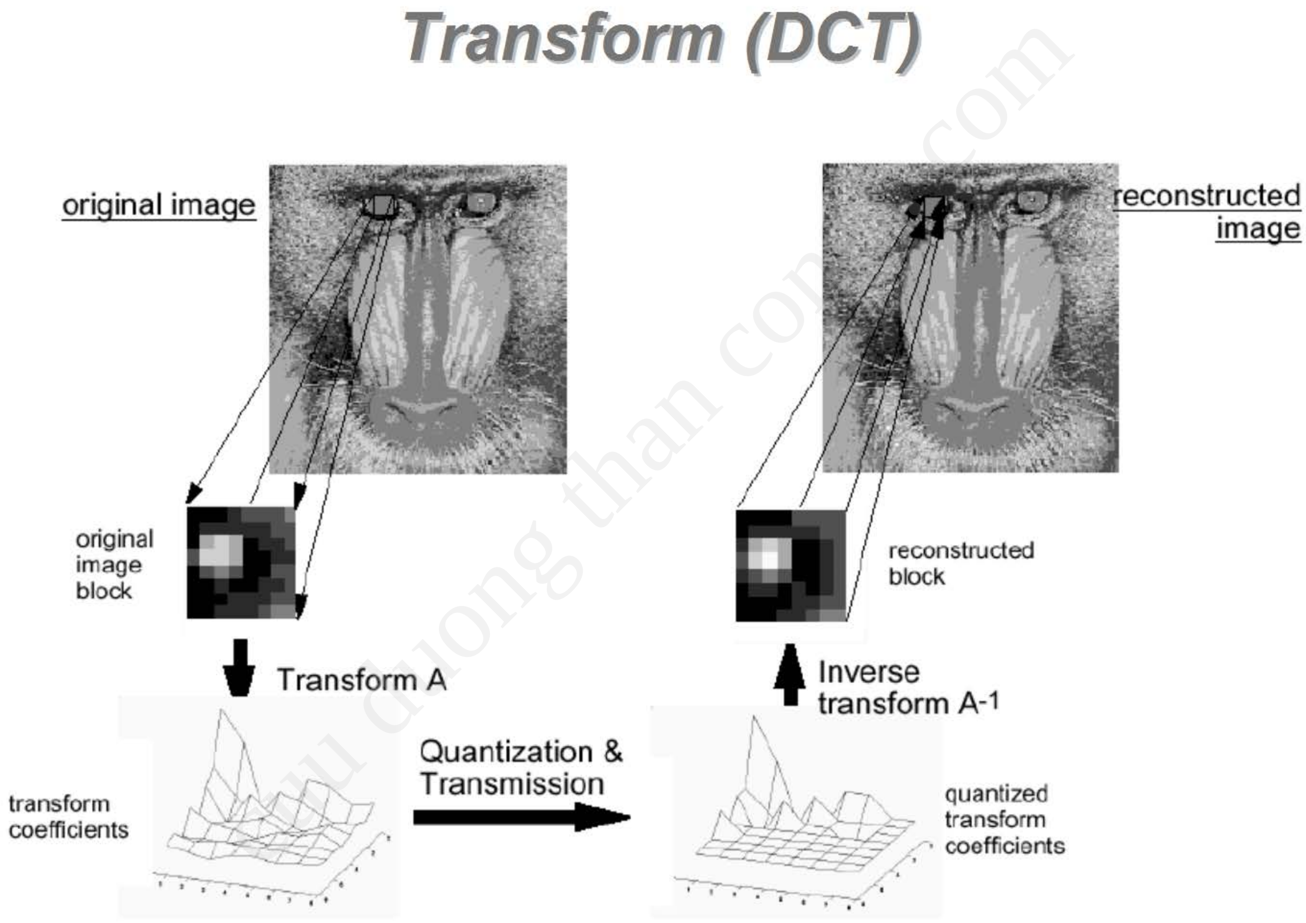


DCT



FFT

1. Image Transforms: 2-D DCT (8)



1. Image Transforms: 2-D DCT (9)

Original

Reconstructed from IDCT, (J < 5)

Reconstructed from IDCT, (J < 10)

Reconstructed from IDCT, (J < 20)

1. Probabilistic Methods (1)

Let $z_i,\ i = 0,\ 1,\ 2,\ \ldots,\ L\text{-}1$, 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

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

The variance of the intensities is given by

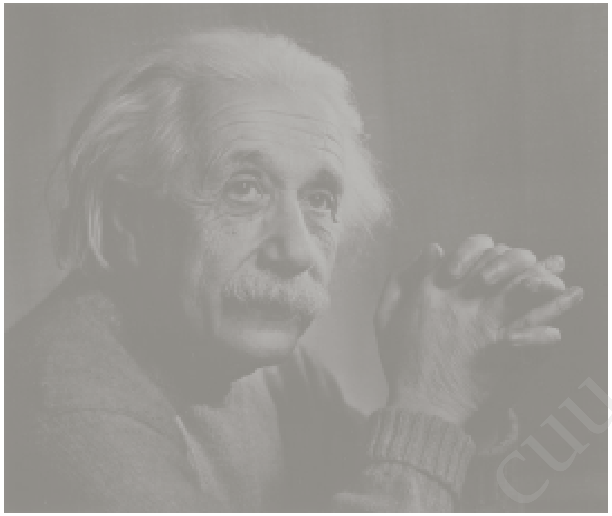
$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2\, p(z_k)$$

1. Probabilistic Methods (2)

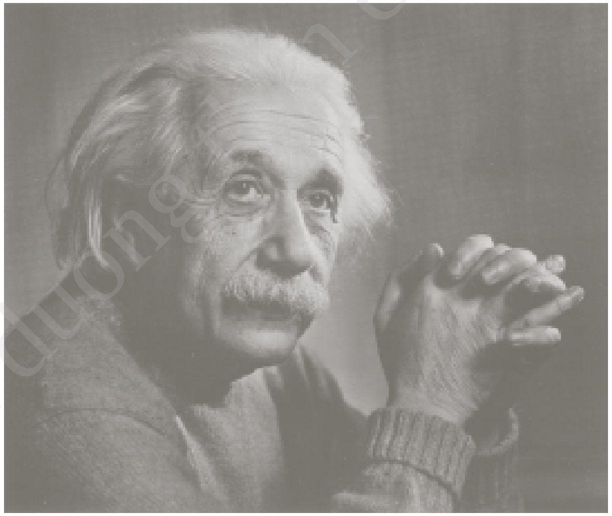
The n^{th} moment of the intensity variable z is

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

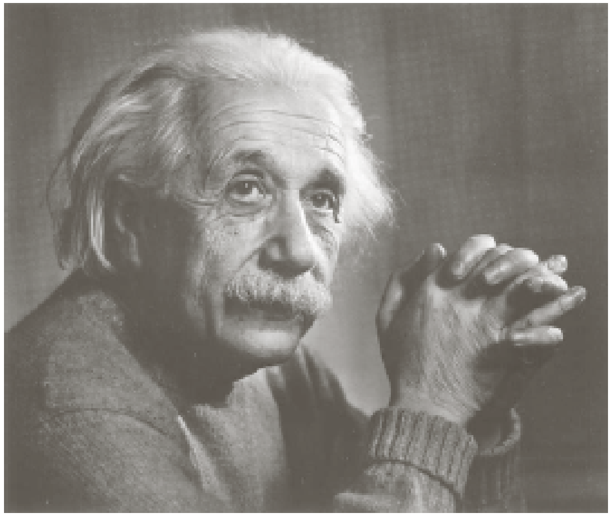
<u>Example</u>: Comparison of standard deviation values



$\sigma = 14.3$



$\sigma = 31.6$



$\sigma = 49.2$