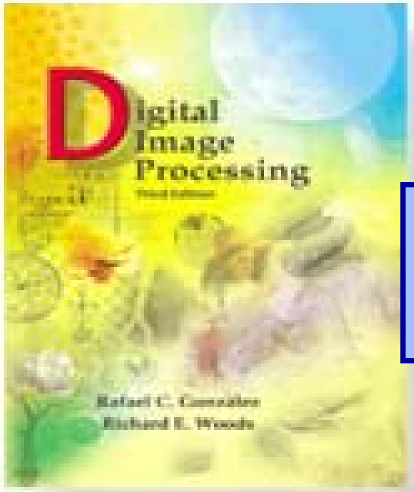


Digital Image Processing

CHAPTER 1: INTRODUCTION

WHAT IS A DIGITAL IMAGE?

- A digital image can be defined to be a function of two real variables, $f(x,y)$, where x and y are spatial coordinates, and the amplitude f at a given pair of coordinates is called the intensity of the image at that point.
- Every digital image is composed of a finite number of elements, called pixels, each with a particular location and a finite value.

WHAT IS DIGITAL IMAGE PROCESSING?

- **Digital image processing**: analysis, manipulation, storage, and display of graphical images from sources such as photographs, drawings, and video.
- **The process is composed of 3 steps**
  - **Input**: the image is either directly captured with a digital camera or an analog image is digitized in a scanner to obtain the binary data a computer can understand.
  - **Processing**: may include any process including image enhancement, restoration, compression, segmentation, or watermarking as well as extraction of image features.
  - **Output**: consists of the display/printing of the processed image, or the features extracted from the input image.

CHAPTER 1: INTRODUCTION

- Interest in DIP methods stem from 2 principal applications:
  - Improvement of pictorial information for human interpretation
  - Processing of image data for storage, transmission, and representation for autonomous machine perception.

BARTLANE CABLE PICTURE TRANSMISSION SYSTEM

- **Early 1920s: Bartlane cable picture transmission system**
  - **Reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours.**
  - **Pictures were coded for cable transmission and then reconstructed at the receiving end on a telegraph printer fitted with type faces simulating a halftone pattern.**
  - **The early Bartlane systems were capable of coding images in five distinct brightness levels. This was increased to fifteen levels in 1929.**

FIGURE 1.1 A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces. (McFarlane.)

OTHER HISTORICAL DEVELOPMENTS

- **1964: Pictures of the moon transmitted by Ranger 7 were processed by a computer at the Jet Propulsion Lab to correct various types of image distortion inherent in the on-board TV camera.**
- **Experience with Ranger 7 led to an improvement in image enhancement and restoration in other projects.**
  - **Surveyor missions to the moon**
  - **Mariner series of flyby missions to Mars**
  - **Apollo manned flights to the moon**

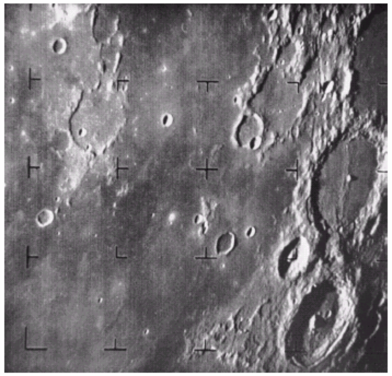


FIGURE 1.4 The first picture of the moon by a U.S. spacecraft. *Ranger 7* took this image on July 31, 1964 at 9:09 A.M. EDT, about 17 minutes before impacting the lunar surface. (Courtesy of NASA.)

APPLICATIONS OF IMAGE PROCESSING

- Early 1970’s: Invention of computerized axial tomography
- Applications of image processing
  - For human interpretation
    - Geography
    - Archeology.
    - Physics
    - Astronomy
    - Biology
    - Nuclear medicine
    - Law enforcement
    - Defense
  - For machine perception
    - Automatic character recognition
    - Industrial machine vision for product assembly and inspection
    - Military recognizance
    - Automatic processing of fingerprints
    - Screening of X-rays and blood samples
    - Machine processing of aerial and satellite imagery for weather prediction and environment assessment

ENERGY SOURCES FOR IMAGES

- Electromagnetic energy spectrum: Principal source of energy
  - The full range of frequencies, from radio waves to gamma rays, that characterizes light.
  - Gamma rays: highest energy, shortest wavelength
  - Radio waves: lowest energy, longest wavelength
- Electromagnetic radiation can be described in terms of a stream of photons, each traveling in a wave-like pattern, moving at the speed of light and carrying some amount of energy.
- The amount of energy a photon has makes it sometimes behave more like a wave and sometimes more like a particle. This is called the "wave-particle duality" of light.
- Low energy photons (such as radio) behave more like waves, while higher energy photons (such as X-rays) behave more like particles.
- Other sources of energy: acoustic, ultrasonic, electronic

ELECTROMAGNETIC SPECTRUM

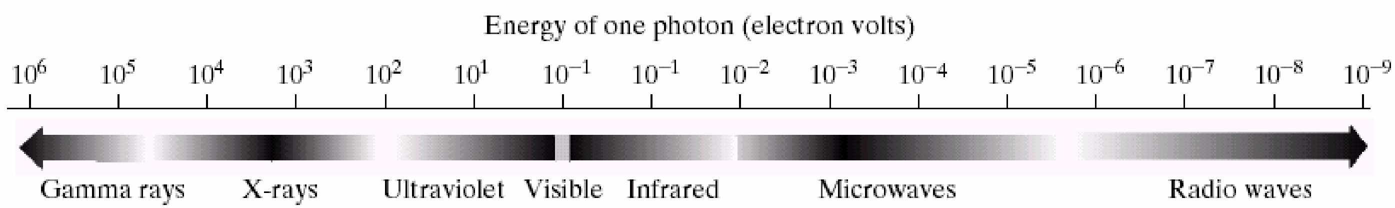
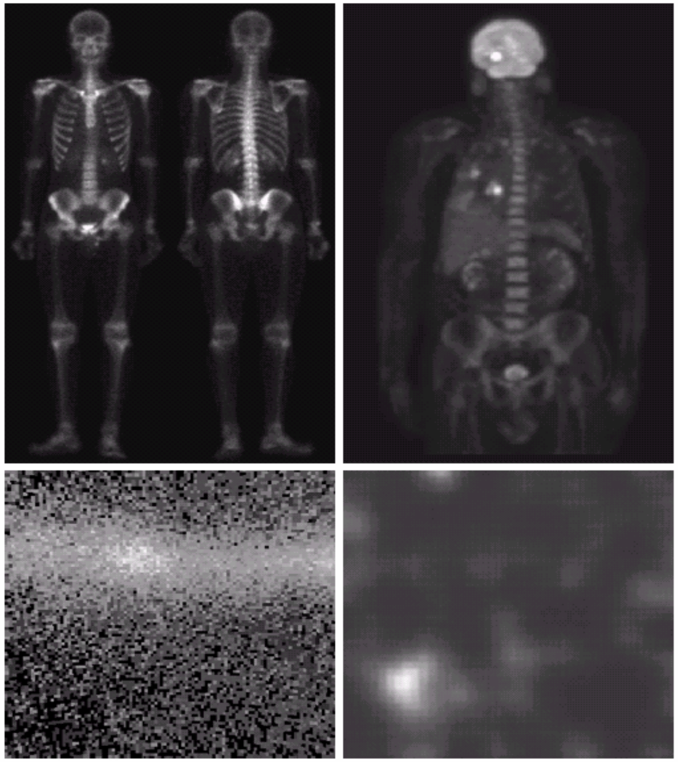


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

GAMMA-RAY

Major uses: Nuclear medicine & astronomical observations

Bone scan: locates sites of bone pathology



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: another major modality of nuclear imaging

One sample of a sequence that constitutes a 3d rendition of the patient.

A **star** in the constellation Cygnus exploded about 15K years ago.

The superheated stationary gas cloud (Cygnus Loop) glows in a spectacular array of colors.

Gamma radiation from a **valve** in a nuclear reactor

X-RAY

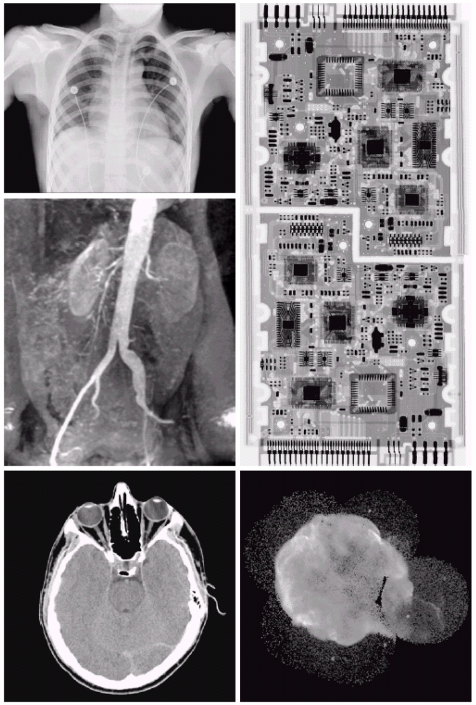
X-rays are among the oldest sources of EM radiation used for imaging.

Major uses: Medical diagnostics, astronomy, angiography, CAT scans

Chest X-ray

Aortic angiogram

Head CAT slice



X-ray image of an electronic circuit board

Cygnus Loop imaged in the X-ray band

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.)

ULTRAVIOLET

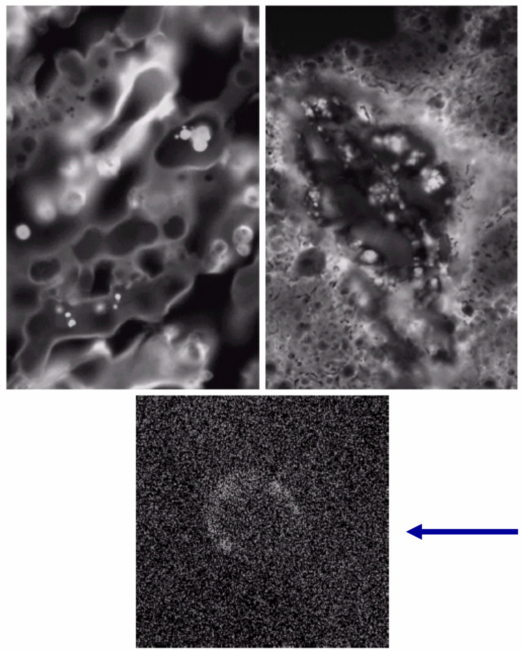
Major uses: **lithography, industrial inspection, microscopy, lasers, biological imaging, astronomical observations.**

Fluorescence is a phenomenon discovered in the middle of 19th century.

Fluorescence microscopy is a excellent method for studying materials that can be made to fluoresce (either in natural form or when treated with chemicals capable of fluorescing).

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.)



Fluorescence microscope image of normal corn

Corn infected by "smut," a disease of cereals, corn, etc.

Cygnus Loop imaged in the high-energy region of the ultraviolet band.

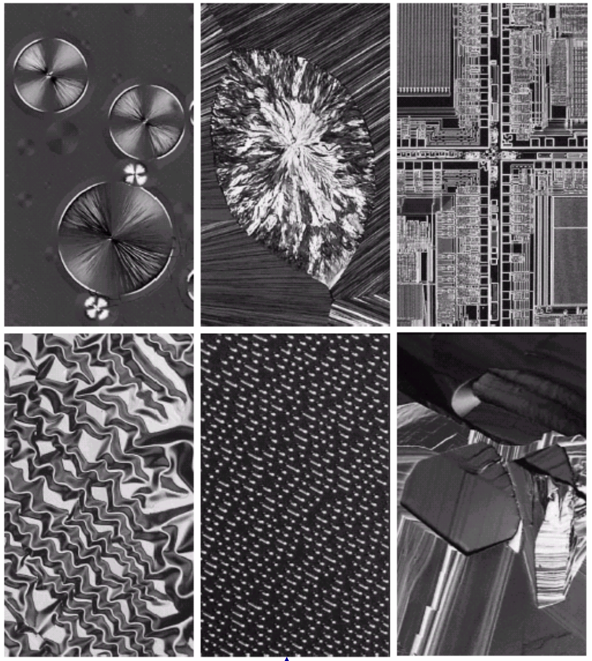
VISIBLE AND INFRARED – LIGHT MICROSCOPE

Major uses: **light microscopy, astronomy, remote sensing, industry, law enforcement.**

Cholesterol

Taxol (anticancer agent)

Microprocessor



Nickel oxide thin film

Surface of audio CD

Organic superconductor

a b c
d e f

FIGURE 1.9 Examples of light microscopy images. (a) Taxol (anticancer agent), magnified 250×. (b) Cholesterol—40×. (c) Microprocessor—60×. (d) Nickel oxide thin film—600×. (e) Surface of audio CD—1750×. (f) Organic superconductor—450×. (Images courtesy of Dr. Michael W. Davidson, Florida State University.)

VISIBLE AND INFRARED – REMOTE SENSING

Remote sensing: usually includes several bands in the visual and infrared regions.

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

LANDSAT obtains and transmits images of the Earth from space, for purposes of monitoring environmental conditions.

Multispectral imaging:

One image for each band in the above table.

The differences between visual and infrared image features are quite noticeable.

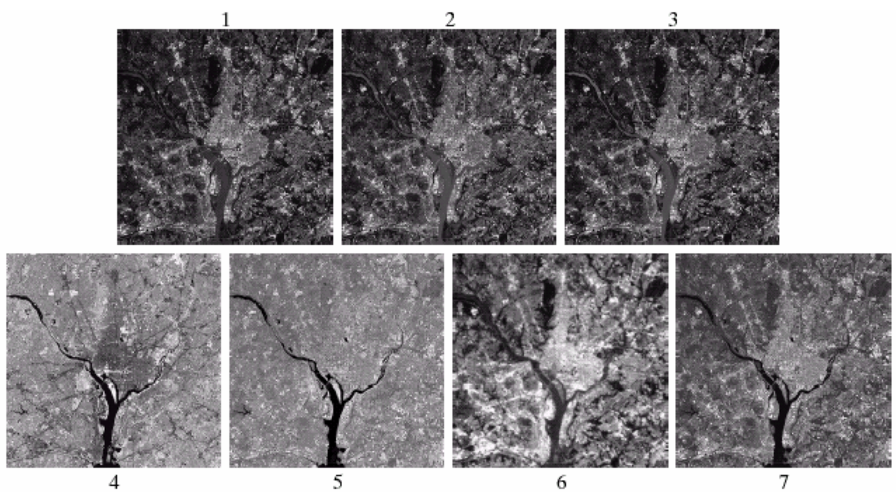


FIGURE 1.10 LANDSAT satellite images of the Washington, D.C. area. The numbers refer to the thematic bands in Table 1.1. (Images courtesy of NASA.)

VISIBLE AND INFRARED – WEATHER OBSERVATION AND PREDICTION

An image of a hurricane taken by a satellite using sensors in the visible and infrared bands.

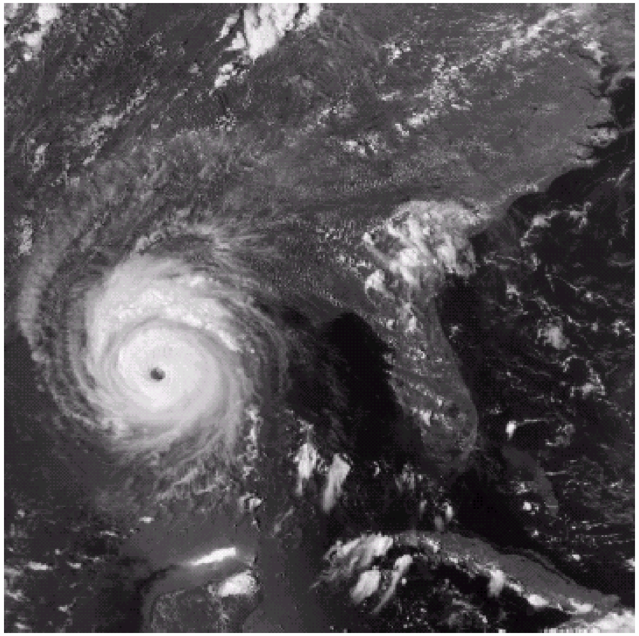


FIGURE 1.11 Multispectral image of Hurricane Andrew taken by NOAA GEOS (Geostationary Environmental Operational Satellite) sensors. (Courtesy of NOAA.)

INFRARED – HUMAN SETTLEMENTS (THE AMERICAS)

FIGURE 1.12 Infrared satellite images of the Americas. The small gray map is provided for reference. (Courtesy of NOAA.)

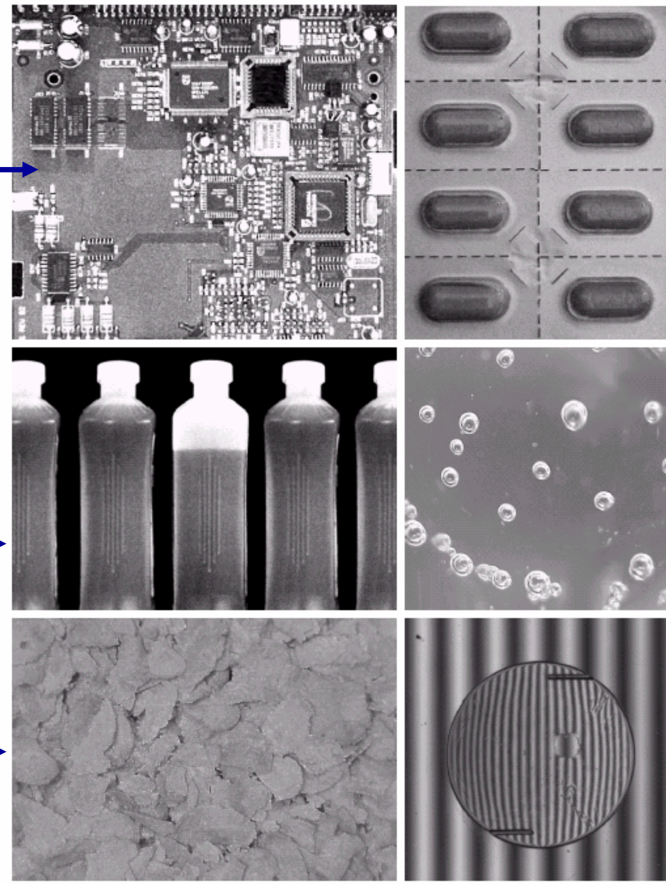
These images are part of Nighttime Lights of the World data set.

This set provides a global inventory of human settlements.

VISIBLE – AUTOMATED VISUAL INSPECTION OF MANUFACTURED GOODS

a b
c d
e f

FIGURE 1.14 Some examples of manufactured goods often checked using digital image processing. (a) A circuit board controller. (b) Packaged pills. (c) Bottles. (d) Bubbles in clear-plastic product. (e) Cereal. (f) Image of intraocular implant. (Fig. (f) courtesy of Mr. Pete Sites, Perceptics Corporation.)



A circuit board controller (the back square is a missing component)

Pill container

A bottle that is not filled up to an acceptable level.

A clear plastic part with an unacceptable number of air pockets.

A batch of cereal inspected for color and the presence of anomalies such as burned flakes.

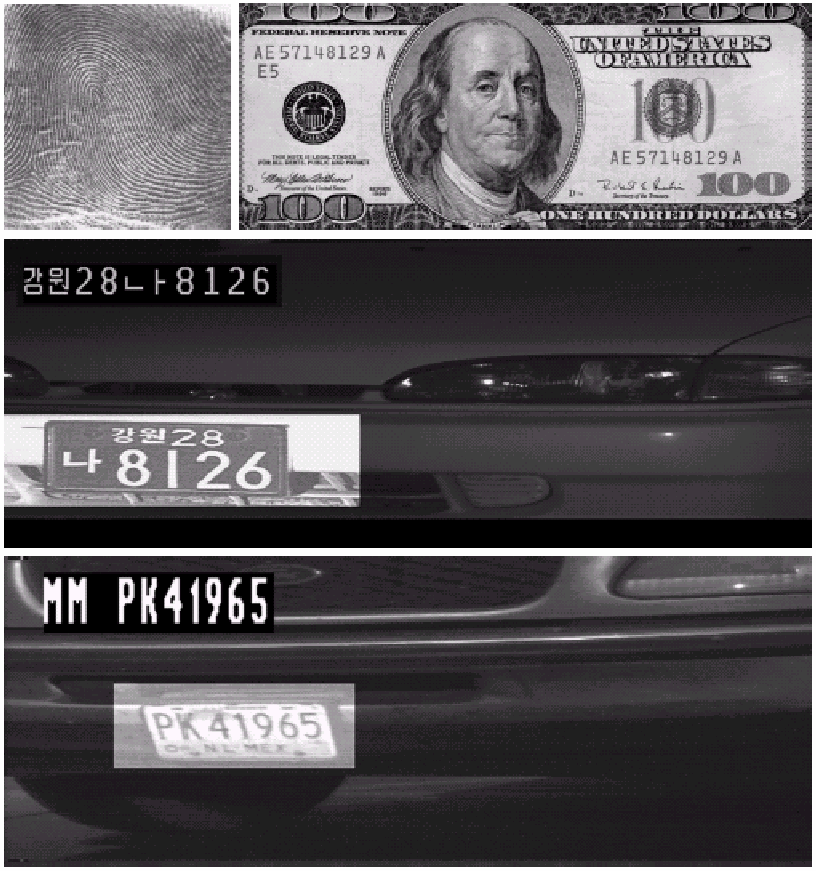
An intraocular implant (replacement lens for the human eye)

VISIBLE – OTHER APPLICATIONS

A thumb print

Paper currency

Automated license plate reading

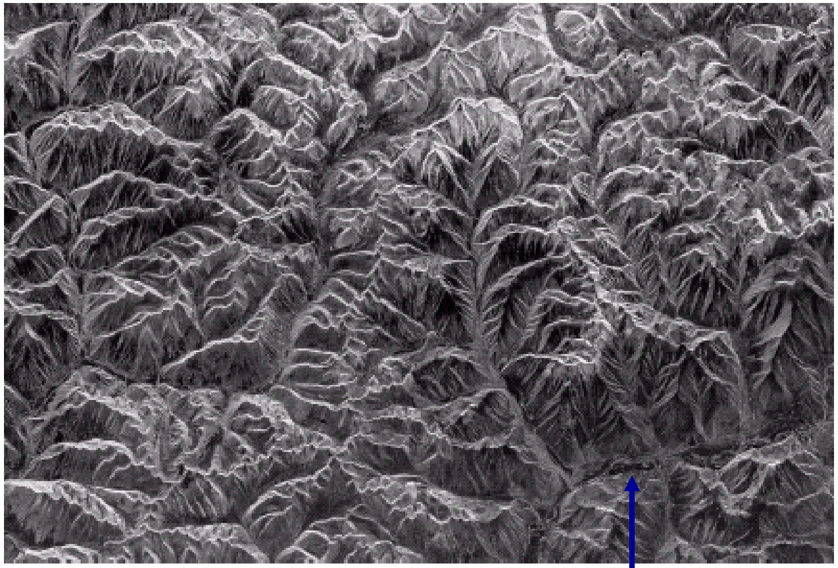


a b
c
d

FIGURE 1.15 Some additional examples of imaging in the visual spectrum. (a) Thumb print. (b) Paper currency. (c) and (d). Automated license plate reading. (Figure (a) courtesy of the National Institute of Standards and Technology. Figures (c) and (d) courtesy of Dr. Juan Herrera, Perceptics Corporation.)

MICROWAVE – IMAGING RADAR

FIGURE 1.16 Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)



Imaging radar

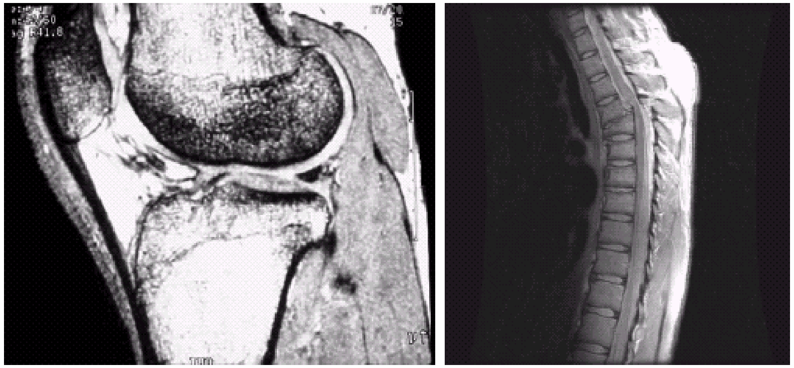
- **has the unique ability to collect data over virtually any region at any time, regardless of weather or ambient lightning conditions.**
- **Works like a flash camera with its own illumination.**
- **Uses an antenna and digital computer processing to record images.**

Note the clarity and detail of the image, unencumbered by clouds or other atmospheric conditions that normally interfere with image in the visual band!

Lhasa River

RADIO – MRI

Major uses: medicine and astronomy



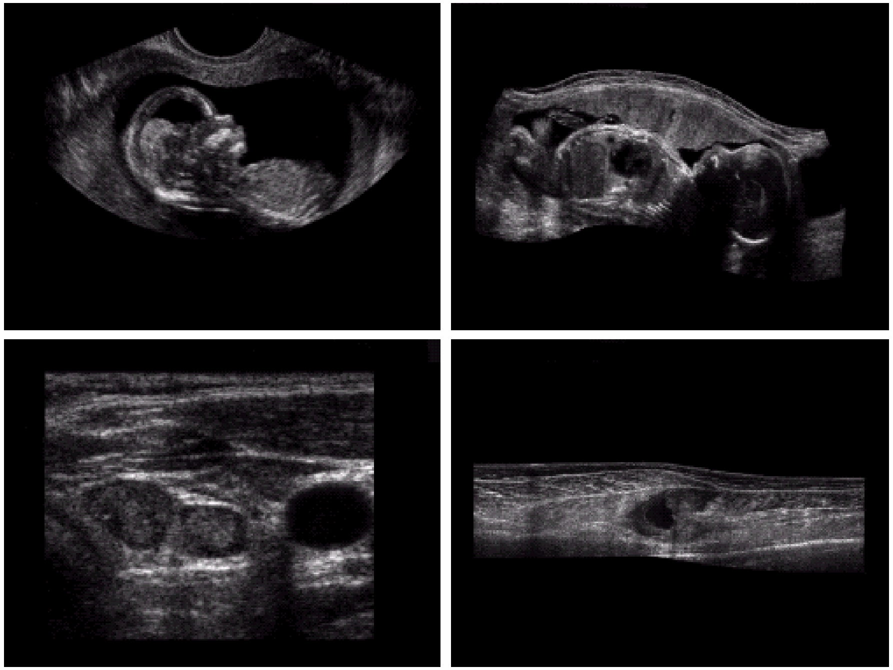
a b

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.)

Magnetic resonance imaging (MRI):

- **places a patient in a powerful magnet and passes radio waves through the body in short pulses.**
- **each pulse causes a responding pulse of radio waves to be emitted by the patient's tissues.**
- **produces a 2D picture of section of the patient.**

OTHER IMAGING MODALITIES - ULTRASOUND IMAGING

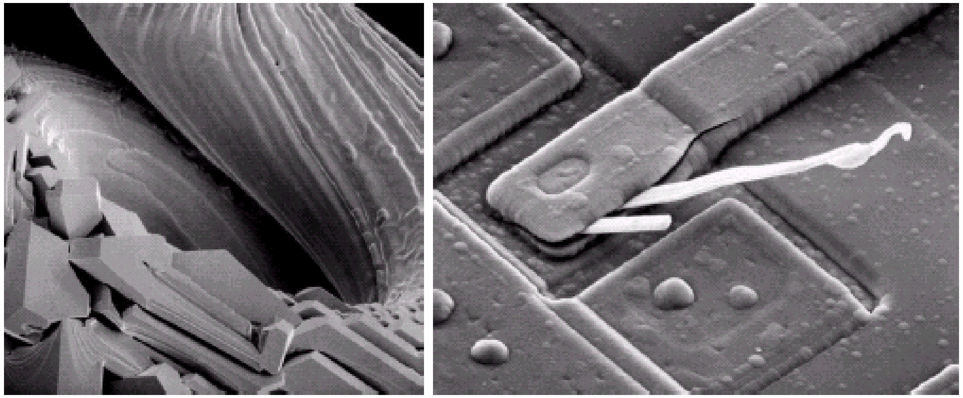


a b
c d

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.)

Ultrasound imaging: millions of HF sound pulses and echoes are sent and received each second.

OTHER IMAGING MODALITIES - ELECTRON MICROSCOPY



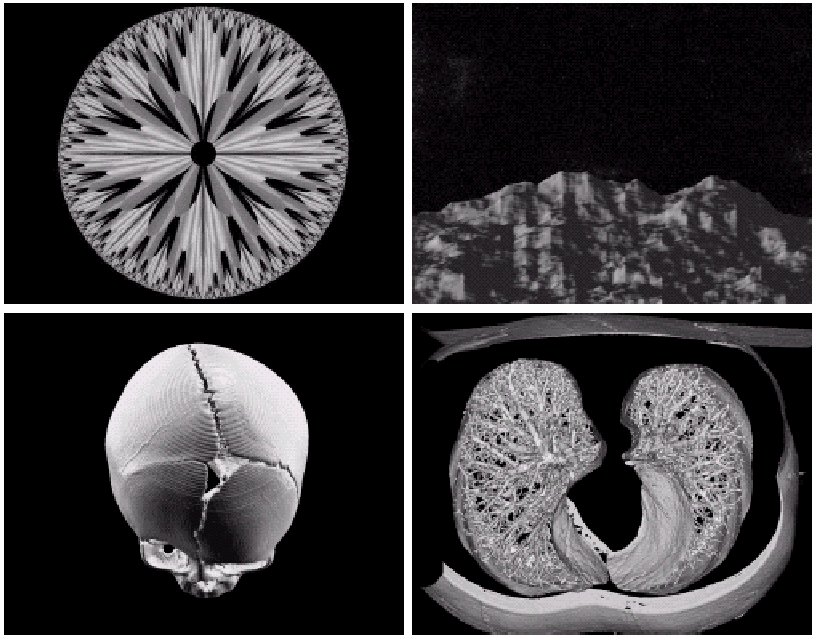
a b

FIGURE 1.21 (a) 250× SEM image of a tungsten filament following thermal failure. (b) 2500× SEM image of damaged integrated circuit. The white fibers are oxides resulting from thermal destruction. (Figure (a) courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene; (b) courtesy of Dr. J. M. Hudak, McMaster University, Hamilton, Ontario, Canada.)

Electron microscopes

- **use a focused beam of electrons instead of light.**
- **are capable of very high magnification (10,000X or more).**
- **transmission electron microscope (TEM), scanning electron microscope (SEM)**

OTHER IMAGING MODALITIES - COMPUTER-GENERATED OBJECTS

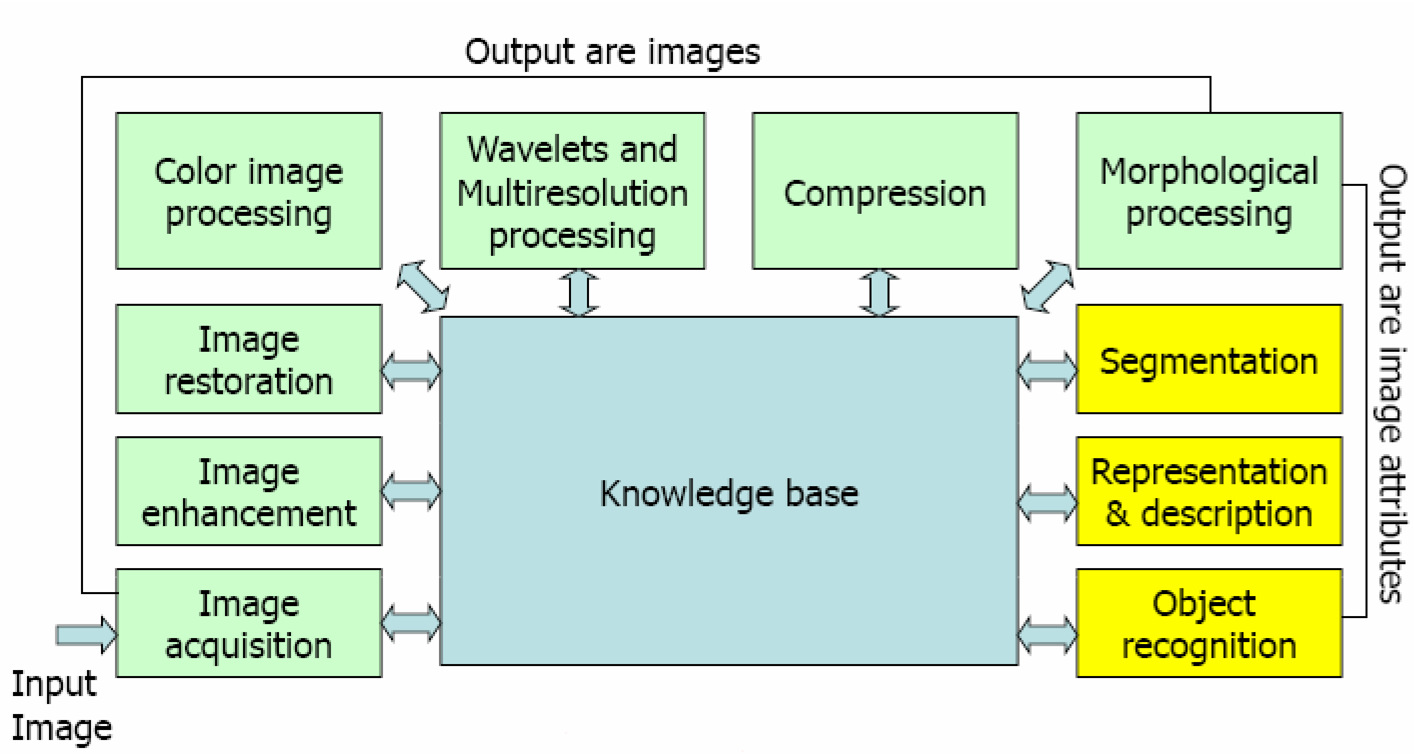


a b
c d

FIGURE 1.22 (a) and (b) Fractal images. (c) and (d) Images generated from 3-D computer models of the objects shown. (Figures (a) and (b) courtesy of Ms. Melissa D. Binde, Swarthmore College, (c) and (d) courtesy of NASA.)

Images that are not obtained from physical objects.

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING



WHAT ARE THE FUNDAMENTAL STEPS?

- Image acquisition: capturing an image in digital form.
- Image enhancement: making an image look better in a subjective way.
- Image restoration: improving the appearance of ana image objectively.
- Color image processing: color models and basic color processing
- Wavelets and multiresolution processing: Wavelet transform in one and two dimensions.
- Image compression: reducing the stored and transmitted image data.
- Morphological image processing: extracting image components that are useful in the representation and description of shape.
- Image segmentation: partitioning an image into its constituent parts or objects.
- Representation and description: boundary representation vs. region representation. Boundary descriptors vs. region descriptors.
- Recognition: assigning a label to an object based on its descriptors.

COMPONENTS OF AN IMAGE PROCESSING SYSTEM

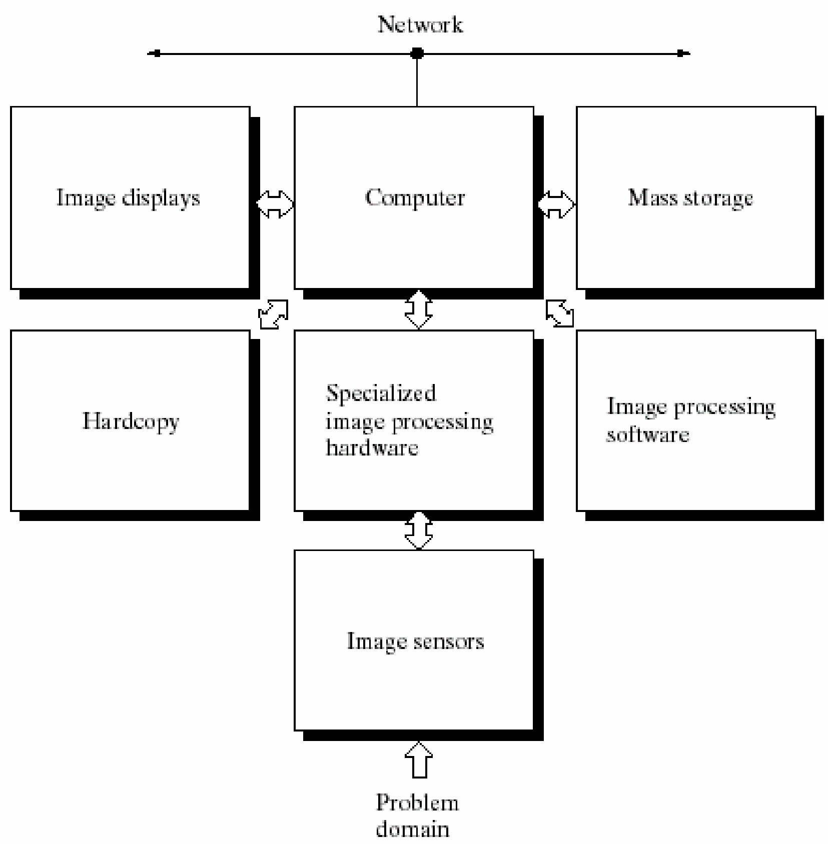


FIGURE 1.24 Components of a general-purpose image processing system.

WHAT ARE THE COMPONENTS?

- **Image sensors and specialized IP HW**
  - **A physical device that is sensitive to the energy radiated by the object.**
  - **A digitizer that converts the output of the physical sensing device into digital form.**
  - **HW that performs other primitive operations.**
- **Computer: Can range from a PC to a supercomputer.**
- **Image processing SW: Specialized modules that perform specific tasks.**
- **Mass storage**
  - **Short term storage: main memory, frame buffers**
  - **On-line storage: magnetic disks and optical disks**
  - **Archival storage: magnetic tapes and optical disks in "jukeboxes"**
- **Hard copy: all types of printers, film cameras, optical disks (CDs and DVDs).**
- **Networking: bandwidth is the key consideration.**

Image Processing

Ch2: Digital image Fundamentals

What is digital Image?

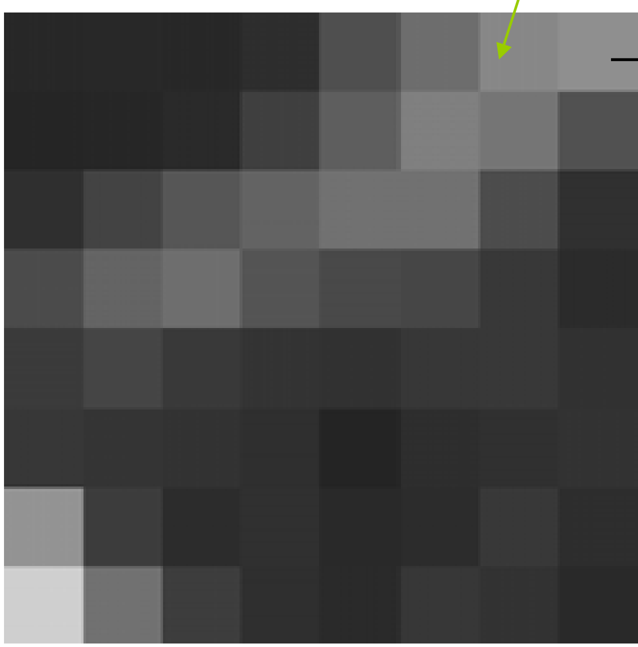
- **digital image:** function of 2 variables, $f(x,y)$, where x and y are spatial coordinates,
- **Pixels (pels)**: Elements of the digital image , each has intensity.
- **Intensity** of pixel: the amplitude غزارة *of gray level (in gray scale images)*

Note: images can be: binary, grayscale, color.

What is digital image?

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

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



pixel

every pixel has intensity كثافة

Ex:

f(4,3)= 123

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

Remember: *images can be: binary, grayscale, color.*

Binary Images

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

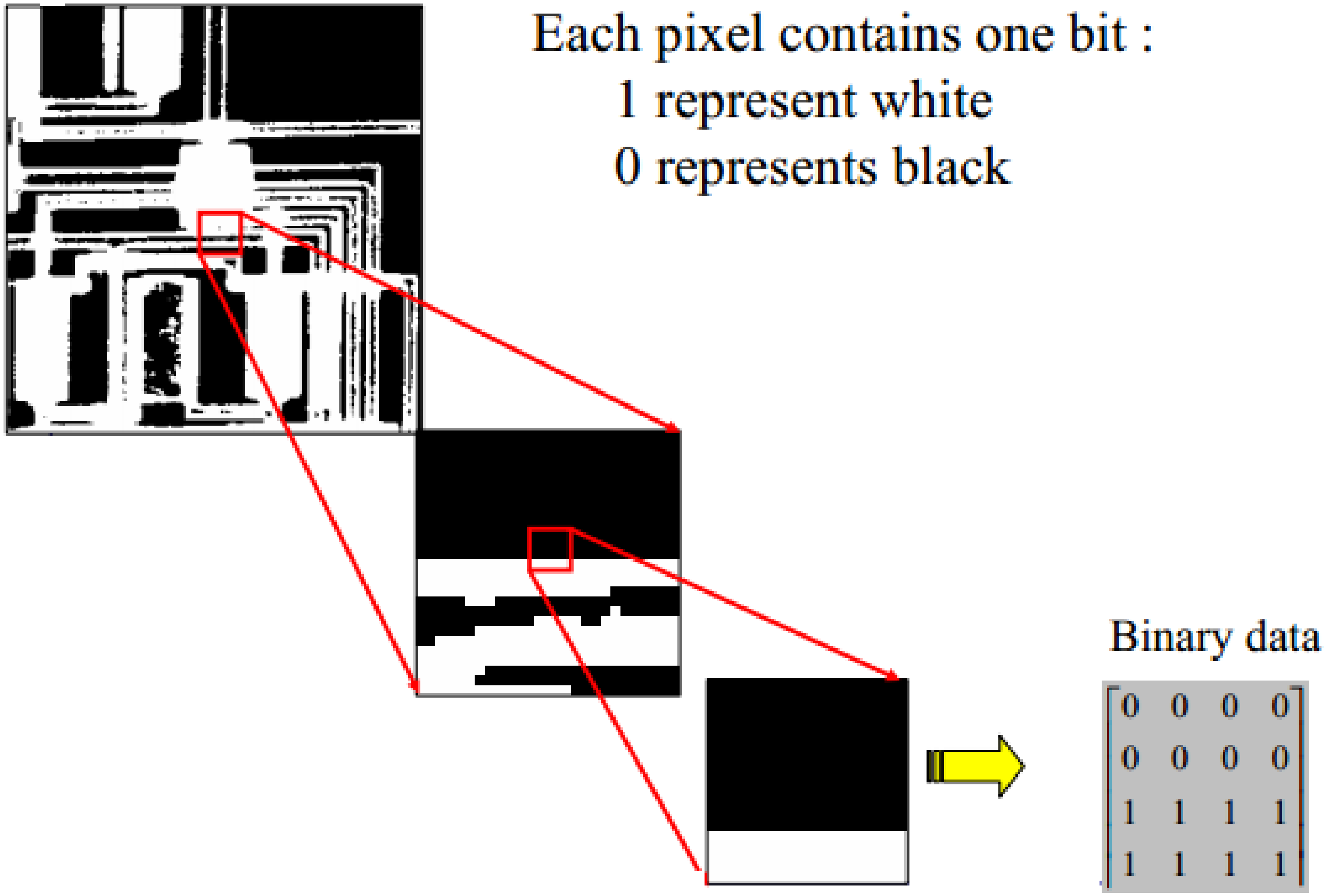
Binary Images

Binary image or black and white image

Each pixel contains one bit :

1 represent white

0 represents black



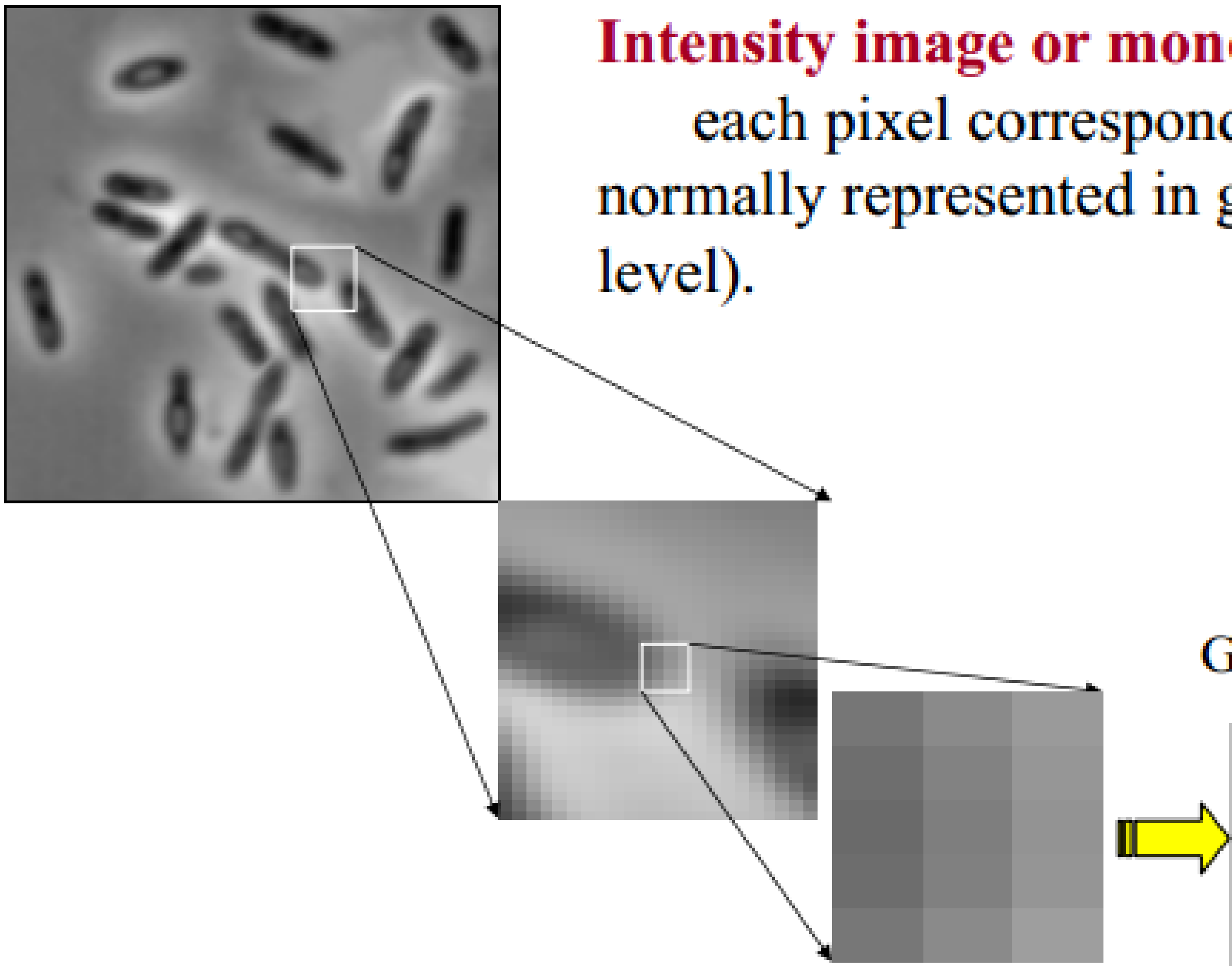
Grayscale Images -monochromatic

- A grayscale (or graylevel) image is simply one in which the only colors are shades of gray (0 – 255)

Grayscale Images -monochromatic

Intensity image or monochrome image

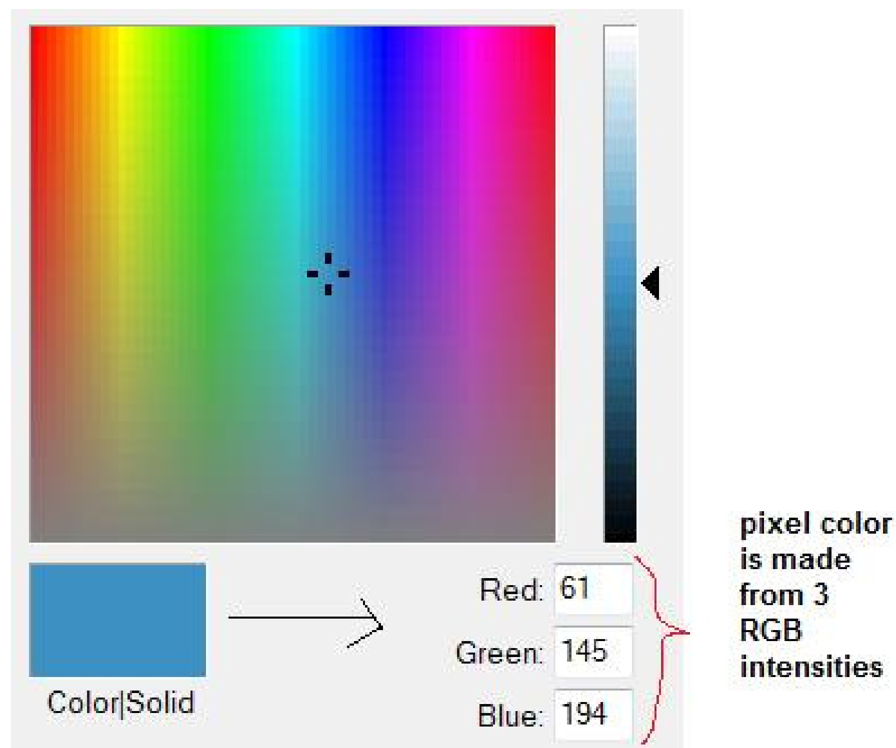
each pixel corresponds to light intensity normally represented in gray scale (gray level).



$$\begin{bmatrix} 10 & 10 & 16 & 28 \\ 9 & 6 & 26 & 37 \\ 15 & 25 & 13 & 22 \\ 32 & 15 & 87 & 39 \end{bmatrix}$$

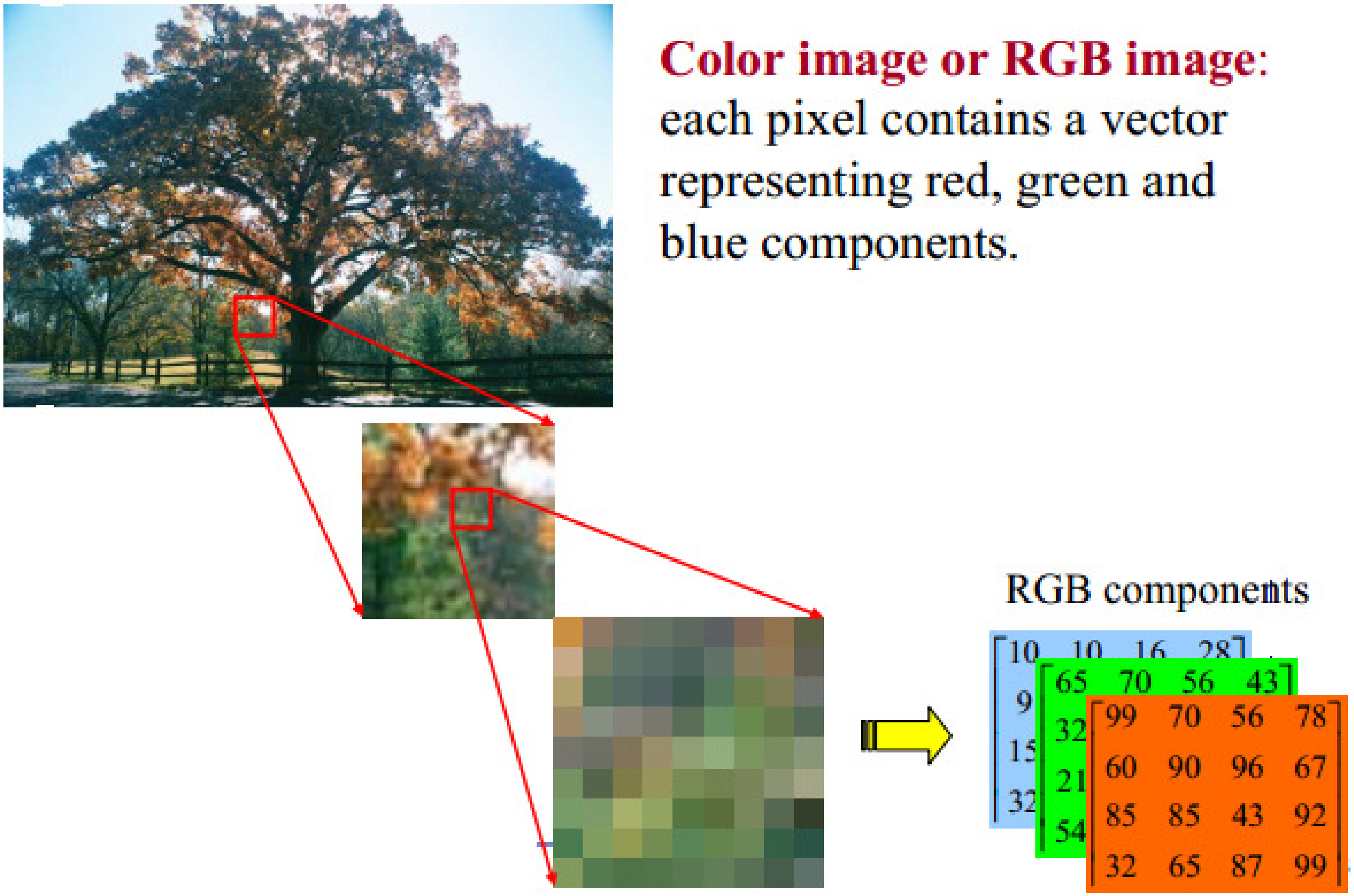
Color Images - chromatic

- <u>Color image:</u> A color image contains pixels each of which holds three intensity values corresponding to the red, green, and blue or(RGB)



Color Images - chromatic

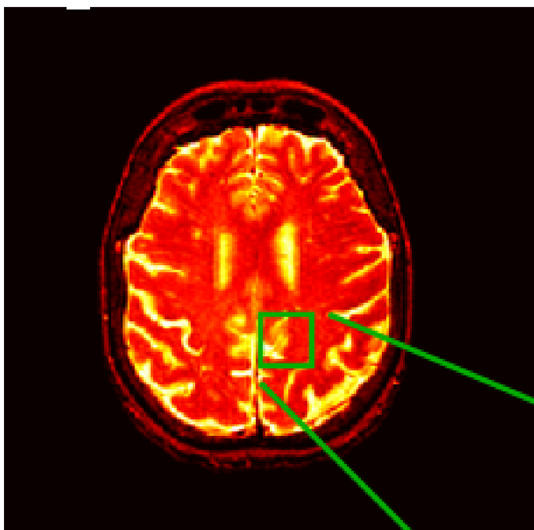
Color image or RGB image: each pixel contains a vector representing red, green and blue components.



Index Image

Index image

Each pixel contains index number pointing to a color in a color table



$$\begin{bmatrix} 1 & 4 & 9 \\ 6 & 4 & 7 \\ 6 & 5 & 2 \end{bmatrix}$$

Index value

Color Table

Index No.	Red component	Green component	Blue component
1	0.1	0.5	0.3
2	1.0	0.0	0.0
3	0.0	1.0	0.0
4	0.5	0.5	0.5
5	0.2	0.8	0.9
...	...	...	...

What is digital image processing? (DIP)

processing digital images by means of a digital computer.

Image sampling and quantization

- In order to process the image, it must be saved on computer.
- The image output of most sensors is continuous voltage waveform.
- But computer deals with digital images not with continuous images, thus: continuous images should be converted into digital form.

continuous image (in real life) → digital (computer)

Image sampling and quantization

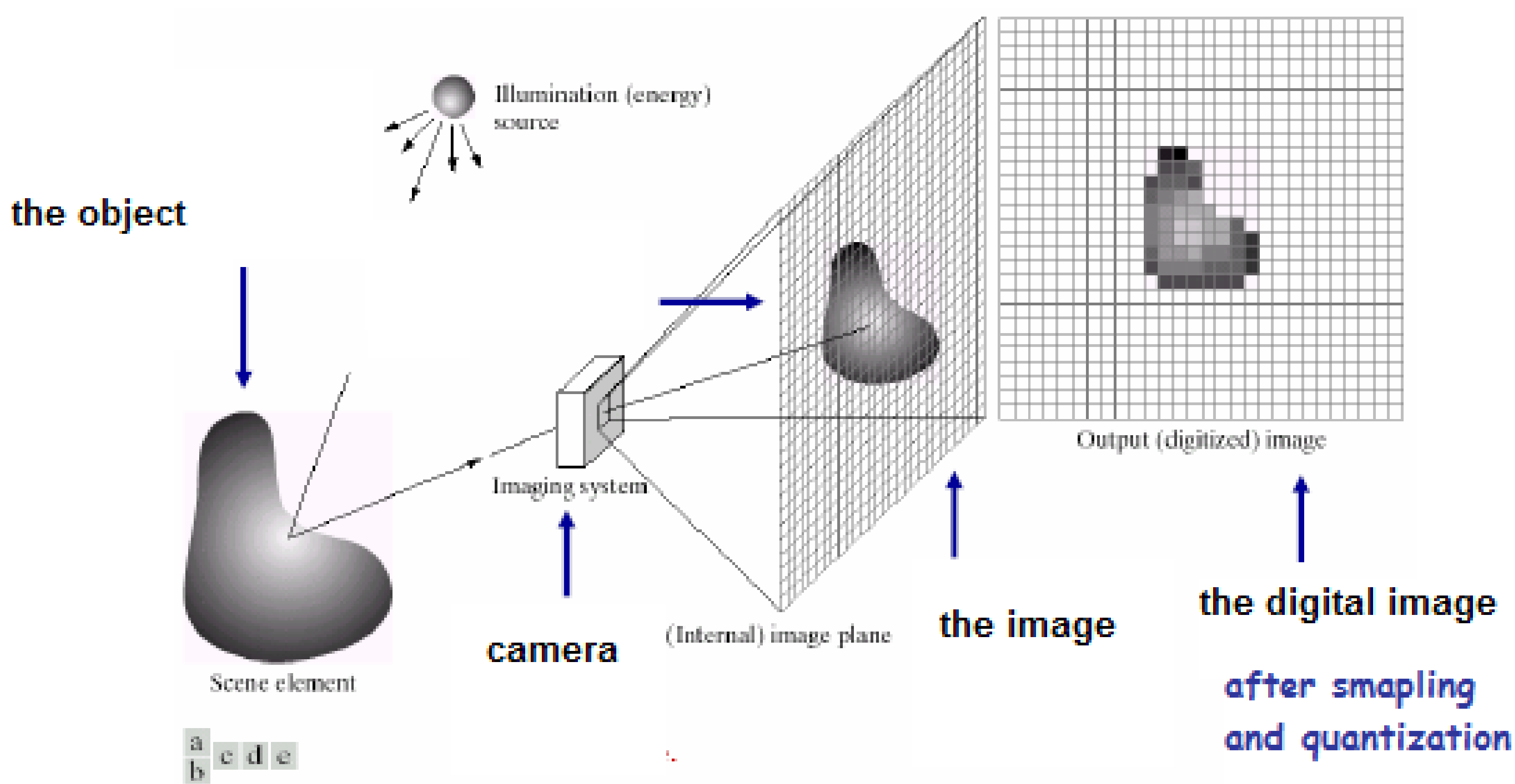


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image sampling and quantization

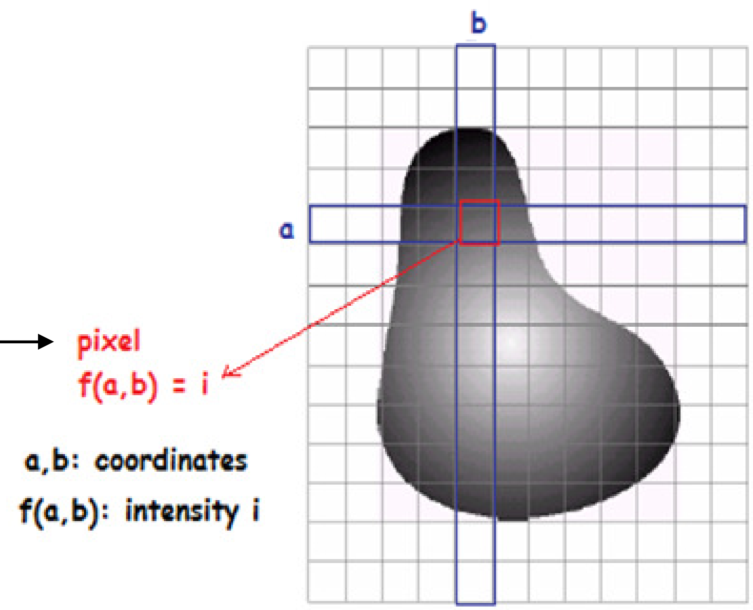
continuous image (in real life) → digital (computer)

To do this we use Two processes:
sampling and **quantization**.

Remember that:

the image is a function f(x,y),

- x and y are coordinates
- F: intensity value (Amplitude)



Sampling: digitizing the coordinate values
Quantization: digitizing the amplitude values

How does the computer digitize the continuous image?

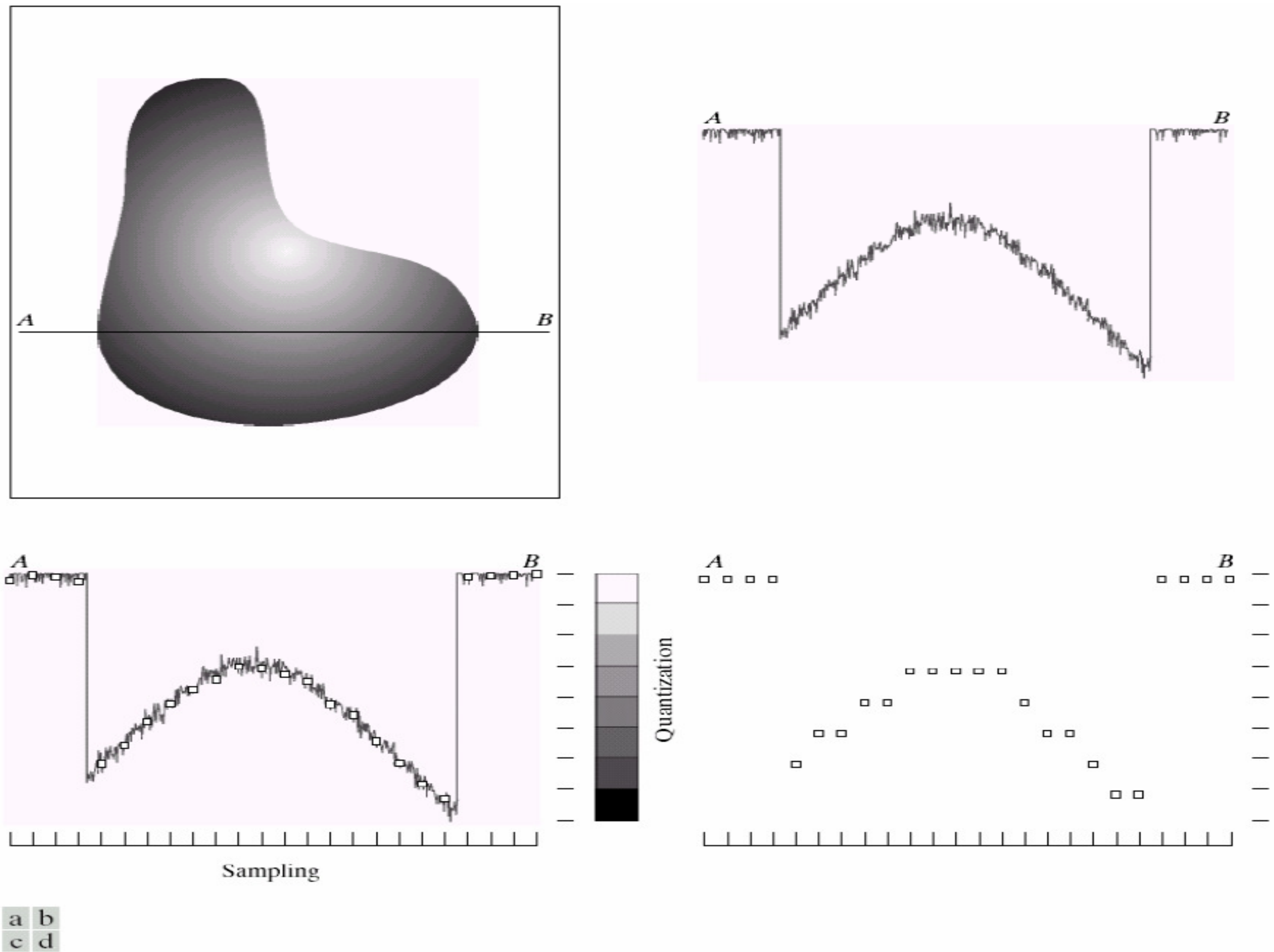
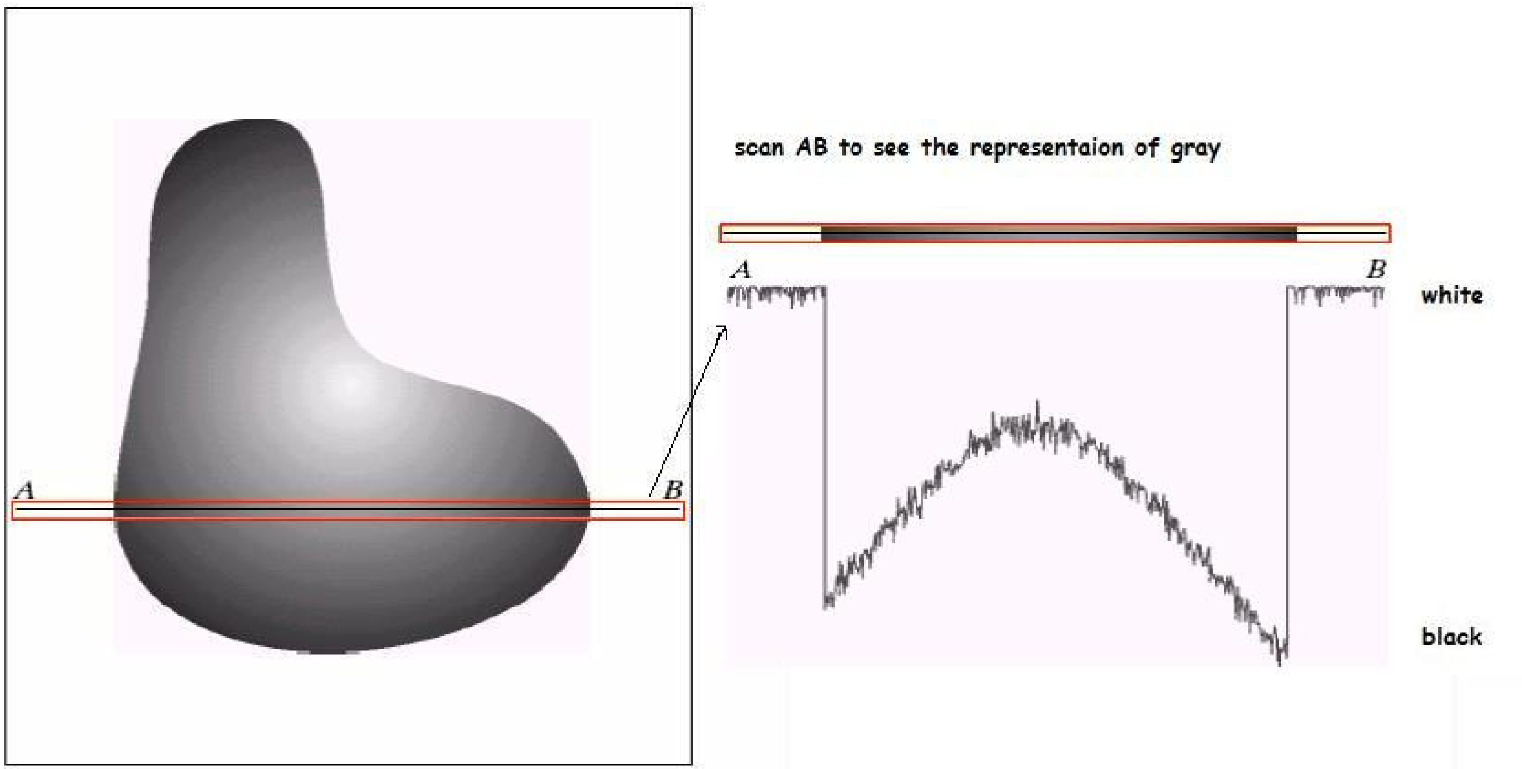


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

How does the computer digitize the continuous image?

Ex:

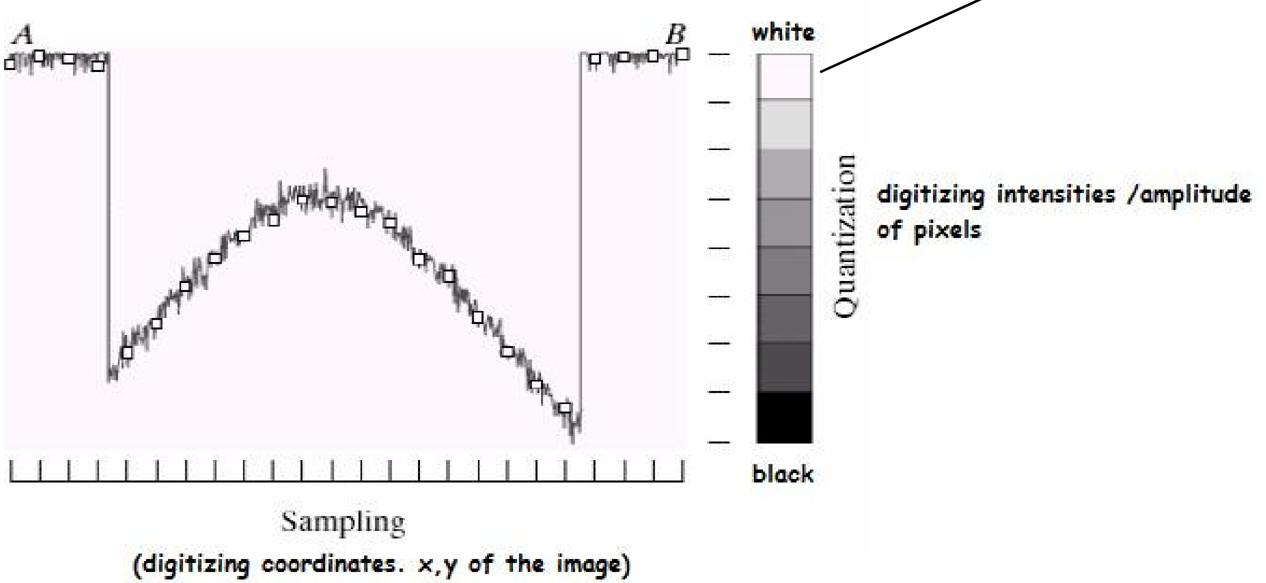
scan a line such as AB from the continuous image, and represent the gray intensities.



How does the computer digitize the continuous image?

Sampling: digitizing coordinates
Quantization: digitizing intensities

Gray-level scale that divides gray-level into 8 discrete levels



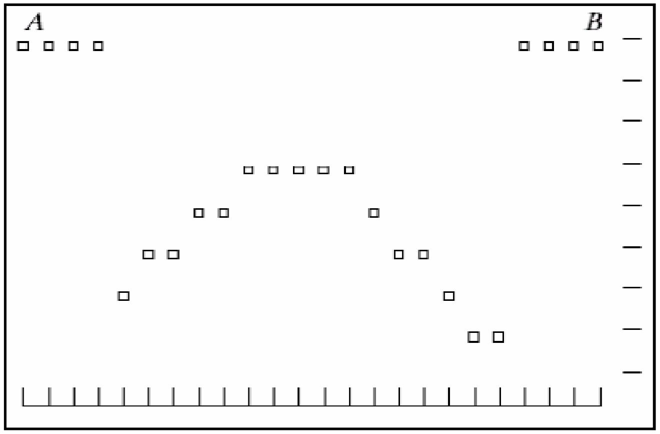
Quantization: converting each sample gray-level value into discrete digital quantity.

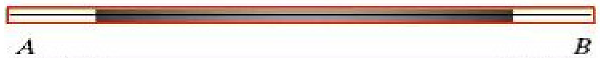
sample is a small white square, located by a vertical tick mark as a point x,y

How does the computer digitize the continuous image?

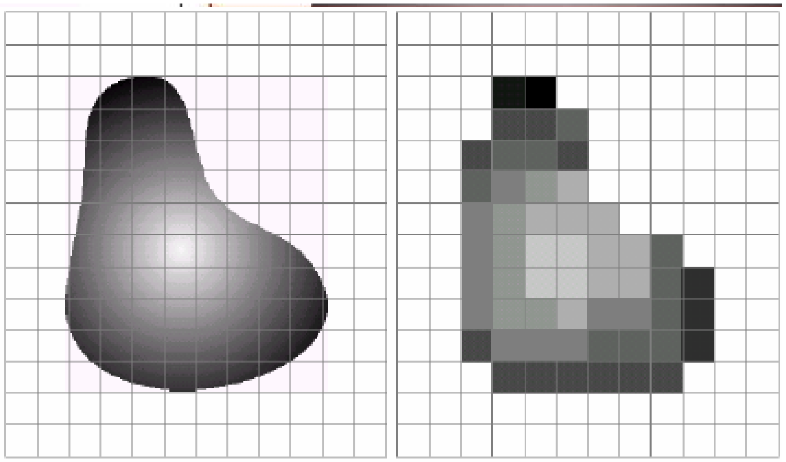
Now:

the digital scanned line AB representation on computer:





The continuous image VS the result of digital image after sampling and quantization



a b

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

Representing digital images

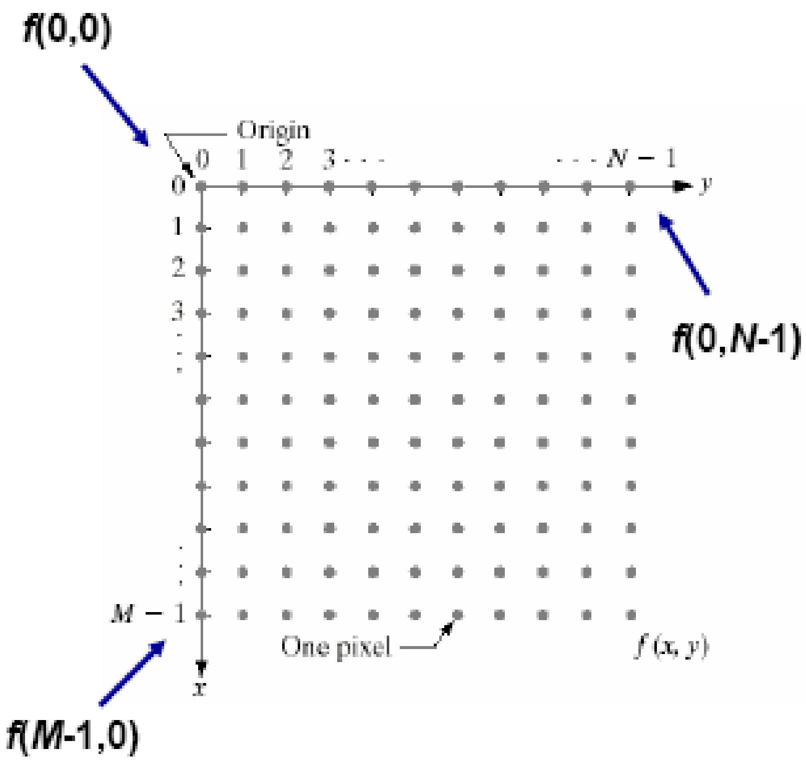


FIGURE 2.18 Coordinate convention used in this book to represent digital images.

L: # of discrete gray levels

$L = 2^k$

$b = M \times N \times k$

Every pixel has a # of bits.

Representing digital images

The notation introduced in the preceding paragraph allows us to write the complete $M \times N$ digital image in the following compact matrix form:

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

The right side of this equation is by definition a digital image. Each element of this matrix array is called an *image element*, *picture element*, *pixel*, or *pel*. The terms *image* and *pixel* will be used throughout the rest of our discussions to denote a digital image and its elements.

In some discussions, it is advantageous to use a more traditional matrix notation to denote a digital image and its elements:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}. \qquad (2.4\text{-}2)$$

Clearly, $a_{ij} = f(x = i, y = j) = f(i, j)$, so Eqs. (2.4-1) and (2.4-2) are identical matrices.

Pixels!

- Every pixel has # of bits (k)

- Q: Suppose a pixel has 1 bit, how many gray levels can it represent?
  Answer: 2 intensity levels only, black and white.
  Bit (0,1) → 0:black , 1: white

- Q: Suppose a pixel has 2 bit, how many gray levels can it represent?
  Answer: 4 gray intensity levels
  2Bit (00, 01, 10 ,11).

Now ..

if we want to represent 256 intensities of grayscale, how many bits do we need?

Answer: 8 bits → which represents: $2^8=256$

so, the gray intensities (L) that the pixel can hold, is calculated according to according to number of bits it has (k).

$$L= 2^k$$

Number of storage of bits:

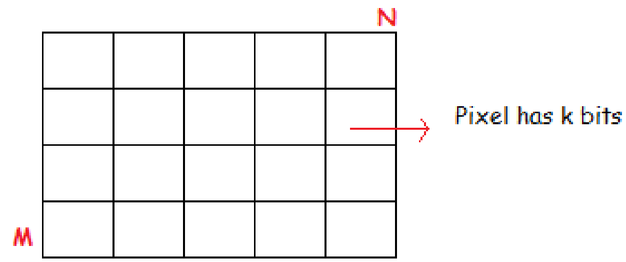
N * M: the no. of pixels in all the image.

K: no. of bits in each pixel

L: grayscale levels the pixel can represent

L= 2^K

all bits in image= N*N*k



NO of pixels= N *M
NO of bits= N*M*k

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

Number of storage of bits:

EX: Here: N=32,   K=3,   L = 2^3 =8

# of pixels=N*N = 1024 . (because in this example: M=N)

# of bits = N*N*K = 1024*3= 3072

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

N=M in this table, which means no. of horizontal pixels= no. of vertical pixels. And thus:

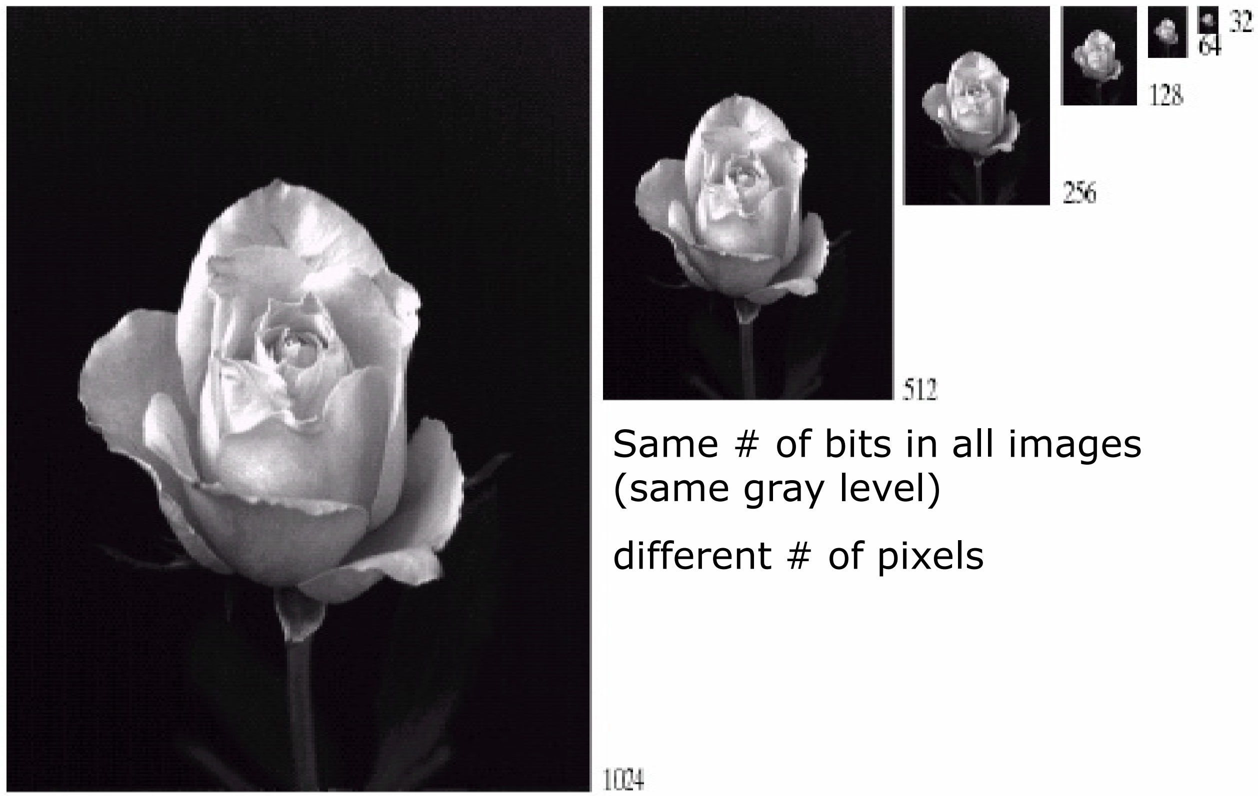
# of pixels in the image= N*N

Spatial and Gray-Level Resolution

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

Spatial and gray-level resolution - reducing SR (subsampling)

The **lower resolution** images are smaller than the original



Same # of bits in all images (same gray level)

different # of pixels

FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

subSampling is performed by deleting rows and columns from the original image.

Spatial and gray-level resolution

Re sampling

(pixel replication)

A special case of nearest neighbor zooming.

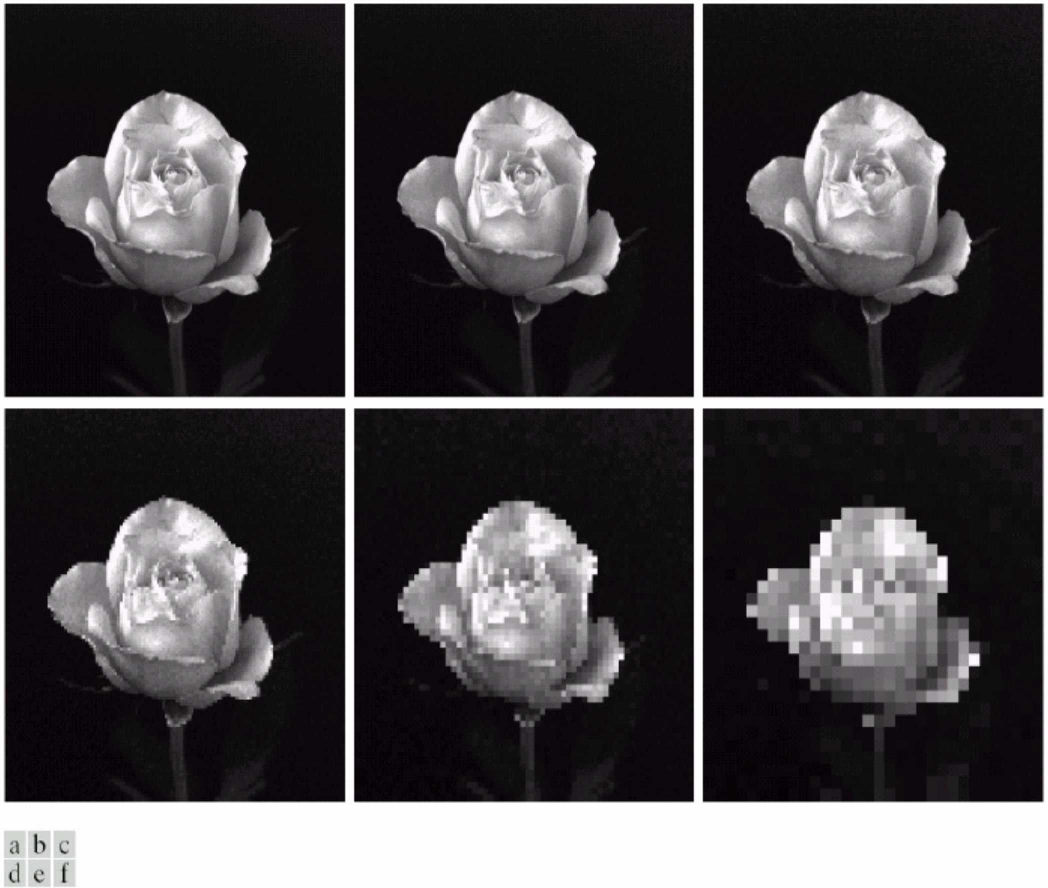


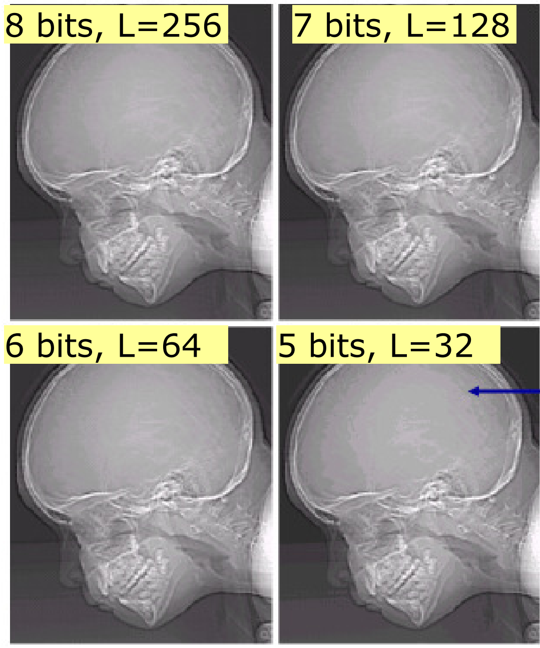
FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

Resampling is performed by row and column duplication

Reducing IR

Here we keep the number of samples constant and **reduce** the number of intensity

In these images, the # of samples is constant but the # of gray levels was reduced from 256 to 32.



a b
c d

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

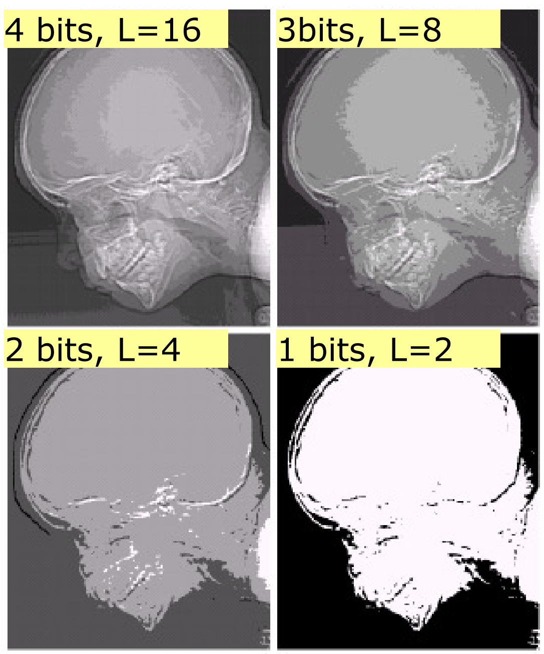
In this example,

all images has 452*374 pixels, but different # of bits per pixel (as shown in yellow)

In these images, the # of samples is constant but the # of gray levels was reduced from 16 to 2.

e f
g h

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



different # of bits in all images (different gray level)

same # of pixels

☝ Remember that:

every pixel has # of bits k
And # of gray levels is $L=2^k$

More pixels → more resolution
More bit/pixel → more accuracy

Image Interpolation

Image Interpolation is the process of using known data to estimate values at unknown location.

Image interpolation is used for zooming, shrinking, rotating, geometric corrections.

- Zooming + shrinking are Image **resizing** tasks and come under image re-sampling methods and we will study them in the following

Zooming (over sampling) images

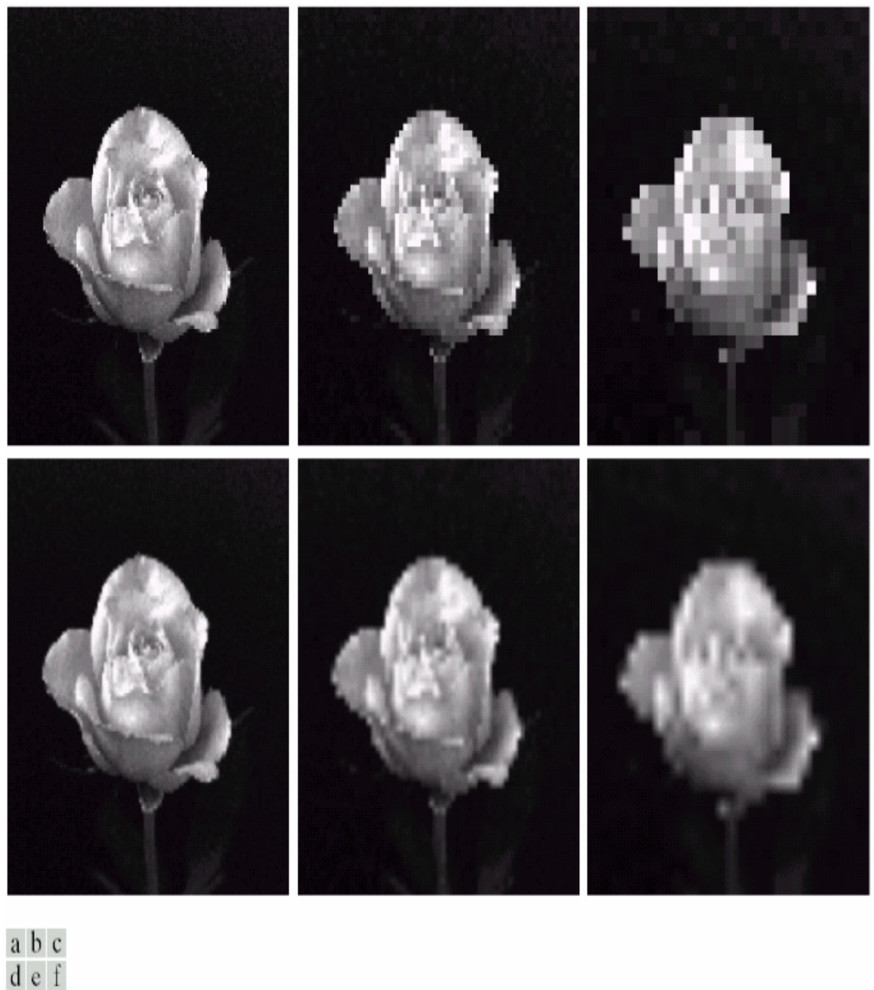


FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Zoomed by using nearest neighbor

Zoomed by using Bilinear

Zooming (over sampling) images

Zooming requires 2 steps:

- The creation of new pixel locations.
- The assignment of gray levels to these new locations.

Two techniques for zooming:

1. Nearest neighbor interpolation
2. Bilinear interpolation
3. Bicubic interpolation

Nearest neighbor interpolation

Example:

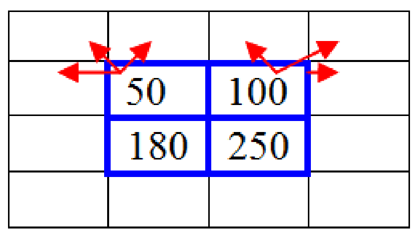
Suppose A 2x2 pixels image will be enlarged 2 times by the nearest neighbor method:

1. Lay an imaginary 4*4 grid over the original image..
2. For any point in the overlay, look for the closest pixel in the original image, and assign its gray level to the new pixel in the grid. (copy)
3. When all the new pixels are assigned values, expand the overlay grid to the original specified size to obtain the zoomed image.

- Pixel replication (re sampling) is a special case that is applicable when the size of the image needs to be increased an integer number of times (like 2 times not 1.5 for example).

50	100
180	250

The original image pixels value



Nearest neighbor zooming

50	50	100	100
50	50	100	100
180	180	250	250
180	180	250	250

The new image

+ ve : Nearest neighbor is fast

-ve: it produces a checkerboard effect like this!

shrinking

- Similar to image zooming.

Shrinking an image an integer number of times

☐ Pixel replication is replaced by row&column deletion.

Bilinear Interpolation:

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

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

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

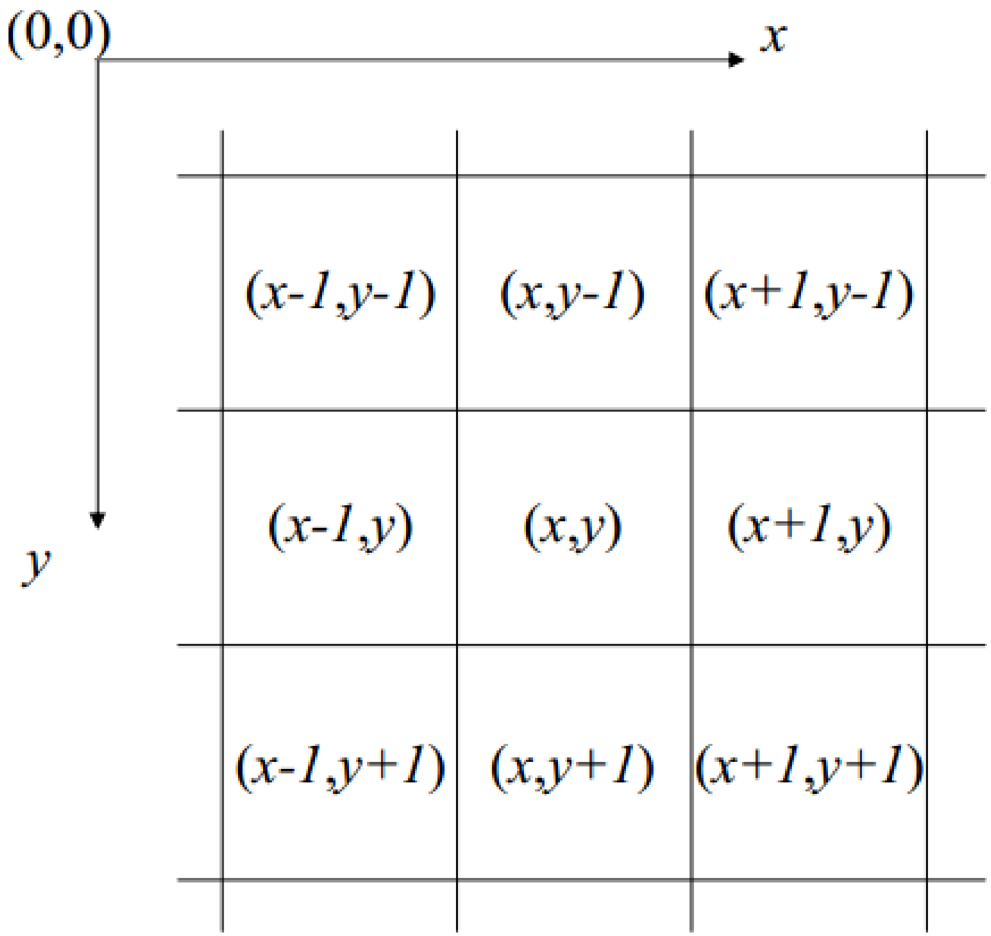
Bicubic Interpolation:

- Involves 16 nearest neighbours of a point.

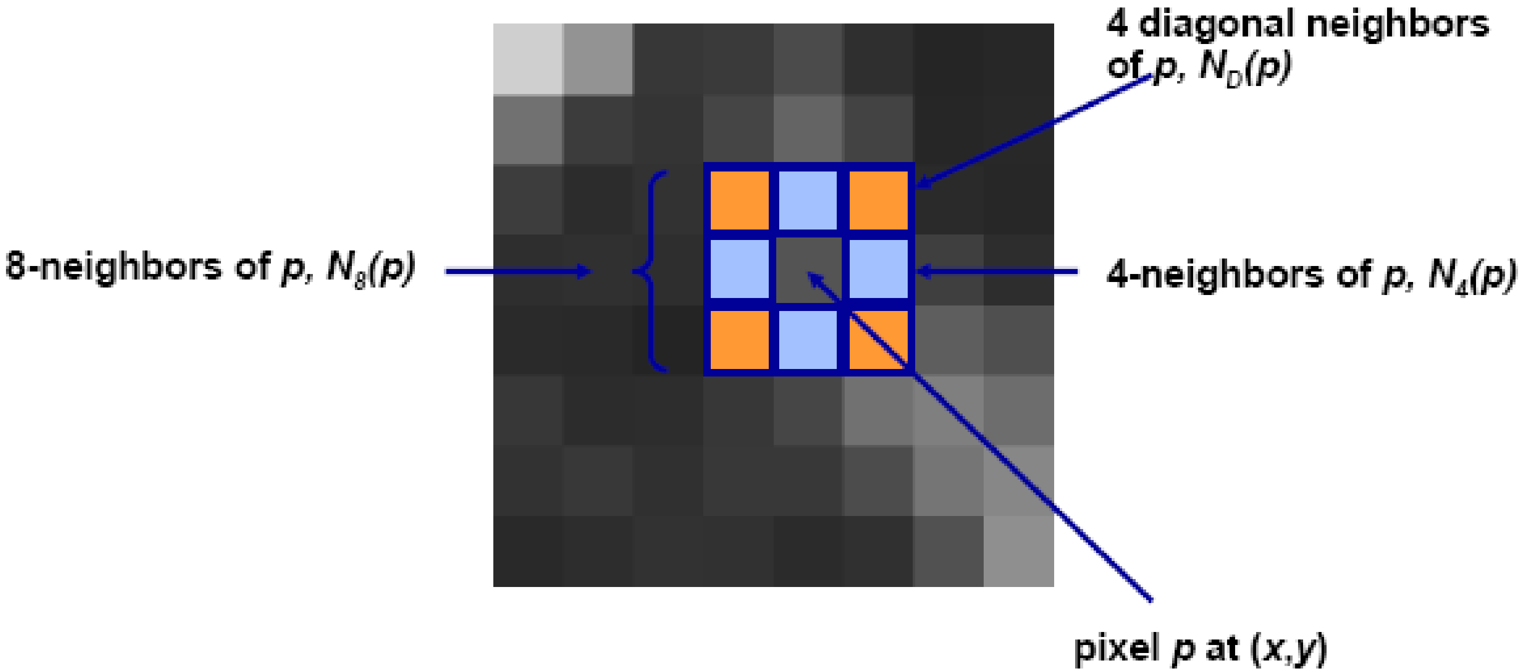
$$v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j$$

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

Some basic relationships between pixels



Neighbors of a pixel



Relationships Between Pixels: Neighbors of a Pixel

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

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

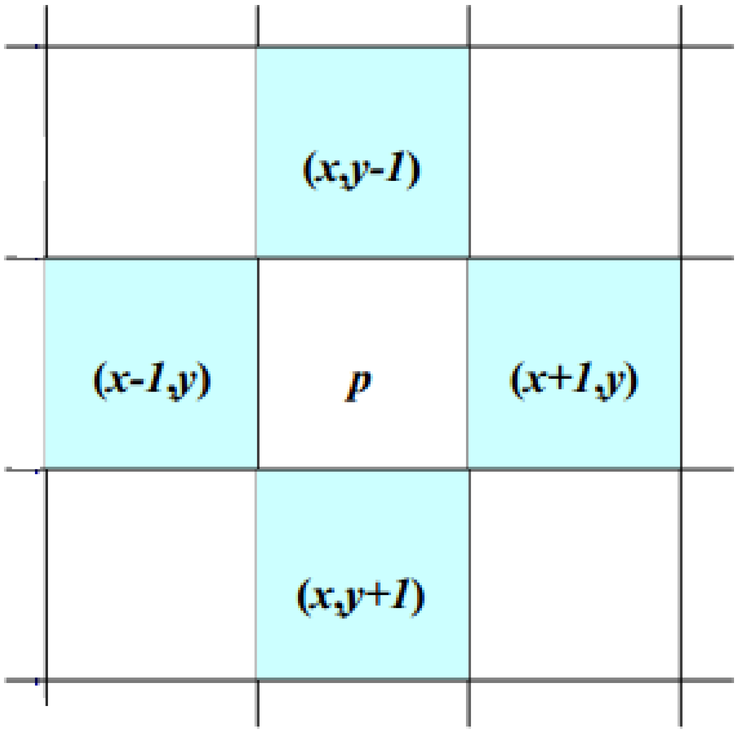
This set of pixels, called the 4-*neighbors* of p, is denoted by $N_4(p)$. Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

The four *diagonal* neighbors of p have coordinates

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

and are denoted by $N_D(p)$. These points, together with the 4-neighbors, are called the 8-*neighbors* of p, denoted by $N_8(p)$. As before, some of the points in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of the image.

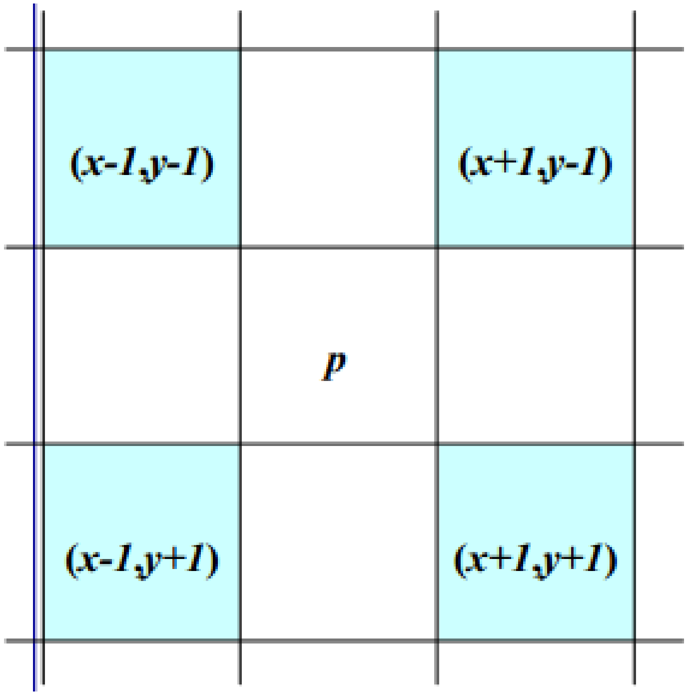
Neighbors of a pixel



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

Neighbors of a pixel



Diagonal neighbors of *p*

$$N_D(p) = \left\{ \begin{pmatrix} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{pmatrix} \right\}$$

Neighbors of a pixel

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

8-neighbors of *p*:

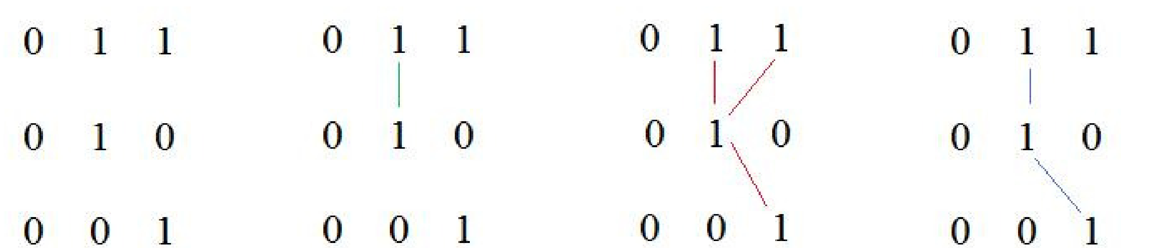
$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

Adjacency

V: set of gray level values (L), (V is a subset of L.)

3 types of adjacency

- 4- adjacency: 2 pixels *p* and *q* with values from *V* are 4- adjacent if *q* is in the set $N_4(p)$
- 8- adjacency: 2 pixels *p* and *q* with values from *V* are 8- adjacent if *q* is in the set $N_8(p)$
- *m*- adjacency: 2 pixels *p* and *q* with values from *V* are *m*adjacent if
  1. *q* is in $N_4(p)$, *or*
  2. *q* is in $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V



Arrangement of pixels in a binary image — 4-adjacent pixels — 8 - adjacent pixels — m - adjacent pixels

Digital Path

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

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

where

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

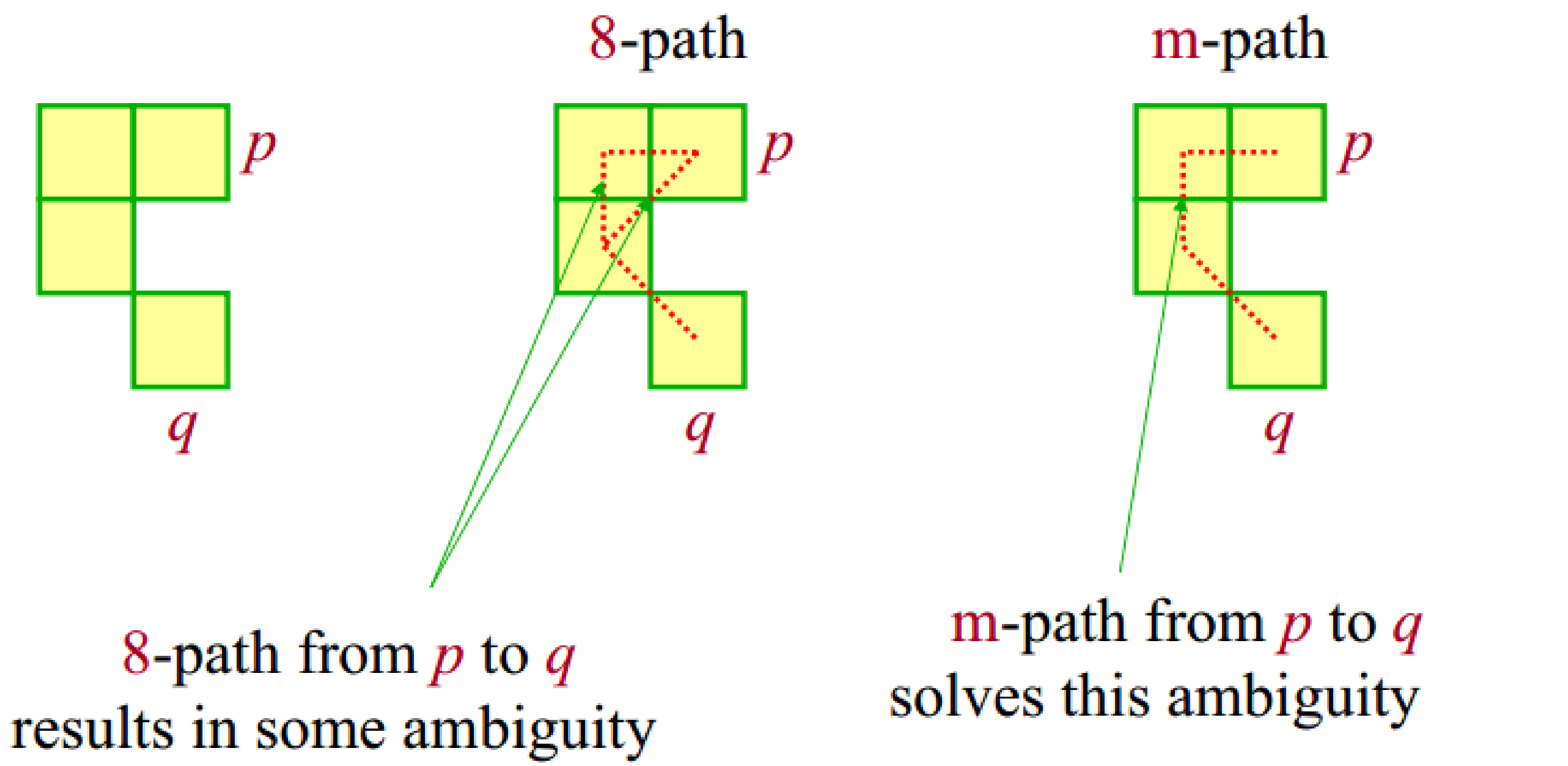
are adjacent for

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

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

Digital Path

- *We can define **4-, 8-, or m-paths** depending* on the type of adjacency specified.



connectivity

- *S*: a subset of pixels in an image.
- Two pixels *p* and *q* are said to be connected in *S* if there exists a path between them consisting entirely of pixels in *S*.
- For any pixel *p* in *S*, the set of pixels that are connected to it in *S* is called a connected component of *S*.
- If S has only one connected component, it is called a connected set.

Regions and boundaries

- *R*: a subset of pixels in an image.
- *R* is a region of the image if *R* is a connected set.

- The boundary of a region *R* is the set of pixels in the region that have one or more neighbors that are not in *R*.

Foreground and background

Suppose that the image contains K disjoint regions Rk none of which touches the image border .

Ru : the union of all regions .

$(Ru)^c$: is the complement .

so Ru is called foreground , and $(Ru)^c$: is the background .

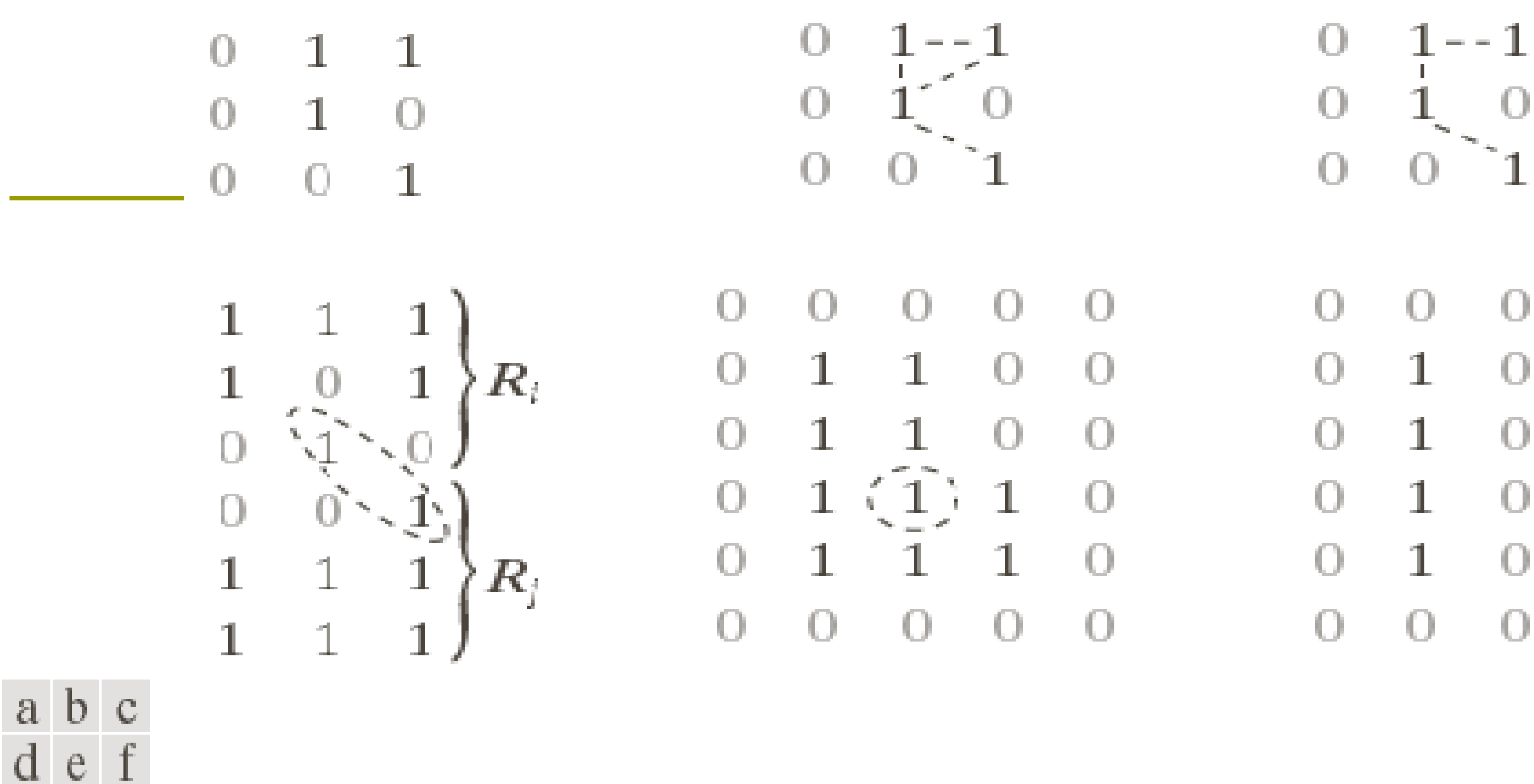


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) m-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

Distance measures

If we have 3 pixels: p,q,z:

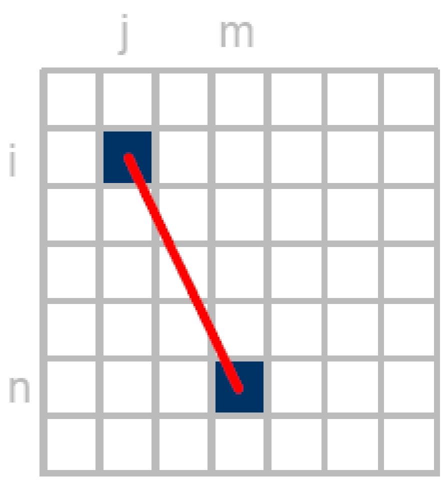
p with (x,y)
q with (s,t)
z with (v,w)

Then:

$D(p,q) = 0$ iff $p = q$
$D(p,q) = D(q,p)$
$D(p,z) \leq D(p,q) + D(q,z)$

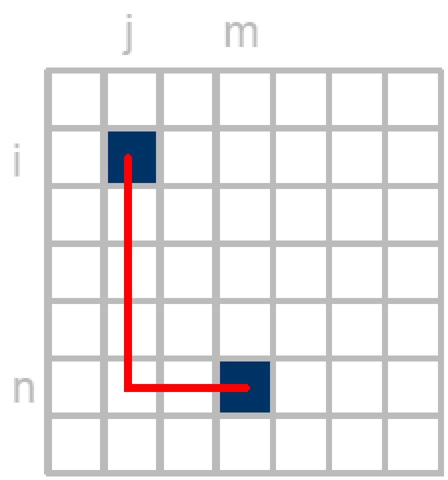
- Euclidean distance between p and q: $De(p,q) = [(x-s)^2 + (y-t)^2]1/2$
- $D4$ distance: $D4(p,q) = |x-s| + |y-t|$
- $D8$ distance: $D8(p,q) = \max(|x-s|, |y-t|)$
- $D4$ and $D8$ distances between p and q are independent of any paths that might exist between the points.
- For m-adjacency, Dm distance between two points is defined as the shortest m-path between the points.

Distance measures



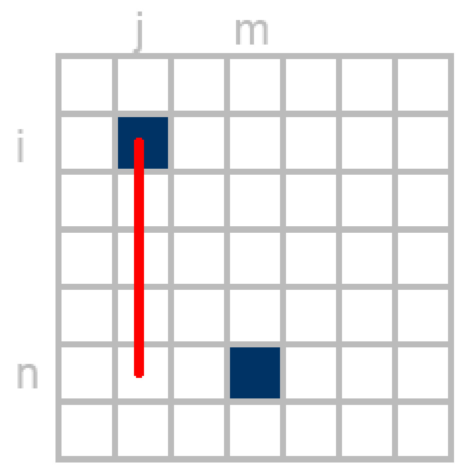
Euclidean Distance

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



City Block Distance

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



Chessboard Distance

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

Distance measures

D_4-distance (*city-block distance*) is defined as

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

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

Pixels with $D_4(p) = 1$ is 4-neighbors of p.

Distance measures

D_8-distance (*chessboard distance*) is defined as

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

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

Pixels with $D_8(p) = 1$ is 8-neighbors of p.

Example

Compute the distance between the two pixels using the three distances :

q:(1,1)

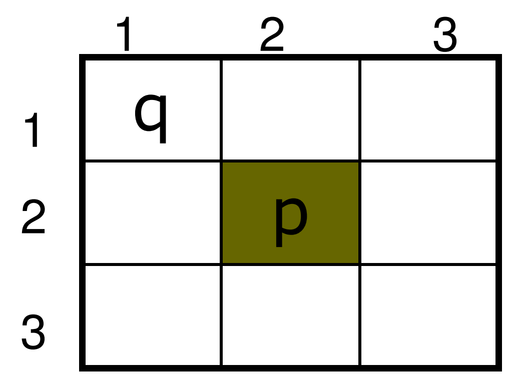
P: (2,2)

Euclidian distance : $((1-2)^2+(1-2)^2)^{1/2}$ = sqrt(2).

D4(City Block distance): |1-2| +|1-2| =2

D8(chessboard distance) : max(|1-2|,|1-2|)= 1

(because it is one of the 8-neighbors)



Distance measures

Example :

Use the city block distance to prove 4-neighbors ?

Pixel A : | 2-2| + |1-2| = 1

Pixel B: | 3-2|+|2-2|= 1

Pixel C: |2-2|+|2-3| =1

Pixel D: |1-2| + |2-2| = 1

	1	2	3
1		d	
2	a	p	c
3		b	

Now as a homework try the chessboard distance to proof the 8- neighbors!!!!

Image Processing

Ch3: Intensity Transformation and spatial filters

Image Enhancement?

☞ **Enhancement :** is to process an image so that the result is more suitable than the original image for a *specific* application.

Enhancement techniques fall into 2 types:

- Spatial domain: direct manipulation of pixels in the image plane
- Frequency domain: modifying Fourier transform of the image.

☞ *In this chapter, we are going to discuss spatial domain techniques*

Spatial domain

- <u>Spatial domain</u>: aggregate of pixels composing an image

Spatial domain processes:

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

Processed (output) image

Input image

Operator T defined on some <u>neighborhood</u> of f(x,y)

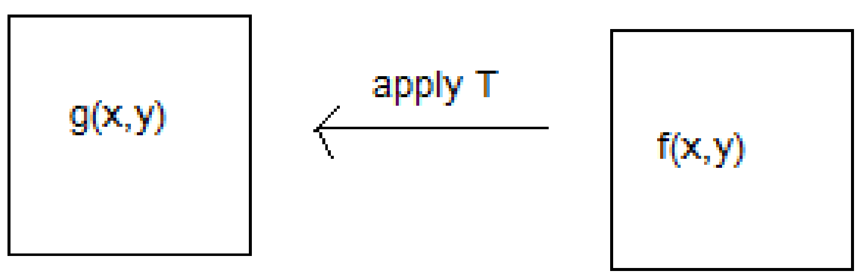
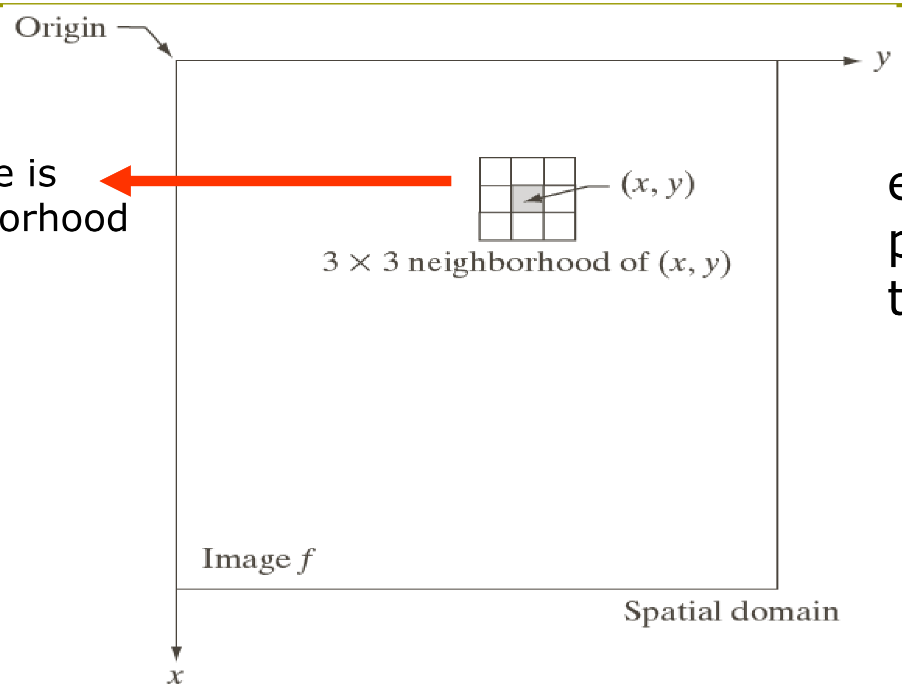


image after processing

original image

Defining a neighborhood (T)

This rectangle is called neighborhood or mask



Operator T is applied at each location (x,y) to produce the output g at that location.

types of neighborhood:

1. **intensity transformation:** neighborhood of size 1x1
2. **spatial filter** (or mask ,kernel, template or window): neighborhood of larger size , like in the above example.

Intensity transformations functions

- The smallest mask is of size 1x1 (1 pixel)
- Here, T is called intensity transformation function or (mapping, gray level function)

$$\underbrace{g(x,y)}_{s} = T[\underbrace{f(x,y)}_{r}]$$

$$s = T(r)$$

s,r : denote the intensity of g and f at any point (x,y) .

Intensity transformations functions

Intensity transformation functions fall into 2 approaches:

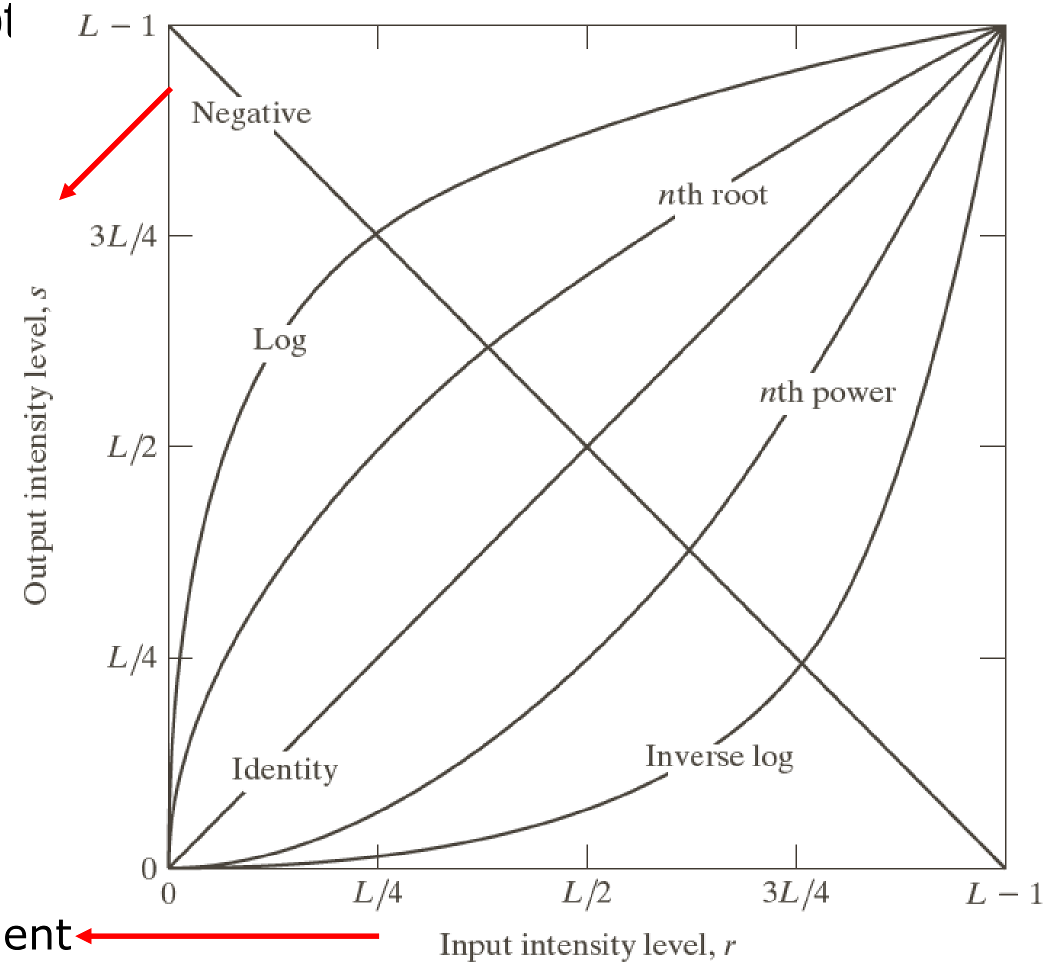
1) **Basic intensity transformations**
   a) Linear (negative and identity).
   b) logarithmic (Log and Inverse Log) .
   c) Power(nth power and nth root).

2) **piecewise Linear transformation functions.**
   a) Contrast stretching, thresholding
   b) Gray-level slicing
   c) Bit-plane slicing

Basic intensity (gray level) transformations

a) Linear (negative and identity).
b) logarithmic (Log and Inverse Log) .
c) Power(nth power and nth root

New gray level



Old gray level before enhancement

Basic intensity (gray level) transformations

Linear (negative and identity)

The negative of an image with intensity levels in the range [0,L-1] is obtained by using the negative transformation :

$$s = L-1-r$$

Image (r)

Image (s) after applying T (negative)

Advantages of negative :

✓ Produces an equivalent of a photographic negative.

✓ Enhances white or gray detail embedded in dark regions especially when the black areas are dominant in size..

Basic intensity (gray level) transformations

Linear (negative and identity)

The negative of an image with intensity levels in the range [0,L-1] is obtained by using the negative transformation :

$$s= L-1-r$$

Example

the following matrix represents the pixels values of an 8-bit image (r) , apply negative transform and find the resulting image pixel values.

Image (r)

100	110	90	95
98	140	145	135
89	90	88	85
102	105	99	115

solution:

$L= 2^8 = 256$

$s=L-1-r$

$s =255-r$

Apply this transform to each pixel to find the negative

Image (s)

155	145	165	160
157	115	110	120
166	165	167	170
153	150	156	140 s

Basic intensity (gray level) transformations

Linear (negative and identity)

Exercise:

the following matrix represents the pixels values of a 5-bit image (r) , apply negative transform and find the resulting image pixel values.

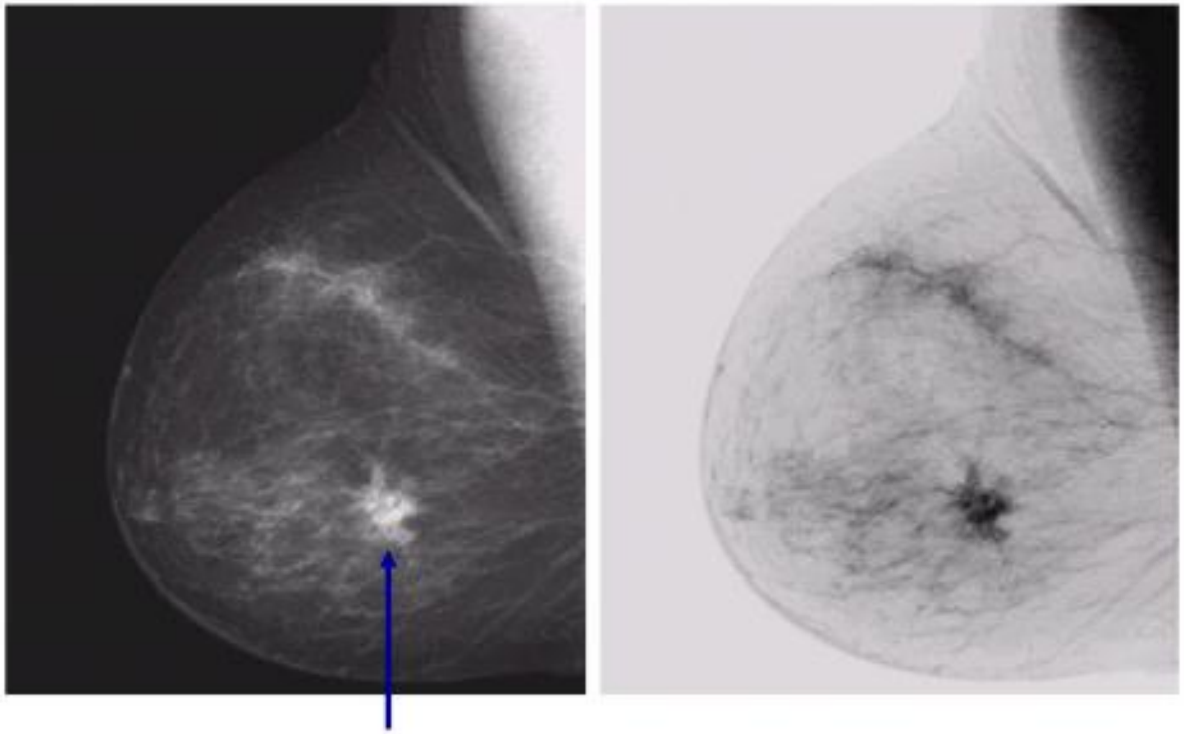
Image (r)

21	26	29	30
19	21	20	30
16	16	26	31
19	18	27	23

solution:

Image (s)

Image Negation (Example)



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.)

Shows a small lesion.

$s = L - 1 - r$

Produces an equivalent of a photographic negative.
Enhances white or gray detail embedded in dark regions.

2. Log Transformations

- *The general form is:*

$$S = c\log(1+r) \qquad\qquad r >= 0$$

where *c is a constant, and it is assumed that*

- *It* maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- The opposite is true of higher values of input levels.
- Used to expand the values of dark pixels in an image while compressing the higher-level values.
- The opposite is true of the inverse log.
- The power-law transformations are much more versatile for this purpose than the log transformation

Basic intensity (gray level) transformations.

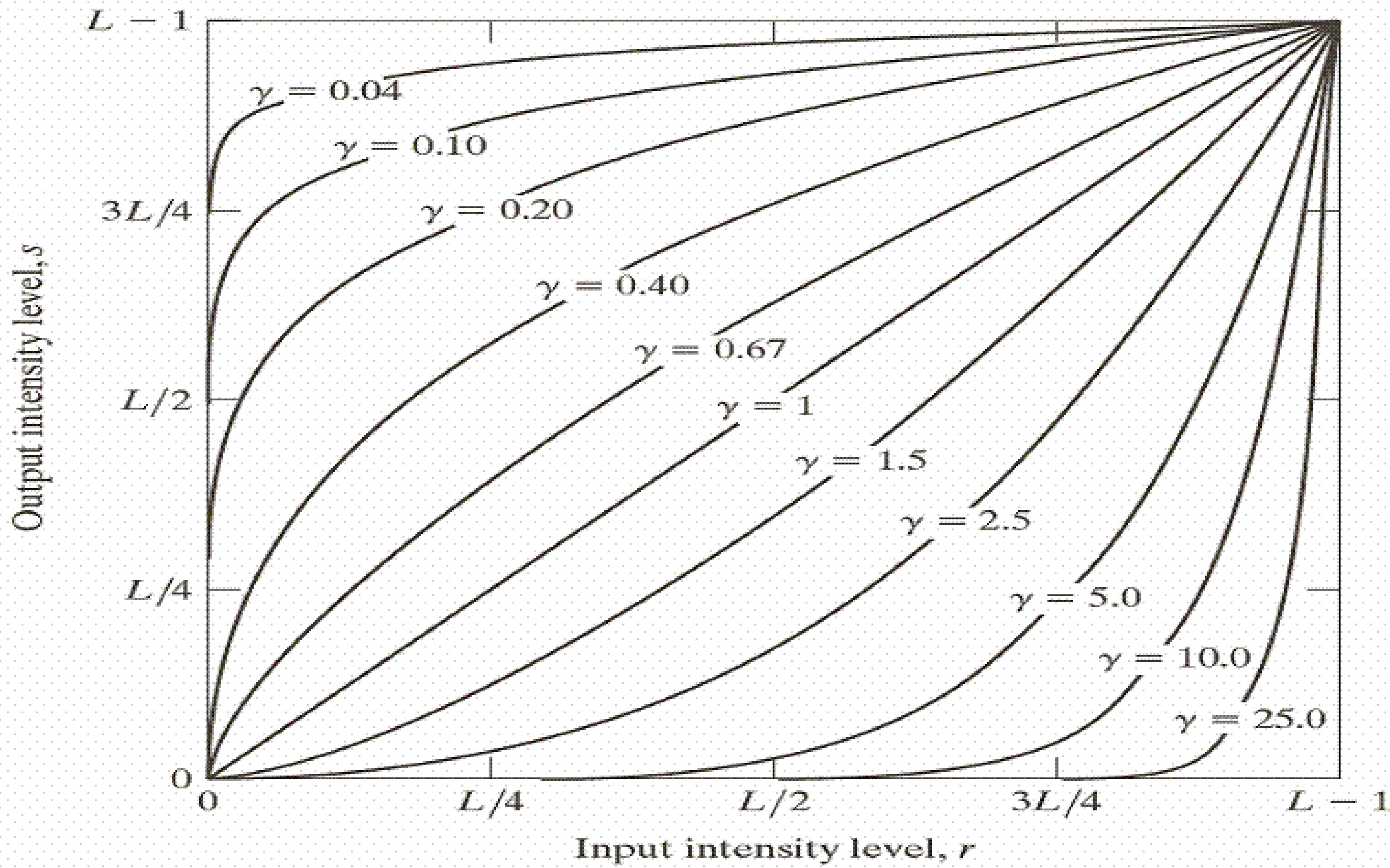
3. Power-Law Transformations

- Power-law transformations have the basic **form**

$$S=cr^{\gamma}$$

- Where c and gamma are positive constants.
- Plots of *s versus r for various* values of gamma are shown in **Fig. 3.6**.
- As in the case of the log transformation, power-law curves with fractional values of gamma map a **narrow range** of dark input values into a **wider range** of output values, with the opposite being true for higher values of input levels.

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.



Power-law transformation

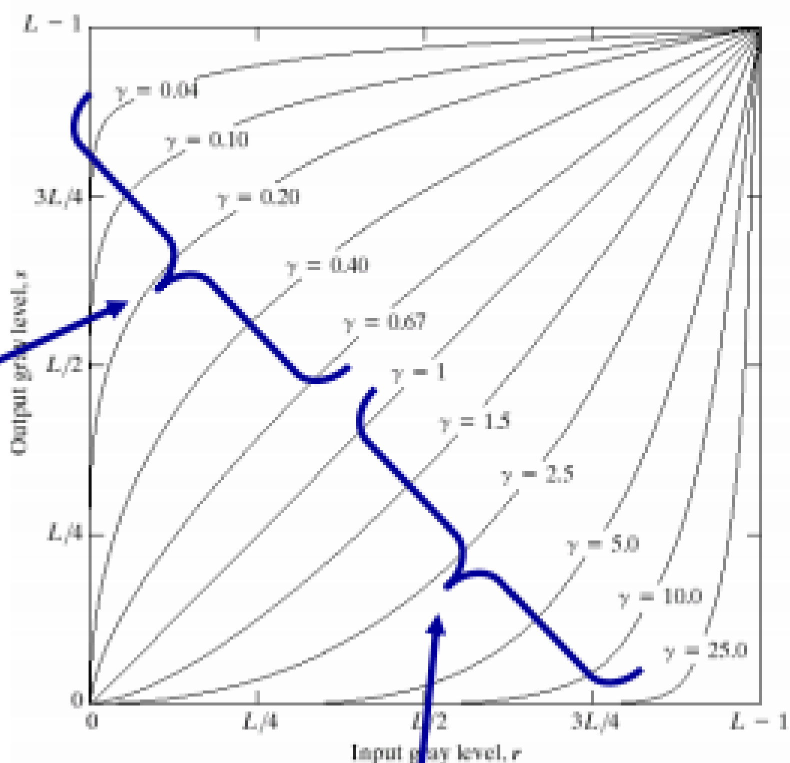
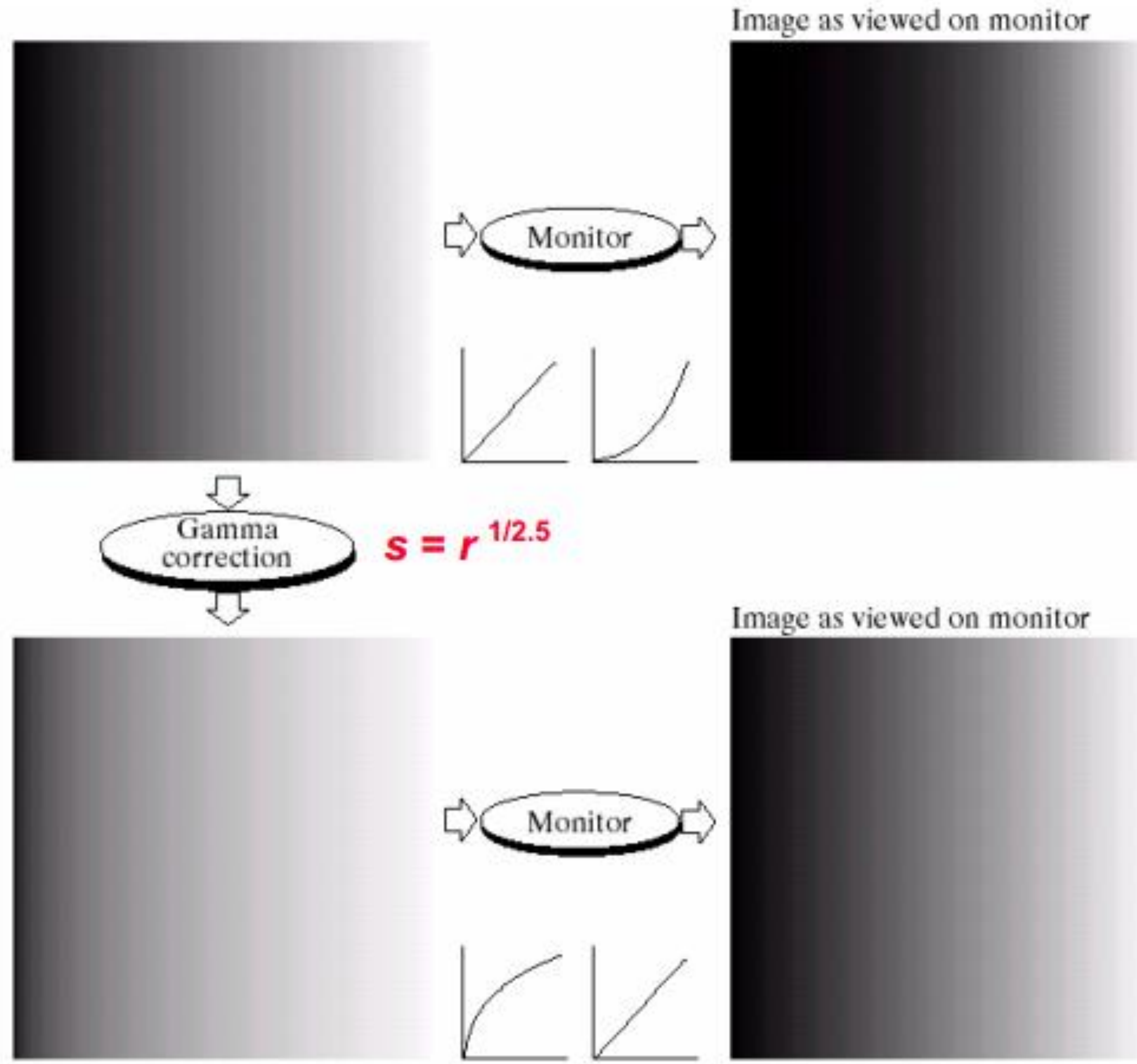


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Map a narrow range of dark input values Into a wider range of output values.

Map a wide range of light input values into a narrower range of output values.

Power_Law Transformation



a b
c d

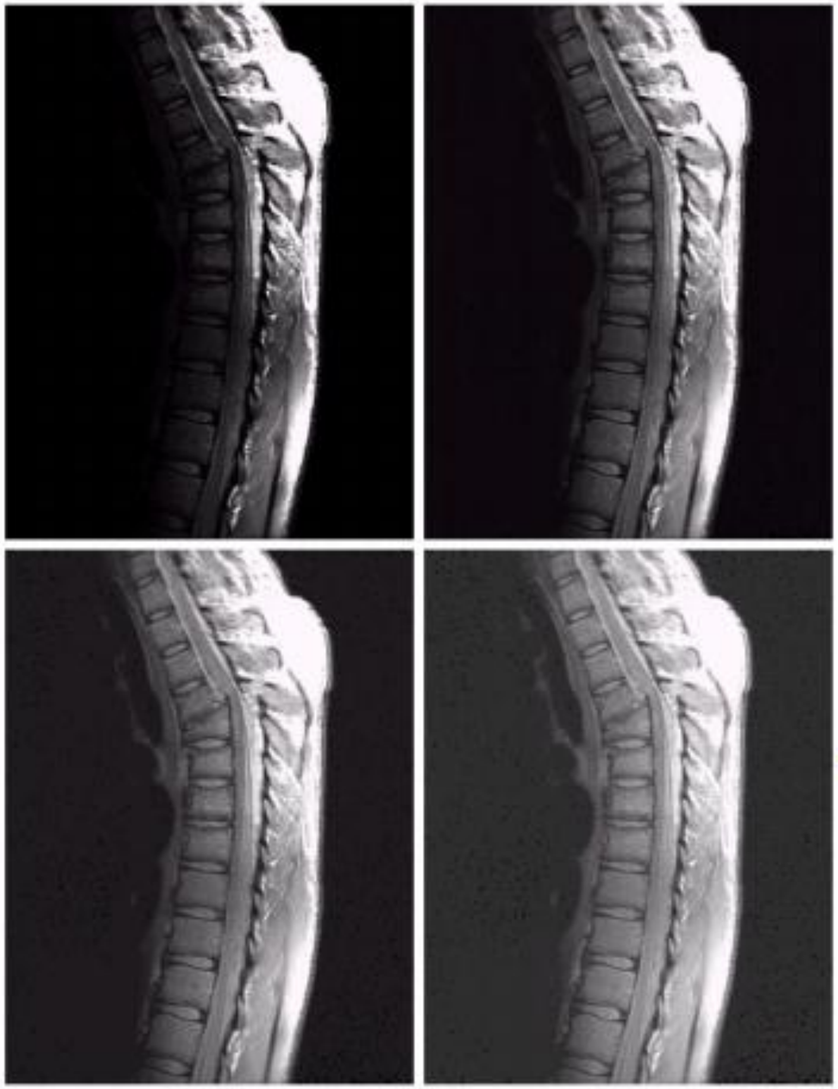
FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

A variety of devices for image capture, printing and display respond according to power law.

Such systems tend to produce images that are darker than intended.

Example:

Original image is predominantly dark!



a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image 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 for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

$s = r^{0.6}$

$s = r^{0.4}$

$s = r^{0.3}$

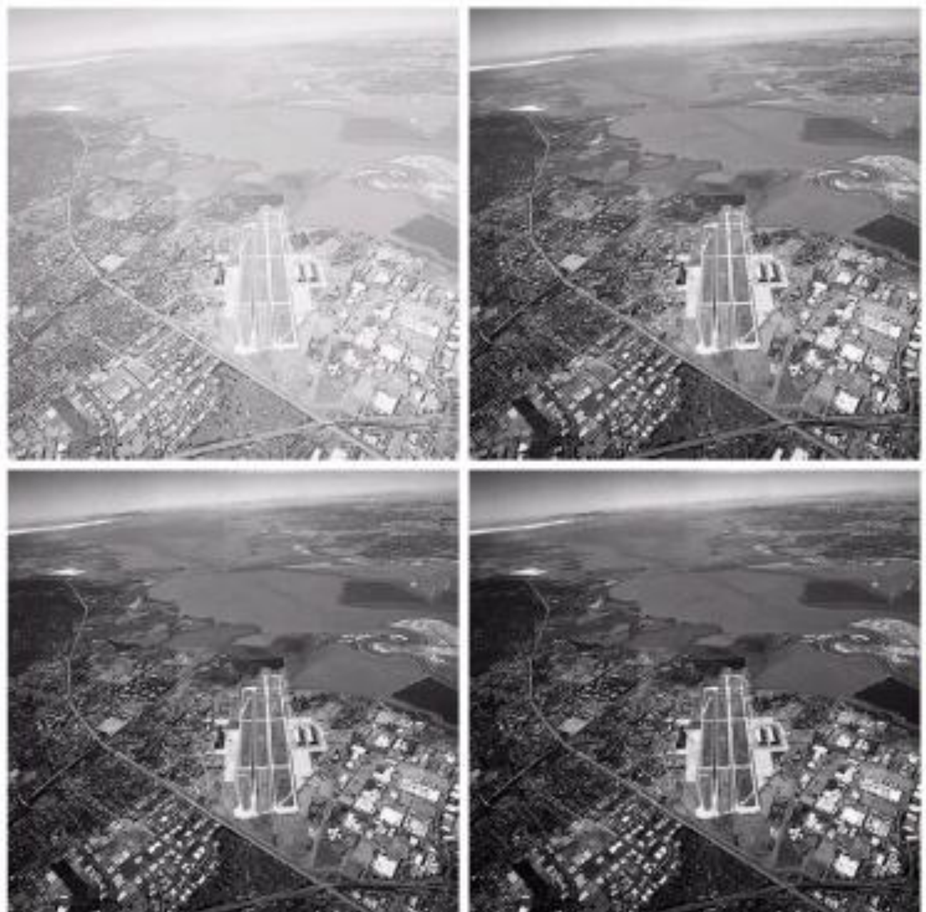
Example:

Original image has a washed-out appearance!

$s = r^{3.0}$

a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)



$s = r^{4.0}$

$s = r^{5.0}$

piecewise Linear transformation functions.

1. **Contrast stretching and thresholding**

- One of the simplest **piecewise linear** functions is a contrast-stretching transformation.
- **Low-contrast** images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition.
- The idea behind contrast stretching is to **increase** the **dynamic range** of the gray levels in the image being processed.

piecewise Linear transformation functions.

1. **Contrast stretching and thresholding**

- Is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the display device.

- For example:

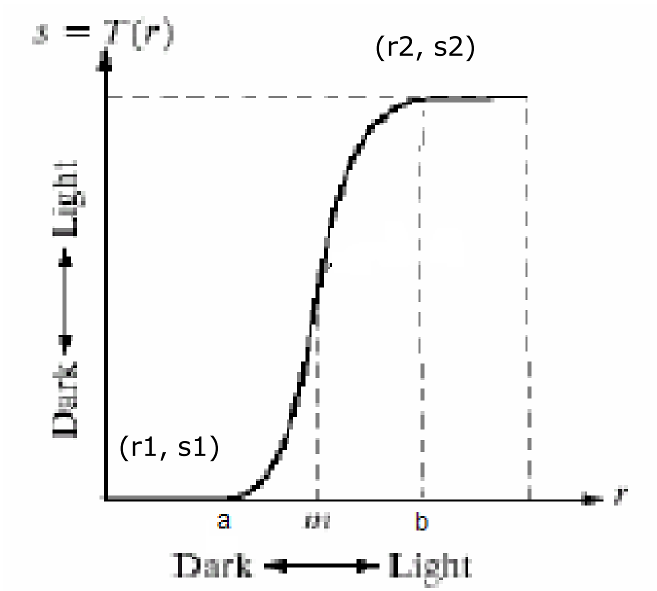
Input Image: 8-bit image have range [80 – 150]

After Contrast Stretching: the new range becomes [0 – 255] using the following equation:

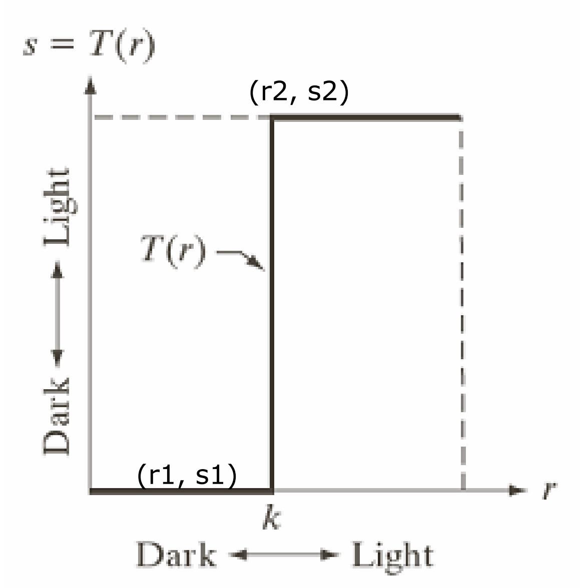
$$I_{new} = (I - Min)\frac{NewMax - NewMin}{Max - Min} + NewMin$$

piecewise Linear transformation functions.

1. Contrast stretching and thresholding



Contrast stretching



Thresholding:

Assume that
a: rmin,
b:rmax,
k : intensity

Contrast stretching: (r1,s1)=(rmin,0) , (r2,s2)=(rmax,L-1)
Thresholding: (r1,s1)=(k,0) , (r2,s2)=(k,L-1)

piecewise Linear transformation functions.

Contrast stretching

contrast stretching, which means that the bright pixels in the image will become brighter and the dark pixels will become darker, this means : higher contrast image.

piecewise Linear transformation functions.

Contrast stretching

Example☺

Image (r)

Image (s) after applying T (contrast stretching)

Notice that the intensity transformation function T, made the pixels with dark intensities darker and the bright ones even more brighter, this is called contrast stretching>

piecewise Linear transformation functions.

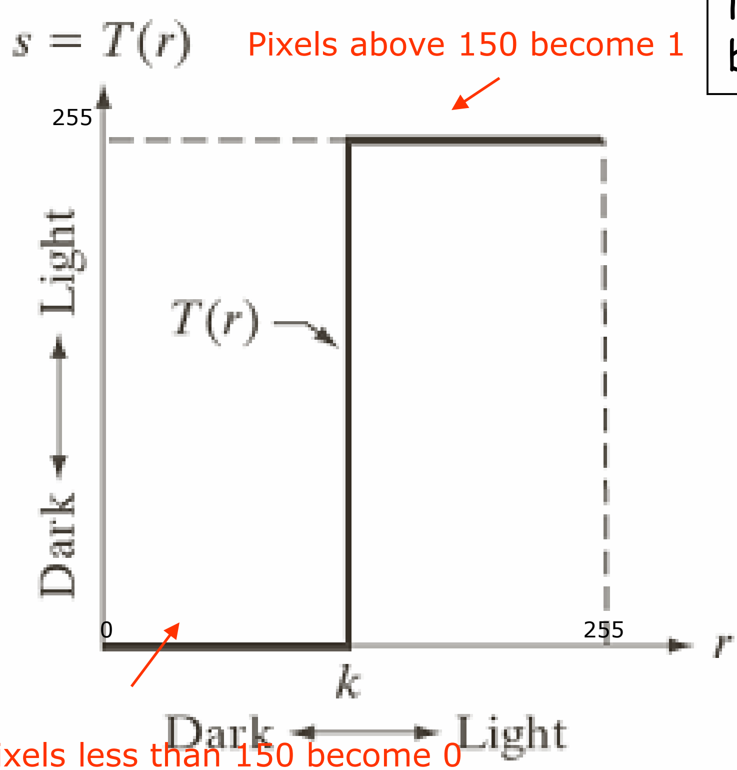
thresholding

Remember that:

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

Or

$$s = T(r)$$



Example: suppose m= 150 (called threshold),

if r (or pixel intensity in image f الصورة الاصلية) is above this threshold it becomes 1 in s (or pixel intensity in image g الصورة بعد التعديل), otherwise it becomes zero.

$$T = \begin{cases} \text{If } f(x,y)>150;\ g(x,y)=1 \\ \text{If } f(x,y)<150;\ g(x,y)=0 \end{cases}$$

Or simply…

$$T = \begin{cases} \text{If } r>150;\ s=1 \\ \text{If } r<150;\ s=0 \end{cases}$$

This is called thresholding, and it produces a binary image!

piecewise Linear transformation functions.

thresholding

Image (r)

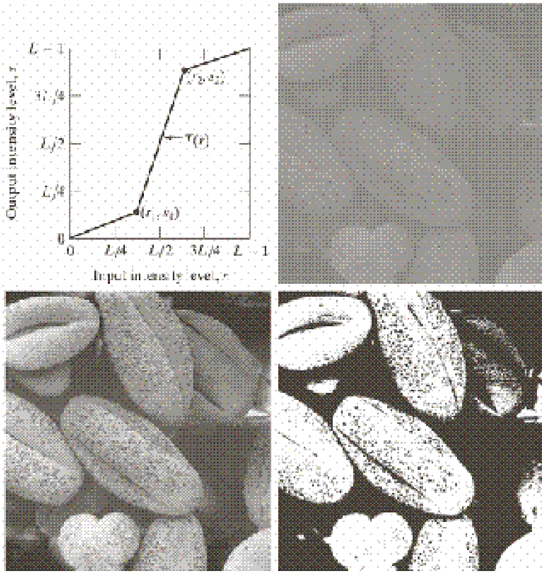
Image (s) after applying T (Thresholding)

Notice that the intensity transformation function T, convert the pixels with dark intensities into black and the bright pixels into white. Pixels above threshold is considered bright and below it is considered dark, and this process is called thresholding.

piecewise Linear transformation functions.

Application on Contrast stretching and thresholding

8-bit image with low contrast



After contrast stretching

$(r1,s1)=(r_{min},0)$, $(r2,s2)=(r_{max},L-1)$

Thresholding function

(r1,s1)=(m,0) , (r2,s2)=(m,L-1)

m : mean intensity level in the image

piecewise Linear transformation functions.

Exercise on Contrast stretching and thresholding

Exercise:

the following matrix represents the pixels values of a 8-bit image (r) , apply thresholding transform assuming that the threshold m=95, find the resulting image pixel values.

Image (r)

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

solution:

Image (s)

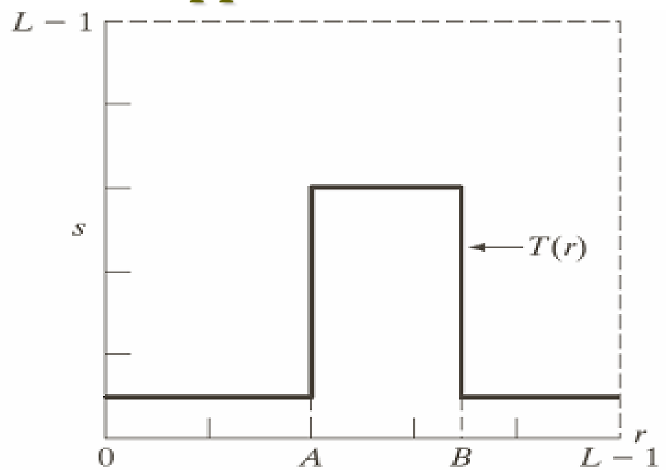
piecewise Linear transformation functions.

2. Intensity-Level Slicing (gray level slicing)

Highlighting a specific range of intensities in an image.

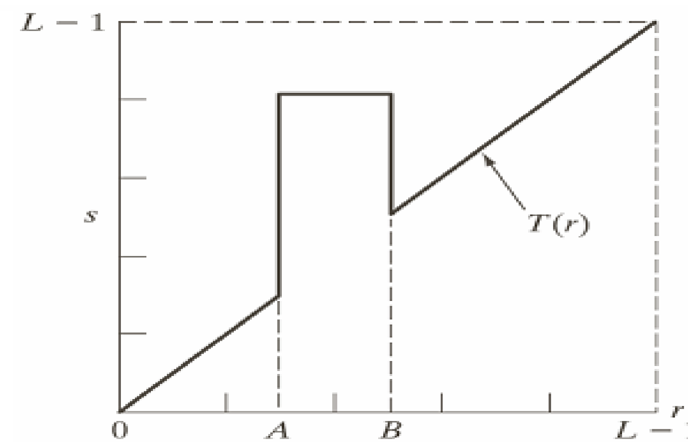
Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images

Approach 1



display in one value(e.g white) all the values in the range of interest , and in another (e.g black) all other intensities

Approach 2



Brightens or darkens the desired range of intensities but leaves all other intensity levels in the image unchanged. It preserves the background

piecewise Linear transformation functions.

2. Intensity-Level Slicing (gray level slicing)

example: approach 1

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 150) convert it in the new image into 255 (white). Otherwise convert it to 0 (black).

Solution:

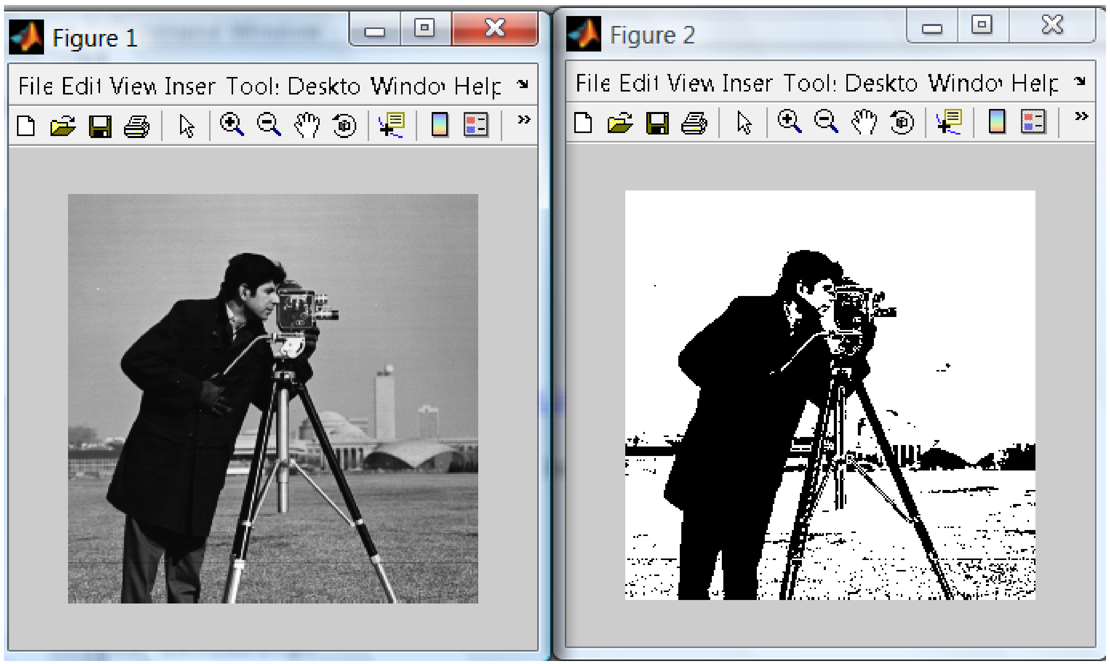
```
x=imread('cameraman.tif');
y=x;
[w h]=size(x);
for i=1:w
    for j=1:h
        if x(i,j)>=100 && x(i,j)<=200
              y(i,j)=255;
        else
              y(i,j)=0;
        end
    end
end
figure, imshow(x);
figure, imshow(y);
```

piecewise Linear transformation functions.

2. Intensity-Level Slicing (gray level slicing) –

example: approach 1

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 150) convert it in the new image into 255 (white). Otherwise convert it to 0 (black).



piecewise Linear transformation functions.

2. Intensity-Level Slicing (gray level slicing)

example: approach 2

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 150) convert it in the new image into 255 (white). Otherwise it leaves it the same.

Solution:

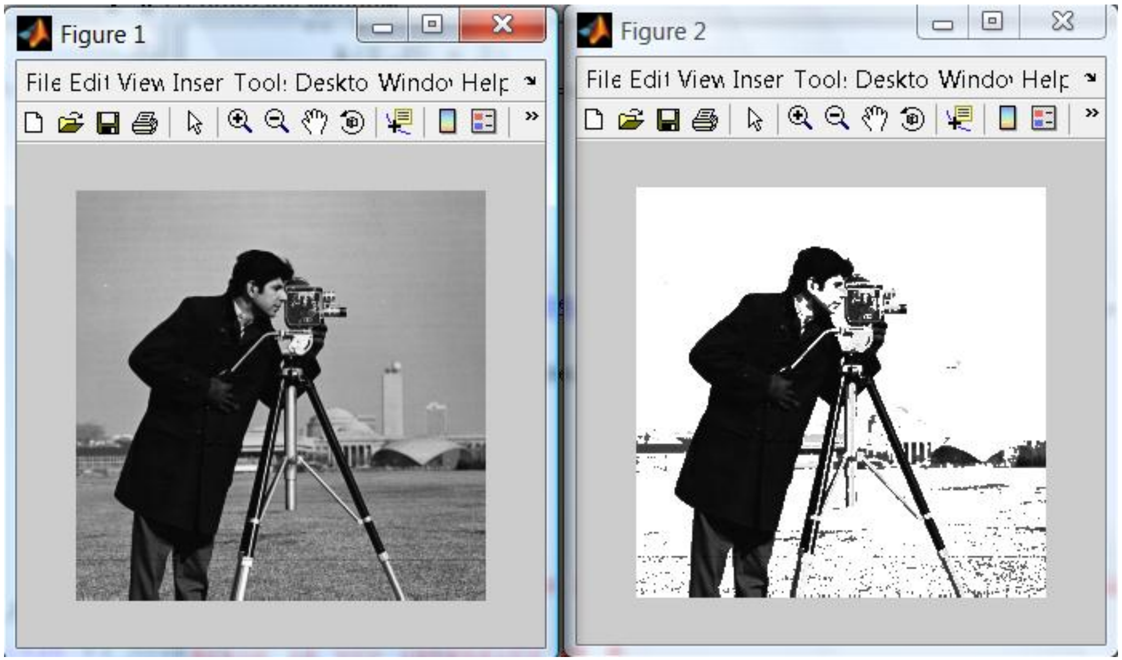
```
x=imread('cameraman.tif');
y=x;
[w h]=size(x);
for i=1:w
    for j=1:h
        if x(i,j)>=100 && x(i,j)<=200
              y(i,j)=255;
        else
              y(i,j)=x(i,j);
        end
    end
end
figure, imshow(x);
figure, imshow(y);
```

piecewise Linear transformation functions.

2. Intensity-Level Slicing (gray level slicing)

example: approach 2

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 150) convert it in the new image into 255 (white). Otherwise it leaves it the same.



piecewise Linear transformation functions.

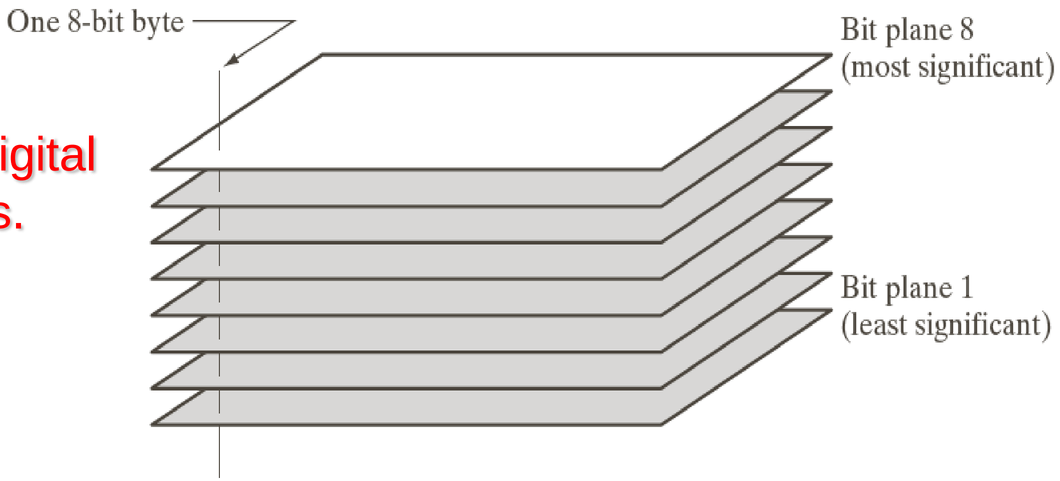
2. Intensity-Level Slicing (gray level slicing)

Homework

example: apply intensity level slicing (approch2) in Matlab to read moon image , then If the pixel intensity in the old image is between (0 → 20) convert it in the new image into 130.

Piecewise Linear Transformation Functions.
3. Bit-Plane Slicing

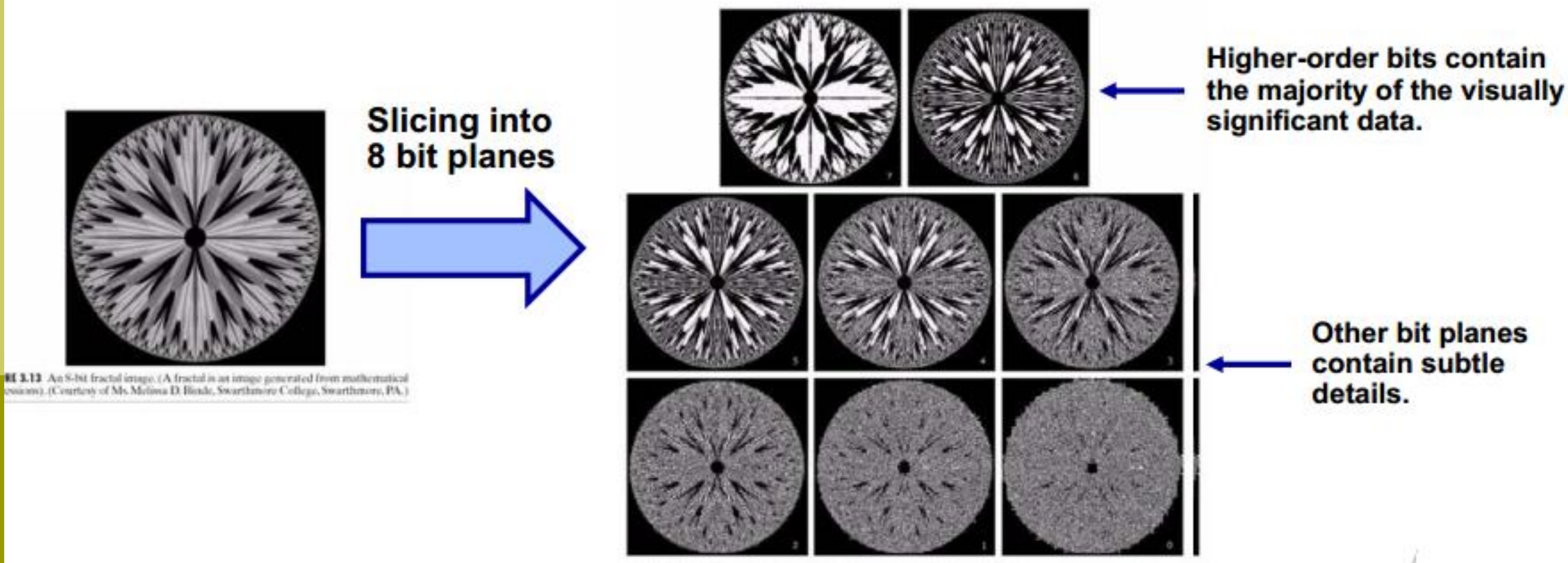
Remember that pixels are digital numbers composed of 8 bits.



8-bit Image composed of 8 1-bit planes

piecewise Linear transformation functions.

3. Bit-Plane Slicing



Bit-plane slicing :

- Suppose that each pixel in an image is represented by **8 bits**.
- Imagine that the image is composed of **eight 1-bit planes,** ranging from bit-plane 0 to bit-plane 7
- **plane 0** contains all the lowest order bits and **plane 7** contains all the high-order bits.
- The **higher-order** bits (especially the top four) contain the majority of the **visually significant data**.
- The other bit planes contribute to more **subtle details** in the image.
- **Separating** a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel. Also, this type of decomposition is useful for image **compression**.

3. Bit-Plane Slicing

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

3. Bit-Plane Slicing

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

3. Bit-Plane Slicing (example)

We have to use bit get and bit set to extract 8 images;

100			

0 1 1 0 0 1 0 0

Image of bit1:
0000000**0**

0			

Image of bit2:
000000**0**0

0			

Image of bit3:
00000**1**00

4			

Image of bit4:
0000**0**000

0			

Image of bit5:
000**0**0000

0			

Image of bit6:
00**1**00000

32			

Image of bit7:
0**1**000000

64			

Image of bit8:
00000000

0			

3. Bit-Plane Slicing- programmed

example: apply bit-plane slicing in Matlab to read cameraman image , then extract the image of bit 6.

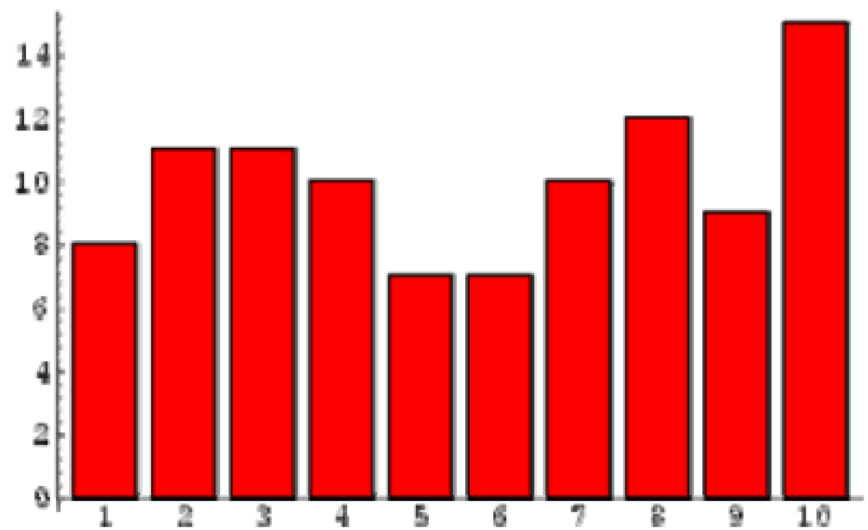
Solution:

```
x=imread('cameraman.tif');
y=x*0;
[w h]=size(x);
for i=1:w
    for j=1:h
        b=bitget(x(i,j),6);
        y(i,j)=bitset(y(i,j),6,b);

    end
end

figure, imshow(x);
figure, imshow(y);
```

Histogram?

What is a histogram?

- A graph indicating the number of times each gray level occurs in the image, i.e. frequency of the brightness value in the image
- The image histogram carries important information about the image content
- Algorithm:
  - Assign zero values to all elements of the array h_f;
  - For all pixels (x,y) of the image f, increment $h_f[f(x,y)]$ by 1.



Histogram?

- Histogram of a digital image

$$h(r_k) = n_k$$

Where:

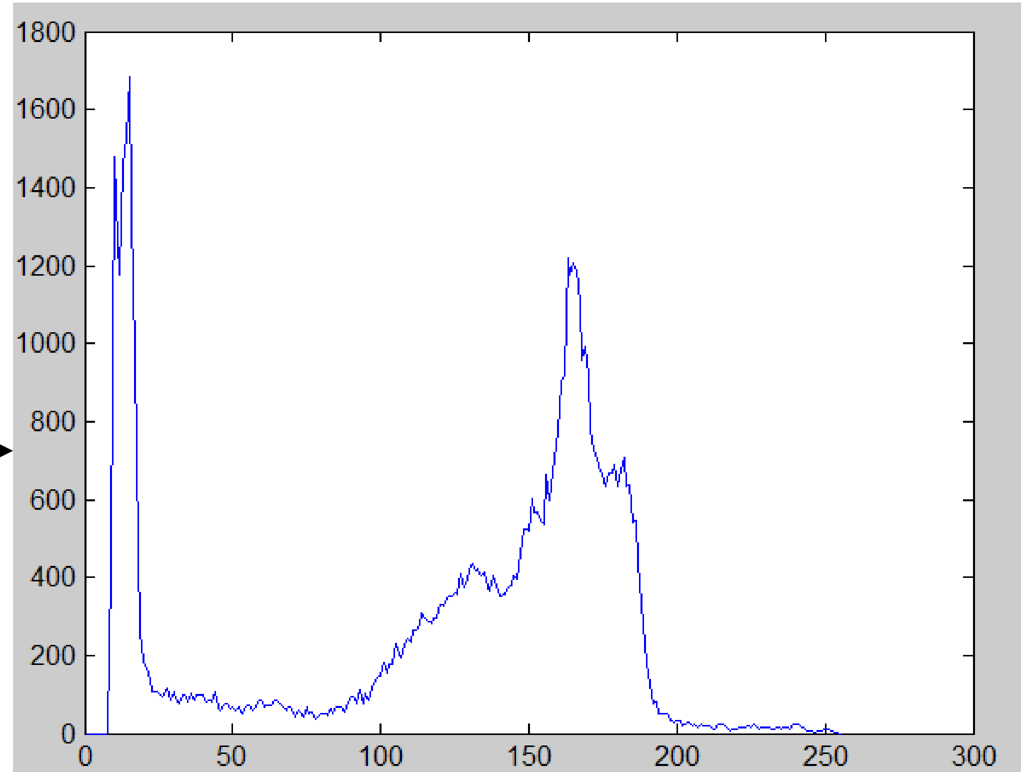
r_k : kth gray level

n_k : # of pixels with having gray level rk

- We manipulate Histogram for image enhancement.
- Histogram data is useful in many applications like image compression and segmentaion.

Histogram of the image:

histogram



gray levels   هو تمثيل لعدد البكسل في كل قيمة لونية من درجات

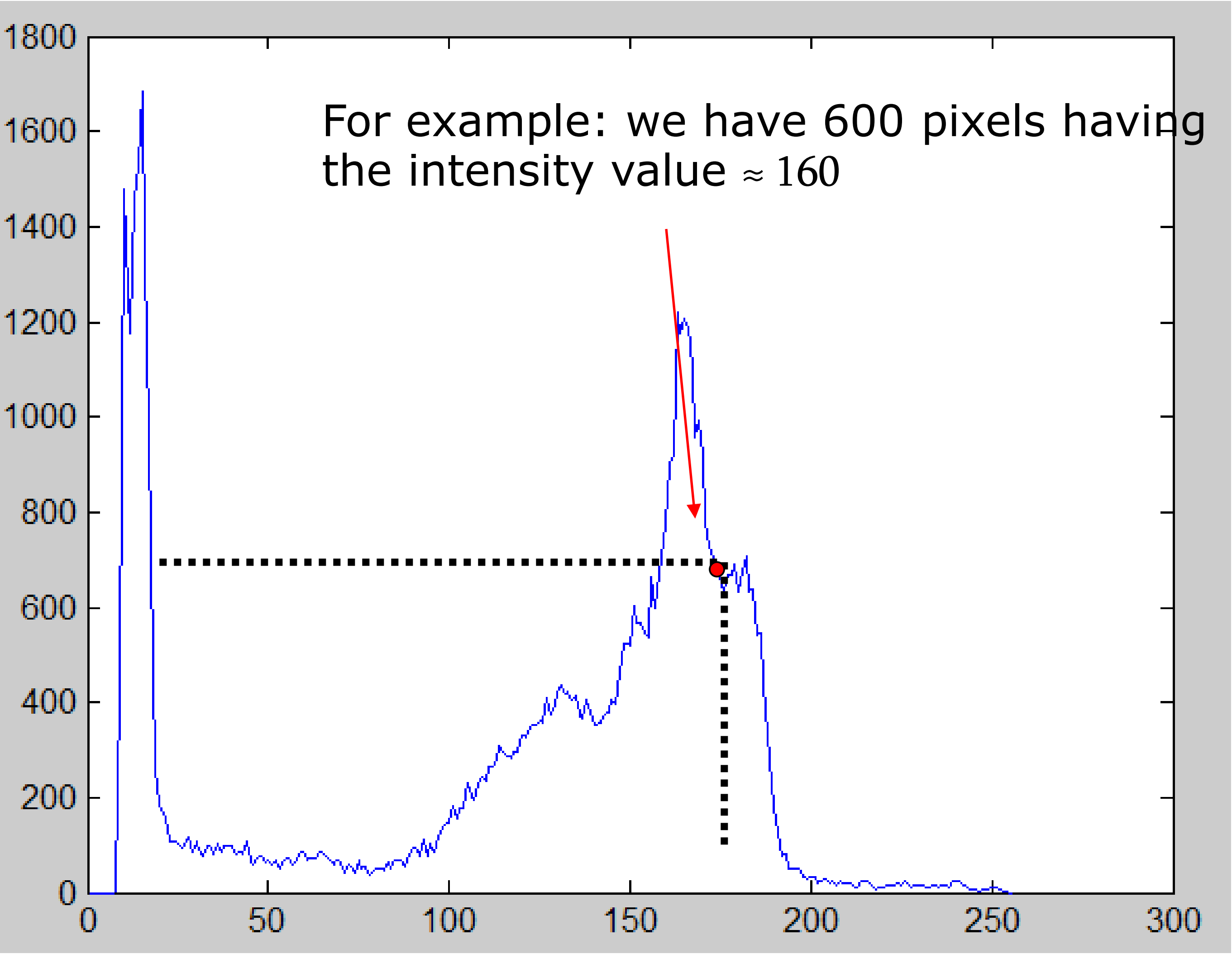
$\boldsymbol{h(r_k) = n_k}$

Where:

r_k : kth gray level

n_k : # of pixels with having gray level rk

Histogram of the image:



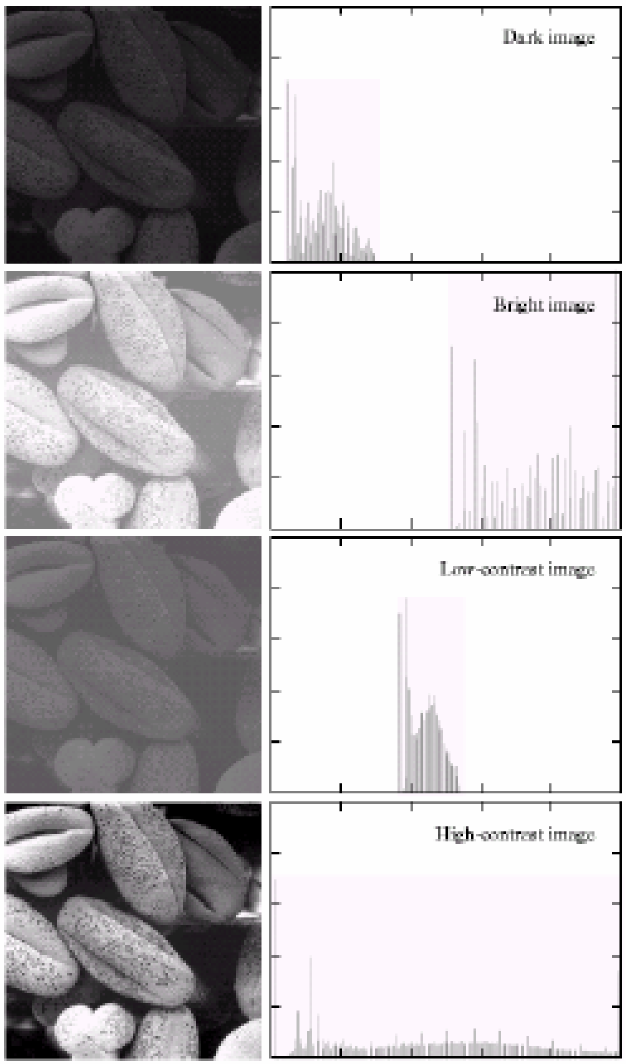
Histogram of the image:

Dark image

Bright image

Low contract image

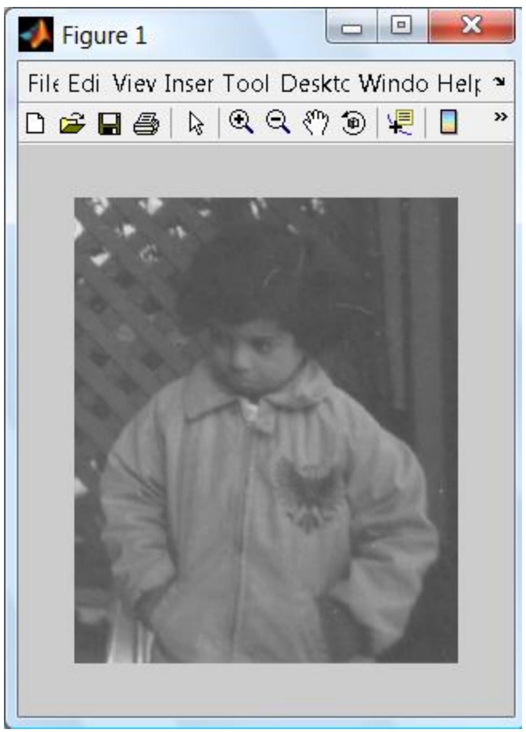
High contrast image

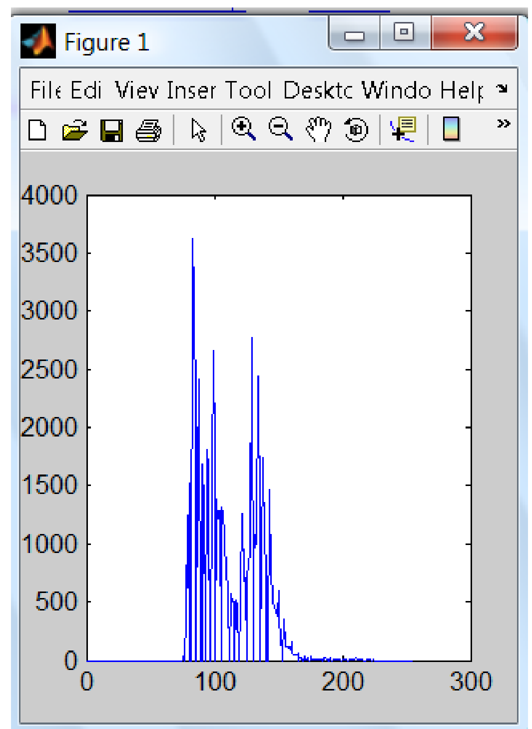


Compare the four images and their histograms.

Histogram equalization of the image:

We have this image in matlab called pout.tif, when we plot its histogram it is showed like this:





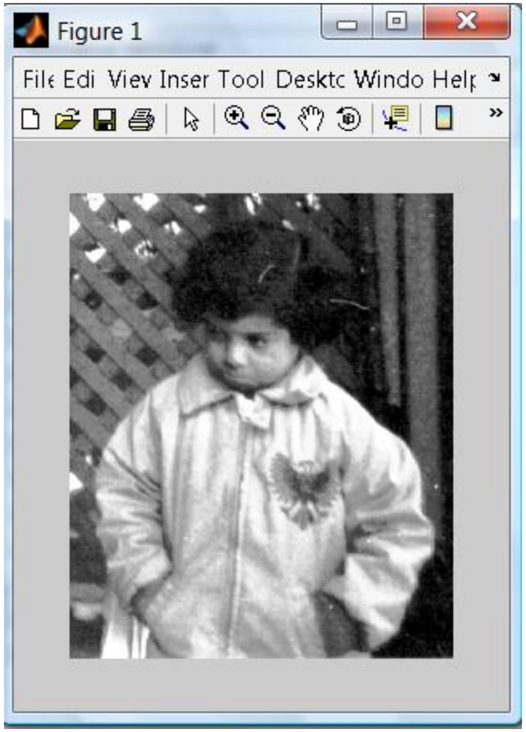
Notice that the pixels intensity values are concentrated on the middle

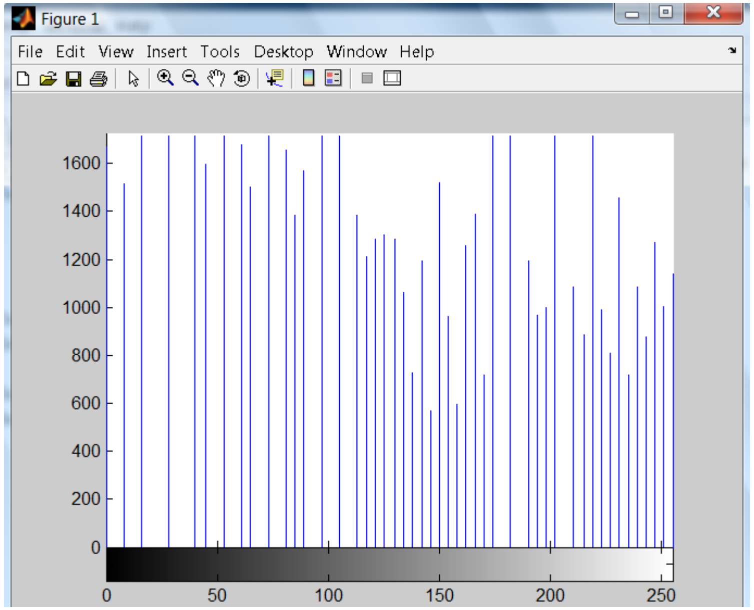
(low contrast)

Histogram equalization of the image:

histogram equalization : is the process of adjusting intensity values of pixels.

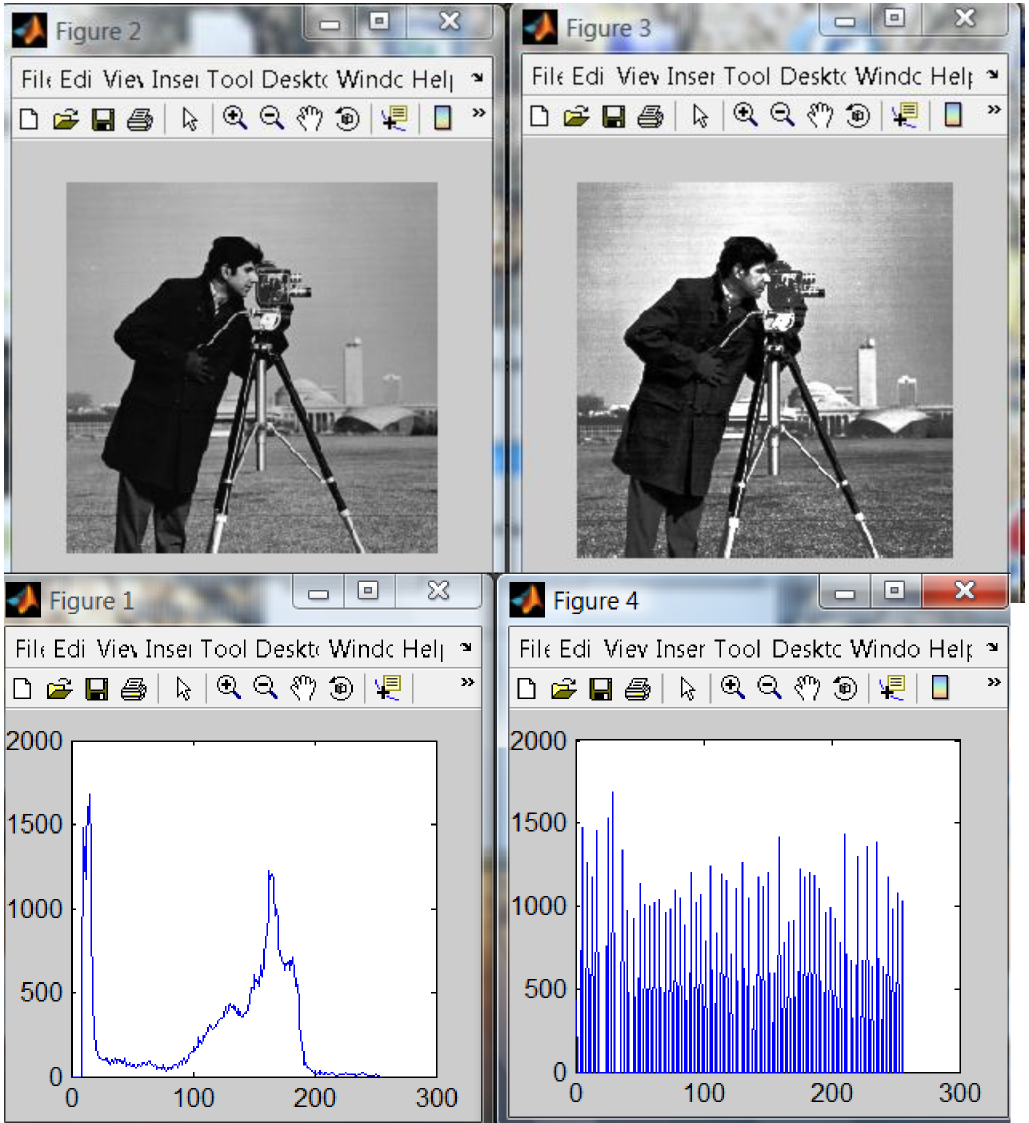
Im matlab : we use **histeq** function





Histogram produces pixels having values that are distributed throughout the range

Histogram equalization of the image:



Notice that histogram equalization does not always produce a good result

Equalization (mathematically)

$g(x) = (L/n).\ T(X) - 1$

Where,

$G(X)$: the new image after equalization

L: No of gray levels 2^n

n: No of pixels

$T(x)$: cumulative sum of each gray level

Equalization (mathematically)

L graylevels	X عدد البكسل لكل Graylevel	T(X)مجموع تراكمي للبكسل	G(x)
0	1	1	٠
1	3	4	٠
2	5	9	١
3	6	15	٢
4	6	21	٤
5	6	27	٥
6	2	29	٦
7	3	32	٧

٨ عدد ال graylevel

عدد البكسلات الكلي No of pixels

Assume that we have (3bits per pixels) or 8 levels of grayscale, and we want to equalize the following image example.

G(x)=(L/n). T(X) -1

=(8/32). T(x) -1

Example

Table 2.1: Example of histogram equalization.

Original value x	Frequency $\#(x)$	Cumulative frequency $t(x)$	New value $f(x)$
0	1	1	0
1	9	10	2
2	8	18	5
3	6	24	7
4	1	25	7
5	1	26	8
6	1	27	8
7	1	28	8
8	2	30	9
9	0	30	9

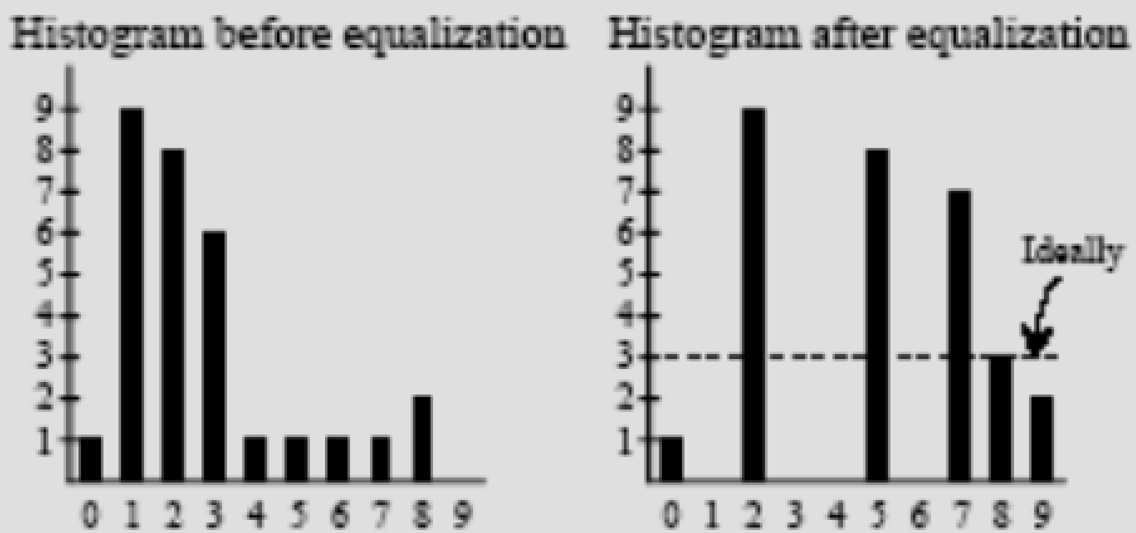


Figure 2.10: Example of histogram equalization.

ENHANCEMENT USING ARITHMETIC/LOGIC OPERATIONS

- **Arithmetic operations**
  - Subtraction
  - Addition

  Most useful in image enhancement

  - Multiplication: used as a masking operation
  - Division

- **Logical operations**
  - AND
  - OR

  Used for masking

  - NOT

  For these operations, gray scale pixel values are processed as strings of binary numbers.

  - Frequently used in conjunction with morphological operations.

And/Or Masks

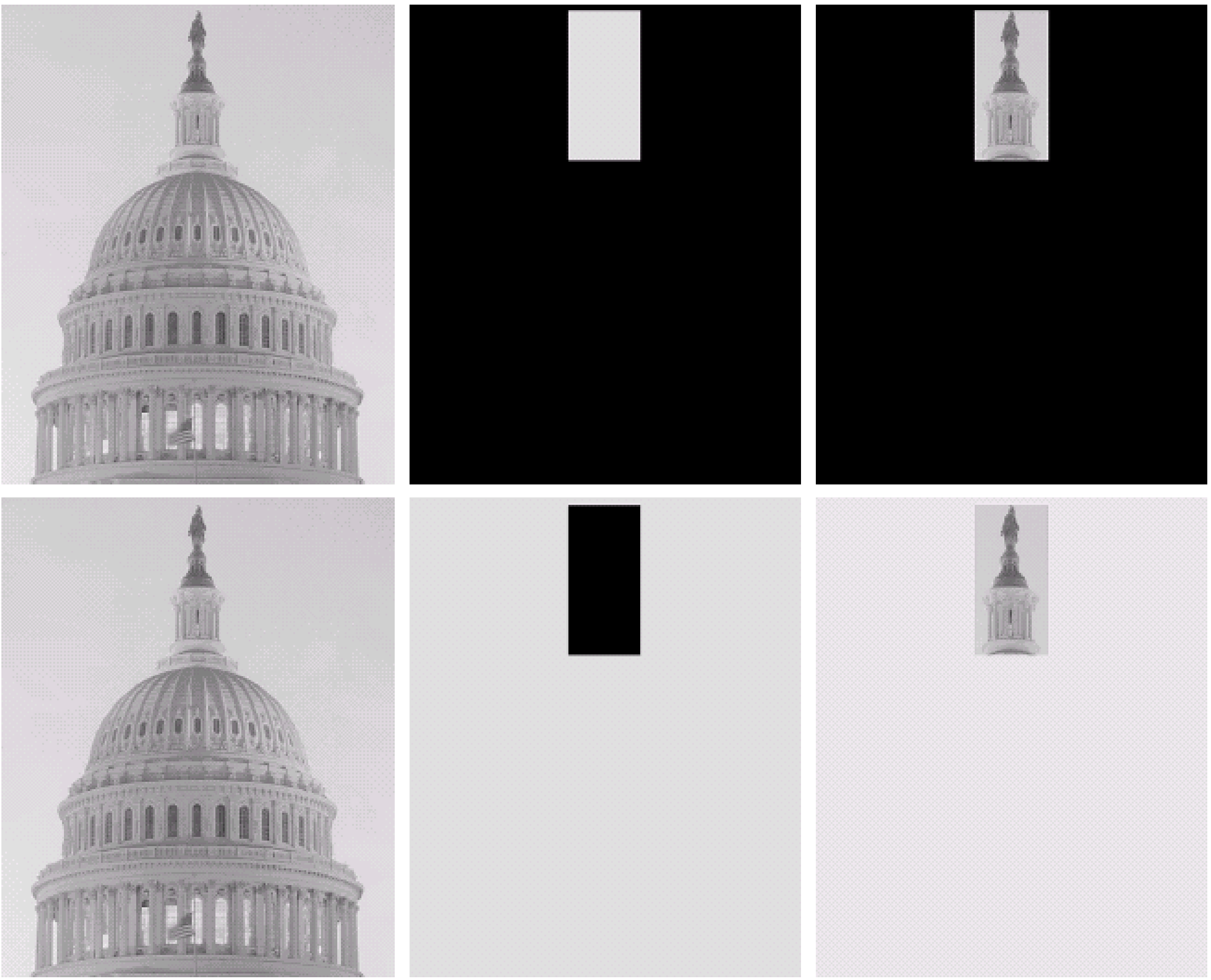


FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

Image Subtraction: Example

Subtraction of two images is often used to detect motion. Consider the case where nothing has changed in a scene; the image resulting from subtraction of two sequential images is filled with zeros-a black image. If something has moved in the scene, subtraction produces a nonzero result at the location of the movement.

Image Subtraction: Motion Tracking

background image

live image

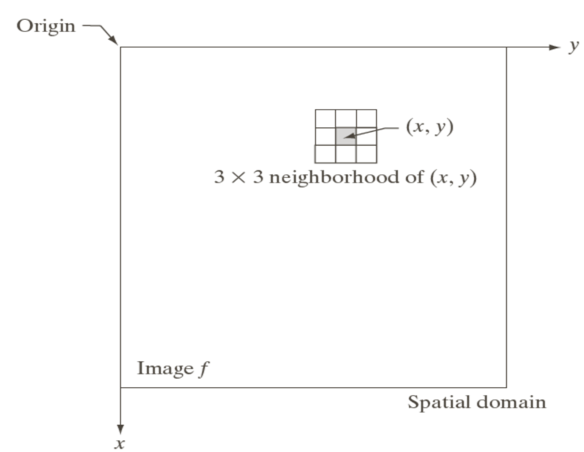
difference image

Spatial filters

Remember that types of neighborhood:

- **intensity transformation**: neighborhood of size 1x1
- **spatial filter** (or mask ,kernel, template or window): neighborhood of larger size , like 3*3 mask

- It is the **procedure** of moving the location of neighbourhood and performing a predefined operation. so the spatial filter mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated



Spatial Filters

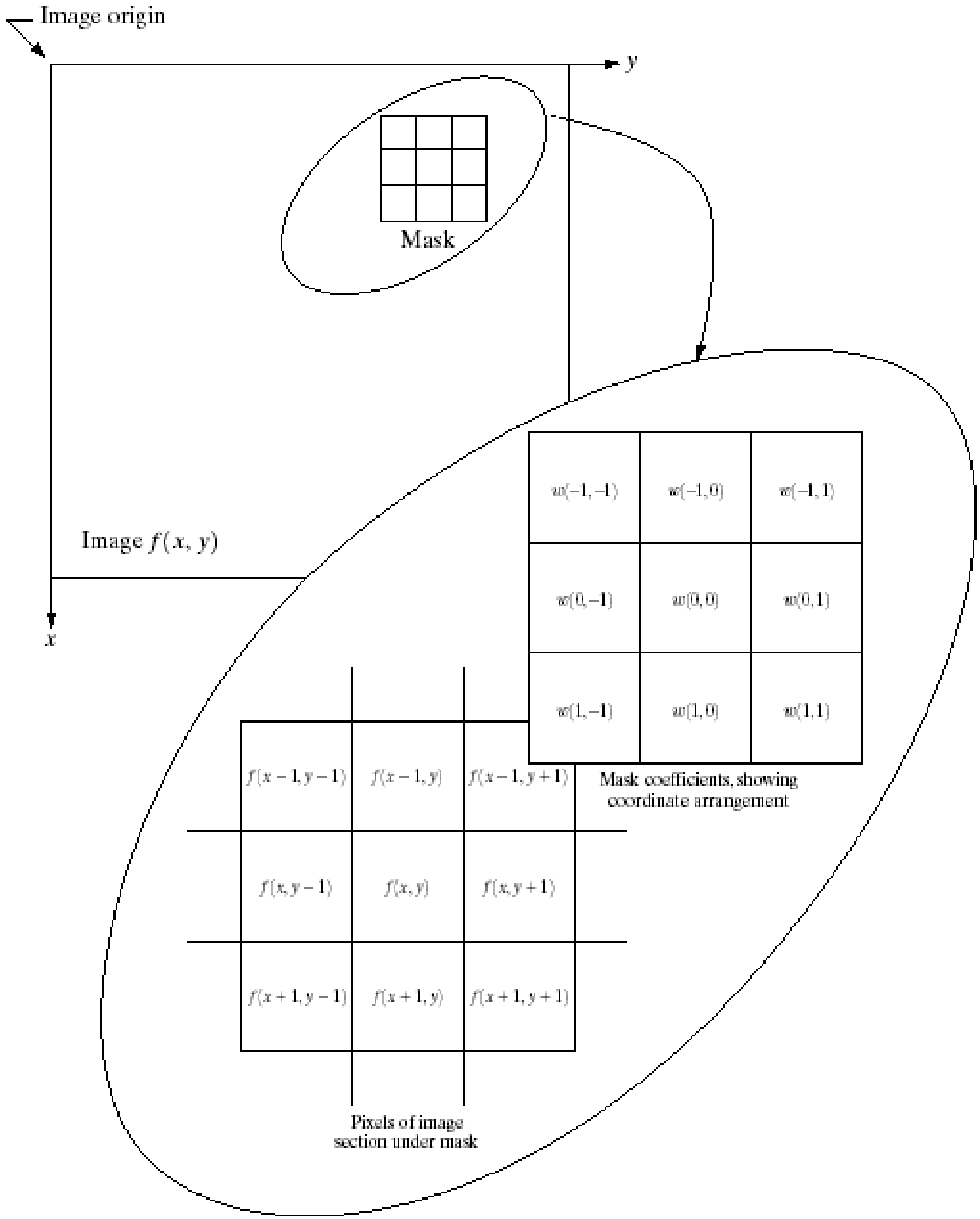


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Spatial filters

- The filter mask is moved from point to point in an image.
- At each point (x,y), the response of the filter is calculated.
- Linear spatial filtering
  - 3x3 mask
  - $R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + w(0,0)f(x,y) + \ldots$
    $+ w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$
  - $w(0,0)$ coincides with $f(x,y)$ indicating that the mask is centered at $f(x,y)$.
- Our focus will be on masks of odd sizes.
- Image of size MxN, filter mask of size mxn.

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

where $a = (m-1)/2$ and $b = (n-1)/2$

Spatial filters

Origin $f(x, y)$

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

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

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

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Spatial filters

1)Smoothing filters - low pass

2)Sharpening filters – high pass

Spatial filters : Smoothing (low pass)

Use for:

1)blurring

2) noise reduction.

How it works? The value of every pixel is replaced by the average of the gray levels in the neighborhood.

Type of smoothing filters:

1. Standard average
2. weighted average.

linear

3. Median filter

Order statistics

Spatial filters : Smoothing
linear smoothing : averaging kernels

The output (response) of a smoothing, linear spatial filter is simply the **average** of the pixels contained in the neighborhood of the filter mask.

Desirable effect: the most application of smoothing is **noise reduction**, because random noise typically consists of sharp transitions in gray levels,

Undesirable effect**:** the undesirable side effect is **blur edges.** edges (which almost always are desirable features of an image) also are characterized by sharp transitions in graylevels.

Standard average

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline\end{array}$$

A box filter

weighted average.

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

The basic strategy is to reduce blurring.

Spatial filters : Smoothing
Standard and weighted Average- example

110	120	90	130
91	**94**	98	200
90	91	99	100
82	96	85	90

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

Standard averaging filter:

(110 +120+90+91+94+98+90+91+99)/9 =883/9 = 98.1

Weighted averaging filter:

(110 +2 x 120+90+2 x 91+4 x 94+2 x 98+90+2 x 91+99)/16 =

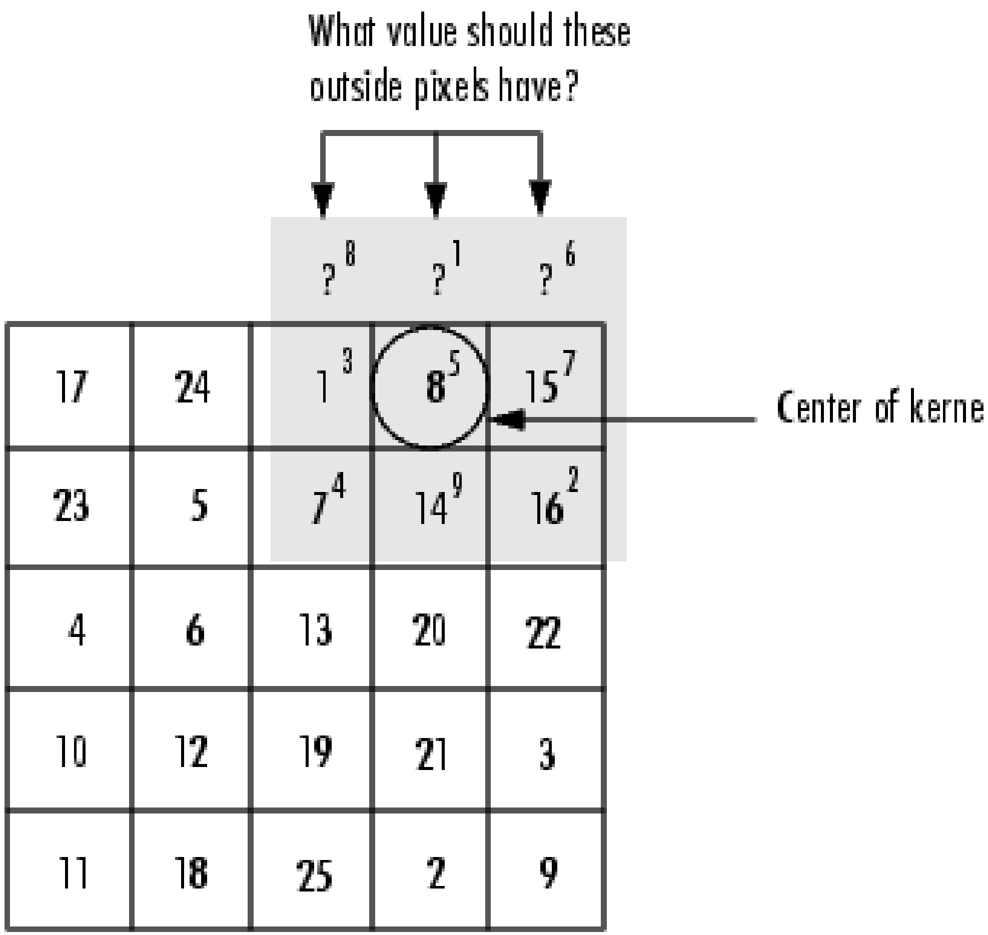
General implementation:

General implementation for filtering an

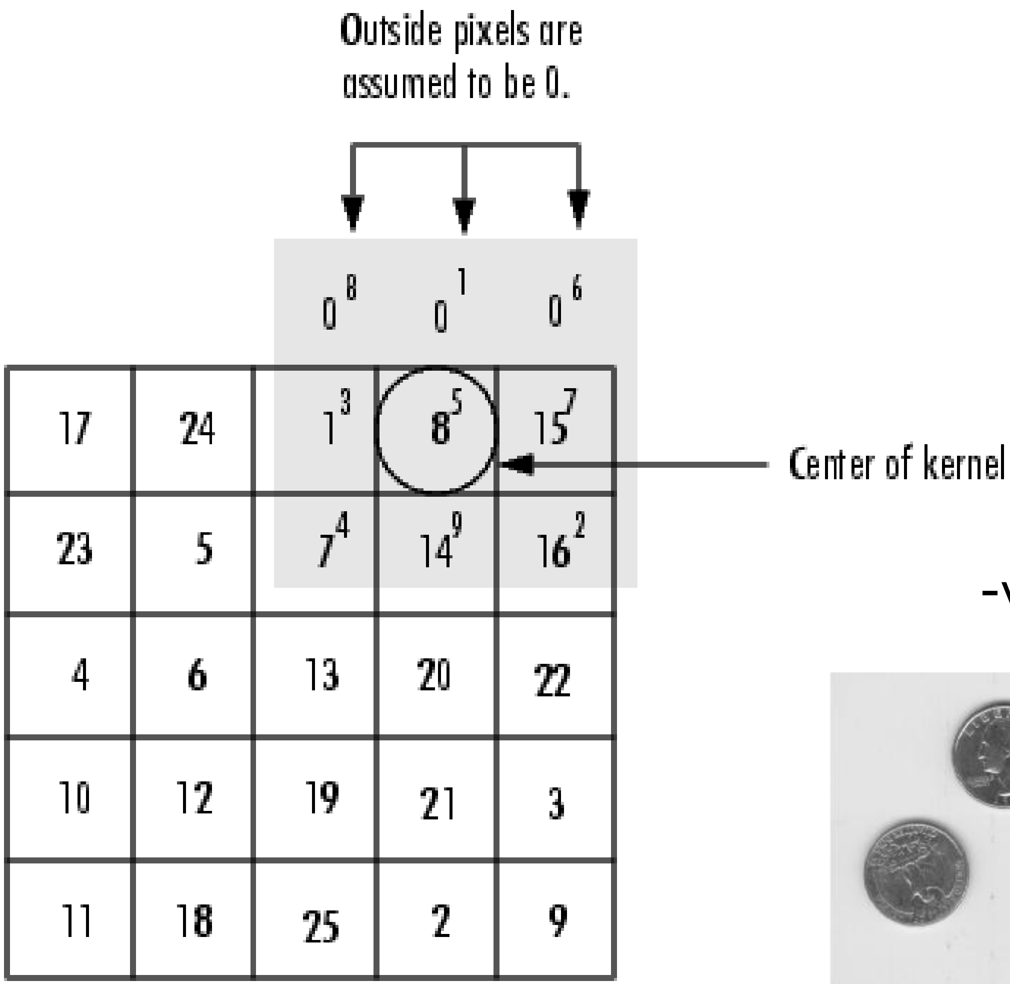
MxN image with a weighted averaging filter of size mxn (m and *n odd) is* given by the expression:

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

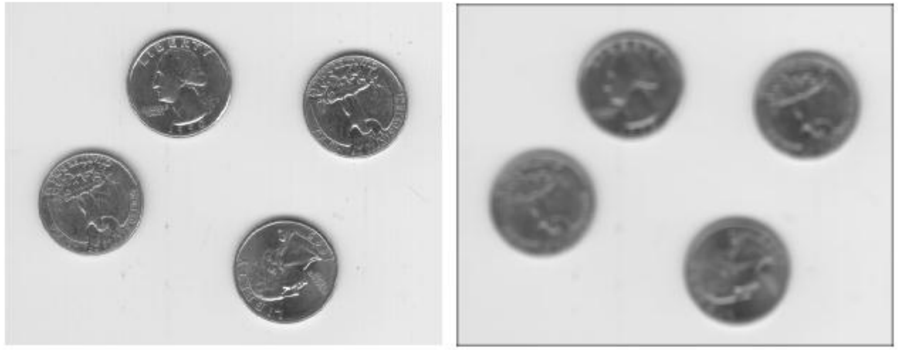
What happens when the Values of the Kernel Fall Outside the Image??!



First solution :Zero padding,

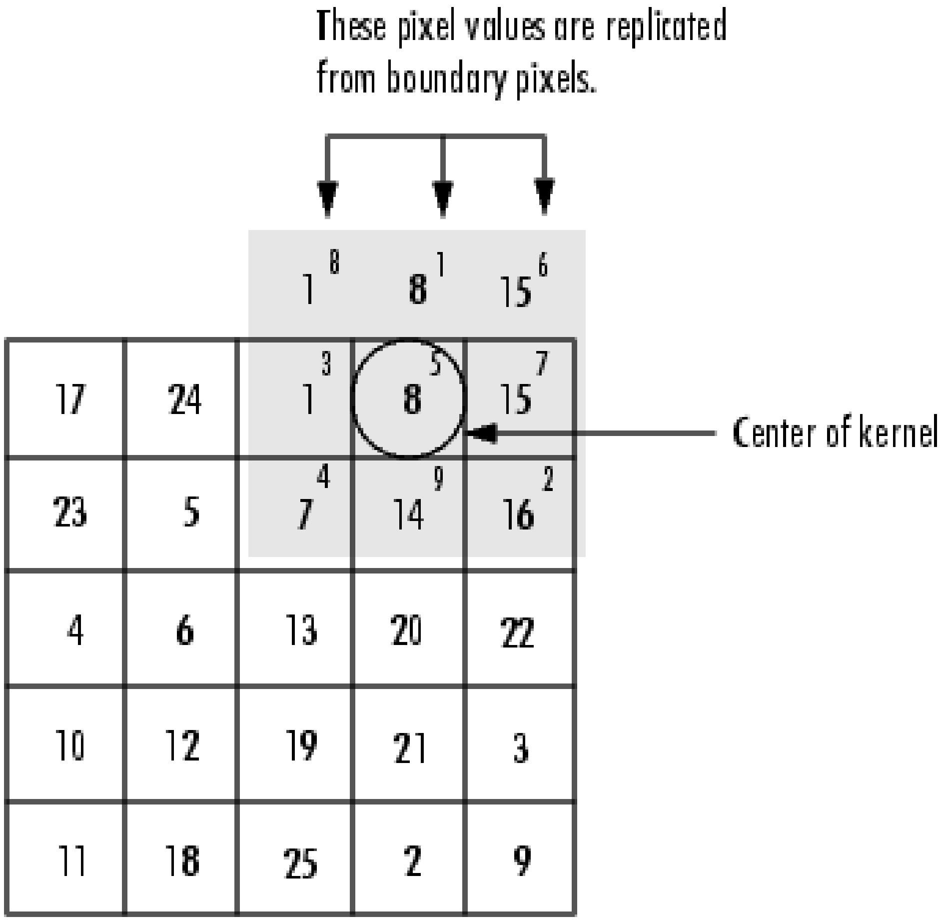


-ve: black border



Original Image   Filtered Image with Black Border

border padding



Spatial filters : Smoothing
Averaging effects: blurring + reducing noise

Original image

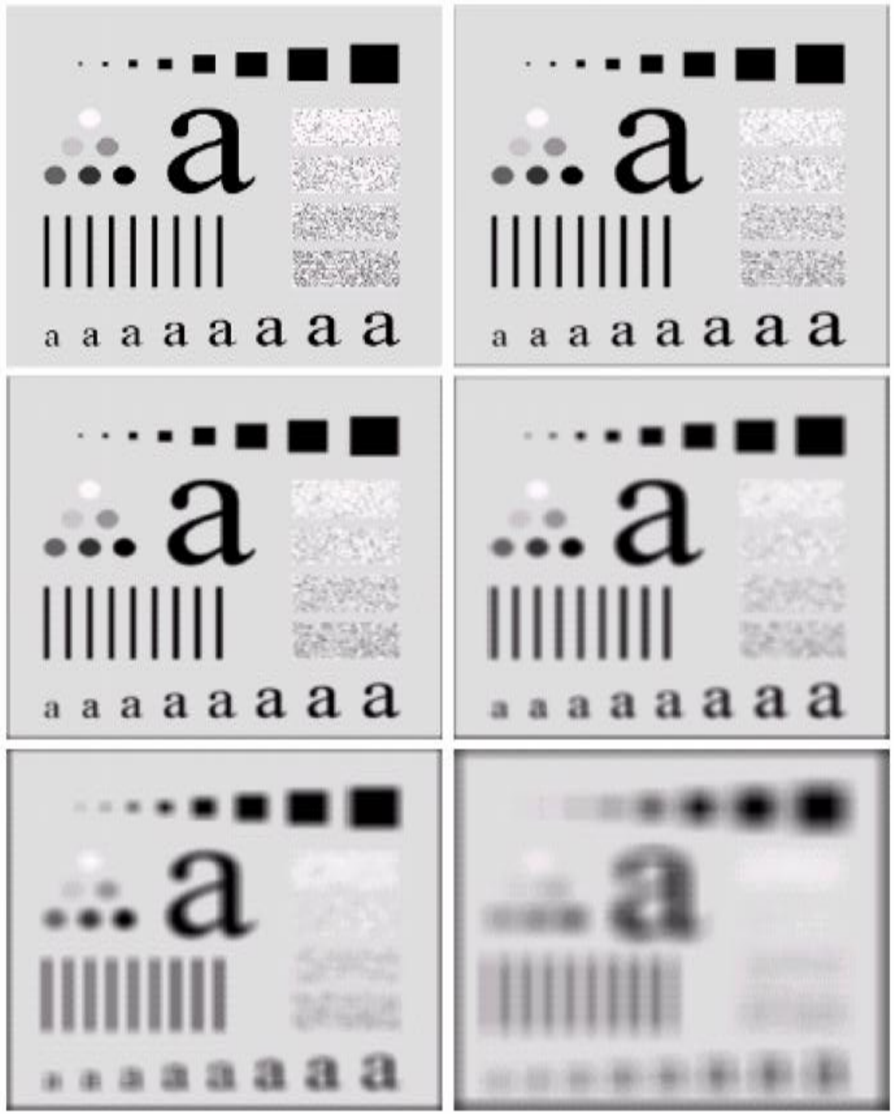
3 x 3 averaging

5 x 5 averaging

9 x 9 averaging

15 x 15 averaging

35 x 35 averaging



Spatial filters : Smoothing order statistics: Median filter

110	120	90	130
91	94	98	200
90	95	99	100
82	96	85	90

becomes

95

Steps:

1. Sort the pixels in ascending order:

90,90, 91, 94, 95, 98, 99, 110, 120

2. replace the original pixel value by the median :

95

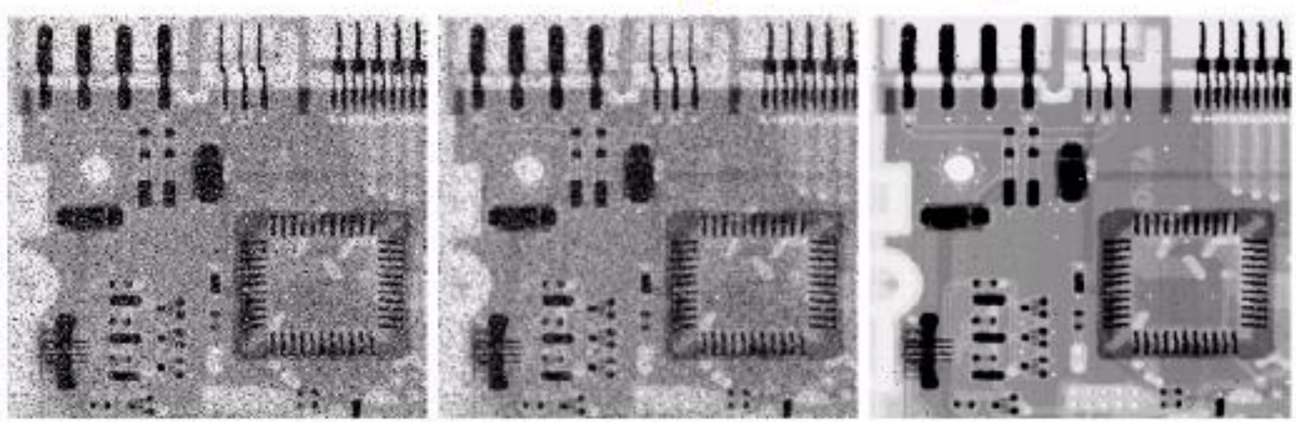
the ***median*** *filter* is based on ordering (**ranking**) the pixels , then *replacing the value of a pixel by the* median of the gray levels in the neighborhood of that pixel.

Median filters are quite **popular** because, for certain types of random noise, they provide **excellent noise reduction + less blurring** than linear smoothing filters of similar size.

Median filters are particularly effective in the presence of *impulse noise, also called* ***salt-and-pepper noise*** *because of its* appearance as white and black dots superimposed on an image.

Spatial filters : Smoothing
order statistics: Median filter
use : blurring + **reduce salt and pepper noise**

Which one has removed the
salt-and-pepper noise??



The original image with salt and pepper noise

The smoothed image using averaging

The smoothed image using median

Spatial filters : Sharpening (high pass)

3.6 Sharpening Spatial Filters

- The principal **objective** of sharpening is to highlight **transition** in intensity.
- Sharpening is accomplished by spatial **differentiation**.
- Image differentiation **enhances edges** and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values

3.6.1 Foundation

- The **derivatives** of a digital function are defined in terms of **differences**
- Definitions of the **first and 2nd-order** derivatives of a 1-D function f(x) are the differences:

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

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

3.6.2 Using the Second Derivative for Image Sharpening–The Laplacian

- The Laplacian, for a function (image) f(x, y) of two variables, is defined and given below:

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

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

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

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

Sharpening Spatial filters

- Used for highlighting fine detail or enhancing detail that has been blurred.
- Averaging is analogous to integration.
- Sharpening is analogous to differentiation.
- A Basic defintion of the first-order derivative of 1-D function *f(x)* is the difference:
  - $f'(x) = f(x + 1) - f(x)$
  - Must be zero in flat areas
  - Must be non-zero at the onset of a gray-level step or ramp
  - Must be nonzero along ramps
- Second derivatives
  - $f''(x) = f(x + 1) + f(x - 1) - 2f(x)$
  - Must be zero in flat areas
  - Must be non-zero at the onset and end of a gray-level step or ramp
  - Must be zero along ramps of constant slope

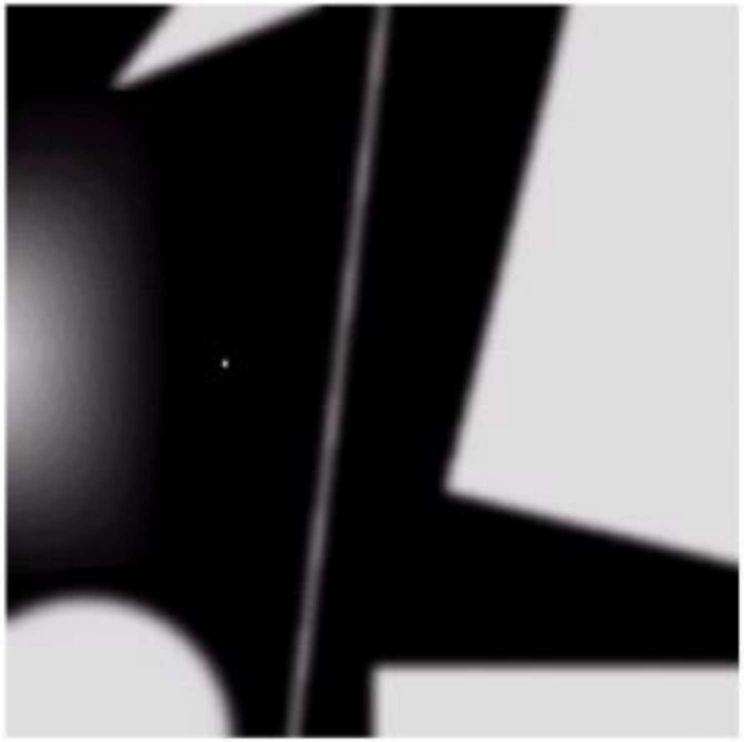
Spatial filters : Sharpening
1^{st} VS 2^{nd} derivative sharpening

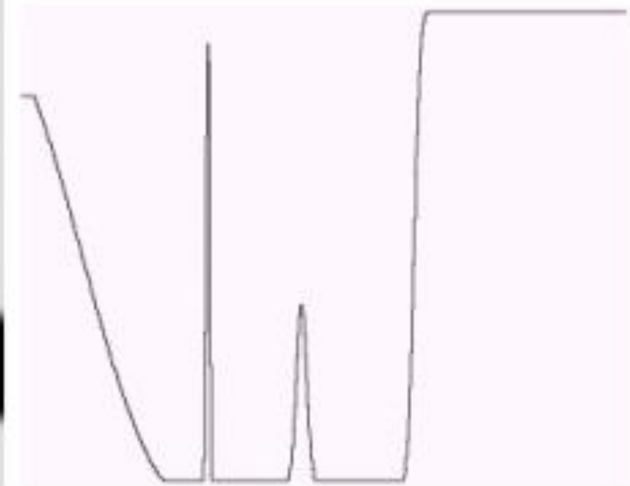
1^{st} derivative sharpening produces thicker edges in an image
1st derivative sharpening has stronger response to gray level change

2^{nd} derivative sharpening has stronger response to fine details, such as thin lines and isolated points.

2^{nd} derivative sharpening has double response to gray level change

1st vs 2nd derivative





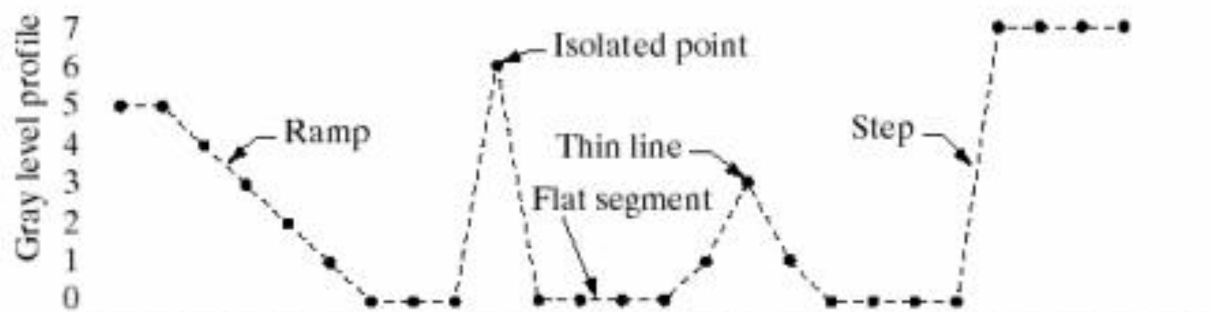


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		–1	–1	–1	–1	–1	0	0	6	–6	0	0	0	1	2	–2	–1	0	0	0	7	0	0	0			
Second Derivative		–1	0	0	0	0	1	0	6	–12	6	0	0	1	1	–4	1	1	0	0	7	–7	0	0			

Spatial filters : Sharpening (high pass)

1. LAPLACE
2. Unsharp
3. High boost
4. sobel

2nd Derivative for Image Sharpening–The Laplacian

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

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

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

$$h_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

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

2nd Derivative for Image Sharpening–The Laplacian

$$h_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

or

$$h_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

or

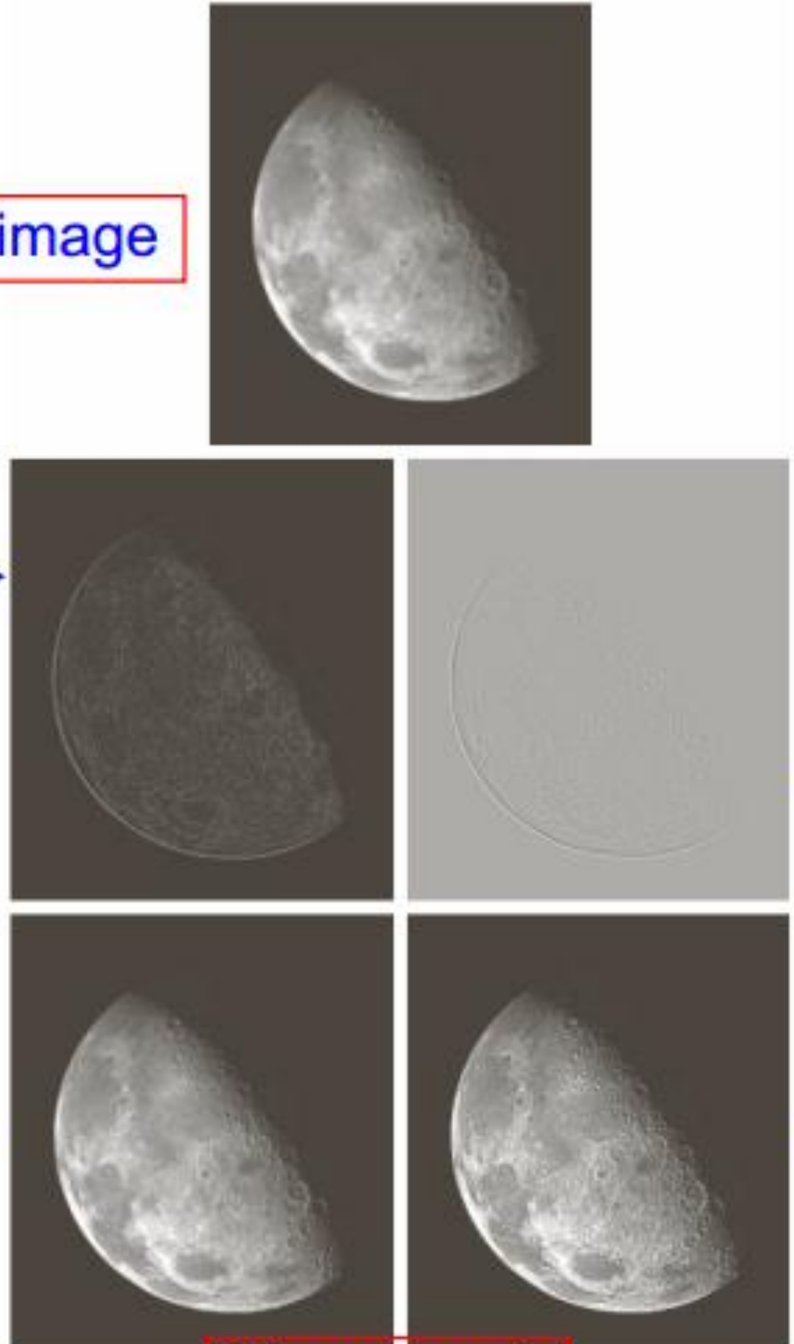
$$h_4 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Spatial filters : Sharpening
1) LAPLACE (2nd derivative)

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Original image

Laplacian



Scaled image

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpened

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).

The Laplacian (Better Implementation)

In practice we use the Laplacian for Image enhancement

In a better way by increasing (decreasing)

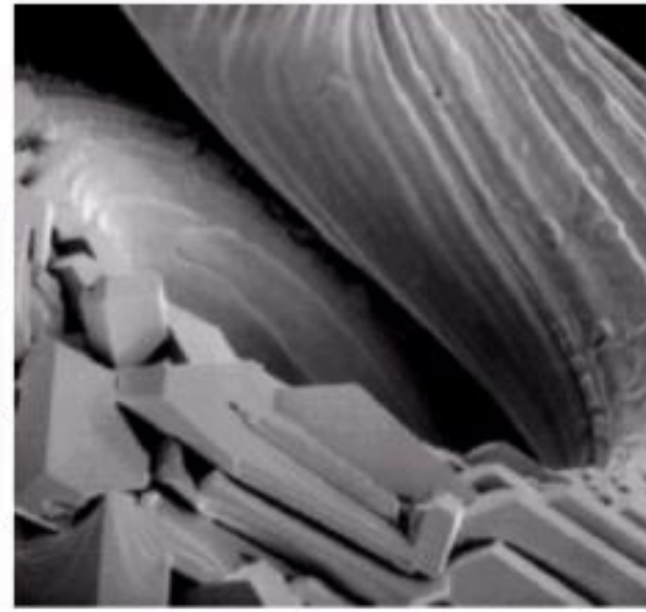
the center coefficient by 1; e.g.

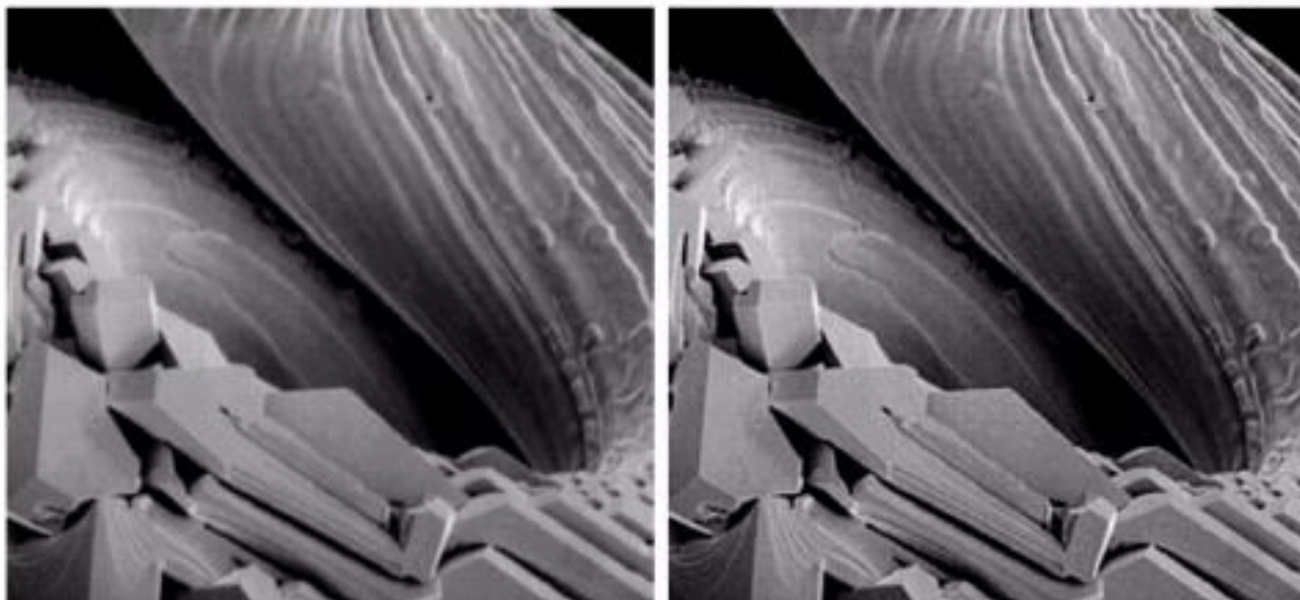
0	-1	0
-1	5	-1
0	-1	0

The Laplacian (more practical masks)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Spatial filters : Sharpening LAPLACE – 2^{st} derivative

Use: for highlighting fine detail or enhancing detail that has been blurred.

Example: apply the following laplace on the highlighted pixel

0	−1	0
−1	4	−1
0	−1	0

١٥٣	١٥٧	١٥٦	١٥٣	١٥٥
١٥٩	١٥٦	١٥٨	١٥٦	١٥٩
١٥٥	١٥٨	**١٥٤**	١٥٦	١٦٠
١٥٤	١٥٧	١٥٨	١٦٠	١٦٠
١٥٧	١٥٧	١٥٧	١٥٦	١٥٥

154*4 – 158- 156-158-158 = -14

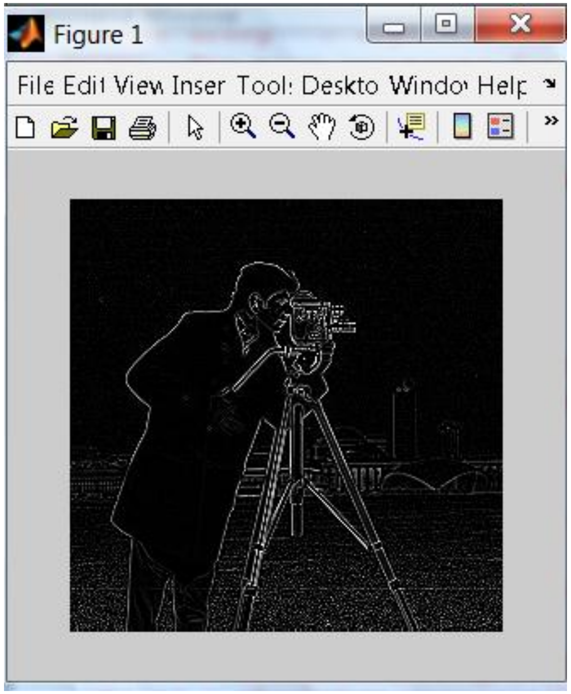
So the value after filter = -14

We call the resultant image: **sharpened image.**

Filtered image=original +sharpened image

The value in the filter image=154-14 =130

La place Sharpened image



La place filtered image

Unsharp Masking

- This operation is referred to as edge enhancement, edge crispening or *unsharp masking*.
- To make edges slightly sharper and crisper.
- A very popular practice in industry (where the name came).
- Subtracting a blurred version of an image from the image itself.

$$f_s(x,y) = f(x,y) - \overline{f}(x,y)$$

Sharpened image

Original image

Blurred image

Unsharp masking and high-boost filtering

- To sharpen images by **subtracting** a blurred or **unsharp (smoothed)** version of an image from the original image itself.
- This process, called unsharp masking, **consists of:**

1. Blur the original image.
2. Subtract the blurred image from the original image (the resulting difference is called the mask).
3. Add the mask to the original.

Unsharp masking and Highboost filtering expressions:

- Let $\bar{f}(x, y)$ denote the **blurred** image.
- The **mask** is obtained as:

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

- A weighted portion of mask is added to original:

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

- **Unsharp Making**: when k=1
- **Highboost filtering**: when $k > 1$

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.

1st Derivatives of Enhancement – The Gradient

First derivatives in image processing are implemented using the magnitude of the gradient.

For a function *f(x, y)*, the gradient of *f* at coordinates *(x, y)* is defined as the 2-dimensional column vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}.$$

1^{st} Derivatives of Enhancement – The Gradient

The magnitude of the vector is given by:

$$\begin{aligned}\nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}.\end{aligned}$$

The magnitude of the gradient vector is often referred to
As the *gradient*.

Because of the computational burden of the above Eq.,
it is common Practice to approximate the magnitude of
the gradient using absolute values as :

$$\nabla f \approx |G_x| + |G_y|$$

This Eq. is simpler to compute.

1st Derivatives of Enhancement – The Gradient

We can now define digital approximations to the preceding equations, and from there formulate the appropriate filter masks.

We will use the following notation to denote image points in a 3x3 region:

...

1st Derivatives of Enhancement – The Gradient

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

The center point, z5 , denotes f(x, y), z1 denotes f(x-1, y-1), and so on.

1st Derivatives of Enhancement – The Gradient

$$\nabla f \approx |G_x| + |G_y|$$

a
b c
d e

FIGURE 3.44 A 3×3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

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

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Roberts operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operators

$$\nabla f \approx |(z_7+2z_8+z_9)-(z_1+2z_2+z_3)| + |(z_3+2z_6+z_9)-(z_1+2z_4+z_7)|$$

The weight value of 2 is used to achieve some smoothing by giving more <u>importance</u> to the center point.

Sobel Operators

−1	−2	−1
0	0	0
1	2	1

Detects horizontal edges

−1	0	1
−2	0	2
−1	0	1

Detects vertical edges

Sobel operators

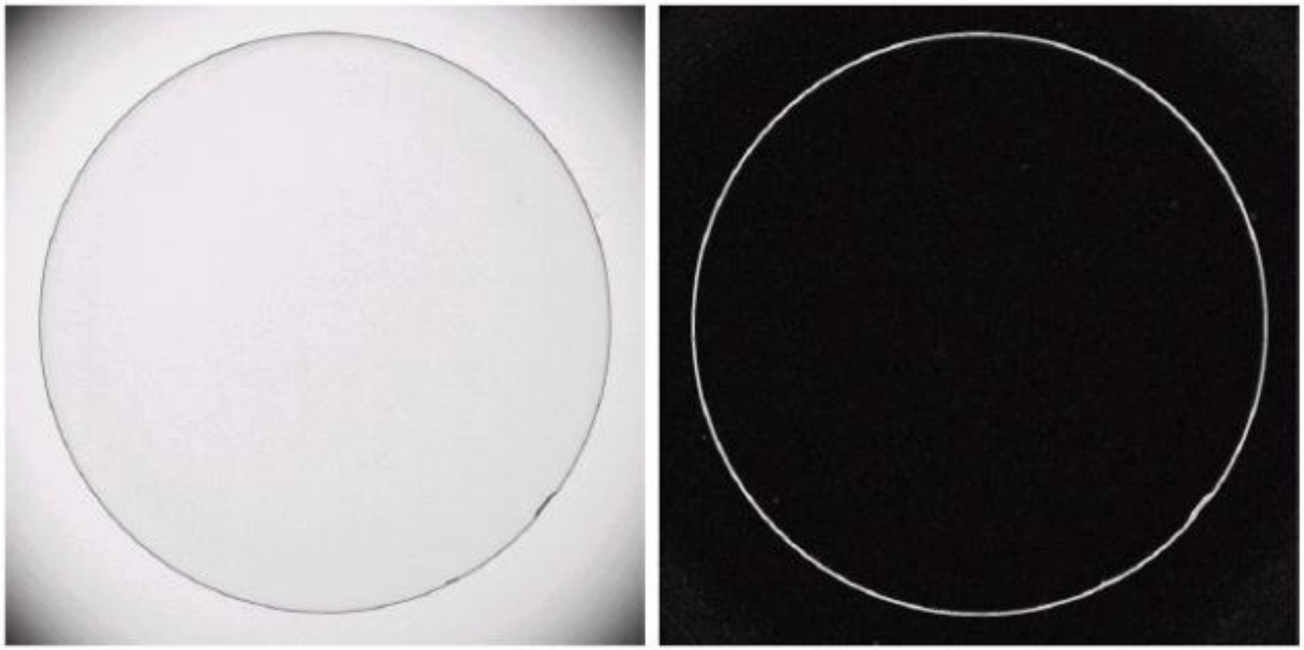
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

we can apply the sobel horizontal kernel or the sobel vertical kernel or both and adding them together.

1st Derivatives of Enhancement – The Gradient

Gradient is frequently used in industrial inspection.



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Constant or slowly varying shades of gray have been eliminated.

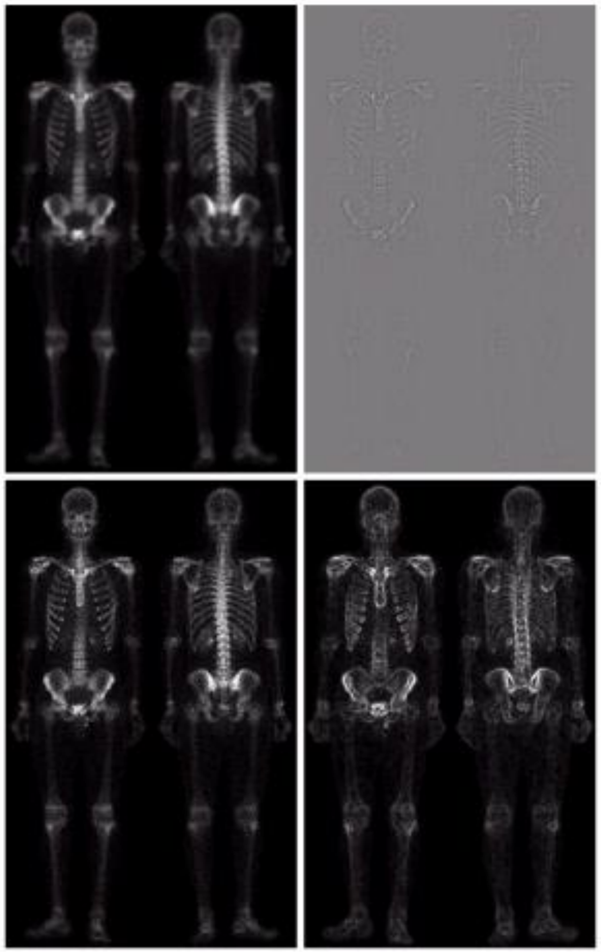
Combining spatial Enhancement Methods

In this chapter we have focused attention thus far on individual enhancement approaches.

Frequently, a given enhancement task will require application of *several complementary enhancement techniques* in order to achieve an acceptable result.

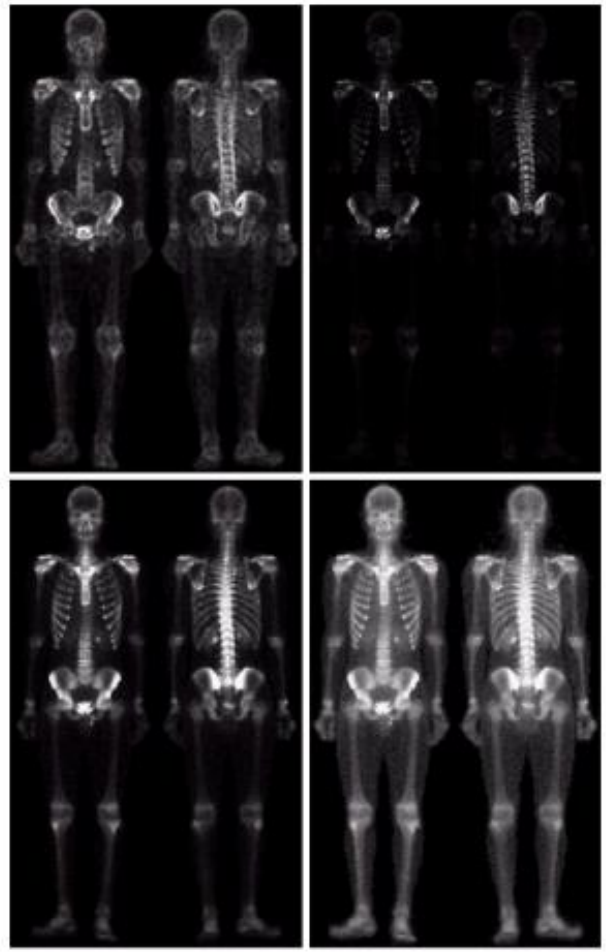
In this section we illustrate by means of an example how to combine several of the approaches developed in this chapter to address a difficult enhancement task.

Combining spatial Enhancement Methods



a b
c d

FIGURE 3.46
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).



e f
g h

FIGURE 3.46 *(Continued)*
(e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Combining spatial Enhancement Methods

- Our **objective** is to enhance the image by sharpening it and by bringing out more of the skeletal detail.
- The narrow dynamic range of the gray levels and high noise content make this image **difficult** to enhance.
- The **strategy** we will follow is:
  - to utilize the **Laplacian** to highlight fine detail, and
  - the **gradient** to enhance prominent edges.
  - a smoothed version of the gradient image will be used to mask the Laplacian image
  - Finally, we will attempt to increase the dynamic range of the gray levels by using a **gray-level transformation**.

Conclusions

With image enhancement we try to make images look better according to subjective criteria.

Contrast enhancement of a gray image can be achieved by manipulating the gray values of the pixels so that they become more diverse.
This can be done by defining a transformation that converts the distribution of the gray values to a pre-specified shape. The choice of this shape may be totally arbitrary.

Smoothing spatial filters have been used for blurring and for noise reduction, while sharpening spatial filters have been used to highlight fine details in the image or to enhance detail that has been blurred ...

MATLAB (average +sobel+laplace)

Fspecial : for choosing the filter:

X=fspecial('average',[3 3])

p=fspecial('laplacian', 0)

v=fspecial('sobel') → horizontal sobel

Y=v' → vertical sobel

Imfilter : for applying filter.

m= imread('cameraman.tif');

f= imfilter(m,x) → this command will apply average filter on image

Fp=imfilter(m,p) → this command will apply laplace filter on image

Imshow(fp) → this command will show the laplace sharpened image

imshow(m+fp) → this command will show the filtered image after applying la place

MATLAB – cont. (unsharp + highboost)

Applying unsharp:.

X=imread(‛cameraman.tif‛)

p=fspecial(’average’,[3 3])

xblur=imfilter(x,p) → *for bluring*

Mask= xblur-x → *for subtracting the mask*

Y=x+mask → *for unsharp the image*

Revise slide 57 + 58

For highboost , only multiply the mask any any constant in the last step: y=x+3*mask for example

MATLAB – cont. (border padding)

Ex.

```
X=imread(‘cameraman.tif‘)
p=fspecial(‘laplacian’, 0)
Xp=imfilter(x,p, ‘replicate‘)
```

This command will apply border padding instead of zero padding