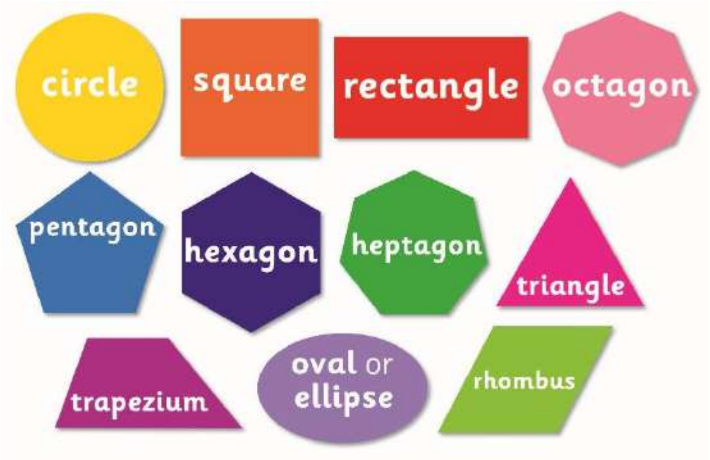
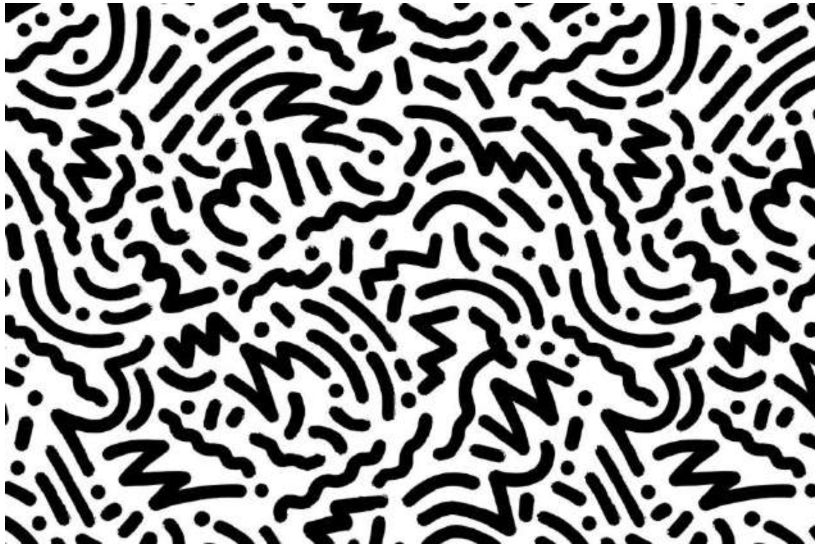
CSE395

Image processing

Module-1: Introduction: Fundamentals





CONTENTS

IMAGES

ANALOG

Continuous

Each Image Point

Brightness

Film Density

Color

For Human Viewing

DIGITAL

Matrix of Pixels

Number

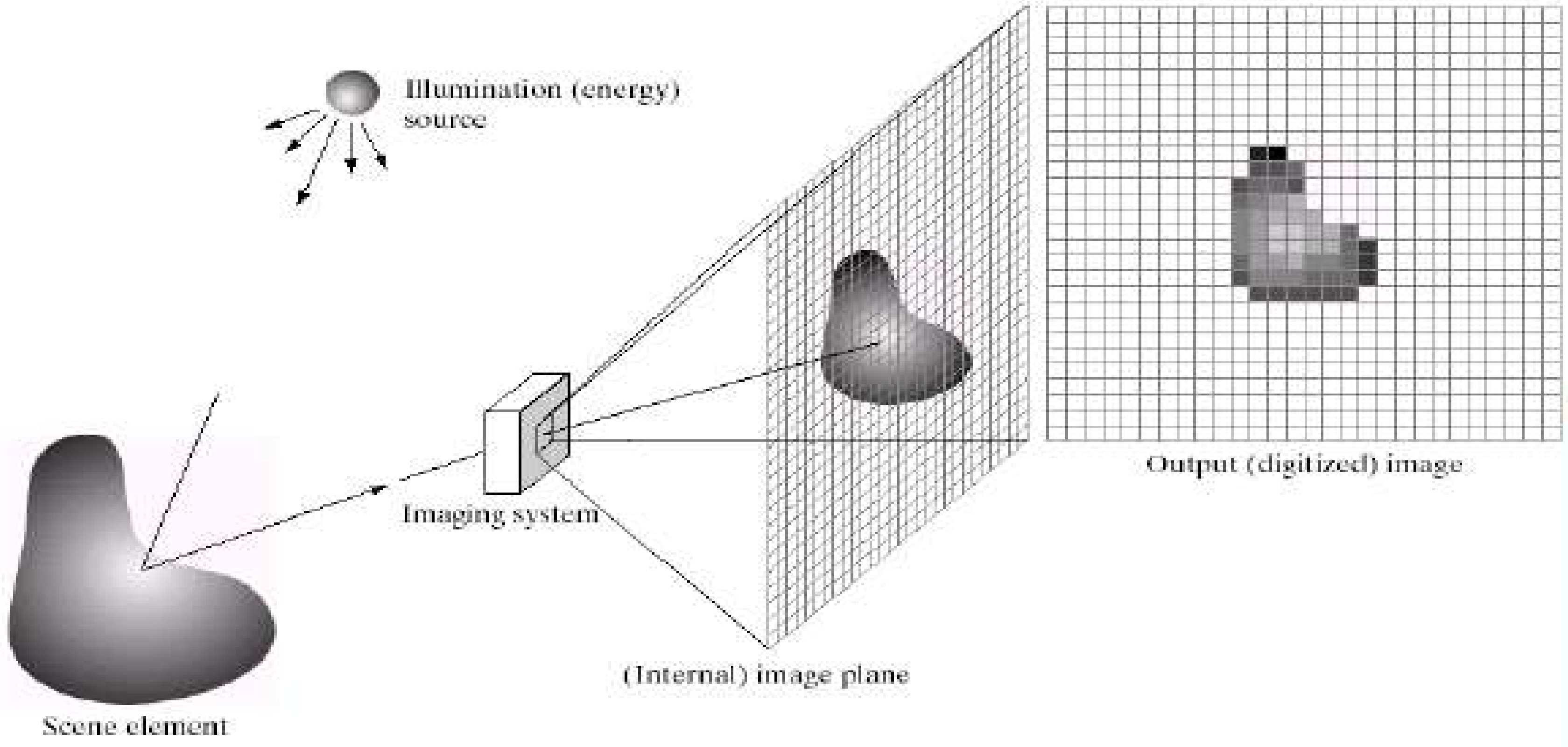
56	56	57	56
56	56	57	56
57	57	57	59
58	58	58	60

For Computer Systems

Sprawls

WHAT IS A DIGITAL IMAGE?

A digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels



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

- What is Digital Image Processing?

Digital Image

— a two-dimensional function $f(x, y)$

x **and** ***y*** **are spatial coordinates**

The amplitude of ***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: an ensemble of recognition of individual objects

Pixel

— the elements of a digital image

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and "Alpha", a.k.a. Opacity)

For most of this presentation we will focus on greyscale images.

The continuum from image processing to computer vision can be broken up into low-, mid- and high-level processes

Low Level Process

Input: Image
Output: Image

Examples: Noise removal, image sharpening

Mid Level Process

Input: Image
Output: Attributes

Examples: Object recognition, segmentation

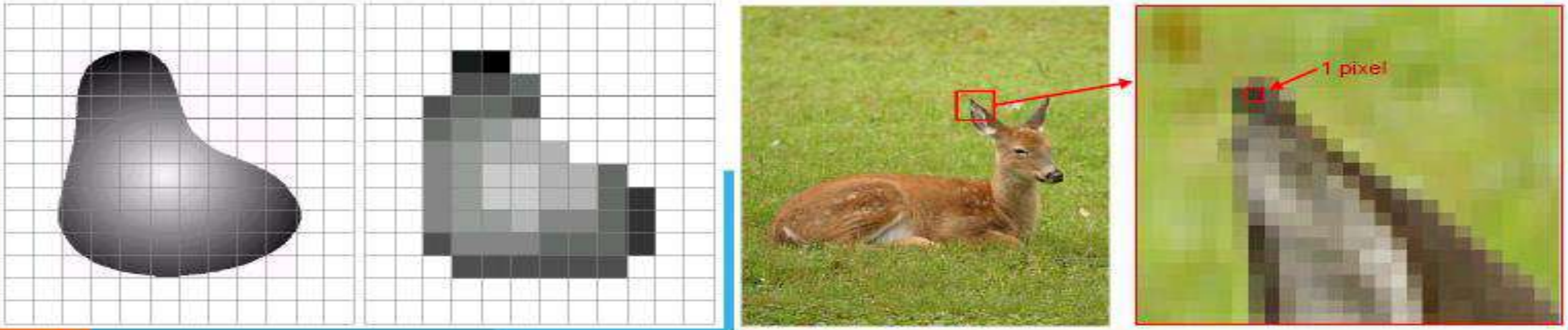
High Level Process

Input: Attributes
Output: Understanding

Examples: Scene understanding, autonomous navigation

Pixel values typically represent gray levels, colours, heights, opacities etc

Remember *digitization* implies that a digital image is an *approximation* of a real scene



HISTORY OF DIGITAL IMAGE PROCESSING

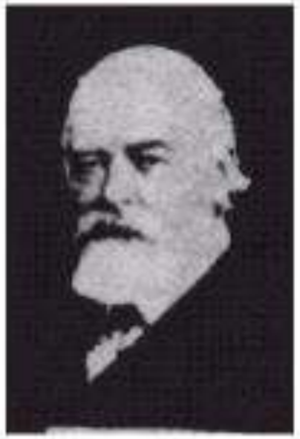
Early 1920s: One of the first applications of digital imaging was in the news-paper industry

- The Bartlane cable picture transmission service
- Images were transferred by submarine cable between London and New York
- Pictures were coded for cable transfer and reconstructed at the receiving end on a telegraph printer

Early digital image

Mid to late 1920s: Improvements to the Bartlane system resulted in higher quality images

- New reproduction processes based on photographic techniques
- Increased number of tones in reproduced images

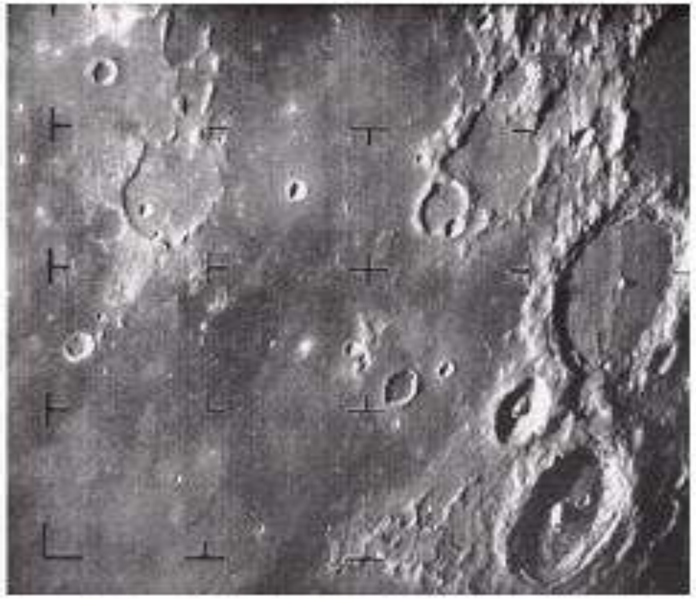


Improved digital image

Early 15 tone digital image

1960s: Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing

- 1964: Computers used to improve the quality of images of the moon taken by the *Ranger 7* probe
- Such techniques were used in other space missions including the Apollo landings



A picture of the moon taken by the Ranger 7 probe minutes before landing

1970s: Digital image processing begins to be used in medical applications

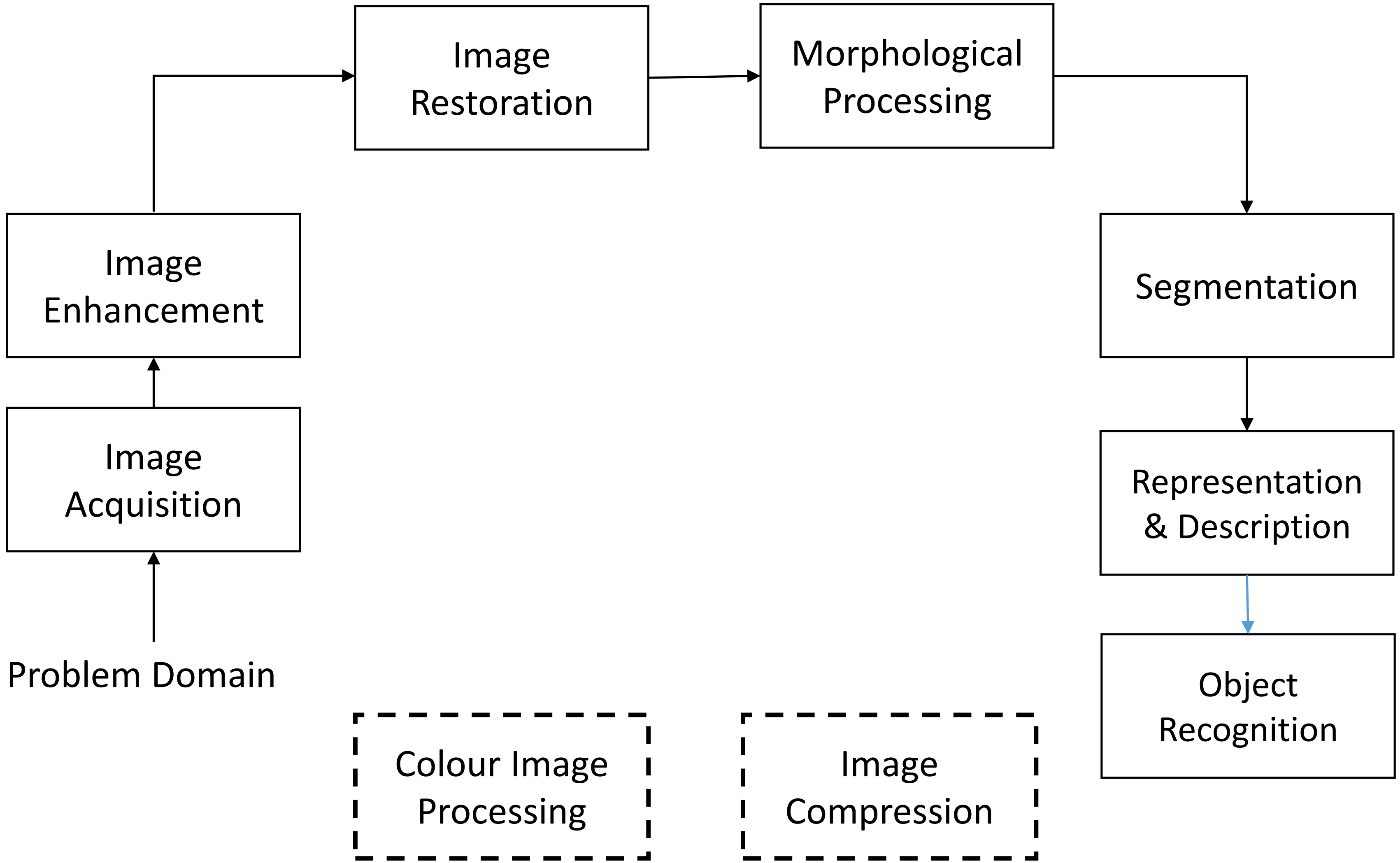
- **1979**: Sir Godfrey N. Hounsfield & Prof. Allan M. Cormack share the Nobel Prize in medicine for the invention of tomography, the technology behind Computerised Axial Tomography (CAT) scans

Typical head slice CAT image

1980s - Today: The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas

- Image enhancement/restoration
- Artistic effects
- Medical visualisation
- Industrial inspection
- Law enforcement
- Human computer interfaces

Key Stages in Digital Image Processing



Key Stages in Digital Image Processing: Image Acquisition

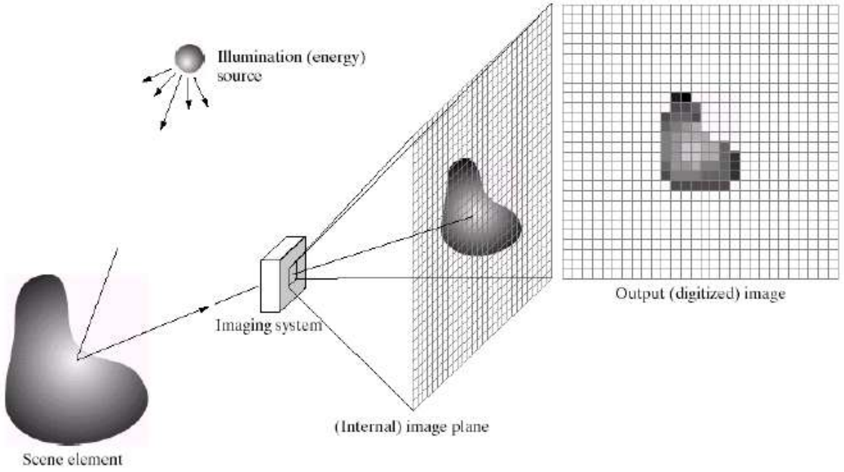
Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition



Representation & Description

Problem Domain

Object Recognition

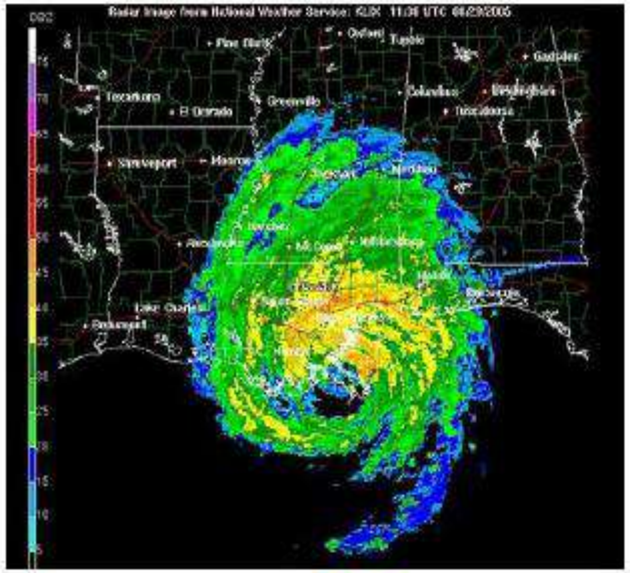
Colour Image Processing

Image Compression

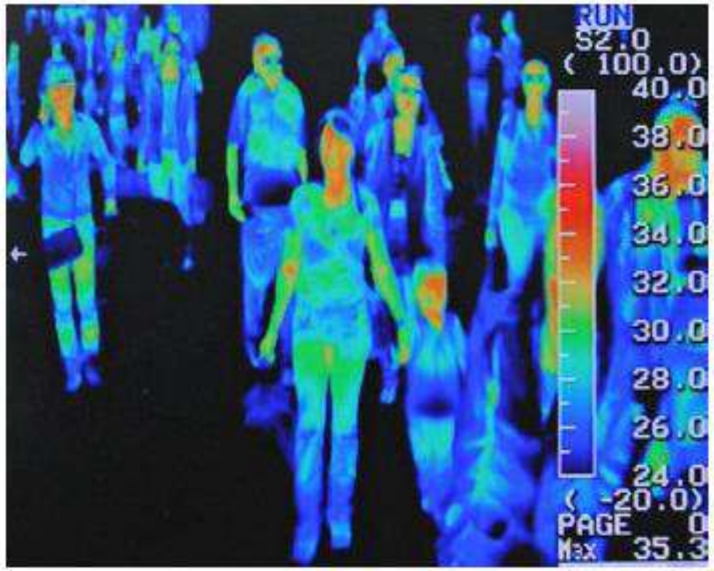
Image Acquisition

- A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc.
- Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence.
- The pixel values typically correspond to light intensity in one or several spectral bands (gray images or colour images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

Image Acquisition – Other Sensors



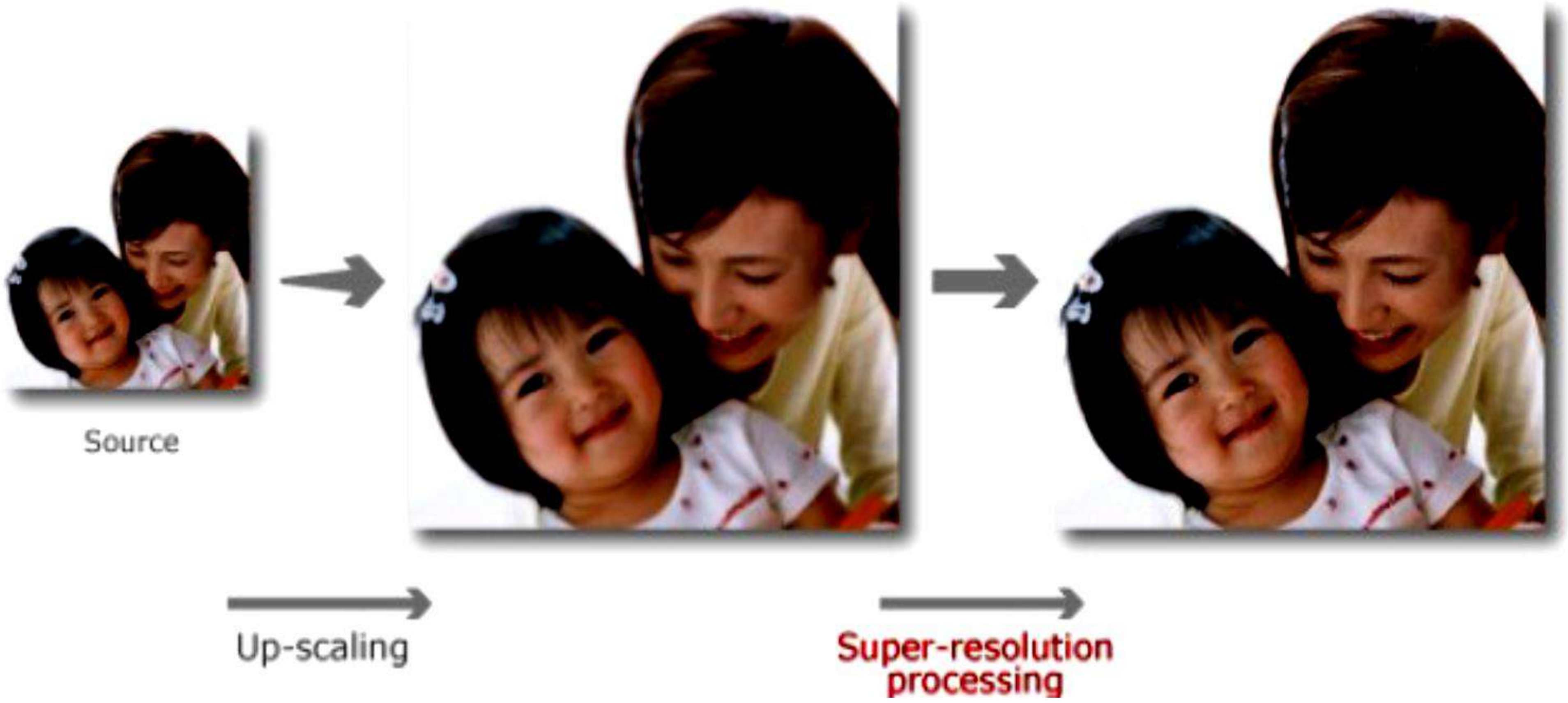
Weather radar image



Thermal image

Super-Resolution Technology

"Super resolution" is a technology that is used to sharpen out-of-focus images or smooth rough edges in images that have been enlarged using a general up-scaling process (such as a bilinear or bicubic process), thereby delivering an image with high-quality resolution.



Key Stages in Digital Image Processing: Image Enhancement

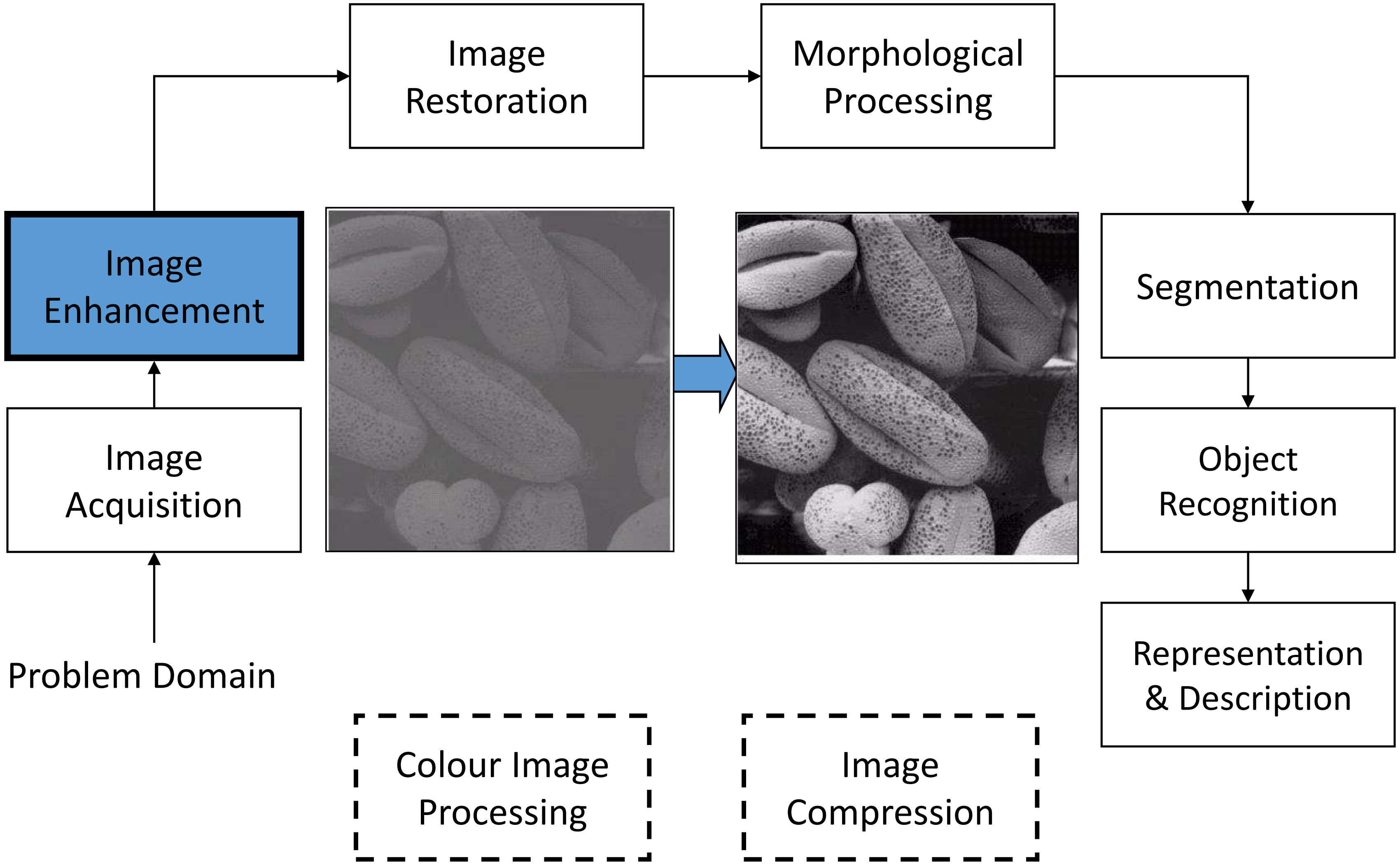
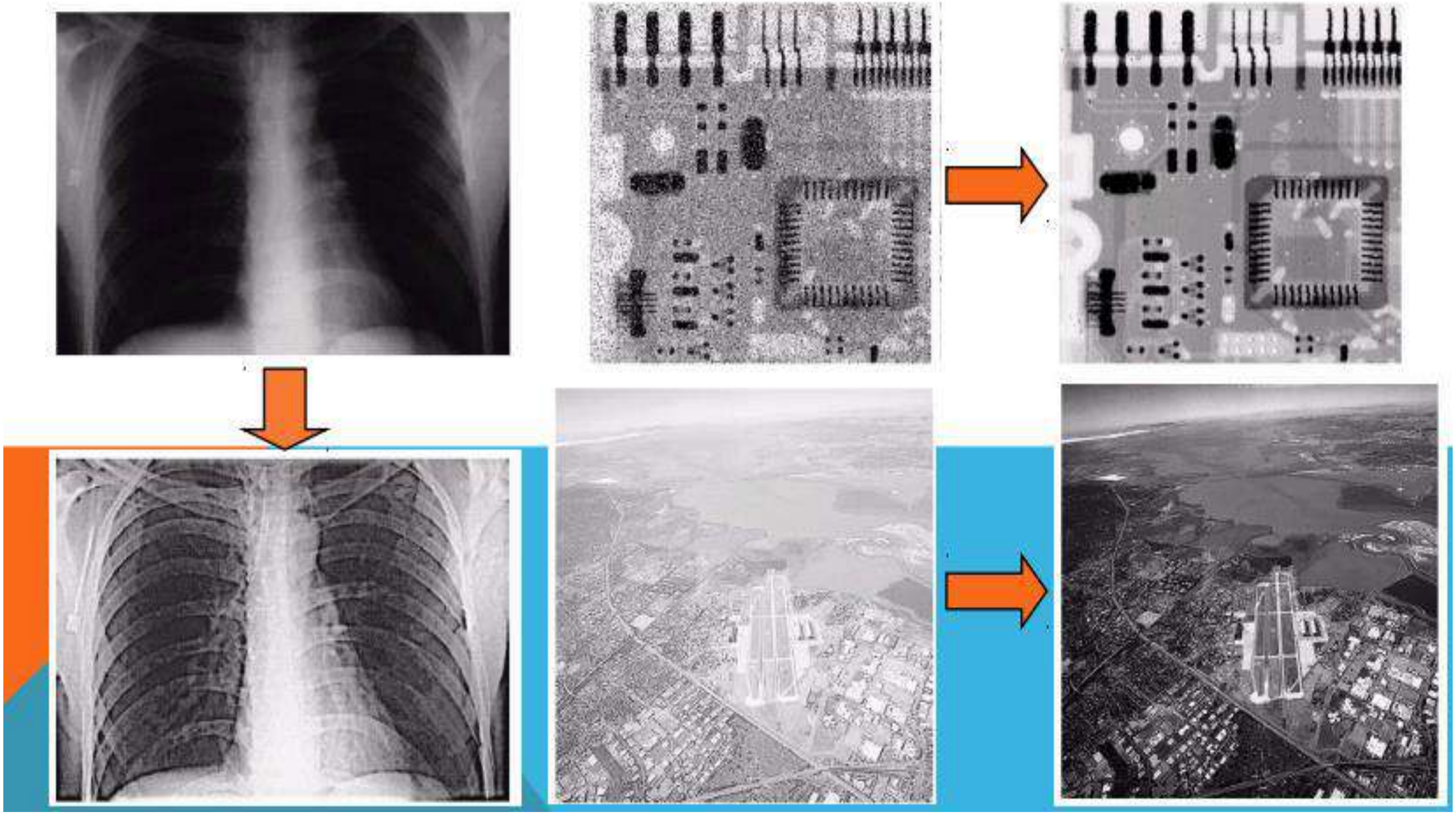


Image Enhancement

- Process of manipulating an images
- Suitable for specific applications eg medical Imaging such as X-ray
- Satellite Images



Key Stages in Digital Image Processing: Image Restoration

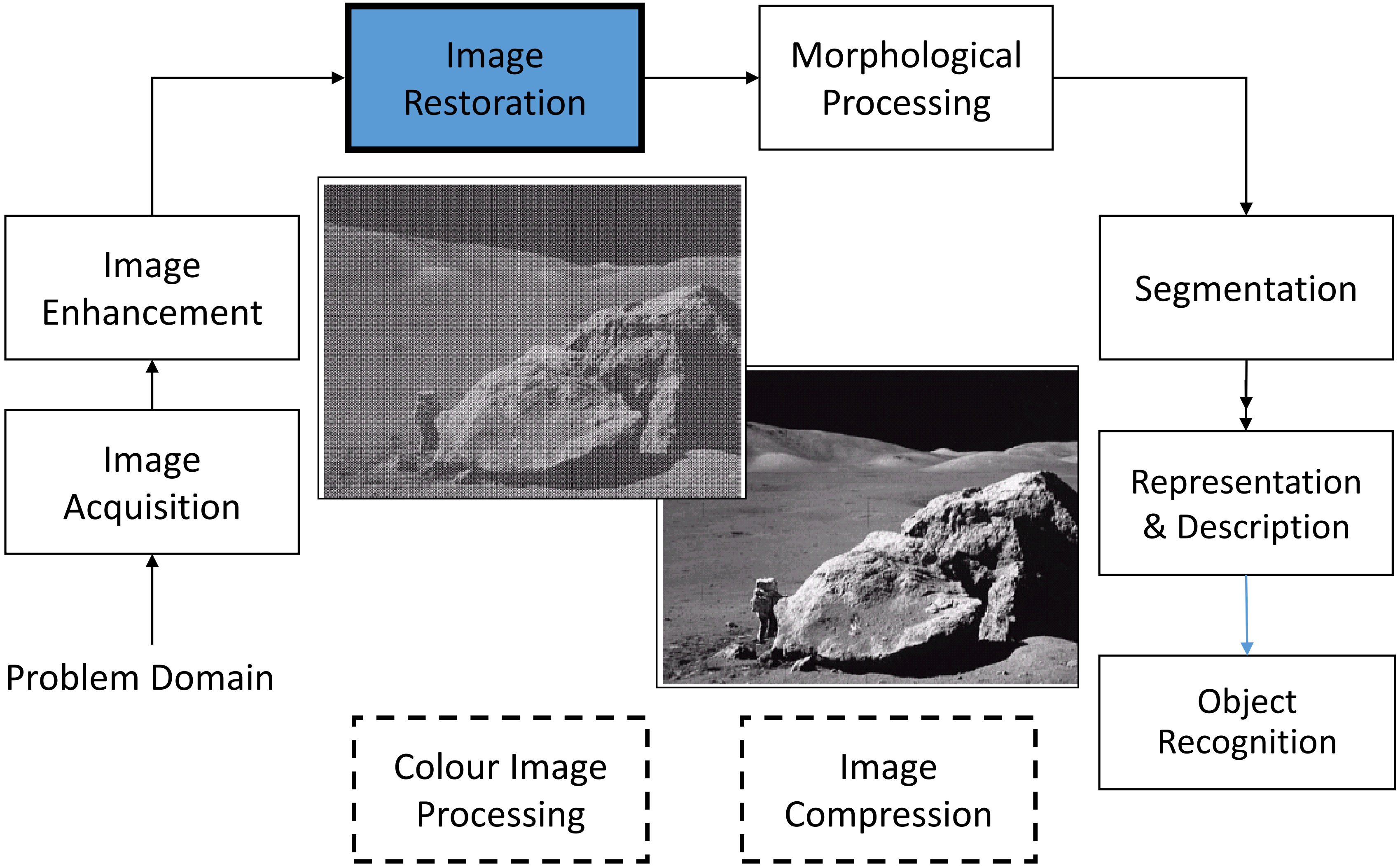
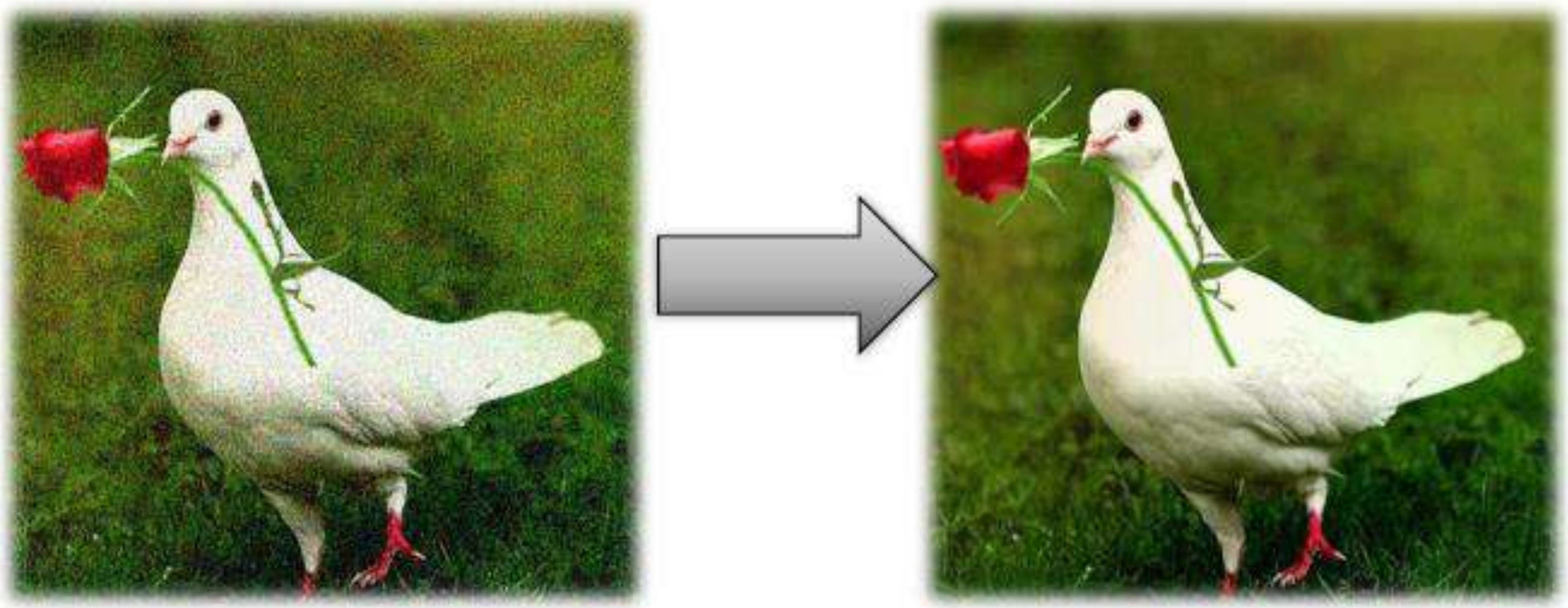


Image Restoration

- Improving the appearance of an image
- Based on mathematical or probabilistic based model



- Image restoration attempts to restore images that have been degraded
- Identify the degradation process and attempt to reverse it.
- Almost Similar to image enhancement, but more objective.

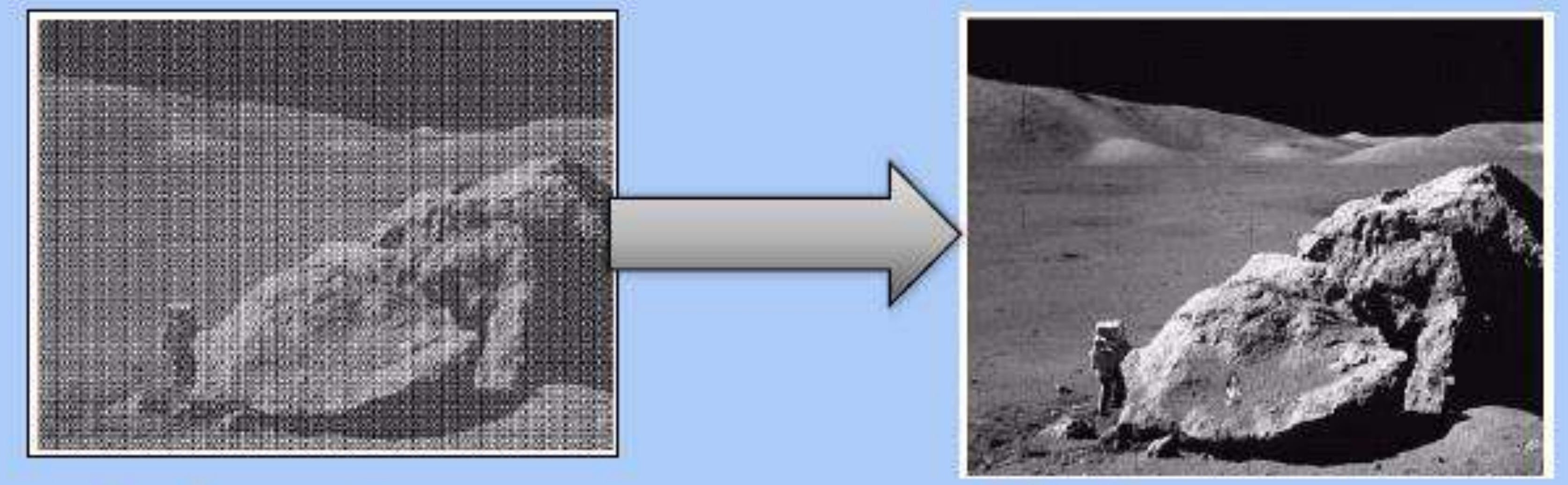
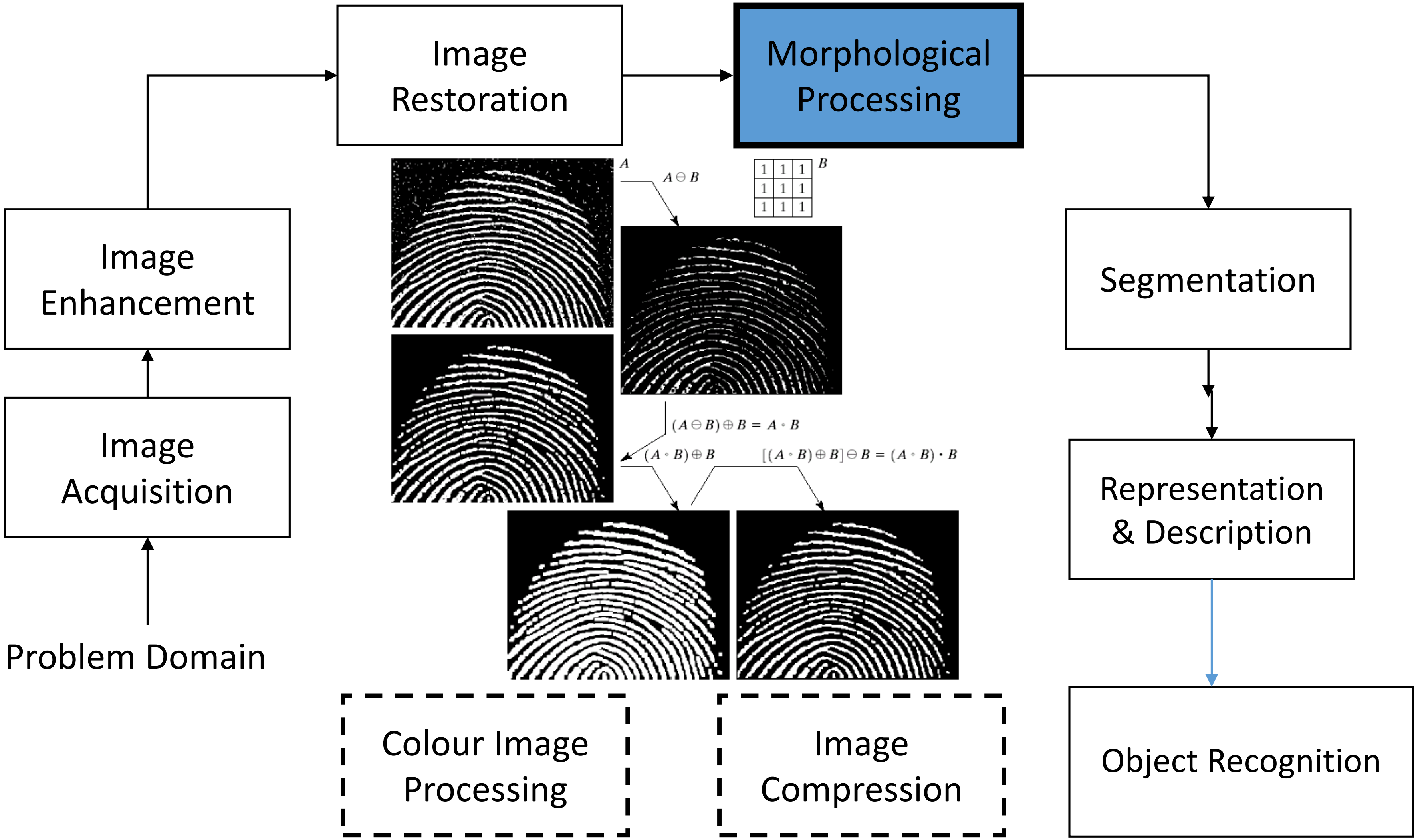


Fig: Degraded image

Fig: Restored image

S.no	Image Enhancement	Image Restoration
1	Image Enhancement is subjective	Image restoration is objective process.
2	Better visual representation	Its removes effects of sensing environment
3	No Quantitative measure is required	Mathematical model is required
4	It concerns about the extraction of features	It concerns about the restoration of degradation

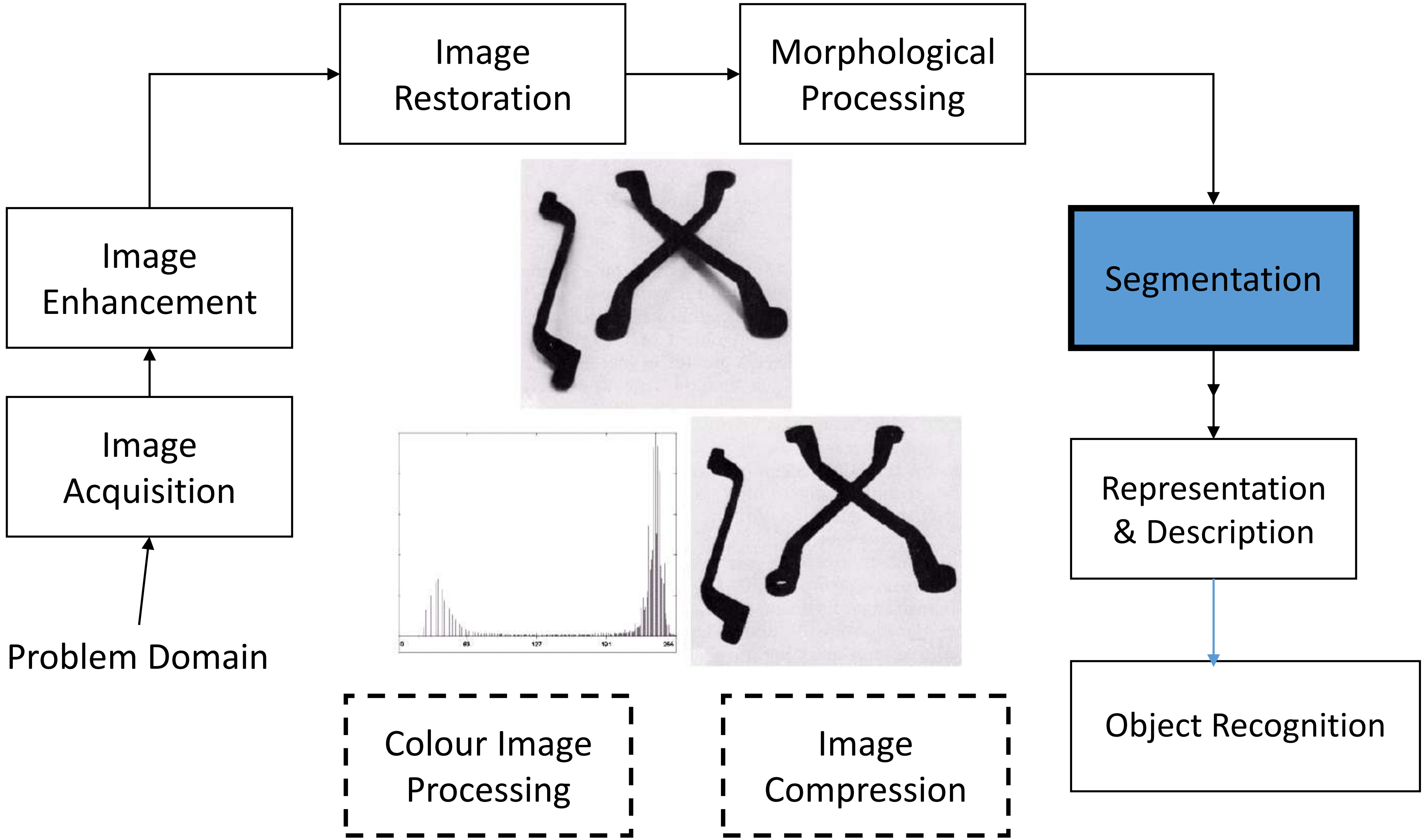
Key Stages in Digital Image Processing: Morphological Processing



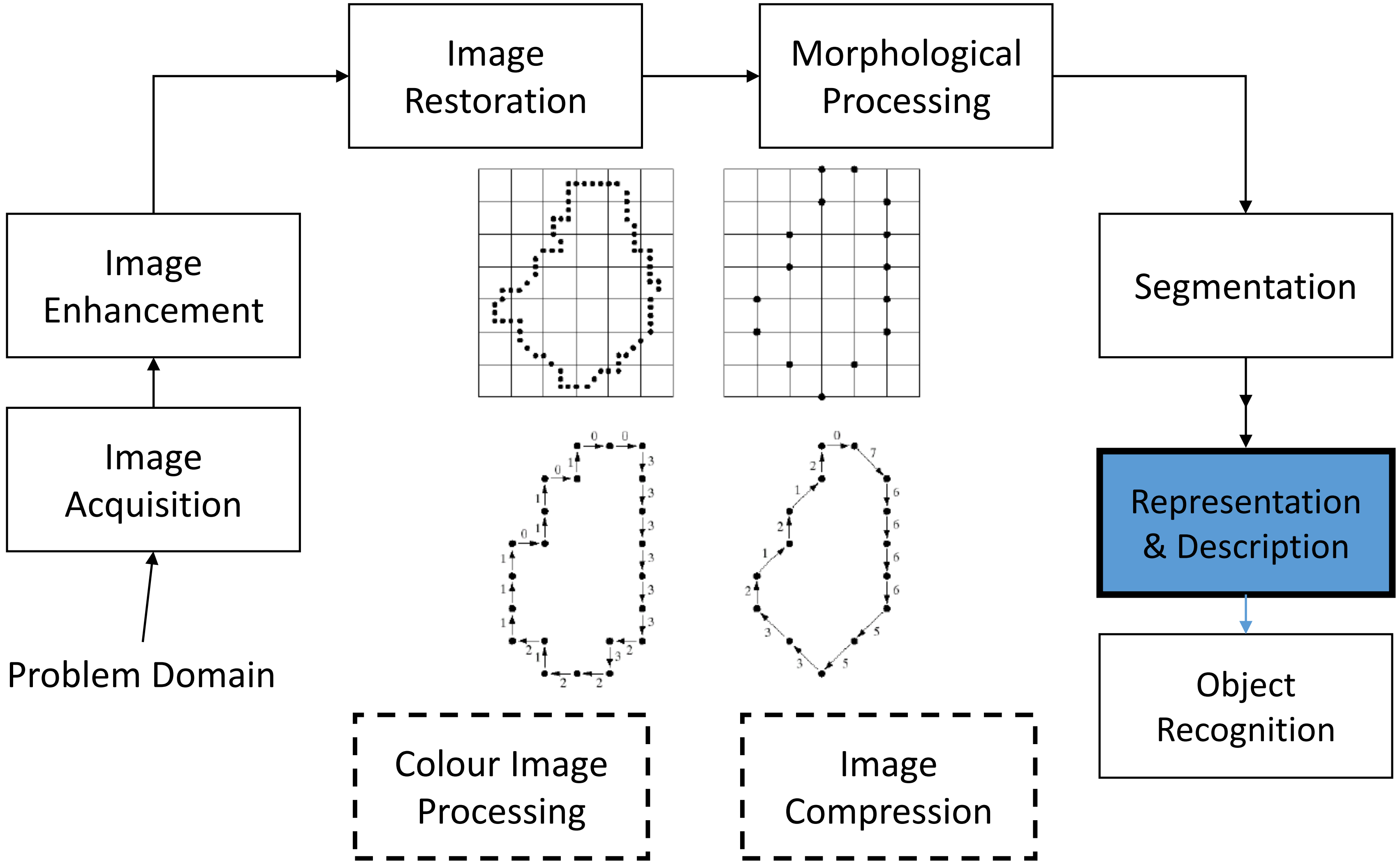
Morphological Processing

- *Morphology* is a broad set of image processing operations that process images based on shapes.
- Morphological operations apply a structuring element to an input image, creating an output image of the same size.
- In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.
- Step deals with tools for extracting image components those are useful in the representation and description of shape

Key Stages in Digital Image Processing: Segmentation



Key Stages in Digital Image Processing: Representation & Description



Representation & Description

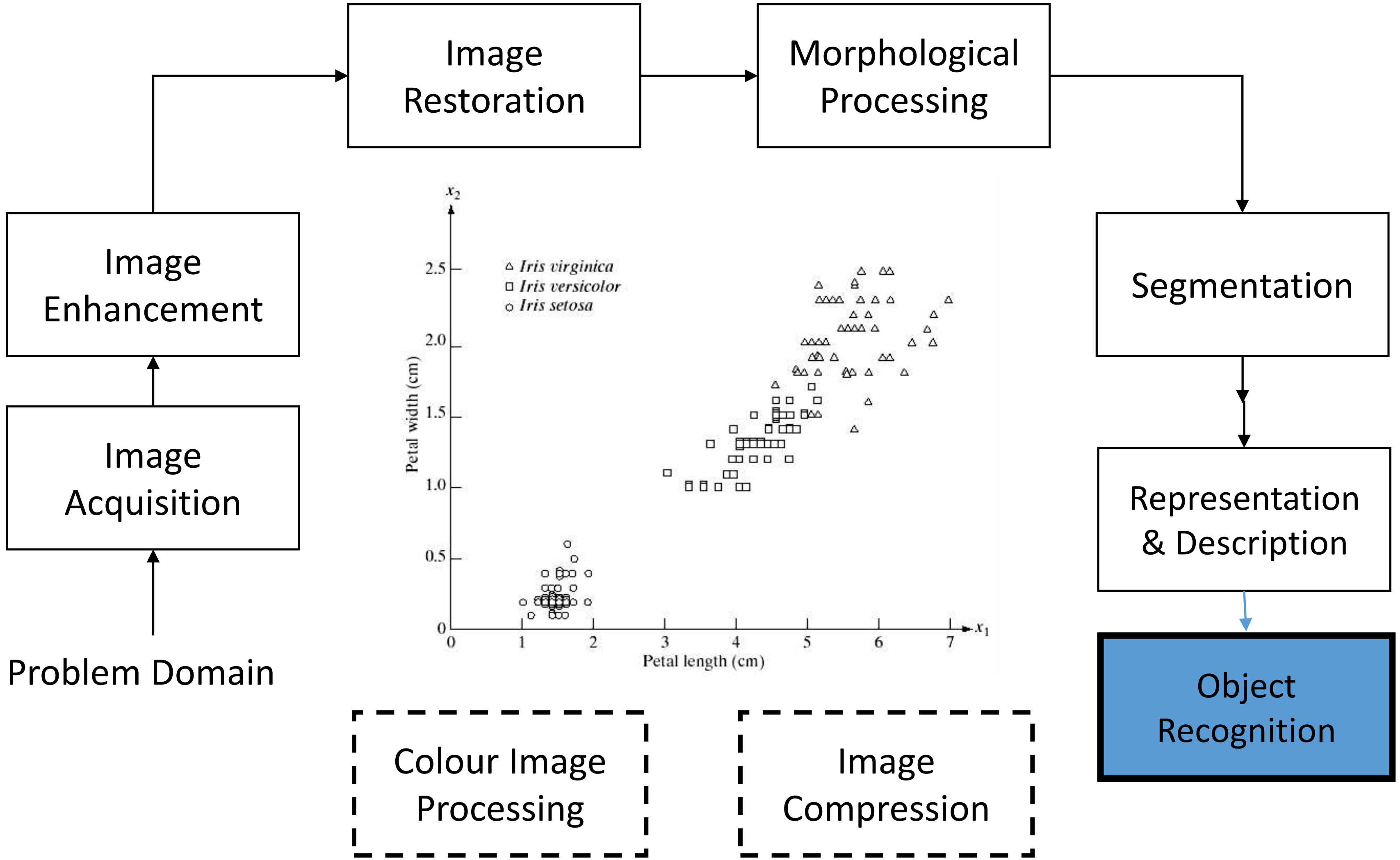
Representation

- Follow the output of Segmentation usually either the boundary of regions or all the point of region
- Converting to the suitable form for computer processing
- Decision should be made either a boundary or complete region
- Boundary representation -for external shape characteristics such corner and inflections
- Regional representation –for internal part such as texture or Skeletal shape

Description

- Features selection
- Some quantitative information of Interest
- Differentiate from class to another class

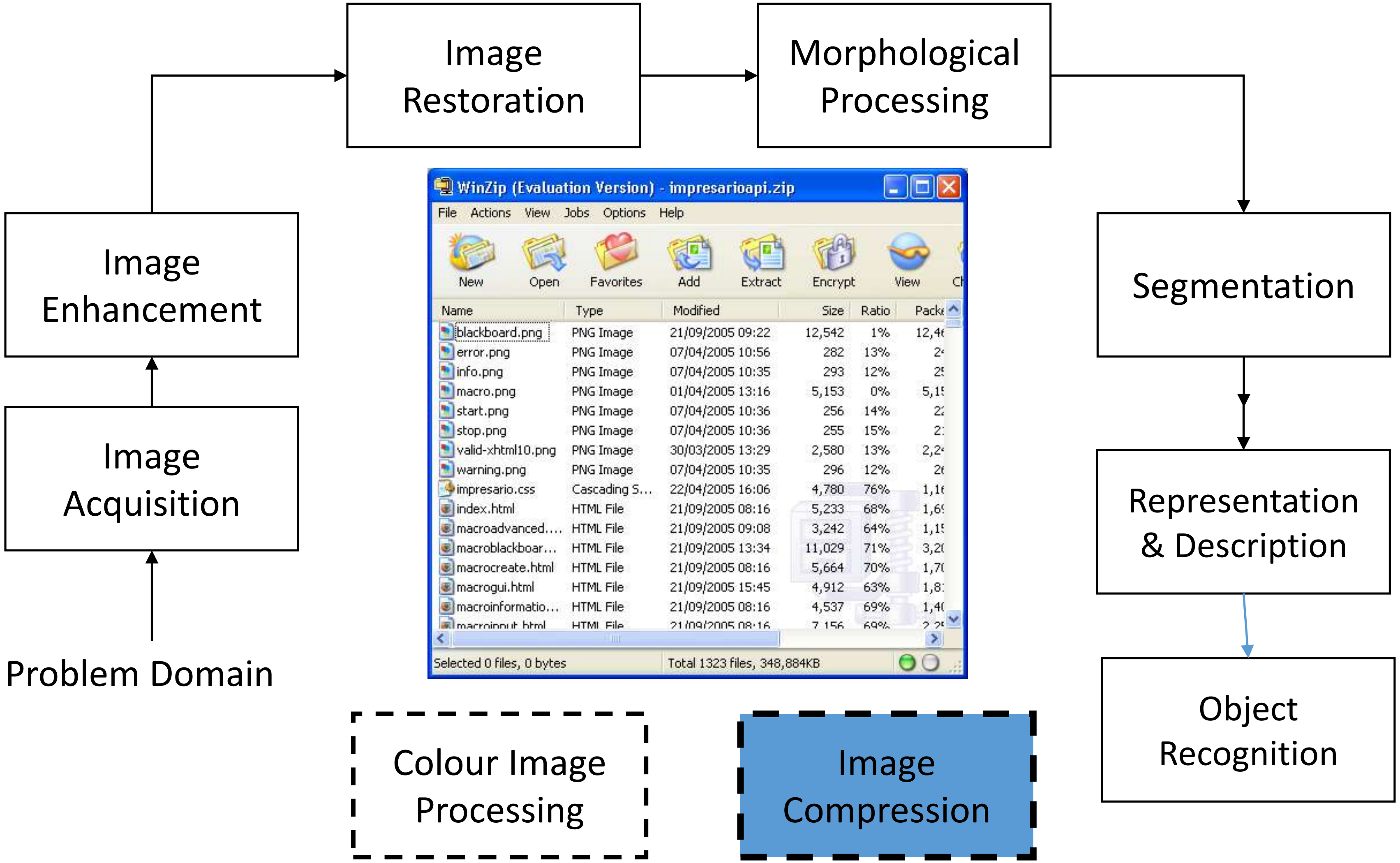
Key Stages in Digital Image Processing: Object Recognition



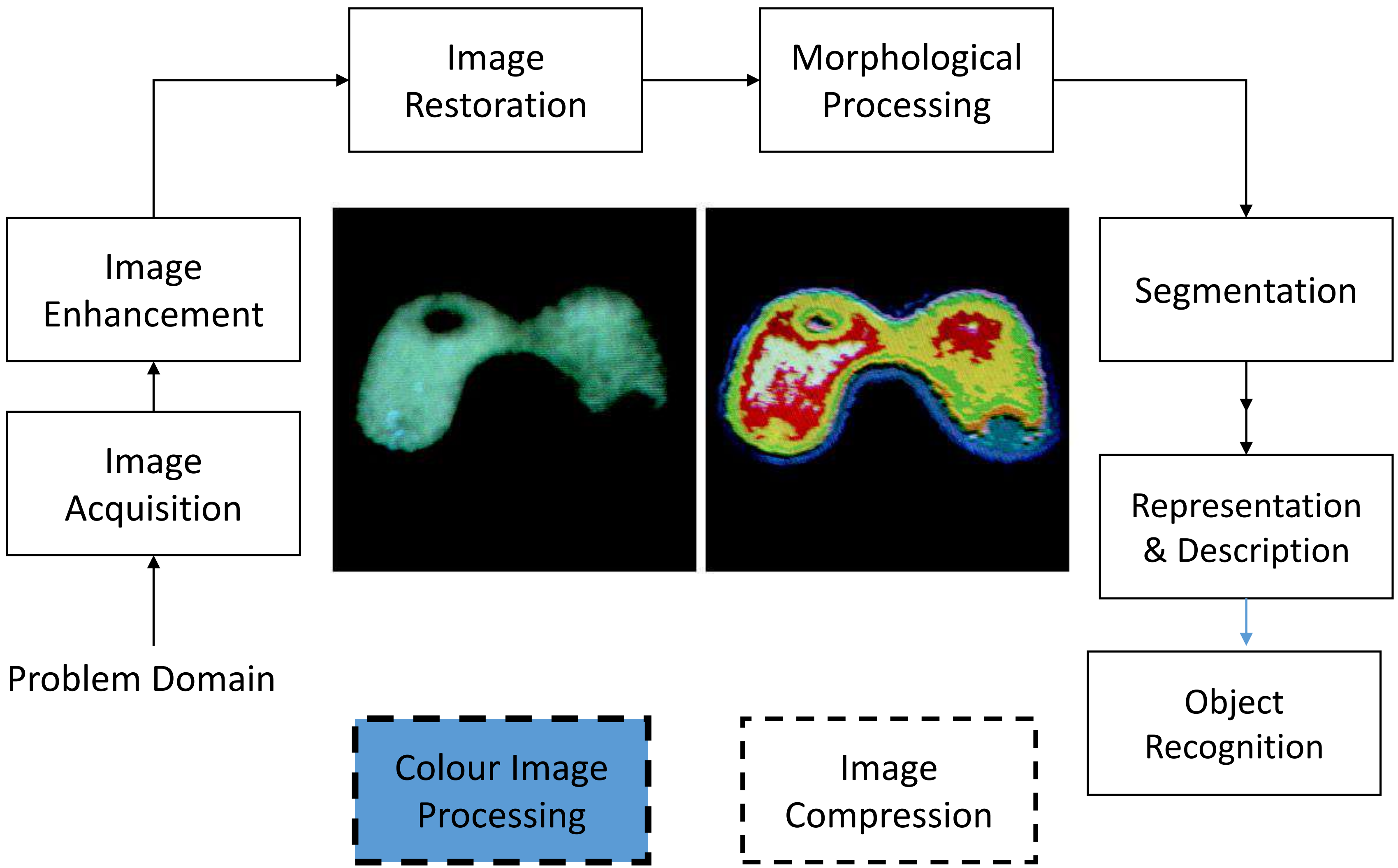
Object Recognition

- Process that assign a label such as vehicle ,animals etc. based on object descriptors
- Recognitions of individual objects in the image
- E.g Face detection , Biometrics, Tumors classifications

Key Stages in Digital Image Processing: Image Compression



Key Stages in Digital Image Processing: Colour Image Processing



Sample Questions

1)Identify the majors steps in Digital Image processing.

2)Mathematical formula for Image representation.

3)Differentiate Image Enhancement and Image restoration.

Multiple Choice Questions

What is the first and foremost step in Image Processing?
a) Image restoration
b) Image enhancement
c) Image acquisition
d) Segmentation

Answer is
c) Image acquisition

In which step of the processing, assigning a label (e.g., “vehicle”) to an object based on its descriptors is done?

a) Object recognition
b) Morphological processing
c) Segmentation
d) Representation & description

Answer is
a) Object recognition

What role does the segmentation play in image processing?

a) Deals with extracting attributes that result in some quantitative information of interest
b) Deals with techniques for reducing the storage required saving an image, or the bandwidth required transmitting it
c) Deals with partitioning an image into its constituent parts or objects
d) Deals with property in which images are subdivided successively into smaller regions

Answer is
c) Deals with partitioning an image into its constituent parts or objects

Which of the following step deals with tools for extracting image components those are useful in the representation and description of shape?

a) Segmentation
b) Representation & description
c) Compression
d) Morphological processing

Answer is
d) Morphological processing

Fields that use Image Processing

Examples of Fields that Use Digital Image Processing

- Electromagnetic energy spectrum

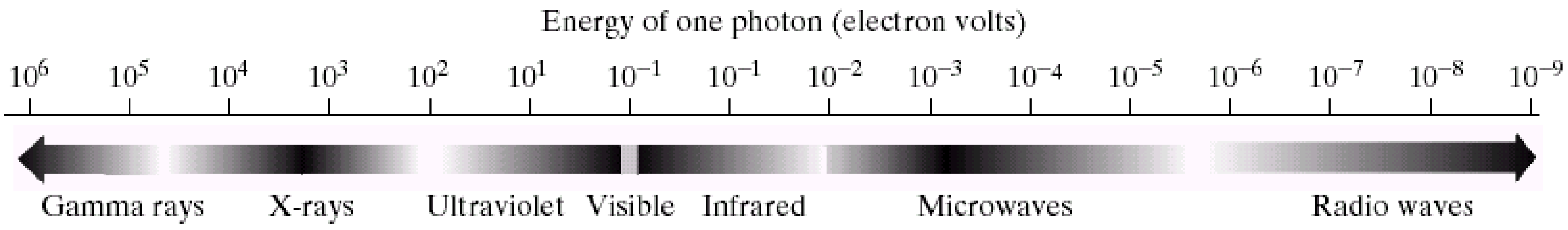


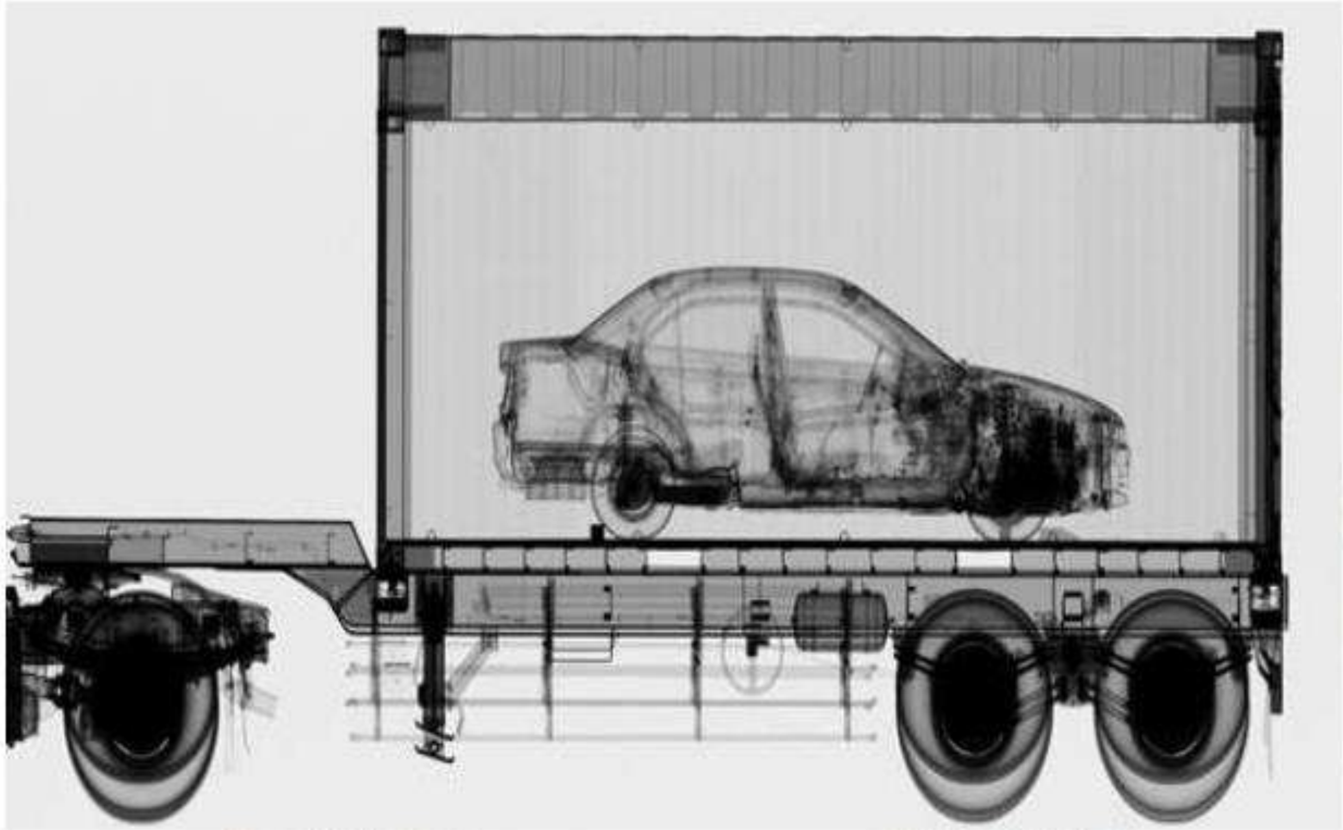
FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

- **GAMMA-RAY IMAGING**
- **X-RAY IMAGING**
- **IMAGING IN THE ULTRAVIOLET BAND**
- **IMAGING IN THE VISIBLE AND INFRARED BANDS**
- **IMAGING IN THE MICROWAVE BAND**
- **IMAGING IN THE RADIO BAND**

Gamma Rays Imaging

- Gamma rays are an energetic form of electromagnetic radiation
- Produced by radioactivity or nuclear or subatomic processes such as electron-positron annihilation.
- Gamma rays are the rays that have the most powerful of emerge power in comparison with alpha and beta rays
- Gamma rays are so light that has a wavelength higher than the other beam.
- Gamma Camera Equipment is a tool used in nuclear medical depiction.
- To see and analyze or diagnose overview of the human body by detecting the radiation beam from a radio isotope that is inserted into the patient's body

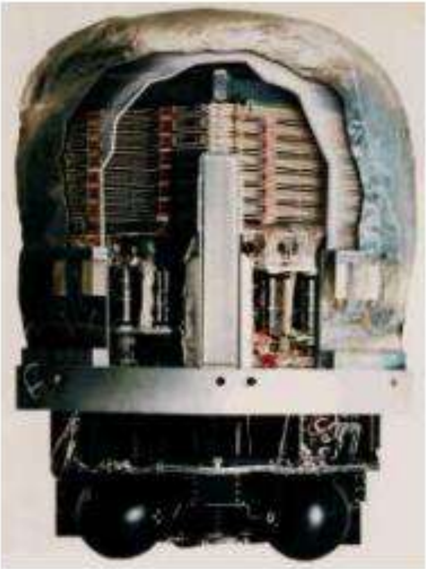
Gamma Ray Container Scanner



Application

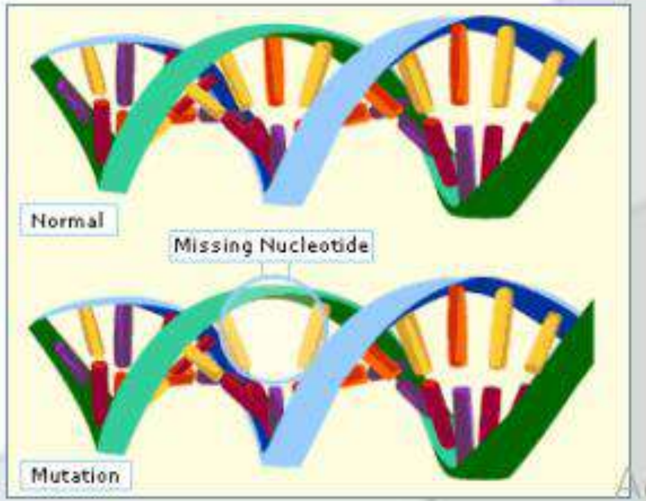
Fermi Gamma-ray Space Telescope

Compton Gamma Ray Observatory



Energetic Gamma Ray Experiment Telescope

Preserving of sorghum



Mutasi Gen

X RAYS

- X-rays are among the oldest sources of EM radiation used for imaging.
- Use of X-rays is medical diagnostics, and extensively in industry and other areas, like astronomy.
- X-rays for medical and industrial imaging are generated using an X-ray tube, which is a vacuum tube with a cathode and anode.
- The cathode is heated, causing free electrons to be released. These electrons flow at high speed to the positively charged anode.
- When the electrons strike a nucleus, energy is released in the form of X-ray radiation. The energy (penetrating power) of X-rays is controlled by a voltage applied across the anode, and by a current applied to the filament in the cathode.

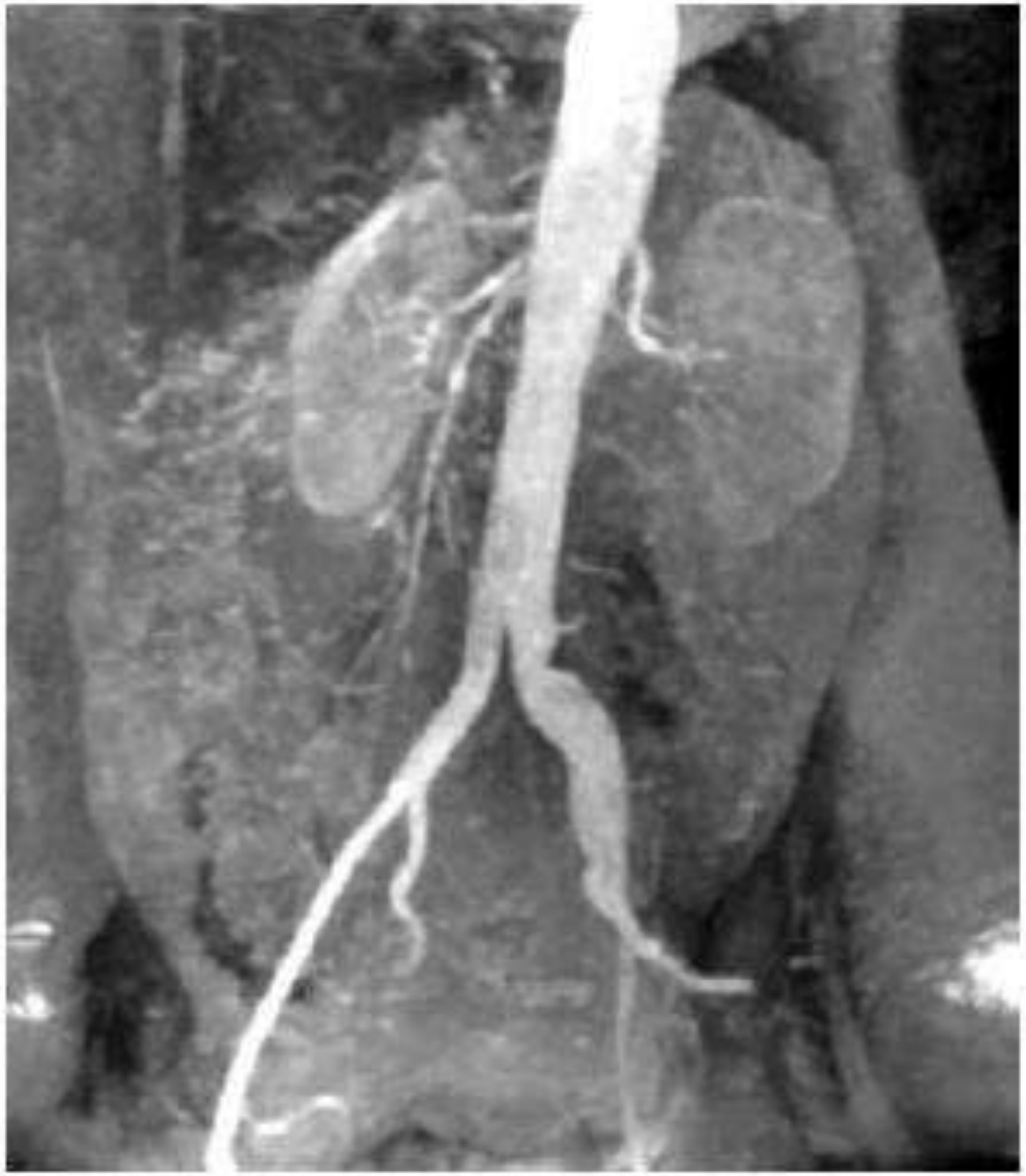
- In digital radiography, digital images are obtained by one of two methods:

    (1) by digitizing X-ray films; or

    (2) by having the X-rays that pass through the patient fall directly onto devices (such as a phosphor screen) that convert X-rays to light.

- The light signal in turn is captured by a light-sensitive digitizing system.

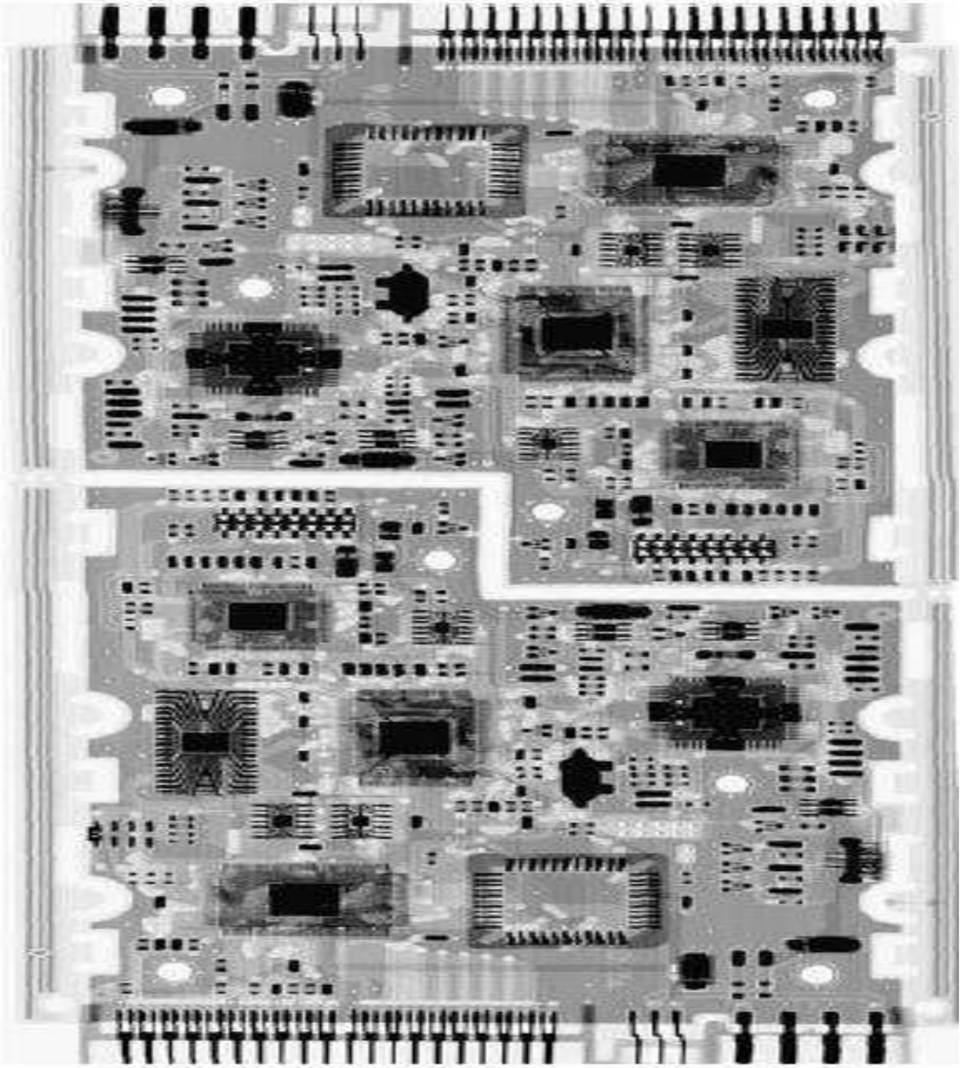
- Angiography is another major application in an area called contrast enhancement radiography.
- This procedure is used to obtain images (called angiograms) of blood vessels.
- A catheter (a small, flexible, hollow tube) is inserted, for example, into an artery or vein in the groin.
- The catheter is threaded into the blood vessel and guided to the area to be studied.
- When the catheter reaches the site under investigation, an X-ray contrast medium is injected through the tube.
- This enhances contrast of the blood vessels and enables the radiologist to see any irregularities or blockages



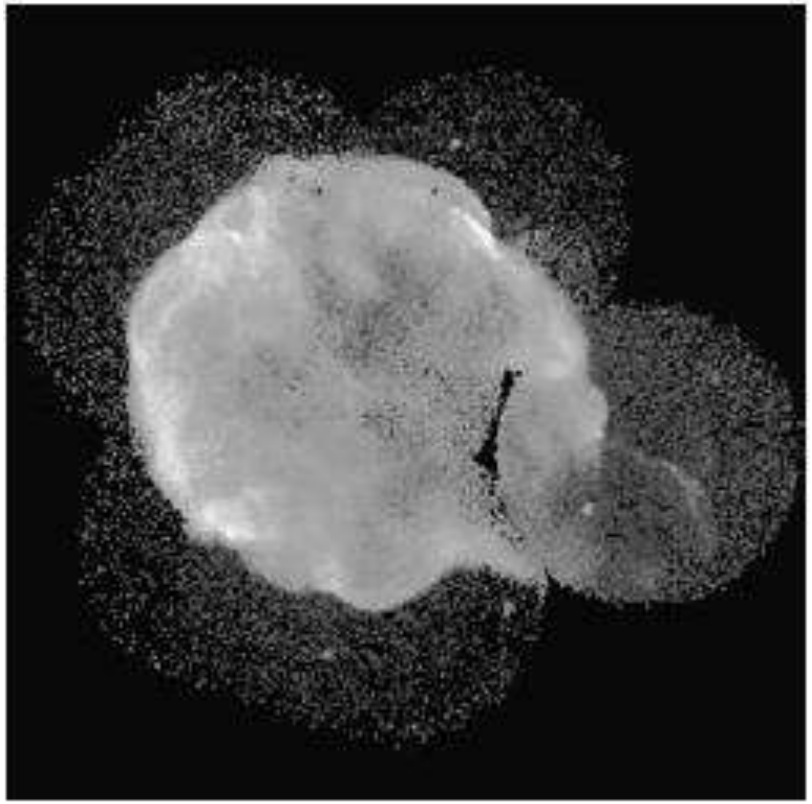
CAT

- Another important use of X-rays in medical imaging is computerized axial tomography (CAT).
- Each CAT image is a “slice” taken perpendicularly through the patient. Numerous slices are generated as the patient is moved in a longitudinal direction.
- The ensemble of such images constitutes a 3-D rendition of the inside of the body, with the longitudinal resolution being proportional to the number of slice images taken.

Higher energy X-rays, are applicable in industrial processes. Figure shows an X-ray image of an electronic circuit board are used to examine circuit boards for flaws in manufacturing, such as missing components or broken traces.



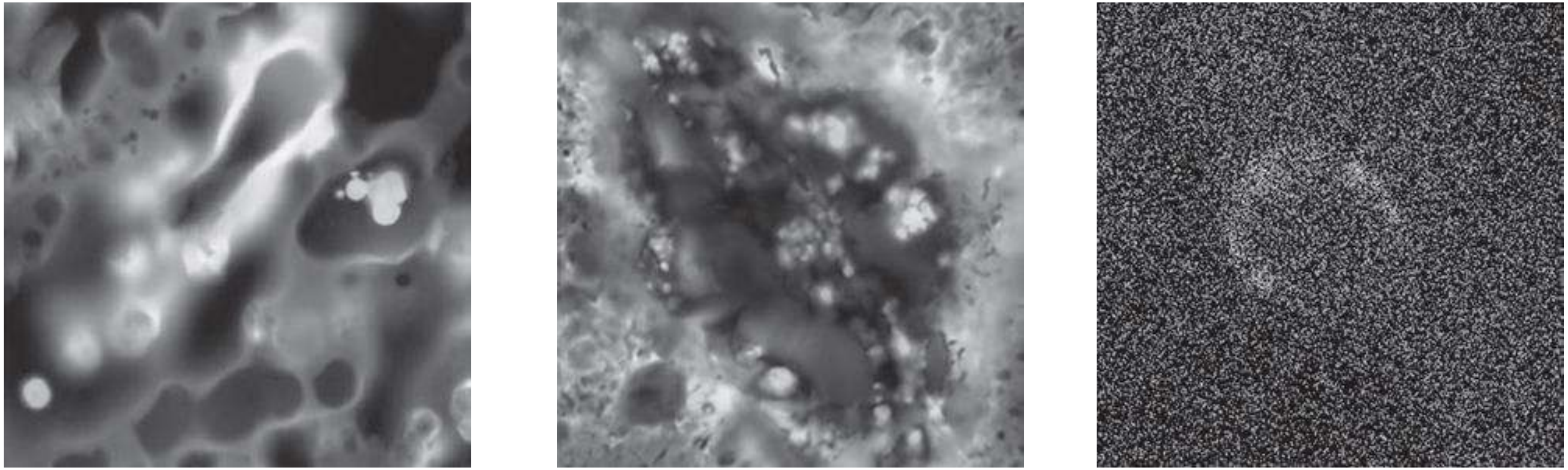
X-ray imaging in astronomy



IMAGING IN THE ULTRAVIOLET BAND

- Applications of ultraviolet “light” are varied.
- They include lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations.
- We illustrate imaging in this band with examples from microscopy and astronomy

Examples of ultraviolet imaging



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

- Ultraviolet light is used in fluorescence microscopy, one of the fastest growing areas of microscopy.
- Fluorescence microscopy is an excellent method for studying materials that can be made to fluoresce, either in their natural form (primary fluorescence) or when treated with chemicals capable of fluorescing (secondary fluorescence).

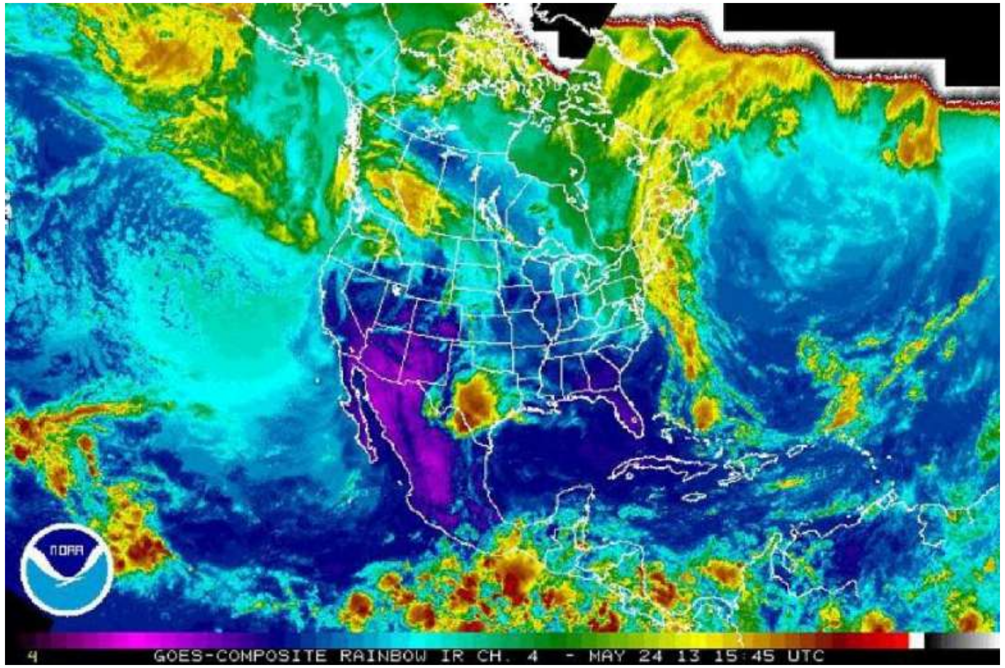
IMAGING IN THE VISIBLE AND INFRARED BANDS

- The infrared band often is used in conjunction with visual imaging, so we have grouped the visible and infrared bands

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.53–0.61	Measures plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.78–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content: soil/vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Short-wave infrared	2.09–2.35	Mineral mapping

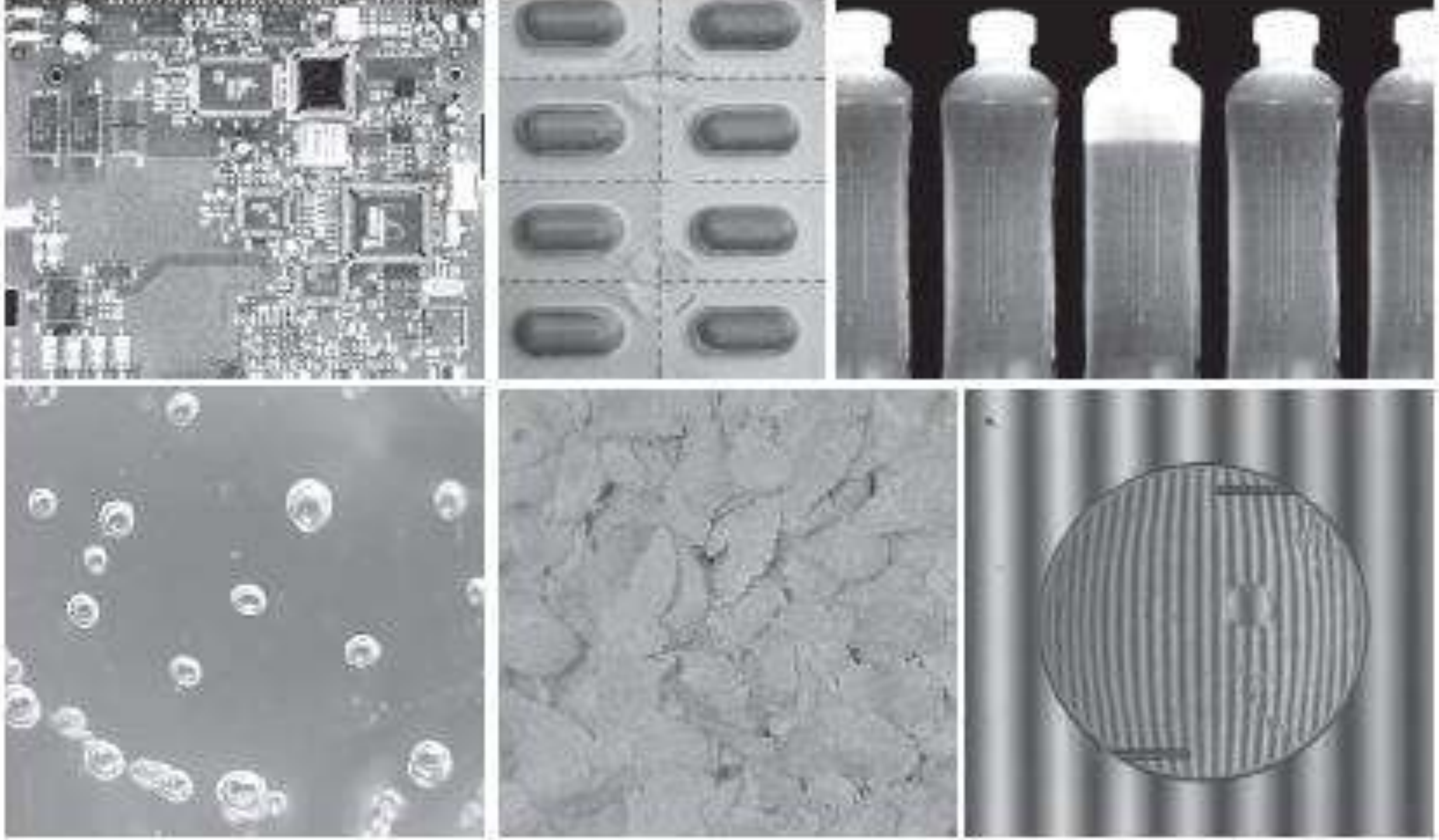
- Images of population centers are used over time to assess population growth and shift patterns, pollution, and other factors affecting the environment.
- The differences between visual and infrared image features are quite noticeable in these images.

- Weather observation and prediction also are major applications of multispectral imaging from satellites.



- The infrared system operates in the band 10.0 to 13.4 mm
- Has the unique capability to observe faint sources of visible, near infrared emissions present on the Earth’s surface, including cities, towns, villages, gas flares, and fires.
- Even without formal training in image processing, it is not difficult to imagine writing a computer program that would use these images to estimate the relative percent of total electrical energy used by various regions of the world.

- A major area of imaging in the visible spectrum is in automated visual inspection of manufactured goods. Figure shows some examples.(a) is a controller board for a CD-ROM drive.
- A typical image processing task with products such as this is to inspect them for missing parts (the black square on the top, right quadrant of the image is an example of a missing component).



a b c
d e f

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

IMAGING IN THE MICROWAVE BAND

- The principal application of imaging in the microwave band is radar.
- The unique feature of imaging radar is its ability to collect data over virtually any region at any time
- Regardless of weather or ambient lighting conditions.
- Some radar waves can penetrate clouds, and under certain conditions, can also see through vegetation, ice, and dry sand.
- In many cases, radar is the only way to explore inaccessible regions of the Earth’s surface.
- An imaging radar works like a flash camera in that it provides its own illumination (microwave pulses) to illuminate an area on the ground and take a snapshot image.

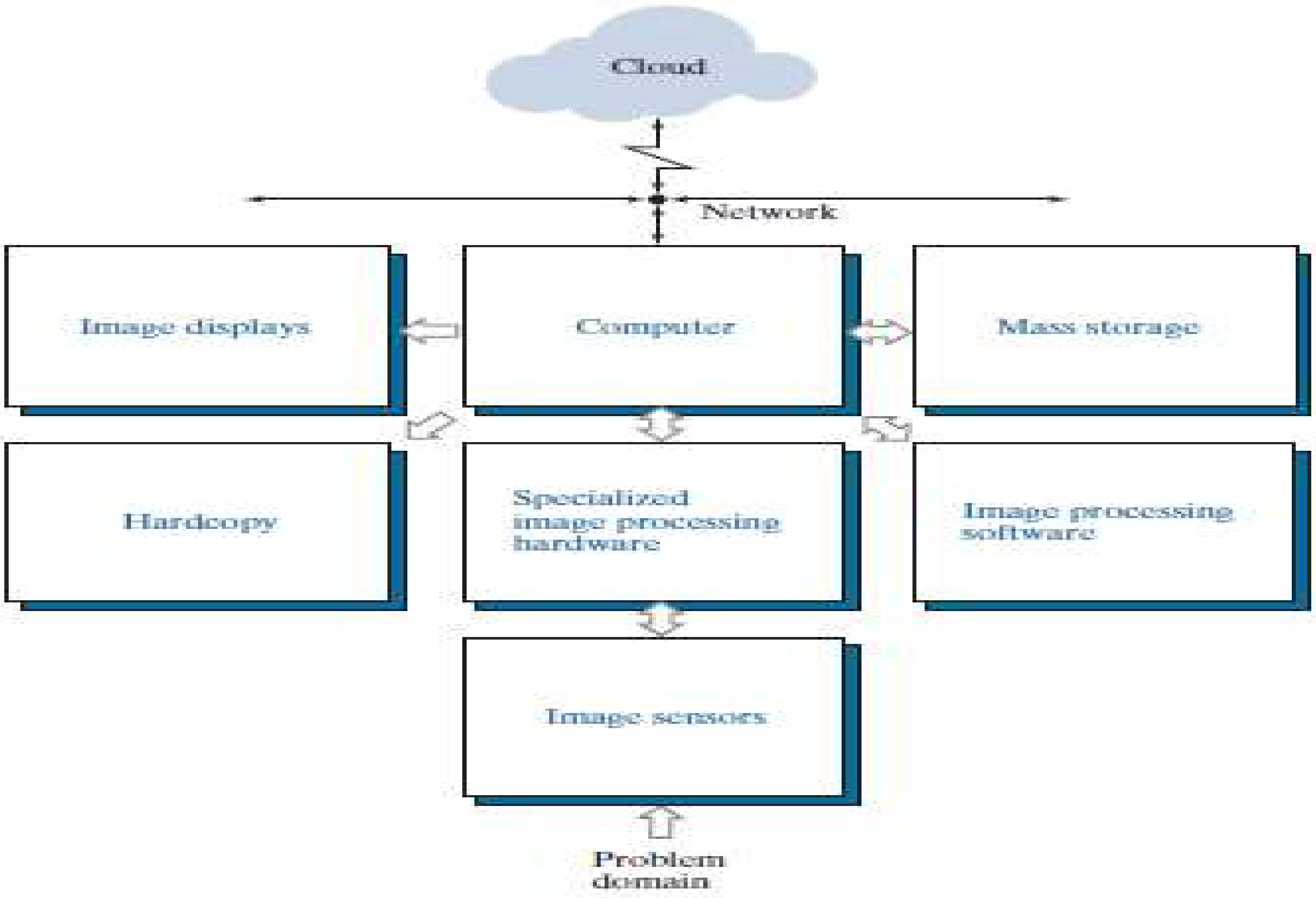
- Instead of a camera lens, a radar uses an antenna and digital computer processing to record its images.
- In a radar image, one can see only the microwave energy that was reflected back toward the radar antenna.

Space borne radar image of mountainous region in southeast Tibet.

IMAGING IN THE RADIO BAND

- Imaging at the other end of the spectrum (gamma rays)
- The major applications of imaging in the radio band are in medicine and astronomy.
- In medicine, radio waves are used in magnetic resonance imaging (MRI).
- This technique places a patient in a powerful magnet and passes radio waves through the individual's body in short pulses.
- Each pulse causes a responding pulse of radio waves to be emitted by the patient's tissues.
- The location from which these signals originate and their strength are determined by a computer.
- Produces a two-dimensional image of a section of the patient.
- MRI can produce images in any plane.

Components of an Image Processing System



i) Image Sensors

- With reference to sensing, two elements are required to acquire digital image.
- The first is a physical device that is sensitive to the energy radiated by the object we wish to image

ii) Specialize image processing hardware

- – It consists of the digitizer just mentioned, plus hardware that performs other primitive operations such as an arithmetic logic unit, which performs arithmetic such addition and subtraction and logical operations in parallel on images

iii) Computer

- It is a general purpose computer and can range from a PC to a supercomputer depending on the application.
- In dedicated applications, sometimes specially designed computer are used to achieve a required level of performance.

iv) Software

- It consist of specialized modules that perform specific tasks a well designed package also includes capability for the user to write code, as a minimum, utilizes the specialized module.
- More sophisticated software packages allow the integration of these modules.

v) Mass storage

- This capability is a must in image processing applications.
- An image of size 1024 x1024 pixels, in which the intensity of each pixel is an 8- bit quantity requires one megabytes of storage space if the image is not compressed.
- Image processing applications falls into three principal categories of storage

  i) Short term storage for use during processing

  ii) On line storage for relatively fast retrieval

  iii) Archival storage such as magnetic tapes and disks

vi) Image displays

- Image displays in use today are mainly color TV monitors.
- These monitors are driven by the outputs of image and graphics displays cards that are an integral part of computer system

vii) Hardcopy devices

- The devices for recording image includes laser printers, film cameras, heat sensitive devices inkjet units and digital units such as optical and CD ROM disk.
- Films provide the highest possible resolution, but paper is the obvious medium of choice for written applications.

viii) Networking

- It is almost a default function in any computer system in use today because of the large amount of data inherent in image processing applications.
- The key consideration in image transmission bandwidth.

Sample Questions

1) Explain the applications and fields where image processing is used.
2) Describe the components of Image processing with neat diagram.

MCQ

1) The infrared system operates in the band:

A) 10.0 to 13.4 mm

B)10.0 to 12.0 mm

C)8.0 to 10.0 mm

D)10.0 to 11.0 mm

2) The principal application of imaging in the microwave band is:

a)X-ray

b) Radar

c)Gamma ray

d) Infra red

3.Which imaging is for weather forecasting

a)Ultraviolet

b)X-ray

c)MRI

d)Near Infrared

Elements of visual perception

- Eye characteristics
  - nearly spherical
  - approximately 20 mm in diameter
  - three membranes
- cornea (transparent) & sclera (opaque) outer cover
- choroid contains a network of blood vessels, heavily pigmented to reduce amount of extraneous light entering the eye. Also contains the iris diaphragm (2-8 mm to allow variable amount of light into the eye)
- retina is the inner most membrane, objects are imaged on the surface

Imaging in the eye

- Variable thickness lens: thick for close focus, thin for distant focus
- Distance of focal center of the lens to the retina (14-17 mm)
- Image of a 15m tree at 100m
    - 15/100 = X/17 or approximately 2.55 mm
- Image is almost entirely on the fovea

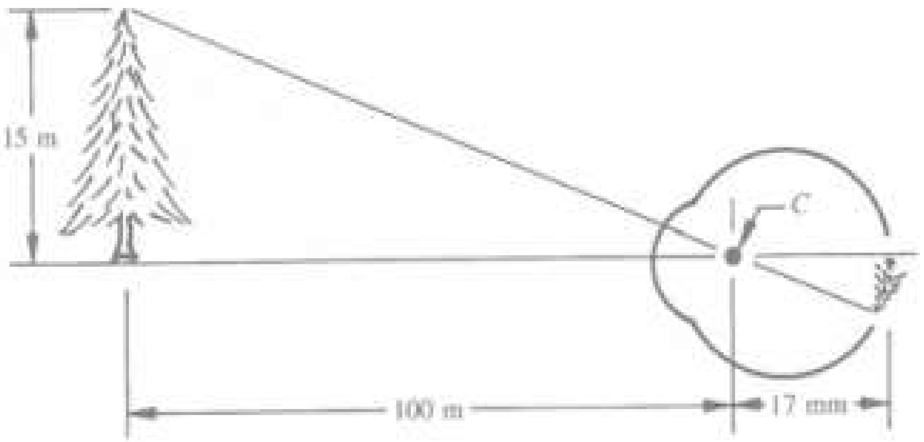


Image Formation

- First step in understanding Image formation is to known about Physical Illumination
- Second is Reflectance Model

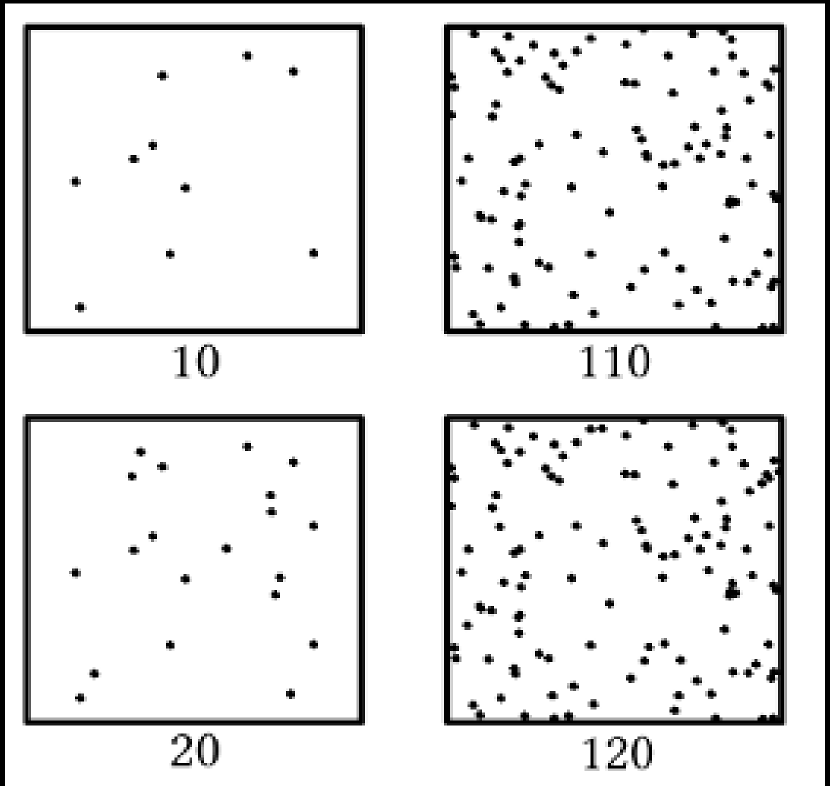
1)Illumination

- Fundamental components of Image formation process which generates sense in our visual organs.
- The strength of Sensation which is the sensation of brightness can be quantified by averaging the responses of many human observers.
- The average response ,i.e. the psychovisual sensation is determined at different spectral wavelength.
- The peak spectral sensitivity of a human observer happens at 555 nm Wavelengths

1)Illumination

- Equal amounts of Luminous flux produce equal brightness, which is proportional to the logarithm of luminous flux.
- Fechner's law defines the brightness by the relation

$$B = K \log\left(\frac{F}{F_0}\right)$$

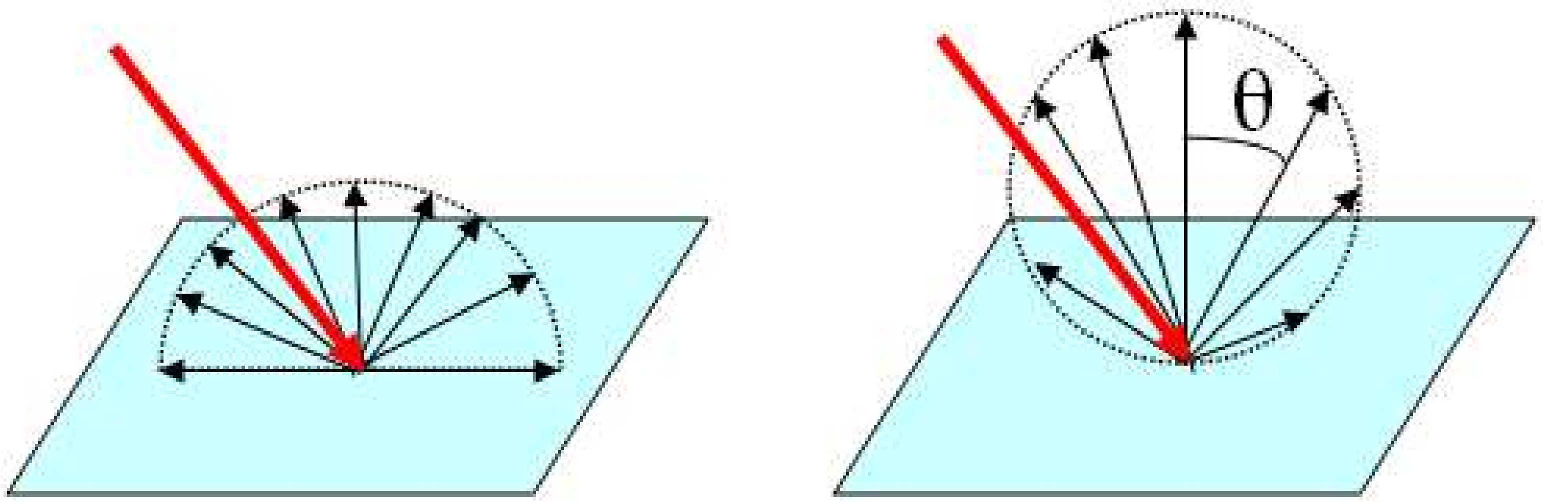


10
110
20
120

2) Reflectance Model

- Depending on the nature reflection
- Three categories :
    1. Lambertian
    2. Specular
    3. Hybrid

1.Lambertian reflection



1.Lambertian reflection

- Lambertian reflectance usually refers to the reflection of light by an object
- Light are reflected in all directions and diffused one
- It can be used to refer to the reflection of any wave
- The reflectance from the wall paint with flat paints, papers , fabrics, ground surface
- For example, in ultrasound imaging, "rough" tissues are said to exhibit Lambertian reflectance.

- Entire incident light in all directions covering solid angle 2π radians
- Equally bright in all directions
- The reflectance map of Lambertian surface may be modelled as

$$I_L = E_0 A \cos\theta$$

Where E_0 is the strength of the incident light source

A is the surface area of the Lambertian patch

θ is the angle of Incidence

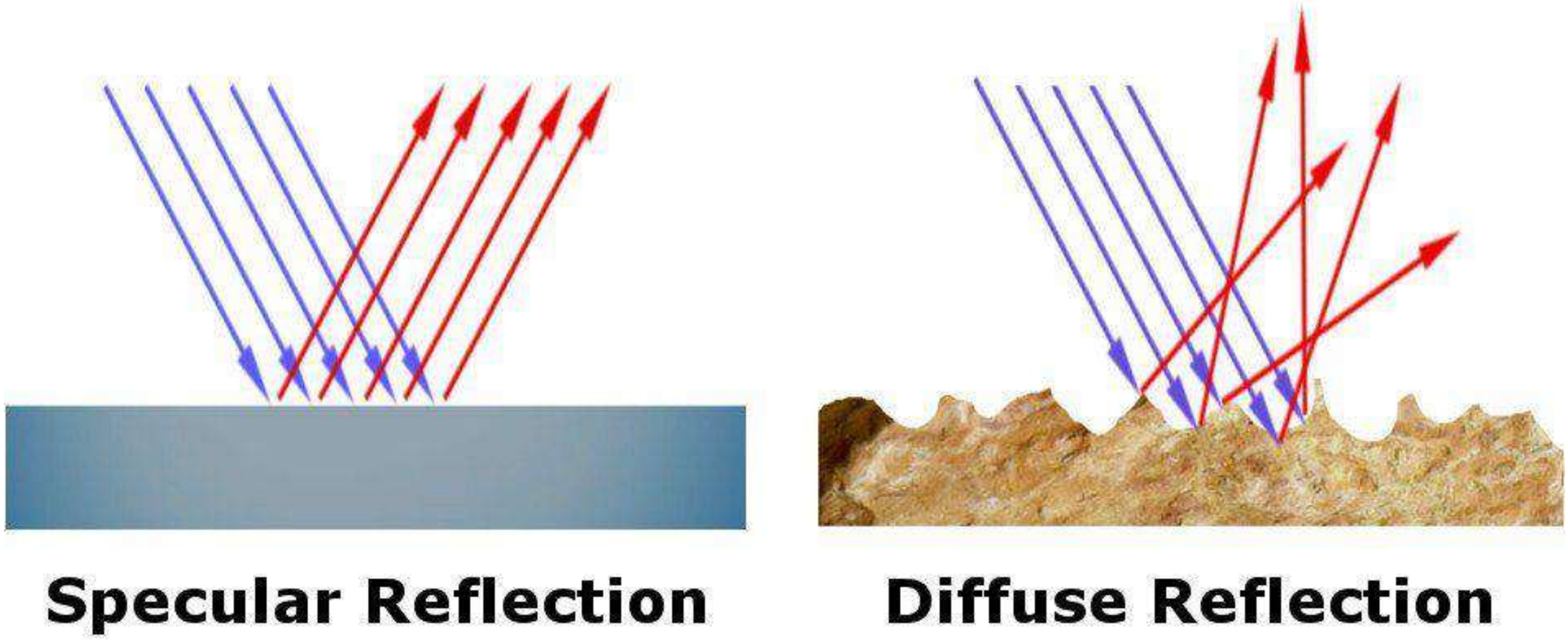
For an Image

- The reflectance relationship for a
- Lambertian model of image $E(x,y)$

$$E(x,y) = A\,(n \cdot s)$$

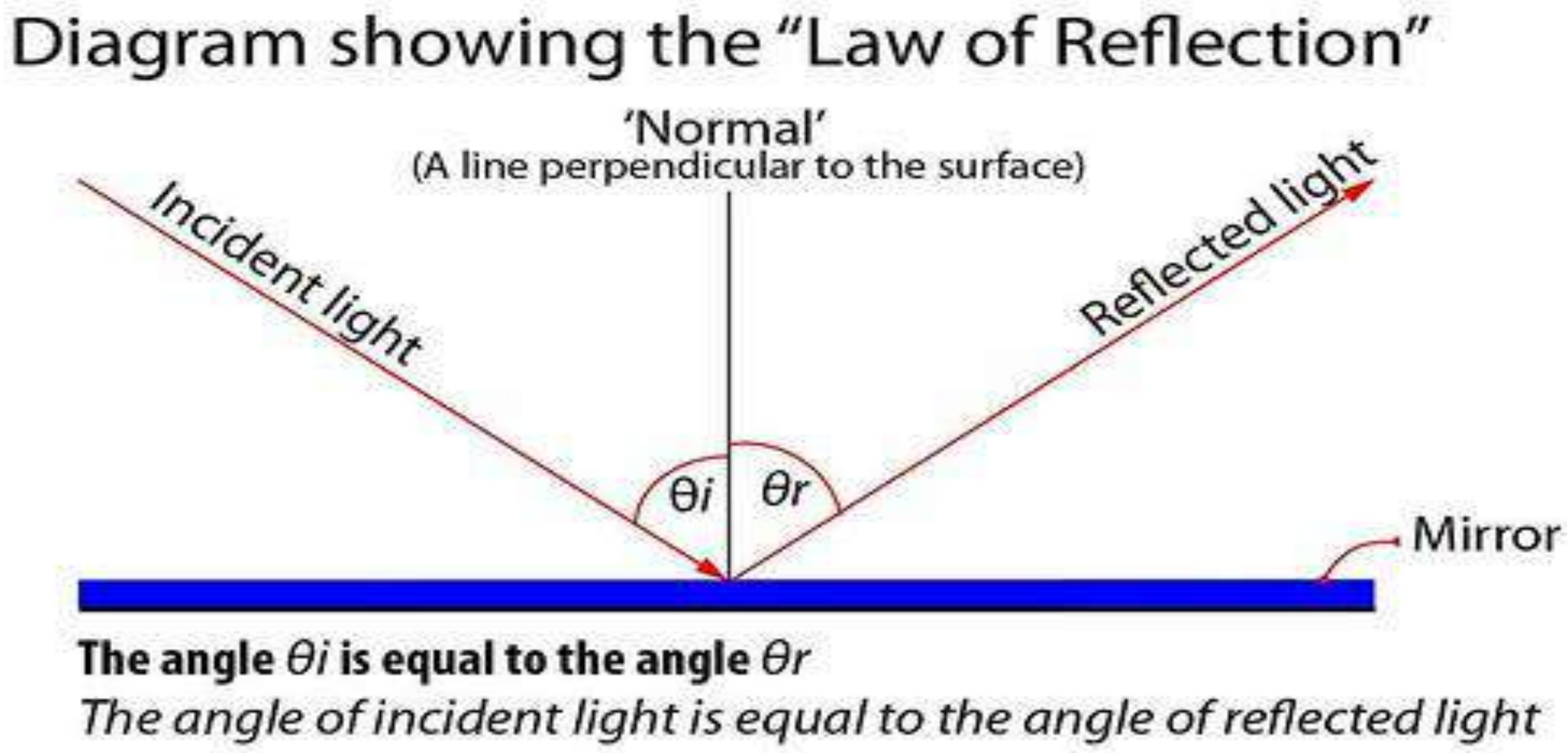
  - n = surface normal (unit vector)
  - s = source direction (unit vector)
  - A = constant related to illumination intensity and surface

2.Specular reflectance



- In specular reflection a mirror surface reflects a beam of light so that the angle of reflection is equal to the angle of incidence
- Eg. the angle of the incoming light is the same as the angle of the outgoing/reflected light

- The diagram shows how the incoming (incident) light is reflected off the mirror at the same angle to the perpendicular line (which in geometry is called the ‘normal’).



Hybrid Reflectance

- Mostly found in Display devices
- Known as Hazes
- In real world ,neither Lambertian nor specular
- Mixture of Lambertian and Specular reflectance
- The reflectance surface of model can described as:

$$I = wI_S+(1-w)I_L$$

Where I_S&I_L specular and Lambertian intensities

Face Image processing

(a) (b) (c)

Sample Questions

- Describe various reflectance model
- Draw a diagram that show how image is formed in the eyes
- Define Fechner’s law

MCQ

1) Mixture of Lambertian and Specular reflectance is called as

a)Diffuse Model

b)Hyper Model

c)Hybrid Model

d)None of above

2)Lambertian model of image $E(x,y)$ can be:

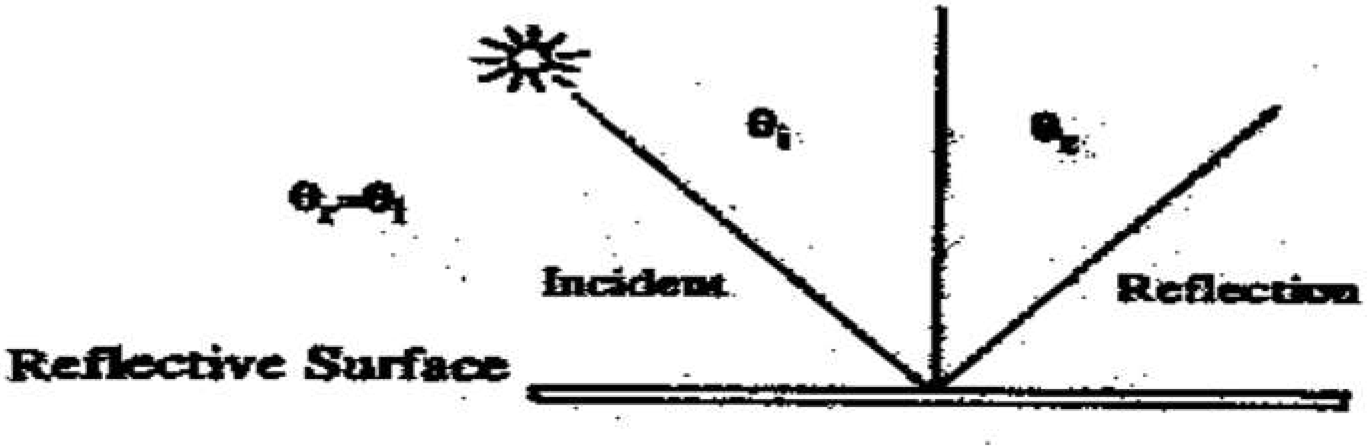
a)$E(x,y) = A\ (n\ .\ s)$

b)$E(x,y) = A\ (s)$

c)$E(x,y) = nA\ (s)$

d)$E(x,y) = A\ (n\ )$

3)The below diagram is represent which reflectance model



a)Diffuse Model
b)Hyper Model
c)Hybrid Model
d)Specular Model

References

- Tinku Acharya and Ajoy K. Ray, "*Image Processing Principles and Applications*", John Wiley and Sons publishers.

Image Sampling and Quantization

- The output of most of the image sensors is an analog signal, and we can not apply digital processing on it because we can not store it. We can not store it because it requires infinite memory to store a signal that can have infinite values.
- So we have to convert an analog signal into a digital signal.
- To create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done.
    1. **Sampling**
    2. **Quantization**
- We will discuss sampling now, and quantization will be discussed later on but for now on we will discuss just a little about the difference between these two and the need of these two steps.

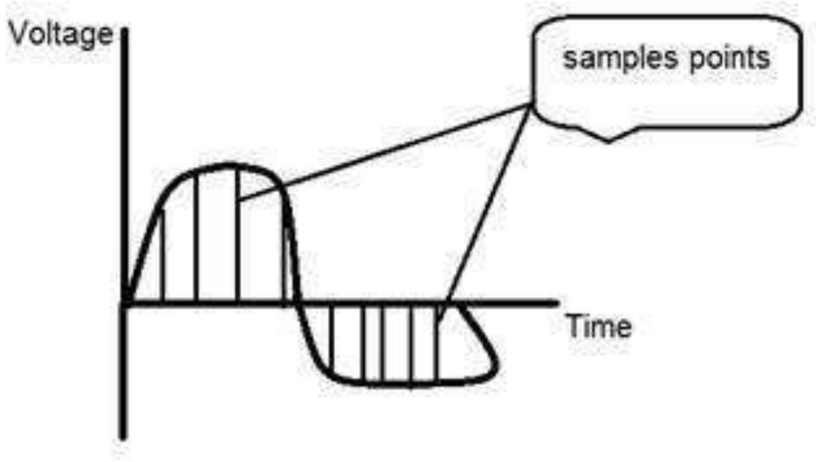
Basic idea:

- The basic idea behind converting an analog signal to its digital signal is

- To convert both of its axis (x,y) into a digital format.
- Since an image is continuous not just in its co-ordinates (x axis), but also in its amplitude (y axis), so the part that deals with the digitizing of co-ordinates is known as sampling.
- And the part that deals with digitizing the amplitude is known as quantization.

Sampling

- The term sampling refers to take samples
- We digitize x axis in sampling
- It is done on independent variable
- It is further divided into two parts , up sampling and down sampling

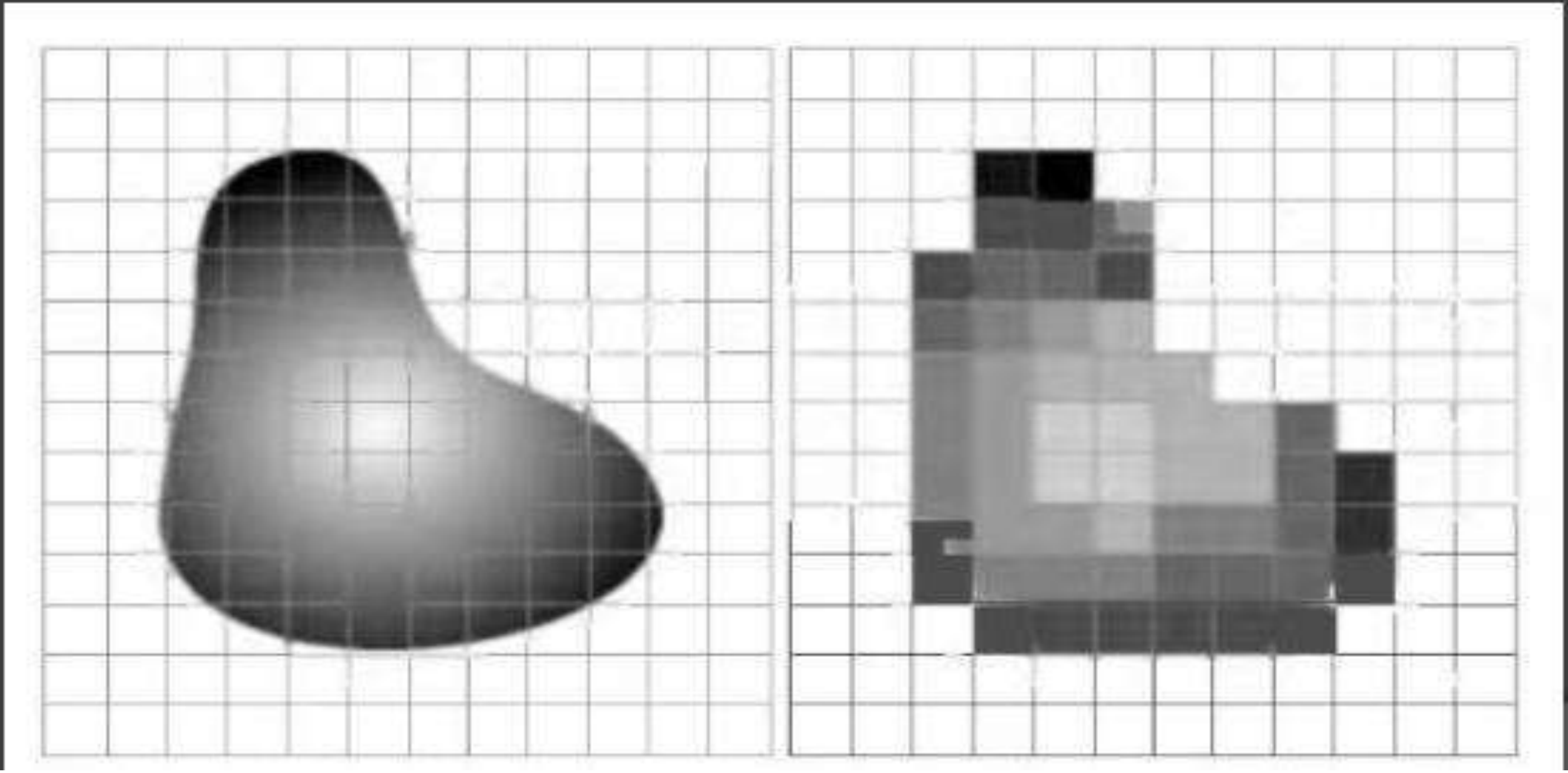


- If you will look at the above figure, you will see that there are some random variations in the signal.
- These variations are due to noise. In sampling we reduce this noise by taking samples.
- It is obvious that more samples we take, the quality of the image would be more better, the noise would be more removed and same happens vice versa.
- However, if you take sampling on the x axis, the signal is not converted to digital format, unless you take sampling of the y-axis too which is known as quantization.
- The more samples eventually means you are collecting more data, and in case of image, it means more pixels.

Image Sampling & Quantization

Sampling: Digitizing the coordinate value.

Quantization: Digitizing the amplitude value.



Recall: a pixel is a point…

- It is NOT a box, disc or teeny wee light
- It has no dimension
- It occupies no area
- It can have a coordinate
- *More* than a point, it is a *SAMPLE*

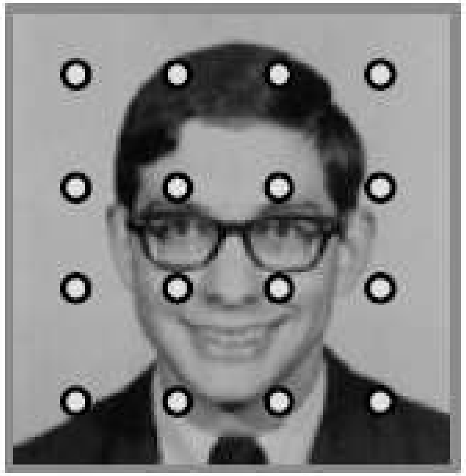
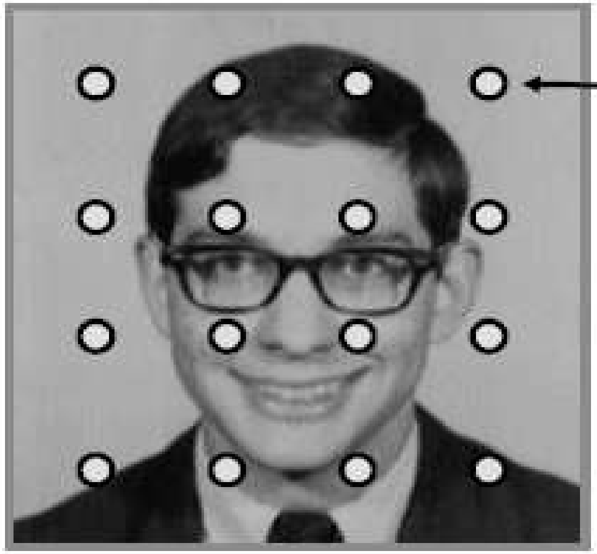


Image Sampling

- An image is a 2D rectilinear array of samples
  - Quantization due to limited intensity resolution
  - Sampling due to limited spatial and temporal resolution



Pixels are infinitely small point samples

Sampling and Reconstruction

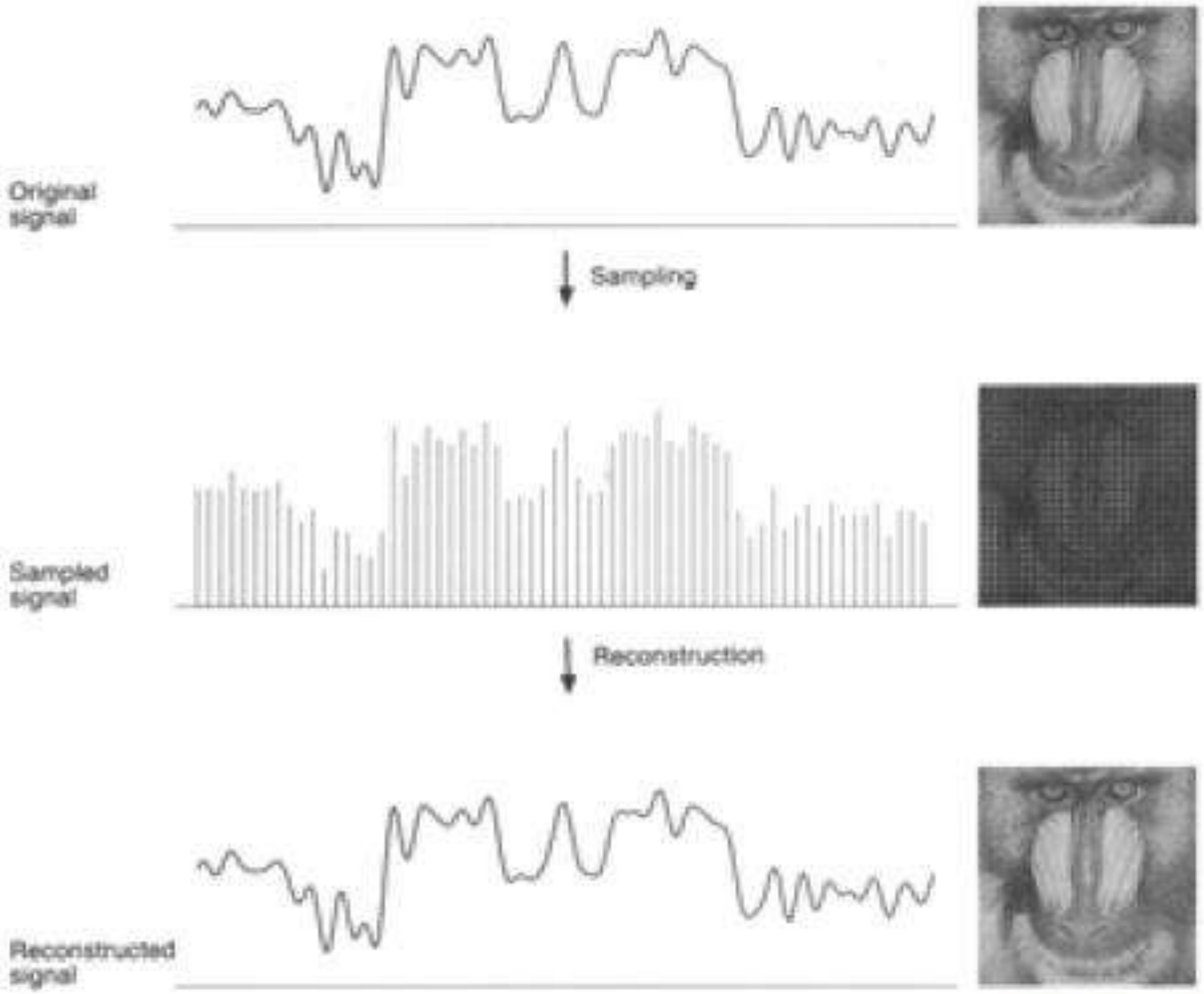


Figure 19.9 FvDFH

Image Sampling

- A static image is a two dimensional spatially varying signal.
- A sampling period should be smaller than or at the most equal to half of the period of the finest detail in the image according to Nyquist rate.
- This implies that sampling frequency along x axis $\omega_x \geq 2\omega_x^L$ and along y axis $\omega_y \geq 2\omega_y^L$

where ω_x^L and ω_y^L are the limiting factors along x and y axis

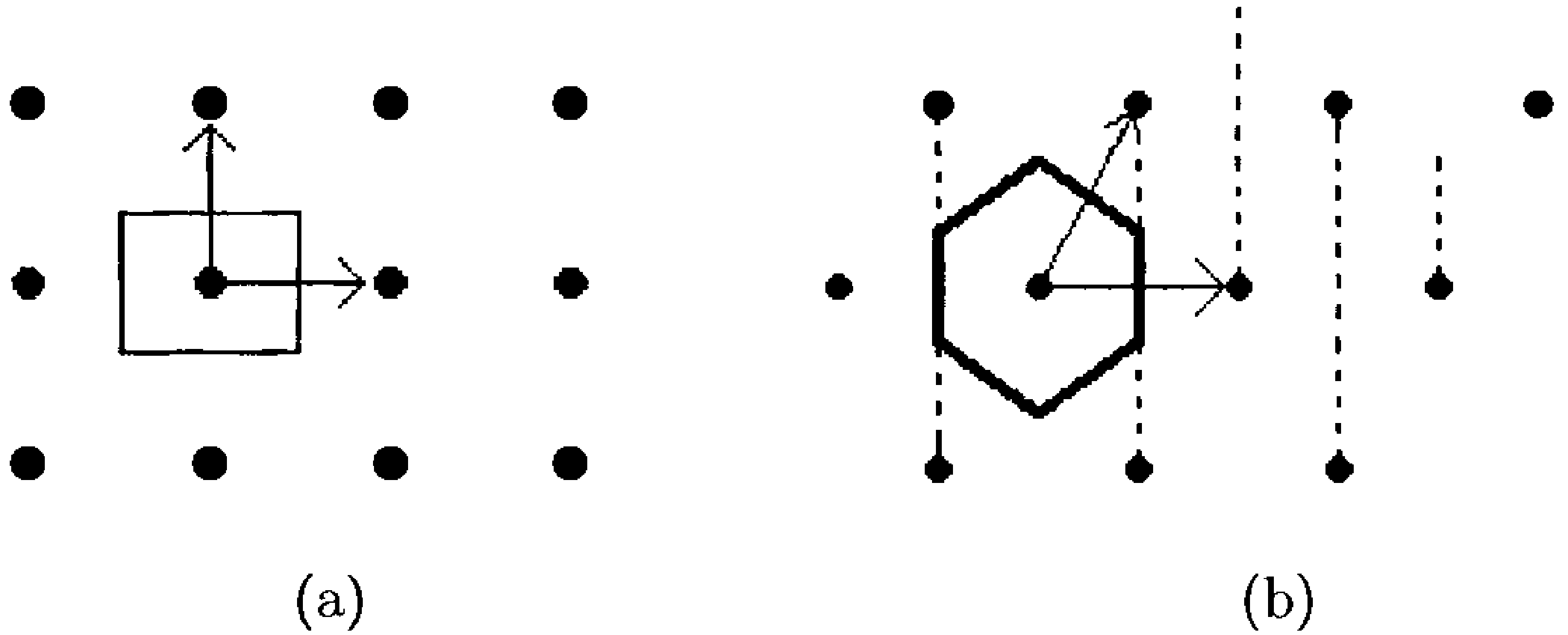


Fig. 2.5 (a) Rectangular and (b) hexagonal lattice structure of the sampling grid.

- $\Delta x\ is$ chosen sampling along x-axis then,

$$\Delta x \leq \frac{\pi}{\omega_x^L}$$

- $\Delta y\ is$ chosen sampling along y-axis then,

$$\Delta y \leq \frac{\pi}{\omega_y^L}$$

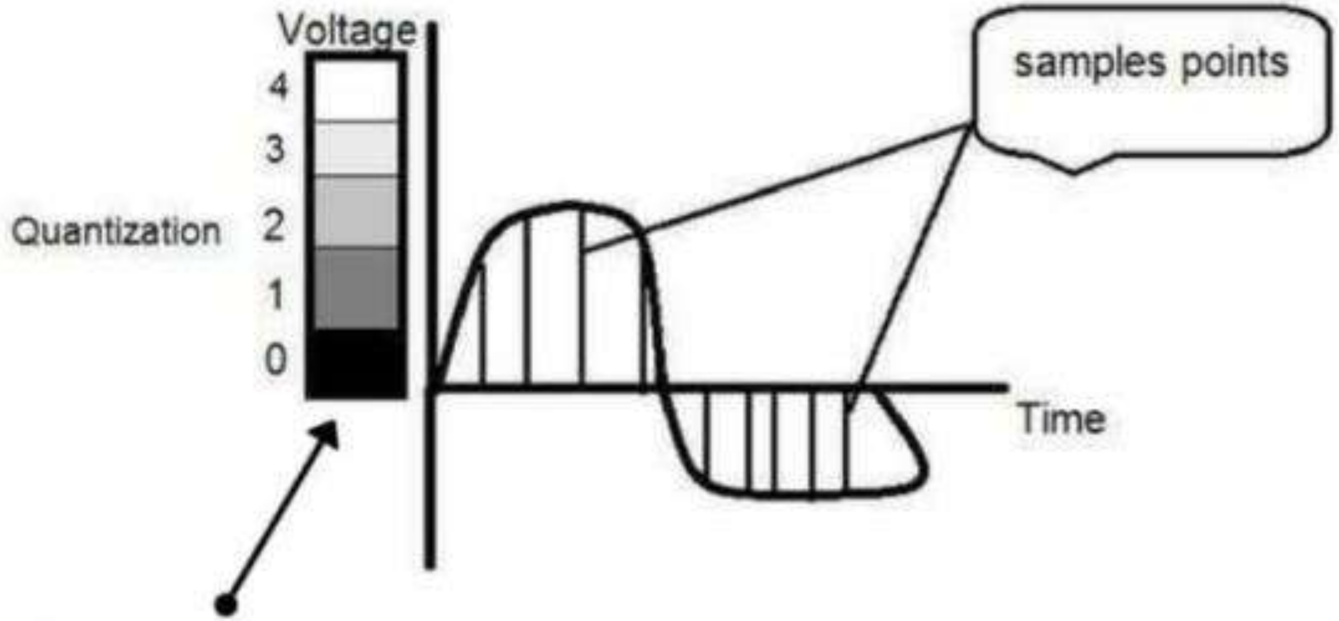
If $\Delta x\ and\ \Delta y$ are smaller, its called oversampling and $\Delta x\ and\ \Delta y$ are larger then it is called undersampling.

- If images are oversampled or exactly sampled, it is possible to reconstruct the band limited image
- If images are undersampled , there will be spectral overlapping which results in aliasing effect
- This aliasing error can be reduced by substantially by using a presampling filter

Fig. Images sampled at 256 × 256, 128 × 128 , 64 × 64, 32 × 32, and 16 × 16 rectangular sampling grids.

Image Quantization

- It is opposite side of Sampling
- As sampling is done in x-axis whereas Quantization is done on the y-axis
- Digitizing the amplitudes is called Quantization
- Quantization involves in assigning a single value to each samples
- We divide signal amplitudes into quanta(partitions)



Vertically ranging values have been quantized into 5 different levels or partitions.

Relation of Quantization and gray level resolution:

Number of quantas (partitions) = Number of gray levels

- Number of gray levels here means number of different shades of gray.
- To improve image quality, we number of gray levels or gray level resolution up.
- If we increase this level to 256, it is known as the grayscale image.

$$\boxed{L = (2^k)}$$

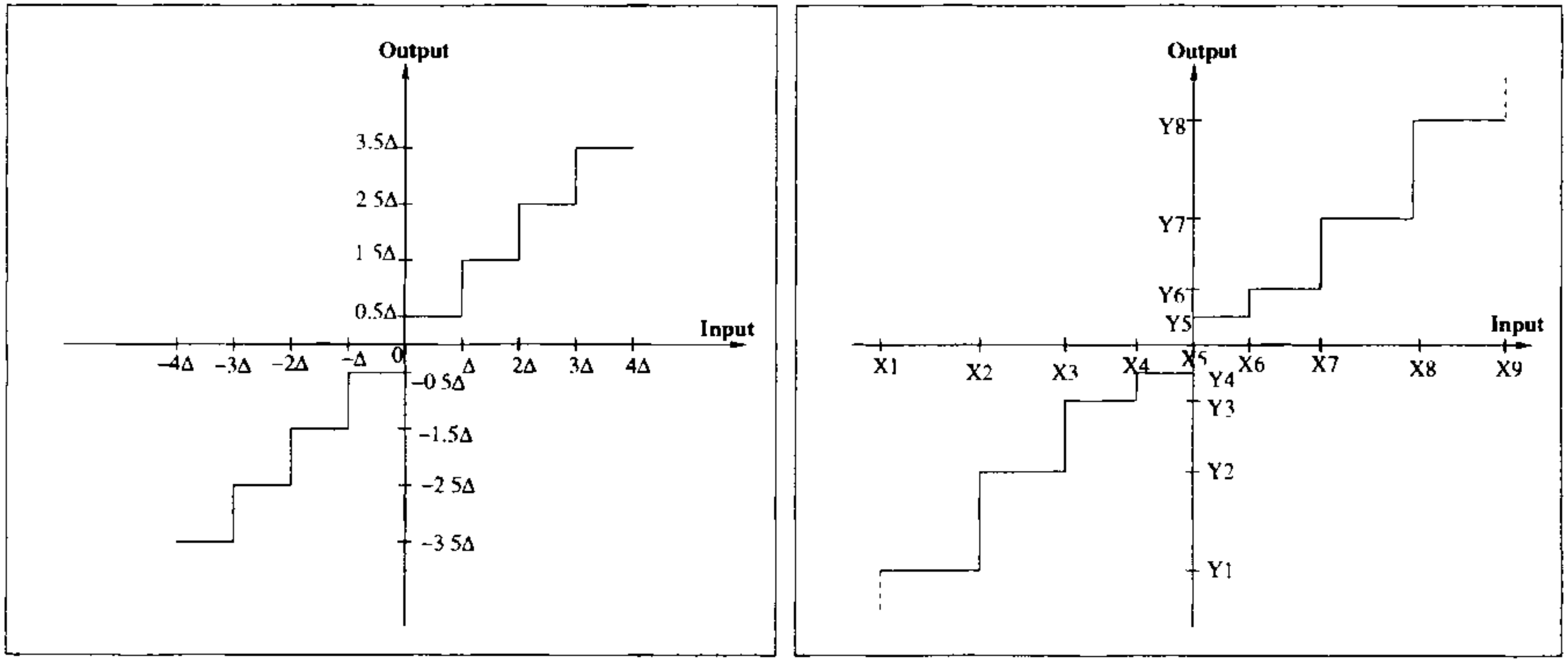
Where,

L = gray level resolution

k = gray level

Types of Quantization

- Uniform Quantization
- Random Quantization



(a) (b)

Fig. 2.7 Two-dimensional (a) uniform quantization (b) nonuniform quantization.

- As the number of quantization levels increases, the quantized image will be approximately to the original image with less error
- When quantization levels are chosen equally spaced at equal interval, it is known Uniform Quantization
- If levels are spaced at different intervals, it is called Random Quantization or Non-uniform Quantization
- When sample intensity values are equally like to occur at different intervals, then Uniform Quantization is followed
- If samples assumes values in a small range quite frequently and other values infrequently,then Non-uniform Quantization is used


Original 8bit
3bit Quantized

Quantification IV


N=64


N=32


N=16


N=8


N=4


N=2

Relationship with pixels

- Since a pixel is a smallest element in an image. The total number of pixels in an image can be calculated as

  Pixels = total no of rows * total no of columns.

- Lets say we have total of 25 pixels, that means we have a square image of 5 X 5.
- Then as we have discussed above in sampling, that more samples eventually result in more pixels. So it means that of our continuous signal, we have taken 25 samples on x axis. That refers to 25 pixels of this image.

MCQ

1) 1. A continuous image is digitised at _______ points.
a) random
b) vertex
c) contour
d) sampling

- 1. A continuous image is digitised at _______ points.
  a) random
  b) vertex
  c) contour
  d) sampling

2. The transition between continuous values of the image function and its digital equivalent is called _______________

a) Quantisation

b) Sampling

c) Rasterisation

d) None of the Mentioned

2. The transition between continuous values of the image function and its digital equivalent is called _______________

a) Quantisation

b) Sampling

c) Rasterisation

d) None of the Mentioned

3.Quantitatively, spatial resolution cannot be represented in which of the following ways

a) line pairs

b) pixels

c) dots

d) none of the Mentioned

3.Quantitatively, spatial resolution cannot be represented in which of the following ways

a) line pairs

b) pixels

c) dots

d) none of the Mentioned

1)Write in detail about Image Sampling and Quantization

2)Identify the different types of Quantization in Images

Reference

- T1. Tinku Acharya and Ajoy K. Ray, "*Image Processing Principles and Applications*", John Wiley and Sons publishers.

Binary Image Processing

Binary Images

- Images only consist of two colors (tones): white or black

Binary Images

- Images with only two values (0 or 1)
- Simple to process and analyze
- Very useful for industrial applications

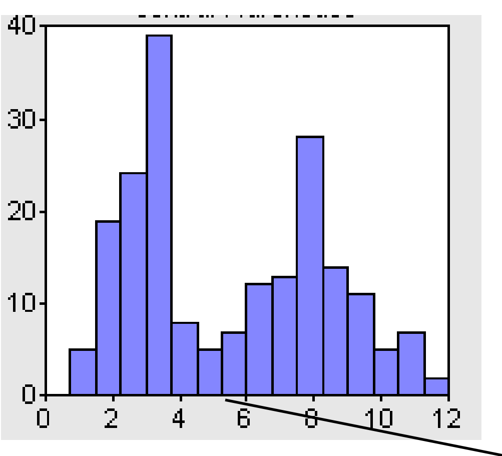
Binary Images

- Obtained from gray-level (or color) image $g(x, y)$ by Thresholding

- Characteristic Function

$$b(x, y) = \begin{cases} 1 & \text{if } g(x, y) > T \\ 0 & \text{if } g(x, y) <= T \end{cases}$$

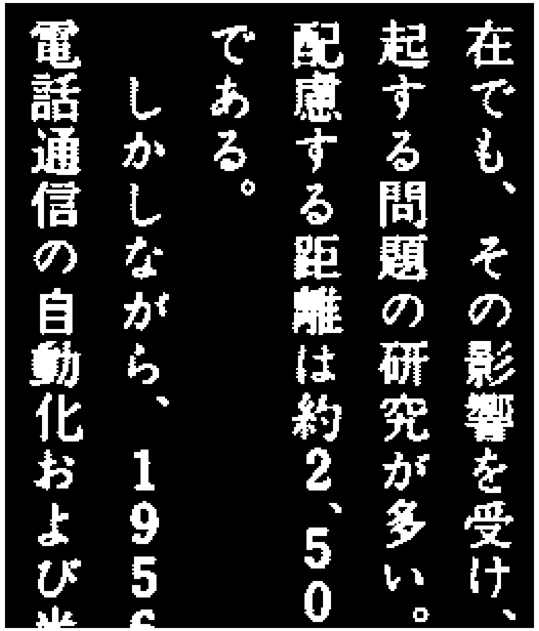
Selecting a Threshold

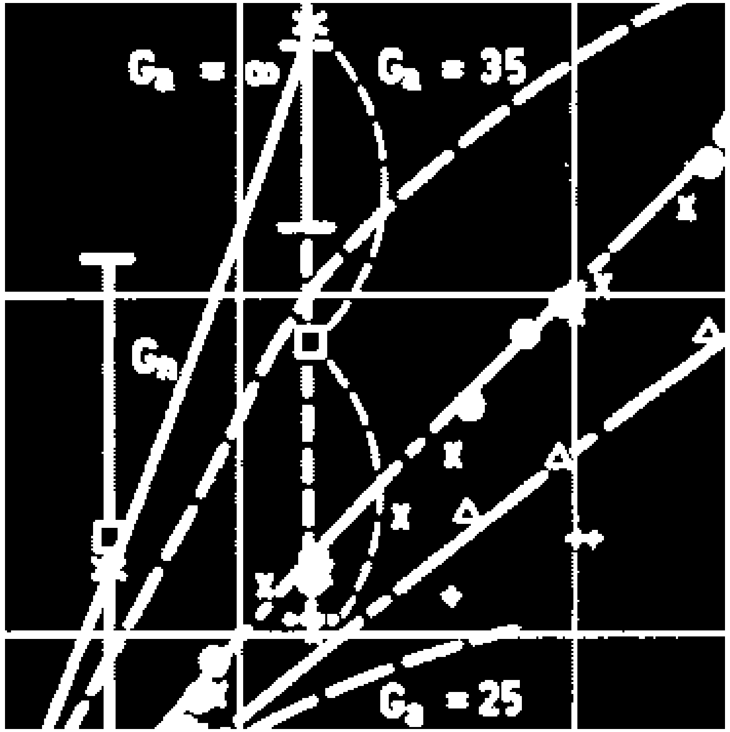


Threshold

Bimodal Histogram

Binary Image Examples


在でも、その影響を受け、
起する問題の研究が多い。
配慮する距離は約2、50
である。
　しかしながら、195
電話通信の自動化および



Why are binary images special?

- Since pixels are either white or black, the locations of white(black) pixels carry ALL information of binary images

Example

$$
\begin{matrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 255 & 255 \\
0 & 0 & 255 & 255
\end{matrix}
\quad \Longleftrightarrow \quad L=\{(3,3),(3,4),(4,3),(4,4)\}
$$

f(m,n)

location of white pixels

matrix representation

set representation

It is often more convenient to consider the set representation than the matrix representation for binary images

- Binary images are least expensive since its requires least storage and also less processing requirements.
- E.g line drawings, printed text in White papers
- This images have enough information about the objects in the images
- The gray images can be converted to binary using thersolding

Image Example (Con't)

Binary fingerprint image

Skeleton image

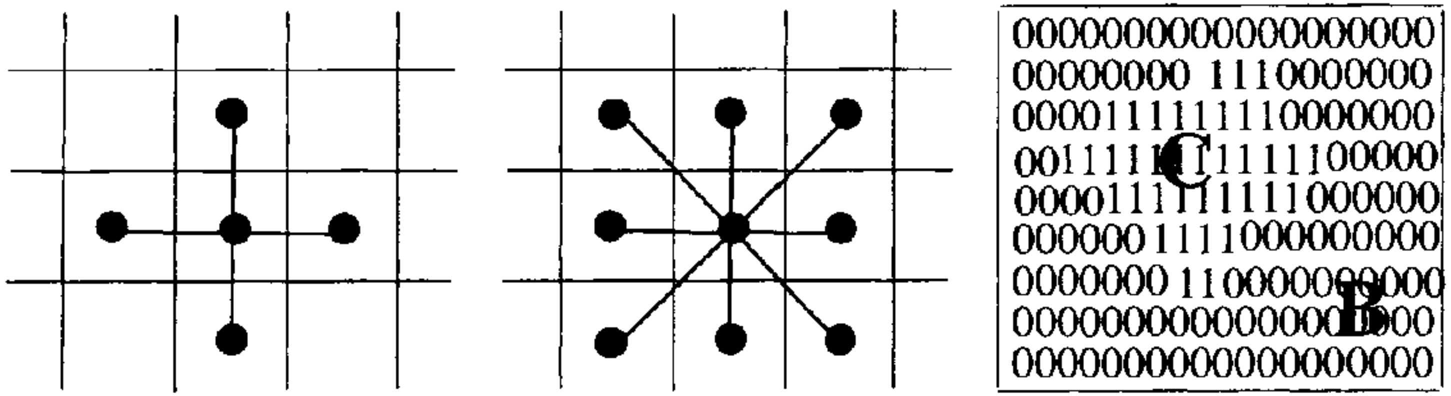
- The binary is quantified in two levels. The grey level can increased based on applications
- As number of intensity increased, we get better approximation images.
- Although the storage requirements also increases approximately

Geometric properties of the Images

- Connectivity
- Projection
- Area
- Perimeter

Connectivity

- A pixel P_0 at (i_0, j_0) is connected to another pixel P_n at (i_n, j_n)
- If and only if there exists a path from P_0 to P_n, which is a sequence of points (i_0, j_0) , (i_1, j_1) ……., (i_n, j_n)
- (i_k, j_k) is a neighboring pixel at (i_{k+1}, j_{k+1})



(a) (b) (c)

Fig. 2.9 Examples of (a) 4-connected pixels, (b) 8-connected pixels, and (c) connected component and background.

- **4-connected:** When a pixel at location (i, j) has four immediate neighbors at $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, and $(i, j-1)$, they are known as *4-connected*. Two four connected pixels share a common boundary as shown in Figure 2.9(a).

- **8-connected:** When the pixel at location (i, j) has, in addition to above four immediate neighbors, a set of four corner neighbors at $(i+1, j+1)$, $(i+1, j-1)$, $(i-1, j+1)$, and $(i-1, j-1)$, they are known as *8-connected*. Thus two pixels are eight neighbors if they share a common corner. This is shown in Figure 2.9(b).

- **Connected component:** A set of connected pixels (4 or 8 connected) forms a *connected component*. Such a connected component represents an object in a scene as shown in Figure 2.9(c).

- **Background:** The set of connected components of 0 pixels forms the *background* as shown in Figure 2.9(c).

AREA

Once an object is identified, some of the attributes of the object may be defined as follows.

- *Area of an object*: The area of a binary object is given by

$$A = \sum_i \sum_j O[i,j],$$

where $O[i,j]$ represents the object pixels (binary 1). The area is thus computed as the total number of object pixels in the object.

- *Location of object*: The location of the object is usually given by the center of mass and is given as

$$x_c = \frac{\sum_i \sum_j [i\,O(i,j)]}{A}, \qquad y_c = \frac{\sum_i \sum_j [j\,O(i,j)]}{A}$$

where x_c and y_c are the coordinates of the centroid of the object and A is the area of the object. In effect thus the location of the object is given by the first-order moment.

Fig.3 Occluded car

- *Orientation of an object*: When the objects have elongated shape, the axis of elongation is the orientation of the object. The axis of elongation is a straight line so that the sum of the squared distances of all the object points from this straight line is minimum. The distance here implies the perpendicular distance from the object point to the line.

- *Perimeter of an object*: To compute the perimeter of an object, we identify the boundary object pixels covering an area. Perimeter is defined by the sum of these boundary object pixels.

Projection of Object

- The projections of a binary image provide good information about the image
- Computed along horizontal, vertical or diagonal lines
- The horizontal projection is obtained by counting the number of object pixels in column wise
- Total no. each rows will give vertical projection

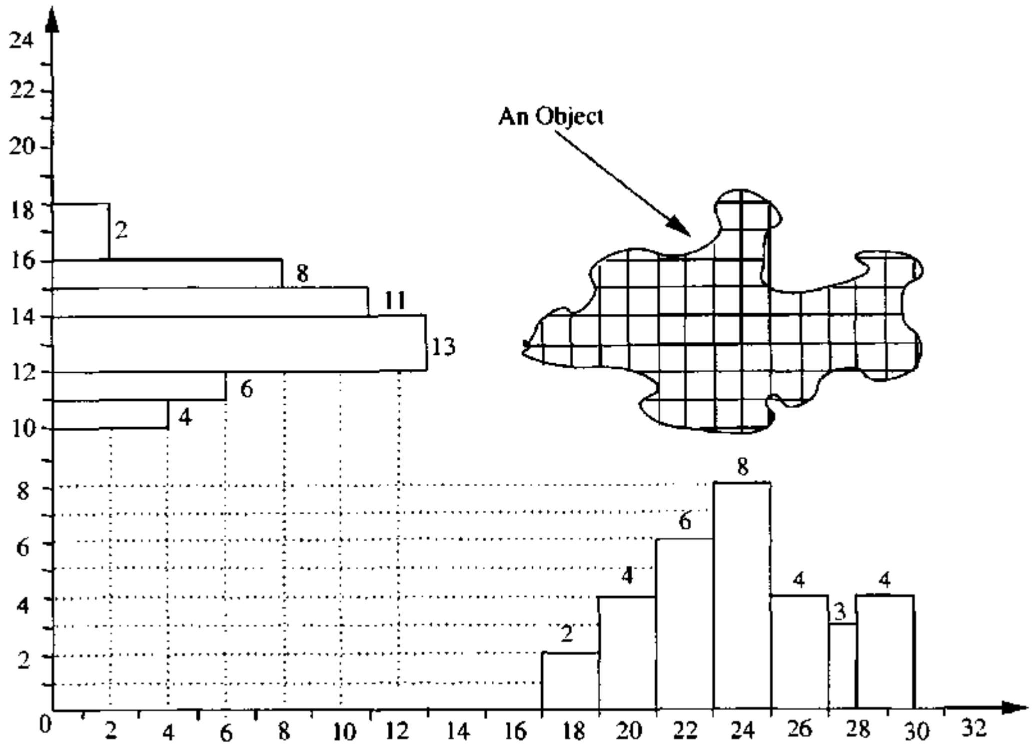


Fig. 2.10 Histogram of a binary object along horizontal and vertical axes.

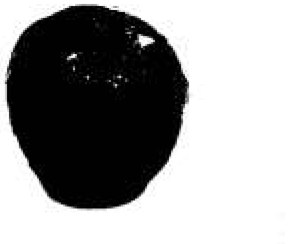
Sample Questions

- How the Geometric properties are helpful for processing binary images?
- What is the procedure for converting gray image to binary Image?

The process of converting grey image to binary is called as

a) Grey scale inversion (negative of the original image);
b) Binary thresholding;
c) Histogram equalization;
d) Some grey scale slicing

1)The process of converting grey image to binary is called as



a) Grey scale inversion (negative of the original image);
b) Binary thresholding;
c) Histogram equalization;
d) Some grey scale slicing

2)Validate the statement “When in an Image an appreciable number of pixels exhibit high dynamic range, the image will have high contrast.”

a) True

b) False

3)The area of object can be defined as

a) $A = \sum_i \sum_j O(i, j)$

b) $A = \sum_i O(i)$

c) $A = \sum_i \sum_j O(i, j, k)$

d) $A = \sum_j O(j)$

3)The area of object can be defined as

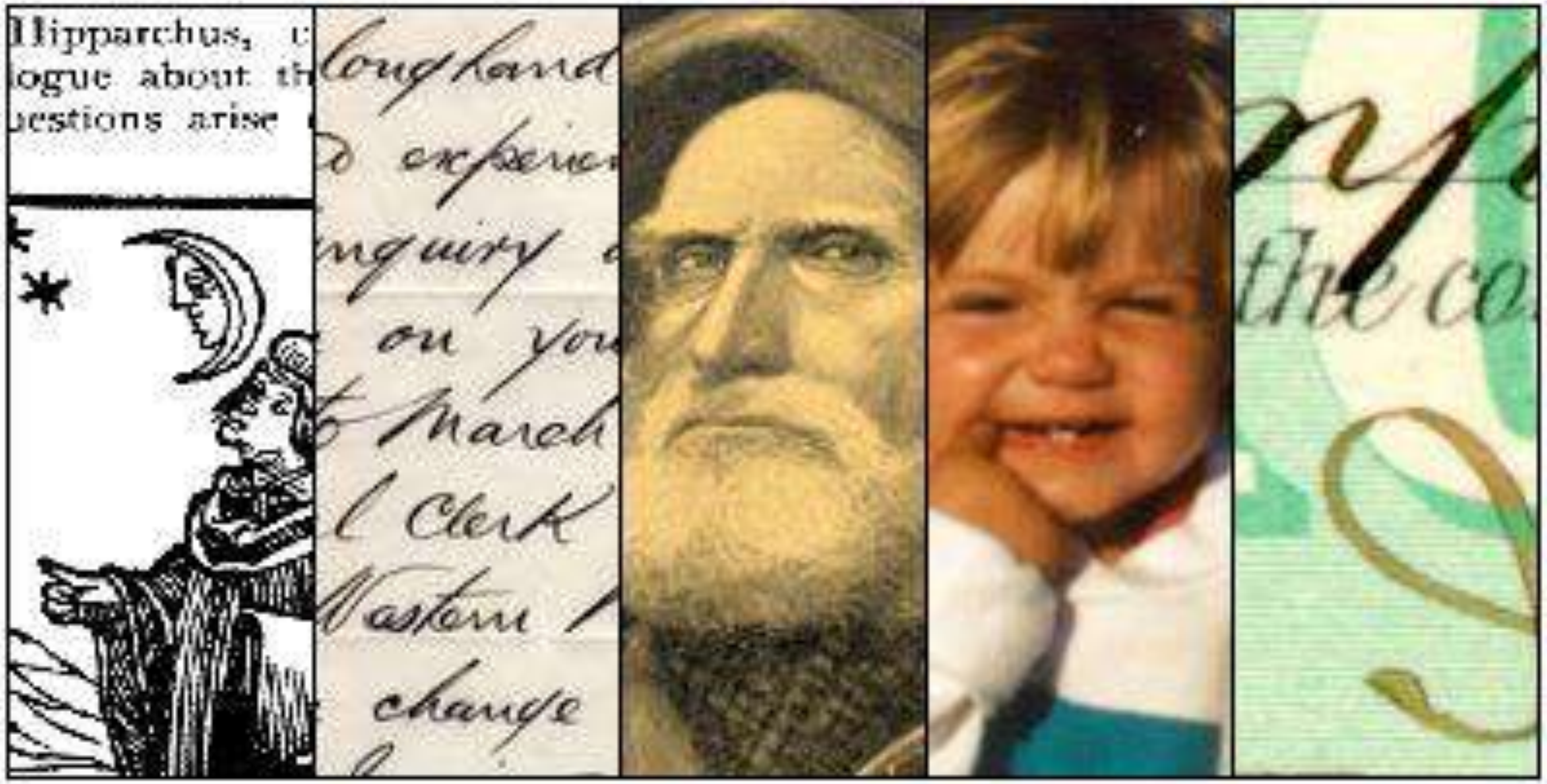
a) $\boldsymbol{A = \sum_i \sum_j O(i,j)}$

b) $A = \sum_i O(i)$

c) $A = \sum_i \sum_j O(i,j,k)$

d) $A = \sum_j O(j)$

T1. Tinku Acharya and Ajoy K. Ray, “*Image Processing Principles and Applications*”, John Wiley and Sons publishers.


Hipparchus,
logue about th
estions arise
March
Clerk
change
the co

RAW (UNPROCESSED CAMERA IMAGE CAPTURING DATA)

- **A camera raw image file contains minimally processed data from the image sensor of a digital camera. Raw files are named so because they are not yet processed and therefore are not ready to be printed or edited. Normally, the image is processed by a raw converter where precise adjustments can be made before conversion to a file format such as a TIFF or JPEG for storage, printing, or further manipulation. There are hundreds, of raw formats in use by different models of digital equipment.**

PROS

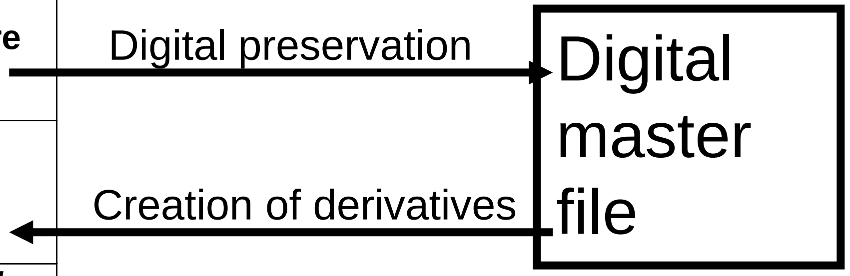
- Format can be used by many applications.
- File size may be smaller than TIFF.
- Saved at the camera's maximum colour bit depth.

CONS

- Large file size means that it can take longer to save an image to the memory card, i.e. less images captured per second.
- Difficult to work with images because they need to be converted to something else (like TIFF) before they can be shared and manipulated.

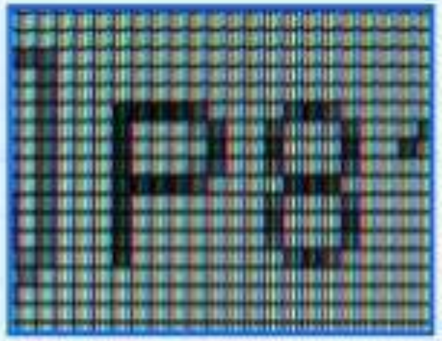
Digital master images

	Conventional photography	Digital photography
step 1. Capture	‘Snapshot’	Scanning / digital capture
step 2. Process	Chemical film processing	Image processing
step 3. Output	Imprint	Screen view / Printer output

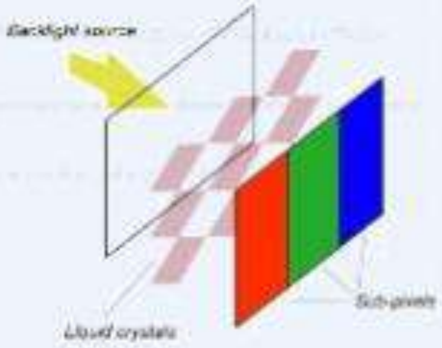


Three steps in the photographic process

Image Formation



- The primary concept of image formation signify the presence of dots.
- In digital mediums these dots are often called PIXELS (Picture Elements) for 2-D medium or VOXELS (Volumetric Picture Elements) for 3-D images.



- Pixels can be of single colour in an entire image drawing or may be of primary colours in a multi-colour drawing.
- The proximity of pixels (DPI - Dots Per Inch) and the count over a given dimension represent the Image Resolution which is the primary measure of Visual Quality of an Image.

Graphic File Formats

… - PCX – PBM – TGA – TIFF – GIF – JPEG – PSD – DXF - CGM – PNG – SVG – RAW – WPG – FITS – BMP – PCD – RAS – TGA – BPS – EPS – PDF – PCT – WBM – FITS – XBM – VFF – RIB – PCX – DMP – AVS – IMG – ICO – JFIF – IFF – WMF - …

Why are there so many different graphic file formats?

- There are a number of fundamental different types of graphical data
  - raster data (sampled values)
  - geometry data (mathematical description of space)
  - latent image data (data transformed into useful images by some algorithmic process)

Types of Digital Images (Based on Drawing Units)

- **RASTER IMAGES:**
  Images made of pixels or dots.
  A Raster means a frame of Screen which is made using scanning of entire screen line by line. Raster images are resolution dependent and can not be scaled up or down without loss of quality.

- **VECTOR IMAGES:**
  Images made using lines and curves that are drawn using mathematical equations. These images are not related to resolution and they can be scaled up or down to any size. While drawing on a medium vector images are also drawn using pixels, but they retain their scaling quality and redrawn again when resized.

Both Raster and Vector Images can have Colours. In case of Raster Images, colours is a pixel property while in case of Vector Images Colours are given as another mathematical property.

Types of Digital Images (Based On Dimensions)

- **2-D IMAGES:**
  Images that are seen on X and Y axis only.

- **3-D IMAGES on 2-D Medium:**
  Images that are displayed on 2-D medium but still have feeling of depth associated with it. These images might require a viewer to wear special glasses to feel the 3D effect.

- **3-D IMAGES on 3-D Medium:**
  Images that are produced in all 3-Dimensions of space such as 3-D holographic images, 3-D Laser Images.

Features List

- Resolution and DPI
- Colour/Bit Depth
- Compression
- Transparency
- Alpha Channels
- Gamma Correction
- Anti-Aliasing
- Interlacing
- Palletes
- Half Tone/ Duotone
- Animation Support

Colour Depth/Bit Depth

- 2 Colour: This is mostly a black and white image. It may however be any other two colour image also. It requires just 1 bit for colour information.
- 16 Colour: These are very basic image types used on old machines. The colour information here is encoded in 4 bits only.
- 256 Color: These are 8 Bit images often used in BMP and GIF formats.
- 16 Bit: This is also known as HI-COLOR setting. These have more than 64 Thousand colours and are suitable for most picture related work on computers.
- 24 Bit: This is known as TRUE COLOR Setting. These have a better colour depth of more than 16.7 Million Colours and are visually excellent.
- 32 Bit: These have even better picture quality to show excellent picture shading effects and shades of special colours like silver, gold, colours across a glass etc. This quality is also available on modern computers.

Image Compression.

- With the advent of faster computers the technology of reducing the image file size gained momentum.
- It is possible to reduce file size of a pixel based image to the tune of 1/10 using modern compression techniques without much loss in image quality.
- Image compression can be either completely lossless like in case of GIF and PNG standards, while in case of lossy compression even more size reduction can be achieved as in the case of JPG formats (commonly used for photographs).

Watch the minor loss of image quality near the Cat's nose.

Image Transparency.

- Transparency is a feature of image where a certain colour of image can be replaced by the background.
- Image Transparency feature is available with certain file formats only. Most common one with transparency features are GIF, PNG and TIFF formats.
- In some formats multiple levels of transparency is supported by adding additional byte for transparency level. So instead of RGB colours it becomes RGBA colours. Here A stands for Alpha Channel. Typical example of this is the PNG file format.

White Background Colour has been made transparent in this case.

More About Alpha Channel

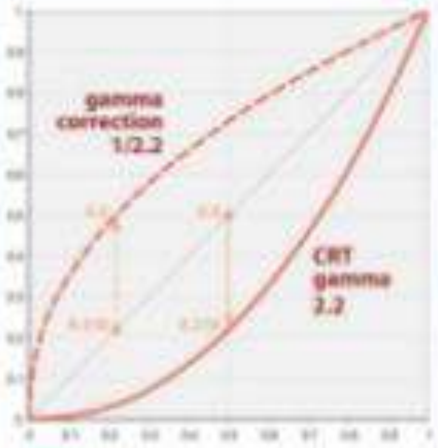
- It is a way to produce partial transparency.
- This often provides perfect smoothening of edges which is otherwise difficult due to rectangular nature of pixels.
- The drop shadow effect is also correctly possible due to partial transparency feature.

Gamma Correction

- If the intensity element of image is encoded in a linear fashion then the image will not look correct. This is due to the non-linear behaviour of displays devices like CRT, LCD etc. The intensity value with respect to impulse applied does not change linearly.

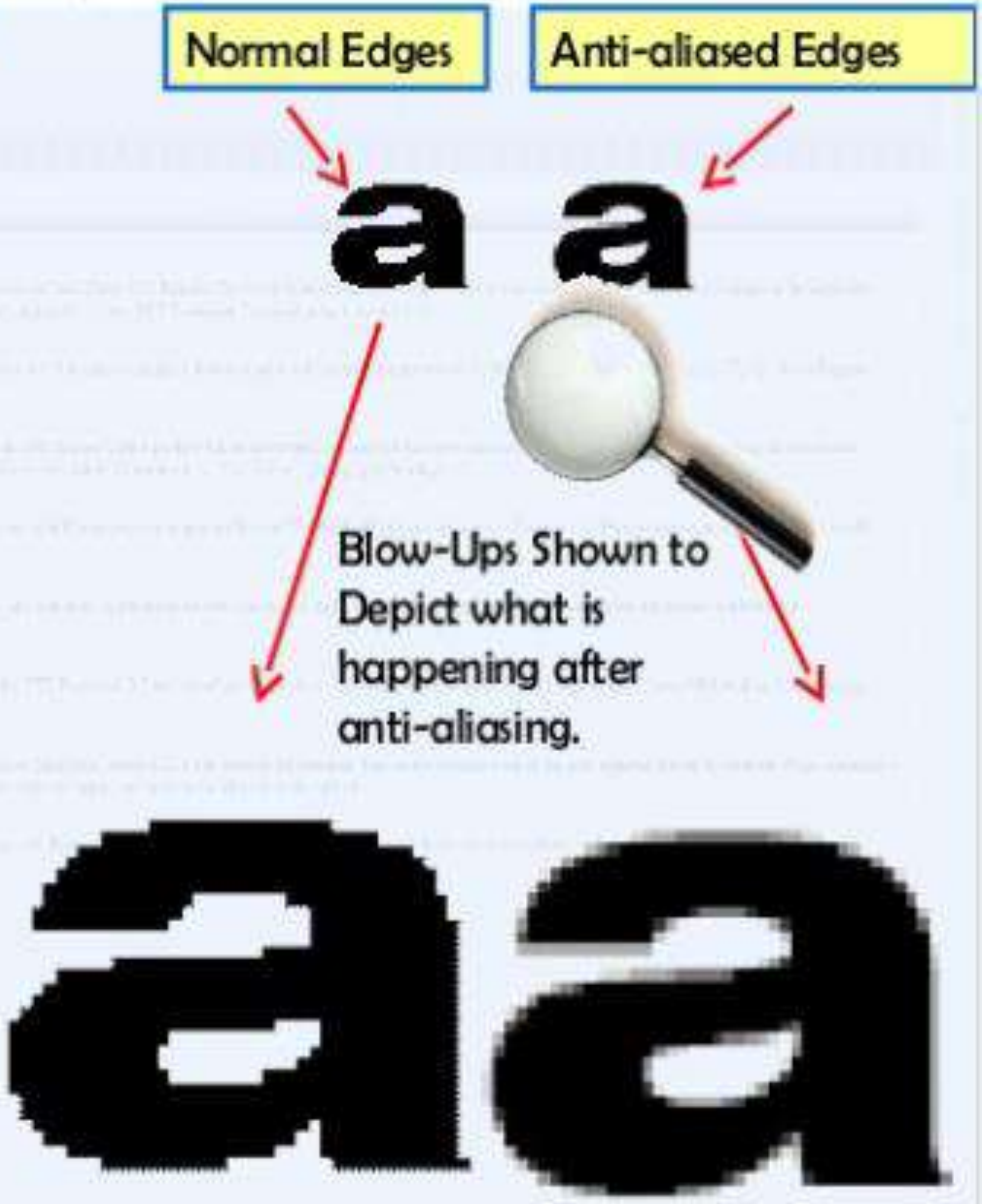
- Due to this nonlinear behaviour intensity value is pre-encoded in image files itself to show it correctly on various displays.

- As an image might be displayed on different mediums and makes of computers one may need to adjust the value of this encoding for different devices. This is called GAMMA CORRECTION. Some image formats like PNG allow this correction in a straight forward manner.



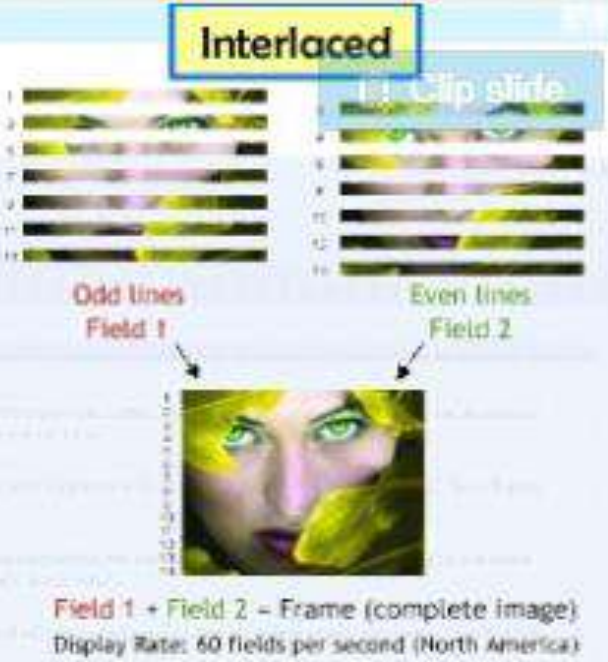
Anti-Aliasing

- Anti-Aliasing is a way to give a smoother perception of edges of an image.
- Anti-aliasing is done by using mathematical calculations to calculate grey shades of existing pixels or by addition/deletion of pixels near the edges.
- Microsoft's Clear-type Technology is an advanced form of anti-aliasing using colored pixels on edges.
- Anti-aliasing algorithms may be applied to full screens for better edge smoothening on the entire picture scenes. This is often done in graphics cards for better look of 3-D effect games.



Interlacing

- Interlacing indicates the way an image is drawn on the screen.
- Interlaced Image: When the image is drawn in a way like it is done on a TV where ODD frame is drawn first and then EVEN frame is drawn, it is called an Interlaced image. In this case the whole image frame is drawn at once and then gradually the other interpolated frames are drawn to give a feel of improving quality.
- Non-Interlaced Image: In this type the image is drawn from top to bottom pixel by pixel and line by line. The rest portion of image looks blank. Non-Interlacing is also termed as "Progressive Scan".
- Image Formats Like GIF, JPG and TIFF support interlacing.



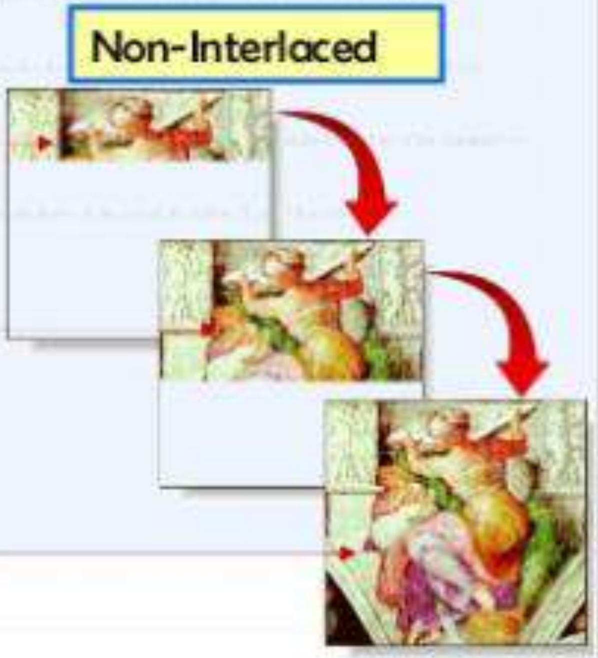
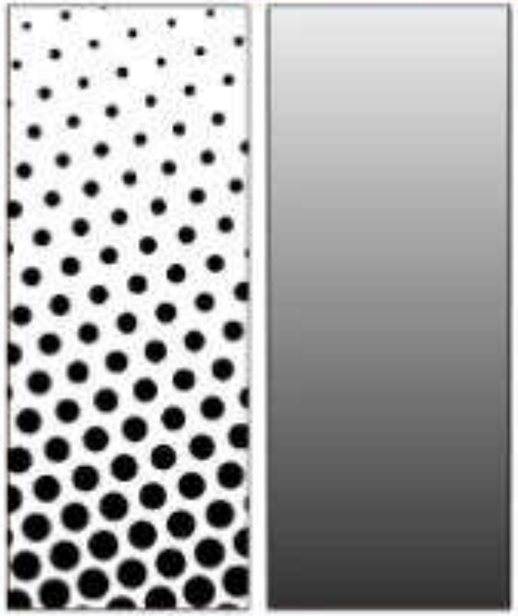


Image Palettes

- The set of colour an image is made of is called image palette.
- A typical digital image may contain an 8 bit palette (256 colours) or a set of more colours like 64K colours or 16.7 Million Colours.
- A palette based image has an advantage of changing the entire colour set in one go by just switching the palettes.
- 8-Bit BMP and GIF images are examples of palette based images.

Half Tone / Duo Tone

- Half tone is a method where continuous tone perception is generated with the use of dots pitched far from regular dots density.
- Duotone is produced by superimposing another contrasting colour halftone over halftone drawing in another colour.

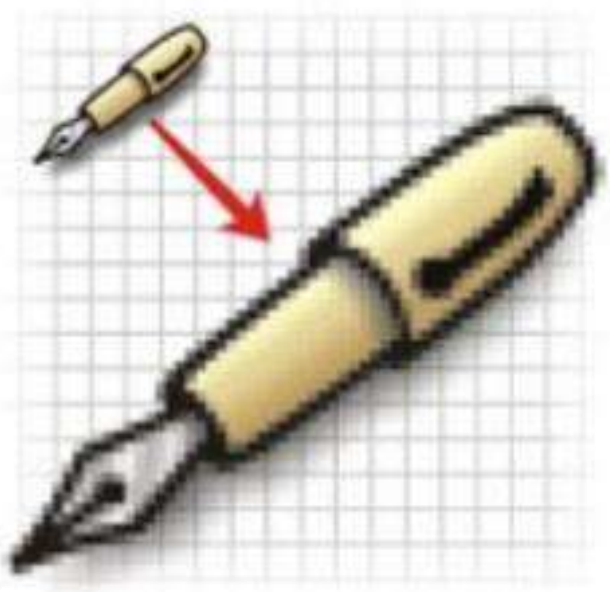


Popular 2-D Digital Image Formats

Categorisation of 2-D Image Formats

- RASTER IMAGE FORMATS
  - Non-Proprietary:
    JPG, GIF, PNG, TIFF, BMP
  - Proprietary:
    PSD, PSP
- VECTOR IMAGE FORMATS
  - Non-Proprietary:
    CGM, SVG, WMF/EMF, EPS
  - Proprietary:
    CDR, AI, PDF, SWF

RASTER FORMATS



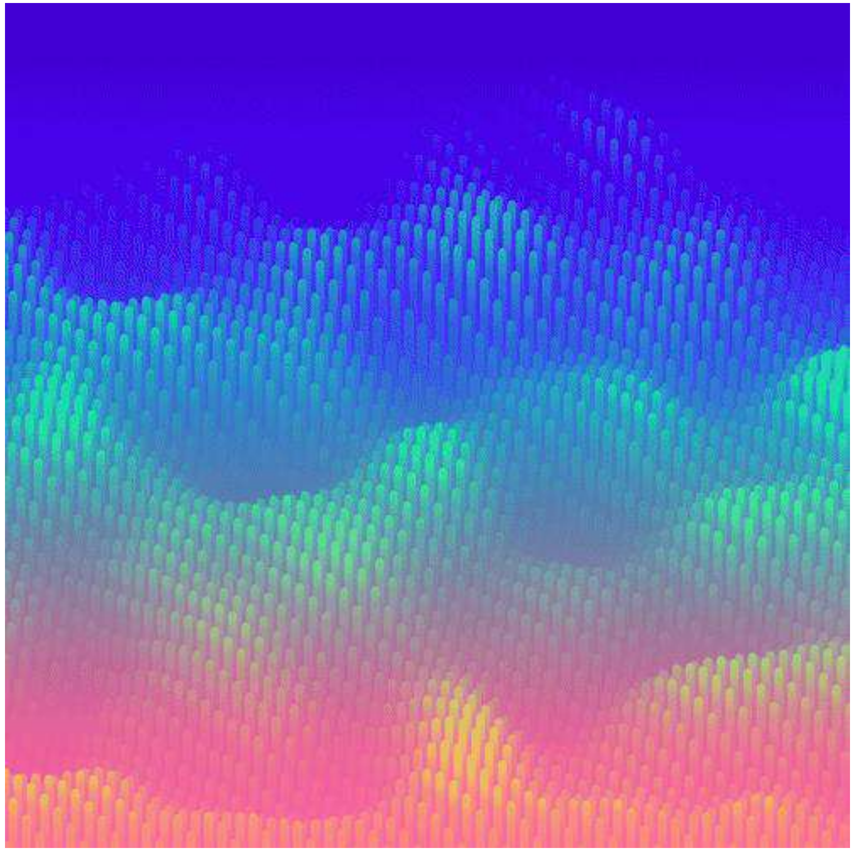
BMP – Bitmap Format

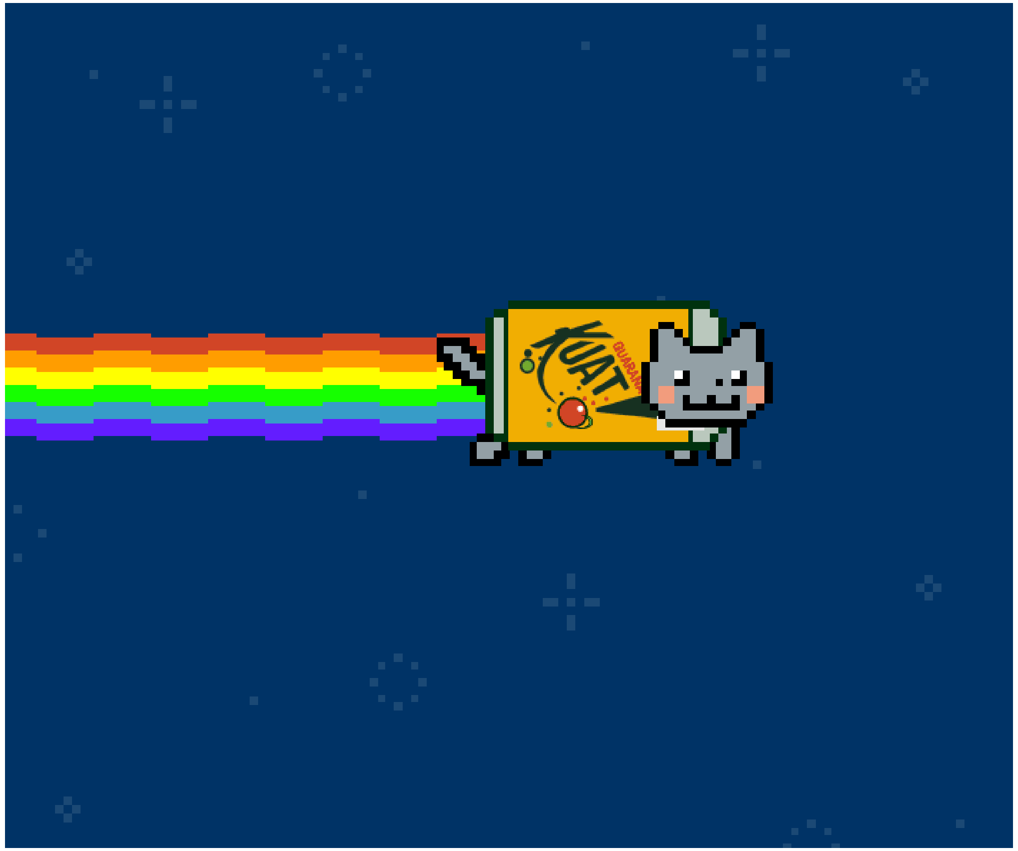
- Used to store direct pixel colour information.
- It is mainly used on Windows family of OS.
- One of its variations is DIB (device independent bitmaps).
- It has file extensions BMP and DIB.
- Its 8 bit version can store only 256 colours.
- It creates a very large file-size as it has no major compression technique in use.
- Sometimes low compression ratio techniques like RLE (Run Length Encoding) can be used along with BMPs.
- BMP files may be zipped to reduce file sizes but still BMP format is not suitable for internet transfer.
- BMP format is however useful for standalone applications and creation of a base file that stores entire image information without any quality loss.

GIF – Graphics Interchange Format

- This is also a pixel based format with a good image compression technique called LZW compression.
- The image quality is not reduced while using this compression.
- It supports upto 256 colours.
- It is good for images with lines, curves and solid colours.
- It is most widely supported on internet and other software.
- It can support animated images with different set of colours for each frame.
- It also supports transparency.


uploading




KUAT

GIF (GRAPHICS INTERCHANGE FORMAT)

- **A GIF is a lossless file format for image files that supports both animated and static images.**

PROS

- 256 number of colours.
- Uses lossless compression.
- Support for transparency.
- Small file format.

CONS

- The oldest format for web – 1989.
- In most cases it has a bigger file size than PNG.
- Loss of colour variation.

JPG/JPEG – Joint Photographic Expert Group

- It has been designed under the banner of ISO (International standard Organisation) by Joint Photographic Experts Group.
- It is a highly compressed raster image format best suited for photographs.
- It allows for adjustment of degree of compression.
- It is a lossy compression. Once compressed perfect reproduction of original is impossible.
- For very little image loss 1:10 compression is often possible.
- It has an advanced version called JPEG 2000 which is still less used as it is a licensed version.
- Progressive JPGs allow multipass drawing when loaded.

Original
1.5 MB

High
Lossy
Compression
92 KB

Compression

- Lossless

RLE (Run Length Encoding) – Windows bitmap files (bmp, ico)

LZW (Lempel-Ziv-Welch) – GIF & TIFF files

ZIP – TIFF files

- Lossy

JPEG (Joint Photographic Experts Group)

Best suited to photos and paintings of realistic scenes with smooth variations of tone and colour

JPEG (JOINT PHOTOGRAPHIC EXPERTS GROUP)

- **A JPEG is an image file format used for compressing image files. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality.**

PROS

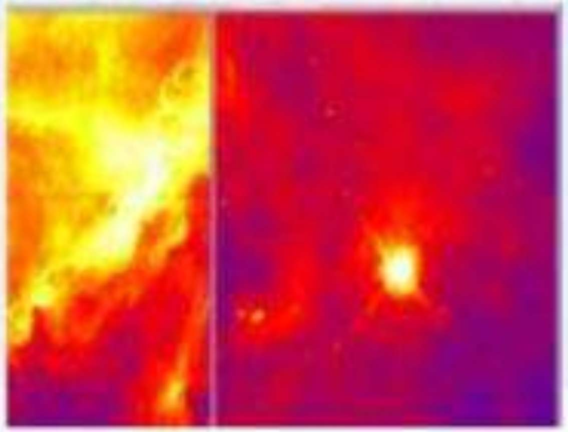
- Retains up to 16,000,000 colours.
- Suitable for images, high details & quality pictures.
- It is the most used graphic file format.
- Approved as standard in 1994.

CONS

- It does not support transparency.
- File size larger than GIF because of colour information.

TIFF – Tagged Image File Format

- This format was designed for images for Desktop Publishing Purposes.
- This format is often the default format in which a scanner saves the file.
- It can work as an image container along with its editing/cropping status.
- It can support compression without any loss of quality.
- It is highly recommended for archival images so that every thing can remain intact along with the image.
- It can even act as container of lossy JPEG files.
- Adobe Corp. holds the copyright of this format but for usage purpose license is not required.
- It is also useful for high precision scientific images. The other option for high precision scientific images such as astronomical images is FITS (Flexible Image Transport System)



TIFF (TAGGED IMAGE FILE FORMAT)

- **Tagged Image File Format (TIFF) is a computer file format for storing raster images, popular among the publishing industry and photographers. The TIFF format is widely supported by image manipulation applications and publishing and page layout applications.**

PROS

- No image data is lost.
- Better image quality than even the JPEG fine quality.
- Good for images that will be heavily manipulated in a photo editing program.

CONS

- File size is very large.
- Still need to make sure that exposure, white balance and colour saturation are properly set because fixing these in the photo editing program will degrade the image to a certain degree.

Extension	Colour	Compression	Common Uses
JPG, JPEG	24-bit	Lossy	Photos, web pics
GIF	8-bit	Lossless	Web graphics – buttons, icons, etc
PNG	up to 24-bit	Lossless	Web – replacement for GIF
TIF, TIFF	24-bit	Lossless	Professional Photos etc

PSD – Photoshop Document

- It is a application specific proprietary format for Adobe Photoshop.
- Application specific formats support layered image storage, editing marks, clipping, padding and position storage.
- It supports colour modes like RGB, CMYK, LAB, GREY, PALLETTED etc.
- It supports unlimited colours.
- It produces native bitmap format with RLE compression.

VECTOR FORMATS

WMF/EMF – Windows Metafile/Enhanced Meta File

- It can store both vector and bitmap components.
- It is the native format for applications like Word, Power-point, Publisher etc.
- The EMF enhancement has commands for printer graphics also.
- WMF/EMF have direct function calls for windows GDI – Graphics Device Interface.
- WMF produces very small size files and is quite suitable for usage in form of clipart libraries.
- WMZ and EMZ are compressed formats of WMF and EMF respectively.

CDR - CorelDraw

- It is native Vector Based Format for Corel Applications.
- It has capability to store both vector based and bitmap images.
- It can store multiple pages of data.
- It stores all editing status and other image configuration data.
- It produces an extremely low file size.
- Every Version of CorelDraw has different formats and it is essentially required to convert files to a lower version while transferring to previous version of Corel Application.

EPS – Encapsulated Postscript

- This format is frequently used for production quality drawing in printing.
- This is a vector based format that can contain PostScript Style Description of Drawing.
- Its purpose is for printing only and EPS files are not meant for Editing.
- Converted EPS files may only be viewed by special viewers. Regular graphics applications are not able to open these file.

PPM-Portable Pixmap Image

- A PPM file is a 24-bit color image formatted using a text format.
- It stores each pixel with a number from 0 to 65536, which specifies the color of the pixel.
- PPM files also store the image height and width, whitespace data, and the maximum color value.
- Either P3(ASCII form) or P6(Binary form)

```
P3
# feep.ppm
8 8
255
255 255 0   255 255 0   255 255 0   255 255 0  255 255 0   255 255 0
255 255 0   255 255 0
255 255 0   255 255 0   255 255 0   255 255 0  255 255 0   255 255 0
255 255 0   255 255 0
0  0  0     0  0  0     0  0  0     255 0 255  0  0  0     0  0  0
0  0  0     255  0 255
0  0  0     0 255 127   0  0  0     0  0  0    0  0  0     0 255 127
0  0  0     0  0  0
0  0  0     0  0  0     0 255 127   0  0  0    0  0  0     0  0  0
0 255 127   0  0  0
255  0 255  0  0  0     0  0  0     0  0  0    255  0 255  0  0  0
0  0  0     0  0  0
255 255 0   255 255 0   255 255 0   255 255 0  255 255 0   255 255 0
255 255 0   255 255 0
255 255 0   255 255 0   255 255 0   255 255 0  255 255 0   255 255 0
255 255 0   255 255 0
```

PGM

- A **PGM** file is a grayscale **image** file saved in the portable gray map (**PGM**) format and encoded with one or two bytes (8 or 16 bits) per pixel.
- It contains header information and a grid of numbers that represent different shades of gray from black (0) to white (up to 65,536)

PBM

- A file with the PBM file extension is mostly likely a Portable Bitmap Image file.
- These files are text-based, black and white image files that contain either a 1 for a black pixel or a 0 for a white pixel.

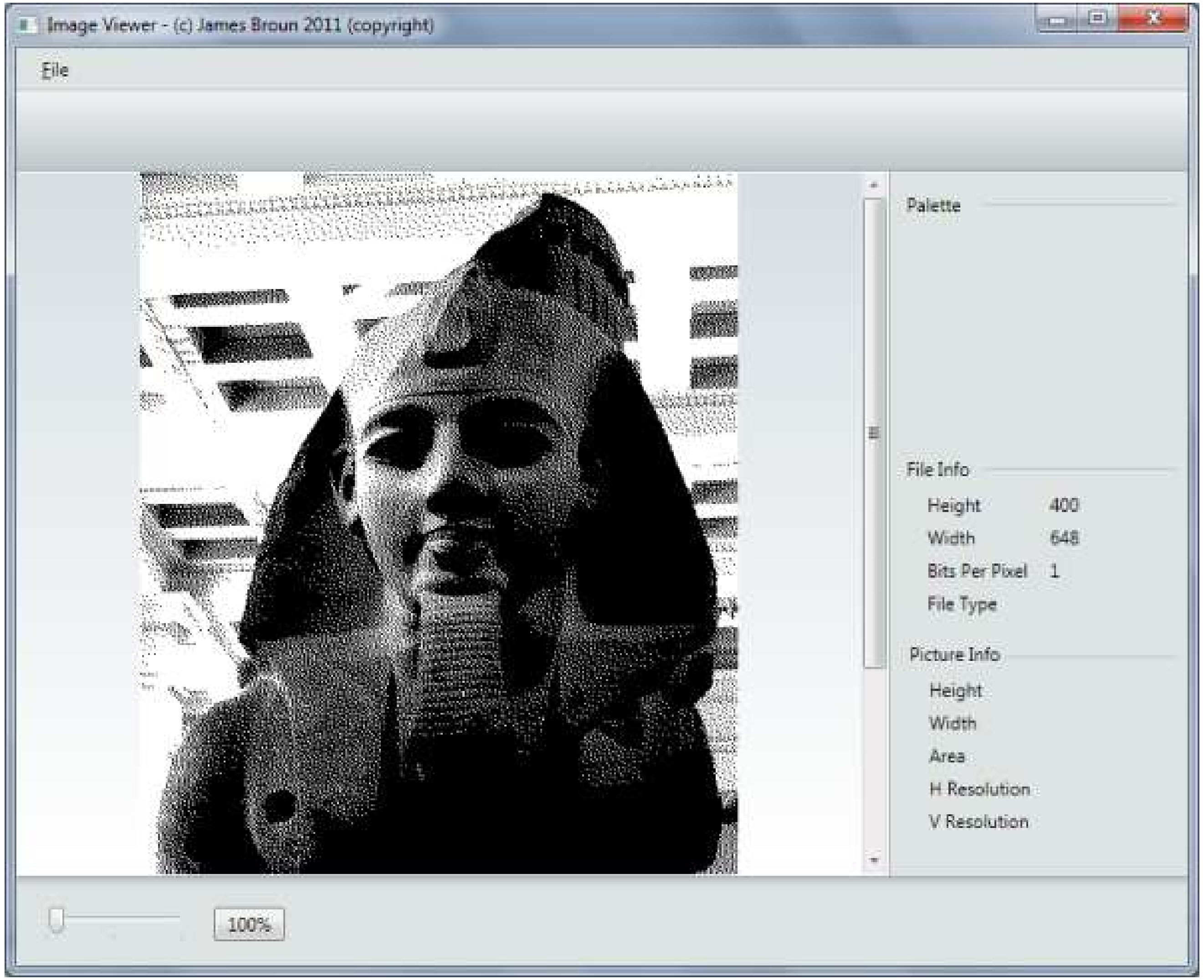


Image Viewer - (c) James Broun 2011 (copyright)
File
Palette
File Info
Height 400
Width 648
Bits Per Pixel 1
File Type
Picture Info
Height
Width
Area
H Resolution
V Resolution
100%

OTHER FORMATS

ICO - ICON

- It is a file format for Windows Icons.
- One can produce 16x16 16 colour icon to 48x48 16.7 Million Colour Icons using this format.
- Technically ICO file can even store a 1x1 pixel image.
- Vista and above supports even the 256x256 pixel icons.
- Modern systems also supports PNG files to be used as icons.
- One may require to use a special pixels based editor to create icons.

SVG – Scalable Vector Graphics

- It is a modern XML based file for vector graphics, that can be used on internet also.
- It is an standard from W3C (World Wide Web Consortium) that created standards for HTML, XML etc.
- It supports standard compression techniques and is saved as SVGZ when compressed.
- It supports dynamic, interactive drawing using Javascript.
- It supports Vector Graphics, Raster Graphics and Normal Text.

Introduction to Mathematical Operations in DIP

- **Array vs. Matrix Operation**

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

Array product operator

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

Array product

Matrix product operator

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

Matrix product

Introduction to Mathematical Operations in DIP

- **Linear vs. Nonlinear Operation**

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

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

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

Additivity

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

Homogeneity

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

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

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

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, $\{g_i(x,y)\}$

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

Example: Addition of Noisy Images for Noise Reduction

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

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

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

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

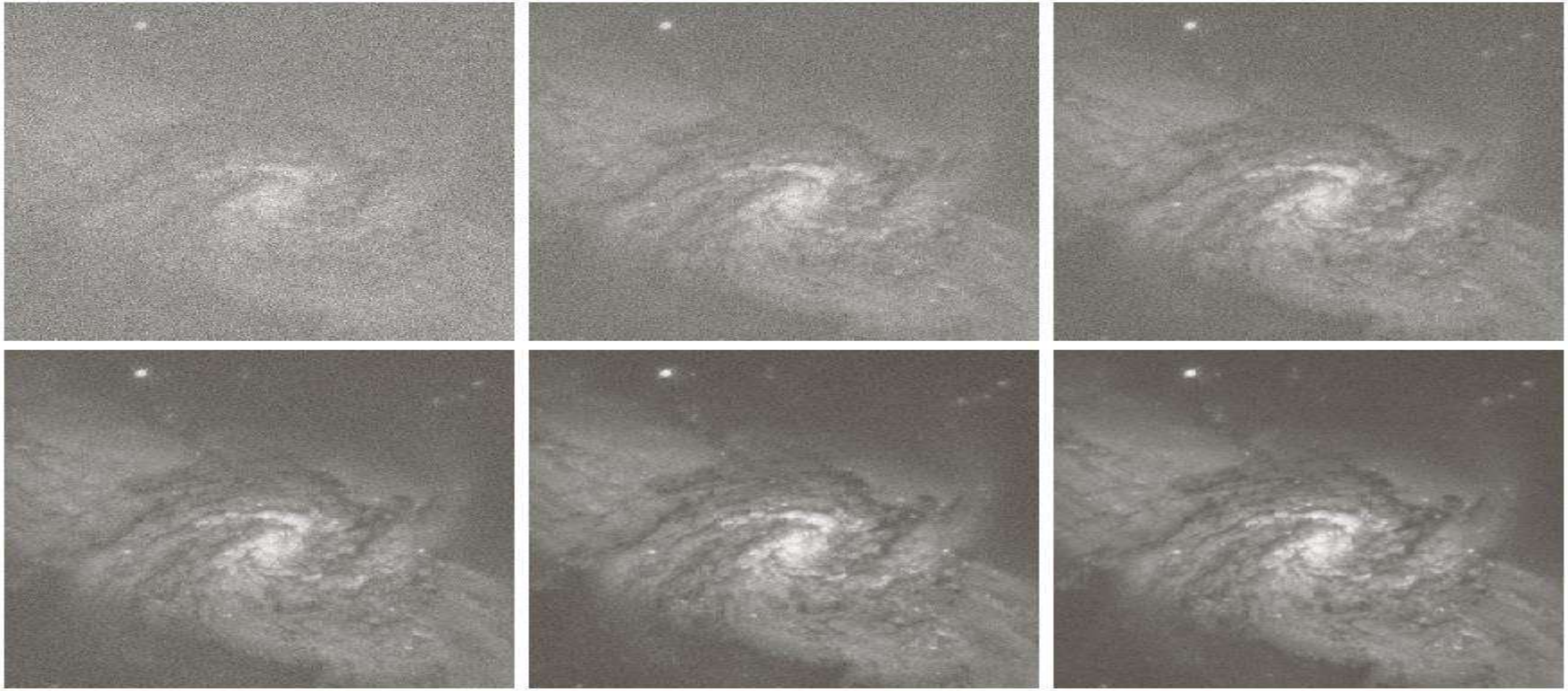
$$= f(x,y)$$

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

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

Example: Addition of Noisy Images for Noise Reduction

- In astronomy, imaging under very low light levels frequently causes sensor noise to render single images 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

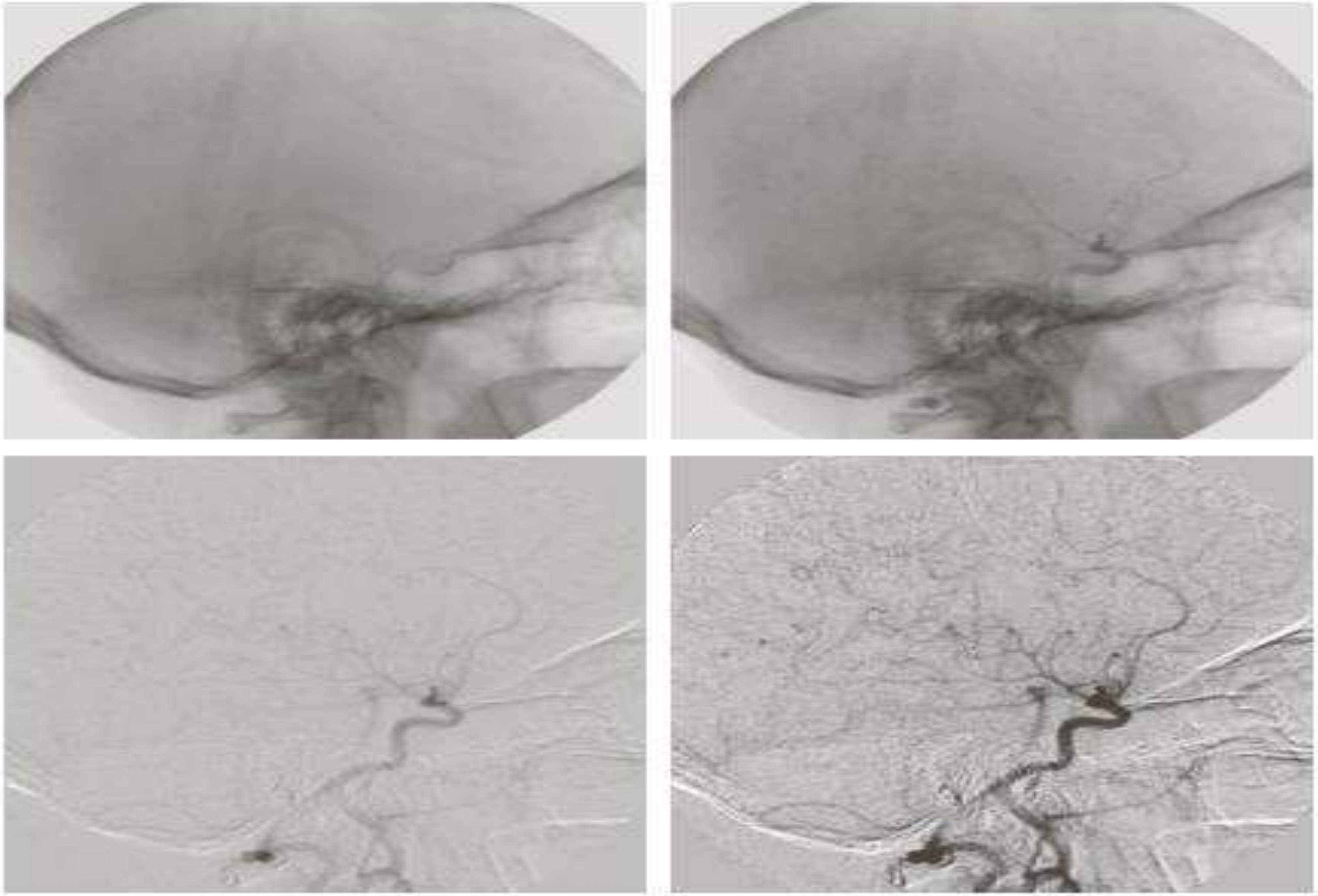
Mask $h(x,y)$: an X-ray image of a region of a patient's body

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

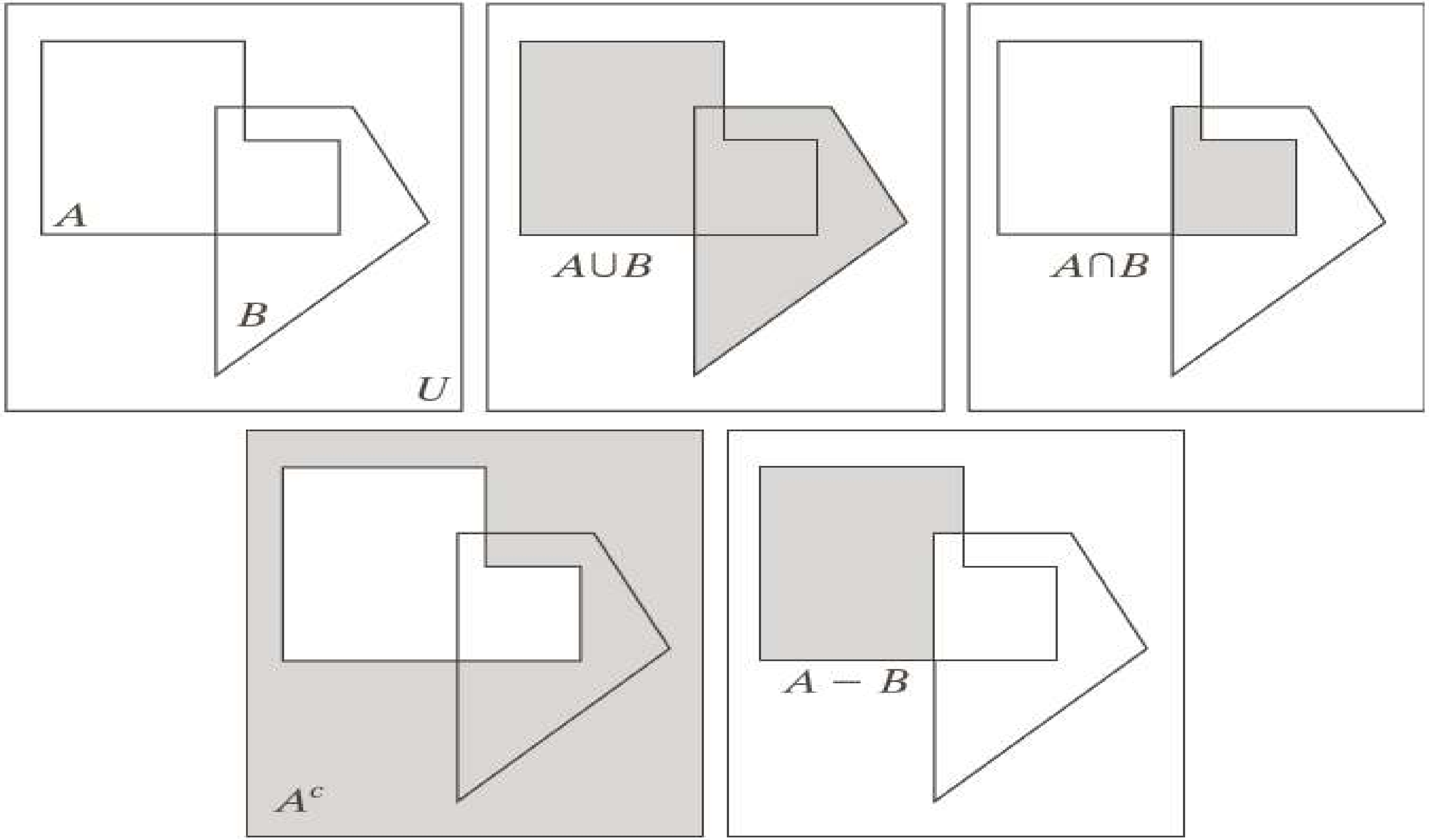
FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

An Example of Image Multiplication

a b c

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

Set and Logical Operations



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

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

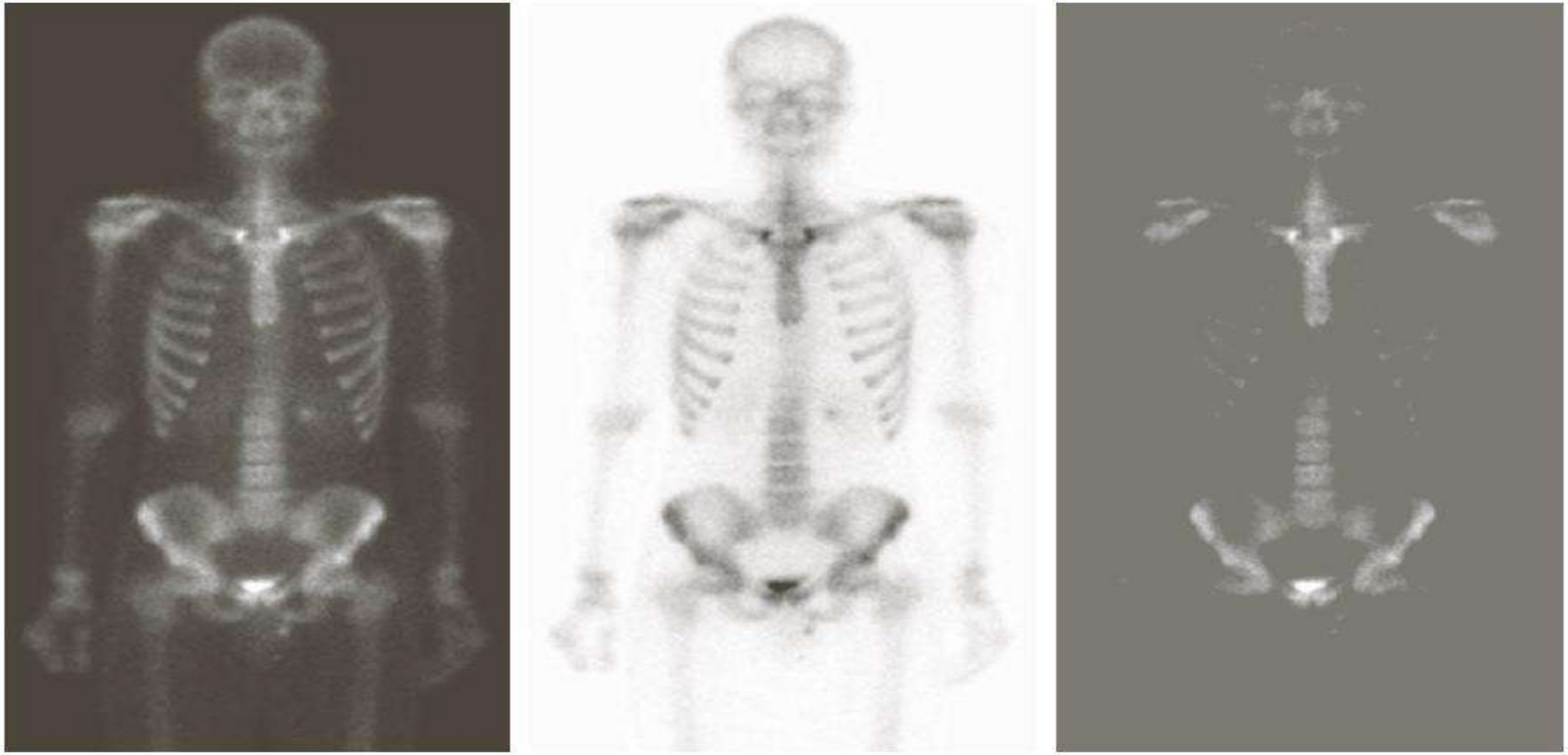
$$K = 2^k - 1; k \text{ is the number of intensity bits used to represent } z$$

Set and Logical Operations

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

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

Set and Logical Operations



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.)

Set and Logical Operations

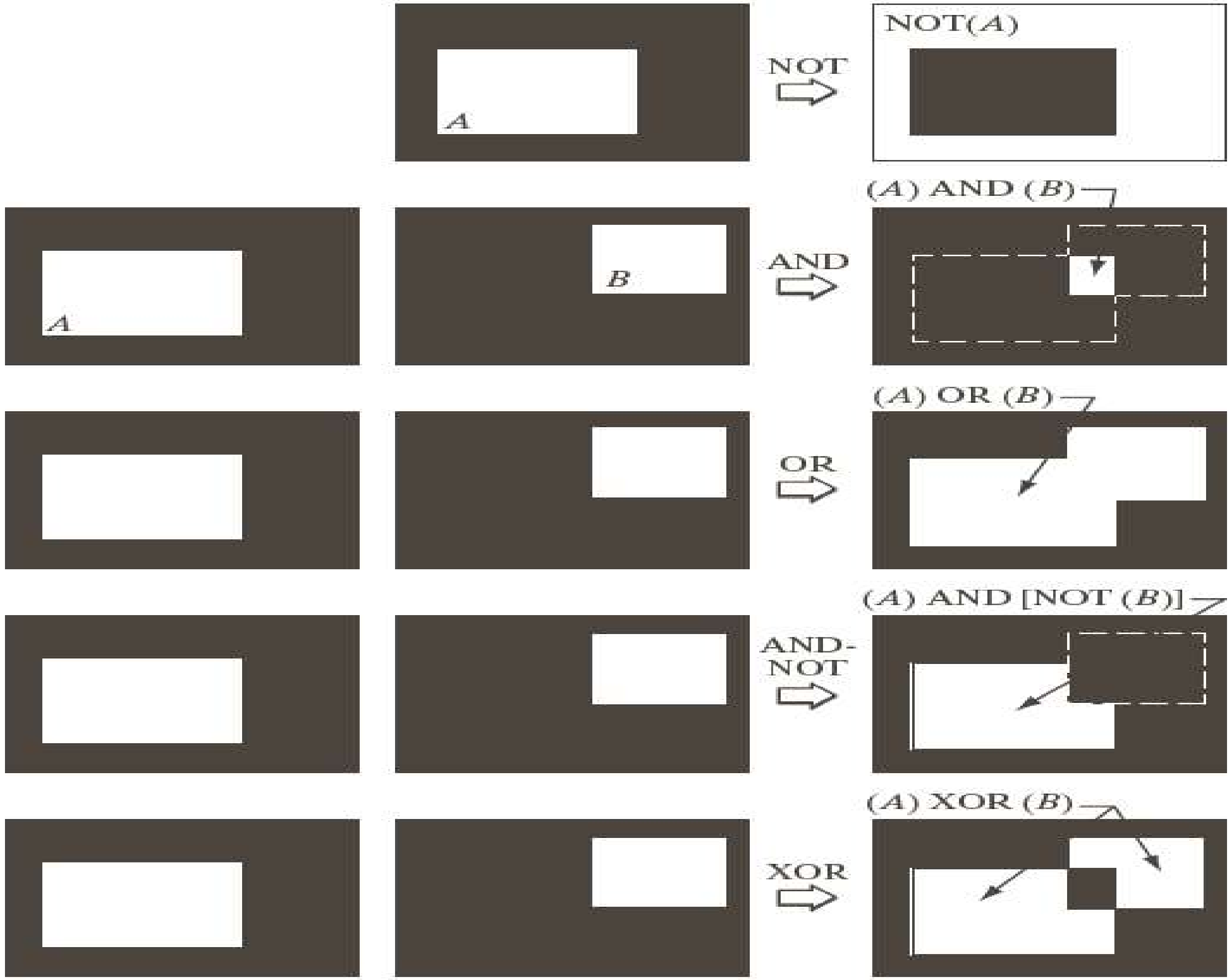


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Spatial Operations

- Single-pixel operations

  Alter the values of an image's pixels based on the intensity.

$$s = T(z)$$

e.g.,

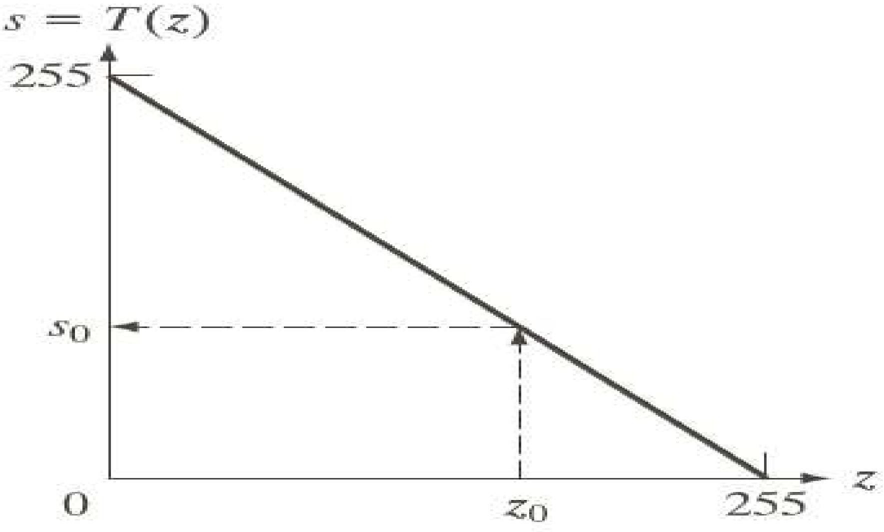
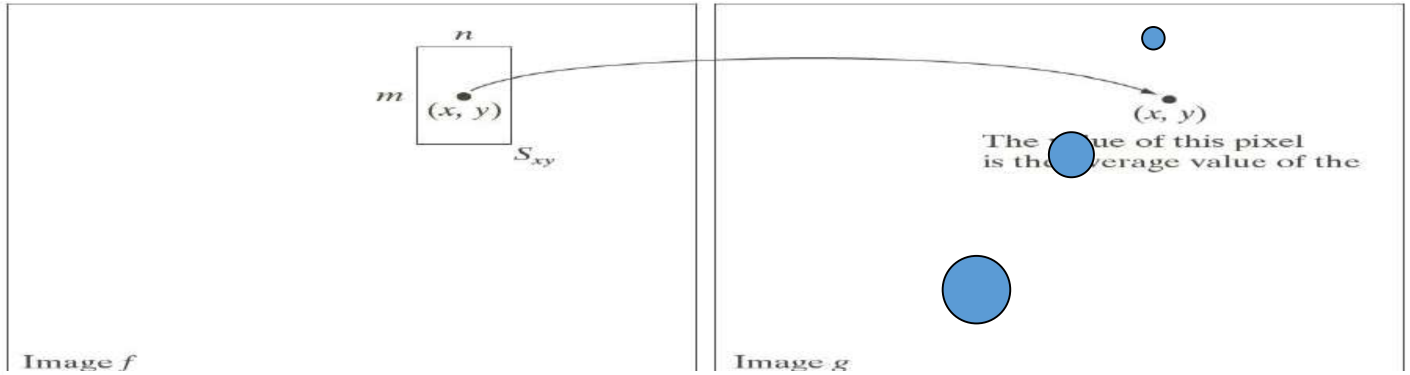


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

Spatial Operations

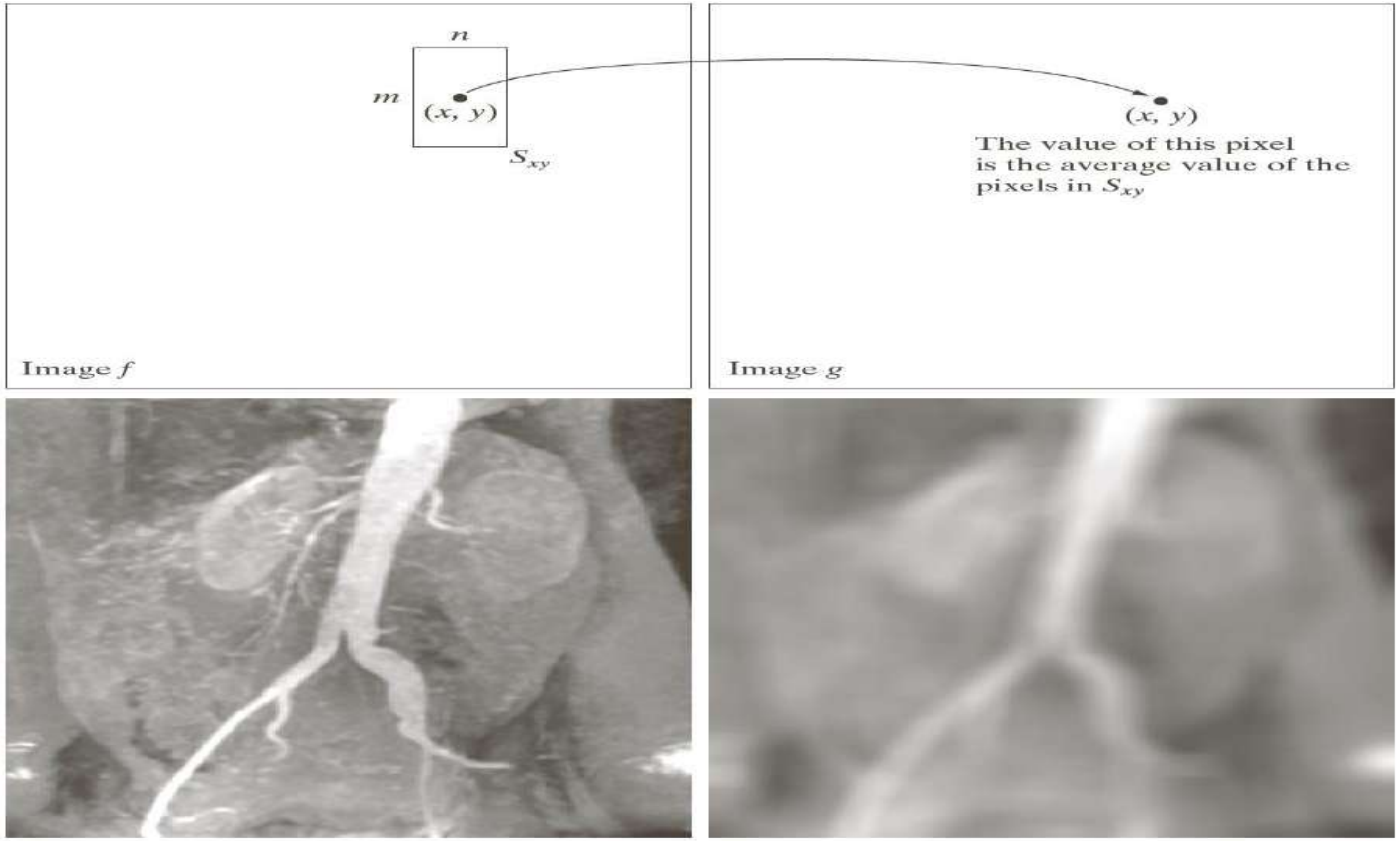
- Neighborhood operations



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

Spatial Operations

- Neighborhood operations



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

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

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

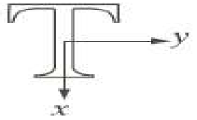
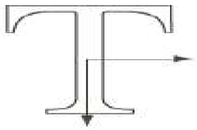
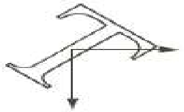
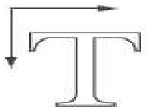
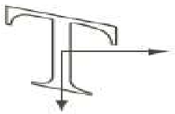
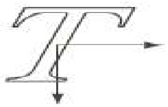
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta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

Image Registration

- Input and output images are available but the transformation function is unknown.

  Goal: estimate the transformation function and use it to register the two images.

- One of the principal approaches for image registration is to use ***tie points*** (also called ***control points***)

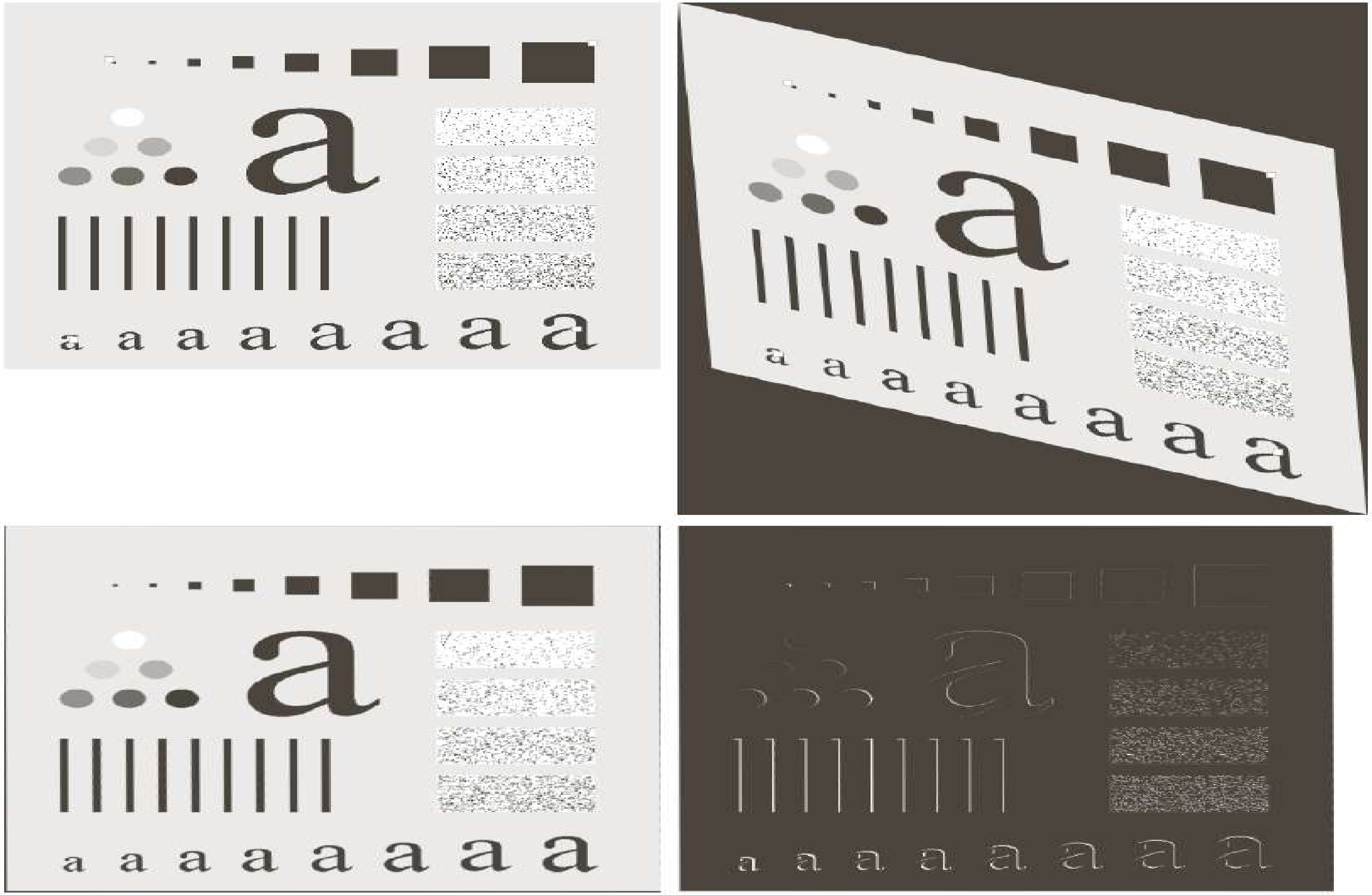
➢ The corresponding points are known precisely in the input and output (**reference**) images.

Image Registration

- A simple model based on bilinear approximation:

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

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



a b
c d

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

Image Transformation

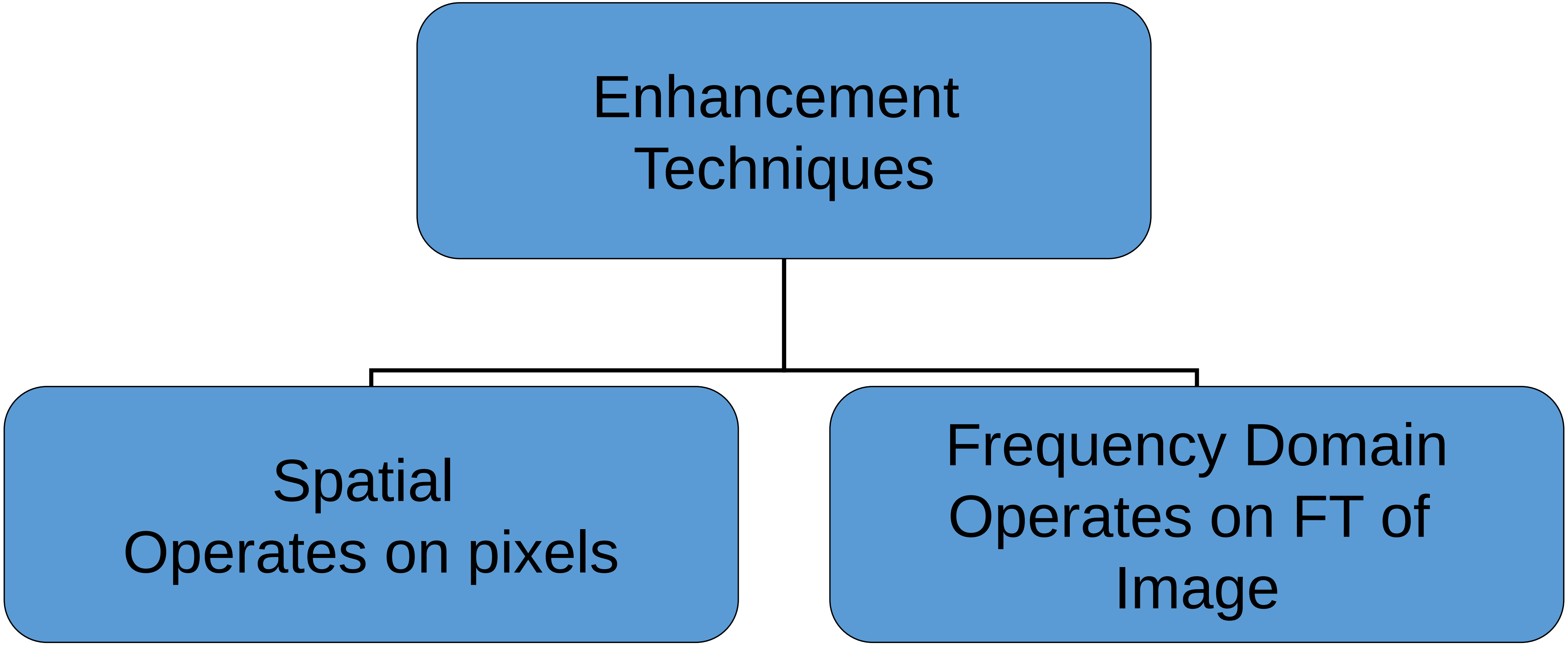

Enhancement Techniques
Spatial
Operates on pixels
Frequency Domain
Operates on FT of Image

Image Enhancement Definition

- **Image Enhancement:** is the process that improves the quality of the image for a specific application

Image Enhancement Methods

- **Spatial Domain Methods (Image Plane)**

  Techniques are based on direct manipulation of pixels in an image
- **Frequency Domain Methods**

  Techniques are based on modifying the Fourier transform of the image.
- **Combination Methods**

  There are some enhancement techniques based on various combinations of methods from the first two categories

Spatial Domain Methods

- As indicated previously, the term *spatial domain* refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

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

  Where f(x,y) in the input image, g(x,y) is the processed image and ***T*** is as operator on f, defined over some neighborhood of (x,y)

- In addition, ***T*** can operate on a set of input images.

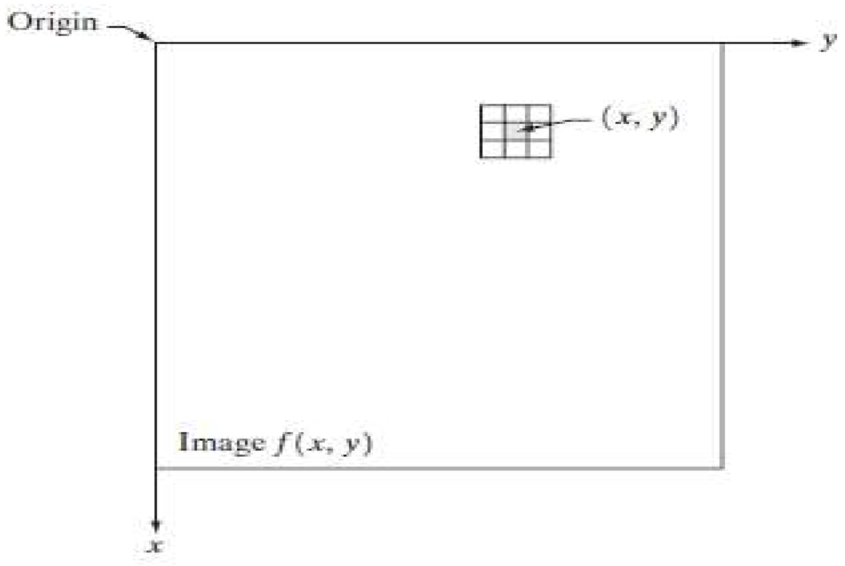


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image.

- The simplest form of ***T***, is when the neighborhood of size 1X1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

$$s = T(r)$$

Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

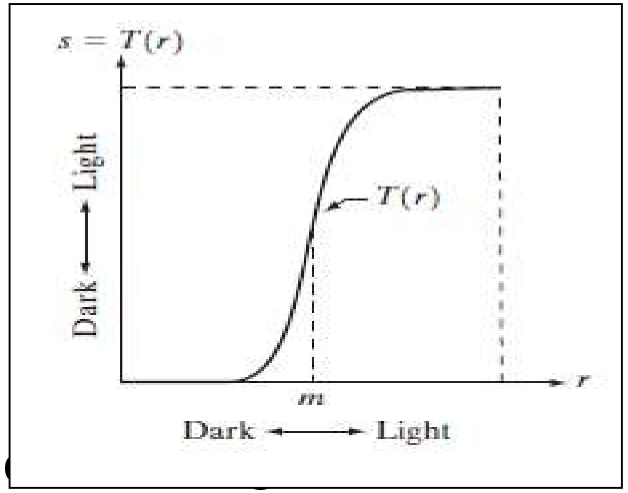
Examples of Enhancement Techniques

- **Contrast Stretching**:

If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

Produce higher contrast than the original image, by:

- Darkening the levels below m in the original image
- Brightening the levels above m in the original image



So, Contrast Stretching: is a simple image

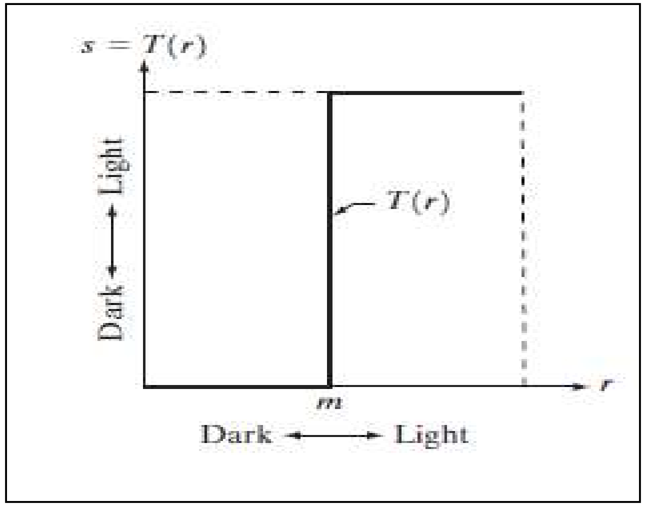
enhancement technique that improves the contrast

in an image by 'stretching' the range of intensity values it contains to span a (
Typically, it uses a linear function

Examples of Enhancement Techniques

- **Thresholding**

Is a limited case of contrast stretching, it produces a two-level (binary) image.



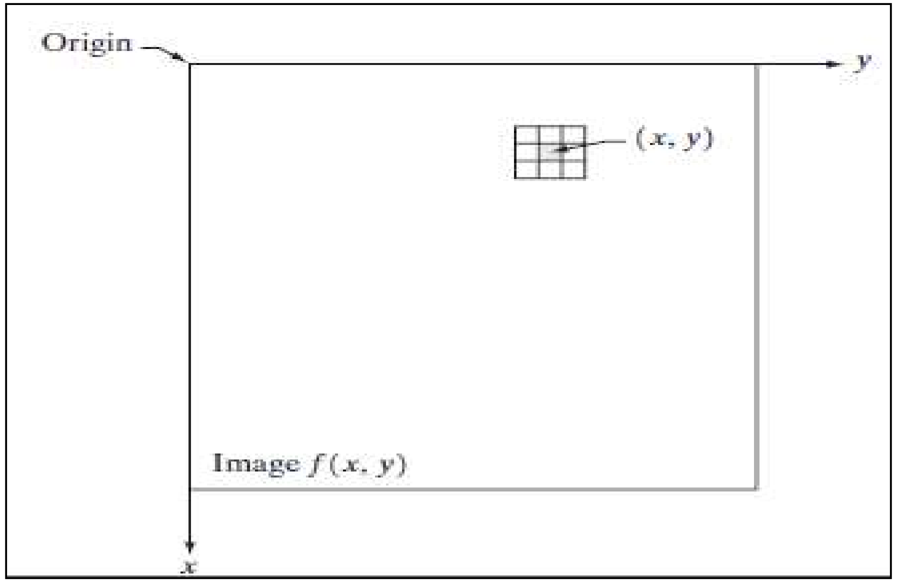
Some fairly simple, yet powerful, processing approaches can be formulated with grey-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as *point processing*.

Examples of Enhancement Techniques

Larger neighborhoods allow considerable more flexibility. The general approach is to use a function of the values of *f* in a predefined neighborhood of (x,y) to determine the value of *g* at (x,y).

One of the principal approaches in this formulation is based on the use of so-called *masks* (also referred to as *filters)*

So, a mask/filter: is a small (say 3X3) 2-D array, such as the one shown in the figure, in which the values of the mask coefficients determine the nature of the process, such as *image sharpening*. Enhancement techniques based on this type of approach often are referred to as *mask processing* or *filtering*.



Some Basic Intensity (Gray-level) Transformation Functions

- Grey-level transformation functions (also called, intensity functions), are considered the simplest of all image enhancement techniques.
- The value of pixels, before an after processing, will be denoted by *r* and *s*, respectively. These values are related by the expression of the form:

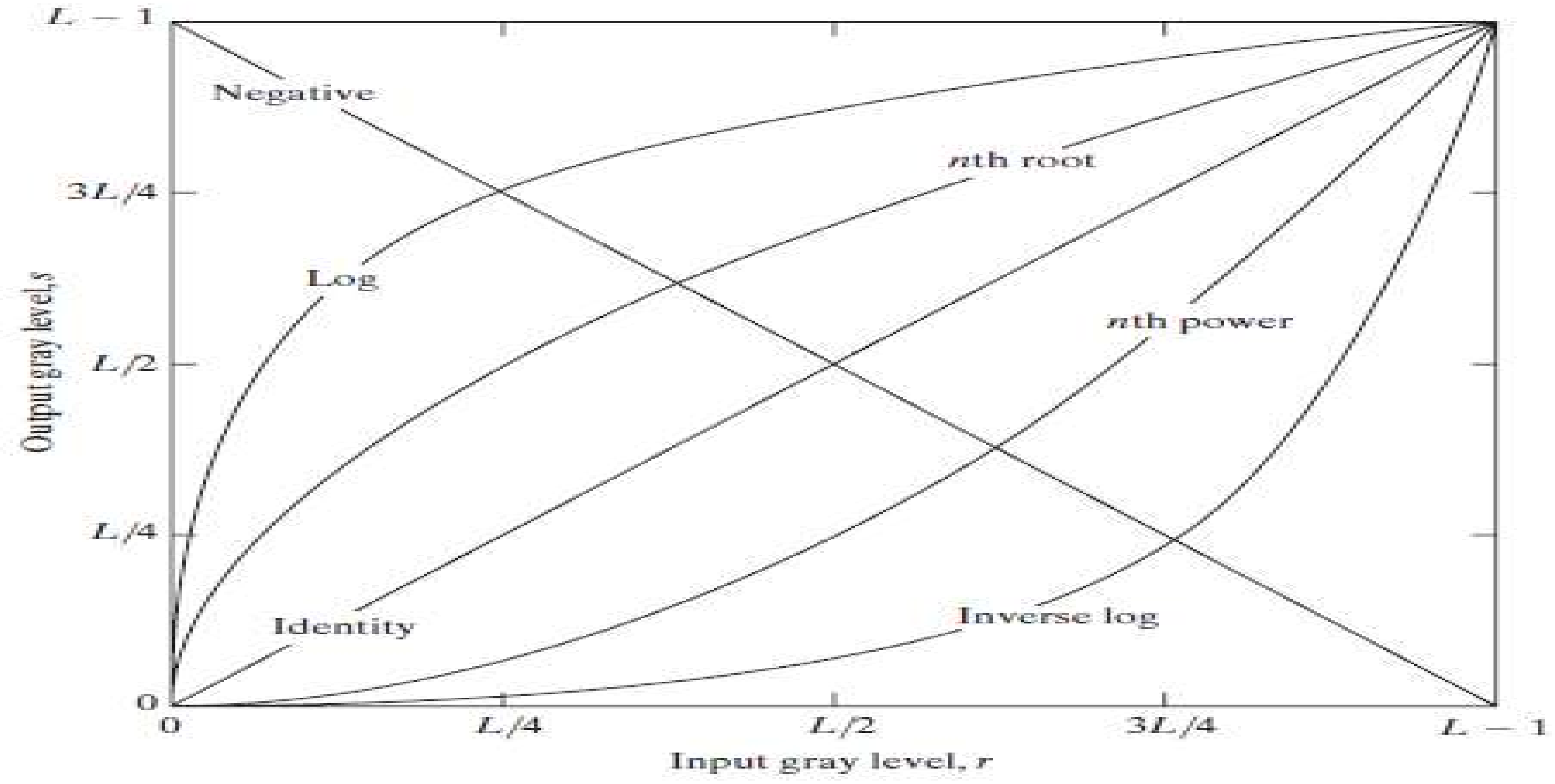
$$s = T(r)$$

where T is a transformation that maps a pixel value r into a pixel value s.

Some Basic Intensity (Gray-level) Transformation Functions

Consider the following figure, which shows three basic types of functions used frequently for image enhancement:

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Some Basic Intensity (Gray-level) Transformation Functions

- The three basic types of functions used frequently for image enhancement:
  - Linear Functions:
    - Negative Transformation
    - Identity Transformation
  - Logarithmic Functions:
    - Log Transformation
    - Inverse-log Transformation
  - Power-Law Functions:
    - n^{th} power transformation
    - n^{th} root transformation

Linear Functions

- **Identity Function**
  - Output intensities are identical to input intensities
  - This function doesn’t have an effect on an image, it was included in the graph only for completeness
  - Its expression:

    s = r

Linear Functions

- **Image Negatives (Negative Transformation)**
  - The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

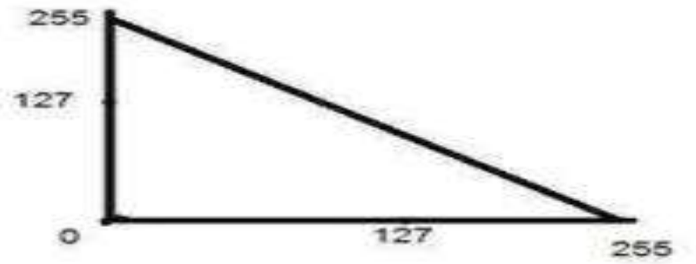
    $$s = L - 1 - r$$

    Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative.

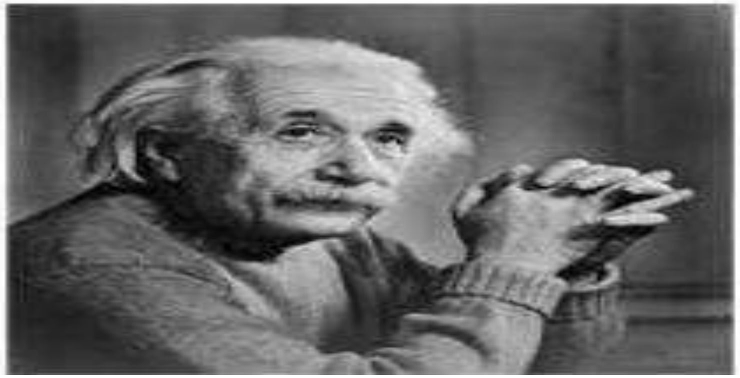
  - The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

Negative transformation example

Graph representation

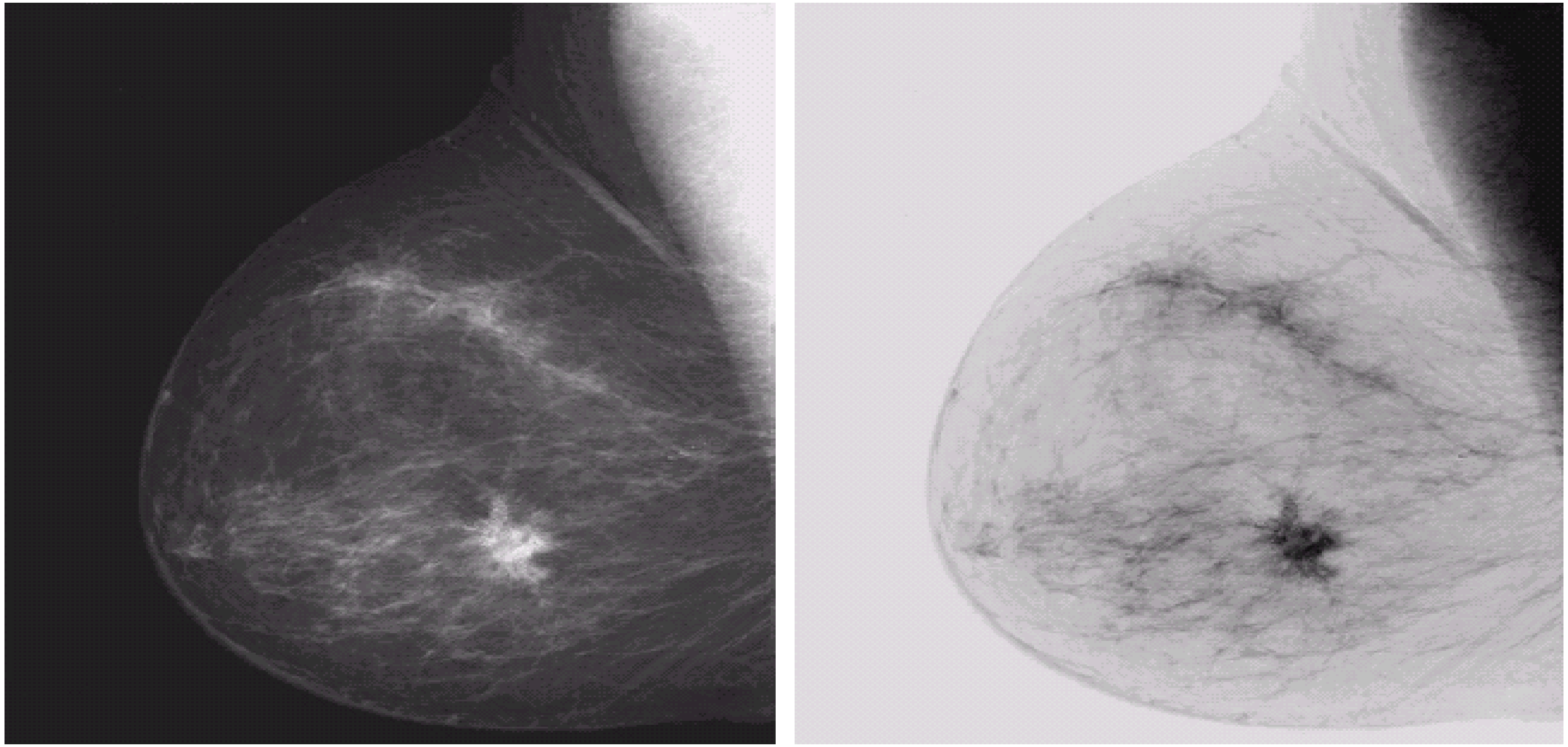


Input image



Output image

Image Negative



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.)

Image Negative: $s = L - 1 - r$

Logarithmic Transformations

- **Log Transformation**

  The general form of the log transformation:

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

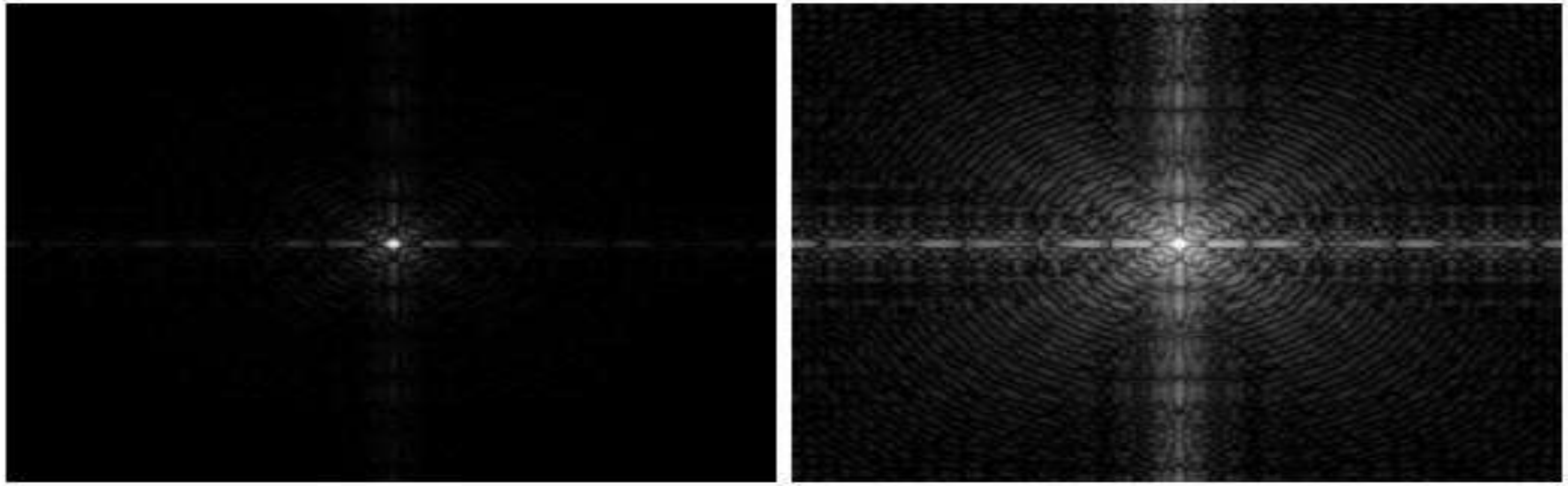
  Where c is a constant, and $r \geq 0$

  - Log curve maps a narrow range of low gray-level values in the input image into a wider range of the output levels.
  - Used to expand the values of dark pixels in an image while compressing the higher-level values.
  - It compresses the dynamic range of images with large variations in pixel values.

Logarithmic Transformations

a b

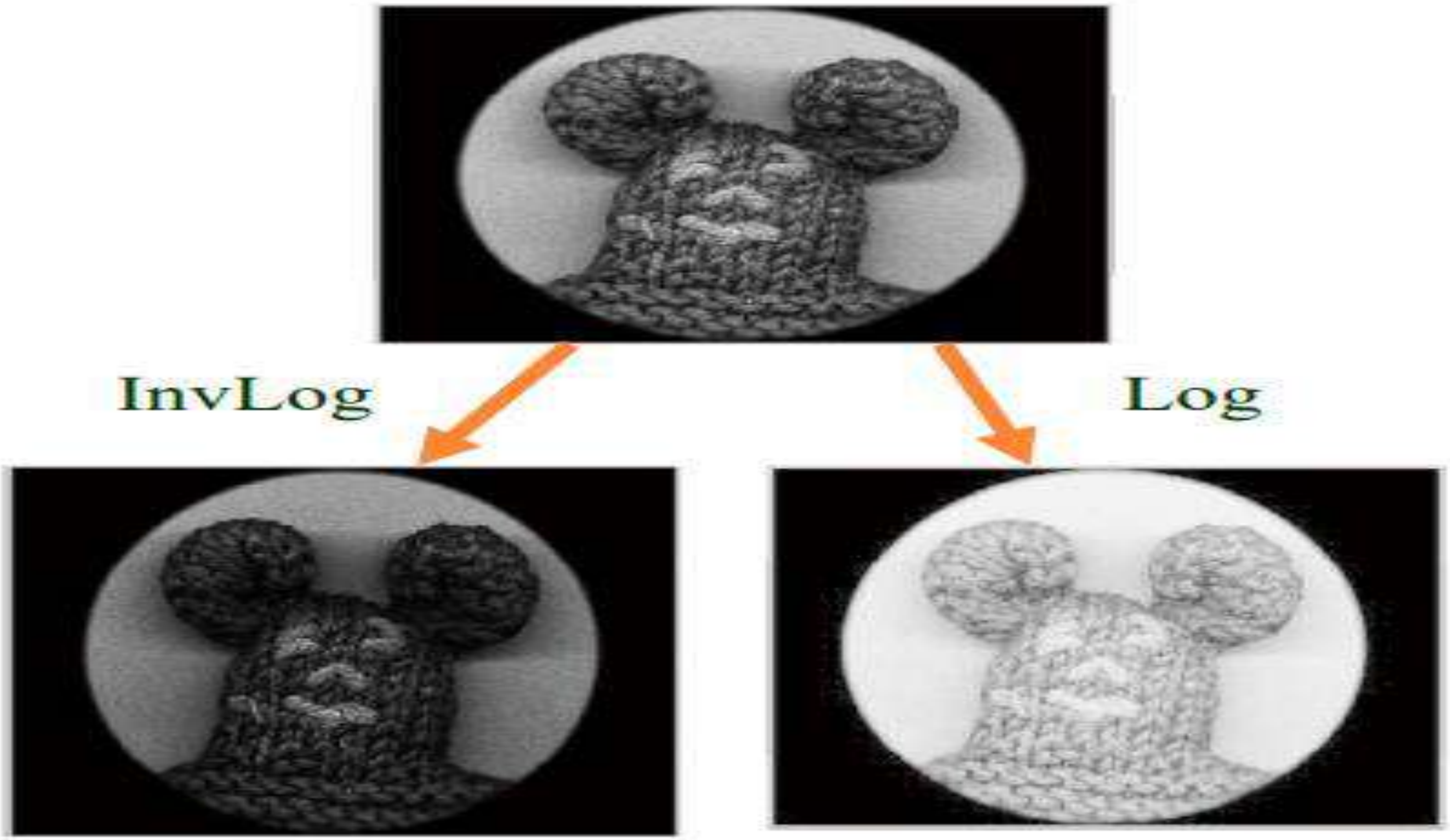
FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Logarithmic Transformations

- **Inverse Logarithm Transformation**
  - Do opposite to the log transformations
  - Used to expand the values of high pixels in an image while compressing the darker-level values.

Log transformation example



Power-Law Transformations

- Power-law transformations have the basic form of:

  $$s = c.r^{\gamma}$$

  Where c and γ are positive constants

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)

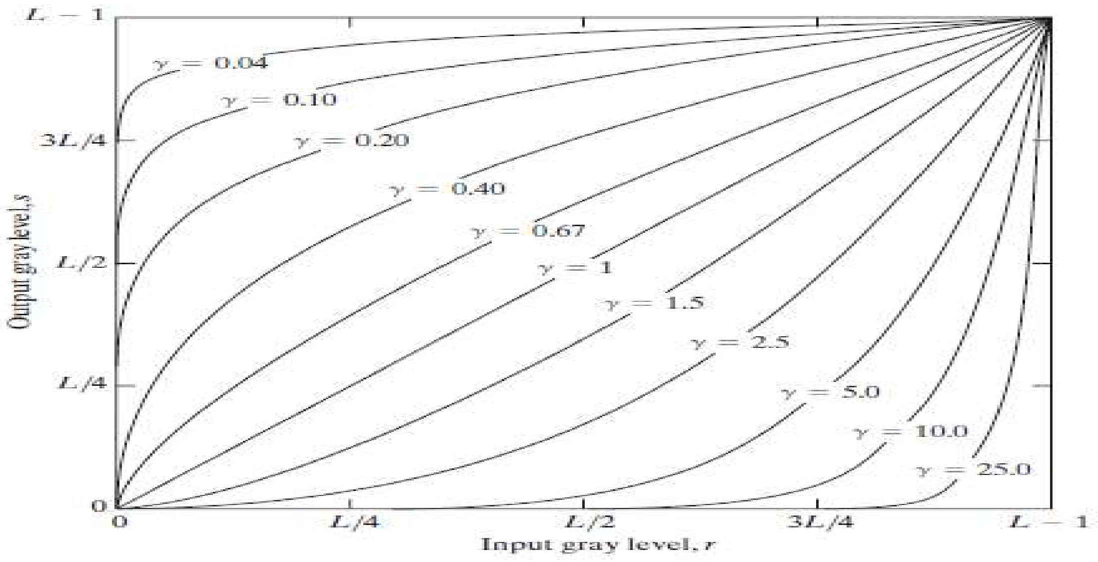


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

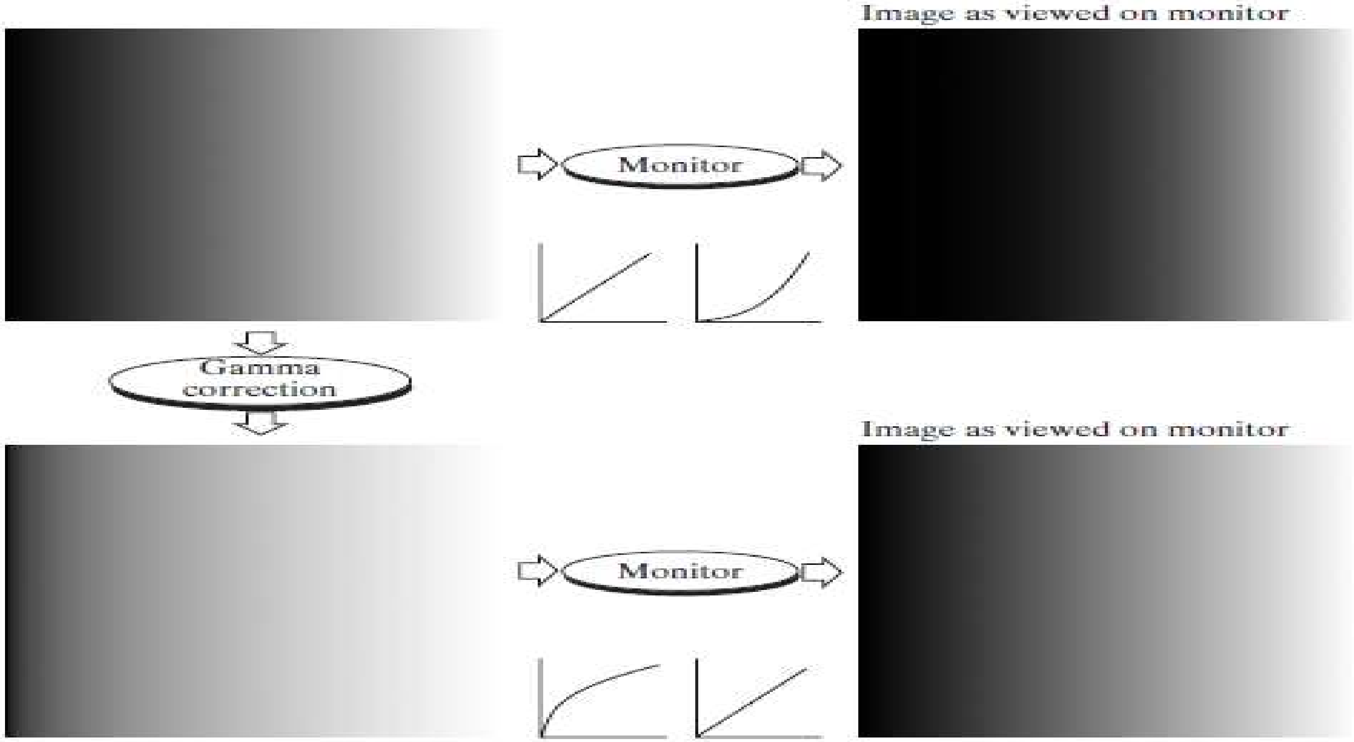
Power-Law Transformations

- Variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power-law response phenomena is called ***gamma correction.***

  For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple gray-scale linear wedge input into a CRT monitor. As expected, the output of the monitor appears darker than the input, as shown in Fig. 3.7(b). Gamma correction

  in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation. The result is shown in Fig. 3.7(c).When input into the same monitor, this gamma-corrected input produces an output that is close in appearance to the original image, as shown in Fig. 3.7(d).

Power-Law Transformation

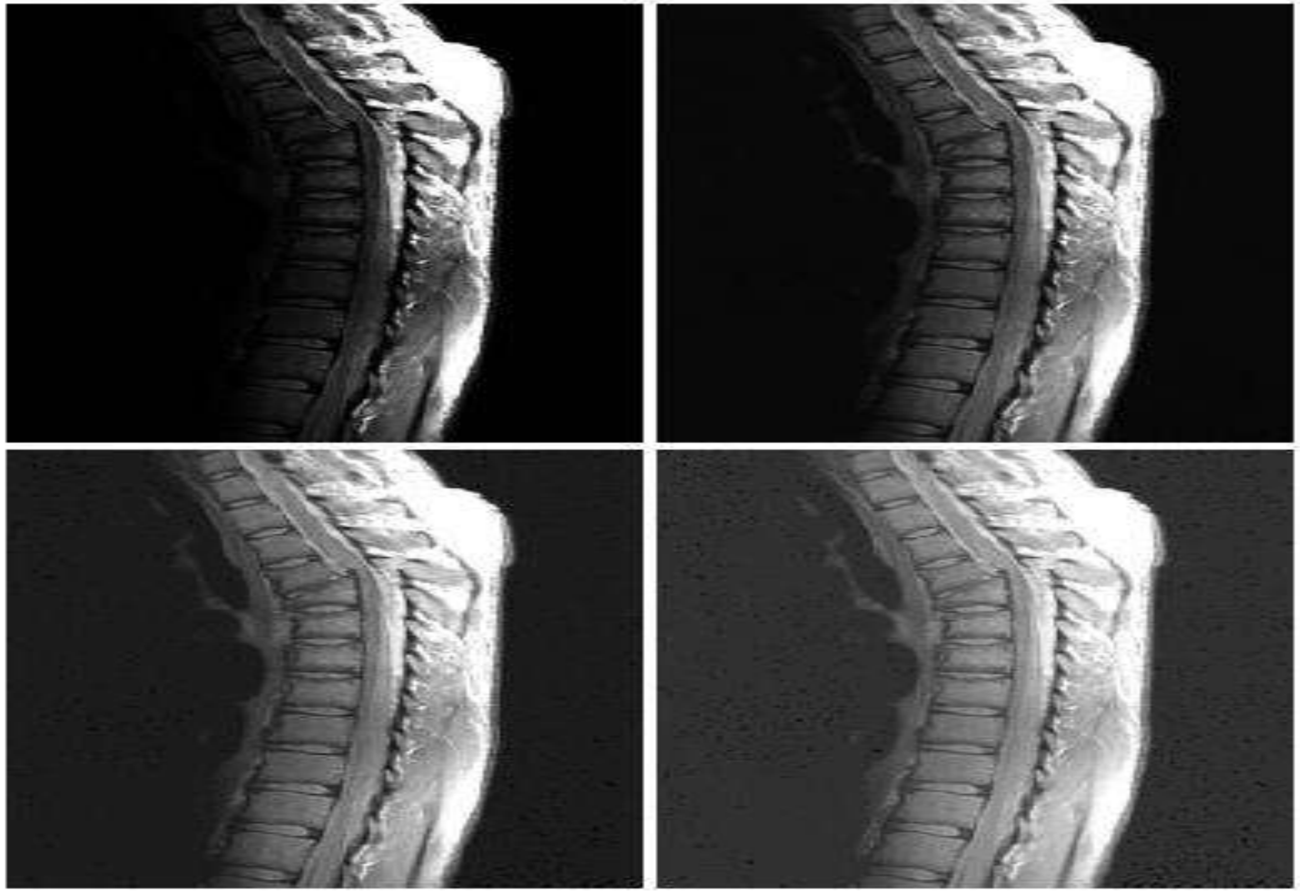


a b
c d

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

Power-Law Transformation

- In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation. See figure 3.8



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.)

Power-Law Transformation

- Another illustration of Power-law transformation

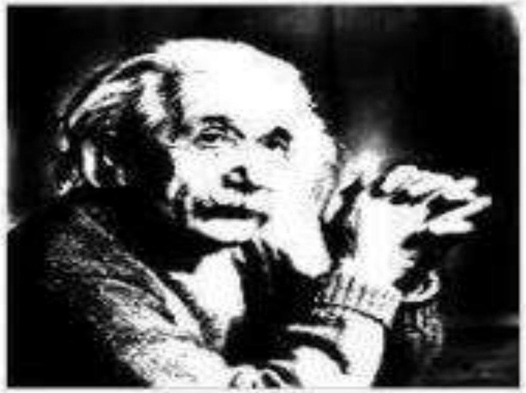
a b
c d

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.)

Power law transformation example

Gamma=10

Gamma=8



Gamma=6

Piecewise-Linear Transformation Functions

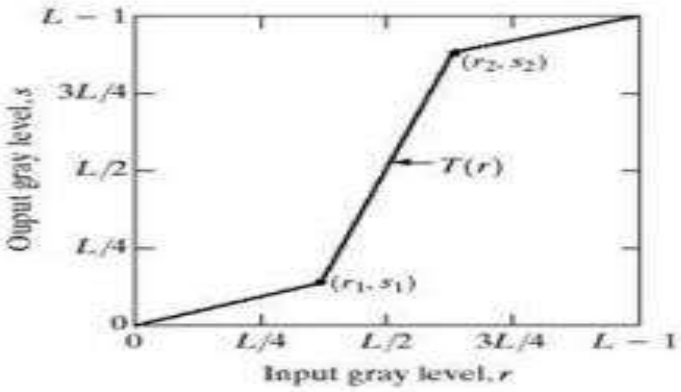
- **Principle Advantage:** Some important transformations can be formulated only as a piecewise function.
- **Principle Disadvantage:** Their specification requires more user input that previous transformations
- **Types of Piecewise transformations are:**
  - Contrast Stretching
  - Gray-level Slicing
  - Bit-plane slicing

Contrast Stretching

- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

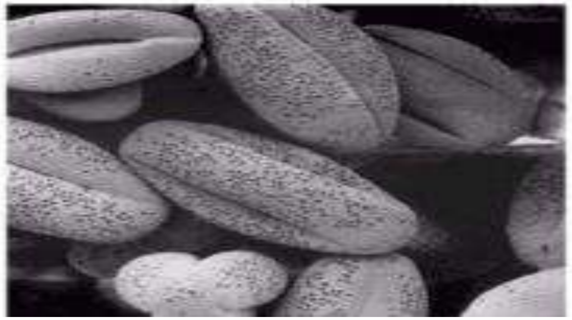
CONTRAST STRETCHING EXAMPLE

Transformation function

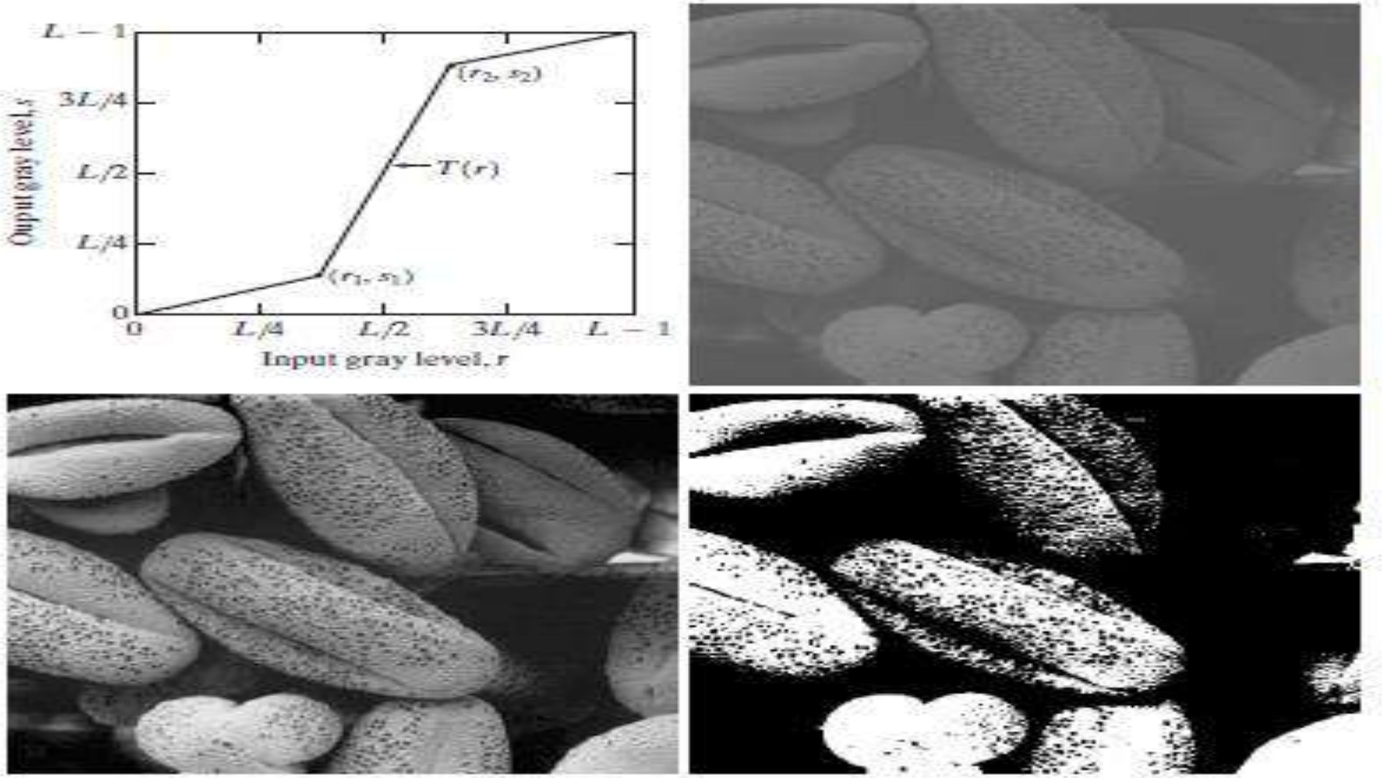


Low contrast image

Contrast stretching image



Contrast Stretching



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast Stretching

- Figure 3.10(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.
- If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels.
- If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a *thresholding function* that creates a binary image. As shown previously in slide 7.
- Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, r1 ≤ r2 and s1 ≤ s2 is assumed, so the function is always increasing.

Contrast Stretching

- Figure 3.10(b) shows an 8-bit image with low contrast.

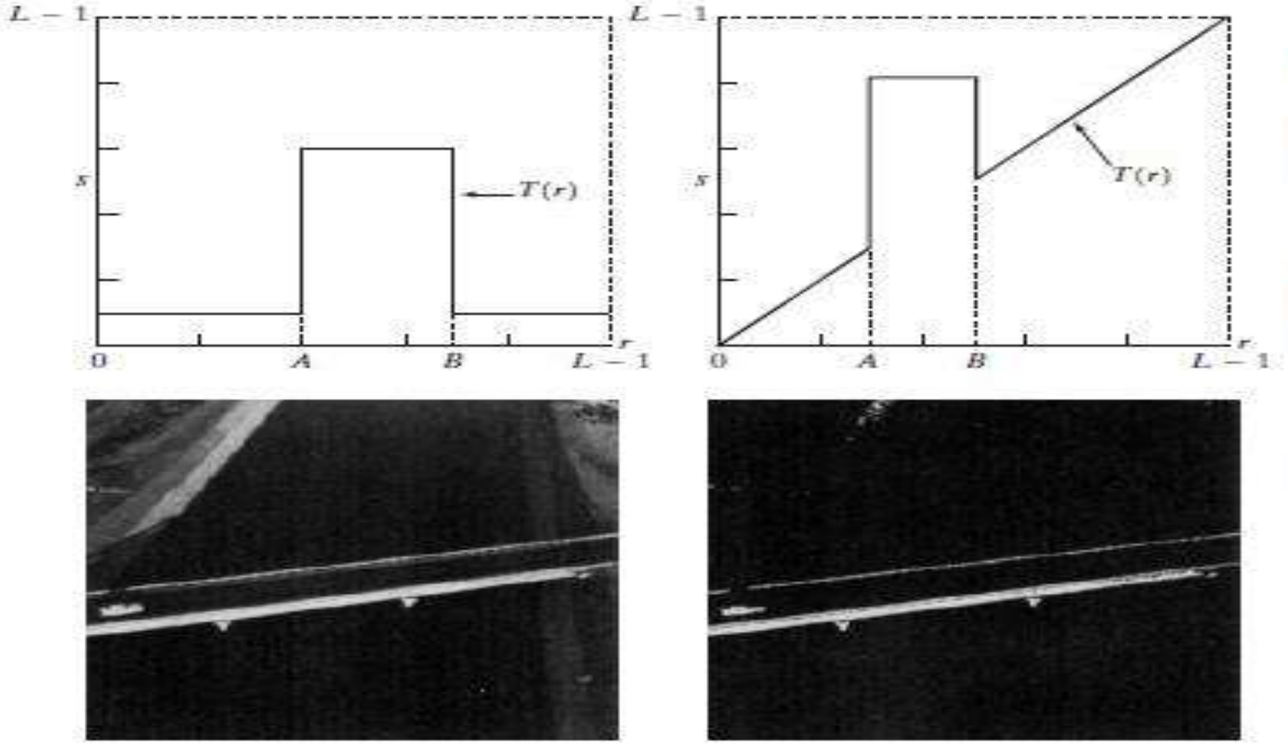
- Fig. 3.10(c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

- Finally, Fig. 3.10(d) shows the result of using the *thresholding function* defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig 3.11 (a), produces a binary image.
  - The second approach, based on the transformation shown in Fig 3.11 (b), brightens the desired range of gray levels but preserves gray levels unchanged.
  - Fig 3.11 (c) shows a gray scale image, and fig 3.11 (d) shows the result of using the transformation in Fig 3.11 (a).

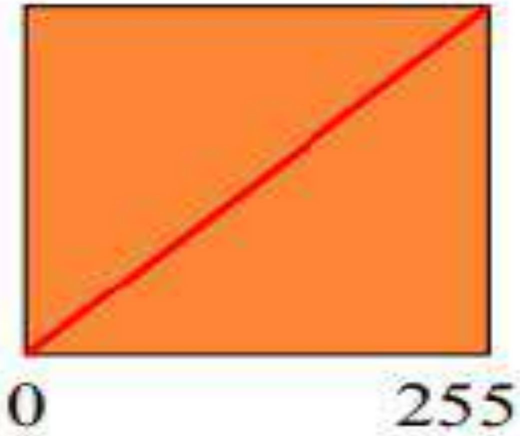
Gray-level Slicing



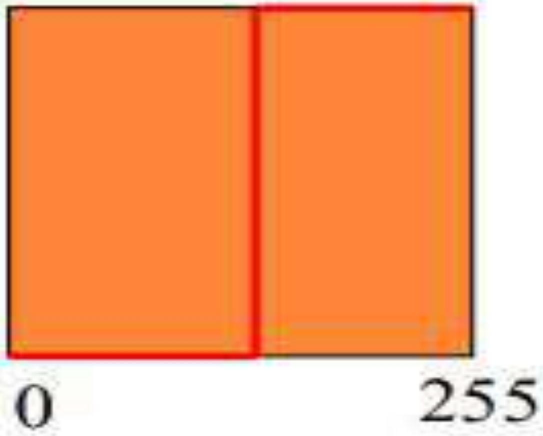
a b
c d

FIGURE 3.11
(a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level.
(b) This transformation highlights range $[A, B]$ but preserves all other levels.
(c) An image.
(d) Result of using the transformation in (a).

Input image



Output image



Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

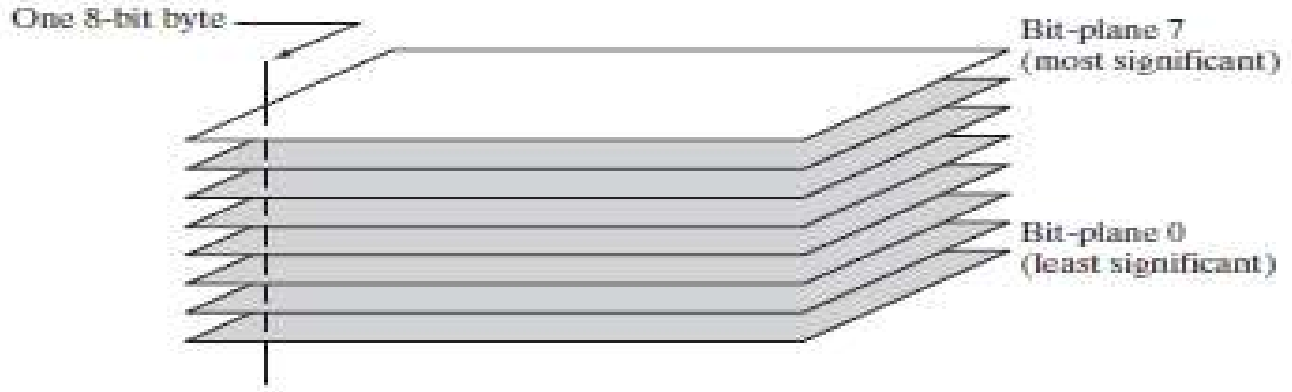


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

BIT PLANE SLICING EXAMPLE

Original image

Bit plane 7

Bit plane 6

Bit plane 4

Bit plane 1

1. Which of the following expression is used to denote spatial domain process?
a) g(x,y)=T[f(x,y)]
b) f(x+y)=T[g(x+y)]
c) g(xy)=T[f(xy)]
d) g(x-y)=T[f(x-y)]

1. Which of the following expression is used to denote spatial domain process?

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

b) f(x+y)=T[g(x+y)]

c) g(xy)=T[f(xy)]

d) g(x-y)=T[f(x-y)]

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

Image Transform

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

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

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

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

variables u and v are the transform variables,

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

Image Transform

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

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

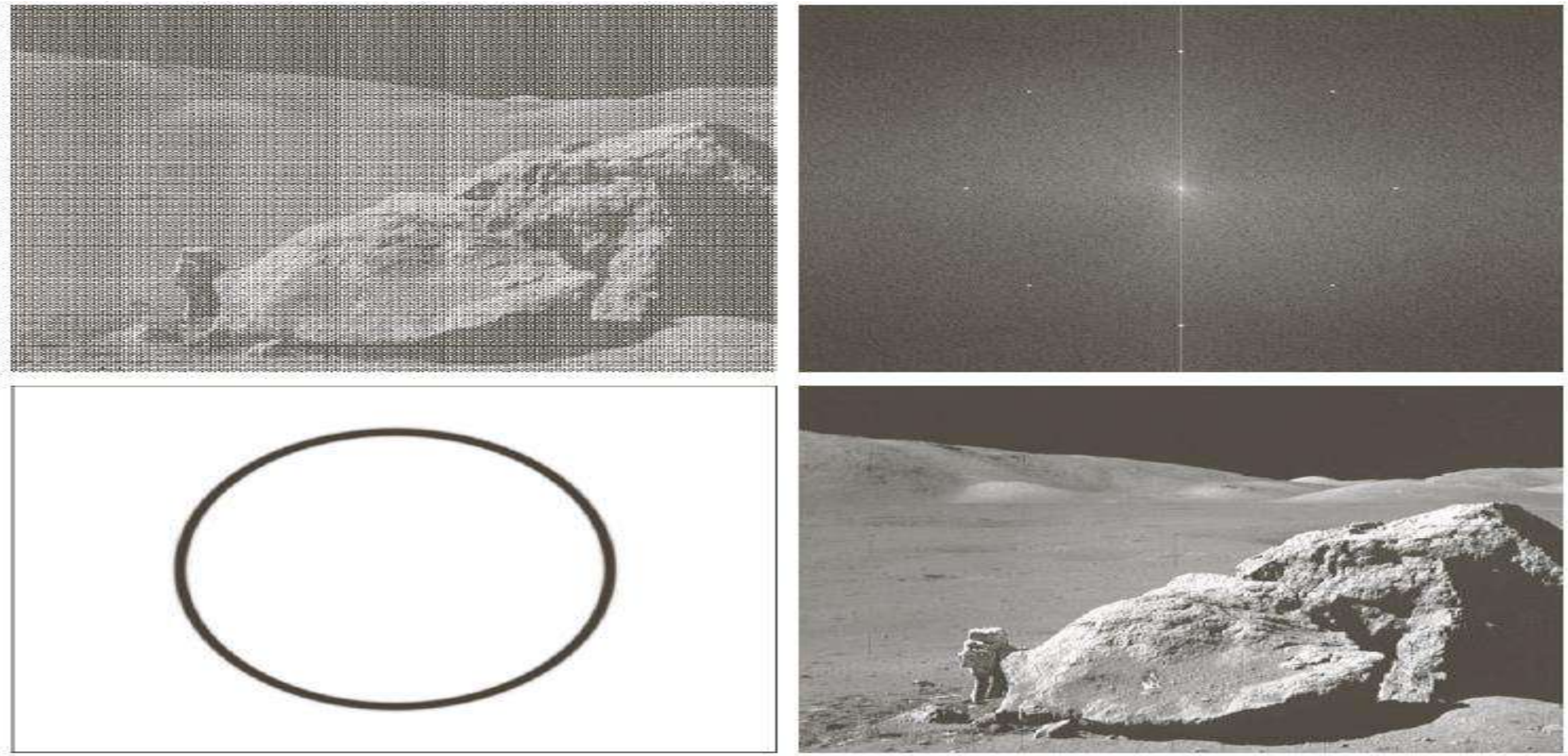
where $s(x,y,u,v)$ is the *inverse transformation kernel*,

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

Image Transform

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

Example: Image Denoising by Using DCT Transform



a b
c d

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

Forward Transform Kernel

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

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

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

In addition, the kernel is said to be SYMMETRIC if $r_1(x,u)$ is functionally equal to $r_2(y,v)$, so that

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

The Kernels for 2-D Fourier Transform

The *forward* kernel

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

Where $j = \sqrt{-1}$

The *inverse* kernel

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

2-D Fourier Transform

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

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

Probabilistic Methods

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

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

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

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

The mean (average) intensity is given by

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

Probabilistic Methods

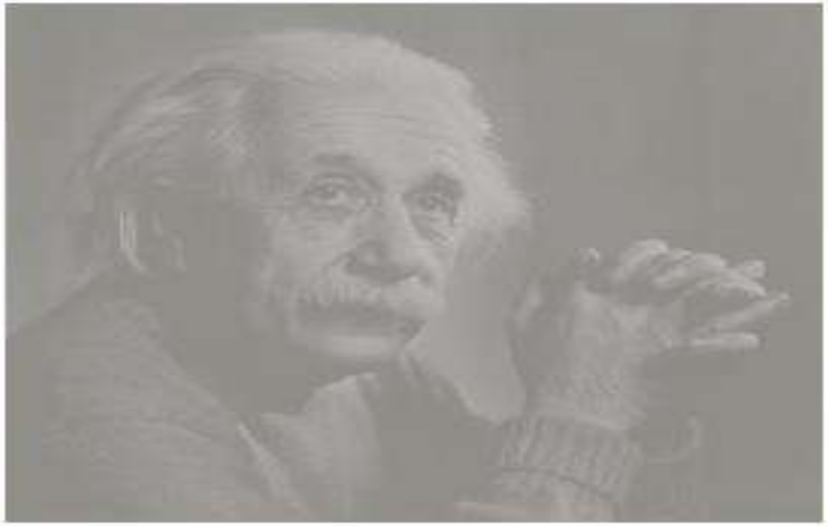
The variance of the intensities is given by

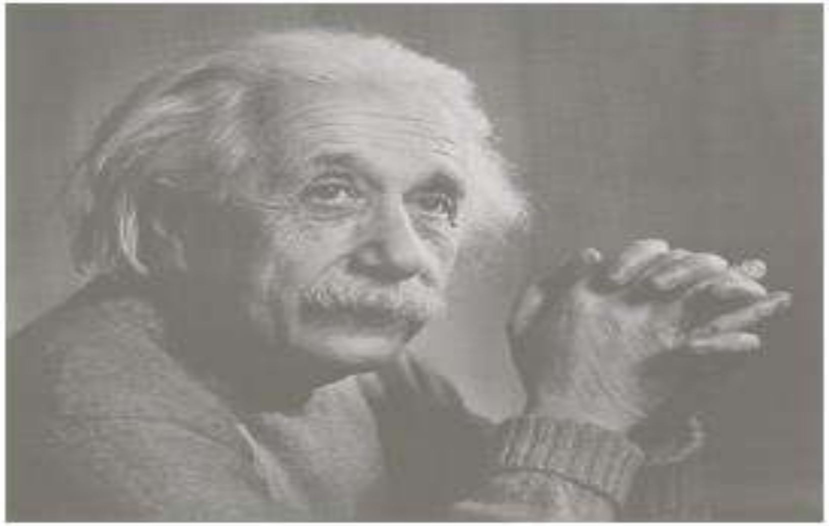
$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 \, p(z_k)$$

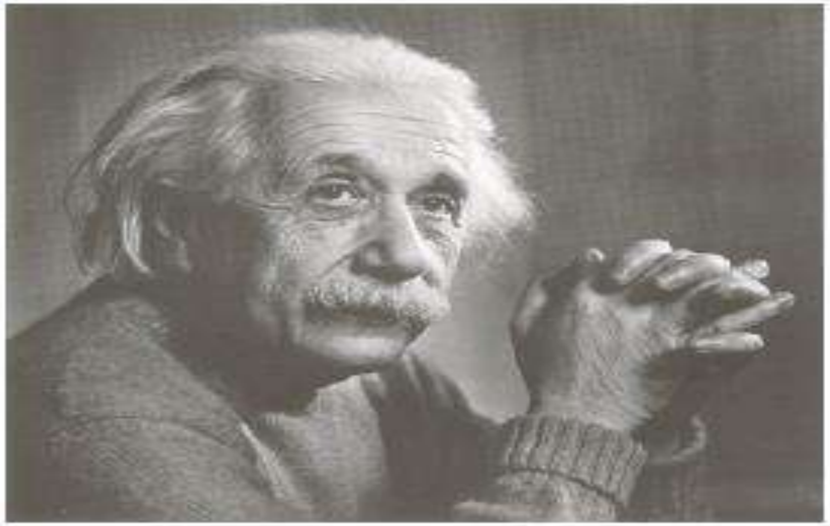
The n^{th} moment of the intensity variable z is

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

Example: Comparison of Standard Deviation Values







$\sigma = 14.3$ $\sigma = 31.6$ $\sigma = 49.2$

Image Transform

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

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

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

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

Image Transform

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

Example: Image Denoising by Using DCT Transform

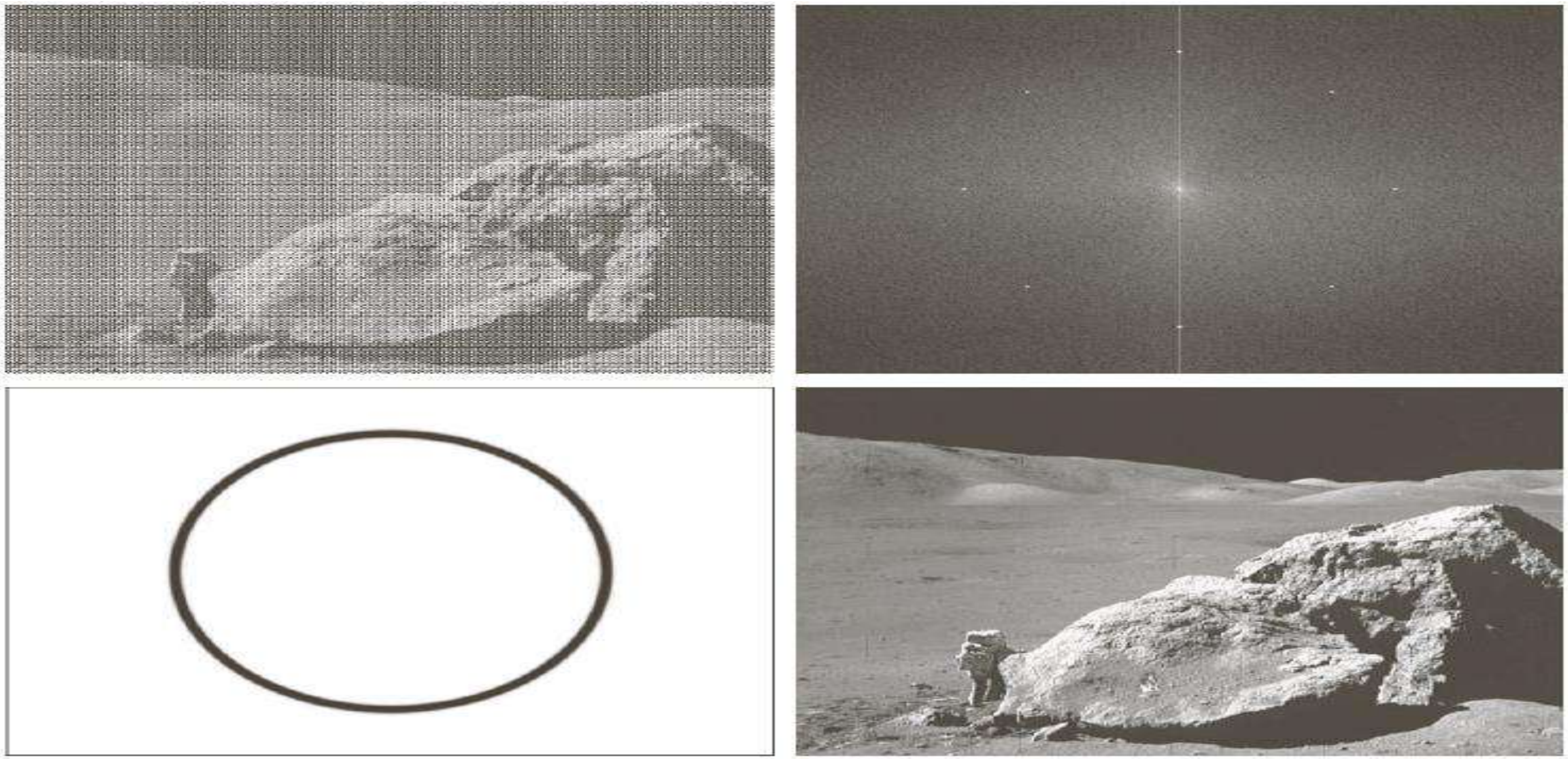


FIGURE 2.40 (a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy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

Forward Transform Kernel

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

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

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

In addition, the kernel is said to be SYMMETRIC if

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

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

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

The Kernels for 2-D Fourier Transform

The *forward* kernel

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

Where $j = \sqrt{-1}$

The *inverse* kernel

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

2-D Fourier Transform

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

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

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013

Probabilistic Methods

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

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

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

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

The mean (average) intensity is given by

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

Probabilistic Methods

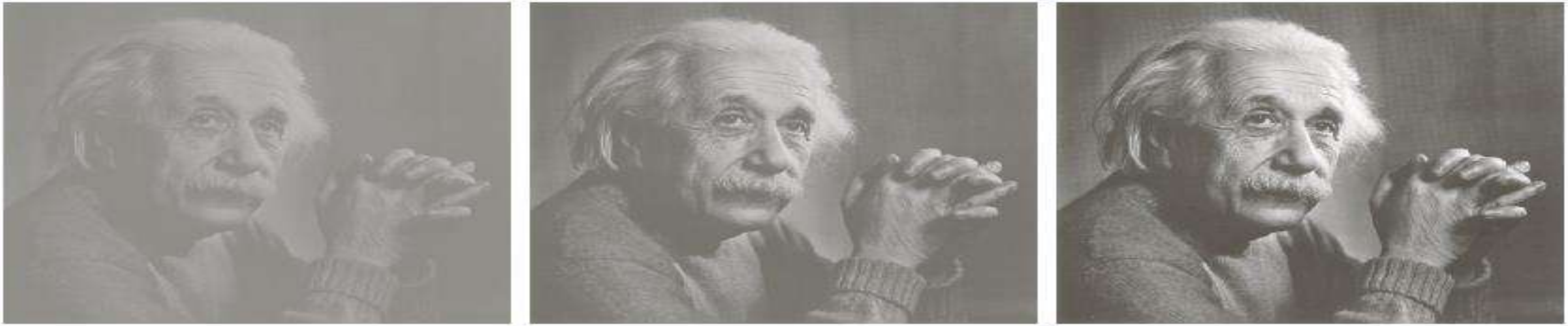
The variance of the intensities is given by

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

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

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

Example: Comparison of Standard Deviation Values



$\sigma = 14.3$ $\sigma = 31.6$ $\sigma = 49.2$

Arithmetic and Logic Operations for Image processing

Arithmetic and Logic Operations

- Arithmetic and logic operations are often applied a preprocessing steps in image analysis in order to combine images in various way.
- Addition, subtraction, division and multiplication comprise the arithmetic operation, while AND , OR, and NOT make up the logic operations.
- These operation performed on two image , except the NOT logic operation which require only one image, and are done on a pixel by pixel basis.

Image Arithmetic

- For input images f_1 and f_2 and some function Op:

$$g(x, y) = \mathrm{Op}\,(f_1(x, y), f_2(x, y))$$

  The operator is applied pairwise to each pixel in the images.

- Pseudocode:

```
for all pixel positions x, y:
  out[x,y] = func( image1[x,y] , image2[x,y]
)
```

- Possibilities: addition, subtraction, and, or, ...

Arithmetic operations

- Addition: $g(x,y) = f_1(x,y) + f_2(x,y)$
- Subtraction: $g(x,y) = f_1(x,y) - f_2(x,y)$
- Multiplication: $g(x,y) = f_1(x,y) \cdot f_2(x,y)$
- Division: $g(x,y) = f_1(x,y) / f_2(x,y)$

Addition/Blending

Used to create double-exposures or composites:

$$g(x, y) = f_1(x, y) + f_2(x, y)$$

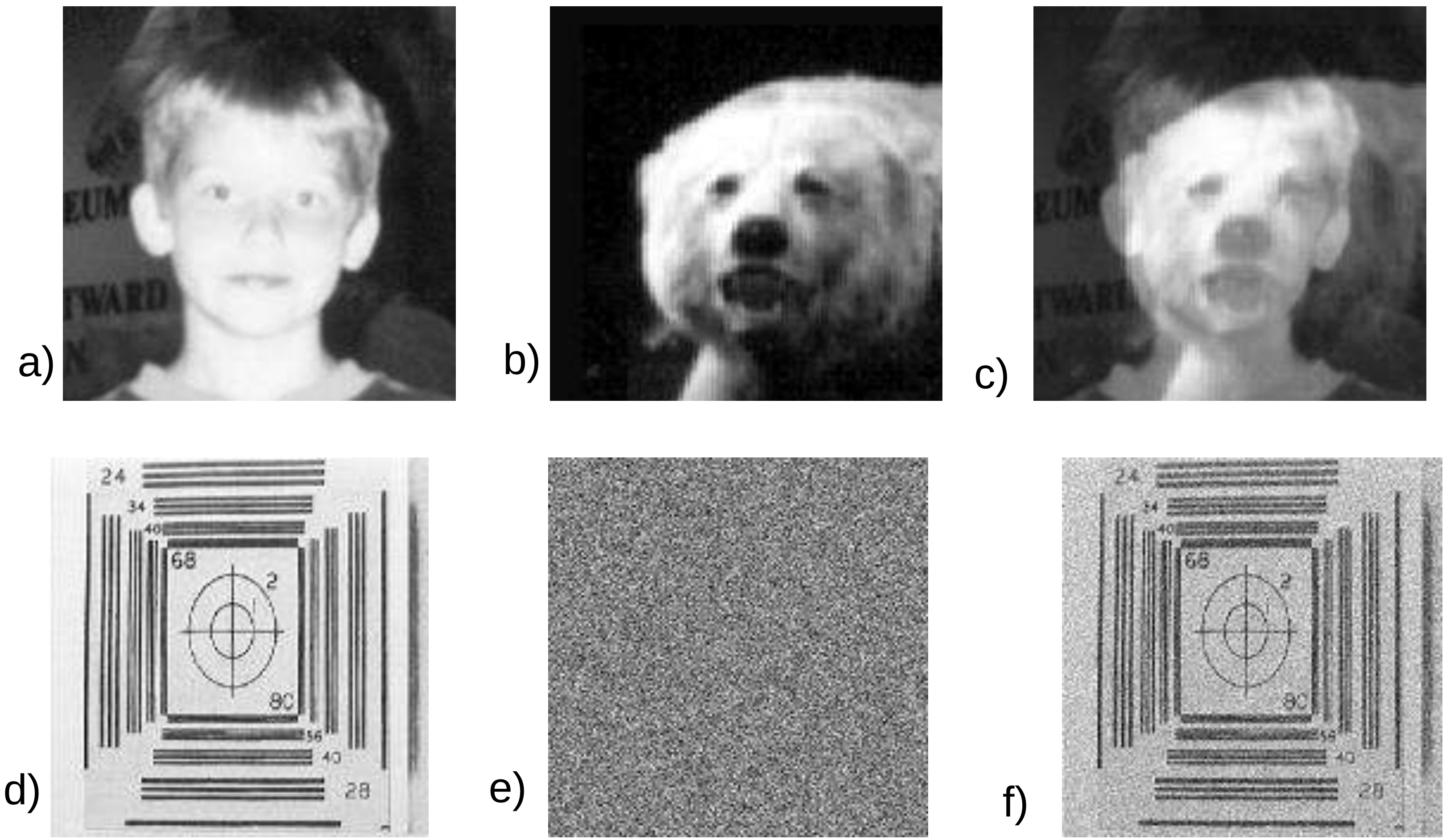
+

=

Can also do a weighted *blend*:

$$g(x, y) = \alpha_1 f_1(x, y) + \alpha_2 f_2(x, y)$$

Figure 3.2-6 Image Addition Examples. This example shows one step in the *image morphing* process where an increasing percentage of the second image is slowly added to the first, and a geometric transformation is usually required to align the images. a) first original, b) second original, c) addition of images (a) and (b). This example shows adding noise to an image which is often useful for developing image restoration models. d) original image, e) Gaussian noise, variance = 400, mean = 0, f) addition of images (d) and (e).



Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y) / f_2(x, y)$

Subtraction

- Useful for finding changes between two images

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

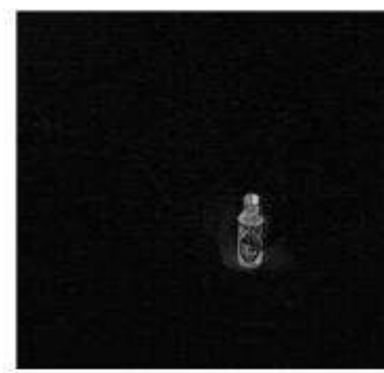
- Sometimes more useful to use *absolute difference*

$$g(x, y) = |f_1(x, y) - f_2(x, y)|$$

- What's changed?

-

=



Subtraction

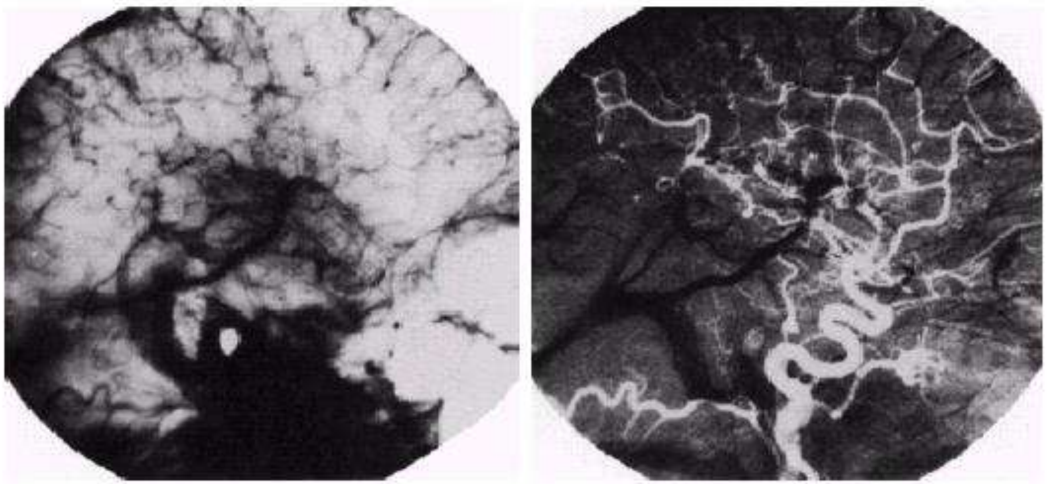
- Subtraction of two image is often used to detect motion.
- Consider the case where nothing has changed in a scene; the image resulting from subtraction of two sequential image is filled with zeros - a black image.
- If something has moved in the scene, subtraction produce a nonzero result at the location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

Subtraction

- Medical imaging often uses this type of operation to allow the doctor to more readily see changes which are helpful in the diagnosis.
- The technique is also used in law enforcement and military applications; for example, to find an individual in a crowd or to detect changes in a military installation.

Digital Subtraction Angiography

1. Take an x-ray
2. Inject patient with a radio-opaque dye
   (and tell them not to move!)
3. Take another x-ray
4. Subtract the two



a b

FIGURE 3.29
Enhancement b
image subtracti
(a) Mask image
(b) An image
(taken after
injection of a
contrast mediu
into the
bloodstream) w
mask subtracte
out.

Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

Multiplication n Division

- used to adjust the brightness of an image.
  - is done on a pixel by pixel basis and the options are to multiply or divide an image by a constant value, or by another image.
  - Multiplication of the pixel value by a value greater than one will brighten the image (or division by a value less than 1), and division by a factor greater than one will darken the image (or multiplication by a value les than 1).
- Brightness adjustment by a constant is often used as a preprocessing step in image enhancement and is shown in Figure 3.2.8.

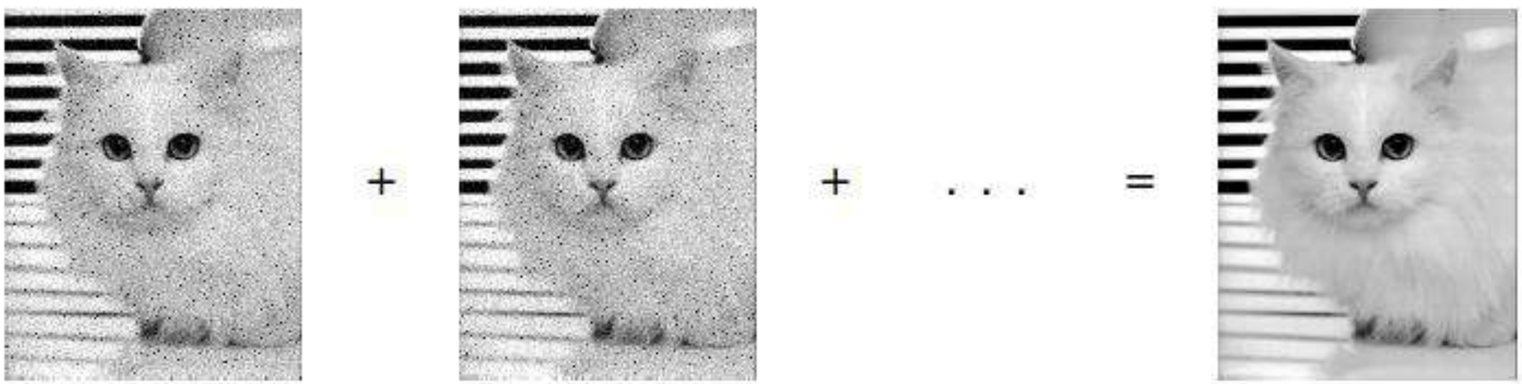
Figure 3.2-8 Image Division. a) original image, b) image divided by a value less than 1 to brighten, c) image divided a value greater than 1 to darken

a) b) c)

Image Averaging

Idea:
Average multiple pictures of the same scene to reduce noise

Similar in principle to acquiring the image for a longer duration.



Logic operations

- The logic operations AND, 0R and NOT operate in a bit-wise fashion on pixel data.
- Example
  - performing a logic AND on two images. Two corresponding pixel values are 111_{10} in one image and 88_{10} in the second image. The corresponding bit string are:

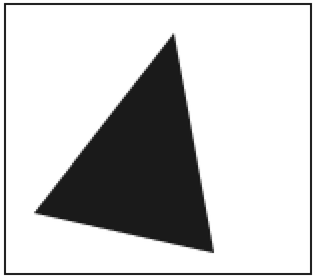
$111_{10} = 01101111_2$ $88 = 01011000_2$

$$
\begin{array}{ll}
 & 01101111_2 \\
\text{AND} & \underline{01011000}_2 \\
 & 01001000_2
\end{array}
$$

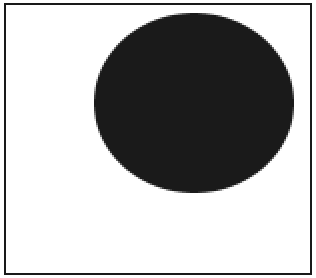
Master Layout 1

Image after logical operation is done

Original Images



IMG_1



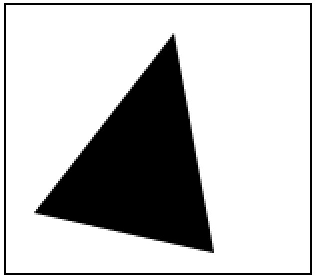
IMG_2

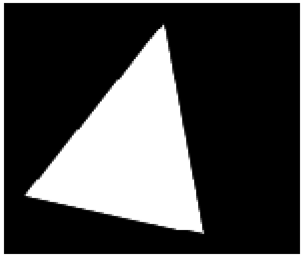
Give a dropdown box to select the operation

- The operations are: NOT, AND, OR, XOR, NOT-AND
- Give Start, Pause, Reset buttons

Step 1:

NOT

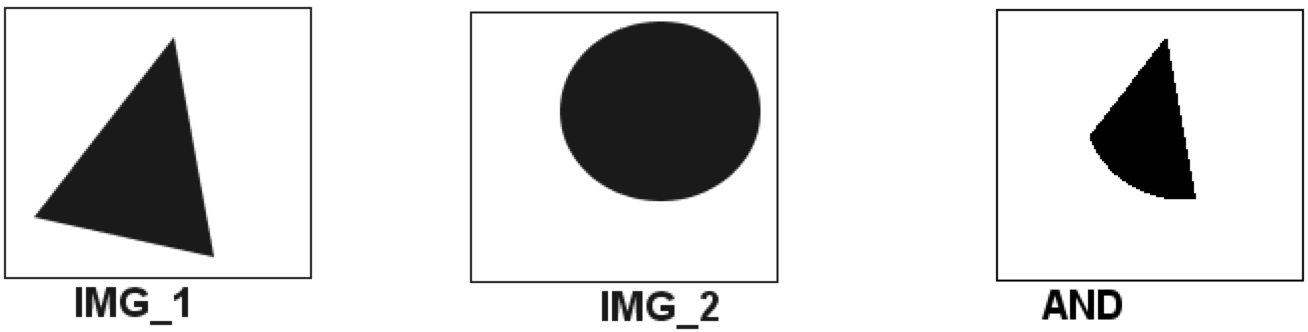




NOT

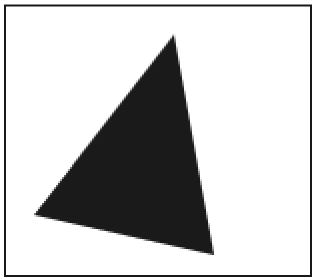
Step 2:

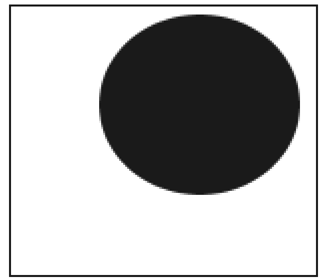
AND

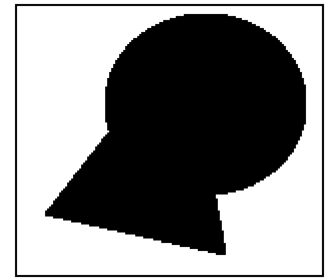


Step 3:

OR

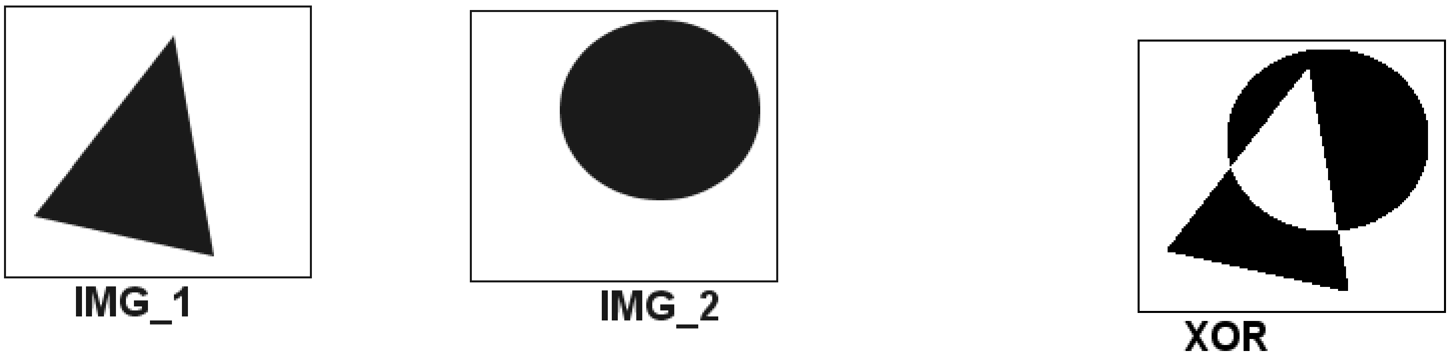

IMG_1


IMG_2


OR

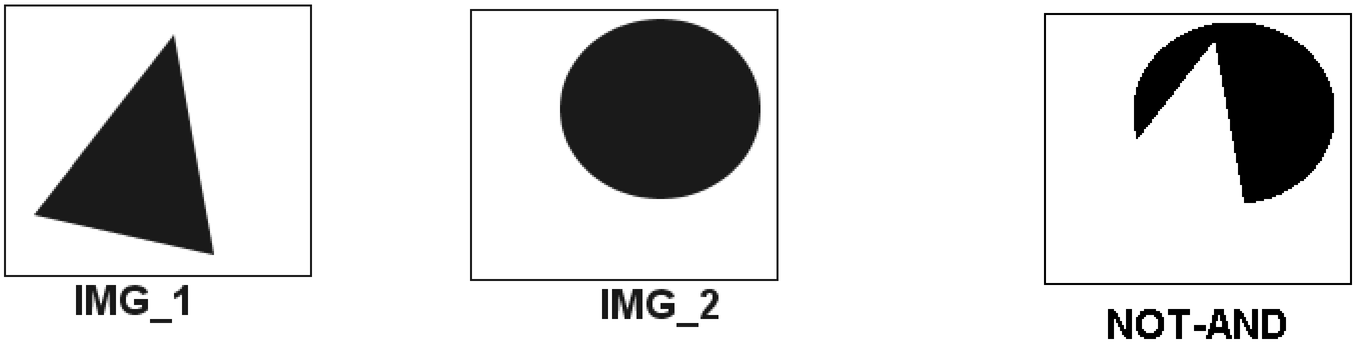
Step 4:

XOR



Step 6:

NOT-AND



- The logic operations AND and OR are used to combine the information in two images.
- This may be done for special effects but a more useful application for image analysis is to perform a masking operation.
- AND and OR can be used as a simple method to extract a ROI from an image.

- For example, a white mask ANDed with an image will allow only the portion of the image coincident with the mask to appear in the output image, with the background turned black; and a black mask ORed with an image will allow only the part f the image corresponding to the black mask to appear in the output image, but will turn the return of the image white.
- This process is called *image masking*

Figure 3.2-10 Image Masking. a) Original image, b) image mask for AND operation, c) Resulting image from (a) AND (b), d) image mask for OR operation, created by performing a NOT on mask (b), e) Resulting image from (a) OR (d).

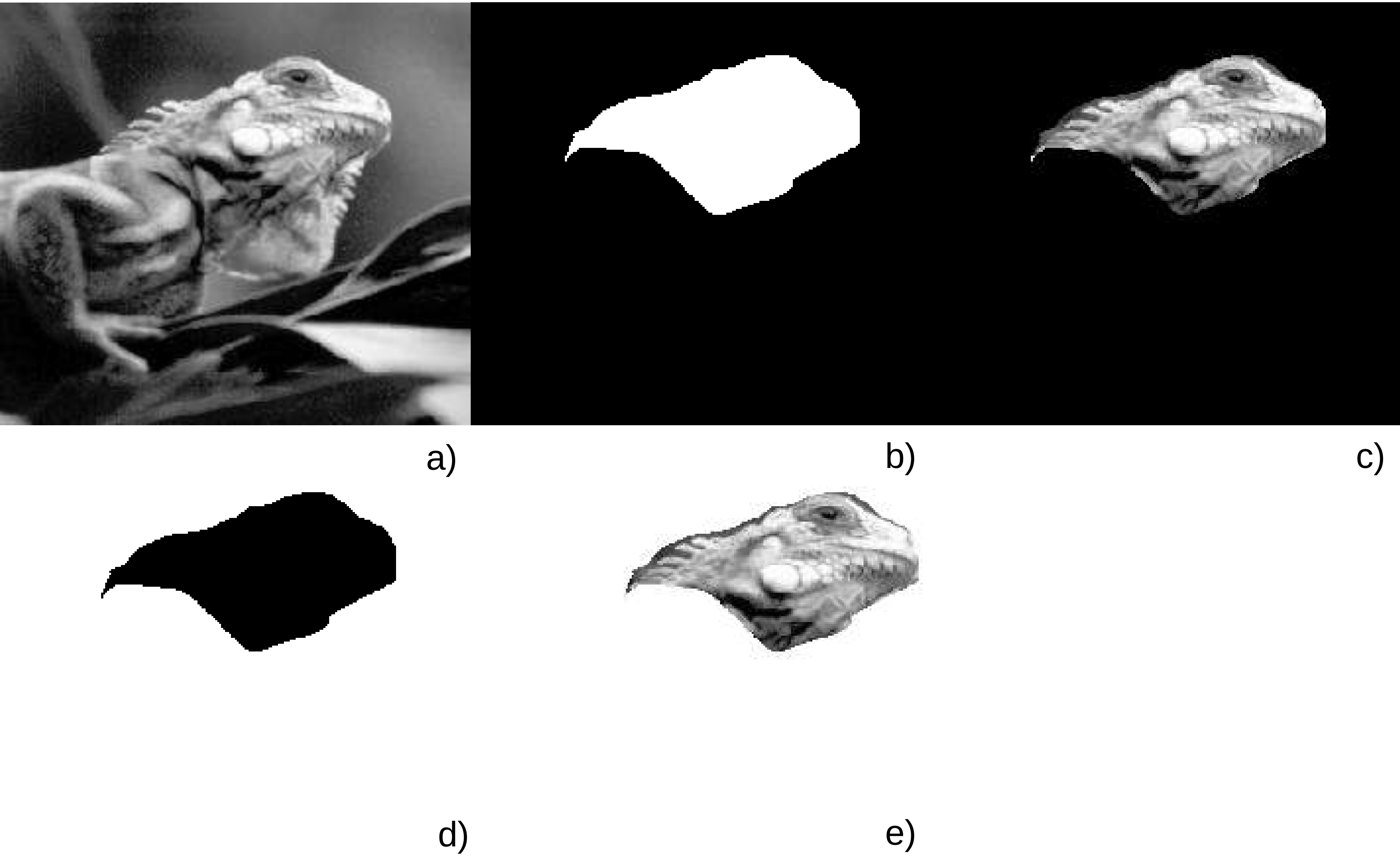


Figure 3.2-11 Complement Image – NOT Operation. a) Original, b) NOT operator applied to the image

a)

b)

Module-2
Digital Image Processing

Image Transformation

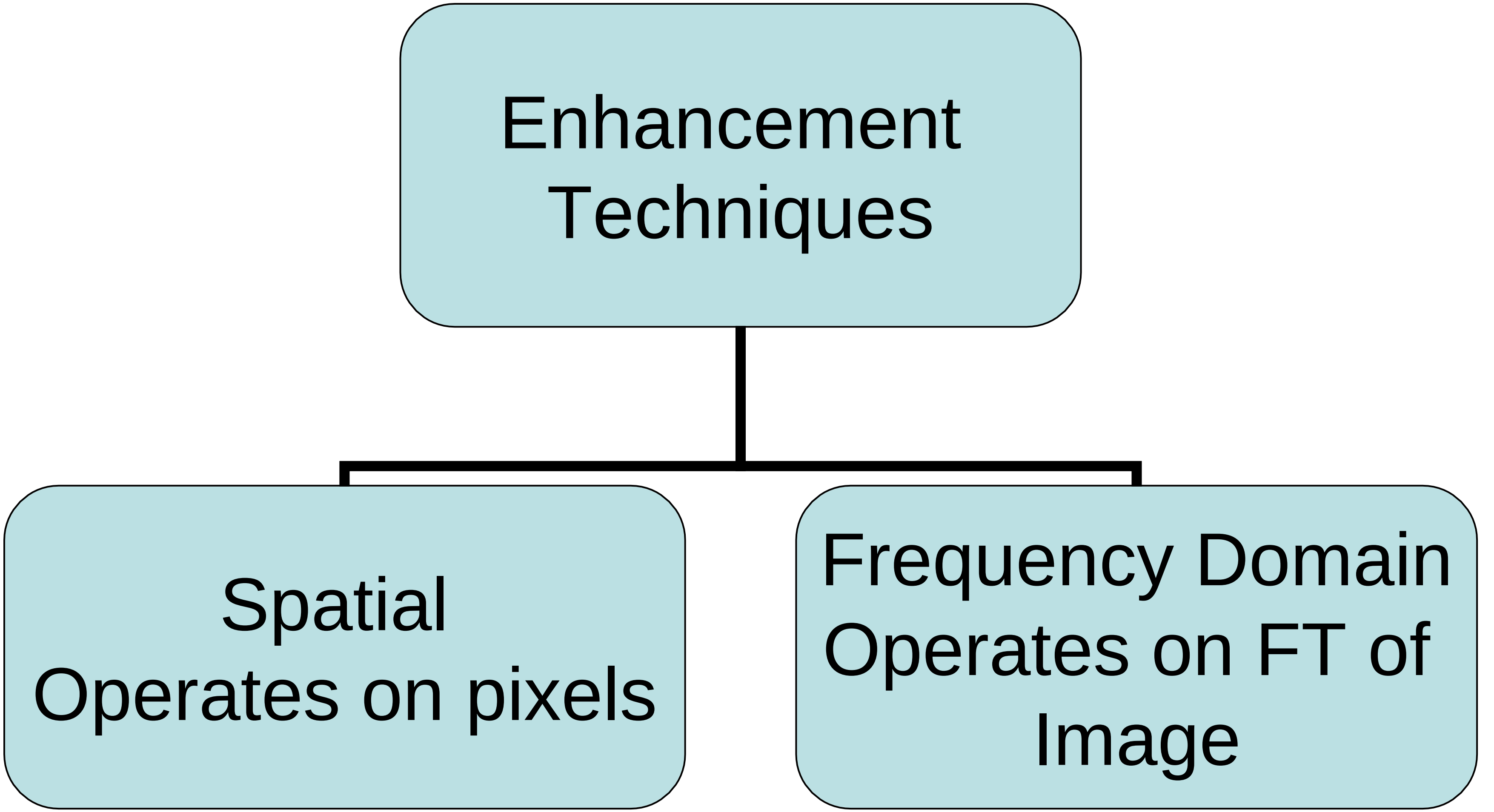

Enhancement Techniques
Spatial
Operates on pixels
Frequency Domain
Operates on FT of Image

Image Enhancement Definition

- **Image Enhancement:** is the process that improves the quality of the image for a specific application

Image Enhancement Methods

- **Spatial Domain Methods (Image Plane)**

  Techniques are based on direct manipulation of pixels in an image

- **Frequency Domain Methods**

  Techniques are based on modifying the Fourier transform of the image.

- **Combination Methods**

  There are some enhancement techniques based on various combinations of methods from the first two categories

Spatial Domain Methods

- As indicated previously, the term *spatial domain* refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

  $g(x,y) = \boldsymbol{T}\,[f(x,y)]$

  Where f(x,y) in the input image, g(x,y) is the processed image and ***T*** is as operator on f, defined over some neighborhood of (x,y)

- In addition, ***T*** can operate on a set of input images.

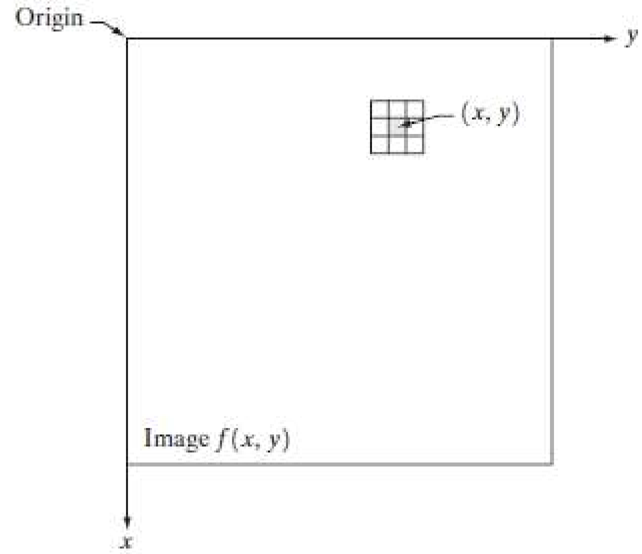


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

- The simplest form of ***T***, is when the neighborhood of size 1X1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

$$s = T\ (r)$$

Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

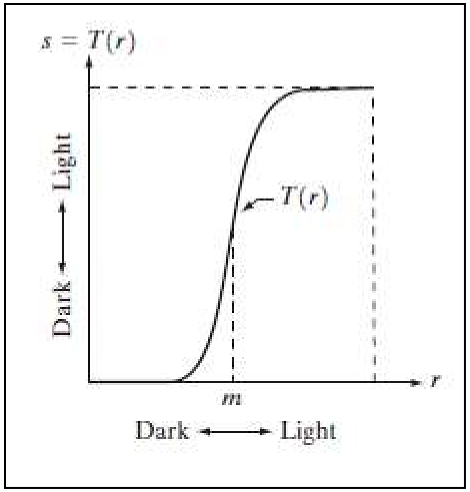
Examples of Enhancement Techniques

- **Contrast Stretching:**

If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

Produce higher contrast than the original image, by:

- Darkening the levels below m in the original image
- Brightening the levels above m in the original image

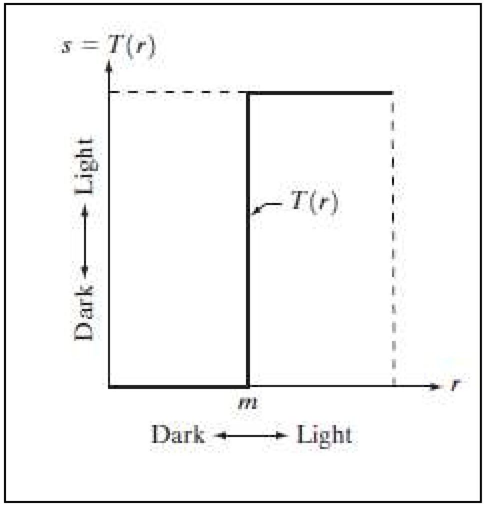


So, Contrast Stretching: is a simple image enhancement technique that improves the contrast in an image by ‘stretching’ the range of intensity values it contains to span a desired range of values. Typically, it uses a linear function

Examples of Enhancement Techniques

- **Thresholding**

  Is a limited case of contrast stretching, it produces a two-level (binary) image.



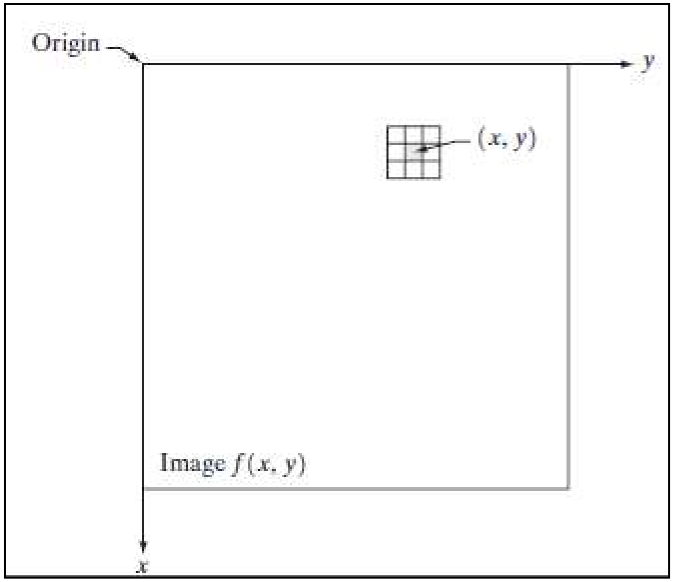
Some fairly simple, yet powerful, processing approaches can be formulated with grey-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as *point processing*.

Examples of Enhancement Techniques

Larger neighborhoods allow considerable more flexibility. The general approach is to use a function of the values of *f* in a predefined neighborhood of (x,y) to determine the value of *g* at (x,y).

One of the principal approaches in this formulation is based on the use of so-called *masks* (also referred to as *filters*)

So, a mask/filter: is a small (say 3X3) 2-D array, such as the one shown in the figure, in which the values of the mask coefficients determine the nature of the process, such as *image sharpening*. Enhancement techniques based on this type of approach often are referred to as *mask processing* or *filtering*.



Some Basic Intensity (Gray-level) Transformation Functions

- Grey-level transformation functions (also called, intensity functions), are considered the simplest of all image enhancement techniques.
- The value of pixels, before an after processing, will be denoted by *r* and *s*, respectively. These values are related by the expression of the form:

$$\mathbf{s = \mathit{T}\,(r)}$$

where T is a transformation that maps a pixel value r into a pixel value s.

Some Basic Intensity (Gray-level) Transformation Functions

Consider the following figure, which shows three basic types of functions used frequently for image enhancement:

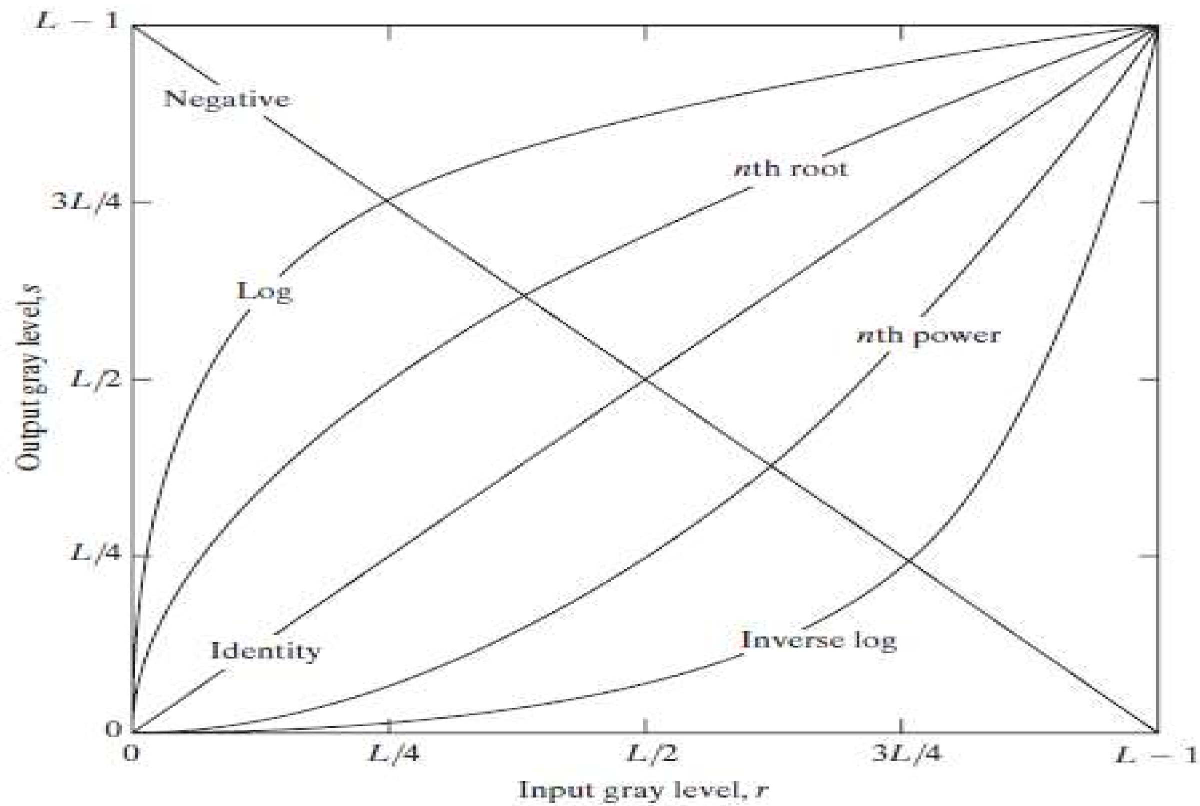


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Some Basic Intensity (Gray-level) Transformation Functions

- The three basic types of functions used frequently for image enhancement:
  - Linear Functions:
    - Negative Transformation
    - Identity Transformation
  - Logarithmic Functions:
    - Log Transformation
    - Inverse-log Transformation
  - Power-Law Functions:
    - n^{th} power transformation
    - n^{th} root transformation

Linear Functions

- **Identity Function**
  - Output intensities are identical to input intensities
  - This function doesn't have an effect on an image, it was included in the graph only for completeness
  - Its expression:

    $s = r$

Linear Functions

- **Image Negatives (Negative Transformation)**
  - The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

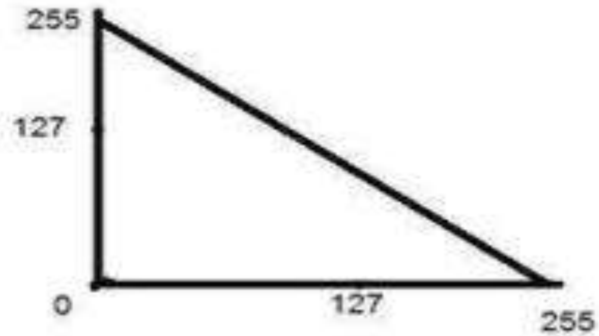
$$s = L - 1 - r$$

  Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative.

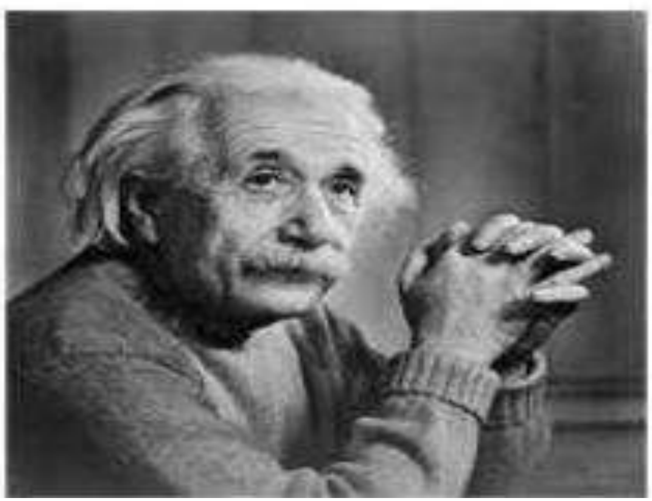
  - The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

NEGATIVE TRANSFORMATION EXAMPLE

Graph representation

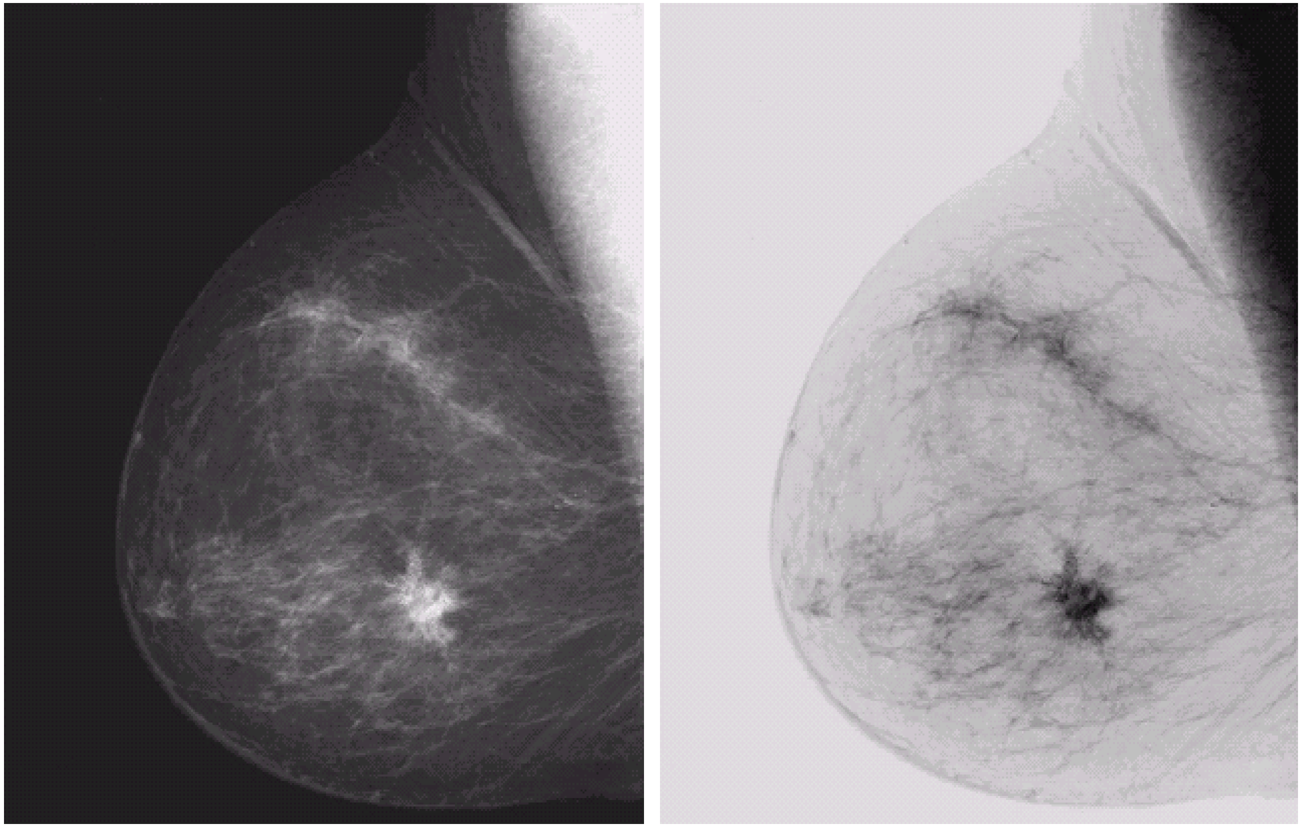


Input image



Output image

Image Negative



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.)

Image Negative: $s = L - 1 - r$

Logarithmic Transformations

- **Log Transformation**

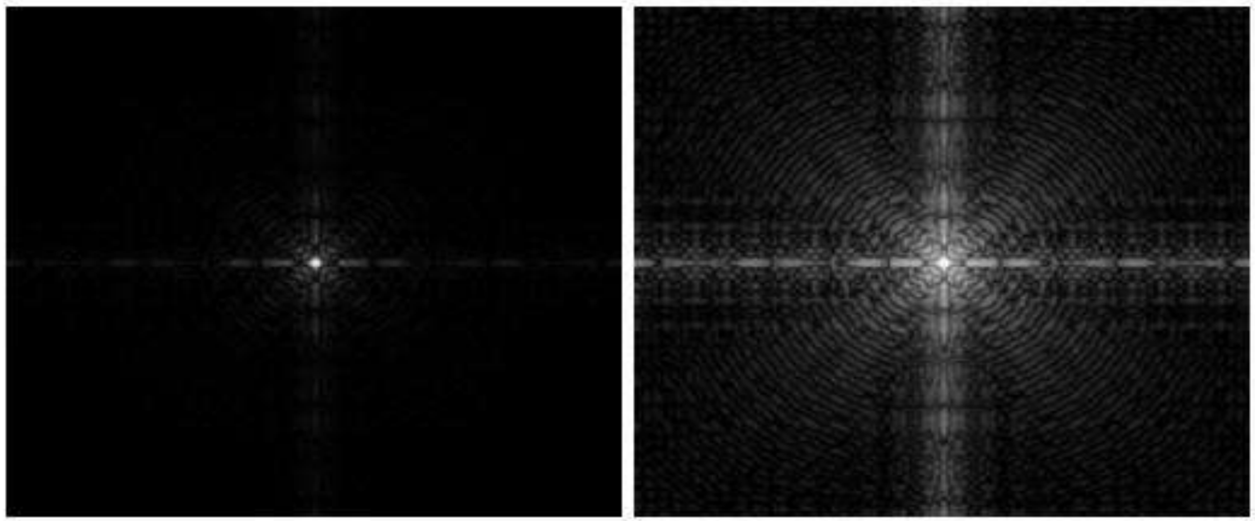
  The general form of the log transformation:

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

  Where c is a constant, and $r \geq 0$

  - Log curve maps a narrow range of low gray-level values in the input image into a wider range of the output levels.
  - Used to expand the values of dark pixels in an image while compressing the higher-level values.
  - It compresses the dynamic range of images with large variations in pixel values.

Logarithmic Transformations



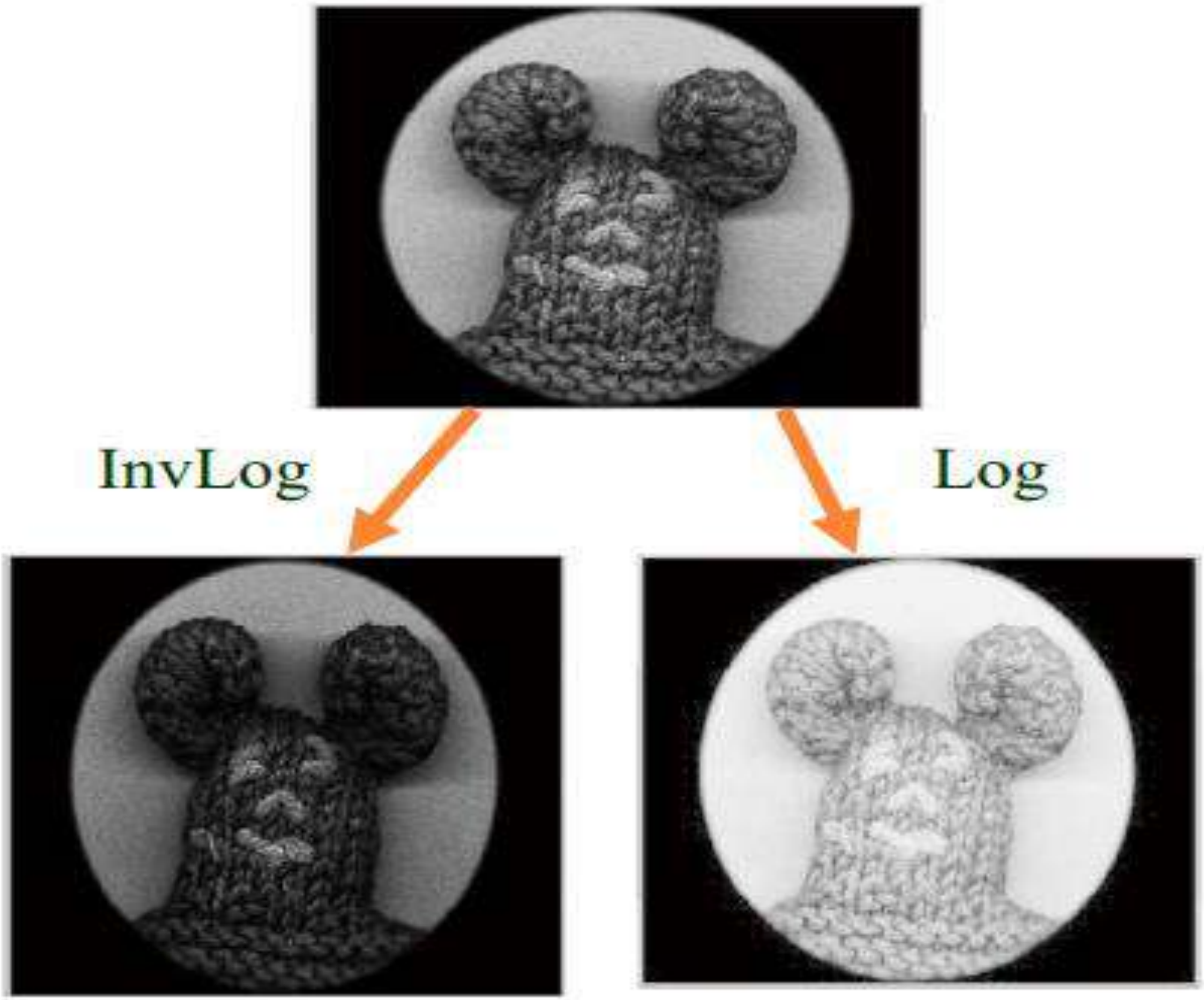
a b

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

Logarithmic Transformations

- **Inverse Logarithm Transformation**
  - Do opposite to the log transformations
  - Used to expand the values of high pixels in an image while compressing the darker-level values.

Log transformation example



Power-Law Transformations

- Power-law transformations have the basic form of:

  $$s = c.r^{\gamma}$$

  Where c and γ are positive constants

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)

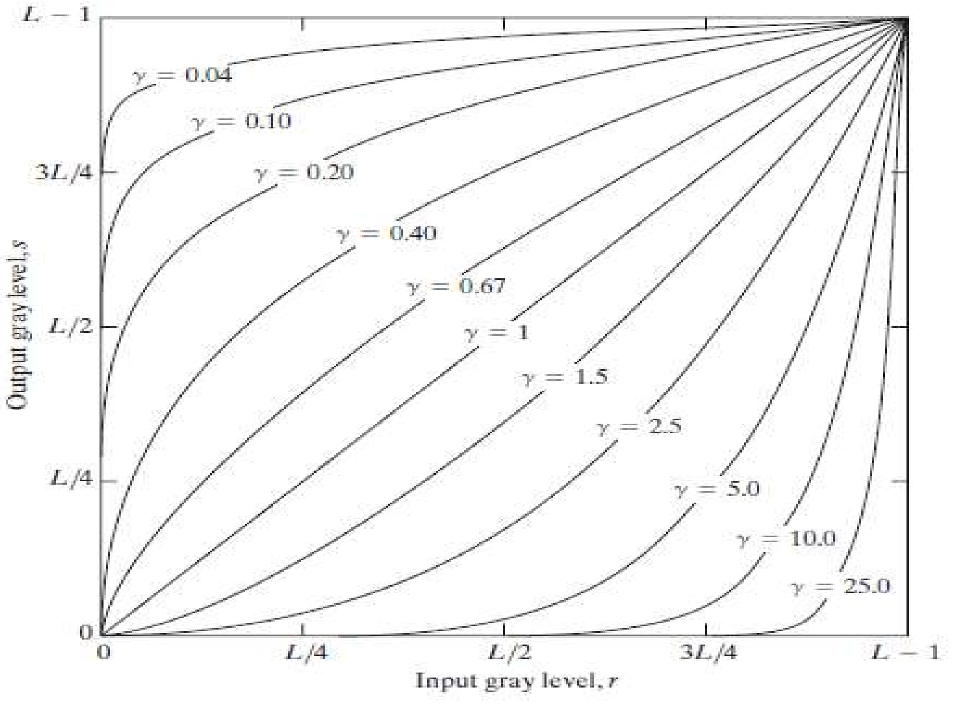


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

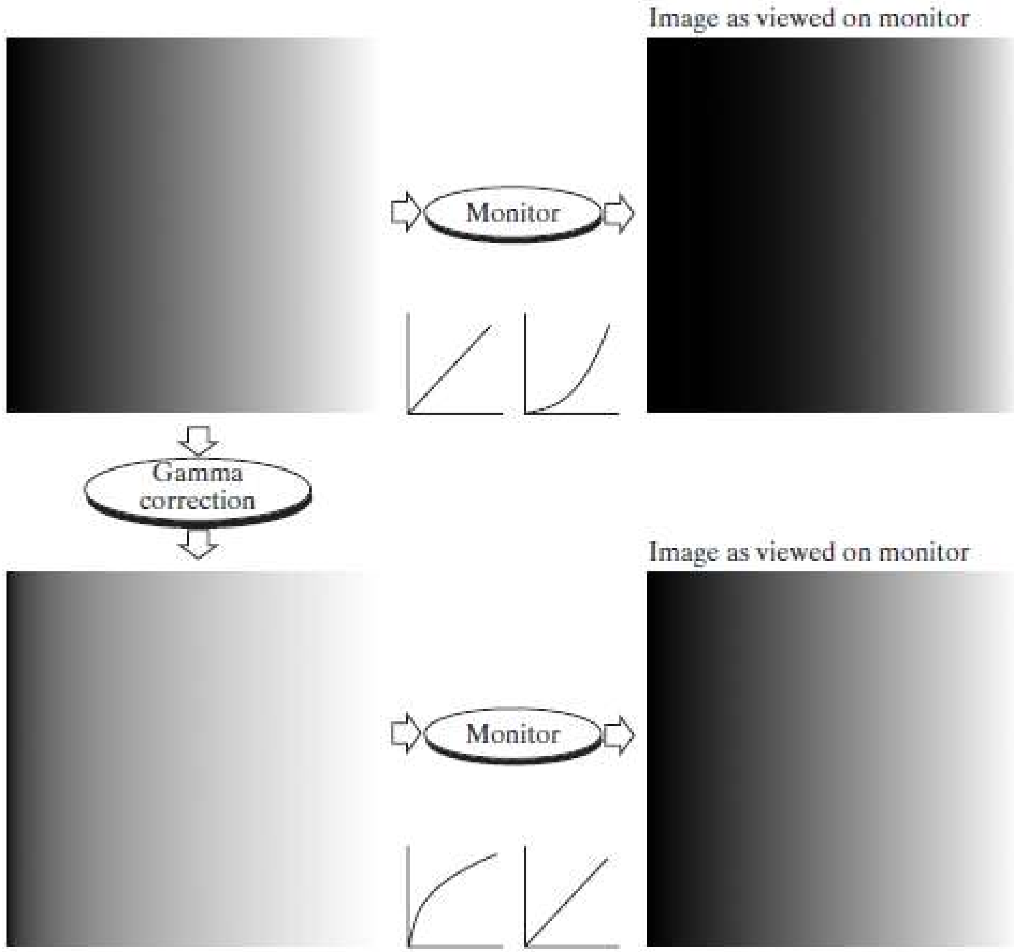
Power-Law Transformations

- Variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power-law response phenomena is called ***gamma correction****.*

  For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple gray-scale linear wedge input into a CRT monitor. As expected, the output of the monitor appears darker than the input, as shown in Fig. 3.7(b). Gamma correction

  in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation. The result is shown in Fig. 3.7(c).When input into the same monitor, this gamma-corrected input produces an output that is close in appearance to the original image, as shown in Fig. 3.7(d).

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.

Power-Law Transformation

- In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation. See figure 3.8

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.)

Power-Law Transformation

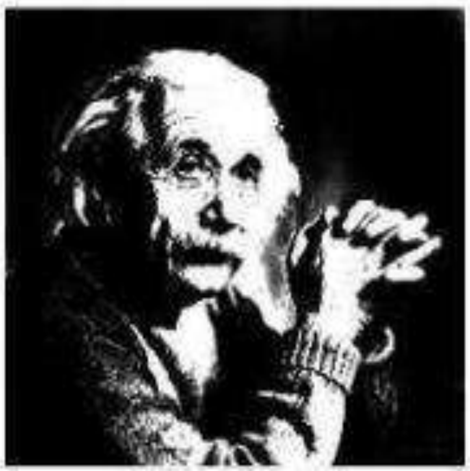
- Another illustration of Power-law transformation

a b
c d

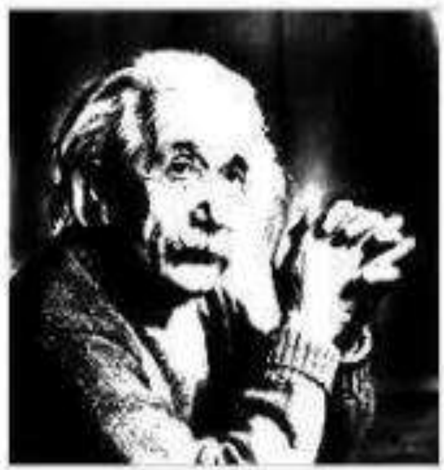
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

POWER LAW TRANSFORMATION EXAMPLE

Gamma=10



Gamma=8



Gamma=6

Piecewise-Linear Transformation Functions

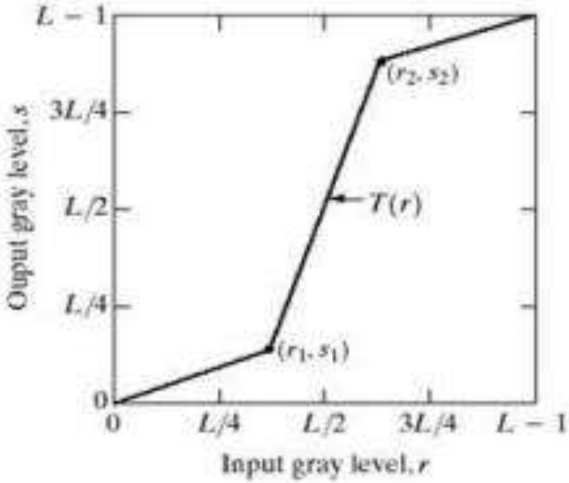
- **Principle Advantage:** Some important transformations can be formulated only as a piecewise function.
- **Principle Disadvantage:** Their specification requires more user input that previous transformations
- **Types of Piecewise transformations are:**
  - Contrast Stretching
  - Gray-level Slicing
  - Bit-plane slicing

Contrast Stretching

- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

CONTRAST STRETCHING EXAMPLE

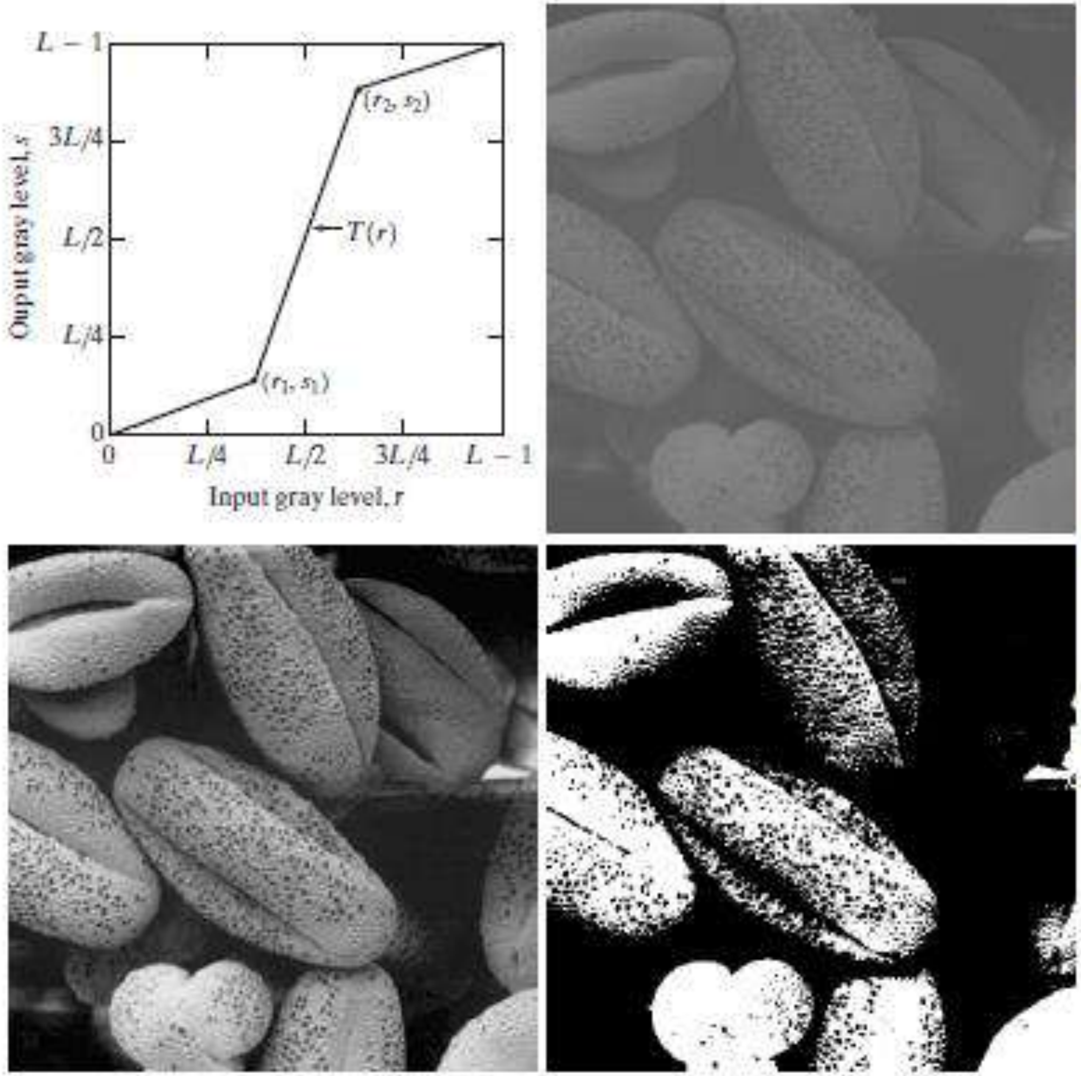
Transformation function



Low contrast image

Contrast stretching image

Contrast Stretching



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast Stretching

- Figure 3.10(a) shows a typical transformation used for contrast stretching. The locations of points (r_1, s_1) and (r_2, s_2) control the shape of the transformation function.
- If $r_1 = s_1$ and $r_2 = s_2$, the transformation is a linear function that produces no changes in gray levels.
- If $r_1 = r_2$, $s_1 = 0$ and $s_2 = L-1$, the transformation becomes a *thresholding function* that creates a binary image. As shown previously in slide 7.
- Intermediate values of (r_1, s_1) and (r_2, s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, $r_1 \leq r_2$ and $s_1 \leq s_2$ is assumed, so the function is always increasing.

Contrast Stretching

- Figure 3.10(b) shows an 8-bit image with low contrast.

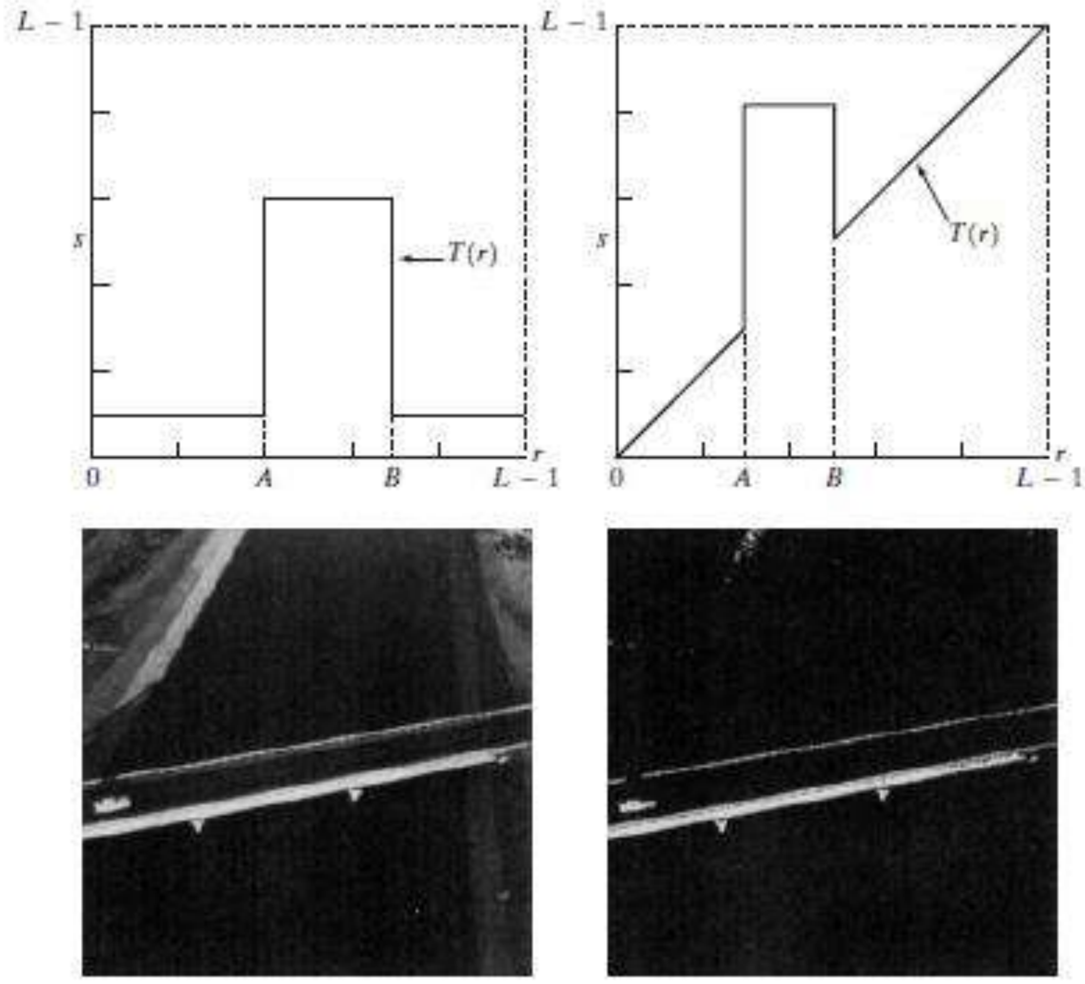
- Fig. 3.10(c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

- Finally, Fig. 3.10(d) shows the result of using the *thresholding function* defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig 3.11 (a), produces a binary image.
  - The second approach, based on the transformation shown in Fig 3.11 (b), brightens the desired range of gray levels but preserves gray levels unchanged.
  - Fig 3.11 (c) shows a gray scale image, and fig 3.11 (d) shows the result of using the transformation in Fig 3.11 (a).

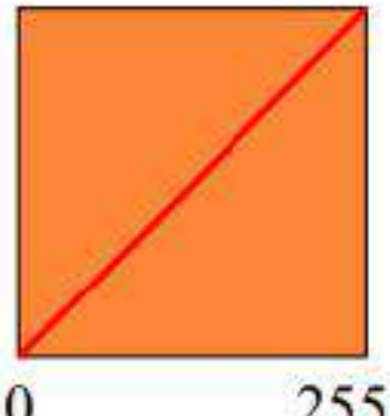
Gray-level Slicing



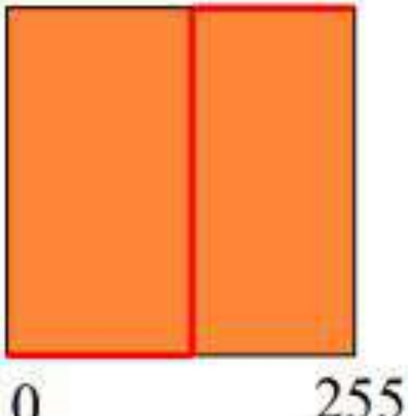
a b
c d

FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Input image



Output image



Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

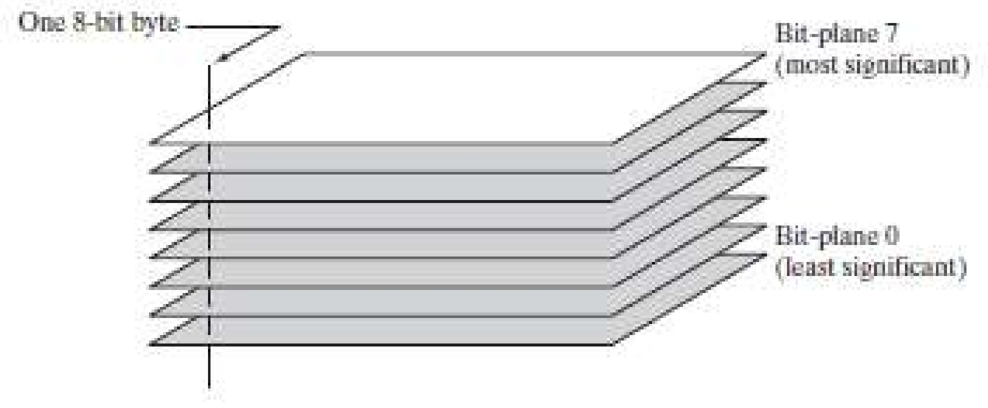


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

BIT PLANE SLICING EXAMPLE

Original image

Bit plane 7

Bit plane 6

Bit plane 4

Bit plane 1

1. Which of the following expression is used to denote spatial domain process?

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

b) $f(x+y)=T[g(x+y)]$

c) $g(xy)=T[f(xy)]$

d) $g(x-y)=T[f(x-y)]$

1. Which of the following expression is used to denote spatial domain process?

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

b) $f(x+y)=T[g(x+y)]$

c) $g(xy)=T[f(xy)]$

d) $g(x-y)=T[f(x-y)]$

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $\mathbf{s=c\log_{10}(1+r)}$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

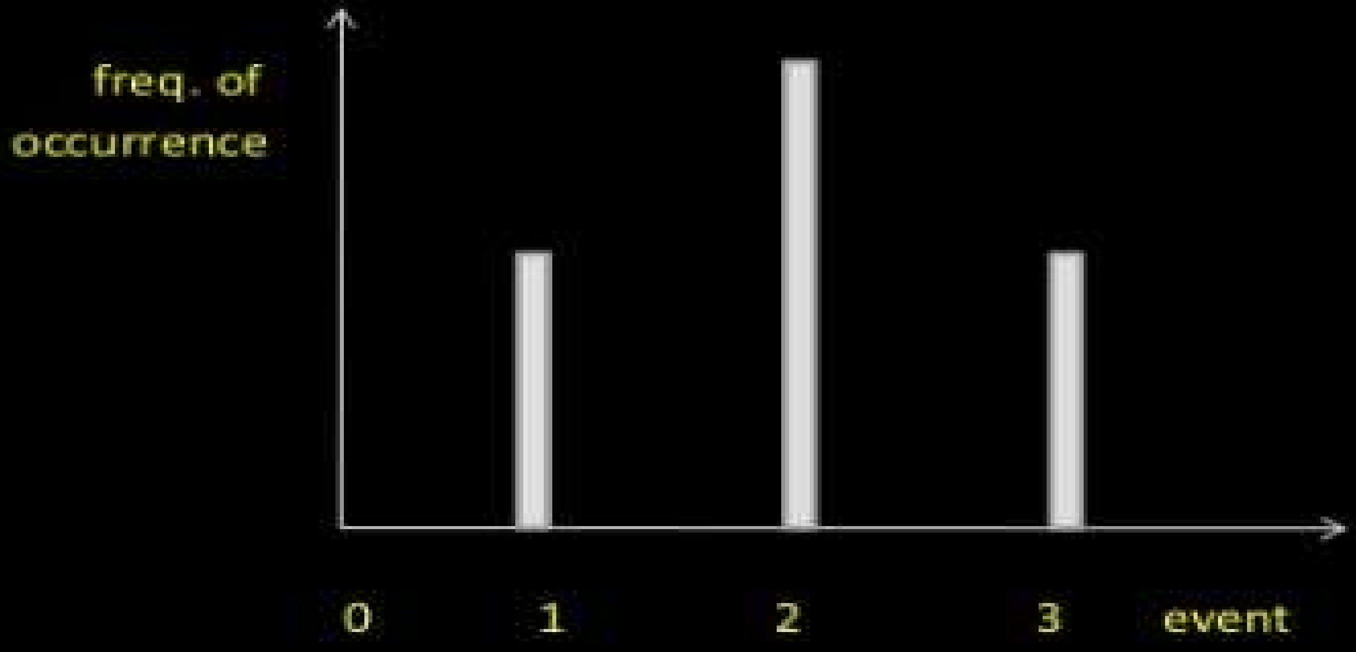
d) $s=rc^{\gamma}$

Histogram Processing

Histogram Processing

Histogram:

It is a plot of frequency of occurrence of an event.



Histogram Processing

- Histogram of images provide a global description of their appearance.
- Enormous information is obtained.
- It is a spatial domain technique.
- Histogram of an image represents relative frequency of occurrence of various gray levels.
- Histogram can be plotted in two ways:

Histogram Processing

- First Method:

- X-axis has gray levels & Y-axis has No. of pixels in each gray levels.

Gray Level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2

Consider a 5x5 image with integer intensities in the range between zero and seven:

Image

$$\begin{matrix} 0 & 7 & 3 & 2 & 3 \\ 0 & 0 & 0 & 6 & 7 \\ 7 & 7 & 2 & 2 & 0 \\ 1 & 1 & 0 & 4 & 1 \\ 0 & 0 & 7 & 4 & 1 \end{matrix}$$

Image matrix

Gray scale

Consider a 5x5 image with integer intensities in the range between one and Seven

Image

$$
\begin{matrix}
0 & 7 & 3 & 2 & 3 \\
0 & 0 & 0 & 6 & 7 \\
7 & 7 & 2 & 2 & 0 \\
1 & 1 & 0 & 4 & 1 \\
0 & 0 & 7 & 4 & 1
\end{matrix}
$$

Image matrix

Black

White

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	**1**	0	4	**1**
0	0	7	4	**1**

Image matrix

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8
Similarly for **1** h(r1) = 4

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

Image matrix

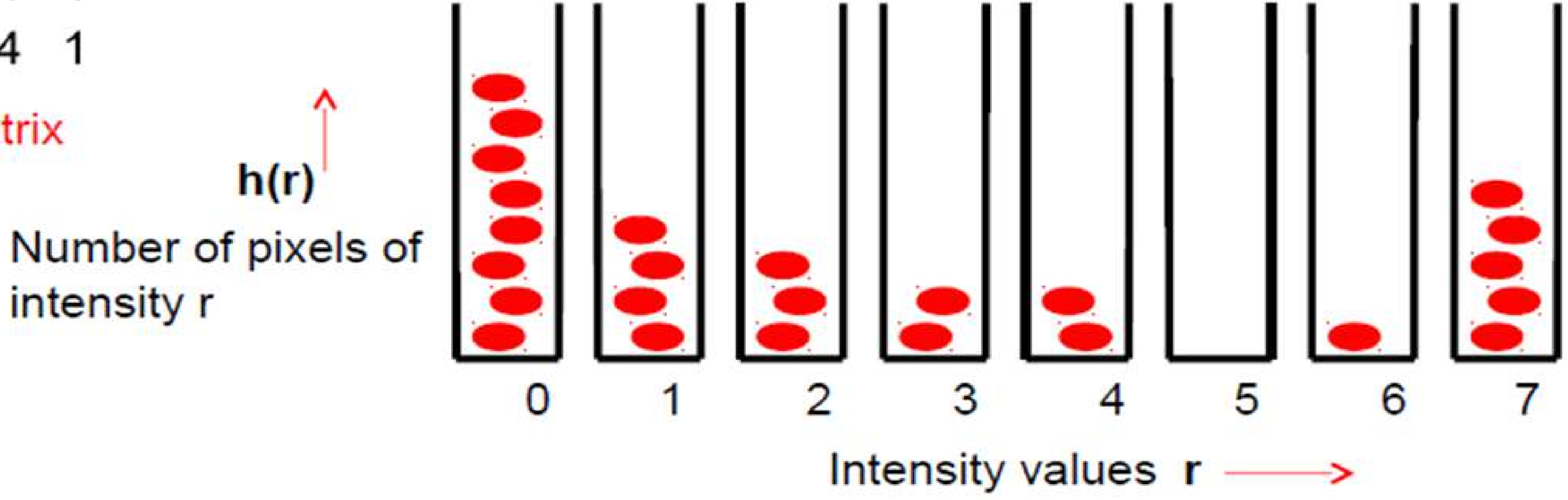
Similarly

INTENSITY r	0	1	2	3	4	5	6	7
NUMBER of pixels of r h(r)	h(r0)=**8**	h(r1)=**4**	h(r2)=**3**	h(r3)=**2**	h(r4)=**2**	h(r5)=**0**	h(r6)=**1**	h(r7)=**5**

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

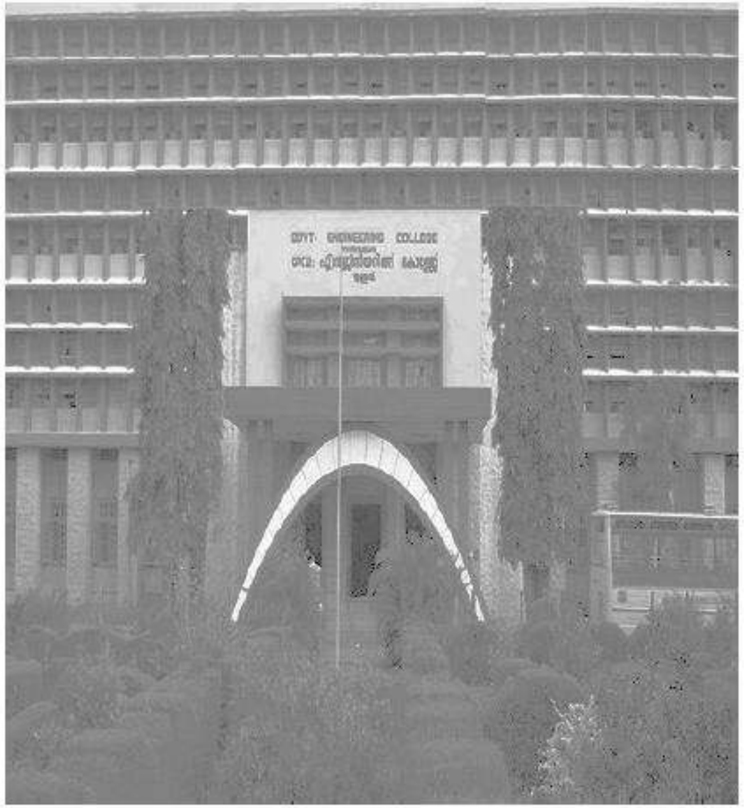
Image matrix

r	0	1	2	3	4	5	6	7
h(r)	8	4	3	2	2	0	1	5



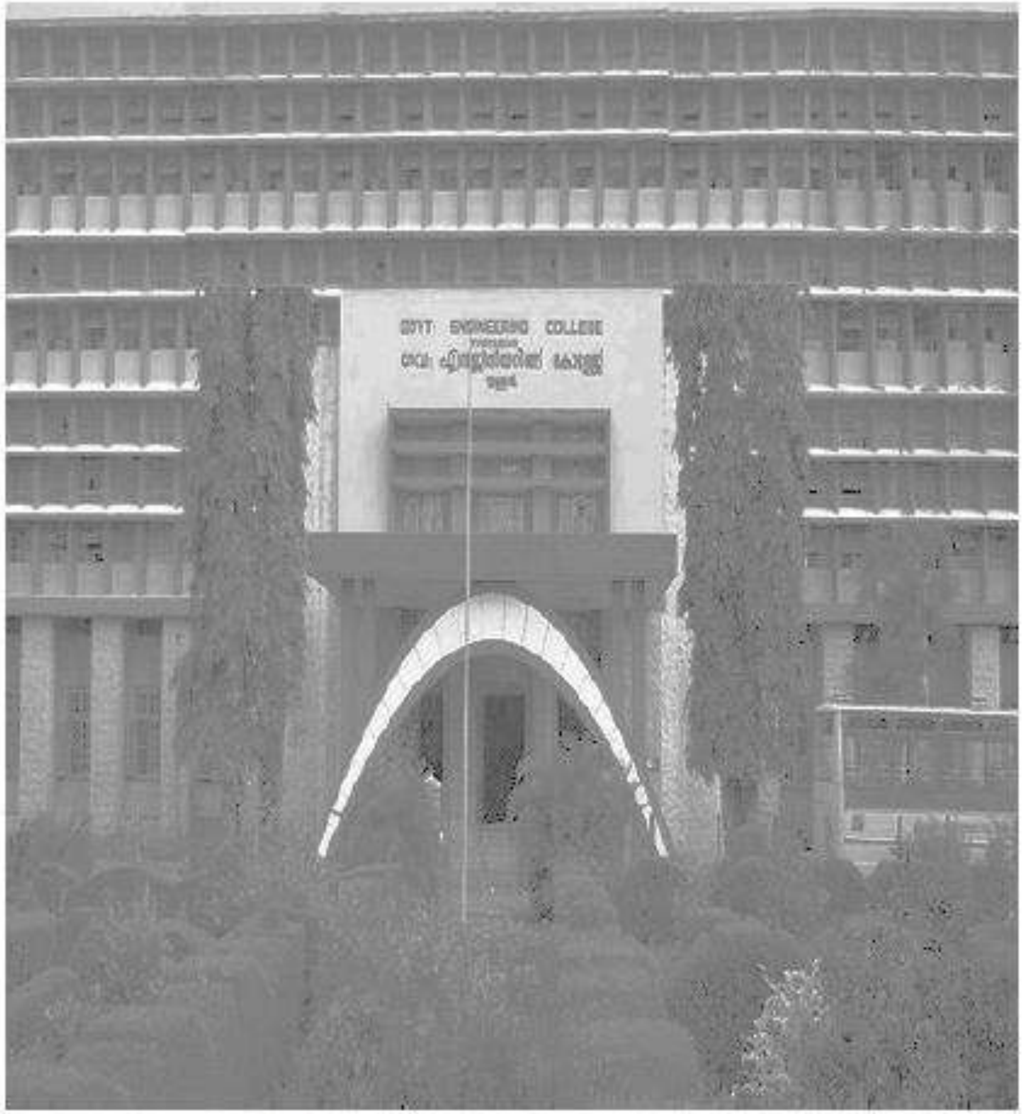
HISTOGRAM

SAMPLE IMAGES AND ITS HISTOGRAM

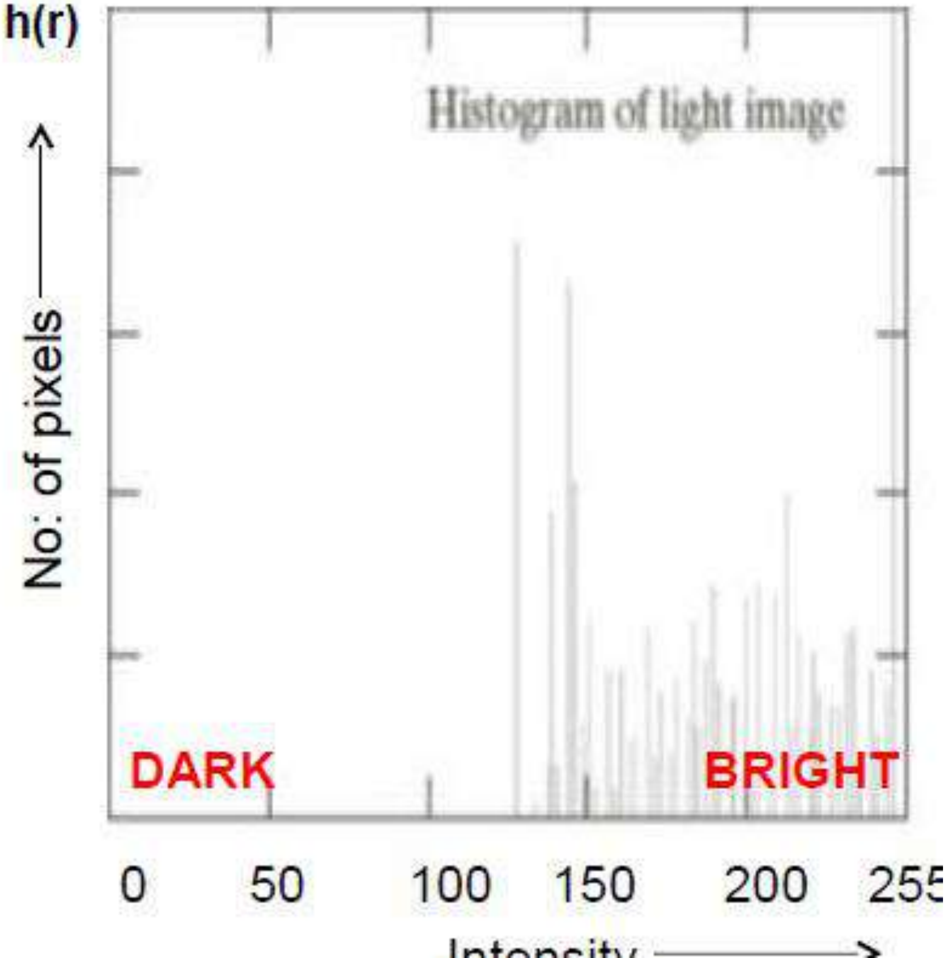


Bright image
Intensity range 0 - 255

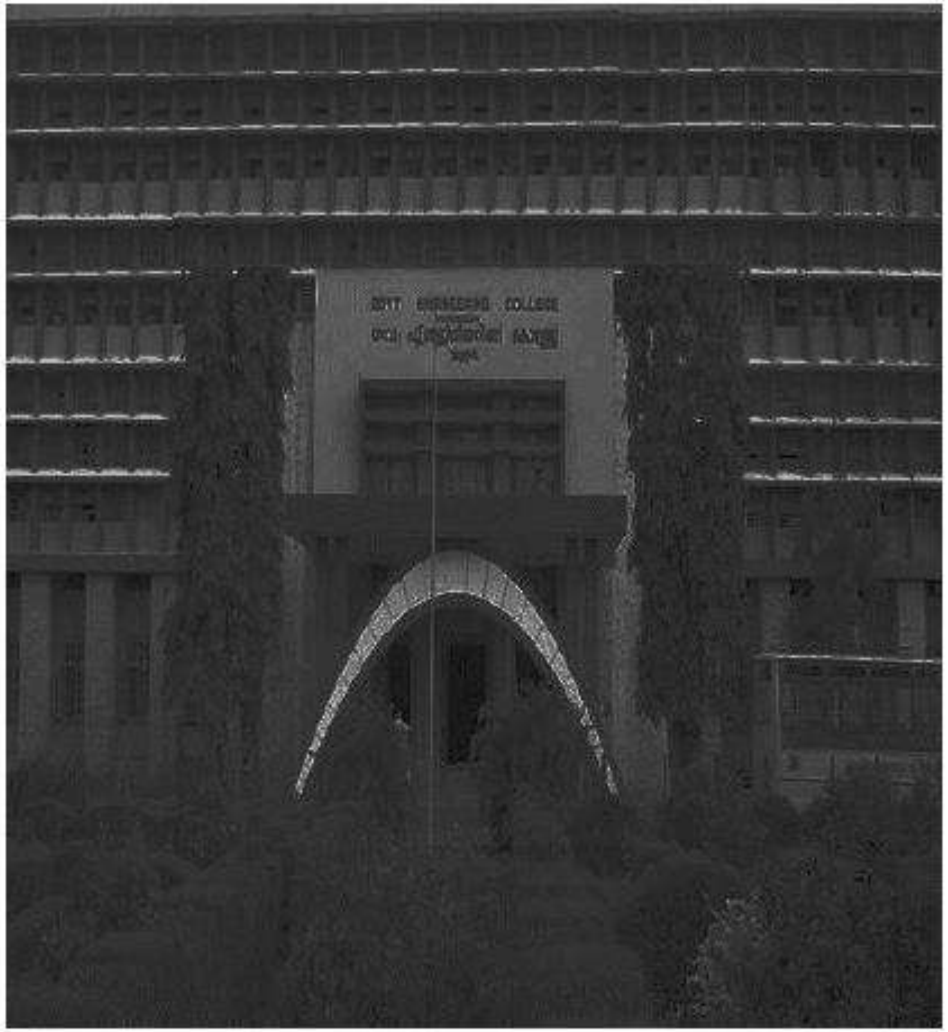
SAMPLE IMAGES AND ITS HISTOGRAM



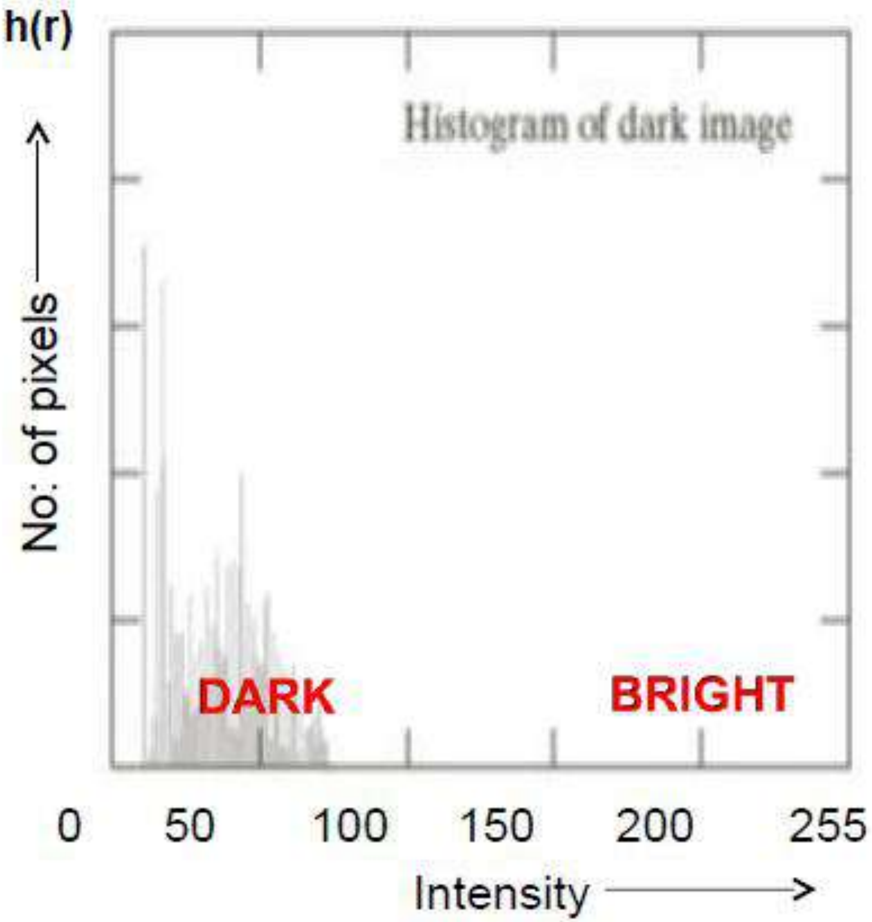
Bright image



SAMPLE IMAGES AND ITS HISTOGRAM

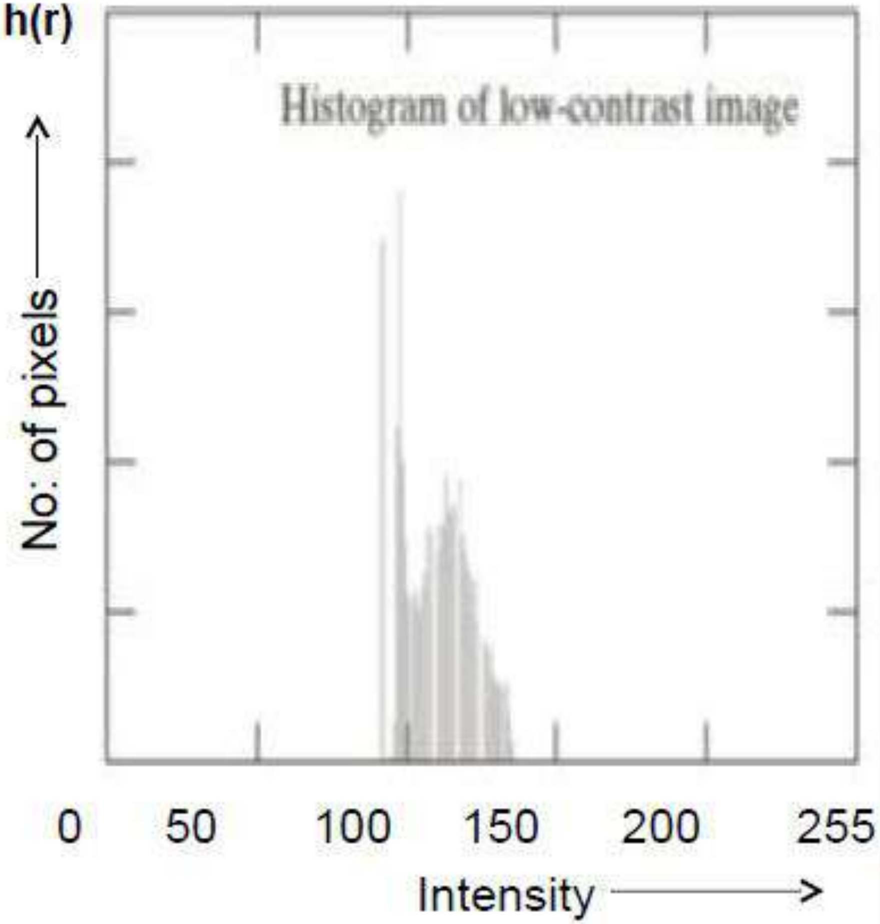


Dark image
Intensity range 0 - 255

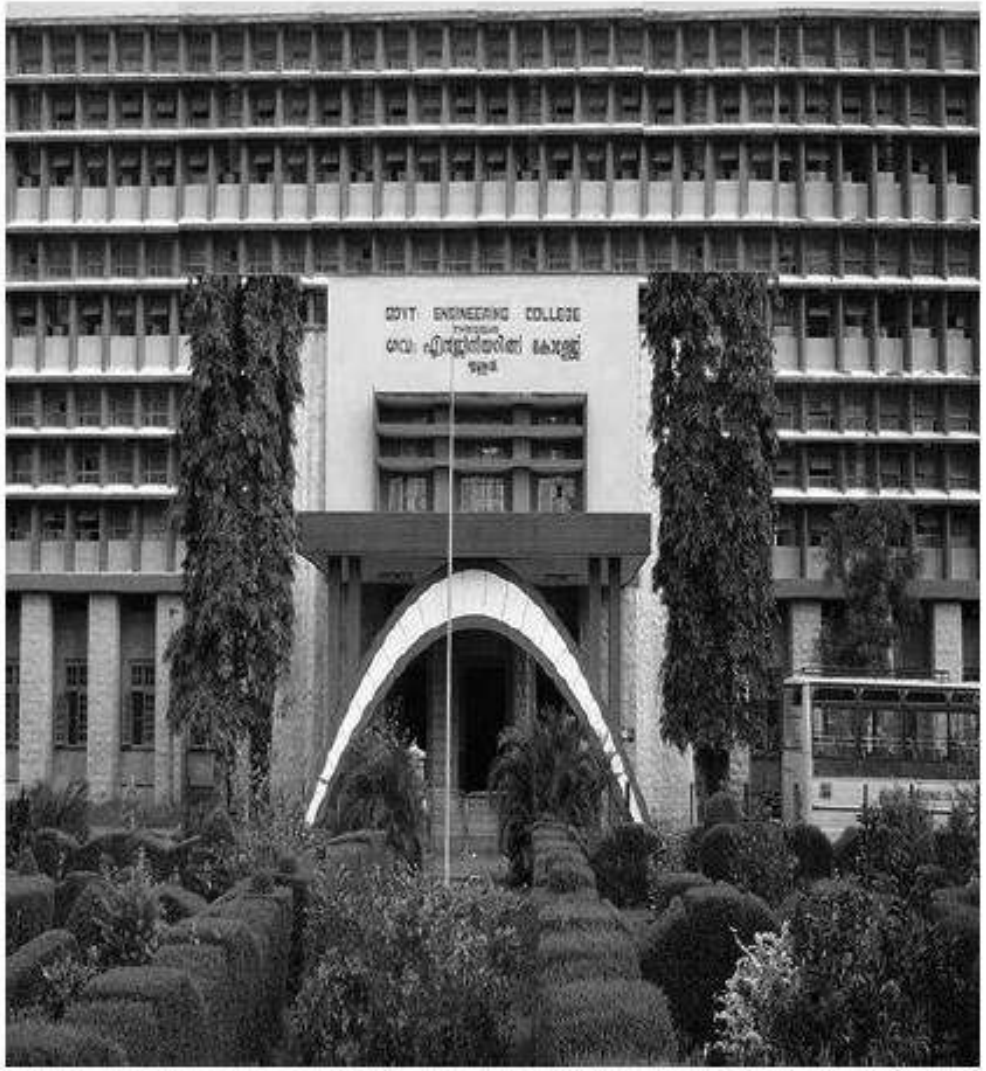


SAMPLE IMAGES AND ITS HISTOGRAM

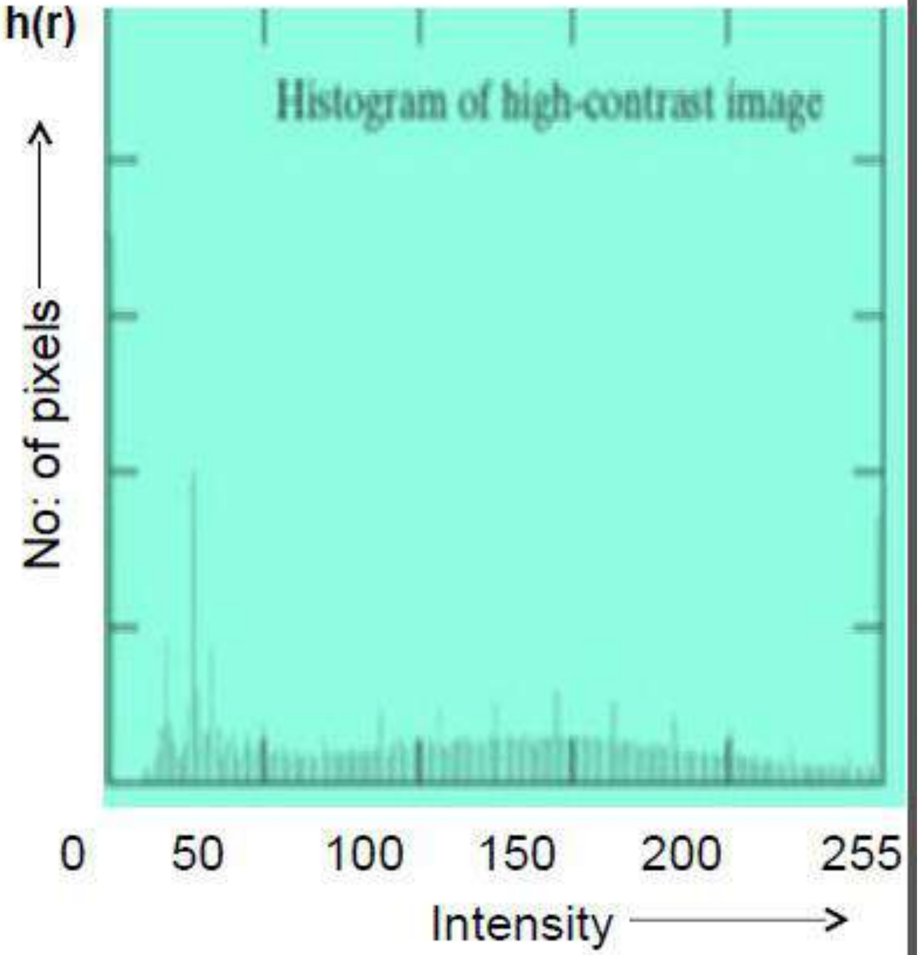
Light image
Intensity range 0 - 255



SAMPLE IMAGES AND ITS HISTOGRAM



High contrast image
Intensity range 0 - 255



Histogram Processing

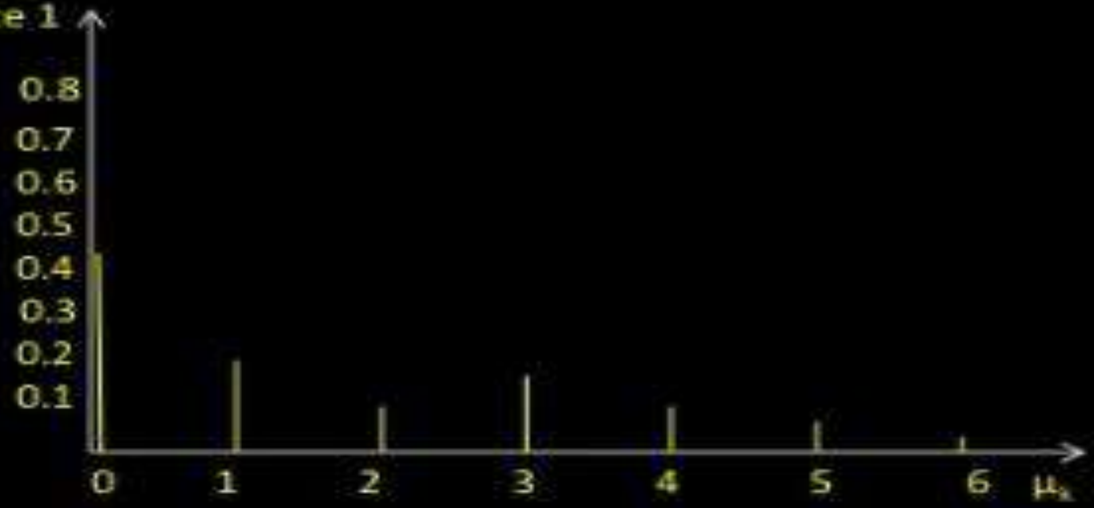
- **Second Method:**
- X-axis has gray levels & Y-axis has probability of occurrence of gray levels.

$$P(\mu_k) = n_k / n; \quad \text{where, } \mu_k - \text{gray level}$$

n_k – no. of pixels in k^{th} gray level

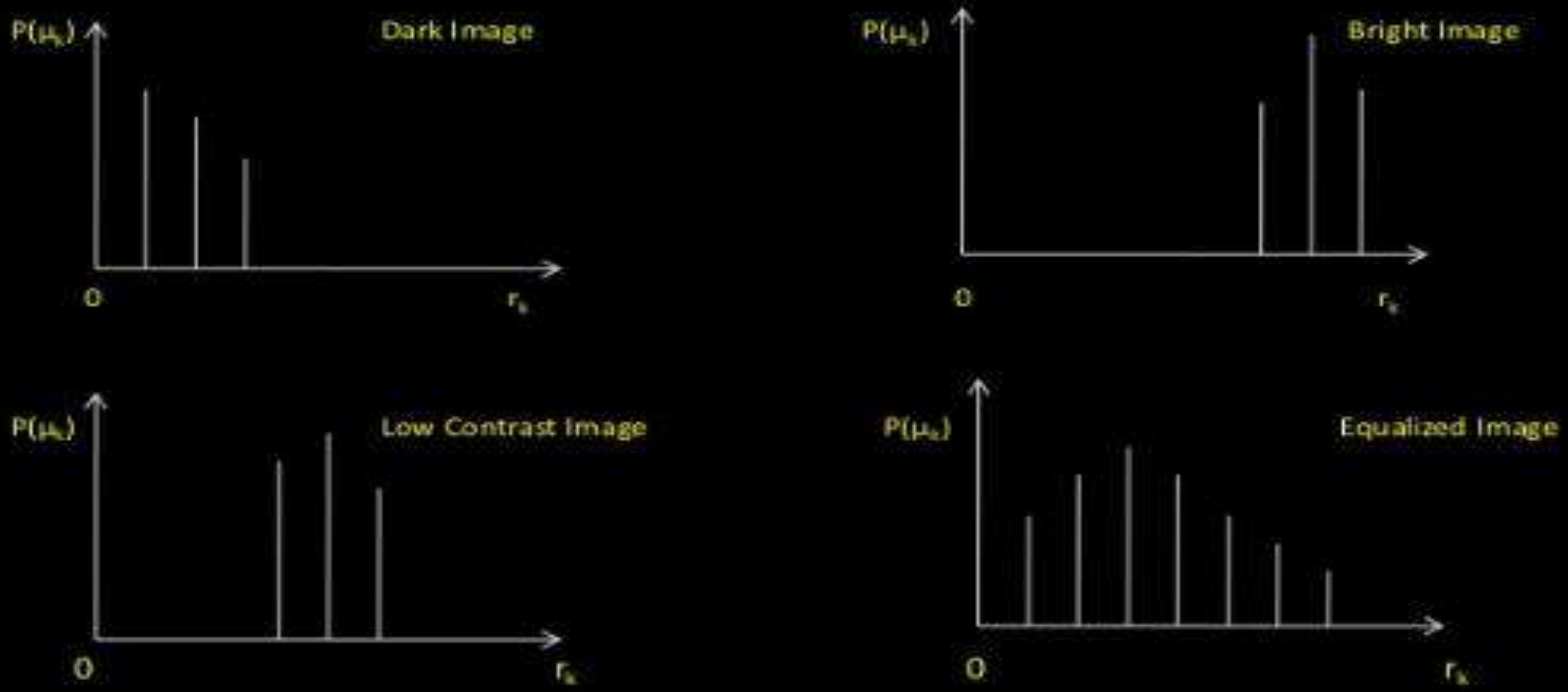
n – total number of pixels in an image

Gray Level	No. of Pixels (n_k)	$P(\mu_k)$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
	n = 100	1



Histogram Processing

- Advantage of 2^{nd} method: Maximum value plotted will always be 1.
- White – 1, Black – 0.
- Great deal of information can be obtained just by looking at histogram.
- Types of Histograms:



Histogram Processing

- The last graph represent the best image.
- It is a high contrast image.
- Our aim would be to transform the first 3 histograms into the 4^{th} type.
- In other words we try to increase the dynamic range of the image.

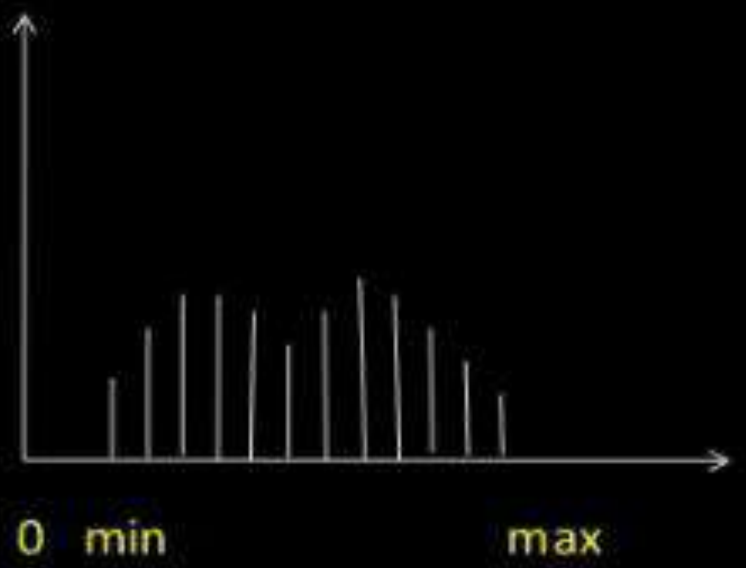
Histogram Stretching

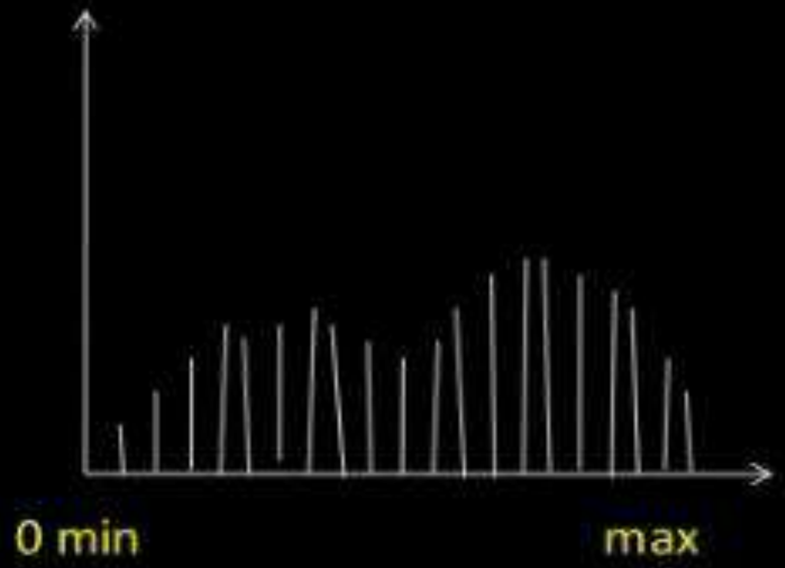
1) <u>Linear stretching:</u>

- Here, we don't alter the basic shape.
- We use basic equation of a straight line having a slope.

$$(S_{max} - S_{min})/(r_{max} - r_{min})$$

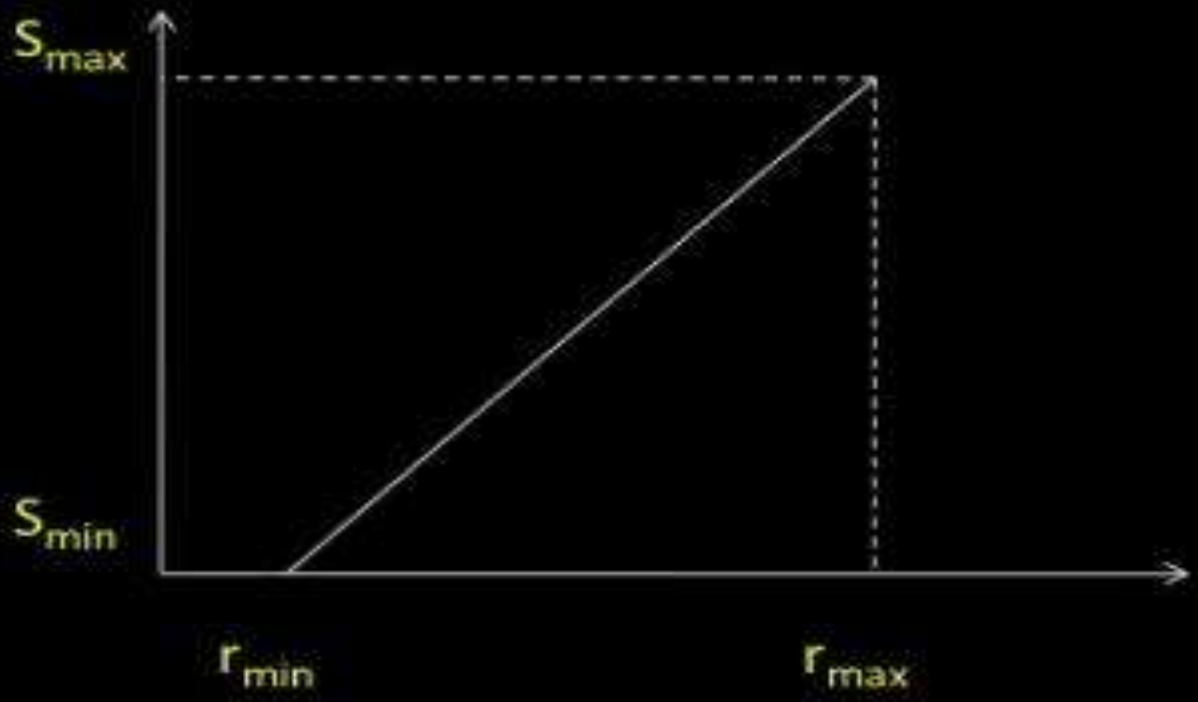
Where, Smax – max gray level of output image
Smin – min gray level of output image
rmax – max gray level of input image
rmin – min gray level of input image





Histogram Processing

$$S = T(r) = ((S_{max} - S_{min})/(r_{max} - r_{min}))(r - r_{min}) + Smin$$



Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Sol^n:- $r_{min} = 2$; $r_{max} = 6$; $s_{min} = 0$; $s_{max} = 7$;

$$slope = ((s_{max} - s_{min})/(r_{max} - r_{min})) = ((7 - 0)/(6 - 2)) = 7 / 4 = 1.75.$$

$S = (7 / 4)(r - 2) + 0;$

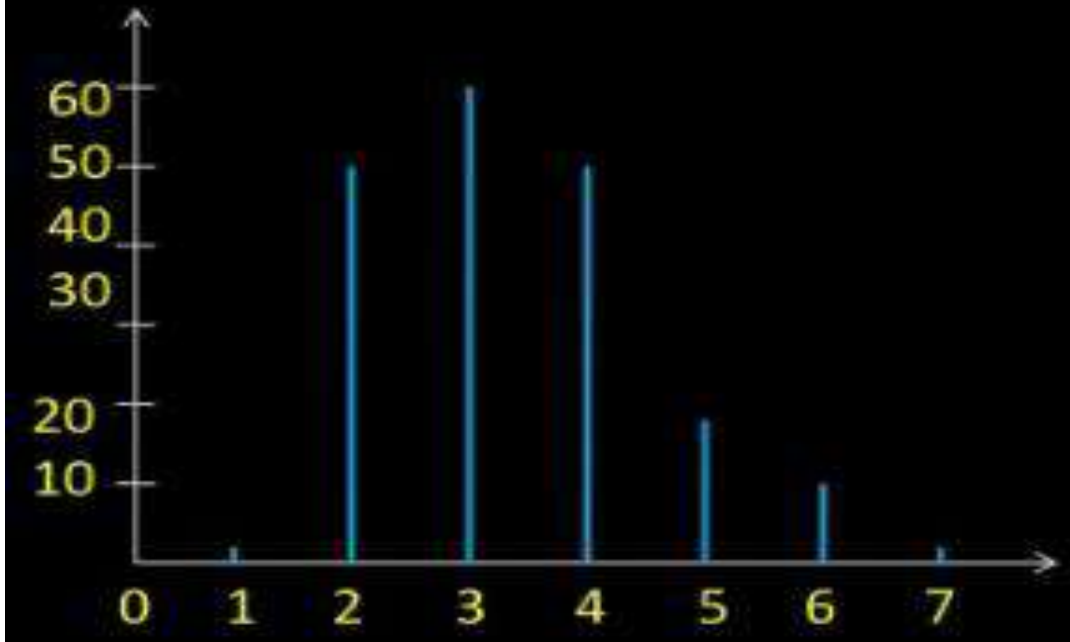
$S = (7 / 4)(r - 2)$

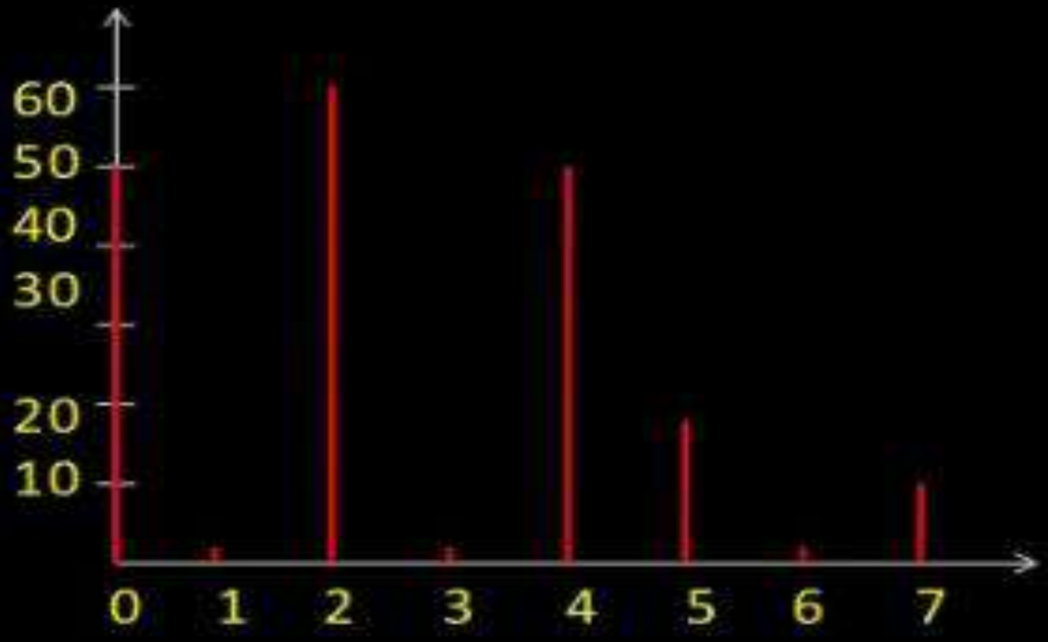
r	(7 / 4)(r - 2) = S
2	0 = 0
3	7/4 = 1.75 = 2
4	7/2 = 3.5 = 4
5	21/4 = 5.25 = 5
6	7 = 7

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	50	0	60	0	50	20	0	10





Ex. 2) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 .

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	100	90	85	70	0	0	0	0

- Rmin=0, rmax=3, smin=0, smax=7
- Slope= (7/3)=2.3
- R s
- 0 0
- 1 2
- 2 5
- 3 7

Ex. 2) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 .

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	100	0	90	0	0	85	0	70

Dynamic range increases

BUT

No. of pixels at each gray level remains constant.

Histogram Equalization

Histogram Equalization

- What is the histogram equalization?

- The histogram equalization is an approach to enhance a given image. The approach is to design a transformation $T(.)$ such that the gray values in the output is uniformly distributed in [0, 1].

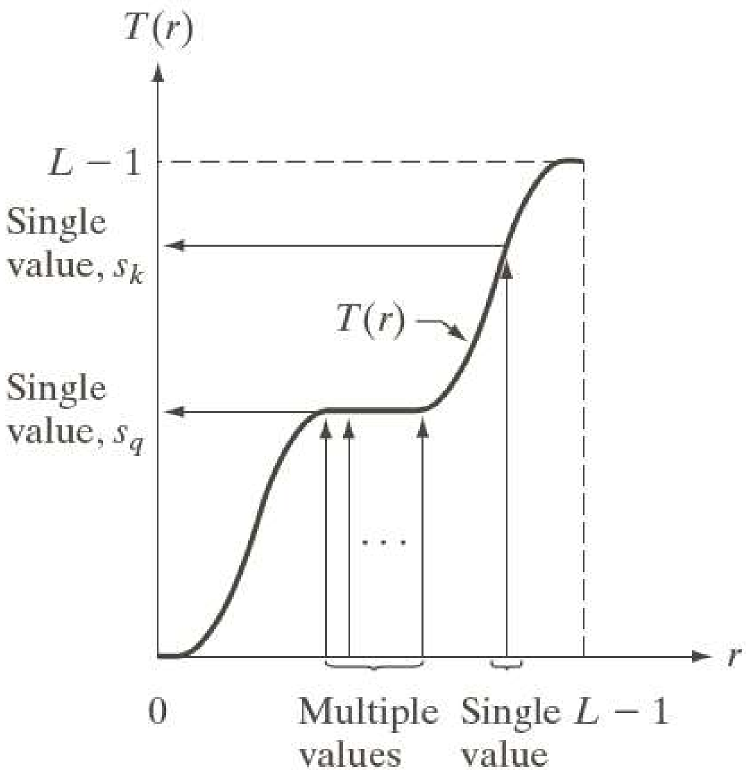
- Let us assume for the moment that the input image to be enhanced has continuous gray values, with $r = 0$ representing black and $r = 1$ representing white.

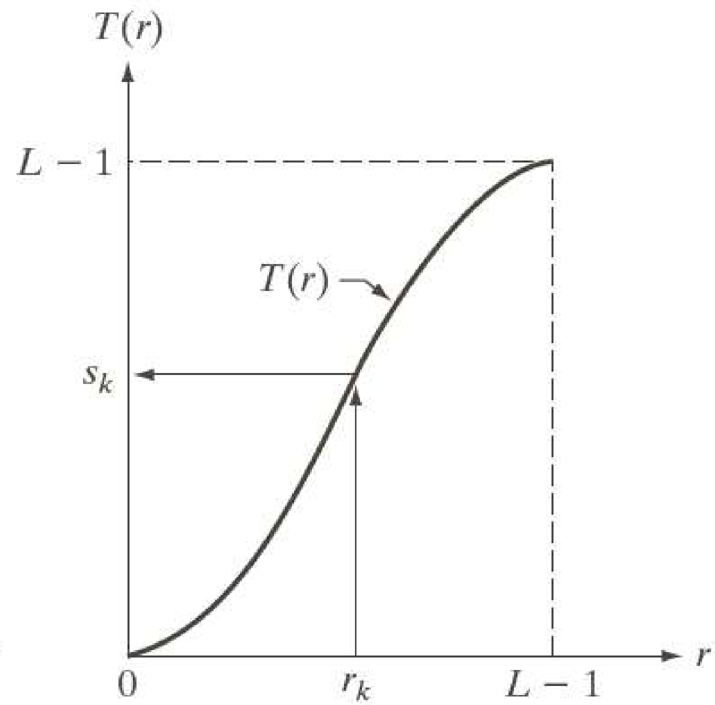
- We need to design a gray value transformation $s = T(r)$, based on the histogram of the input image, which will enhance the image.

- As before, we assume that:

(1) *T(r)* is a monotonically increasing function for $0 \le r \le 1$(preserves order from black to white).

(2) *T(r)* maps [0,1] into [0,1] (preserves the range of allowed Gray values).





- Let us denote the inverse transformation by $r = T^{-1}(s)$. We assume that the inverse transformation also satisfies the above two conditions.

- We consider the gray values in the input image and output image as random variables in the interval [0, 1].

- Let $p_{in}(r)$ and $p_{out}(s)$ denote the probability density of the Gray values in the input and output images.

$$p_s(s)ds = p_r(r)dr$$

- If $p_{in}(r)$ and T(r) are known, and $r = T^{-1}(s)$ satisfies condition 1, we can write (result from probability theory):

$$p_{out}(s) = \left[p_{in}(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

- One way to enhance the image is to design a transformation *T(.)* such that the gray values in the output is uniformly distributed in [0, 1], i.e. $p_{out}(s) = 1, \quad 0 \leq s \leq 1$

- In terms of histograms, the output image will have all gray values in "equal proportion" .

- This technique is called **histogram equalizatio**n.

Next we derive the gray values in the output is uniformly distributed in [0, 1].

- Consider the transformation

$$s = T(r) = \int_0^r p_{in}(w)dw, \qquad 0 \le r \le 1$$

- Note that this is the cumulative distribution function (CDF) of p_{in} (r) and satisfies the previous two conditions.

- From the previous equation and using the fundamental

theorem of calculus,

$$\frac{ds}{dr} = p_{in}(r)$$

- Therefore, the output histogram is given by

$$p_{out}(s) = \left[p_{in}(r) \cdot \frac{1}{p_{in}(r)} \right]_{r=T^{-1}(s)} = [1]_{r=T^{-1}(s)} = 1, \qquad 0 \le s \le 1$$

- The output probability density function is uniform, regardless of the input.

- Thus, using a transformation function equal to the CDF of input gray values r, we can obtain an image with uniform gray values.

- This usually results in an enhanced image, with an increase in the dynamic range of pixel values.

How to implement histogram equalization?

Step 1:For images with discrete gray values, compute:

$$p_{in}(r_k) = \frac{n_k}{n} \qquad 0 \le r_k \le 1 \qquad 0 \le k \le L-1$$

L: Total number of gray levels

n_k: Number of pixels with gray value r_k

n: Total number of pixels in the image

Step 2: Based on CDF, compute the discrete version of the previous transformation :

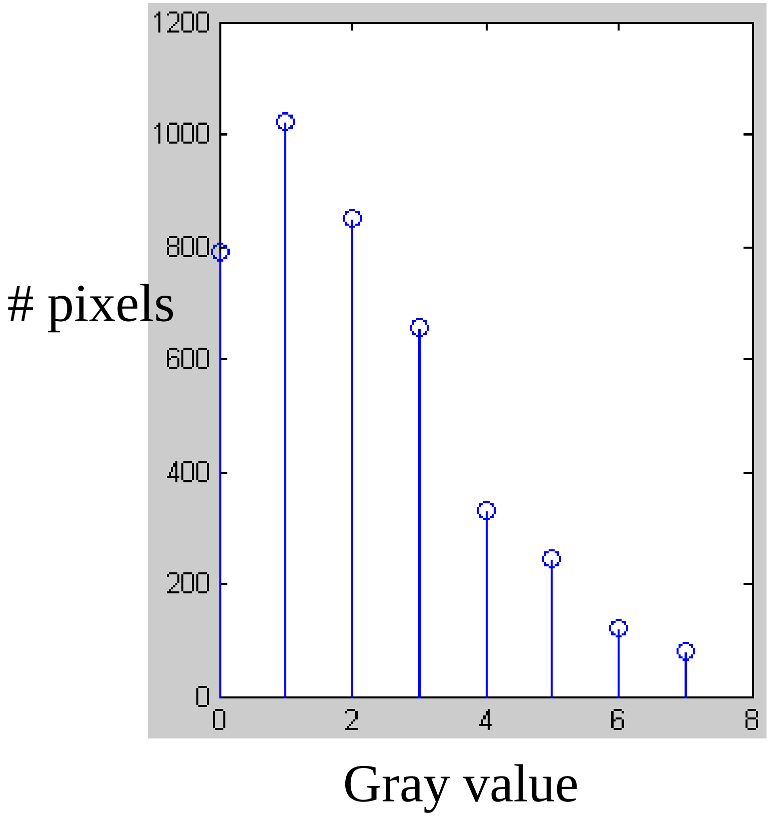
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j) \qquad 0 \le k \le L-1$$

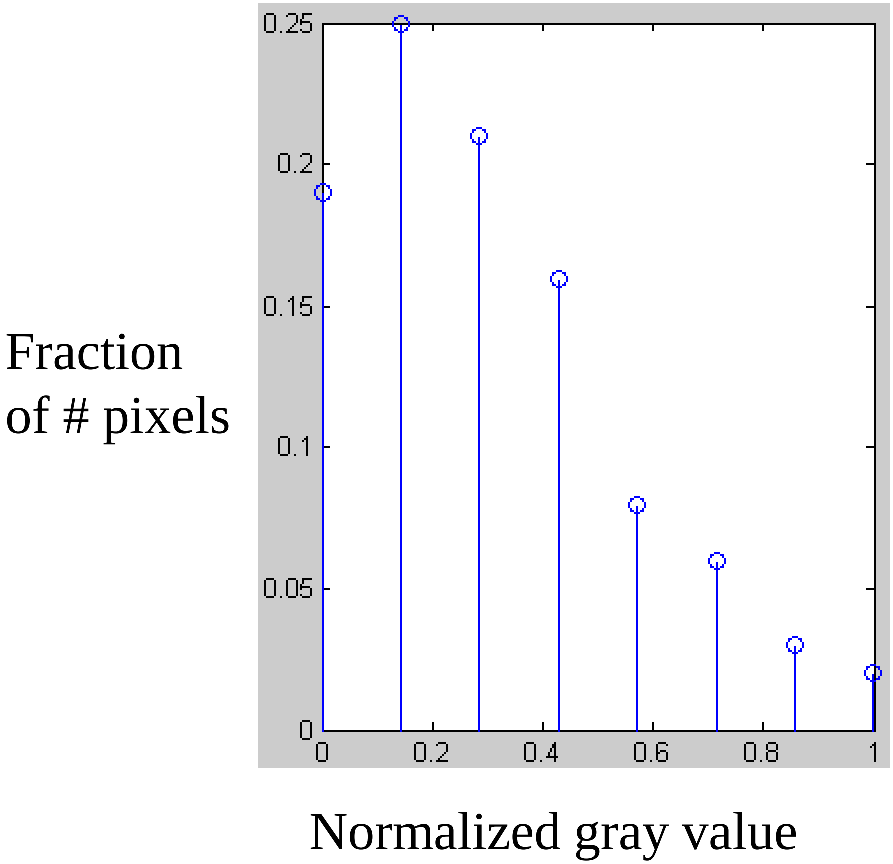
Example:

- Consider an 8-level 64 x 64 image with gray values (0, 1, …, 7). The normalized gray values are (0, 1/7, 2/7, …, 1). The normalized histogram is given below:

k	r_k	n_k	$p(r_k) = n_k/n$
0	0	790	0.19
1	1/7	1023	0.25
2	2/7	850	0.21
3	3/7	656	0.16
4	4/7	329	0.08
5	5/7	245	0.06
6	6/7	122	0.03
7	1	81	0.02

NB: The gray values in output are also (0, 1/7, 2/7, …, 1).






```
>> clear
>> h=[790 1023 850 656 329 245 122 81];
>> stem(0:7,h)
```

```
>> clear
>> h=[0.19 0.25 0.21 0.16 0.08 0.06 0.03 0.02];
>> stem(0:0.142857:1,h)
```

- Applying the transformation, $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j)$ we have

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \qquad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \qquad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad\qquad s_3 = 5.67 \quad \rightarrow 6$$

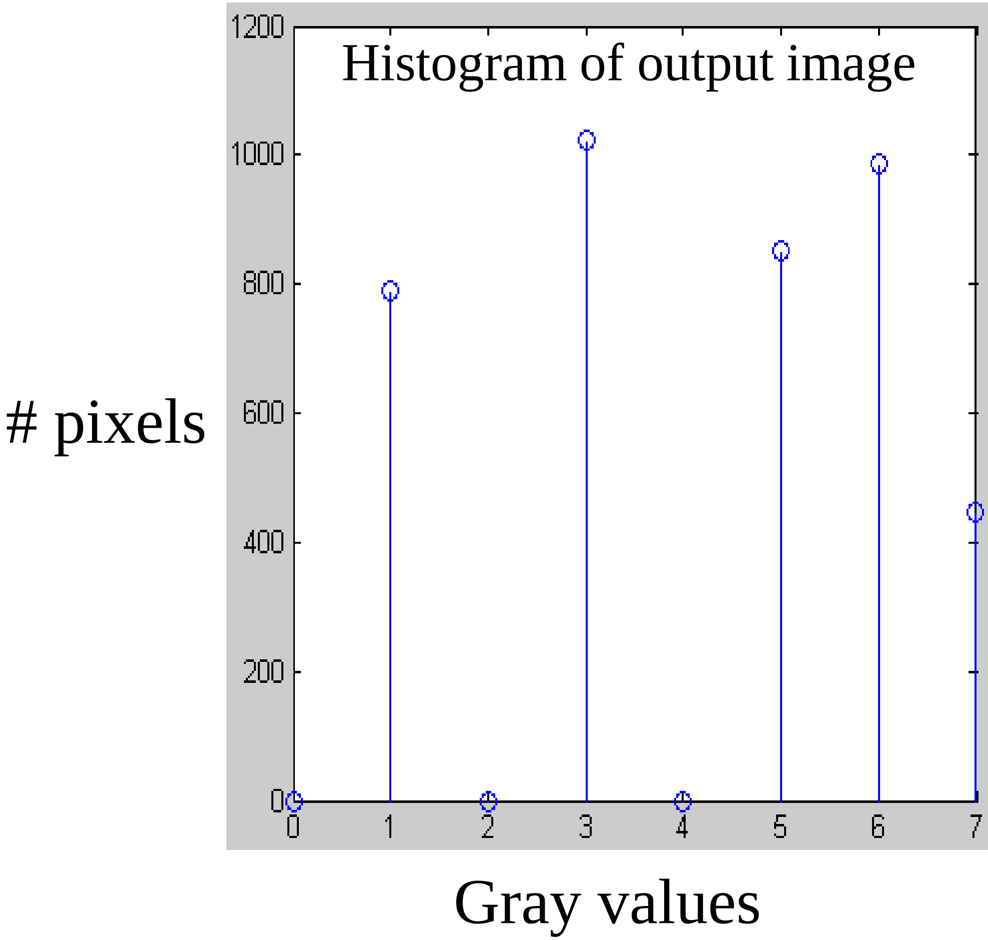
$$s_4 = 6.23 \quad \rightarrow 6 \qquad\qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad\qquad s_7 = 7.00 \quad \rightarrow 7$$

- Notice that there are only five distinct gray levels --- (1/7, 3/7, 5/7, 6/7, 1) in the output image. We will relabel them as (s_0, s_1, …, s_4).

- With this transformation, the output image will have histogram

k	s_k	n_k	$p(s_k) = n_k/n$
0	1/7	790	0.19
1	3/7	1023	0.25
2	5/7	850	0.21
3	6/7	985	0.24
4	1	448	0.11

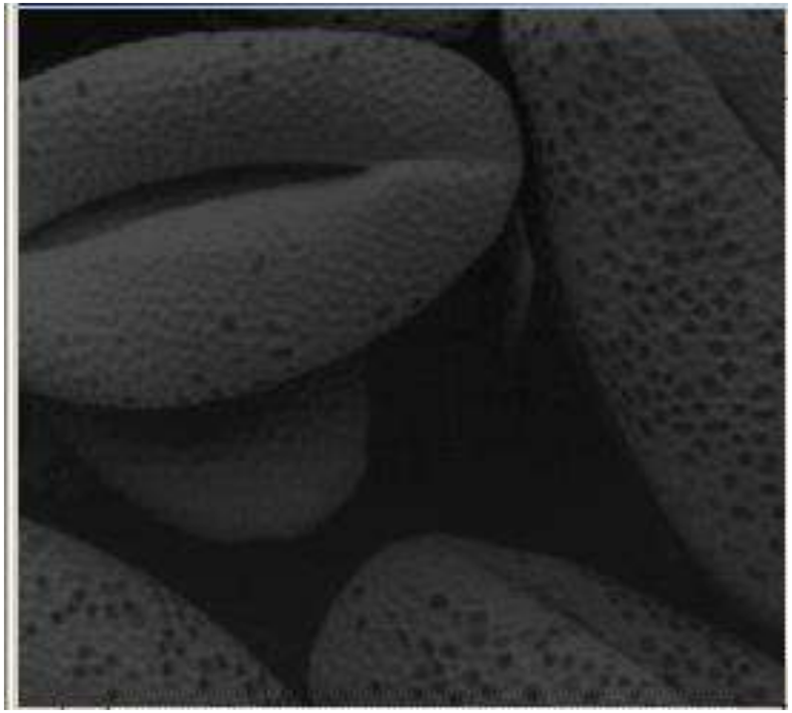


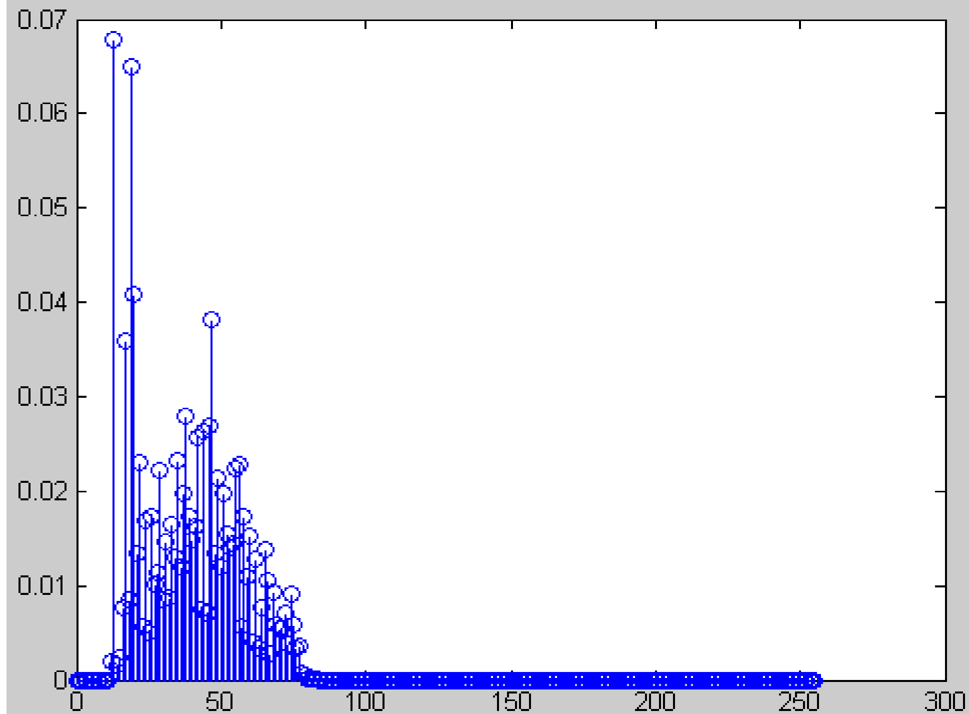

```
>> clear
>> h=[0 790 0 1023 0 850 985 448];
>> stem(0:7,h)
```

• Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

Example

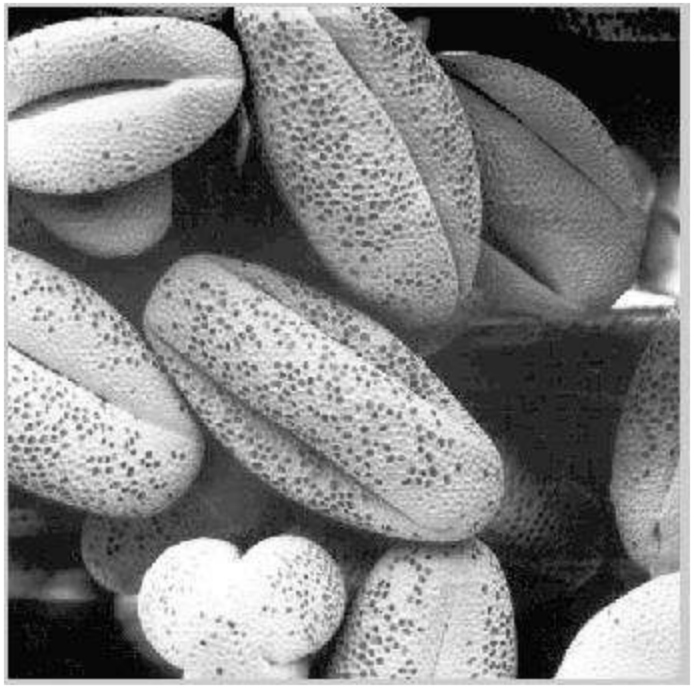
Original image and its histogram

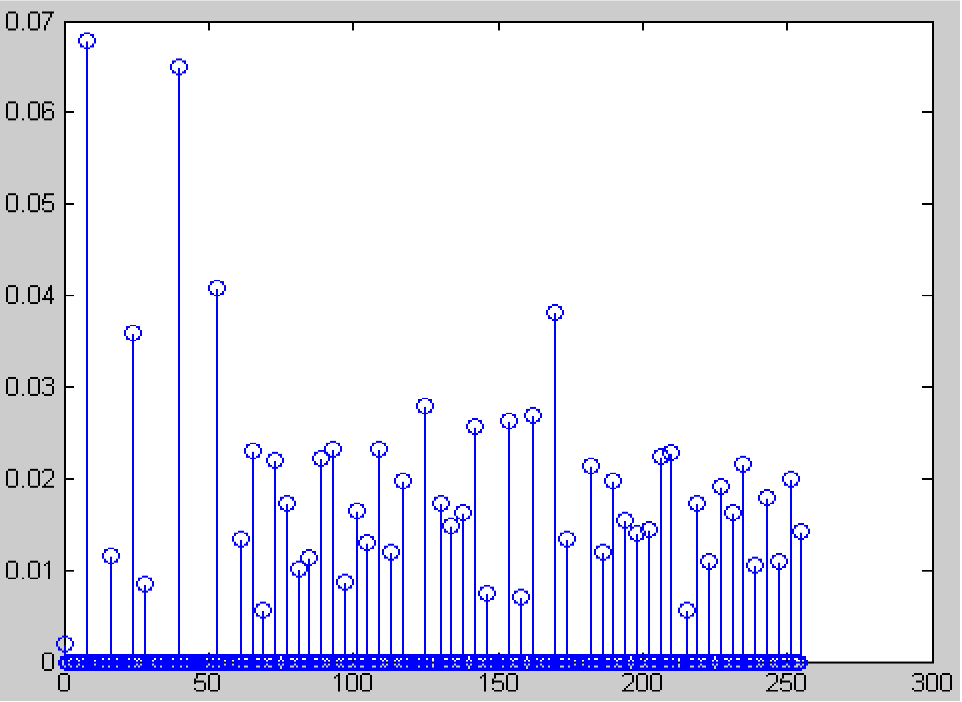





```
>> clear;
[ix,map]=imread('Fig3_15a.jpg');
imshow(ix)
figure;
ix=double(ix);
h=histogram(ix);
stem(0:255,h);
```

Histogram equalized image and its histogram



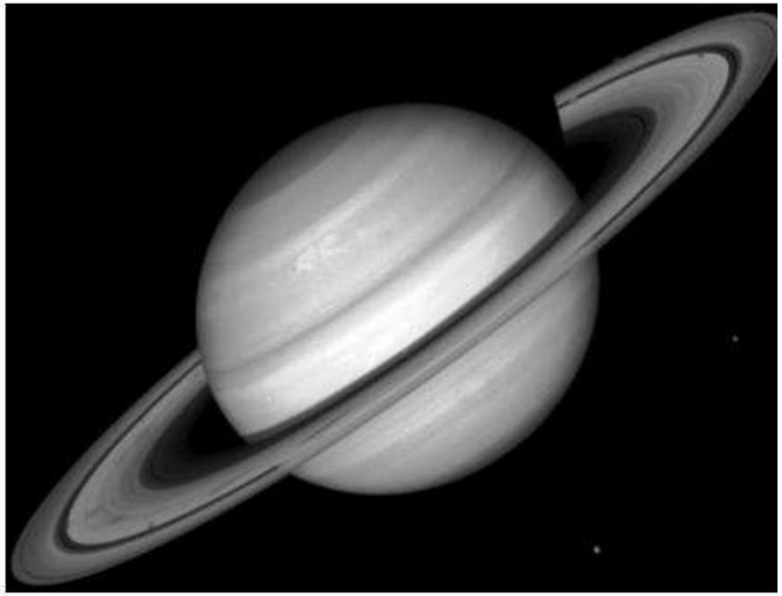
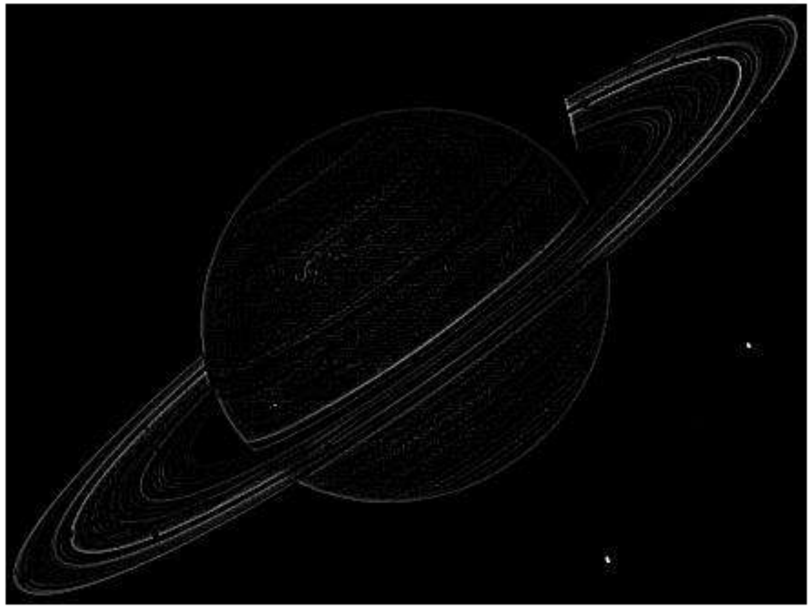
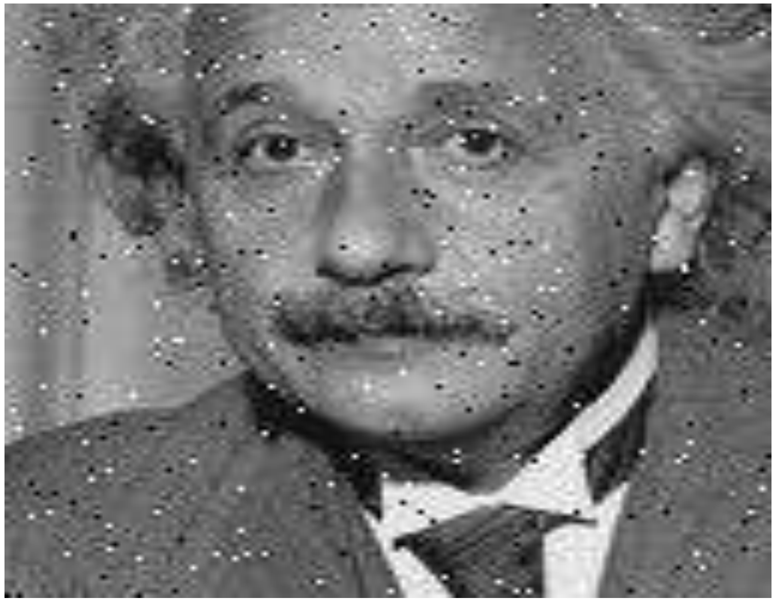
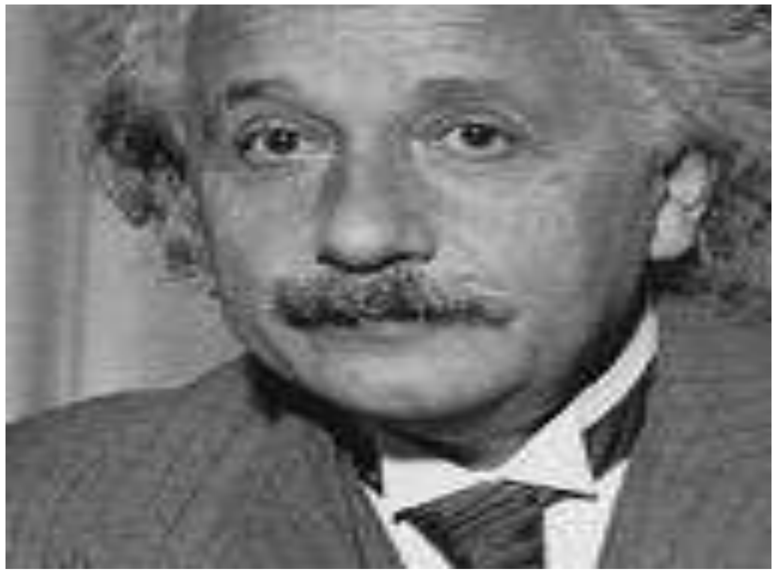



```
>> clear
>> [ix,map]=imread('Fig3_15a.jpg');
>> imshow(ix);
>> iy=histeq(ix);
>> figure
>> imshow(iy);
>> iy=double(iy);
>> hy=histogram(iy);
>> figure
>> stem(0:255,hy);
```

Basics of Spatial Filtering

OUTLINE

- Fundamentals Spatial Filtering
- Smoothing Spatial Filters
  - Smoothing filters are used for blurring and for noise reduction
- Sharpening Spatial Filters
  - Highlight fine detail or enhance detail that has been blurred

Spatial Filtering

- The word "filtering" has been borrowed from the frequency domain.

- Filters are classified as:
  - Low-pass (i.e., preserve low frequencies)
  - High-pass (i.e., preserve high frequencies)
  - Band-pass (i.e., preserve frequencies within a band)
  - Band-reject (i.e., reject frequencies within a band)

Spatial Filtering

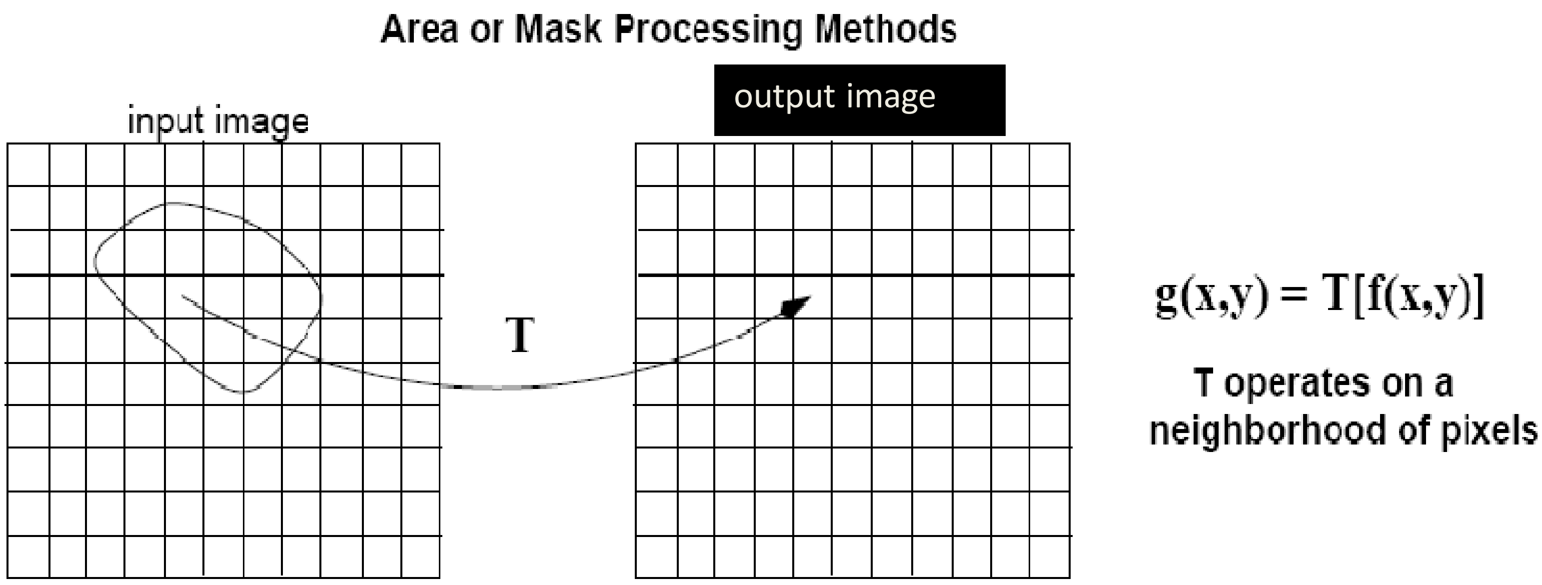
- ***Background :***
  - Filter term in “Digital image processing” is referred to the subimage
  - There are others term to call subimage such as mask, kernel, template, or window
  - The value in a filter subimage are referred as coefficients, rather than pixels.
- ***Basics of Spatial Filtering :***
  - The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.
  - Spatial filtering term is the filtering operations that are performed directly on the pixels of an image

❖ *Definition*

- Spatial filtering are defined by:

  (1) A **neighborhood**

  (2) A **predefined operation** that is performed on the pixels inside the neighborhood



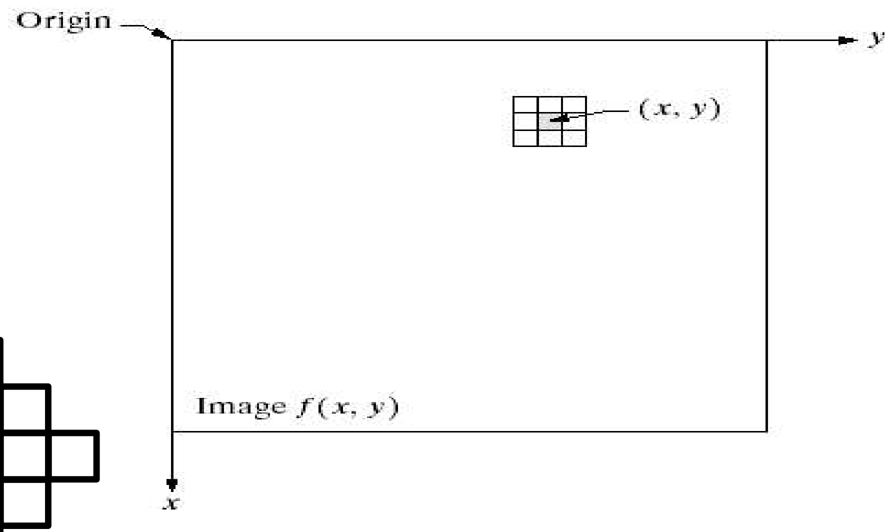
Operation with 3x3 Filter

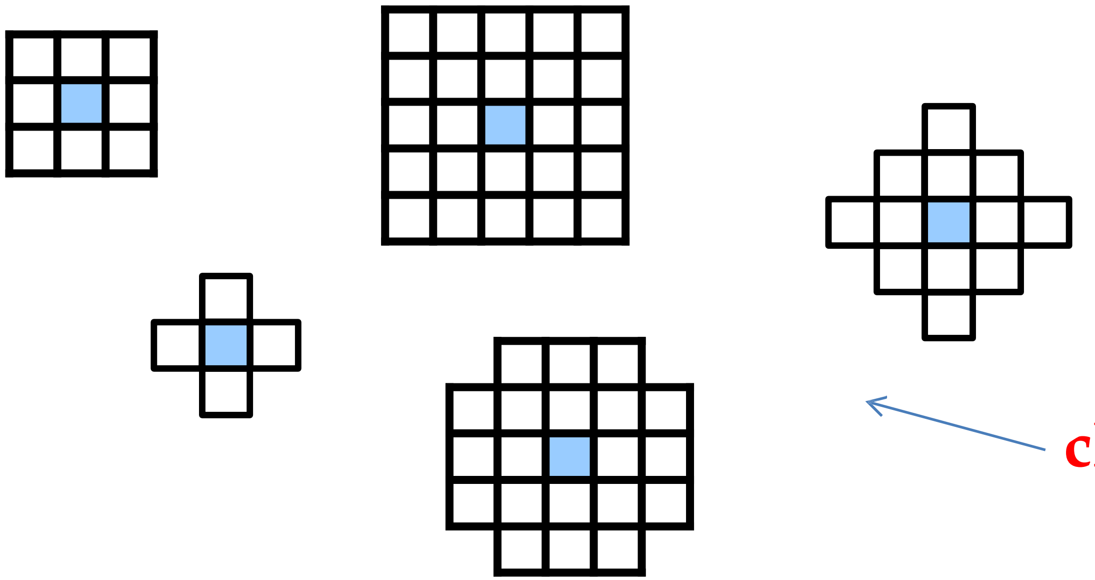
- 3 x 3 Neighborhood / Mask / Window / Template:

	(y - 1)	y	(y + 1)	Y
(x - 1)	w(-1,-1) f(x-1, y-1)	w(-1,0) f(x-1, y)	w(-1,1) f(x-1, y+1)	
x	w(0,-1) f(x, y-1)	w(0,0) f(x, y)	w(0,1) f(x, y+1)	
(x + 1)	w(1,-1) f(x+1, y-1)	w(1,0) f(x+1, y)	w(1,1) f(x+1, y+1)	
X				

❖Spatial Neighborhood

FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.





choices of neighborhood

- Typically, the neighborhood is **rectangular** and its size is much smaller than that of *f(x,y)*
  - e.g., 3x3 or 5x5

❖ *Mechanics of spatial filtering*

- The process consists simply of moving the filter mask from point to point in an image.

- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship

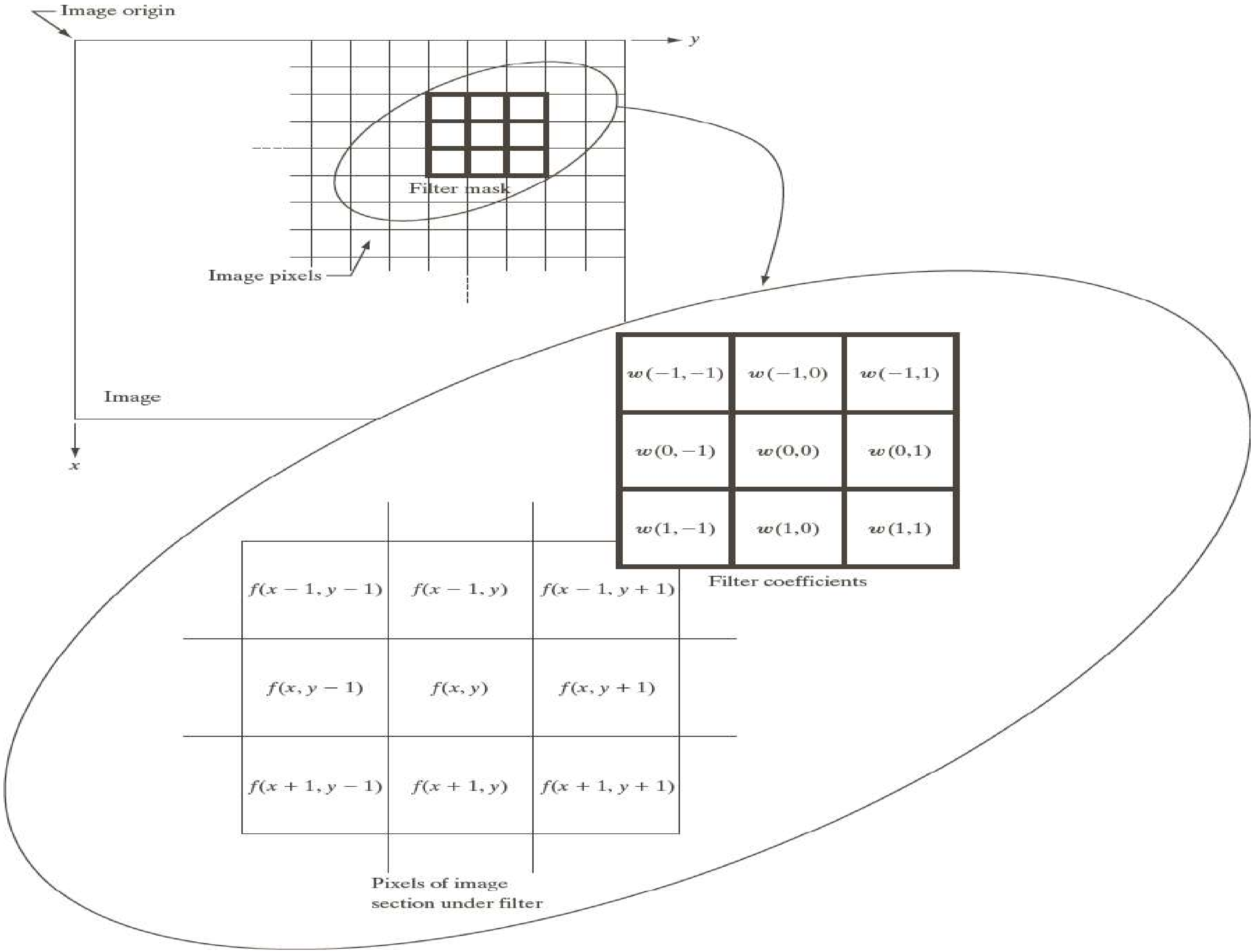


Image origin
y
Filter mask
Image pixels
Image
x
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image
section under filter

❖ *Type*

- Linear spatial filtering
- Nonlinear spatial filtering

➢ A filtering method is linear when the output is a weighted sum of the input pixels.

w1	w2	w3
w4	w5	w6
w7	w8	w9

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

➢ Methods that do not satisfy the above property are called non-linear.

z1	z2	z3
z4	z5	z6
z7	z8	z9

$$z_5' = max(z_k,\ k = 1, 2, \ldots, 9)$$

Linear Spatial Filtering Methods

- Two main linear spatial filtering methods:
  - Correlation
  - Convolution

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

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

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

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

Correlation & Convolution

- Correlation & Convolution are two closely related concepts used in linear spatial filtering.
- *Correlation*: It is a process of moving a filter mask over an image & computing the sum of products at each location.
- *Convolution*: Here, the mechanics are same, except that the filter is first rotated by 180˚.
- Correlation & Convolution are function of displacement. Correlation & Convolution are exactly same if the filter mask is symmetric.
- 1D correlation and convolution of a filter with a discrete unit impulse is shown below.

	Correlation	Convolution	
	Origin f w	Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 3 2 8	0 0 0 1 0 0 0 0 8 2 3 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 3 2 1	(j)
	Zero padding		
(c)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(k)

(d)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
	  1 2 3 2 8
	Position after one shift
		0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
		  8 2 3 2 1
			(l)

(e)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
	    1 2 3 2 8
	Position after four shifts
		0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
		    8 2 3 2 1
			(m)

(f)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
	        1 2 3 2 8
	Final position
		0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
		        8 2 3 2 1
			(n)

	Full correlation result	Full convolution result	
(g)	0 0 0 8 2 3 2 1 0 0 0 0	0 0 0 1 2 3 2 8 0 0 0 0	(o)
	Cropped correlation result	Cropped convolution result	
(h)	0 8 2 3 2 1 0 0	0 1 2 3 2 8 0 0	(p)

Correlation & Convolution

- Correlation is a function of displacement of the filter.
- Correlating a filter w with a function that contains all ‘0’ & single ‘1’ yields a 180˚ rotated copy of w.
- Correlating a function with discrete unit impulse yields a rotated (time inverted) version of the function.
- Convolving a function with a unit impulse yields the same function.
- Thus, to perform convolution all we have to do is rotate one function by 180˚ & perform same operation as in correlation.

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)

Rotated w

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	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

(f)

Full convolution 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	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	0	0
0	0	0	0	0	0	0	0	0

(g)

Cropped convolution result

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

(h)

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.

Some fundamental properties of convolution and correlation

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	-
Associative	$f \star (g \star h) = (f \star g) \star h$	-
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \star (g + h) = (f \star g) + (f \star h)$

❖ *Linear spatial filtering*

Pixels of image

f(x-1.y-1)	f(x-1,y)	f(x,y-1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

- The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask

$$
\begin{aligned}
g(x,y) = {} & w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + w(-1,1)\,f(x-1,y+1) + \\
& w(0,-1)\,f(x,y-1) + w(0,0)\,f(x,y) + w(0,1)\,f(x,y+1) + \\
& w(1,-1)\,f(x+1,y-1) + w(1,0)\,f(x+1,y) + w(1,1)\,f(x+1,y+1)
\end{aligned}
$$

❖ Nonlinear spatial filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration

  - Ex: Noise reduction can be achieved effectively with a nonlinear filter with a basic function of computing the median gray level value in the neighbourhood

  - Computation of median is a non-linear operation, as is that of variance

- **Motivation: Limitation of Linear Filters**
  - Frequency shaping

    enhance some frequency components and suppress the others
  - For individual frequency component, cannot differentiate its "desirable" and "undesirable" parts

- Nonlinear Filters
  - Cannot be expressed as convolution
  - Cannot be expressed as frequency shaping

- **"Nonlinear" Means Everything** (other than linear)
  - Need to be more specific
  - Often heuristic
  - We will study some "nice" ones

SMOOTHING FILTERS(Low Pass) IN SPATIAL DOMAIN

Smoothing Spatial Filter

- Smoothing filters are used for
  - blurring
  - noise reduction.
- Blurring is used in preprocessing steps to removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves
- Noise reduction can be accomplished by blurring

Types of Smoothing Filter

There are 2 way of smoothing spatial filters

- Linear Filters – operations performed on image pixel
- Order-Statistics (non-linear) Filters - based on ranking the pixels

Linear Filter

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- The idea is replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

Linear Filter (cont..)

- This process result in an image reduce the sharp transitions in intensities.
- Two mask
  - Averaging filter
  - Weighted averaging filter

Averaging Filter

- A major use of averaging filters is in the reduction of irrelevant detail in image.
- mxn mask would have a normalizing constant equal to 1/mn.
- Its also known as low pass filter.
- A spatial averaging filter in which all coefficients are equal is called a box filter.

Averaging Filter - Example

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothing Filters: Averaging

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\frac{1}{49} \times \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Weighted Average Filter - Example

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

Weighted Averaging Filter

- The general implementation for filtering an MxN image with a weighted averaging filter of size m x n is given by the expression

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

- For complete filtered image apply **x = 0,1,2,3……..m-1** and **y=0,1,2,3,……..n-1** in the above equation.

Ex. 1) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size averaging mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

0	0	0						
0	10	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

0	0	0						
0	4.44	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 10 & 10 \\ \hline 0 & 10 & 10 \\ \hline\end{array}$$

0	0	0	0					
0	4.44	6.66	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 10 & 10 & 10 \\ \hline 10 & 10 & 10 \\ \hline \end{array}$$

0	0	0	0	0				
0	4.44	6.66	6.66	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0			
0	4.44	6.66	6.66	6.66	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 10 & 10 & 10 \\ \hline 10 & 10 & 10 \\ \hline\end{array}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	10	0
	10	10	10	10	10	10	10	10	
	10	10	10	10	10	10	10	10	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	10	10	10	10	10	10	10	0
0	50	50	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 0 & 10 & 10 \\ 0 & 10 & 10 \\ 0 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 10 & 10 & 10 \\ \hline 50 & 50 & 50 \\ \hline 50 & 50 & 50 \\ \hline\end{array}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 50 & 50 & 50 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	33.33	0
0	33.33	50	50	50	50	50	50	33.33	0
0	22.22	33.33	33.33	33.33	33.33	33.33	33.33	22.22	0
	0	0	0	0	0	0	0	0	0

$$\frac{1}{9}\begin{bmatrix} 50 & 50 & 0 \\ 50 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	23.33	23.33	23.33	23.33	23.33	23.33	10
50	36.66	36.66	36.66	36.66	36.66	36.66	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

- In the resultant image the Low frequency region has remained unchanged.
- Sharp transition between 10 & 50 has changed from 10 to 23.33 to 36.66 and finally to 50.
- Thus, Sharp edges has become blurred.
- Best result when used over image corrupted by Gaussian noise.
- Other types of low pass averaging mask are:

$$\frac{1}{6}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \qquad \frac{1}{10}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Smoothing filters – Example

input image **smoothed image**

Order-Statistics Filter

- Order-statistics filters are nonlinear spatial filters.
- It is based on ordering (ranking) the pixels contained in the image area encompassed by the filter,
- It replacing the value of the center pixel with the value determined by the ranking result.

Order- statics filter

- The filter selects a sample from the window, does not average
- Edges are better preserved than with liner filters
- Best suited for “salt and pepper” noise

Types of order-statics filter

Different types of order-statics filters are

- Minimum filter
- Maximum filter
- Median filter

Minimum Filter

- The 0^{th} percentile filter is the min filter.
- Minimum filter selects the smallest value in the window and replace the center by the smallest value
- Using comparison the minimum value can be obtained fast.(not necessary to sort)
- It enhances the dark areas of image

.

Minimum Filter - Example

(mask size =3 x 3)

(mask size =7 x 7)

Maximum Filter

- The maximum filter selects the largest value within of pixel values, and replace the center by the largest value.

- Using comparison the maximum value can be obtained fast.(not necessary to sort)
- Using the 100^{th} percentile results in the so-called *max filter*
- it enhances bright areas of image

Maximum Filter

mask (3 x 3)

mask (7 x 7)

Median Filter

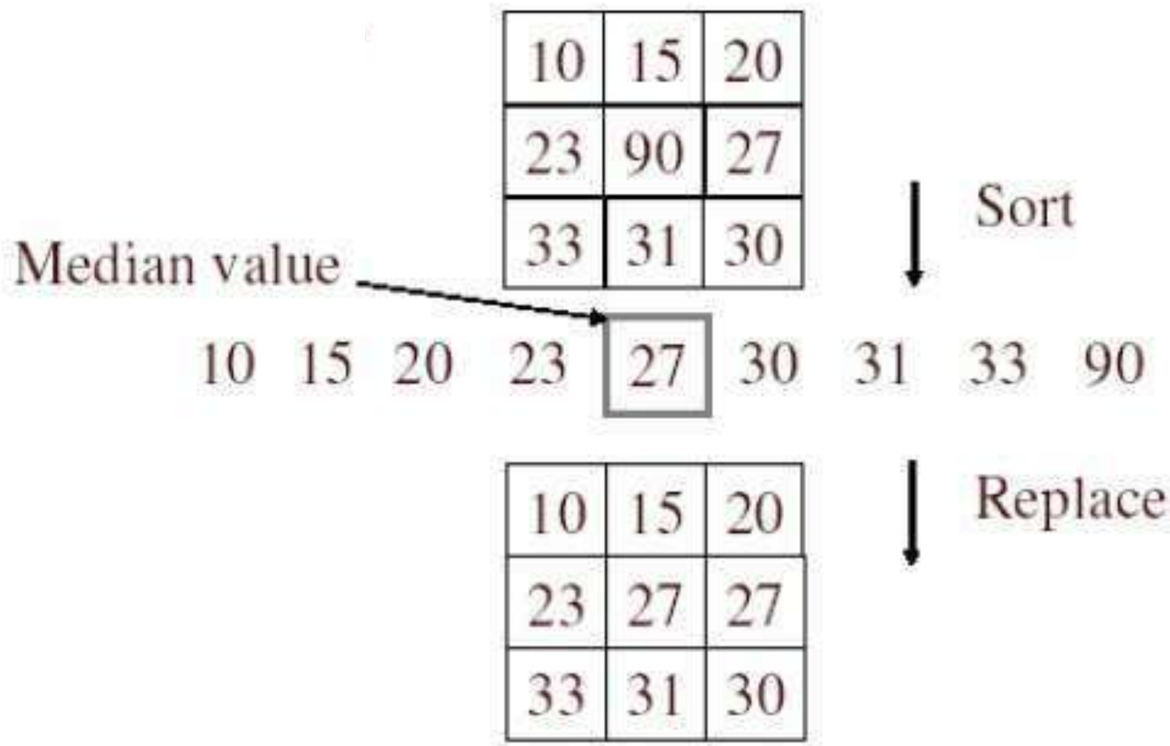
Three steps to be followed to run a median filter:

1. Consider each pixel in the image

2. Sort the neighboring pixels into order based upon their intensities

3. Replace the original value of the pixel with the median value from the list.

Median Filter - Process



Ex. 2) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size median filter mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Median Filter – Process

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Median Filter – Process

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

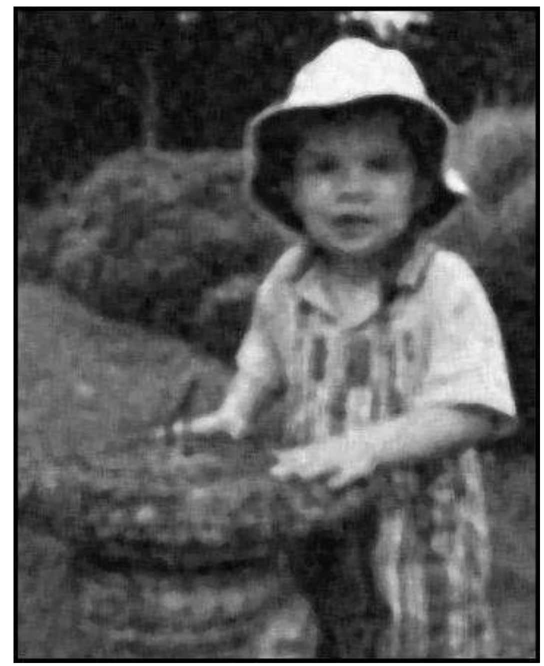
8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Median Filter - Example

Median Filter size =3 x 3



Median Filter size =7 x 7

Conclusion

- A linear filter cannot totally eliminate impulse noise, as a single pixel which acts as an intensity spike can contribute significantly to the weighted average of the filter.

- Non-linear filters can be robust to this type of noise because single outlier pixel intensities can be eliminated entirely.

Original image

Median filter

Mean filter


Thank you

Sharpening Filters(High pass filter)

1. The concept of sharpening filter
2. First and second order derivatives
3. Laplace filter
4. Unsharp mask
5. High boost filter
6. Gradient mask

Sharpening Spatial Filters

- To highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

- Blurring vs. Sharpening

  - Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process of integration

  - Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

Derivative operator

- The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

- Image differentiation
  - enhances edges and other discontinuities (noise)
  - deemphasizes area with slowly varying gray-level values.

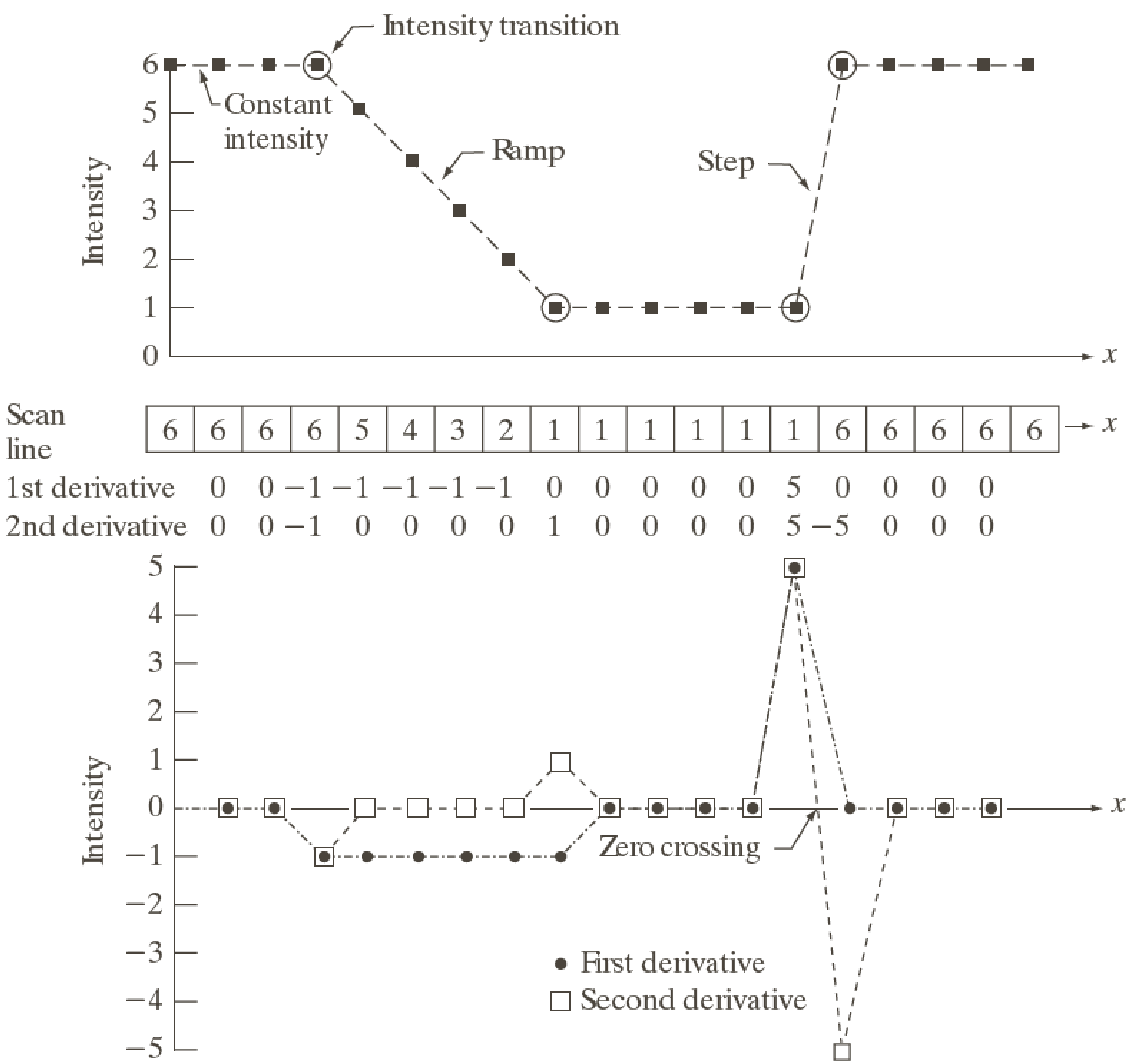
First and second order difference of 1D

- The basic definition of 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 a one-dimensional function f(x) is 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.

First and Second-order derivative of 2D

- when we consider an image function of two variables, f(x, y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

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

(linear operator)

Laplacian operator
(non-linear)

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

Discrete form of Laplacian

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

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

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

Result Laplacian mask

0	1	0
1	−4	1
0	1	0

Laplacian mask implemented an extension of diagonal neighbors

1	1	1
1	−8	1
1	1	1

Other implementation of Laplacian masks

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

give the same result, but we have to keep in mind that when combining (add / subtract) a Laplacian-filtered image with another image.

Effect of Laplacian Operator

- as it is a derivative operator,
  – it highlights gray-level discontinuities in an image
  – it deemphasizes regions with slowly varying gray levels
- tends to produce images that have
  – grayish edge lines and other discontinuities, all superimposed on a dark,
  – featureless background.

Correct the effect of featureless background

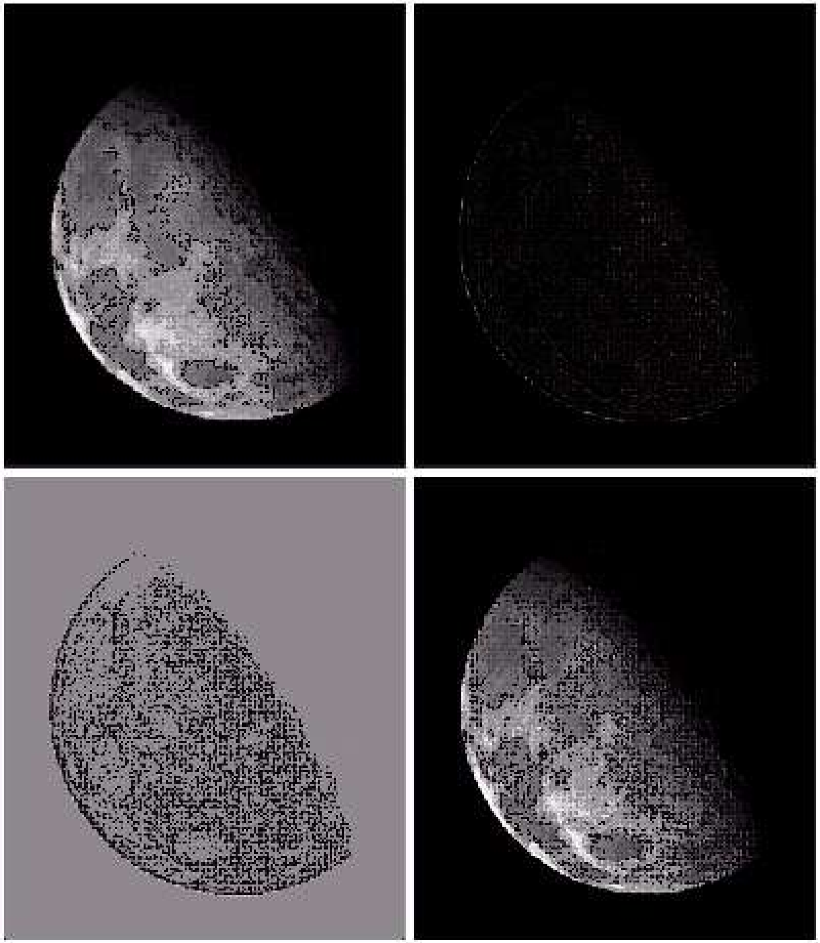
- easily by adding the original and Laplacian image.
- be careful with the Laplacian filter used

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

Example



- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Mask of Laplacian + addition

- to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.

Mask of Laplacian + addition

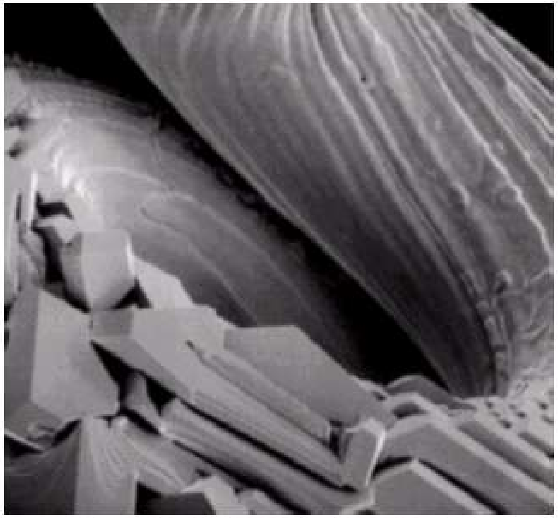
$$
\begin{aligned}
g(x,y) &= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\quad + f(x,y+1) + f(x,y-1) + 4f(x,y)] \\
&= 5f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\quad + f(x,y+1) + f(x,y-1)]
\end{aligned}
$$

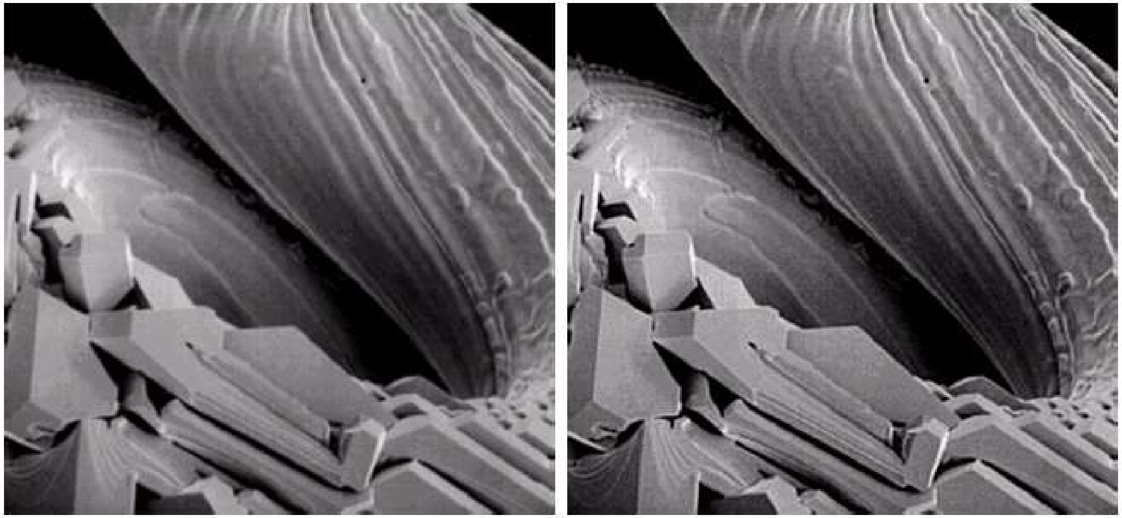
0	-1	0
-1	5	-1
0	-1	0

Example

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.)

Note

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 9 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

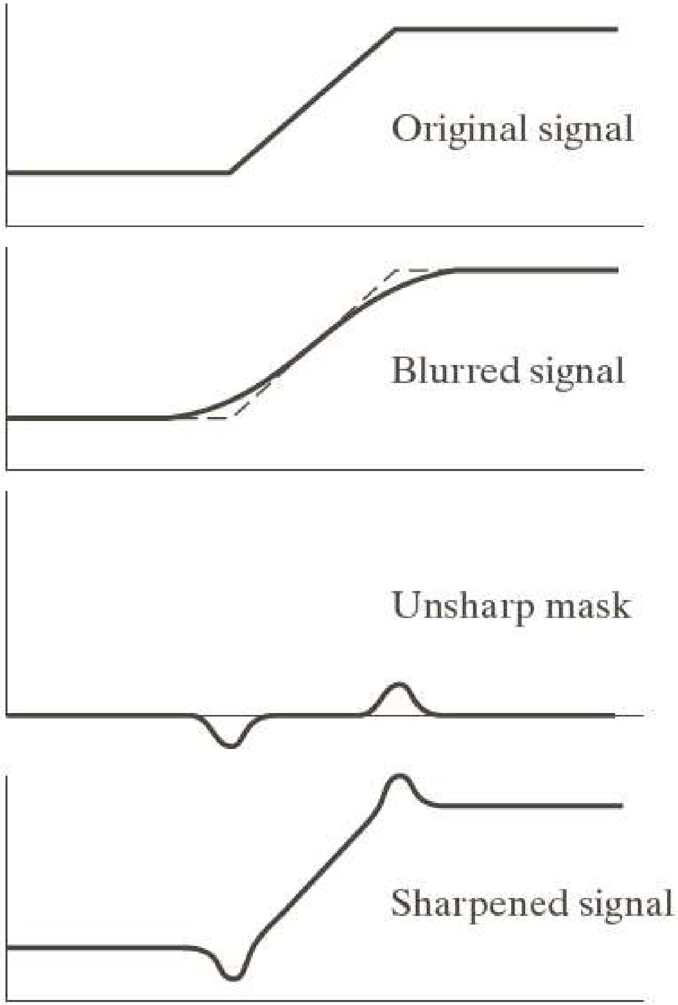
Unsharp masking

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

sharpened image = original image – blurred image

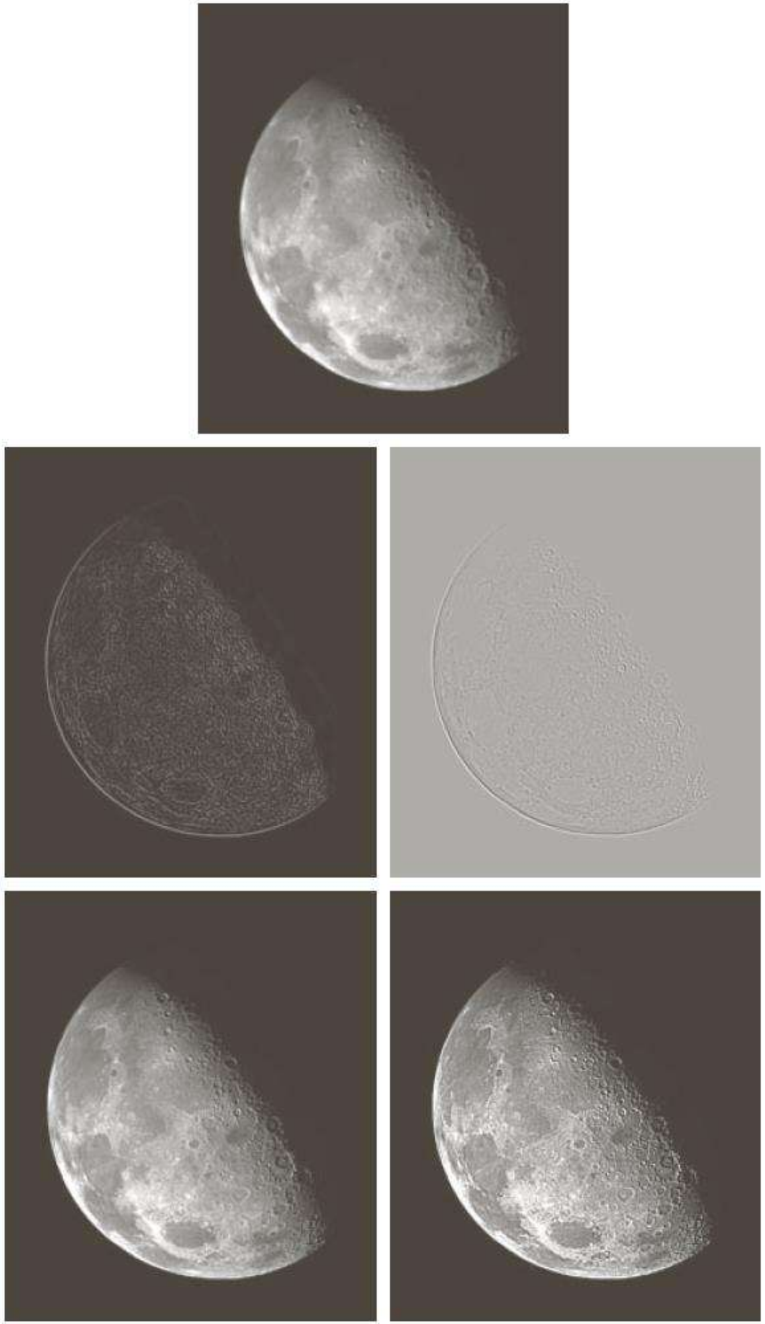
- to subtract a blurred version of an image produces sharpening output image.

Unsharp mask



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).



	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.)

High-boost filtering

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = (A-1)f(x, y) - f(x, y)\bar{f}(x, y)$$

$$= (A-1)f(x, y) - f_s(x, y)$$

- generalized form of Unsharp masking
- A $\geq$ 1

High-boost filtering

$$f_{hb}(x,y) = (A-1)f(x,y) - f_s(x,y)$$

- if we use Laplacian filter to create sharpen image $f_s(x,y)$ with addition of original image

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y) \end{cases}$$

High-boost Masks

0	−1	0
−1	$A + 4$	−1
0	−1	0

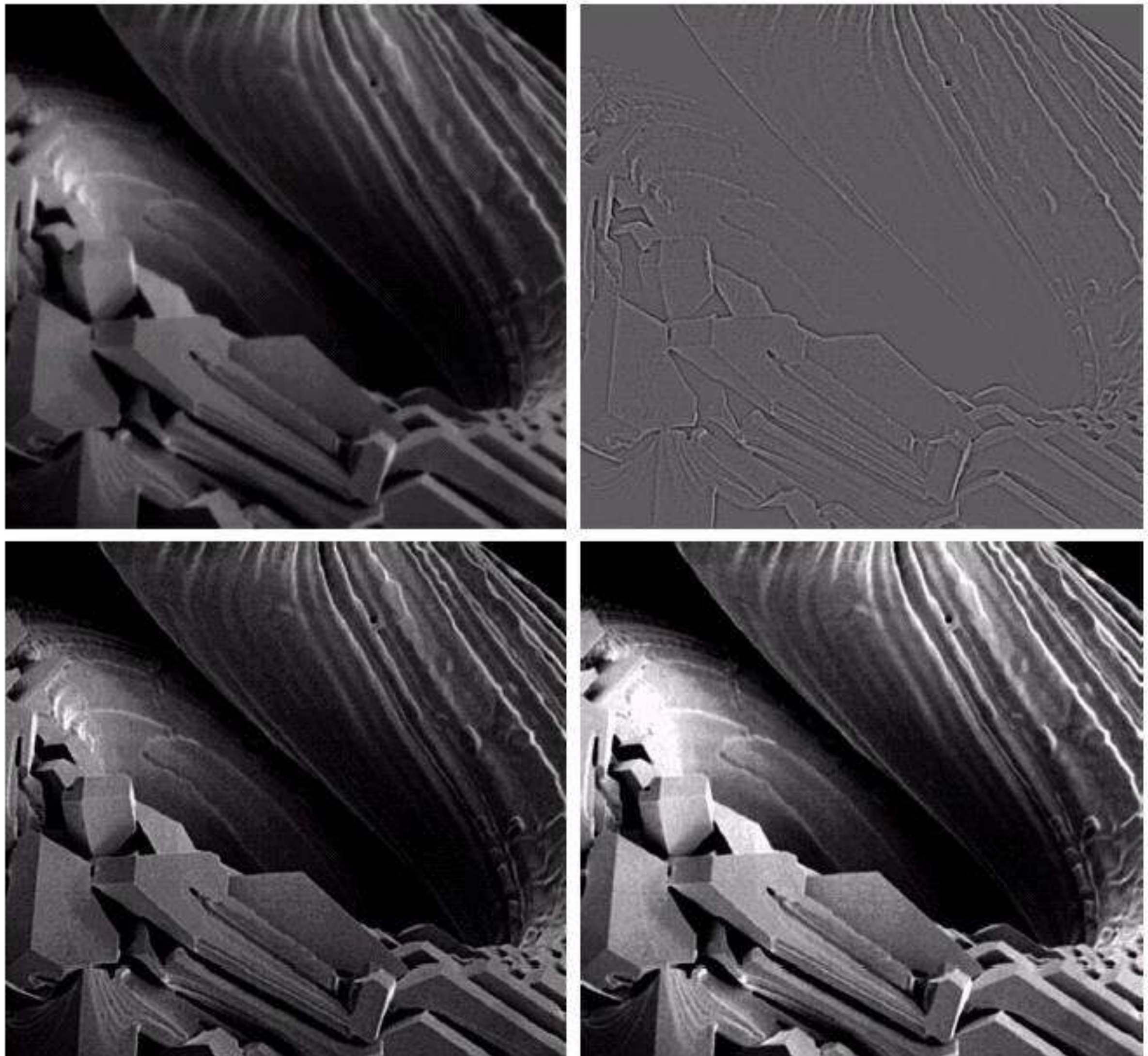
−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

- $A \geq 1$
- if A = 1, it becomes “standard” Laplacian sharpening

Example

a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.



Gradient Operator

- first derivatives are implemented using the magnitude of the gradient.

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

$$\nabla f = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

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

the magnitude becomes nonlinear

Gradient Mask

- simplest approximation, 2x2

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

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

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

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

−1	0
0	1

0	−1
1	0

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

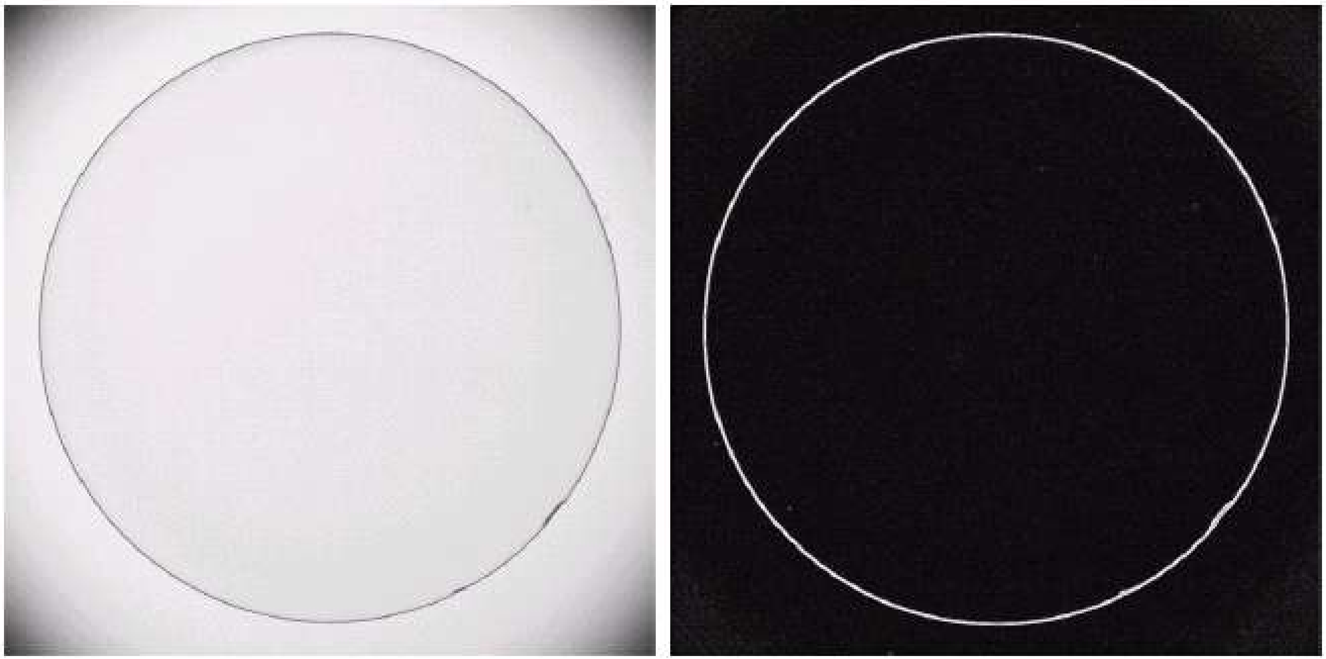
$$\nabla f \approx |G_x| + |G_y|$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

the weight value 2 is to achieve smoothing by giving more important to the center point

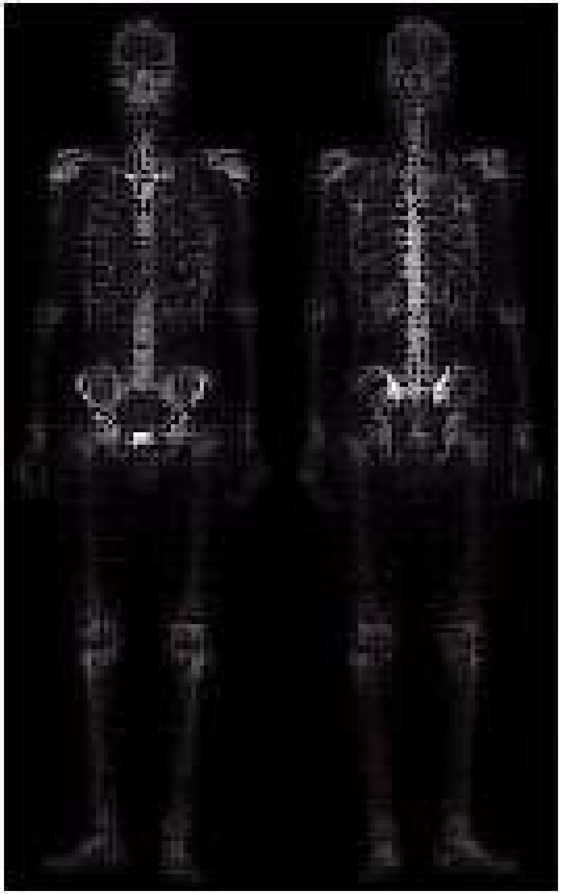
Example



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.)

Combining Spatial Enhancement Methods



- want to sharpen the original image and bring out more skeletal detail.
- problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

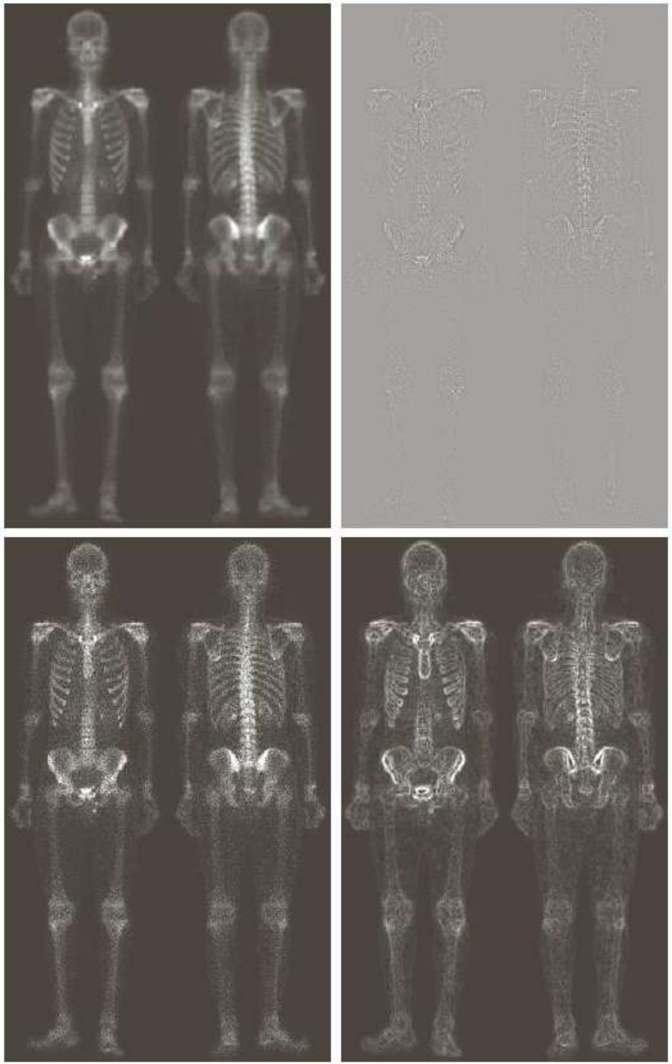
Example of Combining Spatial Enhancement Methods

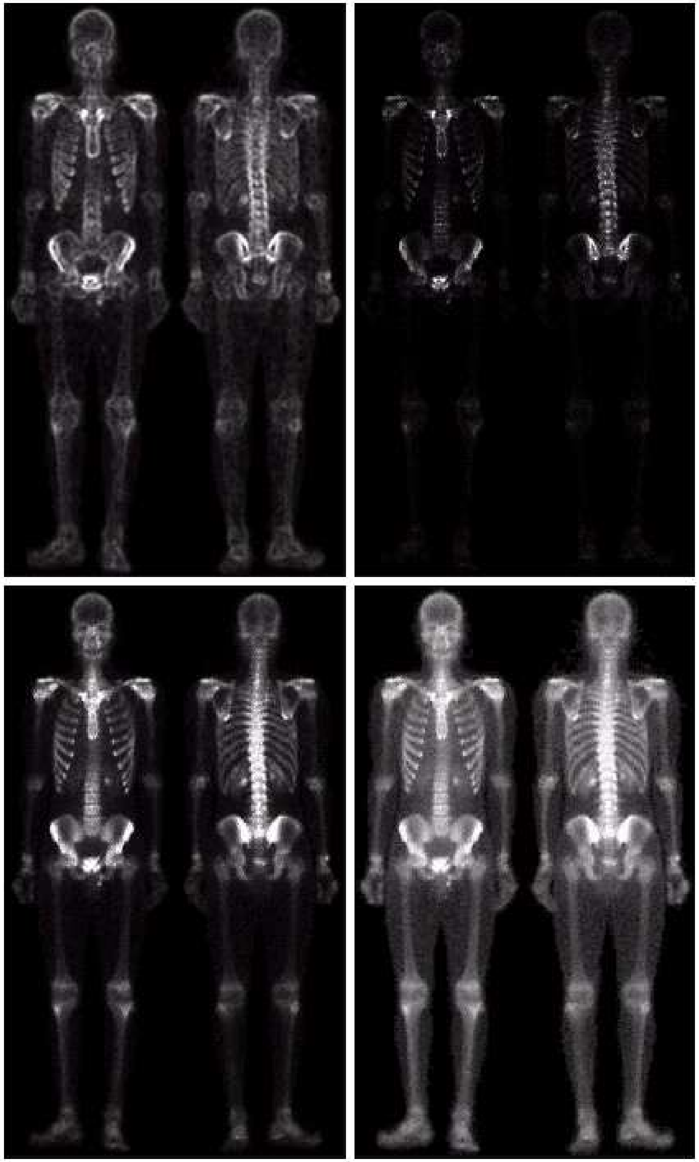
To solve :

1. Laplacian to highlight fine detail
2. gradient to enhance prominent edges
3. gray-level transformation to increase the dynamic range of gray levels

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).





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.)

Digital Image Processing

MODULE-3

Image Restoration

What is Image Restoration:

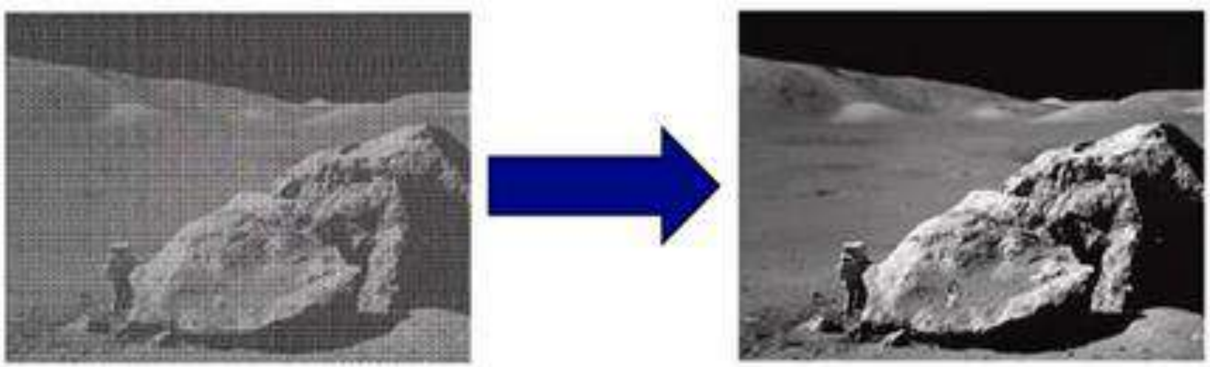
Image restoration aim to improve an image in some predefined sense.

What about image enhancement?

Image enhancement also improves an image by applying filters.

What is Image Restoration?

- Image restoration attempts to restore/reconstruct/ recover images that have been degraded.
  - It identifies the degradation process and attempt to reverse it
  - It is similar to image enhancement, but more objective



Difference:

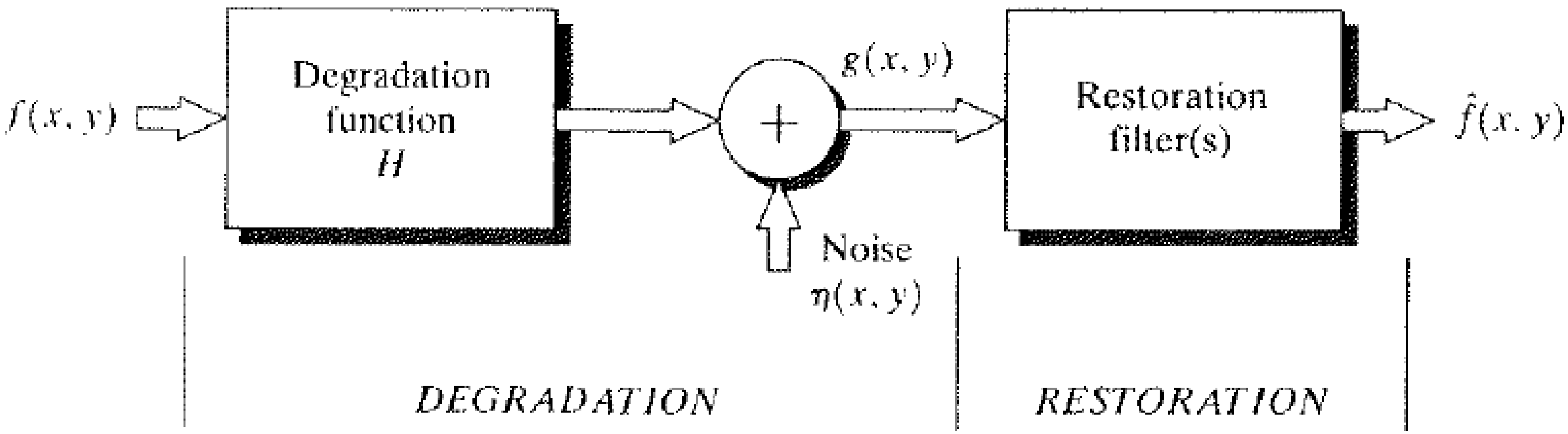
Image Enhancement --- Subjective process
Image Restoration --- Objective Process

- Restoration tries to recover / restore degraded image by using a prior knowledge of the degradation phenomenon.

- Restoration techniques focuses on:

  1. Modeling the degradation
  2. Applying inverse process in order to recover the original image.

Model of the Image Degradation / Restoration Process



- Degradation function along with some additive noise operates on f(x, y) to produce degraded image g(x, y)

- Given g(x, y), some knowledge about the degradation function H and additive noise η(x, y), objective of restoration is to obtain estimate f'(x, y) of the original image.

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

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

- h(x, y) = Spatial representation of H
- * indicates convolution

- Since convolution in Spatial domain = multiplication in Frequency Domain

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

- We Assume that H is identity operator

- We deal only with degradation due to Noise

<u>Noise Models</u>: Noise in digital image arises during

1. Image Acquisition
2. Transmission

<u>During Image Acquisition</u>

- ➢ Environmental conditions (Light Levels)
- ➢ Quality of sensing element

<u>During Transmission</u>

- ➢ Interference during transmission

Spatial Noise Descriptor

- Statistical behavior of the gray level values in the noise component.

- Can be considered as random variables

- Characterized by Probability Density Functions (PDFs)

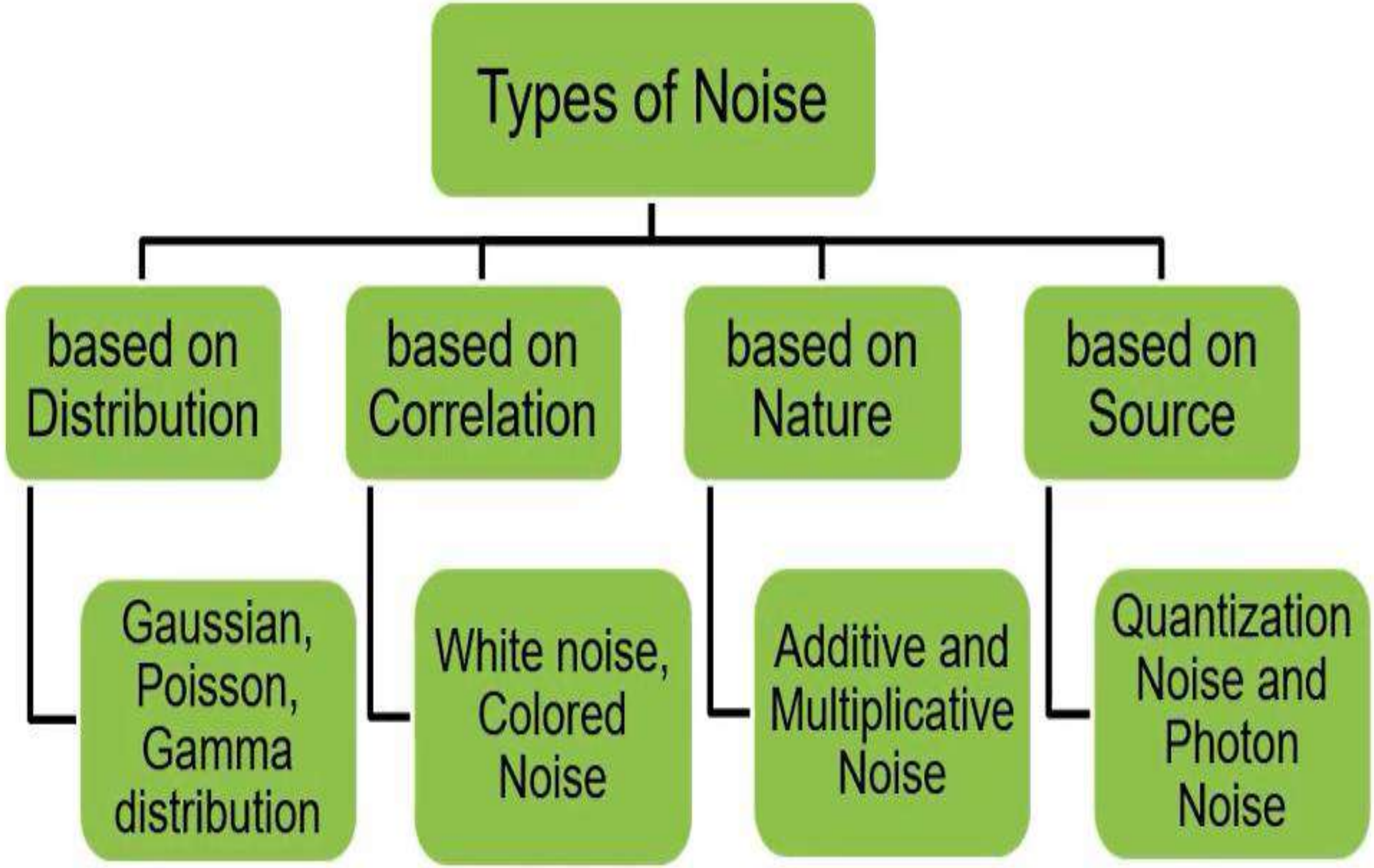
Noise

- Noise is signal similar to image.
- It’s an unwanted signal which distorts the original pixel intensity values, thus, degrading the quality of image
- Noise Model Assumption
  - Independent of Spatial Coordinates
  - Un-correlated with the image, i.e., no relation between the pixel values and noise components

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

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

Some of the types of frequent occurred noise that are encountered in Image Processing can be categorized into four categories.



Some Noise Probability Density Functions (PDFs):

- Uniform Noise
- Gaussian Noise
- Rayleigh Noise
- Erlang (Gamma) Noise
- Exponential Noise
- Impulse (Salt & Pepper Noise)
- Periodic Noise

Noise Modelling

Types of Noise based on Distribution

- Noise is a fluctuation in pixel values and it is characterized by random variable
- A random variable probability distribution is an equation that links the values of the statistical result with its probability of occurrence
- Categorization of noise based on probability distribution is very popular

1. Gaussian Noise

- Random noise that enters a system can be modelled as a Gaussian or normal distribution
- Gaussian noise affects both dark and light areas of image

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

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

- Where z represents gray level
- μ is the **mean of average** values of z
- σ is the standard deviation.
- The standard deviation squared, σ^2, is known as **variance** of z.

Noise Modelling

2. Impulse Noise

- It is also known as Shot Noise, Salt and Pepper Noise, and Binary Noise
- It occurs mostly because of sensor and memory problem because of which pixels are assigned incorrect maximum values

PDF of impulse noise

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

$z \rightarrow$ a & $b \rightarrow$ grey levels

$P_a = P_b = 0$ unipolar Noise

3. Poisson Noise

- This type of noise manifests as a random structure or texture in images
- It is very common in X-ray images

PDF

$$P(z) = \frac{(np)^z}{z!} e^{-np}$$

$n \rightarrow$ total no. of pixels

$p \rightarrow$ Ratio of noise pixels to the total no. of pixels

Noise Modelling

4. Exponential Noise

- This type of noise occurs mostly due to the illumination problems
- ✓ It is present in laser imaging

PDF

$$P(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

✓ Mean $\rightarrow \frac{1}{a}$

✓ Variance $\rightarrow \frac{1}{a^2}$

5. Gamma Noise

- This type of noise also occurs mostly due to the illumination problems

PDF

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

* Mean $\rightarrow \frac{b}{a}$

* Variance $\rightarrow \frac{b}{a^2}$

Noise Modelling

6. Rayleigh Noise

- This type of noise is mostly present in range images
- Range images are mostly used in remote sensing applications

PDF

$$P(z) = \begin{cases} \frac{2}{b}(z-a)\, e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{otherwise} \end{cases}$$

* Mean $\rightarrow a + \sqrt{\frac{\pi b}{4}}$
* Variance $\rightarrow \frac{b(4-\pi)}{4}$

7. Uniform Noise

- It is also very popular noise occurs in images where different values of noise are equally probable
- It occurs because of Quantization noise

$a \rightarrow$ min grey value
$b \rightarrow$ max gray value

PDF

$$P(z) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq f(x,y) \leq b \\ 0 & \text{otherwise} \end{cases}$$

* Mean $\rightarrow a+b/2$
* Variance $\rightarrow (b-a)^2/12$

Noise Modelling

Types of Noise based on Correlation

- Statistical dependence among pixels is known as correlation
- If a pixel is independent of its neighbouring pixels, it is known as uncorrelated pixel otherwise it is known as correlated pixel

- Uncorrelated Noise is known as White Noise
- Mathematically for white noise, the noise power spectrum or power spectral density remains constant with frequency

- Characterization of colored noise is quite difficult because its origin is mostly unknown
- One popular colored noise is Pink Noise
- Its power spectrum is not constant, rather it is proportional to reciprocal of frequency
- This is also known as 1/f or Flicker Noise

Noise Modelling

Types of Noise based on Nature

Based on the nature of its presence, noise can be modelled in two ways

1. Additive Noise

- In this case an image can be perceived as the image plus noise
- This is a linear problem

Perceived image,

$$g(x,y) = \underbrace{f(x,y)}_{\text{i/p image}} + \underbrace{n(x,y)}_{\text{noise}}$$

2. Multiplicative Noise

- It can be modelled as multiplicative process
- Speckle noise is most encountered multiplicative noise in image processing
- It is mostly present in medical images
- It can be modelled as pixel value multiplied by the random value

$$I = f(x,y) + [f(x,y) \times N_g]$$

N_g: Random noise having a zero mean gaussian PDF

Noise Modelling

Types of Noise based on Source

Noise based on source, commonly encountered in image processing are:

- ❑ Quantization Noise
- ❑ Photon Noise

- ❑ Quantization Noise occurs due to a difference between the actual and allocated values
- ❑ It is inherent in the quantization process

- ❑ Photon Noise occurs due to the statistical nature of electromagnetic waves
- ❑ Generation of photon is not constant because of statistical variation
- ❑ This causes variation in photon count which is known as photon noise
- ❑ It is present in many medical images

Graph representation for Uniform Noise Model

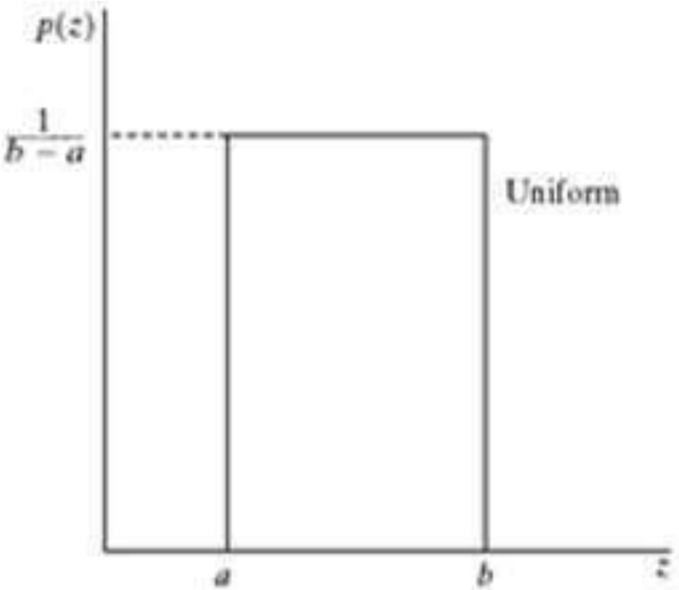
PDF of Uniform Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}.$$



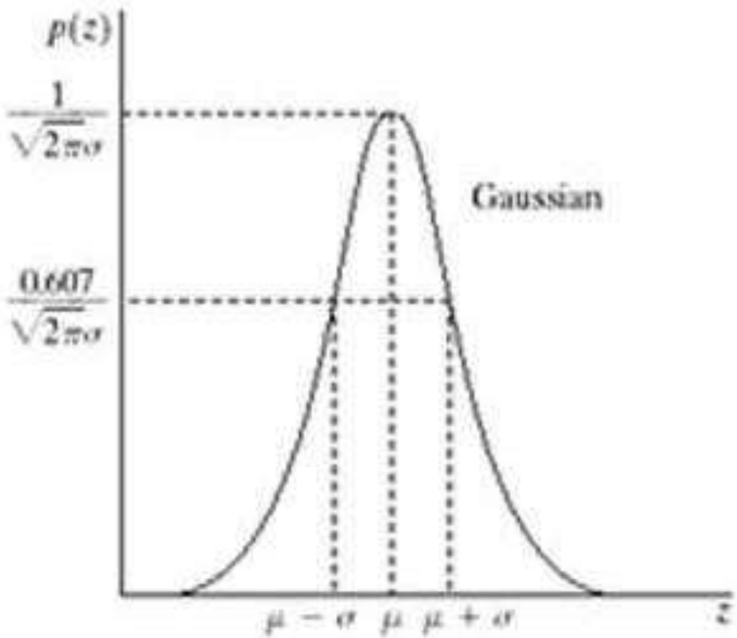
Plot of function

Graph representation for Gaussian / Normal Noise Model

1. Most frequently used.
2. PDF of Gaussian random variable z is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ → Standard Deviation of z
- σ^2 → Variance of z



Plot of function

When z is defined by this equation then

- About 70% of its values will be in the range [(μ - σ),(μ + σ)] and
- About 95% of its values will be in the range [(μ - 2σ),(μ + 2σ)]

Graph representation for Rayleigh Noise Model

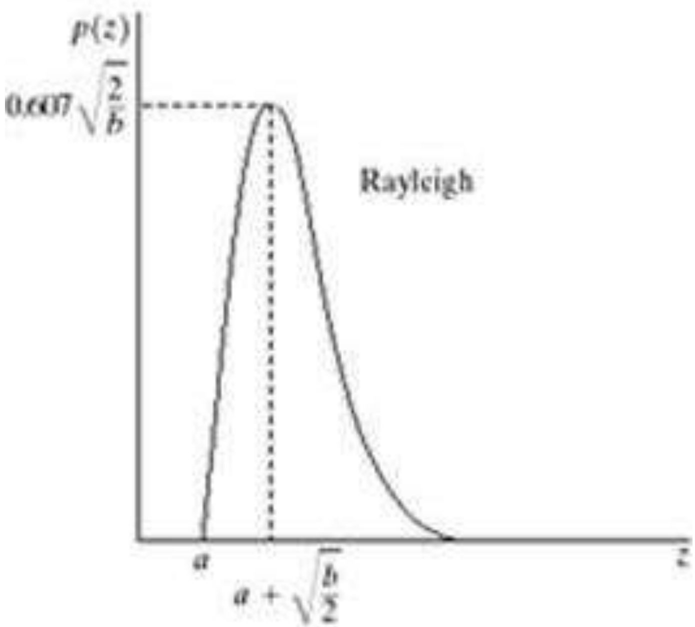
PDF of Rayleigh Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

$$\mu = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$



Plot of function

- Basic shape of this density is skewed to the right.
- Quite useful for approximating skewed histograms.

Graph representation for Erlang (Gamma) Noise Model

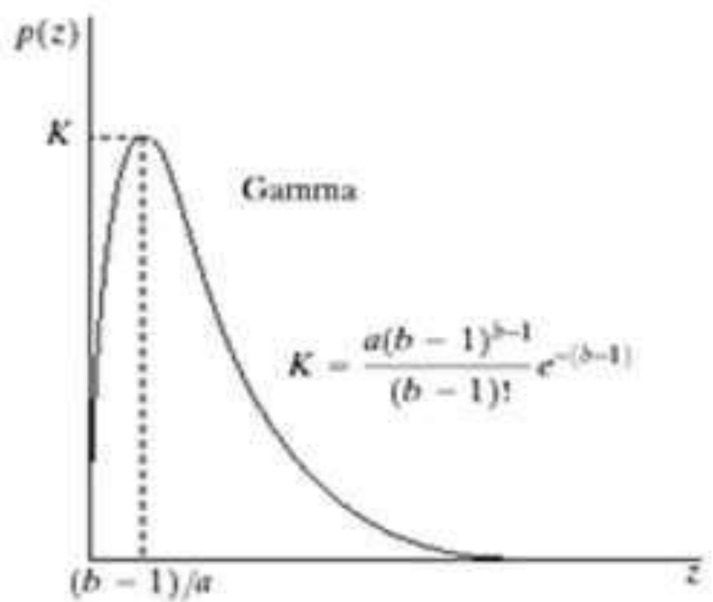
PDF of Erlang Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ^2 → Variance of z

- a > 0
- b = positive integer

$$\mu = \frac{b}{a} \qquad \sigma^2 = \frac{b}{a^2}.$$



Plot of function

- Above equation is also called Erlang Density
- If denominator is Gamma function $\Gamma(b)$ then it is called Gamma density

Graph representation for Exponential Noise Model

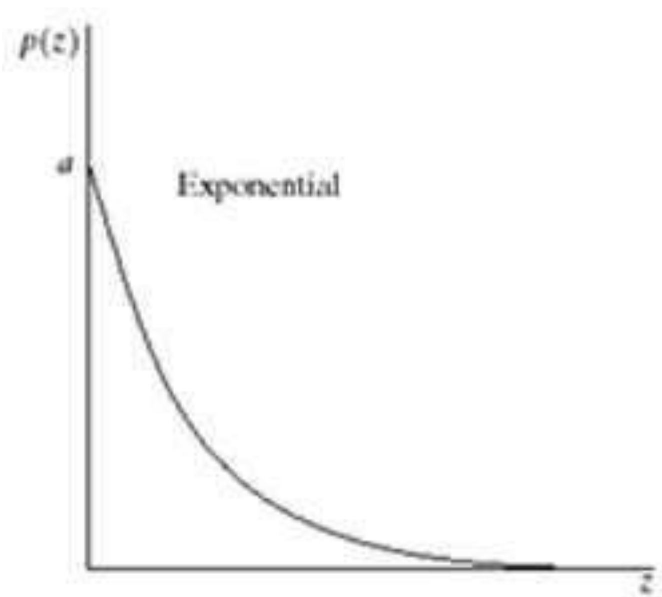
PDF of Exponential Noise is given by:

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

- z → Gray level
- μ → Mean of average value of z
- σ² → Variance of z
- a > 0

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}.$$



Plot of function

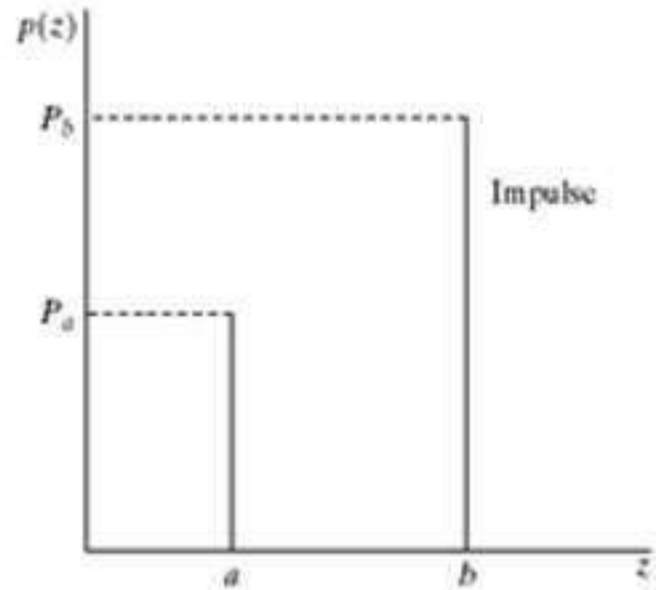
- Special case of Erlang Density Where b=1

Graph representation for Impulse (Salt & Pepper) Noise Model

PDF of Uniform Noise is given by:

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

- z → Gray level
- If b > a then b → light dot and a→ dark dot
- If either P_a or P_b = 0 → Unipolar Impulse Noise otherwise Bipolar Impulse Noise.



Plot of function

- If Neither probability is 0 and approximately equal then noise values will resemble salt & pepper granules randomly distributed over the image.
- Also referred as Shot and Spike Noise

Example

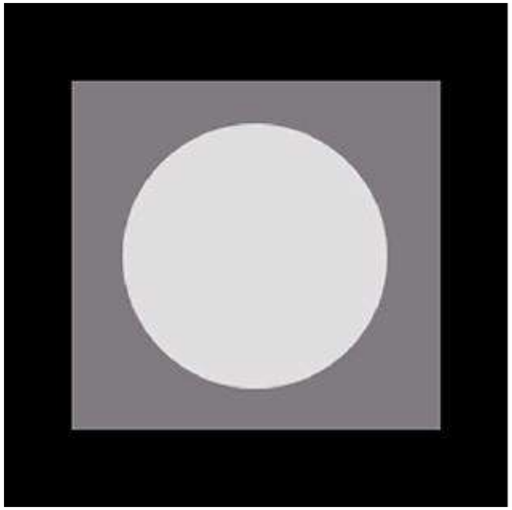
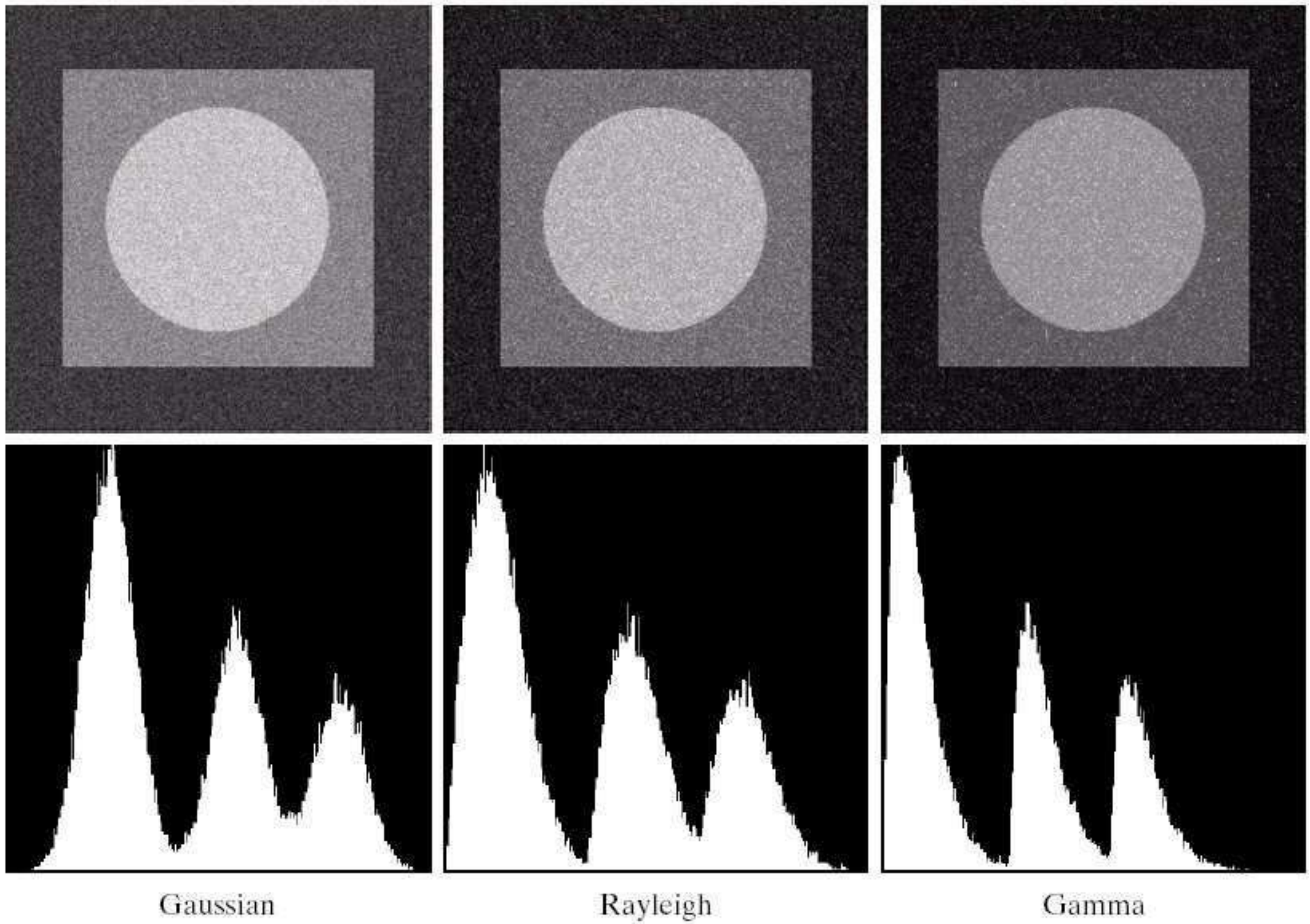


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

This test pattern is well-suited for illustrating the noise models, because it is composed of simple, constant areas that span the grey scale from black to white in only three increments. This facilitates visual analysis of the characteristics of the various noise components added to the image.



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.

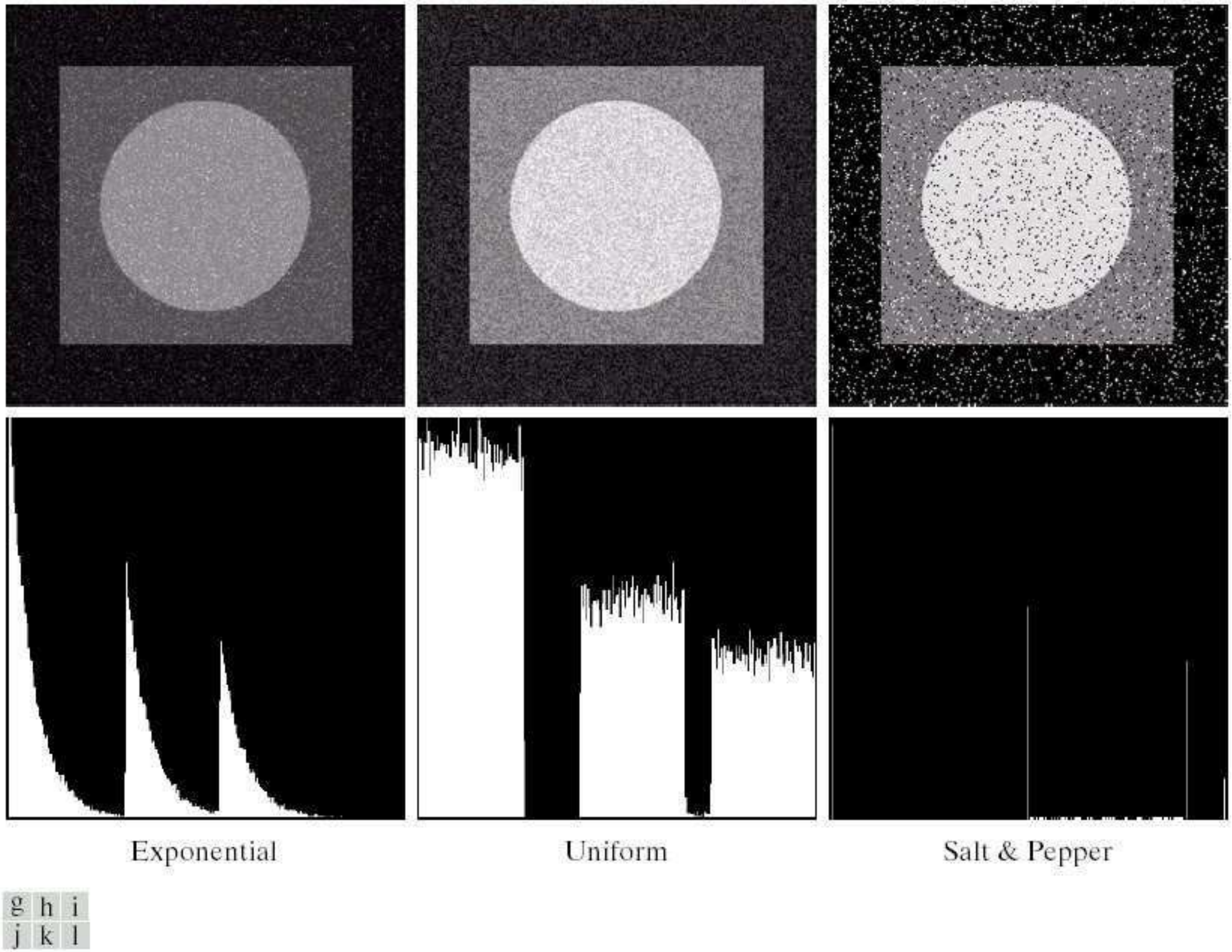


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

Periodic Noise

➢Typically arises due to electrical / Electro-mechanical interference during image acquisition.

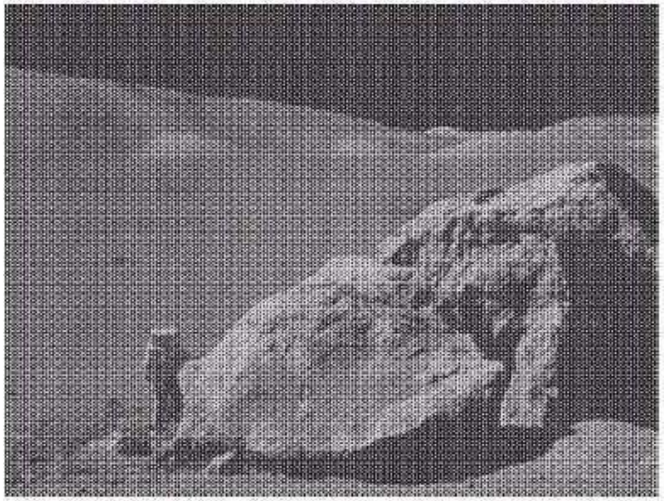
➢Spatially dependent noise.

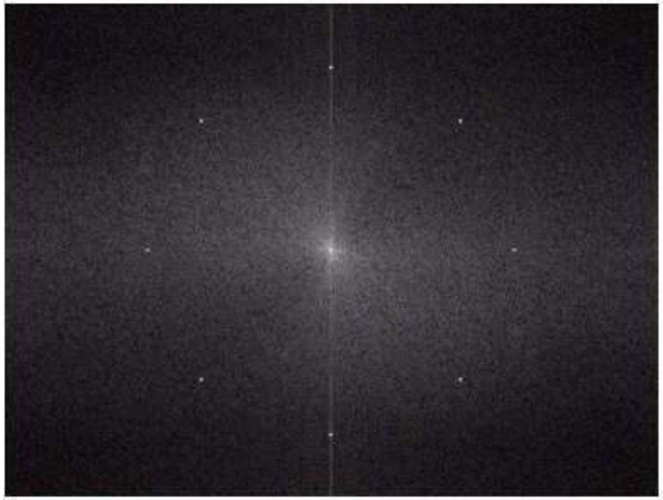
➢Can be reduced significantly via Frequency Domain Filtering.

➢Parameters can be estimated by inspecting the Frequency Spectrum of the image.

➢Periodic noise tend to produce frequency spikes

Image corrupted by Sinusoidal noise



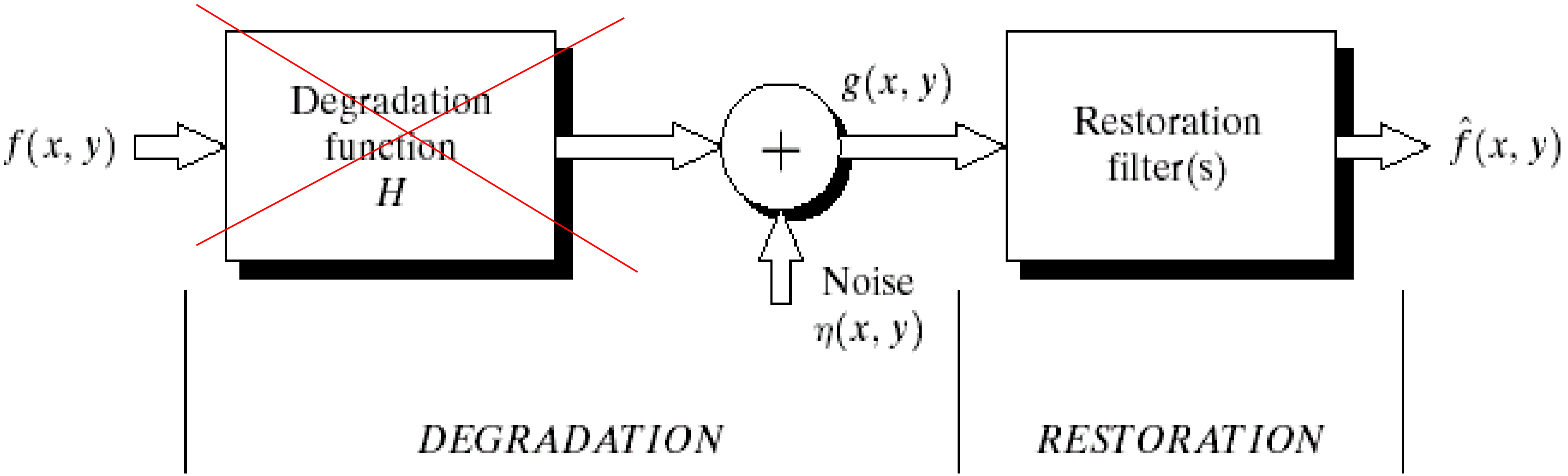


Spectrum (Each pair of conjugate impulses corresponds to one sine wave)

Restoration in the Presence of Noise Only using Spatial Filtering

- Mean filters
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contra-harmonic mean filter
- Order statistics filters
  - Median filter
  - Max and min filters
  - Mid-point filter
  - alpha-trimmed filters
- Adaptive filters
  - Adaptive local noise reduction filter
  - Adaptive median filter

Additive noise only



$$\begin{cases} g(x,y)=f(x,y)+\eta(x,y) \\ G(u,v)=F(u,v)+N(u,v) \end{cases}$$

Spatial filters for de-noising additive noise

- Skills similar to image enhancement
- Mean filters
- Order-statistics filters
- Adaptive filters

Mean filters

- Arithmetic mean

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

Window centered at (x,y)

- Geometric mean

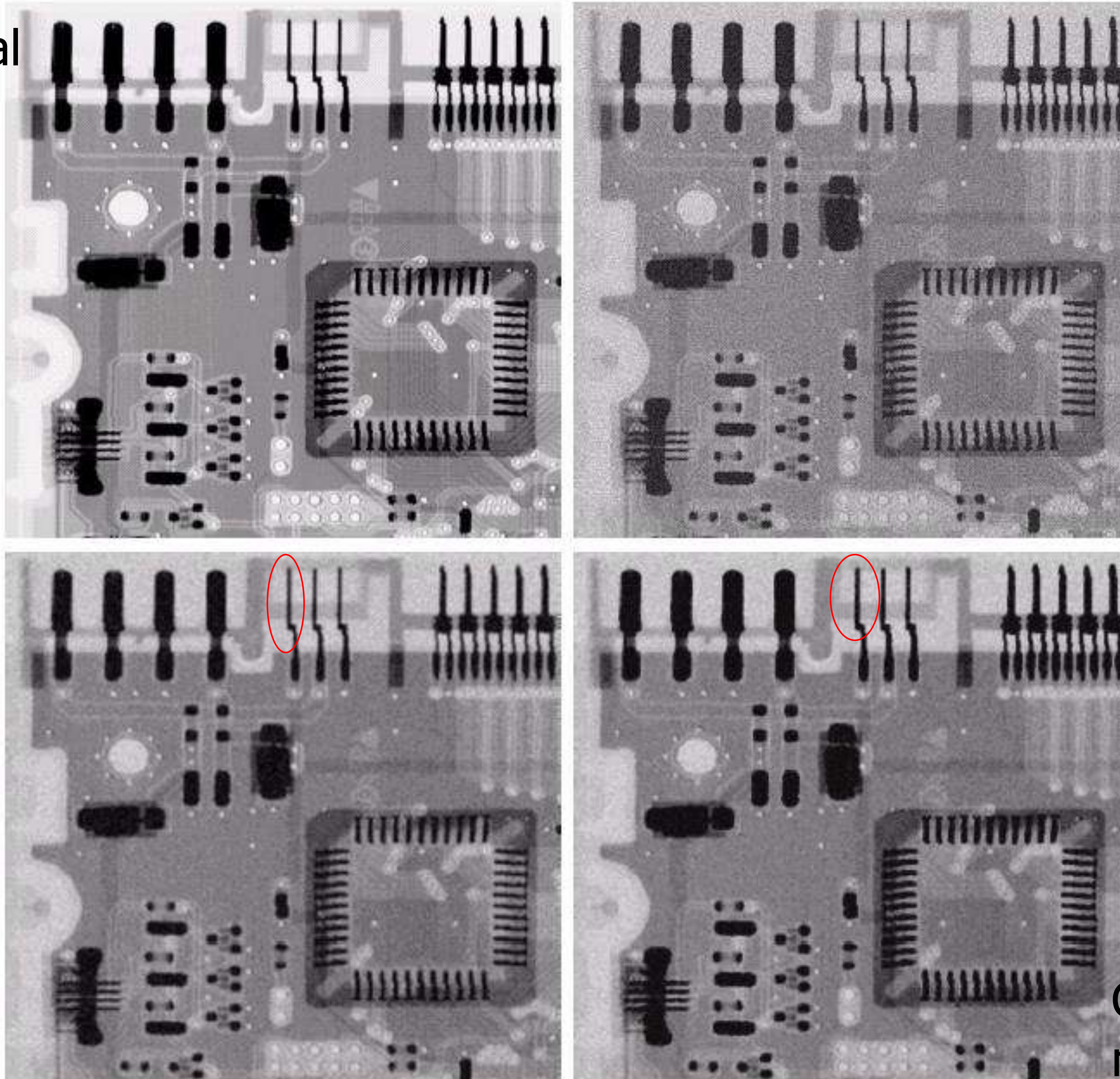
$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{1/mn}$$

original

Noisy Gaussian

$\mu=0$

$\sigma=20$



Arith. mean

Geometric mean

Mean filters (cont.)

- Harmonic mean filter

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

- Contra-harmonic mean filter

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

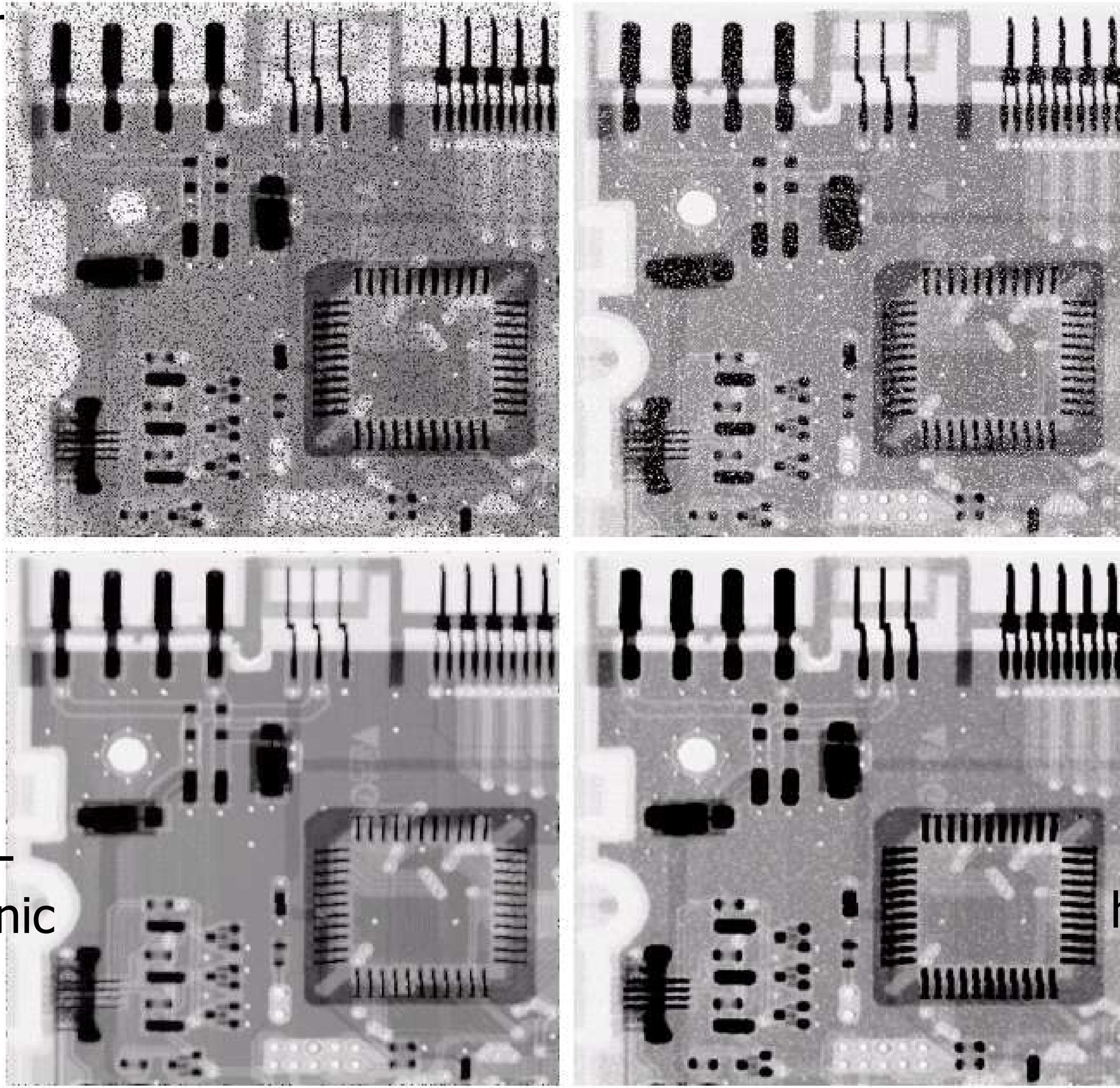
Q=-1, harmonic

Q=0, airth. mean

Q=+, ?

Pepper Noise

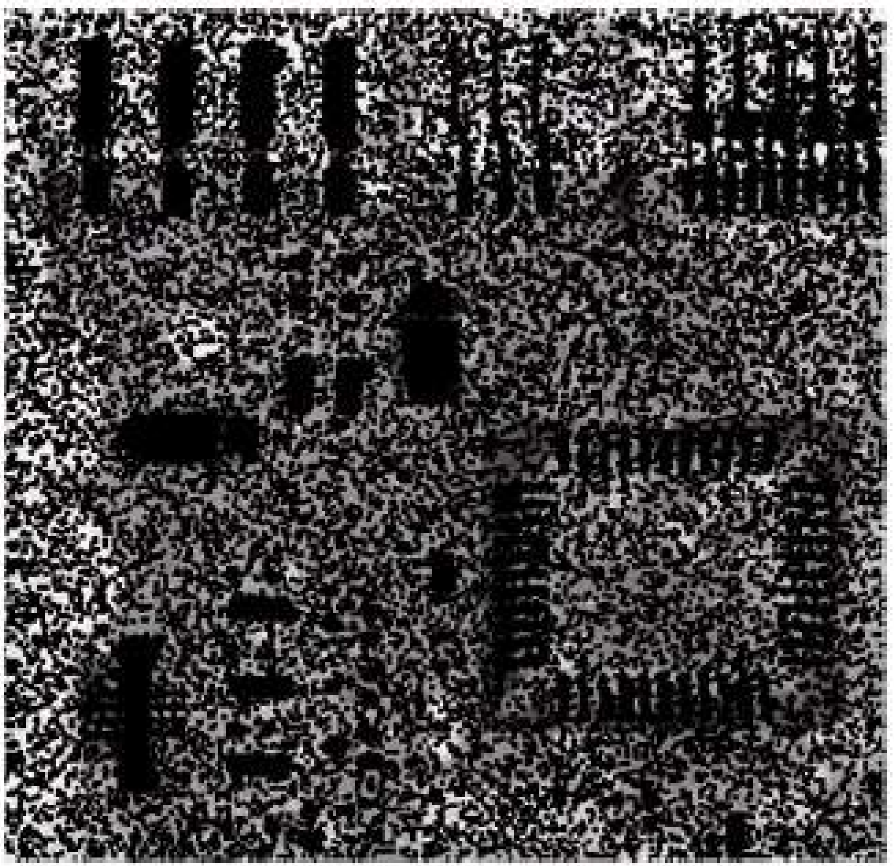
Salt Noise



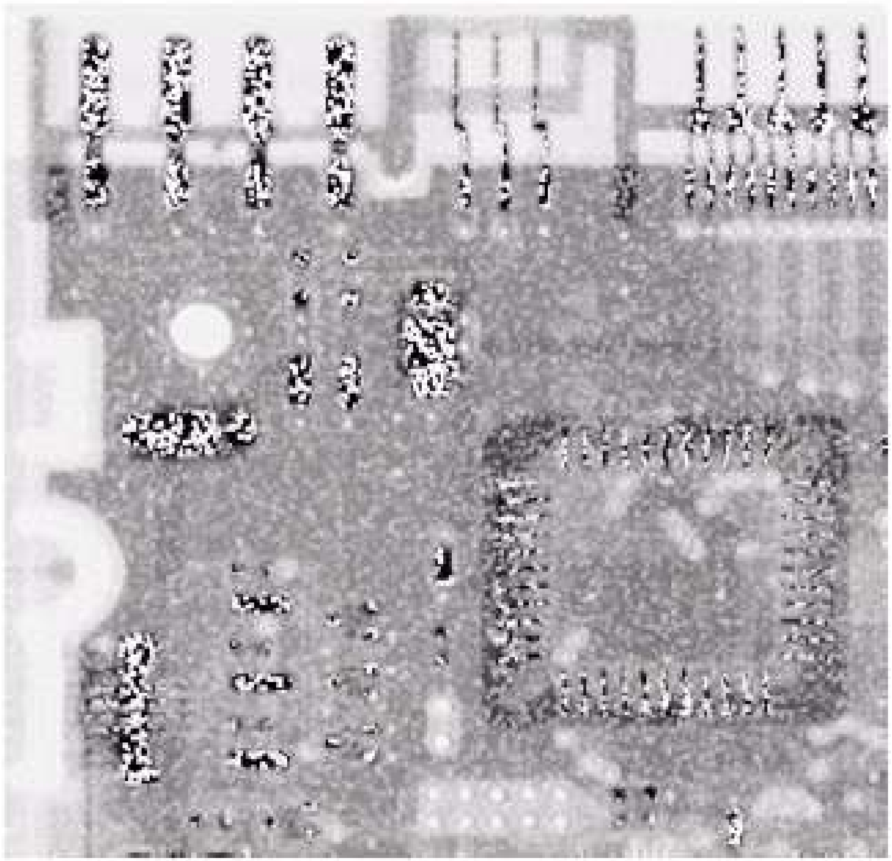
Contra-harmonic Q=1.5

Contra-harmonic Q=-1.5

Wrong sign in contra-harmonic filtering



Q=-1.5

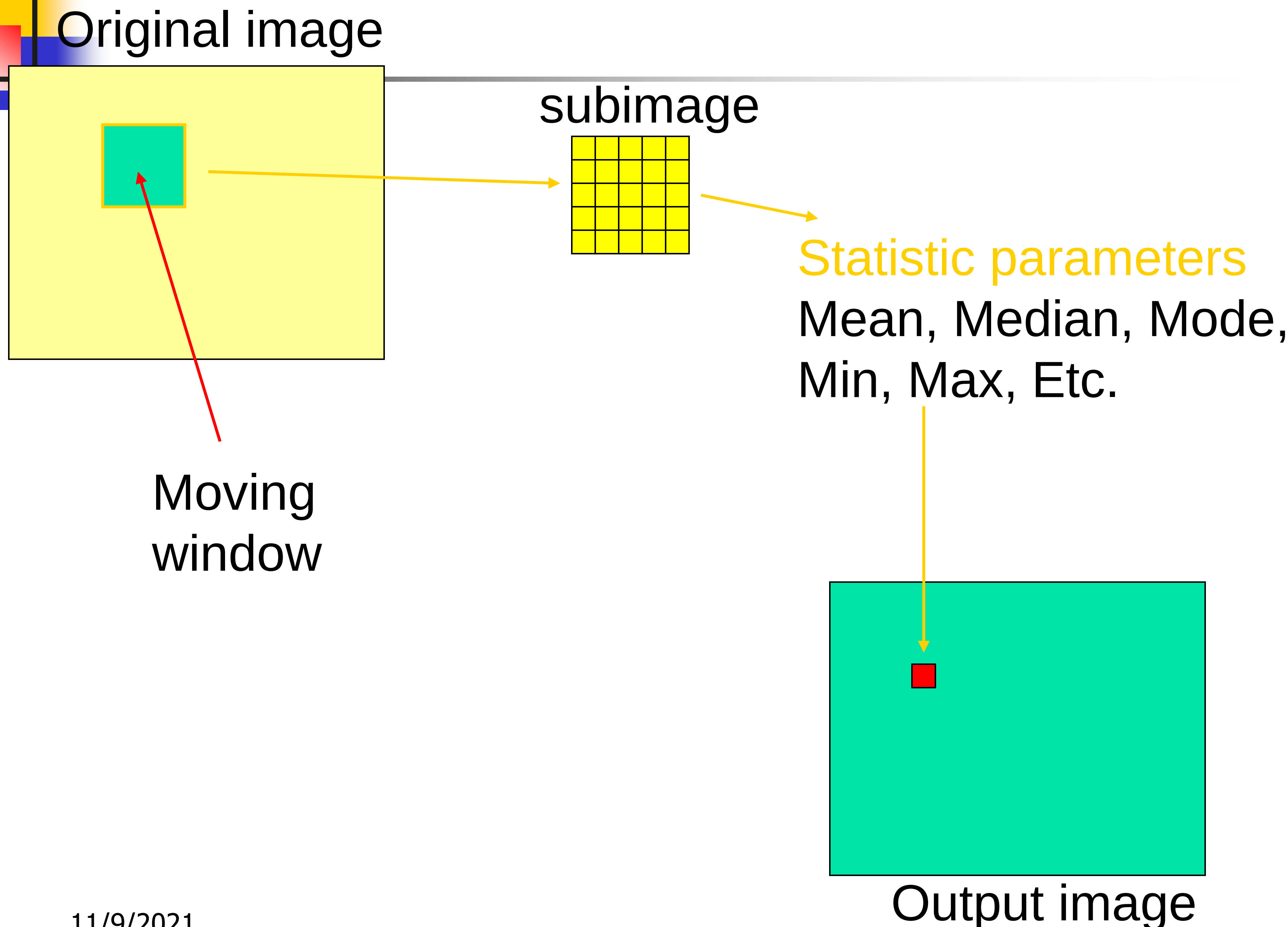


Q=1.5

Order-statistics filters

- Based on the ordering(ranking) of pixels
  - Suitable for unipolar or bipolar noise (salt and pepper noise)
- Median filters
- Max/min filters
- Midpoint filters
- Alpha-trimmed mean filters

Order-Statistic Filters



Order-statistics filters

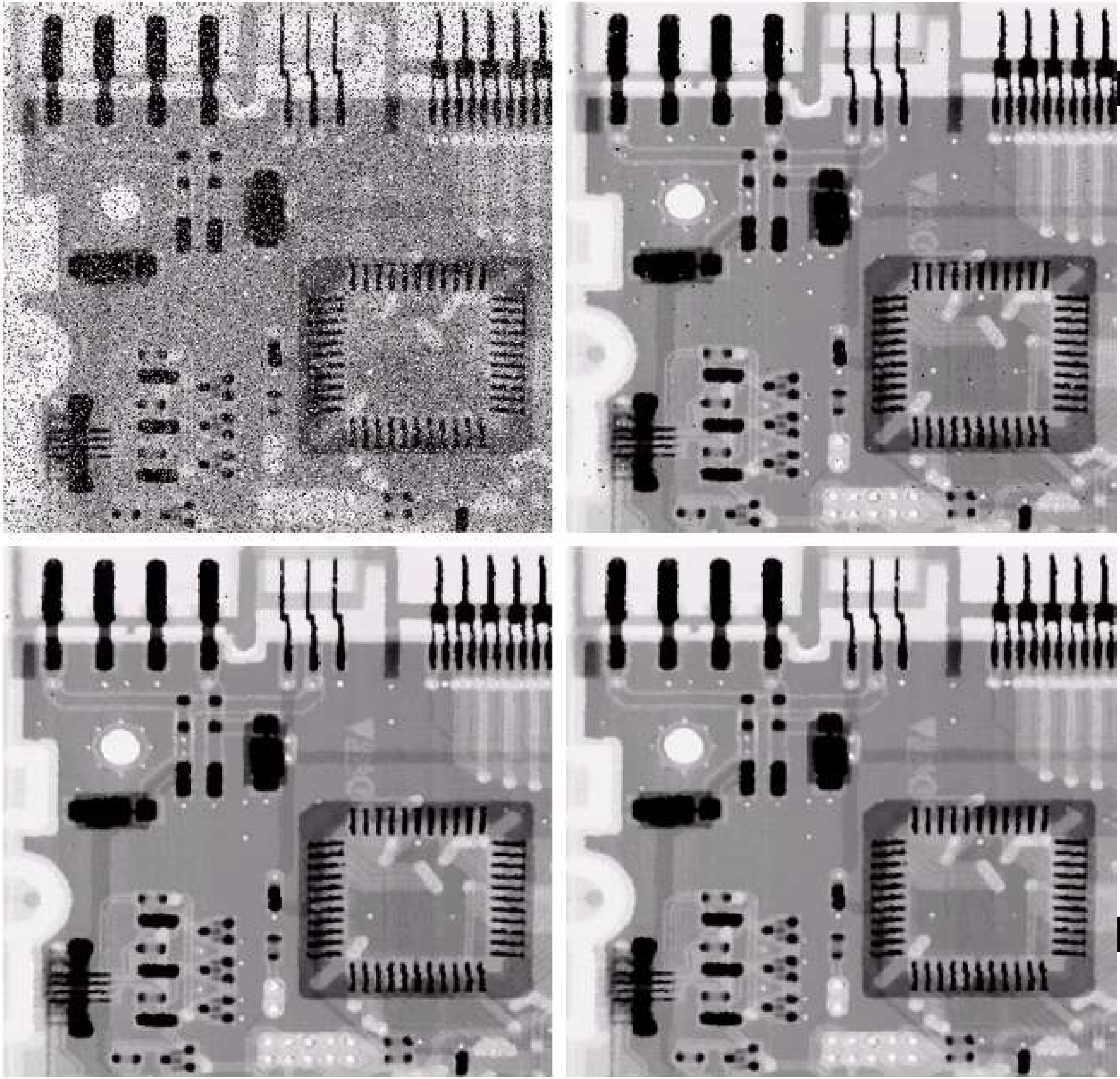
- Median filter

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

- Max/min filters

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

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



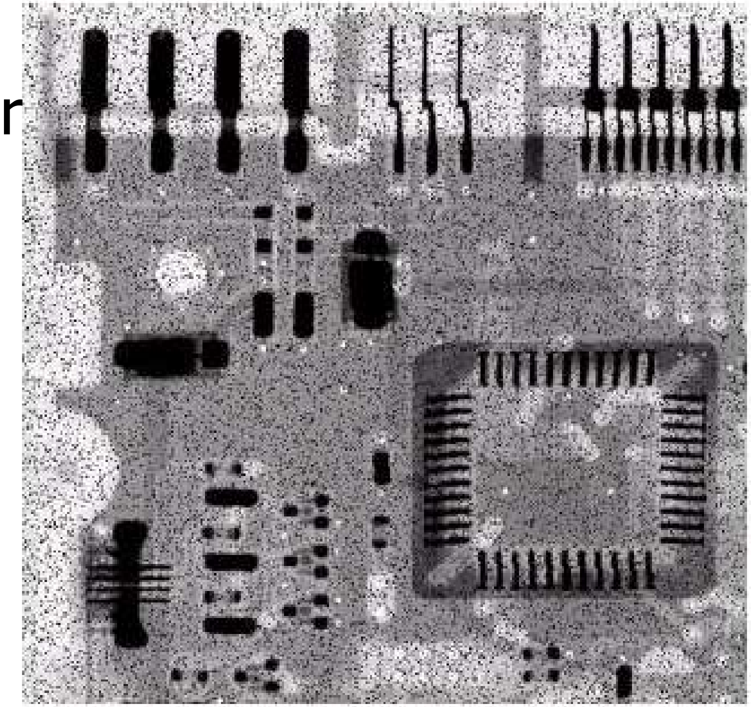
bipolar Noise $P_a = 0.1$ $P_b = 0.1$

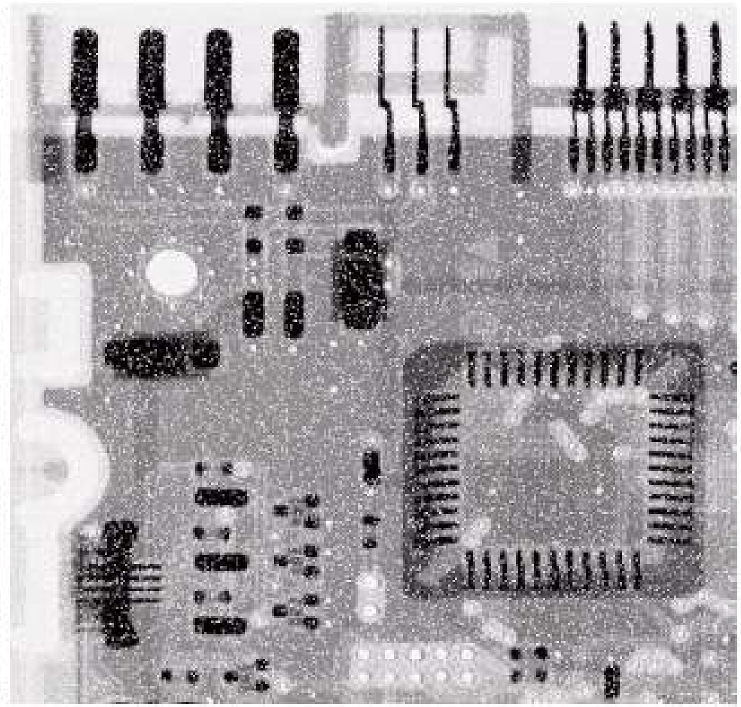
3x3 Median Filter Pass 1

3x3 Median Filter Pass 2

3x3 Median Filter Pass 3

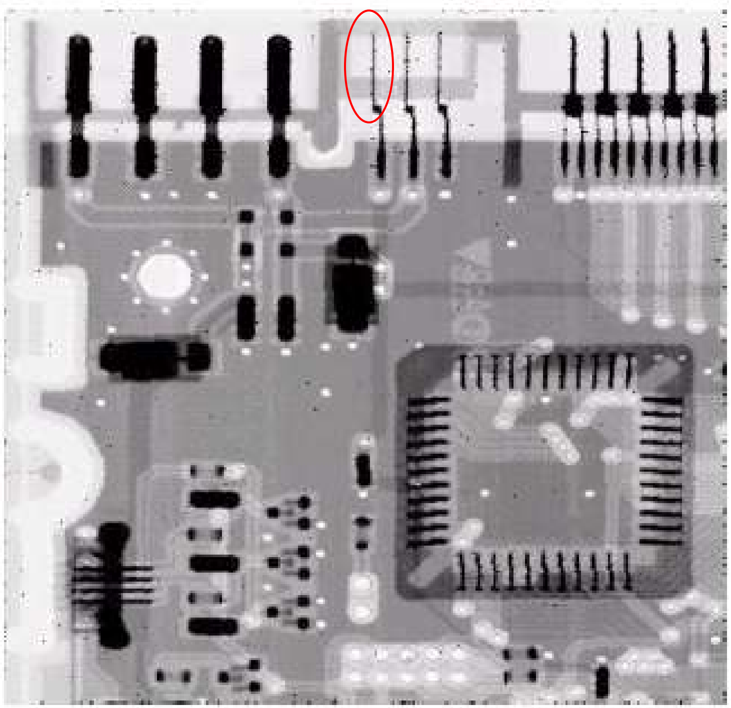
Pepper noise

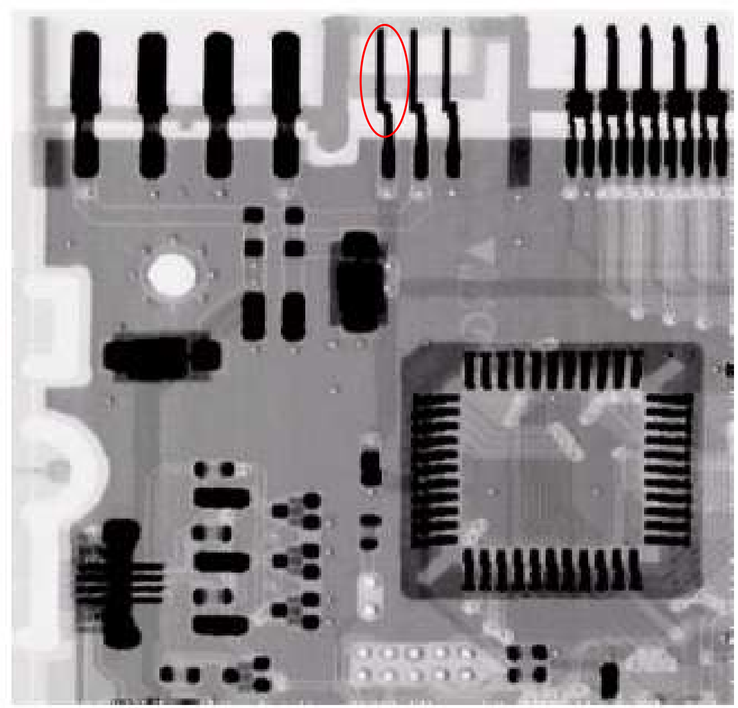




Salt noise

Max filter





Min filter

Order-statistics filters (cont.)

- Midpoint filter

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

- Alpha-trimmed mean filter
  - Delete the d/2 lowest and d/2 highest gray-level pixels

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

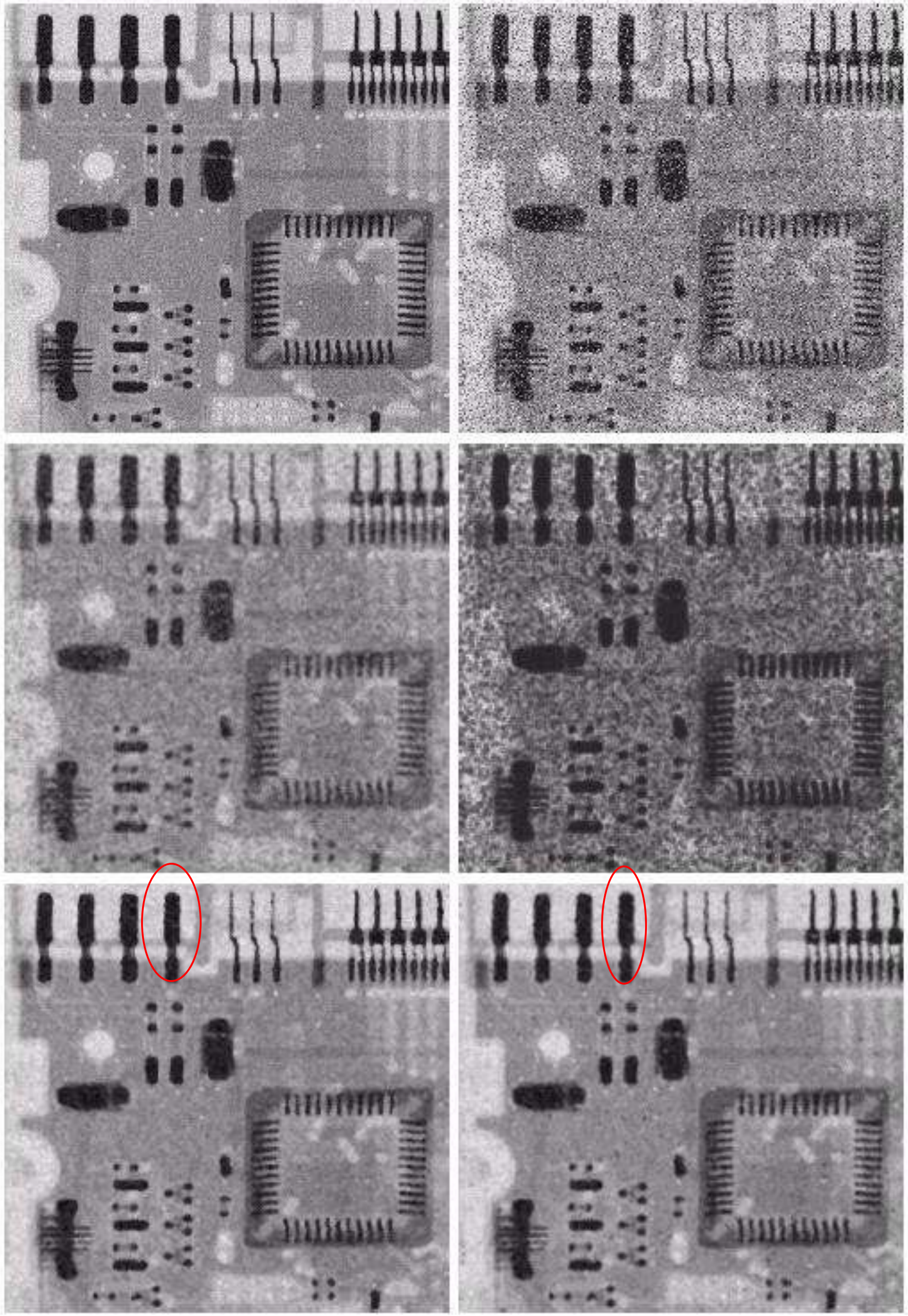
Middle (mn-d) pixels

Uniform noise

$\mu=0$

$\sigma^2=800$

Left +
Bipolar Noise
$P_a = 0.1$
$P_b = 0.1$



5x5
Arith. Mean
filter

5x5
Geometric
mean

5x5
Median
filter

5x5
Alpha-trim.
Filter
d=5

Adaptive filters

- Adapted to the behavior based on the statistical characteristics of the image inside the filter region S_{xy}
- Improved performance v.s increased complexity
- Example: Adaptive local noise reduction filter

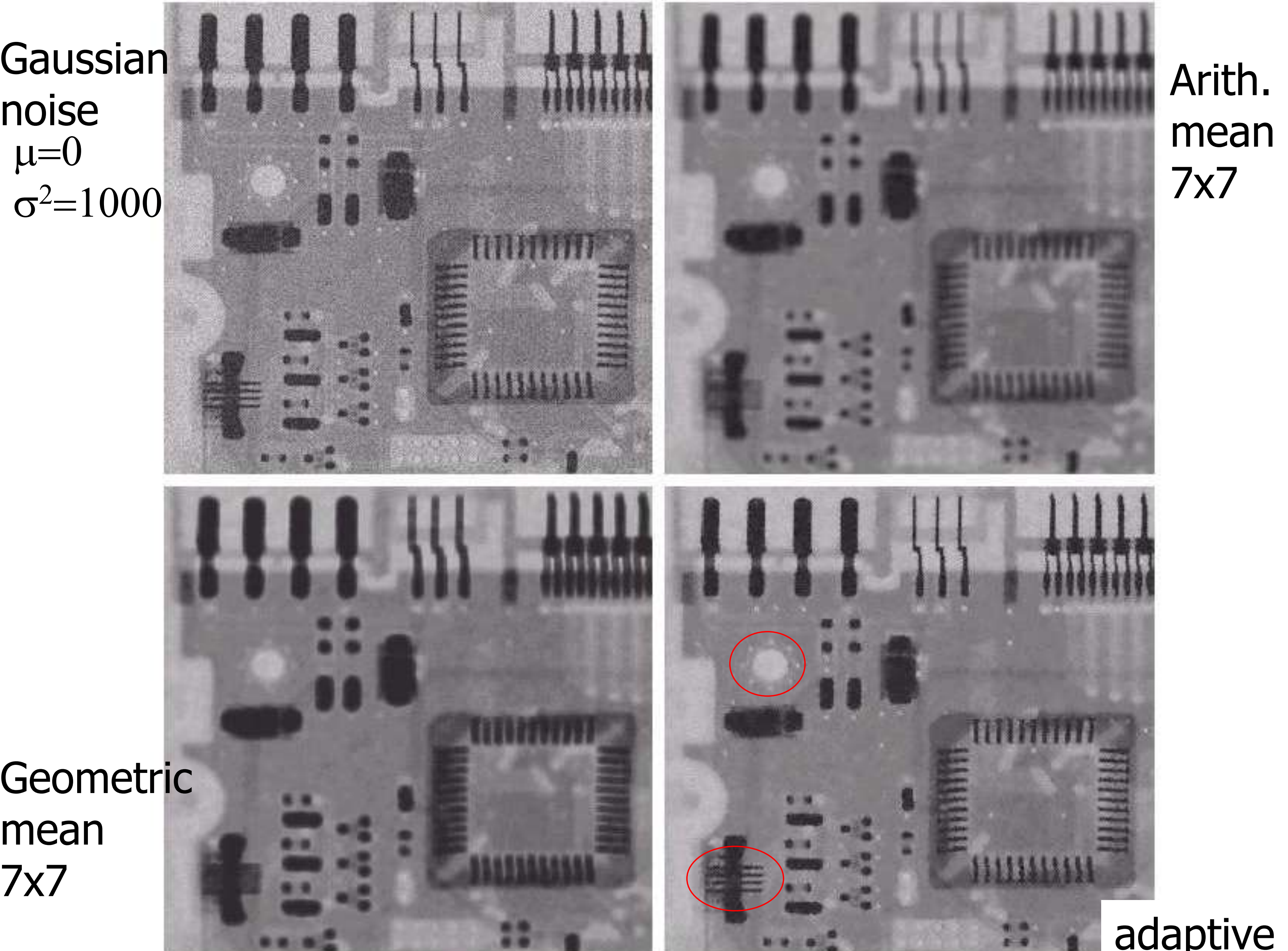
Adaptive local noise reduction filter

- Simplest statistical measurement
  - Mean and variance
- Known parameters on local region S_{xy}
  - g(x,y): noisy image pixel value
  - σ^2_η: noise variance (assume known a prior)
  - m_L : local mean
  - σ^2_L: local variance

Adaptive local noise reduction filter (cont.)

- Analysis: we want to do
  - If σ^2_η is zero, return g(x,y)
  - If $\sigma^2_L > \sigma^2_\eta$, return value close to g(x,y)
  - If $\sigma^2_L = \sigma^2_\eta$, return the arithmetic mean m_L
- Formula

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$



Gaussian noise
$\mu=0$
$\sigma^2=1000$
Arith. mean 7x7
Geometric mean 7x7
adaptive

Inverse filtering

- With the estimated degradation function H(u,v)

Unknown noise

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

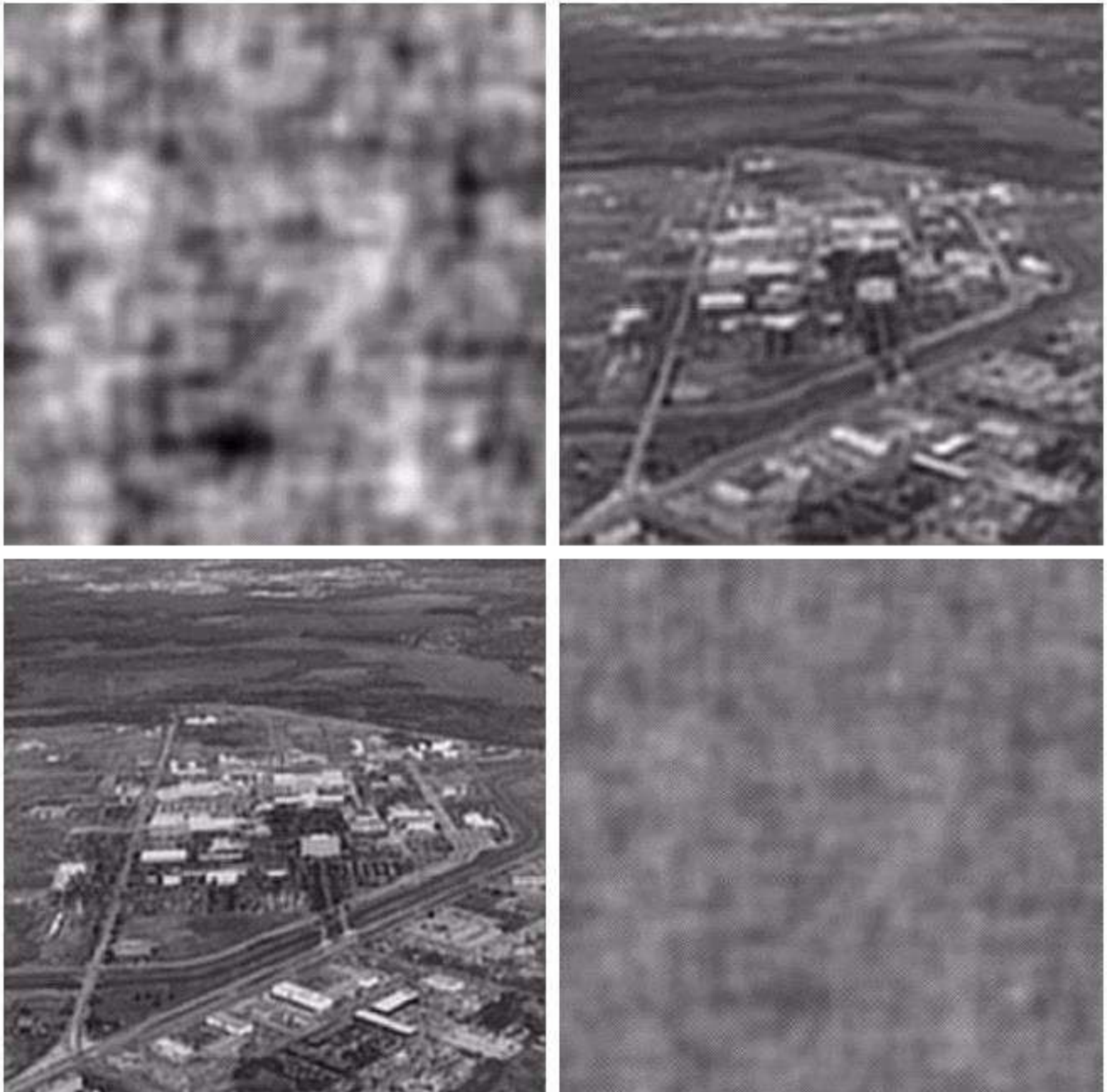
$$=> \quad \hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Estimate of original image

Problem: 0 or small values

Sol: limit the frequency around the origin

Full inverse filter for k=0.0025



Cut Outside 40%

Cut Outside 70%

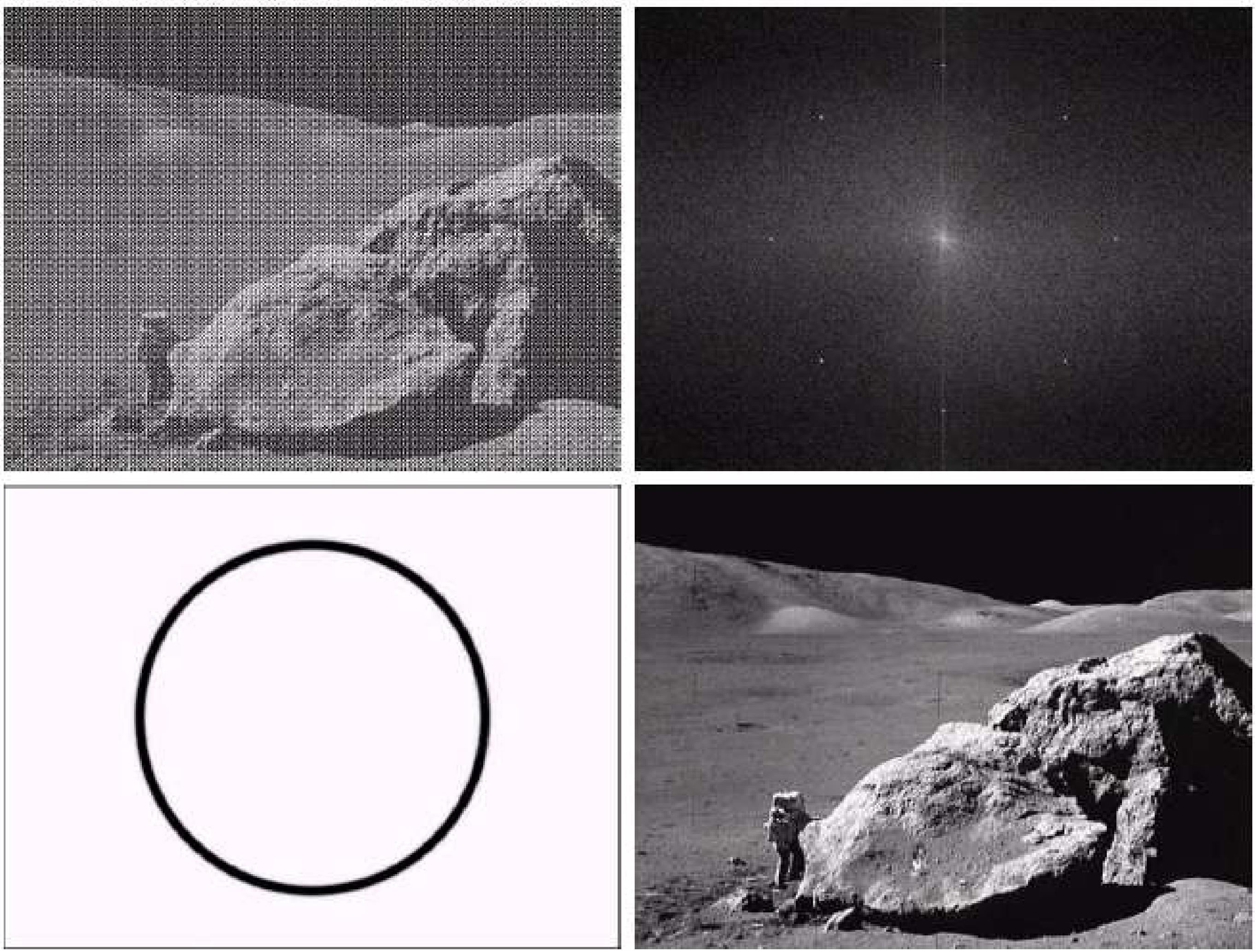
Cut Outside 85%

Periodic noise reduction using Frequency Domain Filtering

Periodic Noise Reduction by Frequency Domain Filtering

- Periodic noise is due to the electrical or electromechanical interference during image acquisition.
- Can be estimated through the inspection of the Fourier spectrum of the image.

Periodic Noise Reduction by Frequency Domain Filtering

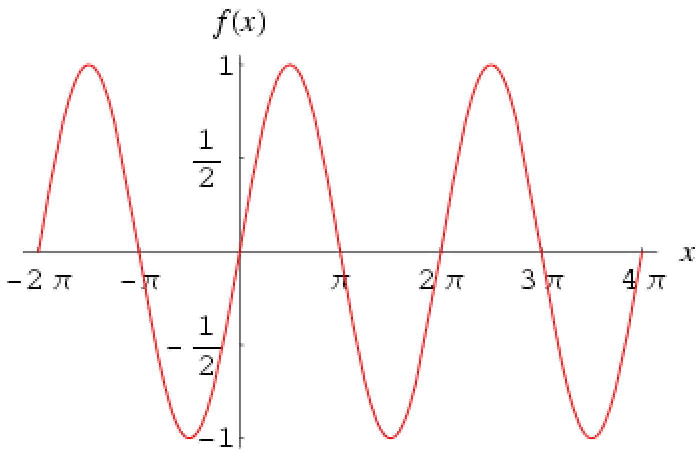


a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Periodic Function

- A function *f* is periodic with period *P* greater than zero if
  - $Af(x + P) = Af(x)$, where A denotes amplitude.



$f(x) = \sin x$, $P = 2\pi$, frequency$=1/2\pi$, $A=1$.

$f(x) = A\sin nx$, $P = 2\pi/n$, frequency$=n/2\pi$.

  - $n\uparrow$, frequency$\uparrow$.

Periodic noise reduction in frequency domain

- Pure sine wave
  - Appear as a pair of impulse (conjugate) in the frequency domain

$$\begin{cases} f(x,y) = A\sin(u_0 x + v_0 y) \\ F(u,v) = -j\dfrac{A}{2}\left[\delta\left(u - \dfrac{u_0}{2\pi}, v - \dfrac{v_0}{2\pi}\right) - \delta\left(u + \dfrac{u_0}{2\pi}, v + \dfrac{v_0}{2\pi}\right)\right] \end{cases}$$

Periodic noise reduction using Frequency Domain Filtering

- Bandreject filters
- Bandpass filters
- Notch filters
- Optimum notch filtering

Periodic Noise Reduction by Frequency Domain Filtering

- Bandreject filters
  - Remove or attenuate a band of frequencies.

$$H(u,v)=\begin{cases}1 & \text{if} \quad D(u,v)<D_0-W/2 \\ 0 & \text{if} \quad D_0-W/2\le D(u,v)\le D_0+W/2 \\ 1 & \text{if} \quad D(u,v)>D_0+W/2\end{cases}$$

- D_0 is the radius.
- $D(u, v)$ is the distance from the origin, and
- W is the width of the frequency band.

Butterworth and Gaussian Bandreject Filters

- Butterworth bandreject filter (order *n*)

$$H(u,v)=\frac{1}{1+\left[\dfrac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}}$$

- Gaussian band reject filter

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}$$

Bandreject Filters

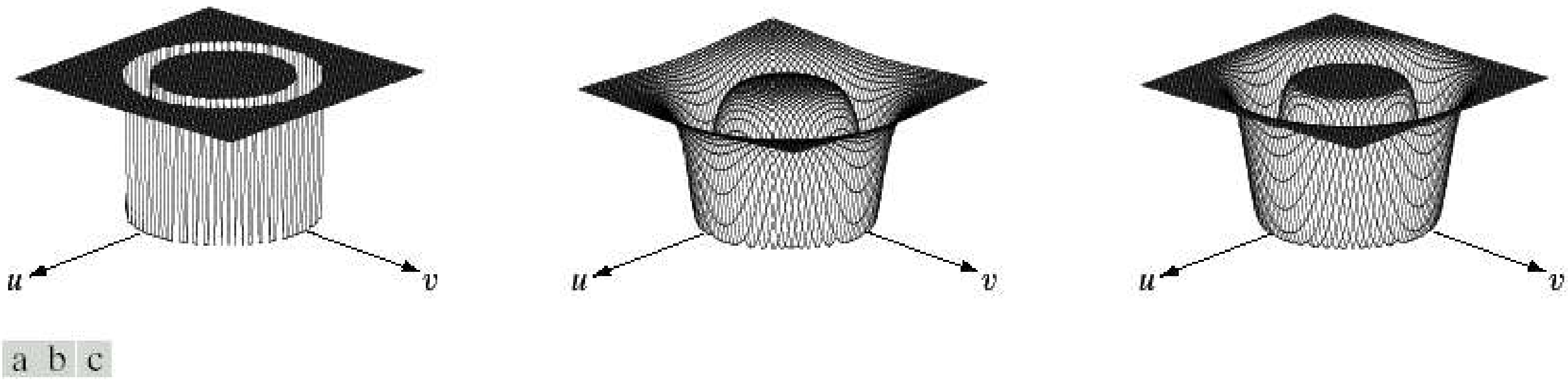


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandpass filter

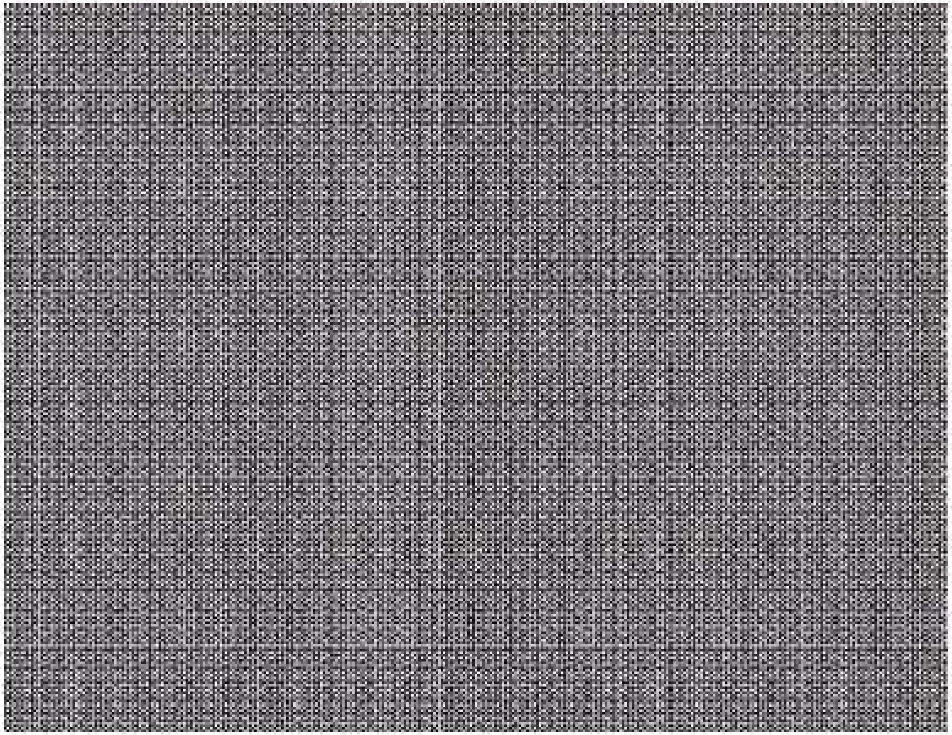
- Obtained form bandreject filter

  $$H_{bp}(u,v)=1-H_{br}(u,v)$$

- The goal of the bandpass filter is to isolate the noise pattern from the original image, which can help simplify the analysis of noise, reasonably independent of image content.

Result of The BandPass Filter

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Notch filters

- Notch filter rejects (passes) frequencies in predefined neighborhoods about a center frequency.

$$H(u,v)=\begin{cases}0 & if\ D_1(u,v)\le D_0 \quad or \quad D_2(u,v)\le D_0\\ 1 & otherwise\end{cases}$$

where

$$D_1(u,v)=\left[(u-M/2-u_0)^2+(v-N/2-v_0)^2\right]^{1/2}$$

$$D_2(u,v)=\left[(u-M/2+u_0)^2+(v-N/2+v_0)^2\right]^{1/2}$$

Notch filters

- Butterworth notch filter

$$H(u,v)=\frac{1}{1+\left[\dfrac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^n}$$

- Gaussian notch filter

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$

- Note that these notch filters will become highpass when $u_0=v_0=0$

Notch filters

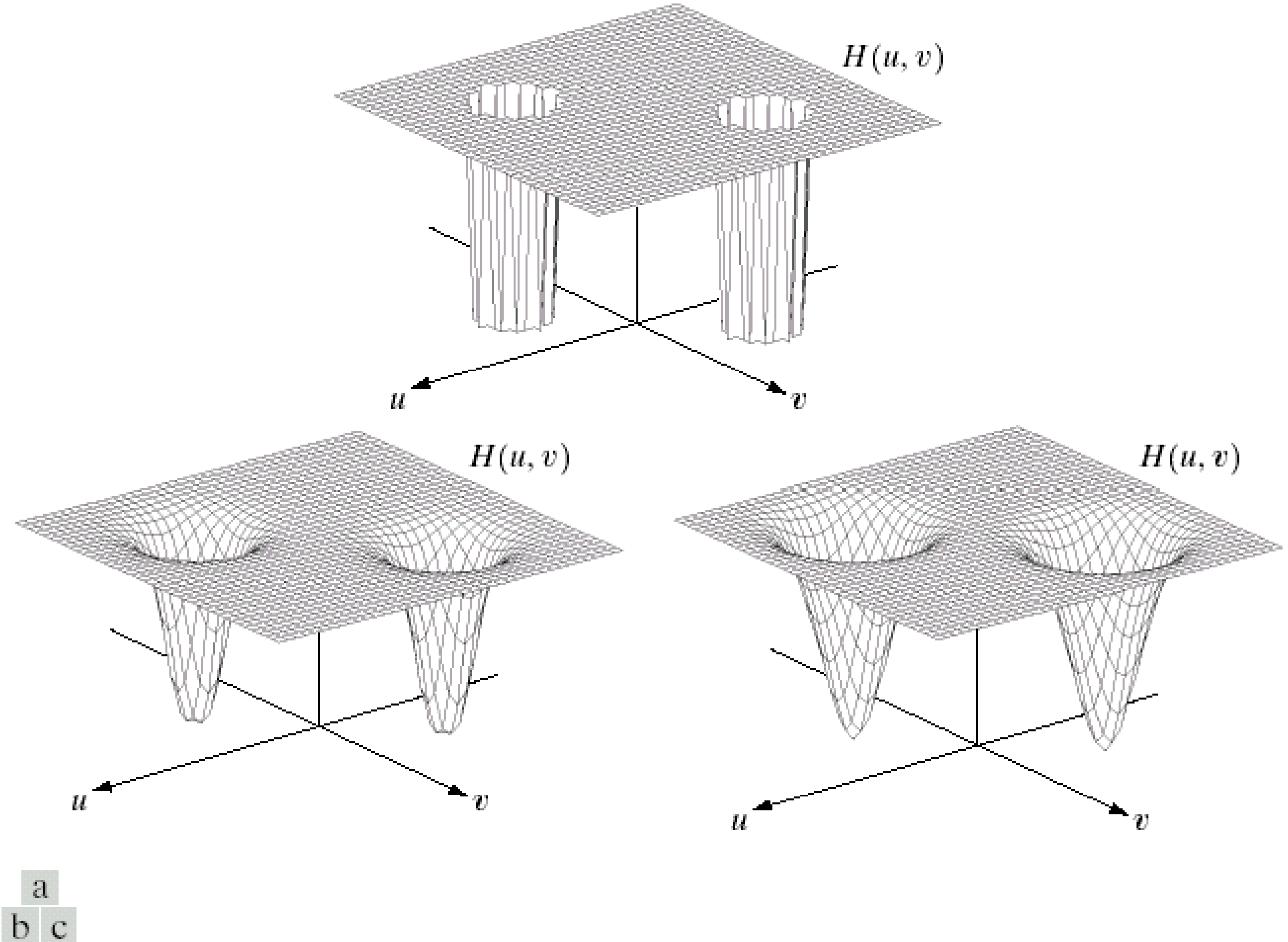
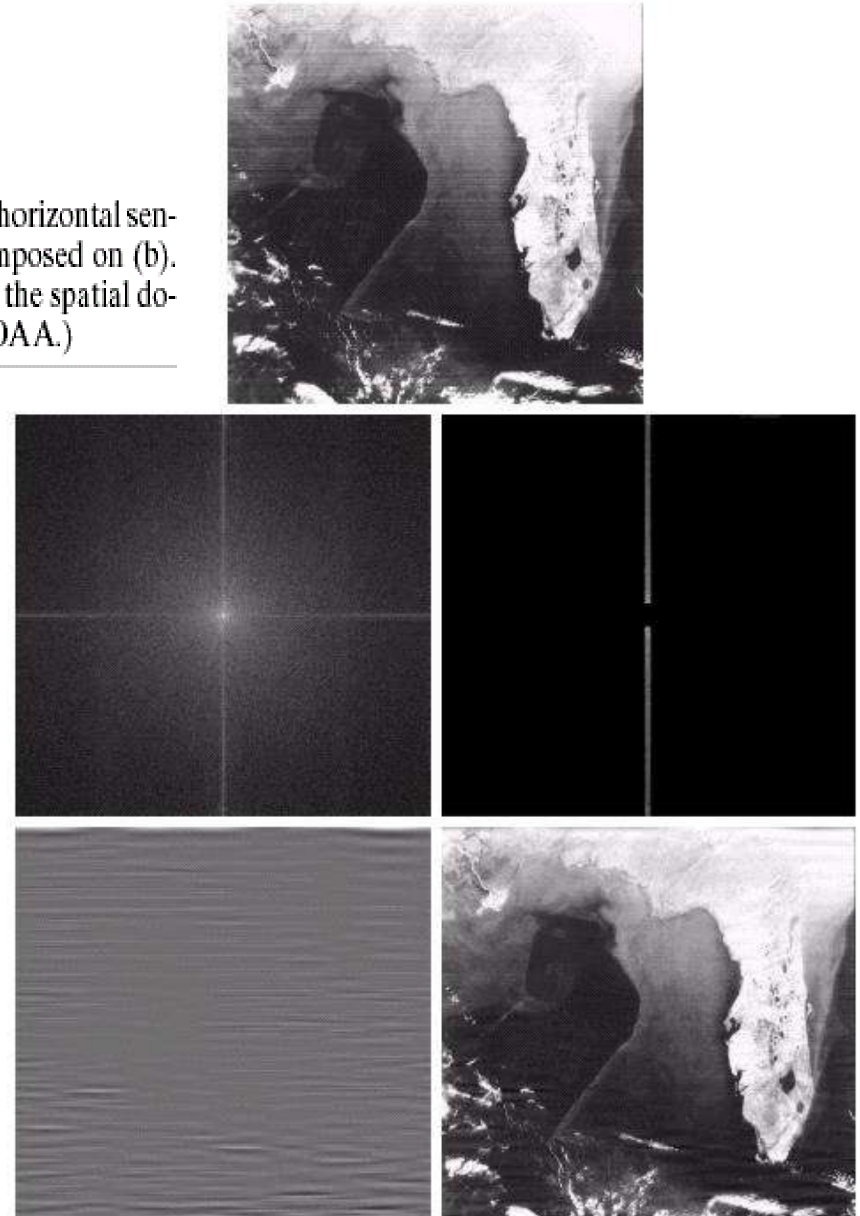


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)



Use 1-D Notch pass filter to find the horizontal ripple noise

Unit-4

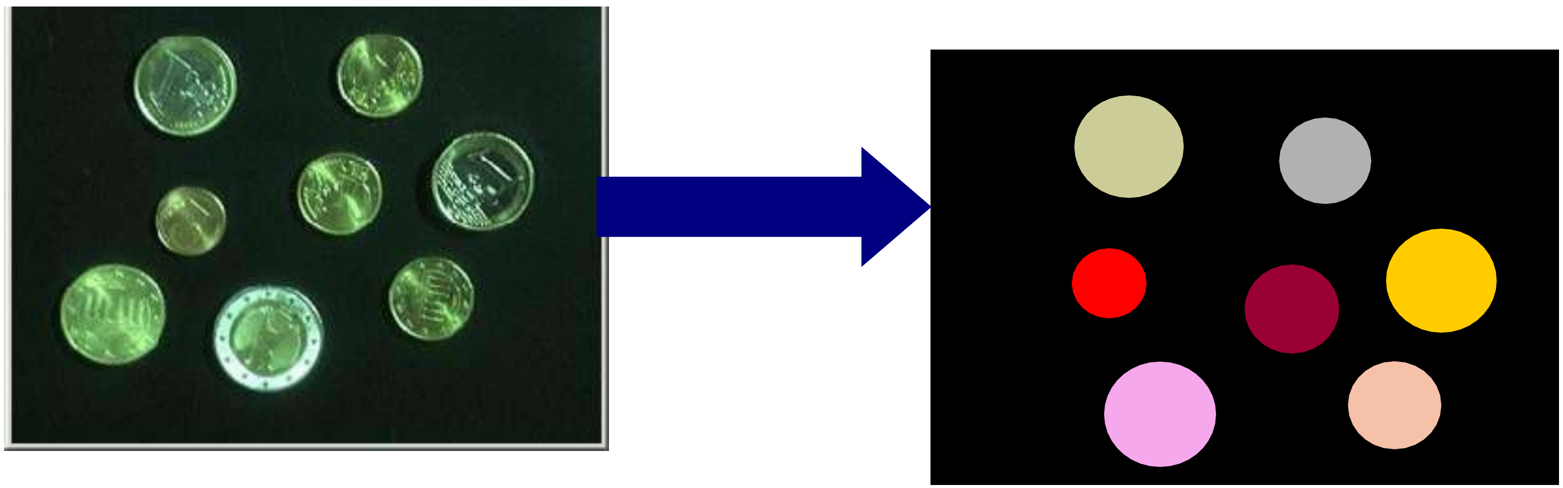
Segmentation

Introduction to Image Segmentation

Segmentation is the process of partitioning a digital image into multiple regions and extracting the meaningful region which is known as Region of Interest (ROI)

- Region of Interest (ROI) vary with applications
- In fact no single universal segmentation algorithm exists for segmenting the ROI in all images
- Therefor many segmentation algorithms need to apply and pick that algorithm which performs the best for given requirement

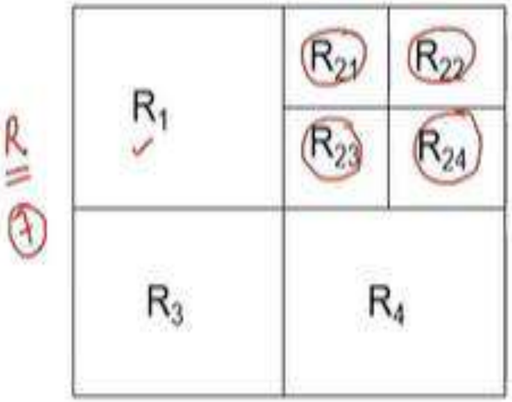
Image Segmentation



Introduction to Image Segmentation

Definition of Image Segmentation

An image can be portioned into many regions $R_1, R_2, R_3 \ldots R_n$



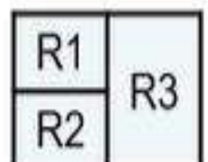
Example

5×5

10	10	20	20	20
10	10	20	20	20
10	10	20	20	20
30	30	20	20	20
30	30	20	20	20

R_1 R_2 R_3

10	10	20	20	20
10	10	20	20	20
10	10	20	20	20
30	30	20	20	20
30	30	20	20	20



Introduction to Image Segmentation

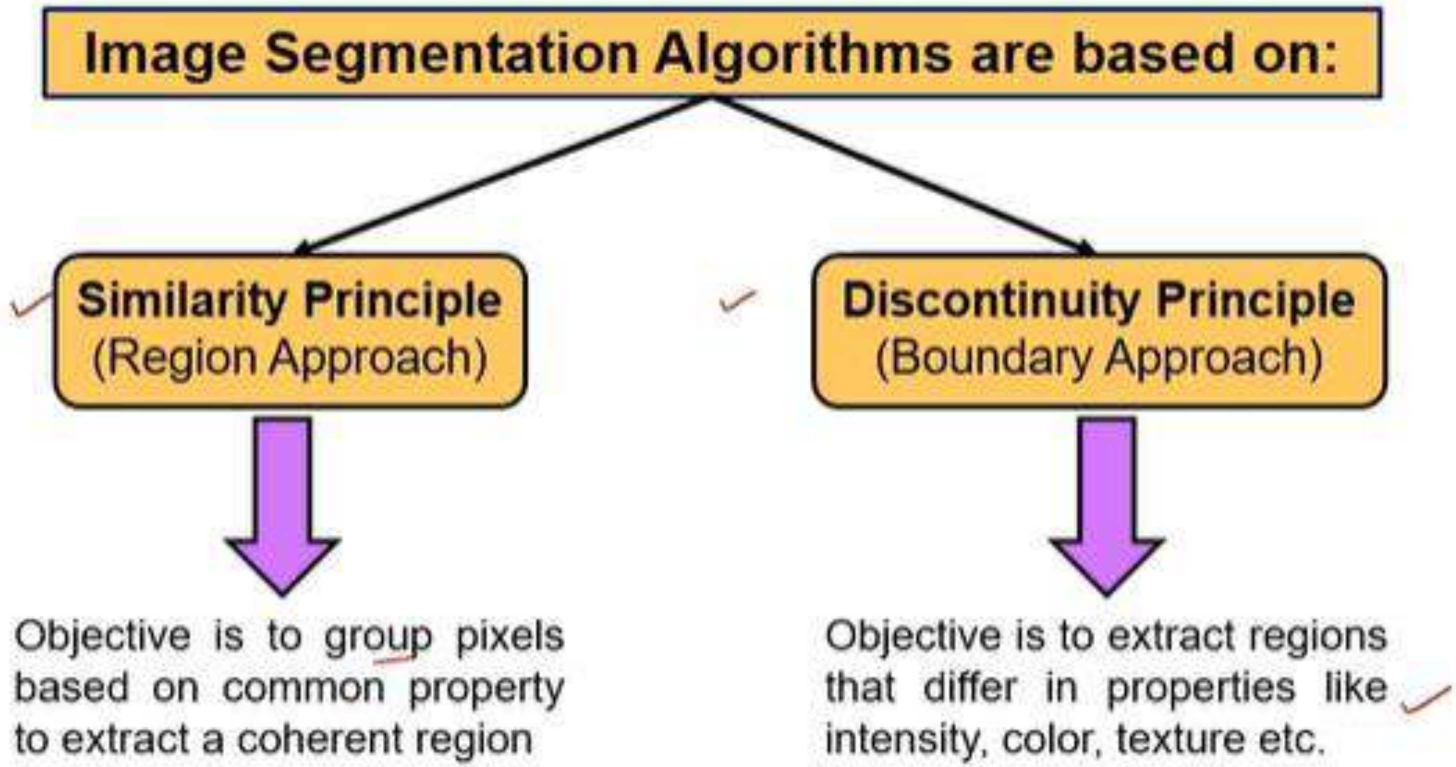
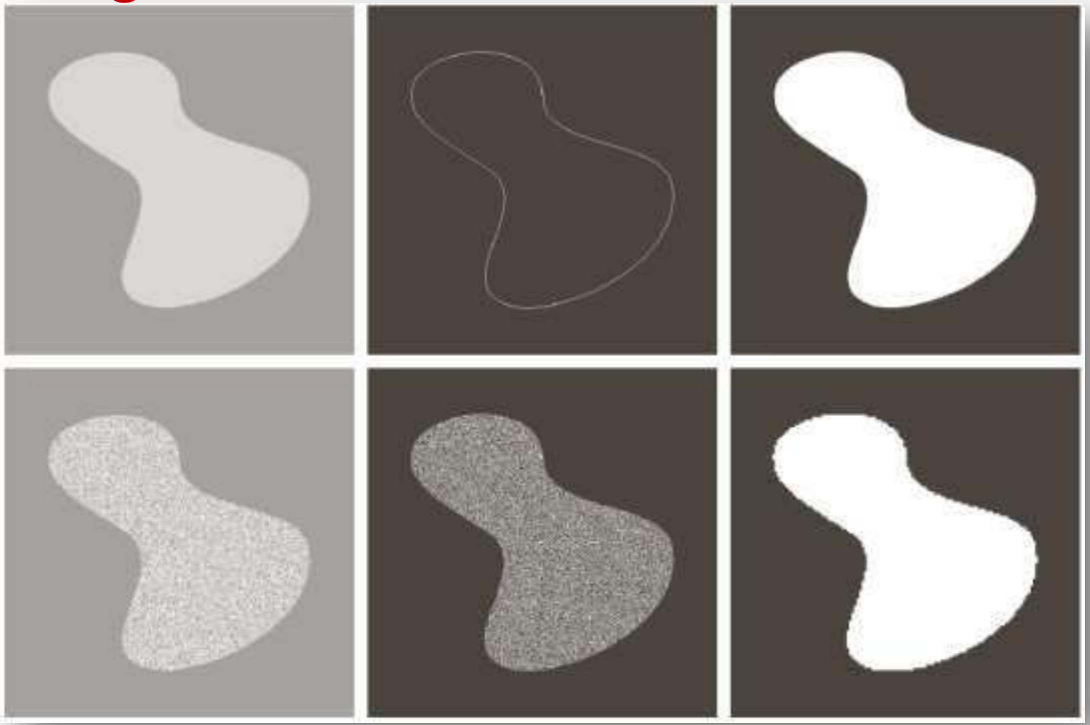


Image Segmentation

- Segmentation algorithms generally are based on one of two basis properties of intensity values
  - Discontinuity: to partition an image based on abrupt changes in intensity (such as edges)
  - Similarity: to partition an image into regions that are similar according to a set of predefined criteria.

Image Segmentation

- Image Segmentation

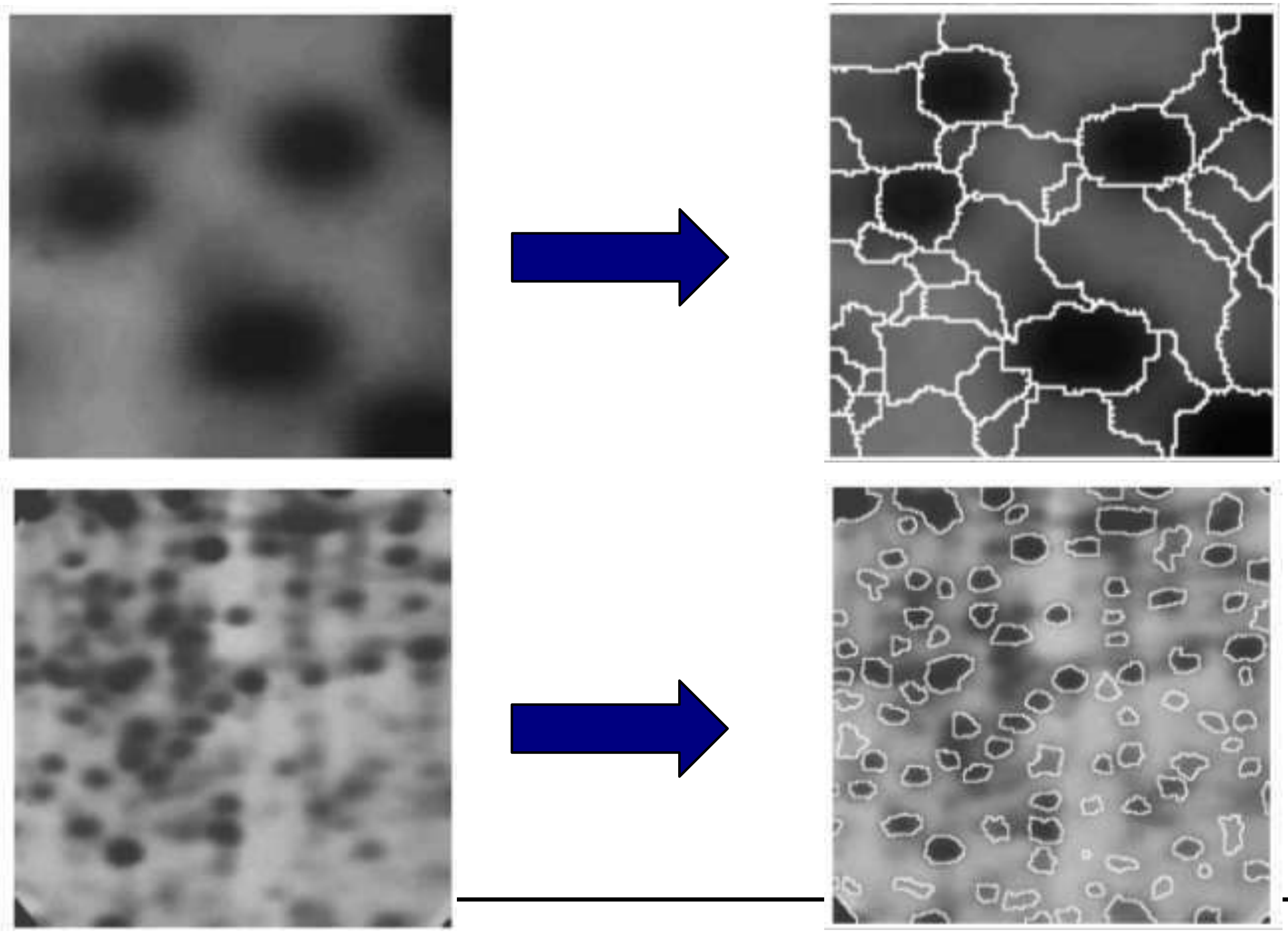


a b c
d e f

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.

Image Segmentation

- Image Segmentation



Introduction to Image Segmentation

Characteristics of Segmentation Process

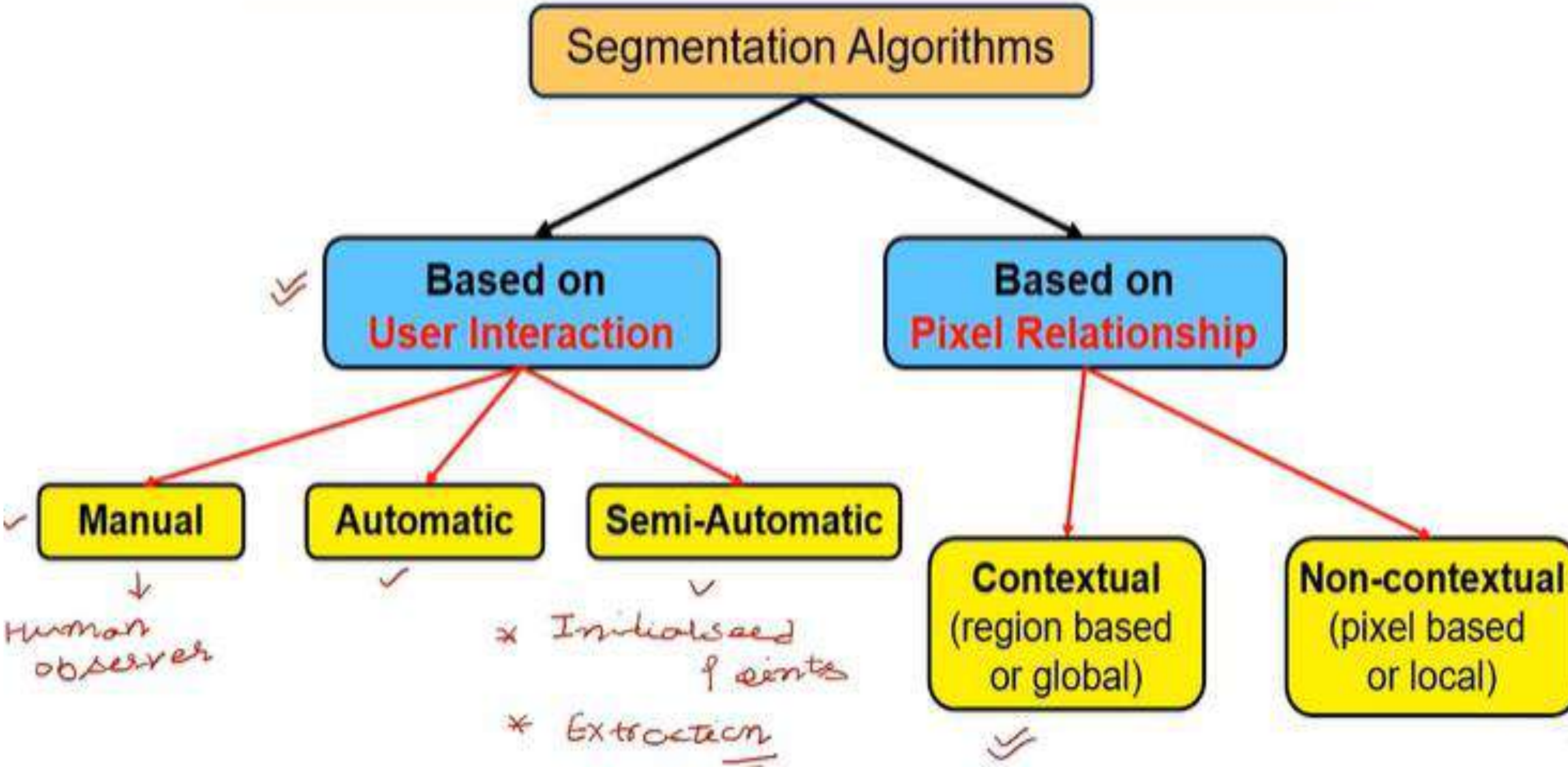
Let R represent the entire image region and Segmentation is partitioning R into n subgroups R_i

- $\cup_{i=1}^{n} Ri = R$ $\quad i = 1, 2, \ldots n \quad$ R
- Ri should be connected region :i=1,2,3,……..,n
- $Ri \cap R_1 = \emptyset\ (for\ all\ i\ and\ j): i \neq j$
- $P(R_i) = TRUE\ for\ i = 1,2,3, \ldots \ldots n$
- $P(R_i \cup R_j) = FALSE\ for\ i \neq j$

Here P(R_i) is a predicate that indicates some property over the region

Introduction to Image Segmentation

Classification of Image Segmentation Algorithms



Detection of discontinuities(Non-contextual)

Detection of Discontinuities

Types of Grey Level discontinuities are:

- **Point**
  - * Isolated Point
- **Line**
  - * Horizontal
  - * Vertical
  - * Slanted
- **Edge**
  - * Object outline

Image Segmentation

- Detection of discontinuities:
  - There are three basic types of gray-level discontinuities:
    - points , lines , edges
  - the common way is to run a mask through the image

Detection of Discontinuities

Point Detection

An isolated point is a point whose grey level is significantly different from its background in a homogeneous area

3x3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3x3

Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image

Response of the mask:

$$R = \sum_{i=1}^{9} w_i z_i$$

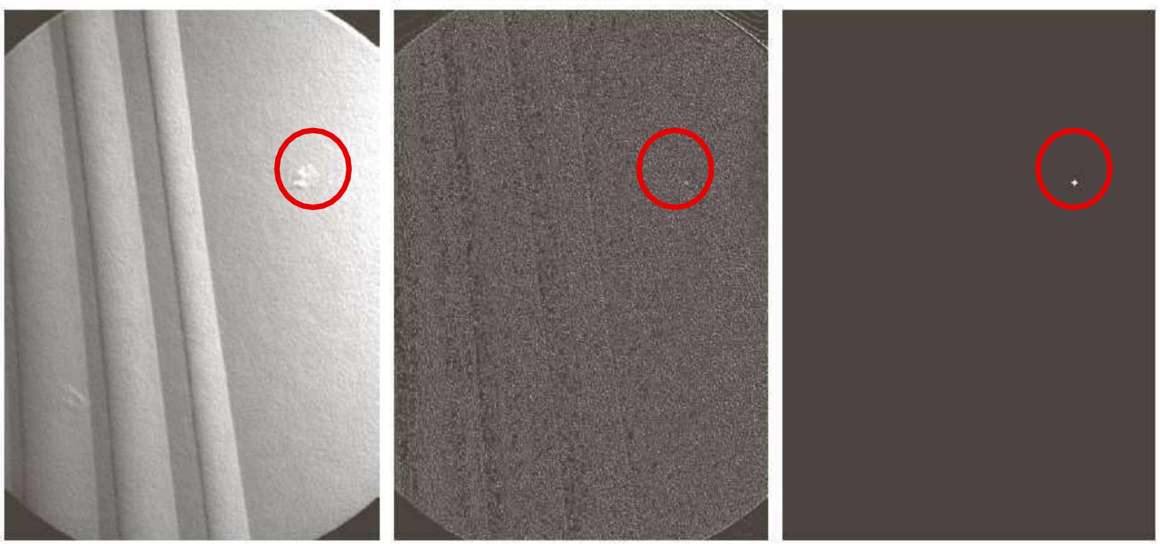
If,
$|R| \geq T$, a point is detected
where,
T is a non negative integer

-1	-1	-1
-1	8	-1
-1	-1	-1

Sample Mask for Point Detection

Point Detection:

1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Detection of Discontinuities

Line Detection

In line detection, **four types of masks** are used to get the responses i,e, R_1, R_2, R_3 and R_4 for the directions vertical, horizontal, $+45^0$ and -45^0 respectively

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

$+45^0$

2	−1	−1
−1	2	−1
−1	−1	2

-45^0

Response of the mask:

$$R_k = \sum_{k=1}^{4} \omega_k z_k$$

If, at a certain point in the image, $|R_i|>|R_j|$ for all $j \neq i$, that **point** is said to be more likely associated with a **line** in the direction of mask i

Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

R_1 Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

R_2 +45°

−1	2	−1
−1	2	−1
−1	2	−1

R_3 Vertical

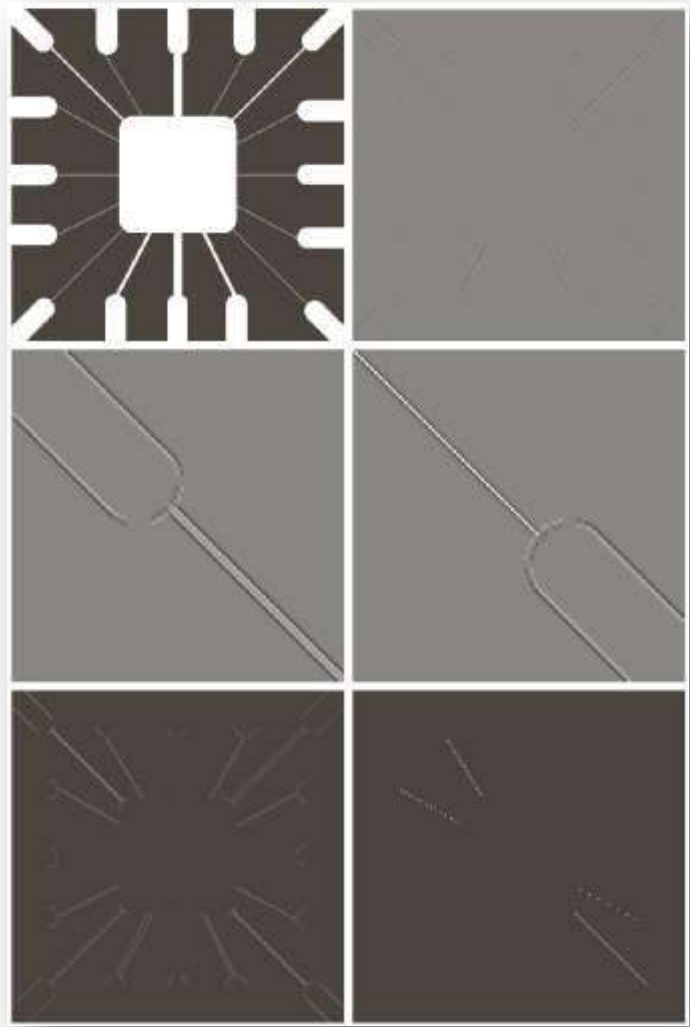
−1	−1	2
−1	2	−1
2	−1	−1

R_4 −45°

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

- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- the similar idea is used with other masks.
- note: the preferred direction of each mask is weighted with a larger coefficient (i.e.,2) than other possible directions.

Line Detection



a b
c d
e f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the $+45°$ line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Detection of Discontinuities

Edge Detection

- An edge is a set of connected pixels that lies on the boundary between two regions which differ in grey value. Pixels on edge is known as edge points
- **Edges provide an outline of the object**

In physical plane, **edge corresponds** to the **discontinuities in** depth, Surface orientation, change in material properties, light variations etc.

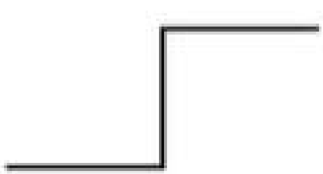
- It locates sharp changes in the intensity function
- Edges are pixels where brightness changes abruptly
- An edge can be extracted by computing the derivative of the image function
  - ✓ Magnitude of the derivative, indicates the strength or contrast of edge
  - ✓ Direction of the derivative vector, indicates the edge orientation

Detection of Discontinuities

Edge Detection

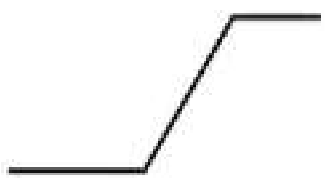
Some of the commonly encountered edges in image processing are:

Step Edge



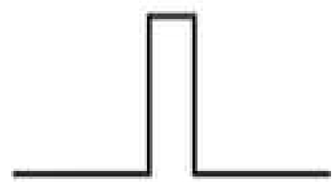
An abrupt change in intensity

Ramp Edge



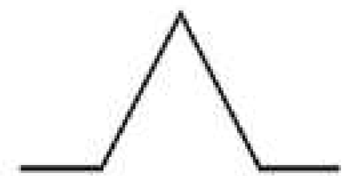
A slow and gradual change in intensity

Spike Edge



Quick change or immediately returns to original intensity level

Roof Edge



It is not instantaneous over short distance

Edge Detection Approach

- Segmentation by finding pixels on a region boundary.
- Edges found by looking at neighboring pixels.
- Region boundary formed by measuring gray value differences between neighboring pixels

Edge Detection

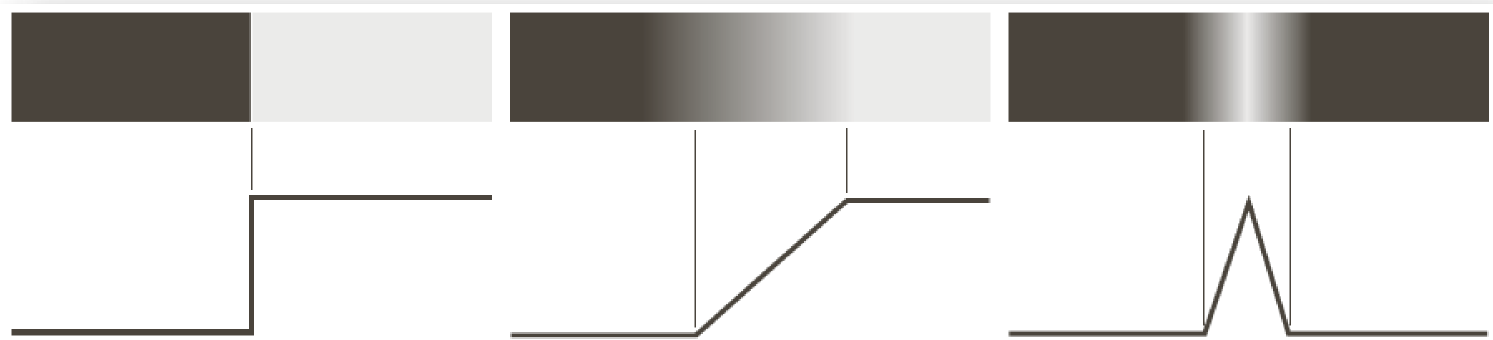


FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

Edge Detection

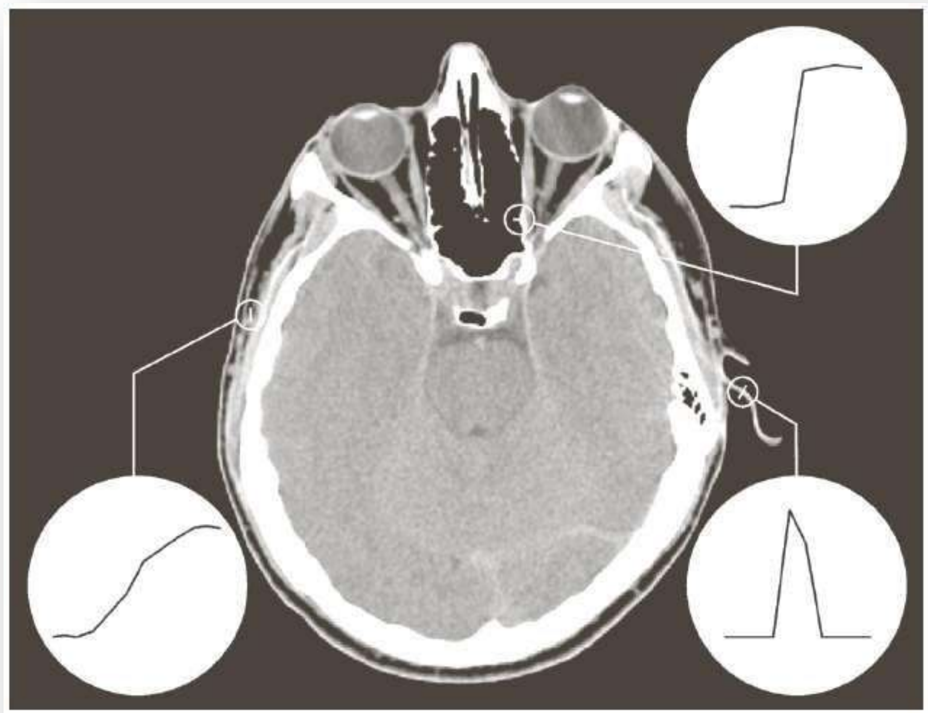
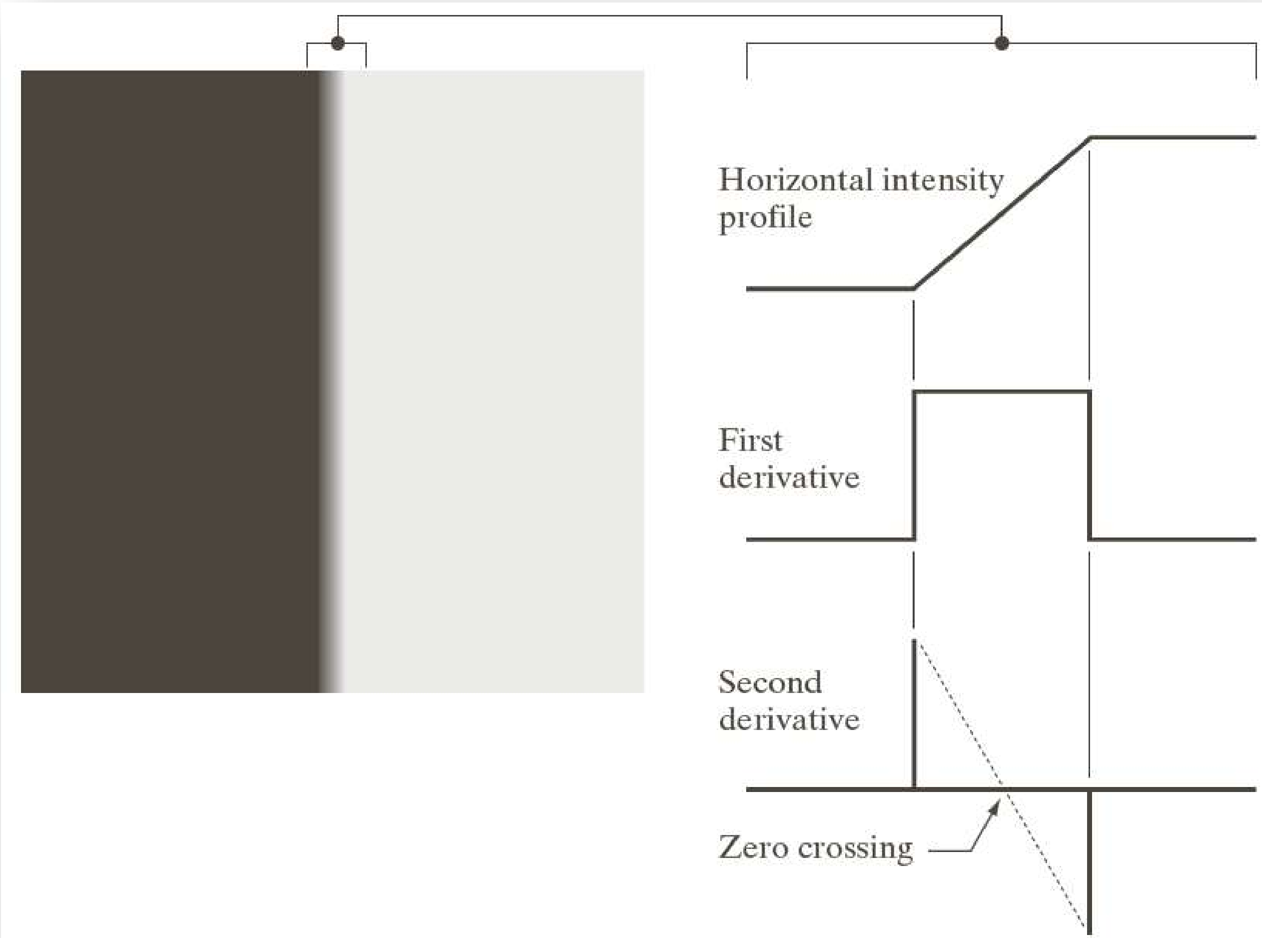


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Edge Detection



a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal vertical ramp edge.
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Detection of Discontinuities

Stages in Edge Detection

Start ⇨ Input Image ⇨ Filtering ⇨ Differn-tiation ⇨ Locali-zation ⇨ Display ⇨ End

It involves smoothing. This stage may be performed explicitly or implicitly

Detection of Discontinuities

Stages in Edge Detection

A B

Start ⇨ Input Image ⇨ Filtering ⇨ Differn-tiation ⇨ Locali-zation ⇨ Display ⇨ End

Filtering: It involves smoothing. This stage may be performed explicitly or implicitly

Differn-tiation: It distinguishes the edge pixels from other pixels

$$\text{Vector: } \nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

$$\text{Magnitude: } \nabla f = \text{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2 \right]^{1/2} \approx |G_x| + |G_y|$$

$$\text{Direction of gradient: } \tan^{-1}\left(\frac{Gy}{Gx} \right)$$

Detection of Discontinuities

Stages in Edge Detection

A B

Start ⇨ Input Image ⇨ Filtering ⇨ Differn-tiation ⇨ Locali-zation ⇨ Display ⇨ End

Filtering: It involves smoothing. This stage may be performed explicitly or implicitly

Differntiation: It distinguishes the edge pixels from other pixels

$f(x,y)$

Vector: $\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$

Magnitude: $\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2} \approx |G_x| + |G_y|$

Direction of gradient: $\tan^{-1}\left(\frac{Gy}{Gx}\right)$

θ

Localization: It determine the exact location of edge

Edge Detection

Gradient Operator

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

$$|\nabla f| = \left[G_x^2 + G_y^2\right]^{1/2}$$

for 3×3 mask

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

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
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−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.

Detection of Discontinuities

Edge Detection

Edge Detection Algorithms

- **Derivative Types**
  - It uses differentiation technique for edge detection
- **Template Matching**
  - It uses templates that resembles the target shapes and match with image
- **Gaussian Derivatives**
  - Very effective for real time images
- **Pattern Fit Approach**
  - Here surface is considered as topographic surface with pixel value representing altitude

Derivative types

First Order Edge Detection Operators

- Local transitions among different image intensities constitute an edge
- Therefore the objective is to measure the **intensity gradient**
- Edge detectors can be viewed as **gradient calculators**

Gradient Operator is represented as:

f(x,y)
2D

$$\text{Vector: } \nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

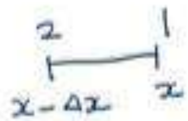
$$\text{Magnitude: } \nabla f = \text{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2} \approx |G_x| + |G_y|$$

$$\text{Direction of gradient: } \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

- An edge can be extracted by computing the derivative of the image function
  - ✓ **Magnitude of the derivative**, indicates the strength or contrast of edge
  - ✓ **Direction of the derivative vector**, indicates the edge orientation

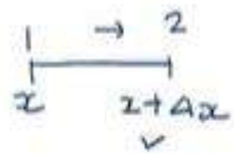
First Order Edge Detection Operators

Backward Difference:



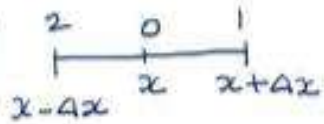
= [f(x) – f(x-Δx)] / Δx

Forward Difference:



= [f(x+ Δx) – f(x)] / Δx

Central Difference:



= [f(x+ Δx) – f(x-Δx)] / 2Δx

These differences can be obtained by applying the following masks, assuming Δx=1:

Backward Difference = f(x) – f(x-1)
= [1 -1]

Forward Difference: = f(x+1) – f(x)
= [-1 +1]

First Order Edge Detection Operators

Robert Operator

- ❑ Robert Kernels are derivatives with respect to the diagonal elements
- ❑ They are known as **Cross-Gradient Operators**
- ❑ They are based on **Cross Diagonal differences**

Generic gradient based algorithm can be:

- Read the image and smooth it
- Convolve the image f with g_x
- Convolve the image f with g_y
- Compute the edge magnitude and edge orientation
- Compare the edge magnitude with a threshold value
  - If edge magnitude is higher, assign it as a possible edge point

This algorithm can be applied to other masks also

Let $f(x,y)$ & $f(x+1,y)$ be neighbouring pixels, then

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

Robert masks may be of:

G_x

1	0
0	–1

G_y

0	1
–1	0

First Order Edge Detection Operators

Prewitt Operator

1 2 3

$x-\Delta x$ x $x+\Delta x$

The Prewitt Method takes the central difference of the neighbouring pixels; This difference can be represented mathematically as:

$$\frac{\partial f}{\partial x} = [f(x+1) - f(x-1)]/2$$

$\Rightarrow$ For 2D $\rightarrow$ (x,y)

$$[f(x+1,y) - f(x-1,y)]/2$$

central difference is obtained by mask $\rightarrow$ $[-1 \quad 0 \quad +1]$

G_x

–1	–1	–1
0	0	0
1	1	1

G_y

–1	0	1
–1	0	1
–1	0	1

3 × 3 digital approximation of Prewitt Operator is given as:

$$G_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

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

First Order Edge Detection Operators

Sobel Operator

- It provides both a differentiating and a smoothing effect
- Sobel Operator relies on the central differences
- It can be viewed as an approximation of first Gaussian Derivative
- Here convolution is both commutative an associative

$$\frac{\partial}{\partial x}(f * G) = f * \frac{\partial}{\partial x}(G)$$

G_x

–1	–2	–1
0	0	0
1	2	1

G_y

–1	0	1
–2	0	2
–1	0	1

3 × 3 digital approximation of Sobel Operator is given as:

$$G_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

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

Additional mask can be used to detect edges in diagonal as:

0	1	2
–1	0	1
–2	–1	0

–2	–1	0
–1	0	1
0	1	2

Edge Detection

Prewitt and Sobel Operators

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

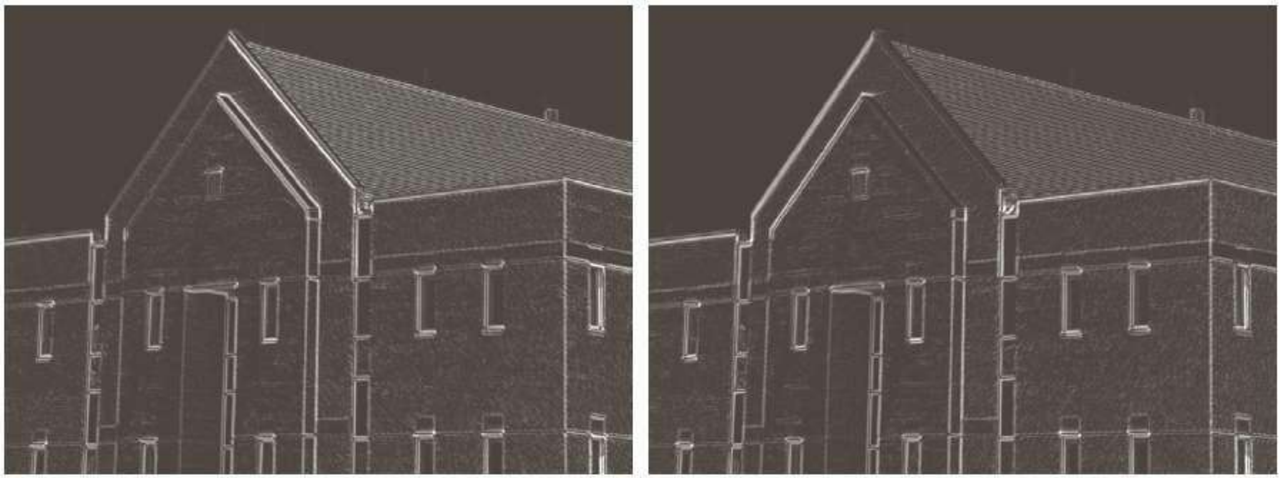
−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

Edge Detection



a b

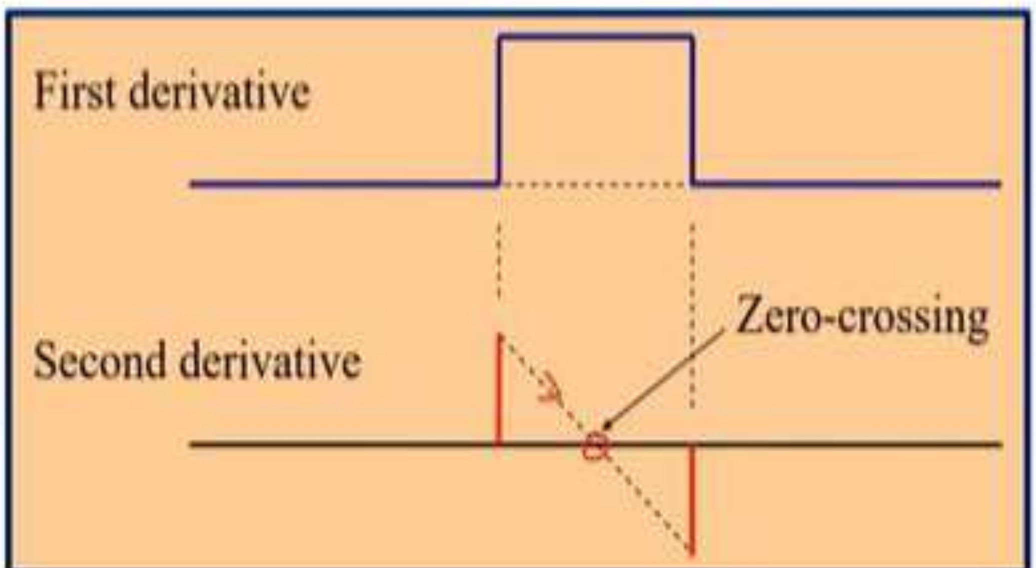
FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

Second Order Derivative Filters

- In first derivative, edges are considered to be present when edge magnitude is large compared to the threshold value
- In case of **second derivative**, edge is present at that location where the **second derivative is zero**
- It is like zero crossing, which can be observed as a sign change

20 40 20

20 -20



Second Order Derivative Filters

Laplacian Operator

- Laplacian Algorithm is one of the **zero-crossing algorithm**
- Laplacian masks are **very sensitive to noise** because there is no magnitude checking, even a small ripple looks like edge
- Therefore image must be filter first and then edge detection process is applied
- Advantage of Laplacian operator is that they are **rotationally invariant**

The Laplacian of a 2D function f(x,y) is a 2nd order derivative is defined as:

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

$x \quad y$

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

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

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

Second Order Derivative Filters

Laplacian Operator

Algorithm can be written as:

- Generate the mask
- Apply the mask
- Detect the zero crossing
  - ✓ Zero crossing is a situation where pixels in neighbourhood differ from each other pixel in sign

p & q $|\nabla^2 f(p)| \leq |\nabla^2 f(q)|$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

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

5×5

Different Laplacian Masks

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

3×3

① $\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$

② for other mask,

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$$

More Advanced Techniques for Edge Detection

- The basic edge detection method is based on simple filtering without taking note of image characteristics and other information
- **More advanced techniques make attempt to improve the simple detection by taking into account factors such as noise, scaling etc.**

Laplacian of Gaussian (Marr-Hildrith) Operator

Discussion of Marr and Hildreth was about:

- Intensity of changes is not independent of image scale
- Sudden intensity change will cause a zero crossing of the second derivative

Therefore, an edge detection operator should:

- Be capable of being tuned to any scale
- Be capable of computing the first and second derivatives

To minimize the noise susceptibility of the Laplacian Operator, Laplacian of Gaussian (LoG) Operator is often preferred

More Advanced Techniques for Edge Detection

Laplacian of Gaussian (Marr-Hildrith) Operator

The LoG Algorithm can be written as:

- Generate the mask and apply LoG to the image
- Detect the Zero Crossing

* For 1D,

$$\nabla^2(f * g) = f * \nabla^2 g = f * LoG$$

Let the 2D, Gaussian f^n is given as →

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right]$$

To suppress the noise, the image is convolved with the Gaussian smoothing function before using the Laplacian for edge detection

$$\nabla\left[G_\sigma(x,y) * f(x,y)\right] = \left[\nabla G_\sigma(x,y)\right] * f(x,y) = LoG * f(x,y)$$

The LoG function can be derived as:

$$\frac{\partial}{\partial x} G_\sigma(x,y) = \frac{\partial}{\partial x} e^{\left[-\frac{x^2+y^2}{2\sigma^2}\right]} = -\frac{x}{\sigma^2} e^{\left[-\frac{x^2+y^2}{2\sigma^2}\right]}$$

More Advanced Techniques for Edge Detection

Laplacian of Gaussian (Marr-Hildrith) Operator

Similarly,

$$\frac{\partial^2}{\partial x^2} G_\sigma(x,y) = \frac{x^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} - \frac{1}{\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} = \frac{x^2-\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

Similarly, by ignoring the normalization constant $1/\sqrt{2\pi\sigma^2}$, we get

$$\frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{y^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} - \frac{1}{\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} = \frac{y^2-\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

The LoG kernel can be described as:

$$LoG \triangleq \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

σ $\uparrow$

More Advanced Techniques for Edge Detection

Canny Edge Detection Algorithm

- ❑ The Canny edge detector is an edge detection operator that uses a multistage algorithm to detect a wide range of edges in images
- ❑ This algorithm was developed by **John F. Canny** in **1986**

Canny approach is based on optimizing the trade off between two performance criteria and can be described as:

- ❑ **Good Edge Detection:** should be capable to detect only real edge points and discard all false edge points
- ❑ **Good Edge Localization:** should have the ability to produce edge points which are close to real edges
- ❑ **Only one response to each edge:** should not produce any false, double or spurious edges

More Advanced Techniques for Edge Detection

Canny Edge Detection Algorithm

Step-1

- Convolve the image with Gaussian Filter
- Compute the gradient of resultant smooth image
- Store the edge magnitude and edge orientation in two separate arrays

Smoothing image and computing coefficients

$$S = G_\sigma * I \qquad G_\sigma = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla S = \left[\frac{\partial}{\partial x} S \quad \frac{\partial}{\partial y} S \right]^T = \left[S_x \quad S_y \right]^T$$

Computing gradient magnitude and direction

$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

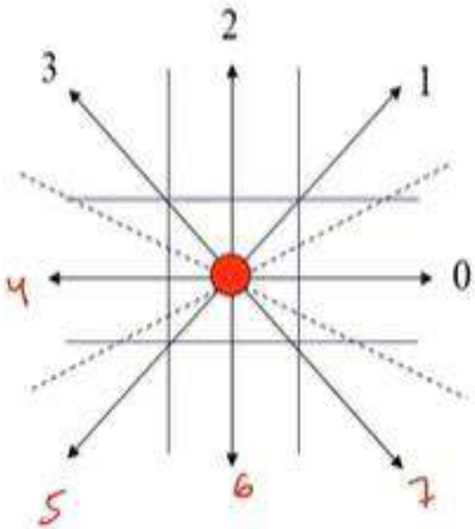
$$\theta = \tan^{-1} \frac{S_y}{S_x}$$

More Advanced Techniques for Edge Detection

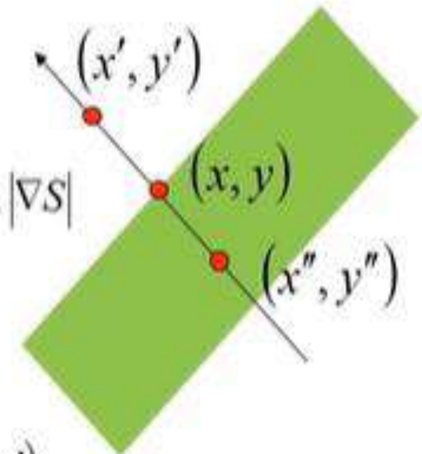
Canny Edge Detection Algorithm

Step-2

- Thinning of edges by a process known as Non-maxima Suppression



(x',y') and (x'',y'') are the neighbors of (x,y) in $|\nabla S|$ along the direction normal to an edge



$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\nabla S|(x',y') \\ & \& \, |\nabla S|(x,y) > |\nabla S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

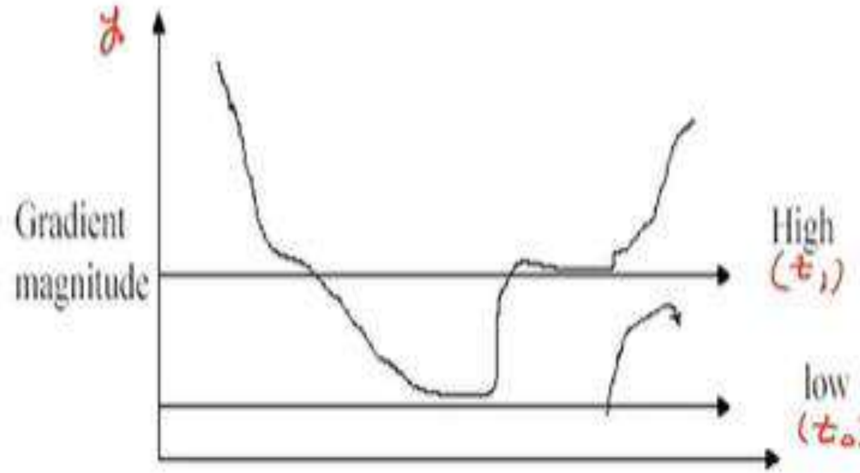
More Advanced Techniques for Edge Detection

Canny Edge Detection Algorithm

Step-3

- Apply Hysteresis Thresholding
  (objective is that only a large amount of change in gradient magnitude)

- If the gradient at a pixel is above **'High'**, declare it an **'edge pixel'**
- If the gradient at a pixel is below **'Low'**, declare it a **'non-edge-pixel'**
- If the gradient at a pixel is between 'Low' and 'High' then declare it an 'edge pixel' if and only if it is connected to an 'edge pixel' directly or via pixels between 'Low' and ' High'



Template matching

Template Matching Masks

- Gradient masks are isotropic and insensitive to direction
- Sometimes it is necessary to design direction sensitive filters, such type of filters are known as Template Matching Filters

Kirsch Masks

- ✓ Kirsch masks are known as compass masks because they are obtained by taking one mask and rotating it to the eight major directions
- ✓ Directions are: East, North East, North, North West, West, South West, South and South East

-3 -3 5
5
5
-3 5 5
-3
8

$$M_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad M_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad M_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad M_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad M_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad M_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad M_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Template Matching Masks

Robinson Compass Masks

The spatial masks for the Robinson edge operator for all the directions are as:

- ✓ Similar to Kirsch masks, the mask that produces the maximum value defines the **direction of the edge**

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

M_1 →

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

M_2 ↗

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

↑

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

↖

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

←

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

↙

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

↓

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

↘

Region based Segmentation

What is a Region?

- Basic definition :- A group of connected pixels with similar properties.

- Important in interpreting an image because they may correspond to objects in a scene.
  - For that an image must be partitioned into regions that correspond to objects or parts of an object.

Region-Based vs. Edge-Based

Region-Based	Edge-Based
Closed boundaries	Boundaries formed not necessarily closed
Multi-spectral images improve segmentation	No significant improvement for multi-spectral images
Computation based on similarity	Computation based on difference

Image Thresholding

- What is thresholding?
- Simple thresholding
- Adaptive thresholding

Thresholding – A Key Aspect

- Most algorithms involve establishing a threshold level of certain parameter.
- Correct thresholding leads to better segmentation.
- Using samples of image intensity available, appropriate threshold should be set automatically in a robust algorithm i.e. no hard-wir of gray values

Automatic Thresholding

- Use of one or more of the following:-
  1. Intensity characteristics of objects
  2. Sizes of objects
  3. Fractions of image occupied by objects
  4. Number of different types of objects
- Size and probability of occurrence – most popular
- Intensity distributions estimate by histogram computation.

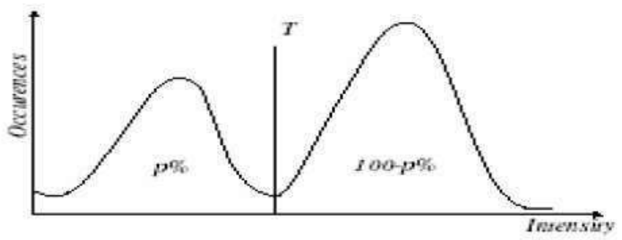
Automatic Thresholding Methods

- Some automatic thresholding schemes:

  1. P-tile method
  2. Iterative threshold selection
  3. Adaptive thresholding

Thresholding Methods

- P-tile Method:- If object occupies P% of image pixels then set a threshold T such that P% of pixels have intensity below T.



- Iterative Thresholding:- Successively refines an approx. threshold to get a new value which partitions the image better.

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

P-Tile Thresholding

- Thresholding is usually the first step in any segmentation approach
- Single value thresholding can be given mathematically as follows:

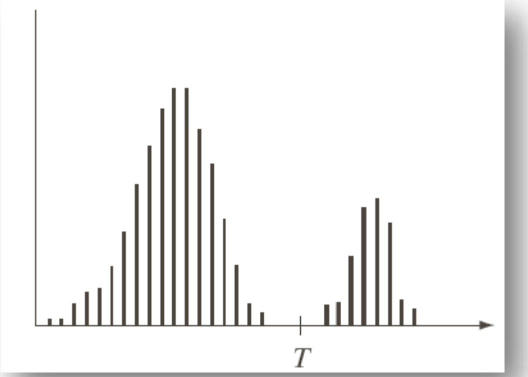
$$g(x,y) = \begin{cases} 1 & \text{if} \quad f(x,y) > T \\ 0 & \text{if} \quad f(x,y) \le T \end{cases}$$

P-Tile Thresholding

- Basic global thresholding:
  - Based on the histogram of an image
    Partition the image histogram using a single global threshold

P-Tile Thresholding

- Basic global thresholding:
  - The success of this technique very strongly depends on how well the histogram can be partitioned



Iterative P-Tile Thresholding

- The Basic global thresholding:
  1. Select an initial estimate for T (typically the average grey level in the image)
  2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels $\leq$ T
  3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Iterative P-Tile Thresholding

- The Basic global thresholding:
  4. Compute a new threshold value:

$$T = \frac{\mu_1 + \mu_2}{2}$$

  5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

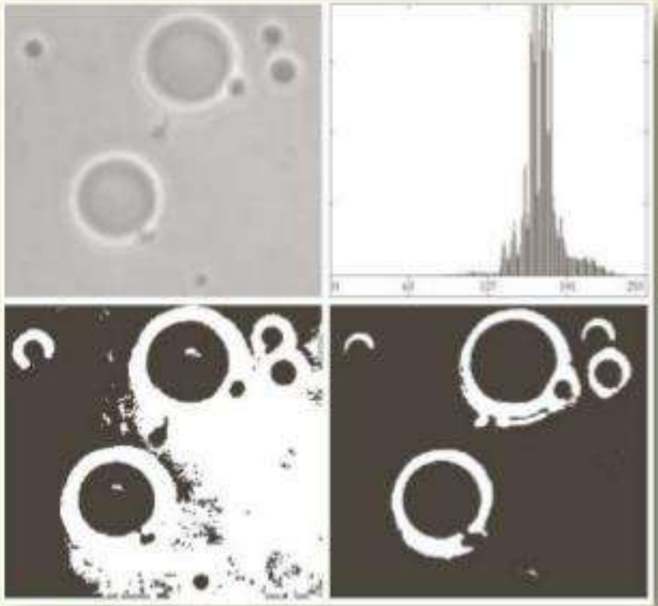
This algorithm works very well for finding thresholds when the histogram is suitable.

Adaptive Thresholding

- Adaptive Thresholding is used in scenes with uneven illumination where same threshold value not usable throughout complete image.
- In such case, look at small regions in the image and obtain thresholds for individual sub-images. Final segmentation is the union of the regions of sub-images.

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding



a b
c d

FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

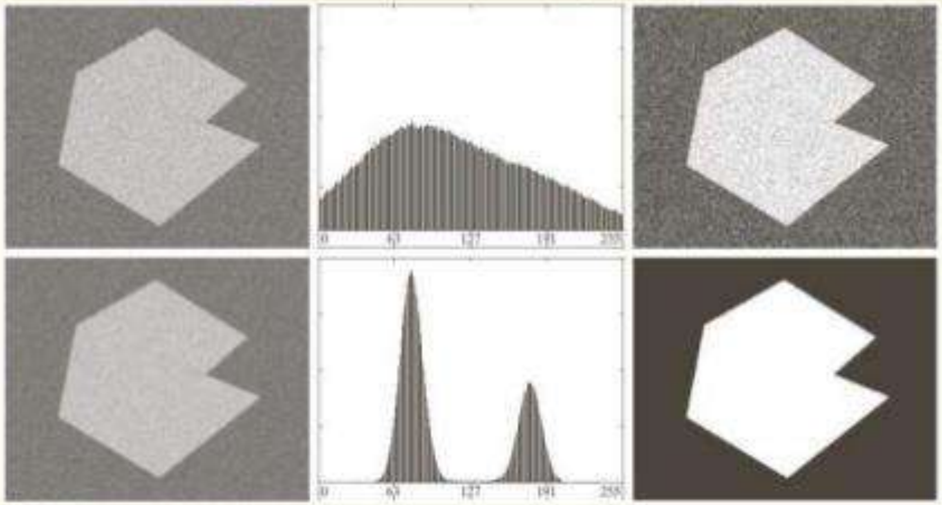
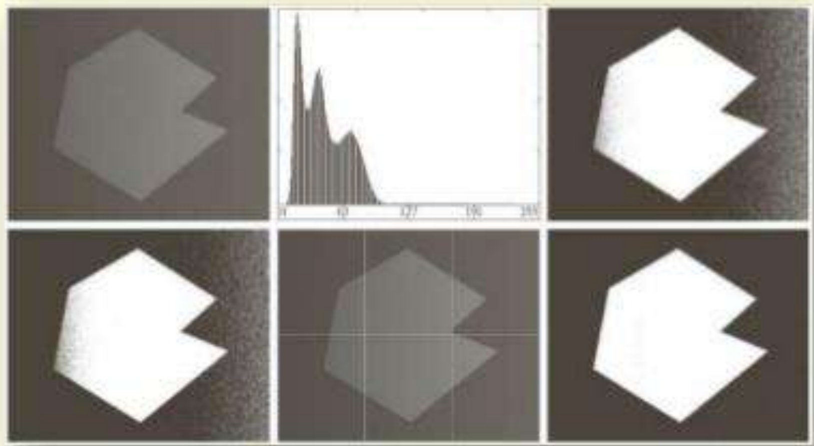


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding



a b c
d e f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

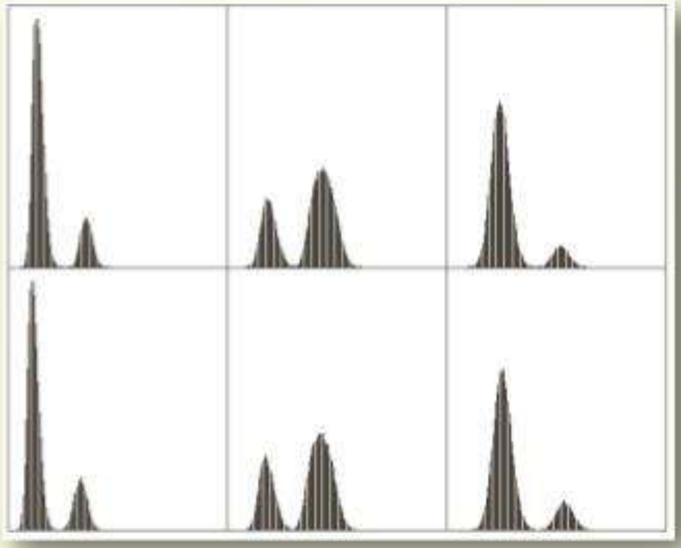


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Summary

- Segmentation is the most essential step in most scene analysis and automatic pictorial pattern recognition problems.

- Choice of the technique depends on the peculiar characteristics of individual problem in hand.

Region-based segmentation

- Find regions directly via pixel inspection
  1. Region growing
  2. Split and merge

Region Growing Algorithm

- Region growing is a procedure that groups pixels or sub regions into larger regions
- Logic behind region growing algorithms is the **principle of similarity**

Similarity Criteria

Principle of similarity states that a region is **coherent** if all the pixels of that region are **homogeneous**

Homogeneity of regions is used as the main segmentation criterion in region growing

- gray level
- Color
- texture
- shape
- model etc.

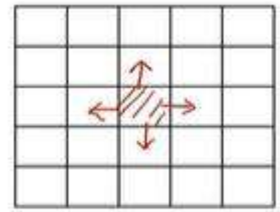
Choice of criteria affects segmentation results dramatically!

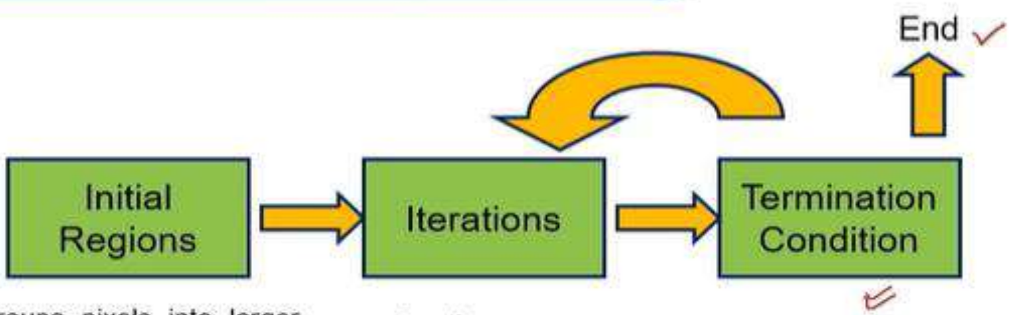
Region Growing Algorithm

Major steps of the region growing algorithm are:

- ✓ ❑ Selection of the initial seed
- ✓ ❑ Seed growing criteria
- ✓ ❑ Termination of the segmentation process

5×5





- Groups pixels into larger regions
- Starts with a **seed** region
- **Grows** region by **merging** neighboring pixels

Iterative process
- How to start?
- How to iterate?
- When to stop?

Region Growing Algorithm

Example: For a given image, show the results of region growing algorithm

5×5

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

* Seed point → S_1 = 9, S_2 = 1
* Threshold, T $\boxed{T \le 4}$

* $S_1 = 9$

$$|f(x,y) - f(x',y')| \le 4$$

$$|f(x,y) - 9| \le 4$$

$$f(x,y) = \{5,6,7,8,9\}$$

* $S_2 = 1$

$$|f(x,y) - f(x',y')| \le 4$$

$$|f(x,y) - 1| \le 4$$

$$f(x,y) = \{0,1,2,3,4,5\}$$

Region Growing Algorithm

Example: For a given image, show the results of region growing algorithm

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

5x5

* Seed point → S_1 = 9, S_2 = 1
* Threshold, T $\boxed{T \le 4}$
* $S_1 = 9$

$$|f(x,y) - f(x',y')| \le 4$$

$$|f(x,y) - 9| \le 4$$

$$f(x,y) = \{5,6,7,8,9\} \Rightarrow A$$

* $S_2 = 1$

$$|f(x,y) - f(x',y')| \le 4$$

$$|f(x,y) - 1| \le 4$$

$$f(x,y) = \{0,1,2,3,4,5\} \Rightarrow B$$

Resultant Image

B	B	A	A	A
B	B	A	A	A
B	B	A	A	A
B	B	A	A	A
B	B	A	A	A

Split and Merge Algorithm

- It is an alternative method of image segmentation
- An image is sub-divided into arbitrary disjoined region
- Arbitrary regions can be split and merged in order to satisfy the condition

The algorithm is stated in two phases

$P(R_i) = False$

Phase-I

- ✓ Split & Continue subdivision process until some stopping criteria is fulfilled
- ✓ Often it is stopped when no further splitting is possible

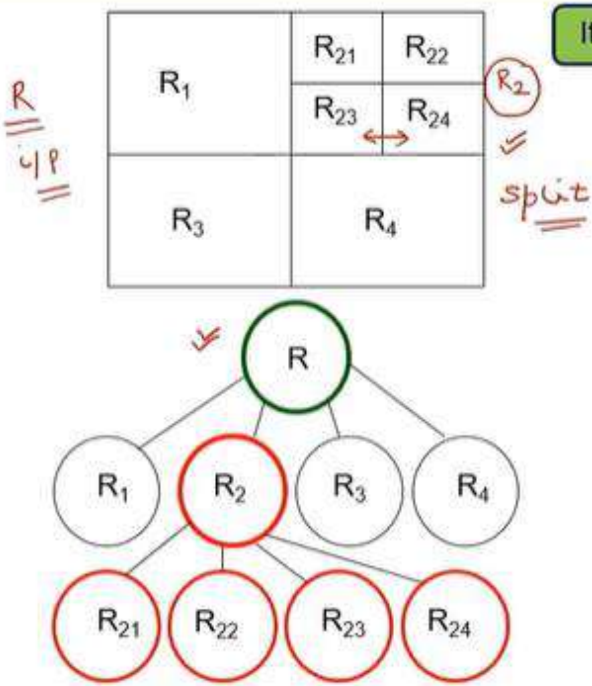
Phase-II

- ✓ Merge adjacent regions if the regions share any common criteria
- ✓ Stop the process when no further merging is possible

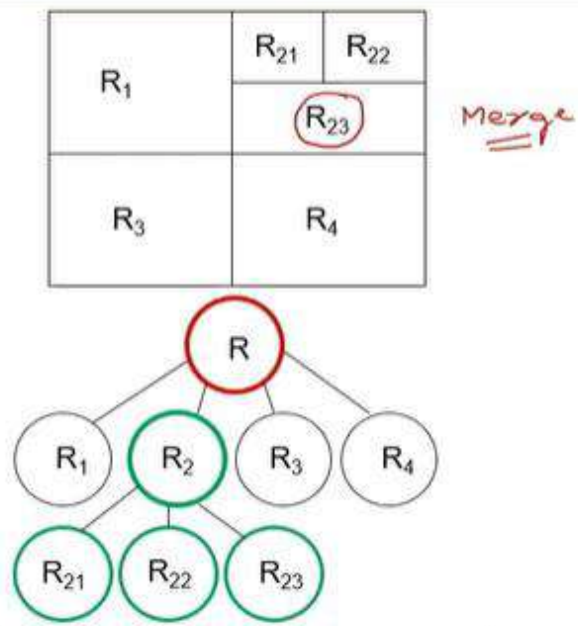
$P(R_i \cup R_j) = True$

Split and Merge Algorithm

It is also known as "**Divide and Conquer**" or "**Top Down**" approach



Quadtree of Splitting



Quadtree of Splitting and Merging

Split and Merge Algorithm

Example: For a given image, apply split and merge algorithm

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

Split and Merge Algorithm

Example: For a given image, apply split and merge algorithm

6	**5**	**6**	**6**	**7**	**7**	**6**	**6**
6	**7**	**6**	**7**	**5**	**5**	**4**	**7**
6	**6**	**4**	**4**	**3**	**2**	**5**	**6**
5	**4**	**5**	**4**	**2**	**3**	**4**	**6**
0	**3**	**2**	**3**	**3**	**2**	**4**	**7**
0	**0**	**0**	**0**	**2**	**2**	**5**	**6**
1	**1**	**0**	**1**	**0**	**3**	**4**	**4**
1	**0**	**1**	**0**	**2**	**3**	**5**	**4**

max. (1) − min (2)
max. (2) − min (1)

$\boxed{T \leq 3}$

Solution:

$\boxed{\text{Threshold, } (T) \leq 3}$

max. Pixel = 7
min. Pixel = 0] 7 − 0 = 7

condition does not satisfies → split

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

A B C D

7, 2] 5 ≰ 3

7, 0] 7

splitting

Merge

R_1

R_2

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

Morphology

Morphology

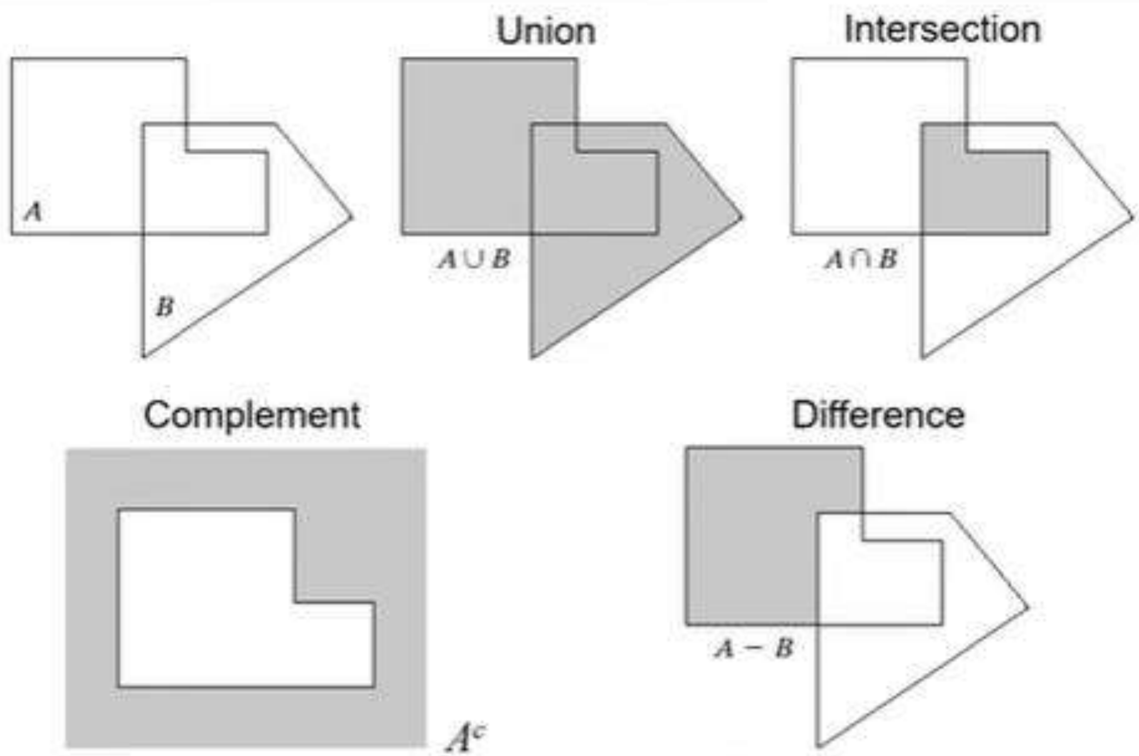
- The word *Morphology* denotes a branch of biology which deals with form and structure of animals and plants.
- Used for extracting image components that are useful in the representation and description of region shape.

Outline of the Chapter

- Basic concepts
- Morphological operations
  - Dilation
  - Erosion
  - Opening
  - Closing
  - Hit-&-Miss Transform
- Basic Morphological Algorithms
  - Boundary Extraction
  - Region Filling
  - Thinning
  - Thickening

Basic concepts

Basic concepts



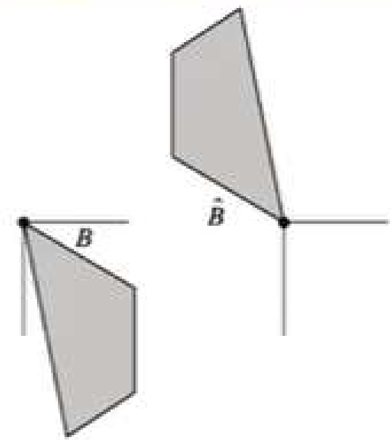
$$A^c = \{w \mid w \notin A\}$$

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

Basic concepts

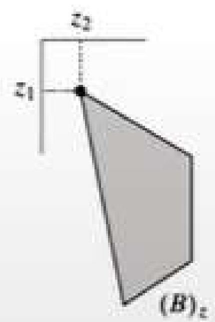
Reflection

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



Translation

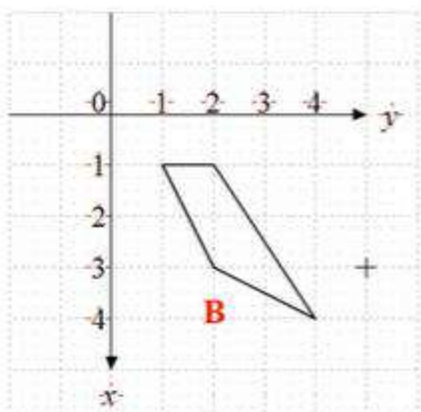
$$(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$$

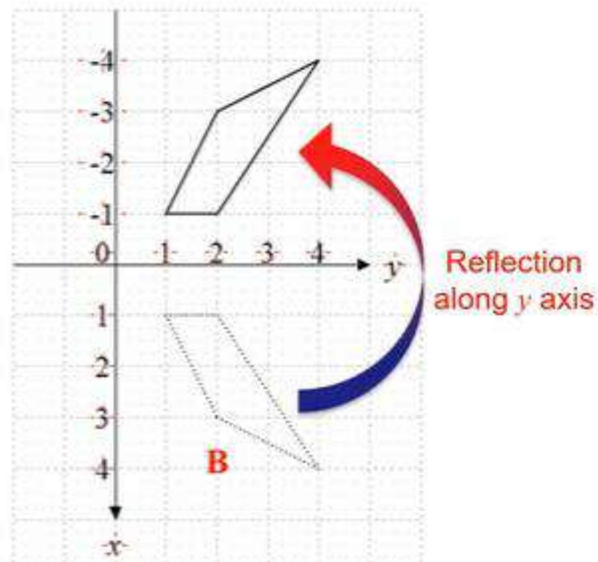


Basic concepts

contd…

Reflection

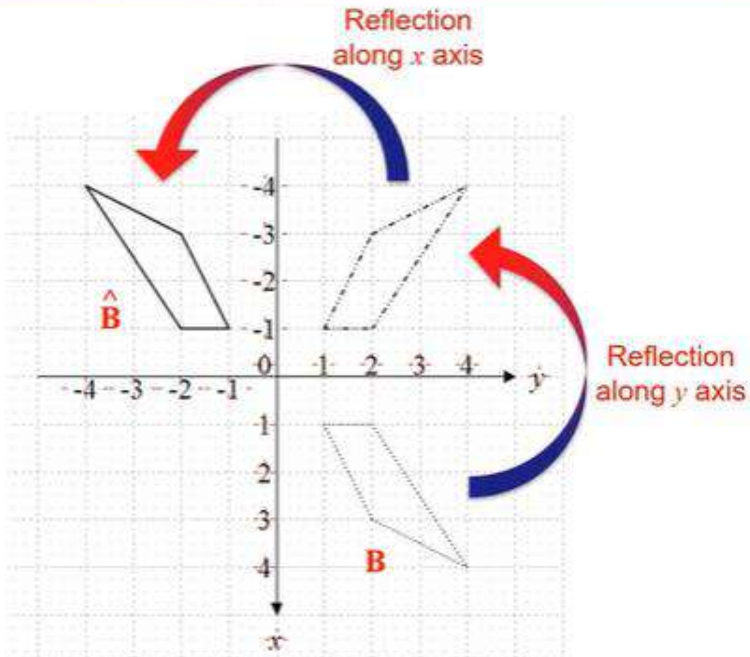




Basic concepts

contd

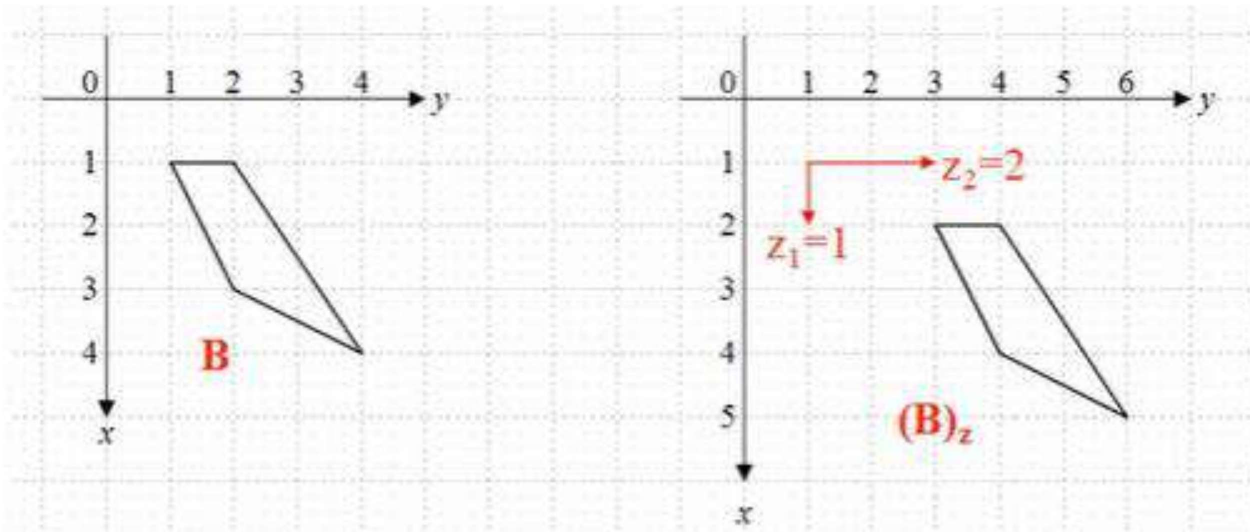
Reflection



Basic concepts

cont

Translation



Morphological operations

❖ Dilation

Dilation of A by B

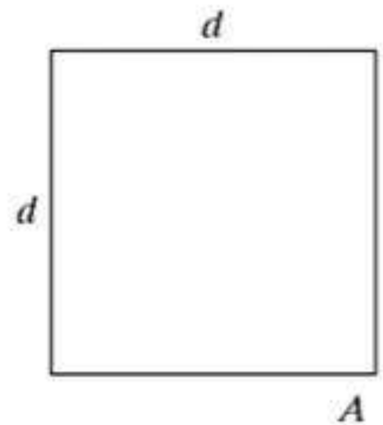
$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$$

This can be rewritten as

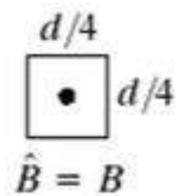
$$A \oplus B = \{z \mid \left[(\hat{B})_z \cap A\right] \subseteq A\}$$

- Set of all displacements z such that $\hat{B}$ and A overlap by at least one point.
- This process can be treated as 2D-Convolution of A with structuring element B.

❖ Dilation



Object A

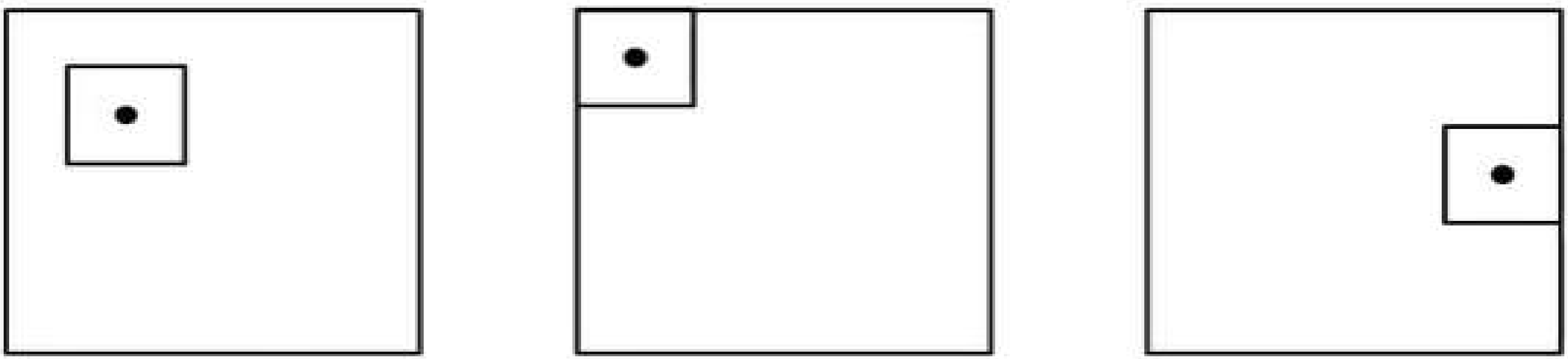


Structuring Element B

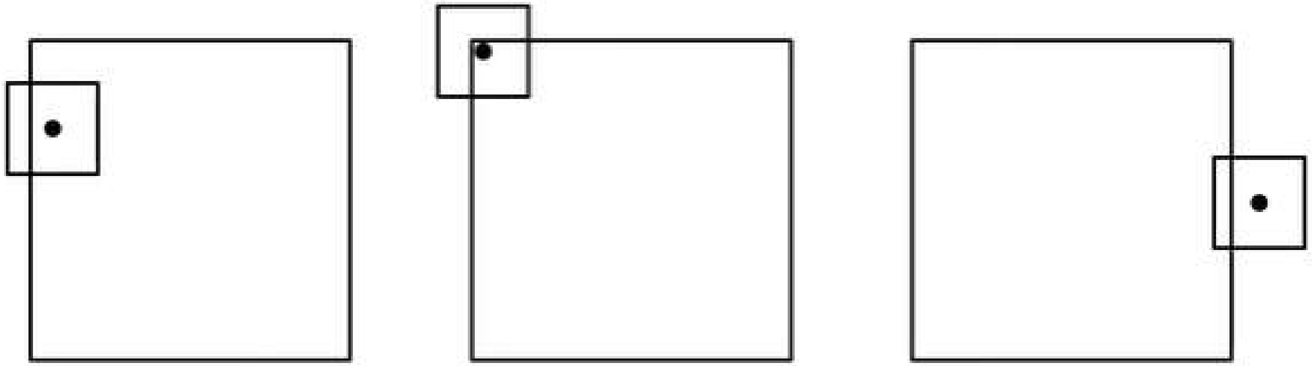
Morphological operations

contd

- Dilation



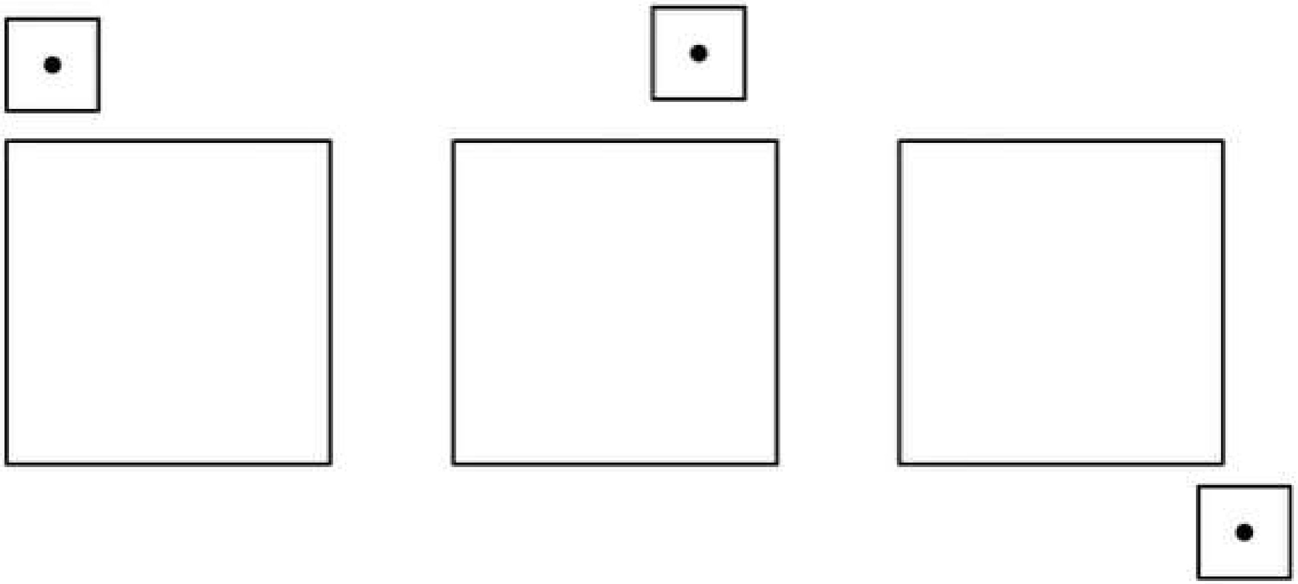
- Dilation



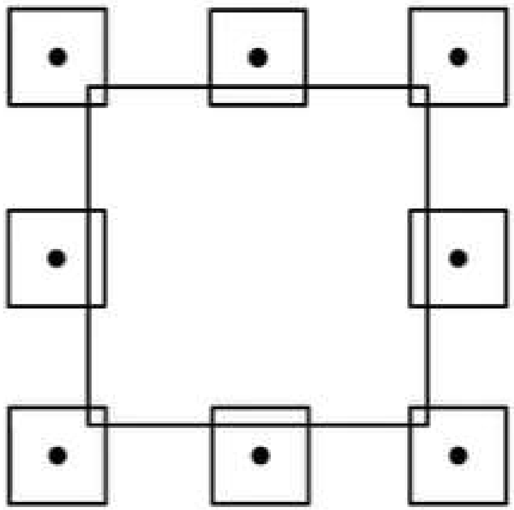
Morphological operations

contd…

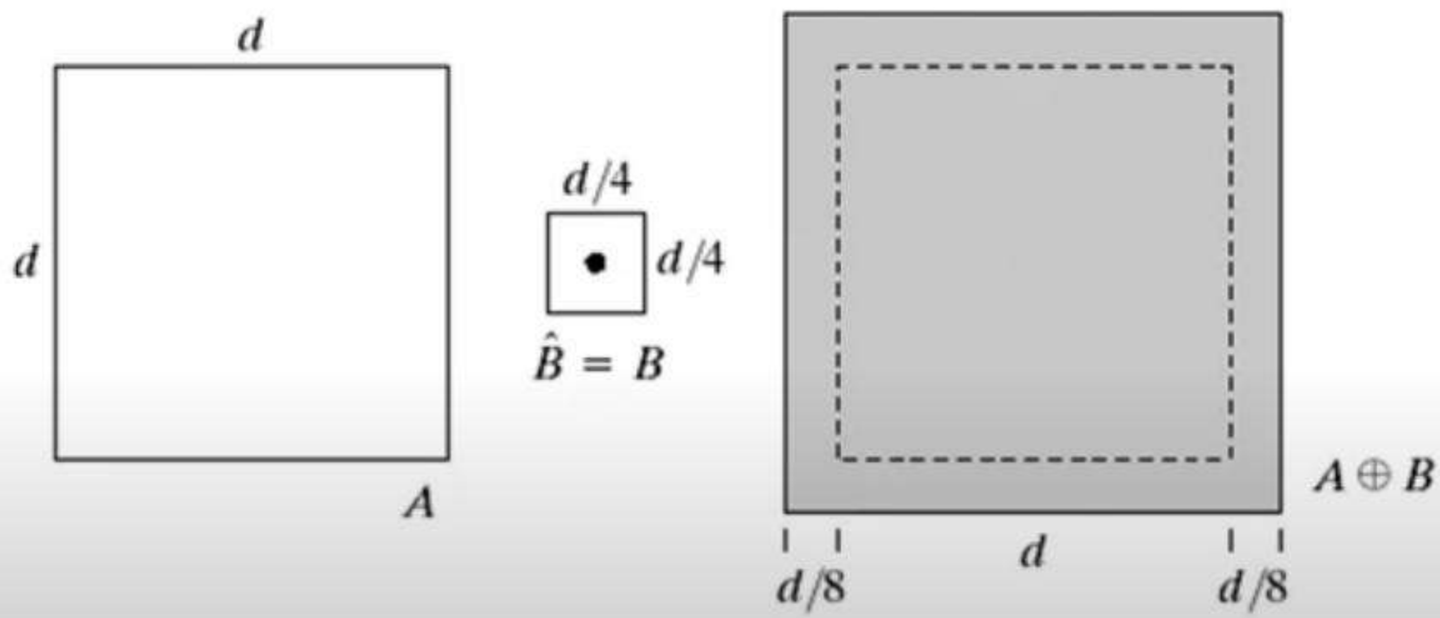
❖ Dilation



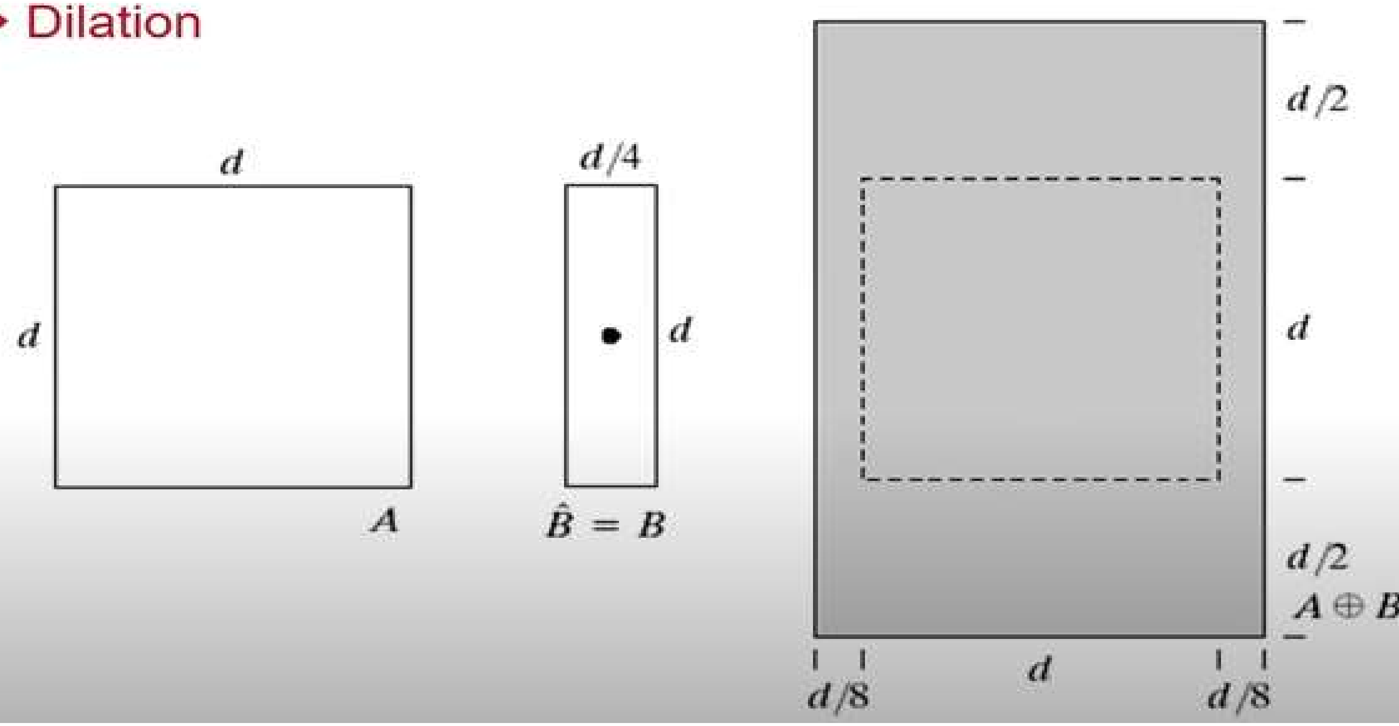
- Dilation



❖ Dilation



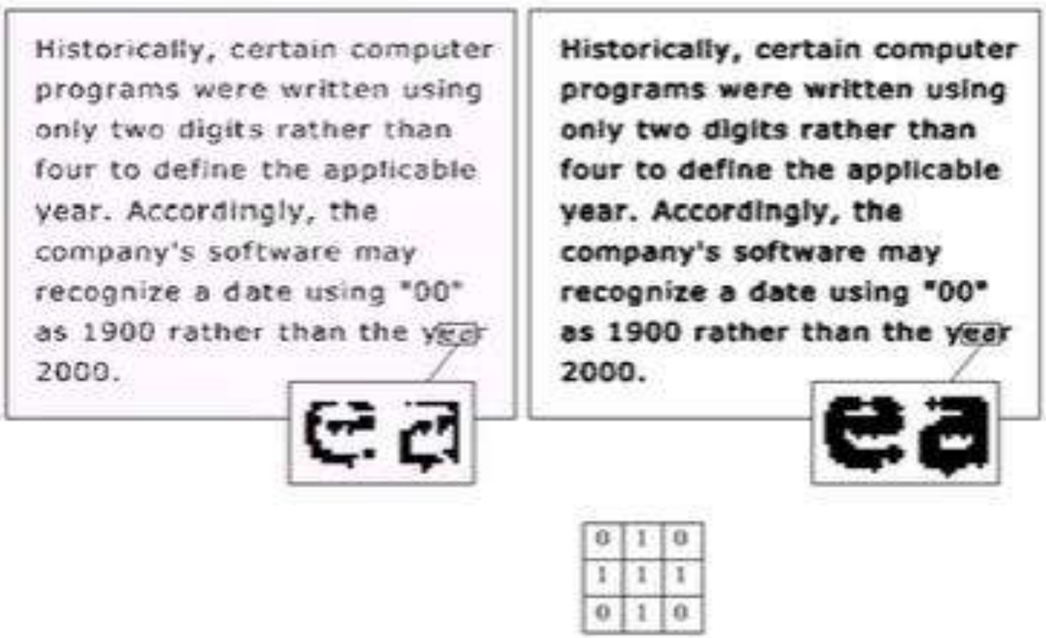
❖ Dilation



❖ Dilation

Applications

- Expansion
- Bridging gaps



a c
b

FIGURE 9.5 (a) Sample text of poor resolution with broken characters (magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

Morphological operations

contd.

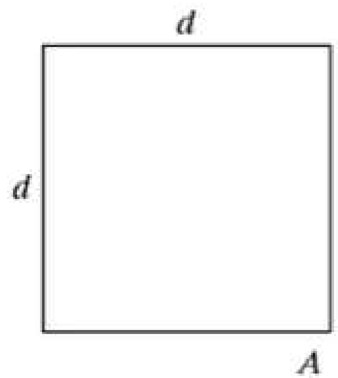
❖ Erosion

Erosion of A by B

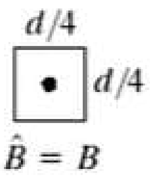
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

– Set of all points z such that B translated by z is completely contained in A.

❖ Erosion

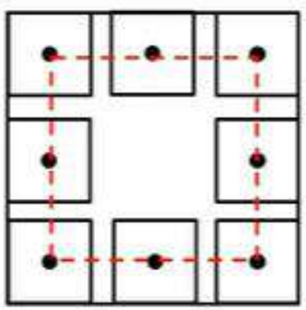


Object A

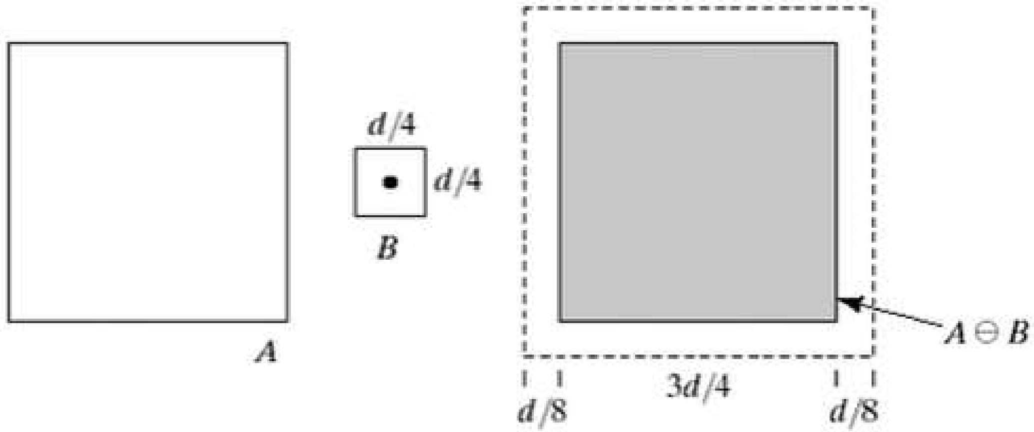


Structuring Element B

❖ Erosion



❖ Erosion



❖ Erosion

Applications

- Removing irrelevant details

a b c

FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Morphological operations

❖ Relation between Dilation & Erosion

– Dilation and erosion are duals of each other with respect to set complementation and reflection

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

❖ Opening

Opening of A by structuring element B

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

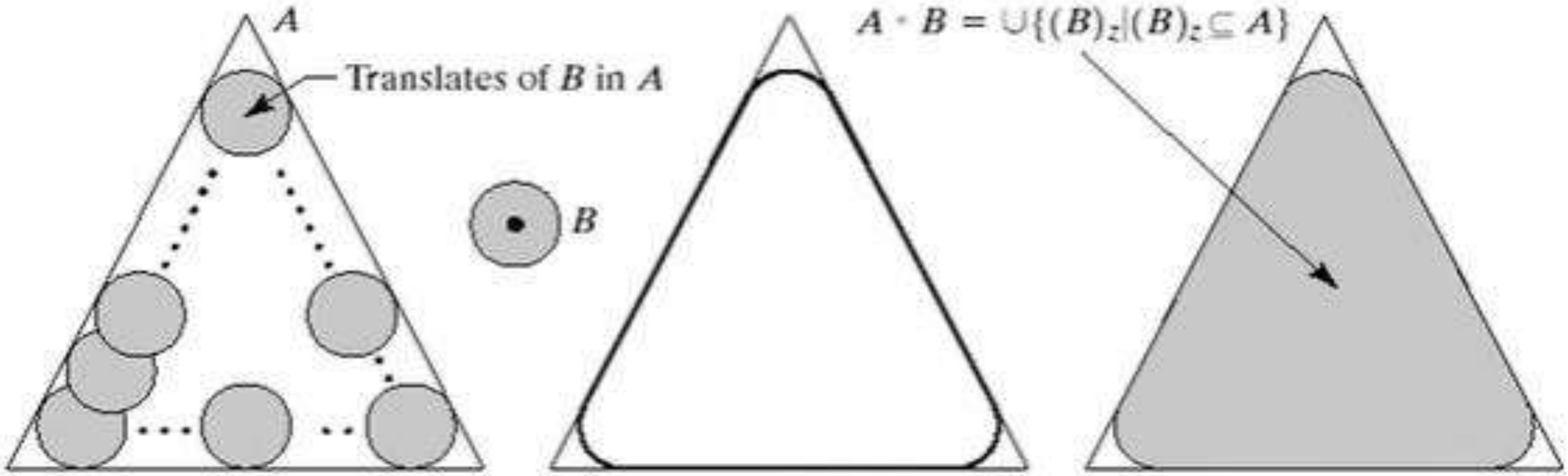
⇨ Opening A by B is the erosion of A by B, followed by a dilation of the result by B.

Geometric interpretation:

– Roll the structuring element B ***inside*** the boundary of A.

$$A \circ B = \cup \{ (B)_z \mid (B)_z \subseteq A \}$$

❖ 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). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Morphological operations

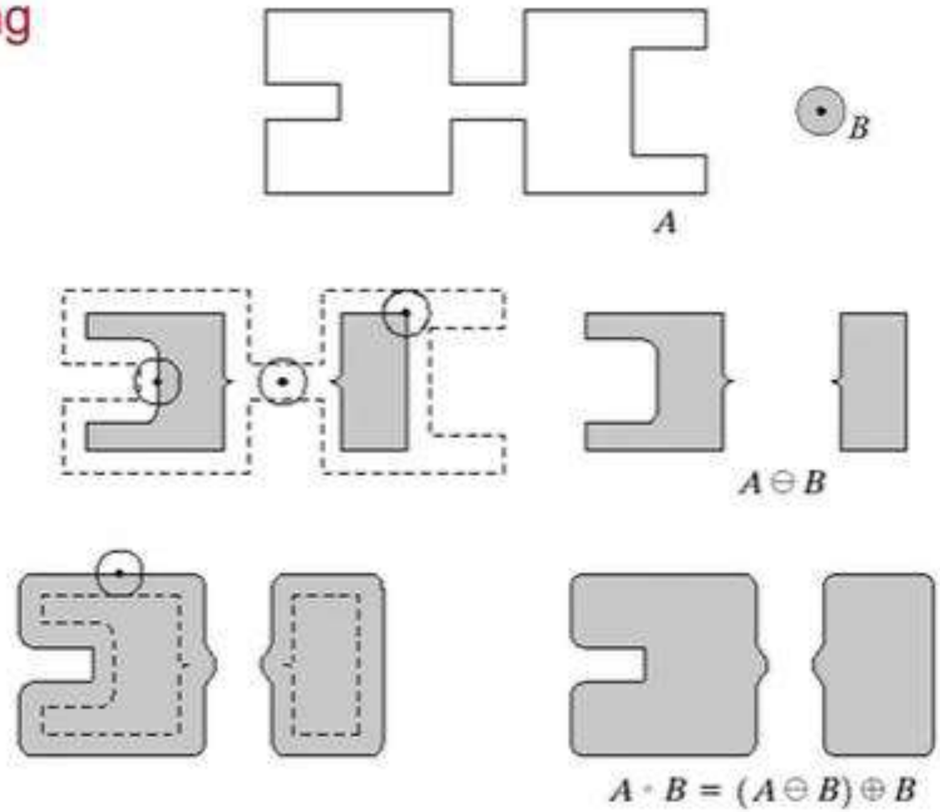
❖ Opening

Applications

- Smoothens contours
- Breaks narrow necks
- Eliminates thin overhangs

Morphological operations

- Opening



❖ Opening

Properties

- $A \circ B$ is a subset (sub image) of A ⇨ opening is a subset of the input
- If C is subset of D, then $C \circ B$ is subset of $D \circ B$ ⇨ Monotonicity is preserved
- $(A \circ B) \circ B = A \circ B$ ⇨applying more than one opening has no effect on the result

Morphological operations

contd..

❖ Closing

Closing of A by structuring element B

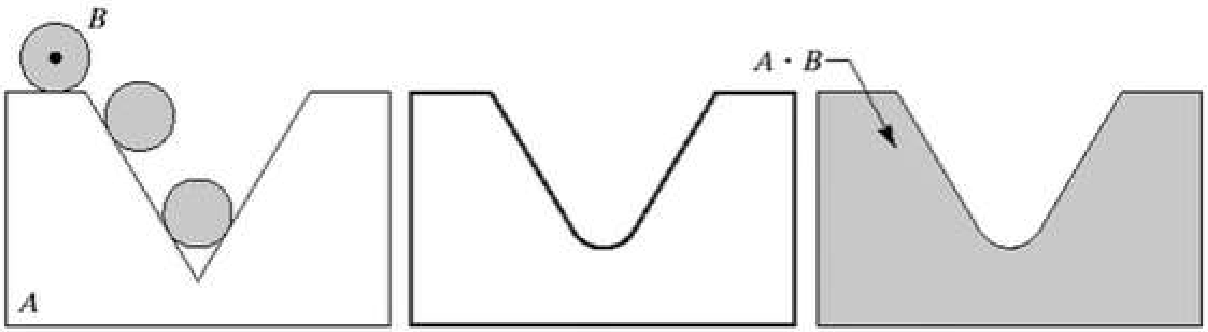
$$A \bullet B = (A \oplus B) \ominus B$$

⇨ Closing A by B is the dilation of A by B, followed by a erosion of the result by B.

Geometric interpretation:

– Roll the structuring element B ***outside*** the boundary of A.

❖ Closing



a b c

FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

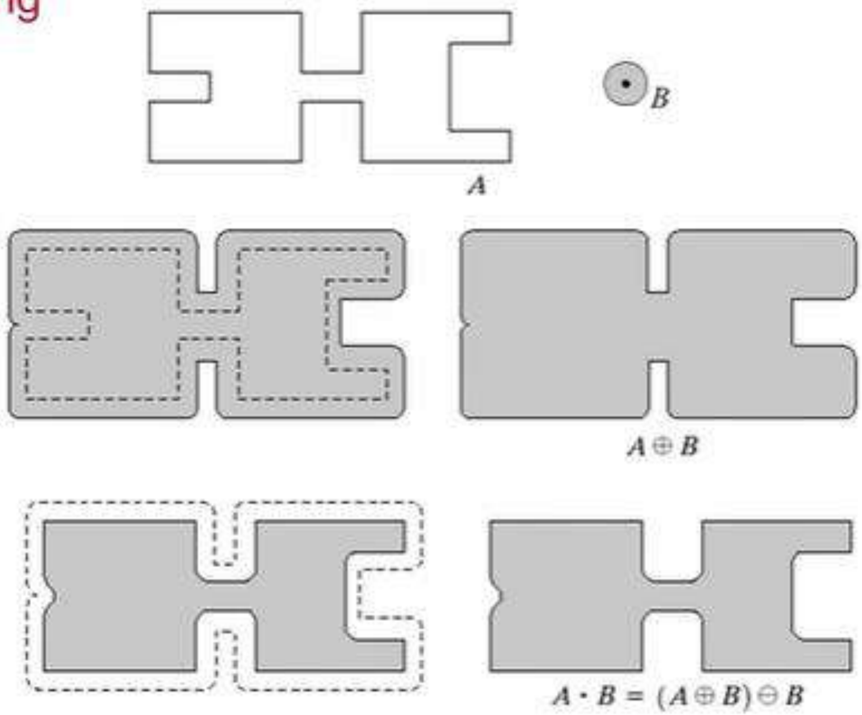
Morphological operations

❖ Closing

Applications

- Fuses narrow breaks & long thin gulfs
- Eliminates small holes
- Fills gap in contours

❖ Closing



❖ Closing

Properties

- A is a subset (sub image) of $A \bullet B$ ⇨ input is a subset of closing
- If C is subset of D, then $C \bullet B$ is subset of $D \bullet B$ ⇨ Monotonicity is preserved
- $(A \bullet B) \bullet B = A \bullet B$ ⇨applying more than one closing has no effect on the result

❖ Closing

Properties

- A is a subset (sub image) of $A \bullet B$ ⇨ input is a subset of closing
- If C is subset of D, then $C \bullet B$ is subset of $D \bullet B$ ⇨ Monotonicity is preserved
- $(A \bullet B) \bullet B = A \bullet B$ ⇨applying more than one closing has no effect on the result

❖ Relation between Opening & Closing

– Opening and closing are duals of each other with respect to set complementation and reflection

$$(A \bullet B)^c = A^c \circ \hat{B}$$

❖ Morphological filtering using Opening & Closing

a b
d c
e f

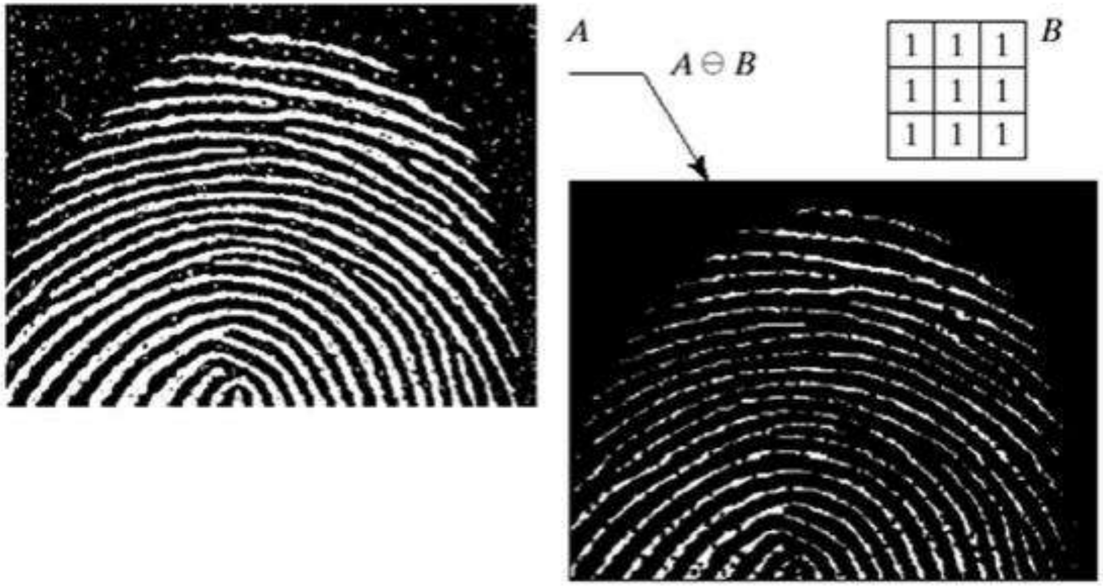
FIGURE 9.11
(a) Noisy image.
(c) Eroded image.
(d) Opening of *A*.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

❖ Morphological filtering using Opening & Closing

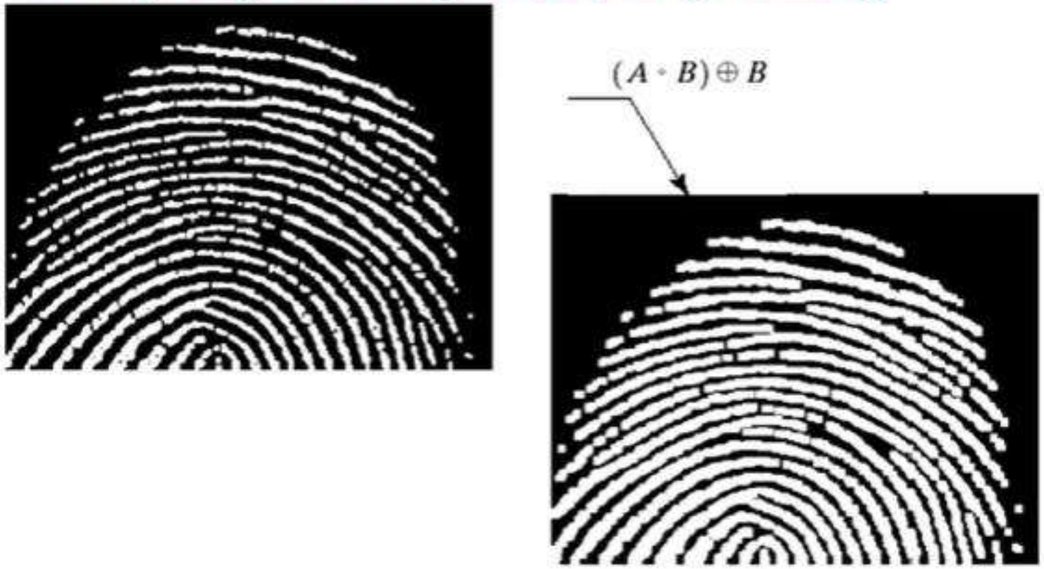
A

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

❖ Morphological filtering using Opening & Closing



❖ Morphological filtering using Opening & Closing



❖ Morphological filtering using Opening & Closing

$$[(A \circ B) \oplus B] \ominus B = (A \circ B) \bullet B$$

Problems Solving based on Dilution,Erosion,Opening and Closing

For the image and structuring element shown below, perform the operation of Dilation, Erosion, Opening and Closing.

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

Image

Center

1	1
1	0

Structuring Element

Dilation

Dilation of A by B

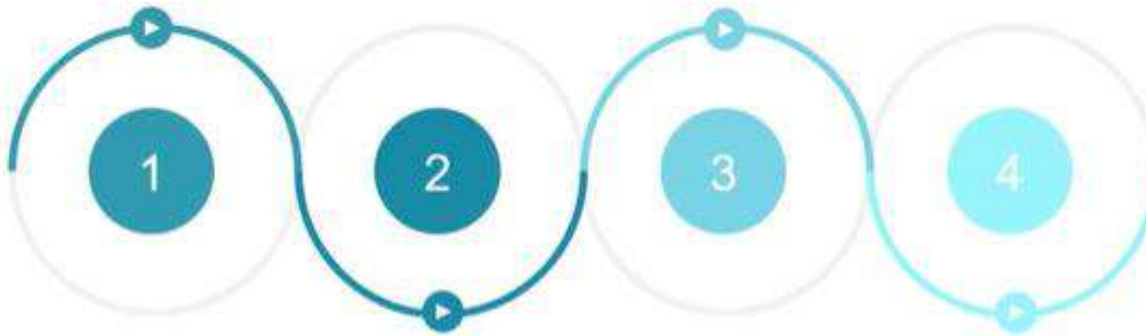
$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$$

where

$A = Image$

$B = Structuring\ Element$

$\Rightarrow$ *Set of all displacements z such that* $\hat{B}$ *and A overlap atleast by one point.*



Step 1

Find $\hat{B}$

Step 2

Pad image A with zeros to get A_{Pad}

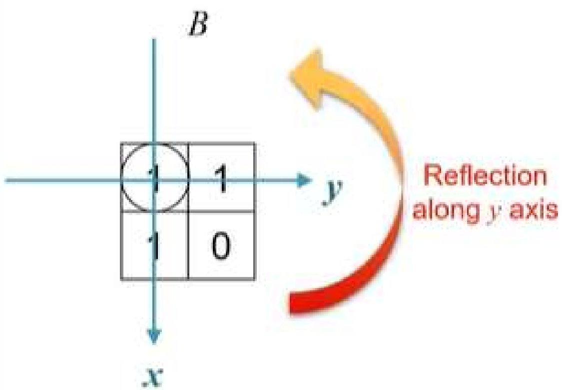
Step 3

Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad} (Keep center inside A)

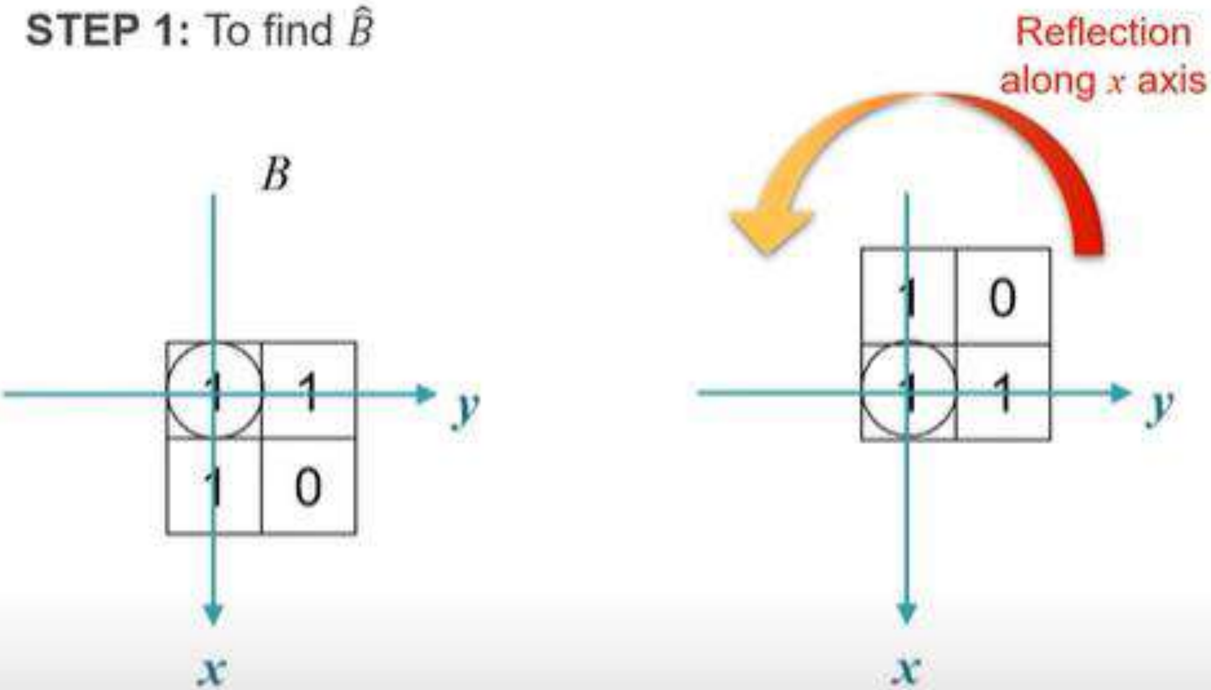
Step 4

Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

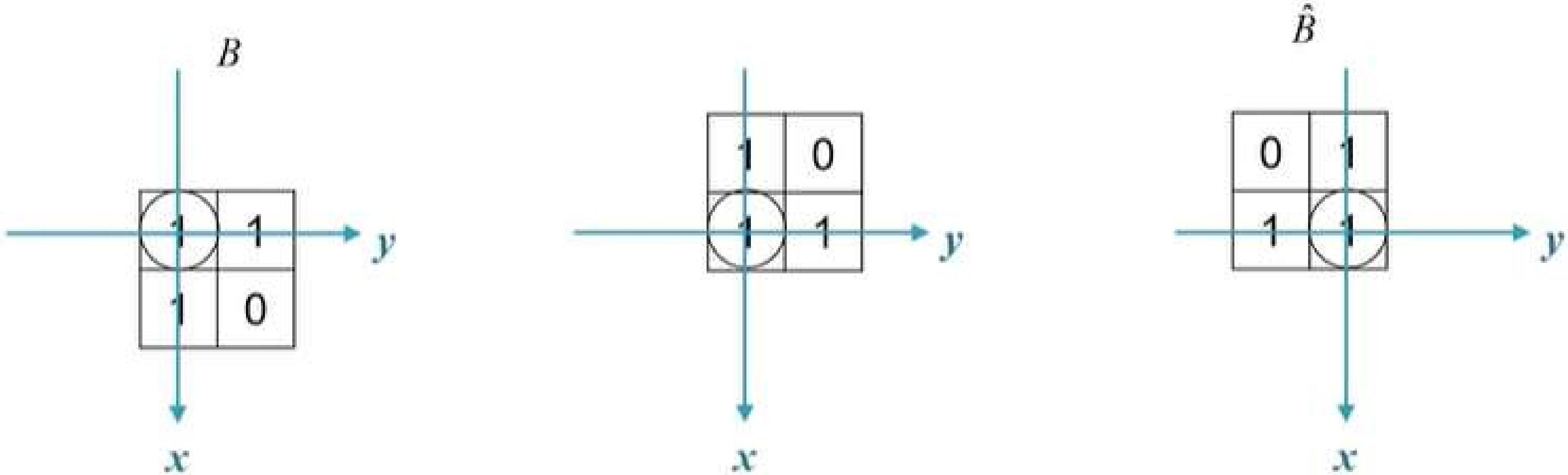
STEP 1: To find $\hat{B}$



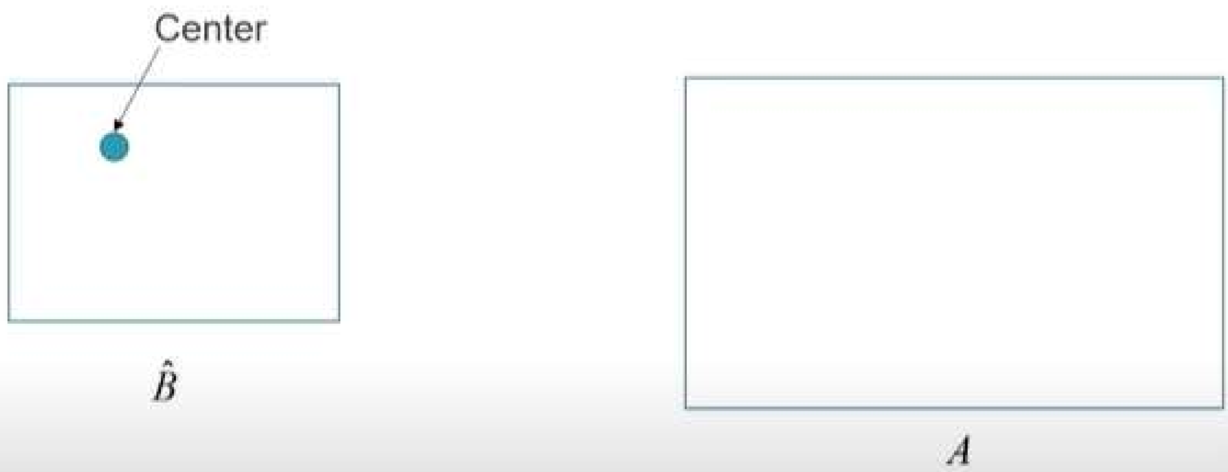
STEP 1: To find $\hat{B}$



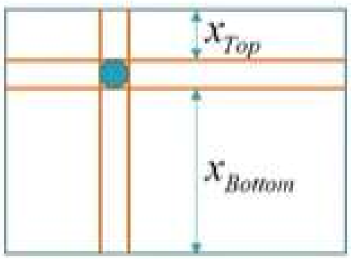
STEP 1: To find $\hat{B}$



STEP 2: Pad image A with zeros



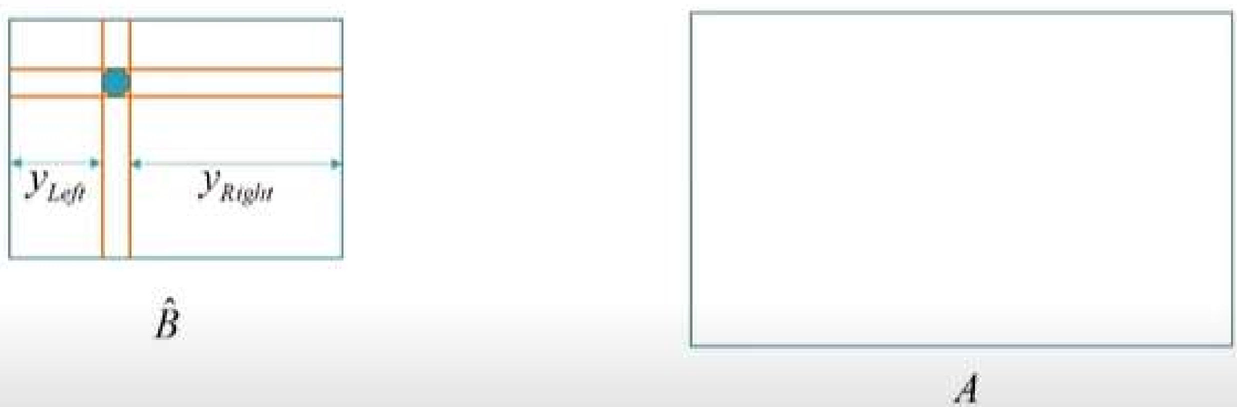
STEP 2: Pad image A with zeros



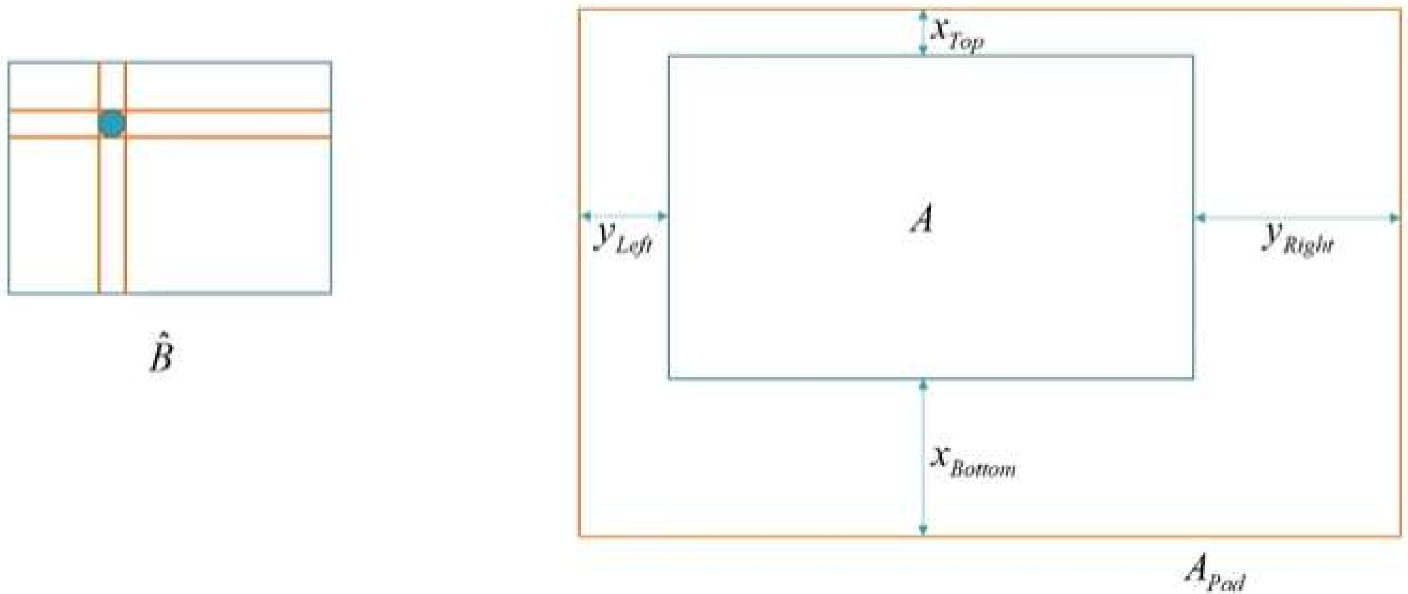
$\hat{B}$

A

STEP 2: Pad image A with zeros



STEP 2: Pad image A with zeros



STEP 2: Pad image A with zeros

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

A

0	1
1	1

$\hat{B}$

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

A_{Pad}

STEP 2: Pad image A with zeros

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

A

0	1
1	1

$\hat{B}$

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

A_{Pad}

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

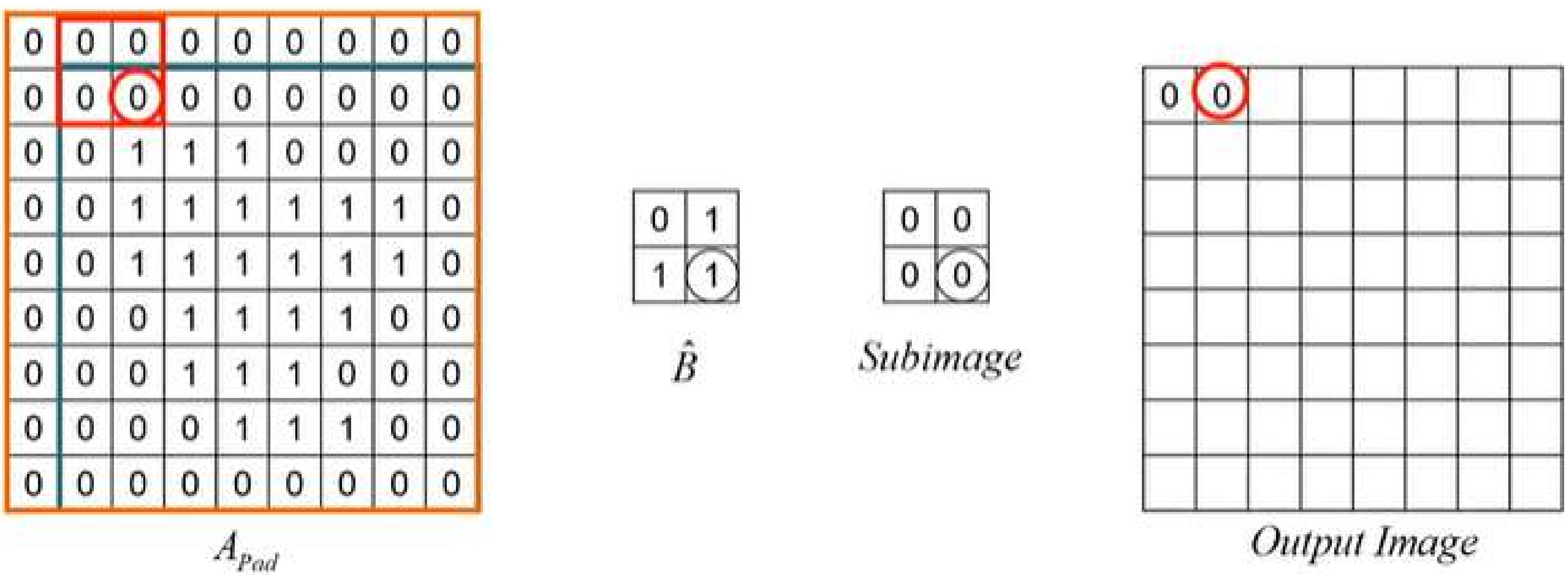
0	0
0	0

Subimage

0							

Output Image

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.



STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

0	0
0	0

Subimage

0	0	0	0	0	0	0	0

Output Image

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

0	0
0	0

Subimage

0	0	0	0	0	0	0	0
0							

Output Image

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

0	0
0	1

Subimage

0	0	0	0	0	0	0	0
0	1						

Output Image

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

0	0
1	1

Subimage

0	0	0	0	0	0	0	0
0	1	1					

Output Image

STEP 3: Keep center of $\hat{B}$ on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 4: Check if there is any overlap between values of A and $\hat{B}$. If there is overlap the corresponding output is 1 else it is 0.

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

A_{Pad}

0	1
1	1

$\hat{B}$

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

Output Image

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

A

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

Output Image

EROSION

Erosion of A by B

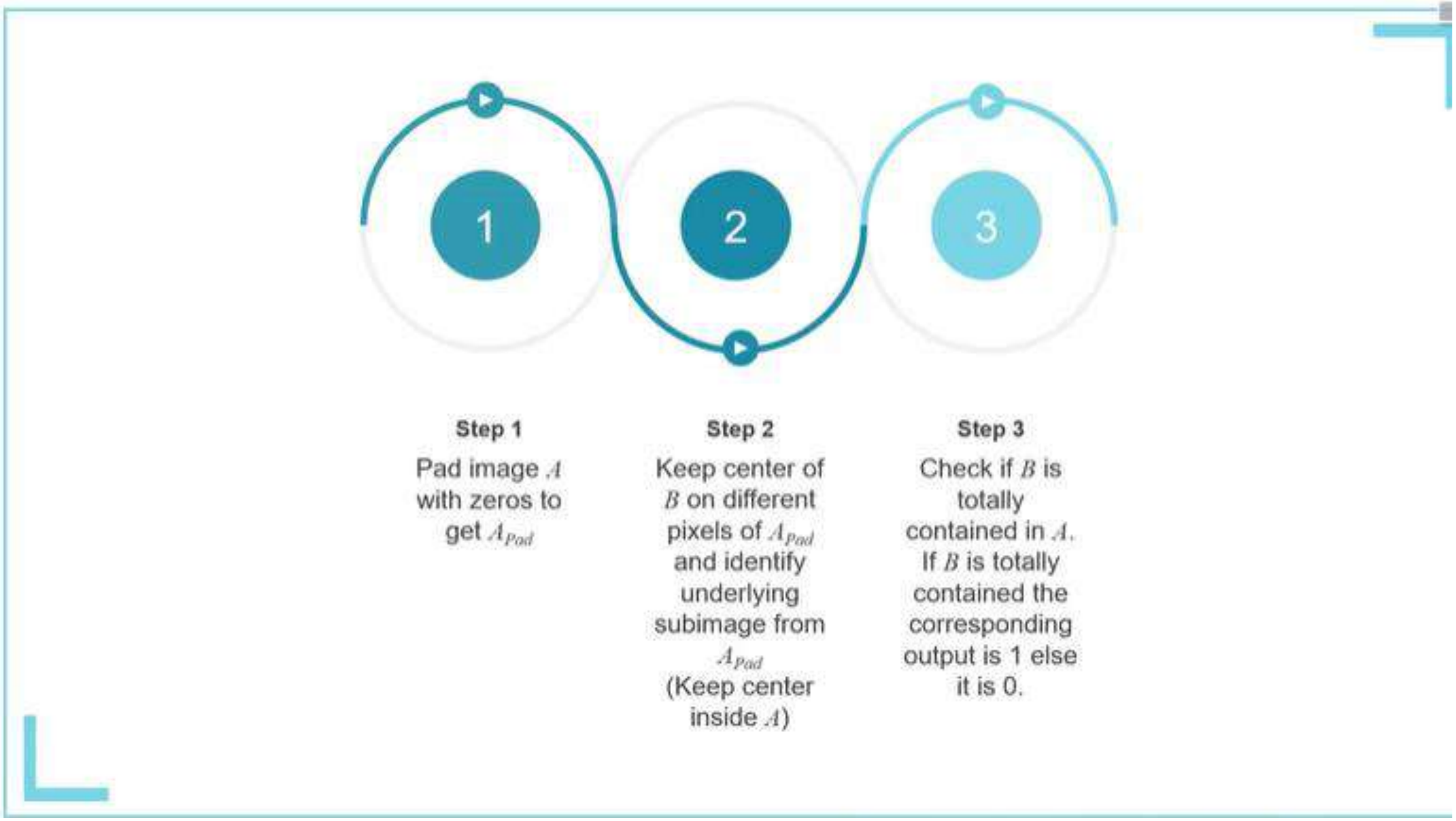
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

where

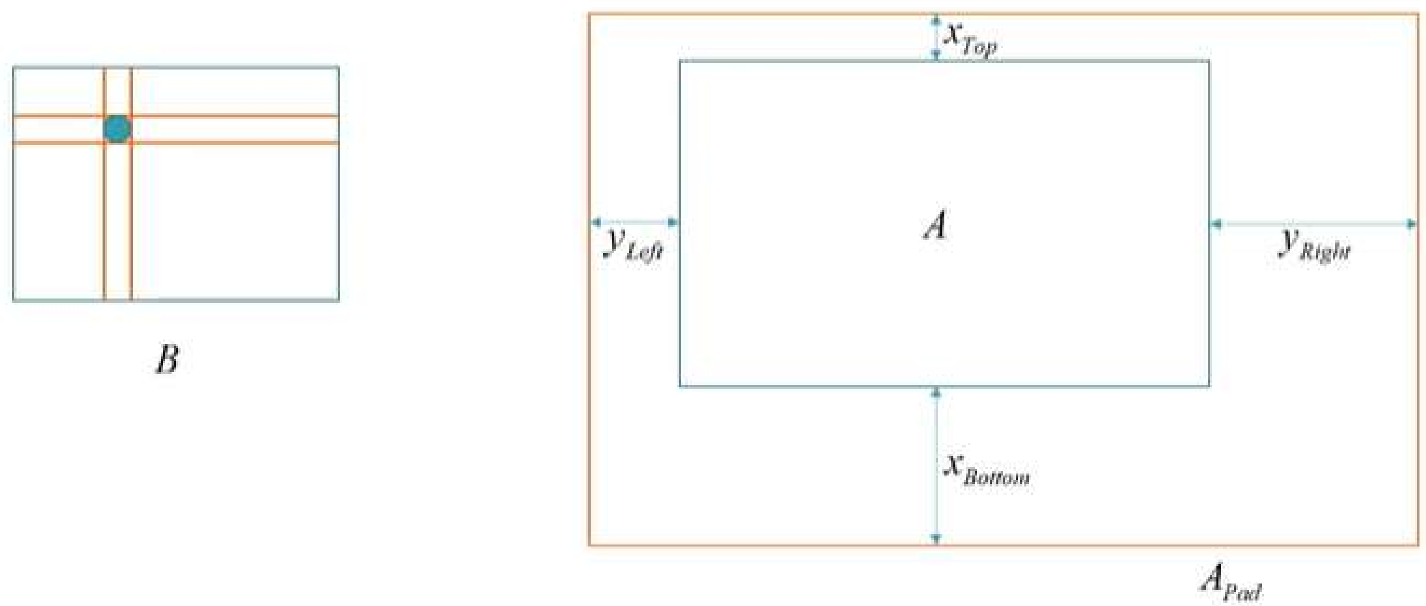
$$A = Image$$

$$B = Structuring\ Element$$

$\Rightarrow$ *Set of all displacements z such that B translated by z is completely contained in A.*


1
2
3
Step 1
Pad image A with zeros to get A_{Pad}
Step 2
Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad} (Keep center inside A)
Step 3
Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else it is 0.

STEP 1: Pad image A with zeros



STEP 1: Pad image A with zeros

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

A

1	1
1	0

B

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

A_{Pad}

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

B

0	0
0	1

Subimage

0							

Output Image

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

B

0	0
1	1

Subimage

0	0						

Output Image

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

B

0	0
0	0

Subimage

0	0	0	0	0	0	0	0

Output Image

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

B

0	1
0	1

Subimage

0	0	0	0	0	0	0	0
0							

Output Image

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}
STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

B

1	1
1	1

Subimage

0	0	0	0	0	0	0	0
0	1						

Output Image

STEP 2: Keep center of B on different pixels of A_{Pad} and identify underlying subimage from A_{Pad}

STEP 3: Check if B is totally contained in A. If B is totally contained the corresponding output is 1 else output is 0.

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

A_{Pad}

1	1
1	0

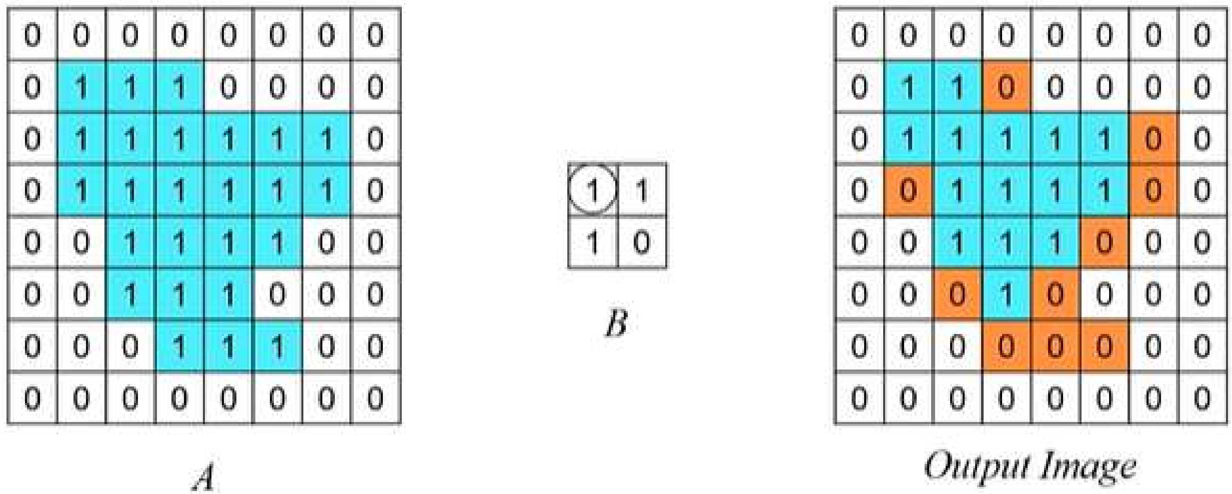
B

1	1
1	1

Subimage

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

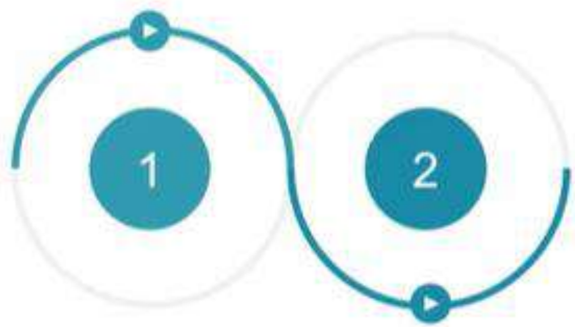
Output Image


A
B
Output Image

OPENING

Opening of A by structuring element B

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



Step 1

Erode A by B

Step 2

Dilate the result by B

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

A

1	1
1	0

B

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

$A \ominus B$

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

$A \ominus B$

1	1
1	0

B

0	1
1	1

$\hat{B}$

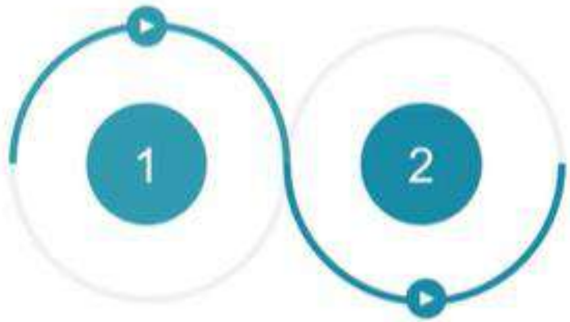
0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	1	1	1	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

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

CLOSING

Closing of A by structuring element B

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



Step 1
Dilate A by B

Step 2
Erode the result by B

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

A

1	1
1	0

B

0	1
1	1

$\hat{B}$

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

$A \oplus B$

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

$A \oplus B$

1	1
1	0

B

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

$A \bullet B = \left(A \oplus B\right) \ominus B$

Q2

Perform the Opening and Closing operation on the following image. Use structuring element as shown.

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

Image

1	1	1

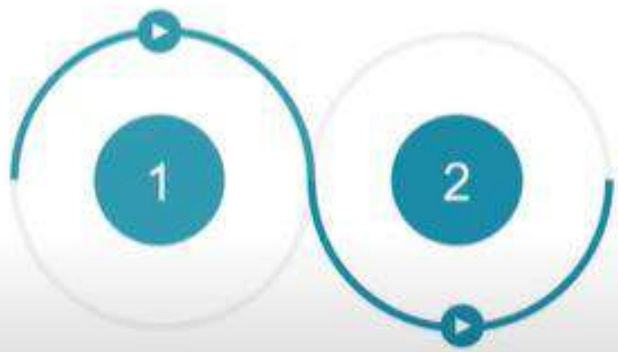
Structuring Element

Opening of A by structuring element B

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

where $A = Image$

$B = Structuring\ Element$



Step 1
Erode A by B

Step 2
Dilate the result by B

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

A

1	1	1

$B = \hat{B}$

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

A_{Pad}

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

A_{Pad}

1	1	1

B

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

$A \ominus B$

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

$A \ominus B$

1	1	1

$B = \hat{B}$

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

$(A \ominus B)_{Pad}$

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

$$\left(A \ominus B\right)_{Pad}$$

1	1	1

$$B = \hat{B}$$

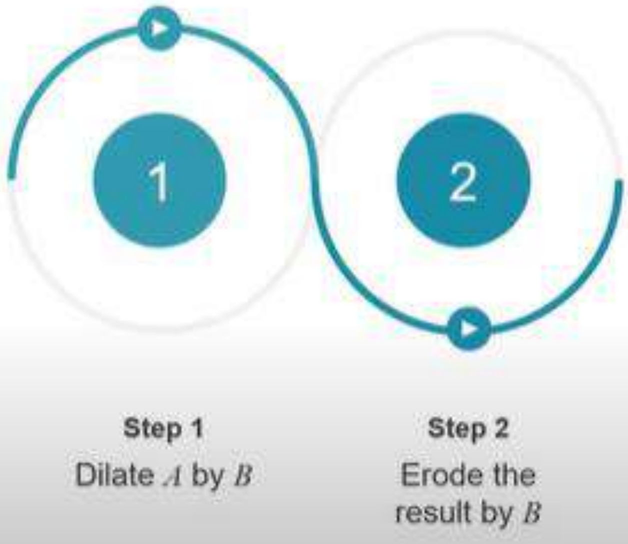
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

$$A \circ B = \left(A \ominus B\right) \oplus B$$

CLOSING

Closing of A by structuring element B

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



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

A

1	1	1

$B = \hat{B}$

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

A_{Pad}

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

A_{Pad}

1	1	1

$B = \hat{B}$

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

$A \oplus B$

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

$(A \oplus B)_{Pad}$

1	1	1

B

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

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

Thinning

Thinning

$\text{thin}(A,B) = A - (A \circledast B)$

thin(A,B)= A- hit-or-miss (A,B)

B1

0	0	0
x	1	x
1	1	1

B2

x	0	0
1	1	0
1	1	x

B3

1	x	0
1	1	0
1	x	0

B4

1	1	x
1	1	0
x	0	0

B5

1	1	1
x	1	x
0	0	0

B6

x	1	1
0	1	1
0	0	x

B7

0	x	1
0	1	1
0	x	1

B8

0	0	x
0	1	1
x	1	1

9.4 The Hit-or-Miss Transformation

- A basic morphological tool for **shape detection**.
- Let the origin of each shape be located at its center of gravity.
- If we want to find the location of a shape , say – **X** ,

  at (larger) image, say – **A** :
  - Let X be enclosed by a small window, say – **W**.
  - The ***local background*** of X with respect to W is defined as the *set difference* (**W - X**).
  - Apply *erosion* operator of A by X, will get us the set of locations of the origin of X, such that X is completely contained in A.
  - It may be also view geometrically as the set of all locations of the origin of X at which X found a match (**hit**) in A.

9.4 The Hit-or-Miss Transformation

<u>Cont.</u>

- Apply *erosion* operator on the *complement of A* by the *local background* set (W – X).
- Notice, that the set of locations for which X <u>**exactly**</u> fits inside A is the ***intersection*** of these two last operators above.

  This intersection is precisely the location sought.

<u>Formally:</u>

If B denotes the set composed of X and it's background –

B = (B1,B2) ; B1 = X , B2 = (W-X).

The match (or set of matches) of B in A, denoted $A \circledast B$ is:

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

Hit-or-Miss exp:

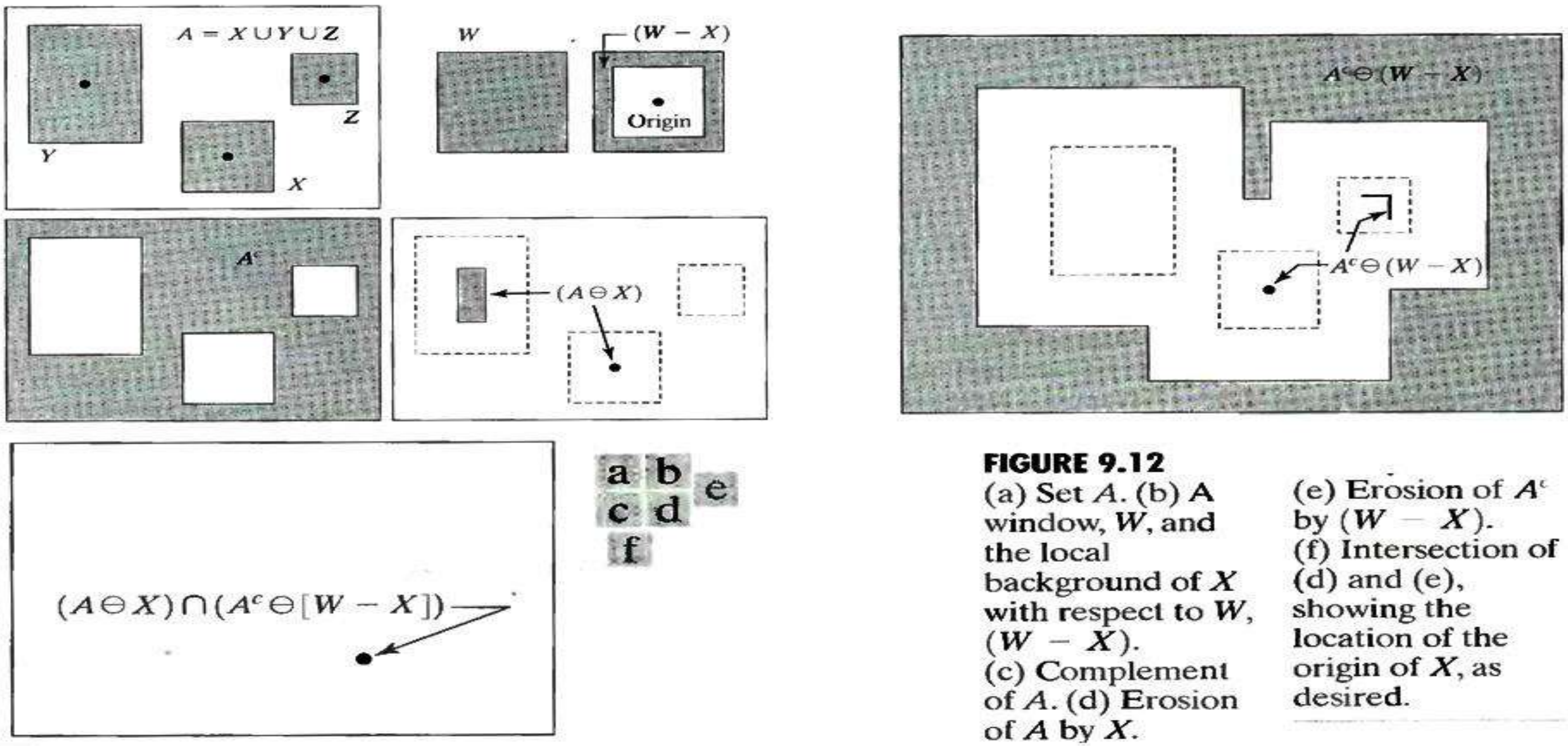


FIGURE 9.12 (a) Set A. (b) A window, W, and the local background of X with respect to W, $(W - X)$. (c) Complement of A. (d) Erosion of A by X. (e) Erosion of A^c by $(W - X)$. (f) Intersection of (d) and (e), showing the location of the origin of X, as desired.

9.4 The Hit-or-Miss Transformation

- The reason for using these kind of structuring element – B = (B1,B2) is based on an assumed definition that,

  two or more objects are distinct only if they are disjoint (disconnected) sets.
- In some applications , we may interested in detecting **certain patterns (combinations)** of 1's and 0's. and not for detecting individual objects.
- In this case a background is not required.

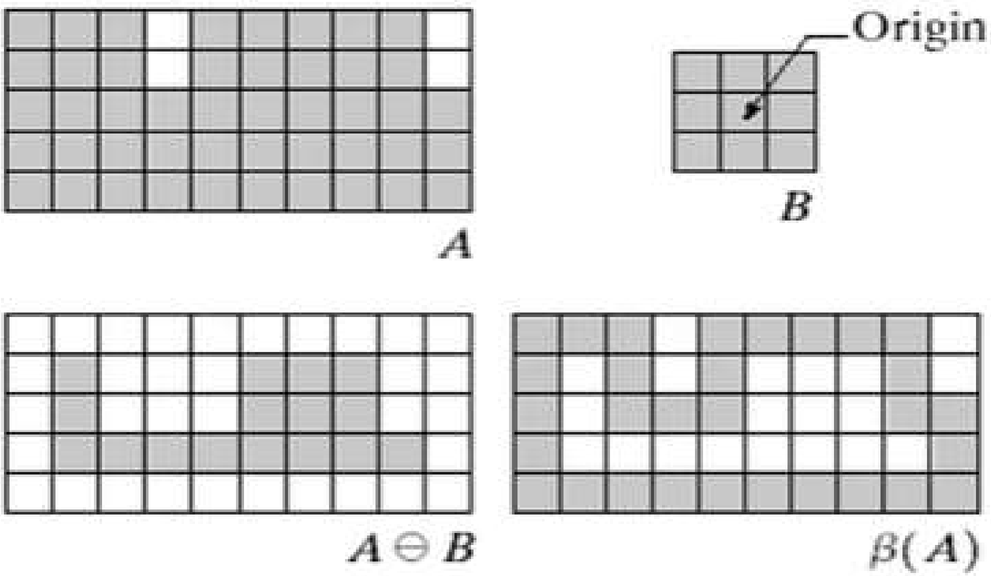
  and the *hit-or-miss transform* reduces to simple *erosion*.
- This simplified pattern detection scheme is used in some of the algorithms for **– identifying characters within a text**.

Basic Morphological Algorithms

Morphological Algorithms

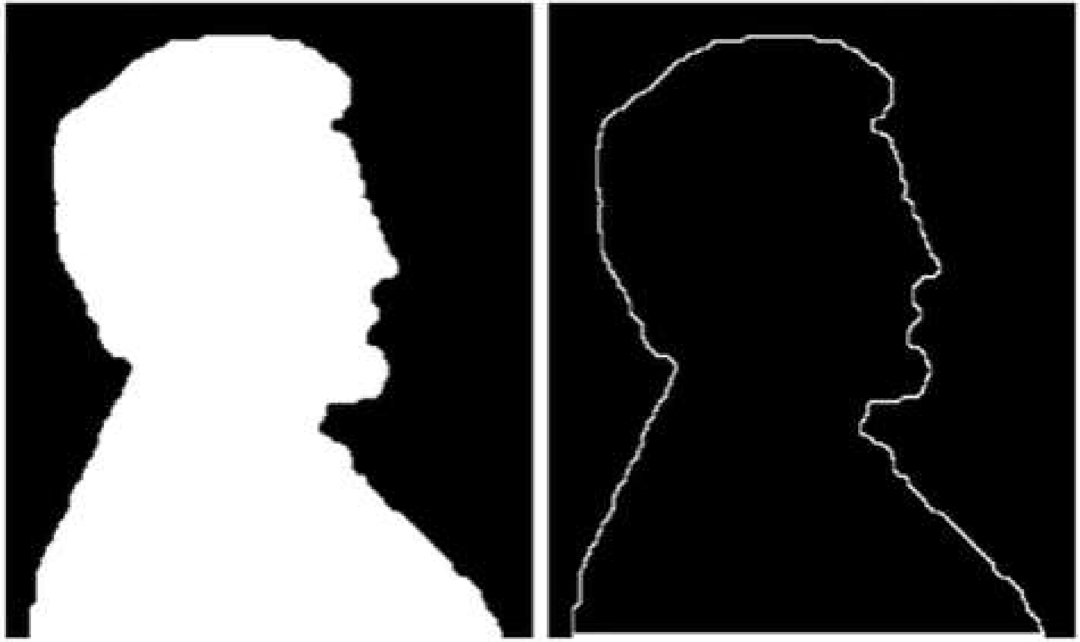
❖ Boundary Extraction

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



Morphological Algorithms

❖ Boundary Extraction



Morphological Algorithms

❖ Region Filling

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

where

$X_0 =$ *point inside the boundry to be filled*

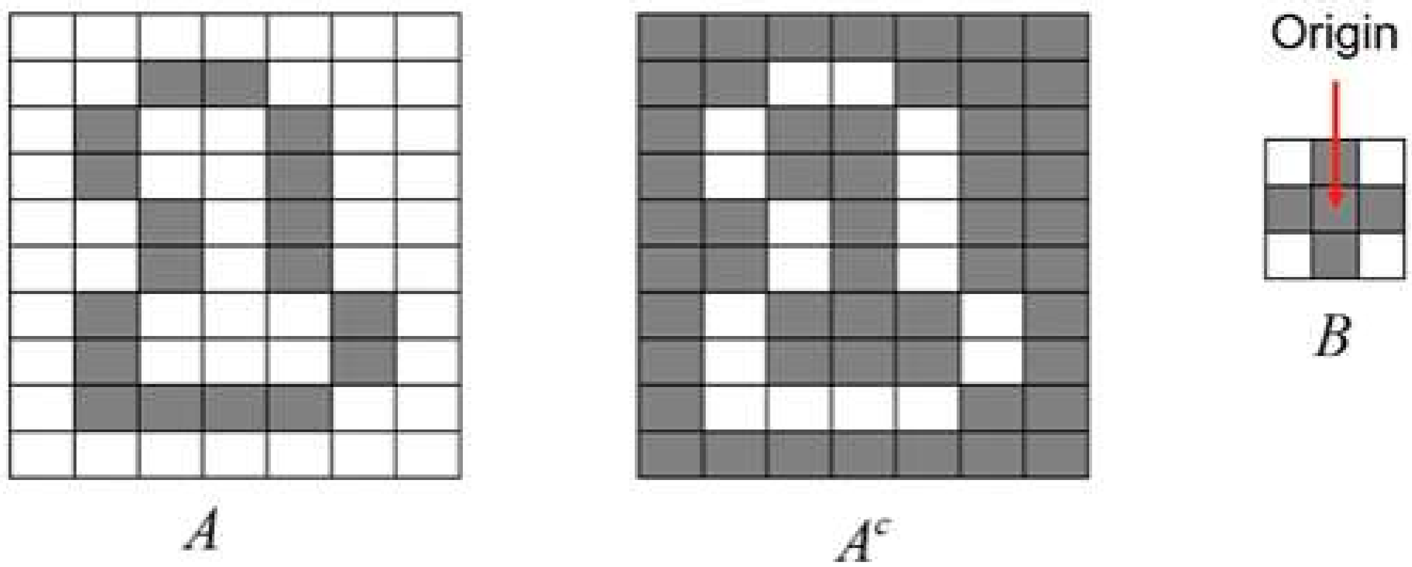
The algorithm terminates if $X_k = X_{k-1}$

$$A_{Filled} = A \cup X_k$$

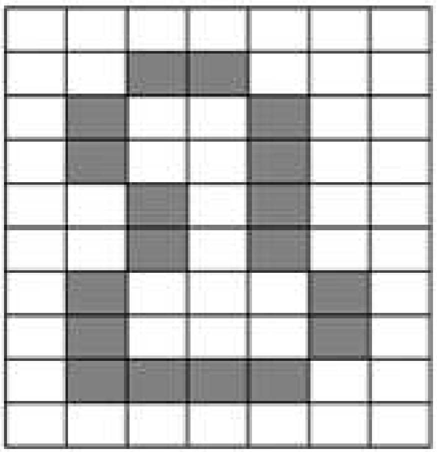
Morphological Algorithms

contd

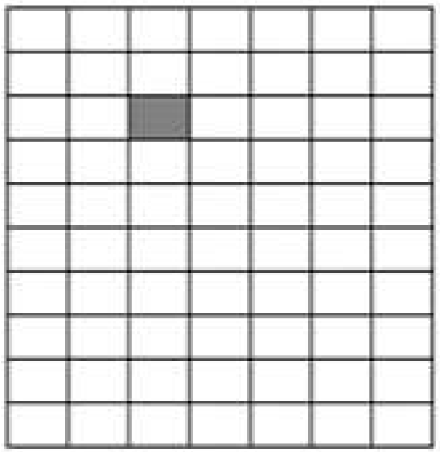
❖ Region Filling



❖ Region Filling



A

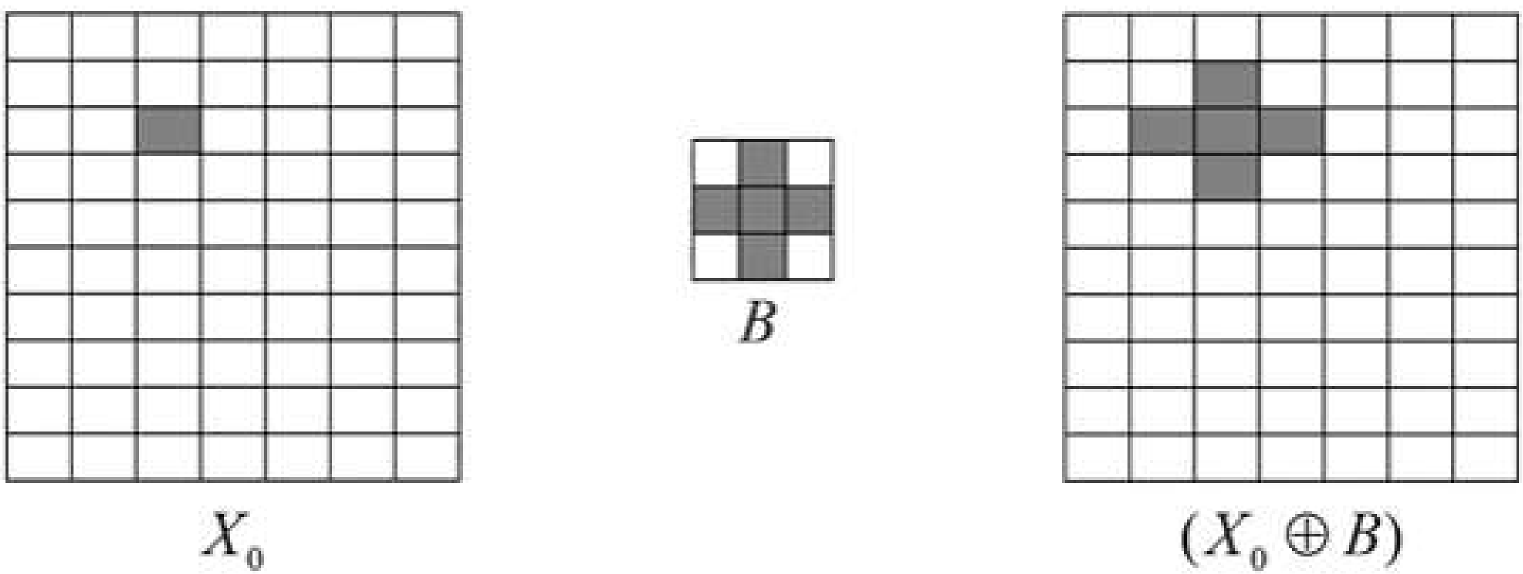


X_0

Morphological Algorithms

contd

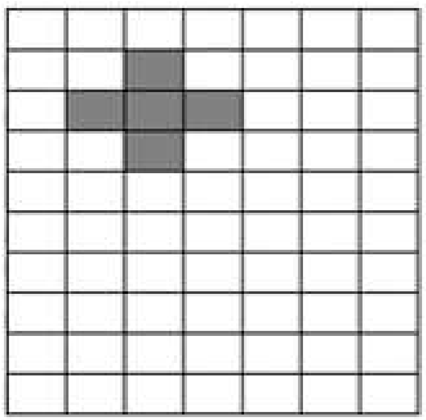
- Region Filling



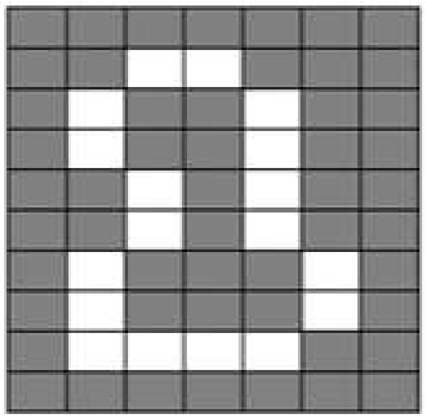
Morphological Algorithms

contd…

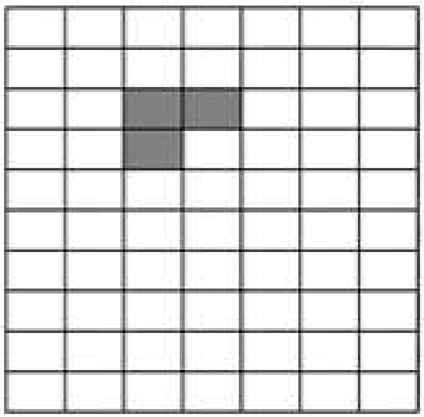
❖ Region Filling



$(X_0 \oplus B)$



A^c

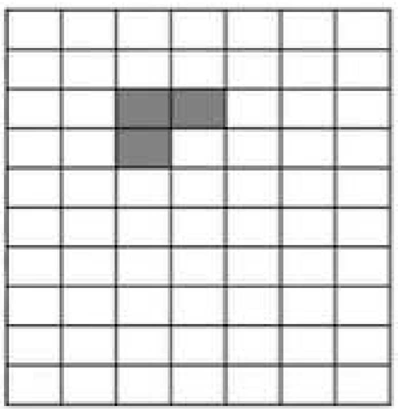


$X_1 = (X_0 \oplus B) \cap A^c$

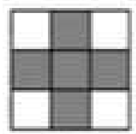
Morphological Algorithms

contd.

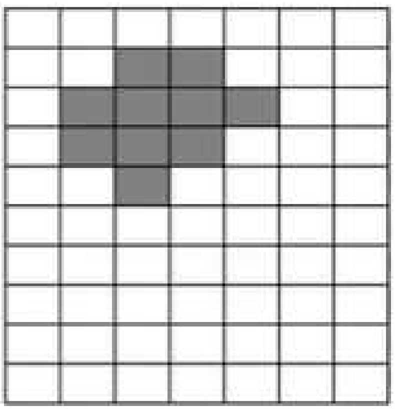
❖ Region Filling



X_1

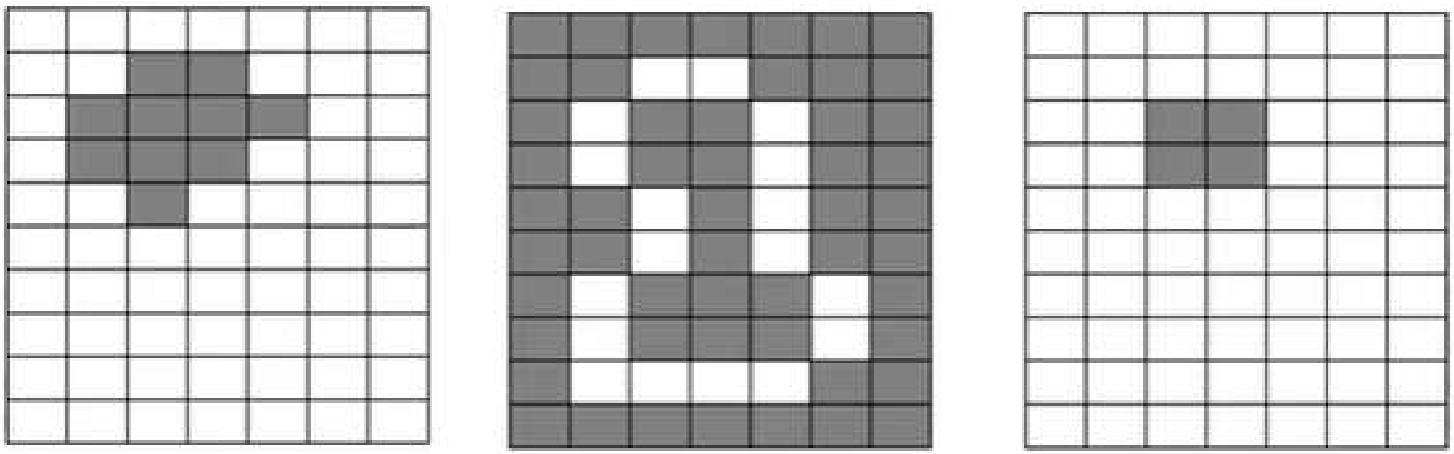


B



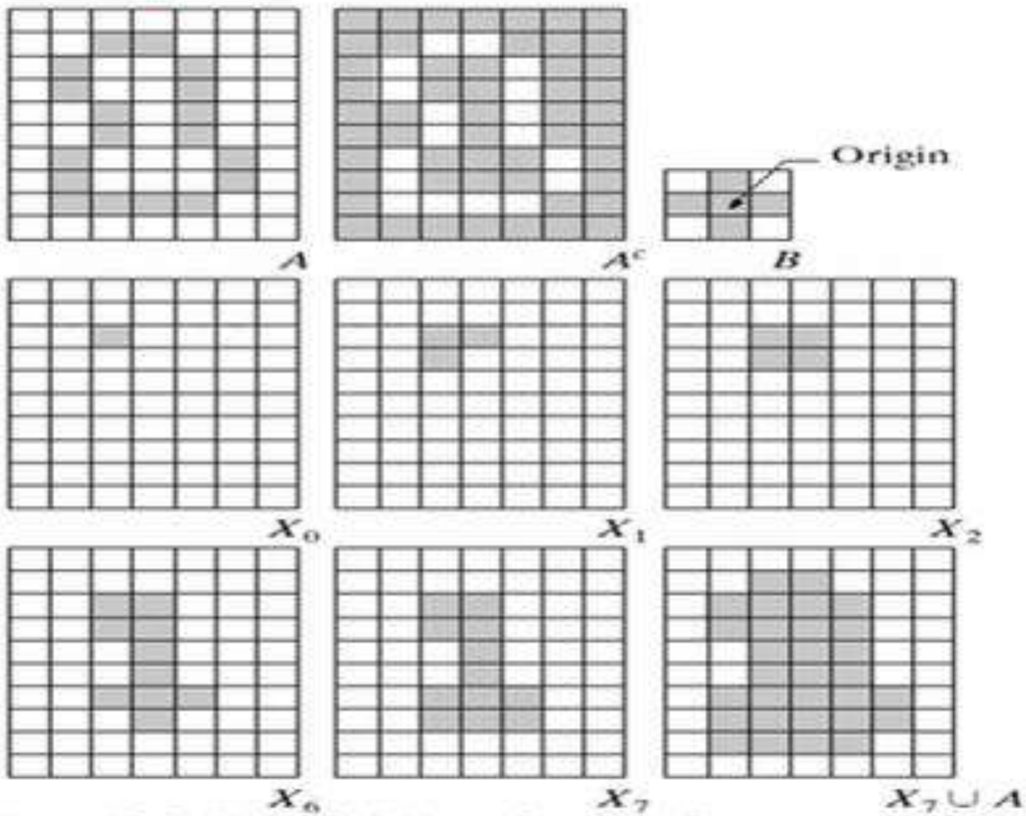
$(X_1 \oplus B)$

❖ Region Filling



$(X_1 \oplus B)$ $\quad$ A^c $\quad$ $X_2 = (X_1 \oplus B) \cap A^c$

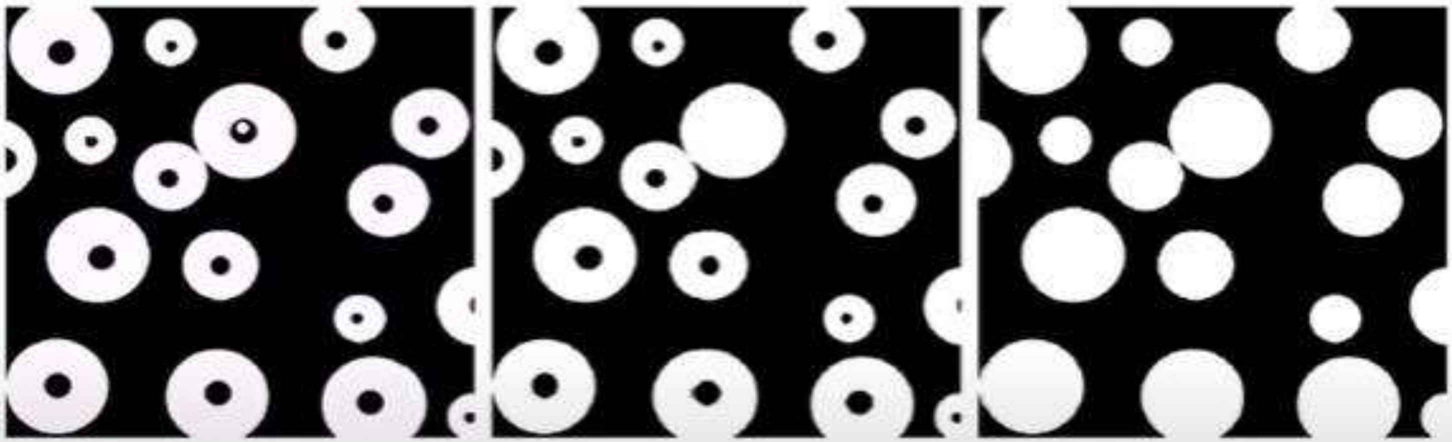
❖ Region Filling



Morphological Algorithms

contd…

❖ Region Filling



a b c

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

Morphological Algorithms

- Thinning

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c$$

In general

$$\{B\} = \{B^1, B^2, \cdots B^n\}$$

and B^i *is rotated version of* B^{i-1}

$$\Rightarrow A \otimes \{B\} = ((\cdots(A \otimes B^1) \otimes B^2) \cdots) \otimes B^n)$$

Morphological Algorithms

- ❖ Thinning

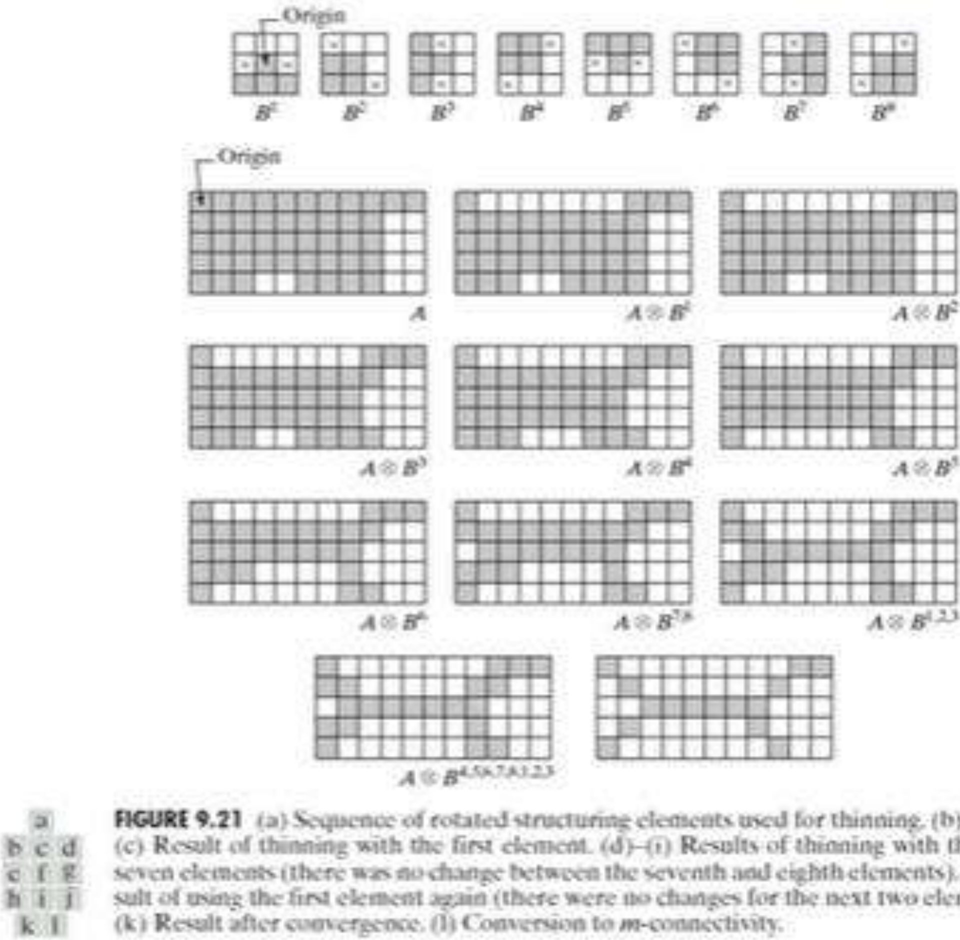


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m-connectivity.

Morphological Algorithms

Thickening

$$A \odot B = A \cup (A \circledast B)$$

In general

$$\{B\} = \{B^1, B^2, \cdots B^n\}$$

and B^i *is rotated version of* B^{i-1}

$$\Rightarrow A \odot \{B\} = ((\cdots(A \odot B^1) \odot B^2)\cdots) \odot B^n)$$

For implementation we form $C = A^c$, *thin* C *and then form* C^c

- Thickening

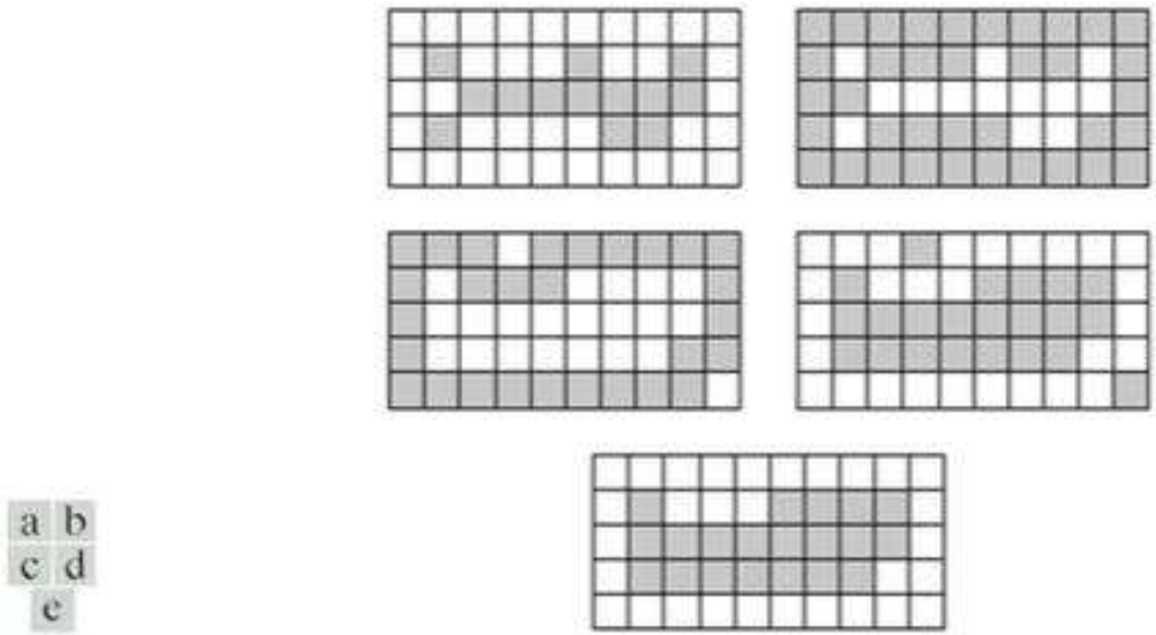


FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

The Hit-or-Miss Transformation

- A basic morphological tool for **shape detection**.
- Let the origin of each shape be located at its center of gravity.
- If we want to find the location of a shape , say – **X** ,

  at (larger) image, say – **A** :
  - Let X be enclosed by a small window, say – **W**.
  - The ***local background*** of X with respect to W is defined as the *set difference* (**W - X**).
  - Apply *erosion* operator of A by X, will get us the set of locations of the origin of X, such that X is completely contained in A.
  - It may be also view geometrically as the set of all locations of the origin of X at which X found a match (**hit**) in A.

The Hit-or-Miss Transformation

<u>Cont.</u>

- Apply *erosion* operator on the *complement of A* by the *local background* set (W – X).
- Notice, that the set of locations for which X **<u>exactly</u>** fits inside A is the ***intersection*** of these two last operators above.

  This intersection is precisely the location sought.

<u>Formally:</u>

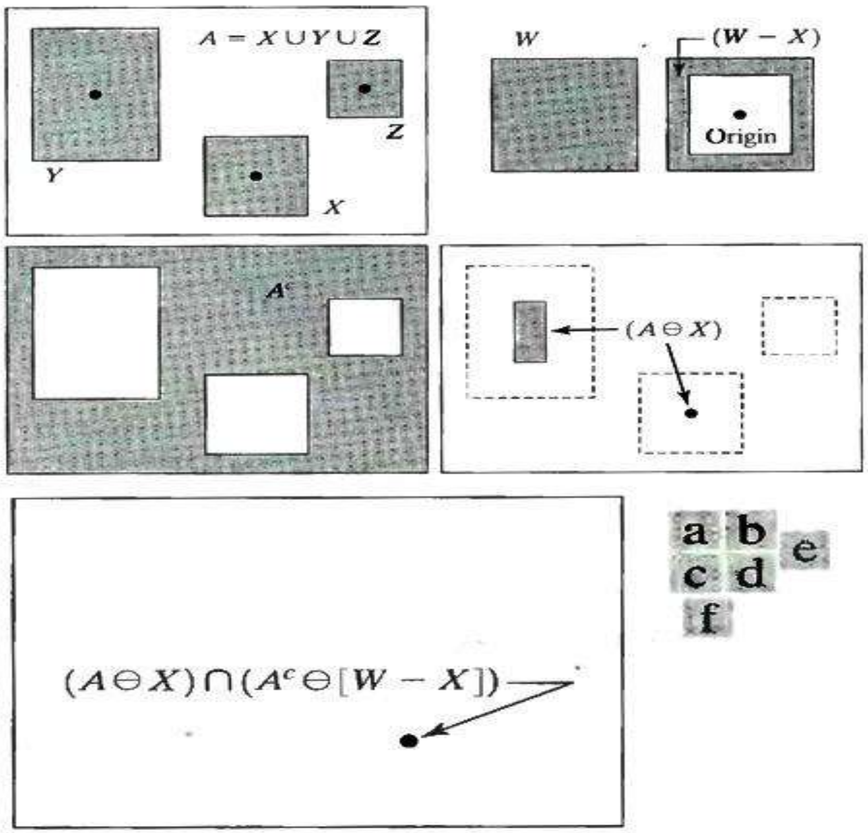
If B denotes the set composed of X and it's background –

B = (B1,B2) ; B1 = X , B2 = (W-X).

The match (or set of matches) of B in A, denoted $A \circledast B$ is:

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

Hit-or-Miss exp:



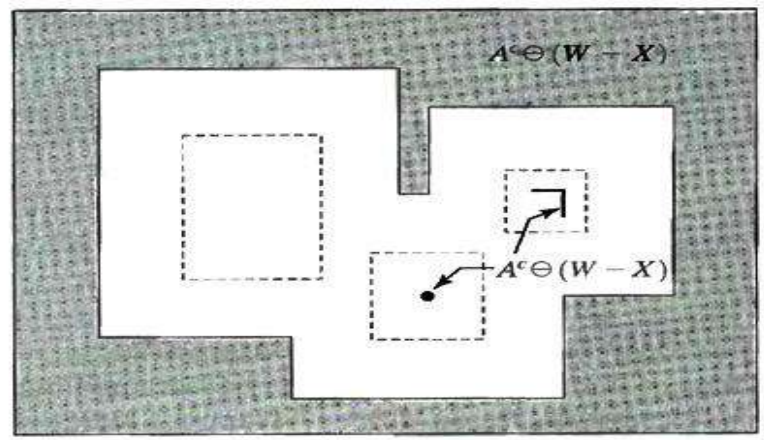


FIGURE 9.12
(a) Set A. (b) A window, W, and the local background of X with respect to W, $(W - X)$. (c) Complement of A. (d) Erosion of A by X. (e) Erosion of A^c by $(W - X)$. (f) Intersection of (d) and (e), showing the location of the origin of X, as desired.

The Hit-or-Miss Transformation

- The reason for using these kind of structuring element – B = (B1,B2) is based on an assumed definition that,

  two or more objects are distinct only if they are disjoint (disconnected) sets.
- In some applications , we may interested in detecting **certain patterns (combinations)** of 1'$_s$ and 0'$_s$. and not for detecting individual objects.
- In this case <u>a background is not required</u>.

  and the *hit-or-miss transform* reduces to <u>simple *erosion*</u>.
- This simplified pattern detection scheme is used in some of the algorithms for **– identifying characters within a text**.

Thinning

Thinning

thin(A,B)= $A - (A \circledast B)$

thin(A,B)= A- hit-or-miss (A,B)

B1

0	0	0
x	1	x
1	1	1

B2

x	0	0
1	1	0
1	1	x

B3

1	x	0
1	1	0
1	x	0

B4

1	1	x
1	1	0
x	0	0

B5

1	1	1
x	1	x
0	0	0

B6

x	1	1
0	1	1
0	0	x

B7

0	x	1
0	1	1
0	x	1

B8

0	0	x
0	1	1
x	1	1

Complete match=> Change centre pixel to 0; Else, pixel retains the same value as in the original image

Step-1 (B1 structuring element)

0	0	0	0	0	0	1	1
1	0	0	0	0	0	1	1
1	1	1	1	1	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	0
1	1	1	0	1	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	1	1
0	0	0	0	0	0	1	1
1	1	0	1	1	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	0
1	1	1	0	1	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	1	1
0	0	0	0	0	0	1	1
0	0	0	1	1	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	0
1	1	1	[illegible]	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

B1

0	0	0
x	1	x
1	1	1

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

Step-2 (B2 structuring element)
No change

Step-3 (B3 structuring element)
No change

Step-4 (B4 structuring element)

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

Step-5 (B5 structuring element)

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

Step-6 (B6 structuring element)
No change

Step-7 (B7 structuring element)
No change

Step-8 (B8 structuring element)

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

a. Successively erodes away foreground pixels
b. Preserves the end-points of line segments
c. Repeated thinning result in obtaining single-pixel wide skeleton (centre line of an object)

Thickening

$\text{thick}(A,B) = A \cup (A \circledast B)$

B1

0	0	0
x	0	x
1	1	1

B2

x	0	0
1	0	0
1	1	x

B3

1	x	0
1	0	0
1	x	0

B4

1	1	x
1	0	0
x	0	0

B5

1	1	1
x	0	x
0	0	0

B6

x	1	1
0	0	1
0	0	x

B7

0	x	1
0	0	1
0	x	1

B8

0	0	x
0	0	1
x	1	1

Complete match=> Change centre pixel to 1; Else, pixel retains the same value as in the original image

Color Image Processing

Spectrum of White Light

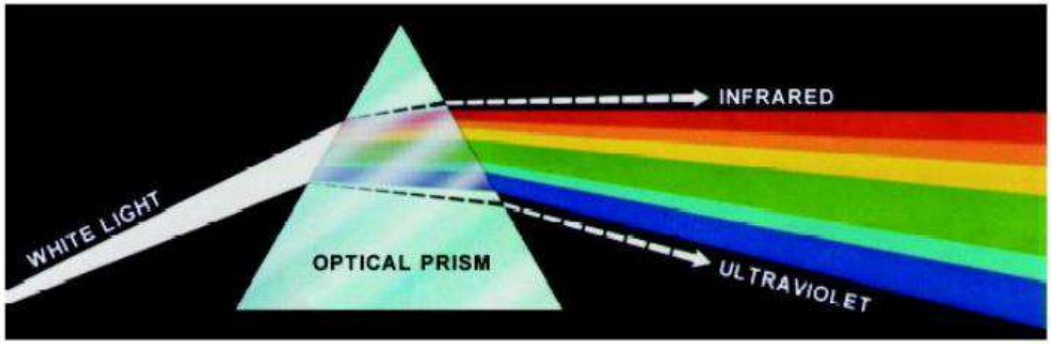
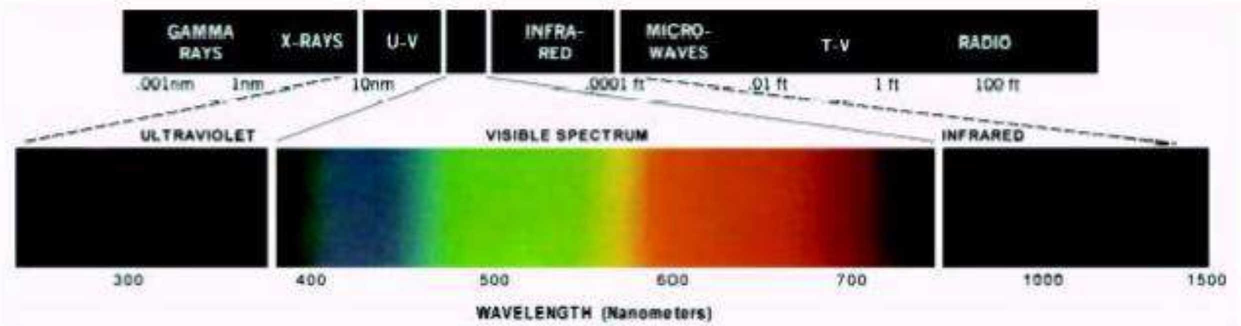


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

1666 Sir Isaac Newton, 24 year old, discovered white light spectrum.

Electromagnetic Spectrum



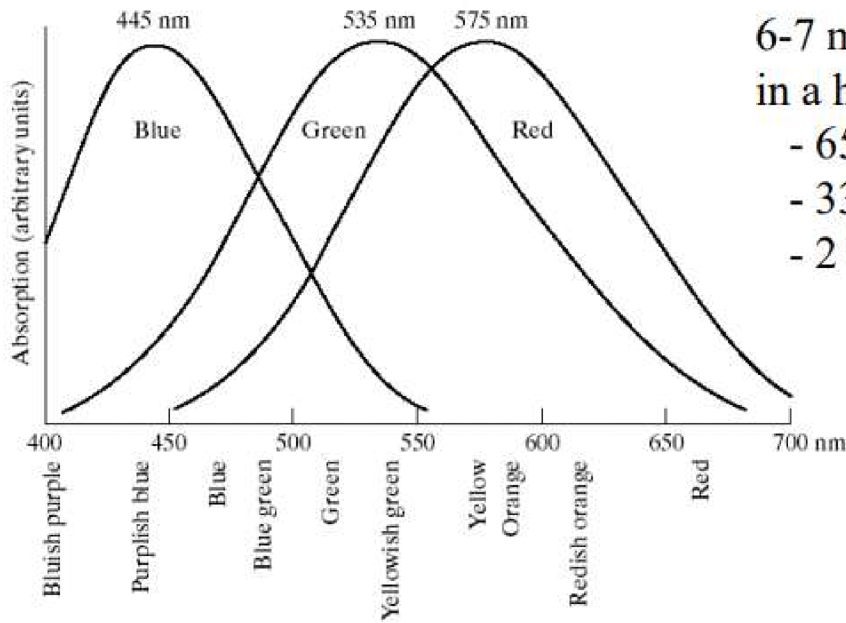
Visible light wavelength: from around 400 to 700 nm

1. For an achromatic (monochrome) light source, there is only 1 attribute to describe the quality: intensity

2. For a chromatic light source, there are 3 attributes to describe the quality:

Radiance = total amount of energy flow from a light source (Watts)
Luminance = amount of energy received by an observer (lumens)
Brightness = intensity

Sensitivity of Cones in the Human Eye



6-7 millions cones in a human eye

- 65% sensitive to Red light
- 33% sensitive to Green light
- 2 % sensitive to Blue light

Primary colors:
Defined CIE in 1931

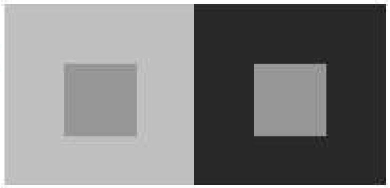
Red = 700 nm
Green = 546.1nm
Blue = 435.8 nm

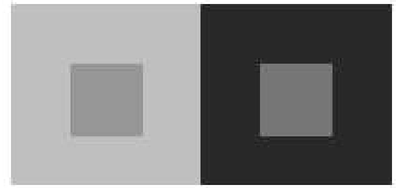
CIE = Commission Internationale de l'Eclairage
(The International Commission on Illumination)

(Images from Rafael C. Gonzalez and Richard E.

Luminance vs. Brightness

Same lum.
Different
brightness





Different lum.
Similar
brightness

- Luminance (or intensity)
  - Independent of the luminance of surroundings

$$L(x, y) = \int_0^{inf} I(x, y, \lambda)V(\lambda)d\lambda$$

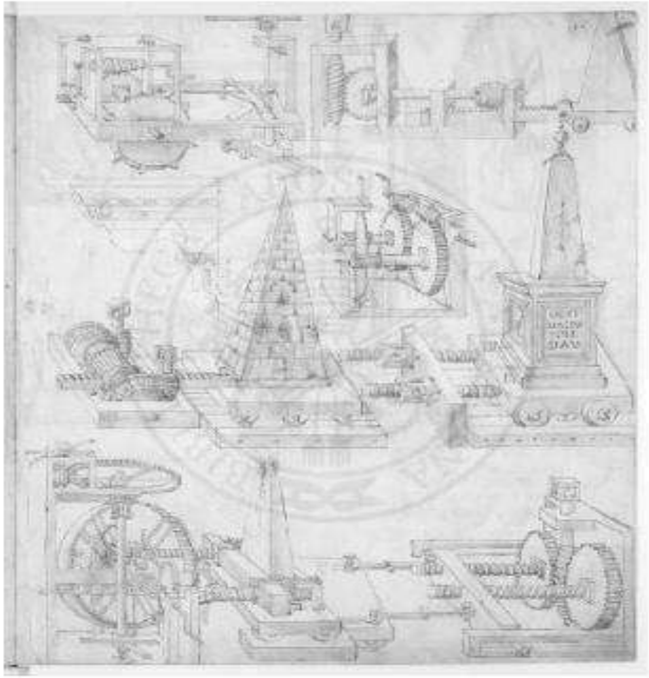
$I(x,y,\lambda)$ -- spatial light distribution

$V(\lambda)$ -- relative luminous efficiency func. of visual system ~ bell shape
(different for scotopic vs. photopic vision;
highest for green wavelength, second for red, and least for blue)

- Brightness
  - Perceived luminance
  - Depends on surrounding luminance

Luminance vs. Brightness (cont'd)

- Example: visible digital watermark
  - How to make the watermark appears the same graylevel all over the image?



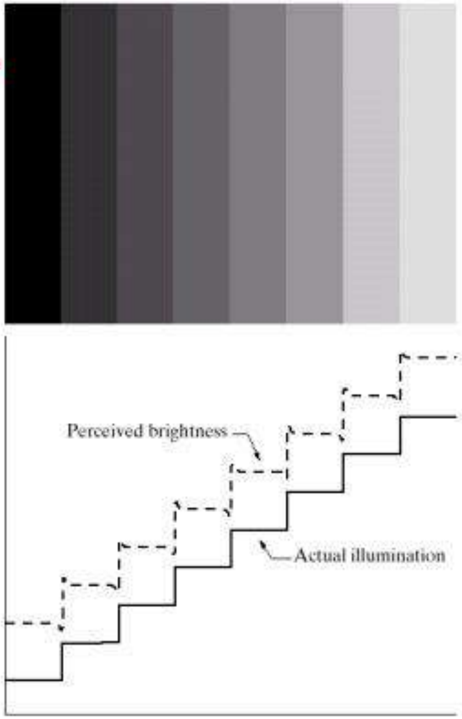
from `IBM Watson web page "Vatican Digital Library"`

Look into Simultaneous Contrast Phenomenon

- Human perception more sensitive to luminance contrast than absolute luminance
- Weber's Law: $| L_s - L_0 | / L_0 = \text{const}$
  - Luminance of an object (L_0) is set to be just noticeable from luminance of surround (L_s)
  - For just-noticeable luminance difference ΔL:

    $$\Delta L / L \approx d(\log L) \approx 0.02 \text{ (const)}$$

    - equal increments in log luminance are perceived as equally different
- Empirical luminance-to-contrast models
  - Assume $L \in [1, 100]$, and $c \in [0, 100]$
  - $c = 50 \log_{10} L$ (logarithmic law, widely used)
  - $c = 21.9 L^{1/3}$ (cubic root law)



a
b

FIGURE 2.7
(a) An example showing that perceived brightness is not a simple function of intensity. The relative vertical positions between the two profiles in (b) have no special significance; they were chosen for clarity.

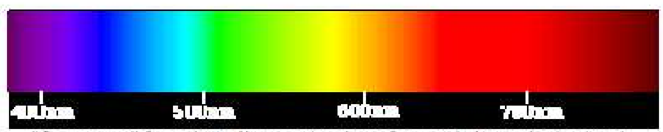
Figure is from slides at Gonzalez/ Woods DIP book website (Chapter 2)

- Visual system tends to undershoot or overshoot around the boundary of regions of different intensities

➔ Demonstrates the perceived brightness is not a simple function of light intensity

Color of Light

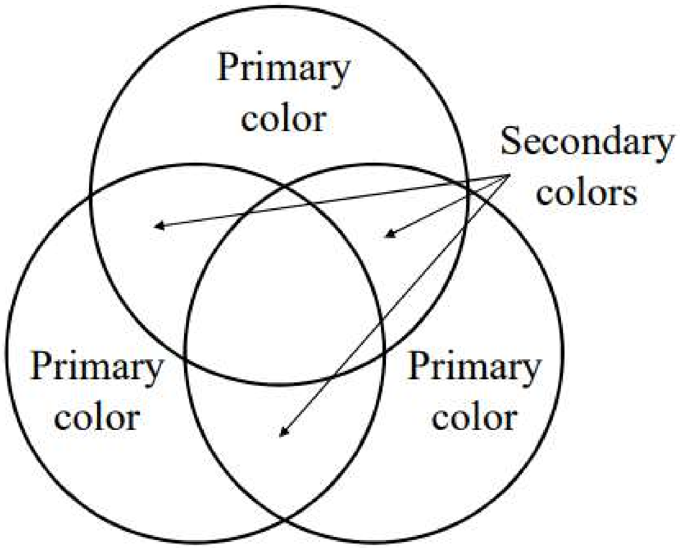
- Perceived color depends on spectral content (wavelength composition)
  - e.g., 700nm ~ red.
  - “spectral color”
    - A light with very narrow bandwidth



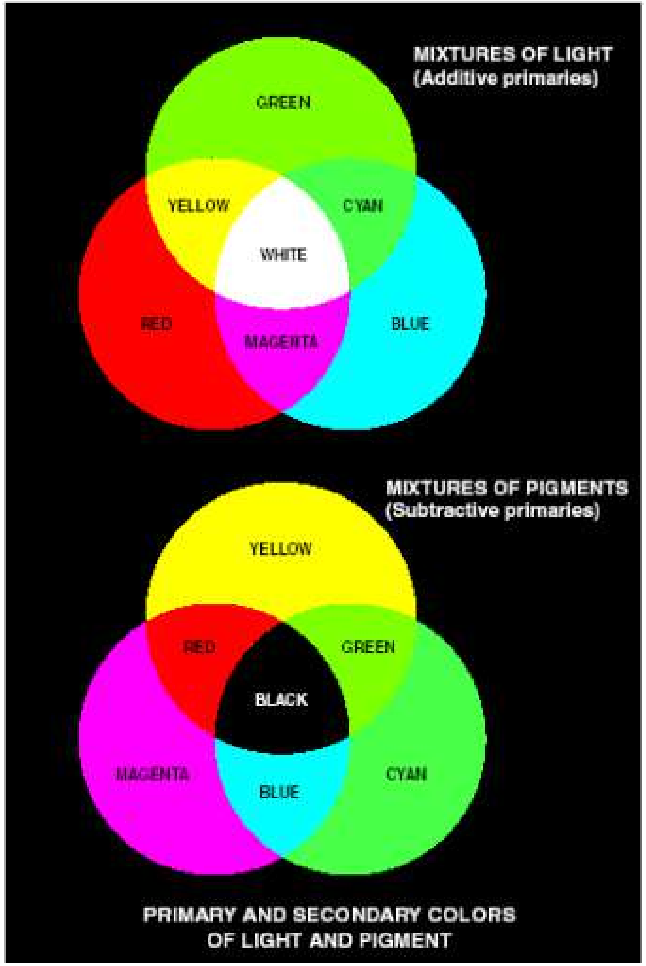
“Spectrum” from http://www.physics.sfasu.edu/astro/color.html

- A light with equal energy in all visible bands appears white

Primary and Secondary Colors



Primary and Secondary Colors (cont.)



Additive primary colors: RGB use in the case of light sources such as color monitors

RGB add together to get white

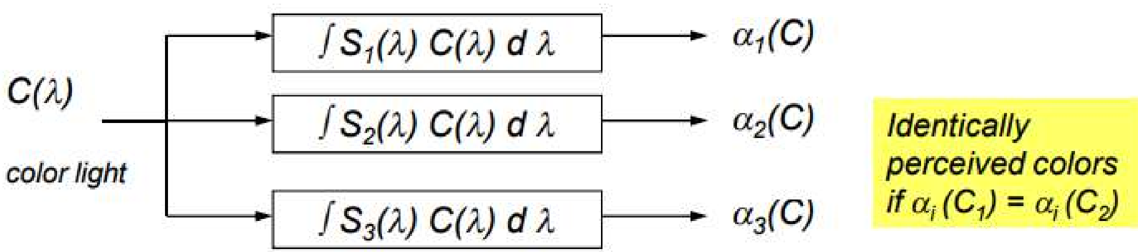
Subtractive primary colors: CMY use in the case of pigments in printing devices

White subtracted by CMY to get Black

(Images from Rafael C. Gonzalez and Richard E.

Representation by Three Primary Colors

- Any color can be reproduced by mixing an appropriate set of three primary colors (Thomas Young, 1802)
- Three types of cones in human retina
  - Absorption response $S_i(\lambda)$ has peaks around 450nm (blue), 550nm (green), 620nm (yellow-green)
  - Color sensation depends on the spectral response $\{\alpha_1(C), \alpha_2(C), \alpha_3(C)\}$ rather than the complete light spectrum $C(\lambda)$



Example: Seeing Yellow Without Yellow

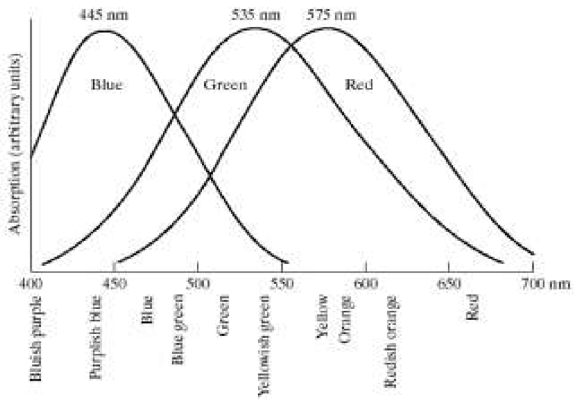
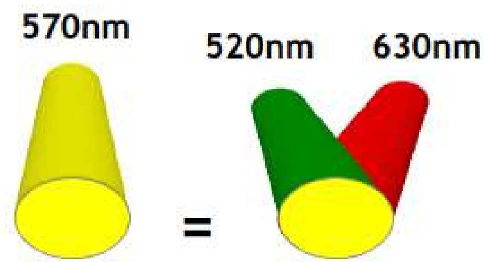


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.



mix <u>green and red</u> light to obtain perception of <u>yellow</u>, without shining a single yellow photon

Color Matching and Reproduction

- Mixture of three primaries: $C = Sum(\beta_k P_k(\lambda))$
- To match a given color C_1
  - adjust β_k such that $\alpha_i(C_1) = \alpha_i(C)$, $i = 1,2,3$.
- Tristimulus values $T_k(C)$
  - $T_k(C) = \beta_k / w_k$

    w_k – the amount of k^{th} primary to match the reference white
- Chromaticity $t_k = T_k / (T_1+T_2+T_3)$
  - $t_1+t_2+t_3 = 1$
  - *visualize (t_1, t_2) to obtain chromaticity diagram*

Color Characterization

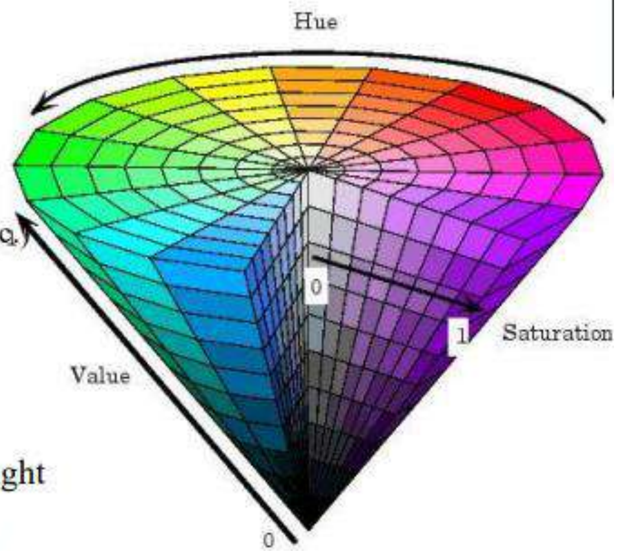
Hue:	dominant color corresponding to a dominant wavelength of mixture light wave
Saturation:	Relative purity or amount of white light mixed with a hue (inversely proportional to amount of white light added)
Brightness:	Intensity

$$\left.\begin{array}{l}\text{Hue}\\ \text{Saturation}\end{array}\right\}\ \text{Chromaticity}$$

amount of red (X), green (Y) and blue (Z) to form any particular color is called *tristimulus.*

Perceptual Attributes of Color

- Value of Brightness (perceived luminance)
- Chrominance
  - **Hue**
    - specify color tone (redness, greenness, etc.)
    - depend on peak wavelength
  - **Saturation**
    - describe how pure the color is
    - depend on the spread (bandwidth) of light spectrum
    - reflect how much white light is added
- RGB ⇔ HSV Conversion ~ *nonlinear*

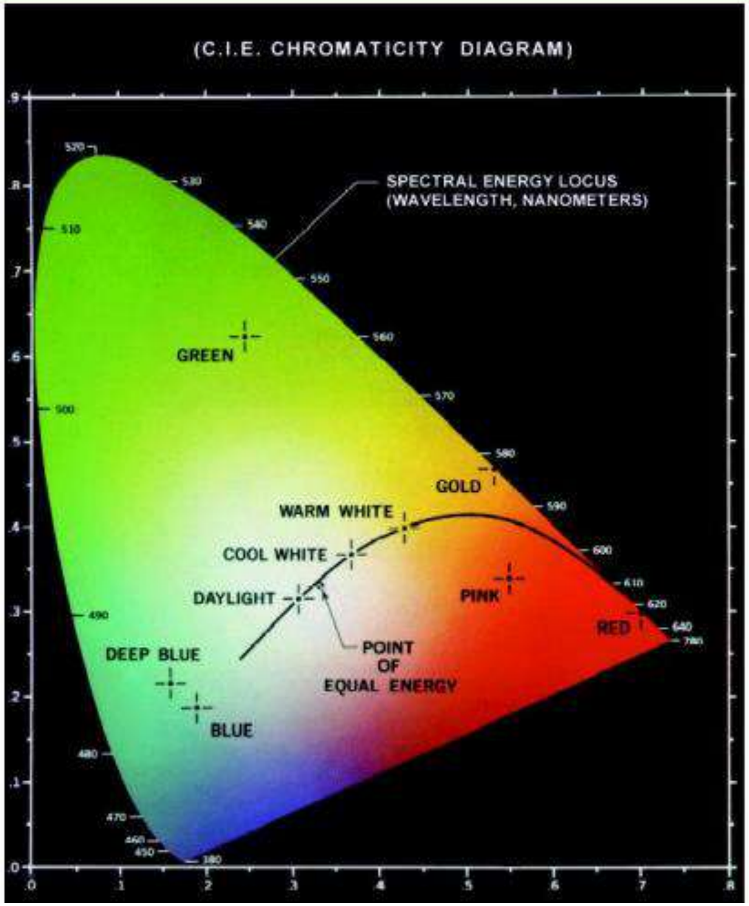


HSV circular cone is from online documentation of Matlab image processing toolbox

http://www.mathworks.com/access/helpdesk/help/toolbox/images/color10.shtml

CIE Chromaticity Diagram

y



Trichromatic coefficients:

$$x = \frac{X}{X+Y+Z}$$

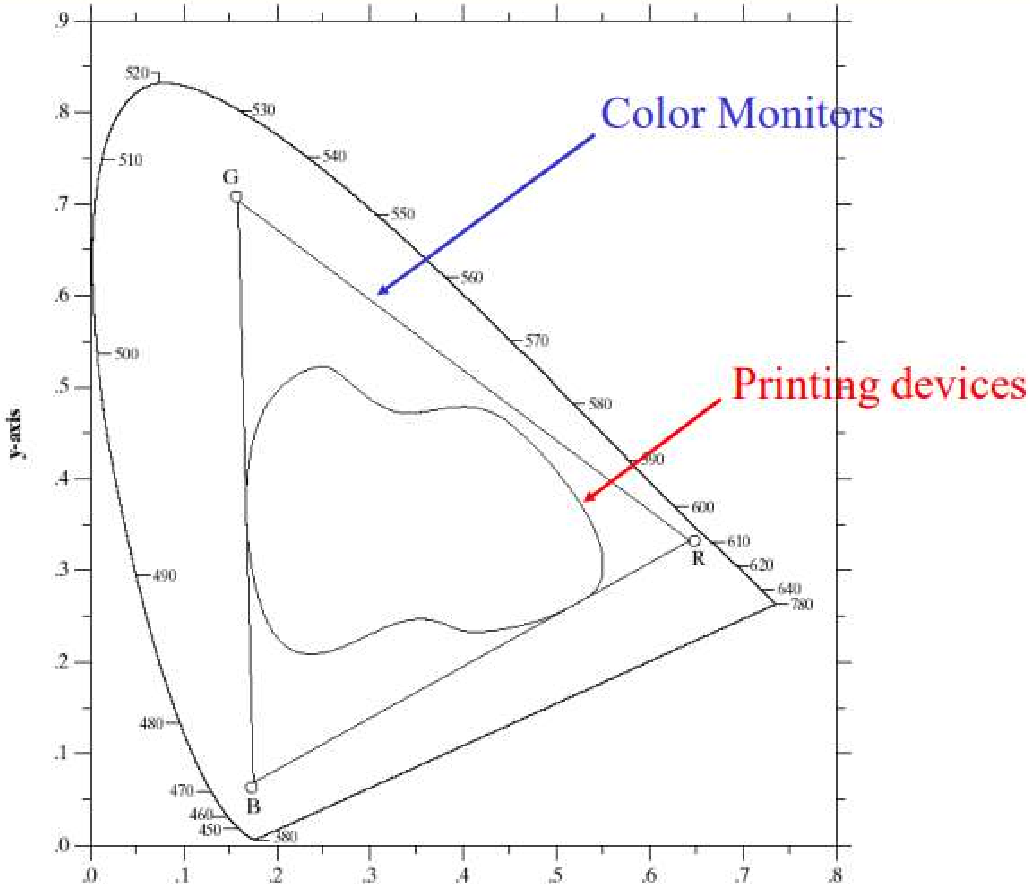
$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

$$x + y + z = 1$$

Points on the boundary are fully saturated colors

Color Gamut of Color Monitors and Printing Devices

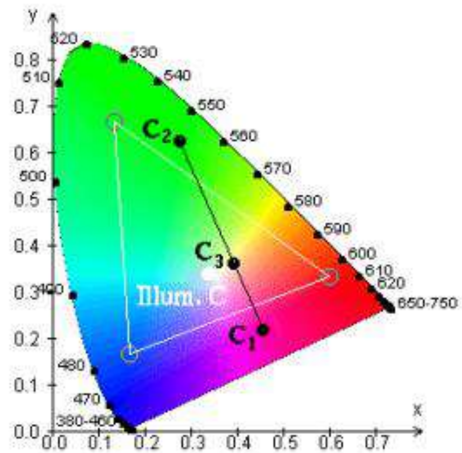


CIE Color Coordinates (cont'd)

- CIE XYZ system

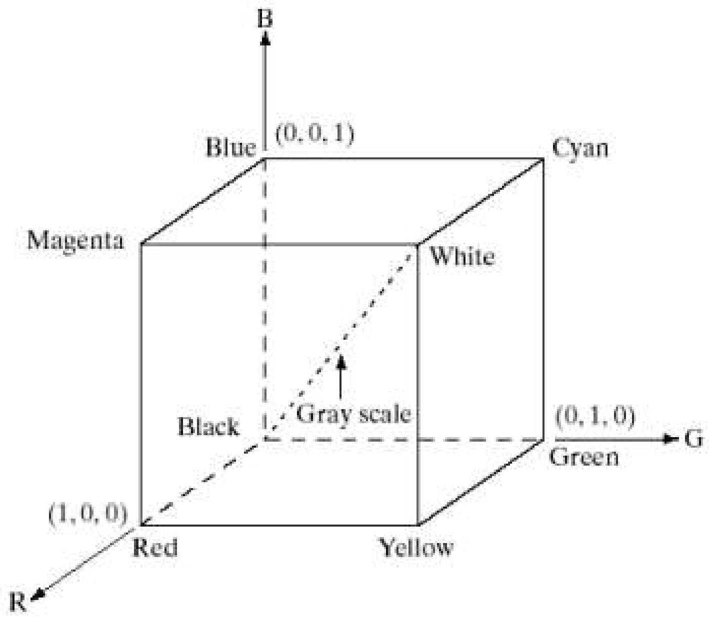
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

  - hypothetical primary sources to yield all-positive spectral tristimulus values
  - Y ~ luminance
- Color gamut of 3 primaries
  - Colors on line C1 and C2 can be produced by linear mixture of the two
  - Colors inside the triangle gamut can be reproduced by three primaries



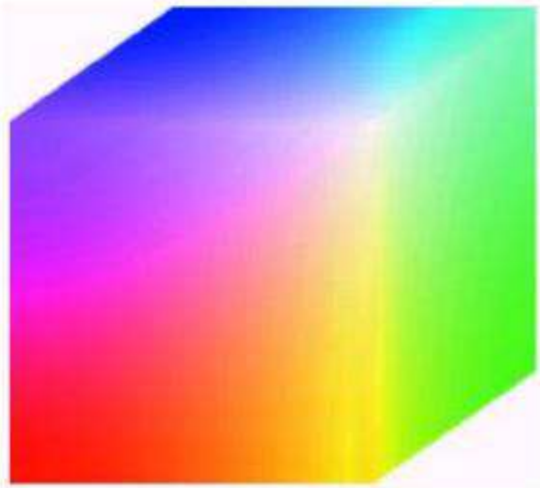
RGB Color Model

Purpose of color models: to facilitate the specification of colors in some standard



RGB color models:
- based on cartesian coordinate system

RGB Color Cube



R = 8 bits
G = 8 bits
B = 8 bits

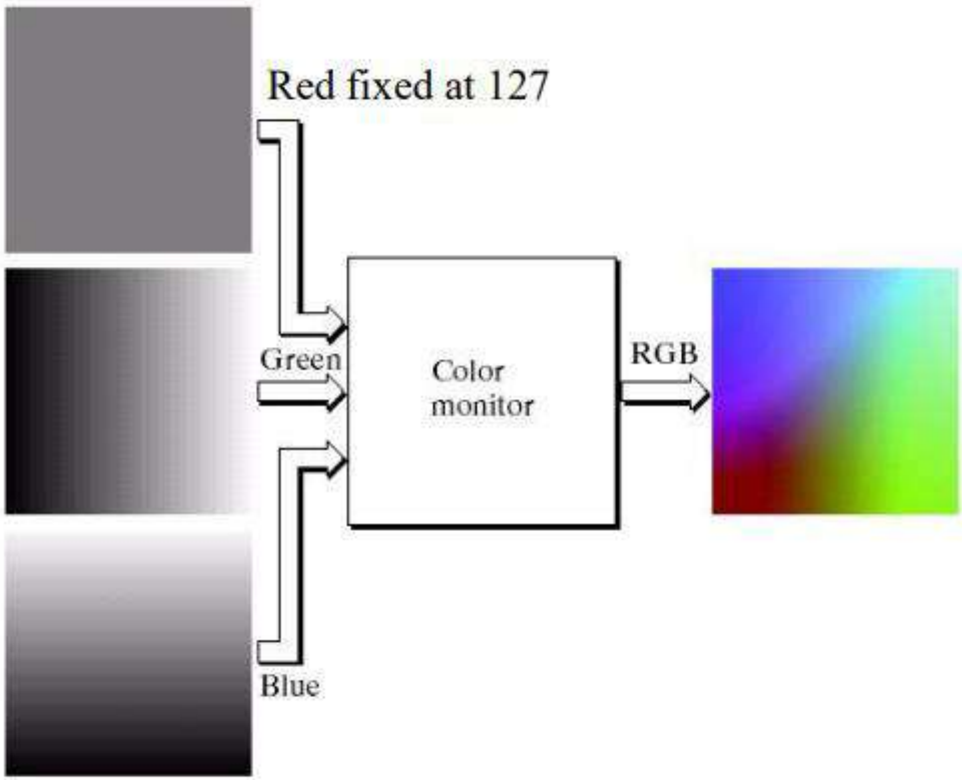
Color depth 24 bits
= 16777216 colors

(R = 0) (G = 0) (B = 0)

Hidden faces of the cube

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

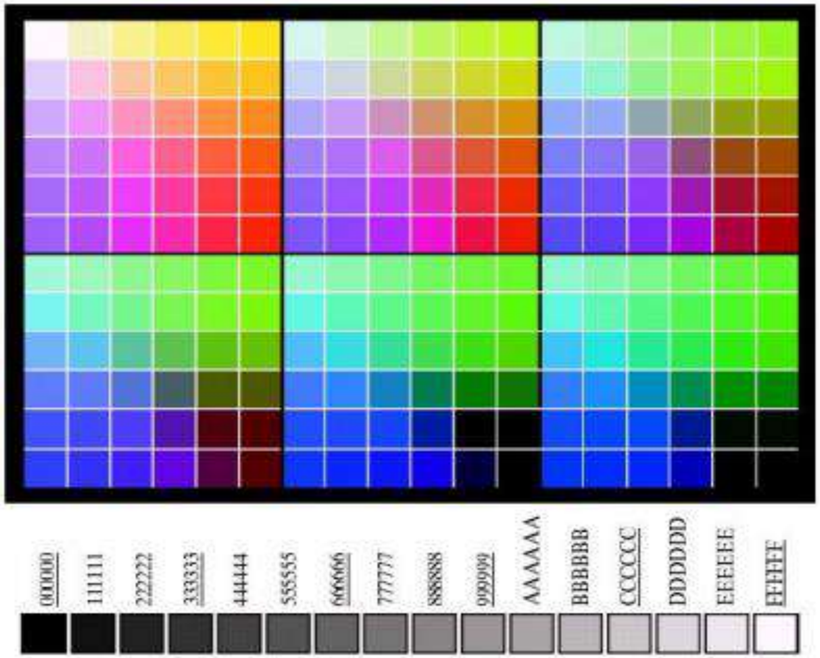
RGB Color Model (cont.)



Safe RGB Colors

Safe RGB colors: a subset of RGB colors.

There are 216 colors common in most operating systems.



a
b

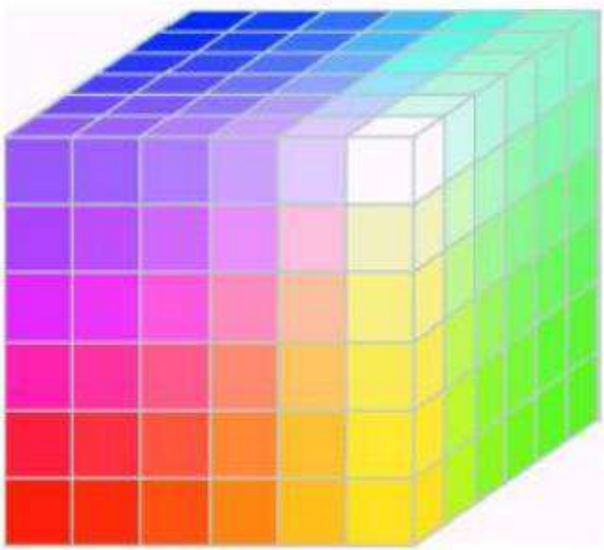
FIGURE 6.10 (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

RGB Safe-color Cube

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.



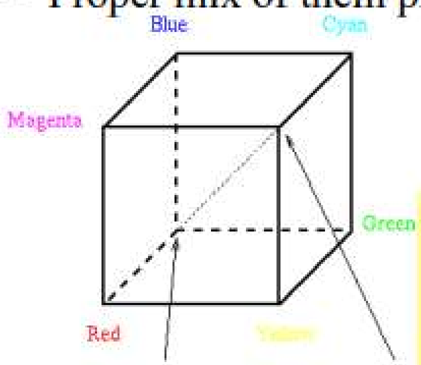
The RGB Cube is divided into 6 intervals on each axis to achieve the total $6^3 = 216$ common colors.

However, for 8 bit color representation, there are the total 256 colors. Therefore, the remaining 40 colors are left to OS.

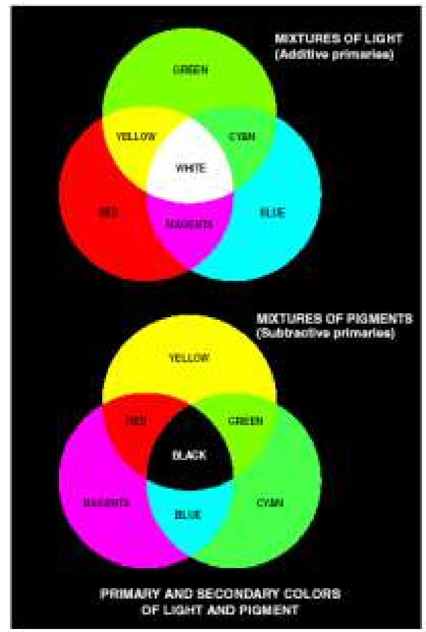
CMY and CMYK Color Models

C = Cyan
M = Magenta
Y = Yellow
K = Black

- Primary colors for pigment
  - Defined as one that subtracts/absorbs a primary color of light & reflects the other two
- CMY – Cyan, Magenta, Yellow
  - Complementary to RGB
  - Proper mix of them produces black



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



HSI Color Model

RGB, CMY models are not good for human interpreting

HSI Color model:

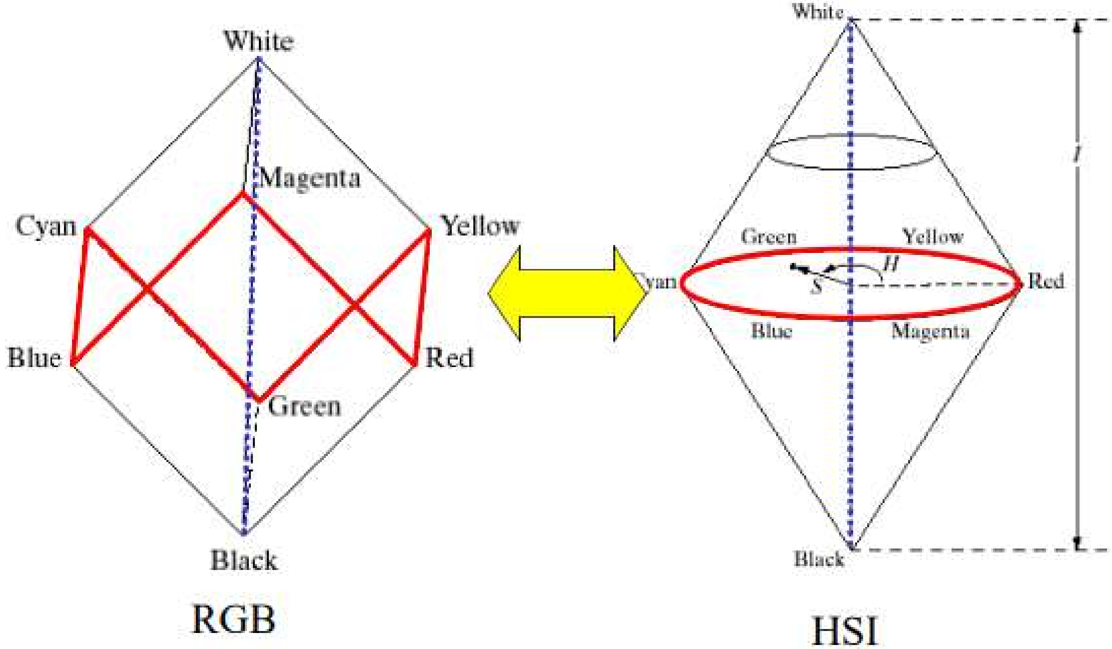
Hue: Dominant color

Saturation: Relative purity (inversely proportional to amount of white light added)

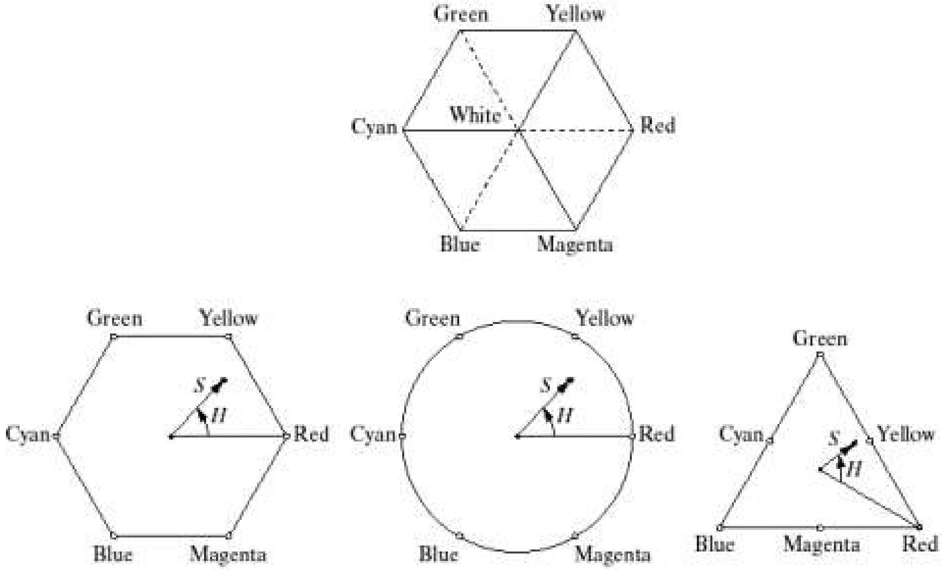
Color carrying information

Intensity: Brightness

Relationship Between RGB and HSI Color Models

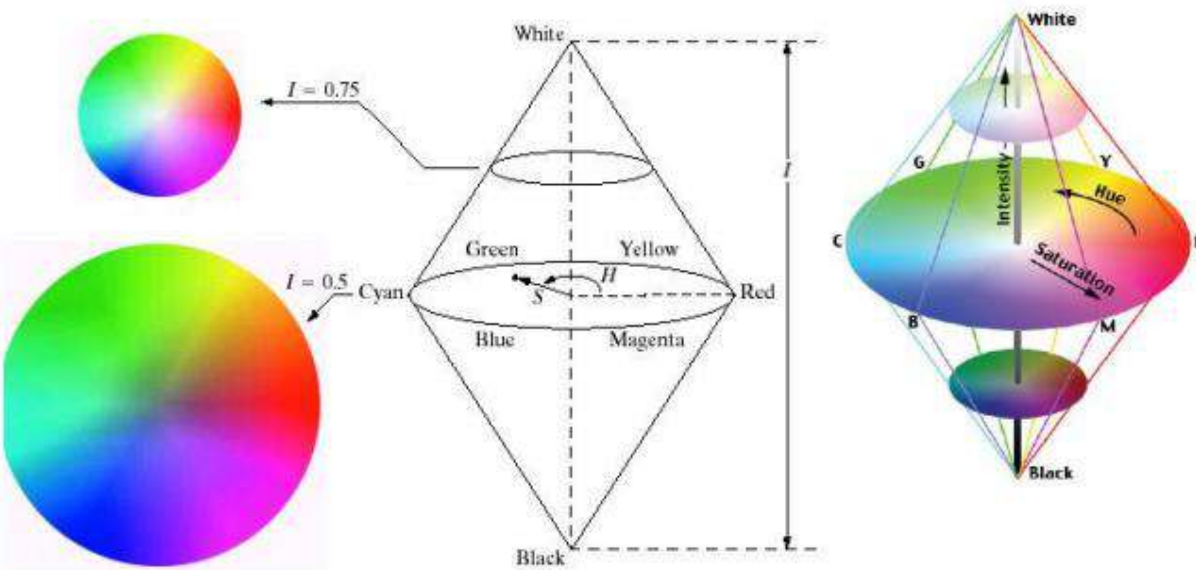


Hue and Saturation on Color Planes



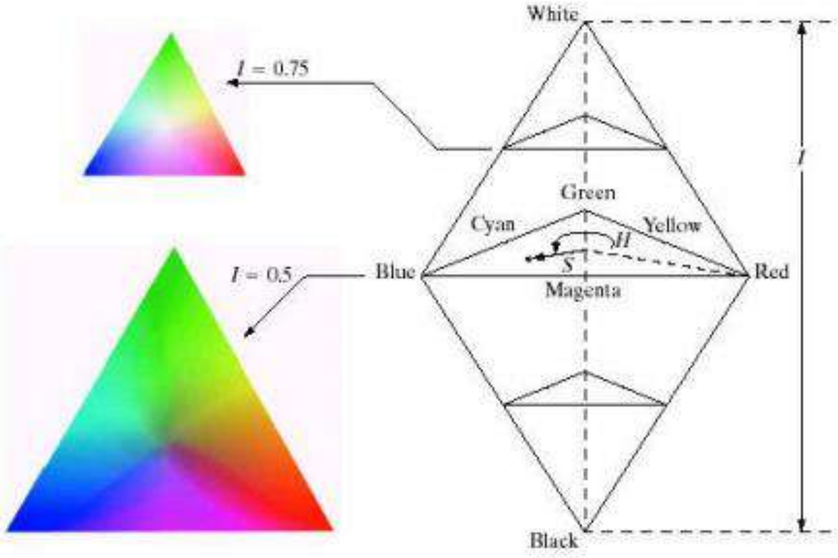
1. A dot is the plane is an arbitrary color
2. Hue is an angle from a red axis.
3. Saturation is a distance to the point.

HSI Color Model (cont.)



Intensity is given by a position on the vertical axis.

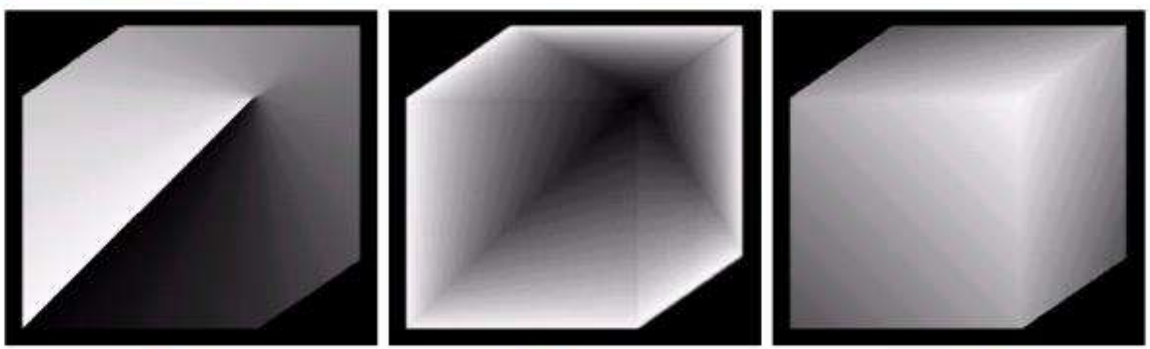
HSI Color Model



Intensity is given by a position on the vertical axis.

Example: HSI Components of RGB Cube

RGB Cube



Hue Saturation Intensity

Converting Colors from RGB to HSI

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{R+G+B}$$

$$I = \frac{1}{3}(R+G+B)$$

Converting Colors from HSI to RGB

RG sector: $0 \leq H < 120$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$$

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

$$G = 1 - (R + B)$$

GB sector: $120 \leq H < 240$

$$H = H - 120$$

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

$$G = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$$

$$B = 1 - (R + G)$$

BR sector: $240 \leq H \leq 360$

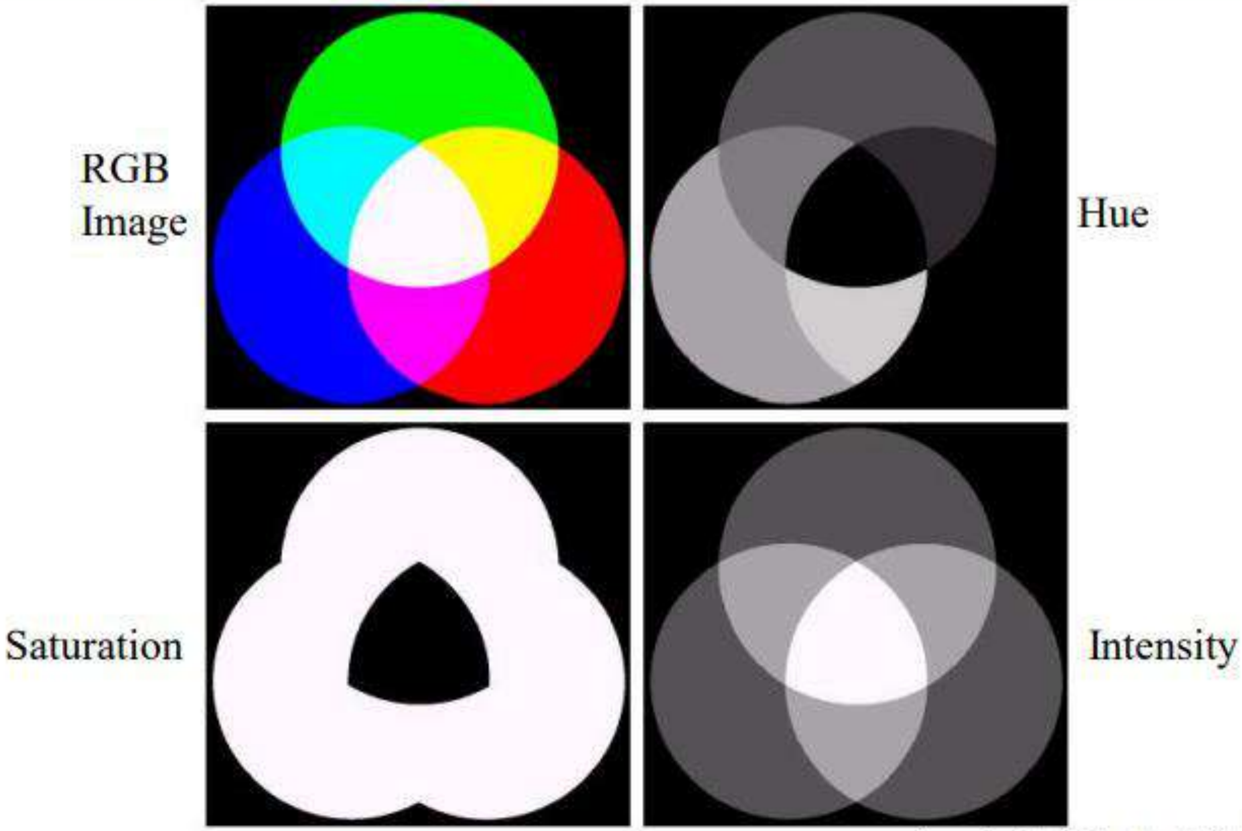
$$H = H - 240$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$$

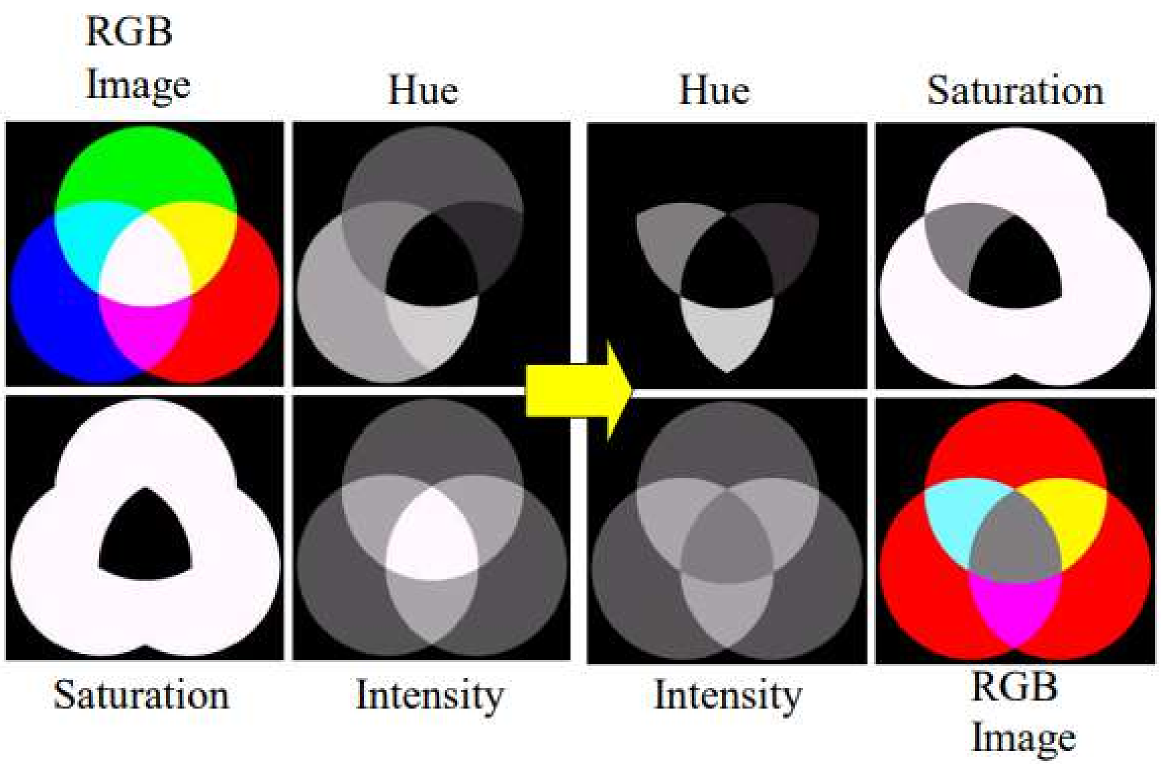
$$G = I(1 - S)$$

$$R = 1 - (G + B)$$

Example: HSI Components of RGB Colors



Example: Manipulating HSI Components



Color Coordinates Used in TV Transmission

- Facilitate sending color video via 6MHz mono TV channel
- YIQ for NTSC (National Television Systems Committee) transmission system
  - Use receiver primary system (R_N, G_N, B_N) as TV receivers standard
  - Transmission system use (Y, I, Q) color coordinate
    - Y ~ luminance, I & Q ~ chrominance
    - I & Q are transmitted in through orthogonal carriers at the same freq.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R_N \\ G_N \\ B_N \end{bmatrix}, \qquad \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R_P \\ G_P \\ B_P \end{bmatrix}.$$

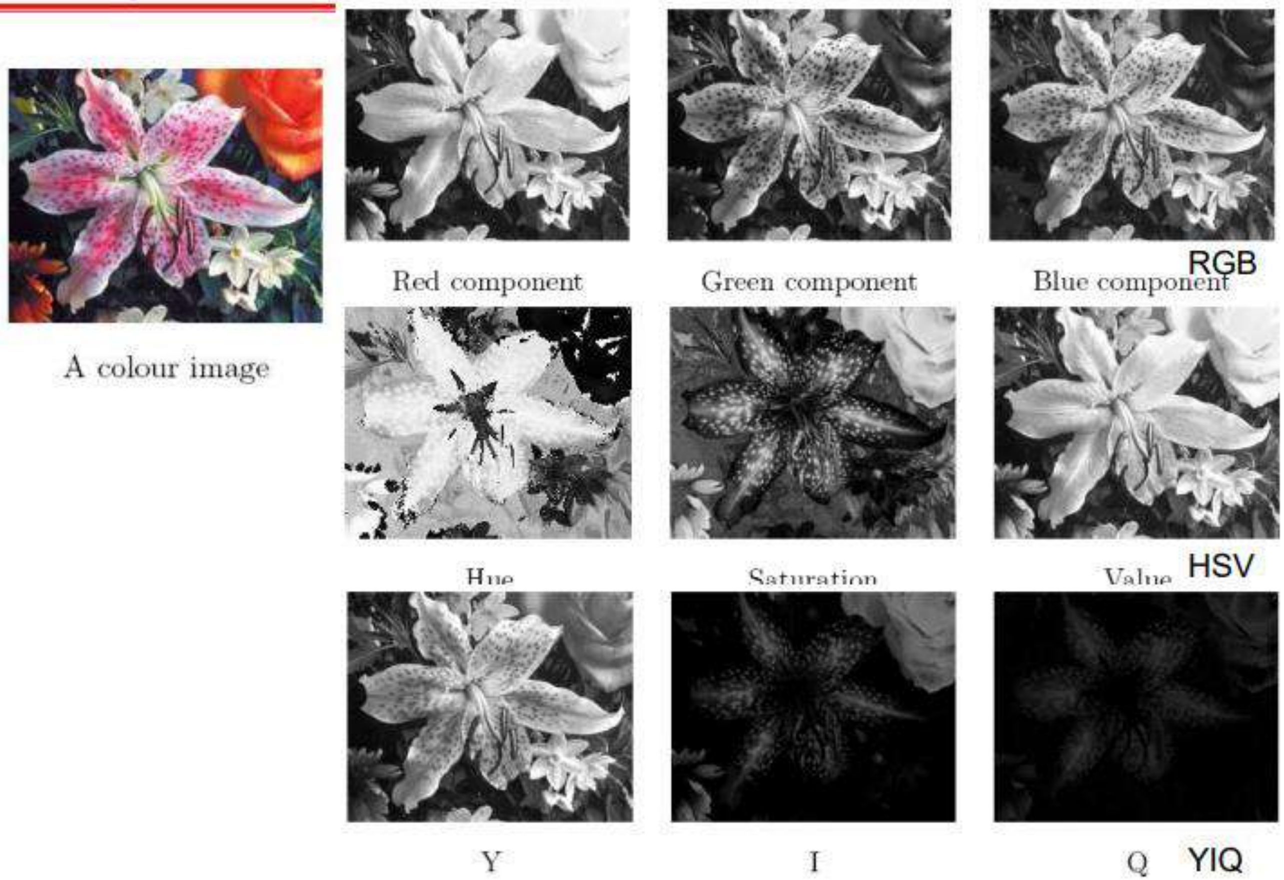
- YUV (YCbCr) for PAL and digital video
  - Y ~ luminance, Cb and Cr ~ chrominance

Color Coordinates

- RGB of CIE
- XYZ of CIE
- RGB of NTSC
- YIQ of NTSC
- YUV (YCbCr)
- CMY

Examples

Examples


A colour image
Red component
Green component
Blue component
RGB
Hue
Saturation
Value
HSV
Y
I
Q
YIQ

Summary

- Monochrome human vision
  - visual properties: luminance vs. brightness, etc.
  - image fidelity criteria
- Color
  - Color representations and three primary colors
  - Color coordinates

Pseudo Color Image Processing

COLOR IMAGE PROCESSING

There are 2 types of color image processes

1. Pseudocolor image process:
    - Assigning colors to gray values based on a specific criterion.
    - Gray scale images to be processed may be a single image or multiple images such as multispectral images

2. Full color image process:
    The process to manipulate real color images such as color photographs.

Pseudo color = false color : In some case there is no "color" concept for a gray scale image but we can assign "false" colors to an image.

Why we need to assign colors to gray scale image?

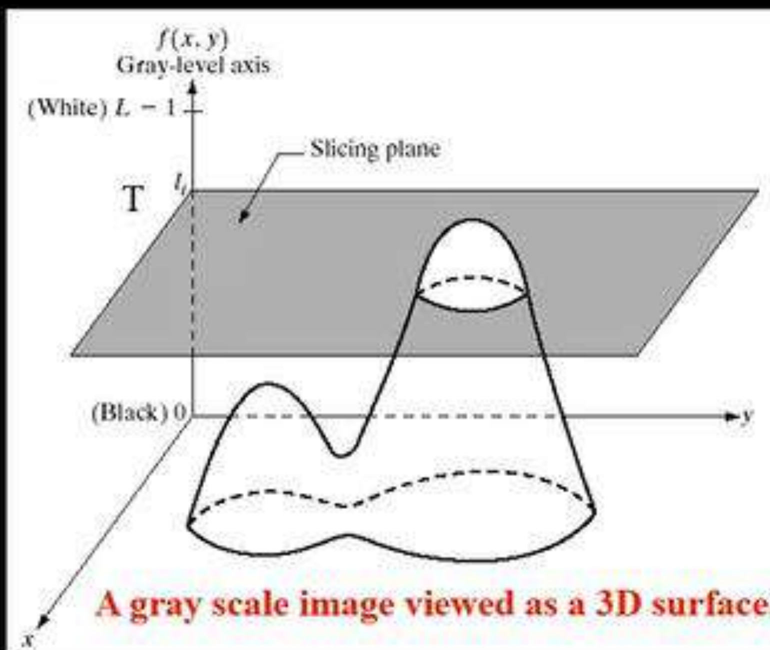
Answer: Human can distinguish different colors better than different shades of gray.

https://damagedphotorestoration.com/blog/how-to-guide/how-to-colorize-a-black-and-white-photo-in-Photoshop.htmlWood, Digital Image Processing, 2nd Edition.

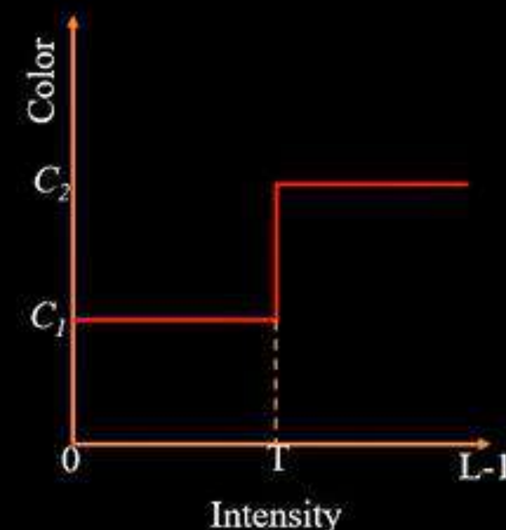
INTENSITY SLICING OR DENSITY SLICING

Formula: $g(x,y) = \begin{cases} C_1 & \text{if } f(x,y) \leq T \\ C_2 & \text{if } f(x,y) > T \end{cases}$

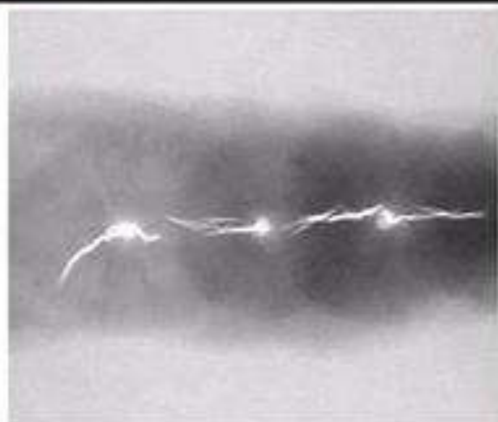
C_1 = Color No. 1
C_2 = Color No. 2



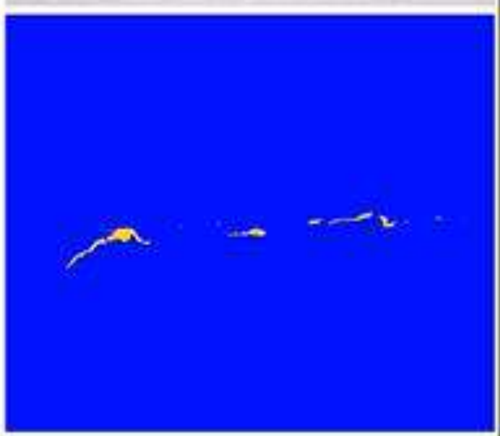
A gray scale image viewed as a 3D surface.



INTENSITY SLICING EXAMPLE



An X-ray image of a weld with cracks

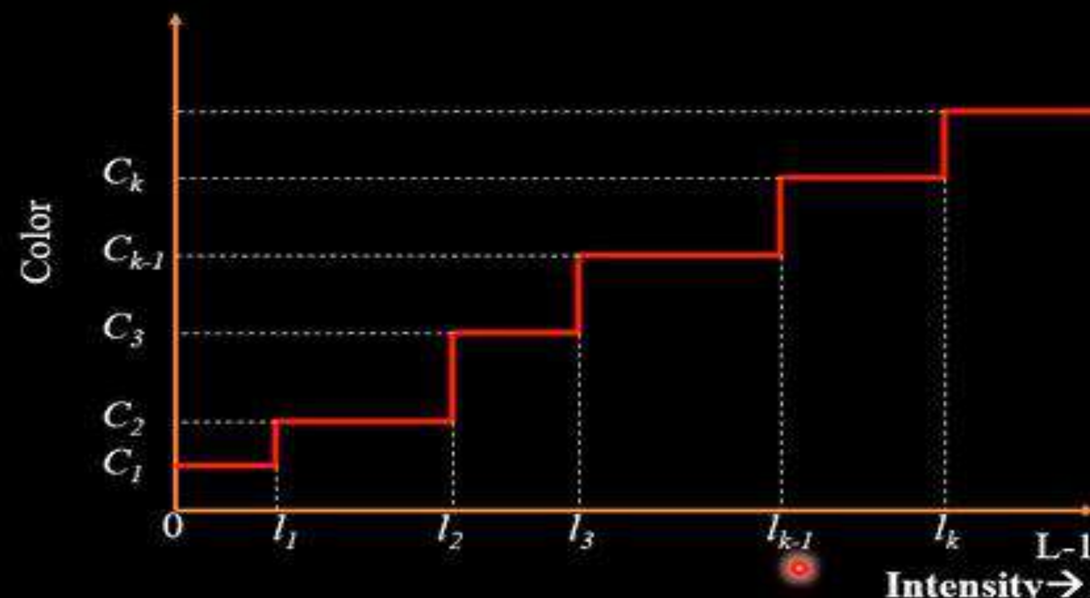


After assigning a yellow color to pixels with value 255 and a blue color to all other pixels.

MULTI LEVEL INTENSITY SLICING

$$g(x,y) = C_k \qquad \text{for } l_{k-1} < f(x,y) \leq l_k$$

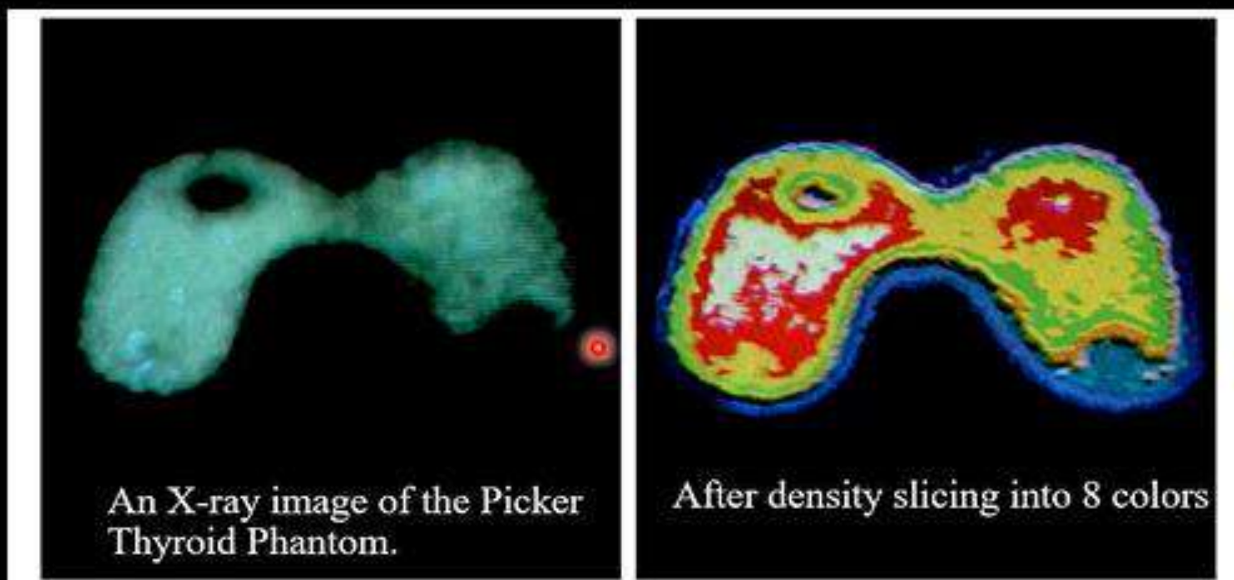
C_k = Color No. k
l_k = Threshold level k



Multi Level Intensity Slicing Example

$$g(x,y) = C_k \qquad \text{for } l_{k-1} < f(x,y) \le l_k$$

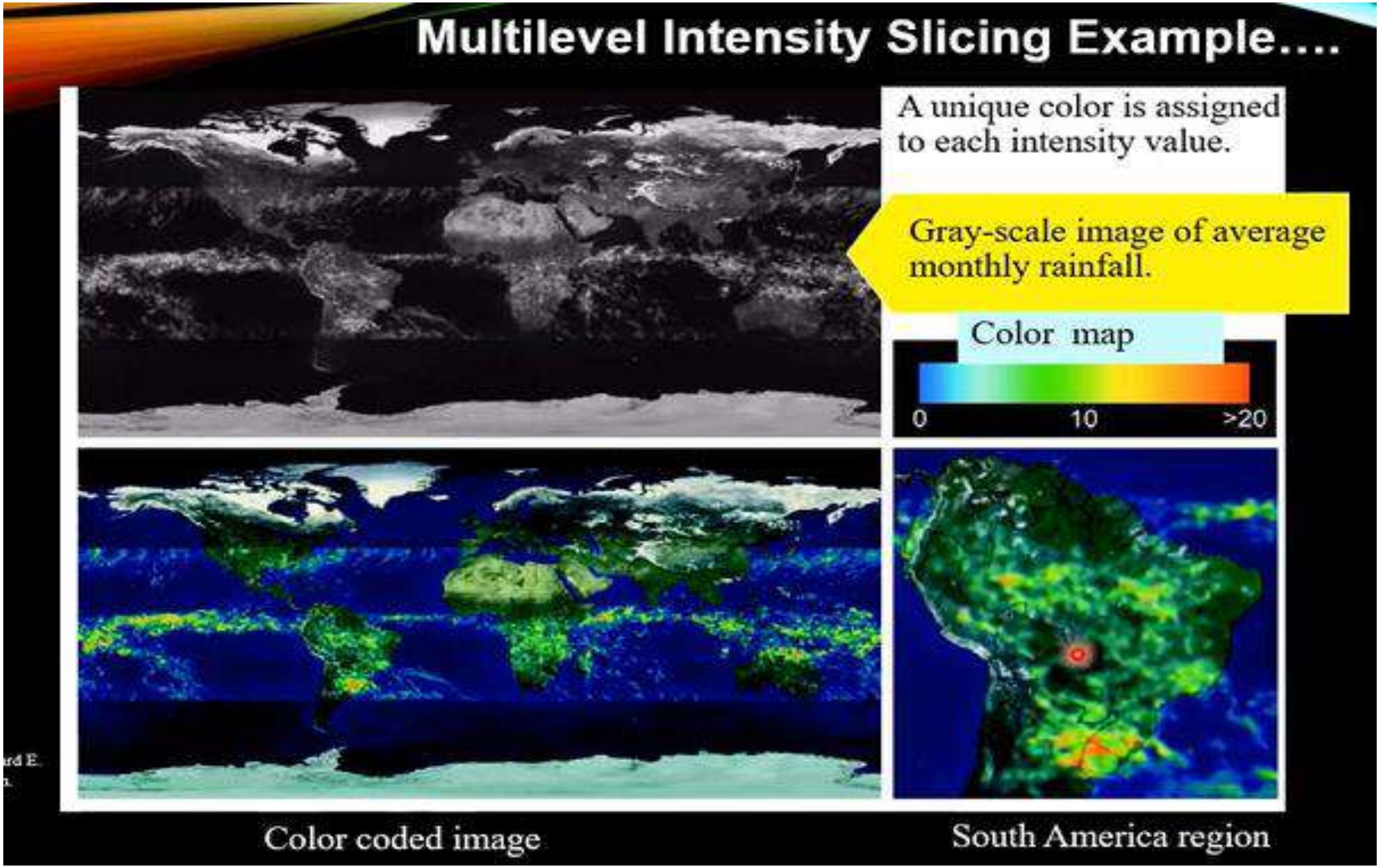
C_k = Color No. k
l_k = Threshold level k



An X-ray image of the Picker Thyroid Phantom.

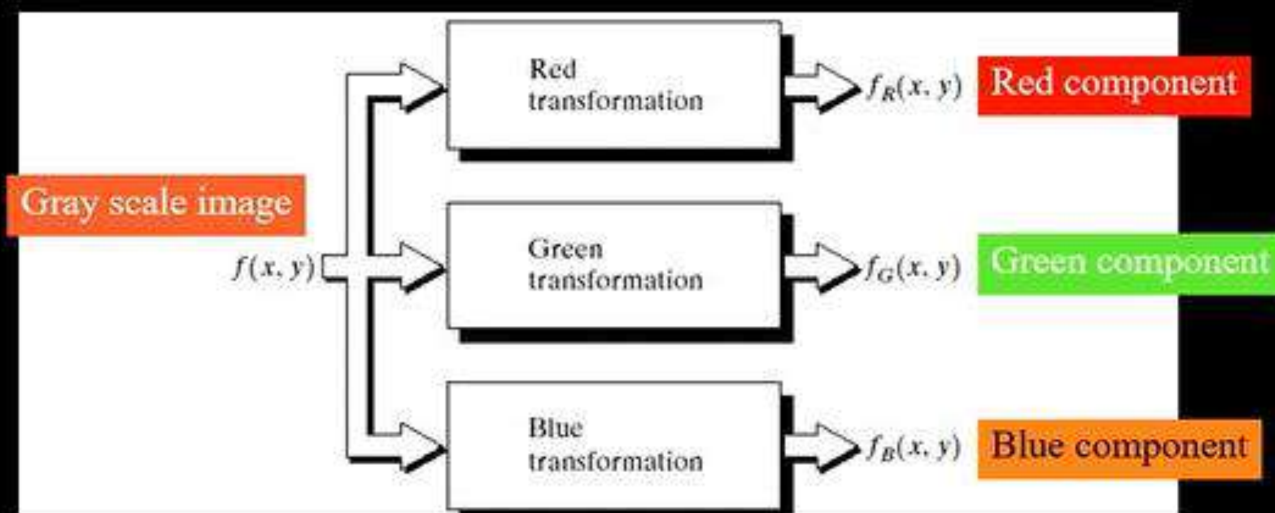
After density slicing into 8 colors

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

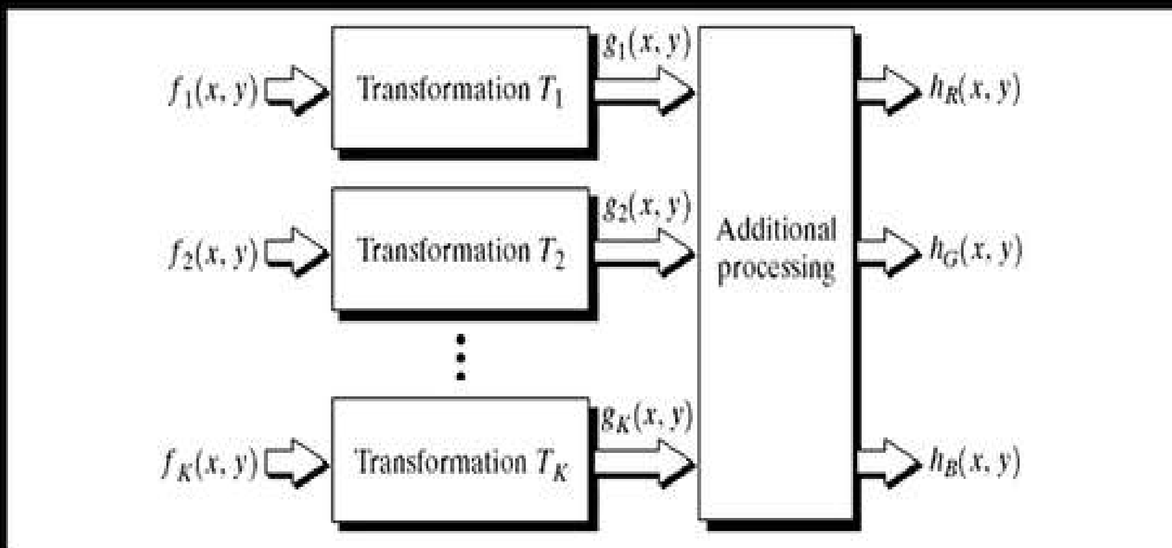

Multilevel Intensity Slicing Example….
A unique color is assigned to each intensity value.
Gray-scale image of average monthly rainfall.
Color map
0
10
>20
Color coded image
South America region

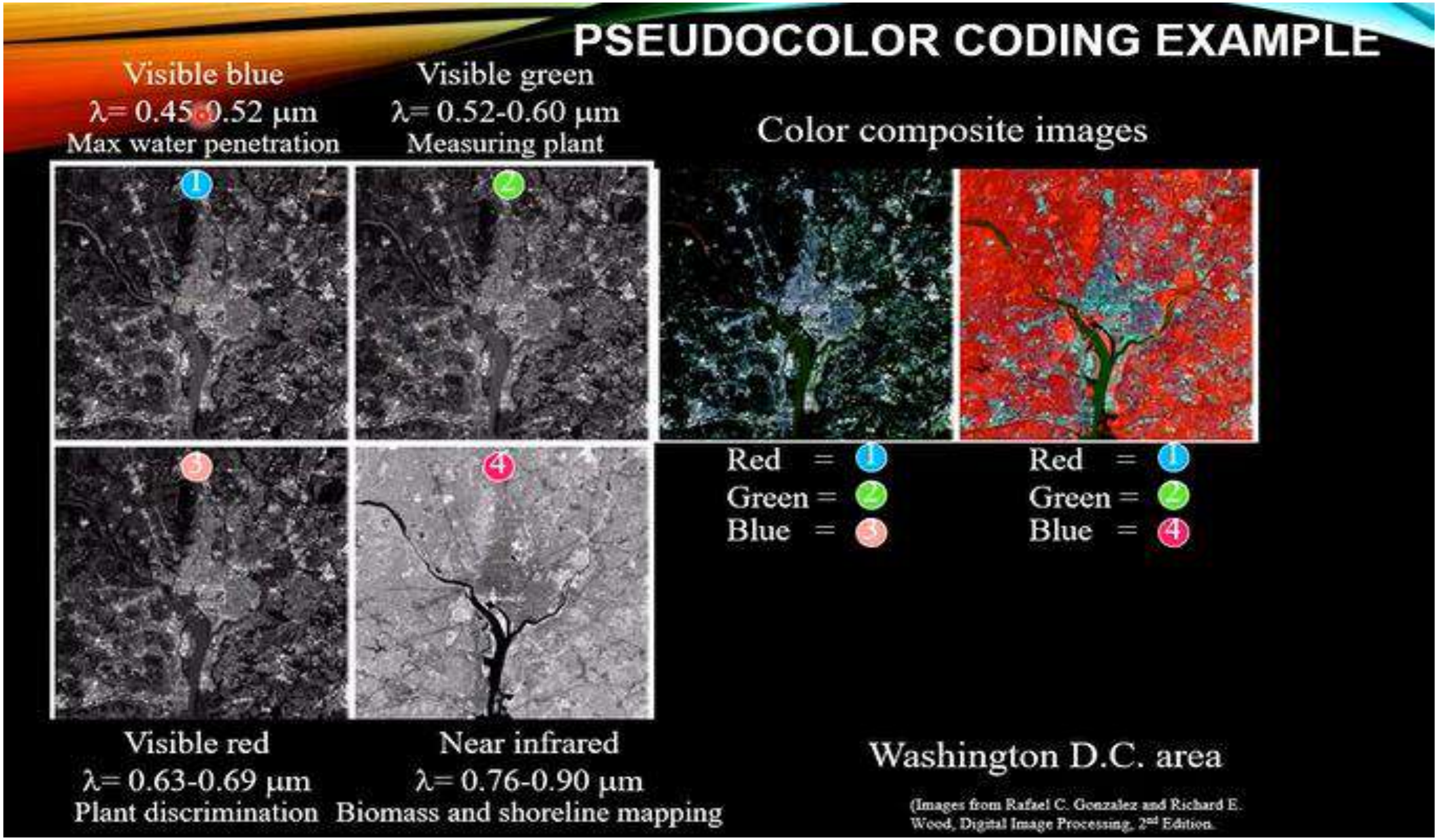
GRAY LEVEL TO COLOR TRANSFORMATION

Assigning colors to gray levels based on specific mapping functions



Used in the case where there are many monochrome images such as multispectral satellite images.




PSEUDOCOLOR CODING EXAMPLE
Visible blue
λ= 0.45-0.52 µm
Max water penetration
Visible green
λ= 0.52-0.60 µm
Measuring plant
Color composite images
1
2
3
4
Red = 1
Green = 2
Blue = 3
Red = 1
Green = 2
Blue = 4
Visible red
λ= 0.63-0.69 µm
Plant discrimination
Near infrared
λ= 0.76-0.90 µm
Biomass and shoreline mapping
Washington D.C. area
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Pseudocolor Coding Example

Psuedocolor rendition of Jupiter moon Io

Yellow areas = older sulfur deposits.
Red areas = material ejected from active volcanoes.

A close-up

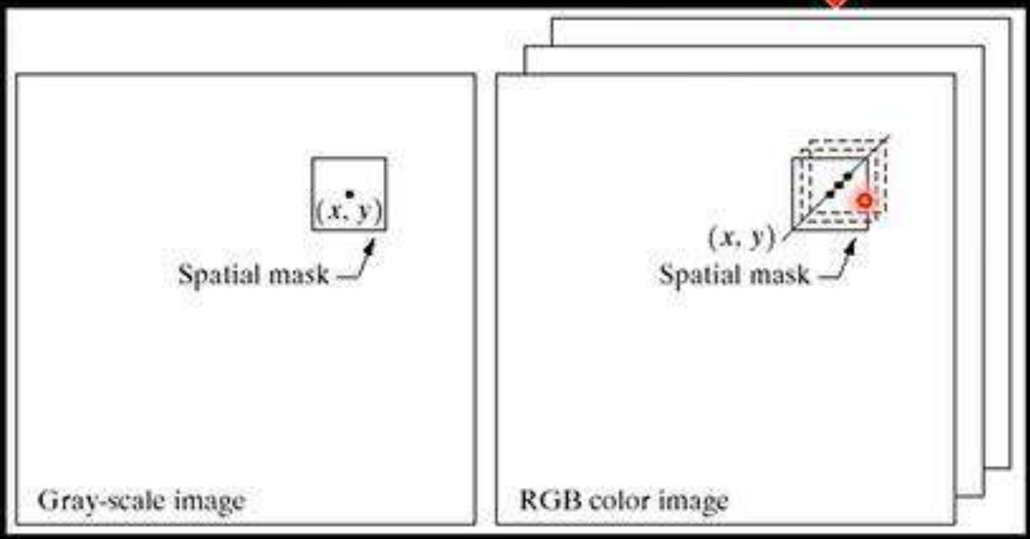
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

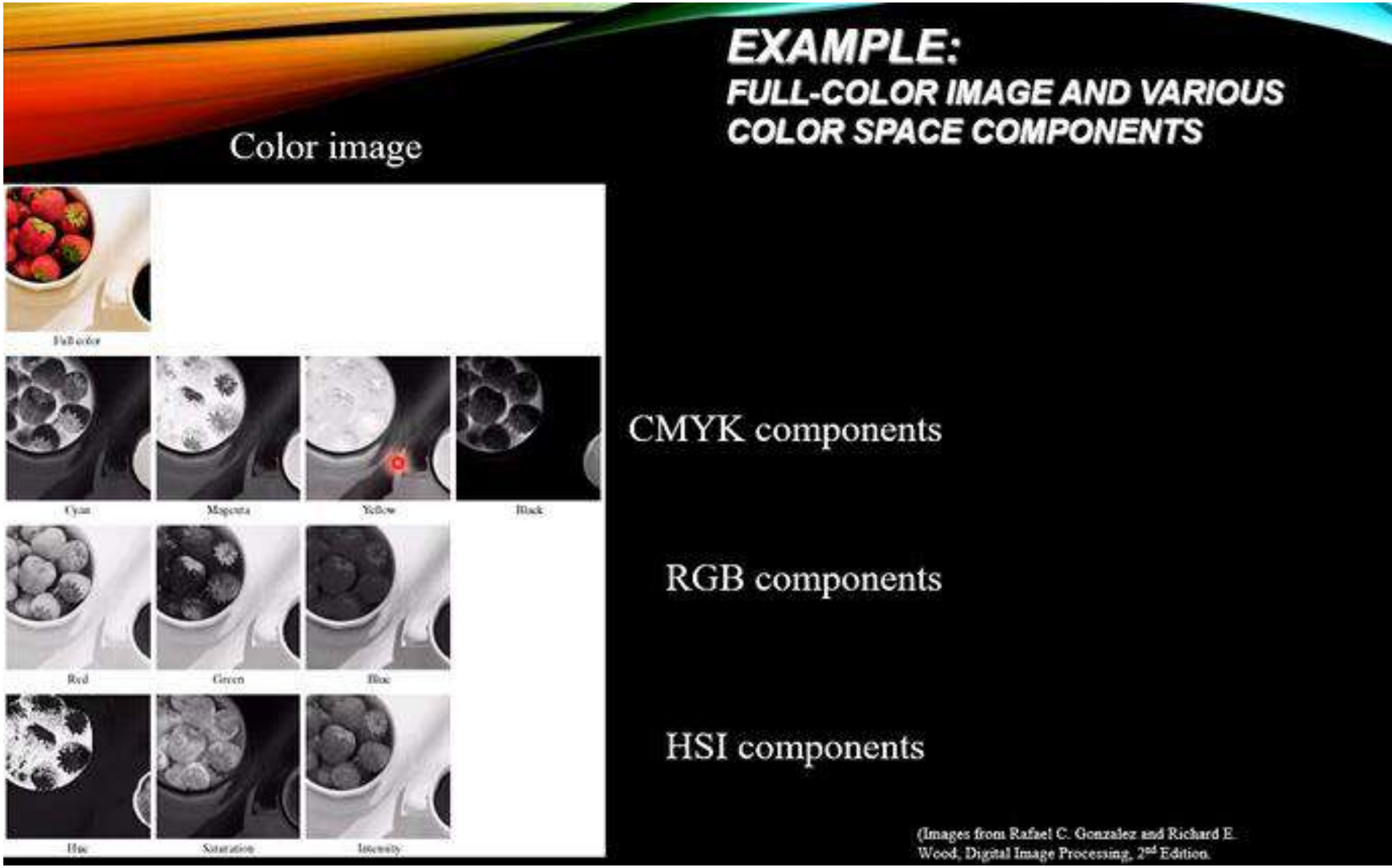
Basics of Full-Color Image Processing

Two Methods:

1. Per-color-component processing: process each component separately.
2. Vector processing: treat each pixel as a vector to be processed.

Example of per-color-component processing: smoothing an image
By smoothing each RGB component separately.




EXAMPLE:
FULL-COLOR IMAGE AND VARIOUS
COLOR SPACE COMPONENTS
Color image
CMYK components
RGB components
HSI components
(Images from Rafael C. Gonzalez and Richard E.
Wood, Digital Image Processing, 2nd Edition.