

Digital Image Processing

Lecture 1 Introduction & Fundamentals

K K Singh, RGUKT Nuzvid



Introduction to the course

► Article Reading and Project

- Develop face Recognition-
- Gender Classification
- Code correction
- Obstacle detection

Journals & Conferences in Image Processing

▶ Journals:

- **IEEE T IMAGE PROCESSING**
- **IEEE T MEDICAL IMAGING**
- **INTL J COMP. VISION**
- **IEEE T PATTERN ANALYSIS MACHINE INTELLIGENCE**
- **PATTERN RECOGNITION**
- **COMP. VISION AND IMAGE UNDERSTANDING**
- **IMAGE AND VISION COMPUTING**

... ..

▶ Conferences:

- **CVPR: Comp. Vision and Pattern Recognition**
- **ICCV: Intl Conf on Computer Vision**
- **ACM Multimedia**
- **ICIP**
- **SPIE**
- **ECCV: European Conf on Computer Vision**
- **CAIP: Intl Conf on Comp. Analysis of Images and Patterns**

... ..

Introduction

- ▶ What is Digital Image Processing?
- ▶ **Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images.**

Digital Image

— a two-dimensional function , x and y are spatial coordinates , The amplitude f is called intensity or gray level at the point (x, y)

Digital Image Processing

— process digital images by means of computer, it covers low-, mid-, and high-level processes

low-level: inputs and outputs are images

mid-level: outputs are attributes extracted from input images

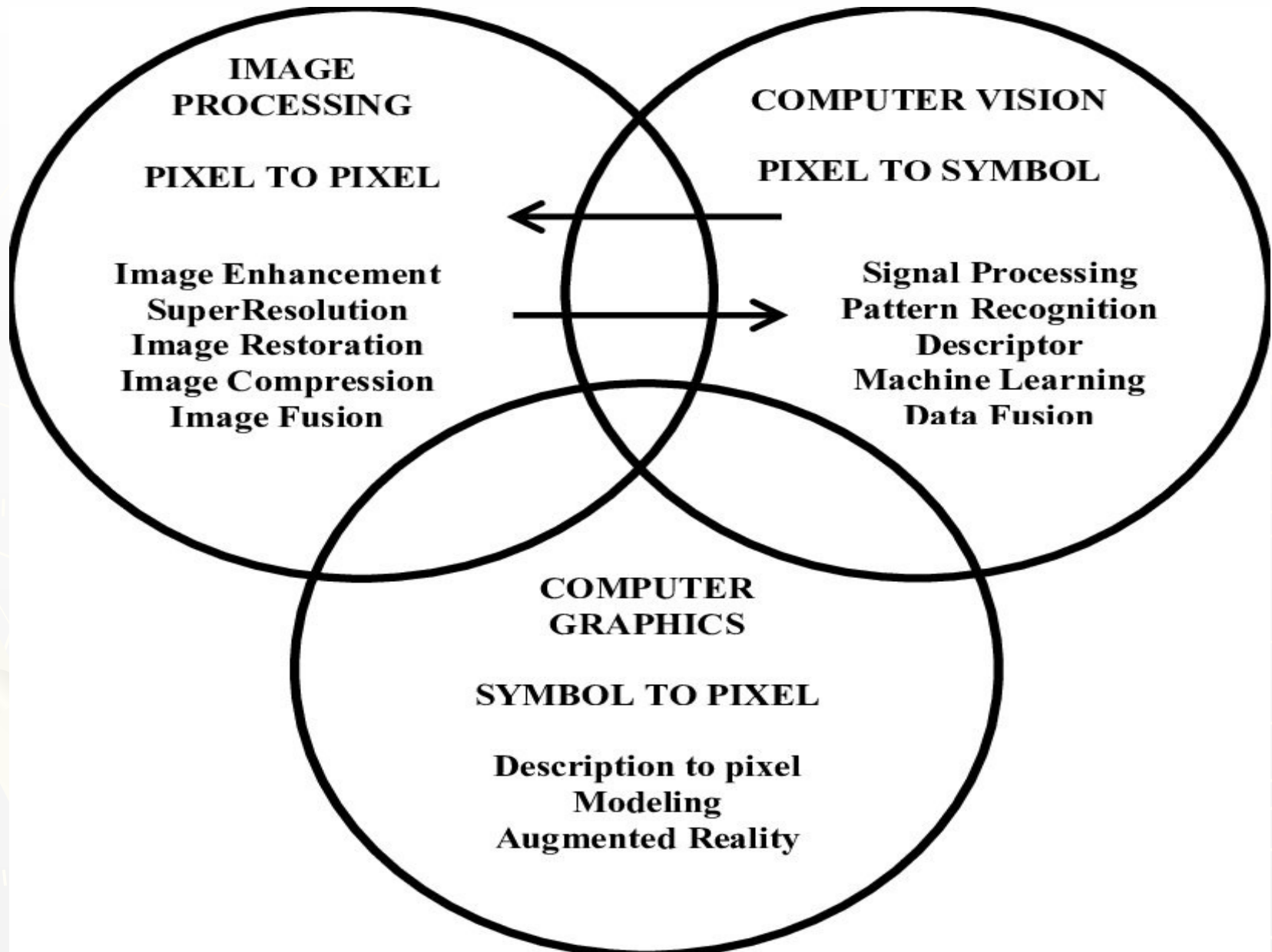
high-level: recognition of individual objects

Pixel — the elements of a digital image

Relationship between a digital image and a signal

- ▶ **If the image is a two dimensional array then what does it have to do with a signal?**
- ▶ Since anything that conveys information or broadcast a message in physical world between two observers is a signal.
- ▶ **How a digital image is formed**
- ▶ Since capturing an image from a camera is a physical process. The sunlight is used as a source of energy.
- ▶ A sensor array is used for the acquisition of the image.
- ▶ So when the sunlight falls upon the object, then the amount of light reflected by that object is sensed by the sensors, and a continuous voltage signal is generated by the amount of sensed data.

Overlapping fields

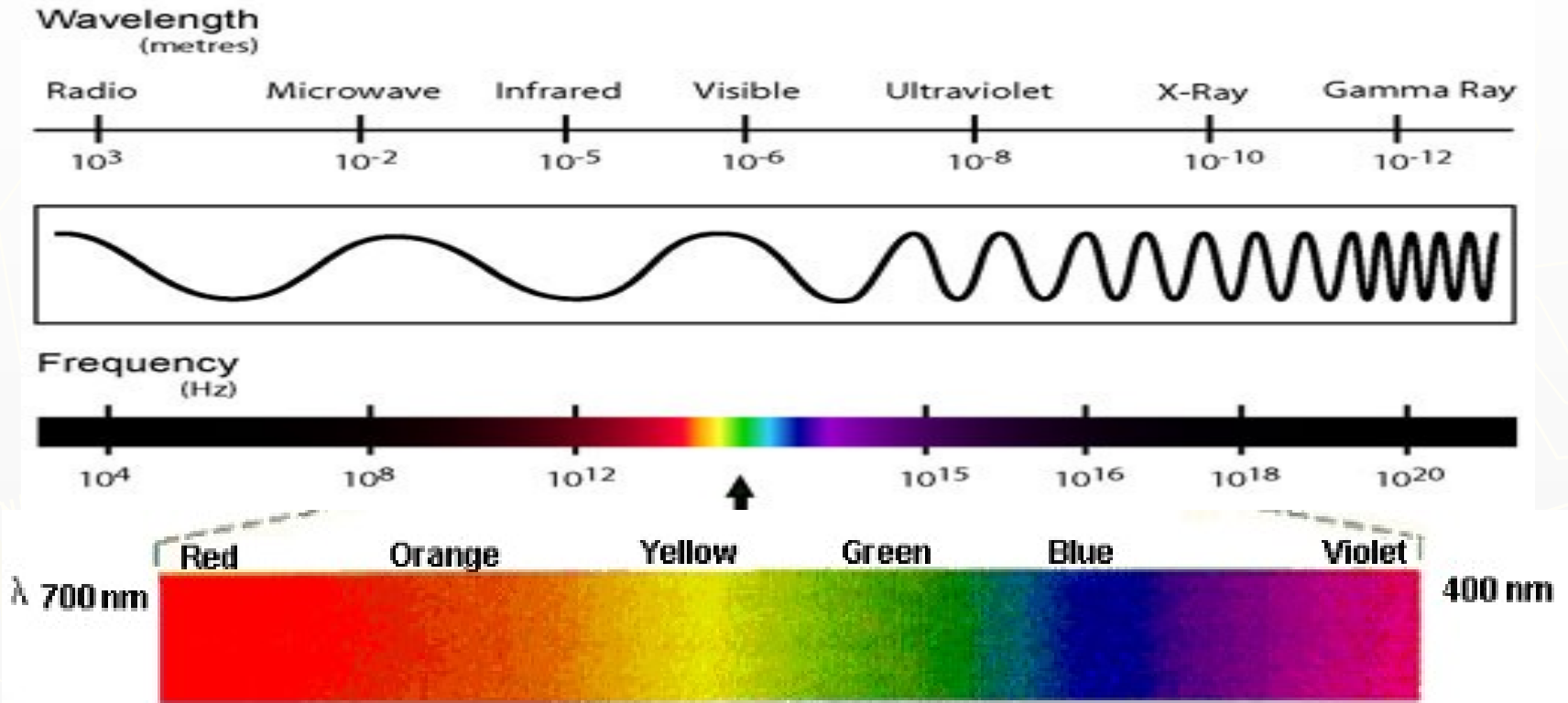


Q &A

- ▶ How can you tell that an image is a form of signal?
- ▶ Difference among image processing, computer vision, and computer graphics.

Light and EM Spectrum

THE ELECTRO MAGNETIC SPECTRUM



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

In physics, electromagnetic radiation refers to the waves of the electromagnetic field, propagating through space, carrying electromagnetic radiant energy. It includes radio waves, microwaves, infrared, light, ultraviolet, X-rays, and gamma rays.

Electromagnetic (EM) energy spectrum

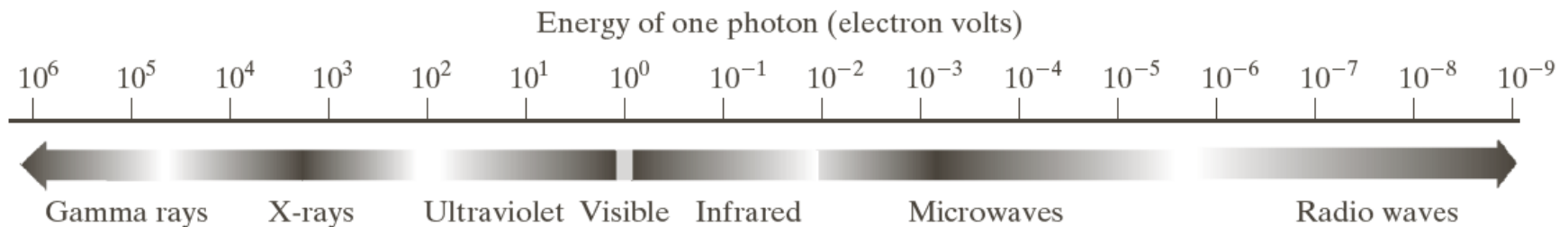


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

Major uses

Gamma-ray imaging: nuclear medicine and astronomical observations

X-rays: medical diagnostics, industry, and astronomy, etc.

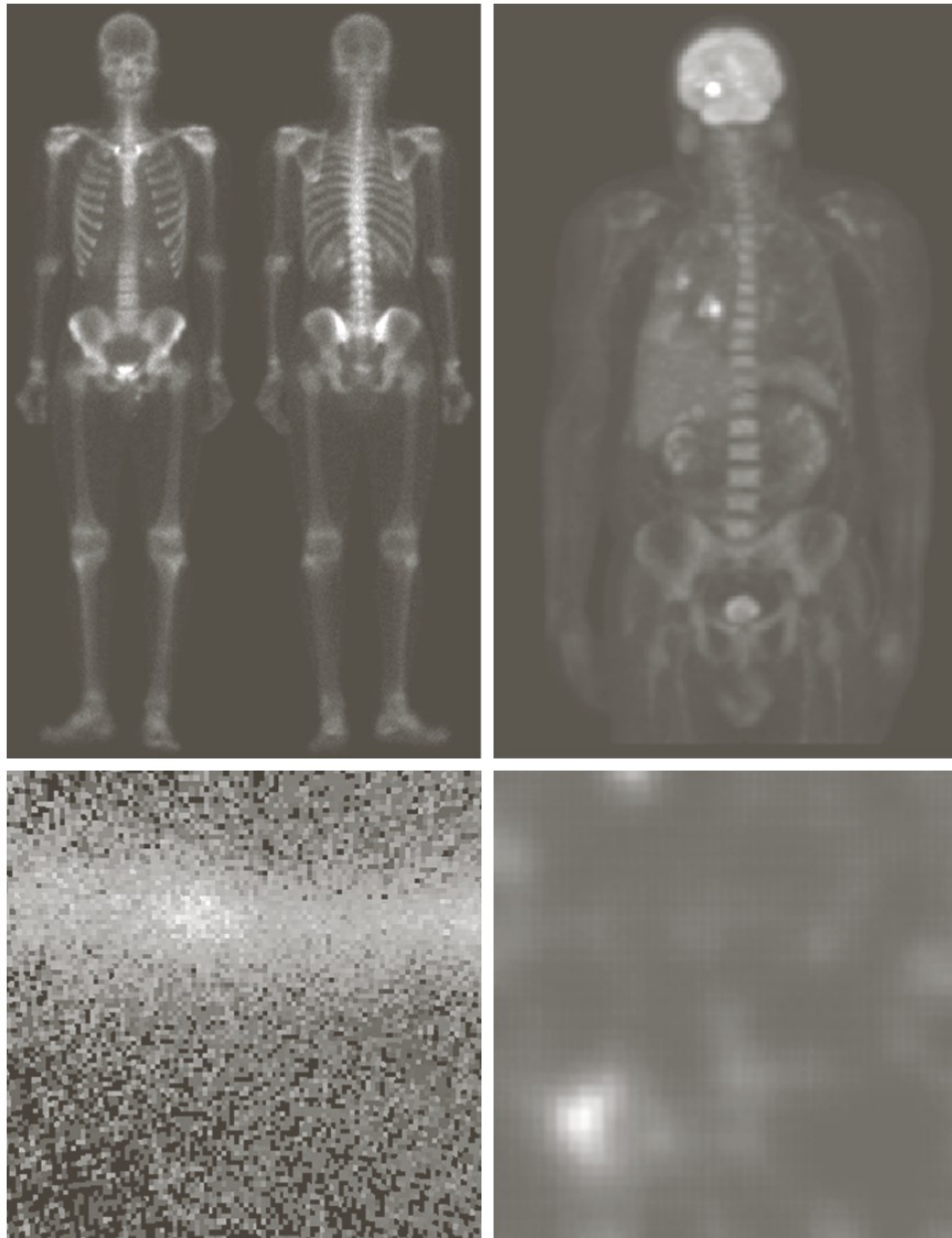
Ultraviolet: lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations

Visible and infrared bands: light microscopy, astronomy, remote sensing, industry, and law enforcement

Microwave band: radar

Radio band: medicine (such as MRI) and astronomy

Examples: Gama-Ray Imaging



a	b
c	d

FIGURE 1.6

Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve.

(Images courtesy of (a) G.E.

Medical Systems,

(b) Dr. Michael

E. Casey, CTI

PET Systems,

(c) NASA,

(d) Professors

Zhong He and

David K. Wehe,

University of

Michigan.)

PET(positron
emission
tomography),
Cygnus
loop(stationary
gas cloud
generated
around 15000
years ago when
a star was
exploded

Examples: X-Ray Imaging

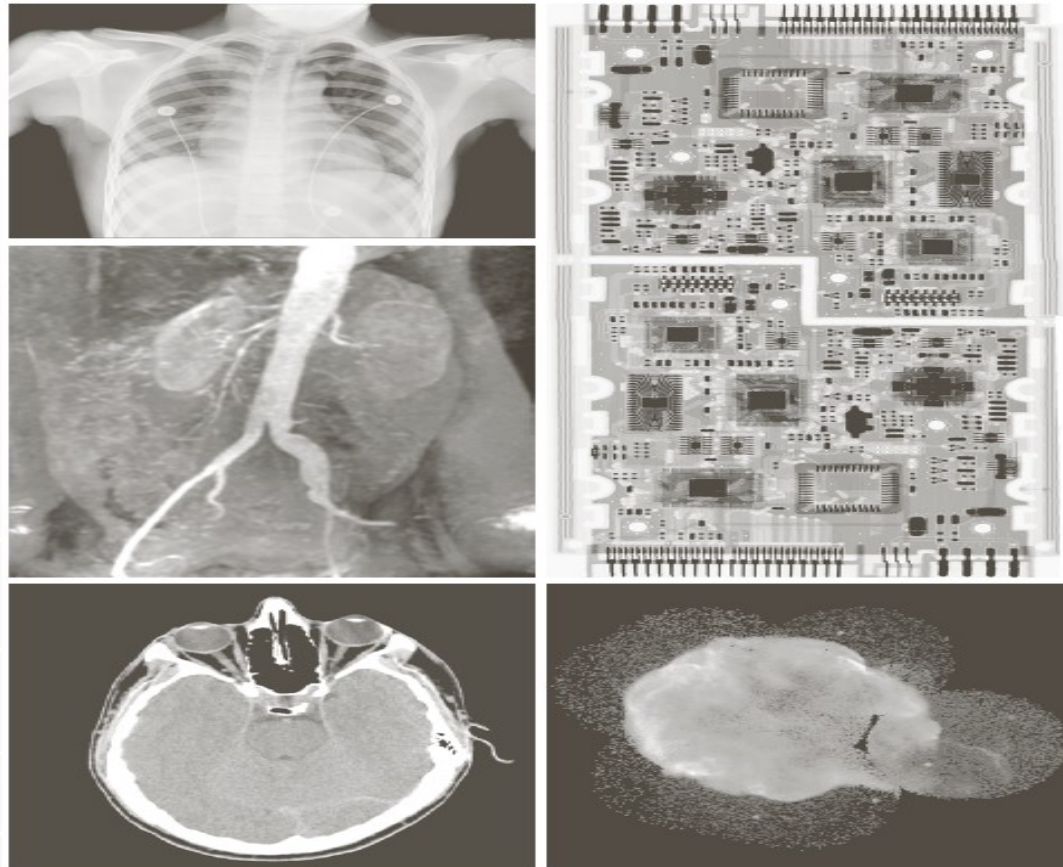
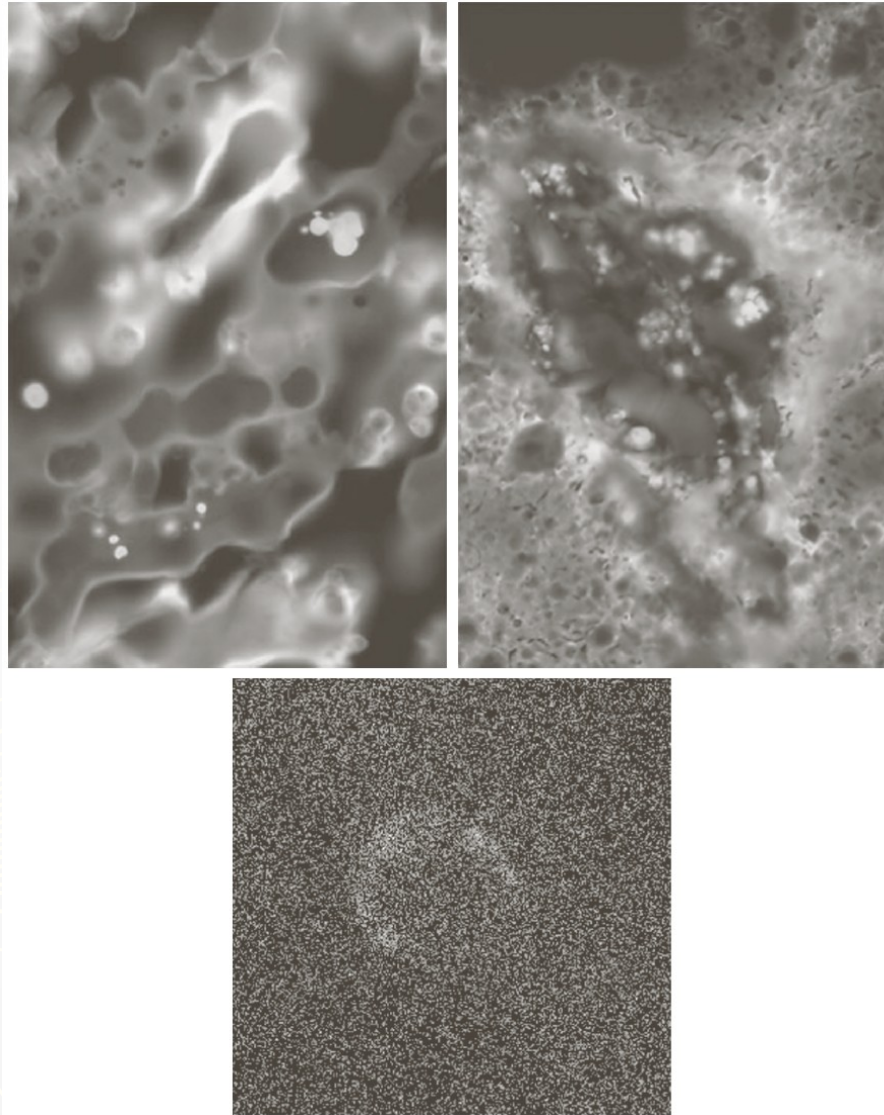


FIGURE 1.7 Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Circuit boards. (e) Cygnus Loop. (Images courtesy of (a) and (c) Dr. David R. Pickens, Dept. of Radiology & Radiological Sciences, Vanderbilt University Medical Center; (b) Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School; (d) Mr. Joseph E. Pascente, Lixi, Inc.; and (e) NASA.)

Examples: Ultraviolet Imaging



a b
c

FIGURE 1.8

Examples of ultraviolet imaging.

(a) Normal corn.

(b) Smut corn.

(c) Cygnus Loop.

(Images courtesy of (a) and

(b) Dr. Michael

W. Davidson,

Florida State

University,

(c) NASA.)

Examples: Visual and Infrared Imaging

TABLE 1.1

Thematic bands
in NASA's
LANDSAT
satellite.

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping

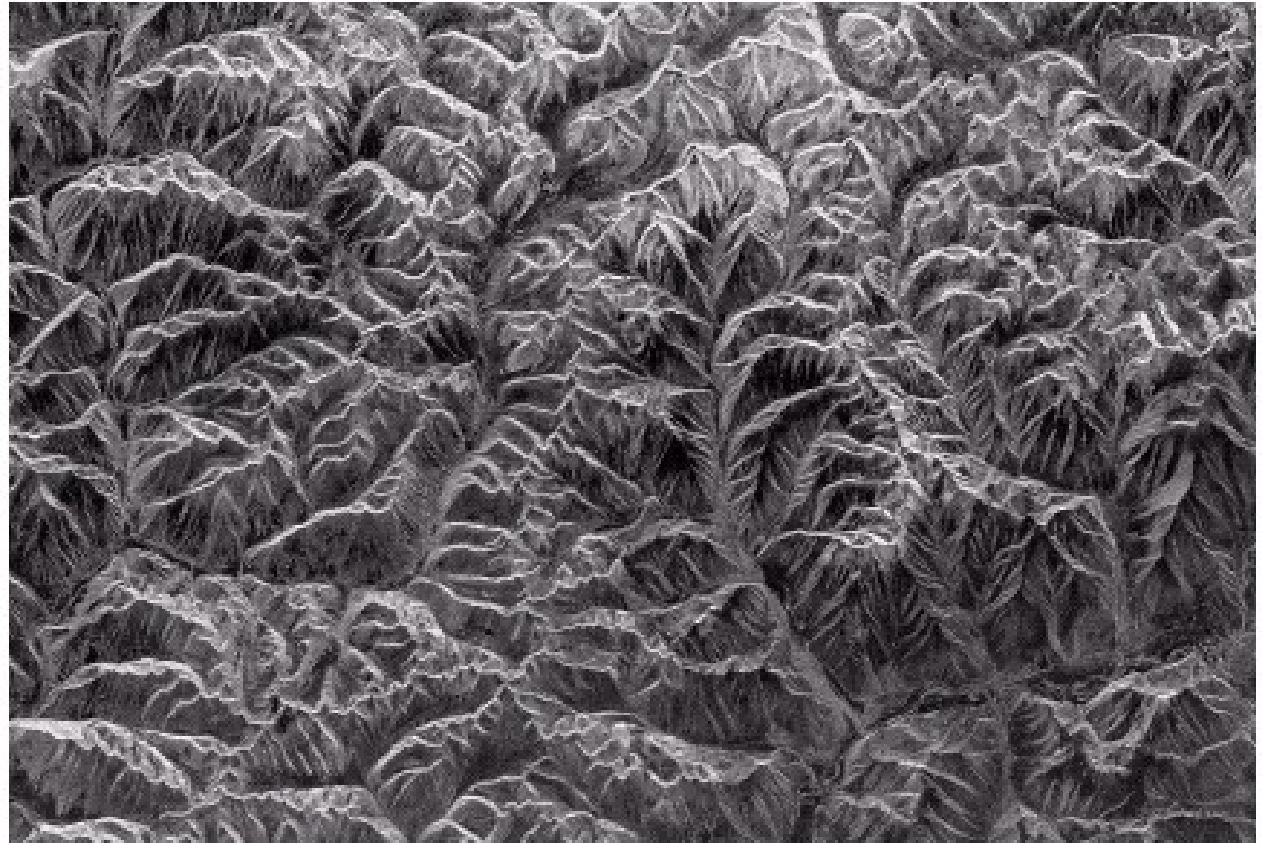
Examples: Infrared Satellite Imaging



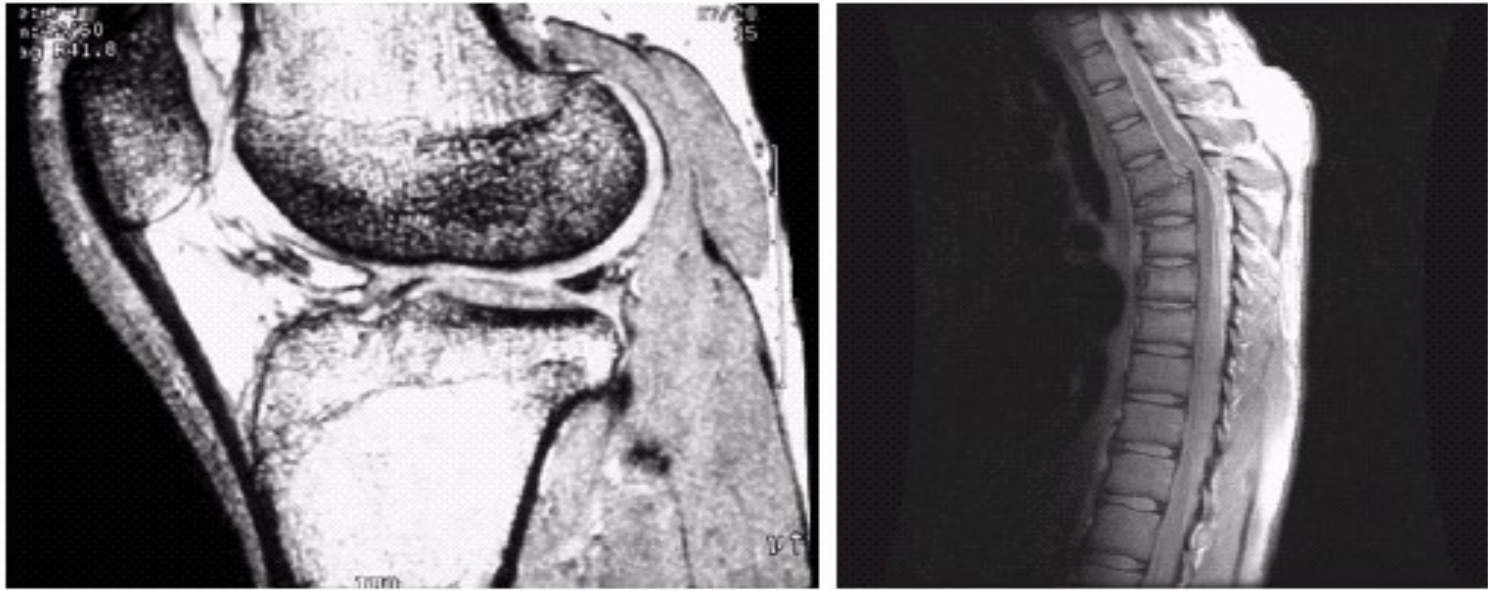
Example of Radar Image (Microwave)

FIGURE 1.16

Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)



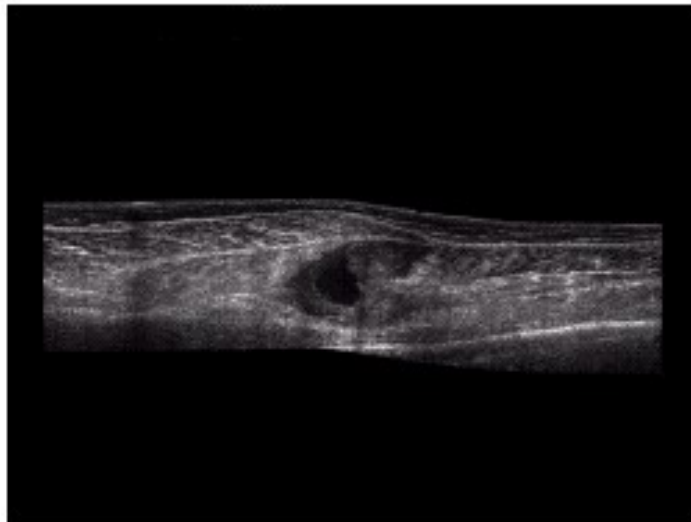
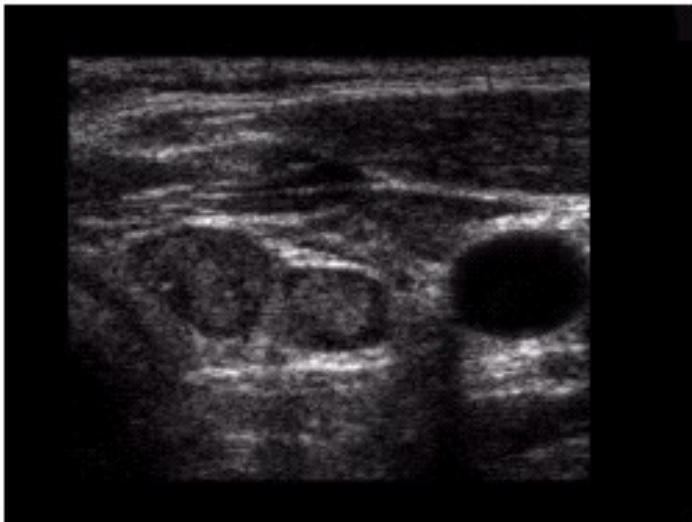
Examples: MRI(Magnetic Resonance Imaging) (Radio Band)



a b

FIGURE 1.17 MRI images of a human (a) knee, and (b) spine. (Image (a) courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, and (b) Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Examples: Ultrasound Imaging



a	b
c	d

FIGURE 1.20
Examples of
ultrasound
imaging. (a) Baby.
(2) Another view
of baby.
(c) Thyroids.
(d) Muscle layers
showing lesion.
(Courtesy of
Siemens Medical
Systems, Inc.,
Ultrasound
Group.)

Q and A

- ▶ What is order of wavelength of visible light?
- ▶ Which EM radiation is used in RADAR?



Image Acquisition Process

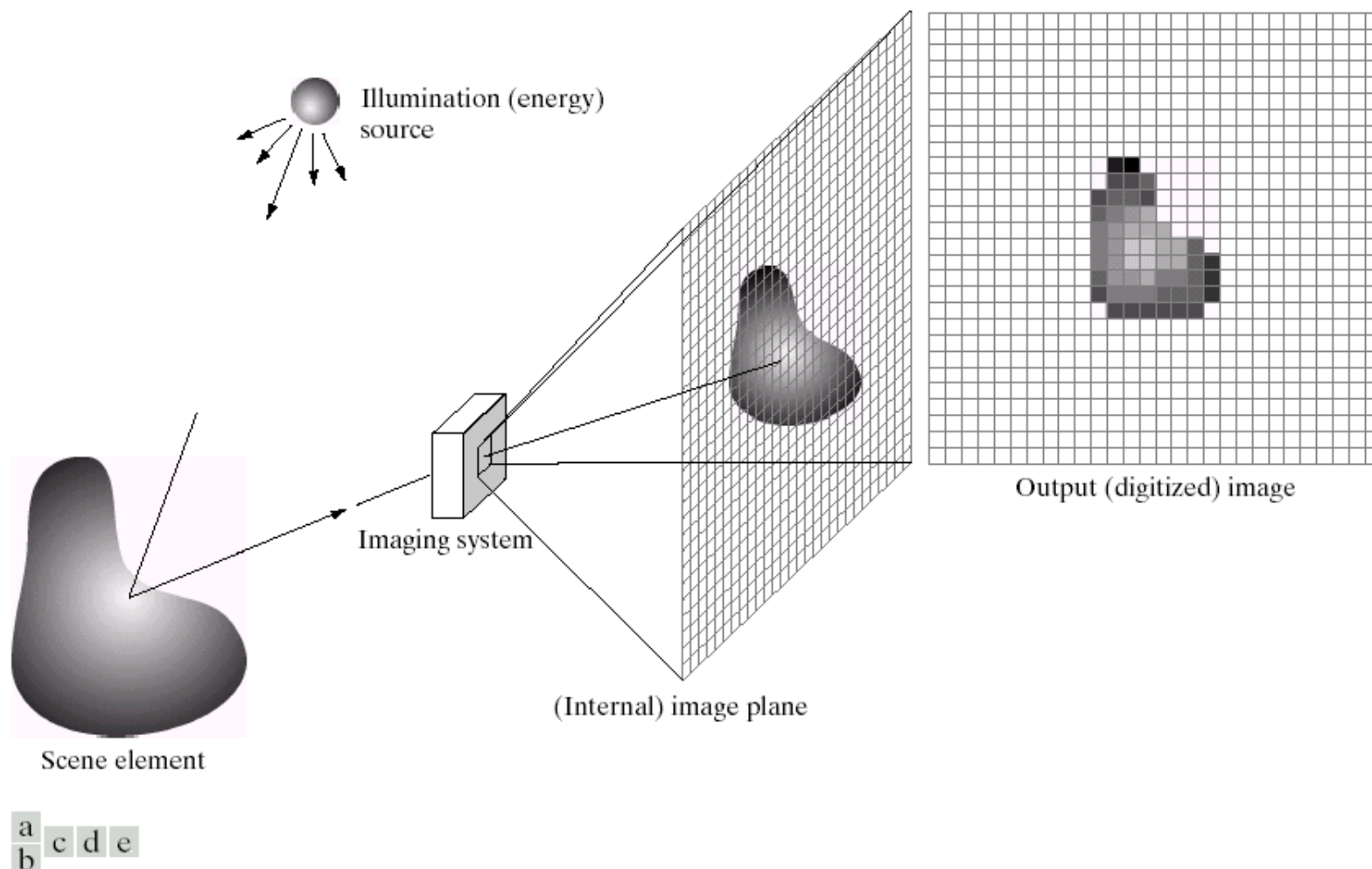


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

A Simple Image Formation Model

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

$f(x, y)$: intensity at the point (x, y)

$i(x, y)$: illumination at the point (x, y)

(the amount of source illumination incident on the scene)

$r(x, y)$: reflectance/transmissivity at the point (x, y)

(the amount of illumination reflected/transmitted by the object)

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

Important Terminologies

- ▶ In an image, **intensity** of a pixel is defined as the value of the pixel. For example in an 8 bit grayscale image there are 256 gray levels.
- ▶ **brightness**, is a relative term. Suppose A, B and C are three pixels having intensities 1, 30 and 250, then C is brighter and A & B are darker with respect to C. In general you can say the higher the intensity the brighter is the pixel.
- ▶ **Luminance** is the luminous intensity, projected on a given area and direction. Luminance is an objectively measurable attribute.
- ▶ The unit is 'Candela per Square Meter' (cd/m^2).
- ▶ It measures of the amount of light emitting, passing through or reflected from a particular surface from a solid angle.
- ▶ **contrast**. It is the difference between maximum and minimum pixel intensities in an image.
- ▶ Consider two images A having pixel intensities between 30 to 200 and B having pixel intensities 70 to 130. Then A has more contrast than B.

Some Typical Ranges of illumination

► Illumination

Lumen — A unit of light flow or luminous flux

Lumen per square meter (lm/m^2) — Illuminance is a term that describes the measurement of the **amount of light** falling onto (illuminating) and spreading over a given surface area.

- On a clear day, the sun may produce in excess of $90,000 \text{ lm}/\text{m}^2$ of illumination on the surface of the Earth
- On a cloudy day, the sun may produce less than $10,000 \text{ lm}/\text{m}^2$ of illumination on the surface of the Earth
- On a clear evening, the moon yields about $0.1 \text{ lm}/\text{m}^2$ of illumination
- The typical illumination level in a commercial office is about $1000 \text{ lm}/\text{m}^2$

Some Typical Ranges of Reflectance

► Reflectance

- 0.01 for black velvet
- 0.65 for stainless steel
- 0.80 for flat-white wall paint
- 0.90 for silver-plated metal
- 0.93 for snow

Image formation on analog cameras

- ▶ In analog cameras , the image formation is due to the chemical reaction that takes place on the strip that is used for image formation.
- ▶ It is denoted in the figure by 35mm film cartridge. This strip is coated with silver halide (a chemical substance).
- ▶ So when light photon particles are passed through the camera, it reacts with the silver halide particles on the strip and it results in the silver which is the negative of the image.
- ▶ although image formation involves many other concepts like shutter and shutter speed and aperture and its opening.



▶ **Aperture**

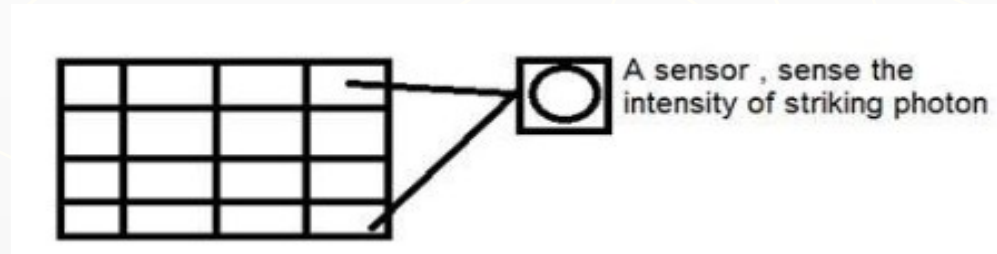
- ▶ Aperture is a small opening which allows the light to travel inside into camera. Here is the picture of aperture
- ▶ If the aperture opening is wide, it would allow more light to pass into the camera. More light would result in more photons, which ultimately result in a brighter image.

▶ **Shutter:**

- ▶ After the aperture, there comes the shutter just before the CCD sensor. Shutter has further two main concepts: Shutter Speed & Shutter time
- ▶ More shutter speed = less shutter time.
- ▶ To capture image of a fast moving object: increase shutter speed & Decrease shutter time.

Image formation on digital cameras

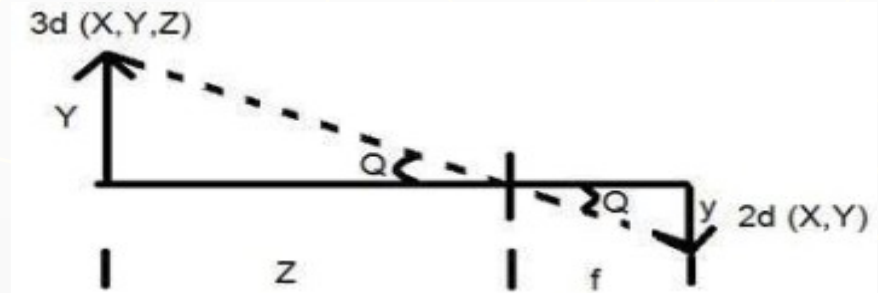
- ▶ In the digital camera , a CCD array of sensors is used for the image formation.
- ▶ CCD stands for charge-coupled device.
- ▶ This CCD is actually in the shape of array or a rectangular grid.



- Each sensor of the CCD array itself is an analog sensor. When photons of light strike on the chip , **it is held as a small electrical charge in each photo sensor.** The response of each sensor is directly equal to the amount of light or (photon) energy strike on the surface of the sensor.
- The value of each sensor of the CCD array refers to each the value of the individual pixel.
- The number of sensors = number of pixels.

Calculating the size of image formed

- ▶ Suppose a person 5m tall, and standing at a distance of 50m from the camera, and we have to tell that what is the size of the image of the person, with a camera of focal length is 50mm.
- ▶ Solution:
- ▶ $Y = 5000$ mm.
- ▶ $f = 50$ mm.
- ▶ $Z = 50000$ mm.
- ▶ Putting the values in the formula, we get



$$Y = -f \frac{Y}{Z} = -50 \times 5000 / 50000$$

$$= -5 \text{ mm.}$$

Q & A

- ▶ What is intensity in an image?
- ▶ Full form of CCD
- ▶ What is difference between optical zoom and digital zoom.

Digitization

- Real world is continuous , The computer world is discrete
- Mapping a continuous function to a discrete one is called sampling
- Mapping a continuous variable to a discrete one is called quantizaion
- To represent or render an image using a computer, we must both sample and quantize

► Two stages in the digitization process:

- **Spatial sampling:** Spatial domain
- **Quantization:** Gray level

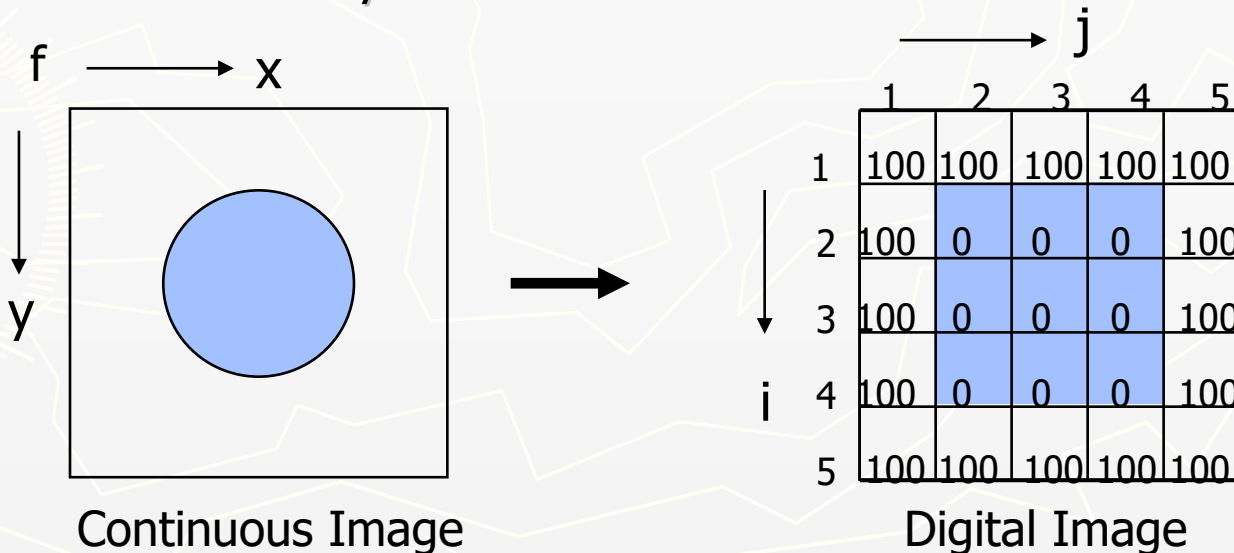


Image Sampling and Quantization

You can see in this image, that the signal has been quantified into four different levels.

And the vertically ranging values have been quantized into 5 different levels or partitions.

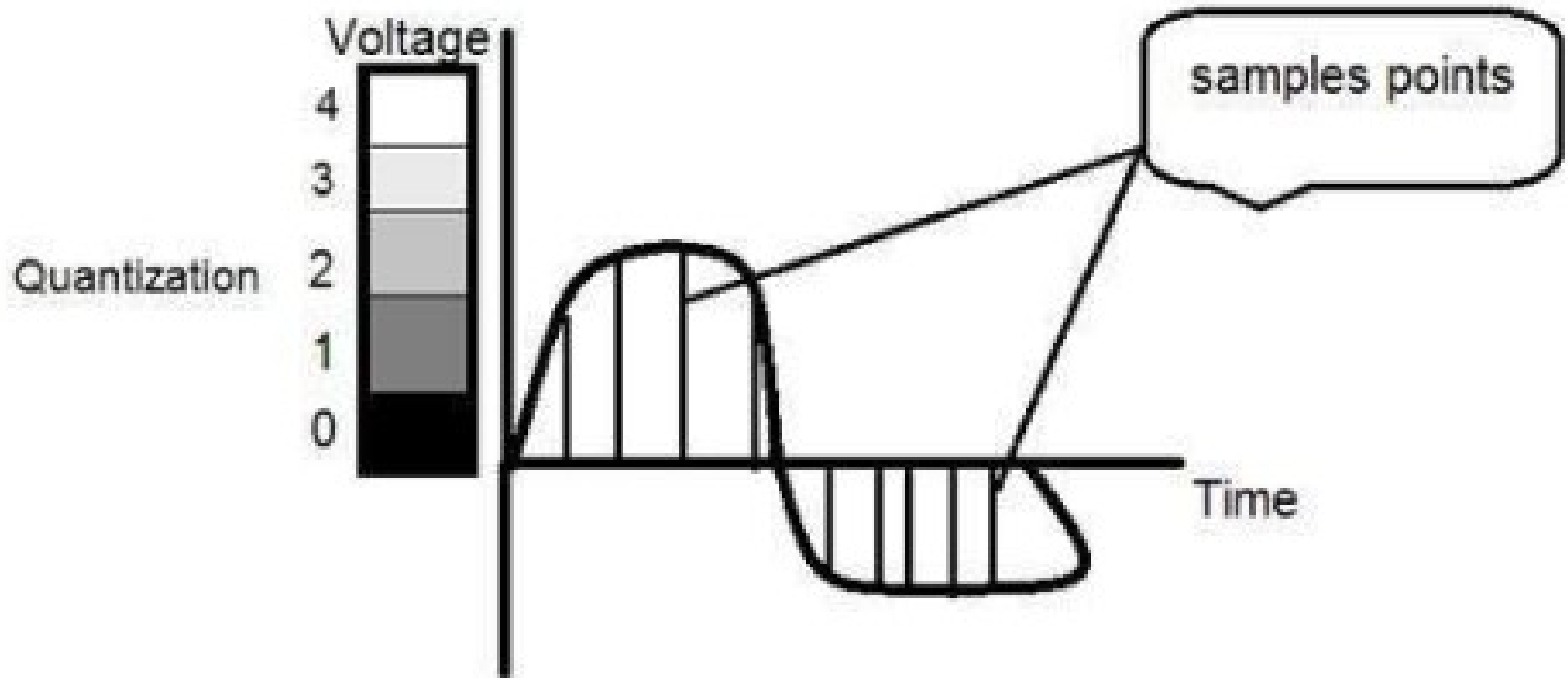
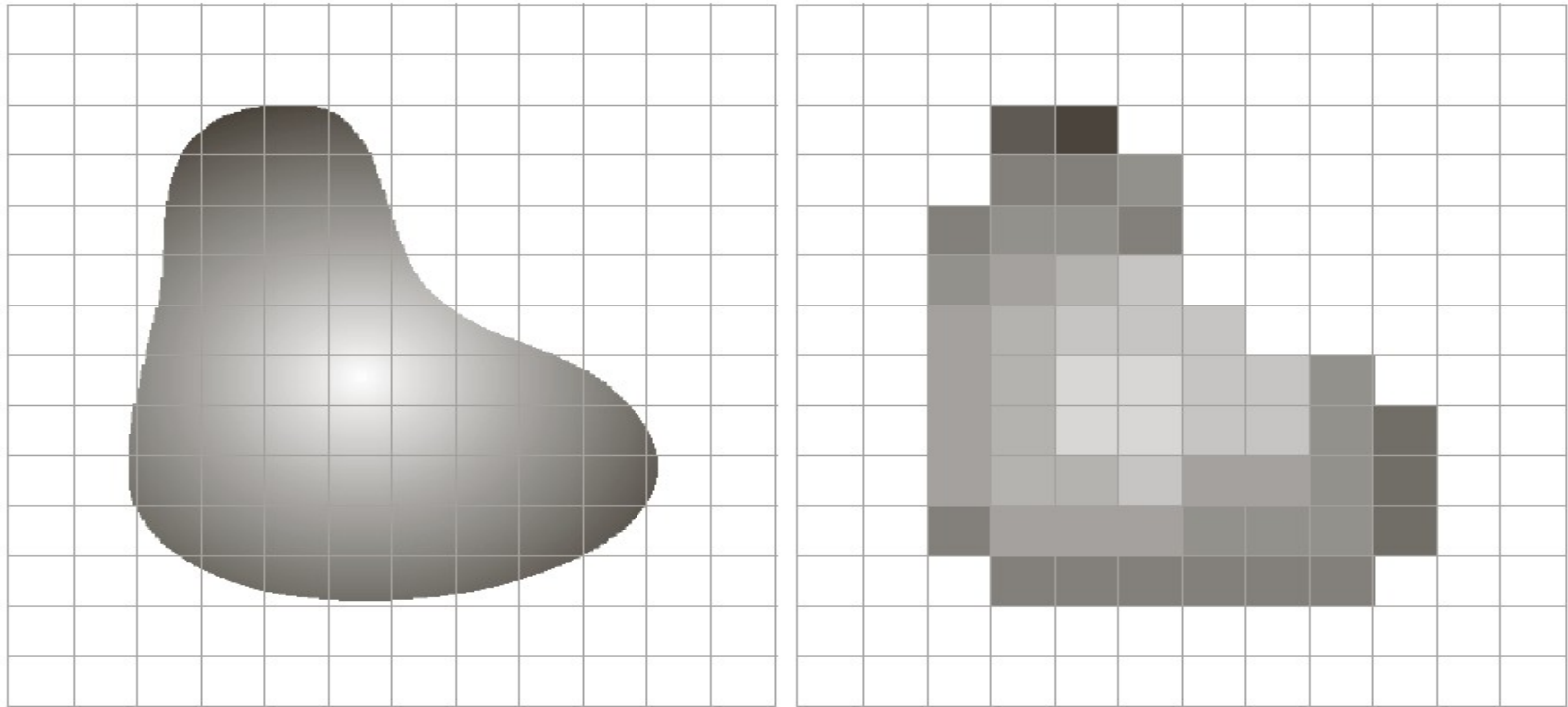


Image Sampling and Quantization



a b

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

The sampling rate determines the spatial resolution of the digitized image, while the quantization level determines the number of grey level resolution in the digitized image

Quantization

256



64



16



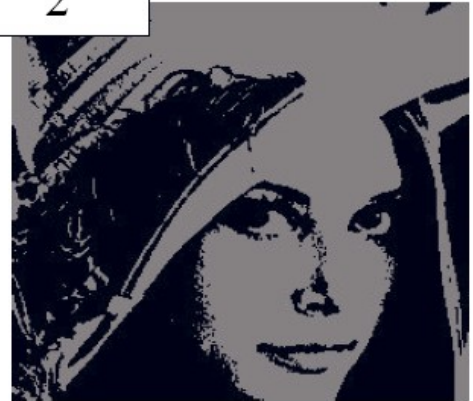
8



4



2



Representing Digital Images

- The representation of an $M \times N$ numerical array as

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

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

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

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

Representing Digital Images

TABLE 2.1

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

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

Resolution

- ▶ The resolution can be defined in many ways. Such as **pixel resolution, spatial resolution (sampling), Intensity (Gray level) resolution, temporal resolution.**
- ▶ **pixel resolution:** You have probably seen that in your own computer settings, you have monitor resolution of 800 x 600, 640 x 480 e.t.c
- ▶ In pixel resolution, the term resolution refers to the total number of count of pixels in an digital image.
- ▶ For example. If an image has M rows and N columns, then its resolution can be defined as $M \times N$.

Resolution (Cont..)

▶ **Megapixels**

- ▶ We can calculate mega pixels of a camera using pixel resolution.
- ▶ Column pixels (width) X row pixels (height) / 1 Million.
- ▶ The size of an image can be defined by its pixel resolution.
- ▶ Size = pixel resolution X bpp (bits per pixel)

▶ **Calculating the mega pixels of the camera**

- ▶ Lets say we have an image of dimension: 2500 X 3192.
- ▶ Its pixel resolution = $2500 * 3192 = 7982350$ bytes.
- ▶ Dividing it by 1 million = $7.9 = 8$ mega pixel (approximately).

▶ **Aspect ratio**

- ▶ Another important concept with the pixel resolution is aspect ratio.
- ▶ Aspect ratio is the ratio between width of an image and the height of an image. 1.33:1, 1.37:1, 1.50:1, 1.56:1, 1.66:1, 1.75:1, e.t.c

Resolution(Cont..)

- ▶ If you are given an image with aspect ratio of 6:2 of an image of pixel resolution of 480000 pixels given the image is an gray scale image.

- ▶ calculate the dimensions of image

- ▶ Given:

- ▶ Aspect ratio: $c:r = 6:2$

- ▶ Pixel resolution:

$$c * r = 480000$$

- ▶ Bits per pixel:

grayscale image = 8bpp

$$\text{Size} = \text{rows} * \text{cols} * \text{bpp}$$

Size of image in bits

$$= 400 * 1200 * 8$$

$$= 3840000 \text{ bits}$$

$$= 48 \text{ kb (approx).}$$

$$\text{Equation 1.} \quad c:r = 6:2 \rightarrow c = 6r / 2$$

$$\text{Equation 2.} \quad c = 480000/r$$

$$\text{Comparing both equations} \rightarrow \frac{6r}{2} = \frac{480000}{r}$$

$$r^2 = \sqrt{\frac{480000 * 2}{6}}$$

$$\text{That gives} \quad r = 400.$$

$$\text{Put } r \text{ in equation 1, we get} \rightarrow c = 1200.$$

$$\text{So rows} = 400 \text{ cols} = 1200.$$

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.*

Q & A

- ▶ What is difference between spatial resolution and pixel resolution?
- ▶ If you are given an image with aspect ratio of 5:3 of an image of pixel resolution of 450000 pixels given the image is an gray scale image. calculate the dimensions of image.
- ▶ What is Intensity resolution for binary image?

Basic Relationships Between Pixels

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

Basic Relationships Between Pixels

- ▶ **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 $\mathbf{N}_8(p)$
$$\mathbf{N}_8(p) = \mathbf{N}_4(p) \cup \mathbf{N}_D(p)$$

Basic Relationships Between Pixels

► Adjacency

Let V 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 $N_8(p)$.

Basic Relationships Between Pixels

► 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 .

Basic Relationships Between Pixels

► 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

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

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

- 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

$$\mathbf{V} = \{1, 2\}$$

0 1 1

0 2 0

0 0 1

0 1 1

0 2 0

0 0 1

0 1 1

0 2 0

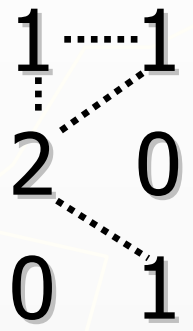
0 0 1

Examples: Adjacency and Path

$$V = \{1, 2\}$$

0	1	1
0	2	0
0	0	1

0	1	1
0	2	0
0	0	1



0	1	1
0	2	0
0	0	1

8-adjacent

Examples: Adjacency and Path

$$V = \{1, 2\}$$

0	1	1
0	2	0
0	0	1

0	1	1
0	2	0
0	0	1

8-adjacent

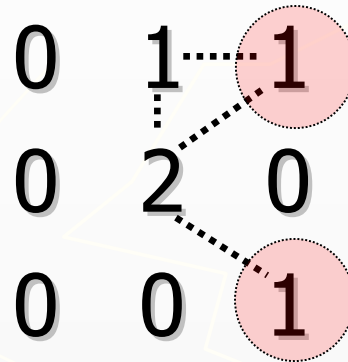
0	1	1
0	2	0
0	0	1

m-adjacent

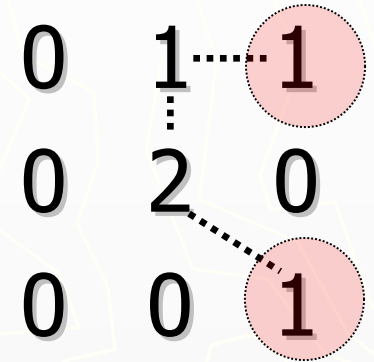
Examples: Adjacency and Path

$$V = \{1, 2\}$$

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)

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

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

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

Basic Relationships Between Pixels

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***.

Basic Relationships Between Pixels

► **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 , $k = 1, 2, \dots, K$. 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**.

Q & A

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



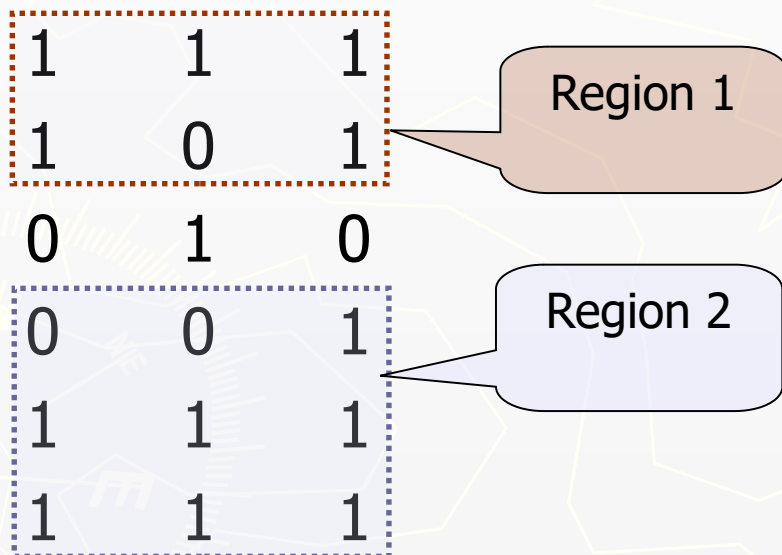
1	1	1
1	0	1
0	1	0
0	0	1
1	1	1
1	1	1

Region 1

Region 2

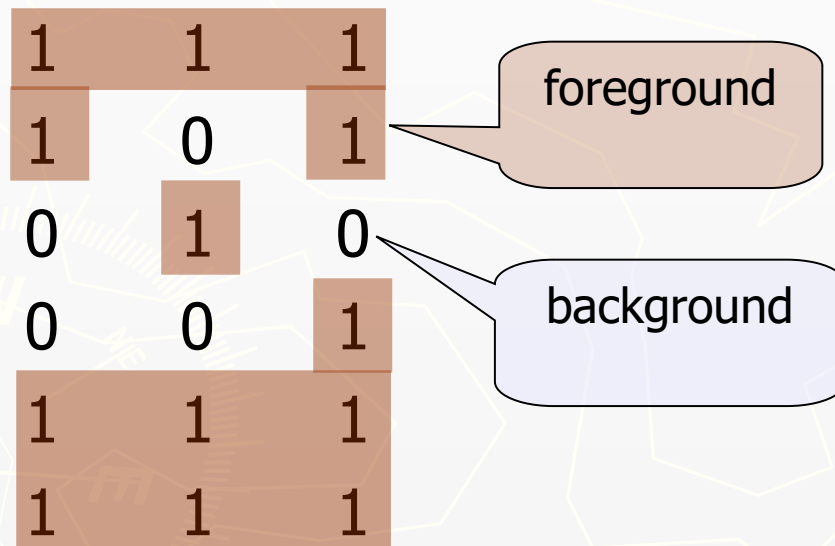
Question 3

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



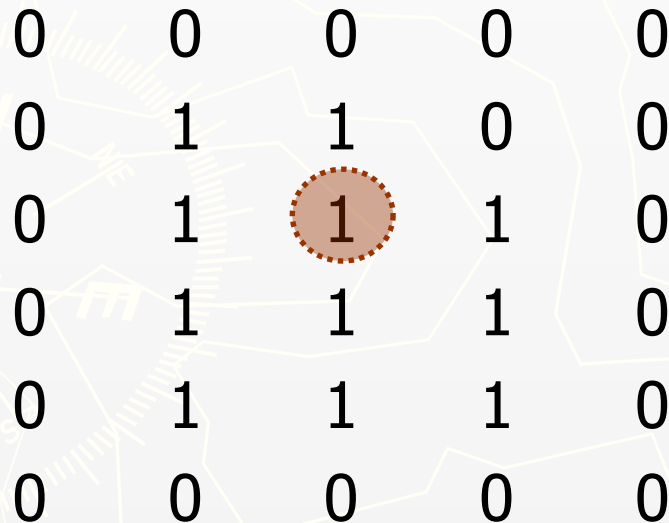
Question 4

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



Question 5

- ▶ In the following arrangement of pixels, the circled point is part of the boundary of the 1-valued pixels
 1. if **8-adjacency** is used,
 2. if **4-adjacency** is used, true or false?



0	0	0	0	0
0	1	1	0	0
0	1	1	1	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. $D(p, q) \geq 0$ $[D(p, q) = 0, \text{ iff } p = q]$

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 :

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

b. City Block Distance:

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

c. Chess Board Distance:

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

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

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

Introduction to Mathematical Operations in DIP

► Array vs. Matrix Operation

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

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

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

Introduction to Mathematical Operations in DIP

► Linear vs. Nonlinear Operation

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

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

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

Additivity

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

Homogeneity

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

H is said to be a **linear operator (sum, average, subtraction)**;

H is said to be a **nonlinear operator (mult, div)** if it does not meet the above qualification.

Arithmetic Operations

- ▶ Arithmetic operations between images are array operations. The four arithmetic operations are denoted as

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

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

$$p(x,y) = f(x,y) \times g(x,y)$$

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

Example: 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: $g(x,y)$

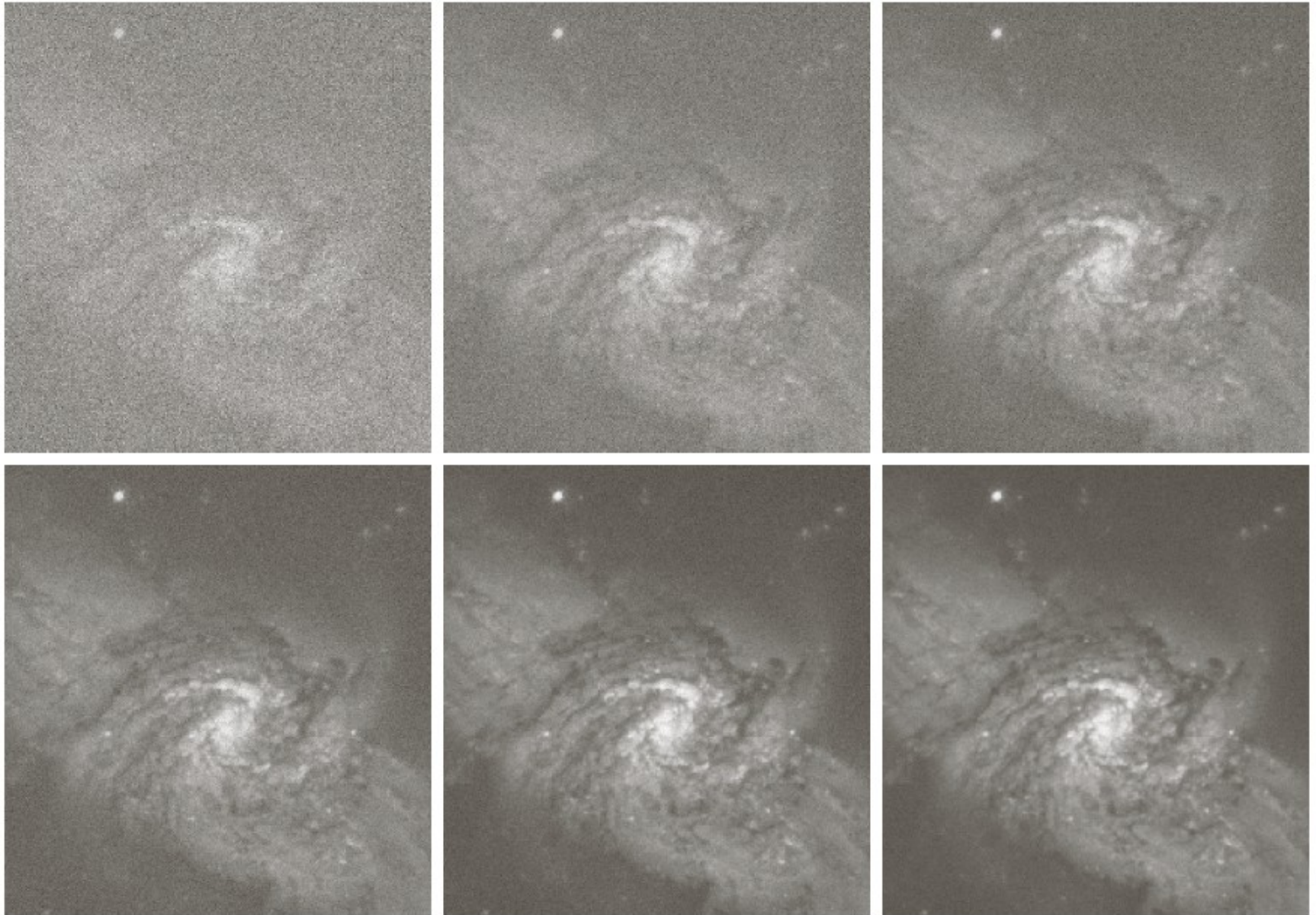
$$g(x,y) = f(x,y) + n(x,y)$$

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

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

Application: Addition of Noisy Images for Noise Reduction

- ▶ In astronomy, imaging under very low light levels frequently causes sensor noise to render single image virtually useless for analysis.
- ▶ 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.)

An Example of Image Subtraction: Mask Mode Radiography

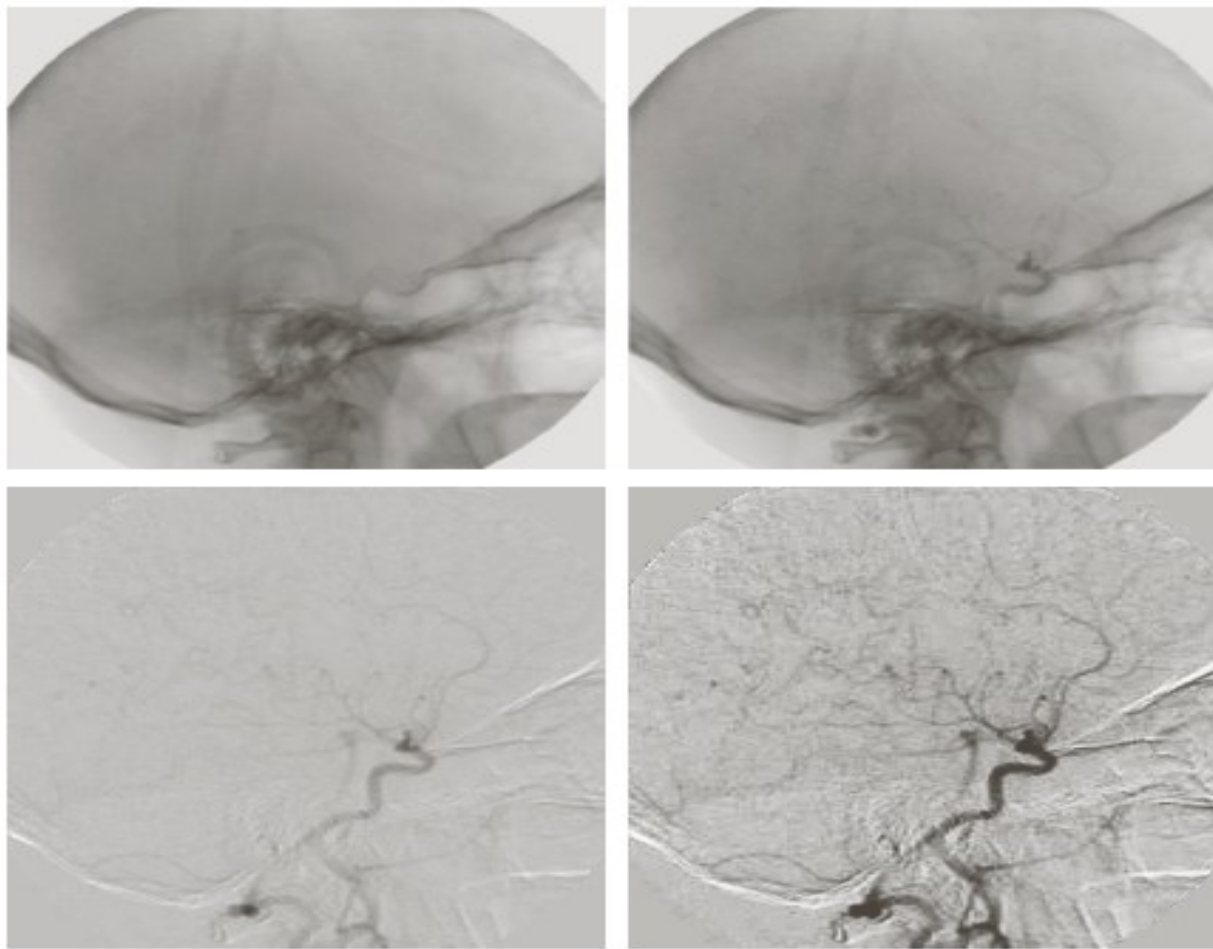
Mask $h(x,y)$: an X-ray image of a region of a patient's body

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.



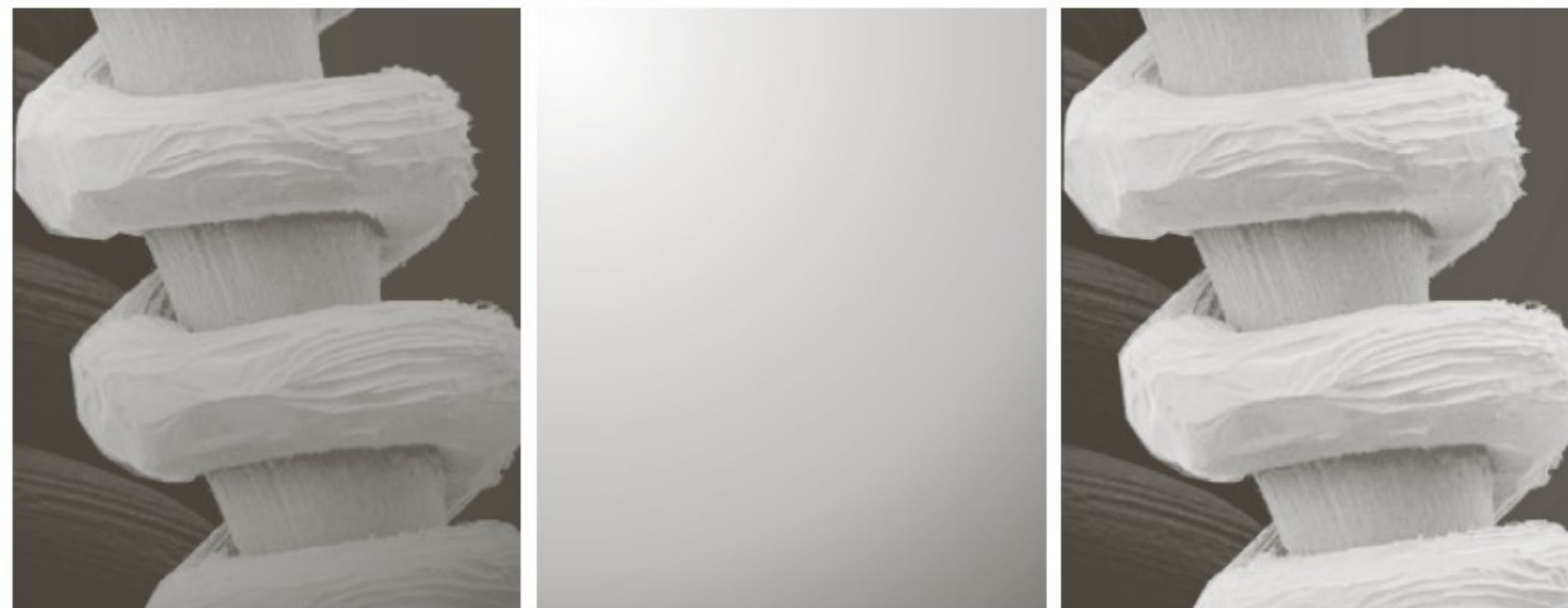
a	b
c	d

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.)

- ❑ A mask X-ray (using intensified TV camera) image of a patient's head prior to injection of an iodine (X-ray contrast) medium into bloodstream, (b) a sample of live image taken after the injection (c) difference between two (d) enhanced image
- ❑ Angiography or arteriography is a medical imaging technique used to visualize the inside, or lumen, of blood vessels and organs of the body, with particular interest in the arteries, veins, and the heart chambers.

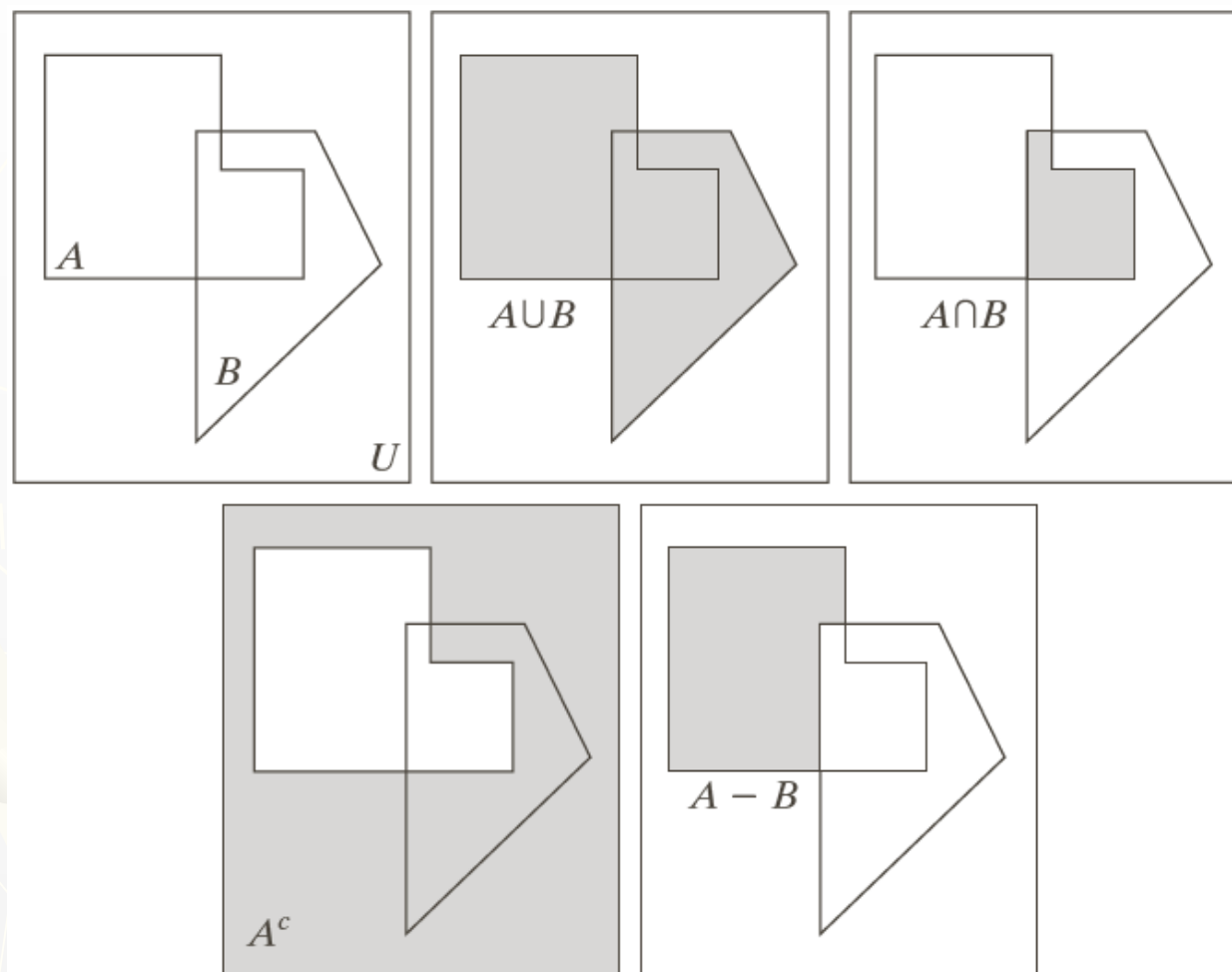
An Example of Image Multiplication (shading correction & masking)



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



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.

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

Set and Logical Operations

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

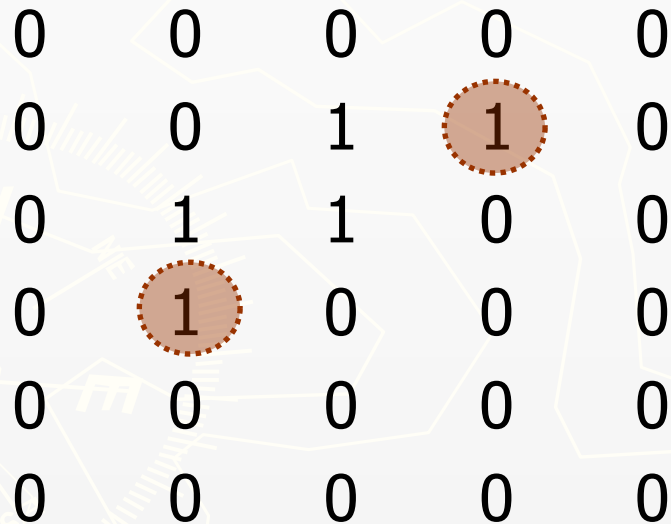
$$A \cup B = \{\max_z(a, b) \mid a \in A, b \in B\}$$

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.)

Question 1

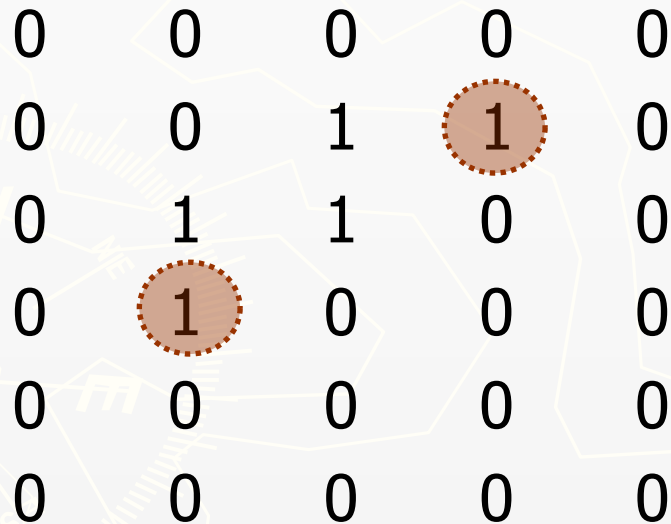
- In the following arrangement of pixels, what's the value of the chessboard distance between the circled two points?



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 2

- 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 3

- ▶ Write Python code to add, subtract, and multiply two images.

Thank You



Lecture 2. Intensity Transformation and Spatial Filtering



Geometric Spatial Transformations

- ▶ Geometric transformation (rubber-sheet transformation)
 - A **spatial transformation** of coordinates

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

- **intensity interpolation** that assigns intensity values to the spatially transformed pixels.

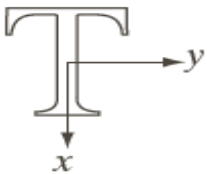
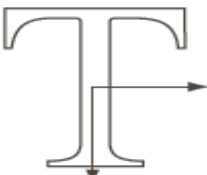
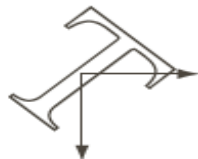
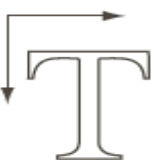
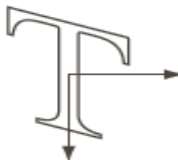
- ▶ Affine transform

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

Application:

1. Rescale the graphics content when displaying it on a desktop vs mobile.
2. Artificially generate more data is to randomly apply an affine transformation to the input data. A technique also known as **augmentation**

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

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

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.

$$(x, y) = T[(w, z)] = (w/2, z/2)$$

► Inverse Mapping

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

$$(w, z) = T^{-1}[(x, y)] = (2x, 2y)$$

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

Inverse mappings are more efficient to implement than forward mappings.

Intensity Transform: Spatial Domain vs. Transform Domain

► Spatial domain

image plane itself, directly process the intensity values of the image plane

► Transform domain

process the transform coefficients, not directly process the intensity values of the image plane

Spatial Domain Process

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

$f(x, y)$: input image

$g(x, y)$: output image

T : an operator on f defined over
a neighborhood of point (x, y)

Spatial Domain Process

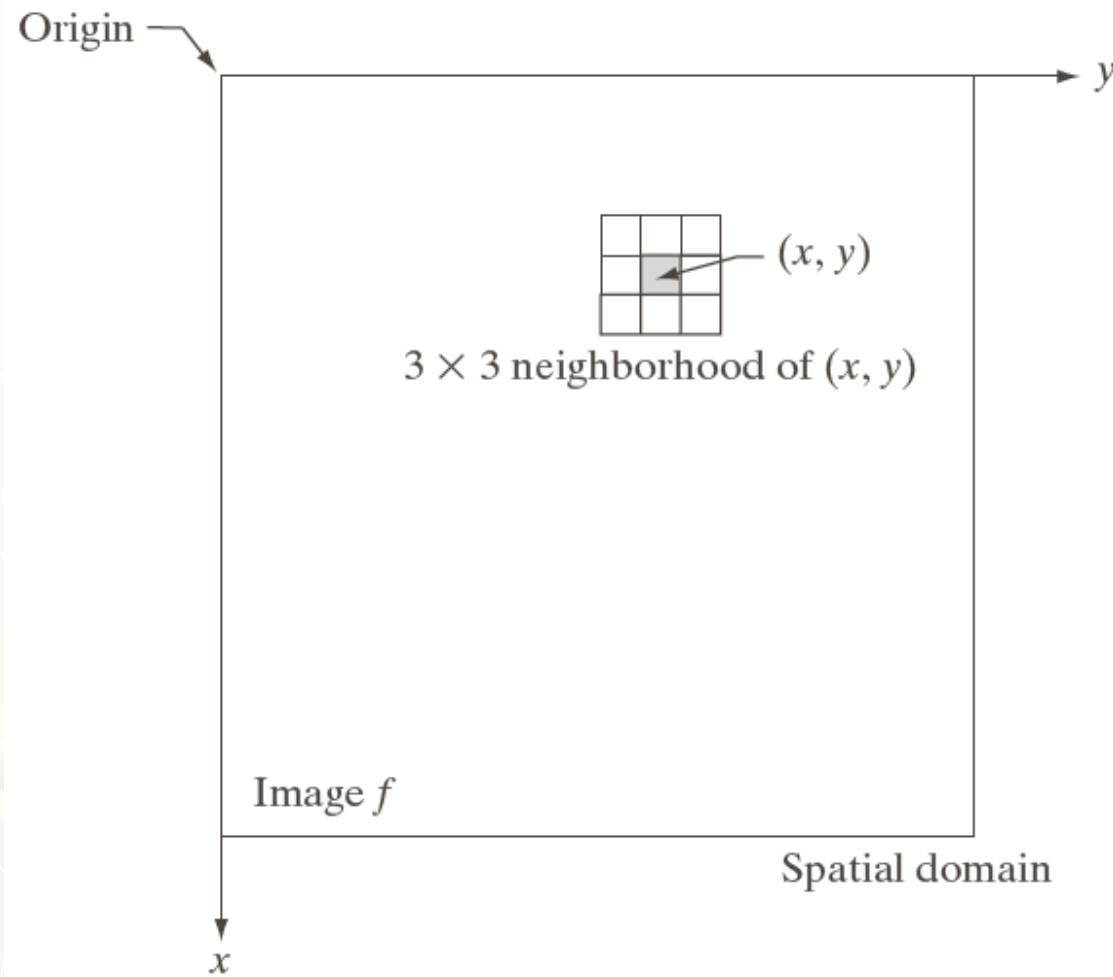


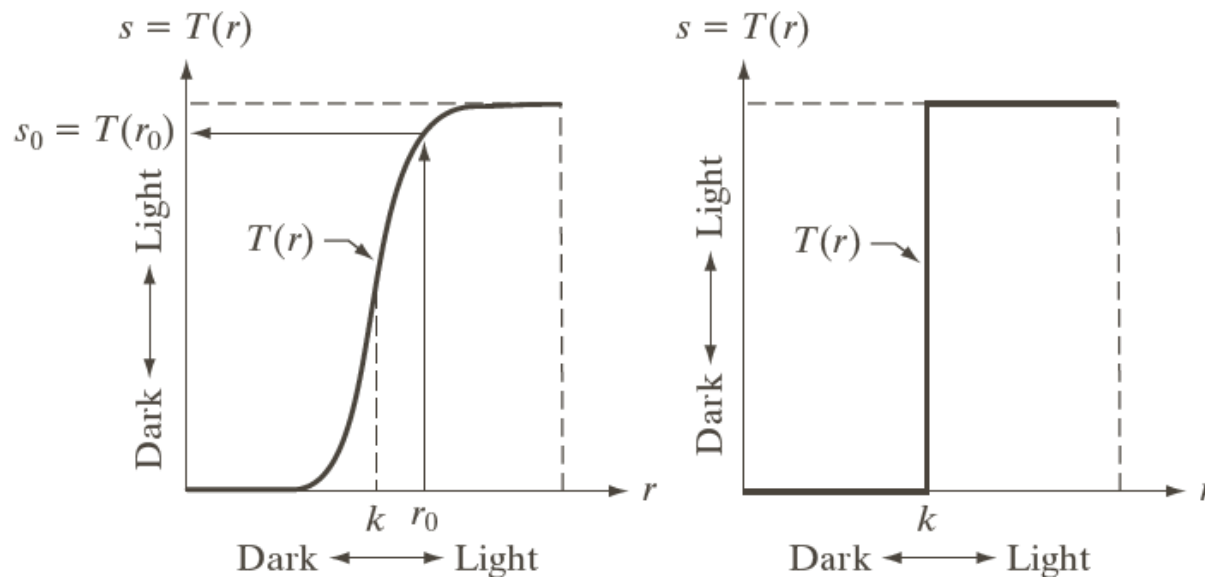
FIGURE 3.1

A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Spatial Domain Process

Intensity transformation function

$$s = T(r)$$



a b

FIGURE 3.2

Intensity transformation functions.

(a) Contrast-stretching function.

(b) Thresholding function.

In contrast stretching, if $r_1=s_1$ and $r_2=s_2$, then transformation is a linear function that produces no changes in gray levels

Example: Image Negatives

Image negatives

$$s = L - 1 - r$$

a b

FIGURE 3.4

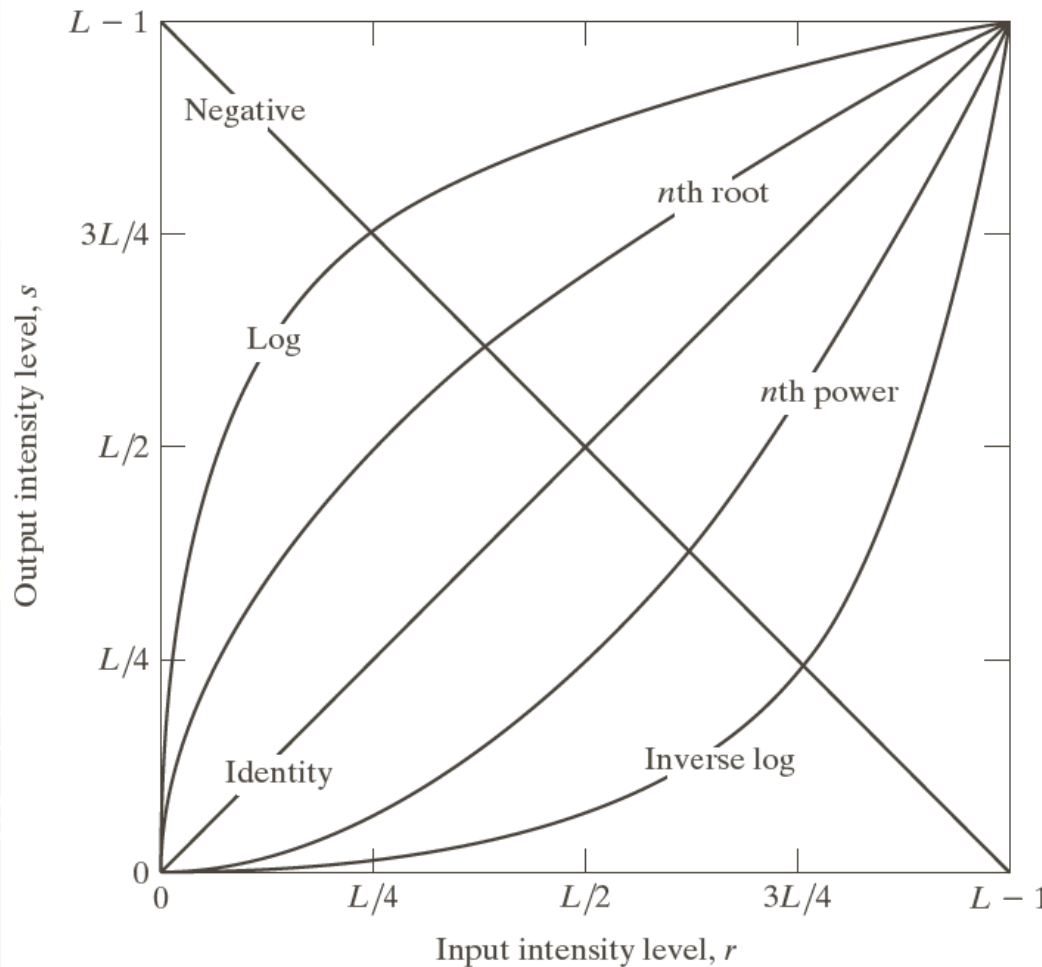
(a) Original digital mammogram.

(b) Negative image obtained using the negative transformation in Eq. (3.2-1).

(Courtesy of G.E. Medical Systems.)

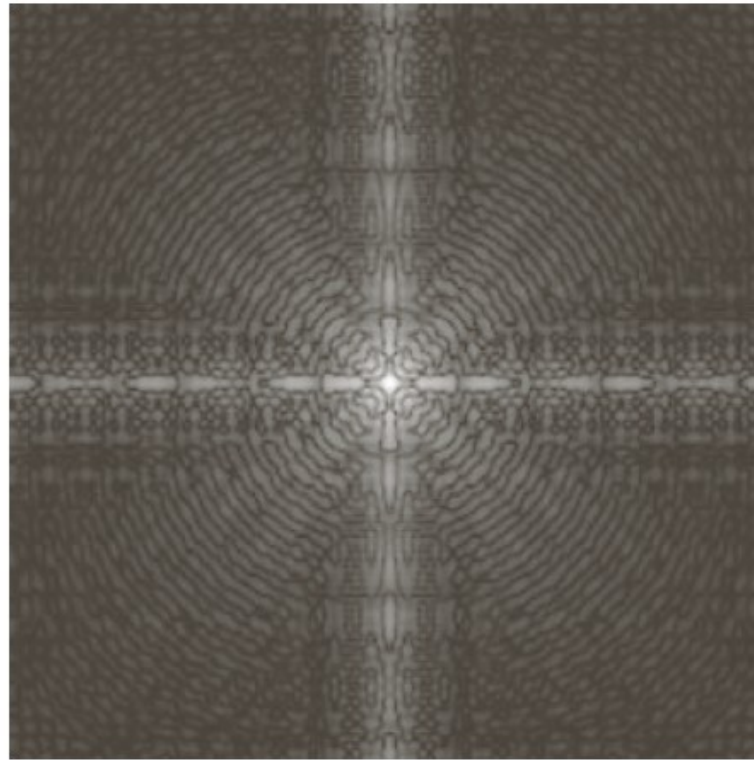
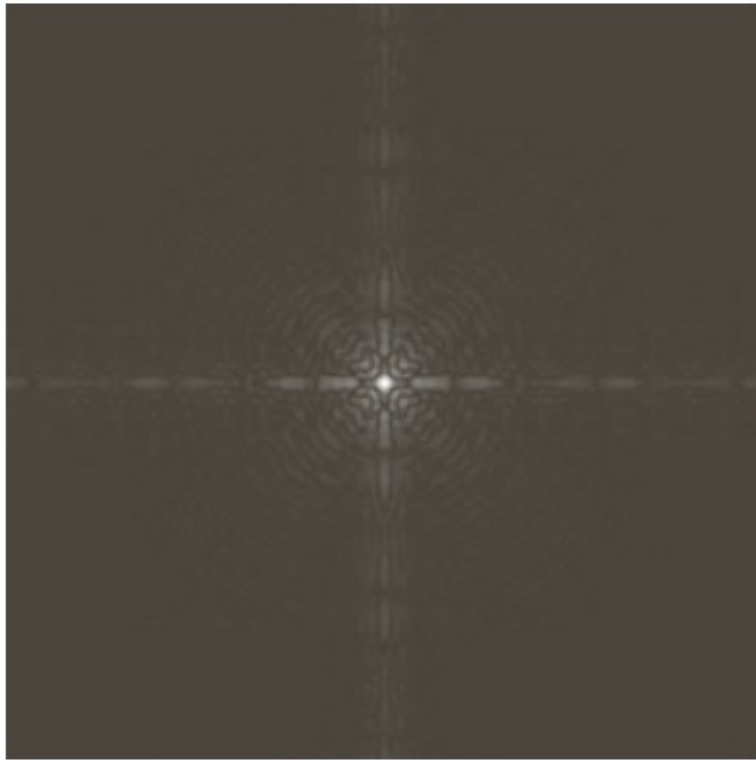
Small
lesion

Log Transformations



Log Transformations
 $s = c \log(1 + r)$

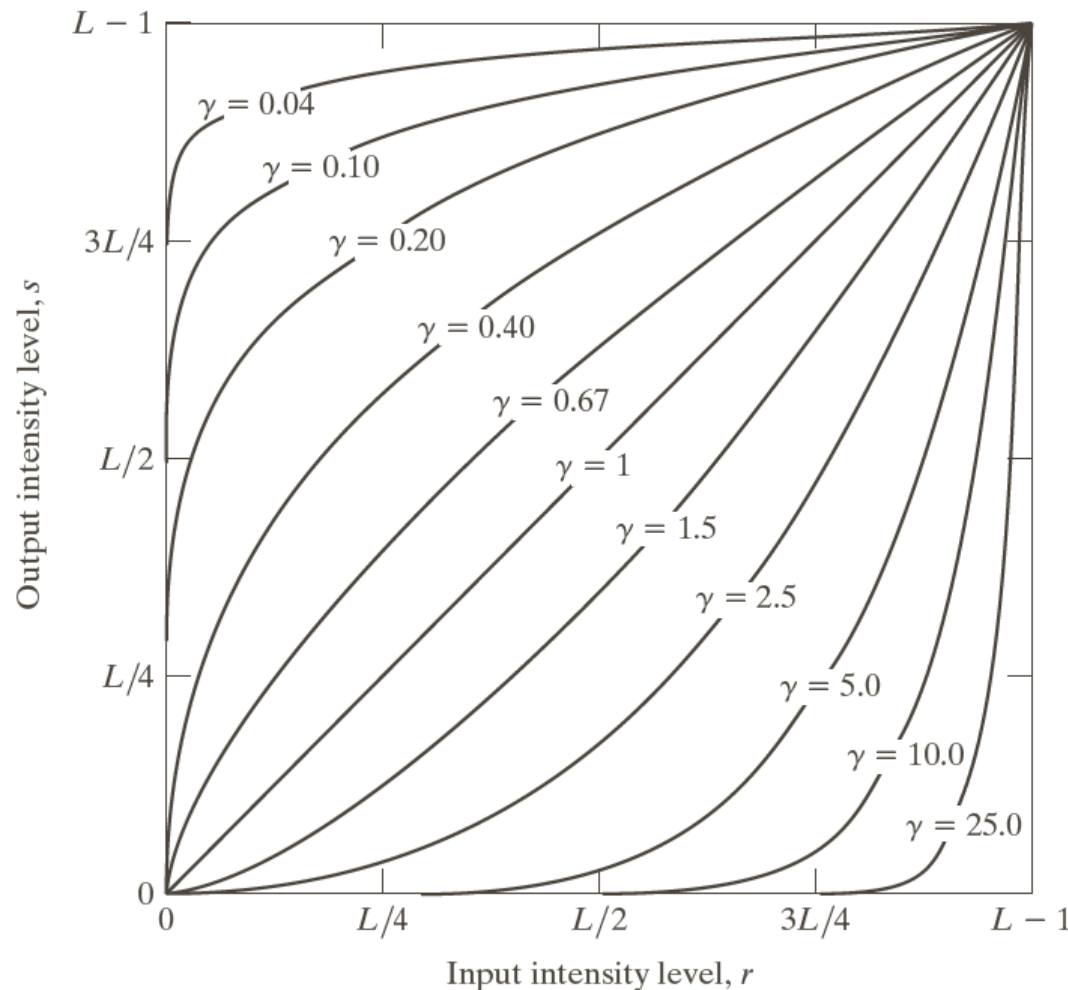
Example: Log Transformations



a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

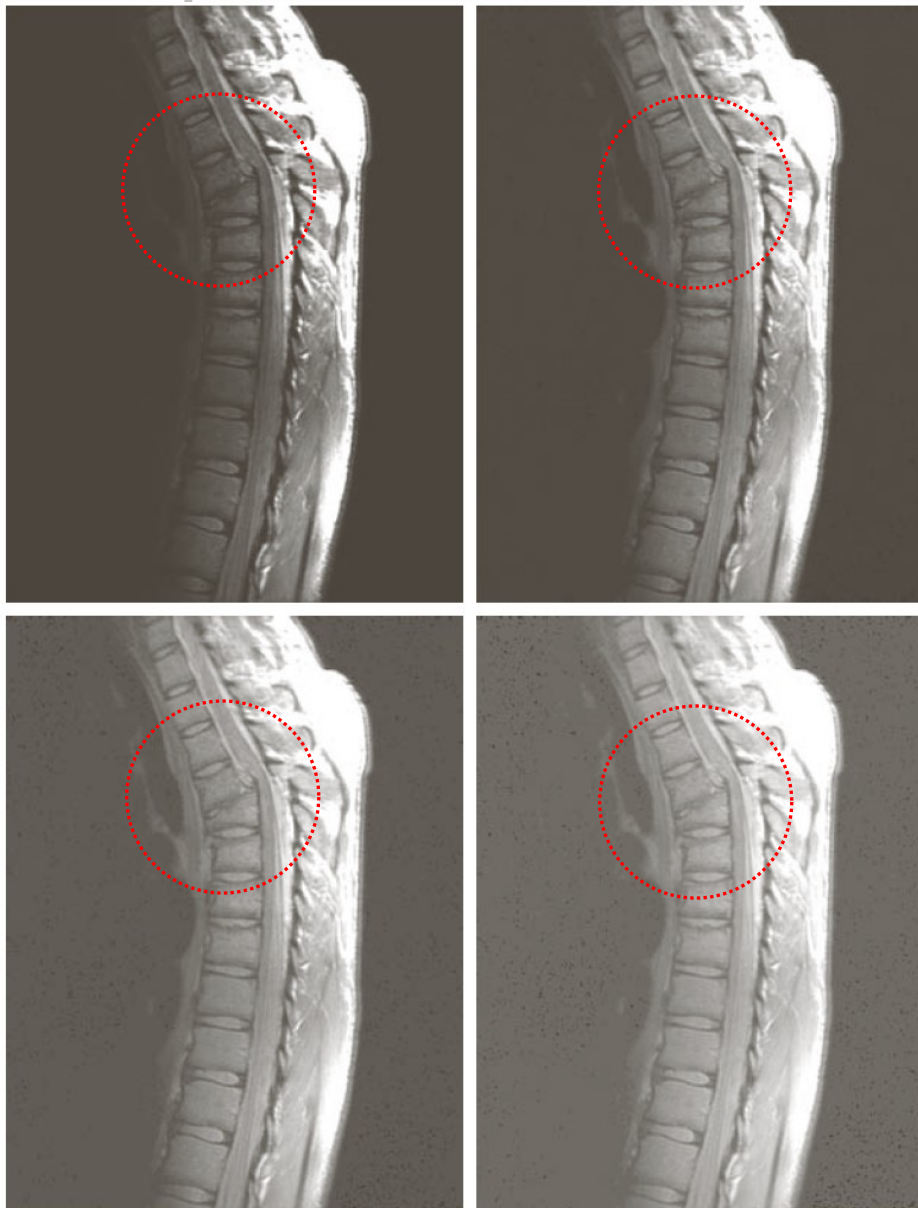
Power-Law (Gamma) Transformations



$$s = cr^\gamma$$

FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Example: Gamma Transformations



a	b
c	d

FIGURE 3.8

(a) Magnetic resonance image (MRI) of a fractured human spine.

(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3 , respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

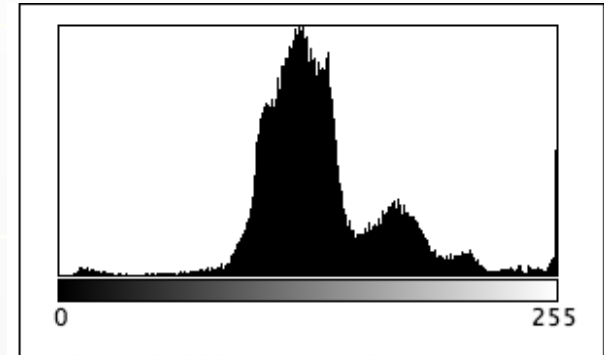
Histogram Processing

► Histogram Equalization

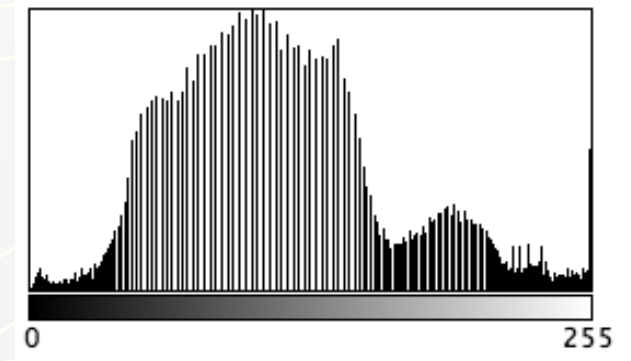
Example: Histogram Equalization

Histogram equalization yields an image whose pixels are (in theory) uniformly distributed among all gray levels.

► Original



► Modified



Example: Histogram Equalization

Suppose that a 3-bit image ($L=8$) of size 3×4 pixels ($MN = 12$) has the intensity distribution shown in following table.

Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k .

A=

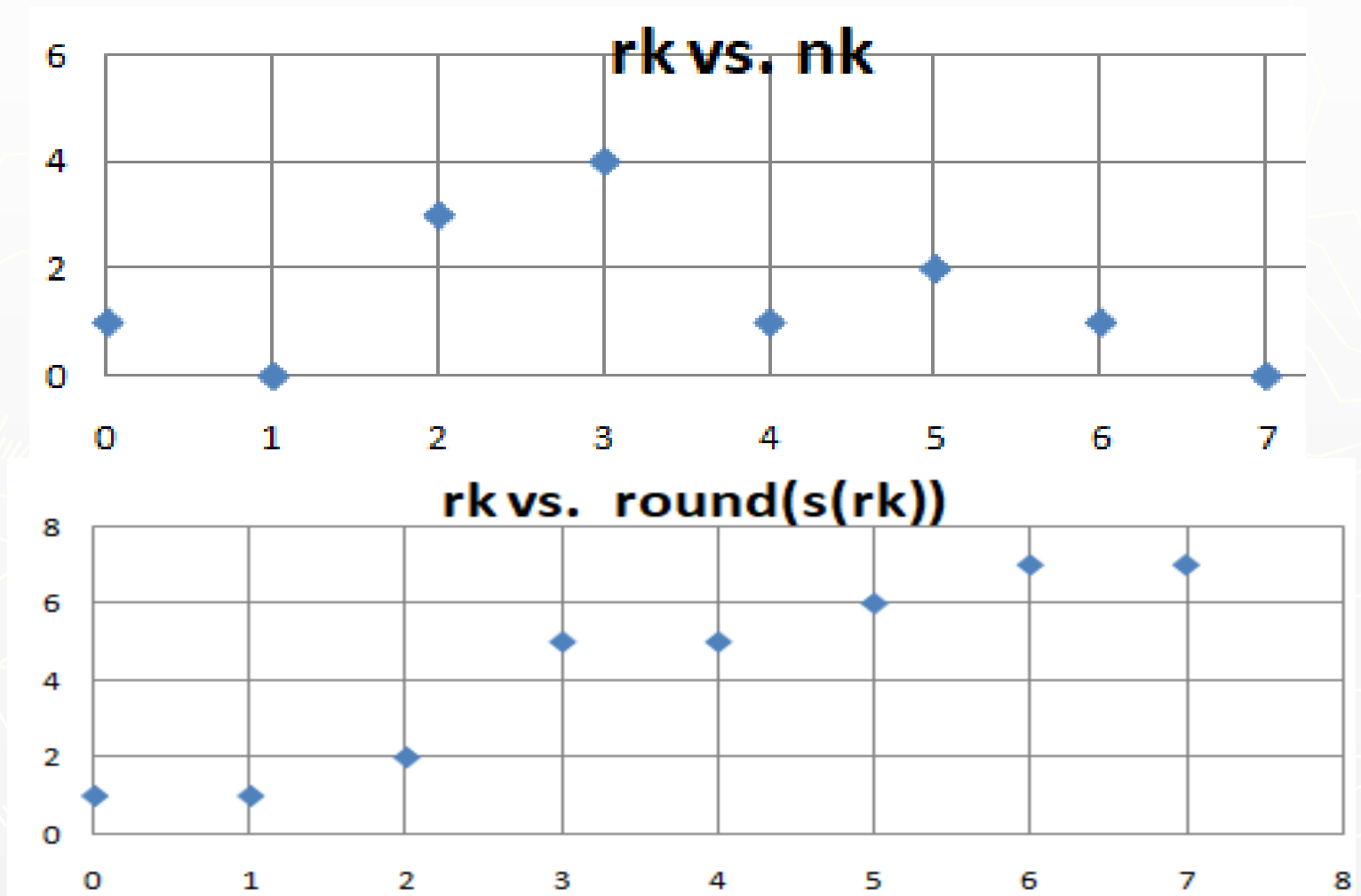
2	3	5
5	6	2
2	3	4
3	3	0

Equalized image look
up table
 $\text{img} = \text{cdf}[A]$

n= 12 rk=[0-7] gray level=L=8

rk	nk	$p(\text{rk}) = \text{nk}/n$	$T(\text{rk}) = 7 * P(\text{rk})$	$S(\text{rk}) = \text{cdf}(T(\text{rk}))$	$\text{floor}(s(\text{rk}))$
0	1	0.08	0.58	0.58	0
1	0	0.00	0.00	0.58	0
2	3	0.25	1.75	2.33	2
3	4	0.33	2.33	4.67	4
4	1	0.08	0.58	5.25	5
5	2	0.17	1.17	6.42	6
6	1	0.08	0.58	7.00	7
7	0	0.00	0.00	7.00	7

Example: Histogram Equalization



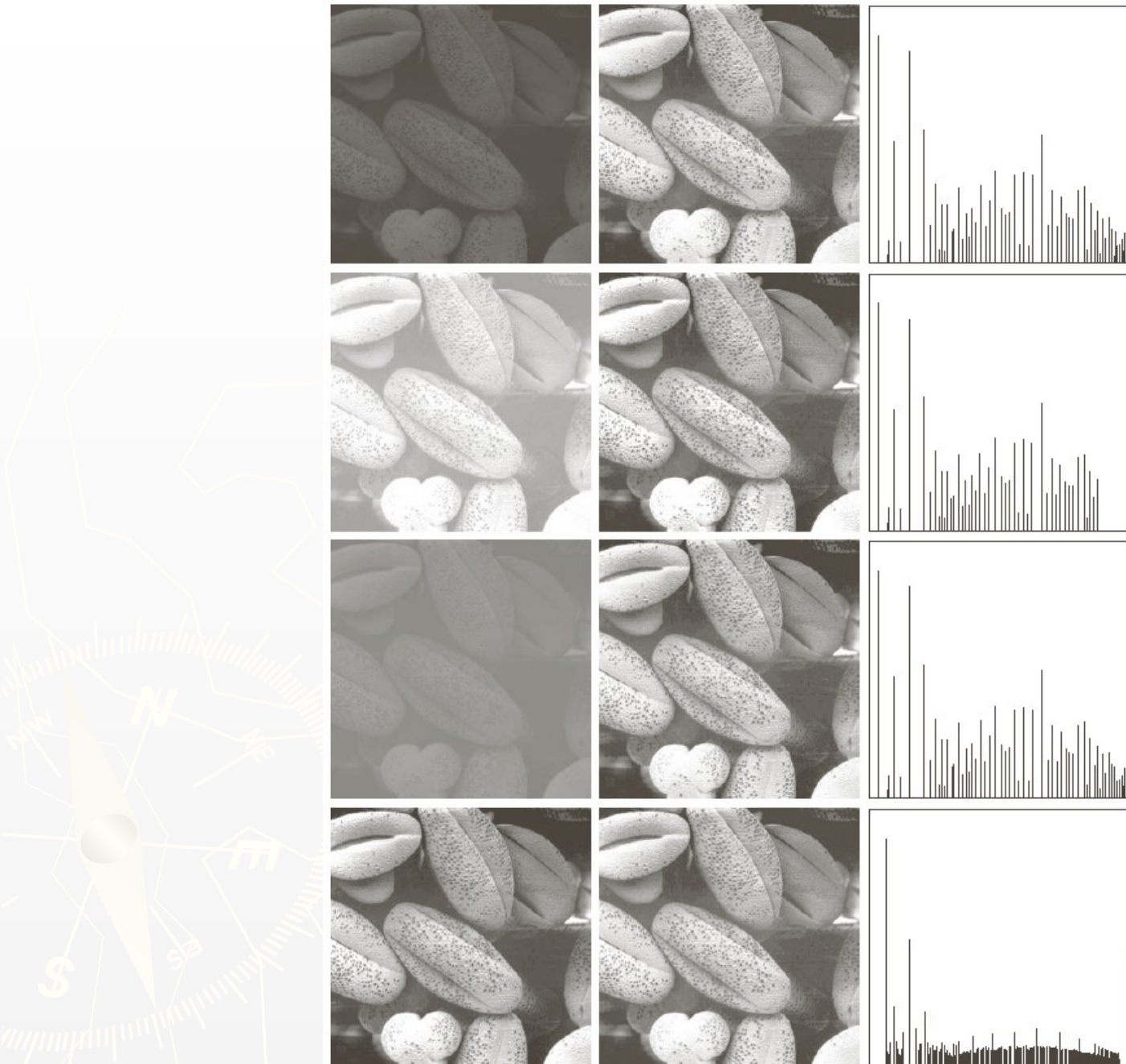


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Question

1. Write python code to perform histogram equalization.

Spatial Filtering

A spatial filter consists of (a) **a neighborhood**, and (b) **a predefined operation**

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression.

$w(x,y)$ is a filter, and $f(x,y)$ is the image

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

Spatial Correlation

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

Spatial Convolution

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x-s, y-t)$$

- Linear filtering of an image is accomplished through an operation called *convolution*.
- Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels.
- The matrix of weights is called the *convolution kernel*, also known as the *filter*.
- A convolution kernel is a correlation kernel that has been rotated 180 degrees.

↙ Origin	$f(x, y)$						$w(x, y)$		
	0	0	0	0	0				
	0	0	0	0	0				
	0	0	1	0	0		1	2	3
	0	0	0	0	0		4	5	6
	0	0	0	0	0		7	8	9

(a)

FIGURE 3.30
Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction

Blurring is used in removal of small details and bridging of small gaps in lines or curves

Smoothing spatial filters include linear filters and nonlinear filters.

Spatial Smoothing Linear Filters

The general implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is given

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

where $m = 2a + 1$, $n = 2b + 1$.

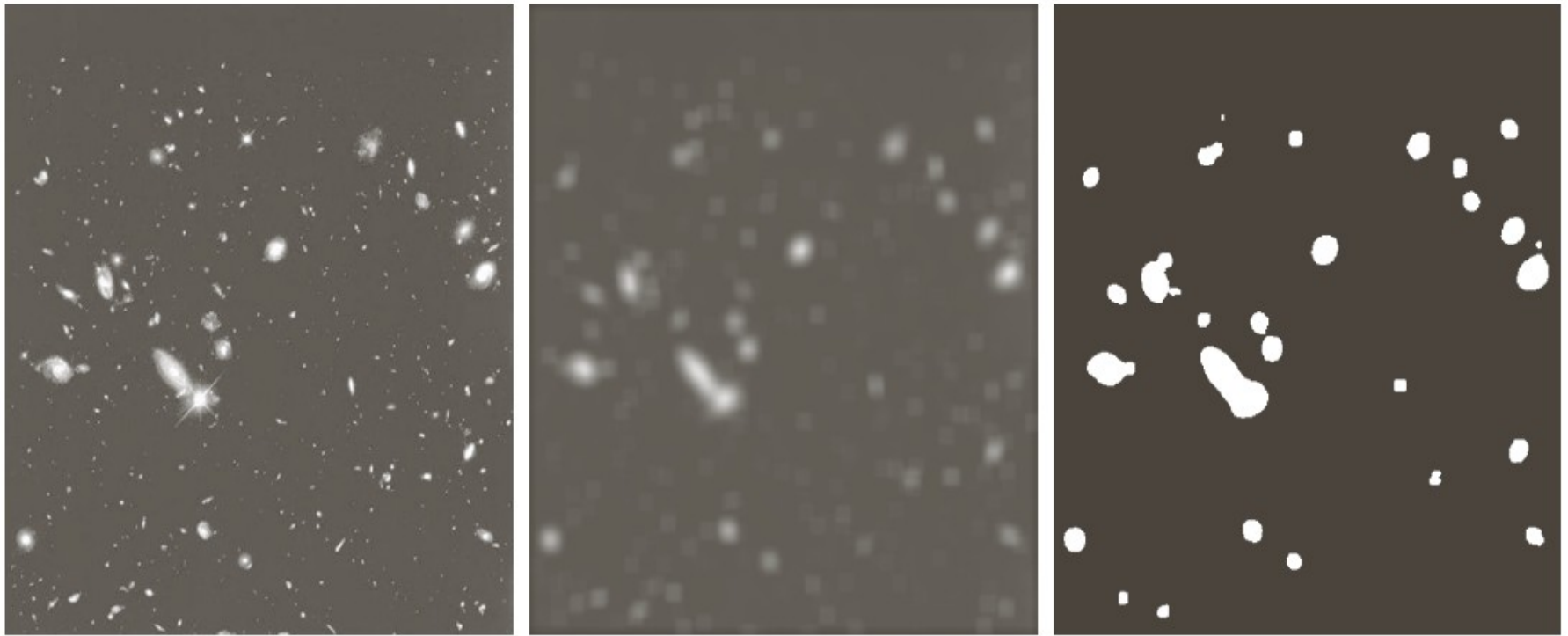
Two Smoothing Averaging Filter Masks

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

Example: Gross Representation of Objects



a b c

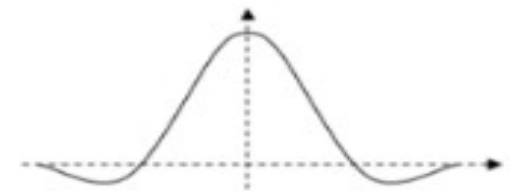
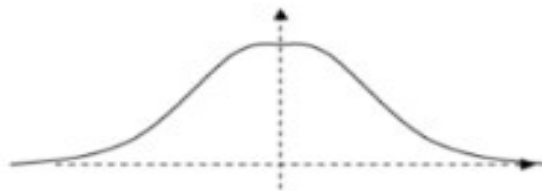
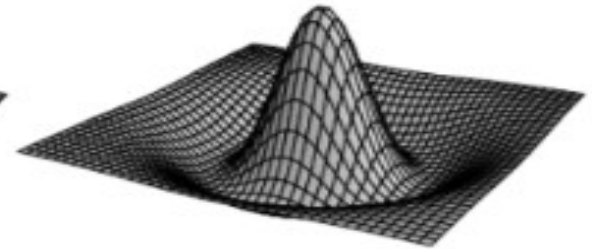
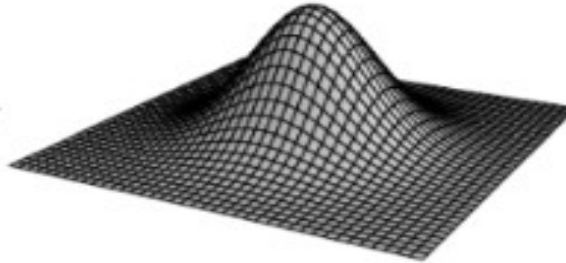
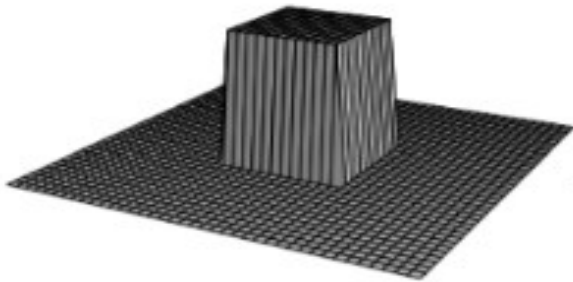
FIGURE 3.34 (a) Image of size 528×485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-statistic (Nonlinear) Filters

- Based on ordering (ranking) the pixels contained in the filter mask
- Replacing the value of the center pixel with the value determined by the ranking result

E.g., median filter, max filter, min filter

Example: filters



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

3D structure, 2D structure and example of filter (a) Box filter (b) Gaussian filter and (c) Laplace filter

Sharpening Spatial Filters

- ▶ Foundation
- ▶ Laplacian Operator
- ▶ Highboost Filtering
- ▶ Using First-Order Derivatives for Nonlinear Image Sharpening — The Gradient

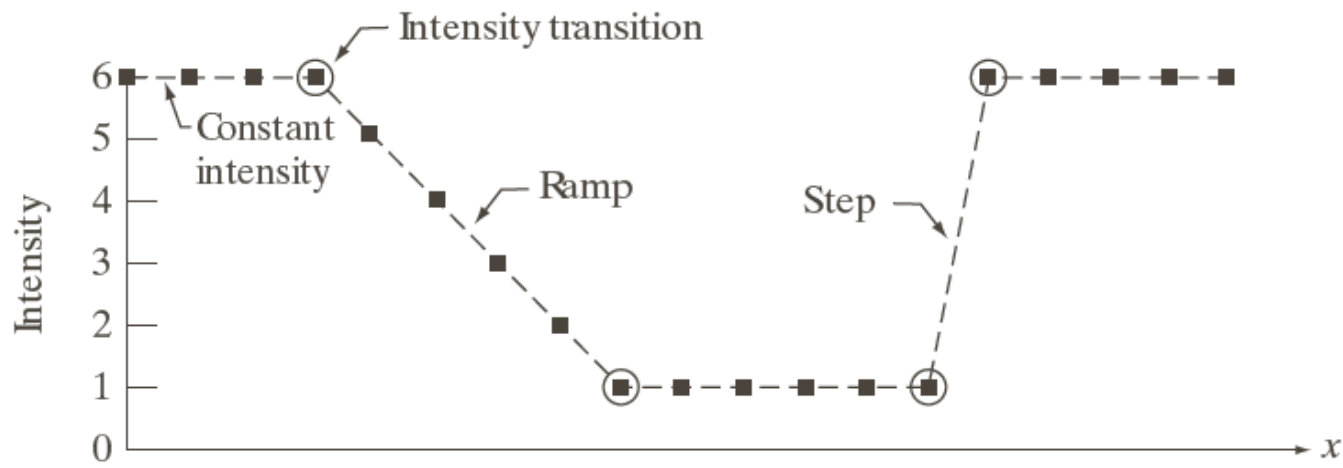
Sharpening Spatial Filters: Foundation

- ▶ The first-order derivative of a one-dimensional function $f(x)$ is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- ▶ The second-order derivative of $f(x)$ as the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Sharpening Spatial Filters: Laplace Operator

The second-order isotropic derivative operator is the Laplacian for a function (image) $f(x,y)$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Sharpening Spatial Filters: Laplace Operator

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a b
c d

FIGURE 3.37

(a) Filter mask used to implement Eq. (3.6-6).

(b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Sharpening Spatial Filters: Laplace Operator

Image sharpening in the way of using the Laplacian:

$$g(x, y) = f(x, y) + c \left[\nabla^2 f(x, y) \right]$$

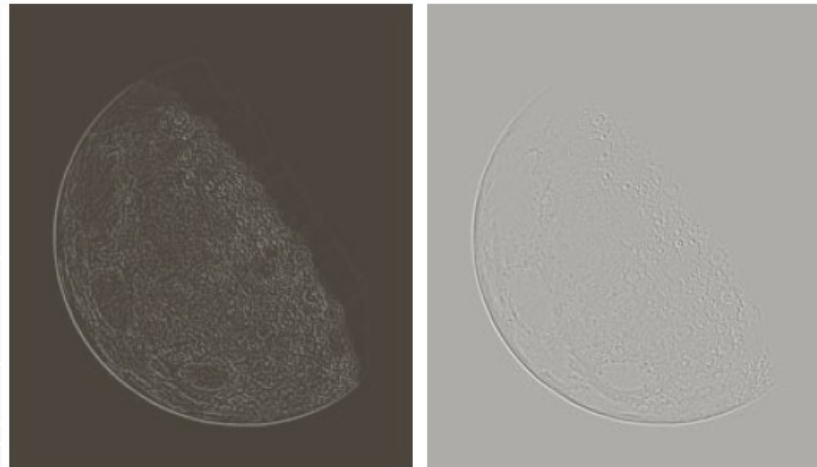
where,

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

$g(x, y)$ is sharpened images,

$c = -1$ if $\nabla^2 f(x, y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.



a	
b	c
d	e

FIGURE 3.38

(a) Blurred image of the North Pole of the moon.
 (b) Laplacian without scaling.
 (c) Laplacian with scaling.
 (d) Image sharpened using the mask in Fig. 3.37(a).
 (e) Result of using the mask in Fig. 3.37(b).
 (Original image courtesy of NASA.)

Unsharp Masking and Highboost Filtering

► Unsharp masking

Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

e.g., printing and publishing industry

► Steps

1. Blur the original image
2. Subtract the blurred image from the original
3. Add the mask to the original

Unsharp Masking and Highboost Filtering

Let $\bar{f}(x, y)$ denote the blurred image, unsharp masking is

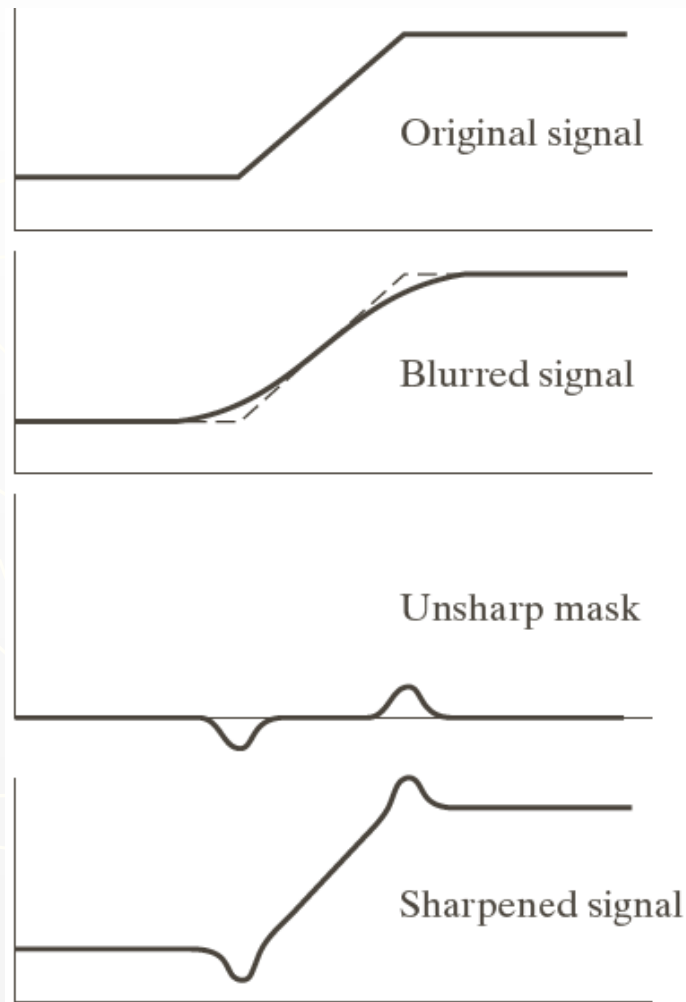
$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \quad k \geq 0$$

when $k > 1$, the process is referred to as highboost filtering.

Unsharp Masking: Demo



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Unsharp Masking and Highboost Filtering: Example



a
b
c
d
e

FIGURE 3.40

(a) Original image.
(b) Result of blurring with a Gaussian filter.
(c) Unsharp mask. (d) Result of using unsharp masking.
(e) Result of using highboost filtering.

Image Sharpening based on First-Order Derivatives

For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector ∇f , denoted as $M(x, y)$

Gradient Image $M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f , denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$M(x, y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$M(5)=$

$$M(x, y) = |z_8 - z_5| + |z_6 - z_5|$$

Image Sharpening based on First-Order Derivatives

Roberts Cross-gradient Operators

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

Sobel Operators

$$M(x, y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image Sharpening based on First-Order Derivatives

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0	0	-1
0	1	1	0

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

a
b c
d e

FIGURE 3.41

A 3×3 region of an image (the z s are intensity values).

(b)–(c) Roberts cross gradient operators.

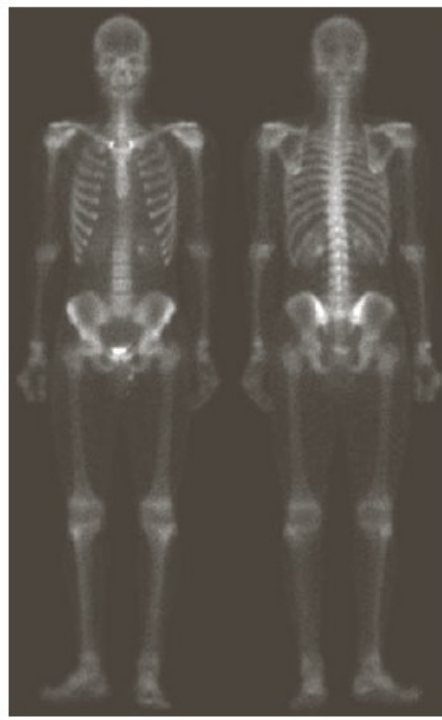
(d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Example:

Combining Spatial Enhancement Methods

Goal:

Enhance the
image by
sharpening it
and by bringing
out more of the
skeletal detail



a	b
c	d

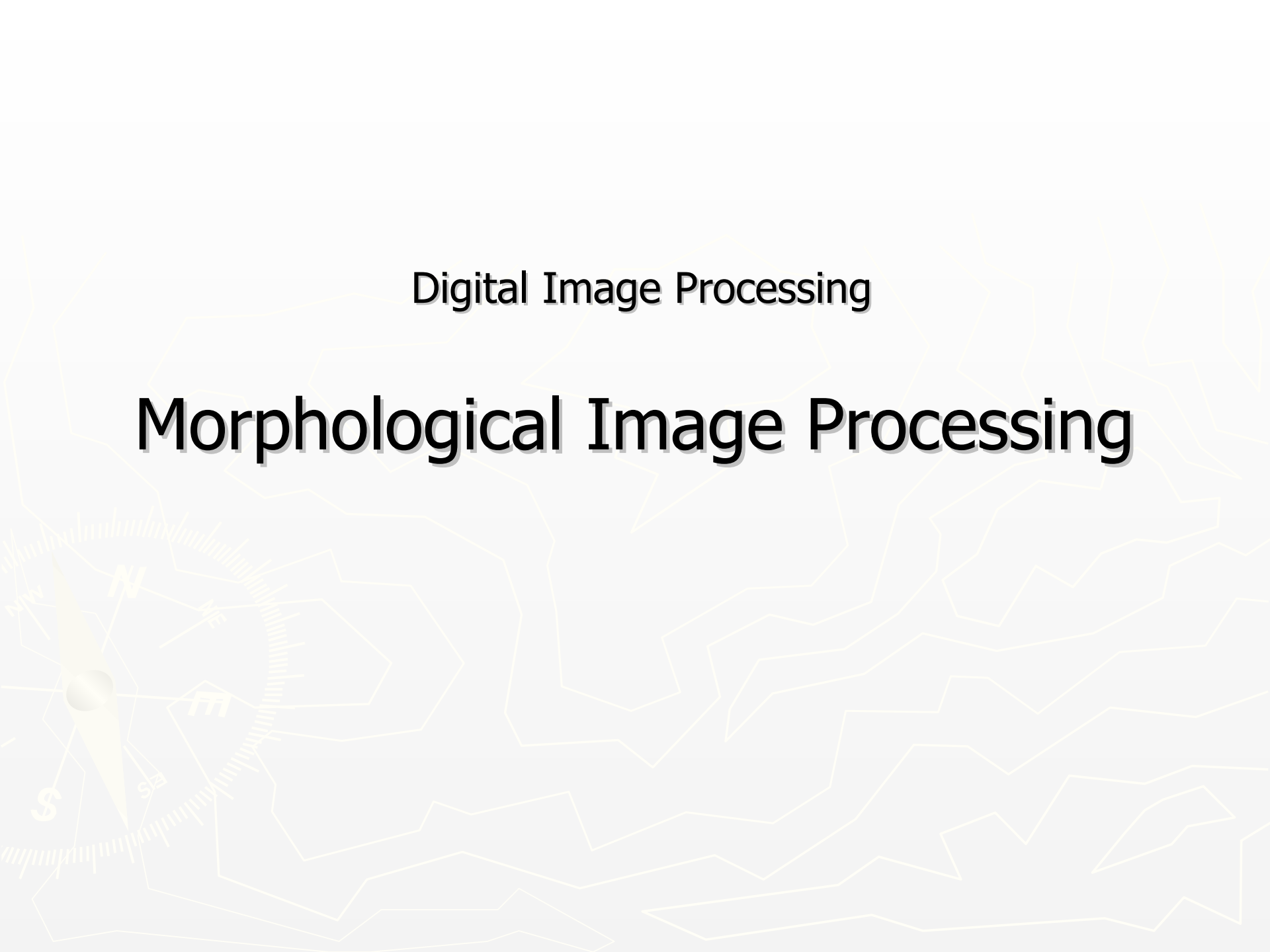
FIGURE 3.43

(a) Image of whole body bone scan.

(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).

What is mirroring and padding?



The background of the slide features a light gray map with yellow topographic contour lines. In the bottom-left corner, there is a yellow compass rose with a black needle pointing towards the top-left. The compass rose includes labels for cardinal and ordinal directions: 'N' for North, 'NE' for Northeast, 'SE' for Southeast, and 'SW' for Southwest. A yellow arrow points from the center of the compass rose towards the top-left.

Digital Image Processing

Morphological Image Processing

Introduction

- ▶ **Morphology:** a branch of biology that deals with the form and structure of animals and plants
- ▶ *Mathematical morphology* is a tool for extracting image components useful in the representation and description of region shape, such as boundaries, skeletons and convex hulls. The language of mathematical morphology is set theory, and as such it can apply directly to binary (two-level) images

Preliminaries (1)

Language of mathematical Morphological is Set Theory.

► Reflection

The reflection of a set B , denoted $\boxed{B}$, is defined as

$$\boxed{B} = \{w \mid w = -b, \text{ for } b \in B\}$$

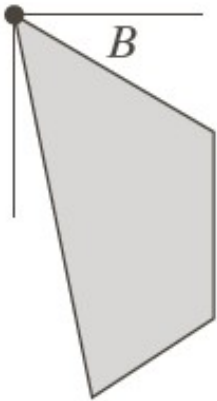
B is the set of pixels (2-D points) representing an object in image. And b is co-ordinate of the pixel

► Translation

The translation of a set B by point $z = (z_1, z_2)$, denoted $(B)_z$, is defined as

$$(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$$

Example: Reflection and Translation



a b c

FIGURE 9.1

(a) A set, (b) its reflection, and (c) its translation by z .

Examples: Structuring Elements (1)

Structure elements (SE): Small sets or sub-images used to probe an image under study for properties of interest

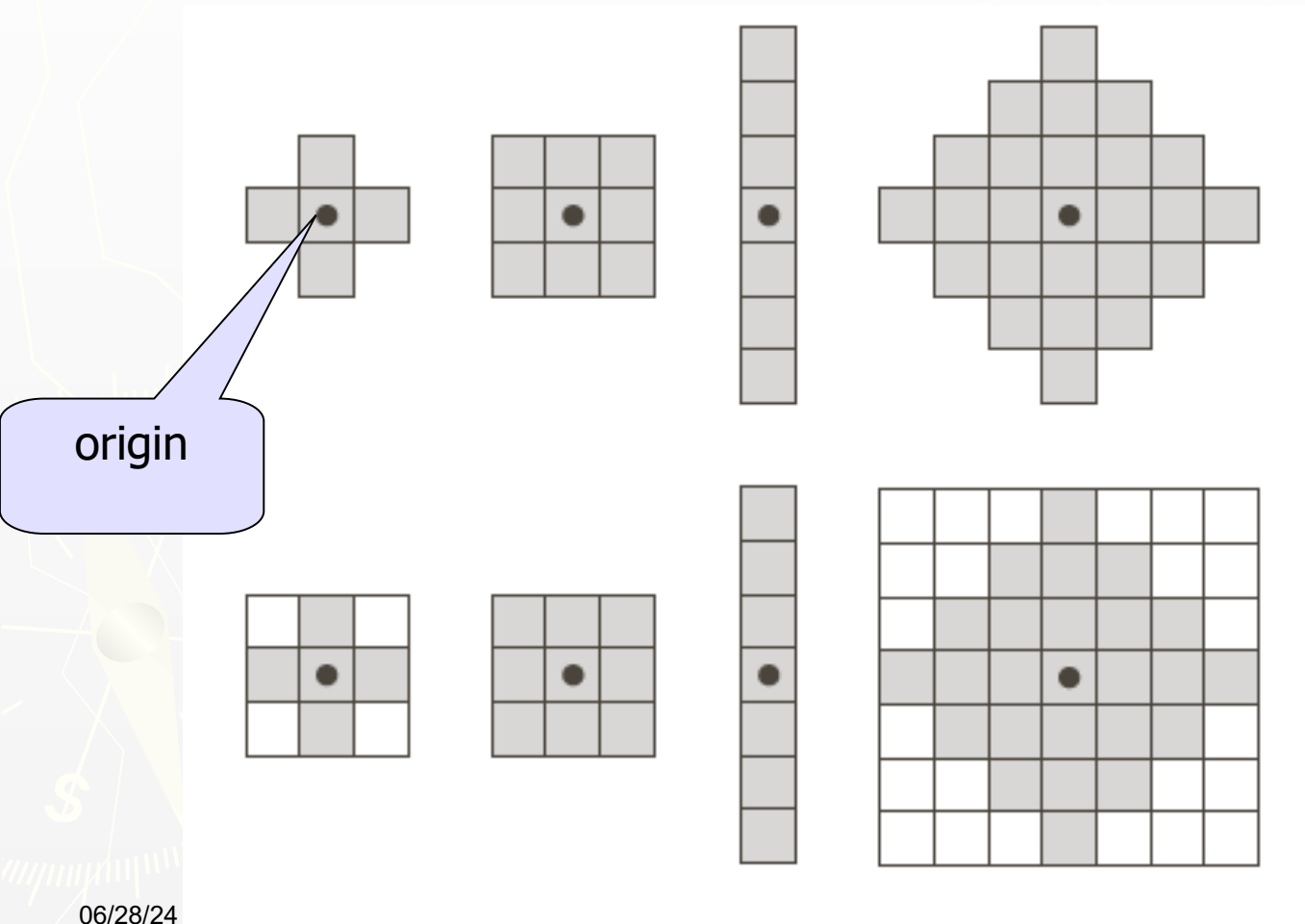
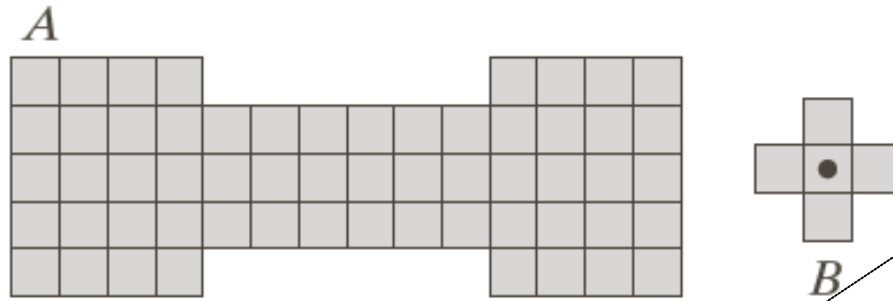


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Examples: Structuring Elements (2)

Accommodate the entire structuring element when its origin is on the border of the original set A



Origin of B visits every element of A

At each location of the origin of B, if B is completely contained in A, then the location is a member of the new set, otherwise it is not a member of the new set.

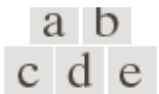


FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

Erosion: It shrinks or thins object in binary image

With A and B as sets in Z^2 , the erosion of A by B , denoted $A \ominus B$, defined

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

The set of all points z such that B , translated by z , is contained by A .

$$A \ominus B = \{z \mid (B)_z \cap A^c = \emptyset\}$$

Erosion The basic effect of erosion operator on a binary image is to erode away the boundaries of foreground pixels (usually the white pixels).

- ▶ Thus areas of foreground pixels shrink in size, and "holes" within those areas become larger.
- ▶ Mathematically, erosion of set A by set B is a set of all points x such that B translated by x is still contained in A .
- ▶ To compute the erosion of a binary input image by this structuring element, we consider each of the foreground pixels in the input image in turn. For each input pixel we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates.
- ▶ With the structuring element chosen as above, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other foreground pixels, assuming 8-connectedness. We can also see that this operation can be performed on binary images simply by applying a logical AND function.

Example of Erosion (1)

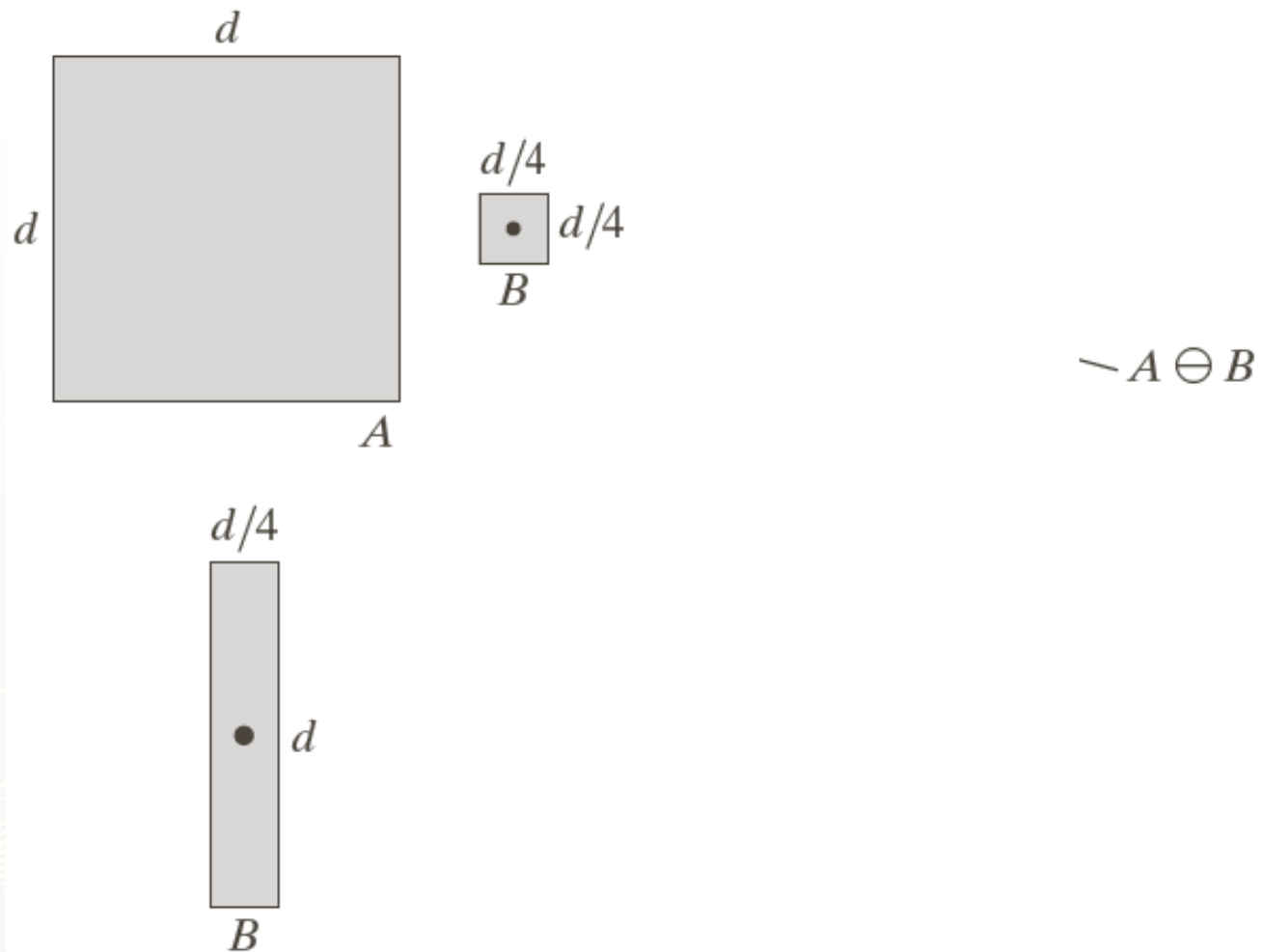
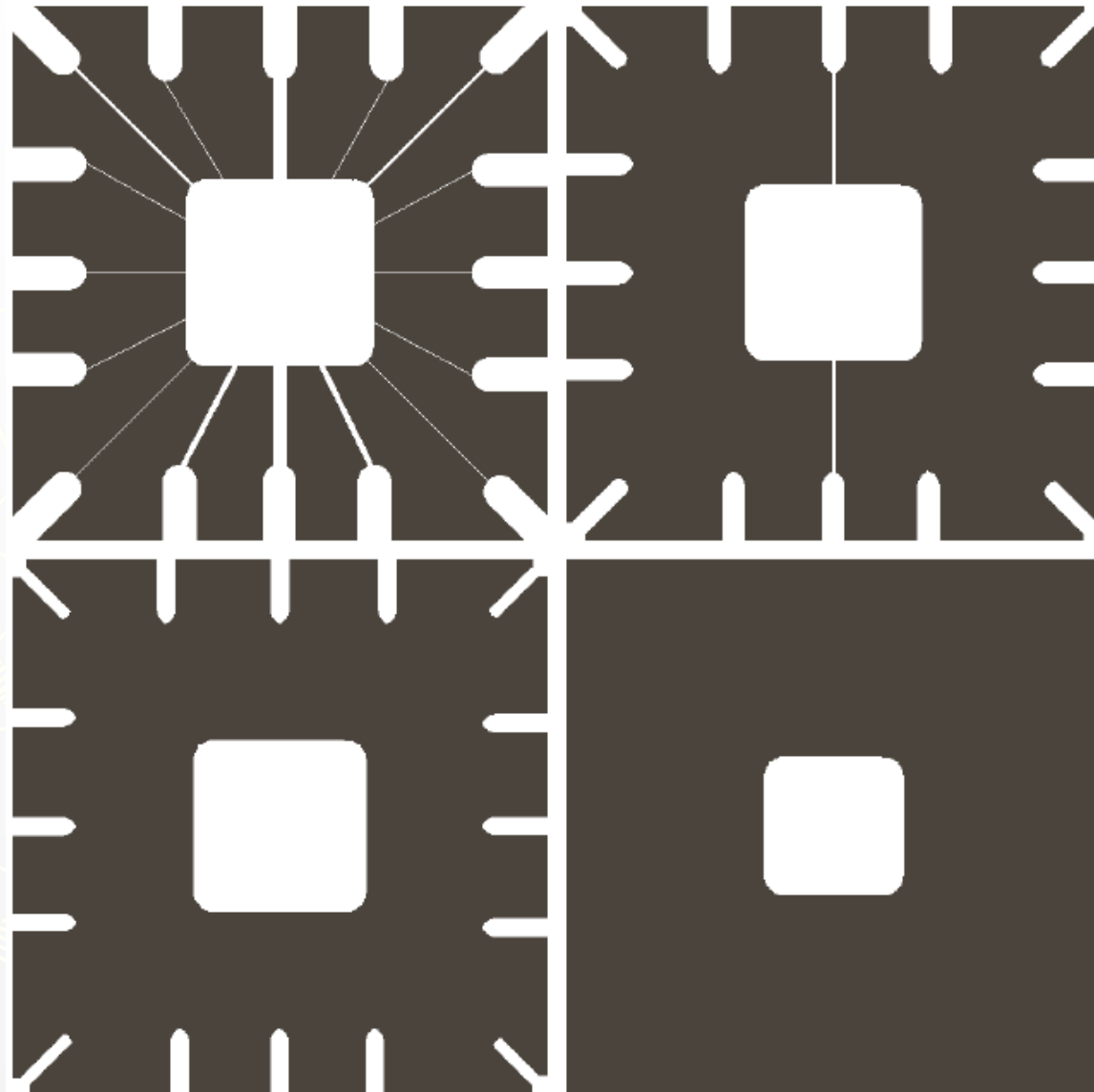


FIGURE 9.4 (a) Set A . (b) Square structuring element, B . (c) Erosion of A by B , shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A , shown only for reference.

Example of Erosion (2)



a	b
c	d

FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11 , 15×15 , and 45×45 , respectively. The elements of the SEs were all 1s.

- ▶ The **dilation** is the other of the two basic operators in mathematical morphology, the first one being [erosion](#). The basic effect of dilation on binary images is to enlarge the areas of foreground pixels (*i.e.* white pixels) at their borders. The areas of foreground pixels thus grow in size, while the background "holes" within them shrink.
- ▶ Mathematically, erosion of A by B is the set of all x displacement such that B and A overlap by at least one non-zero element.
- ▶ Again let us take a 3x3 matrix for the structuring element, with the center pixel used as the origin of the set. B , then the dilation can be performed using the logical OR function:

Dilation : It grows or thickens the object in binary image

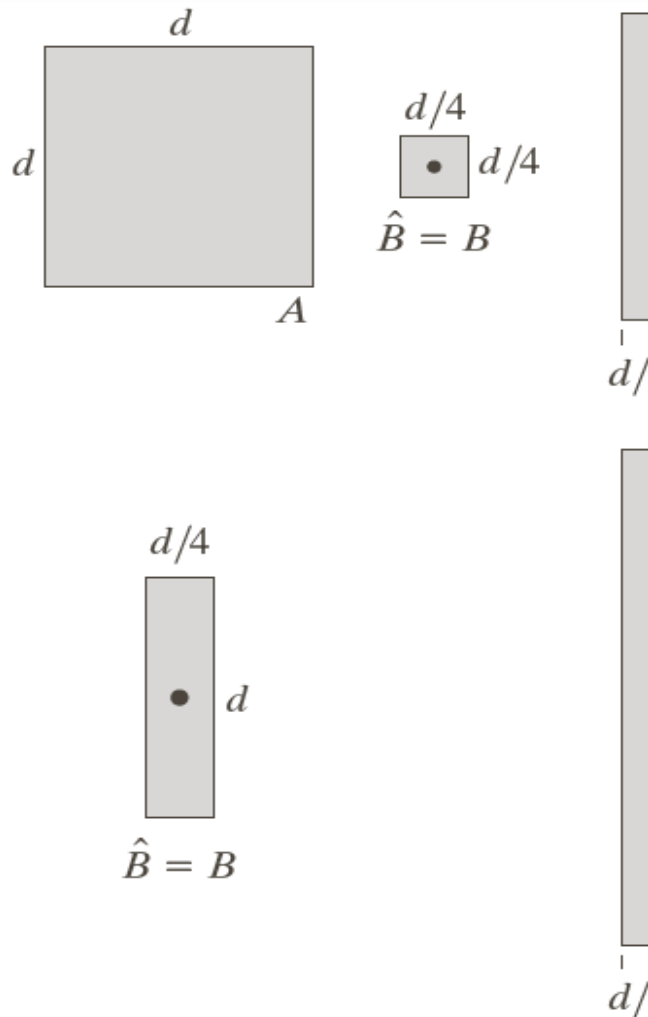
With A and B as sets in Z^2 , the dilation of A by B , denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \mid \left(\overset{\square}{B} \right)_z \cap A \neq \emptyset \right\}$$

The set of all displacements z , the translated $\overset{\square}{B}$ and A overlap by at least one element.

$$A \oplus B = \left\{ z \mid \left[\left(\overset{\square}{B} \right)_z \cap A \right] \subseteq A \right\}$$

Examples of Dilation (1)



a	b	c
d		e

FIGURE 9.6

(a) Set A .
 (b) Square structuring element (the dot denotes the origin).
 (c) Dilation of A by B , shown shaded.
 (d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A , shown only for reference

Examples of Dilation (2)

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



0	1	0
1	1	1
0	1	0

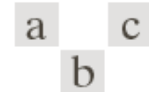


FIGURE 9.7

(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Duality

- ▶ Erosion and dilation are duals of each other with respect to set complementation and reflection

$$(A \ominus B)^c = A^c \oplus B$$

and

$$(A \oplus B)^c = A^c \ominus B$$

Opening and Closing

- ▶ Opening generally smoothes the contour of an object,
- ▶ Smooths convex part of the object
- ▶ Closing tends to smooth sections of contours but it fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour,
- ▶ Smooths concave part of the object

Opening and Closing

The opening of set A by structuring element B , denoted $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \Rightarrow A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A \}$$

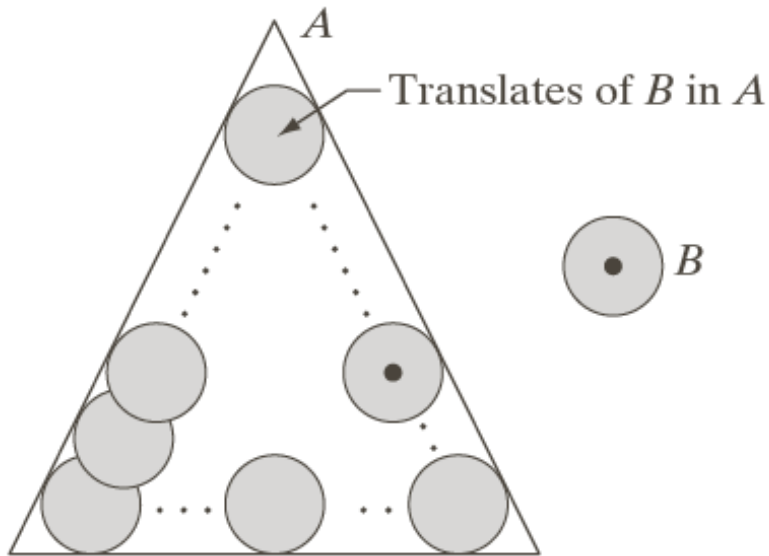
Union of all translates of B that fits into A (B as ball rolls inside the boundary of A). **Smooths convex** part of the object

The closing of set A by structuring element B , denoted $A \sqcap B$, is defined as

$$A \cdot B = (A \oplus B) \ominus B$$

Intersection of all translates of B such that B and A overlap by at least one non-zero element. (B as ball rolls outside the boundary of A .. **Smooths concave** part of the object)

Example: Opening

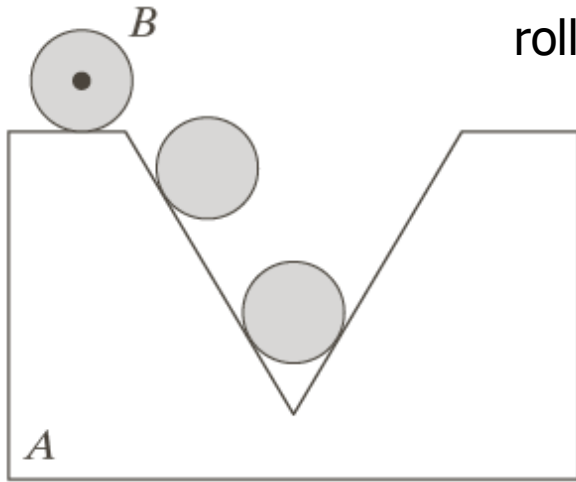


a b c d

FIGURE 9.8 (a) Structuring element B “rolling” along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

Example: Closing

Intersection of all translates of B that fits into A (B as ball rolls outside the boundary of A ..



a b c

FIGURE 9.9 (a) Structuring element B “rolling” on the outer boundary of set A . (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

The Properties of Opening and Closing

► Properties of Opening

(a) $A \circ B$ is a subset (subimage) of A

► Properties of Closing

M-



The Hit-or-Miss Transformation

if B denotes the set composed of D and its background, the match (or set of matches) of B in A , denoted $A \circledast B$,

$$A \circledast B = (A \ominus D) \cap [A^c \ominus (W - D)]$$

$$B = (B_1, B_2)$$

B_1 : object

B_2 : background

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

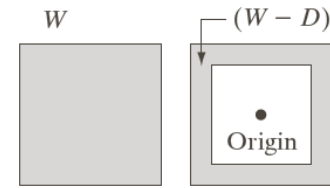
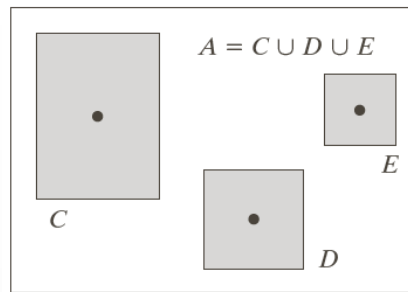


FIGURE 9.12

(a) Set A . (b) A window, W , and the local background of D with respect to W , $(W - D)$. (c) Complement of A . (d) Erosion of A by D . (e) Erosion of A^c by $(W - D)$. (f) Intersection of (d) and (e), showing the location of the origin of D , as desired. The dots indicate the origins of C , D , and E .

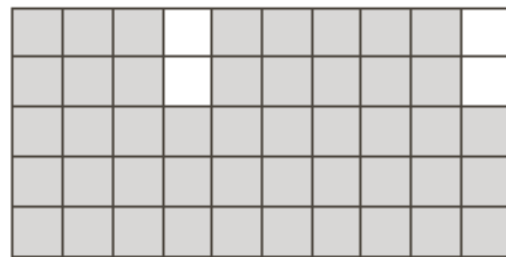
Some Basic Morphological Algorithms (1)

► **Boundary Extraction**

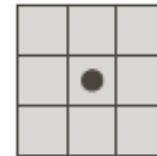
The boundary of a set A , can be obtained by first eroding A by B and then performing the set difference between A and its erosion.

$$\beta(A) = A - (A \ominus B)$$

Example 1



A



B

$A \ominus B$

$\beta(A)$

a	b
c	d

FIGURE 9.13 (a) Set A . (b) Structuring element B . (c) A eroded by B . (d) Boundary, given by the set difference between A and its erosion.

Example 2



a b

FIGURE 9.14

(a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

Some Basic Morphological Algorithms (2)

► **Hole Filling**

A hole may be defined as a background region surrounded by a connected border of foreground pixels.

Let A denote a set whose elements are 8-connected boundaries, each boundary enclosing a background region (i.e., a hole). Given a point in each hole, the objective is to fill all the holes with 1s.

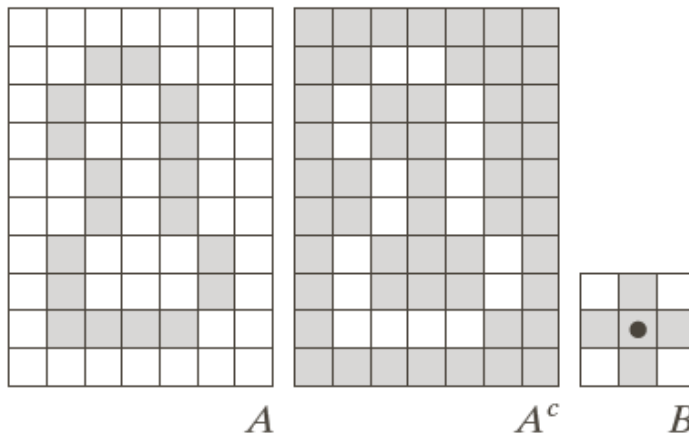
Some Basic Morphological Algorithms (2)

► Hole Filling

1. Forming an array X_0 of 0s (the same size as the array containing A), except the locations in X_0 corresponding to the given point in each hole, which we set to 1.

$$2. X_k = (X_{k-1} \oplus B) \cap A^c \quad k=1,2,3,\dots$$

Stop the iteration if $X_k = X_{k-1}$



a	b	c
d	e	f
g	h	i

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (b) Complement of A . (c) Structuring element B . (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

X_0

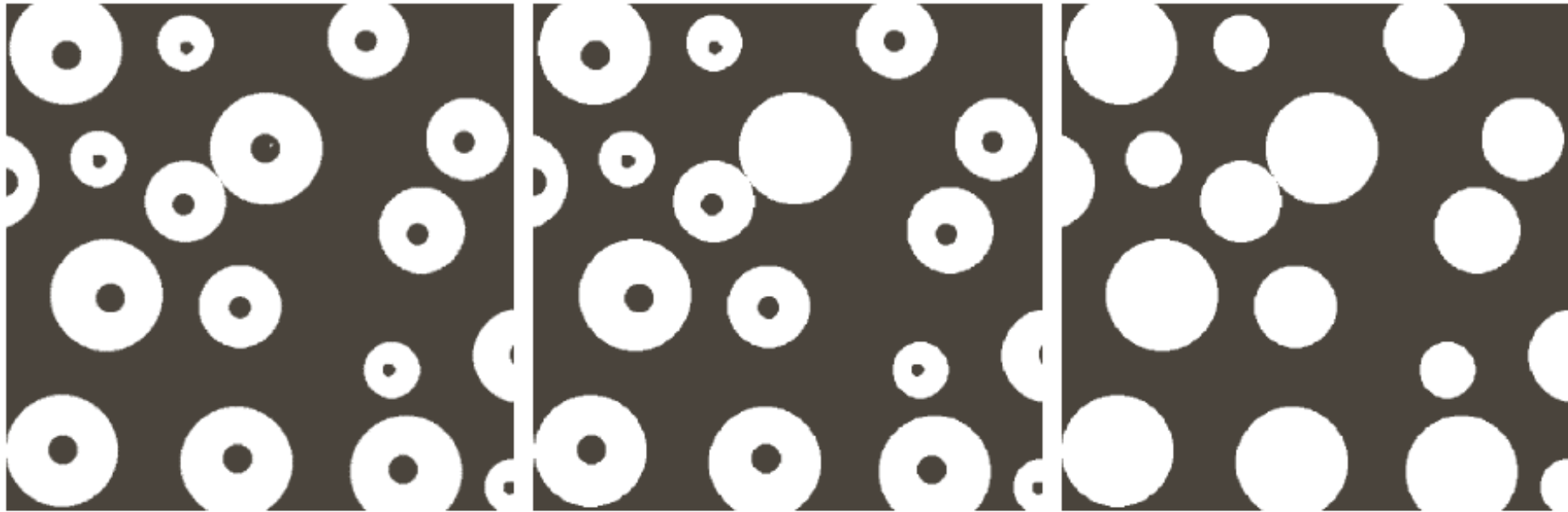
X_1

X_2

X_6

X_8

$X_8 \cup A$



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Some Basic Morphological Algorithms (3)

► **Extraction of Connected Components**

Central to many automated image analysis applications.

Let A be a set containing one or more connected components, and form an array X_0 (of the same size as the array containing A) whose elements are 0s, except at each location known to correspond to a point in each connected component in A , which is set to 1.

Some Basic Morphological Algorithms (3)

► **Extraction of Connected Components**

Central to many automated image analysis applications.

$$X_k = (X_{k-1} \oplus B) \cap A$$

B : structuring element

until $X_k = X_{k-1}$

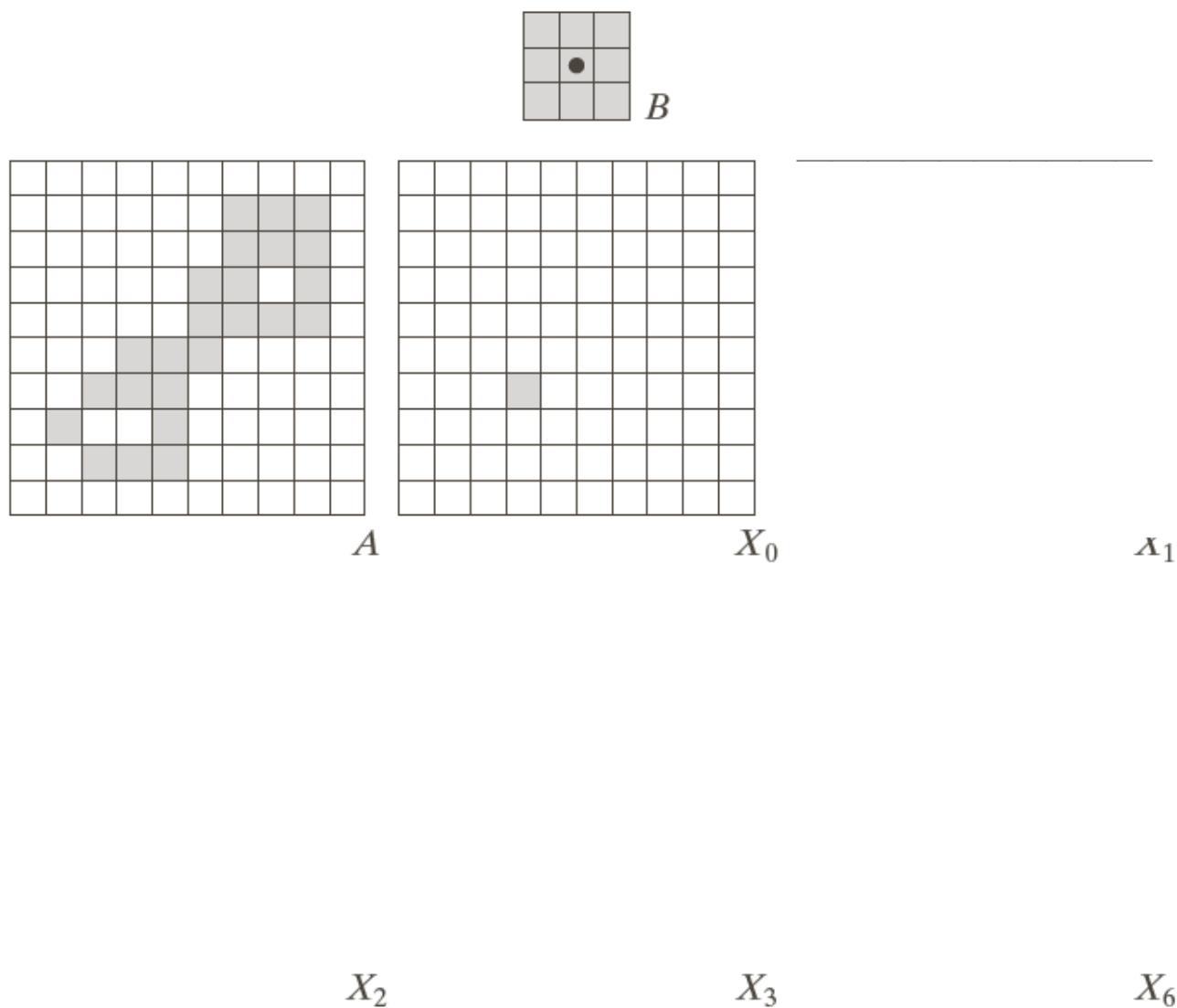


FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

► Thank You



Image Restoration and Reconstruction

K K Singh



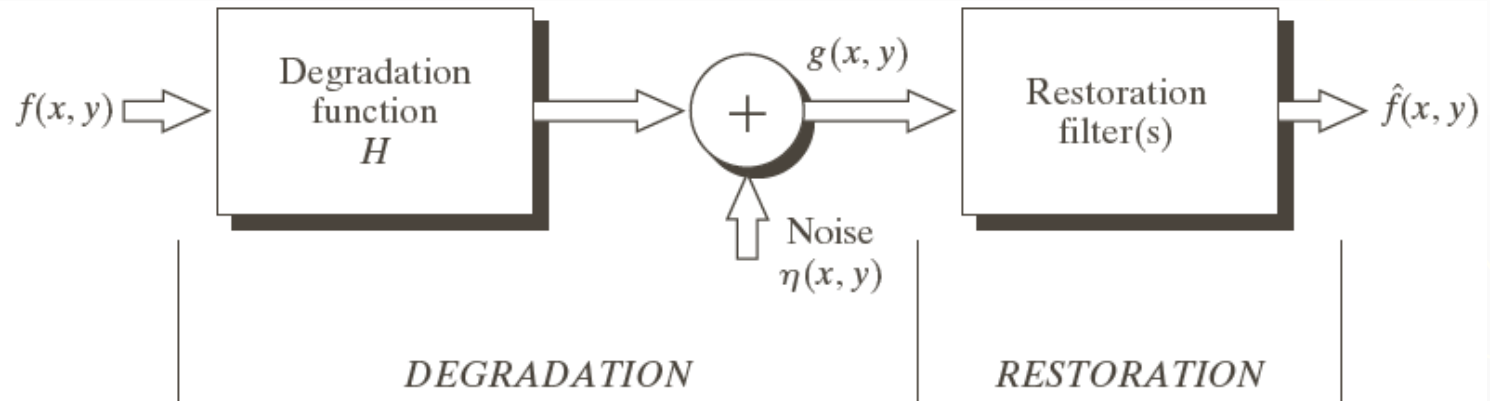
Image Restoration

- ▶ Image restoration: recover an image that has been degraded by using a prior knowledge of the degradation phenomenon.
- ▶ Model the degradation and applying the inverse process in order to recover the original image.

A Model of Image Degradation/Restoration Process

FIGURE 5.1

A model of the image degradation/restoration process.



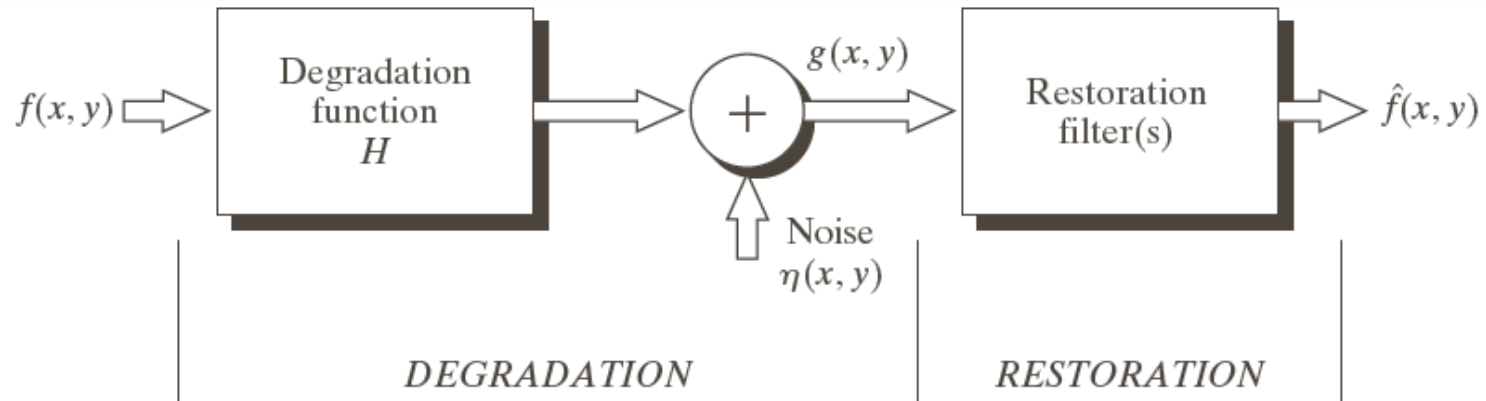
► Degradation

- Degradation function H
- Additive noise $\eta(x, y)$

A Model of Image Degradation/Restoration Process

FIGURE 5.1

A model of the image degradation/restoration process.



If H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y)$$

Noise Sources

- ▶ The principal sources of noise in digital images arise during **image acquisition and/or transmission**
- ✓ Image acquisition
e.g., light levels, sensor temperature, etc.
- ✓ Transmission
e.g., lightning or other atmospheric disturbance in wireless network

Noise Models

► White noise

- The Fourier spectrum of noise is constant
-
- ## ► With the exception of spatially periodic noise, we assume
- Noise is independent of spatial coordinates
 - Noise is uncorrelated with respect to the image itself

Noise Models

- **Gaussian noise**

Electronic circuit noise, sensor noise due to poor illumination and/or high temperature

- **Rayleigh noise**

Range imaging: a collection of techniques that are used to produce a 2D image showing the distance to points in a scene from a specific point

- **Erlang (gamma) noise:** Laser imaging

- **Exponential noise:** Laser imaging

- **Uniform noise:** Least descriptive; Basis for numerous random number generators

- **Impulse noise:** quick transients, such as faulty switching

Gaussian Noise

The PDF of Gaussian random variable, z , is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

- 70% of its values will be in the range

$$[(\mu - \sigma), (\mu + \sigma)]$$

- 95% of its values will be in the range

$$[(\mu - 2\sigma), (\mu + 2\sigma)]$$

Rayleigh Noise

The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b / 4}$$
$$\sigma^2 = \frac{b(4 - \pi)}{4}$$

Erlang (Gamma) Noise

The PDF of Erlang noise is given by

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = b / a$$

$$\sigma^2 = b / a^2$$

Exponential Noise

The PDF of exponential noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = 1 / a$$

$$\sigma^2 = 1 / a^2$$

Uniform Noise

The PDF of uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this density are given by

$$\bar{z} = (a + b) / 2$$

$$\sigma^2 = (b - a)^2 / 12$$

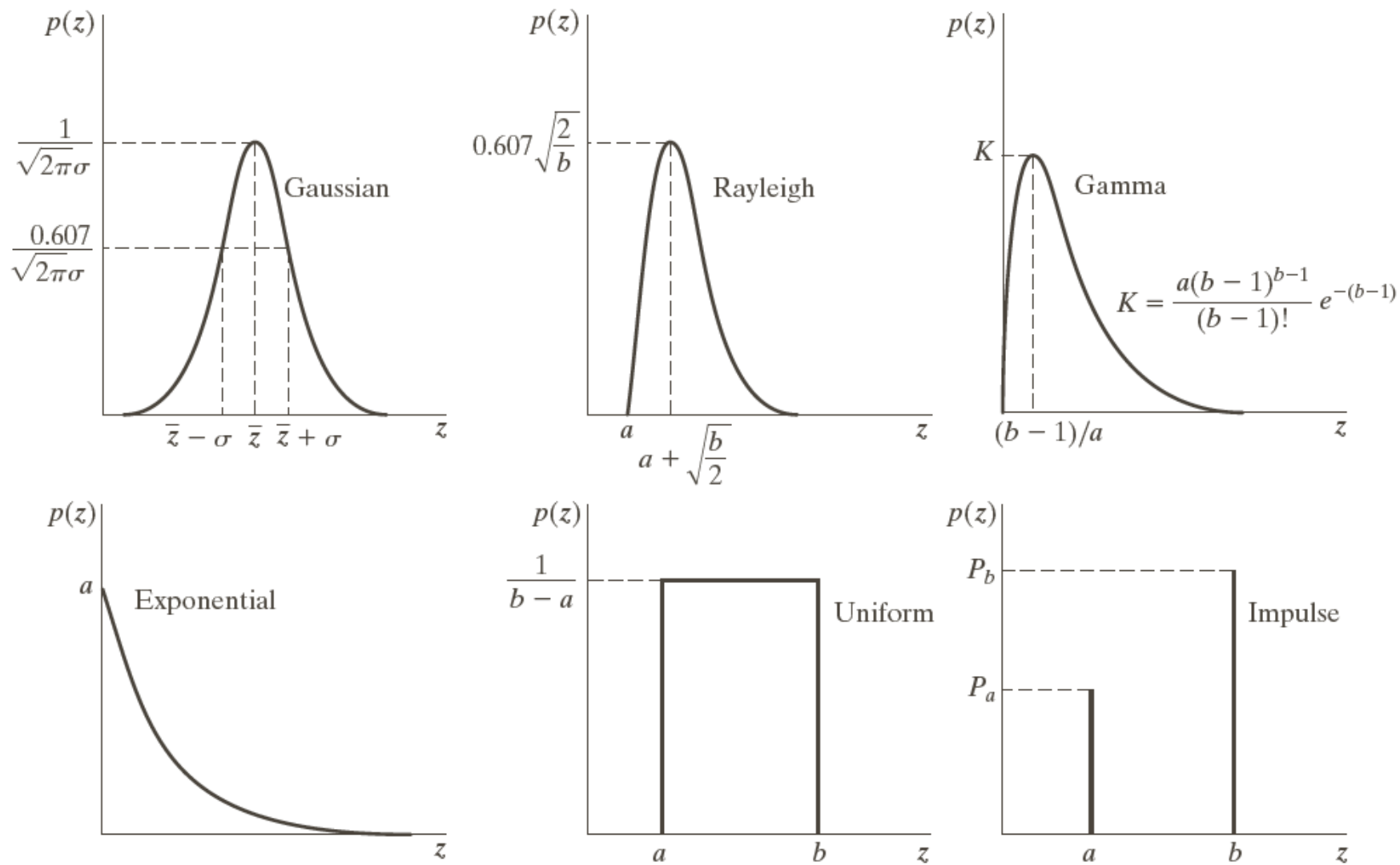
Impulse (Salt-and-Pepper) Noise

The PDF of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

if $b > a$, gray-level b will appear as a light dot, while level a will appear like a dark dot.

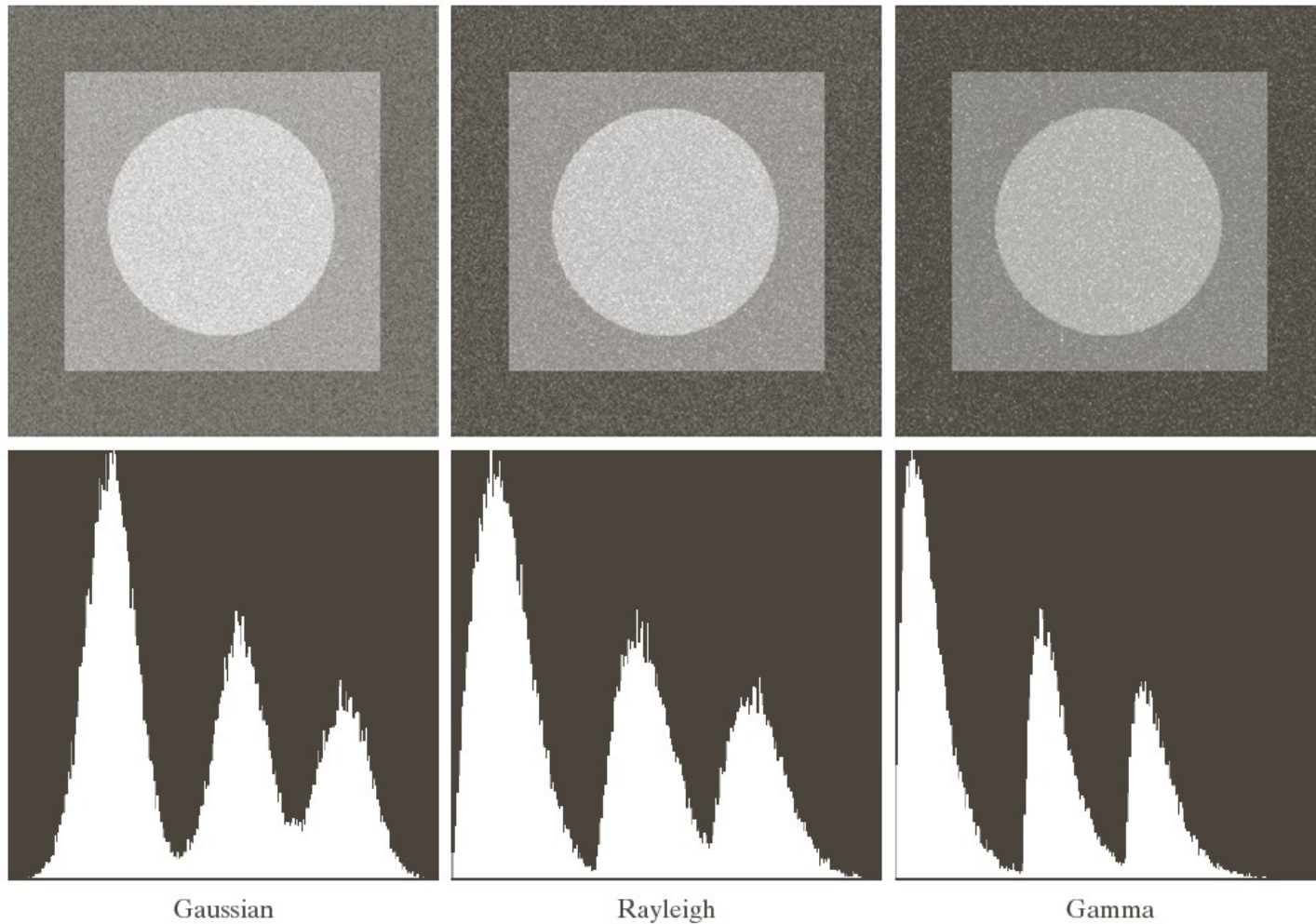
If either P_a or P_b is zero, the impulse noise is called *unipolar*



a	b	c
d	e	f

FIGURE 5.2 Some important probability density functions.

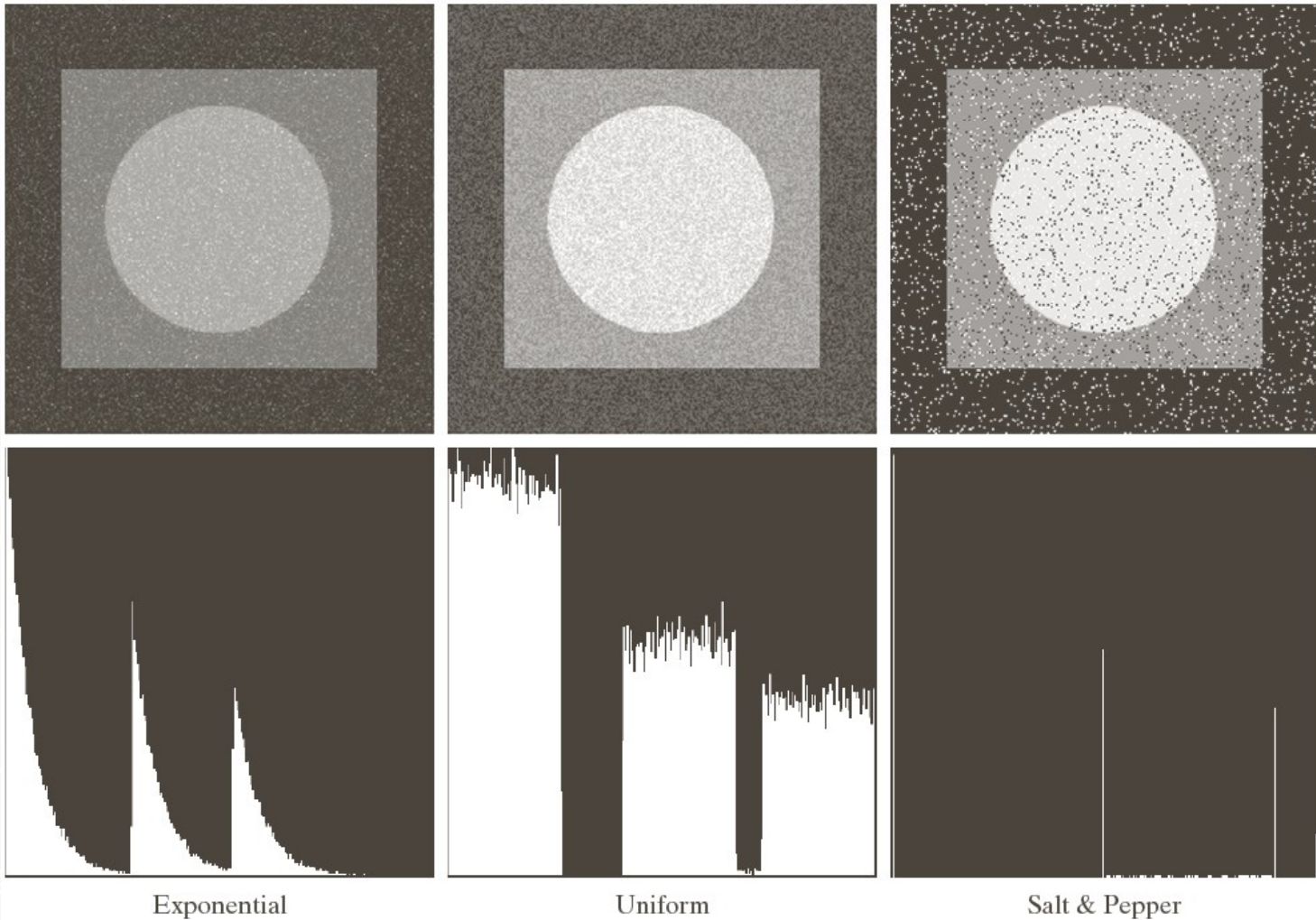
Examples of Noise: Noisy Images(1)



a b c
d e f

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Examples of Noise: Noisy Images(2)



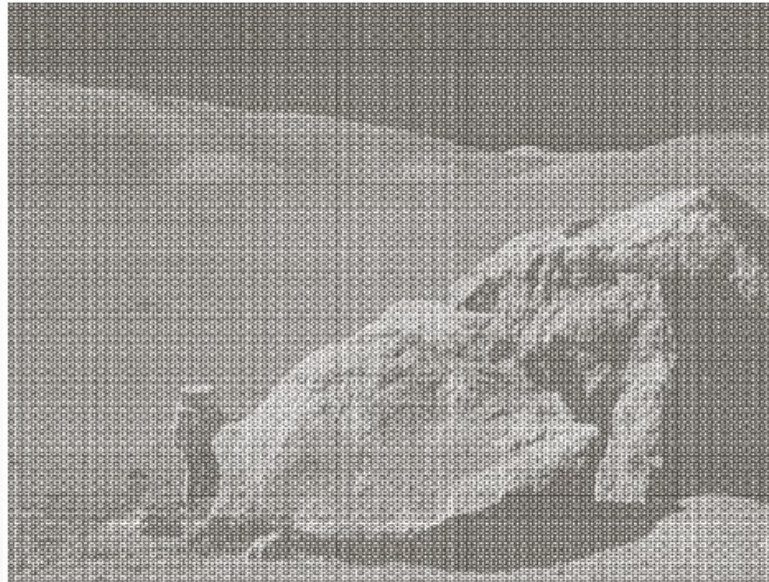
g	h	i
j	k	l

FIGURE 5.4 (Continued) Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Periodic Noise

- ▶ Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition.
- ▶ It is a type of spatially dependent noise
- ▶ Periodic noise can be reduced significantly via frequency domain filtering

An Example of Periodic Noise



a
b

FIGURE 5.5

(a) Image corrupted by sinusoidal noise.
(b) Spectrum (each pair of conjugate impulses corresponds to one sine wave).
(Original image courtesy of NASA.)

Estimation of Noise Parameters (1)

The shape of the histogram identifies the closest PDF match from the histogram plot of a small background strip of the image

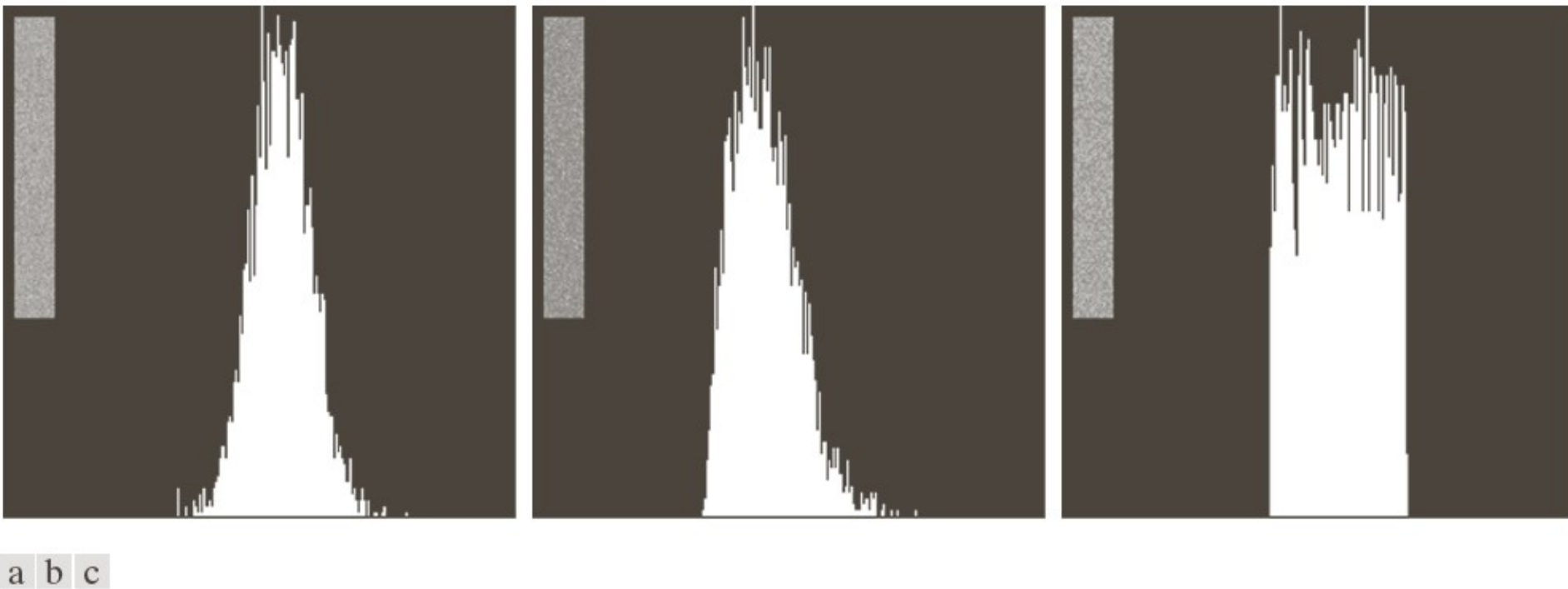


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Estimation of Noise Parameters (2)

Consider a subimage denoted by S , and let $p_s(z_i)$, $i = 0, 1, \dots, L-1$, denote the probability estimates of the intensities of the pixels in S . The mean and variance of the pixels in S :

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_s(z_i)$$

and

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p_s(z_i)$$

For Gaussian, we need mean and variance, and for other distributions we calculate a and b.

Restoration in the Presence of Noise Only and the noise is not periodic is to use

- Spatial Filtering

Noise model without degradation

$$g(x, y) = f(x, y) + \eta(x, y)$$

and

$$G(u, v) = F(u, v) + N(u, v)$$

Spatial Filtering: Mean Filters (1)

Let S_{xy} represent the set of coordinates in a rectangle subimage window of size $m \times n$, centered at (x, y) .

Arithmetic mean filter

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

Spatial Filtering: Mean Filters (2)

Geometric mean filter

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

Generally, a geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process

Spatial Filtering: Mean Filters (3)

Harmonic mean filter

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s, t)}}$$

It works well for salt noise, but fails for pepper noise.
It does well also with other types of noise like Gaussian noise.

Spatial Filtering: Mean Filters (4)

Contraharmonic mean filter

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q}$$

Q is the order of the filter.

It is well suited for reducing the effects of salt-and-pepper noise. $Q > 0$ for pepper noise and $Q < 0$ for salt noise.

06/28/24 If $Q=0$, it is arithmetic mean, If $Q=-1$, it is harmonic mean

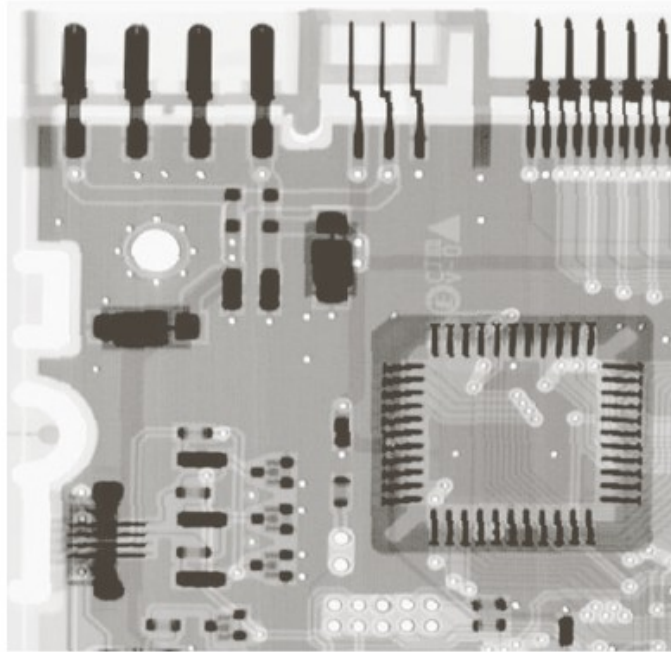
Spatial Filtering: Example (1)

a	b
c	d

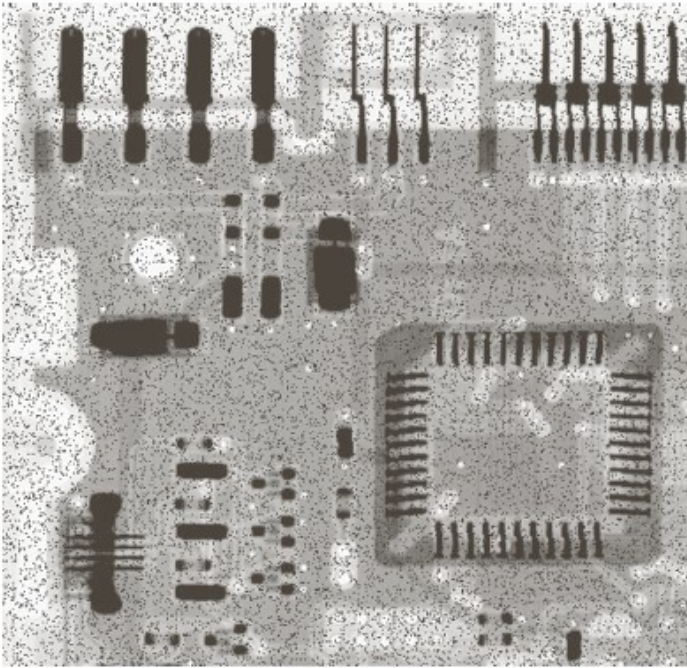
FIGURE 5.7

(a) X-ray image.
(b) Image corrupted by additive Gaussian noise.
(c) Result of filtering with an arithmetic mean filter of size 3×3 .
(d) Result of filtering with a geometric mean filter of the same size.

(Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



Spatial Filtering: Example (2)



a	b
c	d

FIGURE 5.8

(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

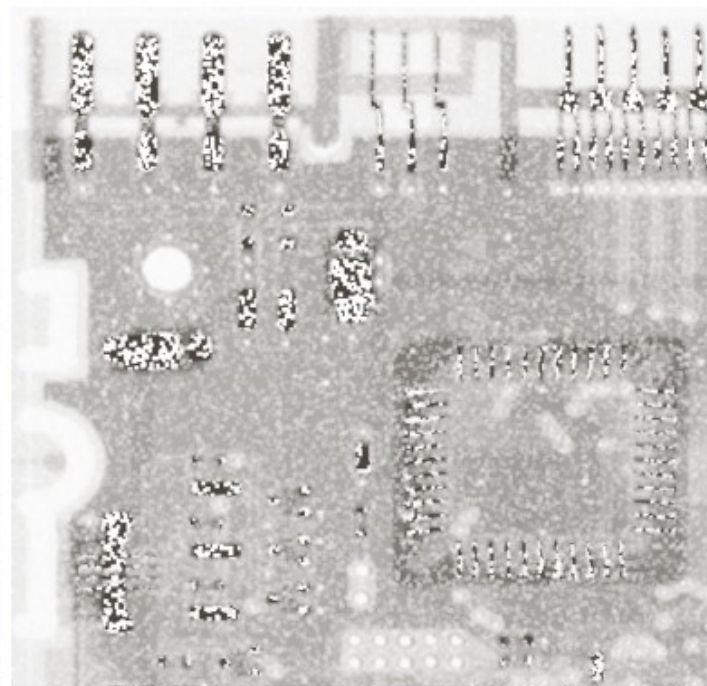
Spatial Filtering: Example (3)

a b

FIGURE 5.9

Results of selecting the wrong sign in contraharmonic filtering.

(a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$.
(b) Result of filtering 5.8(b) with $Q = 1.5$.



Spatial Filtering: Order-Statistic Filters (1)

Median filter

$$\boxed{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s, t)\}$$

Suitable for
Impluse noise

Max filter

$$\boxed{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Suitable for Pepper
noise

Min filter

$$\boxed{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Suitable for salt
noise

Midpoint filter

$$\boxed{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right]$$

Suitable for
Gaussian
noise

a	b
c	d

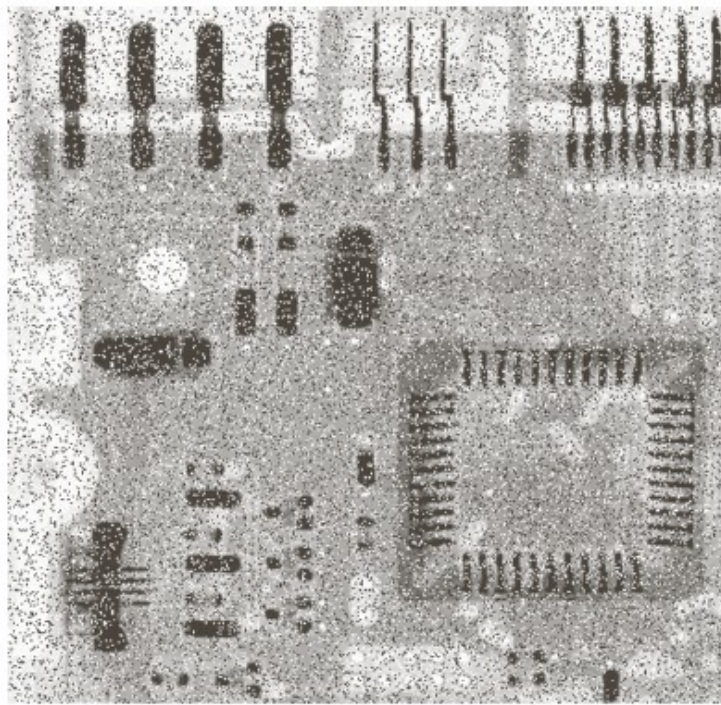
FIGURE 5.10

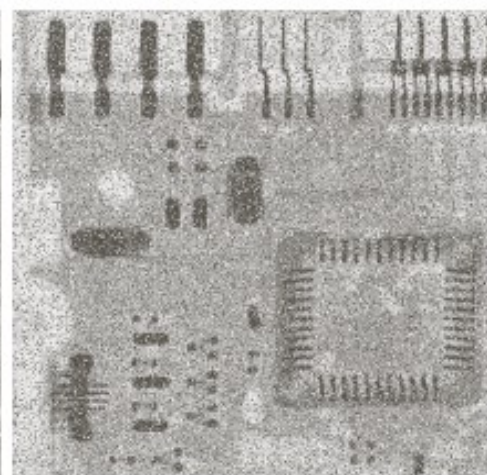
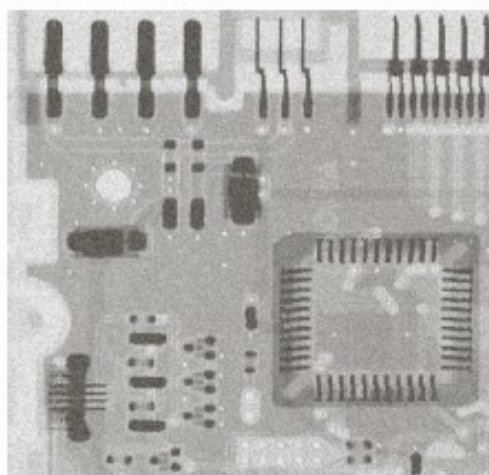
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.

(b) Result of one pass with a median filter of size 3×3 .

(c) Result of processing (b) with this filter.

(d) Result of processing (c) with the same filter.





a	b
c	d
e	f

FIGURE 5.12

(a) Image corrupted by additive uniform noise.

(b) Image additionally corrupted by additive salt-and-pepper noise.

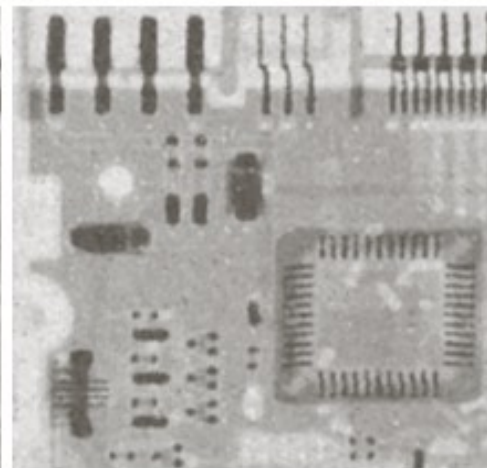
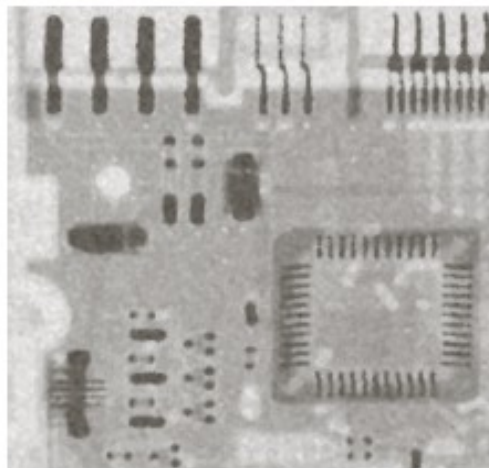
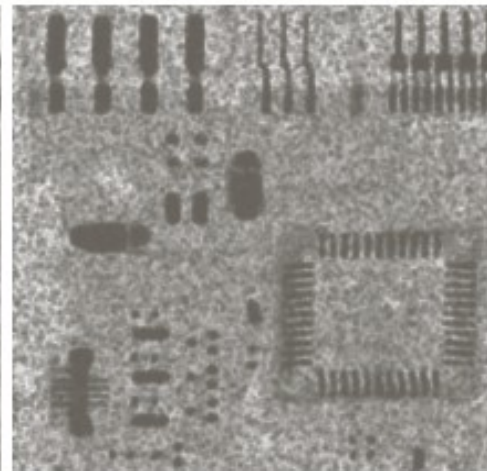
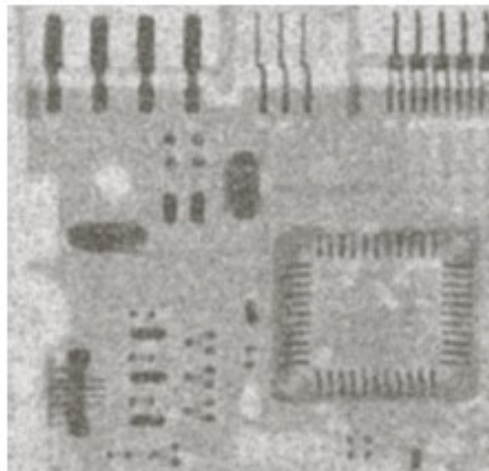
Image (b) filtered with a 5×5 ;

(c) arithmetic mean filter;

(d) geometric mean filter;

(e) median filter;

and (f) alpha-trimmed mean filter with $d = 5$.



Linear, Position-Invariant Degradations

H is linear

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$$

f_1 and f_2 are any two input images.

An operator having the input-output relationship

$g(x, y) = H[f(x, y)]$ is said to be position invariant

if

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta)$$

for any $f(x, y)$ and any α and β .

Linear, Position-Invariant Degradations

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta$$

Assume for a moment that $\eta(x, y) = 0$

if H is a linear operator,

tion (or
olm)
of the
ind

se
se

Mathematical Modeling (1)

- ▶ Environmental conditions cause degradation

A model about atmospheric turbulence

$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$

k : a constant that depends on
the nature of the turbulence

a	b
c	d

FIGURE 5.25

Illustration of the
atmospheric
turbulence model.

(a) Negligible
turbulence.

(b) Severe
turbulence,
 $k = 0.0025$.

(c) Mild
turbulence,
 $k = 0.001$.

(d) Low
turbulence,
 $k = 0.00025$.

(Original image
courtesy of
NASA.)



Mathematical Modeling (2)

- Derive a mathematical model from basic principles

E.g., An image blurred by uniform linear motion between the image and the sensor during image acquisition

Mathematical Modeling (3)

Suppose that an image $f(x, y)$ undergoes planar motion, $x_0(t)$ and $y_0(t)$ are the time-varying components of motion in the x - and y -directions, respectively.

The optical imaging process is perfect. T is the duration of the exposure. The blurred image $g(x, y)$

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt$$

Mathematical Modeling (4)

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt$$

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux+vy)} dx dy$$

Mathematical Modeling (4)

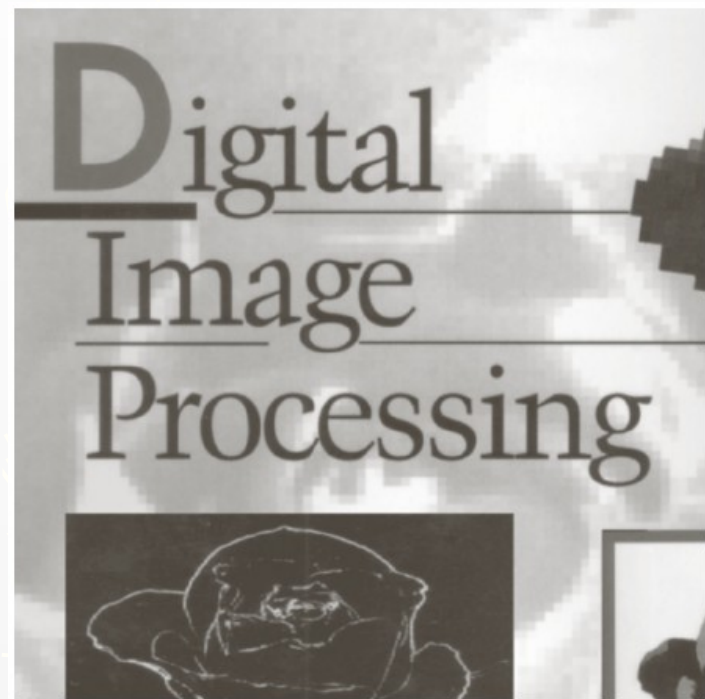
$$H(u, v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt$$

Mathematical Modeling (5)

Suppose that the image undergoes uniform linear motion in the x -direction and y -direction, at a rate given by

$$x_0(t) = at / T \text{ and } y_0(t) = bt / T$$

$$\begin{aligned} H(u, v) &= \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \\ &= \int_0^T e^{-j2\pi[ua+vb]t/T} dt \end{aligned}$$



a b

FIGURE 5.26

(a) Original image.
(b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Minimum Mean Square Error (Wiener) Filtering

➤ **N. Wiener (1942)**

➤ **Objective**

Find an estimate of the uncorrupted image such that the mean square error between them is minimized

$$e^2 = E \left\{ (f - \hat{f})^2 \right\}$$

Minimum Mean Square Error (Wiener) Filtering

The minimum of the error function is given in the frequency domain by the expression

$$\begin{aligned} \hat{F}(u, v) &= \left[\frac{H^*(u, v) S_f(u, v)}{S_f(u, v) |H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \\ &= \left[\frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v) \\ &= \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v) \end{aligned}$$

Minimum Mean Square Error (Wiener) Filtering

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v) \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)}} \right] G(u, v)$$

$H(u, v)$: degradation function

$H^*(u, v)$: complex conjugate of $H(u, v)$

$|H(u, v)|^2 = H^*(u, v)H(u, v)$

$S_\eta(u, v) = |N(u, v)|^2$ = power spectrum of the noise

$S_f(u, v) = |F(u, v)|^2$ = power spectrum of the undegraded image

Minimum Mean Square Error (Wiener) Filtering

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

K is a specified constant. Generally, the value of K is chosen interactively to yield the best visual results.

Minimum Mean Square Error (Wiener) Filtering



FIGURE 5.28 Comparison of inverse and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Some Measures (1)

Singal-to-Noise Ratio (SNR)

$$SNR = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u, v)|^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |N(u, v)|^2}$$

This ratio gives a measure of the level of information bearing singal power to the level of noise power.

Some Measures (2)

Mean Square Error (MSE)

$$\text{MSE} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x, y) - \hat{f}(x, y) \right]^2$$

Root-Mean-Square-Error (RMSE)

$$\text{RMSE} = \frac{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(x, y)^2}{\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |f(x, y) - \hat{f}(x, y)|^2}$$

Constrained Least Squares Filtering

- ▶ In Wiener filter, the power spectra of the undegraded image and noise must be known. Although a constant estimate is sometimes useful, it is not always suitable.
- ▶ Constrained least squares filtering just requires the mean and variance of the noise.

Constrained Least Squares Filtering

$$F(u, v) = \left[\frac{H^*(u, v)}{|H(u, v)|^2 + \gamma |P(u, v)|^2} \right] G(u, v)$$

$P(u, v)$ is the Fourier transform of the function

$$p(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

γ is a parameter

Examples

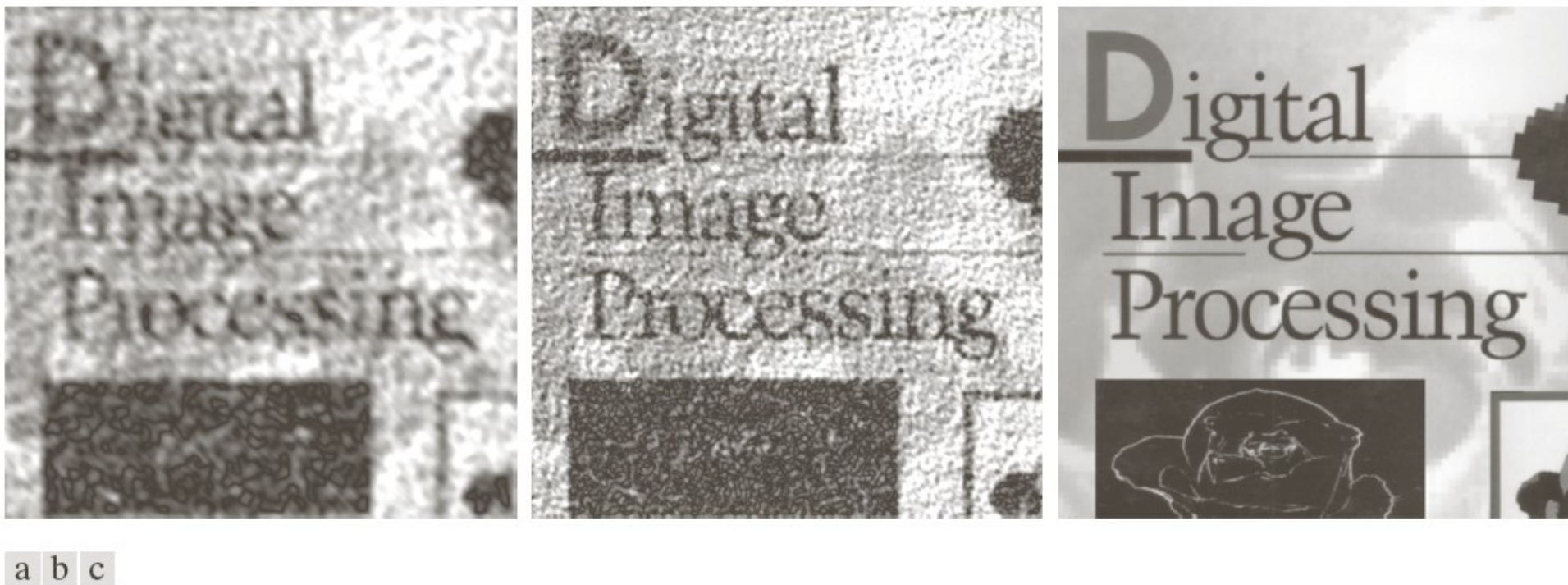
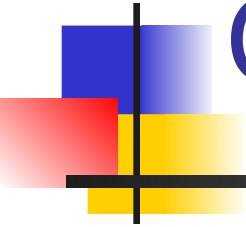


FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

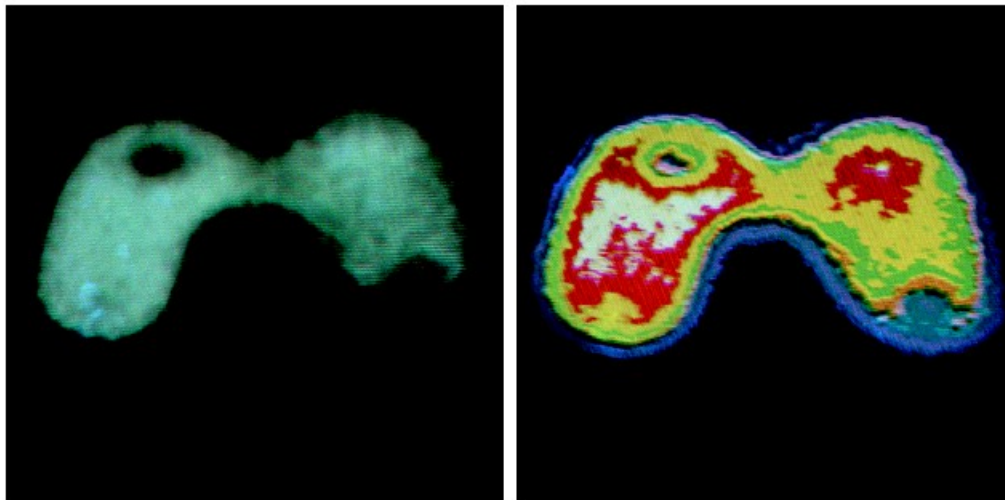


Color Image Processing

K K Singh

Preview

- Why use color in image processing?
 - Color is a **powerful descriptor**
 - Object identification and extraction
 - eg. Face detection using skin colors
 - Humans can **discern** thousands of color shades and intensities
 - c.f. Human discern only two dozen shades of grays



Visible light

- **Chromatic light** span the electromagnetic spectrum (EM) from 400 to 700 nm

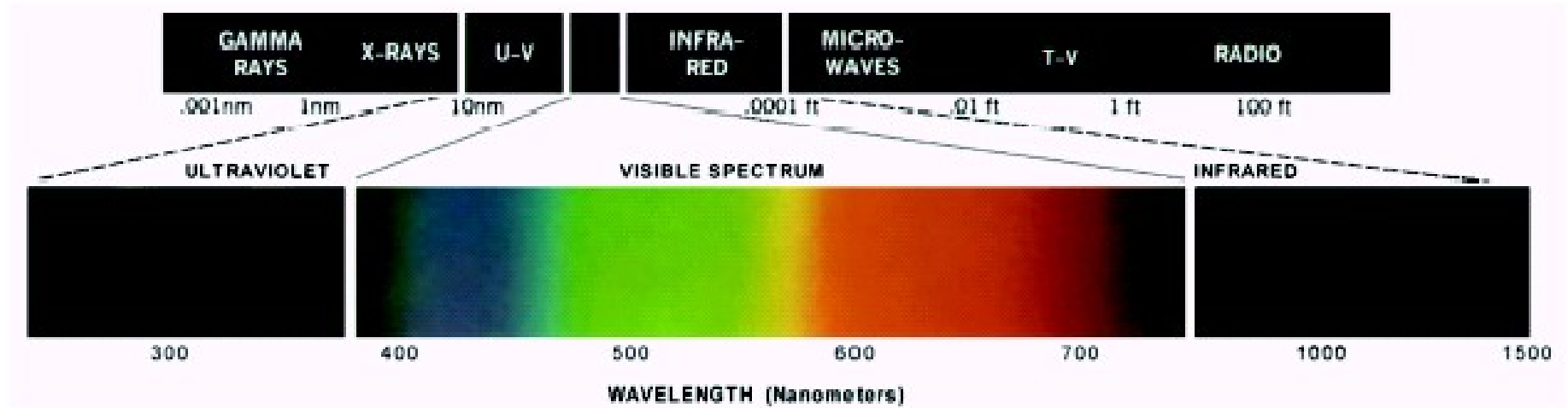
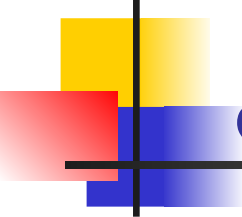


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)



Physical quantities to describe a chromatic light source

- **Radiance**: total amount of energy that flow from the light source, measured in **watts (W)**
- **Luminance**: amount of energy an observer *perceives* from a light source, measured in **lumens (lm)**
 - Far infrared light: high radiance, but 0 luminance
- **Brightness**: subjective descriptor that is hard to measure, similar to the achromatic notion of intensity

How human eyes sense light?

- 6~7M Cones are the sensors in the eye
- 3 principal sensing categories in eyes
 - Red light 65%, green light 33%, and blue light 2%

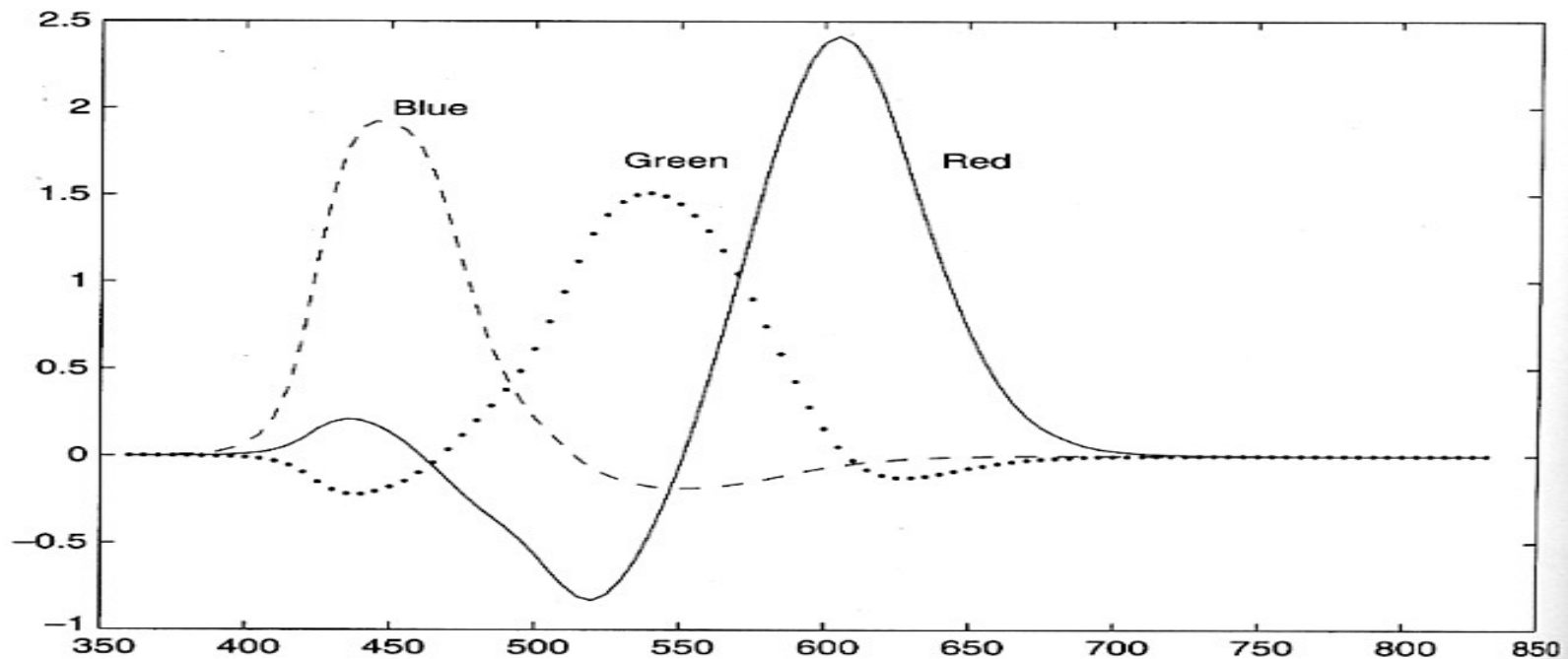
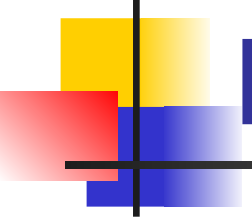


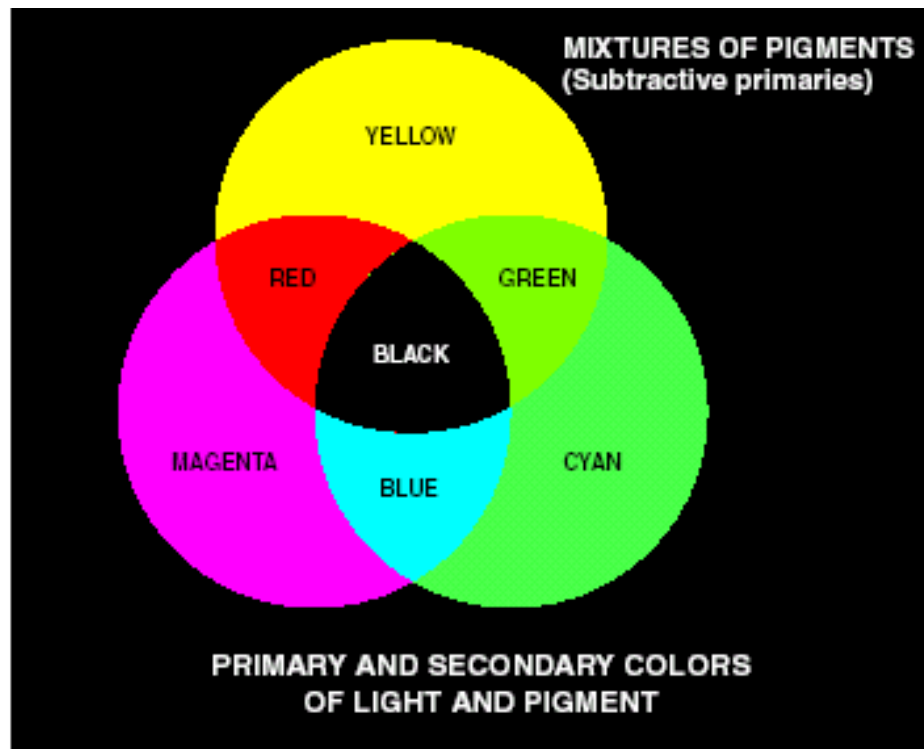
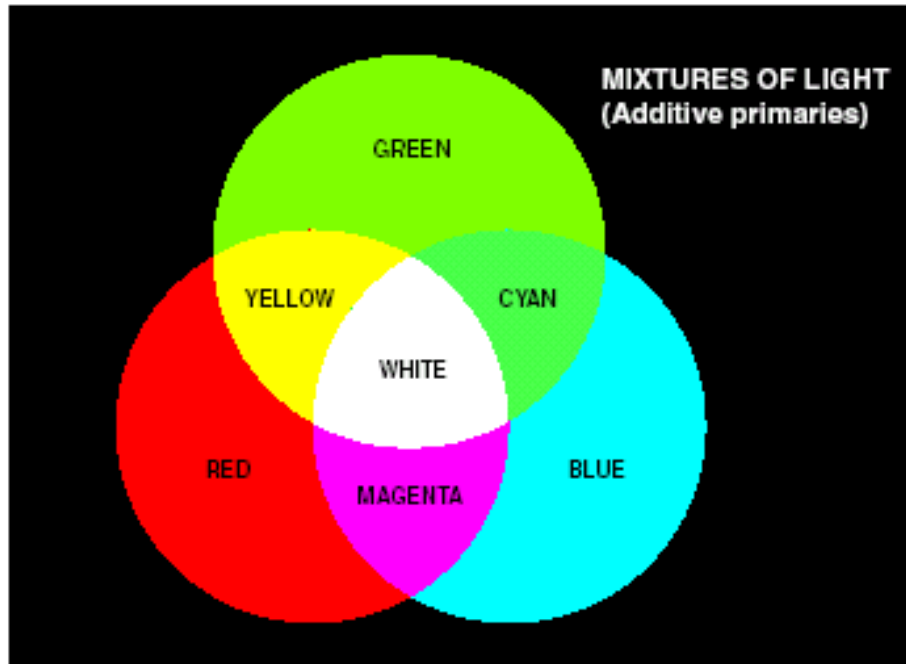
FIGURE B.1 RGB color-matching functions (CIE, 1931).

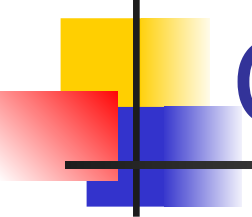


Primary and secondary colors

- In 1931, **CIE**(International Commission on Illumination) defines specific wavelength values to the **primary colors**
 - $B = 435.8 \text{ nm}$, $G = 546.1 \text{ nm}$, $R = 700 \text{ nm}$

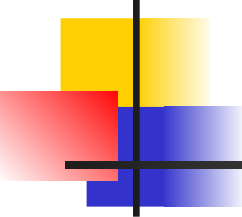
Secondary colors: $G+B=\text{Cyan}$, $R+G=\text{Yellow}$, $R+B=\text{Magenta}$



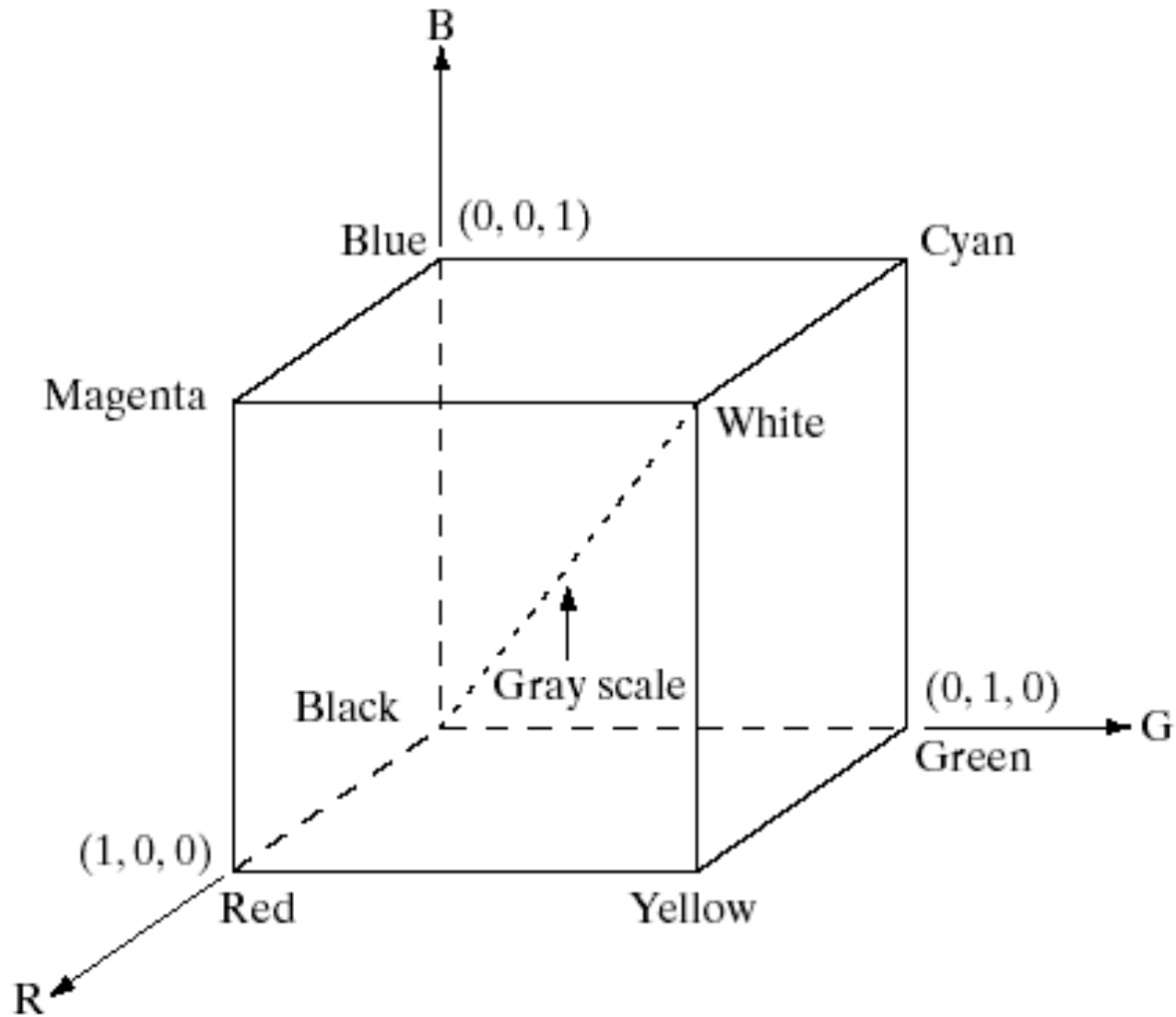


Color models

- Color model, color space, color system
 - Specify colors in a standard way
 - A **coordinate system** that each color is represented by a single point
- RGB model (monitor, video camera)
- CYM model (printer) Suitable for hardware or applications
- CYMK model
- HSI model - match the human description

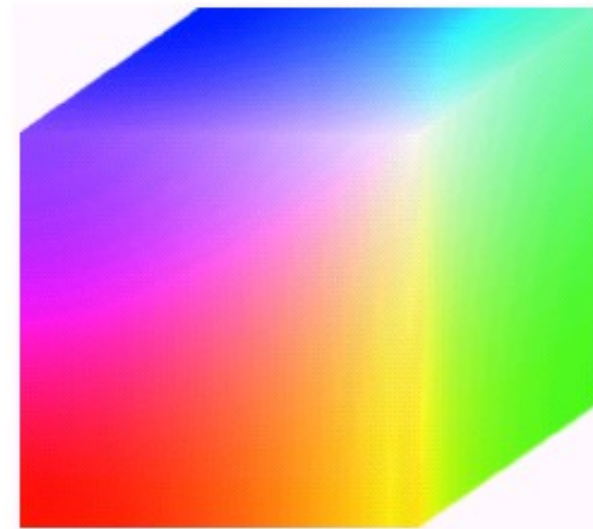
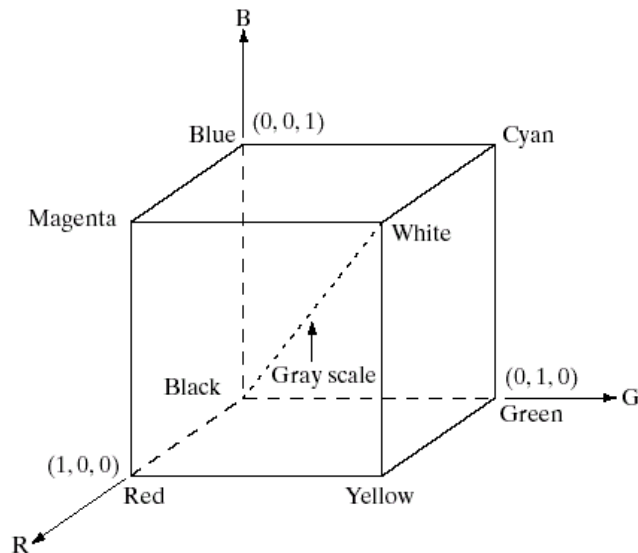


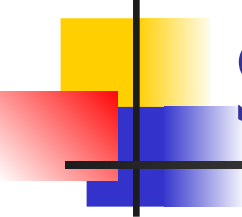
RGB color model



Pixel depth

- **Pixel depth**: the number of **bits** used to represent each pixel in RGB space
- **Full-color** image: 24-bit RGB color image
 - $(R, G, B) = (8 \text{ bits}, 8 \text{ bits}, 8 \text{ bits})$





Safe RGB colors

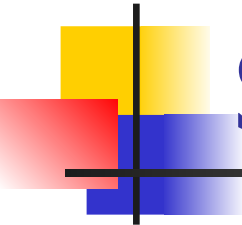
- **Subset of colors** is enough for some application
- **Safe RGB colors** (safe Web colors, safe browser colors)

Number System		Color Equivalents				
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

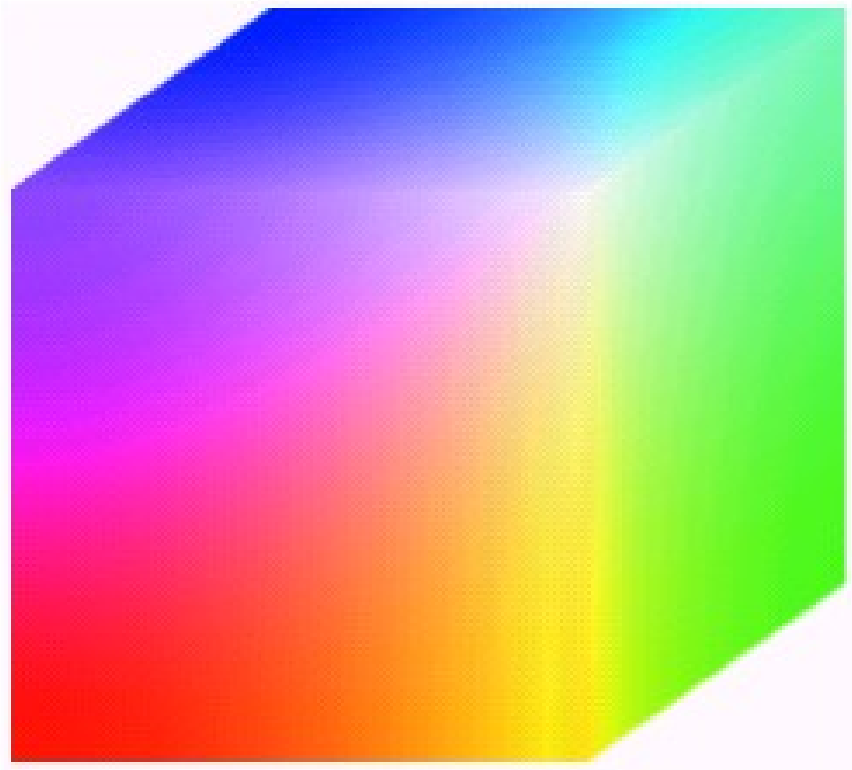
TABLE 6.1

Valid values of each RGB component in a safe color.

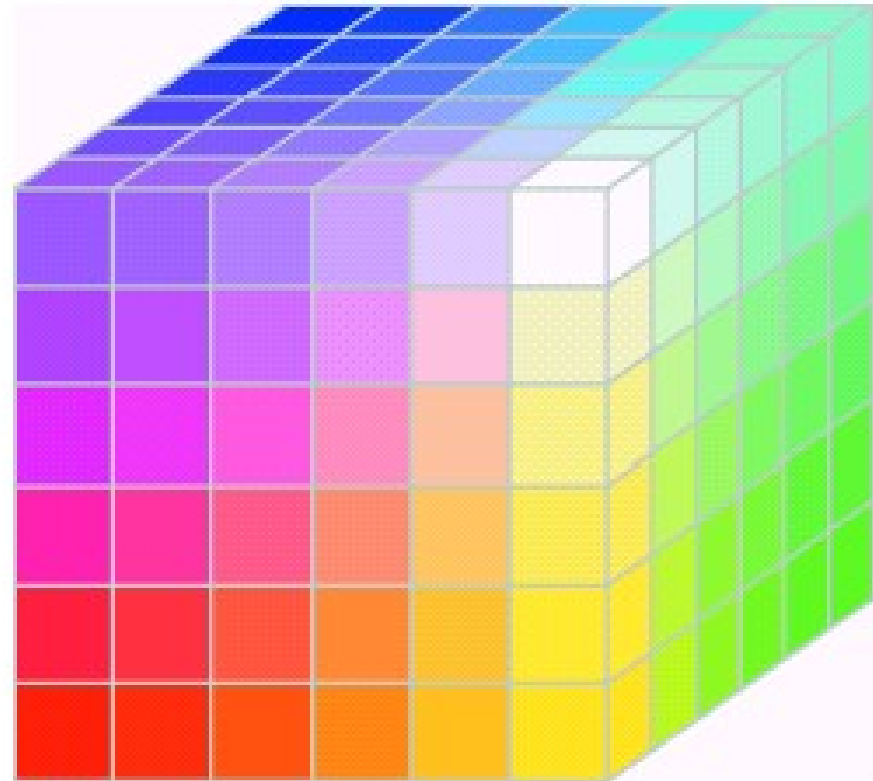
$$(6)^3 = 216$$



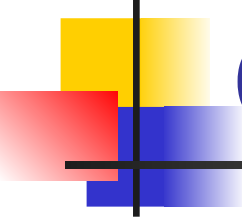
Safe RGB color (cont.)



Full color cube



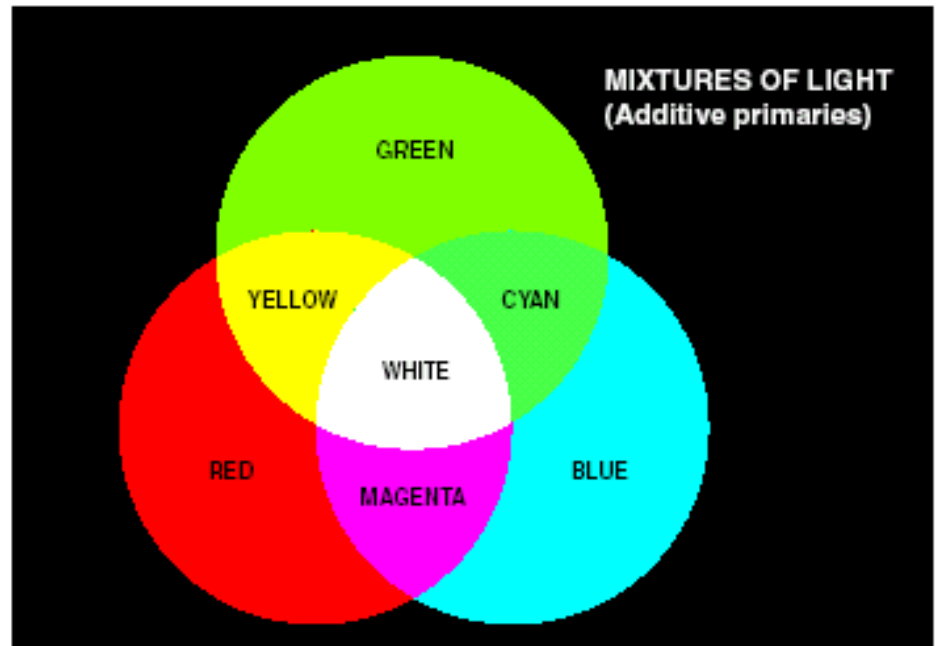
Safe color cube



CMY model (+Black = CMYK)

- **CMY**: secondary colors of light, or primary colors of pigments
- Used to generate hardcopy output

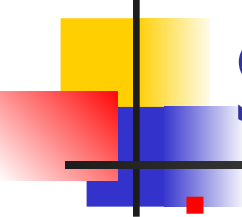
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$





HSI color model

- Will you describe a color using its R, G, B components?
- Human describe a color by its hue, saturation, and brightness
 - **Hue** : color attribute, represents dominant color as perceived by an observer.
 - **Saturation**: purity of color (white- $\rightarrow$ 0, primary color- $\rightarrow$ 1), represents the relative purity or the amount of white mixed with a hue.
 - **Brightness**: achromatic notion of **intensity**
- Hue and saturation together called **chromaticity**.



3D-polar Coordinate Colour Spaces

- ***Hue (dominant colour seen)***

- Wavelength of the pure colour observed in the signal.
- Distinguishes red, yellow, green, etc.
- More the 400 hues can be seen by the human eye.

- ***Saturation (degree of dilution)***

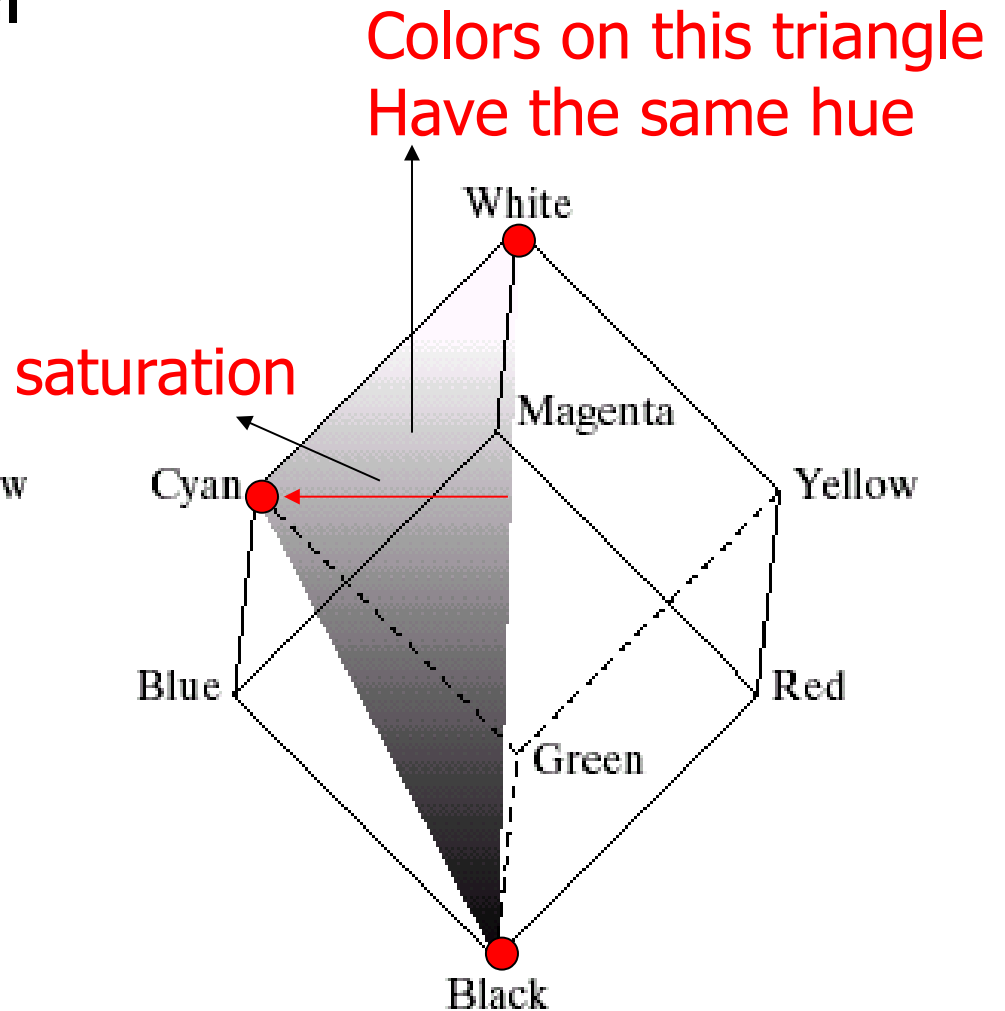
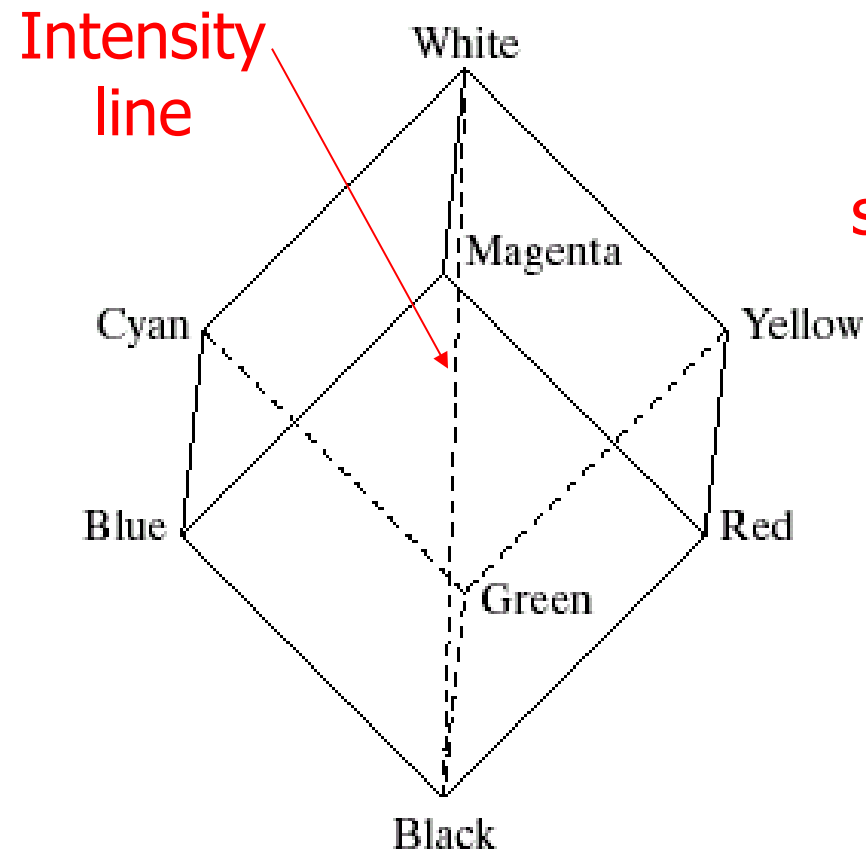
- Inverse of the quantity of “white” present in the signal. A pure colour has 100% saturation, the white and grey have 0% saturation.
- About 20 saturation levels are visible per hue.

- ***Brightness***

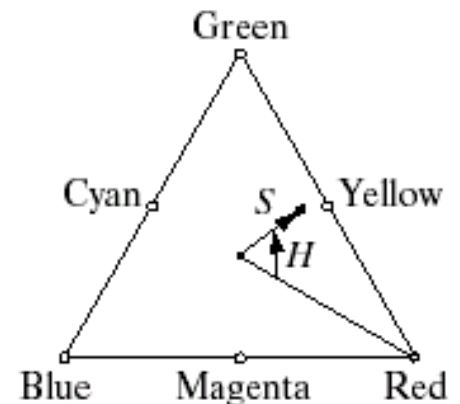
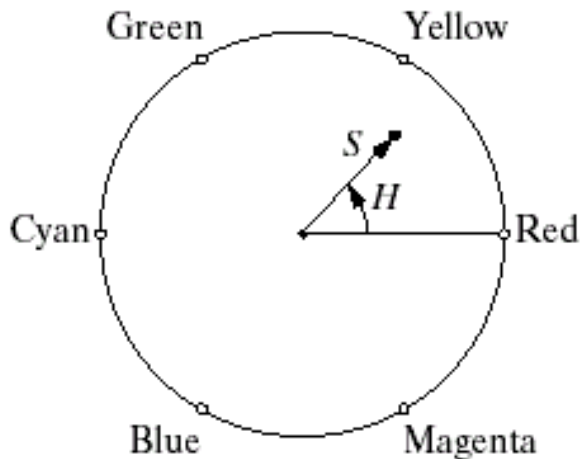
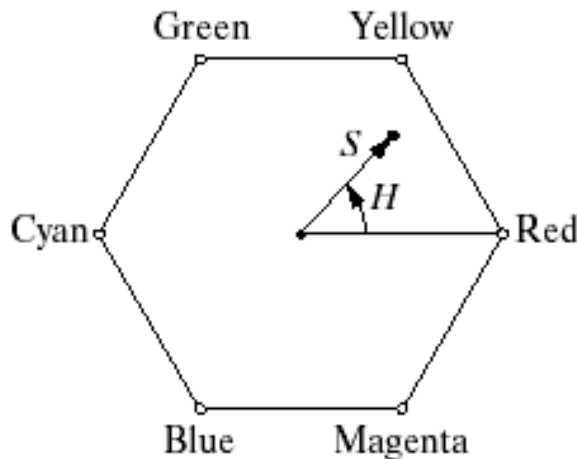
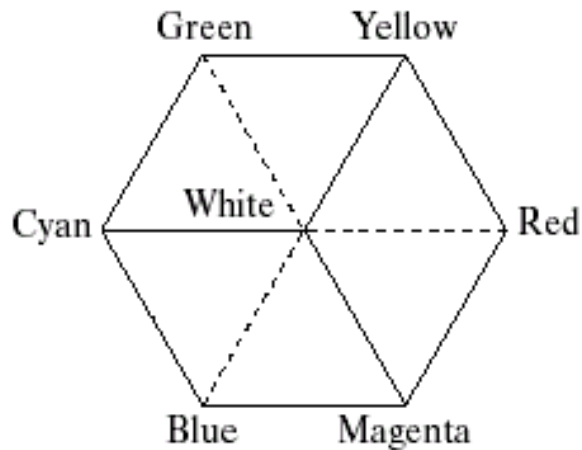
- Amount of light emitted.
- Distinguishes the grey levels.
- The human eye perceives about 100 levels.

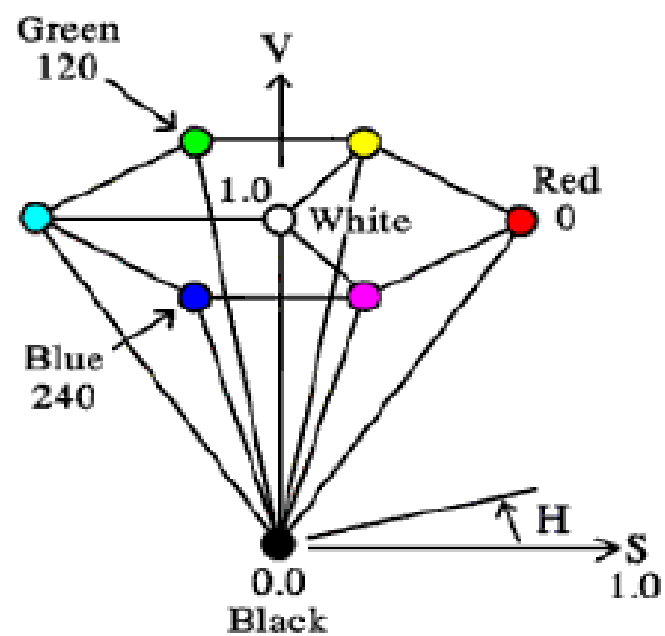
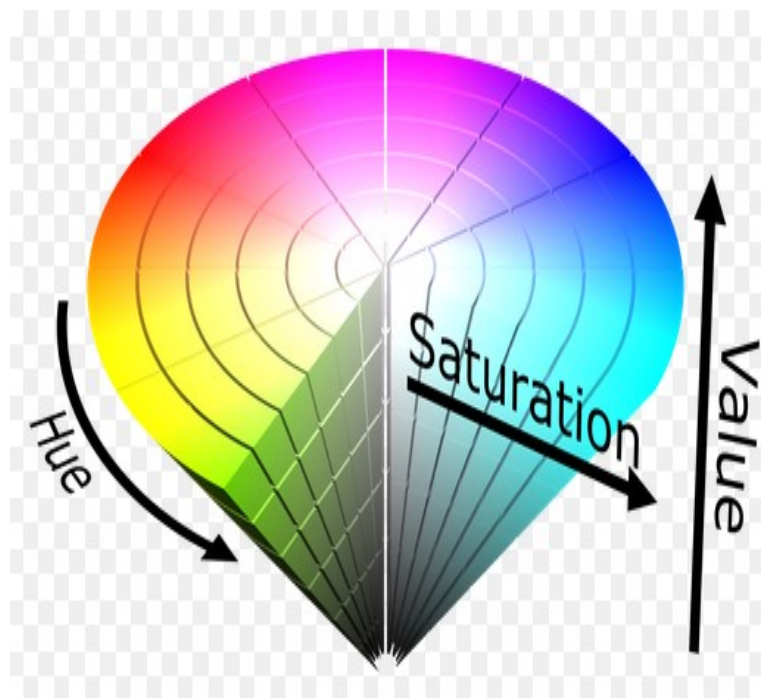
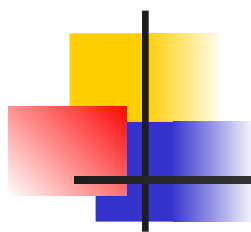
HSI color model (cont.)

- RGB -> HSI model

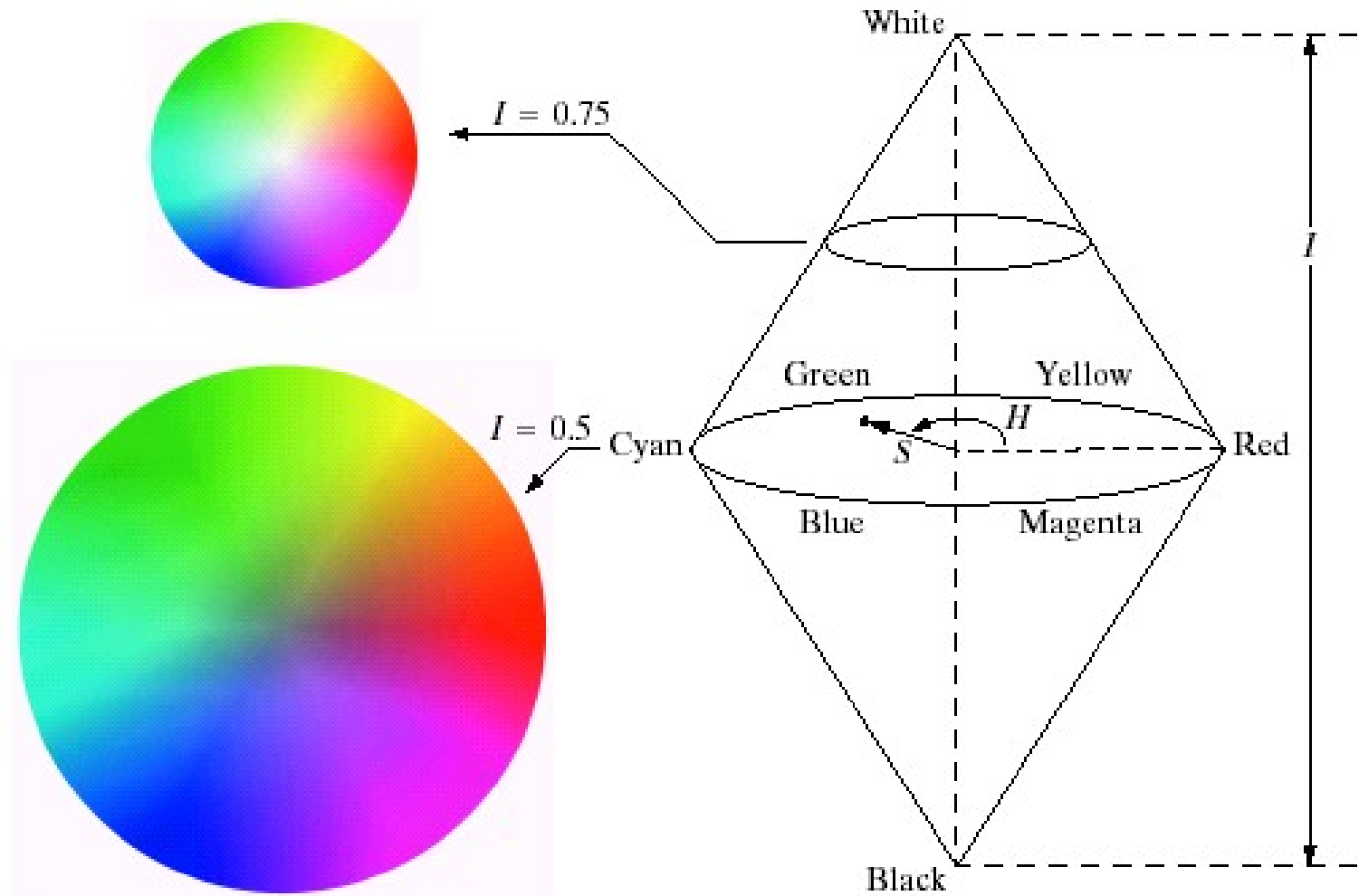


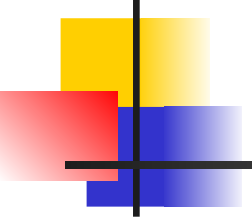
HSI model: hue and saturation





HSI model





Color Image Processing

- RGB to HSI Conversion

$$I = \frac{1}{3}(R + G + B), \quad \text{where } 0 \leq I, R, G, B \leq 1$$

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\}, \quad \text{if } G > B$$

else..... $H = 360^\circ - H$

$$S = 1 - \frac{3}{R + G + B} \times (\min \{R, G, B\})$$



RGB to HSV conversion formula , As per cv2.COLOR_BGR2HSV

The R, G, B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

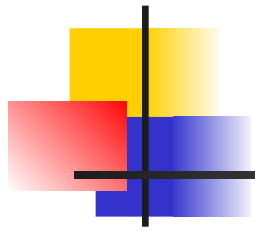
$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{max}$$



■ HSI to RGB Conversion

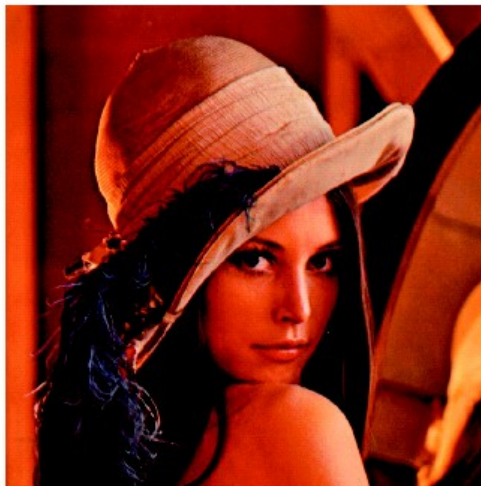
$$B = I(1 - S)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad \text{assume } 0^\circ \leq H \leq 120^\circ$$

$$G = 3I - R - B$$

$$\text{if } 120 < H < 240, \quad H = H - 120^\circ$$

original



R



G



B



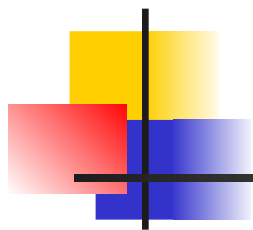
H



S



I



Thank
You

Image Processing in Frequency Domain



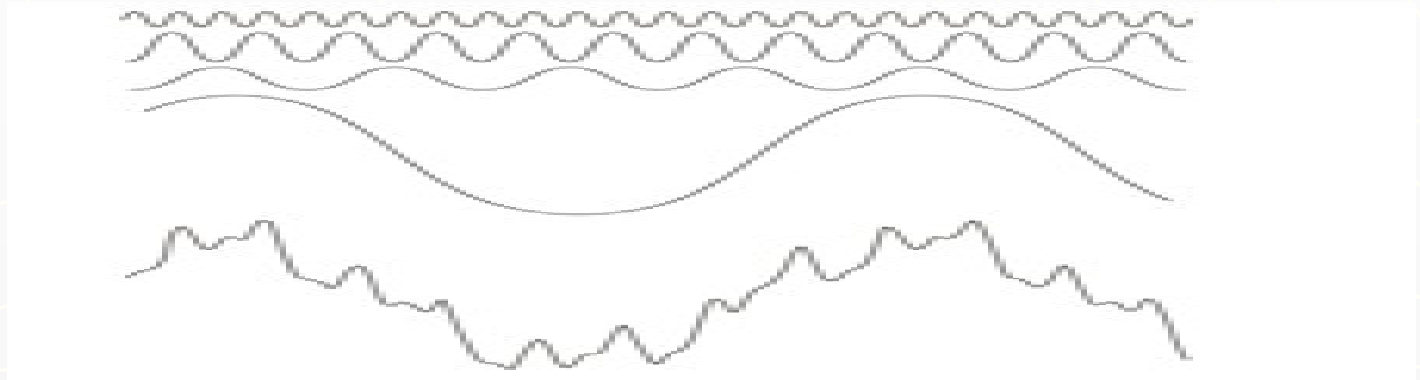
Outline

- ▶ Fourier Series , Fourier Transform
- ▶ Convolution
- ▶ 2-D discrete Fourier transform
- ▶ Filtering in Fourier Transform Domain

Fourier Series and Fourier Transform

► Fourier Series

Any periodic function can be expressed as the sum of sines and /or cosines of different frequencies, each multiplied by a different coefficients



► Fourier Transform

Any function that is not periodic can be expressed as the integral of sines and /or cosines multiplied by a weighing function

Preliminary Concepts

$j = \sqrt{-1}$, a complex number

$$C = R + jI$$

the conjugate

$$C^* = R - jI$$

$$|C| = \sqrt{R^2 + I^2} \text{ and } \theta = \arctan(I / R)$$

$$C = |C| (\cos \theta + j \sin \theta)$$

Using Euler's formula,

$$C = |C| e^{j\theta}$$

Fourier Series

$$\theta = \omega \tau = (2\pi \cdot n / T) \tau$$

A function $f(t)$ of a continuous variable t that is periodic with period, T , can be expressed as the sum of sines and cosines multiplied by appropriate coefficients

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j \frac{2\pi n}{T} t}$$

where

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j \frac{2\pi n}{T} t} dt \quad \text{for } n = 0, \pm 1, \pm 2, \dots$$

Impulses and the Sifting Property (1)

A *unit impulse* of a continuous variable t located at $t=0$, denoted $\delta(t)$, defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases}$$

and is constrained also to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

The *sifting property*

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0)$$

$$\int_{-\infty}^{\infty} f(t) \delta(t) dt = f(0)$$

Impulses (Discrete) and the Sifting Property (2)

A *unit impulse* of a discrete variable x located at $x=0$, denoted $\delta(x)$, defined as

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

and is constrained also to satisfy the identity

$$\sum_{x=-\infty}^{\infty} \delta(x) = 1$$

The *sifting property*

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$$

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0)$$

Impulses and the Sifting Property (3)

impulse train $s_{\Delta T}(t)$,

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$$

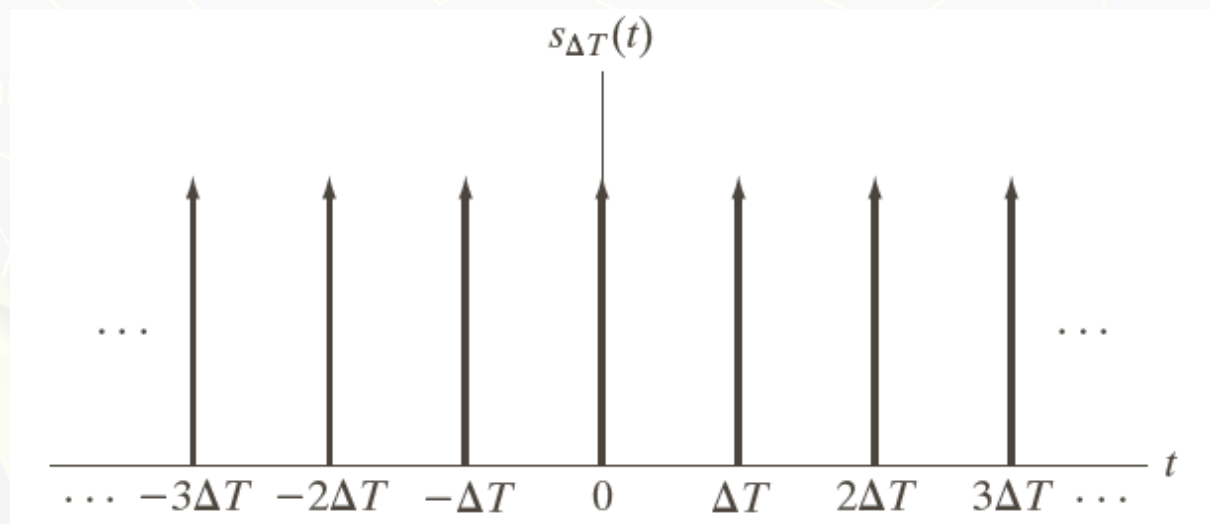


FIGURE 4.3 An impulse train.

Fourier Transform: Continuous Variable

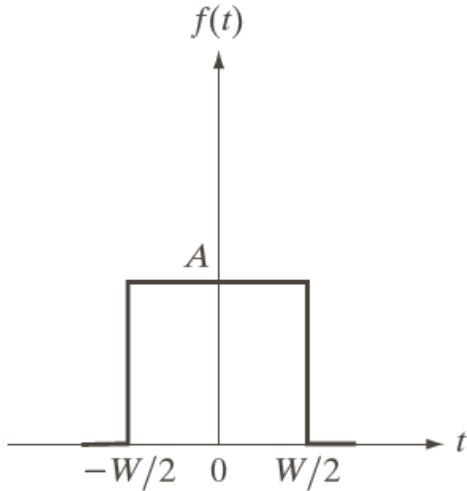
The *Fourier Transform* of a continuous function $f(t)$

$$F(\mu) = \mathfrak{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t} dt$$

The *Inverse Fourier Transform* of $F(\mu)$

$$f(t) = \mathfrak{F}^{-1}\{F(\mu)\} = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t} d\mu$$

Fourier Transform: One Continuous Variable



a b c

FIGURE 4.4 (a) A simple function that is finite in both directions.

Fourier Transform: Impulses

The Fourier transform of a unit impulse located at the origin:

$$F(\mu) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi\mu t} dt$$

The Fourier transform of a unit impulse located at $t = t_0$:

$$F(\mu) = \int_{-\infty}^{\infty} \delta(t - t_0) e^{-j2\pi\mu t} dt$$

Fourier Transform and Convolution

The convolution of two functions is denoted by the operator $\star$

$$f(t) \star h(t) = \int_{-\infty}^{\infty} f(\tau)h(t - \tau)d\tau$$

Fourier Transform and Convolution

Fourier Transform Pairs

$$f(t) \star h(t) \Leftrightarrow H(\mu)F(\mu)$$

$$f(t)h(t) \Leftrightarrow H(\mu) \star F(\mu)$$

The Discrete Fourier Transform (DFT) of One Variable

$$F(\mu) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi\mu x/M}, \quad \mu = 0, 1, \dots, M-1$$

$$f(x) = \frac{1}{M} \sum_{\mu=0}^{M-1} F(\mu) e^{j2\pi\mu x/M}, \quad x = 0, 1, 2, \dots, M-1$$

2-D Impulse and Sifting Property: Continuous

The impulse $\delta(t, z)$,
$$\delta(t, z) = \begin{cases} \infty & \text{if } t = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

and
$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t, z) dt dz = 1$$

The sifting property

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) \delta(t, z) dt dz =$$

2-D Impulse and Sifting Property: Discrete

The impulse $\delta(x, y)$,
$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{otherwise} \end{cases}$$

The sifting property

$$\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f(x, y) \delta(x, y) = f(0, 0)$$

and

$$\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f(x, y) \delta(x - x_0, y - y_0) = f(x_0, y_0)$$

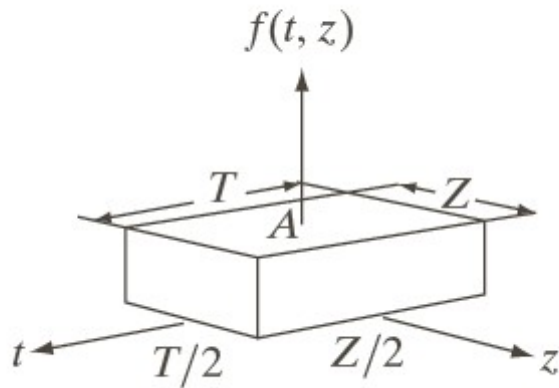
2-D Fourier Transform: Continuous

$$F(\mu, \nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-j2\pi(\mu t + \nu z)} dt dz$$

and

$$f(t, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\mu, \nu) e^{j2\pi(\mu t + \nu z)} d\mu d\nu$$

2-D Fourier Transform: Continuous



$$\begin{aligned} F(\mu, \nu) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-j2\pi(\mu t + \nu z)} dt dz \\ &= \int_{-T/2}^{T/2} \int_{-Z/2}^{Z/2} A e^{-j2\pi(\mu t + \nu z)} dt dz \\ &= ATZ \left[\frac{\sin(\pi\mu T)}{\pi\mu T} \right] \left[\frac{\sin(\pi\nu T)}{\pi\nu T} \right] \end{aligned}$$

a b

FIGURE 4.13 (a) A 2-D function, and (b) a section of its spectrum (not to scale). The block is longer along the t -axis, so the spectrum is more “contracted” along the μ -axis. Compare with Fig. 4.4.

2-D Discrete Fourier Transform and Its Inverse

DFT:

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

$$\mu = 0, 1, 2, \dots, M-1; \nu = 0, 1, 2, \dots, N-1;$$

$f(x, y)$ is a digital image of size $M \times N$.

IDFT:

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

Properties of the 2-D DFT

relationships between spatial and frequency intervals

Let ΔT and ΔZ denote the separations between samples, then the separations between the corresponding discrete, frequency domain variables are given by

$$\Delta\mu = \frac{1}{M\Delta T}$$

and

$$\Delta\nu = \frac{1}{N\Delta Z}$$

Properties of the 2-D DFT

translation and rotation

$$f(x, y)e^{j2\pi(\mu_0 x/M + \nu_0 y/N)} \Leftrightarrow F(\mu - \mu_0, \nu - \nu_0)$$

and

$$f(x - x_0, y - y_0) \Leftrightarrow F(\mu, \nu)e^{-j2\pi(\mu x_0/M + \nu y_0/N)}$$

Using the polar coordinates

$$x = r \cos \theta \quad y = r \sin \theta \quad \mu = \omega \cos \varphi \quad \nu = \omega \sin \varphi$$

results in the following transform pair:

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$

Properties of the 2-D DFT

periodicity

2-D Fourier transform and its inverse are infinitely periodic

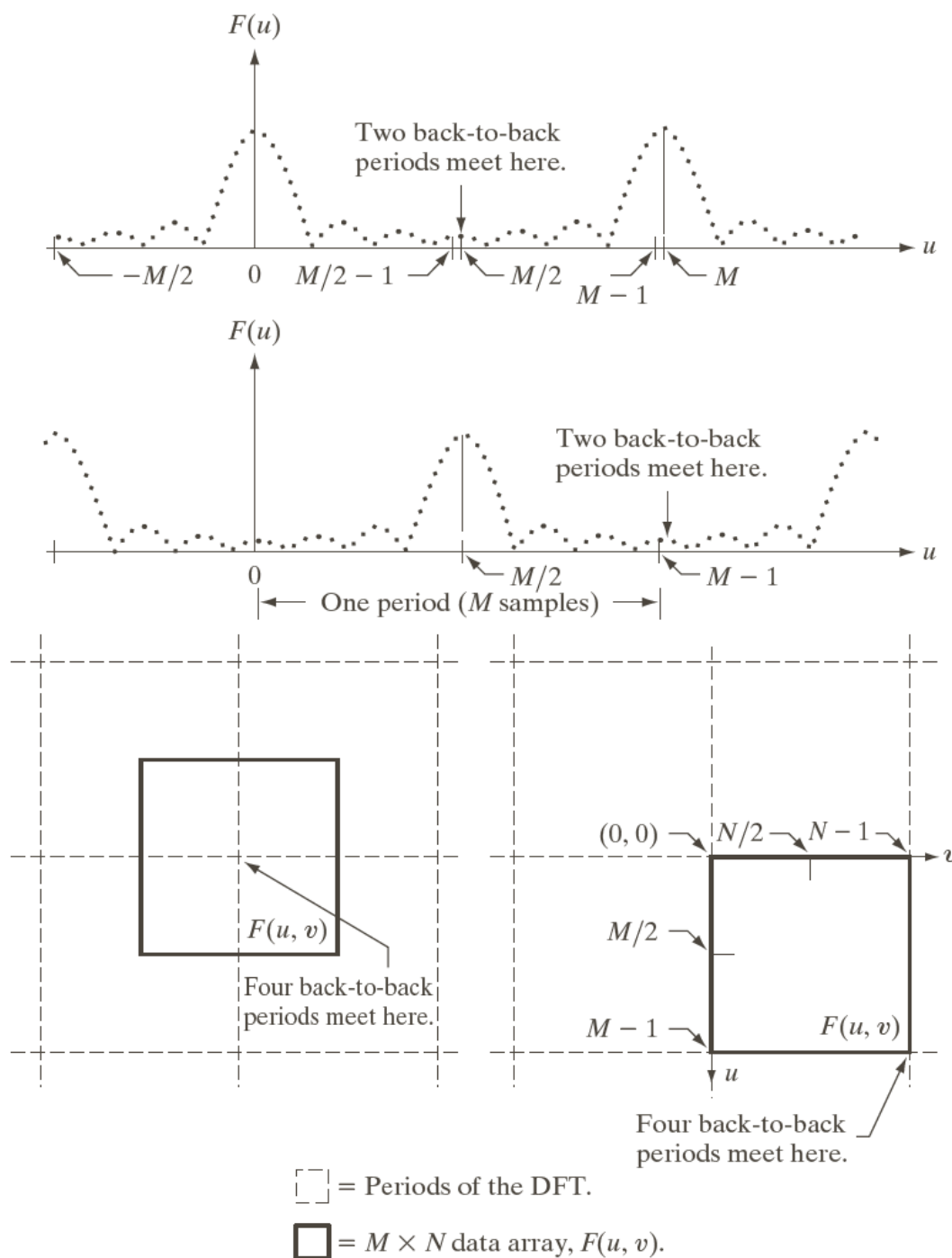
$$F(\mu, \nu) = F(\mu + k_1 M, \nu) = F(\mu, \nu + k_2 N) = F(\mu + k_1 M, \nu + k_2 N)$$

$$f(x, y) = f(x + k_1 M, y) = f(x, y + k_2 N) = f(x + k_1 M, y + k_2 N)$$

$$f(x) e^{j2\pi(\mu_0 x/M)} \Leftrightarrow F(\mu - \mu_0)$$

$$\mu_0 = M / 2, \quad f(x)(-1)^x \Leftrightarrow F(\mu - M / 2)$$

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(\mu - M / 2, \nu - N / 2)$$



a
b
c d

FIGURE 4.23

Centering the Fourier transform.

(a) A 1-D DFT showing an infinite number of periods.

(b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$.

(c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods.

(d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

Properties of the 2-D DFT

Symmetry

Spatial Domain [†]		Frequency Domain [†]
1)	$f(x, y)$ real	$\Leftrightarrow F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow F(u, v)$ complex and odd

TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

[†]Recall that x, y, u , and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y , and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

2-D Convolution Theorem

1-D convolution

$$f(x) \star h(x) = \sum_{m=0}^{M-1} f(m)h(x-m)$$

2-D convolution

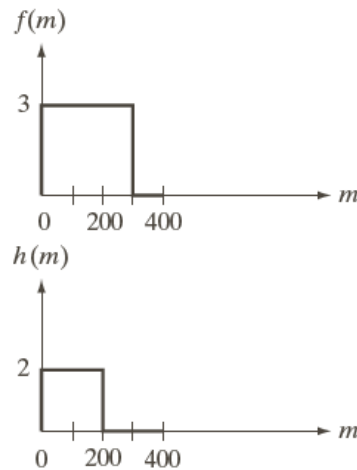
$$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x-m, y-n)$$

$$x = 0, 1, 2, \dots, M-1; y = 0, 1, 2, \dots, N-1.$$

$$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$$

$$f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$$

An Example of Convolution



Mirroring h
about the
origin

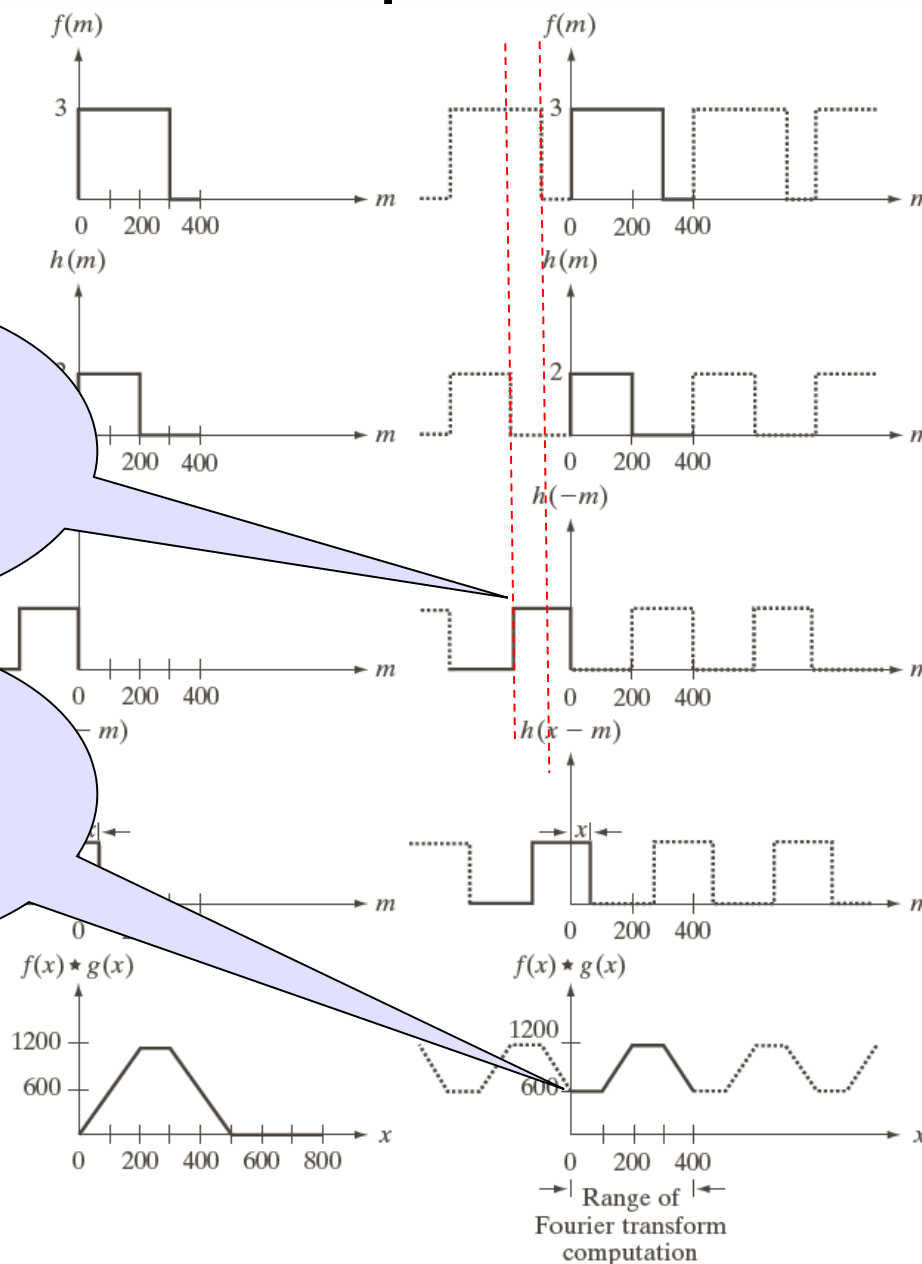
Translating
the mirrored
function by x

Computing
the sum for
each x

a	f
b	g
c	h
d	i
e	j

FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.

An Example of Convolution



a	f
b	g
c	h
d	i
e	j

FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.

Zero Padding

- ▶ Consider two functions $f(x)$ and $h(x)$ composed of A and B samples, respectively
- ▶ Append zeros to both functions so that they have the same length, denoted by P , then wraparound is avoided by choosing

$$P \geq A+B-1$$

Zero Padding

- ▶ Let $f(x,y)$ and $h(x,y)$ be two image arrays of sizes $A \times B$ and $C \times D$ pixels, respectively. Wraparound error in their convolution can be avoided by padding these functions with zeros

$$f_p(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A-1 \text{ and } 0 \leq y \leq B-1 \\ 0 & A \leq x \leq P \text{ or } B \leq y \leq Q \end{cases}$$

$$h_p(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C-1 \text{ and } 0 \leq y \leq D-1 \\ 0 & C \leq x \leq P \text{ or } D \leq y \leq Q \end{cases}$$

Here $P \geq A + C - 1; Q \geq B + D - 1$

Summary

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
3) Polar representation	$F(u, v) = F(u, v) e^{j\phi(u, v)}$
4) Spectrum	$ F(u, v) = [R^2(u, v) + I^2(u, v)]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = F(u, v) ^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

Summary

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M+vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \varphi \quad v = \omega \sin \varphi$
6) Convolution theorem [†]	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

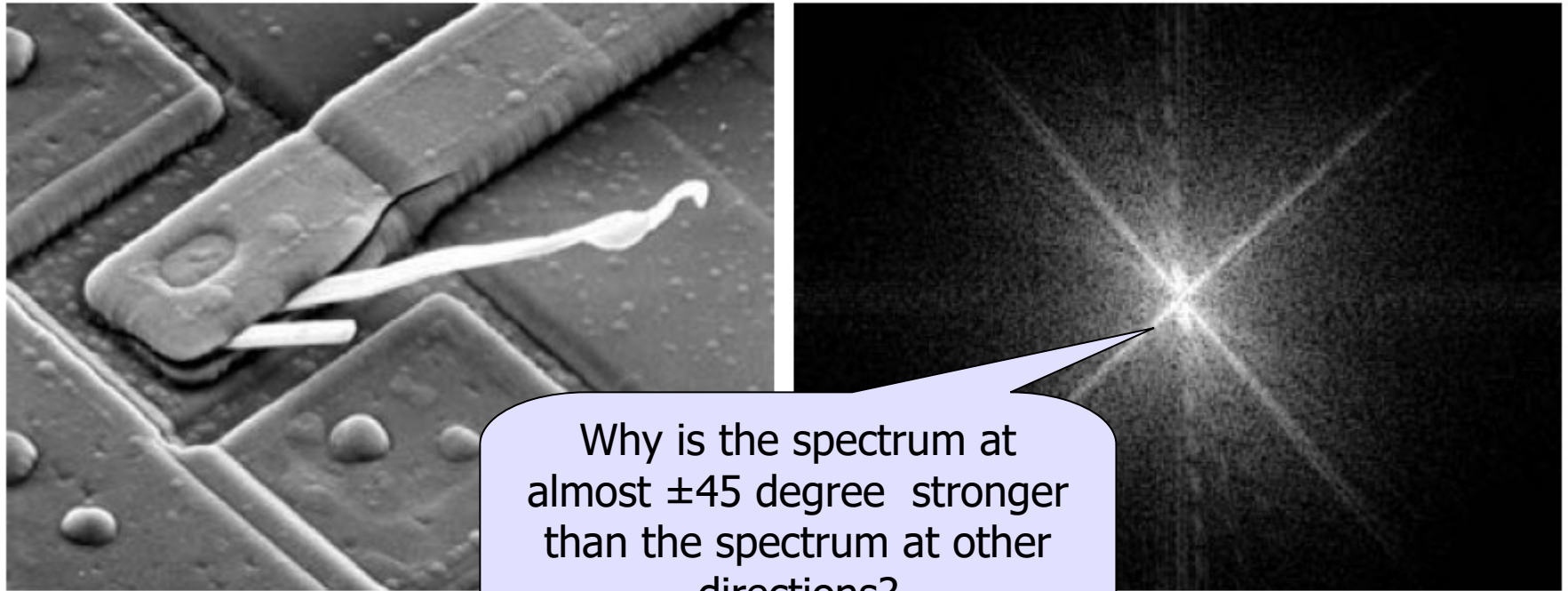
(Continued)

Summary

Name	DFT Pairs
7) Correlation theorem [†]	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$ $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\text{rect}[a, b] \Leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j \frac{1}{2} [\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)]$
11) Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\frac{1}{2} [\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) <i>Differentiation</i> (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)	$\left(\frac{\partial}{\partial t}\right)^m \left(\frac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu, \nu)$ $\frac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \frac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
13) <i>Gaussian</i>	$A 2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow A e^{-(\mu^2+\nu^2)/2\sigma^2} \quad (A \text{ is a constant})$

[†] Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

The Basic Filtering in the Frequency Domain



a b

FIGURE 4.29 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

The Basic Filtering in the Frequency Domain

- ▶ Modifying the Fourier transform of an image
- ▶ Computing the inverse transform to obtain the processed result

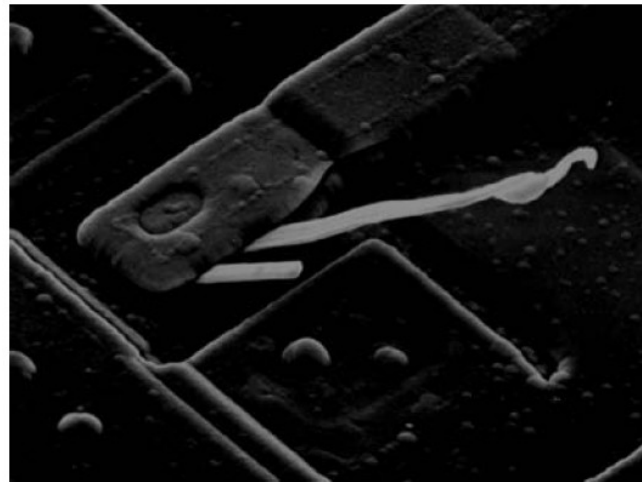
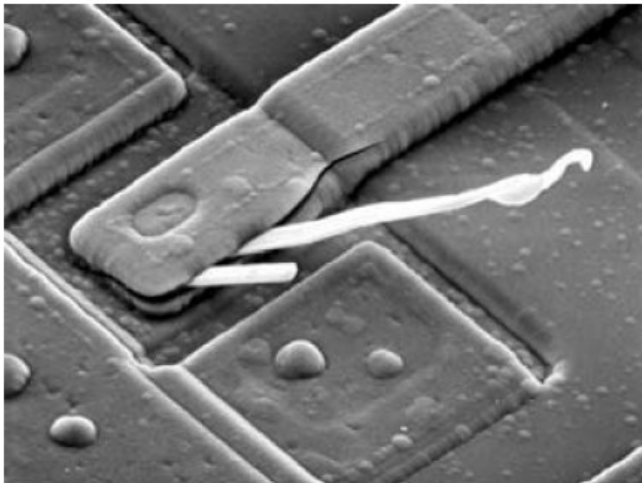
$$g(x, y) = \mathfrak{F}^{-1}\{H(u, v)F(u, v)\}$$

$F(u, v)$ is the DFT of the input image

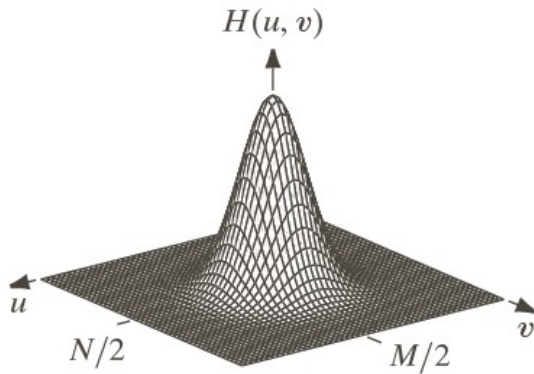
$H(u, v)$ is a filter function.

The Basic Filtering in the Frequency Domain

- ▶ In a filter $H(u,v)$ that is 0 at the center of the transform and 1 elsewhere, what's the output image?



The Basic Filtering in the Frequency Domain



a	b	c
d	e	f

06/28

FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

Correspondence Between Filtering in the Spatial and Frequency Domains (1)

Let $H(u)$ denote the 1-D frequency domain Gaussian filter

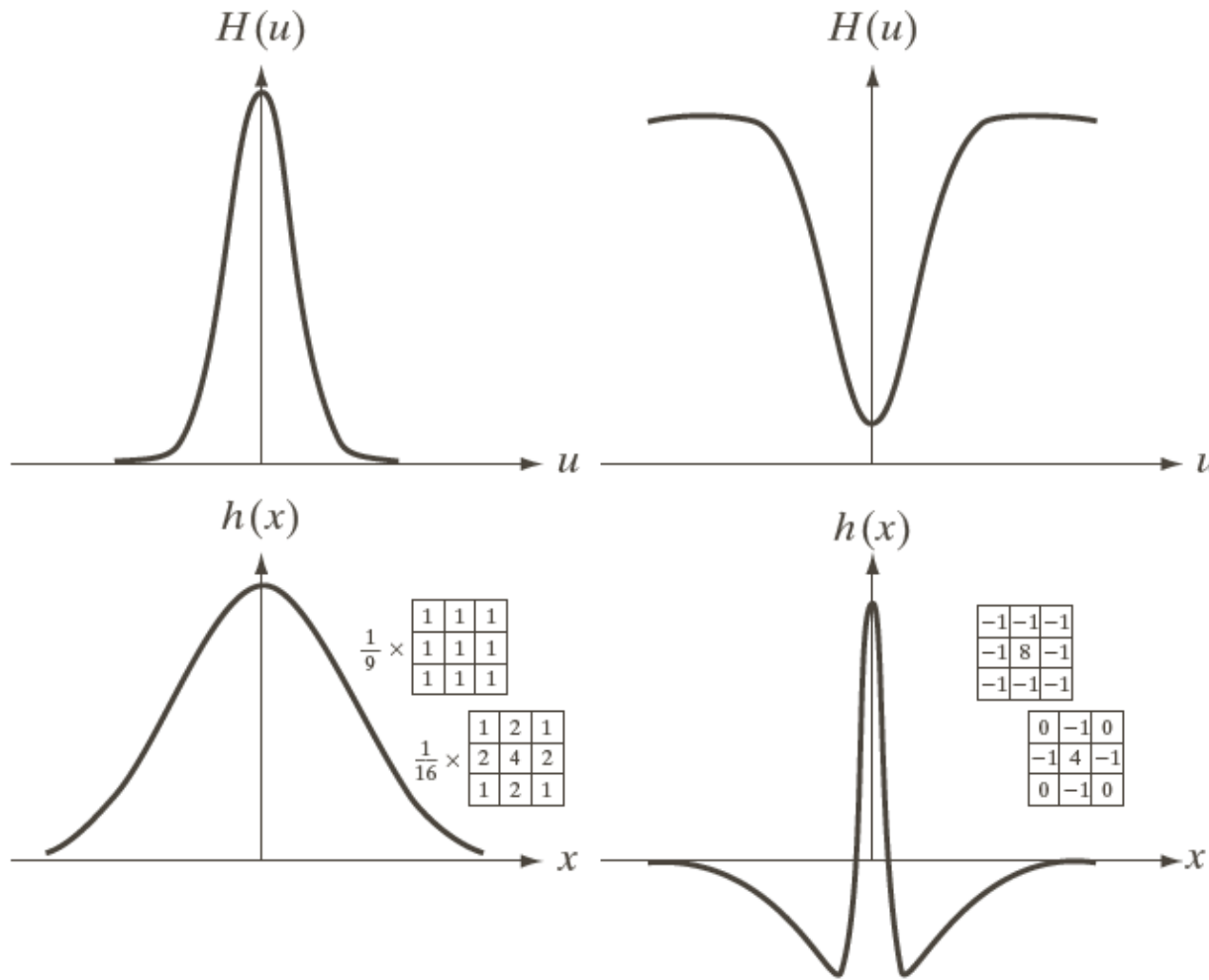
$$H(u) = Ae^{-u^2/2\sigma^2}$$

The corresponding filter in the spatial domain

$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

1. Both components are Gaussian and real
2. The functions behave reciprocally

Correspondence Between Filtering in the Spatial and Frequency Domains (3)



a	c
b	d

FIGURE 4.37

(a) A 1-D Gaussian lowpass filter in the frequency domain. (b) Spatial lowpass filter corresponding to (a). (c) Gaussian highpass filter in the frequency domain. (d) Spatial highpass filter corresponding to (c). The small 2-D masks shown are spatial filters we used in Chapter 3.

Generate $H(u,v)$

1. Multiply $h_p(x, y)$ by $(-1)^{x+y}$ to center the frequency domain filter
2. Compute the forward DFT of the result in (1)
3. Set the real part of the resulting DFT to 0 to account for parasitic real parts
4. Multiply the result by $(-1)^{u+v}$, which is implicit when $h(x, y)$ was moved to the center of $h_p(x, y)$.

Image Smoothing Using Filter Domain Filters: ILPF

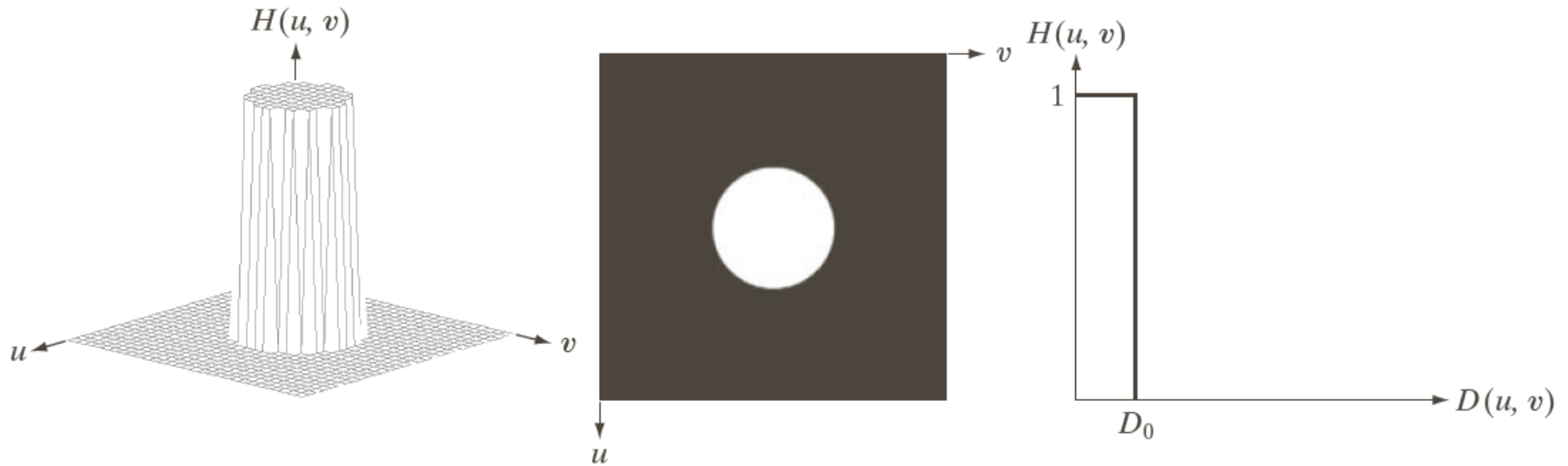
Ideal Lowpass Filters (ILPF)

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

D_0 is a positive constant and $D(u, v)$ is the distance between a point (u, v) in the frequency domain and the center of the frequency rectangle

$$D(u, v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$

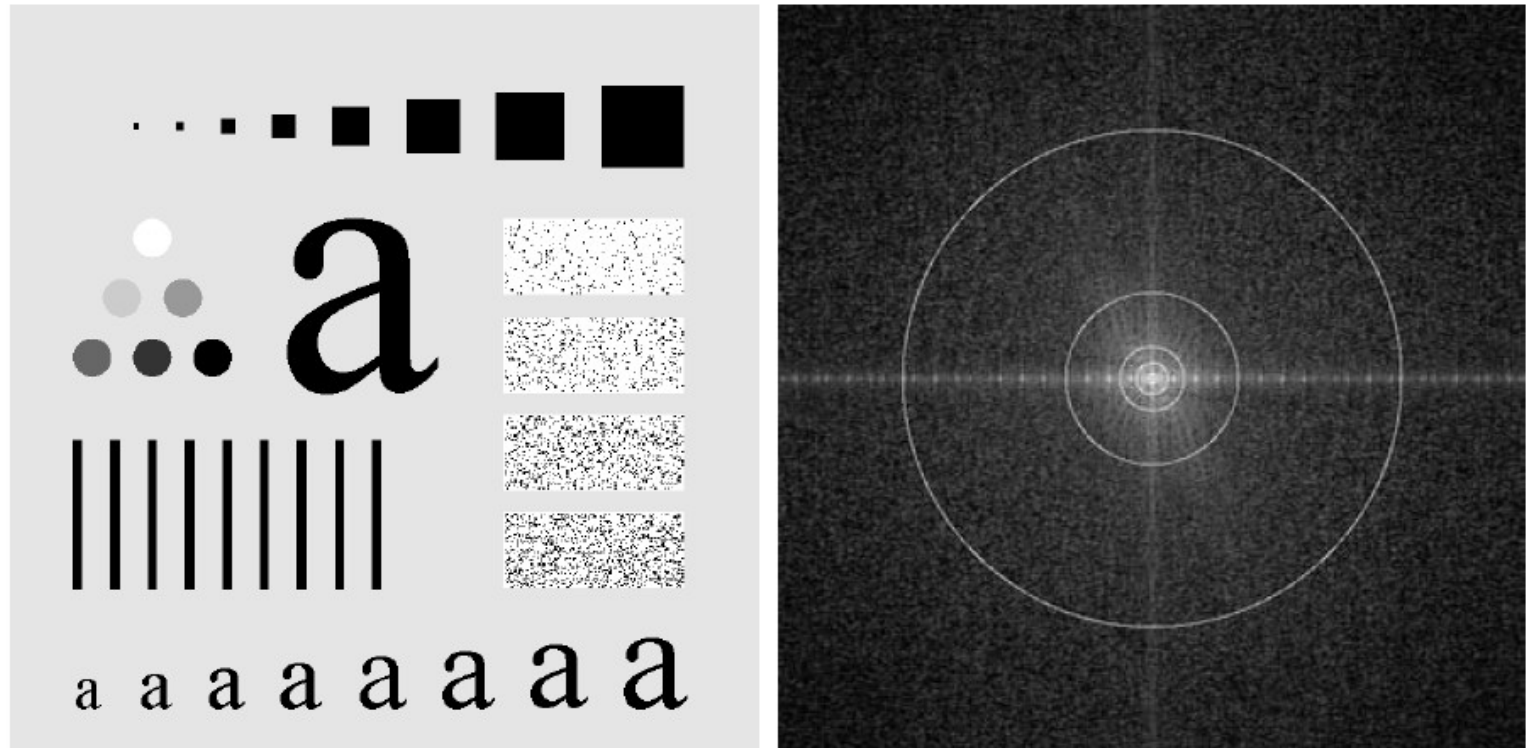
Image Smoothing Using Filter Domain Filters: ILPF



a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

ILPF Filtering Example



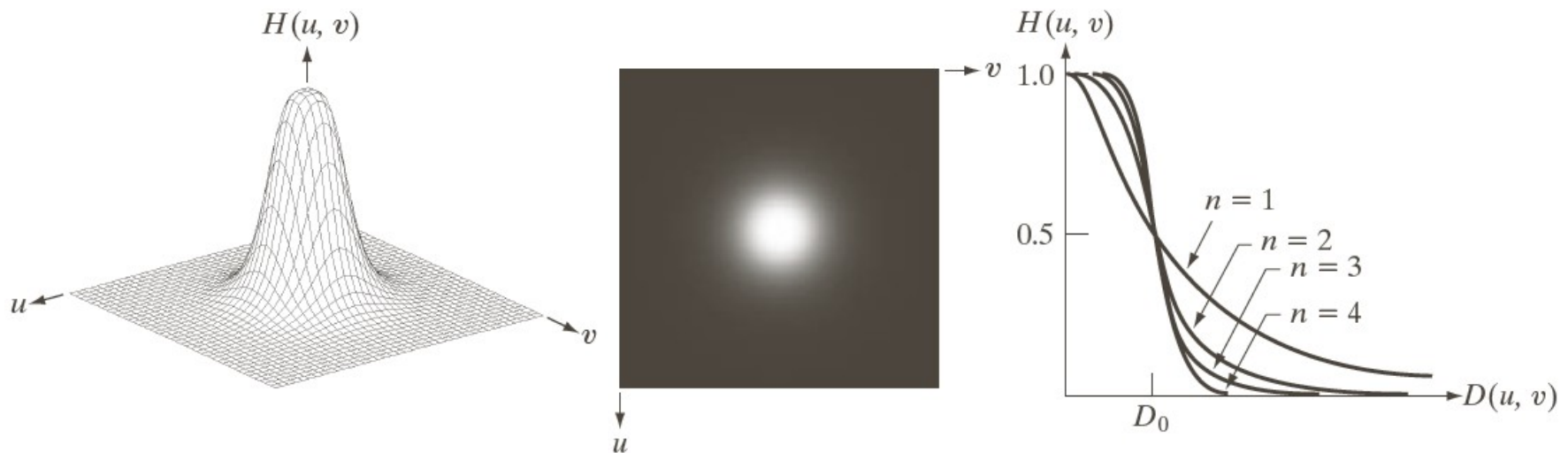
a b

FIGURE 4.41 (a) Test pattern of size 688×688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the padded image power, respectively.

Image Smoothing Using Filter Domain Filters: BLPF

Butterworth Lowpass Filters (BLPF) of order n and with cutoff frequency D_0

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$



a b c

FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

Image Smoothing Using Filter Domain Filters: GLPF

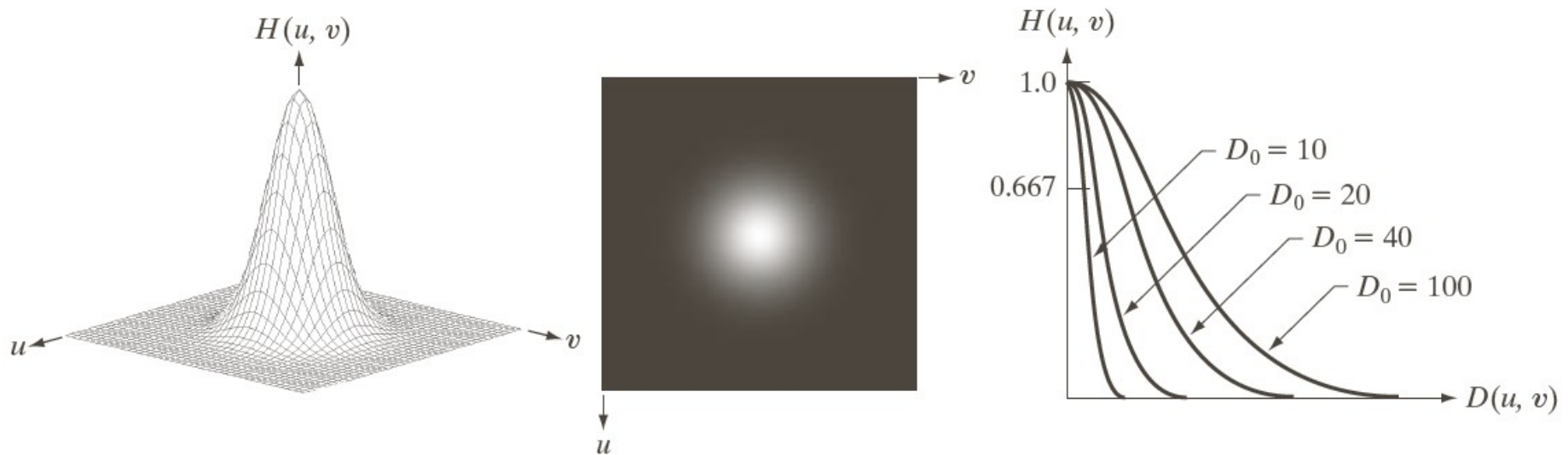
Gaussian Lowpass Filters (GLPF) in two dimensions is given

$$H(u, v) = e^{-D^2(u, v)/2\sigma^2}$$

By letting $\sigma = D_0$

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

Image Smoothing Using Filter Domain Filters: GLPF



a b c

FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0 .



FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

FIGURE 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.

Image Sharpening Using Frequency Domain Filters

A highpass filter is obtained from a given lowpass filter using

$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$

A 2-D ideal highpass filter (IHPL) is defined as

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

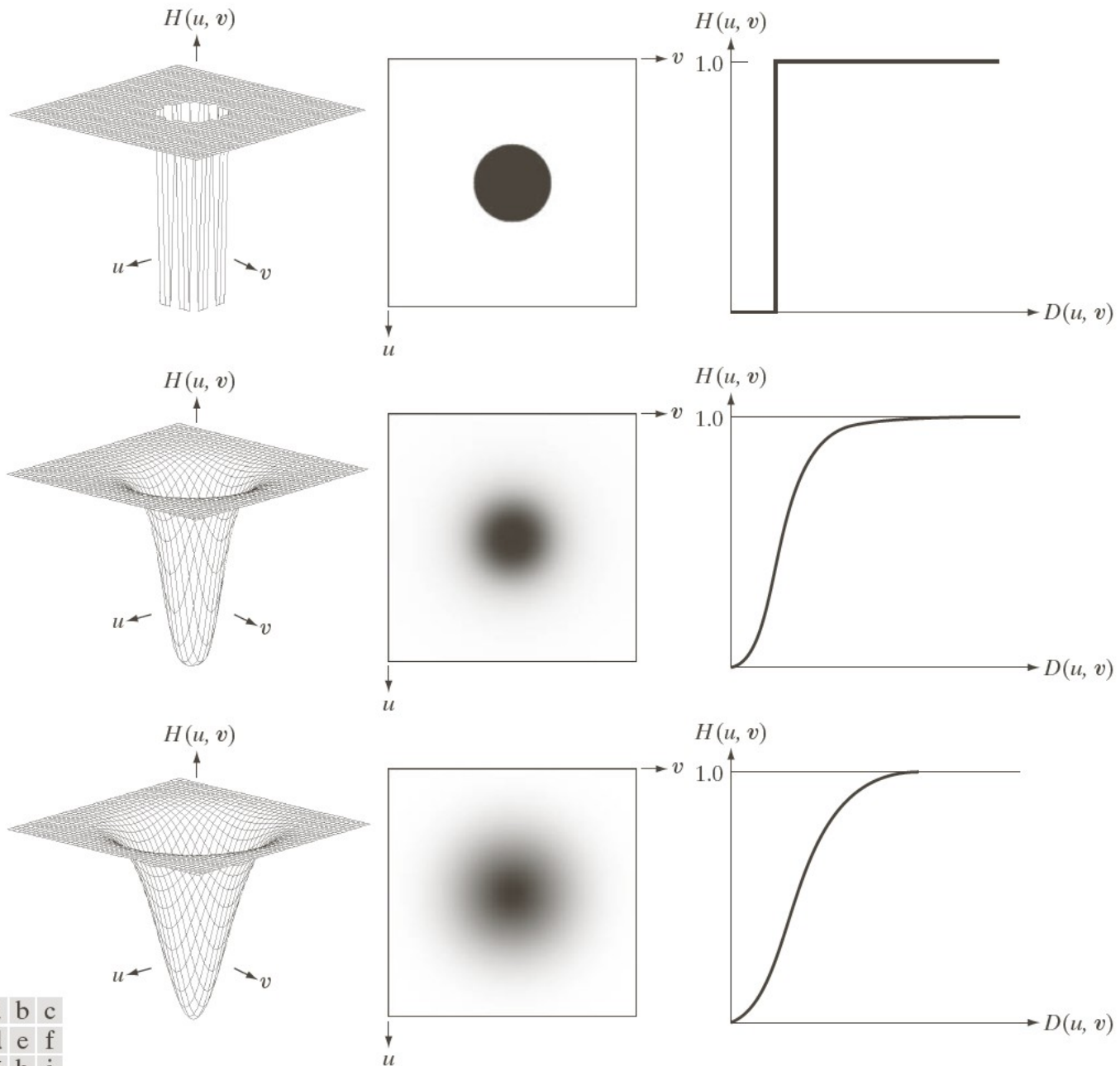
Image Sharpening Using Frequency Domain Filters

A 2-D Butterworth highpass filter (BHPL) is defined as

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

A 2-D Gaussian highpass filter (GHPL) is defined as

$$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2}$$



a	b	c
d	e	f
g	h	i

FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Selective Filtering: Bandreject and Bandpass Filters

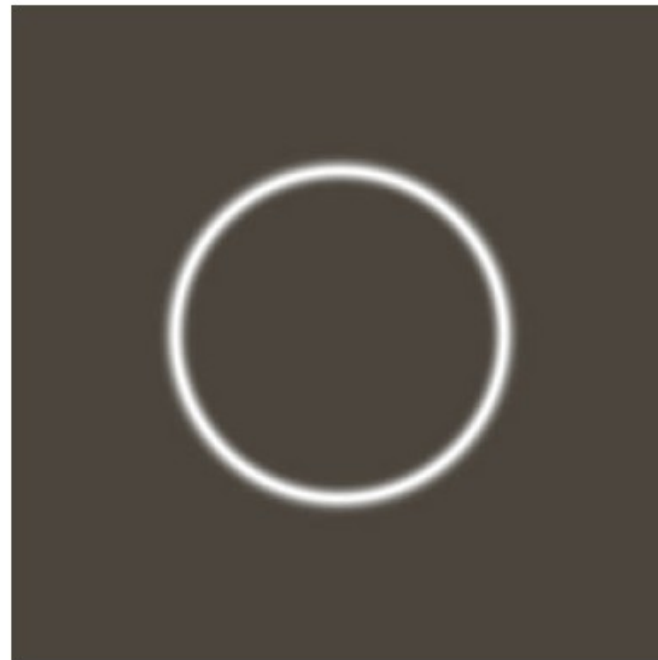
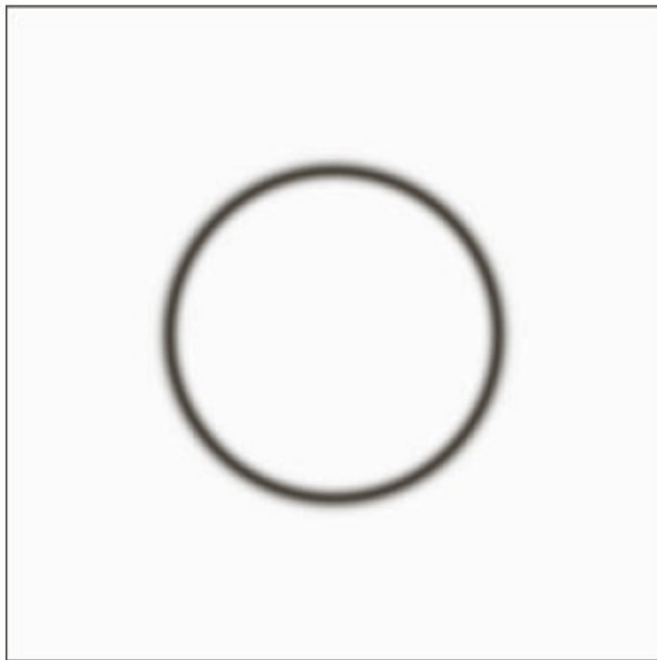
TABLE 4.6

Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2} \right]^{2n}}$	$H(u, v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW} \right]^2}$

$$H_{BP}(u, v) = 1 - H_{BR}(u, v)$$

Selective Filtering: Bandreject and Bandpass Filters



a b

FIGURE 4.63

(a) Bandreject
Gaussian filter.

(b) Corresponding
bandpass filter.

The thin black
border in (a) was
added for clarity; it
is not part of the
data.

Digital Image Processing

Image Compression



Relative Data Redundancy

- ▶ Let b and b' denote the number of bits in two representations of the same information, the relative data redundancy R is

$$R = 1 - 1/C$$

C is called the compression ratio, defined as

$$C = b/b'$$

e.g., $C = 10$, the corresponding relative data redundancy of the larger representation is 0.9, indicating that 90% of its data is redundant

Why do we need compression?

- ▶ Data storage
- ▶ Data transmission

How can we implement compression?

- ▶ **Coding redundancy**

Most 2-D intensity arrays contain more bits than are needed to represent the intensities

- ▶ **Spatial redundancy:** elements that are duplicated within a structure, such as pixels in a still image and bit patterns in a file. Exploiting spatial redundancy is how compression is performed.

- ▶ **Temporal redundancy:** pixels in two video frames that have the same values in the same location. Exploiting temporal redundancy is one of the primary techniques in video compression.

- ▶ **Irrelevant information**

Most 2-D intensity arrays contain information that is ignored by the human visual system

Examples of Redundancy



FIGURE 8.1 Computer generated $256 \times 256 \times 8$ bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

Coding Redundancy

TABLE 8.1
Example of
variable-length
coding.

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

The average number of bits required to represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k) = 0.25(2) + 0.47(1) + 0.24(3) + 0.03(3) = 1.81 \text{ bits}$$

$$C = \frac{8}{1.81} \approx 4.42$$

$$R = 1 - 1/4.42 = 0.774$$

Spatial and Temporal Redundancy

1. All 256 intensities are equally probable.
2. The pixels along each line are identical.
3. The intensity of each line was selected randomly.

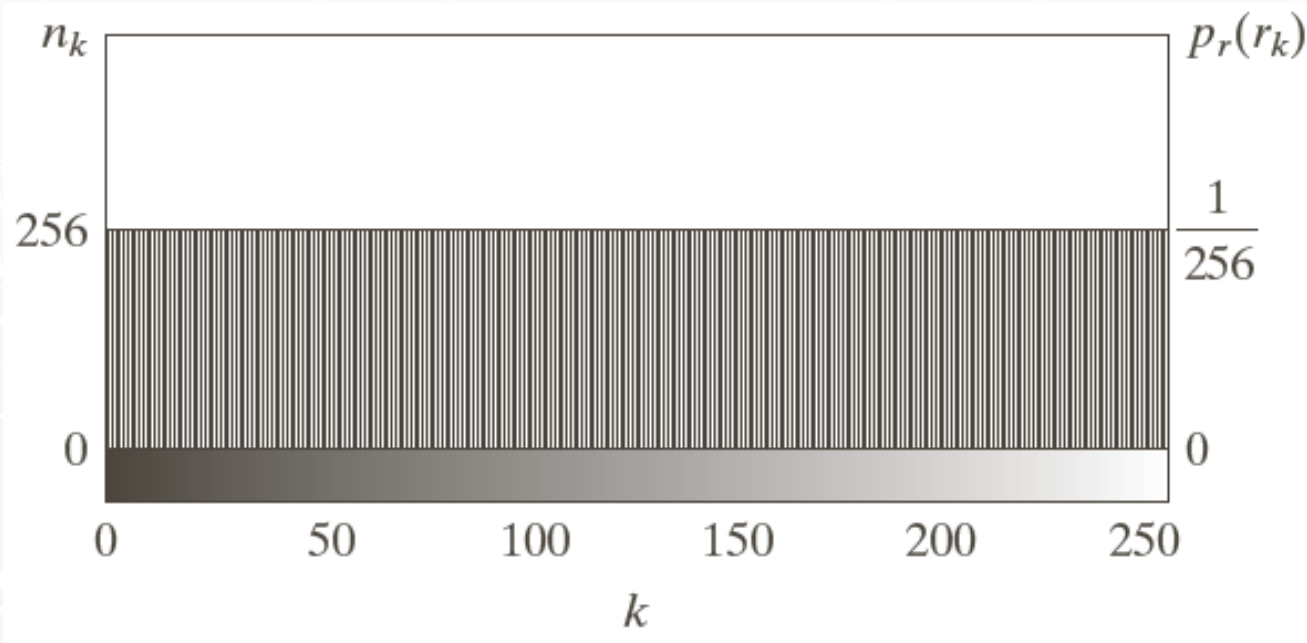


FIGURE 8.2 The intensity histogram of the image in Fig. 8.1(b).

Spatial and Temporal Redundancy

1. All 256 intensities are equally probable.
2. The pixels along each line are identical.
3. The intensity of each line was selected randomly.

Run-length pair specifies the start of a new intensity and the number of consecutive pixels that have that intensity.

Each 256-pixel line of the original representation is replaced by a single 8-bit intensity value and length 256 in the run-length representation.

The compression ratio is

$$\frac{256 \times 256 \times 8}{(256 + 256) \times 8} = 128:1$$

Measuring Image Information

Given a source of statistically independent random events from a discrete set of possible events $\{a_1, a_2, \dots, a_J\}$ with associated probabilities $\{P(a_1), P(a_2), \dots, P(a_J)\}$, the average information per source output, called the entropy of the source

$$H = -\sum_{j=1}^J P(a_j) \log P(a_j)$$

a_j is called source symbols. Because they are statistically independent, the source called *zero – memory source*.

Measuring Image Information

If an image is considered to be the output of an imaginary zero-memory "intensity source", we can use the histogram of the observed image to estimate the symbol probabilities of the source. The intensity source's entropy becomes

$$H = - \sum_{k=0}^{L-1} p_r(r_k) \log p_r(r_k)$$

$p_r(r_k)$ is the normalized histogram.

Measuring Image Information

For the fig.8.1(a),

$$H =$$

$$-[0.25 \log_2 0.25 + 0.47 \log_2 0.47 + 0.25 \log_2 0.25 + 0.03 \log_2 0.03]$$

$$\approx 1.6614 \text{ bits/pixel}$$

Image Compression Models

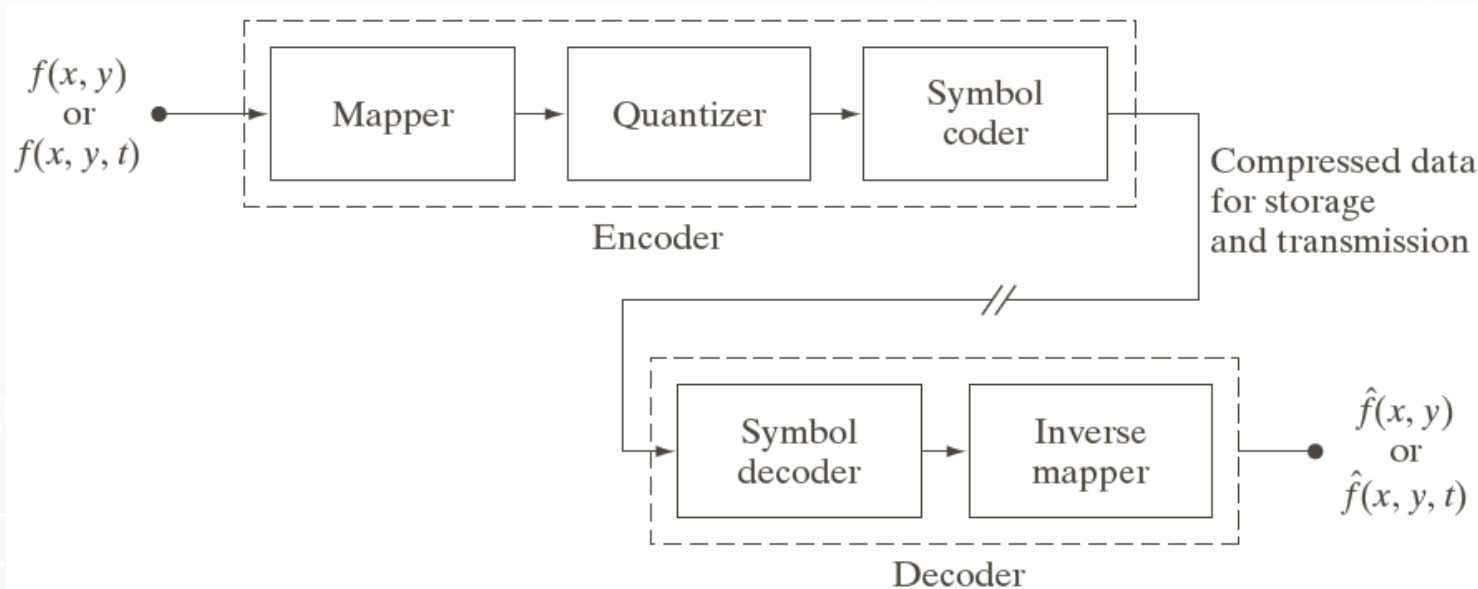


FIGURE 8.5
Functional block diagram of a general image compression system.

Mapper: Reduce spatial and temporal redundancy e.g. run-length coding

Quantizer: Reduces the quality in accord with predefined fidelity criteria (irreversible)

Symbol coder: Generates a fixed or variable length code to represent quantizer output.

Decoder, Inverse of Encoder's Mapper and coder.

Image Compression Standards

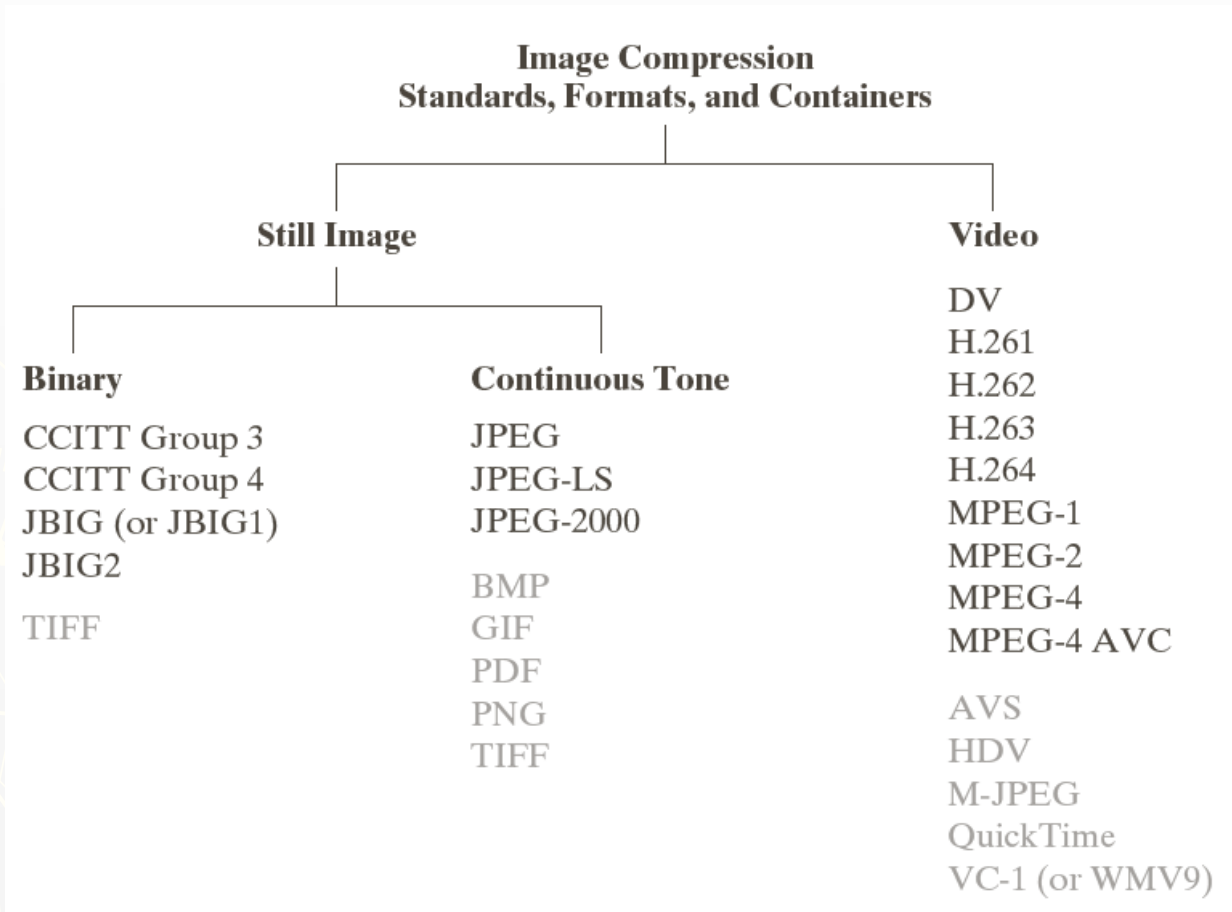


FIGURE 8.6 Some popular image compression standards, file formats, and containers. Internationally sanctioned entries are shown in black; all others are grayed.

Name	Organization	Description
<i>Bi-Level Still Images</i>		
CCITT Group 3	ITU-T	Designed as a facsimile (FAX) method for transmitting binary documents over telephone lines. Supports 1-D and 2-D run-length [8.2.5] and Huffman [8.2.1] coding.
CCITT Group 4	ITU-T	A simplified and streamlined version of the CCITT Group 3 standard supporting 2-D run-length coding only.
JBIG or JBIG1	ISO/IEC/ ITU-T	A <i>Joint Bi-level Image Experts Group</i> standard for progressive, lossless compression of bi-level images. Continuous-tone images of up to 6 bits/pixel can be coded on a bit-plane basis [8.2.7]. Context sensitive arithmetic coding [8.2.3] is used and an initial low resolution version of the image can be gradually enhanced with additional compressed data.
JBIG2	ISO/IEC/ ITU-T	A follow-on to JBIG1 for bi-level images in desktop, Internet, and FAX applications. The compression method used is content based, with dictionary based methods [8.2.6] for text and halftone regions, and Huffman [8.2.1] or arithmetic coding [8.2.3] for other image content. It can be lossy or lossless.
<i>Continuous-Tone Still Images</i>		
JPEG	ISO/IEC/ ITU-T	A <i>Joint Photographic Experts Group</i> standard for images of photographic quality. Its lossy <i>baseline coding system</i> (most commonly implemented) uses quantized discrete cosine transforms (DCT) on 8×8 image blocks [8.2.8], Huffman [8.2.1], and run-length [8.2.5] coding. It is one of the most popular methods for compressing images on the Internet.
JPEG-LS	ISO/IEC/ ITU-T	A lossless to near-lossless standard for continuous tone images based on adaptive prediction [8.2.9], context modeling [8.2.3], and Golomb coding [8.2.2].
JPEG-2000	ISO/IEC/ ITU-T	A follow-on to JPEG for increased compression of photographic quality images. Arithmetic coding [8.2.3] and quantized discrete wavelet transforms (DWT) [8.2.10] are used. The compression can be lossy or lossless.

TABLE 8.3
Internationally sanctioned image compression standards. The numbers in brackets refer to sections in this chapter.

(Continues)

TABLE 8.3
(Continued)

Name	Organization	Description
<i>Video</i>		
DV	IEC	<i>Digital Video</i> . A video standard tailored to home and semiprofessional video production applications and equipment—like electronic news gathering and camcorders. Frames are compressed independently for uncomplicated editing using a DCT-based approach [8.2.8] similar to JPEG.
H.261	ITU-T	A two-way videoconferencing standard for ISDN (<i>integrated services digital network</i>) lines. It supports non-interlaced 352×288 and 176×144 resolution images, called CIF (<i>Common Intermediate Format</i>) and QCIF (<i>Quarter CIF</i>), respectively. A DCT-based compression approach [8.2.8] similar to JPEG is used, with frame-to-frame prediction differencing [8.2.9] to reduce temporal redundancy. A block-based technique is used to compensate for motion between frames.
H.262	ITU-T	See MPEG-2 below.
H.263	ITU-T	An enhanced version of H.261 designed for ordinary telephone modems (i.e., 28.8 Kb/s) with additional resolutions: SQCIF (<i>Sub-Quarter CIF</i> 128×96), 4CIF (704×576), and 16CIF (1408×512).
H.264	ITU-T	An extension of H.261–H.263 for videoconferencing, Internet streaming, and television broadcasting. It supports prediction differences within frames [8.2.9], variable block size integer transforms (rather than the DCT), and context adaptive arithmetic coding [8.2.3].
MPEG-1	ISO/IEC	A <i>Motion Pictures Expert Group</i> standard for CD-ROM applications with non-interlaced video at up to 1.5 Mb/s. It is similar to H.261 but frame predictions can be based on the previous frame, next frame, or an interpolation of both. It is supported by almost all computers and DVD players.
MPEG-2	ISO/IEC	An extension of MPEG-1 designed for DVDs with transfer rates to 15 Mb/s. Supports interlaced video and HDTV. It is the most successful video standard to date.
MPEG-4	ISO/IEC	An extension of MPEG-2 that supports variable block sizes and prediction differencing [8.2.9] within frames.
MPEG-4 AVC	ISO/IEC	MPEG-4 Part 10 <i>Advanced Video Coding</i> (AVC). Identical to H.264 above.

Some Basic Compression Methods: Huffman Coding

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	
a_1	0.1	0.1	0.2	0.3	0.4
a_4	0.1	0.1			
a_3	0.06	0.1	0.1	0.3	0.4
a_5	0.04				

FIGURE 8.7
Huffman source
reductions.

Some Basic Compression Methods:

Huffman Coding

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

FIGURE 8.8
Huffman code
assignment
procedure.

The average length of this code is

$$\begin{aligned}
 L_{avg} &= 0.4 * 1 + 0.3 * 2 + 0.1 * 3 + 0.1 * 4 + 0.06 * 5 + 0.04 * 5 \\
 &= 2.2 \text{ bits/pixel}
 \end{aligned}$$

The background of the slide is a light gray topographic map with yellow contour lines. In the bottom-left corner, there is a yellow compass rose with a black needle pointing towards the top-left. The compass rose has labels for 'N' (North), 'NE' (Northeast), 'SE' (Southeast), and 'SW' (Southwest).

Digital Image Processing

Image Segmentation

Fundamentals

- Let R represent the entire spatial region occupied by an image. Image segmentation is a process that partitions R into n sub-regions, $R_1, R_2, \dots, R_n$, such that

(a) $\bigcup_{i=1}^n R_i = R.$

(b) R_i is a connected set. $i = 1, 2, \dots, n.$

(c) $R_i \cap R_j = \Phi.$

(d) $\Box Q(R_i) = \text{TRUE}$ for $i = 1, 2, \dots, n.$

(e) $\Box Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions
 R_i and $R_j.$

Q is a compound stmt like “average intensity of the region is less than m and standard deviation of their intensity is Greater than s ”.

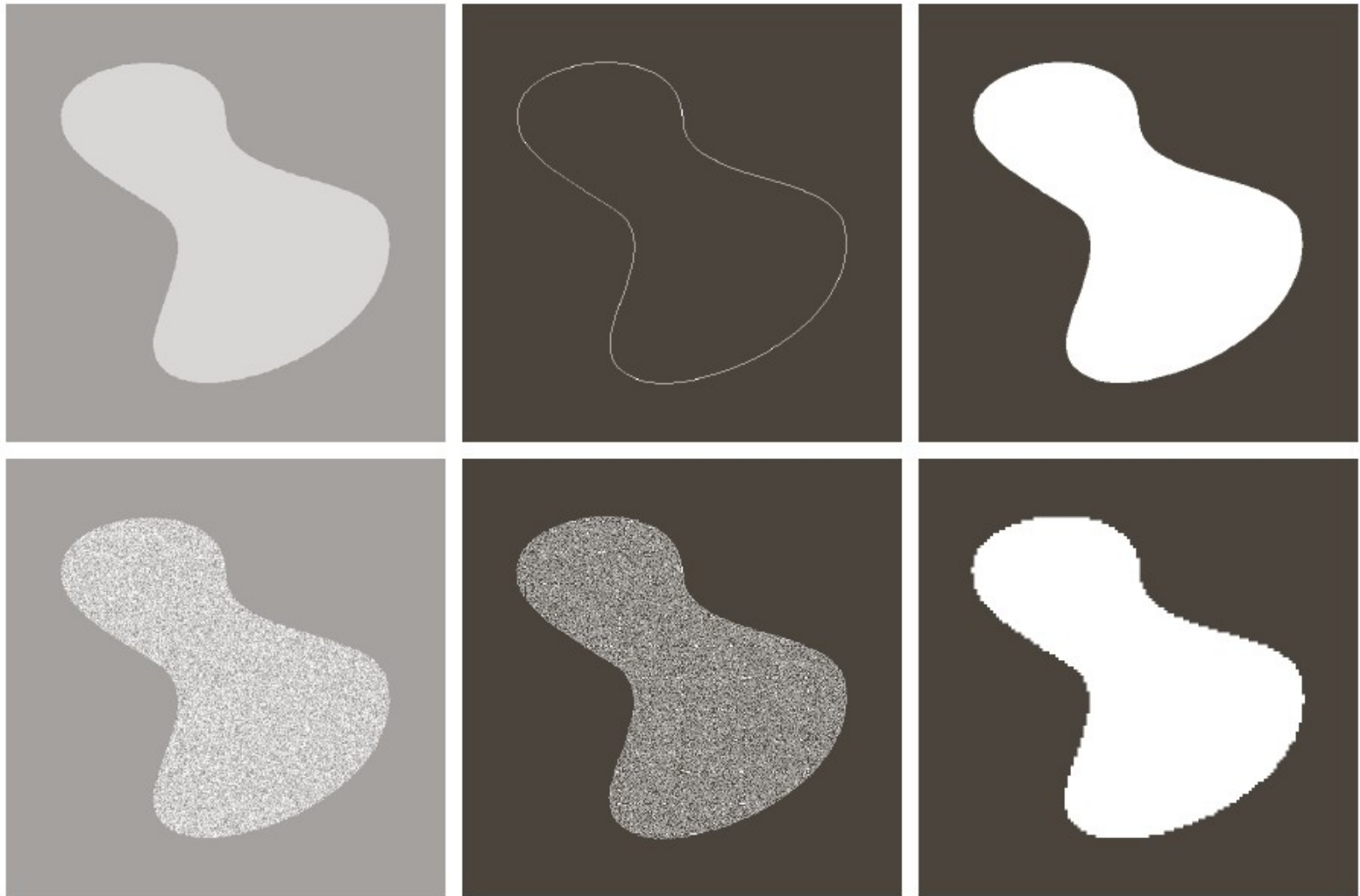


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

a	b	c
d	e	f

Background

- ▶ First-order derivative

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- ▶ Second-order derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Characteristics of First and Second Order Derivatives

- ▶ First-order derivatives generally produce thicker edges in image
- ▶ Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- ▶ The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light.
- ▶ This isotropic (rotation invariant) filter

Detection of Points in 2-D space

► The Laplacian

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

$$g(x, y) = \begin{cases} 1 & \text{if } |R(x, y)| \geq T \\ 0 & \text{otherwise} \end{cases} \quad R = \sum_{k=1}^9 w_k z_k$$

Line Detection :Second derivatives to result in a stronger response and to produce thinner lines than first derivatives

Detecting Line in Specified Directions

-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
Horizontal			+45°			Vertical			-45°		

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- ▶ Let R_1 , R_2 , R_3 , and R_4 denote the responses of the masks. If, at a given point in the image, $|R_k| > |R_j|$, for all $j \neq k$, that point is said to be more likely associated with a line in the direction of mask k .

Edge Detection

- ▶ Edges are pixels where the brightness function changes abruptly
- ▶ Edge models (Step edge, Ramp edge, roof edge model)



FIGURE 10.8
From left to right,
models (ideal
representations) of
a step, a ramp, and
a roof edge, and
their corresponding
intensity profiles.

Basic Edge Detection by Using First-Order Derivative

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of ∇f

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

The direction of ∇f

$$\alpha(x, y) = \tan^{-1} \left[\frac{g_x}{g_y} \right]$$

The direction of the edge

$$\phi = \alpha - 90^\circ$$

Basic Edge Detection by Using First-Order Derivative

$$\text{Edge normal: } \nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\text{Edge unit normal: } \nabla f / \text{mag}(\nabla f)$$

In practice, sometimes the magnitude is approximated by

$$\text{mag}(\nabla f) = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad \text{or} \quad \text{mag}(\nabla f) = \max \left(\left| \frac{\partial f}{\partial x} \right|, \left| \frac{\partial f}{\partial y} \right| \right)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0	0	-1
0	1	1	0

Roberts

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

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

Sobel

a	
b	c
d	e
f	g

FIGURE 10.14

A 3×3 region of an image (the z 's are intensity values) and various masks used to compute the gradient at the point labeled z_5 .

0	1	1	-1	-1	0
-1	0	1	-1	0	1
-1	-1	0	0	1	1

Prewitt

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

Sobel



FIGURE 10.15
Prewitt and Sobel
masks for
detecting diagonal
edges.



a	b
c	d

FIGURE 10.16
 (a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$.
 (b) $|g_x|$, the component of the gradient in the x -direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.
 (c) $|g_y|$, obtained using the mask in Fig. 10.14(g).
 (d) The gradient image, $|g_x| + |g_y|$.



a	b
c	d

FIGURE 10.18
Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.



Edge Linking and Boundary Detection

- ▶ Edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries
- ▶ Three approaches to edge linking
 - Local processing**
 - Regional processing
 - Global processing

Local Processing

- ▶ Analyze the characteristics of pixels in a small neighborhood about every point (x,y) that has been declared an edge point
- ▶ All points that similar according to predefined criteria are linked, forming an edge of pixels.

Establishing similarity: (1) the strength (magnitude) and (2) the direction of the gradient vector.

A pixel with coordinates (s,t) in S_{xy} is linked to the pixel at (x,y) if both magnitude and direction criteria are satisfied.

Local Processing

Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image. An edge pixel with coordinate (s, t) in S_{xy} is similar in *magnitude* to the pixel at (x, y) if

$$|M(s, t) - M(x, y)| \leq E$$

An edge pixel with coordinate (s, t) in S_{xy} is similar in *angle* to the pixel at (x, y) if

$$|\alpha(s, t) - \alpha(x, y)| \leq A$$

Local Processing: Steps (1)

1. Compute the gradient magnitude and angle arrays, $M(x,y)$ and $\alpha(x,y)$, of the input image $f(x,y)$
2. Form a binary image, g , whose value at any pair of coordinates (x,y) is given by

$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_M \text{ and } \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

T_M : threshold A : specified angle direction

T_A : a "band" of acceptable directions about A

Local Processing: Steps (2)

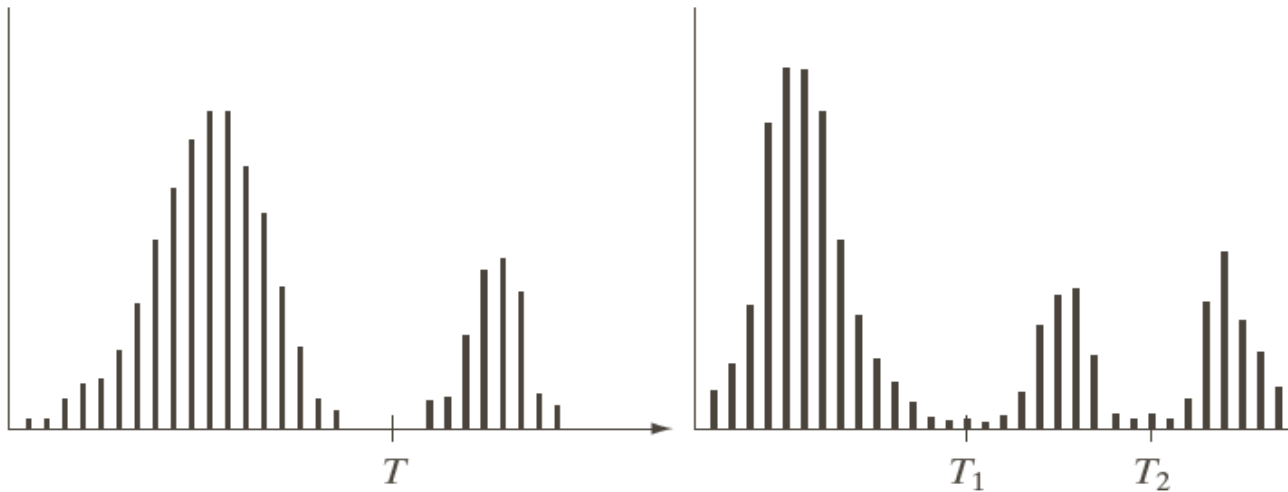
3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K .
4. To detect gaps in any other direction, rotate g by this angle and apply the horizontal scanning procedure in step 3.



Thresholding

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \quad (\text{object point}) \\ 0 & \text{if } f(x, y) \leq T \quad (\text{background point}) \end{cases}$$

T : global thresholding



a b

FIGURE 10.35
Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

Multiple thresholding

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases}$$

The Role of Noise in Image Thresholding

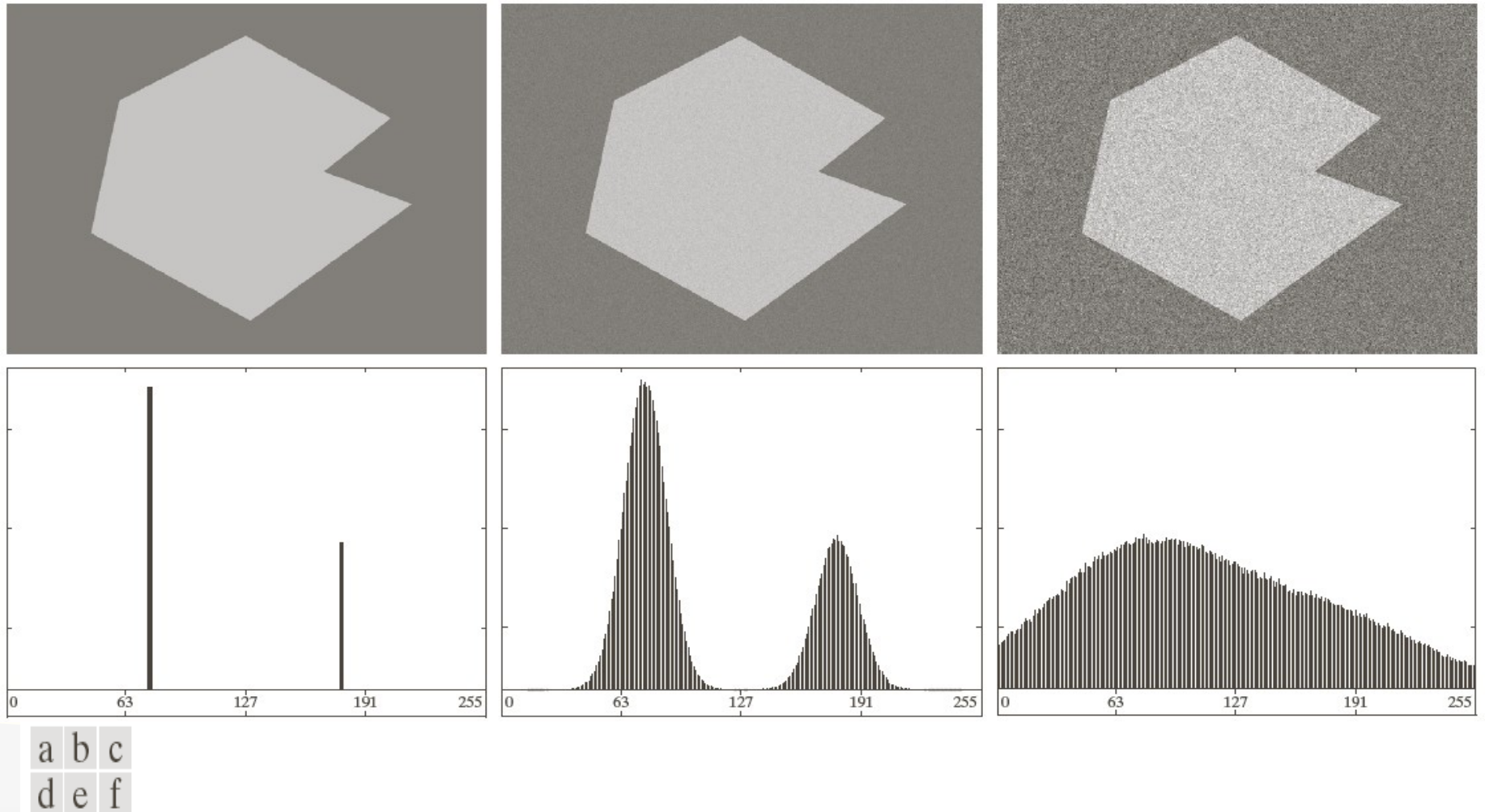


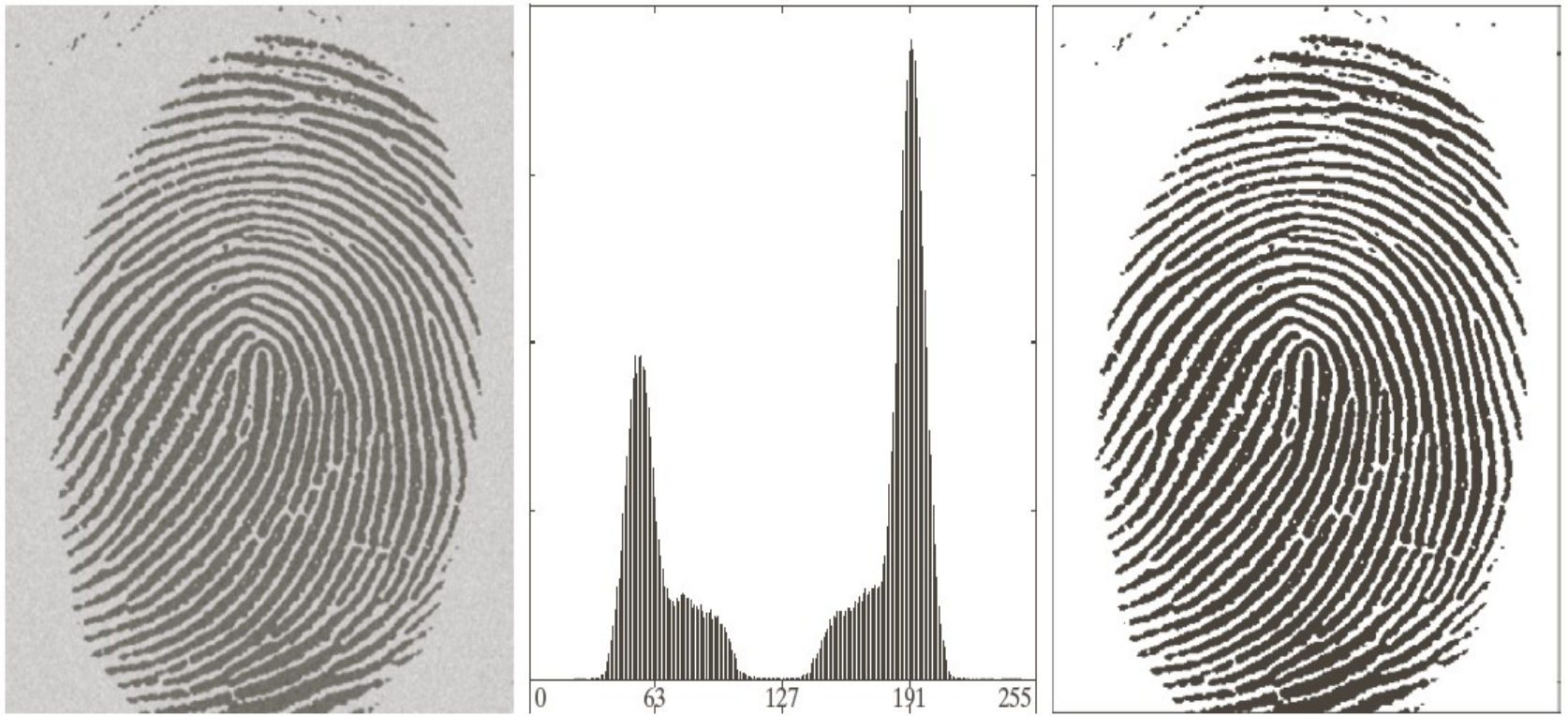
FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Basic Global Thresholding

1. Select an initial estimate for the global threshold, T .
2. Segment the image using T . It will produce two groups of pixels: $G1$ consisting of all pixels with intensity values $> T$ and $G2$ consisting of pixels with values $\leq T$.
3. Compute the average intensity values $m1$ and $m2$ for the pixels in $G1$ and $G2$, respectively.
4. Compute a new threshold value.

$$T = \frac{1}{2}(m1 + m2)$$

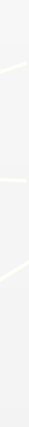
5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT .



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Using Image Smoothing to Improve Global Thresholding



a	b	c
d	e	f

FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Variable Thresholding Based on Local Image Properties

Let σ_{xy} and m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood S_{xy} , centered at coordinates (x, y) in an image. The local thresholds,

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

If the background is nearly constant,

$$T_{xy} = a\sigma_{xy} + bm$$

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T_{xy} \\ 0 & \text{if } f(x, y) \leq T_{xy} \end{cases}$$



a	b
c	d

FIGURE 10.48

(a) Image from Fig. 10.43.

(b) Image segmented using the dual thresholding approach discussed in Section 10.3.6.

(c) Image of local standard deviations.

(d) Result obtained using local thresholding.

$$\begin{aligned} a &= 30 \\ b &= 1.5 \\ m_{xy} &= m_G \end{aligned}$$

Region-Based Segmentation

► Region Growing

1. Region growing is a procedure that groups pixels or subregions into larger regions.
2. The simplest of these approaches is *pixel aggregation*, which starts with a set of “seed” points and from these grows regions by appending to each seed points those **neighboring pixels** that have **similar properties** (such as gray level, texture, color, shape).
3. Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

Region-Based Segmentation

Example: Region Growing based on 8-connectivity

$f(x, y)$: input image array

$S(x, y)$: seed array containing 1s (seeds) and 0s

$Q(x, y)$: predicate

$$Q = \begin{cases} \text{TRUE} & \text{if the absolute difference of the intensities} \\ & \text{between the seed and the pixel at } (x, y) \text{ is } \leq T \\ \text{FALSE} & \text{otherwise} \end{cases}$$

Region Growing based on 8-connectivity

1. Find all connected components in $S(x, y)$ and erode each connected components to one pixel; label all such pixels found as 1. All other pixels in S are labeled 0.
2. Form an image f_Q such that, at a pair of coordinates (x, y) , let $f_Q(x, y) = 1$ if the Q is satisfied otherwise $f_Q(x, y) = 0$.
3. Let g be an image formed by appending to each seed point in S all the 1-value points in f_Q that are 8-connected to that seed point.
4. Label each connected component in g with a different region label. This is the segmented image obtained by region growing.

Example

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

4-connectivity

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

8-connectivity

Region Splitting and Merging

R : entire image R_i :entire image Q : predicate

1. For any region R_i , If $Q(R_i) = \text{FALSE}$,
we divide the image R_i into quadrants.
2. When no further splitting is possible,
merge any adjacent regions R_j and R_k
for which $Q(R_j \cup R_k) = \text{TRUE}$.
3. Stop when no further merging is possible.

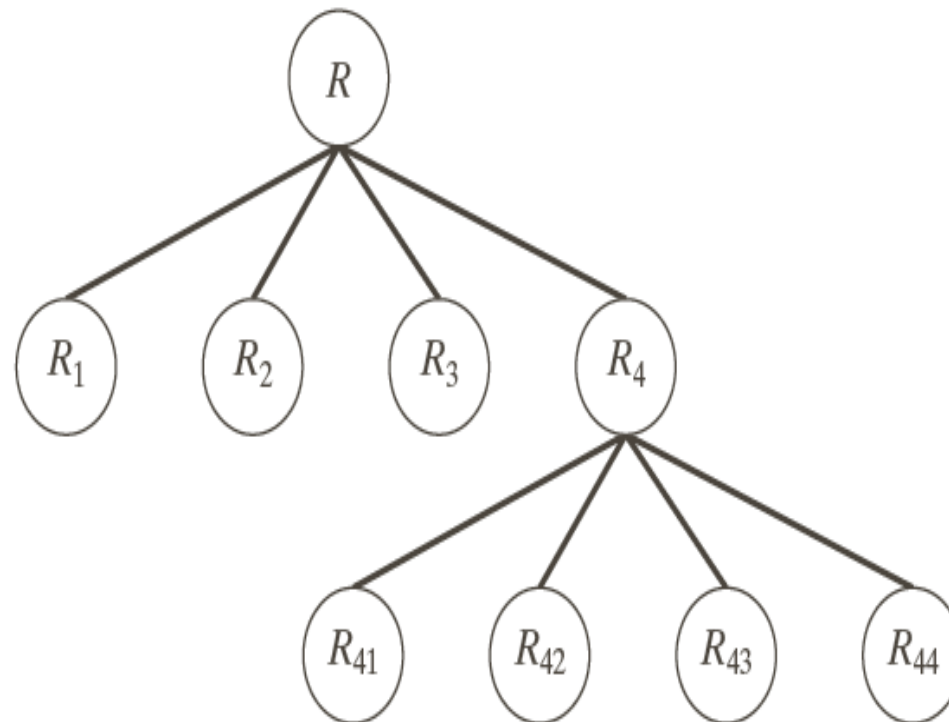
$$Q = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ and } 0 < m < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

a b

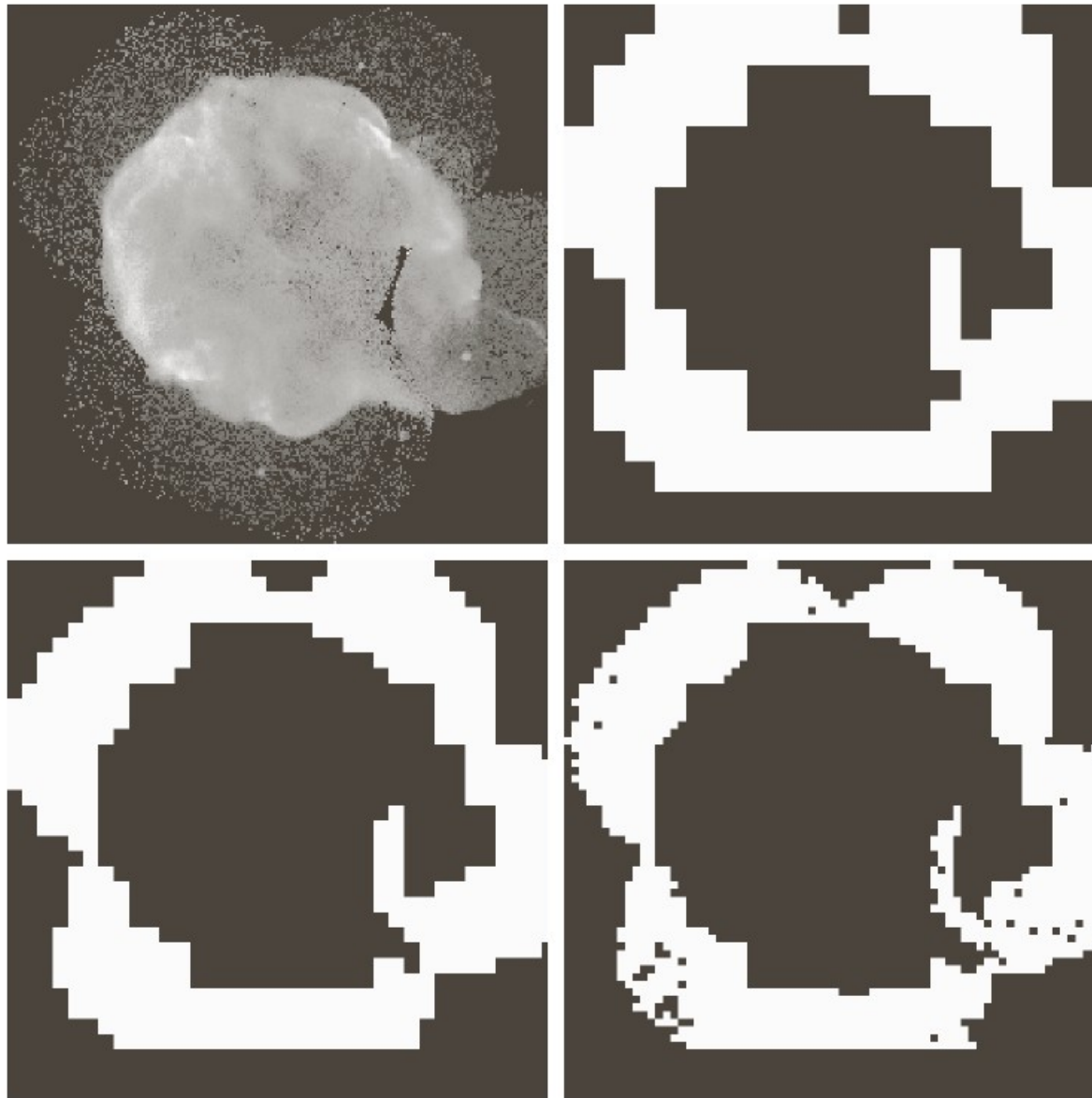
FIGURE 10.52

(a) Partitioned image.

(b) Corresponding quadtree. R represents the entire image region.



R_1	R_2	
R_3	R_{41}	R_{42}
	R_{43}	R_{44}



a	b
c	d

FIGURE 10.53
 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32 , 16×16 , and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Thank You

