ANNAMALAI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

STUDY MATERIAL FOR B.E IIIYEAR ‘B’ BATCH

08PE605- DIGITAL IMAGE PROCESSING

Staff Name: Dr.R.Bhavani

Unit-I

Digital Imaging: Introduction – Steps in Image Processing Systems – Image Acquisition – Image Sampling and Quantization – Pixel Relationships – Linear and Nonlinear Operations. MATLAB: The MATLAB Desktop – Using the MATLAB Editor/Debugger – Getting Help – Saving and Retrieving work Session Data – Digital Image Representation – Image I/O and Display – Classes and Image Types – M-Function Programming.

Introducion

DIGITAL IMAGE

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer.

Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used to denote the elements of a digital image.

Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. The area of image analysis (also called image understanding) is in between image processing and computer vision.

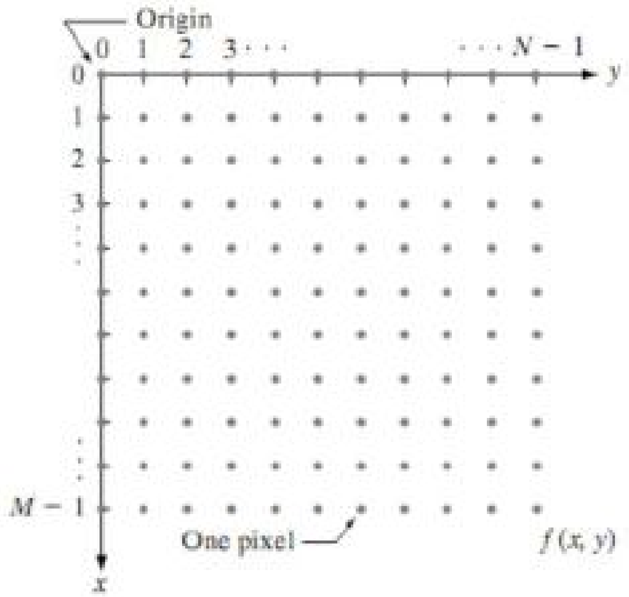
The processes of acquiring an image of the area containing the text, preprocessing that image, extracting (segmenting) the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters are in the scope of what we call digital image processing.

Representing Digital Images:

We will use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting digital image has M rows and N columns. The values of the coordinates (x, y) now become discrete quantities.

For notational clarity and convenience, we shall use integer values for these discrete coordinates. Thus, the values of the coordinates at the origin are (x, y) = (0, 0). The next coordinate values along the first row of the image are represented as (x, y) = (0, 1).

It is important to keep in mind that the notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure 1 shows the coordinate convention used.



Coordinate convention used to represent digital images

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

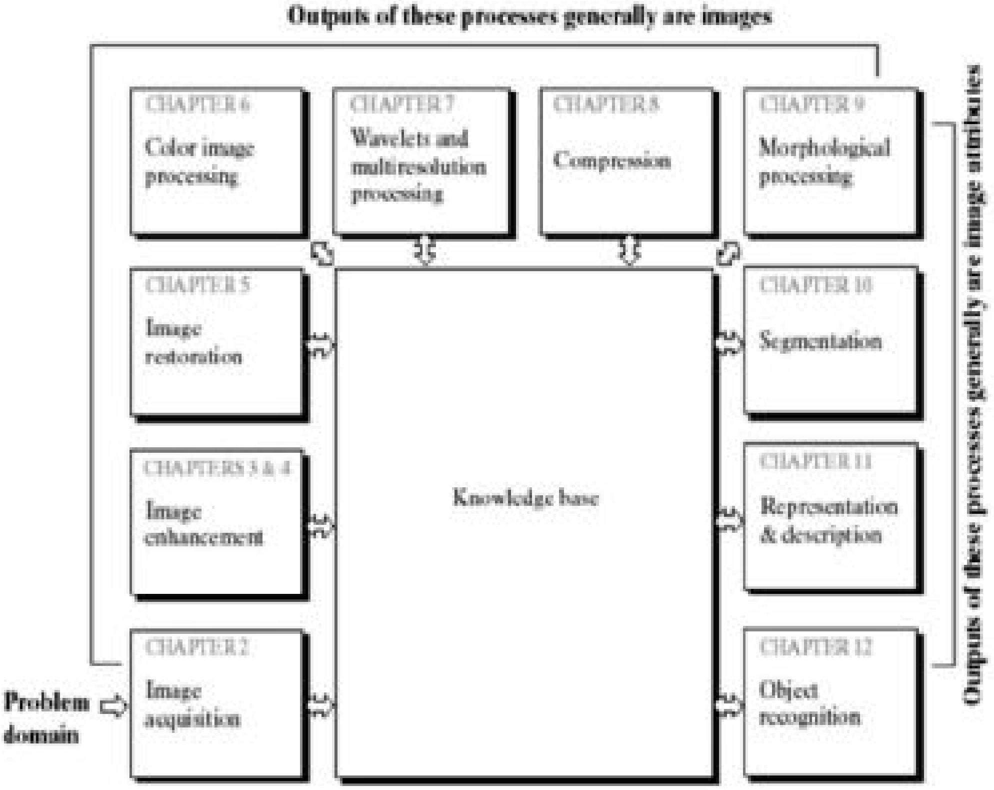
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}.$$

The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, pixel, or pel.

Fundamental Steps in Digital Image Processing

1. Image acquisition
2. Image enhancement
3. Image restoration
4. Color image processing
5. Wavelets
6. Compression
7. Morphological processing
8. Segmentation
9. Representation & description
10. Object recognition

11. Knowledge base



1.Image acquisition

1) Origin of digital image

2) Simple, images already in digital form

3) Involves preprocessing such as scaling

2. Image enhancement

1) Used to bring out the details that is obscured

2) Highlight certain features of interest in image

Eg: If we increase the contrast of the image it looks better

3.Image restoration

1) Improves the appearance of image

2) Restoration techniques –>mathematical or probabilistic models of image degradation

4. Color image processing

Gaining importance -> significant increase in use of digital images over internet

5.Wavelets

1) Foundation for representing images in various degrees of resolution
2) Used for image data compression and pyramidal representation
3) In which images are subdivided into smaller regions

6. Compression

Techniques for reducing the storage required to save an image or the bandwidth requires to transmit it.

7.Morphological processing

1) Tools for extracting image components that are useful in the representation and description of shape
2) Begins a transition from processes that output images to processes that output image attributes

8. Segmentation

1) Its procedures partition an image into its constituent parts or objects
2) Autonomous segmentation is the difficult task in digital image processing

9. Representation & description

1) Follows the output of segmentation stage
2) It has raw pixel data constituting either the boundary of a region or all the points in the region itself
3) Description also called as feature selection deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another

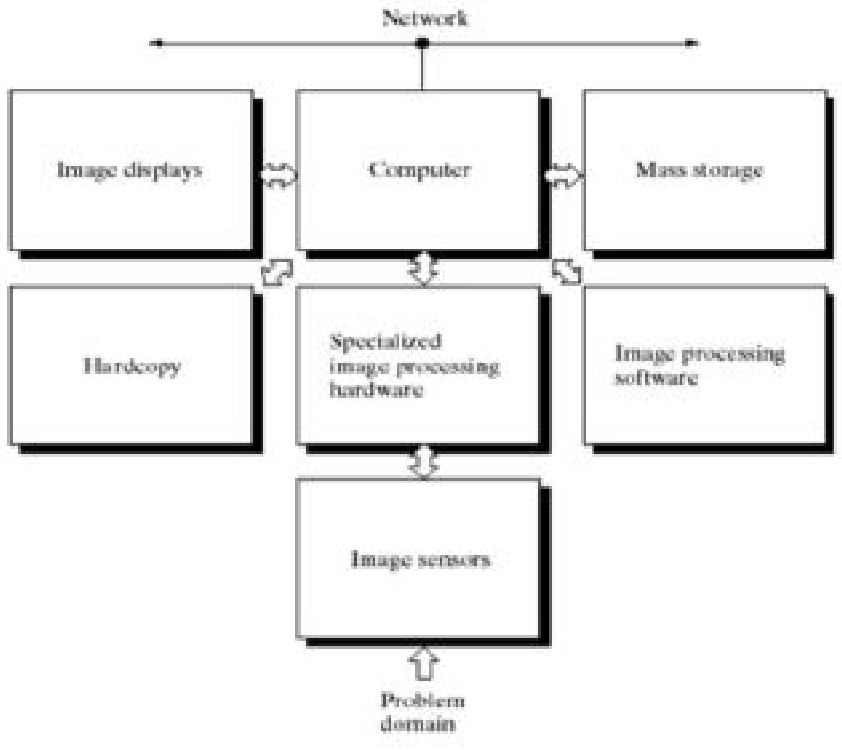
10. Recognition

It is the process that assigns a label to an object based on its descriptors

11.Knowledge base

1) Quite complex-> interrelated list of all major possible defects in a materials inspection problem
2) Controls the interaction between the processing modules

Components of an Image Processing System



1. Sensing

Two elements required to acquire digital images

1) physical device that is sensitive to energy radiated by the object we wish to view image

2) digitizer is a device for converting the output of the physical sensing device into digital form.

2.Specialized image processing hardware

1) Consists of digitizer and hardware

2) Performs primitive operations such as Arithmetic and Logic Unit(ALU) that performs the arithmetic and logical operations in parallel on entire images

3. Computer

1) In image processing system it is a general purpose computer that ranges from a PC to a supercomputer

2) Specially designed computers are used to achieve a required level of performance

4.Software

1) It consists of specialized modules that perform specific tasks

2) Well designed package includes the capability for the user to write code that as a minimum utilizes the specialized modules

5.Mass storage

1) It is must in an image processing applications.

2) An image of size 1024 * 1024 pixels in which the intensity of each pixel is an 8-bit quantity requires one megabyte of storage space if the image is not compressed

6. Image displays

1) It uses color TV monitors

2) Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system

6. Image displays

1) It uses color TV monitors

2) Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system

7. Hardcopy

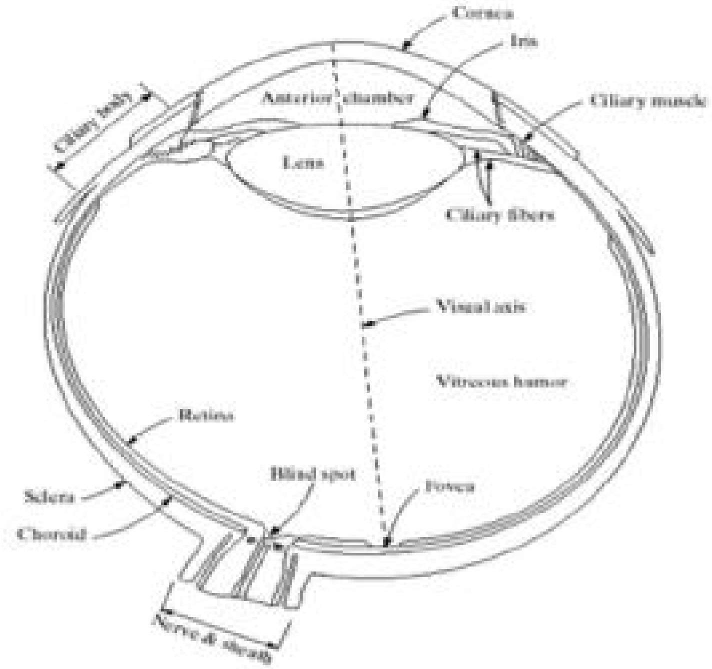
Hardcopy devices for recording images include laser printers, film cameras, heat sensitive devices, inkjet units and digital units such as optical and CD-ROM disks

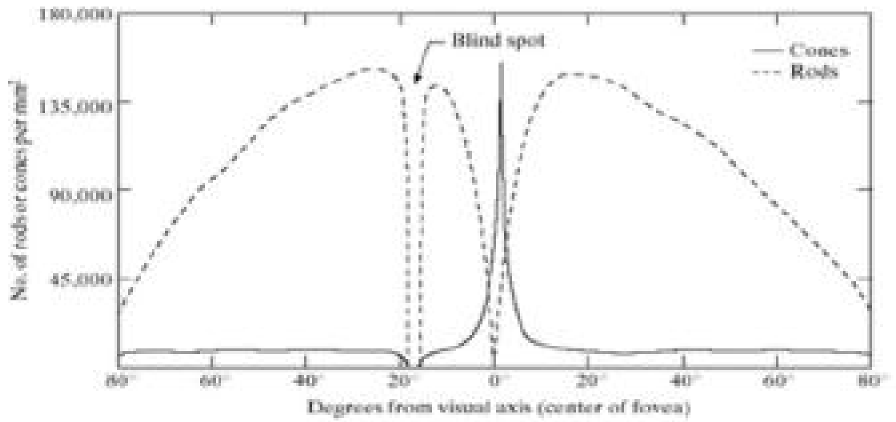
8. Networking

1) It is a default function in any computer system in use today

2) Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth

Elements of Visual Perception





Distribution of rods and cones in the retina.

Structure of the Human Eye

- Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina.

- There are two classes of receptors : cones and rods.

    - The number of cones in each eye: 6 to 7 millions

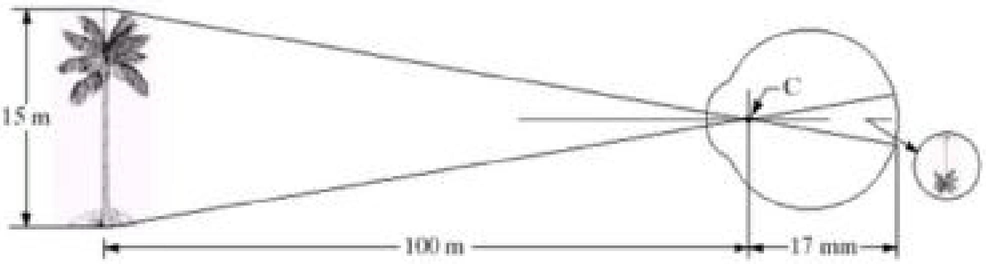
    - The number of rods in each eye: 75 to 150 millions

    - The cones is concentrated in the central portion of the retina (fovea).

    - The rods are distributed over the retinal surface.
- Photopic (bright-light) vision: vision with cones
    - color receptors, high resolution in the fovea, less sensitive to light
- Scotopic (dim-light) vision: vision with rods
    - color blind, much more sensitive to light (night vision), lower resolution

Image Formation in the Eye

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

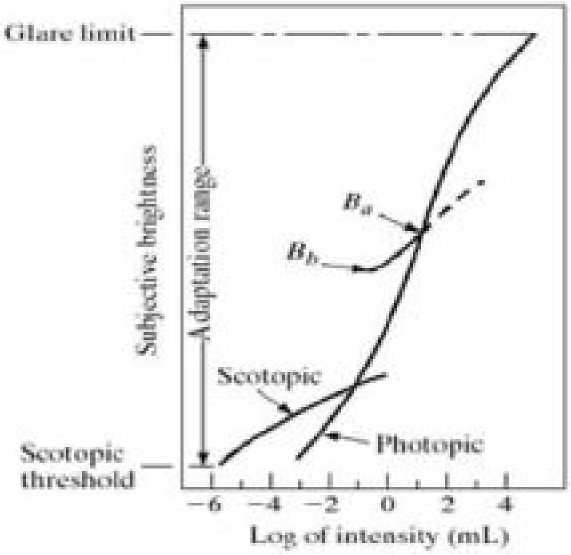


- Focal length of the eye: 17 to 14 mm
- Let *h* be the height in mm of that object in the retinal image, then

  15/100 = *h* / 17 , *h* = 2.55mm
- The retinal image is reflected primarily in the area of the fovea.

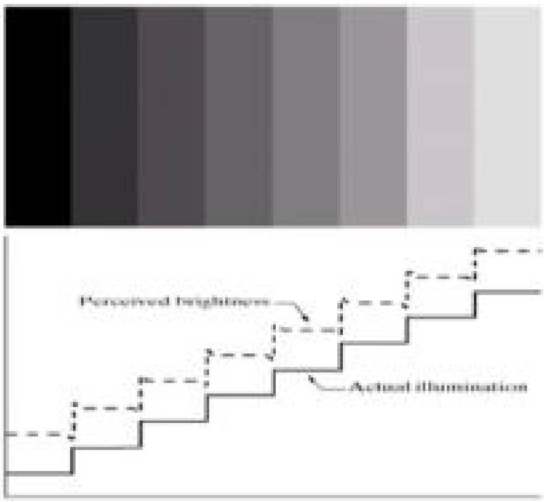
Brightness Adaptation and Discrimination

- The range of brightness that the eye can adapt to is enormous, roughly around 10^{10} to 1.
- Photopic vision alone has a range of around 10^{6} to 1.
- Brightness adaptation: example “ B_a”
- mL: millilambert

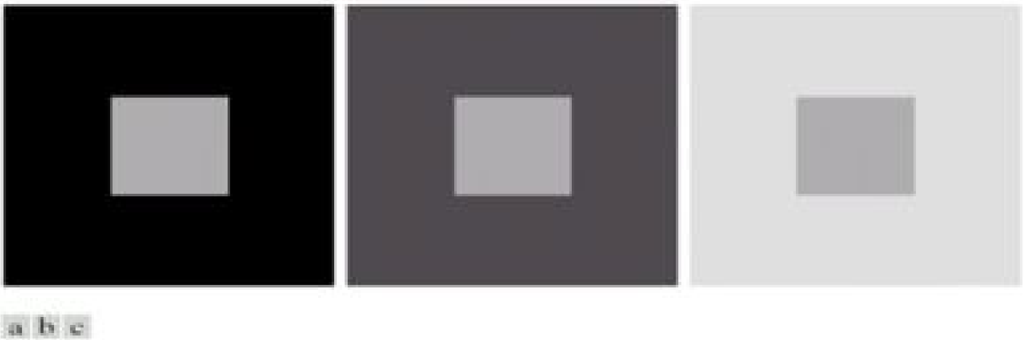


- The essential point in interpreting the impressive dynamic range depicted in Fig.4.3 is that the visual system cannot operate over such a range simultaneously. Rather, it accomplishes this large variation by changes in its overall sensitivity, a phenomenon known as brightness adaptation.
- The total range of distinct intensity levels it can discriminate simultaneously is rather small when compared with the total adaptation range.
- For any given set of conditions, the current sensitivity level of the visual system is called the brightness adaptation level, which may correspond, for example, to brightness Ba in Fig. 4.3.
- The short intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. This range is rather restricted, having a level Bb at and below which all stimuli are perceived as indistinguishable blacks.
- The upper (dashed) portion of the curve is not actually restricted but, if extended too far, loses its meaning because much higher intensities would simply raise the adaptation level higher than Ba.

Example: Mach bands

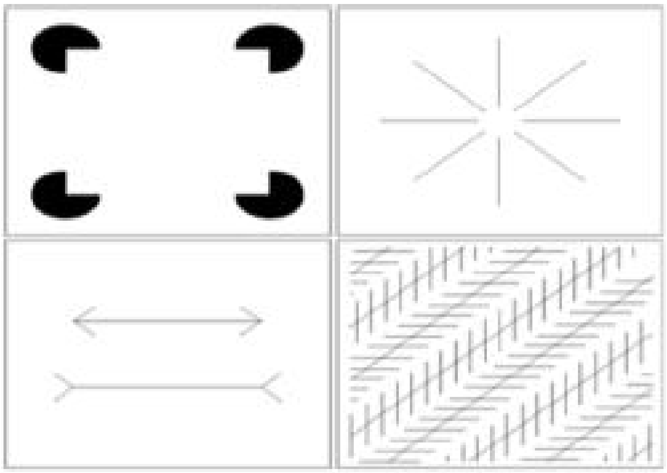


Example: Simultaneous Contrast

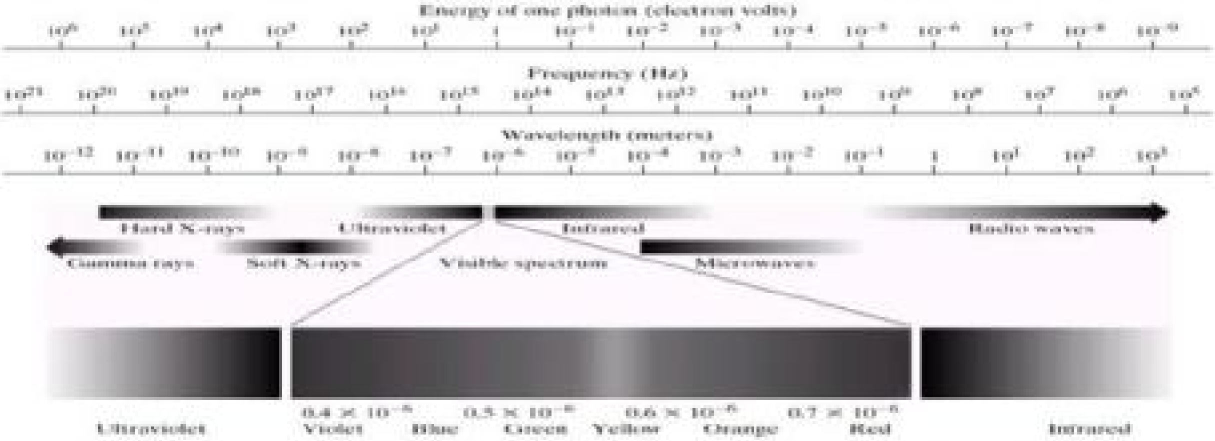


Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

Examples for Human Perception Phenomena



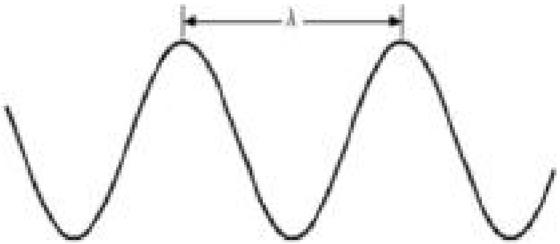
Light and the Electromagnetic Spectrum



The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

- Three basic quantities described the quality of a chromatic light source:
  - Radiance: the total amount energy that flow from the light source (can be measured)
  - Luminance: the amount of energy an observer perceives from a light source (can be measured)
  - Brightness: a subjective descriptor of light perception; perceived quantity of light emitted (cannot be measured)
- Relationship between frequency ( ) and wavelength ( )
  , where *c* is the speed of light
- Energy of a photon
  , where *h* is Planck's constant

Graphical representation of one wavelength.

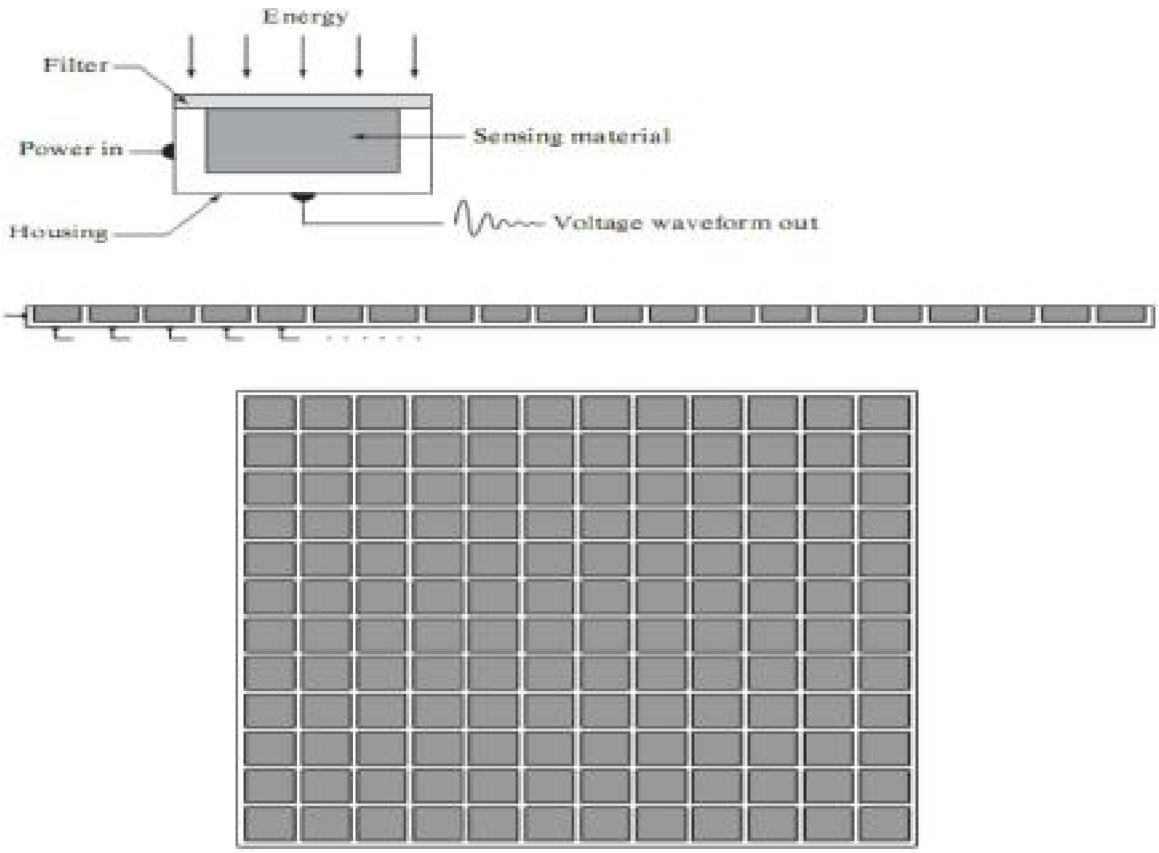


Process of image acquisition

Image Sensing and Acquisition:

- The types of images in which we are interested are generated by the combination of an "illumination" source and the reflection or absorption of energy from that source by the elements of the "scene" being imaged.
- For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern.
- We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects.
- An example in the first category is light reflected from a planar surface.
- An example in the second category is when X-rays pass through a patient's body for the purpose of generating a diagnostic X-ray film.
- In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach.

Figure 5.1 shows the three principal sensor arrangements used to transform illumination energy into digital images. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response.



(a) Single imaging Sensor (b) Line sensor (c) Array sensor

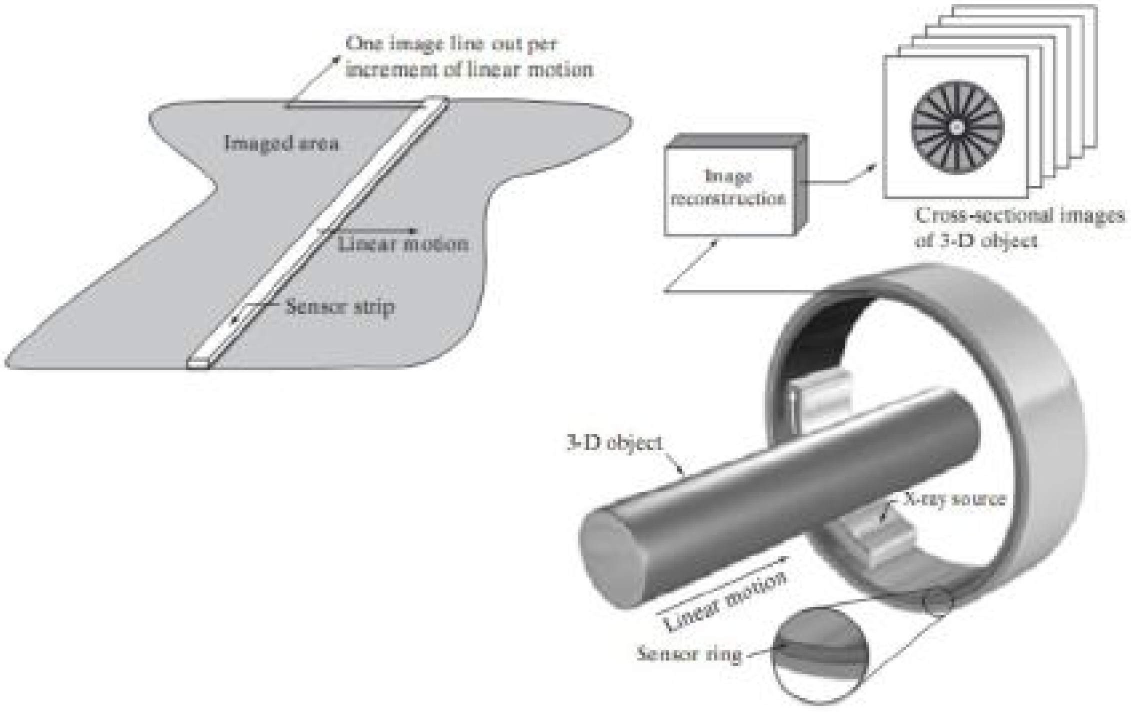
(1)Image Acquisition Using a Single Sensor:

- Perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform is proportional to light. The use of a filter in front of a sensor improves selectivity.
- For example, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.
- In order to generate a 2-D image using a single sensor, there has to be relative displacements in both the x- and y-directions between the sensor and the area to be imaged. Figure 5.2 shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension.
- The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Since mechanical motion can be controlled with high precision, this method is an inexpensive (but slow) way to obtain high-resolution images.
- Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as microdensitometers.

(2) Image Acquisition Using Sensor Strips:

- A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, as Fig. 5.1 (b) shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction, as shown in Fig. 5.3 (a).This is the type of arrangement used in most flat bed scanners.

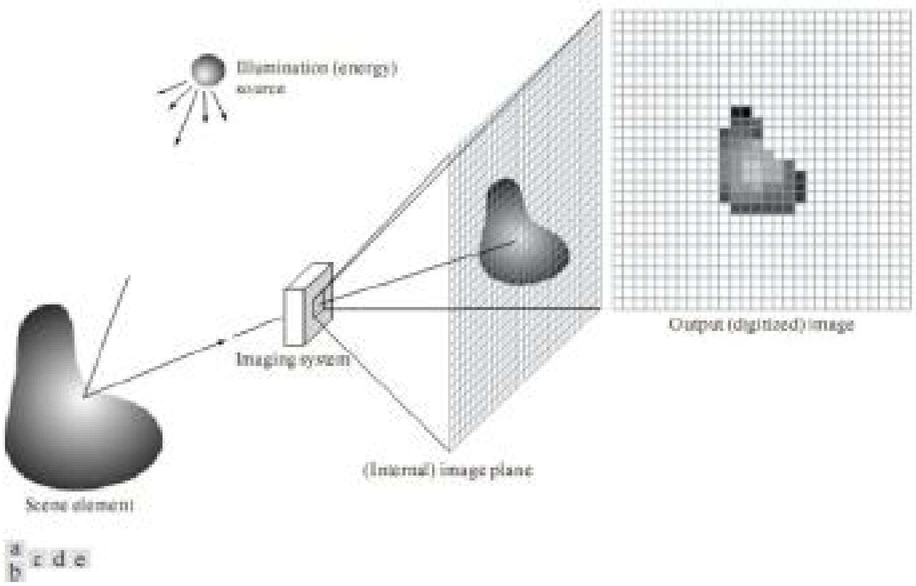
- Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged.
- One-dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image.
- Lenses or other focusing schemes are used to project the area to be scanned onto the sensors. Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional ("slice") images of 3-D objects, as Fig. 5.3 (b) shows.
- A rotating X-ray source provides illumination and the portion of the sensors opposite the source collect the X-ray energy that pass through the object (the sensors obviously have to be sensitive to X-ray energy).This is the basis for medical and industrial computerized axial tomography (CAT).
- It is important to note that the output of the sensors must be processed by reconstruction algorithms whose objective is to transform the sensed data into meaningful cross-sectional images.
- In other words, images are not obtained directly from the sensors by motion alone; they require extensive processing. A 3-D digital volume consisting of stacked images is generated as the object is moved in a direction perpendicular to the sensor ring.
- Other modalities of imaging based on the CAT principle include magnetic resonance imaging (MRI) and positron emission tomography (PET).The illumination sources, sensors, and types of images are different, but conceptually they are very similar to the basic imaging approach shown in Fig. 5.3 (b).



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

(3) Image Acquisition Using Sensor Arrays:

- Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras.
- A typical sensor for these cameras is a CCD array, which can be manufactured with a broad range of sensing properties and can be packaged in rugged arrays of 4000 * 4000 elements or more. CCD sensors are used widely in digital cameras and other light sensing instruments.
- The response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor, a property that is used in astronomical and other applications requiring low noise images.
- The principal manner in which array sensors are used is shown in Fig.5.4. This figure shows the energy from an illumination source being reflected from a scene element.
- The first function performed by the imaging system shown in Fig.5.4 (c) is to collect the incoming energy and focus it onto an image plane.
- If the illumination is light, the front end of the imaging system is a lens, which projects the viewed scene onto the lens focal plane, as Fig. 2.15(d) shows.
- The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital and analog circuitry sweeps these outputs and converts them to a video signal, which is then digitized by another section of the imaging system.



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

Image sampling and quantization process

Image Sampling and Quantization:

The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

Basic Concepts in Sampling and Quantization:

The basic idea behind sampling and quantization is illustrated in Fig.6.1. Figure 6.1(a) shows a continuous image, f(x, y), that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization.

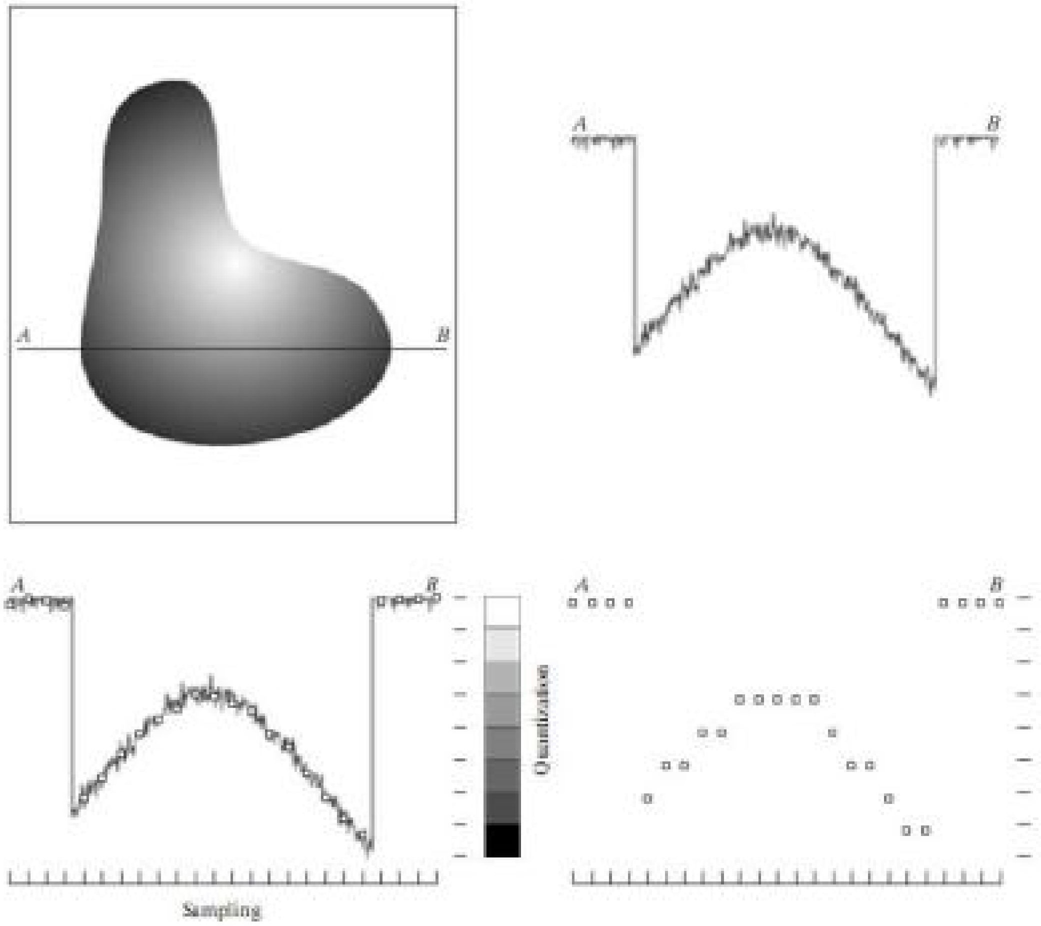
The one-dimensional function shown in Fig.6.1 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig. 6.1(a).The random variations are due to image noise.

To sample this function, we take equally spaced samples along line AB, as shown in Fig.6.1 (c).The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of Fig. 6.1 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark.

The digital samples resulting from both sampling and quantization are shown in Fig.6.1 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion, as in Fig. 2.13, the output of the sensor is quantized in the manner described above. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle; there is almost no limit as to how fine we can sample an image. However, practical limits are established by imperfections in the optics used to focus on the sensor an illumination spot that is inconsistent with the fine resolution achievable with mechanical displacements.

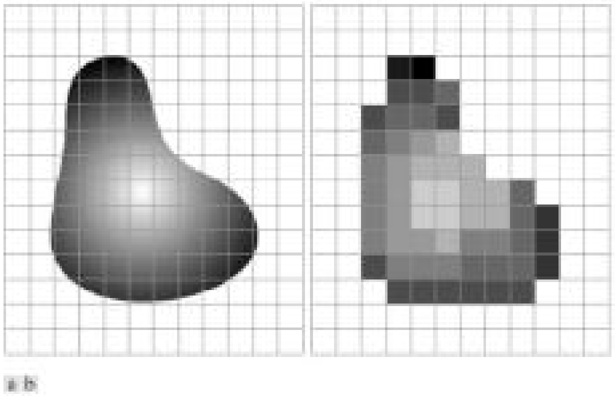
When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor outputs completes the process of generating a digital image.



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

When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Figure 6.2 illustrates this concept. Figure 6.2 (a) shows a continuous image projected onto the plane of an array sensor.

Figure 6.2 (b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization.



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

Basic relationships and distance measures between pixels in a digital image.

Neighbors of a Pixel:

- A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by (x+1, y), (x-1, y), (x, y+1), (x, y-1). This set of pixels, called the 4-neighbors of p, is denoted by N_4 (p). Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.
- The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) and are denoted by N_D (p). These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N_8 (p). As before, some of the points in N_D (p) and N_8 (p) fall outside the image if (x, y) is on the border of the image.

Connectivity:

- Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries.
- To establish if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal).
- For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are said to be connected only if they have the same value.
- Let V be the set of gray-level values used to define adjacency. In a binary image, V={1} if we are referring to adjacency of pixels with value 1. In a grayscale image, the idea is the same, but set V typically contains more elements.
- For example, in the adjacency of pixels with a range of possible gray-level values 0 to 255, set V could be any subset of these 256 values.
- We consider three types of adjacency:

(a) 4-adjacency. Two pixels p and q with values from V are 4-adjacent if q is in the set N_4 (p).
(b) 8-adjacency. Two pixels p and q with values from V are 8-adjacent if q is in the set N_8 (p).
(c) m-adjacency (mixed adjacency).Two pixels p and q with values from V are m-adjacent if
(i) q is in N_4 (p), or
(ii) q is in N_D(p) and the set has no pixels whose values are from V.

- Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.
- For example, consider the pixel arrangement shown in

Fig.9 (a) for V= {1}.The three pixels at the top of Fig.9 (b) show multiple (ambiguous) 8-adjacency, as indicated by the dashed lines. This ambiguity is removed by using m-adjacency, as shown in Fig. 9 (c).Two image subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel in S2. It is understood here and in the following definitions that adjacent means 4-, 8-, or m-adjacent.

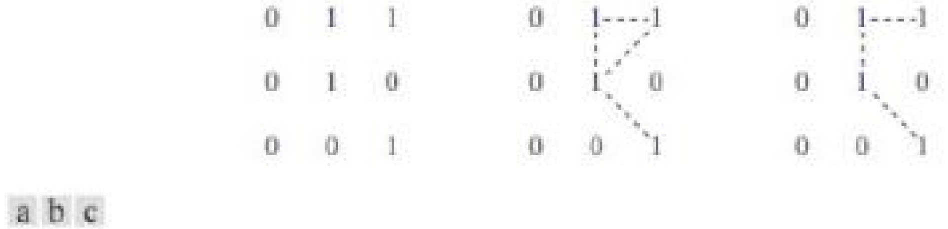
- A (digital) path (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates

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

where $(x_0, y_0) = (x, y), (x_n, y_n) = (s, t),$ and pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$.

- In this case, n is the length of the path. If $(x_0, y_0) = (x_n, y_n)$, the path is a closed path.
- Two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels in S. For any pixel p in S, the set of pixels that are connected to it

in S is called a connected component of S. If it only has one connected component, then set S is called a connected set.



(a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency

Distance Measures:

- For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a distance function or metric if

(a) $D(p, q) \geq 0 \quad (D(p, q) = 0 \quad \text{iff} \quad p = q)$,
(b) $D(p, q) = D(q, p)$, and
(c) $D(p, z) \leq D(p, q) + D(q, z)$.

- The **Euclidean distance** between p and q is defined as

$$D_e(p, q) = \left[(x - s)^2 + (y - t)^2\right]^{\frac{1}{2}}.$$

- For this distance measure, the pixels having a distance less than or equal to some value r from(x, y) are the points contained in a disk of radius r centered at (x, y).
- The **D_4 distance (also called city-block distance)** between p and q is defined as

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

- In this case, the pixels having a D_4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D_4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

The pixels with D_4 =1 are the 4-neighbors of (x, y).
The **D_8 distance (also called chessboard distance)** between p and q is defined as

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

In this case, the pixels with D_8 distance from(x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D_8 distance ≤ 2 from(x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

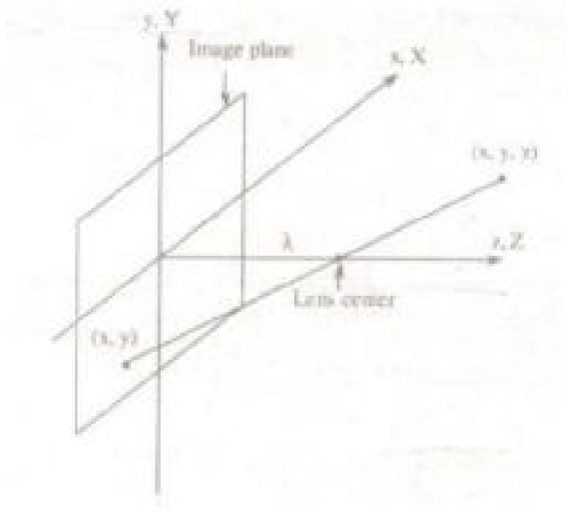
- The pixels with $D_8=1$ are the 8-neighbors of (x, y). Note that the D4 and D8 distances between p and q are independent of any paths that might exist between the points because these distances involve only the coordinates of the points. If we elect to consider m-adjacency, however, the D_m distance between two points is defined as the shortest m-path between the points.
- In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.
- For instance, consider the following arrangement of pixels and assume that p, p_2 , and p_4 have value 1 and that p_1 and p_3 can have a value of 0 or 1:

$$\begin{array}{ccc} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{array}$$

- Suppose that we consider adjacency of pixels valued 1 (i.e. = {1}). If p_1 and p_3 are 0, the length of the shortest m-path (the Dm distance) between p and p_4 is 2. If p1 is 1, then p_2 and p will no longer be m-adjacent (see the definition of m-adjacency) and the length of the shortest m-path becomes 3 (the path goes through the points $pp_1p_2p_4$).
- Similar comments apply if p_3 is 1 (and p_1 is 0); in this case, the length of the shortest m-path also is 3. Finally, if both p_1 and p_3 are 1 the length of the shortest m-path between p and p4 is 4. In this case, the path goes through the sequence of points $pp_1p_2p_3p_4$

Perspective image transformation

- A perspective transformation (also called an imaging transformation) projects 3D points onto a plane.
- Perspective transformations play a central role in image processing because they provide an approximation to the manner in which an image is formed by viewing a 3D world.
- These transformations are fundamentally different, because they are nonlinear in that they involve division by coordinate values.



Basic model of imaging process, The camera coordinate system 9x,y,z) is aligned with world coordinate system (X,Y,Z)

- The camera coordinate system (x, y, z) has the image plane coincident with the xy plane and the optical axis (established by the center of the lens) along the z axis. Thus the center of the image plane is at the origin, and the centre of the lens is at coordinates (0.0, λ).
- If the camera is in focus for distant objects, λ is the focal length of the lens. Here the assumption is that the camera coordinate system is aligned with the world coordinate system (X, Y, Z).
- Let (X,Y,Z) be the world coordinates of any point in a 3D scene, as shown in figure 10. We assume throughout the following discussion that Z> λ (i.e) all points of interest lie in front of lens.
- The first step is to obtain relationship that gives the coordinates(x,y) of the projection of point (X,Y,Z) onto image plane.

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda}$$
$$= \frac{X}{\lambda - Z}$$

$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda}$$
$$= \frac{Y}{\lambda - Z}$$

- Where, negative signs in front of X and Y indicate that image points are actually inverted.
- The image plane coordinates of the projected 3D point follow directly from above equations

$$x = \frac{\lambda X}{\lambda - Z}$$

$$y = \frac{\lambda Y}{\lambda - Z}$$

- These equations are nonlinear because they involve division by the variable Z. Although we could use them directly as shown, it is often convenient to express them in linear matrix form, for rotation, translation and scaling. This is easily accomplished by using homogeneous coordinates.
- The homogeneous coordinates of a point with Cartesian coordinates (X, Y, Z) are defined as (kX, kY, kZ. k), where k is an arbitrary, nonzero constant. Clearly, conversion of homogeneous coordinates back to Cartesian coordinates is accomplished by dividing the first three homogeneous coordinates by the fourth.
- A point in the Cartesian world coordinate system may be expressed in vector form as

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

and its homogeneous counterpart is

$$\mathbf{w}_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

If we define the perspective transformation matrix as

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix}$$

The product Pwh yields a vector denoted ch

$$\begin{aligned} \mathbf{c}_h &= \mathbf{P}\mathbf{w}_h \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \\ &= \begin{bmatrix} kX \\ kY \\ kZ \\ \frac{-kZ}{\lambda} + k \end{bmatrix} \end{aligned}$$

- The element of ch is the camera coordinates in homogeneous form. As indicated, these coordinates can be converted to Cartesian form by dividing each of the first three components of ch by the fourth. Thus the Cartesian of any point in the camera coordinate system are given in vector form by

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}$$

- The first two components of c are the (x, y) coordinates in the image plane of a projected 3-D point (X, Y, Z). The third component is of no interest in terms of the model in Fig. 10. As shown next, this component acts as a free variable in the inverse perspective transformation
- The inverse perspective transformation maps an image point back into 3-D.

$$\mathbf{w}_h = \mathbf{P}^{-1}\mathbf{c}_h$$

Where P-1 is

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix}$$

- Suppose that an image point has coordinates $(x_0, y_0, 0)$, where the 0 in the z location simply indicates that the image plane is located at $z = 0$. This point may be expressed in homogeneous vector form as

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \qquad \mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix}$$

or, in Cartesian coordinates

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}$$

- This result obviously is unexpected because it gives $Z = 0$ for any 3-D point. The problem here is caused by mapping a 3-D scene onto the image plane, which is a many-to-one transformation.
- The image point (x_0, y_0) corresponds to the set of collinear 3-D points that lie on the line passing through $(x_0, y_0, 0)$ and $(0, 0, \lambda)$. The equation of this line in the world coordinate system; that is,

$$X = \frac{x_0}{\lambda}(\lambda - Z)$$

$$Y = \frac{y_0}{\lambda}(\lambda - Z).$$

- Equations above show that unless something is known about the 3-D point that generated an image point (for example, its Z coordinate) it is not possible to completely recover the 3-D point from its image.
- This observation, which certainly is not unexpected, can be used to formulate the inverse perspective transformation by using the z component of c_h as a free variable instead of 0. Thus, by letting

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix} \qquad \mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \dfrac{kz}{\lambda} + k \end{bmatrix}$$

which upon conversion to Cartesian coordinate gives

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda x_0}{\lambda + z} \\ \dfrac{\lambda y_0}{\lambda + z} \\ \dfrac{\lambda z}{\lambda + z} \end{bmatrix}$$

In other words, treating z as a free variable yields the equations

$$X = \frac{\lambda x_0}{\lambda + z}$$

$$Y = \frac{\lambda y_0}{\lambda + z}$$

$$Z = \frac{\lambda z}{\lambda + z}$$

Solving for z in terms of Z in the last equation and substituting in the first two expressions yields

$$X = \frac{x_0}{\lambda}(\lambda - Z)$$

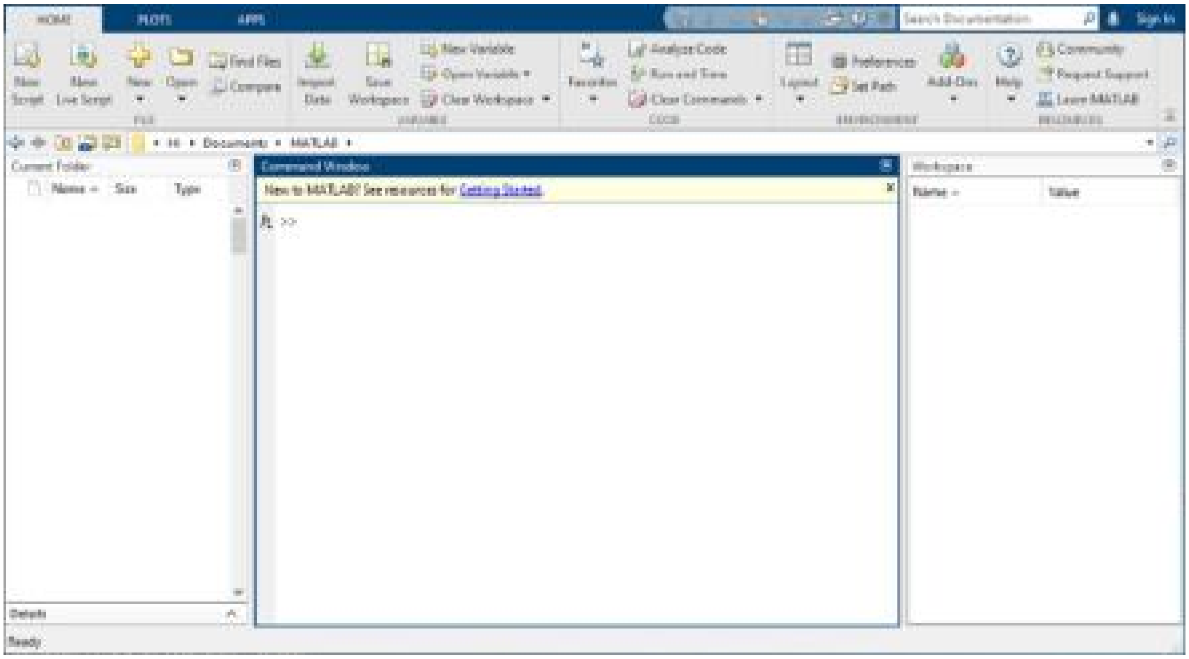
$$Y = \frac{y_0}{\lambda}(\lambda - Z)$$

which agrees with the observation that revering a 3-D point from its image by means of the inverse perspective transformation requires knowledge of at least one of the world coordinates of the point.

MATLAB:

The MATLAB Desktop

When you start MATLAB®, the desktop appears in its default layout.



New to MATLAB? See resources for Getting Started.

```
>>
```

The desktop includes these panels:

- **Current Folder** — Access your files.
- **Command Window** — Enter commands at the command line, indicated by the prompt (>>).
- **Workspace** — Explore data that you create or import from files.

Using the MATLAB Editor/Debugger

1 Statements:

- Statements are entered in the Command Window at the command prompt: >>.
- The comment symbol in MATLAB is the percent sign: %.
- When MATLAB encounters this symbol, it ignores the remaining text on that line. Comments are useful for creating “headers” for scripts.

Example Code: >> % Data Analysis and Plotting Script

- Results from statements issued are by default stored in the single variable named ans. This variable will show up in the Workspace and appear in the Command Window:
  Example Code:
  ```
  >> 3.45
  ans = 3.4500
  >>
  ```
- Pressing the Enter key causes MATLAB to execute the statement in the command prompt.
- Example Code: >> sqrt(1.44)
  ans = 1.2000

2 M-Files

- Files with extension .m are executable in MATLAB. There are two flavors of m-files: Script and Function.
- All m-files contain plain text, and as such can be edited with any text editor. MATLAB provides a customized text editor, as discussed above.
- However, you may read, write, and edit m-files with any plain text editor (such as Notepad, WinEdt, emacs, etc.).

3 Scripts

- Scripts are helpful because they can lump dozens or even thousands of Command Line statements into a single action. The conventions for naming scripts are very similar to those for naming variables:
  - Do Not begin filenames with numbers
  - Do Not include punctuation anywhere in filenames
  - Do Not use a filename that is already used for a MATLAB function or an existing variable

  There are a few useful shortcut keys when writing and debugging scripts. The F5 key saves and executes the entire active script. When dealing with long files, it is often useful to execute only the code you have just written. The F9 key executes only the text that is currently highlighted.

4 Functions

- There are thousands of functions in the MATLAB library. However, one may find it necessary to write a new function—often a collection of other functions—to suit one's specific needs. Just like MATLAB native functions, user-defined functions have a strict yet simple syntax.
- To define a new function, there are four essential components that must be the first executable line in the m-file:
  - The MATLAB command function
  - Output variables
  - The user-defined function name
  - Input variables

Getting Help

- MATLAB's biggest advantage over other programming languages is its library of functions. To aid users in finding a function, MATLAB has two very useful commands.

- To see a list of functions that have a certain word in the title or description, use the lookfor command.

  Example Code:

  ```
  >> lookfor sinc
  DIRIC Dirichlet, or periodic sinc function
  SINC Sin(pi*x)/(pi*x) function.
   INVSINC Desired amplitude and frequency response for invsinc filters
   >>
  ```

- To get explicit information on a single function whose name is known, the help function is extremely useful. It explains the syntax and purpose of every function in the MATLAB library. Example Code:

  ```
  >> help sqrt
  SQRT Square root.
  SQRT(X) is the square root of the elements of X. Complex results are produced if X is not positive.
  See also sqrtm.
   >>
  ```

Saving and Retrieving work Session Data

SPTool Sessions

When you start SPTool, the default startup.spt session is loaded. To save your work in the startup SPTool session, use **File > Save Session** or to specify a session name, use **File > Save Session As.**

To recall a previously saved session, use **File > Open Session.**

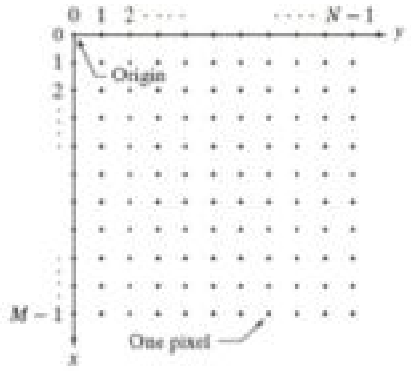
Digital Image Representation

Digital image representation

- The common mathematical representation of an image is a function of two continuous spatial coordinates: f(x,y)f(x,y). The value f(x0,y0)f(x0,y0) is often called the image *intensity* at (x0,y0)(x0,y0), although that term is used loosely because it often does not represent actual light intensity. The term *gray level* is also commonly used.
- You can represent a color image mathematically in a couple of different ways.
  - One way is to use a collection of image functions, one per color component, such as r(x,y)r(x,y), g(x,y)g(x,y), and b(x,y)b(x,y).
  - Another way is to use a vector-valued function, f(x,y)f(x,y).
- Many functions in the Image Processing Toolbox also support higher-dimensional images. For example, ff might be a function of three spatial variable, as in f(x,y,z)f(x,y,z). Image Processing Toolbox documentation calls this a *multidimensional image*.
- Converting the amplitude values of ff to a set of discrete values is called *quantization.* Capturing those values at discrete spatial coordinates is called *sampling.* When xx, yy, and the values of ff are all finite, discrete quantities, we call *f* a *digital image*.

Coordinate conventions

- It's important to pay attention to different coordinate system conventions that may be used in image processing. Often the center of the upper-left pixel is considered to be the origin, (0,0). There is more variation in the assignment of xx and yy axes.
- The coordinate system in the diagram below, with the xx-axis pointing down and the yy-axis pointing to the right



- When displaying images in MATLAB, the usual convention is for the center of the upper-left pixel to be at (1,1), the xx-axis to point to the right, and the yy-axis to point down.

Images as matrices and arrays

- Digital images are very conveniently represented as matrices, which happens to be great for working with in MATLAB. A monochrome image matrix looks like this:

```
f(1,1)   f(1,2)   ...   f(1,N)

f(2,1)   f(2,2)   ...   f(2,N)

  .        .              .

  .        .              .

  .        .              .

f(M,1)   f(M,2)   ...   f(M,N)
```

- We represent color images in MATLAB as multidimensional arrays. \
- For example, an RGB image (with three color components) is represented as an M-by-N-by-3 array.
- A CMYK image (with four color components) would use an M-by-N-by-4 array.
- Multidimensional images are also represented in MATLAB as multidimensional arrays. Therefore we rely on context to distinguish between an RGB image (M-by-N-by-3) and a three-dimensional image with three z-plane slices (also M-b-N-by-3).
- For example, if you pass an M-by-N-by-3 array to rgb2gray, it is clear from the context that the input should be interpreted as an RGB color image and not as a three-dimensional image.

<u>Image I/O and Display</u>

- **imread():** To read an image and store in a matrix.

Syntax: IM=imread('filename')
Eg.: >>im=imread('peppers.png');

- **imshow():** Displays the image. Syntax:

Syntax: imshow('filename')
or imshow(im)
Example:-
>>imshow('cameraman.tif");

Classes and Image Types

Different Types of Images

Binary Image: An image that consists of only black and white pixels. Technically these types of images are called as Black and White Image. (Although it makes me sad to break my reader's heart but till now what you called black and white images have some other technical name).

Grayscale Image: In daily language what we refer to as black-and-white (as in old photos) are actually grayscale. It contains intensity values ranging from a minimum (depicting absolute black) to a maximum (depicting absolute white) and in between varying shades of gray. Typically, this range is between 0 and 255.

Color Image: We all have seen this! Such an image is composed of the three primary colors, Red, Green and Blue, hence also called an RGB image.

Binary Image

Grayscale Image

Color Image

M-File Functions

- Functions are program routines, usually implemented in M-files, that accept input arguments and return output arguments.
- They operate on variables within their own workspace. This workspace is separate from the workspace you access at the MATLAB command prompt.

The Function Workspace

- Each M-file function has an area of memory, separate from the MATLAB base workspace, in which it operates. This area, called the function workspace, gives each function its own workspace context.
- While using MATLAB, the only variables you can access are those in the calling context, be it the base workspace or that of another function.
- The variables that you pass to a function must be in the calling context, and the function returns its output arguments to the calling workspace context.
- You can, however, define variables as global variables explicitly, allowing more than one workspace context to access them.

Simple Function Example

The average function is a simple M-file that calculates the average of the elements in a vector:

```
function y = average(x)
% AVERAGE Mean of vector elements.
% AVERAGE(X), where X is a vector, is the mean of vector elements.
% Nonvector input results in an error.
[m,n] = size(x);
if (~((m == 1) | (n == 1)) | (m == 1 & n == 1))
    error('Input must be a vector')
end
y = sum(x)/length(x);      % Actual computation
```

Try entering these commands in an M-file called average.m. The average function accepts a single input argument and returns a single output argument. To call the average function, enter

```
z = 1:99;

average(z)
ans =
    50
```

Variable conversion

syntax	Conversion
int2str	Convert integers to characters
mat2str	Convert matrix to characters
num2str	Convert numbers to character array
str2double	Convert strings to double precision values
str2num	Convert character array or string to numeric array
bin2dec	Convert text representation of binary number to decimal number
dec2bi	Convert text representation of decimal number to binary number

Variable

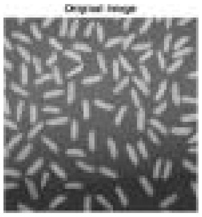
- Read:
  - ➢X = sprintf('%s will be %d this year.',name,age);
- Write:
  - ➢disp(X)
  - ➢X = sprintf('%s will be %d this year.',name,age); disp(X)
  - ➢fprintf('%s will be %d this year.\n',name,age);

Image conversion

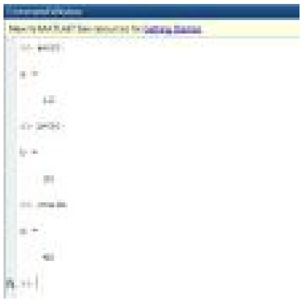
Syntax	Conversion
IM=imbinarize(B)	Conversion of image to binary form
BW = im2bw(I,level)	Conversion of image to binary form -a level value of 0.5 corresponds to an intensity value halfway between the minimum and maximum value of the class.
uint8Image = uint8(255 * binaryImage);	Conversion of image from binary to grey scale image

Sharpening filter

- B = imsharpen(A) sharpens the grayscale or truecolor (RGB) input image A by using the unsharp masking method.
  - ➢ b = imsharpen(a);
- B = imsharpen(A,Name,Value) uses name-value pairs to control aspects of the unsharp masking.
  - ➢ Eg.: b = imsharpen(a,'Radius',2,'Amount',1);

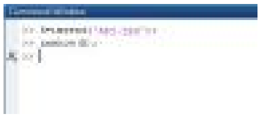


Simple matlab command for addition of 2 numbers

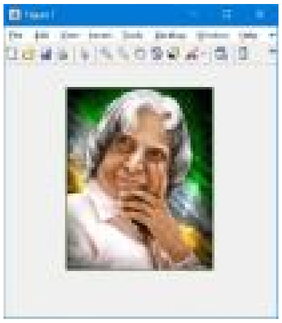


```
>> a=10
a =
    10
>> b=30
b =
    30
>> c=a+b
c =
    40
>> 
```

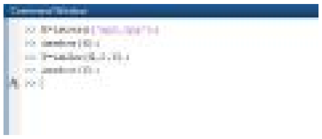
Simple matlab code to read and display an image



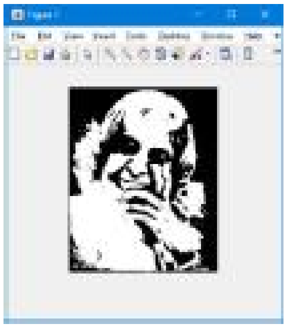
```
>> B=imread('[illegible]');
>> imshow(B);
>> 
```



Simple matlab code to convert image into binary image

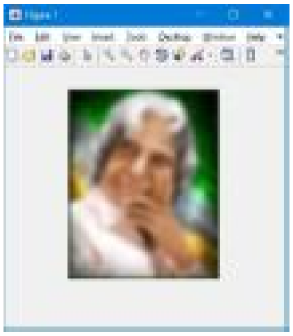


```
>> B=imread('[illegible]');
>> imshow(B);
>> I=im2bw(B,[illegible]);
>> imshow(I);
>> 
```



Simple code for smoothing

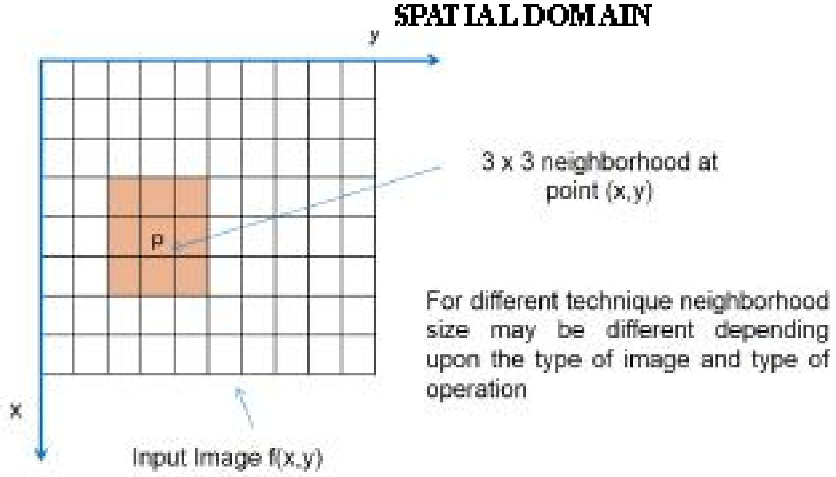
```
>> Iblur1 = imgaussfilt(I,[illegible]);
>> imshow(Iblur1);
>> 
```



Unit-II

Image enhancement: Spatial Domain – Gray level Transformations – Histogram Processing – Spatial Filtering – Smoothing and Sharpening. Frequency Domain: Filtering in Frequency Domain – DFT, FFT, DCT – Smoothing and Sharpening filters – Homomorphic Filtering.

SPATIAL DOMAIN


y
3 x 3 neighborhood at point (x,y)
P
For different technique neighborhood size may be different depending upon the type of image and type of operation
x
Input Image f(x,y)

The simplest of T is when neighbourhood is 1 x 1, where g is dependent on f, when T becomes gray-level.

$$s = T(r)$$

TRANSFORMATIONS:

Gray –level transformations:

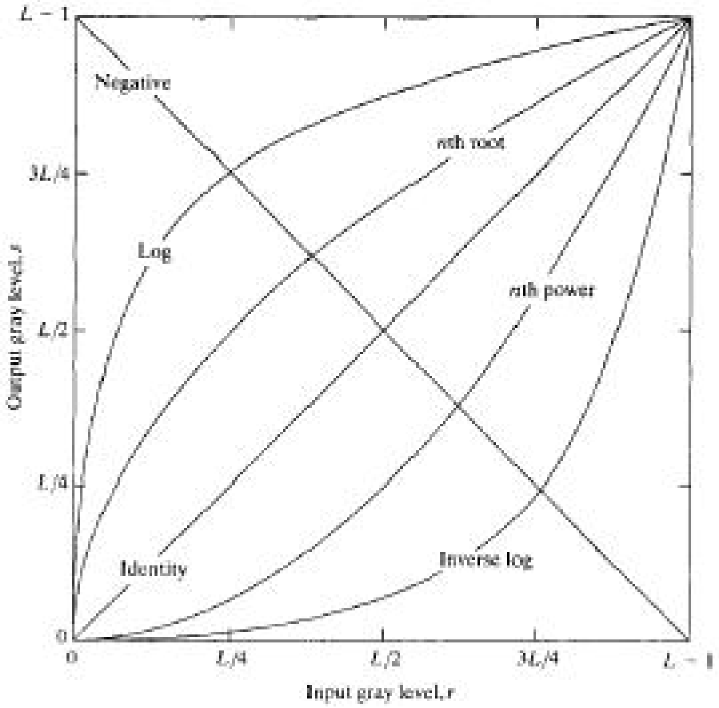
The values of transformation functions are stored in one-dimensional arrays and mapping from r to s via look up table, since digital quantities are dealt. For 8 bitenvironment, look-up table will have 256 entries. Three basic type functions in gray-level transformation are:

- Linear (negative and identity transformations)
- Log arithmetic (log and inverse log transformations)
- Power law (nth power and nth root transformations)

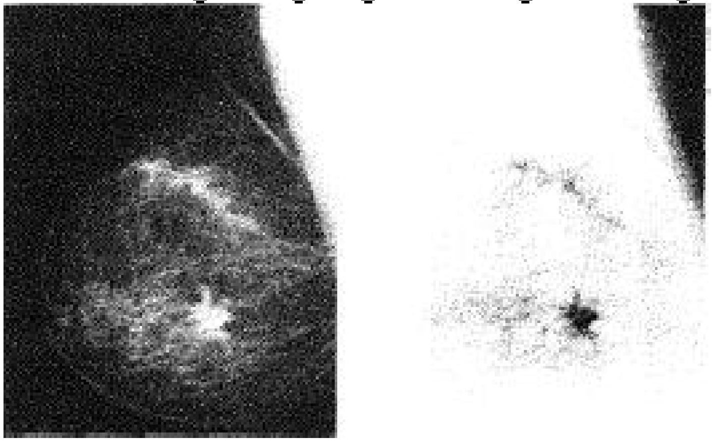
a) *Image negatives:*

The negative of image with Gray levels in range[0,L-1] by,

$$s = L - 1 - r.$$



Reversing intensity levels of image gives photographic negative, which is suited for white or gray detail in dark regions. Eg.: Digital mammogram showing brain tissue.



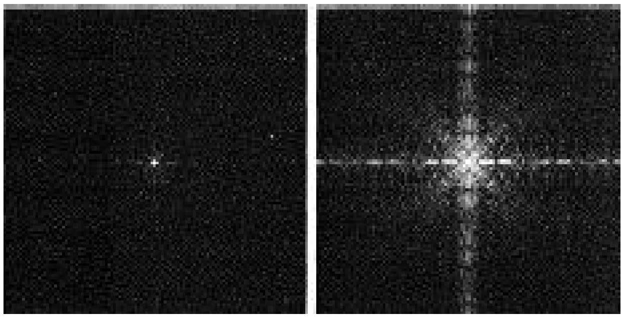
b) Log transformations :

General log transformation

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

Where, C is constant; assume r>=0

A narrow range of low gray level in input is transformed into wider range at output. The opposite is true with higher level of input. This is used to expand values of dark pixels of imagewhile compression.

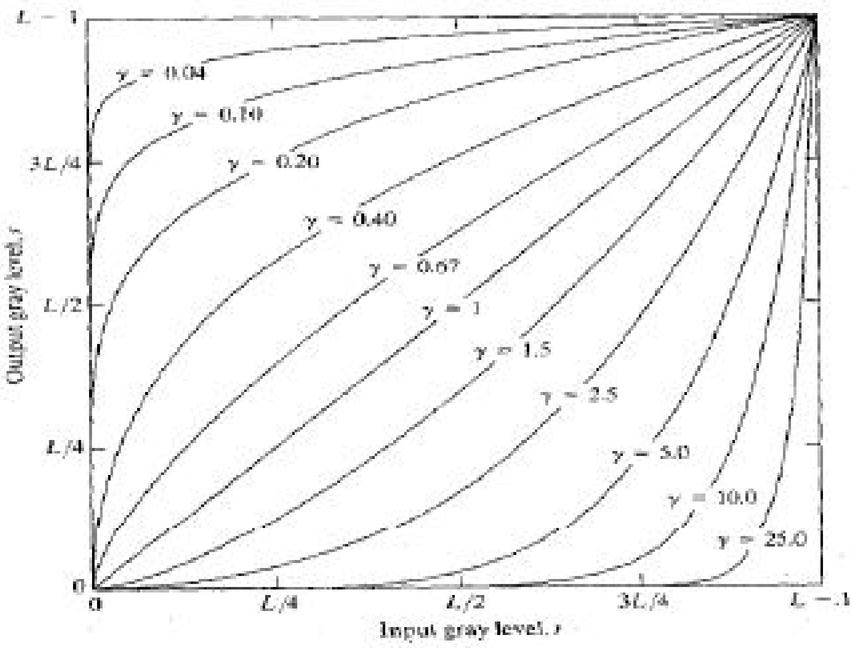


c) Power law transformations:

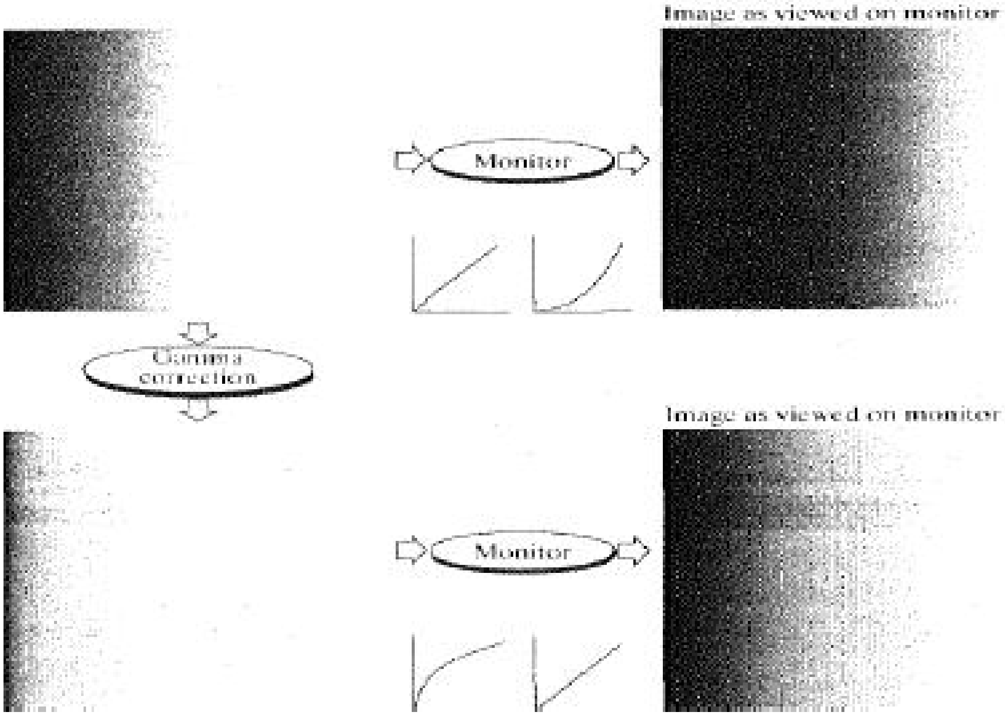
Power law has the basic transformation as ,

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

Where, c and Υ are positive constants. Power law curves with narrow range of dark values are mapped to wider range with opposite being true for higher values



. Unlike log transformation, the possible transformation curves can be obtained by just varying Υ. A variety of devices used for image capturing, printing and displaying follow power transformation law.the process used to correct power law transformation is called *gamma correction*.



A straight-forward gamma correction in case of simple gray scale linear wedge input into Cathode Ray Tube(CRT).

d) Piece-wise linear transformation functions:

Contrast stretching:

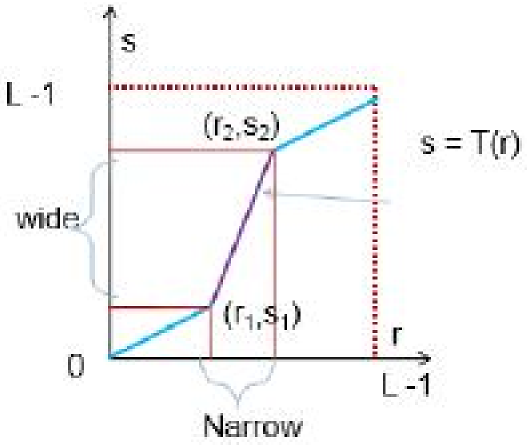
□ Contrast image is process of increasing low contrast image to high contrast image.

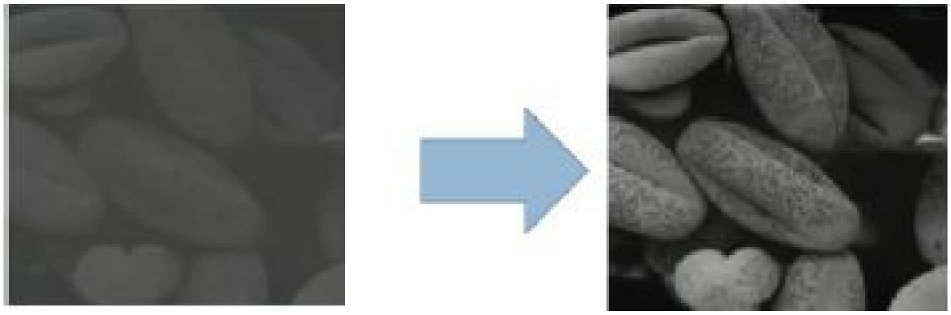
□ Narrow intensity range in original image is converted wide intensity range in processed image

□ For enhancement operation

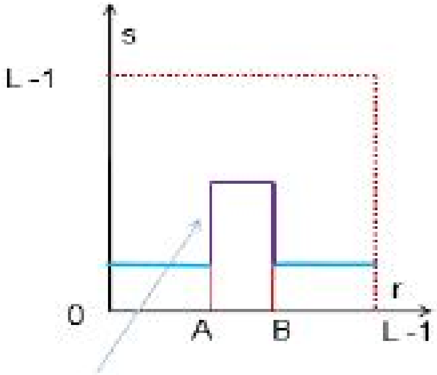
□ $\mathbf{r_1 \leq r_2}$ **and** $\mathbf{s_1 \leq s_2}$

□ For $r_1 = r_2$, $s_1 = 0$ and $s_2 = L\text{-}1$ the transfer function become Thresholding operation



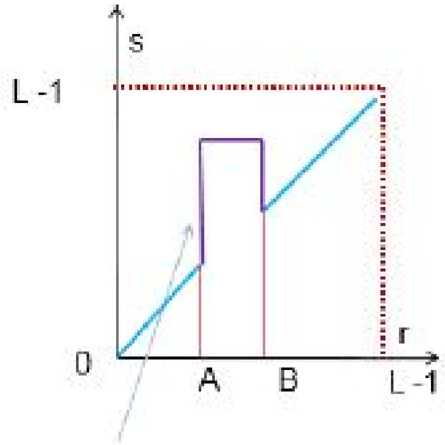


Gray Level Slicing



Concept:

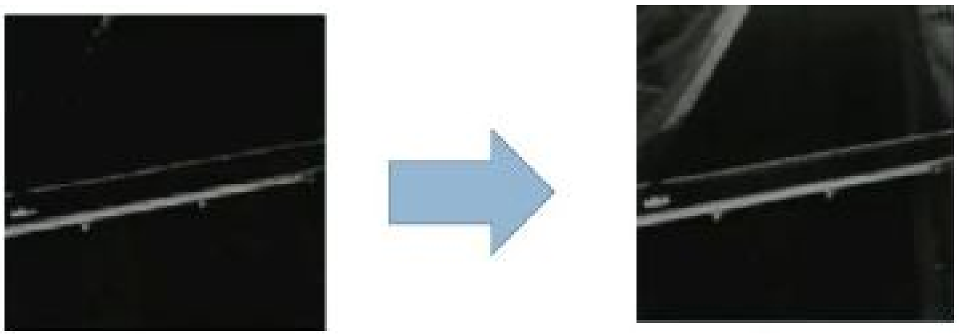
Here it shows that only in the range A – B the image is enhanced and for all other pixel value are suppressed



Concept:

Here it shows that only in the range A – B the image is enhanced and all other pixel value are retained

Gray Level Slicing Results:



Histogram Based Technique

- □ Histogram Definition
- □ A image having (0 , L-1) discrete intensity level
- □ Then Histogram:

$h(r_k) = n_k$

- □ Where r_k: Intensity level
- □ n_k : No. of pixel having intensity level r_k
- □ The plot of distinct intensity level against all possible intensity level is known at Histogram
- □ Nomalized Histogram:

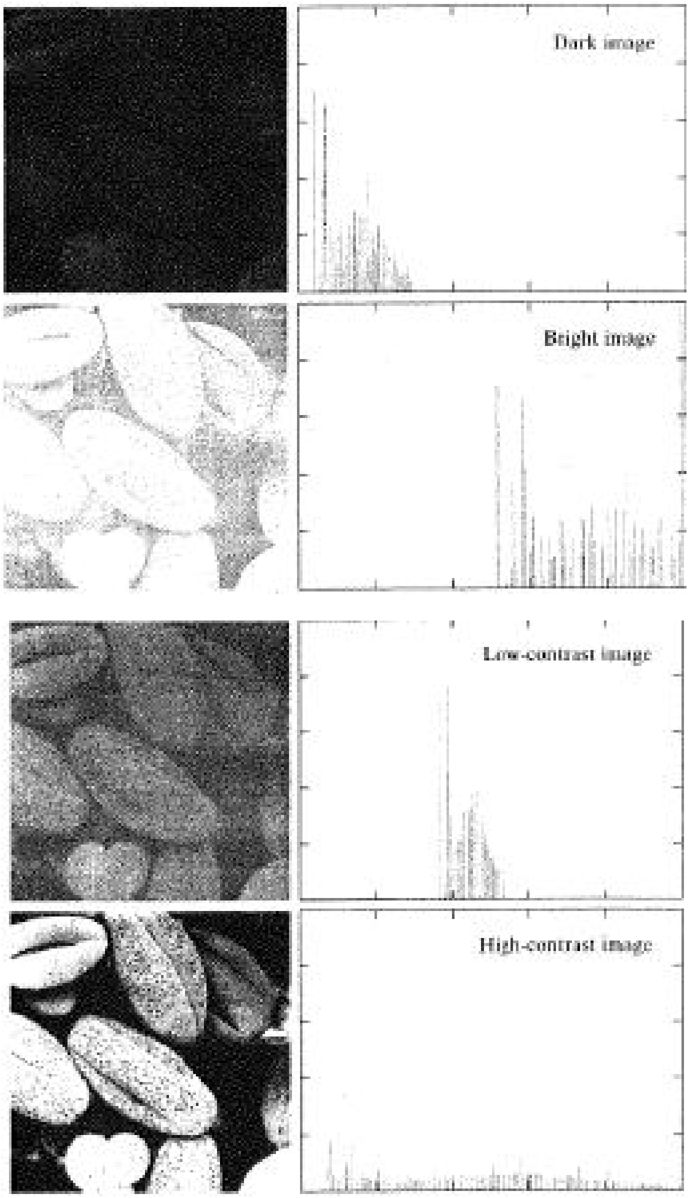
$p(r_k) = n_k/n$

- □ Where n is the total no. of pixel in the image

□ $P(r_k)$ indicate the probability of occurrence of intensity of level r_k in the image

□ Histogram Techniques includes

- ▣ Histogram equalization
- ▣ Histogram specification
- ▣ Image Subtraction
- ▣ Image Averaging



Histogram Equalization

□ Let r represents gray level in an image

□ Assume [0,1] is the normalized pixel in the image where 0 → back pixel and 1 → white pixel

$s = T(r)$

□ where r is the intensity in original image and s is the intensity in the processed image

□ We assume the transfer function satisfy the following condition

1. T(r) must be a single valued & monotonically increasing in the range $0 \leq r \leq 1$ i.e. Darker pixel remains darker in the processed image
2. $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$ i.e. any pixel intensity vale may not be larger than the maxim intensity level.

□ **Inverse Transfer**

$r = T^{1}(s)$

□ It should also satisfy the above condition

□ For continuous domain

□ $p_r(r)$ → pdf of r

□ $p_s(s)$ → pdf of s

□ For elementary probability theory we know that if $p_r(r)$ and $p_s(s)$ are known and T-1(s) is single valued monotonically increasing function

□ Then we can obtaine

$$p_s(s) = p_r(r)|dr/ds|_{at\ r = T\text{-}1(s)}$$

□ Now we take transfer function

□ $s = T(r) = \int_{0 \to r} pr(w)\, dw$

□ From this we can compute

$ds/dr = p_r(r)$

□ $p_s(s) = p_r(r)\ |dr/ds|$

□ $= p_r(r).\ 1/p_r(r) = 1$

□ Discrete Formulation

□ $p_r(r_k) = n_k/n$

□ Histogram equalization become

$s_k = T(r_k) = \sum_{j=0 \to k} p_r(r_j)$

$= \sum_{j=0\to k} n_j/n$

Histogram Specification/Matching:

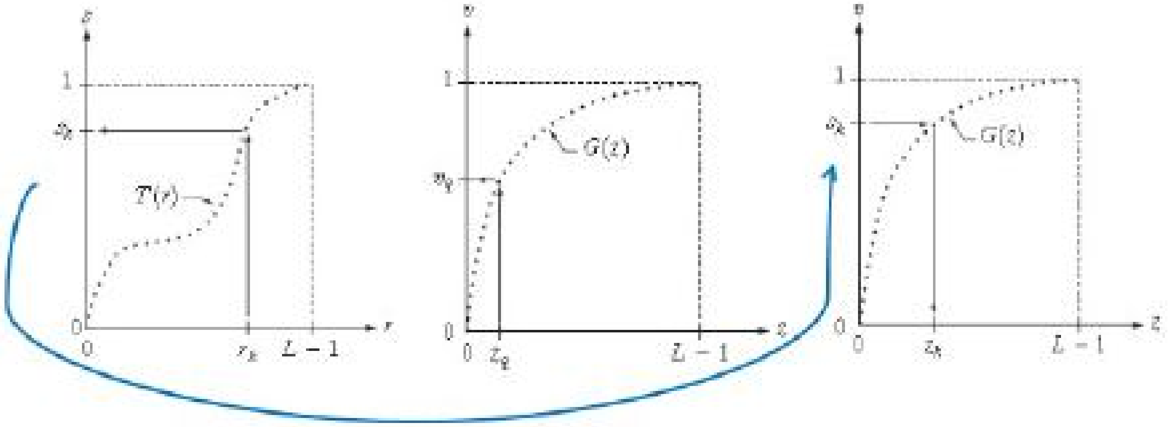
- □ **Target Histogram**
- □ Let r → given image and z → target area in the image
- □ Hence $p_z(z) \to$ target histogram
- □ $s = T(r) = \int_{0\to r} p_r(w)\,dw$
- □ Similarly we have function G(z) instead T(r) for target histogram
- □ $G(z) = \int_{0\to z} p_z(t)\,dt$
- □ Discrete formulation

$s_k = T(r_k) = \sum_{j=0\to k} n_j/n = \sum_{j=0\to k} p_r(r_j)$

Let $p_z(z)$ is the **specified target histogram**

$V_k = G(z_k) = \sum_{i=0\to k} p_z(z_i) = s_k$

$Z_k = G^{-1}\,[T(r_k)] = G^{-1}(s_k)$

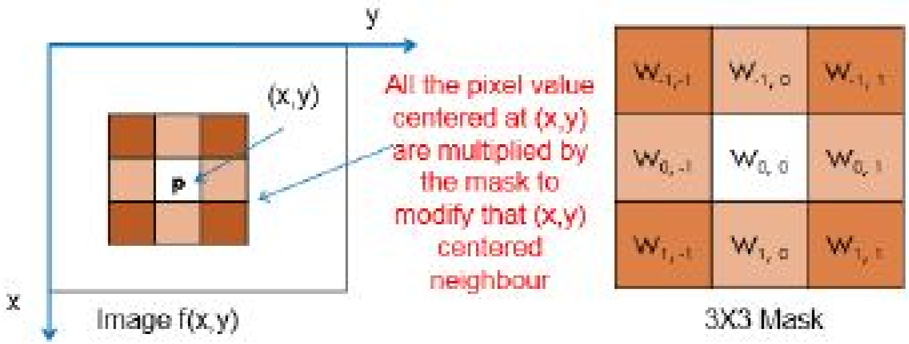


Graphical interpretation of mapping from r_k to s_k via T(r).

Mapping of z_q to its corresponding value v_q via G(z).

Inverse mapping from s_k to its corresponding value of z_k

Spatial Filtering:



All the pixel value centered at (x,y) are multiplied by the mask to modify that (x,y) centered neighbour

g(x,y) = $\sum\sum w_{ij}$ f(x+i, y+i) for i = -1 to 1 & j = -1 to 1

Where w_{ij} is the mask coefficient and this mask coefficient is different for different techniques

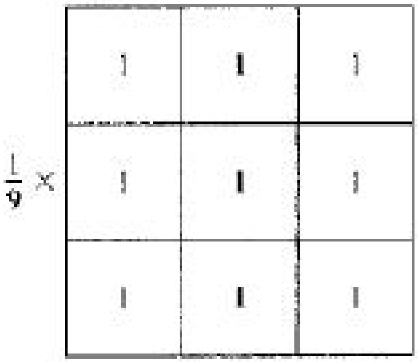
For mask processing neighborhood must be greater than 1

Operations: Image sharpening, Averaging etc.

Spatial filters:

- ***Smoothing:***
  - **Linear Smoothing Filter**
    - Performing averaging on an image means smoothening an image

- If an image is smoothen much more it become blurred
- Averaging Filter



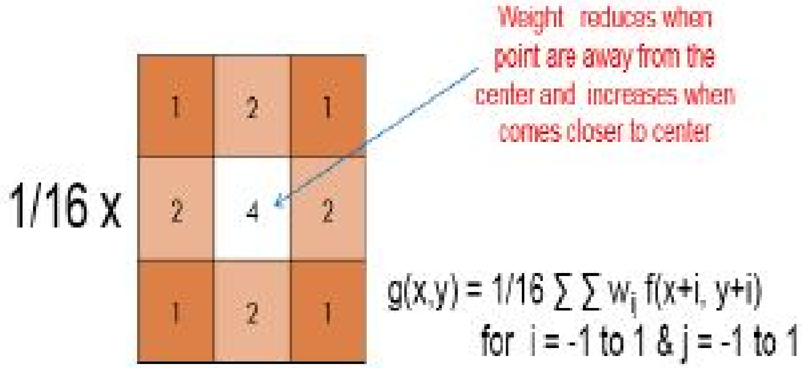
$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$g(x,y) = 1/9 \sum \sum w_{ij} f(x+i, y+i)$

for $i = -1$ to 1 & $j = -1$ to 1

- Weighted Averaging



1/16 x

1	2	1
2	4	2
1	2	1

Weight reduces when point are away from the center and increases when comes closer to center

$g(x,y) = 1/16 \sum \sum w_{i} f(x+i, y+i)$

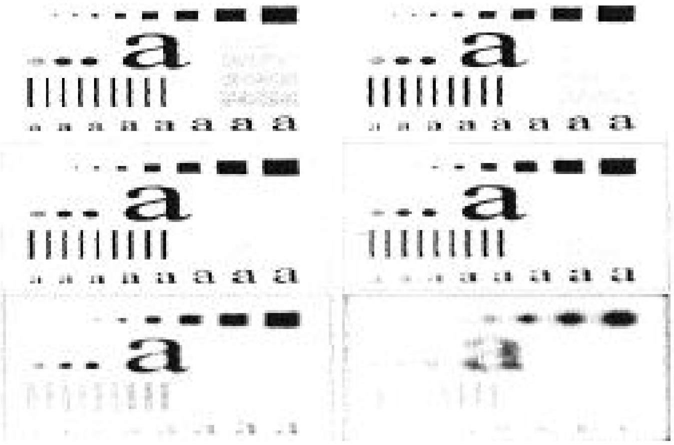
for $i = -1$ to 1 & $j = -1$ to 1

In general form :

$$g(x,y) = \frac{\sum_{i=-a}^{a} \sum_{j=-b}^{b} w_{ij} f(x+i, y+i)}{\sum_{i=-a}^{a} \sum_{j=-b}^{b} w_{ij}}$$

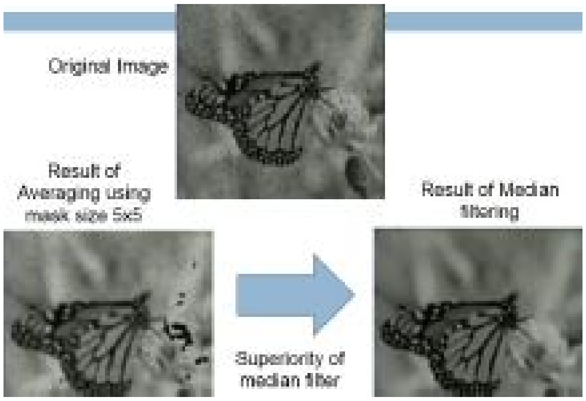
For mask size MXN where $M = 2a + 1$ & $N = 2b + 1$

Results



(a) Original image, of size 500*500 pixels(b)–(f) Results of smoothing with. square averaging filter masks of sizes n=3, 5, 9, 15, and 35, respectively.

- Median Filter (nonlinear)
  - It is also known as Order-statistics filters
  - Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
  - Median filter is a well known nonlinear filter


Original Image
Result of Averaging using mask size 5x5
Result of Median filtering
Superiority of median filter

- **Sharpening:**
  - Sharpening Filter
    - In the smoothening filters (linear &nonlinear) the image are smoothen and as a result image loses it detailing.

- Image sharpening is a process to enhance an image such that it can extract various details of an image
- Averaging is basically an integration operation whereas sharpening uses derivative operation over image
- Two type of derivative operation are First Order Derivative & Second Order Derivative

Desirable Response of Derivative Filters

- First Order Derivative Filter.
  - Must be Zero in area of constant gray level
  - Non Zero at the onset of grey level step or ramp
  - Non Zero along ramp
- Second Order Derivative Filter
  - Zero in flat area
  - Non Zero at onset and end of a gray level step or ramp
  - Zero along ramp of constant slope

- **First Order Derivative Filter.**

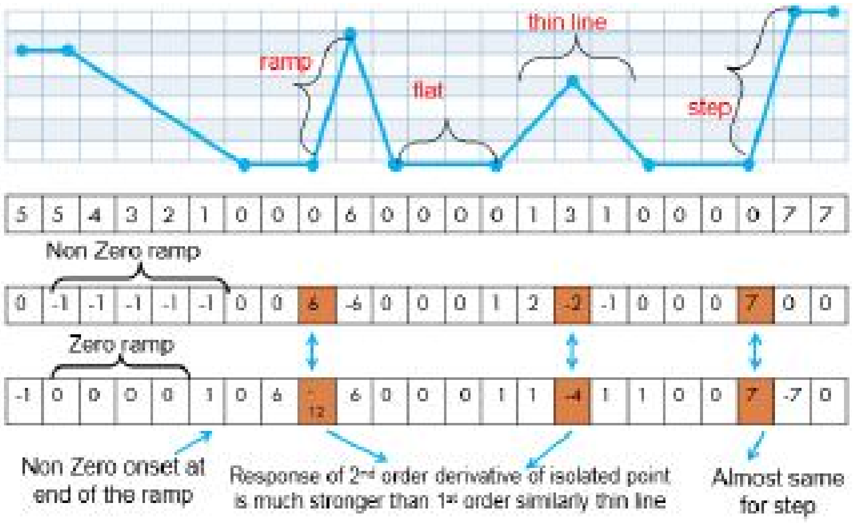
$$\frac{df(x)}{dx} = \underset{\Delta x \to 0}{Lt} \frac{f(x+\Delta x) - f(x)}{\Delta(x)}$$

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

- **Second Order Derivative Filter**

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

Choice between 1st & 2nd Order Derivative


thin line
ramp
flat
step
Non Zero ramp
Zero ramp
Non Zero onset at end of the ramp
Response of 2nd order derivative of isolated point is much stronger than 1st order similarly thin line
Almost same for step

5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7
0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0
-1	0	0	0	0	1	0	6	-12	6	0	0	0	1	1	-4	1	1	0	0	7	-7	0

Observation

- □ 1st Order derivation generally produce thicker edge in an image
- □ 2nd Order derivative gives stronger response to fine details such as thin line and isolated point
- □ 1st order derivative has stronger response to gray level step
- □ 2nd order derivative produces a double response at step edge

Second Order derivative are better suitable for image enhancement

- □ Discrete formulation of 2nd Order derivative filter should Isotropic in nature (The response of filter should be independent of the orientation of the discontinuity in the image)

Laplacian Operator

- □ Popularly known 2nd order derivative operator is Laplacian Operator, it is Isotropic in nature
- □ Laplacian Operator in continuous domain

$$\nabla^2 f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$$

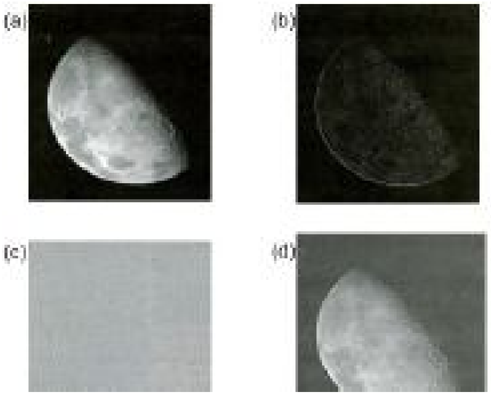
$$\frac{d^2 f}{dx^2} = f(x+1) + f(x-1) - 2f(x)$$

For image f(x,y)

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

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

Laplacian Operator Results



(a) Original Image

(b) Laplacian filtered iage

(c) Laplacian scaled image

(d) Enhanced image using laplacian operator

FREQUENCY DOMAIN

Basics of Filtering in the Frequency Domain

Frequency domain filtering operation

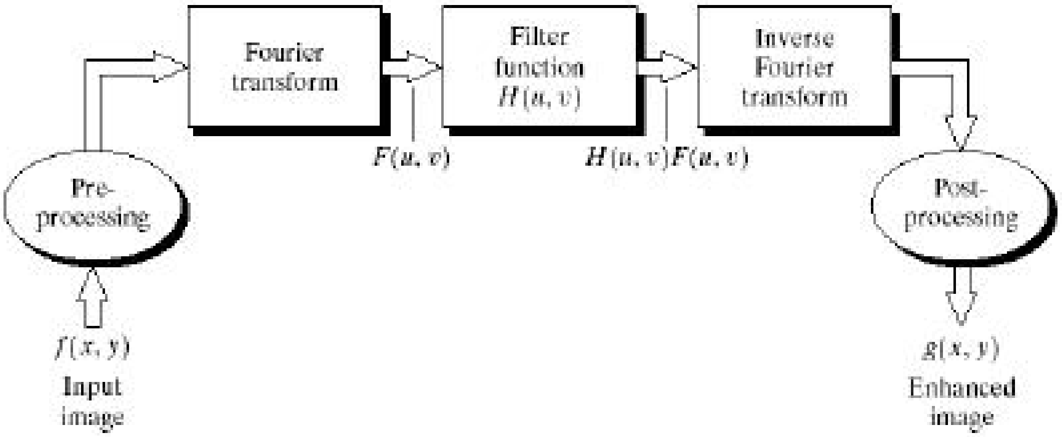


FIGURE 4.5 Basic steps for filtering in the frequency domain.

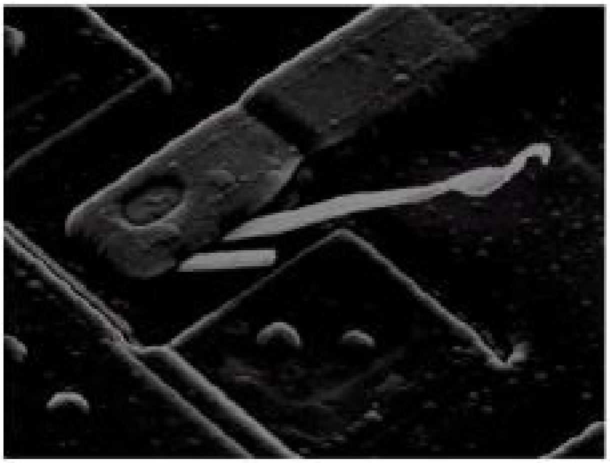
Some Basic Filters and Their Functions

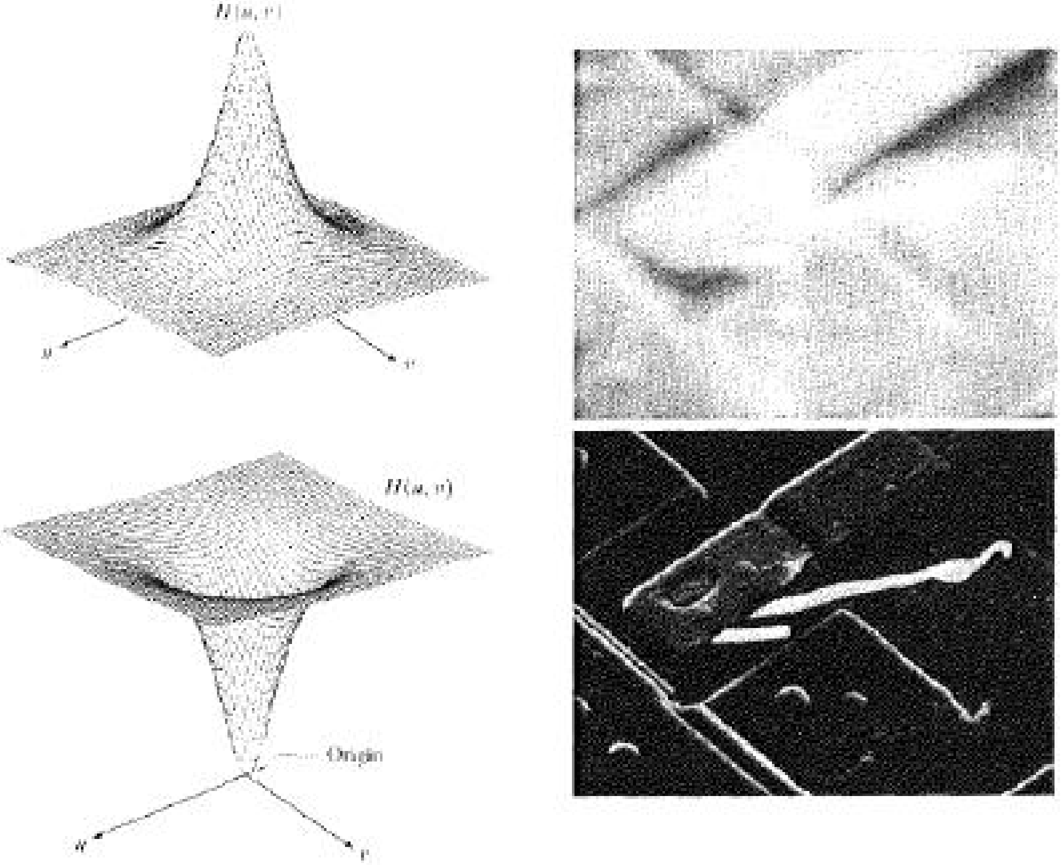
- Multiply all values of $F(u,v)$ by the filter function (notch filter):

$$H(u,v) = \begin{cases} 0 & \text{if } (u,v) = (M/2, N/2) \\ 1 & \text{otherwise.} \end{cases}$$

All this filter would do is set $F(0,0)$ to zero (force the average value of an image to zero) and leave all frequency components of the Fourier transform untouched.

FIGURE 4.6 Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the $F(0, 0)$ term in the Fourier transform.





a b
c d

FIGURE 4.7 **(a)** A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

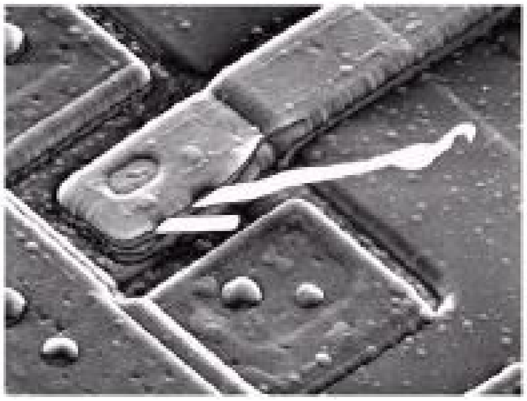
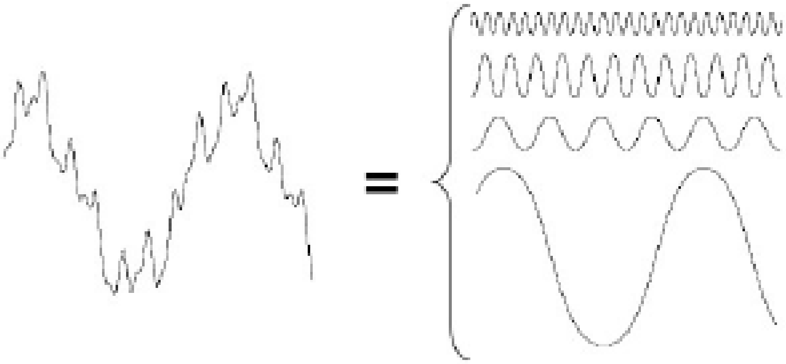


FIGURE 4.8 Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).

DFT

- Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient (Fourier series).
- Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function (Fourier transform).



Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*

- The **frequency domain** refers to the plane of the two dimensional discrete Fourier transform of an image.
- The purpose of the Fourier transform is to represent a signal as a linear combination of sinusoidal signals of various frequencies.
- The one-dimensional Fourier transform and its inverse
  - Fourier transform (continuous case)

    $$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \quad \text{where } j = \sqrt{-1}$$

  - Inverse Fourier transform:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

- The two-dimensional Fourier transform and its inverse
  - Fourier transform (continuous case)

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

  - Inverse Fourier transform:

    $$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} du dv$$

The one-dimensional Fourier transform and its inverse

Fourier transform (discrete case) DTC

$$F(u) = \frac{1}{M}\sum_{x=0}^{M\ 1} f(x) e^{\ j2\pi ux/M} \quad \text{for } u = 0,1,2,\ldots,M-1$$

- Inverse Fourier transform:

  $$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad \text{for } x = 0,1,2,\ldots,M-1$$

- Since $e^{j\theta} = \cos\theta + j\sin\theta$ and the fact $\cos(-\theta) = \cos\theta$

  then discrete Fourier transform can be redefined

  $$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)[\cos 2\pi ux/M - j\sin 2\pi ux/M]$$

  for $u = 0,1,2,\ldots,M-1$

  - Frequency (time) domain: the domain (values of u) over which the values of $F(u)$ range; because u determines the frequency of the components of the transform. Frequency (time) component: each of the M terms of $F(u)$.

$F(u)$ can be expressed in polar coordinates:

$$F(u) = |F(u)| e^{j\phi(u)}$$

where $|F(u)| = \left[R^2(u) + I^2(u)\right]^{\frac{1}{2}}$ (magnitude or spectrum)

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right] \quad \text{(phase angle or phase spectrum)}$$

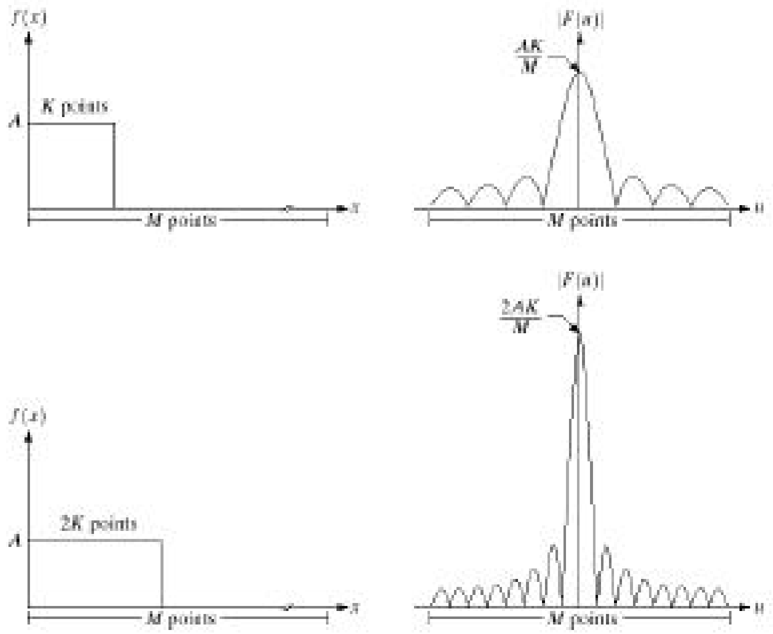
$R(u)$: the real part of $F(u)$

- $I(u)$: the imaginary part of $F(u)$

- Power spectrum:

  $$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

The One-Dimensional Fourier Transform Example

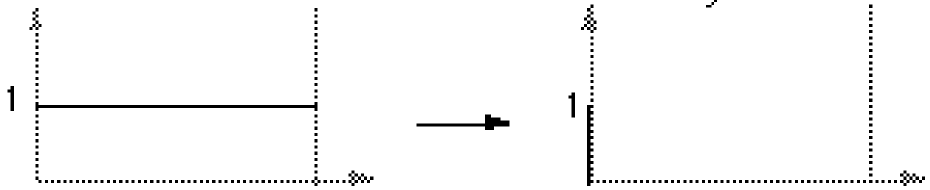


a b
c d

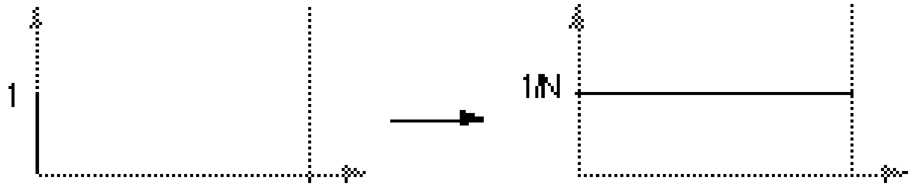
FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

The One-Dimensional Fourier Transform Some Examples

- The transform of a constant function is a DC value only.

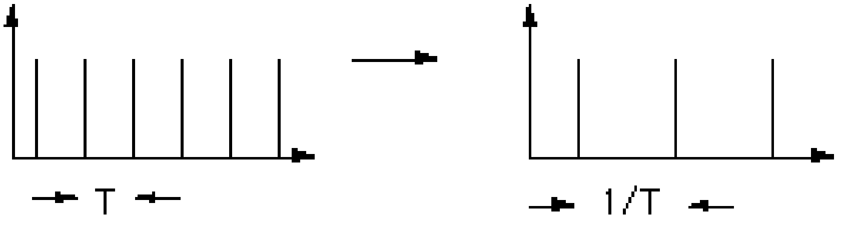


- The transform of a delta function is a constant.

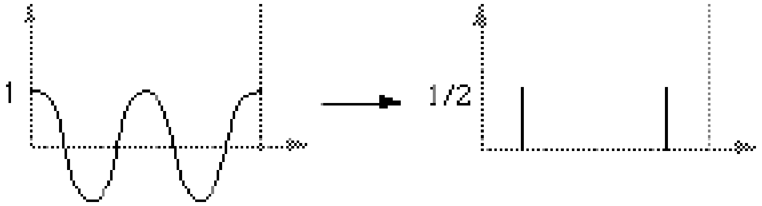


The One-Dimensional Fourier Transform Some Examples

- The transform of an infinite train of delta functions spaced by T is an infinite train of delta functions spaced by 1/T.

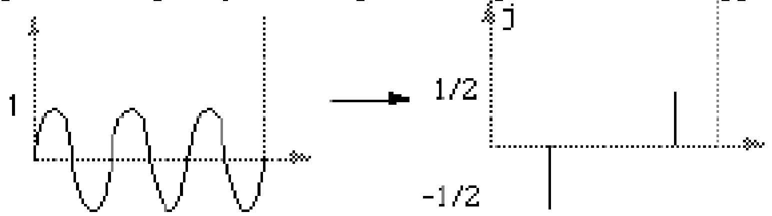


- The transform of a cosine function is a positive delta at the appropriate positive and negative frequency.

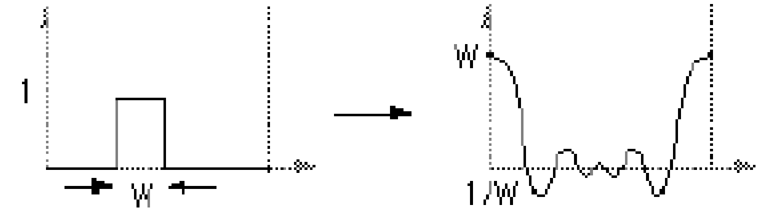


The One-Dimensional Fourier Transform Some Examples

- The transform of a sin function is a negative complex delta function at the appropriate positive frequency and a negative complex delta at the appropriate negative frequency.



- The transform of a square pulse is a sinc function.



- The two-dimensional Fourier transform and its inverse

  Fourier transform (discrete case) DTC

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

  $$\text{for } u=0,1,2,\ldots,M-1, v=0,1,2,\ldots,N-1$$

  - Inverse Fourier transform:

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

    $$\text{for } x=0,1,2,\ldots,M-1, y=0,1,2,\ldots,N-1$$

    - u, v : the transform or frequency variables
    - x,y : the spatial or image variables

- We define the Fourier spectrum, phase anble, and power spectrum as follows:

  $$|F(u,v)|=\left[R^2(u,v)+I^2(u,v)\right]^{\frac{1}{2}} \quad \text{(spectrum)}$$

  $$\phi(u,v)=\tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \quad \text{(phase angle)}$$

  $$P(u,v)=|F(u,v)|^2=R^2(u,v)+I^2(u,v) \quad \text{(power spectrum)}$$

  - $R(u,v)$: the real part of $F(u,v)$
  - $I(u,v)$: the imaginary part of $F(u,v)$

- Some properties of Fourier transform:

$$\Im\left[f(x,y)(-1)^{x+y}\right] = F\left(u-\frac{M}{2}, v-\frac{N}{2}\right) \text{ (shift)}$$

$$F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \qquad \text{(average)}$$

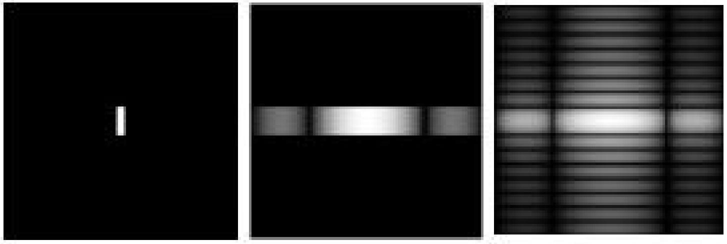
$$F(u,v) = F^*(-u,-v) \qquad \text{(conjugate symmetric)}$$

$$|F(u,v)| = |F(-u,-v)| \qquad \text{(symmetric)}$$

The Two-Dimensional DFT and Its Inverse

The 2D DFT **F(u,v)** can be obtained by

1. taking the 1D DFT of every row of image **f(x,y)**, **F(u,y)**,
2. taking the 1D DFT of every column of **F(u,y)**

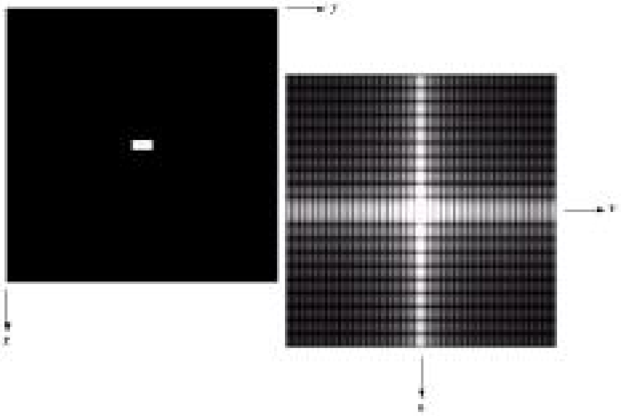


(a)f(x,y) (b)F(u,y) (c)F(u,v)

The Two-Dimensional DFT and Its Inverse

a b

FIGURE 4.3 (a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



Fast Fourier Transformation

□ A 2D Fourier transform

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

□ Has complexity O(N^4)

□ For a 1D Discrete F T complexity become O(N^2)

$$F(u) = \frac{1}{N}\sum_{N=0}^{N-1} f(x)\,e^{-j\frac{2\pi}{N}ux}$$

□ Where we take $W_N^{ux} = e^{-j\frac{2\pi}{N}ux}$ for simplification. We have N = 2^N no. of input and we assume N = 2M

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux}$$

□ Re-write F(u) as

$$F(u) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) W_{2M}^{ux}$$

$$F(u) = \frac{1}{2}\left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_{2M}^{u(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{u(2x+1)}\right]$$

$$F(u) = \frac{1}{2}\left[\frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_{2M}^{ux} + \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{ux} . W_{2M}^{u}\right]$$

$$F(u) = \frac{1}{2}\left[F_{even}(u) + F_{odd}(u) . W_{2M}^{u}\right]$$

□ We take

$$W_M^{u+M} = W_M^u \quad and \quad W_{2M}^{u+M} = - W_{2M}^u$$

$$F(u+M) = \frac{1}{2}\left[F_{even}(u) - F_{odd}(u) . W_{2M}^{u}\right]$$

□ Total complexity reduces to $N \log_2^N$

Discrete cosine transform.

The 1-D discrete cosine transform is defined as

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

For u = 0, 1, 2, . . , N-1.

Similarly the inverse DCT is defined as

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

For u = 0, 1, 2, . . , N-1

Where α is

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

The corresponding 2-D DCT pair is

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

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

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

For x, y= 0, 1, 2, . . , N-1

Correspondence between Filtering in the Spatial and Frequency Domain

- Convolution theorem:
  - The discrete convolution of two functions $f(x,y)$ and $h(x,y)$ of size MXN is defined as

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

  - Let $F(u,v)$ and $H(u,v)$ denote the Fourier transforms of $f(x,y)$ and $h(x,y)$, then

$$f(x,y)*h(x,y)\Leftrightarrow F(u,v)H(u,v) \quad \text{Eq. (4.2-31)}$$

$$f(x,y)h(x,y)\Leftrightarrow F(u,v)*H(u,v) \quad \text{Eq. (4.2-32)}$$

- $A\delta(x-x_0,y-y_0)$:an impulse function of strength A, located at coordinates (x_0,y_0)

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}s(x,y)A\delta(x-x_0,y-y_0)=As(x_0,y_0)$$

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}s(x,y)\delta(x,y)=s(0,0)$$

where $\delta(x,y)$: a unit impulse located at the origin

- The Fourier transform of a unit impulse at the origin (Eq. (4.2-35)) :

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

- Let $f(x,y)=\delta(x,y)$, then the convolution (Eq. (4.2-36))

$$\begin{aligned}f(x,y)*h(x,y)&=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\delta(m,n)h(x-m,y-n)\\&=\frac{1}{MN}h(x,y)\end{aligned}$$

- Combine Eqs. (4.2-35) (4.2-36) with Eq. (4.2-31), we obtain

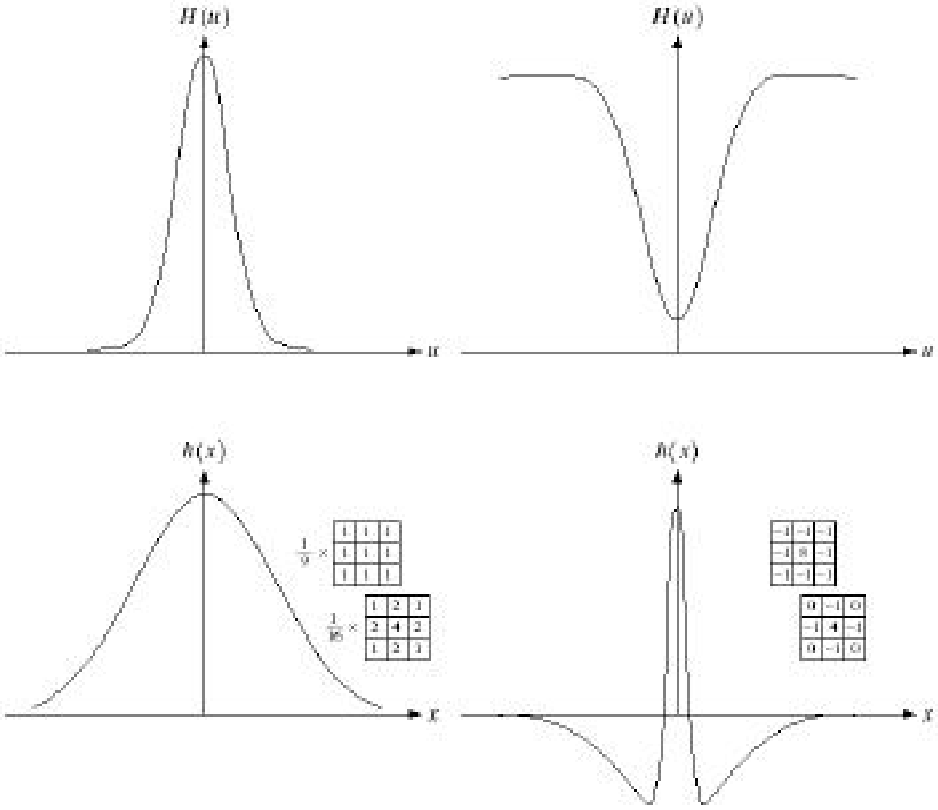
$$f(x,y)*h(x,y)\Leftrightarrow F(u,v)H(u,v)$$

$$\delta(x,y)*h(x,y)\Leftrightarrow \Im[\delta(x,y)]H(u,v)$$

$$\frac{1}{MN}h(x,y) \qquad\qquad \frac{1}{MN}H(u,v)$$

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

- Let $H(u)$ denote a frequency domain, Gaussian filter function given the equation
  $$H(u) = Ae^{u^2/2\sigma^2}$$
  where σ : the standard deviation of the Gaussian curve.
- The corresponding filter in the spatial domain is
  $$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$
- Note: Both the forward and inverse Fourier transforms of a Gaussian function are real Gaussian functions.
- One very useful property of the Gaussian function is that both it and its Fourier transform are real valued; there are no complex values associated with them.
- In addition, the values are always positive. So, if we convolve an image with a Gaussian function, there will never be any negative output values to deal with.
- There is also an important relationship between the widths of a Gaussian function and its Fourier transform. If we make the width of the function smaller, the width of the Fourier transform gets larger. This is controlled by the variance parameter ^{2} in the equations.
- These properties make the Gaussian filter very useful for lowpass filtering an image. The amount of blur is controlled by σ^2. It can be implemented in either the spatial or frequency domain.
- Other filters besides lowpass can also be implemented by using two different sized Gaussian functions.



a b
c d

FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Smoothing Frequency-Domain Filters

- The basic model for filtering in the frequency domain

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

  where $F(u,v)$: the Fourier transform of the image to be smoothed

  $H(u,v)$: a filter transfer function
- Smoothing is fundamentally a lowpass operation in the frequency domain.
- There are several standard forms of lowpass filters (LPF).
  - Ideal lowpass filter
  - Butterworth lowpass filter
  - Gaussian lowpass filter

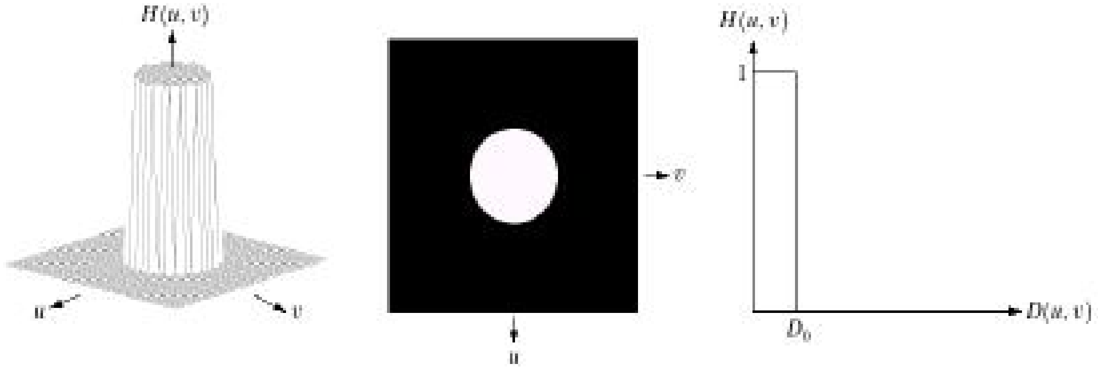
Ideal Lowpass Filters (ILPFs)

- The simplest lowpass filter is a filter that "cuts off" all high-frequency components of the Fourier transform that are at a distance greater than a specified distance D_0 from the origin of the transform.
- The transfer function of an ideal lowpass filter

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

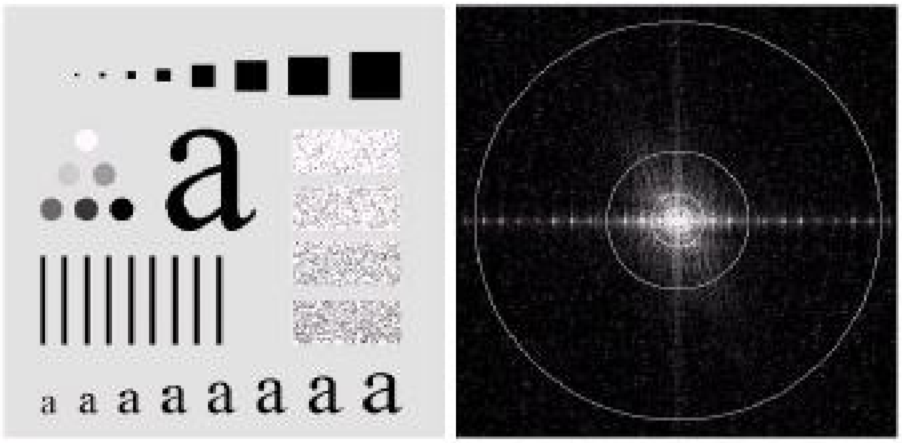
where $D(u,v)$: the distance from point (u,v) to the center of ther frequency rectangle

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



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.



a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

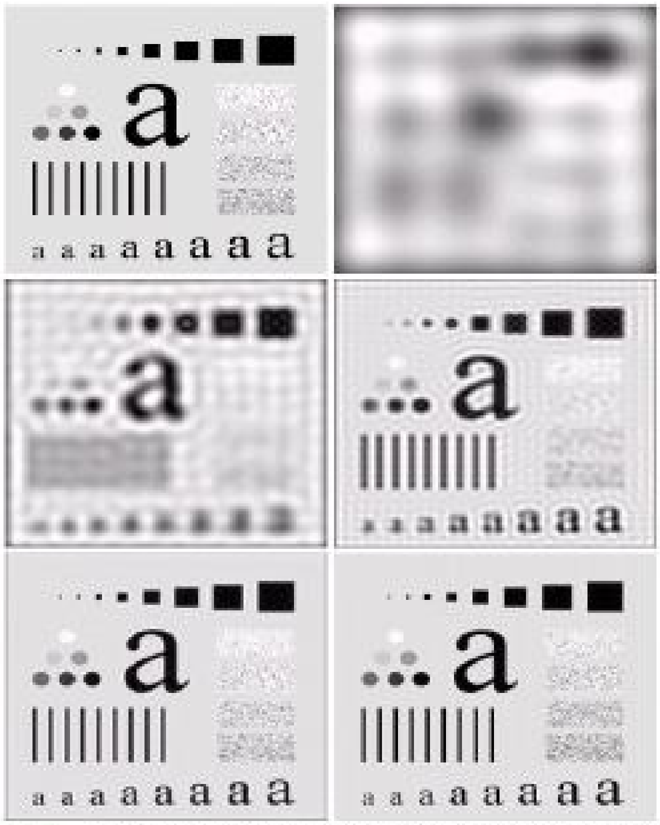


FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Ideal Lowpass Filters Another Example

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

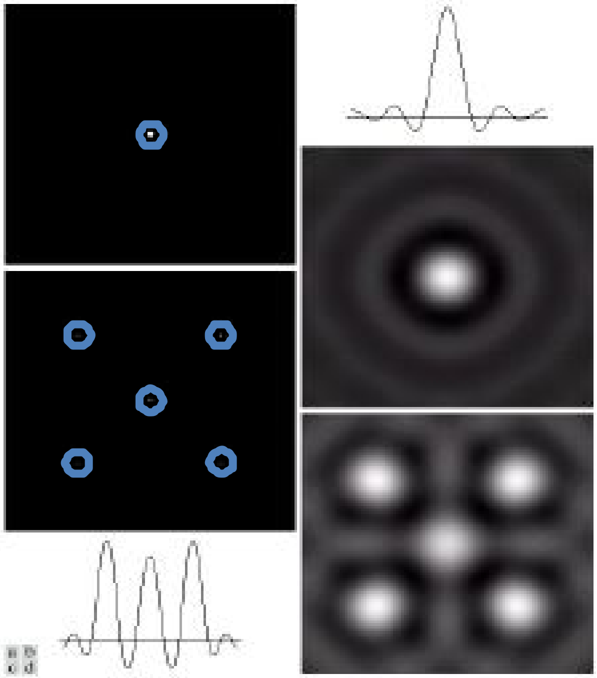
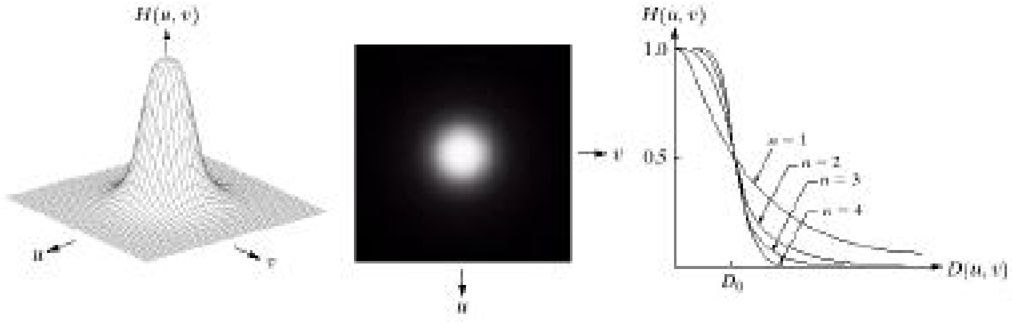


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

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

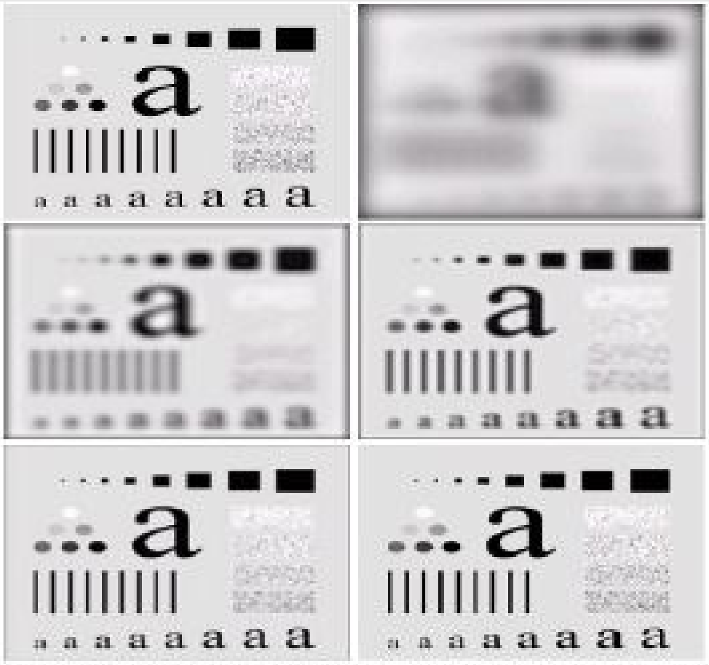
Butterworth Lowpass Filters (BLPFs) With order n

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



a b c

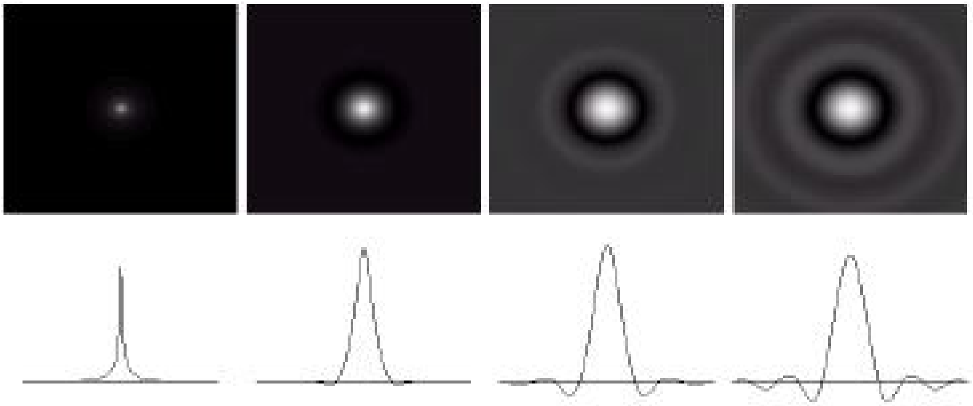
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Spatial Representation

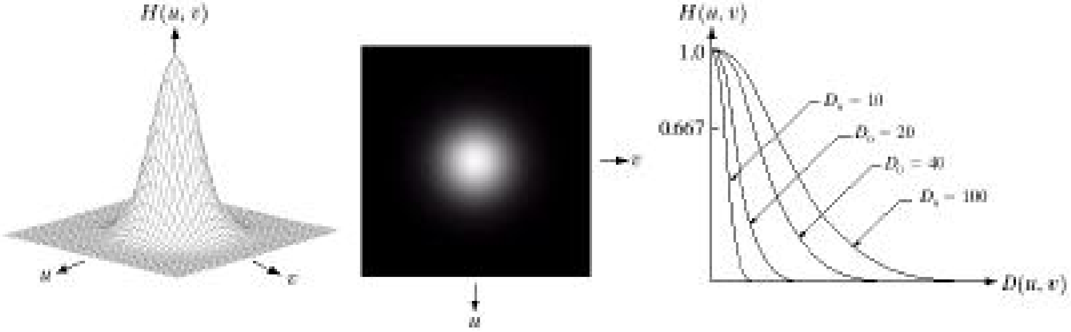


a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Gaussian Lowpass Filters (FLPFs)

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



a b c

FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

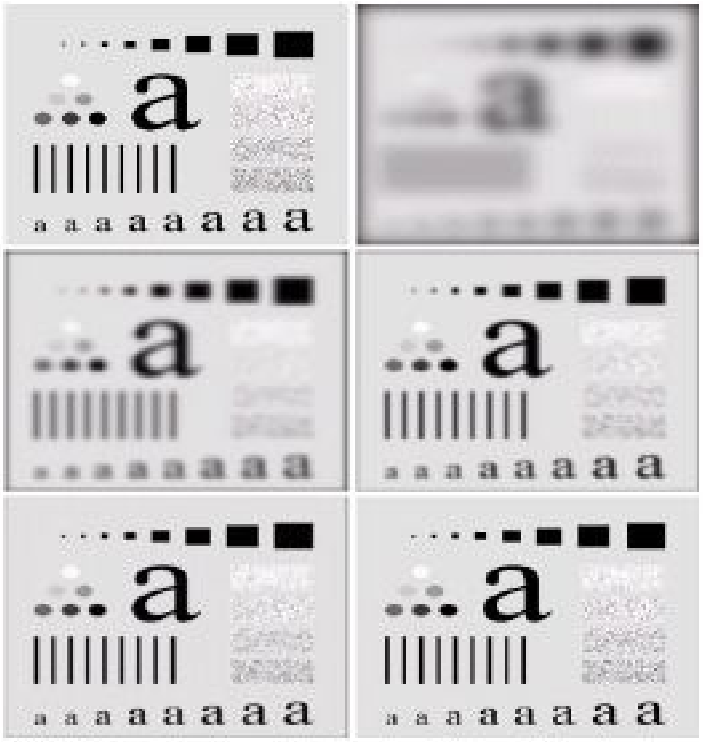


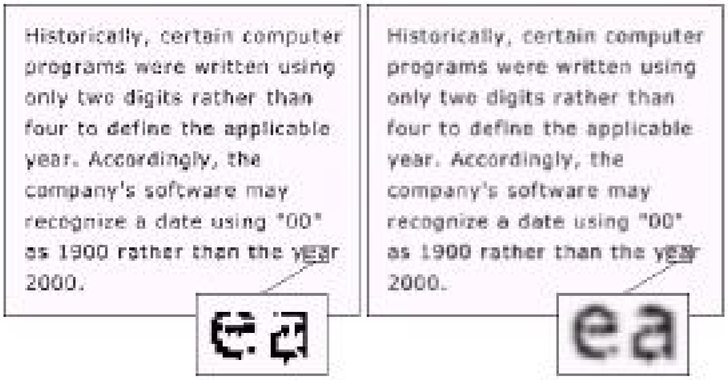
FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

a b
c d
e f

Additional Examples of Lowpass Filtering

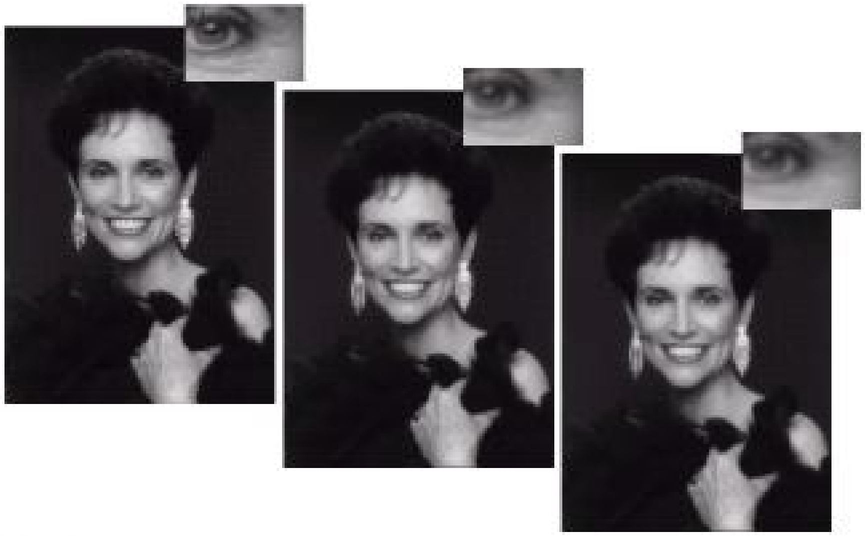
a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).



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

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



a b c

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

Sharpening Frequency Domain Filter

$$H_{hp}(u,v) = H_{lp}(u,v)$$

Ideal highpass filter

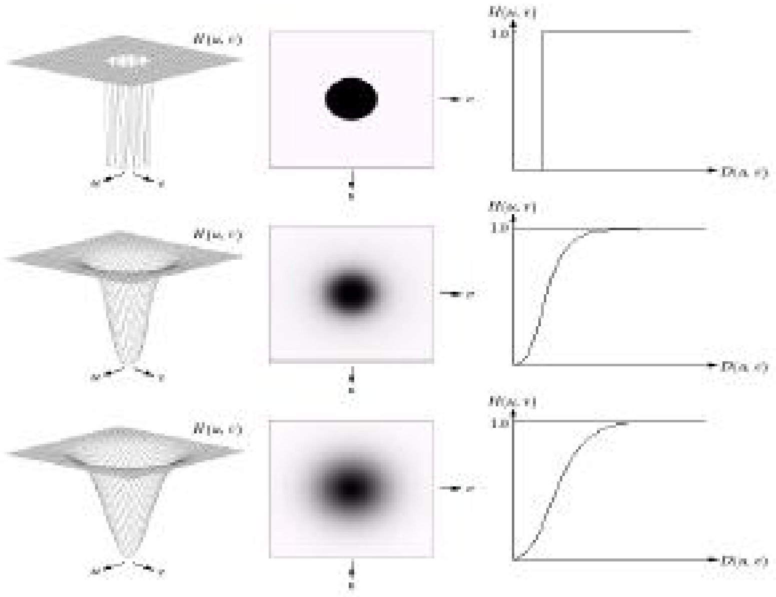
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Butterworth highpass filter

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

Gaussian highpass filter

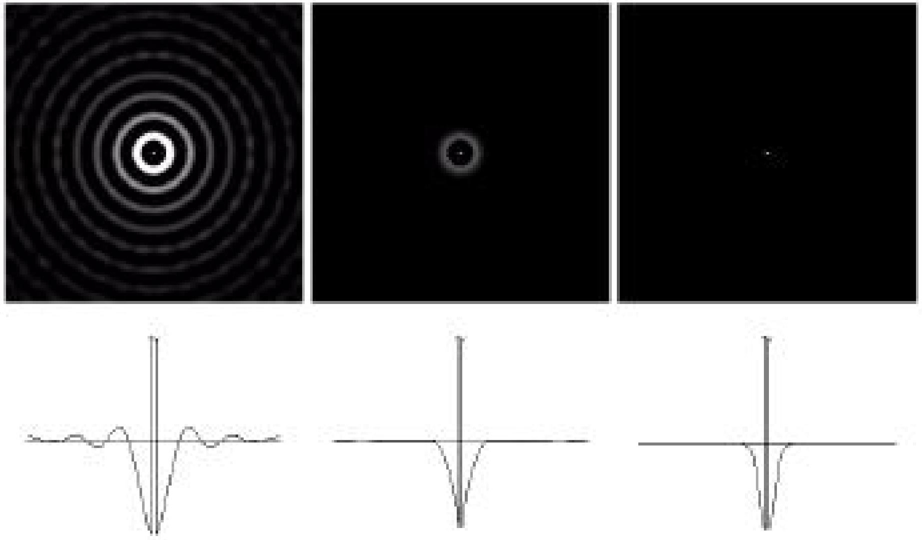
$$H(u,v) = 1 - e^{D^2(u,v)/2D_0^2}$$



a b c
d e f
g h i

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

High pass Filters Spatial Representations



a b c

FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

Ideal Highpass Filters

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

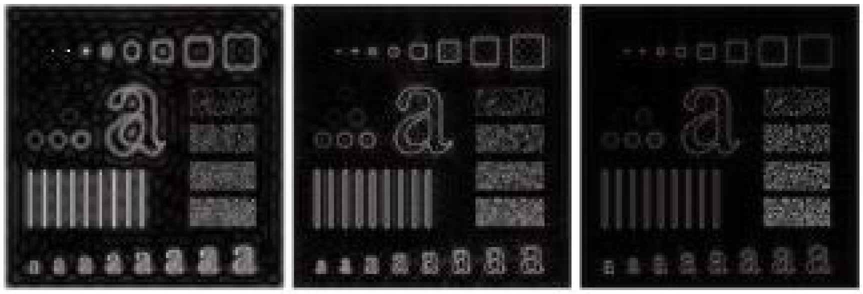


FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Butterworth Highpass Filters

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

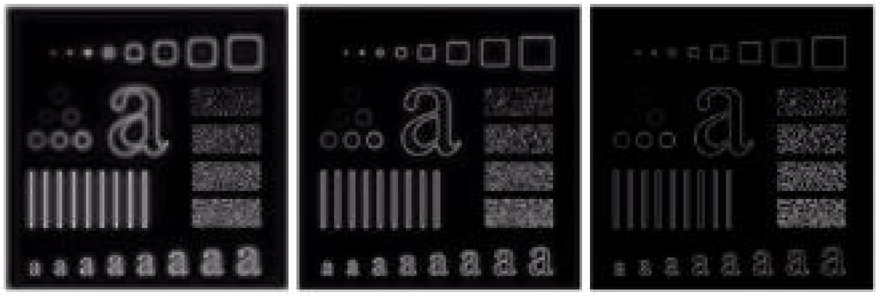


FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

Gaussian Highpass Filters

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

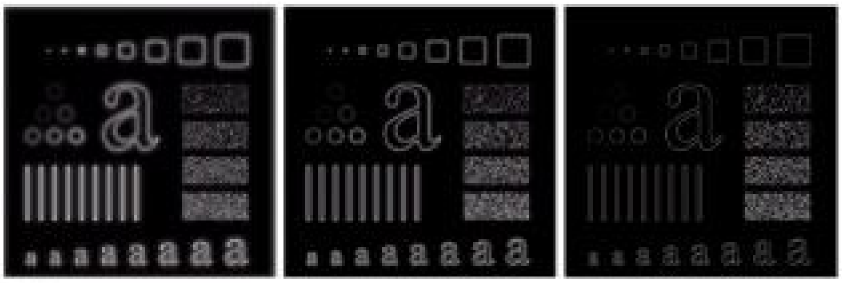


FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

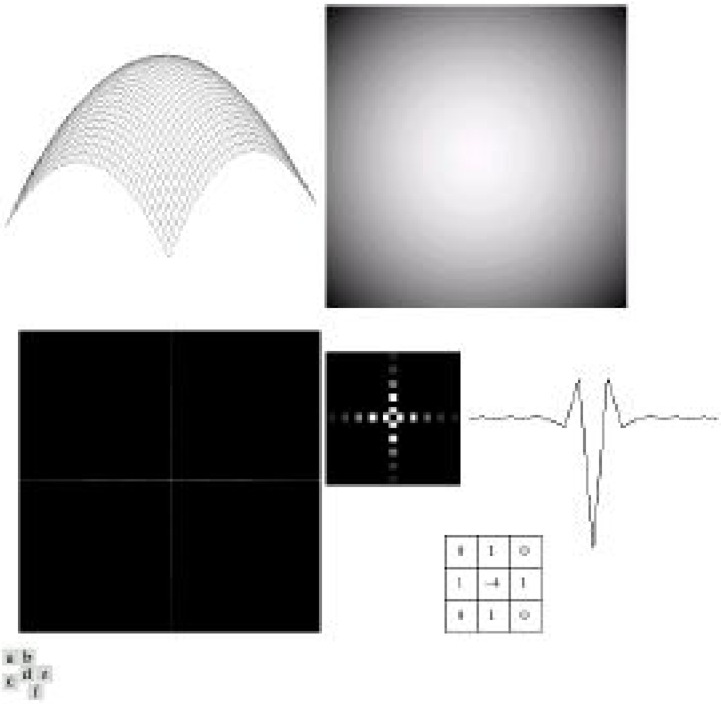
The Laplacian in the Frequency Domain

- The Laplacian filter

  $$H(u,v) = -(u^2+v^2)$$

- Shift the center:

  $$H(u,v) = -\left[(u-\frac{M}{2})^2+(v-\frac{N}{2})^2\right]$$



0	1	0
1	-4	1
0	1	0

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

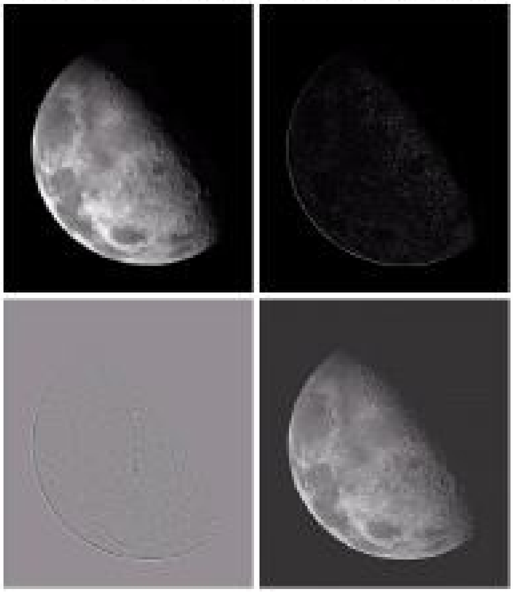


FIGURE 4.28 (a) Image of the North Pole of the moon. (b) Laplacian filtered image. (c) Laplacian image scaled. (d) Image enhanced by using Eq. (4.4-12). (Original image courtesy of NASA.)

Homomorphic filtering:

The illumination-reflectance model can be used to develop a frequency domain procedure forimproving the appearance of an image by simultaneous gray-level range compression andcontrast enhancement. An image f(x, y) can be expressed as the product of illumination andreflectance components:

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

$$\Im\{f(x, y)\} \neq \Im\{i(x, y)\}\Im\{r(x, y)\}.$$

Suppose we define,

$$z(x, y) = \ln f(x, y)$$
$$= \ln i(x, y) + \ln r(x, y).$$

Then

$$\Im\{z(x, y)\} = \Im\{\ln f(x, y)\}$$
$$= \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\}$$ or

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

Where $F_i(u, v)$ *and* $F_r(u, v)$ *are fourier transforms*

If we $Z(u, v)$ process by filter functions $H(u, v)$

$$S(u, v) = H(u, v)Z(u, v)$$
$$= H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

Where $S(u, v)$ *is fourier transform of result*

$$s(x, y) = \Im^{-1}\{S(u, v)\}$$
$$= \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\}.$$

$$i'(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\}$$
$$r'(x, y) = \Im^{-1}\{H(u, v)F_r(u, v)\},$$
$$s(x, y) = i'(x, y) + r'(x, y).$$
$$g(x, y) = e^{s(x, y)}$$
$$= e^{i'(x, y)} \cdot e^{r'(x, y)}$$
$$= i_0(x, y)r_0(x, y)$$

Where

$$i_0(x, y) = e^{i'(x, y)}$$

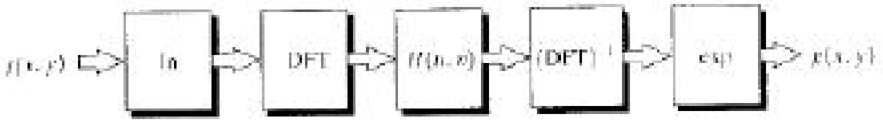


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

are the illumination and reflectance components of the output image. The enhancement approachusing the foregoing concepts is summarized in Fig. 9.1. This method is based on a special case ofa class of systems known as homomorphic systems. In this particular application, the key to theapproach is the separation of the illumination and reflectance components achieved. Thehomomorphic filter function H (u, v) can then operate on these components separately.

The illumination component of an image generally is characterized by slowspatial variations, while the reflectance component tends to vary abruptly, particularly at thejunctions of dissimilar objects. These characteristics lead to associating the low frequencies ofthe Fourier transform of the logarithm of an image with illumination and the high frequencieswith reflectance. Although these associations are rough approximations, they can be used toadvantage in image enhancement.

A good deal of control can be gained over the illumination andreflectance components with a homomorphic filter. This control requires specification of a filterfunction H (u, v) that affects the low- and high-frequency components of the Fourier transform indifferent ways. Figure 9.2 shows a cross section of such a filter. If the parameters γLand γHarechosen so that γL< 1 and γH> 1, the filter function shown in Fig. 9.2 tends to decrease thecontribution made by the low frequencies (illumination) and amplify the contribution made byhigh frequencies (reflectance). The net result is simultaneous dynamic range compression andcontrast enhancement.

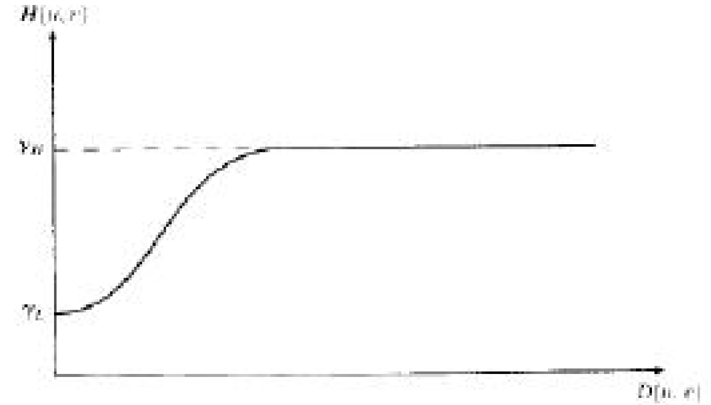


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

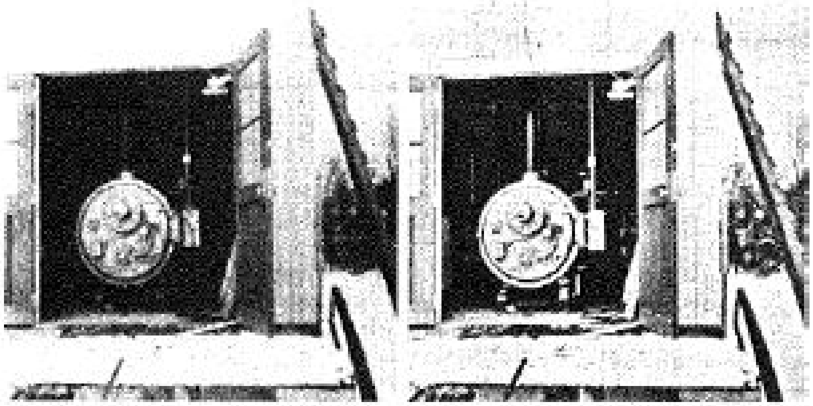


FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)

Unit-III

Image Segmentation: Detection of Discontinuities – Edge Operators – Edge Linking and Boundary Detection – Thresholding – Region Based Segmentation – Morphological Watersheds – Motion Segmentation.

Introduction

1) *Definition:*

- Segmentation is the process of subdividing an image into its constituent regions or objects.
- Its purpose is to further analyze the constituent parts in the image in detail once they are identified.
- Segmentation should stop when the objects of interest in an application have been isolated.
- Eg: One of the applications of Segmentation is to identify Tumor region in Brain MRI image.
- Segmentation algorithms generally are based on one of two basic properties of intensity values: **discontinuity** and **similarity**.
    - The approach based on the discontinuity is to partition an image based on abrupt changes in intensity (gray levels), e.g. edges in the image.
    - The approach based on the similarity is to partition an image into regions that are similar according to a set of predefined criteria, e.g. thresholding, region growing, region splitting and merging.

2) *Detection of discontinuities:*

- 3 basic types of gray-level discontinuities:
    - Points
    - Lines
    - Edges
- Common method of detection:
    - run a mask through the image. general 3 x 3 mask

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig 1. A

A. Point Detection:

- T: non-negative threshold $|R \geq T$
-

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

-1	-1	-1
-1	8	-1
-1	-1	-1

Fig 2. Mask for Point processing

- A point has been detected at the location on which the mask is centered if: $|R|>T$
- The gray level of an isolated point will be quite different from the gray levels of its neighbors
    - measure the weighted differences between the center point and its neighbors

B. Line Detection:

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal
R_1

-1	-1	2
-1	2	-1
2	-1	-1

+45°
R_2

-1	2	-1
-1	2	-1
-1	2	-1

Vertical
R_3

2	-1	-1
-1	2	-1
-1	-1	2

-45°
R_4

- If at a certain point $|R_i|>|R_j|$, this point is more likely associated with a line in the direction of mask *i*.

C. Edge Detection:

- Edge pixels are pixels at which the intensity of an image function abruptly changes and edges or edge segments are the set of connected edge pixels.
- Edge models: There are three basic edge models, namely the **step edge**, the **ramp edge** and the **roof edge**.
    - The **step edge** is a transition between two intensity levels occurring ideally over a distance of 1 pixel.
    - The **ramp edge** is a transition between two intensity levels occurring gradually over a distance of several pixels due to blurring. The slope of the ramp (and hence the width of the edge) is directly proportional to the degree of blurring.
    - The **roof edge** is a model of a line between two regions. The width of the edge is determined by the thickness and sharpness of the line.

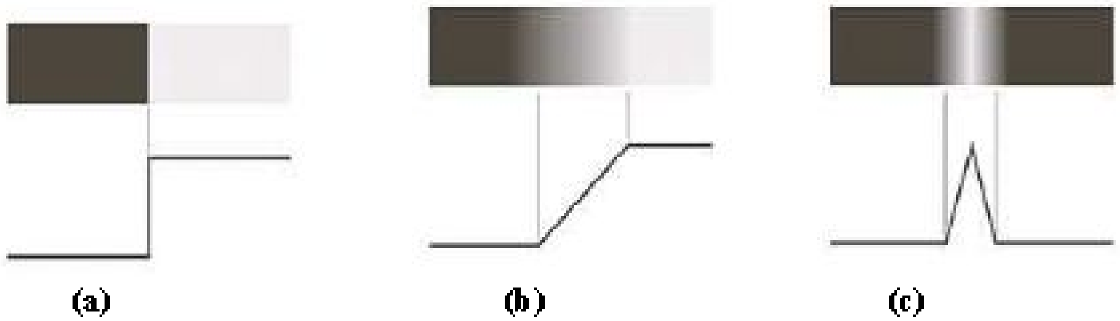
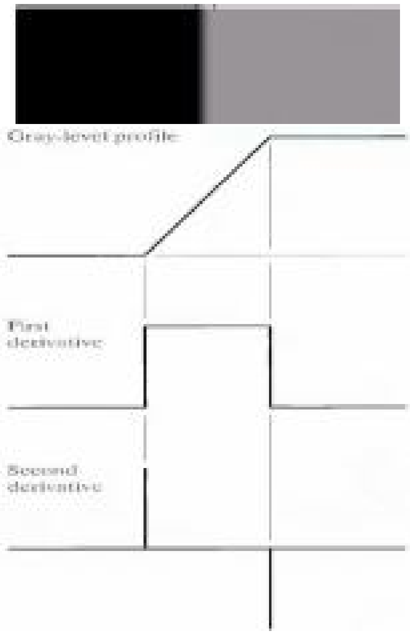


Fig 3. Models (top) and Gray-level profiles (bottom) of (a) step, (b) ramp and (c) roof edges

- The common approach is to run a mask approximating either the first order derivative (Gradient operator) or the second order derivative (Laplacian operator).
    - The magnitude of the first order derivative (Gradient) is used to determine whether a point is on the ramp.
    - The sign of the second order derivative (Laplacian) is used to determine whether an edge pixel is either on the dark (left) side of the edge or on the light (right) side of the edge.



3) Edge Operators:

- *Gradient operator:* The gradient of a function (image), f(x,y) is defined as the vector

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

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

where Gx is the gradient along the x-direction and Gy is the gradient along the y-direction.

- The magnitude ∇f | & phase Θ of the gradient are

$$\nabla f = \sqrt{G_x^2 + G_y^2}$$

z_5	z_6
z_8	z_9

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

- Gradient is a non-linear operator.
- The magnitude of the gradient is often approximated either using the difference along x- and y-directions or using the cross differences along the diagonals as

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

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

- The pair of 2 x 2 masks, known as the **Roberts Cross Gradient Operators,** using the cross differences along the diagonals are shown below.

1	0
0	-1

0	1
-1	0

Roberts Cross Gradient Operators

- Using a 2 x 2 mask has a practical difficulty as it does not have a center. Hence masks of size 3 x 3 are often preferred using either the differences along the x- and y-directions or using the cross differences along the diagonals as

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

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- The two pairs of3 x 3 masks, known as the **Prewitt** and **Sobel Operators**, using the differences along x- and y-directions to detect horizontal and vertical edges are shown below.

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

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

Prewitt Operators

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

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

Sobel Operators

- The two pairs of 3 x 3 masks, known as the **Prewitt** and **Sobel Operators**, using the differences along the diagonals to detect the diagonal edges are shown below.

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

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

Prewitt Operators

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

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

Sobel Operators

- *Laplacian Operator:* Laplacian, for a function (image) f(x,y), is defined as

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- The Laplacian is a linear operator.
- The discrete form of the Laplacian of f(x,y), taking the 4-neighbours into account, is obtained by summing the discrete forms of partial derivatives along x- and y- directions or taking all the 8-neighbours into account, is obtained by summing the discrete forms of partial derivatives along x- and y directions & along the diagonals as

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8)$$

- The corresponding 3 x 3 masks are

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

-1	-1	-1
-1	8	-1
-1	-1	-1

- The Laplacian generally is not used in its original form for edge detection for the following reasons: (i) Second-order derivative is unacceptably sensitive to noise and (ii) The magnitude of the Laplacian produces double edges.
- The Laplacian is often used with Gaussian smoother given by

$$h(x,y) = -\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

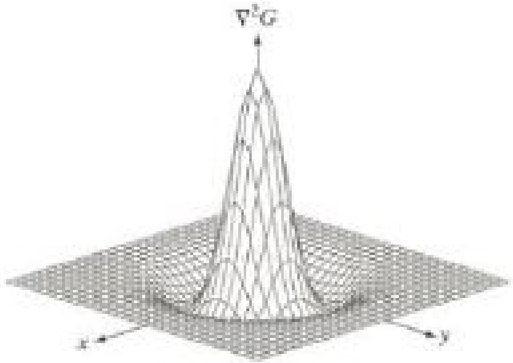
where is the standard deviation which determines the degree of blurring that occurs.

- The Laplacian of h is given by

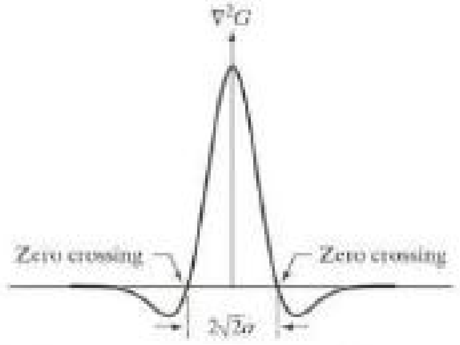
$$\nabla^2 h = -\left(\frac{r^2 - \sigma^2}{\sigma^4}\right)\exp\left(-\frac{r^2}{2\sigma^2}\right)$$

where $r^2 = x^2 + y^2$.

- Here, $\nabla^2 h$ is called the **Laplacian of Gaussian (LoG).**
- The Laplacian of Gaussian is sometimes called the *Mexican Hat function* because of its appearance.



LoG: 3D plot



LoG: Cross section showing zero-crossings

- To implement LoG, the image is convolved with the h and the result is then Laplacianed or the following mask which approximates the LoG is used.

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

- To detect the edges the resulting image is thresholded (setting all its positive values to white and negative values to black) and the zero-crossings between these whites and blacks are found.

4) *Edge Linking and boundary detection:*

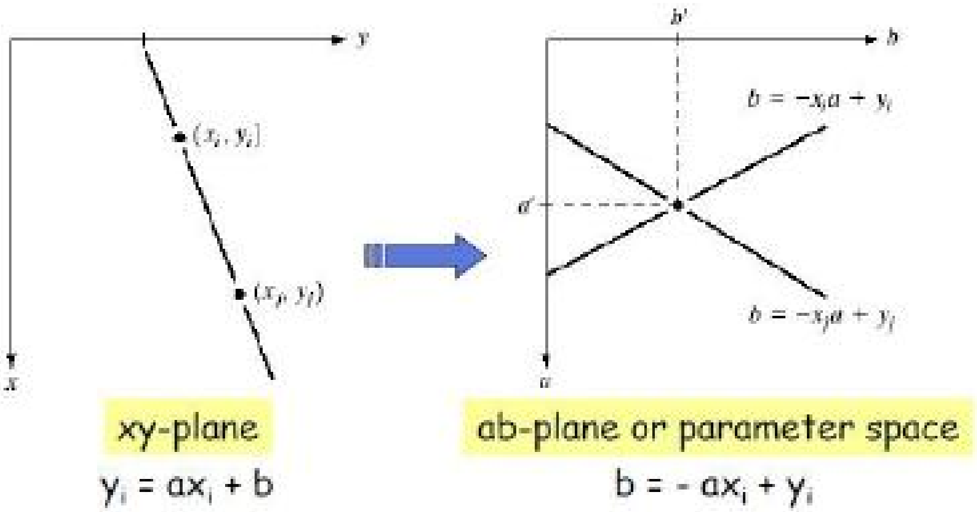
- Edge detection algorithms are followed by linking procedures to assemble edge pixels into meaningful edges.
- Basic approaches
  - Local Processing
  - Global Processing via the Hough Transform
  - Global Processing via Graph-Theoretic Techniques

A *Local processing:*

- Analyze pixels in small neighbourhood $S_{x,y}$ of each edge point
- Pixels that are similar are linked
- Principal properties used for establishing similarity:
    - o $M(x, y) = |\nabla f(x, y)|$: Magnitude of gradient vector
    - o $\alpha(x, y)$: Direction of gradient vector
- Edge pixel with coordinates (s, t) in $S_{x,y}$ is similar in magnitude to pixel at (x, y) if
  $$M(s, t) - M(x, y)| < E$$
- Edge pixel with coordinates (s, t) in Sxy has an angle similar to pixel at (x, y) if
  $$|\alpha(s, t) - \alpha(x, y)| < A$$
- Edge pixel (s, t) in Sxy is linked with (x, y) if both criteria are satisfied

B. *Global Processing via the Hough Transform*

- The Hough transform is a technique which can be used to isolate features of a particular shape within an image.
- Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.*
- The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.
- Here, we develop an approach based on whether sets of pixels lie on curves of a specified shape. Once detected, these curves form the edges or region boundaries of interest.
- Given n points in an image. Suppose that we want to find subset of those points that lie on straight lines. One solution is to find all lines determined by every pair of points & then find all subsets of points that are close to particular lines. This approach involves finding $n(n-1)/2 \approx n^2$ lines & then performing $n(n(n-1))/2 \approx n^3$ comparisons of every point to all lines.



- Subdivide the parameter space ab into a finite number of accumulator cells.
- All cells are initialized to zero.
- For every (x_k,y_k) in the xy-plane, a is chosen to be each subdivision value in the allowed range and corresponding b is calculated using $b=-ax_k+y_k$ and rounded to nearest allowed range.
- If a choice of a_p results in solution b_q then we let $A(p,q) = A(p,q)+1$.
- At the end of the procedure, value Q in A(i,j) corresponds to Q points in the xy-plane lying on the line $y = a_i x+b_j$.

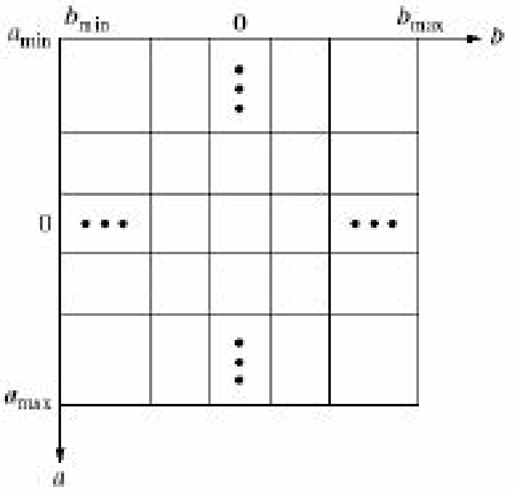


FIGURE 10.18 Subdivision of the parameter plane for use in the Hough transform.

General Steps:

1: Compute the gradient of an image and threshold it to obtain a binary image.
2: Specify subdivisions in the ab-plane.
3: Examine the counts of the accumulator cell for high pixel concentrations.
4: Examine the relationship (principally for continuity) between pixels in a chosen cell

5) *Thresholding:*

- Thresholding may be viewed as an operation that involves tests against a function T of the form

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

where f(x, y) is the gray level of point (x, y) and p(x, y) denotes some local property of this point.

- A thresholded image g(x, y) is defined as

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

- Thus, pixels labeled 1 (or any other convenient gray level) correspond to objects, whereas pixels labeled 0 (or any other gray level not assigned to objects) correspond to the background.

- **Global Thresholding:** the threshold, T depends only on the histogram of the image.
- **Local Thresholding:** the threshold, T at any point, (x,y) depends on the property of neighbourhood around that point.
- **Adaptive Thresholding:** the threshold, T at any point, (x,y) depends on both the neighbourhood property and the spatial coordinates x and y.
- **Multilevel Thresholding:** Here, multilevel thresholding classifies a point (x, y) as belonging to one object class if T1 < f(x, y) < T2, to the other object class if f(x, y) > T2, and to the background if f(x, y) < T1.

A. Basic Global Thresholding:

- A global threshold can be obtained either by a visual inspection of the histogram of the image or automatically from the histogram via a simple algorithm.
- Simple algorithm to automatically obtain a global threshold from the histogram of the image as follows:

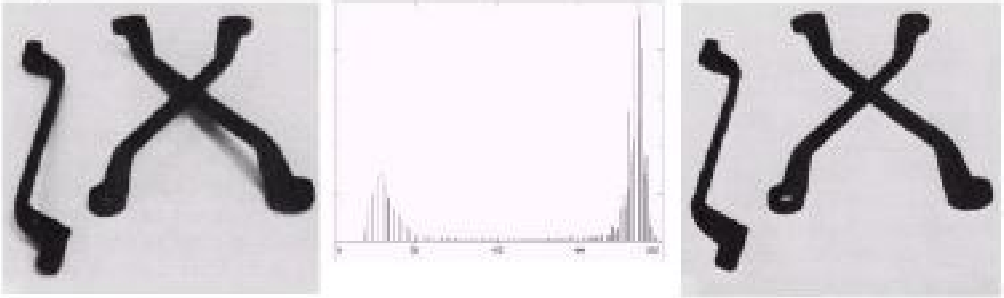
1: Select an initial estimate for T.

2: Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values > T and G_2 consisting of pixels with gray level values $\leq$ T

3: Compute the average gray level values m_1 and m_2 for pixels in regions G_1 and G_2

4: Compute a new threshold value $T = 0.5\ (m_1 + m_2)$

5: Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.



Original Image

Image histogram

Result of global thresholding with T midway between the max and min gray levels

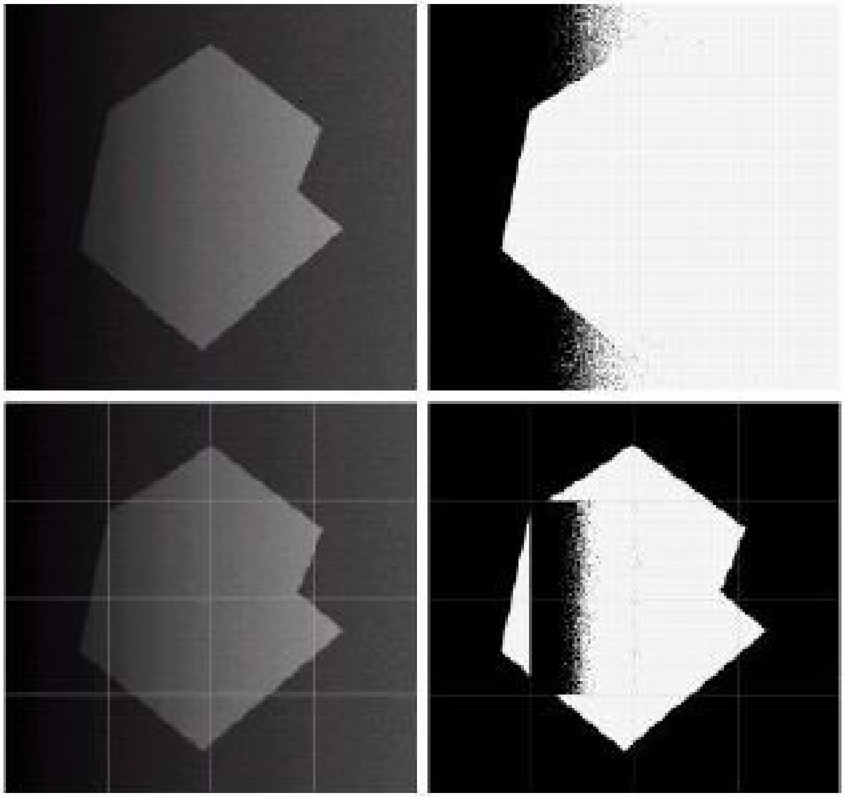
Note: Use of T midway between min and max gray levels produce binary image as shown above.

B. Basic Adaptive Thresholding:

- In an image where a fixed global threshold does not give acceptable results, e.g., an image with poor illumination, a basic adaptive thresholding technique can be used as explained below.

  - The image is divided into smaller sub-images.
  - Individual thresholds are chosen to segment each sub-image.

- The thresholds thus selected are "adaptive" to the pixel values in individual sub images.
- The improper subdivision can result in poor segmentation. Further subdivision of the improperly segmented sub image and subsequent adaptive thresholding can improve the process of segmentation.



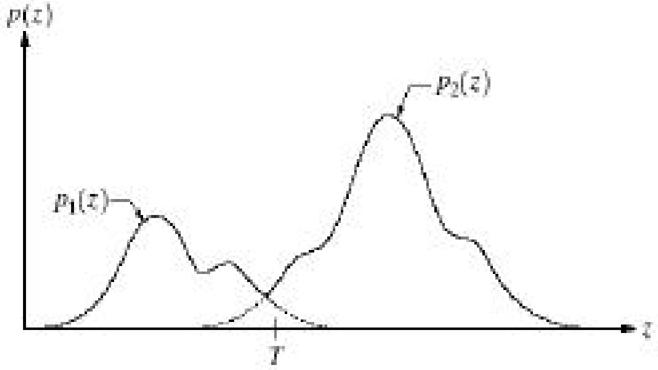
a b
c d

FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

C. *Optimal global and adaptive thresholding:*

- Consider an image with only two principal gray level regions as shown below.

FIGURE 10.32 Gray-level probability density functions of two regions in an image.



- Assume that the larger of the two PDFs corresponds to the background level while the smaller one describes the gray levels of objects in the image.
- The mixture probability density function describing the overall gray-level variation in the image is

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad \& \quad P_1 + P_2 = 1$$

- P1 is the probability (a number) that a pixel is an object pixel.

- Similarly, P2 is the probability that the pixel is a background pixel.
- The image is segmented by classifying all pixels with gray levels greater than a threshold T as background. All other pixels are called object pixels.
- The probability of erroneously classifying a background point as an object point is

$$E_1(T) = \int_{-\infty}^{T} p_2(z)dz$$

- Similarly, probability of erroneously classifying an object point as background is

$$E_2(T) = \int_{T}^{\infty} p_1(z)dz$$

- Then the overall probability of error is

$$E(T) = P_2E_1(T) + P_1E_2(T)$$

- To find the threshold value for which this error is minimal requires differentiating E(T) with respect to T (using Leibniz's rule) and equating the result to 0. The result is

$$P_1p_1(T) = P_2p_2(T)$$

- The above equation is solved for T to obtain an optimum threshold value.
- Note that if $P_1 = P_2$, then the optimum threshold is where the curves for $P_1(z)$ and $p_2(z)$ intersect.
- Obtaining an analytical expression for T requires that the equations for the two PDFs are known.

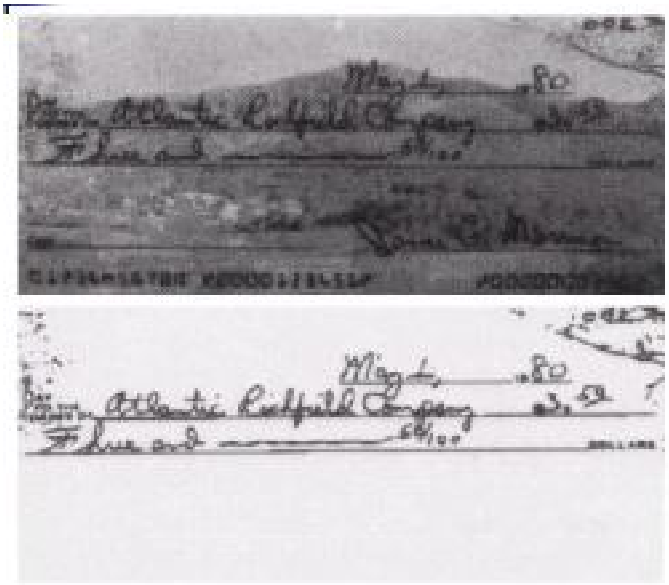
D. Local Thresholding:

- One approach for improving the shape of histograms is to consider only those pixels that lie on or near the edges between objects and the background.
- In this, the histograms are less dependent on the relative sizes of the object and the background.
- This however requires that the edges between the object and the background are known.
- The identification of whether the pixels are on the edges is done using the gradient.
- The identification of whether the pixels are on the left or right side of the edges is done using the Laplacian.
- These two quantities may be used to form a three-level image, as follows:

$$s(x,y) = \begin{cases} 0, & \text{if } \nabla f < T \\ +, & \text{if } \nabla f \geq T \text{and} \nabla^2 f \geq 0 \\ -, & \text{if } \nabla f \geq T \text{and} \nabla^2 f < 0 \end{cases}$$

where
all pixels that are not on an edge are labeled 0
all pixels that are on the dark (left) side of an edge are labeled +
all pixels that are on the light (right) side an edge are labeled –

- A transition from – to + indicates the transition from a light background to a dark object.
- A 0 or + indicates the interior of the object.
- A transition from + to - indicates the transition from a dark object to a light background.



(a) Original image
(b) Image segmented by local thresholding

6) *Region based Segmentation:*

- Let R represent the entire image region. Then, segmentation is a process that partitions R into n sub regions, R_1, R_2, ..., R_n, such that

$$
\begin{aligned}
&(a)\quad \bigcup_{i=1}^{n} R_i = R \\
&(b)\quad R_i \text{ is a connected region, } i = 1, 2, \ldots, n \\
&(c)\quad R_i \cap R_j = \phi \text{ for all i and j, } i \neq j \\
&(d)\quad P(R_i) = TRUE \text{ for } i = 1, 2, \ldots, n \\
&(e)\quad P(R_i \cup R_j) = FALSE \;\; for\ i \neq j
\end{aligned}
$$

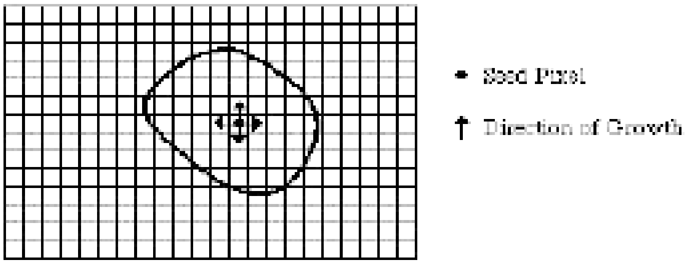
- Condition (a) indicates that the segmentation must be complete; that is, every pixel must be in a region.
- Condition (b) requires that points in a region must be connected in some predefined sense.
- Condition (c) indicates that the regions must be disjoint.
- Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region—for example $P(R_i)$ = TRUE if all pixels in R_i, have the same graylevel.
- Condition (e) indicates that regions R_i, and R_j are different in the sense of predicate P.

A. *Region growing:*

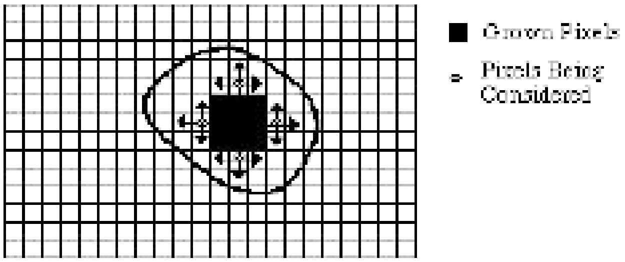
- Region growing groups the pixels or sub-regions into larger regions based on predefined criteria.
- Neighbouring pixels are examined and added to a region class if no edges are found.
- This process is iterated for each boundary pixel in the region.

  - Start with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (specified ranges of gray level or colour).
  - Seed points are found out by computing the same set of properties for every pixel values.
  - The resultant segment is removed from the process.
  - New seed is chosen from the remaining pixels.
  - This continues until all pixels have been allocated to a segment.

- Region-growing algorithms vary depending on the criteria used to decide whether a pixel should be included in the region or not, the connectivity type used to determine neighbours and the strategy to visit neighbouring pixels.



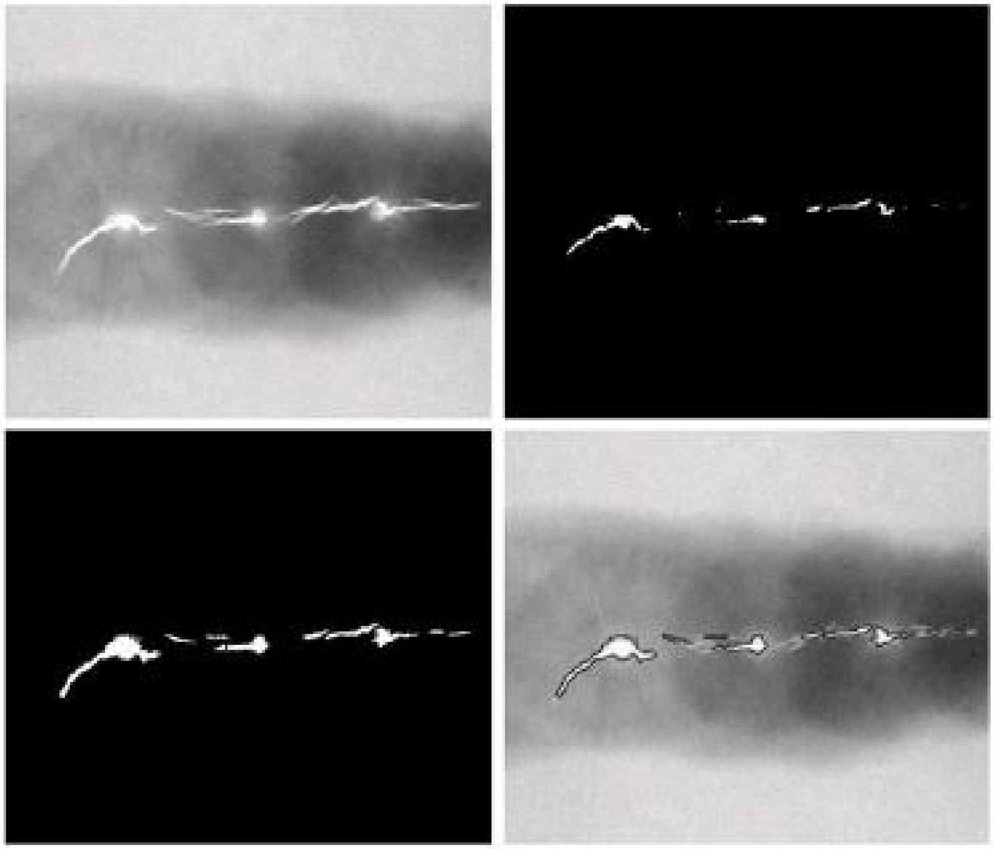
(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

a b
c d

(a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).



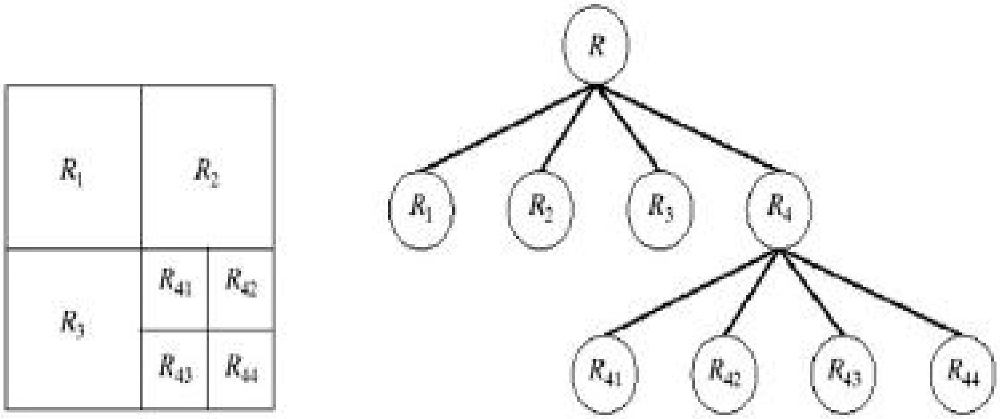
B. Region Splitting and Merging

- An alternative approach to the region growing is to subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions.

- The approach is to segment R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i, $P(R_i)$ = TRUE. If only splitting were used, the final partition likely would contain adjacent regions with identical properties. This drawback may be remedied by allowing merging, as well as splitting.

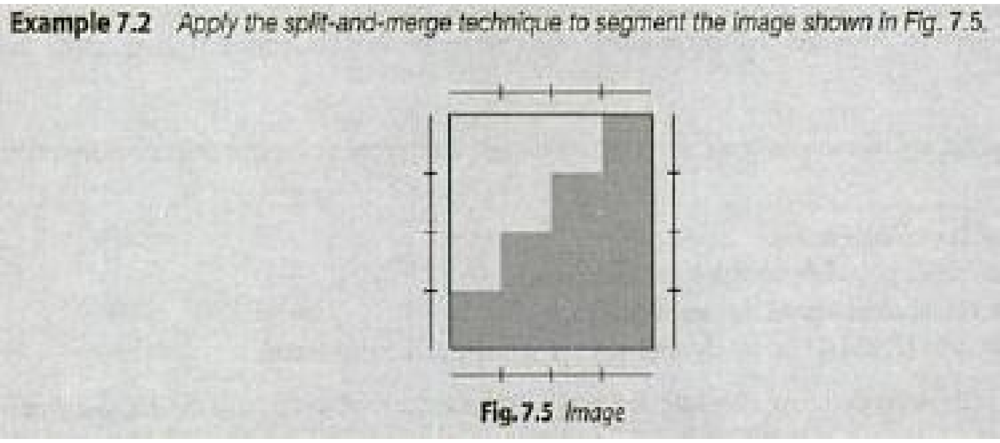
- The procedure is to start with the entire region. If P(R) = FALSE, divide the image into quadrants. If P is FALSE for any quadrant, subdivide that quadrant into subquadrants, and so on. This process results in a so-called **quad tree**.

a b



(a) Partitioned image.
(b) Corresponding quadtree.

- While splitting, two adjacent regions R_j and R_k are merged only if $P(R_j \cup R_k)$ = TRUE.
- When no further merging or splitting is possible, the procedure is stopped.



Example 7.2 *Apply the split-and-merge technique to segment the image shown in Fig. 7.5.*

Fig. 7.5 *Image*

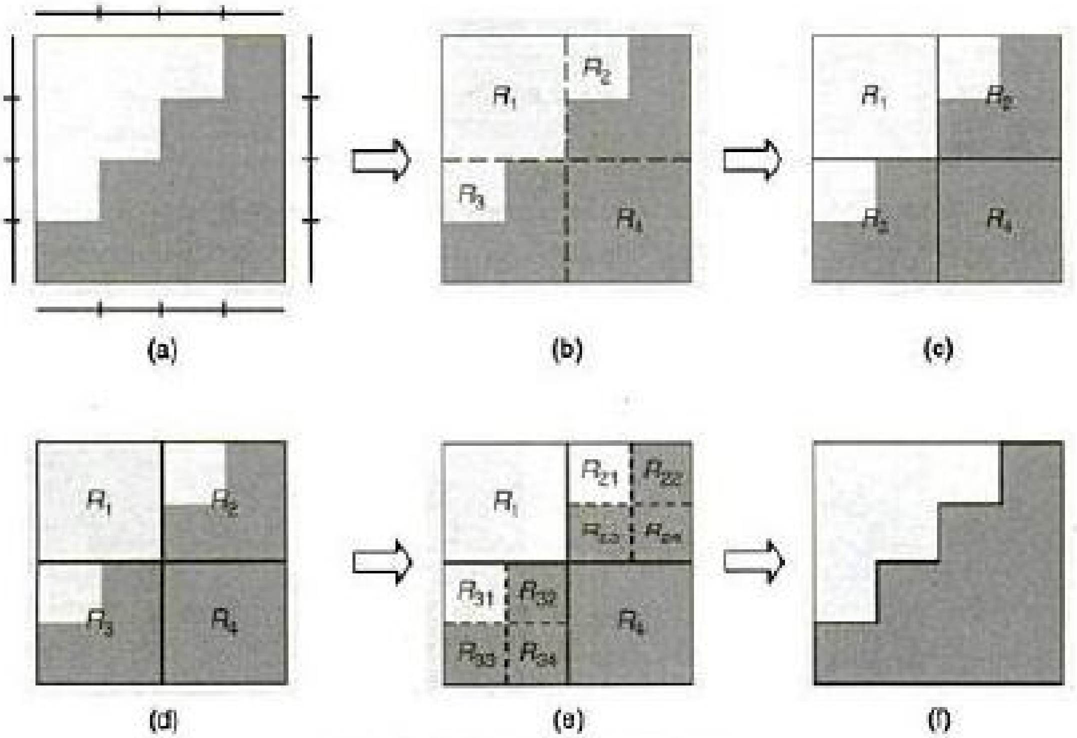


Fig. 7.5 *Split-and-merge technique*

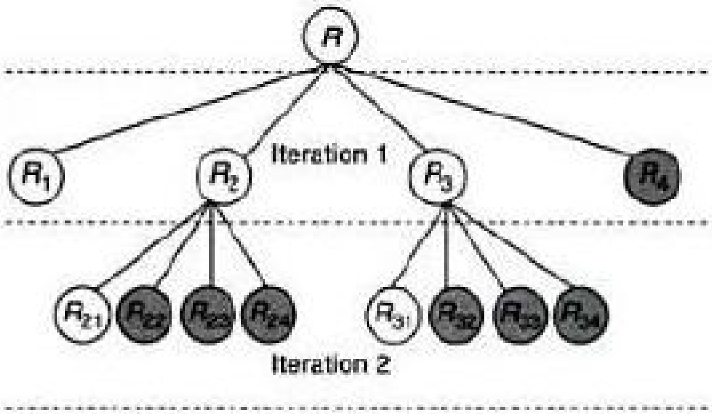
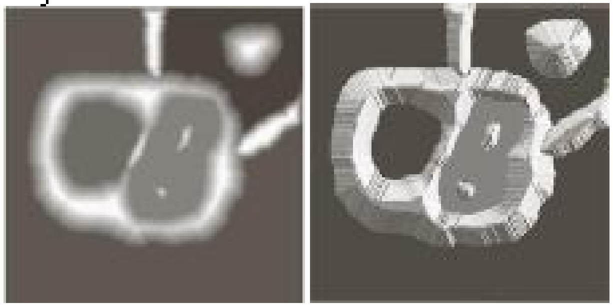


Fig. 7.6 *Quadtree representation*

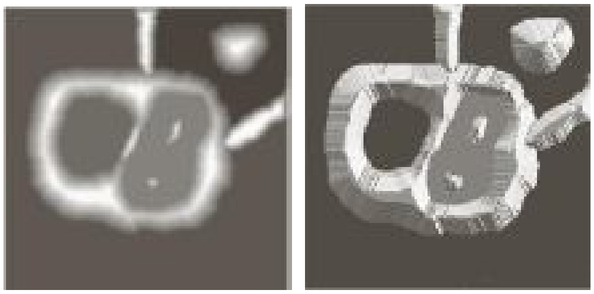
Morphological Watersheds

- Visualize an image topographically in 3D
  - The two spatial coordinates and the intensity (relief representation).
- Three types of points –

  - Points belonging to a regional minimum. –
  - Points ta which a drop of water would fall certainly to a regional minimum (catchment basin). –
  - Points at which the water would be equally likely to fall to more than one regional minimum (crest lines or watershed lines).
- Objective: find the watershed lines.



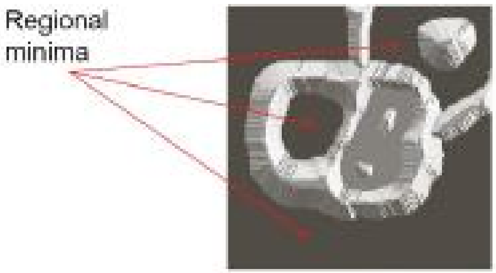
- Topographic representation.
- ▪ The height is proportional to the image intensity. ▪
- Backsides of structures are shaded for better visualization.



A hole is punched in each regional minimum and the topography is flooded by water from below through the holes.

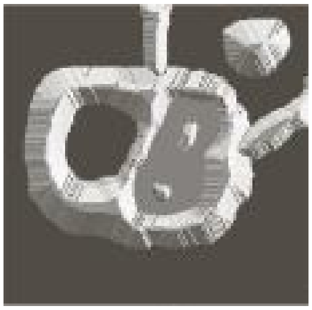
▪ When the rising water is about to merge in catchment basins, a dam is built to prevent merging. ▪ There will be a stage where only the tops of the dams will be visible.

▪ These continuous and connected boundaries are the result of the segmentation.



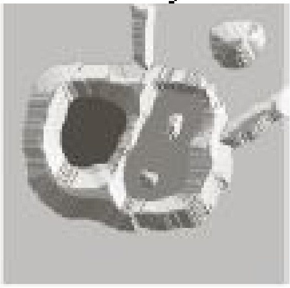
▪Topographic representation of the image.

▪A hole is punched in each regional minimum (dark areas) and the topography is flooded by water (at equal rate) from below through the holes.



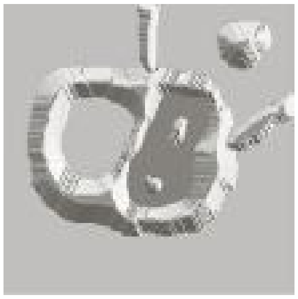
▪Before flooding.

▪To prevent water from spilling through the image borders, we consider that the image is surrounded by dams of height greater than the maximum image intensity.



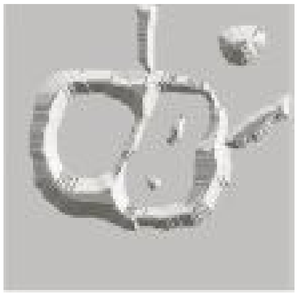
First stage of flooding.

▪The water covered areas corresponding to the dark background.



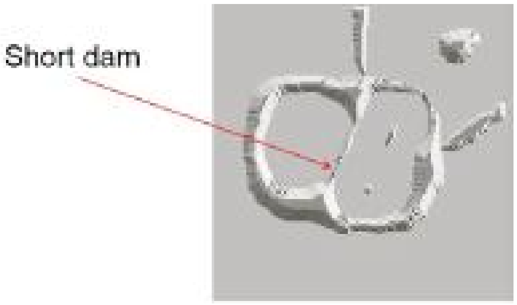
Next stages of flooding.

▪The water has risen into the other catchment basin.



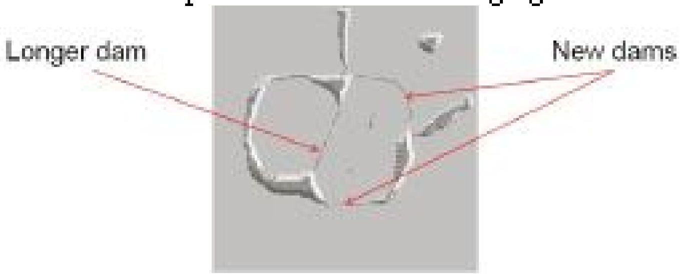
Further flooding.

The water has risen into the third catchment basin.

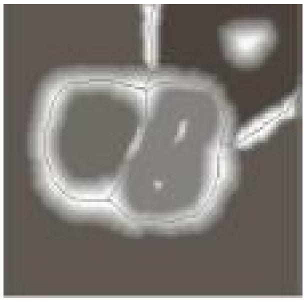


Further flooding.

▪The water from the left basin overflowed into the right basin. ▪A short dam is constructed to prevent water from merging.



▪Further flooding.
▪The effect is more pronounced.
▪The first dam is now longer.
▪ New dams are created.



The process continues until the maximum level of flooding is reached.

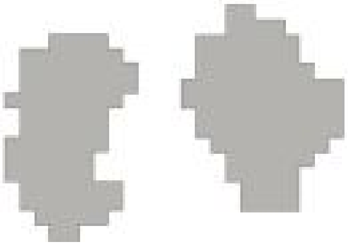
▪ The final dams correspond to the watershed lines which is the result of the segmentation.

▪ Important: continuous segment boundaries.

Dams are constructed by morphological dilation

Flooding step n-1.

Cn-1 (M1) Cn-1 (M2)

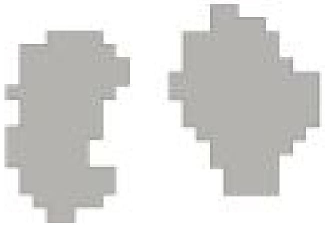


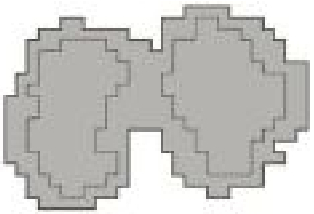
Regional minima: M1 and M2 . Catchment basins associated: Cn-1 (M1) and Cn-1 (M2).

$C[n-1]\ \ \sigma\ C_{n-1}\,C(M_1) U\ C_{n-1}(M_2)$

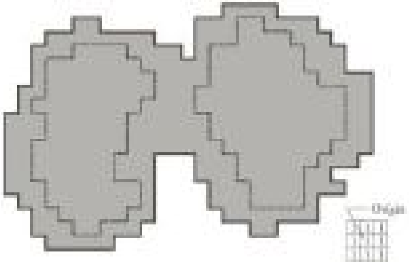
C[n-1] has two connected components

Flooding step n-1. Flooding step n.





- If we continue flooding, then we will have one connected component.
- This indicates that a dam must be constructed.
- Let q be the merged connected component if we perform flooding a step n.

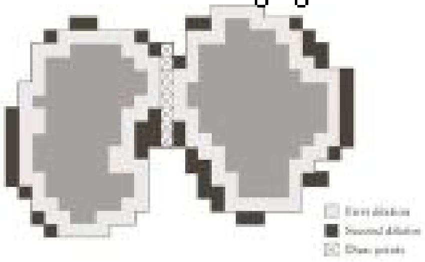


Each of the connected components is dilated by the SE shown, subject to:

1. The center of the SE has to be contained in q.
2. The dilation cannot be performed on points that would cause the sets being dilated to merge.

Conditions:

1. Center of SE in q.
2. No dilation if merging

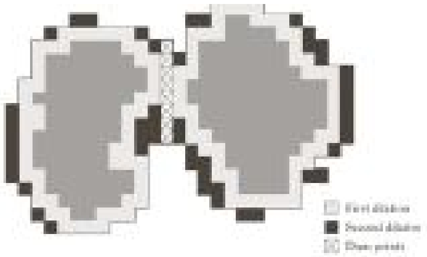


In the first dilation, condition 1 was satisfied by every point and condition 2 did not apply to any point.

- In the second dilation, several points failed condition 1 while meeting condition 2 (the points in the perimeter which is broken).

Conditions:

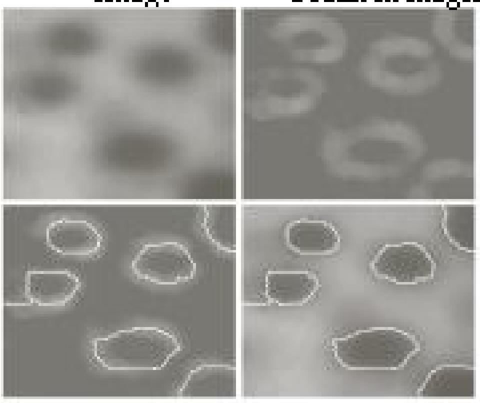
1. Center of SE in q.
2. No dilation if merging.



The only points in q that satisfied both conditions form the 1-pixel thick path.

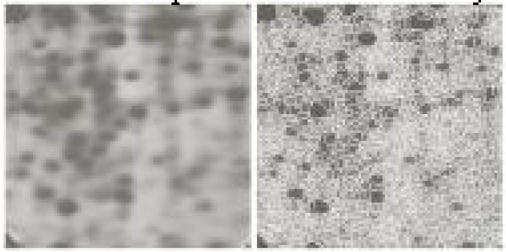
- This is the dam at step n of the flooding process.
- The points should satisfy both conditions.

Image Gradient magnitude



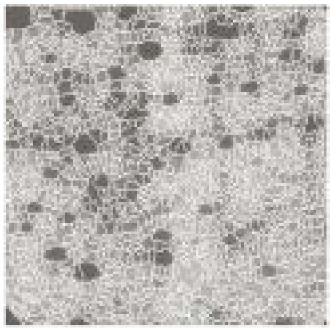
Watersheds Watersheds on the image

- A common application is the extraction of nearly uniform, bloblike objects from their background.
- For this reason it is generally applied to the gradient of the image and the catchment basins correspond to the blob like objects.

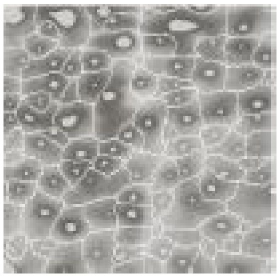


Noise and local minima lead generally to oversegmentation.

- The result is not useful.
- Solution: limit the number of allowable regions by additional knowledge.



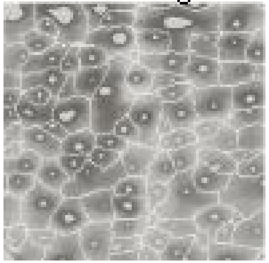
- Markers (connected components):
  – internal, associated with the objects
  – external, associated with the background.
- Here the problem is the large number of local minima.
- Smoothing may eliminate them.
- Define an internal marker (after smoothing):
- Region surrounded by points of higher altitude.
  – They form connected components.
  – All points in the connected component have the same intensity.

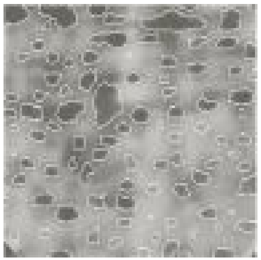


▪After smoothing, the internal markers are shown in light gray.

▪ The watershed algorithm is applied and the internal markers are the only allowable regional minima.

▪ The resulting watersheds are the external markers (shown in white).



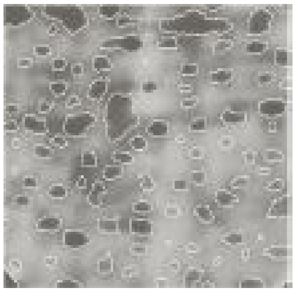


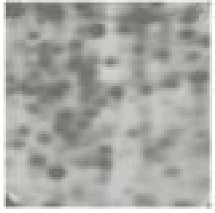
▪Each region defined by the external marker has a single internal marker and part of the background.

▪ The problem is to segment each of these regions into two segments: a single object and background.

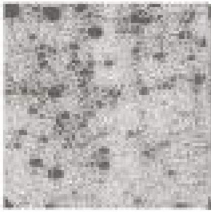
▪ The algorithms we saw in this lecture may be used (including watersheds applied to each individual region)

Final segmentation.

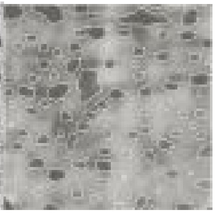




Image

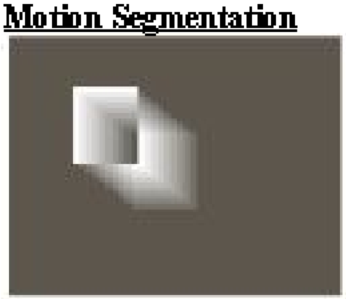


Watersheds

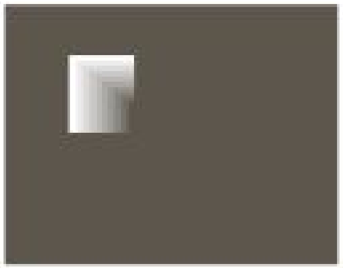


Watersheds with markers

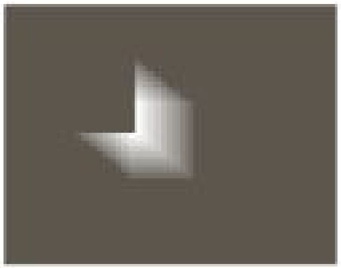
Motion Segmentation



Accumulative Difference Image (ADI) for a moving rectangle



Positive ADI

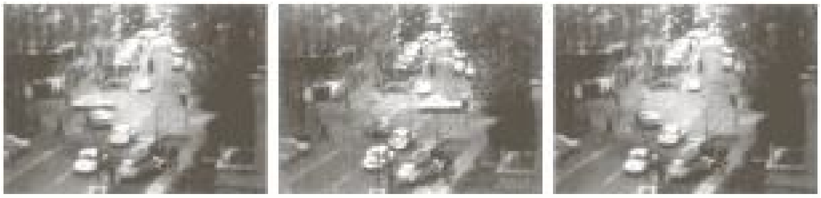


Negative ADI

$$\mathbf{A}_k(x,y)=\begin{cases} A_{k-1}(x,y)+1 & \text{if } \left|R(x,y)-f(x,y,k)\right|>T \\ A_{k-1}(x,y) & \textit{otherwise} \end{cases}$$

- The accumulator is incremented if the absolute difference between the reference frame and each successive frame is above a threshold.
- We can also have positive and negative accumulator images.

$$P_k(x,y)=\begin{cases} P_{k-1}(x,y)+1 & \text{if } \left|R(x,y)-f(x,y,t_k)\right|>T \\ P_{k-1}(x,y) & \textit{otherwise} \end{cases} \qquad N_k(x,y)=\begin{cases} N_{k-1}(x,y)+1 & \text{if } \left|R(x,y)-f(x,y,t_k)\right|\le T \\ N_{k-1}(x,y) & \textit{otherwise} \end{cases}$$



a b c

FIGURE 10.60 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)

- There is a small moving object with a 9 pixel Gaussian distribution moving with v x=0.5 and v y=0.5 pixel/frame.
- This is one of 32 frames.

FIGURE 10.61 LANDSAT frame. (Cowart, Snyder, and Ruedger.)

Basic concept:

Single one pixel object moving against a uniform background. v x=1 pixel/frame

1. Project image onto x-axis (sum columns)

2. At t=0 multiply columns of projection array by ej2 ax t, x=0,1,2,… where a is a positive integer and t is the time interval between frames

3. At t=1 do the same thing except object has moved to x'+1 This gives a accumulator array of zeroes except for the moving object projection. If the velocity is constant the projection is ej2 a(x'+t) t=cos[2 a(x'+t) t]+jsin[2 a(x'+t) t], i.e., projections of moving objects give single frequency sinusoids (for constant velocity)

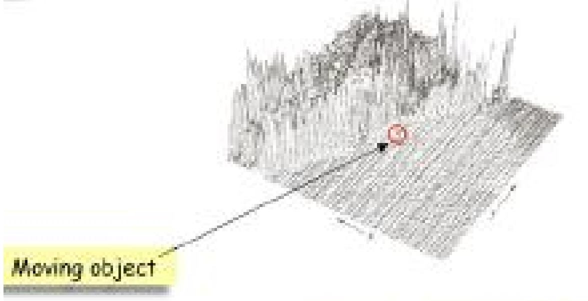
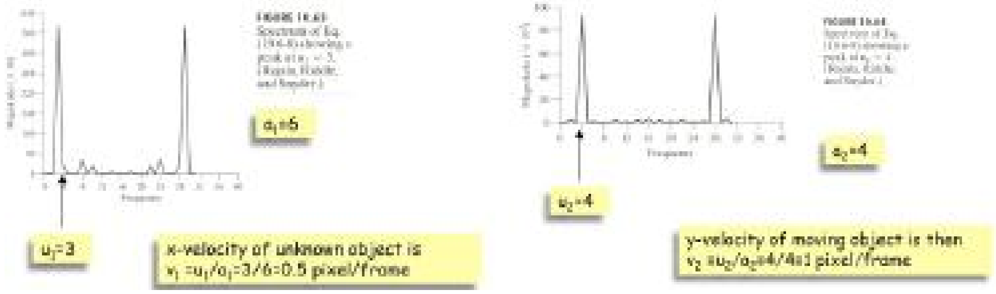


FIGURE 10.62 Intensity plot of the image in Fig. 10.61, with the target circled. (Rajala, Riddle, and Snyder.)

Intensity plot of single frame.

The a's are selected to prevent aliasing in the frequency domain. A rule of thumb is to select a as the integer closest to u_{max}/v_{max} where v_{max} is the maximum velocity and u_{max} is related to the maximum number of frames/second.



The use of motion in segmentation

Powerful cue used by humans to extract objects and regions.
▪Motion arises from
–Objects moving in the scene.
–Relative displacement between the sensing system and the scene (e.g. robotic applications, autonomous navigation).
▪We will consider motion
–Spatially.
–In the frequency domain.

Basic spatial motion segmentation

Difference image and comparison with respect to a threshold:

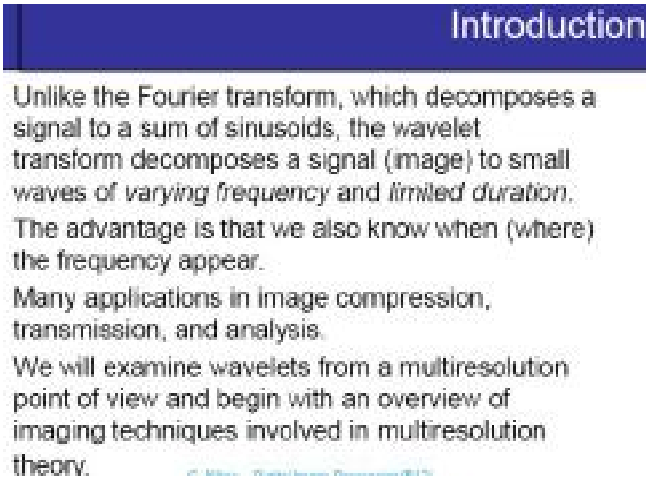
$$d_{ij}(x,y)=\begin{cases}1 & \text{if } \left|f(x,y,t_i)-f(x,y,t_j)\right|>T\\ 0 & \text{otherwise}\end{cases}$$

▪The images should be registered.
▪Illumination should be relatively constant within the bounds defined by *T*.

Unit-IV

Multi Resolution Analysis and Compression: Image Pyramids – Multi resolution expansion – Wavelet Transforms image Compression: Fundamentals – Models – Elements of Information Theory – Error Free Compression – Lossy Compression – Compression Standards.

Introduction



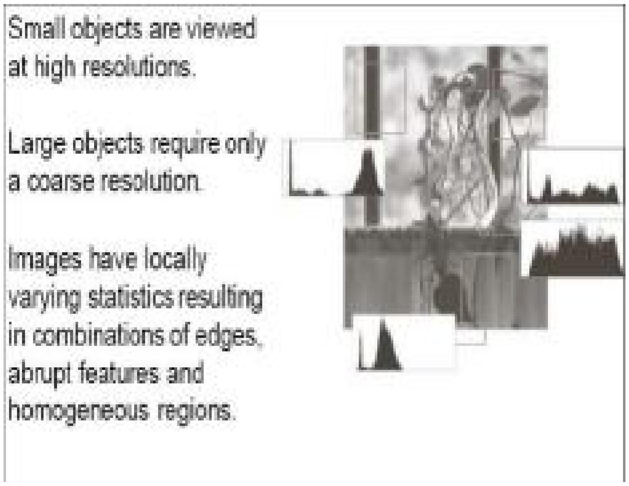
Introduction

Unlike the Fourier transform, which decomposes a signal to a sum of sinusoids, the wavelet transform decomposes a signal (image) to small waves of *varying frequency* and *limited duration*.

The advantage is that we also know when (where) the frequency appear.

Many applications in image compression, transmission, and analysis.

We will examine wavelets from a multiresolution point of view and begin with an overview of imaging techniques involved in multiresolution theory.



Small objects are viewed at high resolutions.

Large objects require only a coarse resolution.

Images have locally varying statistics resulting in combinations of edges, abrupt features and homogeneous regions.

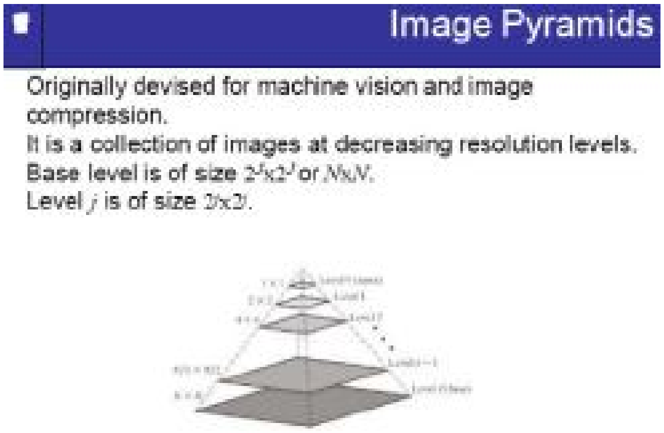


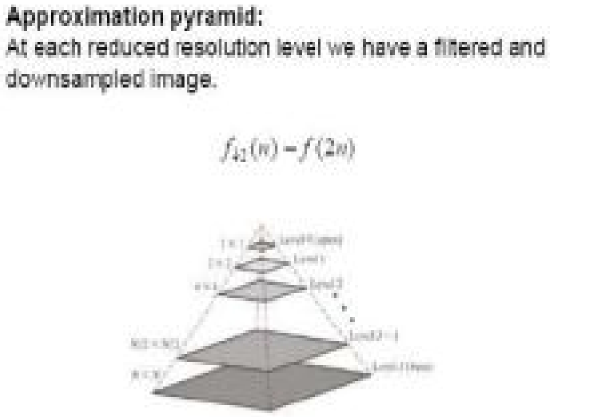
Image Pyramids

Originally devised for machine vision and image compression.

It is a collection of images at decreasing resolution levels.

Base level is of size $2^J x 2^J$ or NxN.

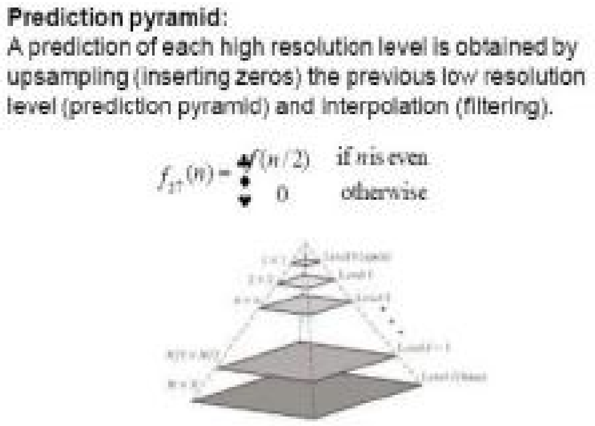
Level j is of size $2^j x 2^j$.



Approximation pyramid:

At each reduced resolution level we have a filtered and downsampled image.

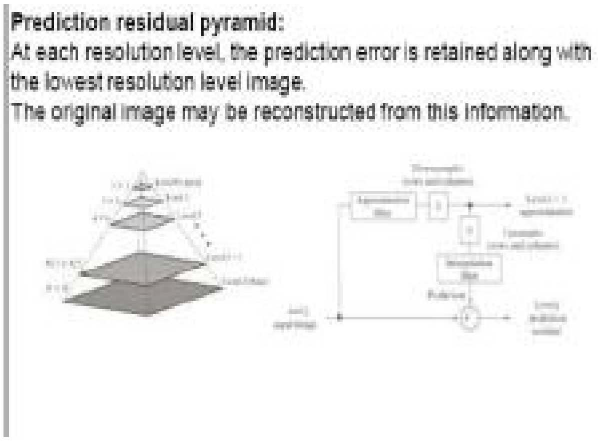
$$f_{2\downarrow}(n) = f(2n)$$



Prediction pyramid:

A prediction of each high resolution level is obtained by upsampling (inserting zeros) the previous low resolution level (prediction pyramid) and interpolation (filtering).

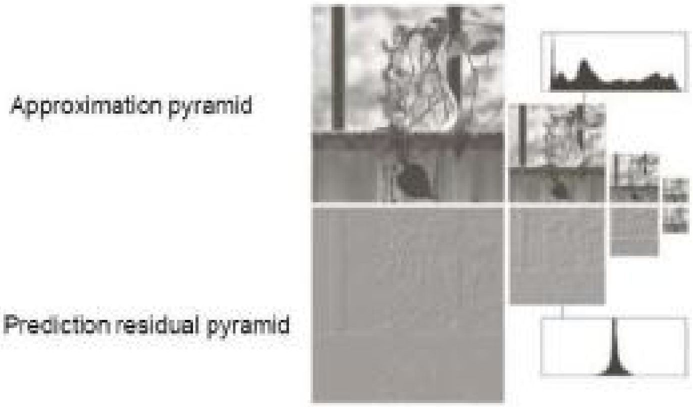
$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$



Prediction residual pyramid:

At each resolution level, the prediction error is retained along with the lowest resolution level image.

The original image may be reconstructed from this information.

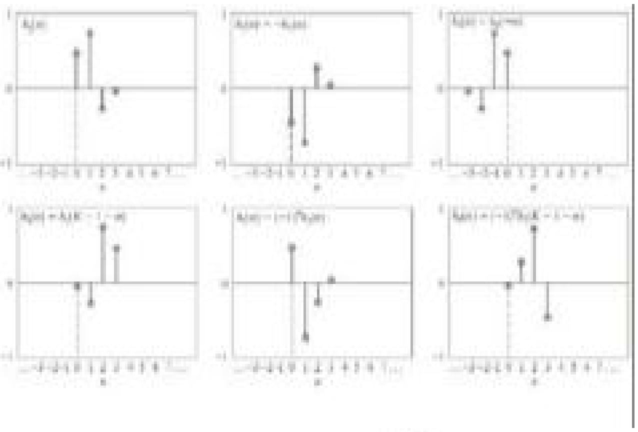


Subband Coding

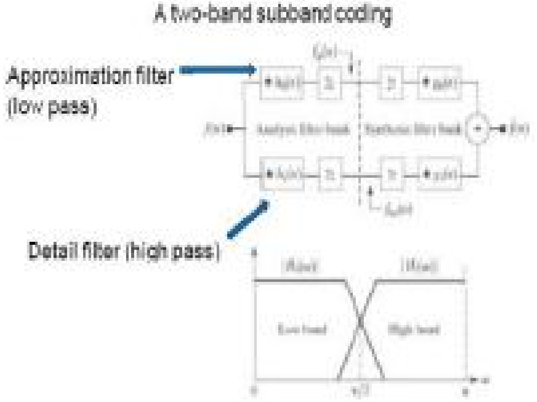
An image is decomposed to a set of bandlimited components (subbands).

The decomposition is carried by filtering and downsampling.

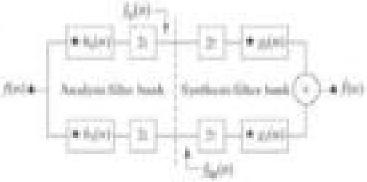
If the filters are properly selected the image may be reconstructed without error by filtering and upsampling.



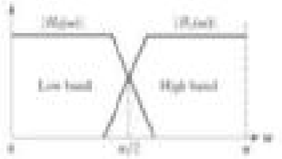
A two-band subband coding



The goal of subband coding is to select the analysis and synthesis filters in order to have *perfect reconstruction* of the signal.



It may be shown that the synthesis filters should be modulated versions of the analysis filters with one (and only one) synthesis filter being sign reversed of an analysis filter.



Also, the filters should be *biorthogonal*:

$$\langle h_i(2n-k), g_j(k) \rangle = \delta(i-j)\delta(n),\ i,j = \{0,1\}$$

Of special interest in subband coding are filters that move beyond biorthogonality and require to be *orthonormal*:

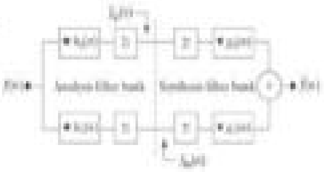
$$\langle g_i(n), g_j(n+2m) \rangle = \delta(i-j)\delta(m),\ i,j = \{0,1\}$$

In addition, orthonormal filters satisfy the following conditions:

$$g_1(n) = (-1)^n g_0(K_{even}-1-n)$$
$$h_i(n) = g_i(K_{even}-1-n),\ i = \{0,1\}$$

where the subscript means that the size of the filter should be even.

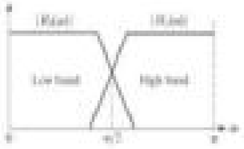
The analysis and synthesis filters should be related in one of the two ways:



$$g_0(n) = (-1)^n h_1(n)$$
$$g_1(n) = (-1)^{n+1} h_0(n)$$

or

$$g_0(n) = (-1)^{n+1} h_1(n)$$
$$g_1(n) = (-1)^n h_0(n)$$



These filters are called *cross-modulated*.

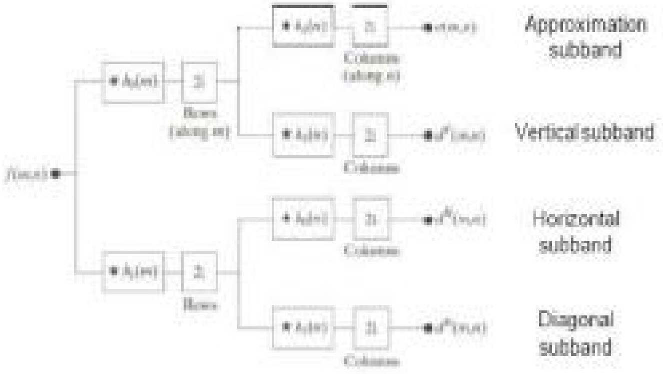
$$g_1(n) = (-1)^n g_0(K_{even}-1-n)$$
$$h_i(n) = g_i(K_{even}-1-n),\ i = \{0,1\}$$

Synthesis filters are related by order reversal and modulation.

Analysis filters are both order reversed versions of the synthesis filters.

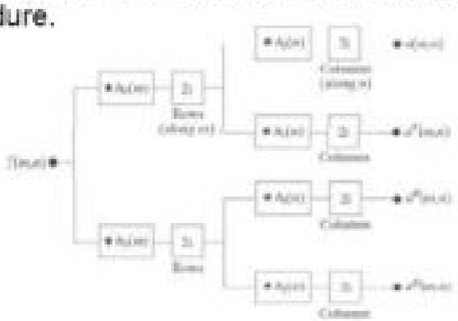
An orthonormal filter bank may be constructed around the impulse response of g_0 which is called the *prototype*.

1-D orthonormal filters may be used as 2-D separable filters for subband image coding.

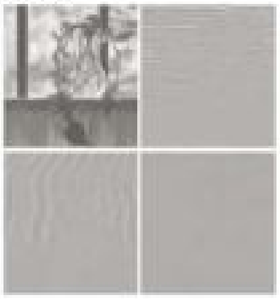


The subbbands may be subsequently split into smaller subbands.

Image synthesis is obtained by reversing the procedure.



The wavy lines are due to aliasing of the barely discernable window screen. Despite the aliasing, the image may be perfectly reconstructed.



The Haar Transform

It is due to Alfred Haar [1910].

Its basis functions are the simplest known orthonormal wavelets.

The Haar transform is both separable and symmetric:

$$\mathbf{T}=\mathbf{HFH},$$

$\mathbf{F}$ is a NxN image and $\mathbf{H}$ is the NxN transformation matrix and $\mathbf{T}$ is the NxN transformed image.

Matrix $\mathbf{H}$ contains the Haar basis functions.

The Haar basis functions $h_k(z)$ are defined for in $0 \le z \le 1$, for $k=0,1,\ldots,N-1$, where $N=2^n$.

To generate $\mathbf{H}$:

- we define the integer $k=2^p+q-1$, with $0 \le p \le N-1$.
- if $p=0$, then $q=0$ or $q=1$.
- if $p \ne 0$, $1 \le q \le 2^p$

For the above pairs of p and q, a value for k is determined and the Haar basis functions are computed

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \; z \in [0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \le z \le (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \le z \le q/2^p \\ 0 & \text{otherwise}, \; z \in [0,1] \end{cases}$$

The ith row of a NxN Haar transformation matrix contains the elements of $h_i(z)$ for $z=0/N, 1/N, 2/N, \ldots, (N-1)/N$.

For instance, for $N=4$, p, q and k have the following values:

k	p	q
0	0	0
1	0	1
2	1	1
3	1	2

and the 4x4 transformation matrix is:

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Similarly, for $N=2$, the 2x2 transformation matrix is:

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The rows of $\mathbf{H}_2$ are the simplest filters of length 2 that may be used as analysis filters $h_0(n)$ and $h_1(n)$ of a perfect reconstruction filter bank.

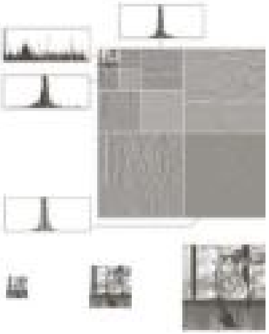
Moreover, they can be used as scaling and wavelet vectors (defined in what follows) of the simplest and oldest wavelet transform.

An introductory example to wavelet analysis

Combination of the key features examined so far:

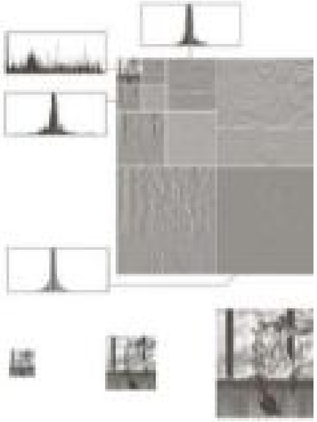
- pyramids,
- subband coding,
- the Haar transform.

The decomposition is called the discrete wavelet transform and it will be developed later in the course.



With the exception of the upper left image, the histograms are very similar with values close to zero. This fact may be exploited for compression purposes.

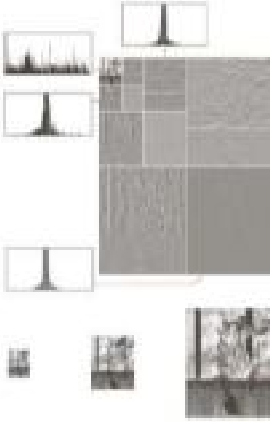
The subimages may be used to construct coarse and fine resolution approximations.



The decomposition was obtained by subband coding in 2-D.

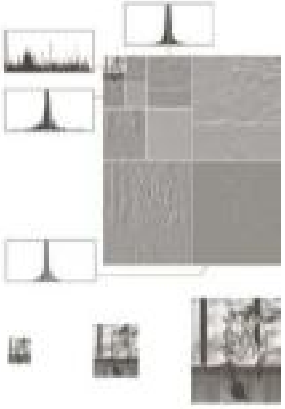
After the generation of the four subbands, the approximation subband was further decomposed into four new subbands (using the same filter bank). The procedure was repeated for the new approximation subband.

This procedure characterizes the wavelet transform as the subimages become smaller in size.



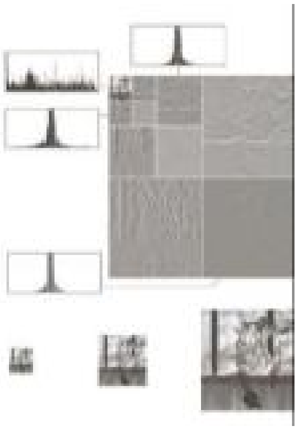
This is not the Haar transform of the image. The Haar transform of the image is different.

Although these filter bank coefficients were taken by the Haar transformation matrix, there is a variety of orthonormal filters that may be used.



Each subimage represents a specific band of spatial frequencies in the original image.

Many of the subimages demonstrate directional sensitivity (e.g. the subimage in the upper right corner captures horizontal edge information in the original image).



Compression

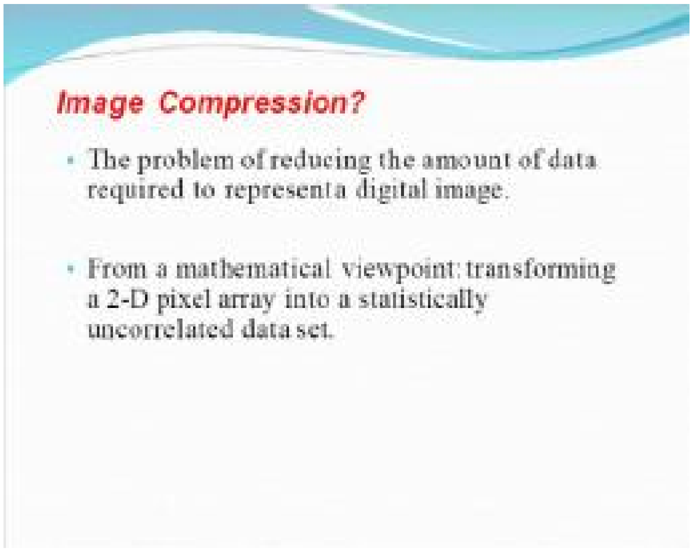
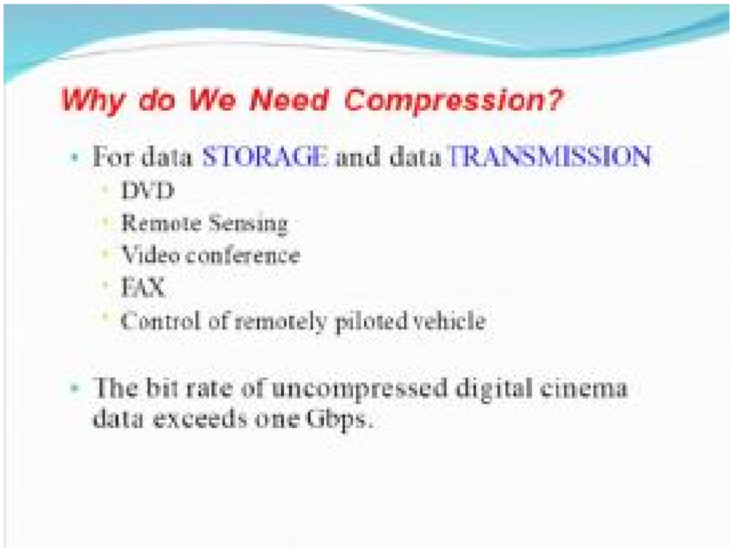


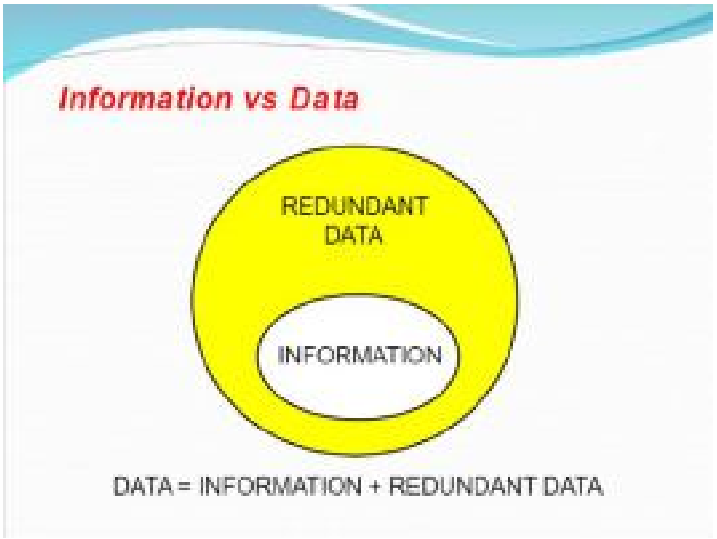
Image Compression?

- The problem of reducing the amount of data required to representa digital image.
- From a mathematical viewpoint: transforming a 2-D pixel array into a statistically uncorrelated data set.



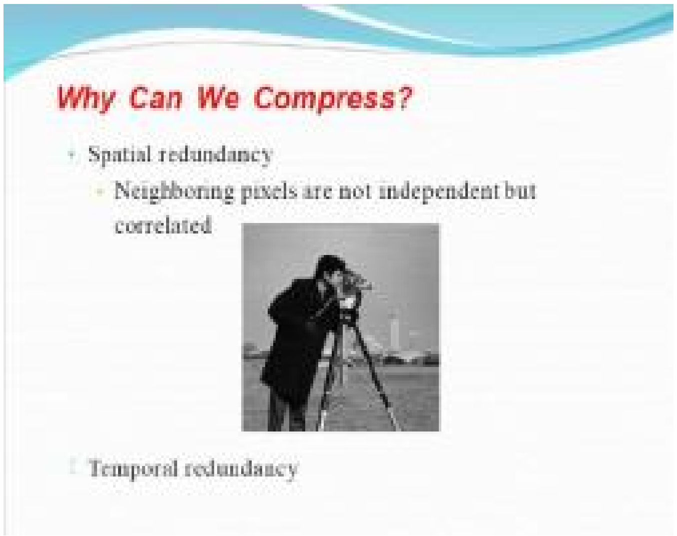
Why do We Need Compression?

- For data STORAGE and data TRANSMISSION
  - DVD
  - Remote Sensing
  - Video conference
  - FAX
  - Control of remotely piloted vehicle
- The bit rate of uncompressed digital cinema data exceeds one Gbps.



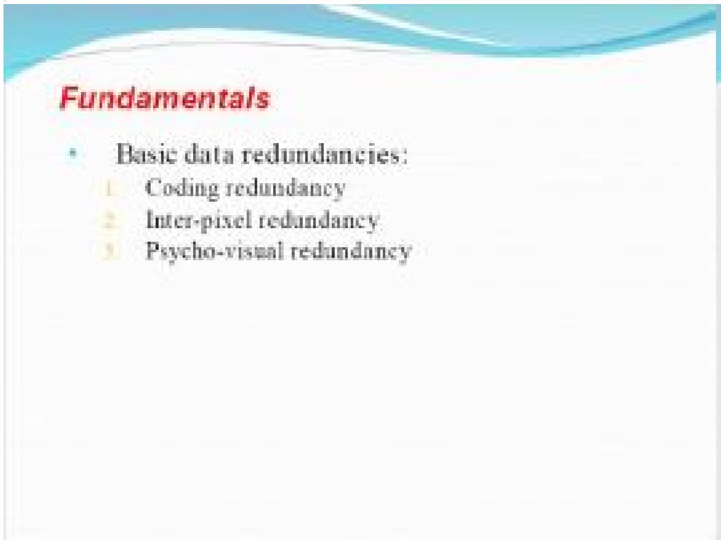
Information vs Data

DATA = INFORMATION + REDUNDANT DATA



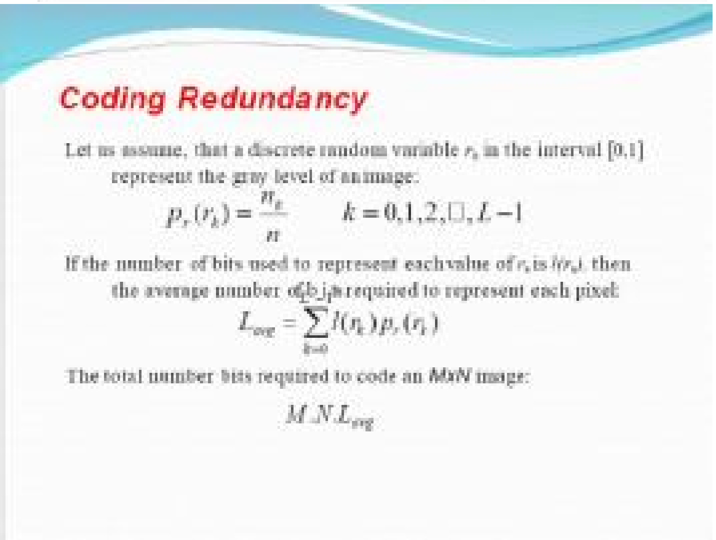
Why Can We Compress?

- Spatial redundancy
  - Neighboring pixels are not independent but correlated
- Temporal redundancy



Fundamentals

- Basic data redundancies:
  1. Coding redundancy
  2. Inter-pixel redundancy
  3. Psycho-visual redundancy



Coding Redundancy

Let us assume, that a discrete random variable r_k in the interval [0,1] represent the gray level of an image:

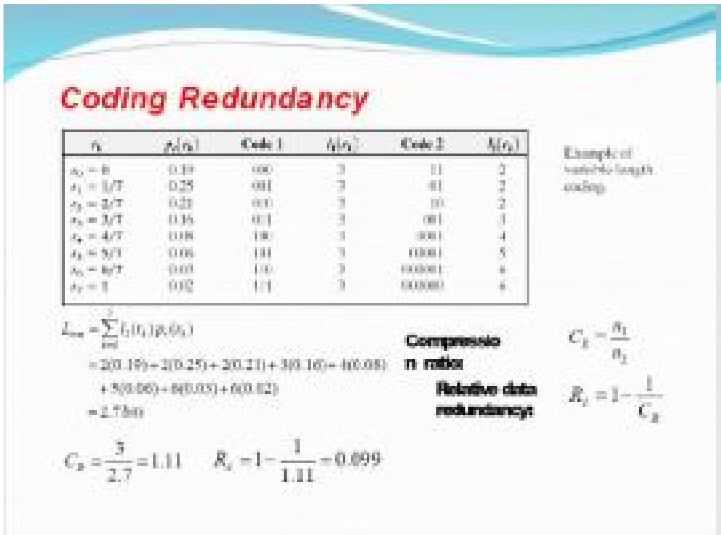
$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0,1,2,\square,L-1$$

If the number of bits used to represent each value of r_k is $l(r_k)$, then the average number of bits is required to represent each pixel:

$$L_{avg} = \sum_{k=0} l(r_k) p_r(r_k)$$

The total number bits required to code an *MxN* image:

$$M.N.L_{avg}$$



Coding Redundancy

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

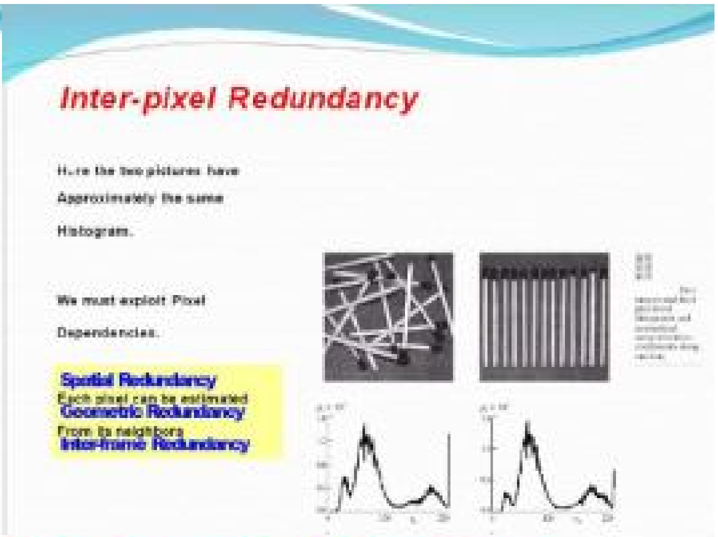
Example of variable-length coding.

$$L_{avg} = \sum_{k=0}^{7} l_2(r_k) p_r(r_k)$$
$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08)$$
$$+ 5(0.06) + 6(0.03) + 6(0.02)$$
$$= 2.7 \text{bits}$$

Compression ratio $C_R = \frac{n_1}{n_2}$

Relative data redundancy: $R_D = 1 - \frac{1}{C_R}$

$$C_R = \frac{3}{2.7} = 1.11 \quad R_D = 1 - \frac{1}{1.11} = 0.099$$



Inter-pixel Redundancy

H.re the two pictures have

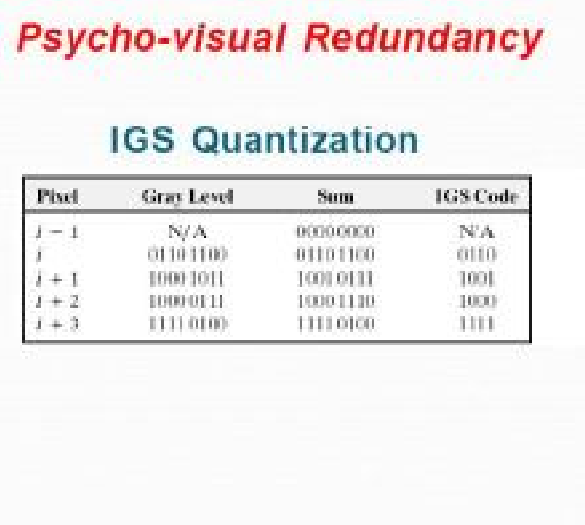
Approximately the same

Histogram.

We must exploit Pixel

Dependencies.

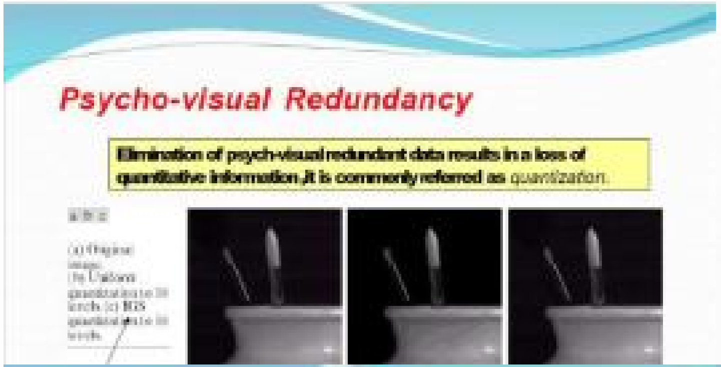
Spatial Redundancy
Each pixel can be estimated
Geometric Redundancy
From its neighbors
Inter-frame Redundancy



Psycho-visual Redundancy

IGS Quantization

Pixel	Gray Level	Sum	IGS Code
$i - 1$	N/A	0000 0000	N/A
i	0110 1100	0110 1100	0110
$i + 1$	1000 1011	1001 0111	1001
$i + 2$	1000 0111	1000 1110	1000
$i + 3$	1111 0100	1111 0100	1111

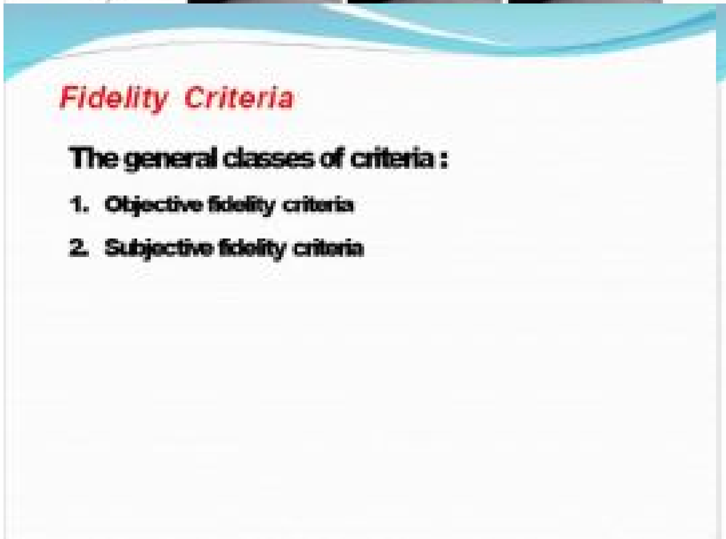


Psycho-visual Redundancy

Elimination of psych-visual redundant data results in a loss of quantitative information ,it is commonly referred as quantization.

a b c

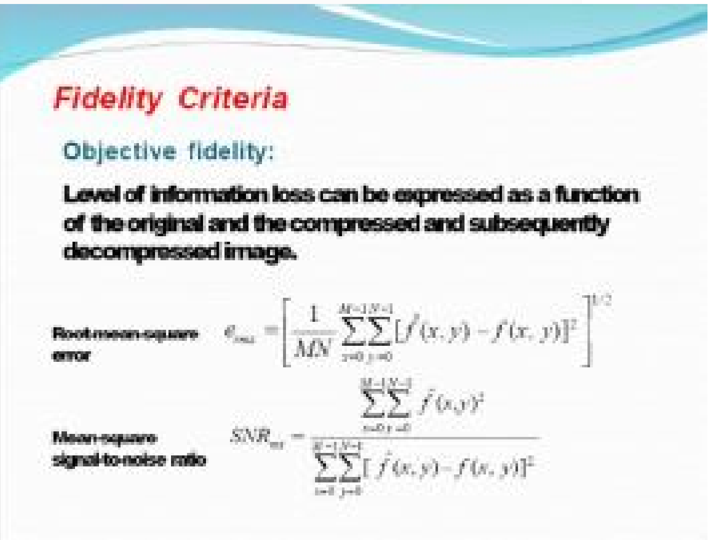
(a) Original image. (b) Uniform quantization to 16 levels. (c) IGS quantization to 16 levels.



Fidelity Criteria

The general classes of criteria :

1. **Objective fidelity criteria**
2. **Subjective fidelity criteria**



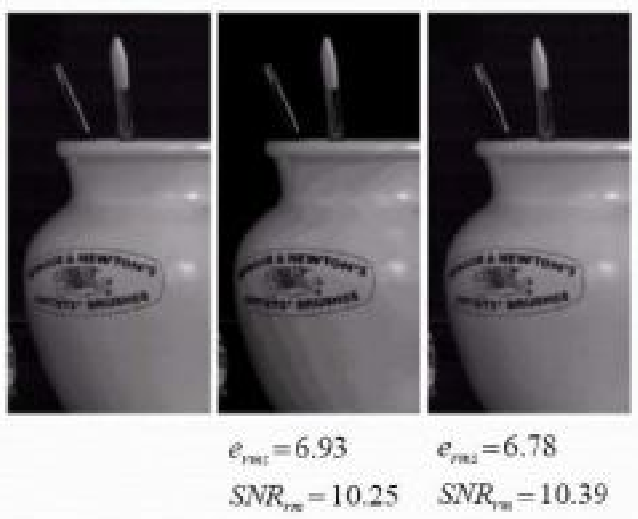
Fidelity Criteria

Objective fidelity:

Level of information loss can be expressed as a function of the original and the compressed and subsequently decompressed image.

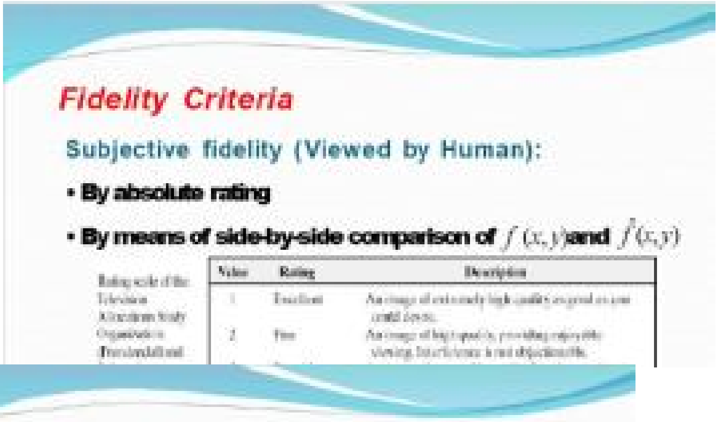
Root-mean-square error
$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \right]^{1/2}$$

Mean-square signal-to-noise ratio
$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2}$$



$e_{rms} = 6.93$ $e_{rms} = 6.78$

$SNR_{rms} = 10.25$ $SNR_{rms} = 10.39$



Fidelity Criteria

Subjective fidelity (Viewed by Human):

- By absolute rating
- By means of side-by-side comparison of $f(x,y)$ and $\hat{f}(x,y)$

Rating scale of the Television Allocations Study Organization (Frendendall and

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.

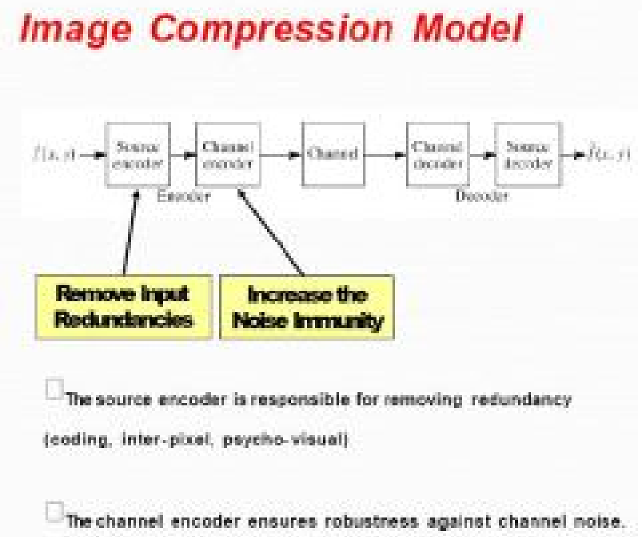


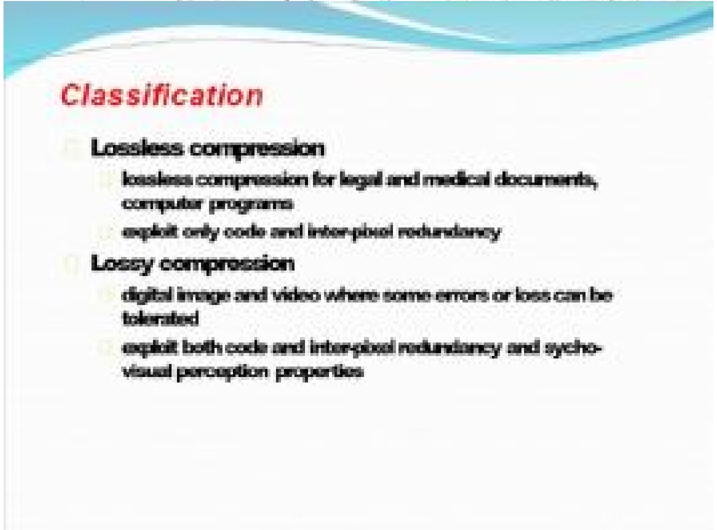
Image Compression Model

Remove Input Redundancies

Increase the Noise Immunity

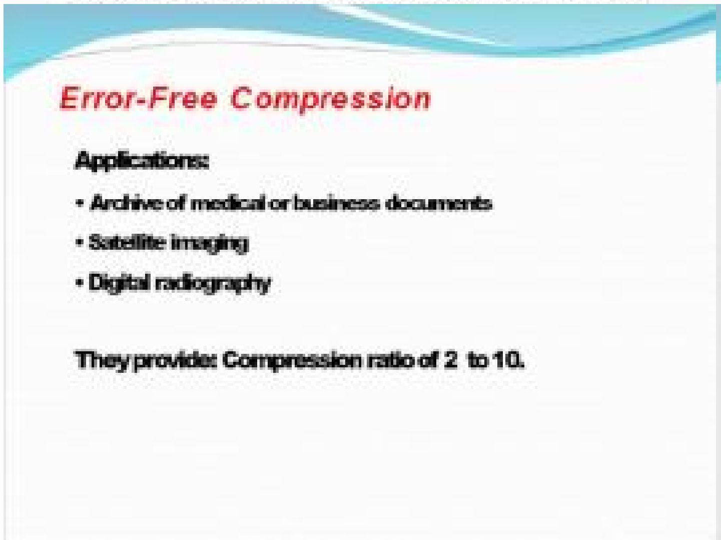
□ The source encoder is responsible for removing redundancy (coding, inter-pixel, psycho-visual)

□ The channel encoder ensures robustness against channel noise.



Classification

- Lossless compression
  - lossless compression for legal and medical documents, computer programs
  - exploit only code and inter-pixel redundancy
- Lossy compression
  - digital image and video where some errors or loss can be tolerated
  - exploit both code and inter-pixel redundancy and sycho-visual perception properties

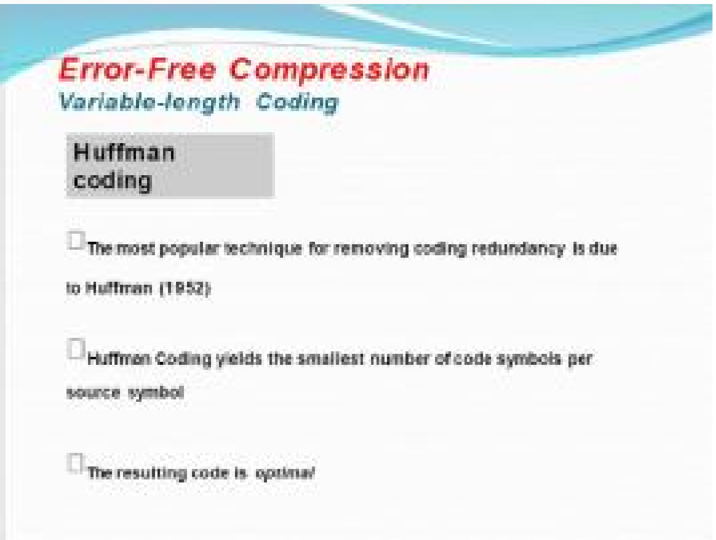


Error-Free Compression

Applications:

- Archive of medical or business documents
- Satellite imaging
- Digital radiography

They provide: Compression ratio of 2 to 10.



Error-Free Compression

Variable-length Coding

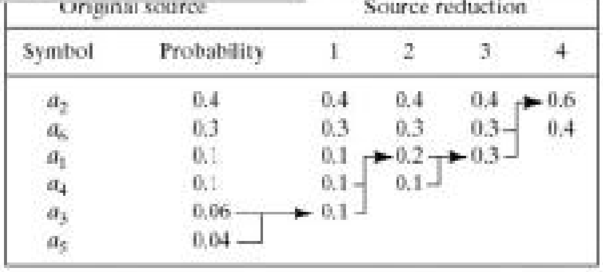
Huffman coding

□ The most popular technique for removing coding redundancy is due to Huffman (1952)

□ Huffman Coding yields the smallest number of code symbols per source symbol

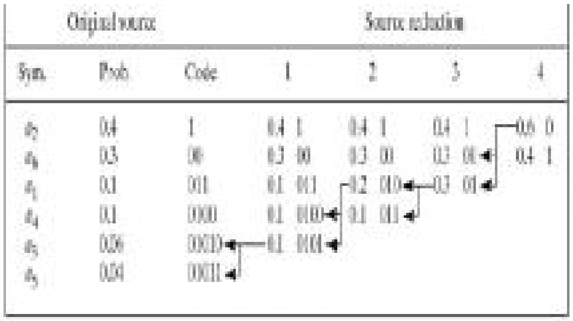
□ The resulting code is optimal

Huffman coding (optimal code)



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

FIGURE 8.12
Huffman code assignment procedure.



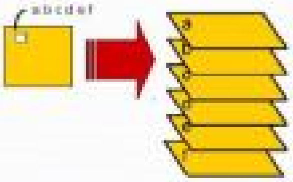
Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

$$L_{avg} = (0.4)(1)+(0.3)(2)+(0.1)(3)+(0.1)(4)+(0.06)(5)+(0.04)(5)$$
$$= 2.2\ bits\ /\ symbol$$
$$entropy = 2.14 bits\ /\ symbol$$

Error-Free Compression

Bit-plane coding

Bit-plane coding is based on decomposing a multilevel image into a series of binary images and compressing each binary image .



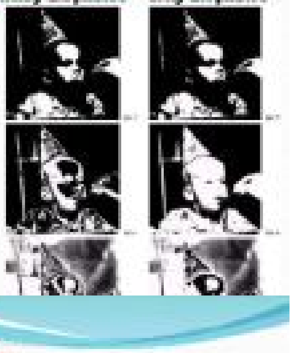
a b
A 1024 × 1024 (a) 8-bit monochrome image and (b) binary image.

Error-Free Compression

Bit-plane coding

Bit-plane decomposition

The four most significant binary (left column) and Gray-coded (right column) bit planes of the image in Fig. 8.14(a).



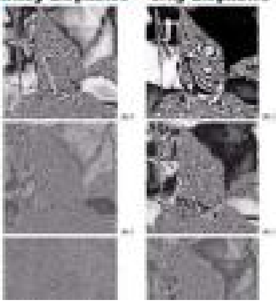
$m-bit\ gray\ scale:\quad a_{m-1}2^{m-1}+a_{m-2}2^{m-2}+\cdots+a_0 2^0$

Error-Free Compression

Bit-plane coding

Bit-plane decomposition

The four least significant binary (left column) and Gray-coded (right column) bit planes of the image in Fig. 8.14(a).



Error-Free Compression

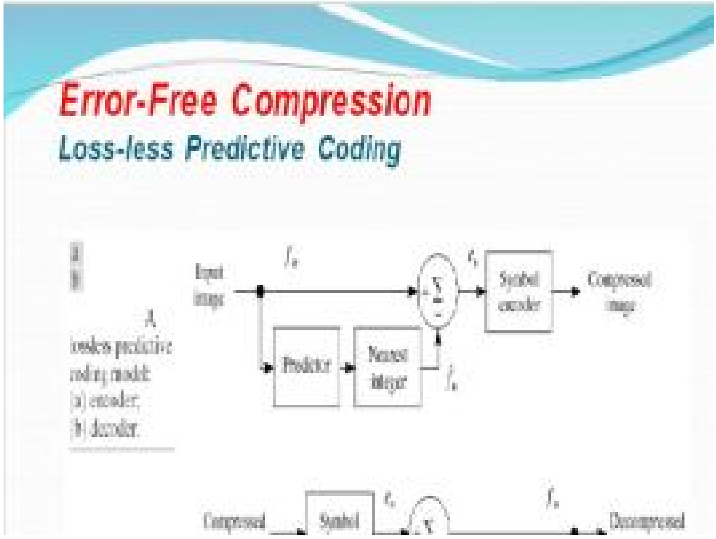
Bit-plane coding

- **Constant area Coding**
- **One-dimensional run-length coding** $H_{RL} = \frac{H_0 + H_1}{L_0 + L_1}$

Average values of black and white run lengths

- **Two-dimensional RLC**

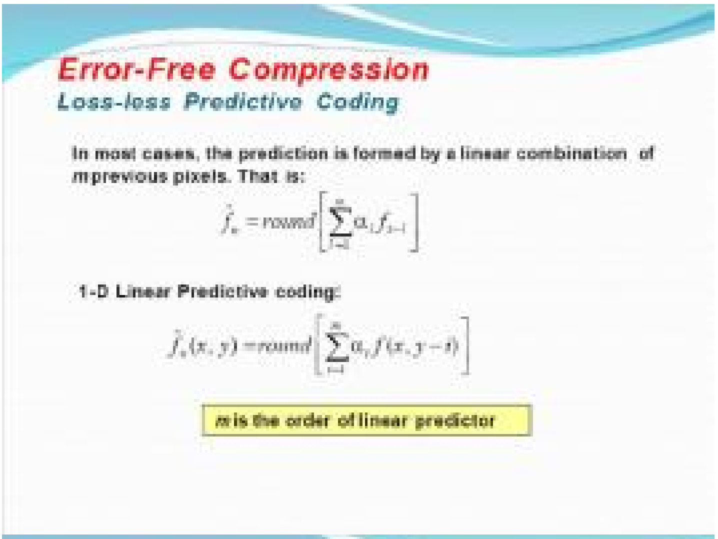
Relative Address Coding (RAC) is based on tracking the binary transitions.



Error-Free Compression

Loss-less Predictive Coding

A lossless predictive coding model: (a) encoder; (b) decoder.



Error-Free Compression

Loss-less Predictive Coding

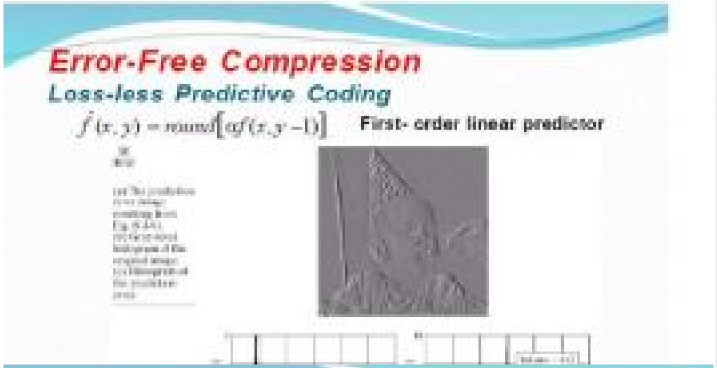
In most cases, the prediction is formed by a linear combination of *m* previous pixels. That is:

$$\hat{f}_n = round\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

1-D Linear Predictive coding:

$$\hat{f}_n(x, y) = round\left[\sum_{i=1}^{m} \alpha_i f(x, y-i)\right]$$

m is the order of linear predictor



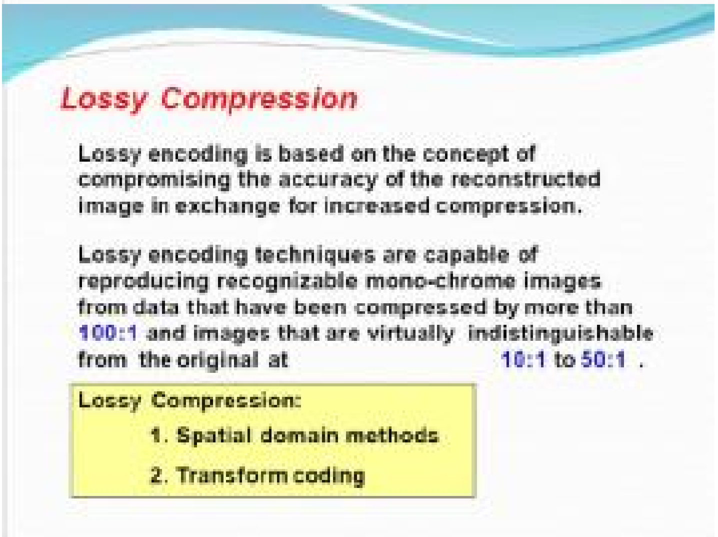
Error-Free Compression

Loss-less Predictive Coding

$$\hat{f}(x, y) = round[\alpha f(x, y-1)]$$ First- order linear predictor

a
b c

(a) The prediction error image resulting from Eq. (8.4-[illegible]). (b) Gray-level histogram of the original image. (c) Histogram of the prediction error.



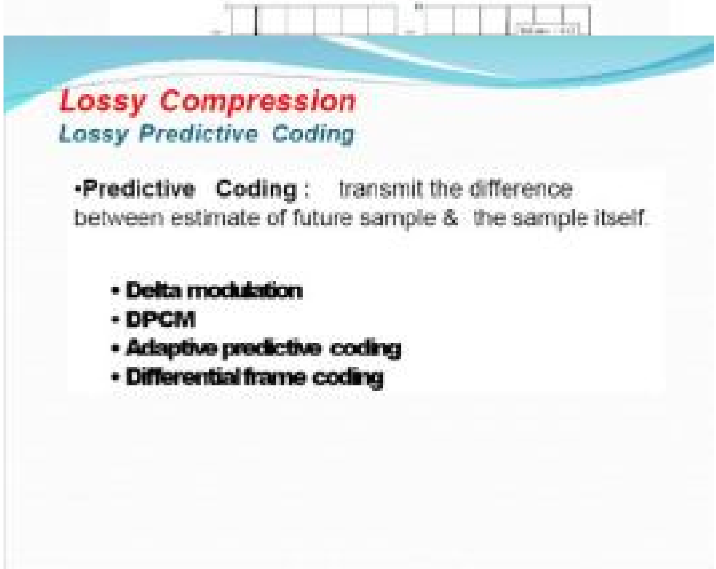
Lossy Compression

Lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression.

Lossy encoding techniques are capable of reproducing recognizable mono-chrome images from data that have been compressed by more than 100:1 and images that are virtually indistinguishable from the original at 10:1 to 50:1 .

Lossy Compression:

1. Spatial domain methods
2. Transform coding

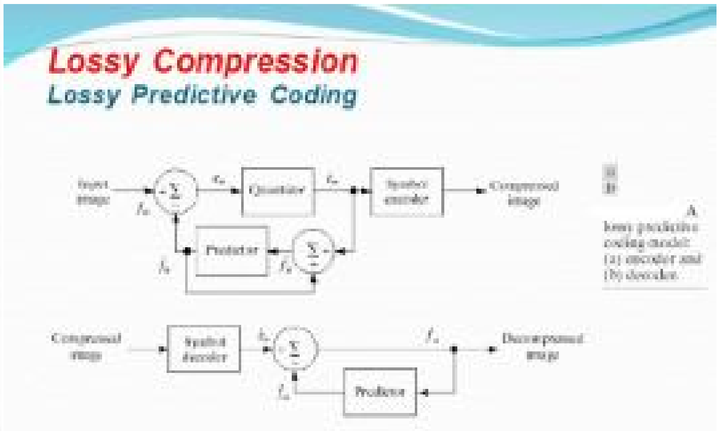


Lossy Compression

Lossy Predictive Coding

•**Predictive Coding :** transmit the difference between estimate of future sample & the sample itself.

- **Delta modulation**
- **DPCM**
- **Adaptive predictive coding**
- **Differential frame coding**



Lossy Compression

Lossy Predictive Coding

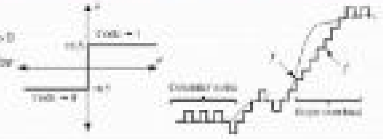
A lossy predictive coding model: (a) encoder and (b) decoder.

Lossy Compression

Lossy Predictive Coding – Delta Modulation (DM)

$\hat{f}_n = \alpha \dot{f}_{n-1}$

$\dot{e}_n = \begin{cases} +\zeta & \text{for } e_n > 0 \\ -\zeta & \text{otherwise} \end{cases}$

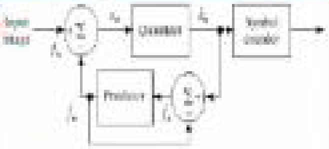


(a) (b) An example of delta modulation.

Input		Encoder				Decoder		Error
n	f	$\hat{f}$	e	$\dot{e}$	$\dot{f}$	$\hat{f}$	$\dot{f}$	$[f - \dot{f}]$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
2	14	20.5	−6.5	−6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.5	53.0	59.5	17.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Lossy Compression

Optimal Prediction



$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n$$

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}$$

$$E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^{m} \alpha_i f_{n-i}\right]^2\right\}$$

$$\sum_{i=1}^{m} \alpha_i \le 1$$

Differential Pulse Code Modulation (DPCM)

Lossy Compression

Optimal Prediction

Prediction Error

$\hat{f}(x, y) = 0.97 f(x, y-1)$ **Pred. #1**

$\hat{f}(x, y) = 0.5 f(x, y-1) + 0.5 f(x-1, y)$ **Pred. #2**

$\hat{f}(x, y) = 0.75 f(x, y-1) + 0.75 f(x-1, y) - 0.5 f(x-1, y-1)$ **Pred. #3**

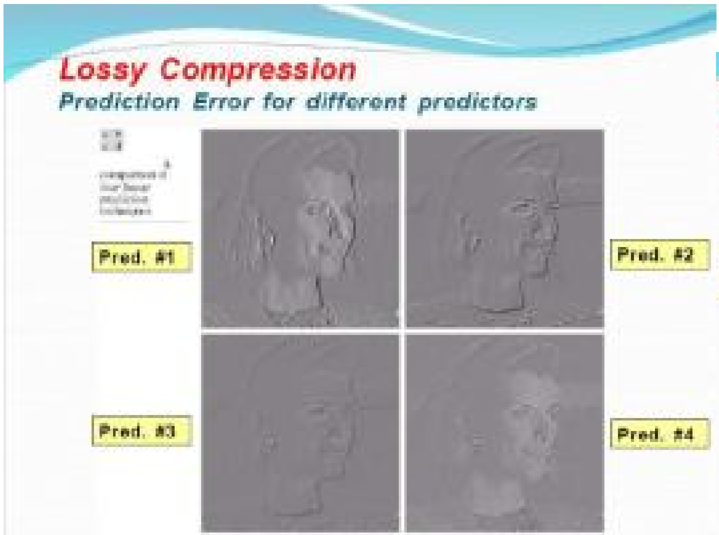
$\hat{f}(x, y) = \begin{cases} 0.97 f(x, y-1) & \text{if } \Delta h \le \Delta v \\ 0.97 f(x-1, y) & \text{otherwise} \end{cases}$ **Pred. #4**

$\Delta h = |f(x-1, y) - f(x-1, y-1)|$ *and* $\Delta v = |f(x, y-1) - f(x-1, y-1)|$

Lossy Compression

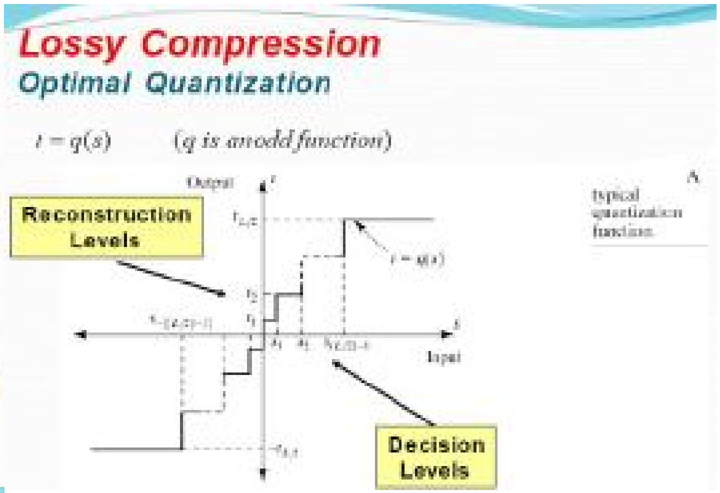
Optimal Prediction

A 512 × 512 8-bit monochrome image.



Lossy Compression

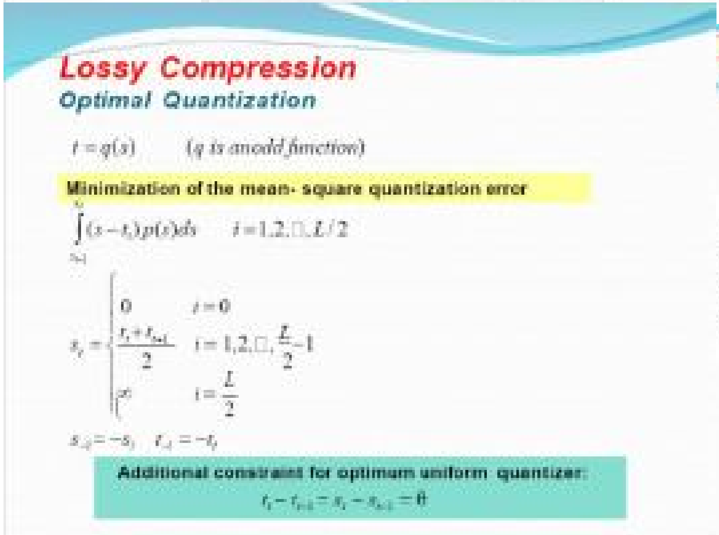
Prediction Error for different predictors



Lossy Compression

Optimal Quantization

$t = q(s)$ (*q is an odd function*)



Lossy Compression

Optimal Quantization

$t = q(s)$ (*q is an odd function*)

Minimization of the mean- square quantization error

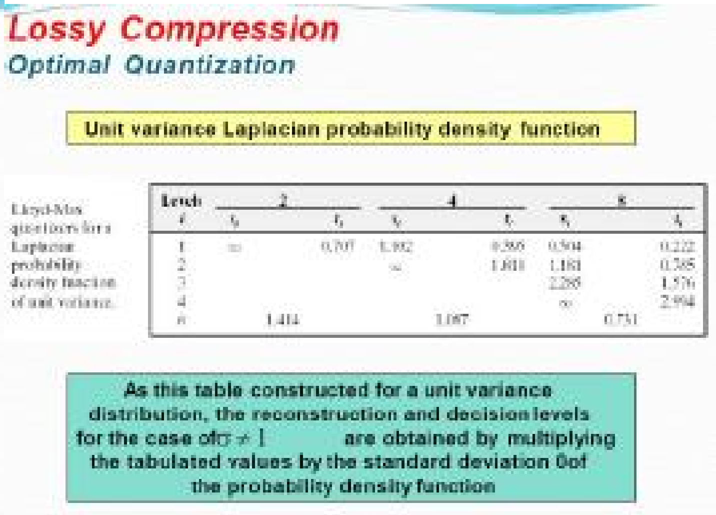
$$\int_{s_{i-1}}^{s_i} (s - t_i)\, p(s)\, ds \qquad i = 1, 2, \square, L/2$$

$$s_i = \begin{cases} 0 & i = 0 \\ \dfrac{t_i + t_{i+1}}{2} & i = 1, 2, \square, \dfrac{L}{2} - 1 \\ \infty & i = \dfrac{L}{2} \end{cases}$$

$s_{-i} = -s_i \quad t_{-i} = -t_i$

Additional constraint for optimum uniform quantizer:

$$t_i - t_{i-1} = s_i - s_{i-1} = \theta$$



Lossy Compression

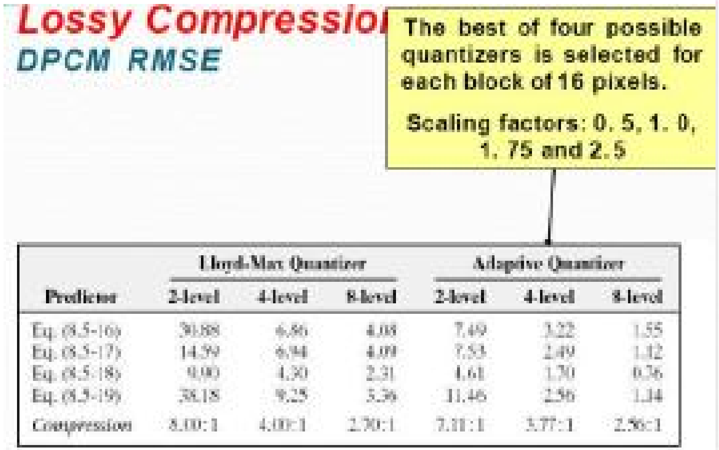
Optimal Quantization

Unit variance Laplacian probability density function

Lloyd-Max quantizers for a Laplacian probability density function of unit variance.

Levels	2		4		8	
i	s_i	t_i	s_i	t_i	s_i	t_i
1	∞	0.707	1.102	0.395	0.504	0.222
2			∞	1.810	1.181	0.785
3					2.285	1.576
4					∞	2.994
θ	1.414		1.087		0.731	

As this table constructed for a unit variance distribution, the reconstruction and decision levels for the case of $\sigma \neq 1$ are obtained by multiplying the tabulated values by the standard deviation 0of the probability density function



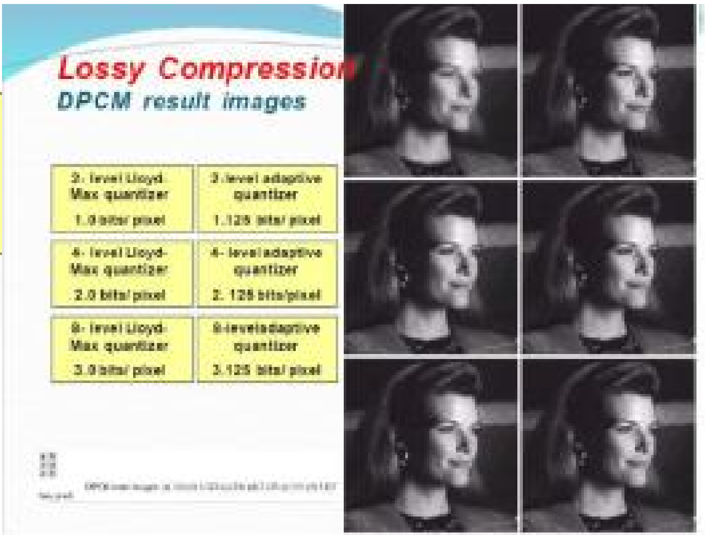
Lossy Compression

DPCM RMSE

The best of four possible quantizers is selected for each block of 16 pixels.

Scaling factors: 0. 5, 1. 0, 1. 75 and 2.5

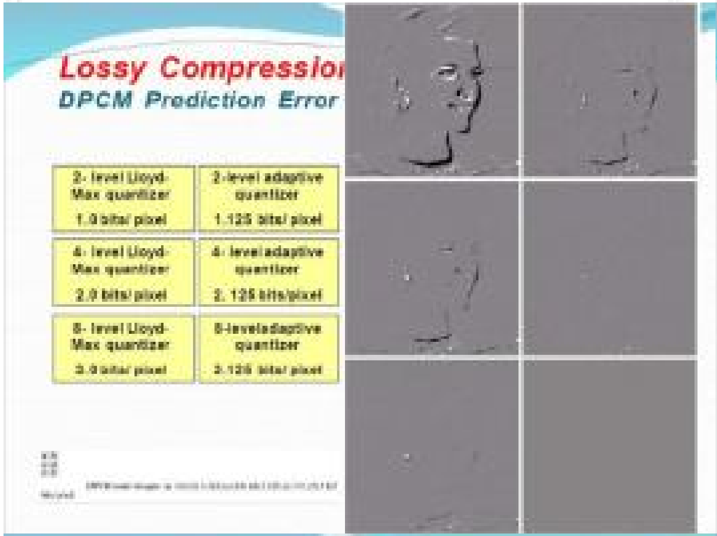
	Lloyd-Max Quantizer			Adaptive Quantizer		
Predictor	2-level	4-level	8-level	2-level	4-level	8-level
Eq. (8.5-16)	30.88	6.86	4.08	7.49	3.22	1.55
Eq. (8.5-17)	14.59	6.94	4.09	7.53	2.49	1.12
Eq. (8.5-18)	9.90	4.30	2.31	4.61	1.70	0.76
Eq. (8.5-19)	38.18	9.25	5.36	11.46	2.56	1.14
Compression	8.00:1	4.00:1	2.70:1	7.11:1	3.77:1	2.56:1



Lossy Compression

DPCM result images

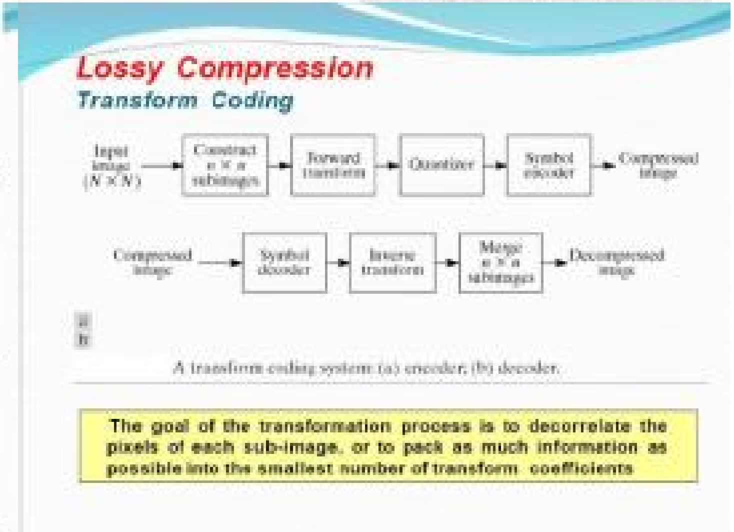
2- level Lloyd-Max quantizer 1.0 bits/ pixel	2-level adaptive quantizer 1.125 bits/ pixel
4- level Lloyd-Max quantizer 2.0 bits/ pixel	4- level adaptive quantizer 2. 125 bits/pixel
8- level Lloyd-Max quantizer 3.0 bits/ pixel	8-leveladaptive quantizer 3.125 bits/ pixel



Lossy Compression

DPCM Prediction Error

2- level Lloyd-Max quantizer 1.0 bits/ pixel	2-level adaptive quantizer 1.125 bits/ pixel
4- level Lloyd-Max quantizer 2.0 bits/ pixel	4- level adaptive quantizer 2. 125 bits/pixel
8- level Lloyd-Max quantizer 3.0 bits/ pixel	8-leveladaptive quantizer 3.125 bits/ pixel



Lossy Compression

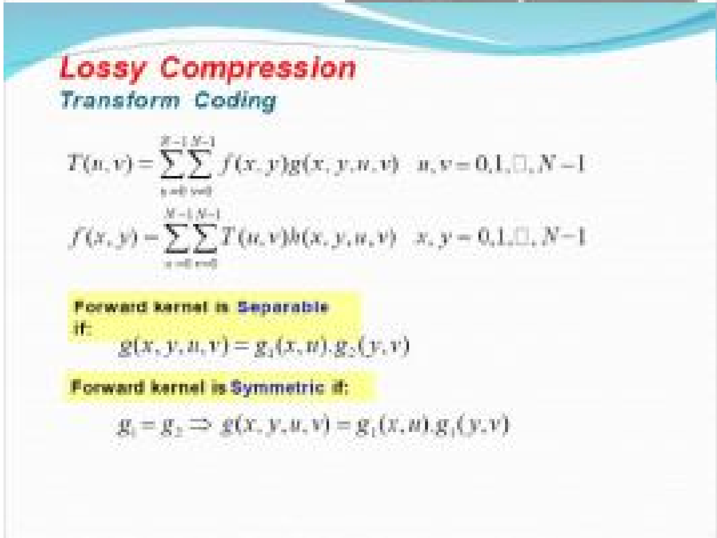
Transform Coding

Input image ($N \times N$) → Construct $n \times n$ subimages → Forward transform → Quantizer → Symbol encoder → Compressed image

Compressed image → Symbol decoder → Inverse transform → Merge $n \times n$ subimages → Decompressed image

A transform coding system: (a) encoder; (b) decoder.

The goal of the transformation process is to decorrelate the pixels of each sub-image, or to pack as much information as possible into the smallest number of transform coefficients



Lossy Compression

Transform Coding

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

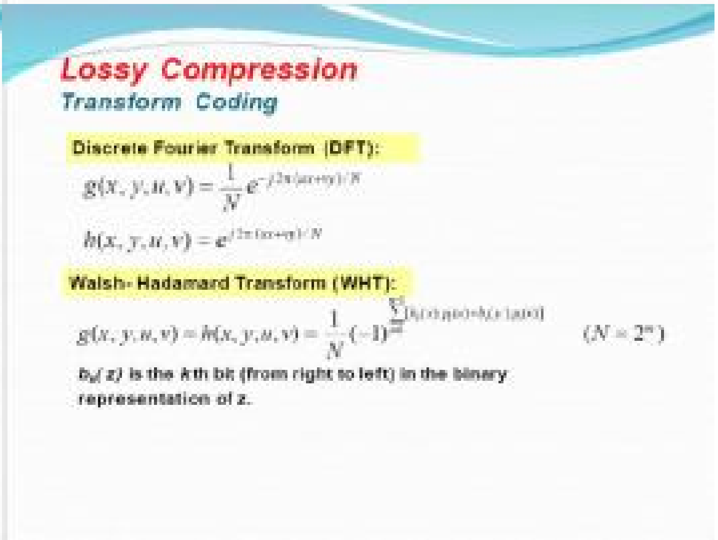
$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} T(u,v)h(x,y,u,v) \quad x,y=0,1,\square,N-1$$

Forward kernel is Separable if:

$$g(x,y,u,v)=g_1(x,u).g_2(y,v)$$

Forward kernel is Symmetric if:

$$g_1=g_2 \Rightarrow g(x,y,u,v)=g_1(x,u).g_1(y,v)$$



Lossy Compression

Transform Coding

Discrete Fourier Transform (DFT):

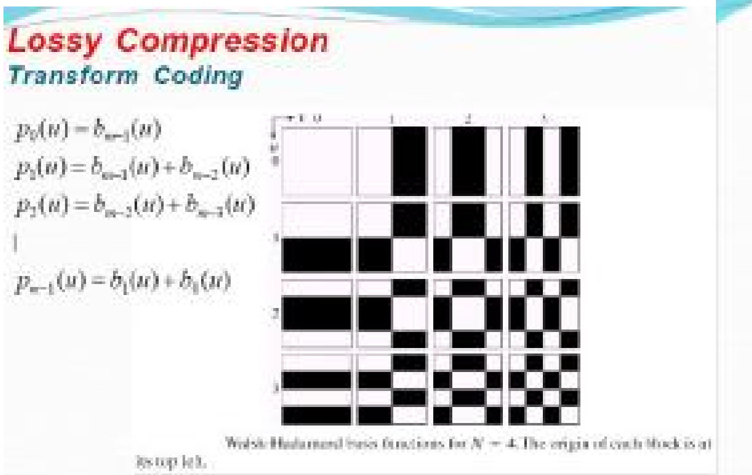
$$g(x,y,u,v)=\frac{1}{N}e^{-j2\pi(ux+vy)/N}$$

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

Walsh- Hadamard Transform (WHT):

$$g(x,y,u,v)=h(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{m-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]} \qquad (N=2^m)$$

$b_k(z)$ is the kth bit (from right to left) in the binary representation of z.

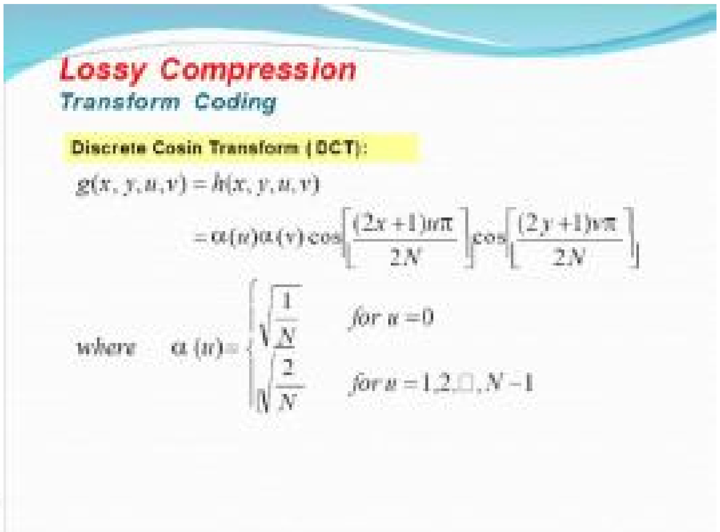


Lossy Compression

Transform Coding

$$p_0(u)=b_{m-1}(u)$$
$$p_1(u)=b_{m-1}(u)+b_{m-2}(u)$$
$$p_2(u)=b_{m-2}(u)+b_{m-3}(u)$$
$$\vdots$$
$$p_{m-1}(u)=b_1(u)+b_0(u)$$

Walsh-Hadamard basis functions for $N = 4$. The origin of each block is at its top left.



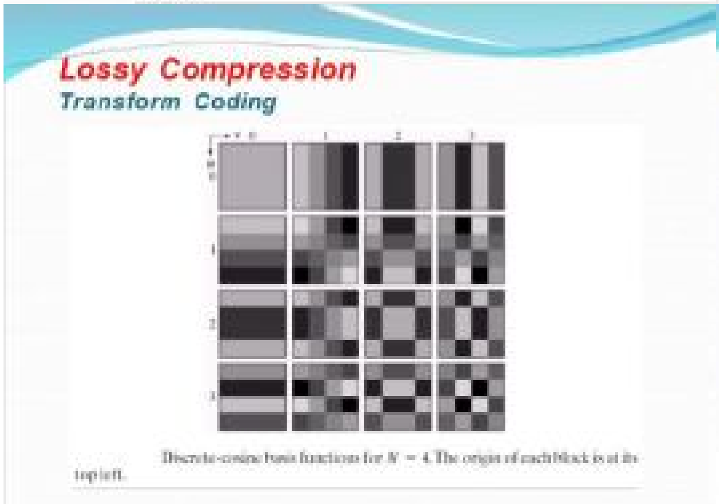
Lossy Compression

Transform Coding

Discrete Cosin Transform (DCT):

$$g(x,y,u,v)=h(x,y,u,v)$$
$$=\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

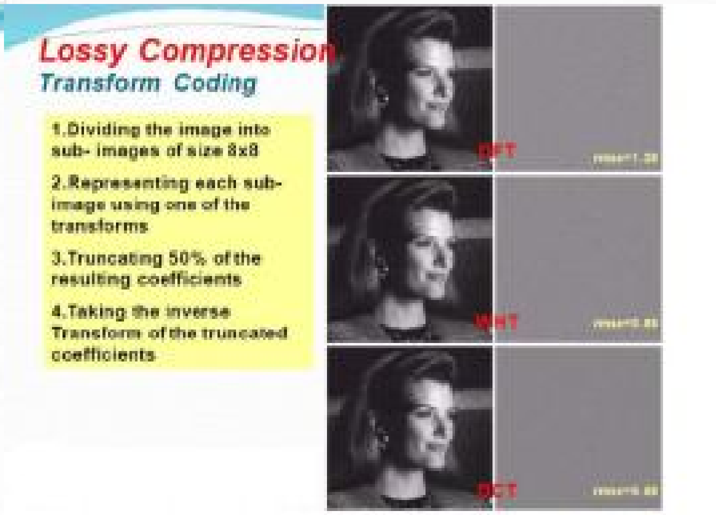
$$\text{where}\quad \alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & \text{for } u=0\\ \sqrt{\frac{2}{N}} & \text{for } u=1,2,\square,N-1\end{cases}$$



Lossy Compression

Transform Coding

Discrete-cosine basis functions for $N = 4$. The origin of each block is at its top left.



Lossy Compression

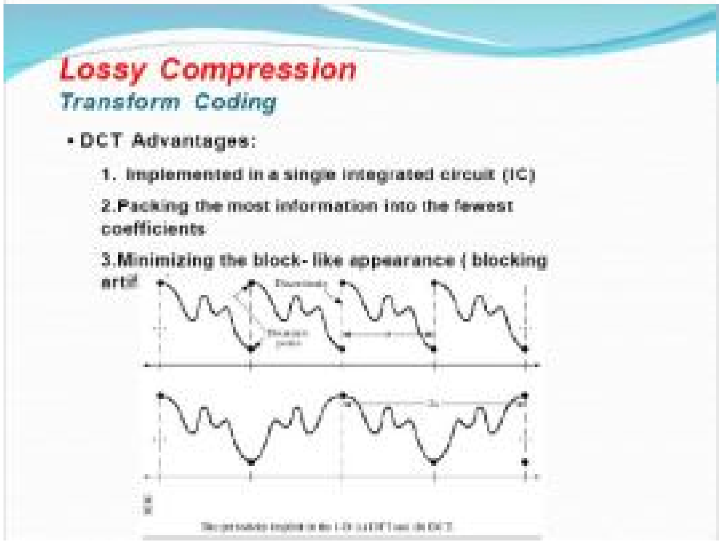
Transform Coding

1.Dividing the image into sub- images of size 8x8

2.Representing each sub-image using one of the transforms

3.Truncating 50% of the resulting coefficients

4.Taking the inverse Transform of the truncated coefficients

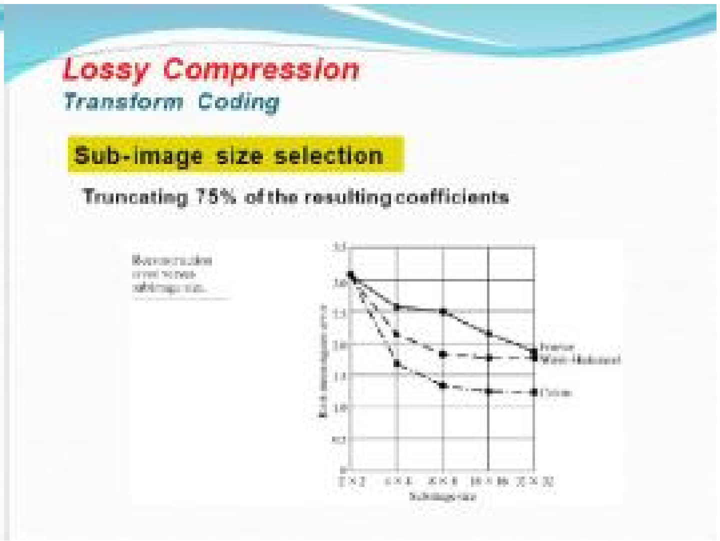


Lossy Compression

Transform Coding

- DCT Advantages:
  1. Implemented in a single integrated circuit (IC)
  2. Packing the most information into the fewest coefficients
  3. Minimizing the block- like appearance (blocking artif

The periodicity implicit in the 1-D (a) DFT and (b) DCT.



Lossy Compression

Transform Coding

Sub-image size selection

Truncating 75% of the resulting coefficients

Reconstruction error versus subimage size.

Lossy Compression

Transform Coding

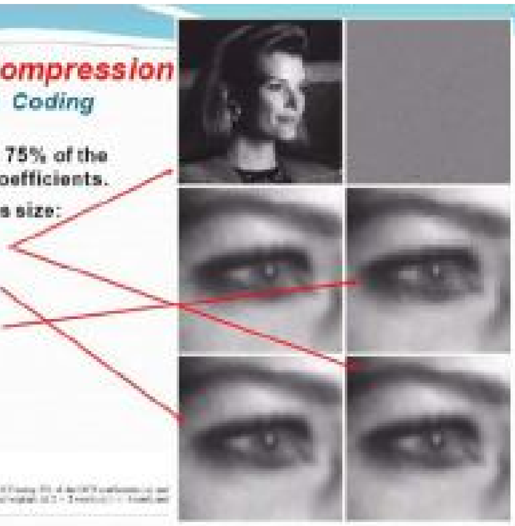
Truncating 75% of the resulting coefficients.

Sub- images size:

8x8

4x4

2x2



[illegible]

Lossy Compression

Transform Coding- Bit allocation

(a)(b)(c)(d) A typical (a) zonal mask, (b) zonal bit allocation, (c) threshold mask, and (d) thresholded coefficient ordering sequence. Shading highlights the coefficients that are retained.

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

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

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

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Lossy Compression

Transform Coding- Bit allocation

- Zonal coding
  1. Fixed number of bits / coefficient
     - Coefficients are normalized by their standard deviations and uniformly quantized
  2. Fixed number of bits is distributed among the coefficients unequally.

     -A quantizer such as an optimal Lloyed- Max is designed for each coeff.:

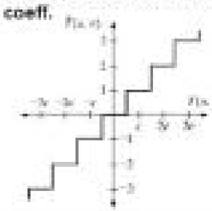
     - DC coeff. Is modeled by Rayleigh density func.
     - The remaining coeff. Are modeled by Laplcian or Gaussian

Lossy Compression

Transform Coding- Bit allocation

- Threshold coding
  1. Single global threshold
  2. Different threshold for each subimage (N- Largest coding)
  3. Threshold can be varied as a function of the location of each coeff.

(a) A threshold coding quantization curve [see Eq. (8.5-40)]. (b) A typical normalization matrix.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Image Compression Standards

Why Do We Need International Standards?

- International standardization is conducted to achieve inter-operability .
  - Only syntax and decoder are specified.
  - Encoder is not standardized and its optimization is left to the manufacturer.
- Standards provide state-of-the-art technology that is developed by a group of experts in the field.
  - Not only solve current problems, but also anticipate the future application requirements.
- Most of the standards are sanction by the International Standardization Organization (ISO) and the Consultative Committee of the International Telephone and Telegraph (CCITT)

Image Compression Standards

Binary Image Compression Standards

CCITT Group 3 and 4

- They are designed as FAX coding methods.
- The Group 3 applies a non-adaptive 1-D run length coding and optionally 2-D manner.
- Both standards use the same non-adaptive 2-D coding approach, similar to RAC technique.
- They sometime result in data expansion. Therefore, the Joint Bilevel Imaging Group (JBIG), has adopted several other binary compression standards, JBIG1 and JBIG2.

Image Compression Standards

Continues Tone Still Image Comp.

What Is JPEG?

- "Joint Photographic Expert Group". Voted as international standard in 1992.
- Works with color and grayscale images, e.g., satellite, medical, ...
- Lossy and lossless

Image Compression Standards

Continues Tone Still Image Comp. - JPEG

- First generation JPEG uses DCT+Run length Huffman entropy coding.
- Second generation JPEG (JPEG2000) uses wavelet transform + bit plane coding + Arithmetic entropy coding.

Image Compression Standards

Continues Tone Still Image Comp. - JPEG

- Still-image compression standard
- Has 3 lossless modes and lossymode
  - sequential baseline encoding
    - encode in one scan
    - input &output data precision is limited to 8 bits, while quantized DCT values are restricted to 11bits
  - progressive encoding
  - hierarchical encoding
  - lossless encoding
- Can achieve compression ratios of up-to 20 to 1without noticeable reduction in image quality

Image Compression Standards

Continues Tone Still Image Comp. - JPEG

- Work well for continuous tone images, but not good for cartoons or computer generatedimages.
- Tend to filter out high frequency data.
- Can specify a quality level(Q)
  - with too low Q, resulting images may contain blocky, contouring and ringing structures.
- 5 steps of sequential baseline encoding
  - transform image to luminance/chrominancespace (YCbCr)
  - reduce the color components (optional)
  - partition image into 8x8 pixel blocks and perform DCT on each block
  - quantize resulting DCT coefficients
  - variable length code the quantizedcoefficients

Image Compression Standards

JPEG Encoding

Original

JPEG 27:1

Image Compression Standards

Video Compression Standards

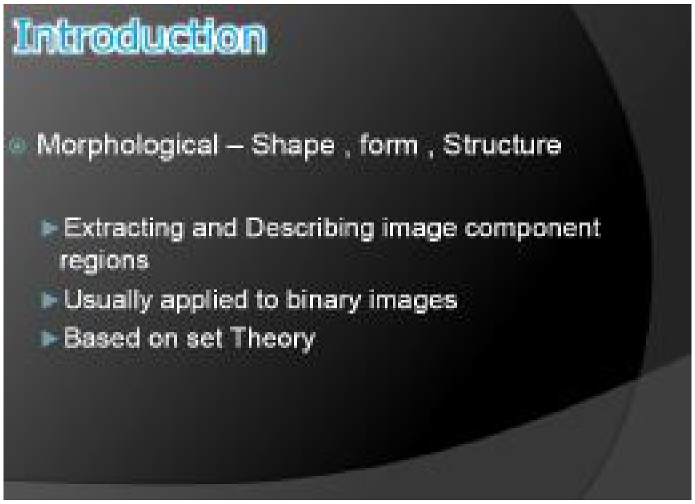
Video compression standards:

1. Video teleconferencing standards
   - H.261 (Px64)
   - H.262
   - H.263 (10 to 30 kbit/s)
   - H.320 (ISDN bandwidth)
2. Multimedia standards
   - MPEG-1 (1.5 Mbit/s)
   - MPEG-2 (2-10 Mbit/s)
   - MPEG-4 (5 to 64 kbit/s for mobile and PSTN and uo to 4 Mbit/s for TV and film application)

Unit-V

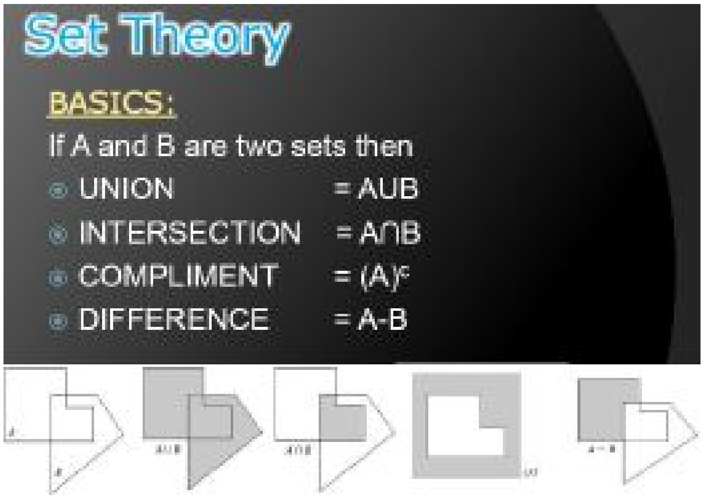
Morphological Image Processing – Preliminaries – Dilation and Erosion – Opening and Closing – The Hit-or-Miss Transformation-Representation – Boundary Descriptors – Regional Descriptors – Use of Principal Components for Description – Relational Descriptors – Applications of Image Processing – Image Watermarking – Fingerprint Recognition – Iris Recognition.

MORPHOLOGICAL IMAGE PROCESSING:



Introduction

- Morphological – Shape , form , Structure
  - Extracting and Describing image component regions
  - Usually applied to binary images
  - Based on set Theory

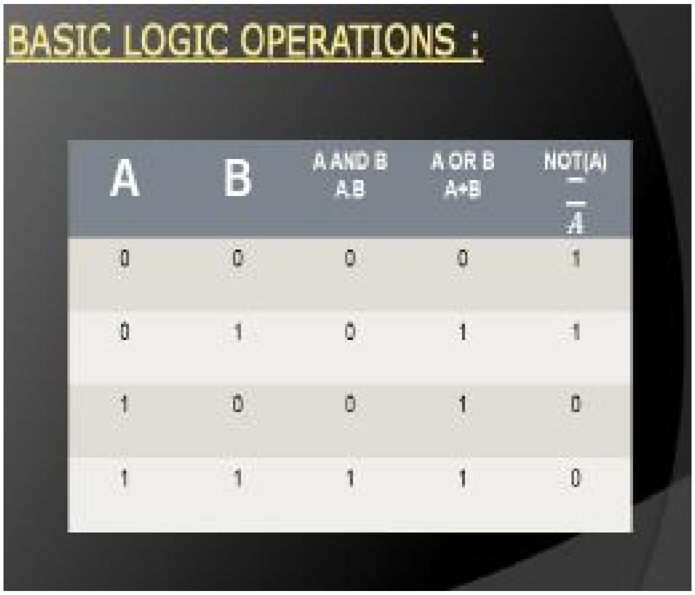


Set Theory

BASICS:

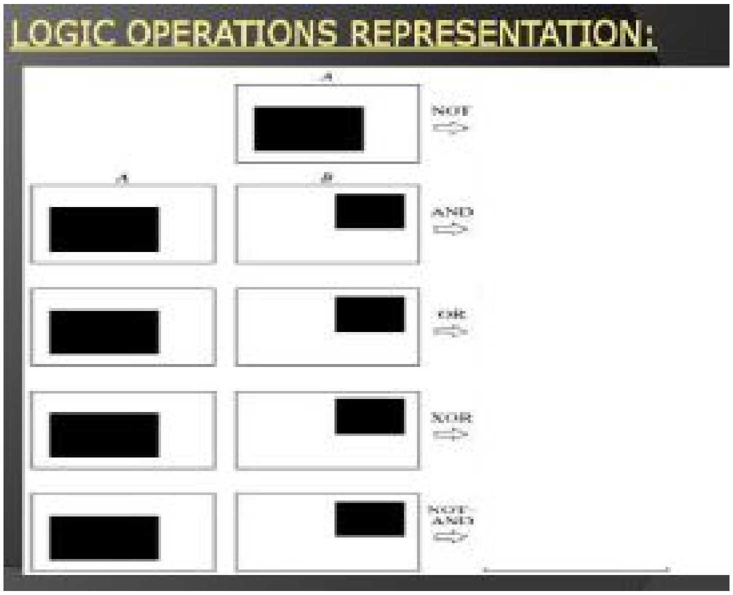
If A and B are two sets then

- UNION = AUB
- INTERSECTION = A∩B
- COMPLIMENT = (A)^c
- DIFFERENCE = A-B

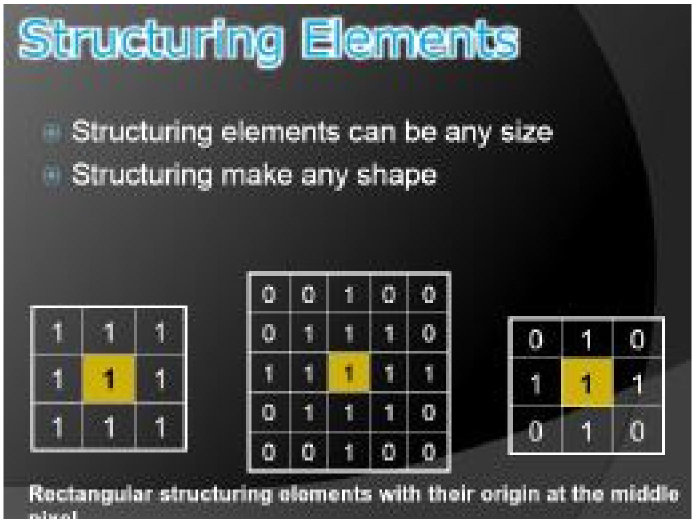


BASIC LOGIC OPERATIONS :

A	B	A AND B A.B	A OR B A+B	NOT(A) $\bar{A}$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



LOGIC OPERATIONS REPRESENTATION:



Structuring Elements

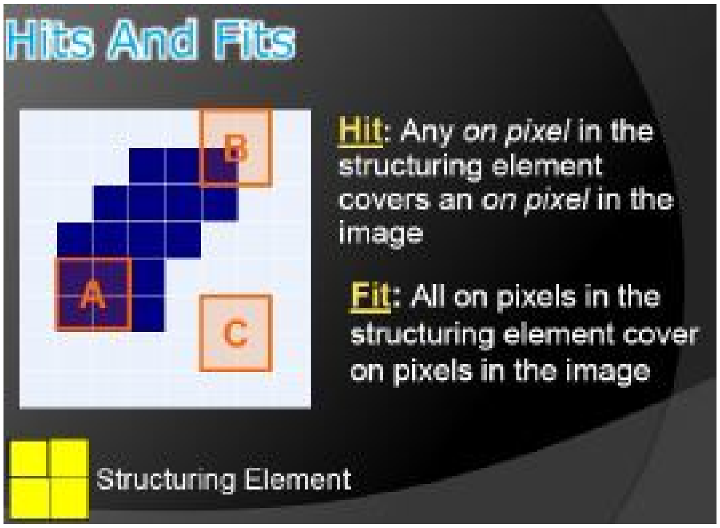
- Structuring elements can be any size
- Structuring make any shape

1	1	1
1	1	1
1	1	1

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

0	1	0
1	1	1
0	1	0

Rectangular structuring elements with their origin at the middle pixel

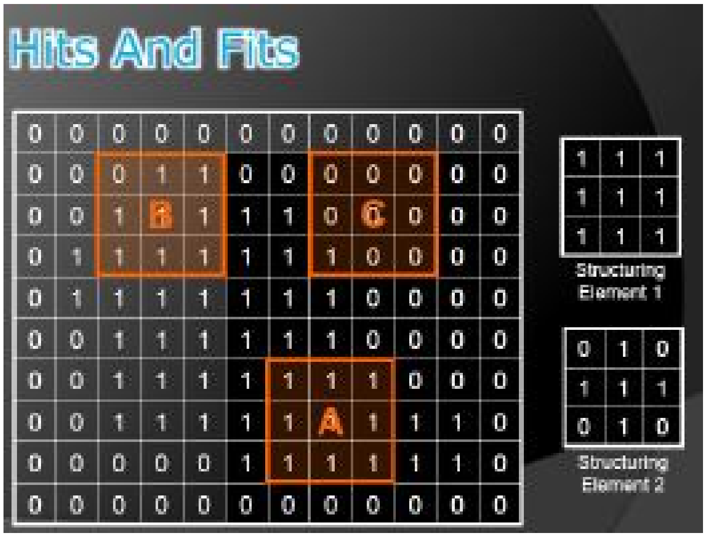


Hits And Fits

Hit: Any *on pixel* in the structuring element covers an *on pixel* in the image

Fit: All on pixels in the structuring element cover on pixels in the image

Structuring Element



Hits And Fits

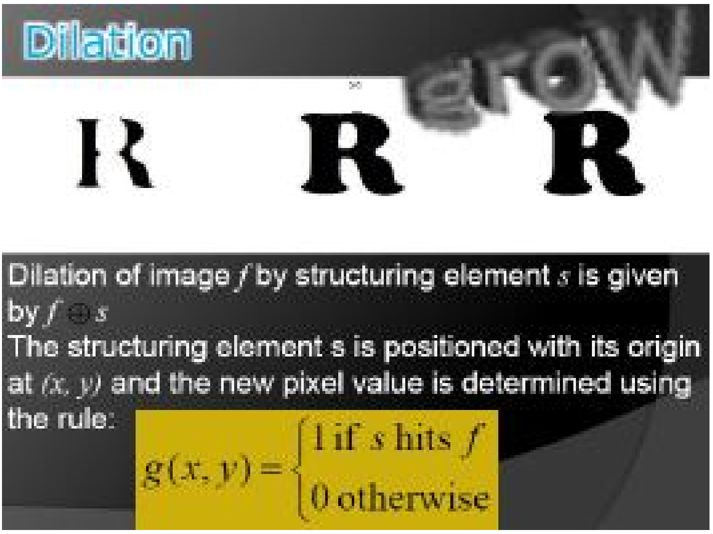
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	B	1	1	1	0	C	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	A	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

Structuring Element 1

0	1	0
1	1	1
0	1	0

Structuring Element 2

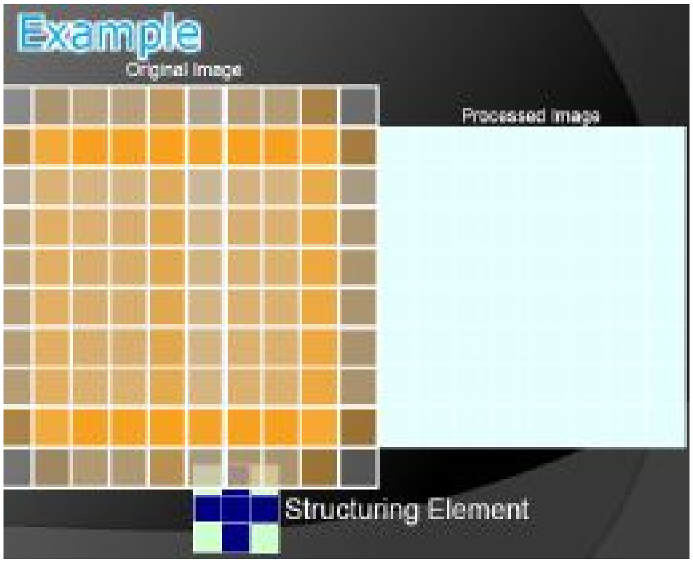


Dilation

Dilation of image f by structuring element s is given by $f \oplus s$

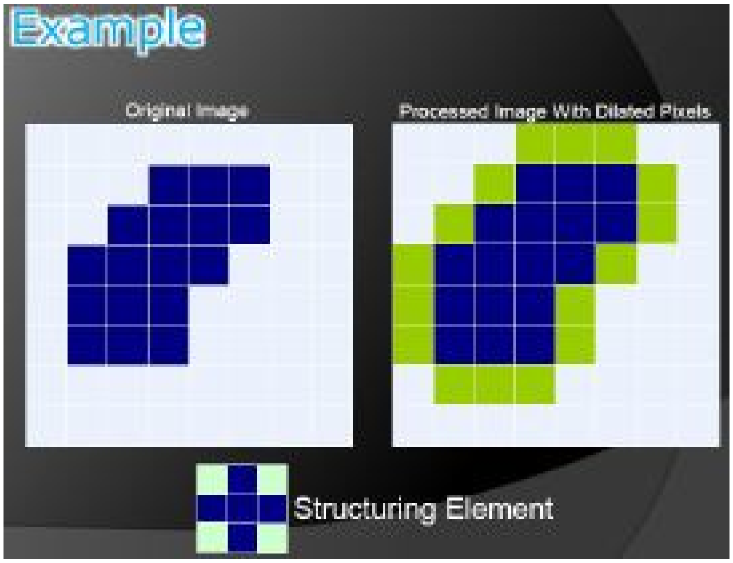
The structuring element s is positioned with its origin at (x, y) and the new pixel value is determined using the rule:

$$g(x, y) = \begin{cases} 1 \text{ if } s \text{ hits } f \\ 0 \text{ otherwise} \end{cases}$$



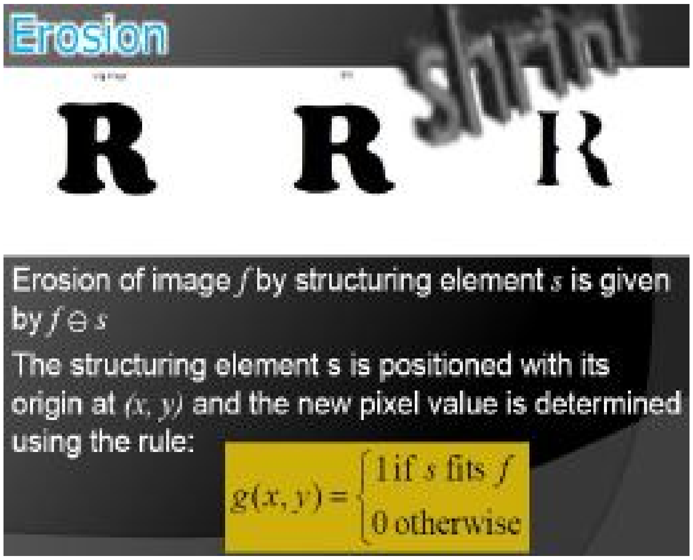
Example

Structuring Element



Example

Structuring Element

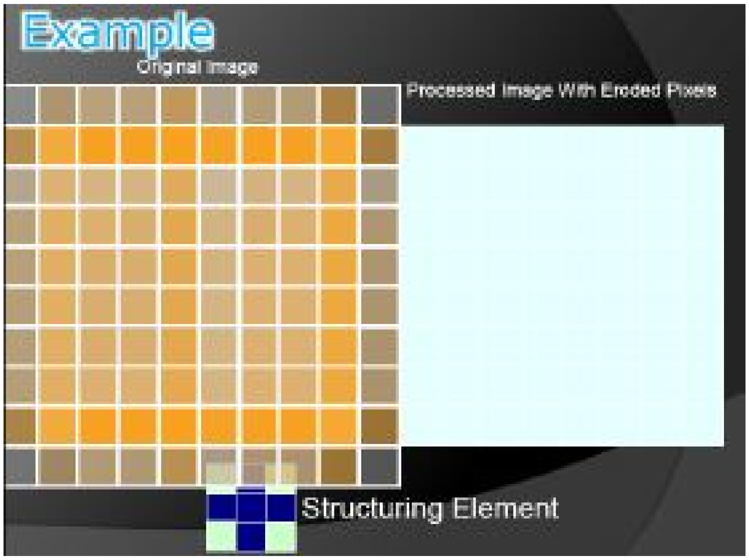


Erosion

Erosion of image f by structuring element s is given by $f \ominus s$

The structuring element s is positioned with its origin at (x, y) and the new pixel value is determined using the rule:

$$g(x,y)=\begin{cases}1 \text{ if } s \text{ fits } f\\ 0 \text{ otherwise}\end{cases}$$

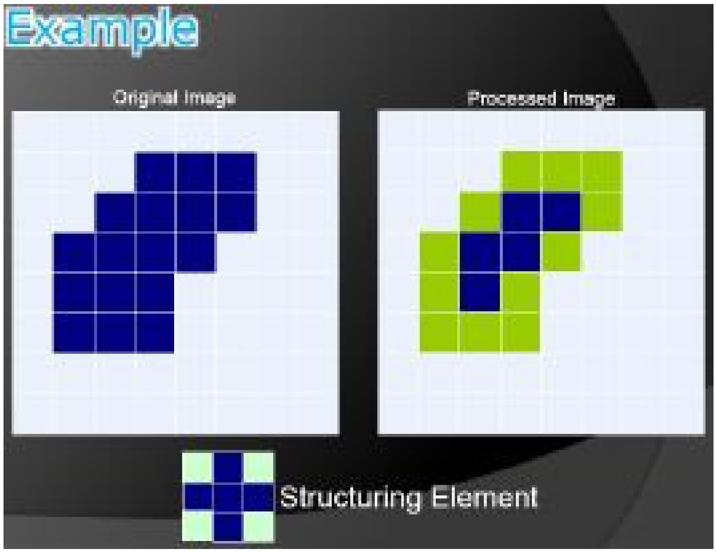


Example

Original Image

Processed Image With Eroded Pixels

Structuring Element

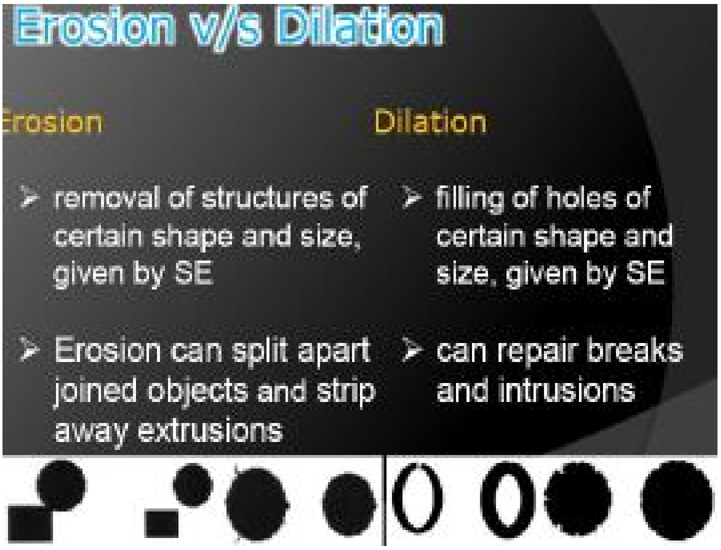


Example

Original Image

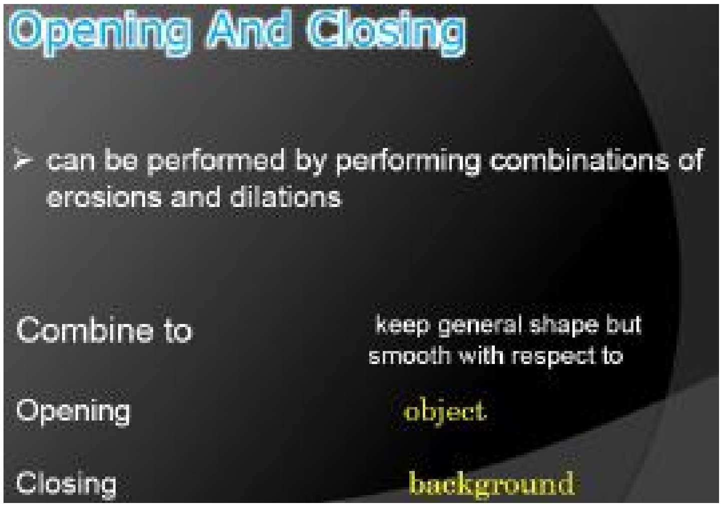
Processed Image

Structuring Element



Erosion v/s Dilation

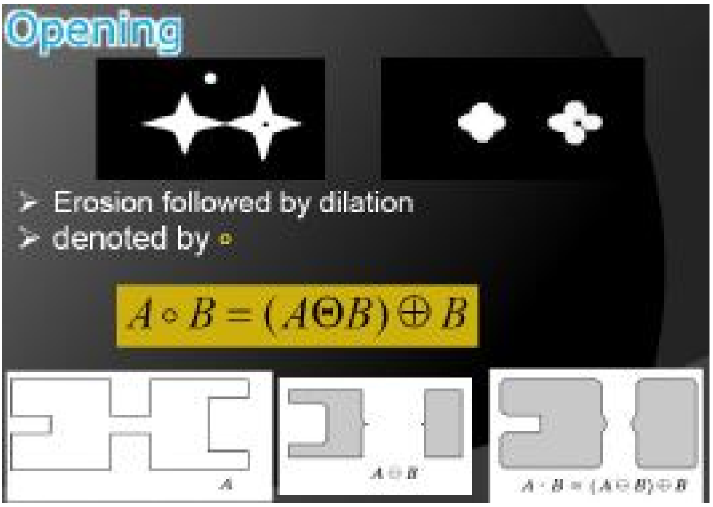
Erosion	Dilation
removal of structures of certain shape and size, given by SE	filling of holes of certain shape and size, given by SE
Erosion can split apart joined objects and strip away extrusions	can repair breaks and intrusions



Opening And Closing

- can be performed by performing combinations of erosions and dilations

Combine to	keep general shape but smooth with respect to
Opening	object
Closing	background

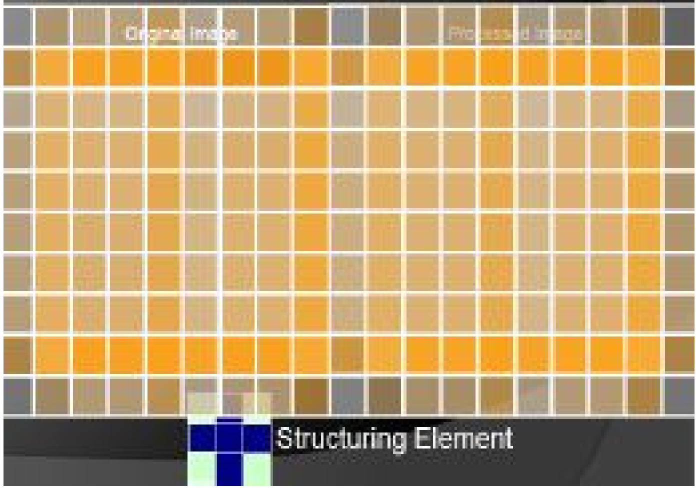


Opening

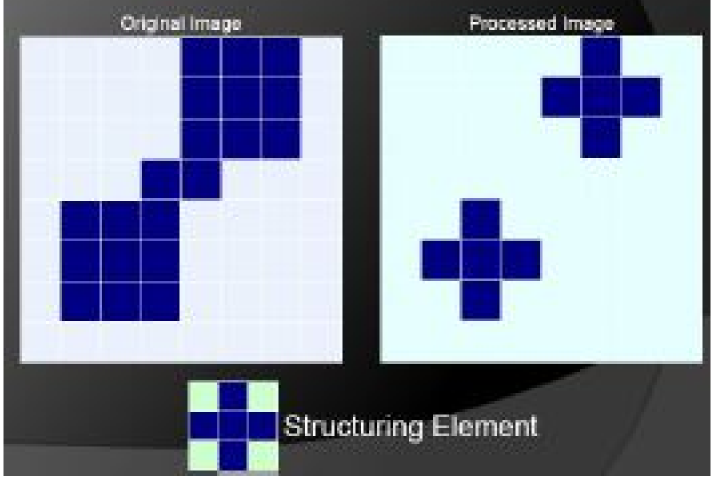
- Erosion followed by dilation
- denoted by ∘

$$A \circ B = (A \Theta B) \oplus B$$

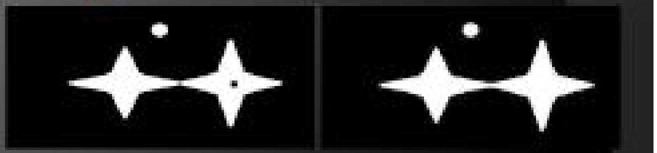
Example



Example

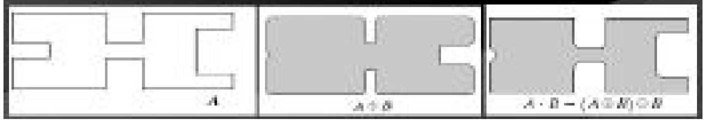


Closing

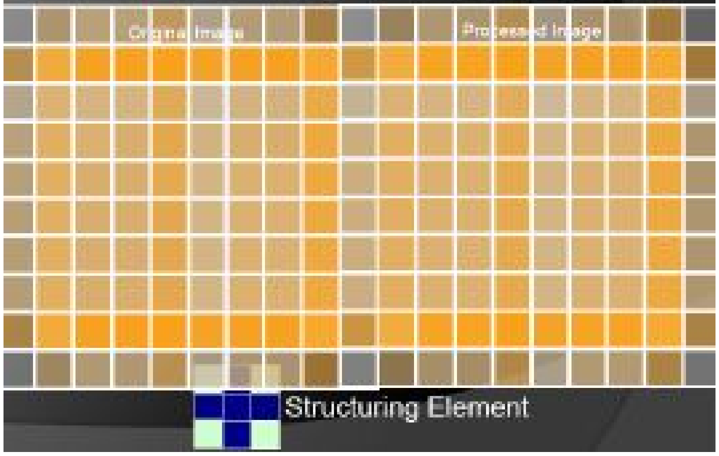


- Dilation followed by erosion
- denoted by •

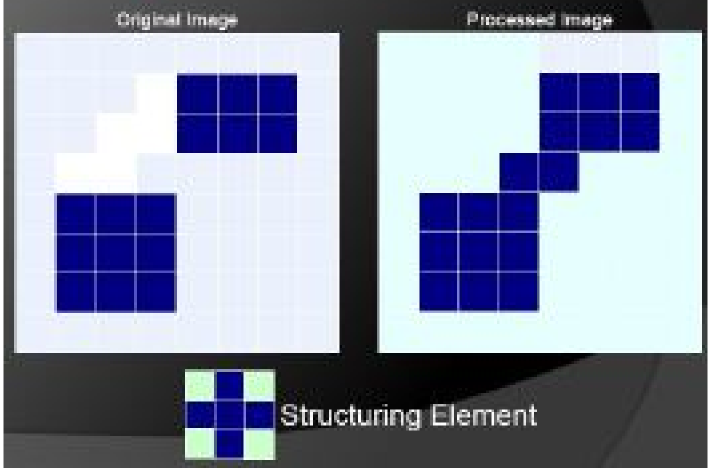
$$f \bullet s = (f \oplus s) \ominus s$$



Example



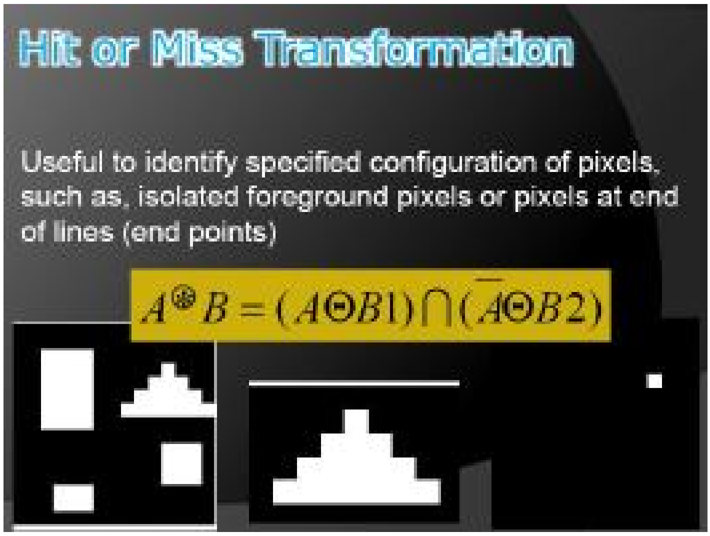
Example



Opening V/S Closing

Opening	Closing
A°B is a subset (subimage) of A	A is a subset (subimage) of A•B
If C is a subset of D, then C °B is a subset of D °B	If C is a subset of D, then C •B is a subset of D •B
(A °B) °B = A °B	(A •B) •B = A •B

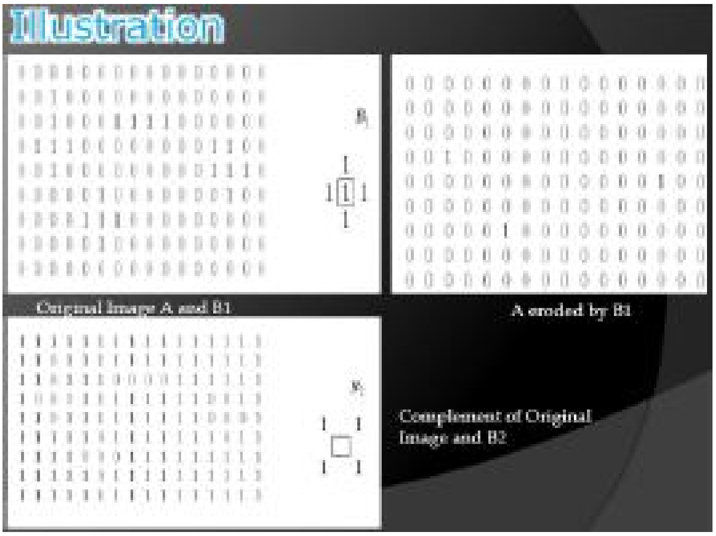
Note: repeated openings/closings has no effect!



Hit or Miss Transformation

Useful to identify specified configuration of pixels, such as, isolated foreground pixels or pixels at end of lines (end points)

$$A \circledast B = (A\Theta B1)\cap(\overline{A}\Theta B2)$$



Illustration

```
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 1 0 0 0 1 1 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0
0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
0 0 0 0 1 1 1 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
```

B_1

$$\begin{array}{ccc} & 1 & \\ 1 & \boxed{1} & 1 \\ & 1 & \end{array}$$

Original Image A and B1

```
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 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 1 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 0 0
```

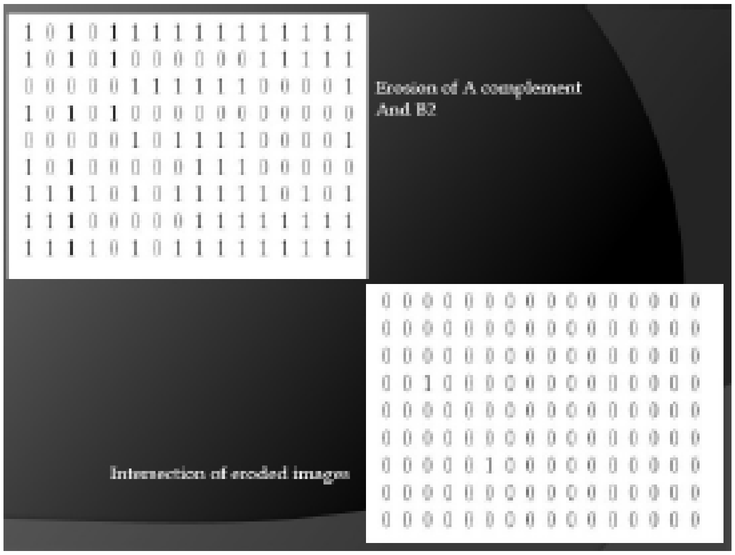
A eroded by B1

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1
1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 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 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

B_2

$$\begin{array}{ccc} 1 & & 1 \\ & \square & \\ 1 & & 1 \end{array}$$

Complement of Original Image and B2



```
1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1
1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1
1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1
```

Erosion of A complement And B2

Intersection of eroded images

```
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 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 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 0 0
```

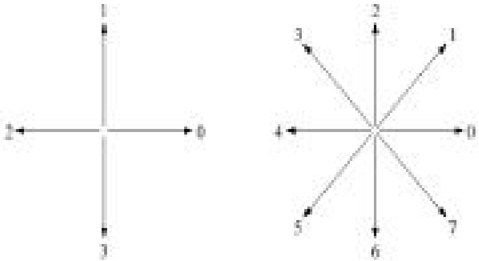
REPRESENTATION:

- Image regions (including segments) can be represented by either the border or the pixels of the region. These can be viewed as external or internal characteristics, respectively.

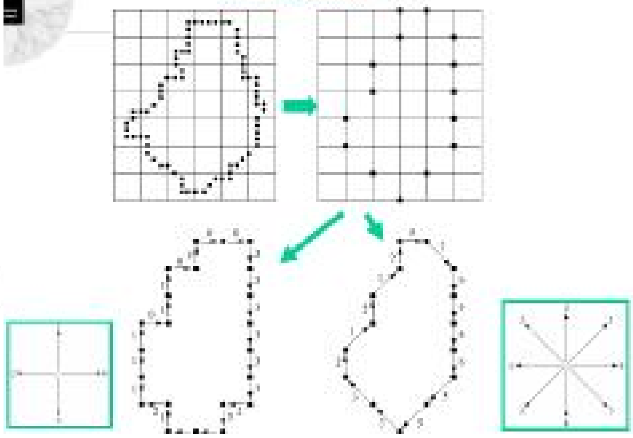
Representation

- Boundary(Border) following
- Chain codes
- Polygonal approximations using Minimum-perimeter polygons
- Other polygonal approximation approaches
- Signatures
- Boundary segments
- Skeletons

- Chain codes: represent a boundary of a connected region.
- Also called as freeman chain codes.
- Eg: 0321, 03322101, 0330322111 etc.,

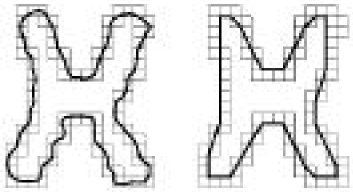


Representation Chain Codes

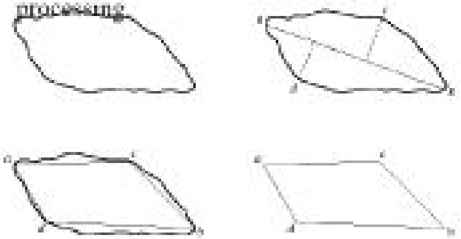


- Chain codes can be based on either 4-connectedness or 8-connectedness.
- The first difference of the chain code:
  - This difference is obtained by counting the number of direction changes (in a counterclockwise direction)
  - For example, the first difference of the 4-direction chain code 10103322 is 3133030.
- Assuming the first difference code represent a closed path, rotation normalization can be achieved by circularly shifting the number of the code so that the list of numbers forms the smallest possible integer.
- Size normalization can be achieved by adjusting the size of the resampling grid.

- Polygonal approximations: to represent a boundary by straight line segments, and a closed path becomes a polygon.
- The number of straight line segments used determines the accuracy of the approximation.
- Only the minimum required number of sides necessary to preserve the needed shape information should be used (Minimum perimeter polygons).
- A larger number of sides will only add noise to the model.



- Other polygonal approximation approaches: (Merging and splitting)
  - Merging and splitting are often used together to ensure that vertices appear where they would naturally in the boundary.
  - A least squares criterion to a straight line is used to stop the processing.



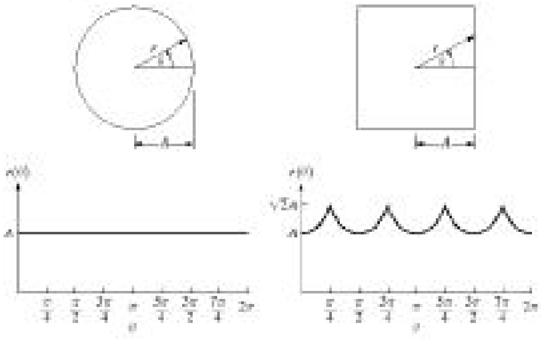
a b
c d

(a) Original boundary. (b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.

- The idea behind a signature is to convert a two dimensional boundary into a representative one dimensional function.

a b

Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A\sec\theta$ for $0 \le \theta \le \pi/4$ and $r(\theta) = A\csc\theta$ for $\pi/4 < \theta \le \pi/2$.

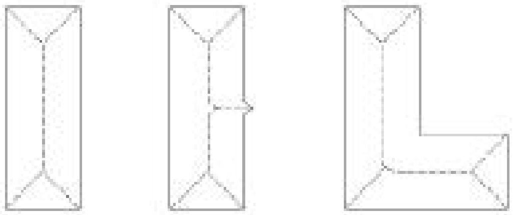


Signature

- Signatures are invariant to location, but will depend on rotation and scaling.
  - Starting at the point farthest from the reference point or using the major axis of the region can be used to decrease dependence on rotation.
  - Scale invariance can be achieved by either scaling the signature function to fixed amplitude or by dividing the function values by the standard deviation of the function.

Skeletons

- Skeletons: produce a one pixel wide graph that has the same basic shape of the region, like a stick figure of a human. It can be used to analyze the geometric structure of a region which has bumps and "arms".



a b c

Medial axes (dashed) of three simple regions.

- Step 1: Flag a contour point p_1 for deletion if the following conditions are satisfied

  (a) $2 \leq N(p_1) \leq 6$ (c) $p_2 \cdot p_4 \cdot p_6 = 0$

  (b) $T(p_1) = 1$ (d) $p_4 \cdot p_6 \cdot p_8 = 0$

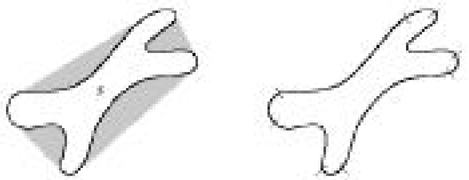
p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

0	0	1
1	p_1	0
1	0	1

Illustration of conditions (a) and (b) in Eq. (11.1-1). In this case $N(p_1) = 4$ and $T(p_1) = 3$.

Boundary Segments

- Boundary segments: decompose a boundary into segments.
- Use of the convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.



a b

(a) A region, S, and its convex deficiency (shaded). (b) Partitioned boundary.

- Before a thinning algorithm:
  - A contour point is any pixel with value 1 and having at least one 8-neighbor valued 0.
  - Let

$$N(p_1) = p_2 + p_3 + \ldots + p_8 + p_9$$

$T(p_1)$: the number of 0-1 transitio ns in the ordered sequence

$$p_2, p_3, \ldots, p_8, p_9, p_2$$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

- Step 2: Flag a contour point p_1 for deletion again. However, conditions (a) and (b) remain the same, but conditions (c) and (d) are changed to

  (d') $p_2 \cdot p_6 \cdot p_8 = 0$

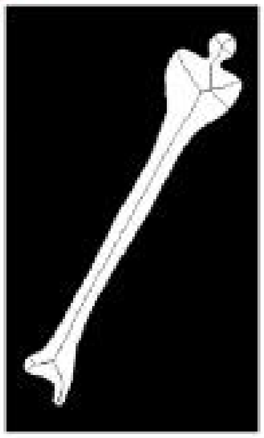
  (c') $p_2 \cdot p_4 \cdot p_8 = 0$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

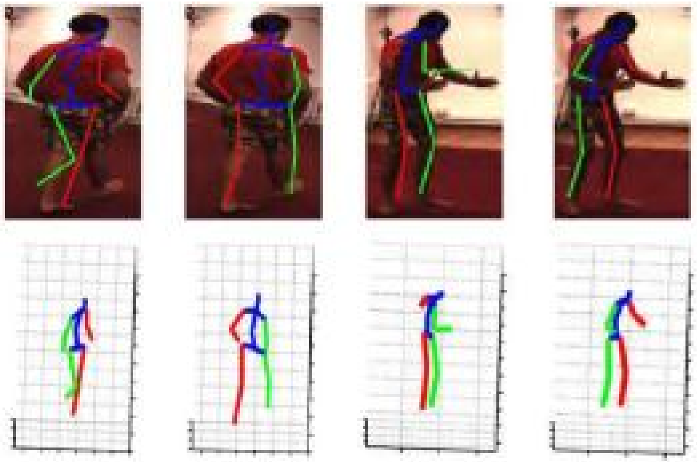
- A thinning algorithm:
  - (1) applying step 1 to flag border points for deletion
  - (2) deleting the flagged points
  - (3) applying step 2 to flag the remaining border points for deletion
  - (4) deleting the flagged points
  - This procedure is applied iteratively until no further points are deleted.

Representation Skeletons: Example

- One application of skeletonization is for character recognition.
- A letter or character is determined by the center-line of its strokes, and is unrelated to the width of the stroke lines.



Human leg bone and skeleton of the region shown superimposed.

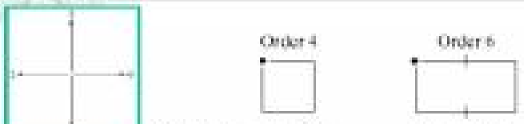


(a) $e^{\chi 2}$-HoG (b) Our (c) $e^{\chi 2}$-HoG (d) Our

Boundary Descriptors

- There are several simple geometric measures that can be useful for describing a boundary.
  - The ***length*** of a boundary: the number of pixels along a boundary gives a rough approximation of its length.
  - ***Curvature***: the rate of change of slope
    - To measure a curvature accurately at a point in a digital boundary is difficult
    - The difference between the slops of adjacent boundary segments is used as a descriptor of curvature at the point of intersection of segments

Shape Numbers

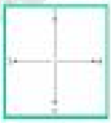


	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

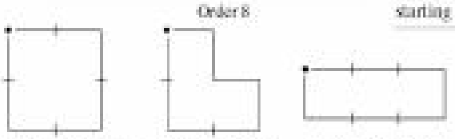
First difference

All shapes of order 4, 6, and 8. The directions are from Fig. 11.1(a), and the dot indicates the starting point.

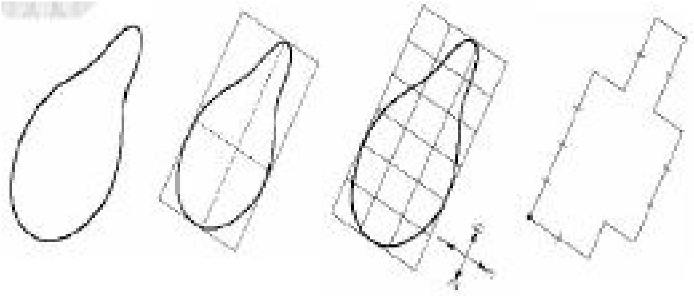
- The ***shape number*** of a boundary is defined as the first difference of smallest magnitude.
- The order *n* of a shape number is defined as the number of digits in its representation.



All shapes of order 4, 6, and 8. The directions are from Fig. 11.1(a), and the dot indicates the starting point.



Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3



Chain code: 0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

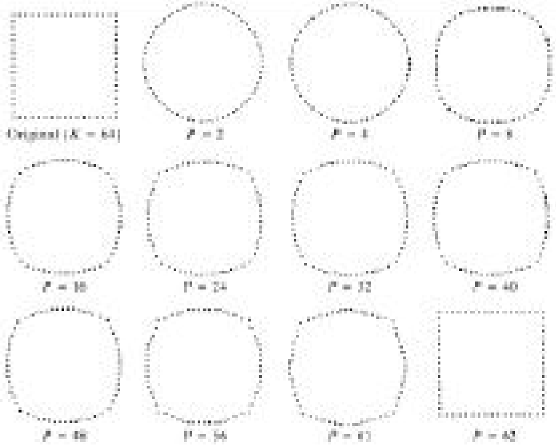
Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape no.: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

Fourier Descriptors

- This is a way of using the Fourier transform to analyze the shape of a boundary.
  - The *x-y* coordinates of the boundary are treated as the real and imaginary parts of a complex number(eg:a+bi).
  - Then the list of coordinates is Fourier transformed using the DFT (chapter 4).
  - The Fourier coefficients are called the Fourier descriptors.
  - The basic shape of the region is determined by the first several coefficients, which represent lower frequencies.
  - Higher frequency terms provide information on the fine detail of the boundary.

FIGURE 11.14 Examples of reconstruction from Fourier descriptors. *P* is the number of Fourier coefficients used in the reconstruction of the boundary.



Statistical Moments

- Moments are statistical measures of data.
  - They come in integer orders.
  - Order 0 is just the number of points in the data.
  - Order 1 is the sum and is used to find the average.
  - Order 2 is related to the variance, and order 3 to the skew of the data.
  - Higher orders can also be used, but don't have simple meanings.

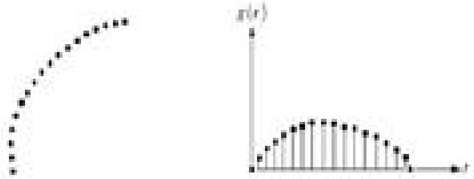
- Let r be a random variable, and $g(r_i)$ be normalized (as the probability of value r_i occurring), then the moments are

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i)$$

$$\text{where } m = \sum_{i=0}^{K-1} r_i g(r_i)$$

a b

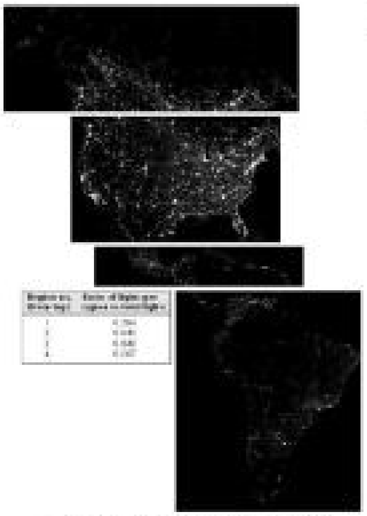
(a) Boundary segment. (b) Representation as a 1-D function.



Regional Descriptors

- Some simple descriptors
  - The area of a region: the number of pixels in the region
  - The perimeter of a region: the length of its boundary
  - The compactness of a region: (perimeter)^2/area
  - The mean and median of the gray levels
  - The minimum and maximum gray-level values
  - The number of pixels with values above and below the mean

Regional Descriptors Example

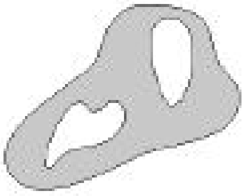


Region no. (from top)	Ratio of lights per region to total lights
1	0.204
2	0.640
3	0.049
4	0.107

Infrared images of the Americas at night. (Courtesy of NOAA.)

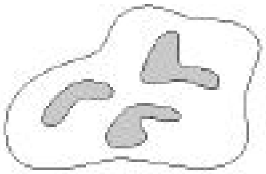
Topological Descriptors

Topological property 1:
the number of holes (H)



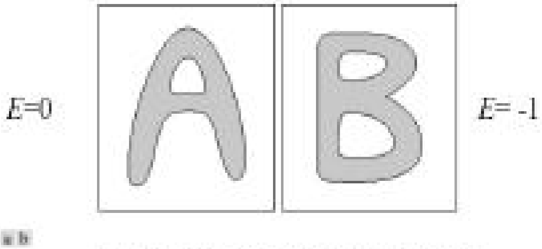
A region with two holes.

Topological property 2:
the number of connected components (C)

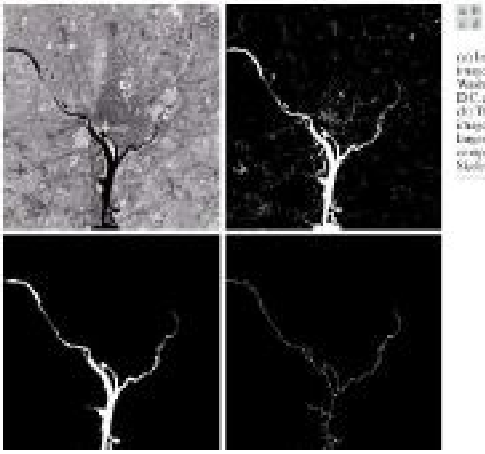


A region with three connected components.

Topological property 3:
Euler number: the number of connected components subtract the number of holes
$E = C - H$



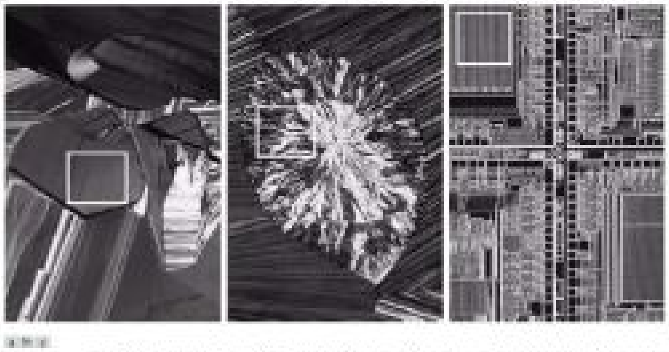
Regions with Euler number equal to 0 and −1, respectively.



(a) Infrared image of the Washington, D.C. area. (b) Thresholded image. (c) The largest connected component of (b). (d) Skeleton of (c).

Topological property 4: the largest connected component.

Texture



The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)

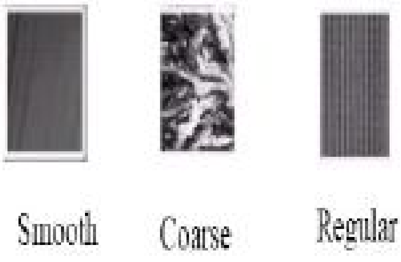
- Texture is usually defined as the smoothness or roughness of a surface.
- In computer vision, it is the visual appearance of the uniformity or lack of uniformity of brightness and color.
- There are two types of texture: random and regular.
  - Random texture cannot be exactly described by words or equations; it must be described statistically. The surface of a pile of dirt or rocks of many sizes would be random.
  - Regular texture can be described by words or equations or repeating pattern primitives. Clothes are frequently made with regularly repeating patterns.
  - Random texture is analyzed by statistical methods.
  - Regular texture is analyzed by structural or spectral (Fourier) methods.

Statistical Approaches

- Let z be a random variable denoting gray levels and let $p(z_i)$, i=0,1,…,L-1, be the corresponding histogram, where L is the number of distinct gray levels.
  - The nth moment of z:

$$\mu_n(z)=\sum_{i=0}^{L-1}(z_i-m)^n p(z_i) \quad \text{where} \quad m=\sum_{i=0}^{L-1} z_i p(z_i)$$

  - The measure R: $R=1-\dfrac{1}{1+\sigma^2(z)}$
  - The uniformity: $U=\sum_{i=0}^{L-1} p^2(z_i)$
  - The average entropy: $e=-\sum_{i=0}^{L-1} p(z_i)\log_2 p(z_i)$

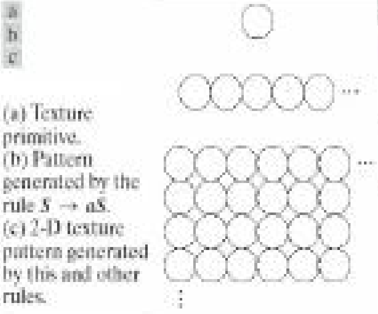


Texture measures for the subimages shown in

Texture	Mean	Standard deviation	R (normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

Structural Approaches

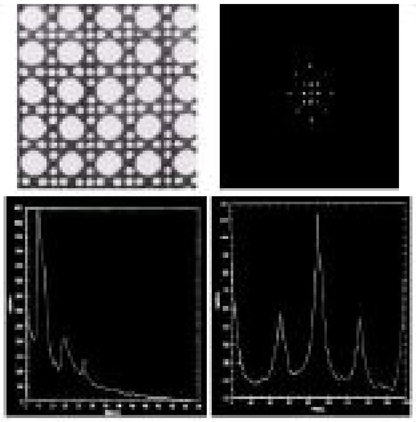
- Structural concepts:
  - Suppose that we have a rule of the form $S\rightarrow aS$, which indicates that the symbol S may be rewritten as aS.
  - If a represents a circle [Fig. 11.23(a)] and the meaning of "circle to the right" is assigned to a string of the form $aaaa\ldots$



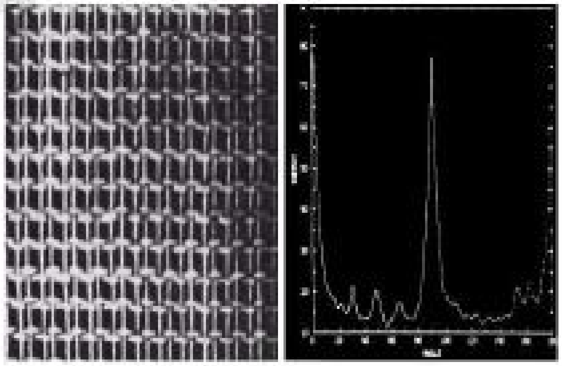
(a) Texture primitive. (b) Pattern generated by the rule $S\rightarrow aS$. (c) 2-D texture pattern generated by this and other rules.

Spectral Approaches

- For non-random primitive spatial patterns, the 2-dimensional Fourier transform allows the patterns to be analyzed in terms of spatial frequency components and direction.
- It may be more useful to express the spectrum in terms of polar coordinates, which directly give direction as well as frequency.
- Let $S(r,\theta)$ is the spectrum function, and r and θ are the variables in this coordinate system.
  - For each direction θ, $S(r,\theta)$ may be considered a 1-D function $S_\theta(r)$.
  - For each frequency r, $S_r(\theta)$ is a 1-D function.
  - A global description: $S(r)=\sum_{\theta=0}^{\pi} S_\theta(r)$ $\quad S(\theta)=\sum_{r=1}^{R_0} S_r(\theta)$



a b
c d
e f
(a) Image showing periodc texture. (b) Spectrum. (c) Plot of $S(r)$. (d) Plot of $S(\theta)$. (e) Another image with a different type of periodic texture. (f) Plot of $S(\theta)$. (Courtesy of Dr. Dragana Brzakovic, University of Tennessee.)



a b
c d
e f
FIGURE 11.24 (a) Image showing periodic texture. (b) Spectrum. (c) Plot of $S(r)$. (d) Plot of $S(\theta)$. (e) Another image with a different type of periodic texture. (f) Plot of $S(\theta)$. (Courtesy of Dr. Dragana Brzakovic, University of Tennessee.)

Moments of Two-Dimensional Functions

- For a 2-D continuous function $f(x,y)$, the moment of order $(p+q)$ is defined as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy \quad \text{for} \quad p,q = 1,2,3,\ldots$$

- The central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,dx\,dy$$

where $\bar{x} = \dfrac{m_{10}}{m_{00}}$ and $\bar{y} = \dfrac{m_{01}}{m_{00}}$

- If $f(x,y)$ is a digital image, then

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

- The central moments of order up to 3 are

$$\mu_{00} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^0 f(x,y) = \sum_x \sum_y f(x,y) = m_{00}$$

$$\mu_{10} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^0 f(x,y) = m_{10} - \frac{m_{10}}{m_{00}}(m_{00}) = 0$$

$$\mu_{01} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^1 f(x,y) = m_{01} - \frac{m_{01}}{m_{00}}(m_{00}) = 0$$

$$\mu_{11} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^1 f(x,y) = m_{11} - \frac{m_{10}m_{01}}{m_{00}}$$

$$= m_{11} - \bar{x}m_{01} = m_{11} - \bar{y}m_{10}$$

The central moments of order up to 3 are

$$\mu_{20} = \sum_x \sum_y (x-\bar{x})^2 (y-\bar{y})^0 f(x,y) = m_{20} - \bar{x}m_{10}$$

$$\mu_{02} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^2 f(x,y) = m_{02} - \bar{y}m_{01}$$

$$\mu_{21} = \sum_x \sum_y (x-\bar{x})^2 (y-\bar{y})^1 f(x,y) = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}m_{01}$$

$$\mu_{12} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^2 f(x,y) = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}m_{10}$$

$$\mu_{30} = \sum_x \sum_y (x-\bar{x})^3 (y-\bar{y})^0 f(x,y) = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10}$$

$$\mu_{03} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^3 f(x,y) = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}$$

The normalized central moments are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

where $\gamma = \dfrac{p+q}{2} + 1 \quad \text{for} \quad p+q = 2,3,\ldots.$

- A seven invariant moments can be derived from the second and third moments:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

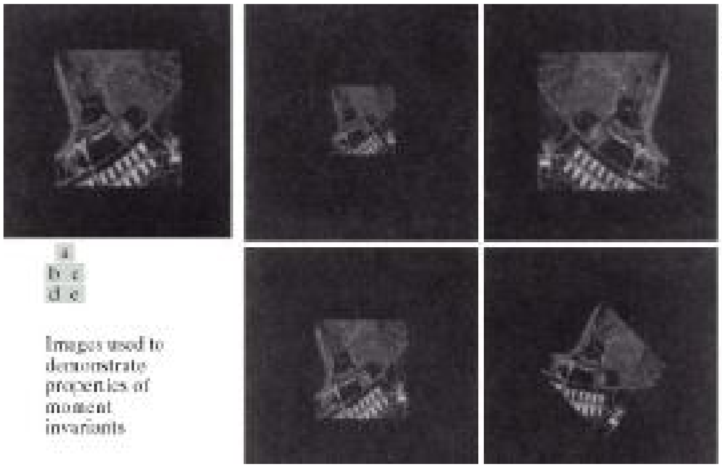
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right]$$
$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

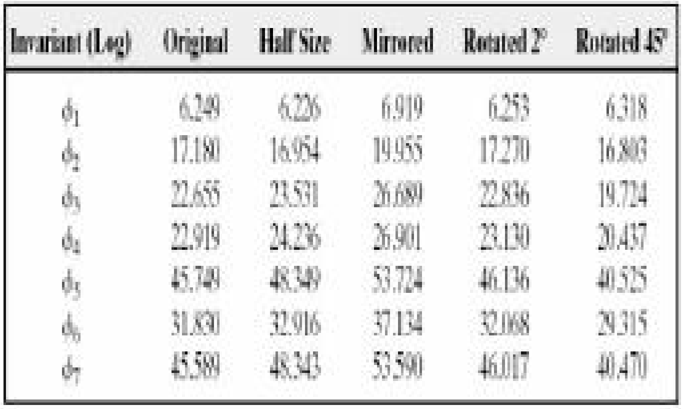
- $$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$
$$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right]$$
$$+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

- This set of moments is invariant to translation, rotation, and scale change.



Images used to demonstrate properties of moment invariants



Invariant (Log)	Original	Half Size	Mirrored	Rotated 2°	Rotated 45°
ϕ_1	6.249	6.226	6.919	6.253	6.318
ϕ_2	17.180	16.954	19.955	17.270	16.803
ϕ_3	22.655	23.531	26.689	22.836	19.724
ϕ_4	22.919	24.236	26.901	23.130	20.437
ϕ_5	45.749	48.349	53.724	46.136	40.525
ϕ_6	31.830	32.916	37.134	32.068	29.315
ϕ_7	45.589	48.343	53.590	46.017	40.470

USE OF PRINCIPAL COMPONENTS FOR DESCRIPTION:

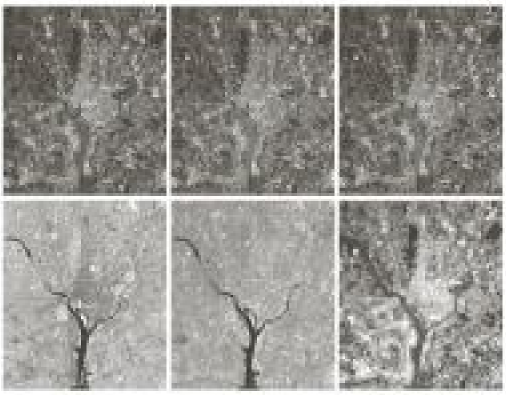
Principal Component Analysis (PCA)

We want to construct a low-dimensional linear subspace that best explains the variation in the components of a multidimensional image:

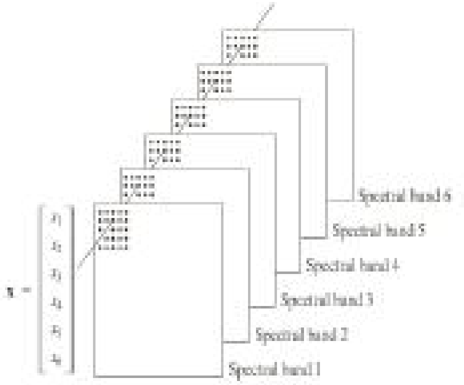
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

n=3 for RGB images or n=6 for LANSAT images.

Multispectral images in the visible blue, visible green, visible red, near infrared, middle infrared and thermal infrared bands.



One n-dimensional vector for each pixel is created.

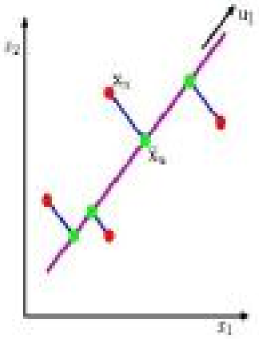


Goal: find a low dimensional representation for the vectors that

- minimizes the projection error (the distance between the initial vectors and their projections)

or equivalently

- maximizes the variance of the projected data.



- Given: N data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ in R^d

- We want to find a new set of features that are linear combinations of the original ones:

$$w(\mathbf{x}_i) = \mathbf{u}^T(\mathbf{x}_i - \mathbf{m}_x), \qquad \mathbf{m}_x = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

- What unit vector $\mathbf{u}$ in R^d captures the most variance of the data?

- We now estimate vector $\mathbf{u}$ maximizing the variance:

$$\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}$$

subject to: $\mathbf{u}^T\mathbf{u} = \|\mathbf{u}\|^2 = 1$

because any multiple of $\mathbf{u}$ maximizes the objective function.

- The Lagrangian is $J(\mathbf{u};\lambda) = \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} + \lambda(1 - \mathbf{u}^T\mathbf{u})$

leading to the solution: $\boldsymbol{\Sigma}\mathbf{u} = \lambda\mathbf{u}$

which is an eigenvector of $\boldsymbol{\Sigma}$. The one maximizing J corresponds to the largest eigenvalue of $\boldsymbol{\Sigma}$.

- Because $\boldsymbol{\Sigma}$ is real, symmetric and positive definite its eigenvalues are non negative and its eigenvectors are orthogonal.

- Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ be the eigenvalues of $\boldsymbol{\Sigma}$ in descending order and

$$\mathbf{A} = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_d^T \end{bmatrix}$$

be a matrix whose rows are the eigenvectors of $\boldsymbol{\Sigma}$.

- The variance of the projected data:

$$\mathrm{var}(w(\mathbf{x}_i)) = \frac{1}{N}\sum_{i=1}^{N} w(\mathbf{x}_i)\,w^T(\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{u}^T\underbrace{(\mathbf{x}_i - \mathbf{m}_x)\big(\mathbf{u}^T(\mathbf{x}_i - \mathbf{m}_x)}_{\text{Projection of data point}}\big)^T$$

$$= \frac{1}{N}\sum_{i=1}^{N}\mathbf{u}^T(\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T\mathbf{u} = \mathbf{u}^T\frac{1}{N}\underbrace{\left[\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T\right]}_{\text{Covariance matrix of data}}\mathbf{u}$$

$$= \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}$$

- The direction that captures the maximum variance of the data is the eigenvector corresponding to the largest eigenvalue of the data covariance matrix.

- The top *k* orthogonal directions that capture the most variance of the data are the *k* eigenvectors corresponding to the *k* largest eigenvalues.

- The transformation that maps $\mathbf{x}$ to the new k-dimensional ($k<d$) subspace (using the first k eigenvectors) is

$$\mathbf{y} = \mathbf{A}_k(\mathbf{x} - \mathbf{m}_x) = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_k^T \end{bmatrix}(\mathbf{x} - \mathbf{m}_x)$$

- Properties of the new subspace:

$$\mathbf{m}_y = \mathbf{0}$$

$$\boldsymbol{\Sigma}_y = \mathbf{A}_k\boldsymbol{\Sigma}\mathbf{A}_k^T = \begin{bmatrix} \lambda_1 & & 0 & 0 \\ & \lambda_1 & & 0 \\ 0 & & \ddots & \\ 0 & 0 & & \lambda_k \end{bmatrix}$$

The components of $\mathbf{y}$ are uncorrelated

- Because the rows of $\mathbf{A}$ are orthonormal $\mathbf{A}^{-1} = \mathbf{A}^T$
- Therefore, we can reconstruct the original data $\mathbf{x}$ from their k-dimensional projections $\mathbf{y}$:

$$\hat{\mathbf{x}} = \mathbf{A}_k^T \mathbf{y} + \mathbf{m}_\mathbf{x}$$

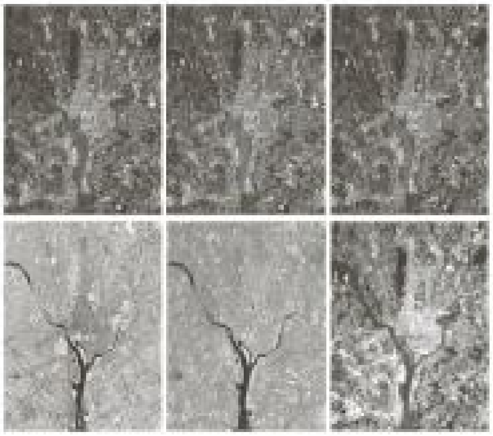
- The mean square reconstruction error is:

$$e = \sum_{j=1}^{d} \lambda_j - \sum_{j=1}^{k} \lambda_j = \sum_{j=k+1}^{d} \lambda_j$$

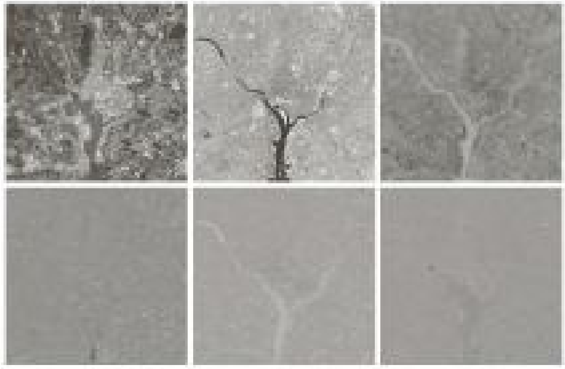
which is zero if we make use of all of the eigenvectors.

- In our example, the eigenvalues of the covariance matrix of the 6-dimensional pixels are:

$\lambda_1 = 10344$
$\lambda_2 = 2966$
$\lambda_3 = 1401$
$\lambda_4 = 203$
$\lambda_5 = 94$
$\lambda_6 = 31$

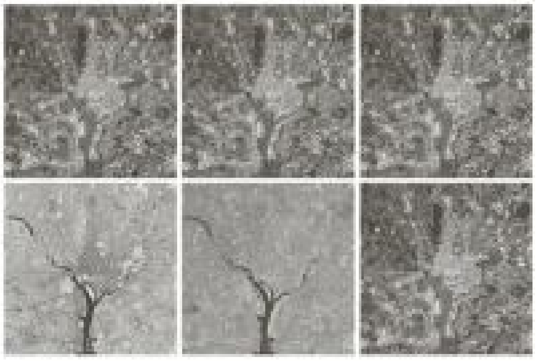


- The six principal component images $\mathbf{y} = \mathbf{A}_6(\mathbf{x} - \mathbf{m}_\mathbf{x})$
- The first two images account for 89% of the total variance.

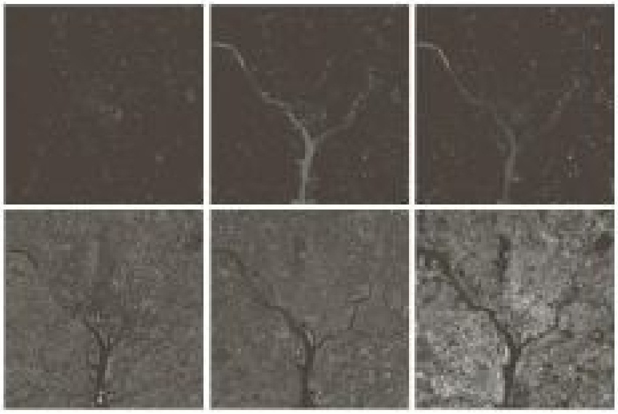


Reconstructed images using the first two principal components

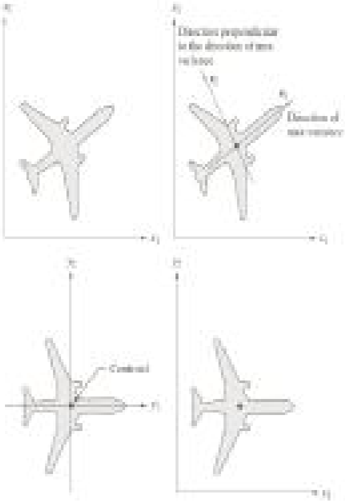
$$\hat{\mathbf{x}} = \mathbf{A}_2^T \mathbf{y} + \mathbf{m}_\mathbf{x}$$



Reconstruction error (enhanced for better visualization).



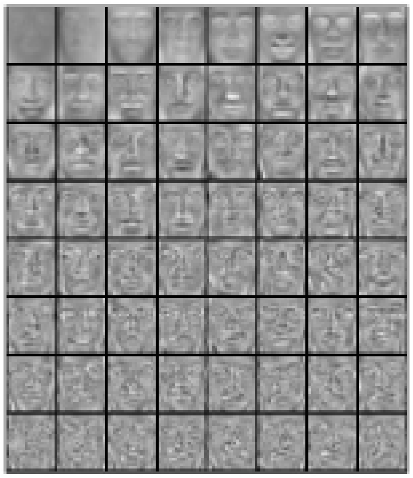
- PCA is a convenient way to normalize 2D boundaries with respect to rotation and translation.
- Shifting by $(y_{1\min}, y_{2\min})$ makes the coordinate non negatives.
- Further division by λ_1 and λ_2 normalizes the scale.


Direction perpendicular to the direction of max variance
Direction of max variance
Centroid

- Application: eigenfaces for face recognition

Eigenfaces

Mean image

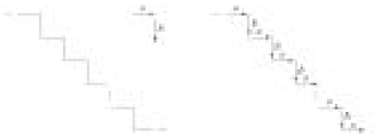


RELATIONAL DESCRIPTORS:

Relational descriptors

- Rewriting rules that capture the basic repetitive pattern.
- It applies to both boundaries and regions.
- Example: the staircase structure has been extracted from an image and we want to describe it.
- We employ two primitive elements and a set of rules.

(1) $S \rightarrow aA$,
(2) $A \rightarrow bS$, and
(3) $A \rightarrow b$.

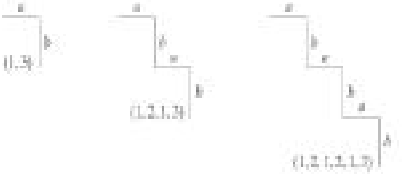


- S and A are variables and a and b are constants.
- Rule 1 indicates that the *starting symbol* S can be replaced by a and a variable A.
- Rules 2 and 3 indicate that variable A in turn can be replaced by b and S or b alone.
  - Replacing A by bS leads to the first rule and the procedure may be repeated.
  - Replacing A by b terminates the process as there ar no more variables to be processed.

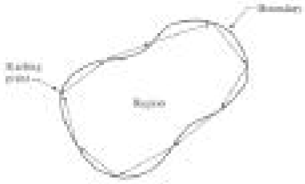
(1) $S \rightarrow aA$,
(2) $A \rightarrow bS$, and
(3) $A \rightarrow b$.

- Example derivations of the rules.
- The relationship between a and b is preserved as the rules force an a to be followed by a b.

(1) $S \rightarrow aA$,
(2) $A \rightarrow bS$, and
(3) $A \rightarrow b$.

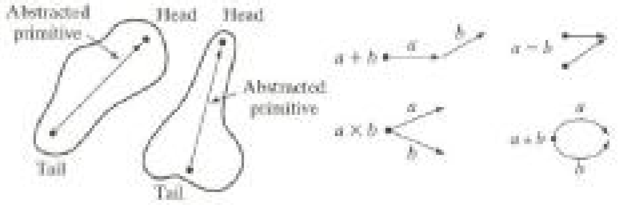


- These strings are 1D structures.
- Applications of the rules to images requires an establishment of methods for reducing 2D positional relations to 1D relations.
- An approach is to follow the contour of an object and code the result with head-to-tail segments of specified direction and length.

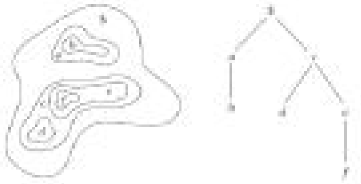


- Another approach is to employ directed line segments with other ways, besides head-to-tail connections.

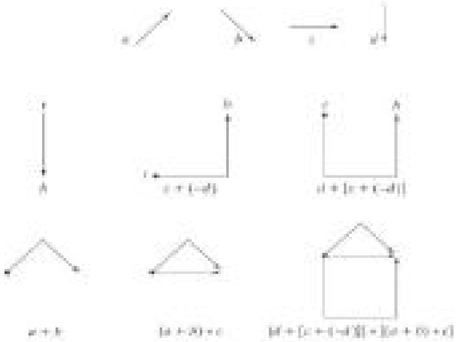
Primitives and operations among them



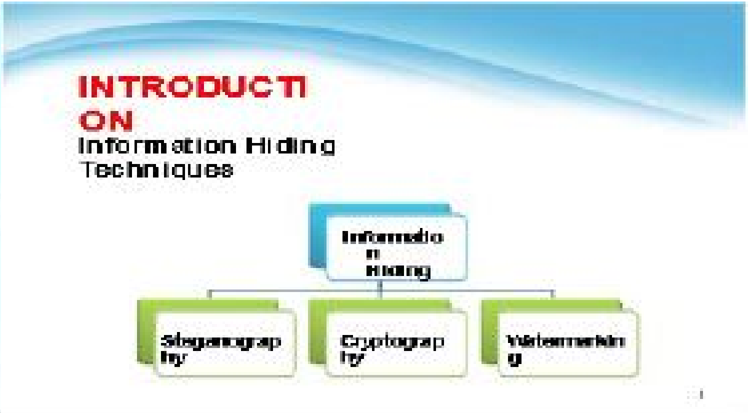
- Tree structures better represent non contiguous textured regions.
- Important information in a tree
  - A set of words describing the node (e.g. image region).
  - Relation between a node and its neighbors (e.g. "inside of").



- Step by step generation of a specific shape.



Digital Watermarking



INTRODUCTION

Information Hiding Techniques

WHAT IS A WATERMARK?

- A watermark is a form, image or text that is impressed onto paper which provides evidence of its authenticity.
- A distinguishing mark impressed on paper during manufacture; visible when paper is held up to the light (e.g. $ bill).

Visible Watermarking

Invisible Watermarking

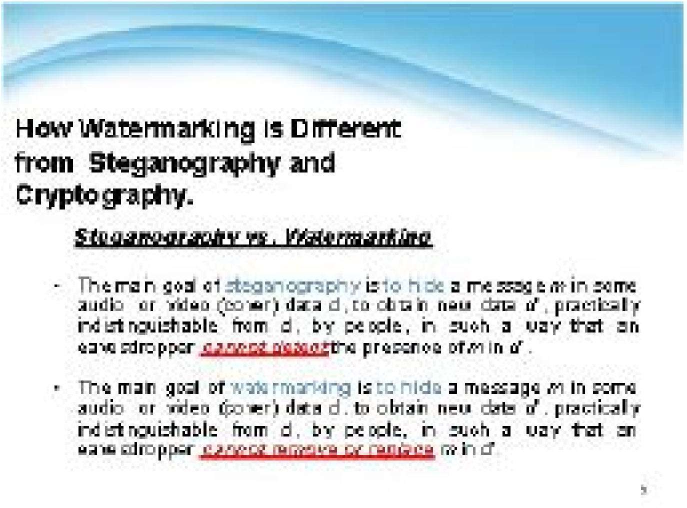
WHAT IS DIGITAL WATERMARKING ?

- Digital is an extension of concept watermarking g
- A digital watermark is a pattern of bits inserted into a digital image, audio or video file that identifies the file's copyright information (author, rights, etc).

Image

Video

Audio

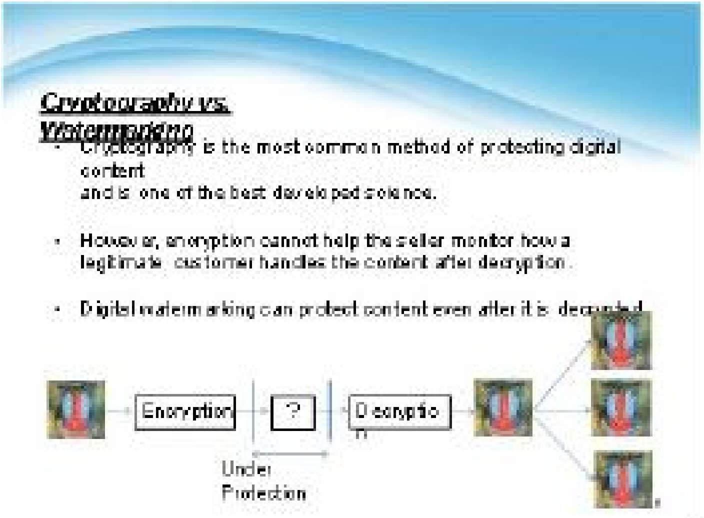


How Watermarking is Different from Steganography and Cryptography.

Steganography vs . Watermarking

- The main goal of steganography is to hide a message *m* in some audio or video (cover) data d, to obtain new data *d'*, practically indistinguishable from d, by people, in such a way that an eavesdropper *cannot detect* the presence of *m* in *d'*.
- The main goal of watermarking is to hide a message *m* in some audio or video (cover) data d, to obtain new data *d'*, practically indistinguishable from d, by people, in such a way that an eavesdropper *cannot remove or replace m* in *d'*.

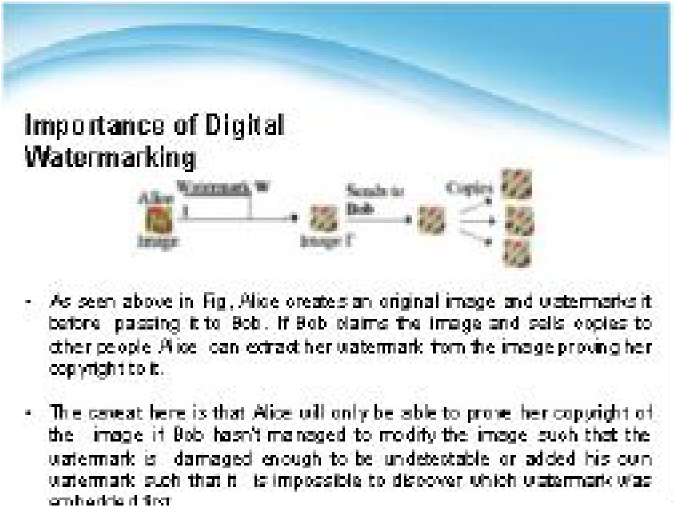
5



Cryptography vs. Watermarking

- Cryptography is the most common method of protecting digital content
  and is one of the best developed science.
- However, encryption cannot help the seller monitor how a legitimate customer handles the content after decryption.
- Digital watermarking can protect content even after it is decrypted.

6



Importance of Digital Watermarking

- As seen above in Fig, Alice creates an original image and watermarks it before passing it to Bob. If Bob claims the image and sells copies to other people Alice can extract her watermark from the image proving her copyright to it.
- The caveat here is that Alice will only be able to prove her copyright of the image if Bob hasn't managed to modify the image such that the watermark is damaged enough to be undetectable or added his own watermark such that it is impossible to discover which watermark was embedded first.

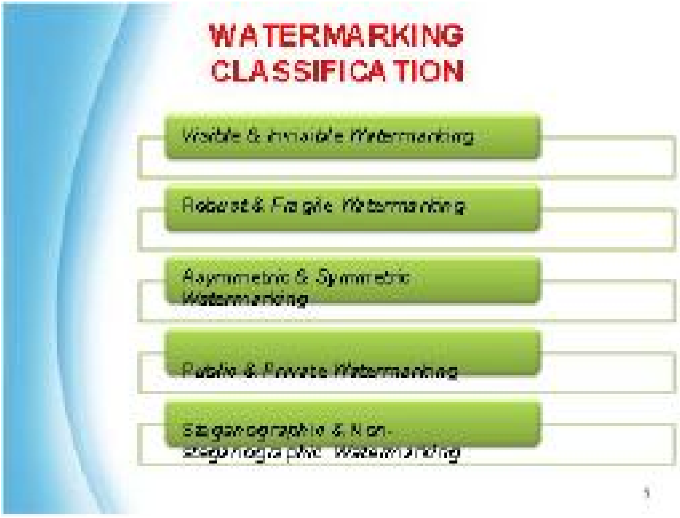
WHAT IS DIGITAL WATERMARKING?

- Digital is an extension of concept of the digital world watermarking g
- A digital watermark is a pattern of bits inserted into a digital image, audio or video file that identifies the file's copyright information (author, rights, etc.).

Imag e

Vide o

Audi o



WATERMARKING CLASSIFICATION

- Visible & Invisible Watermarking
- Robust & Fragile Watermarking
- Asymmetric & Symmetric Watermarking
- Public & Private Watermarking
- Steganographic & Non-steganographic Watermarking

Visible Watermarking

- Visible watermark is a translucent overlaid into an image and is visible to the viewer. Visible watermarking is used to indicate ownership and for copyright protection.

Visible Watermark

Original Image

Watermarked Image

Invisible Watermarking

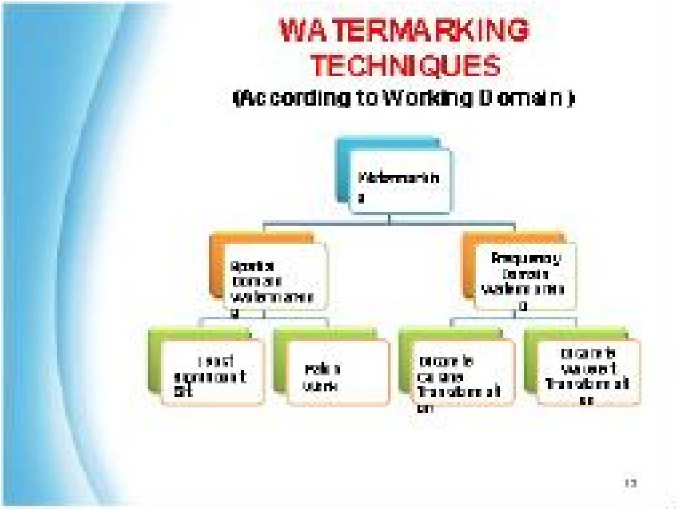
- Invisible watermark is embedded into the data in such a way that the changes made to the pixel values are perceptually not noticed. Invisible watermark is used as evidence of ownership and to detect misappropriated images.

Original Image

Watermarked Image

Dual Watermarking

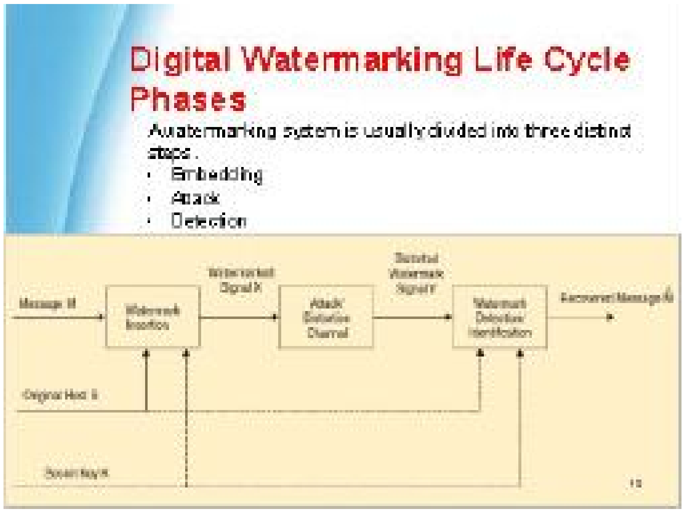
- Dual watermark is the combination of visible and invisible watermark. An invisible watermark is used as a backup for the visible watermark.



WATERMARKING TECHNIQUES

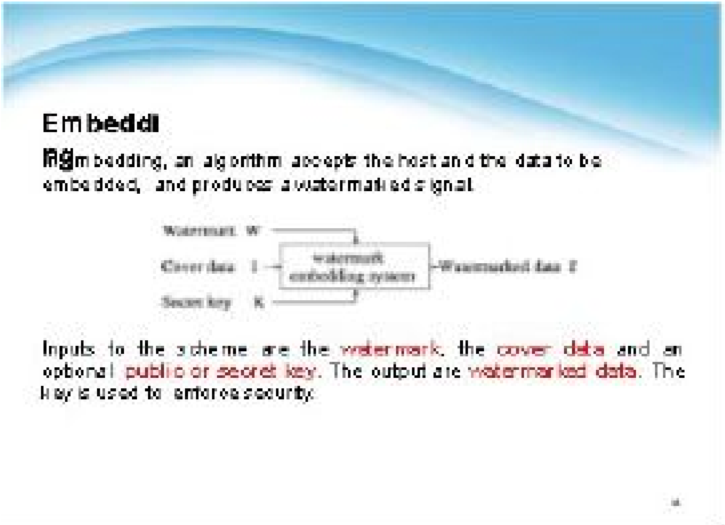
(According to Working Domain)

Watermarking
Spatial Domain Watermarking
Frequency Domain Watermarking
Least Significant Bit
Patch work
Discrete Cosine Transformation
Discrete Wavelet Transformation



Digital Watermarking Life Cycle Phases

A watermarking system is usually divided into three distinct steps .
- Embedding
- Attack
- Detection



Embedding

In embedding, an algorithm accepts the host and the data to be embedded, and produces a watermarked signal.

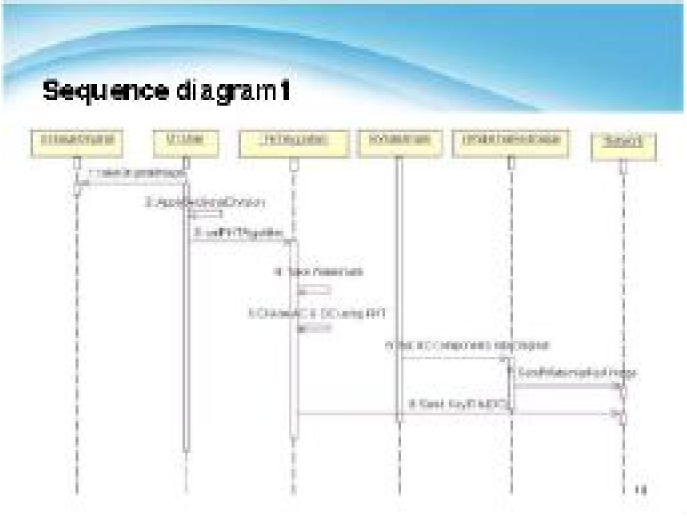
Inputs to the scheme are the watermark, the cover data and an optional public or secret key. The output are watermarked data. The key is used to enforce security.

ALGORITHM

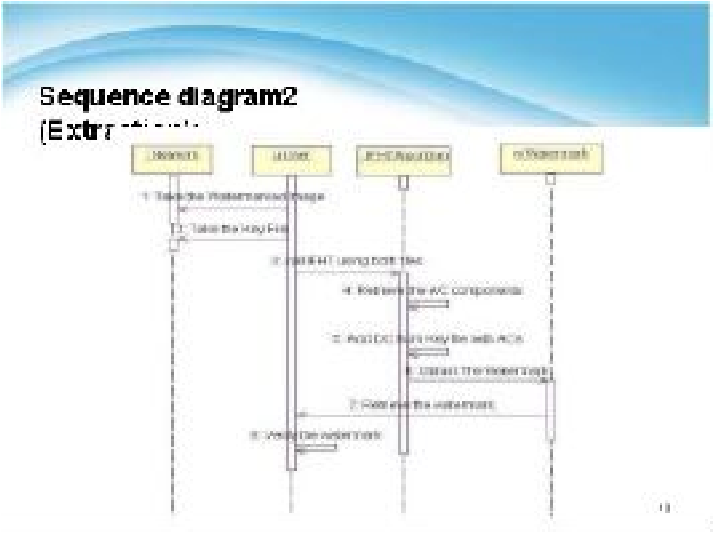
FAST HADAMARD TRANSFORM(FHT)

Advantages of FHT

- Shorter processing time
- Invisibility of the watermark guaranteed
- Increased watermark energy leads to higher robustness

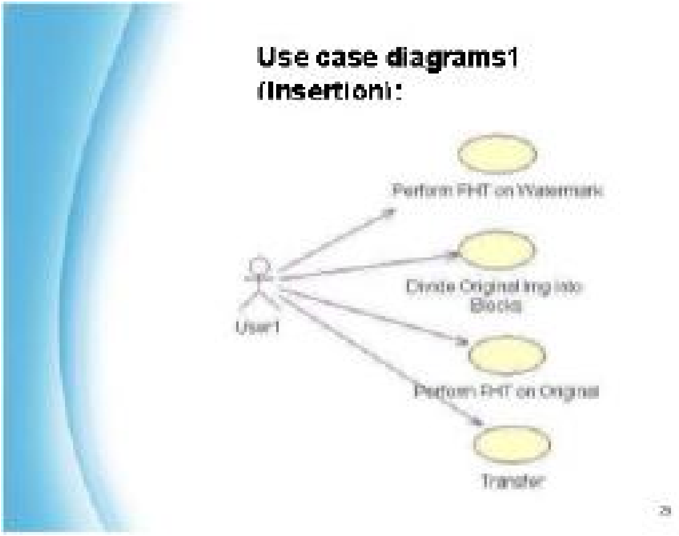


Sequence diagram1



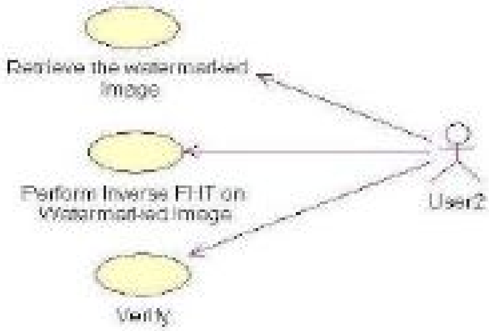
Sequence diagram2 (Extraction):

1: Take the Watermarked Image
2: Take the Key File
3: call FHT using both files
4: Retrieve the AC components
5: Add DC from Key file with ACs
6: Extract The Watermark
7: Retrieve the watermark
8: Verify the watermark



Use case diagrams1 (Insertion):

Use case diagrams2 (Extraction):


Retrieve the watermarked image
Perform Inverse FHT on Watermarked image
Verify
User2

WATERMARKING IN FHT DOMAIN

FAST HADAMARD TRANSFORM(FHT)

$$[V] = \frac{H_n [U] H_n}{N}$$

V ➡ Transformed image

U ➡ Actual image

Hn ➡ N x N Hadamard matrix, N=2^n, where [illegible] values either +1 or -1

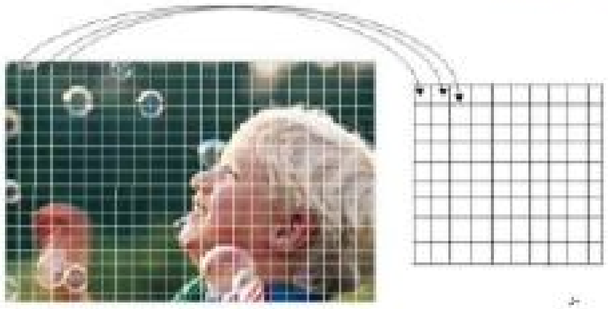
The Hadamard matrix of the order n is generated in terms of Hadamard matrix of order n-1 using Kronecker product, ⊗, as

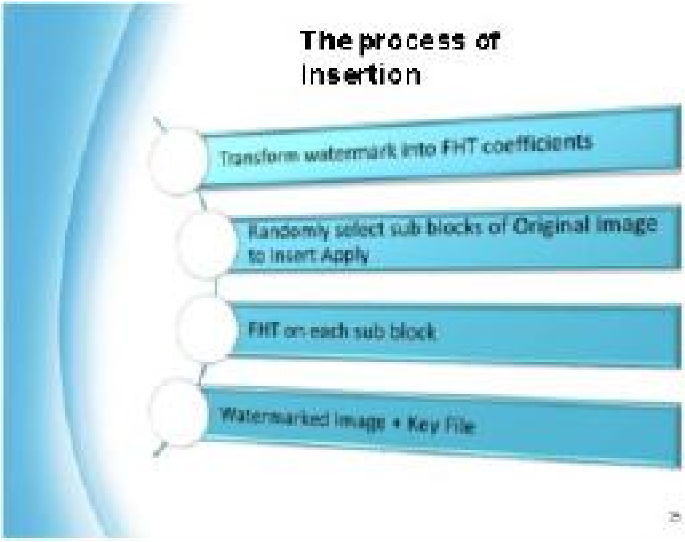
$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$$

Considering 8×8 sub-blocks of the whole image, the third order Hadamard transform matrix $H3$ becomes:

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

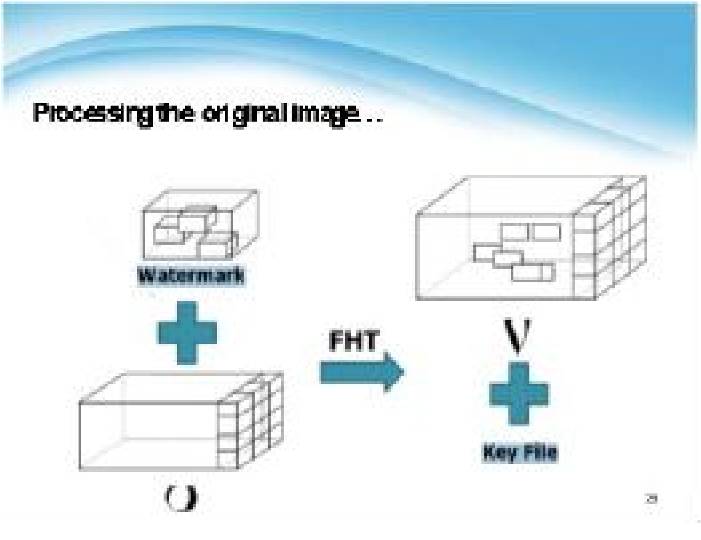
Original image pixel portions being taken into matrix "U"



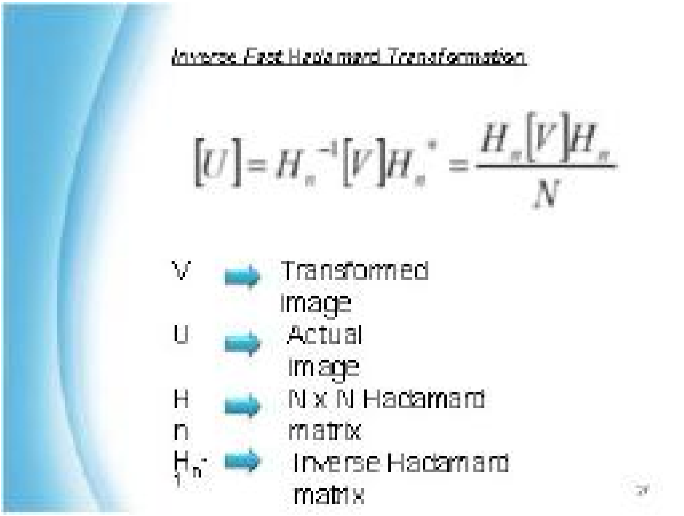


The process of insertion

- Transform watermark into FHT coefficients
- Randomly select sub blocks of Original image to insert Apply
- FHT on each sub block
- Watermarked image + Key File



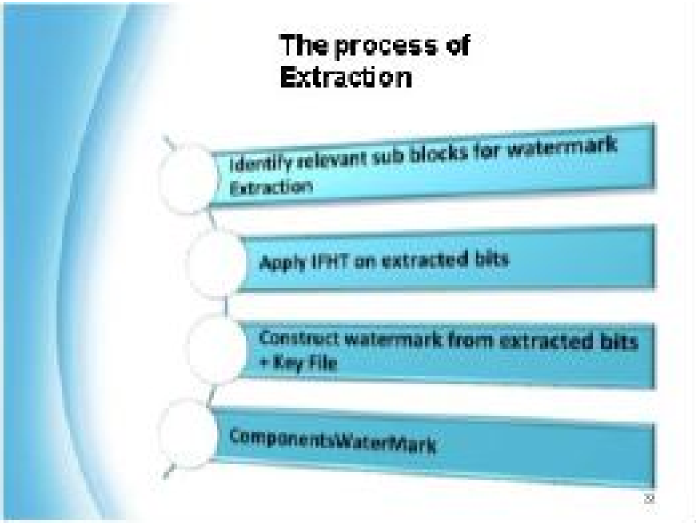
Processing the original image...



Inverse Fast Hadamard Transformation

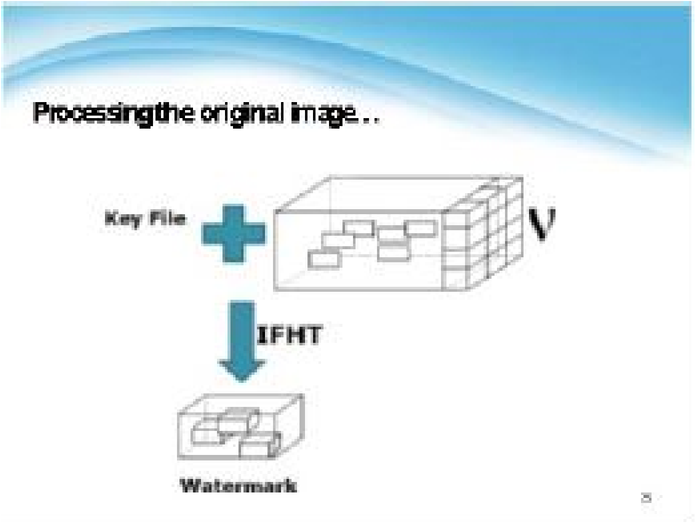
$$[U] = H_n^{-1}[V]H_n^{*} = \frac{H_n[V]H_n}{N}$$

- V ➡ Transformed image
- U ➡ Actual image
- H_n ➡ N x N Hadamard matrix
- H_n^{-1} ➡ Inverse Hadamard matrix



The process of Extraction

- Identify relevant sub blocks for watermark Extraction
- Apply IFHT on extracted bits
- Construct watermark from extracted bits + Key File
- ComponentsWaterMark



Processing the original image...

Biometrics

- Biometrics is the science and technology of measuring and analyzing biological data
- Biometrics refers to technologies that measure and analyze human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns ,facial patterns and hand measurements, for authentication purposes.
- The two categories of biometric identifiers include :

➢physiological characteristics.

➢behavioral characteristics.

○**Physiological** characteristics :

➢Fingerprint

➢face recognition

➢DNA

➢palm print

➢hand geometry

➢iris recognition (which has largely replaced retina)

➢Odour/scent.

○**Behavioral** characteristics :

➢Gait

➢voice

Fingerprint

- A fingerprint is the feature pattern of one finger.
- It is the pattern of ridges and valleys (also called furrows in the fingerprint literature) on the surface of a fingertip.
- Each individual has unique fingerprints so the uniqueness of a fingerprint is exclusively determined by the local ridge characteristics and their relationships
- These local ridge characteristics are not evenly distributed.

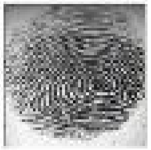


Fig. 1. A Fingerprint image acquired by an Optical Sensor

- Fingerprints are distinguished by Minutiae, which are some abnormal points on the ridges.
- The two most prominent local ridge characteristics, called minutiae, are
  1) ridge ending and
  2) ridge bifurcation.

- A ridge ending is defined as the point where a ridge ends abruptly.
- A ridge bifurcation is defined as the point where
  a ridge forks or diverges into branch ridges

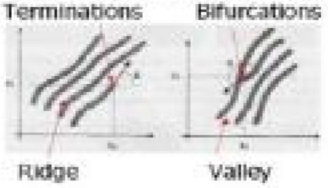


Fig 2.Ridge and valley

What is Fingerprint Recognition?

- Fingerprint recognition (sometimes referred to as dactyloscopy) is the process of comparing questioned and known fingerprint against another fingerprint to determine if the impressions are from the same finger or palm.

- The fingerprint recognition problem can be grouped into two sub-domains:

➢Fingerprint verification :

Fingerprint verification is to verifythe authenticity of one person by his fingerprint.

➢Fingerprint identification:

Fingerprint identification is to specifyone person's identity by his fingerprint(s).

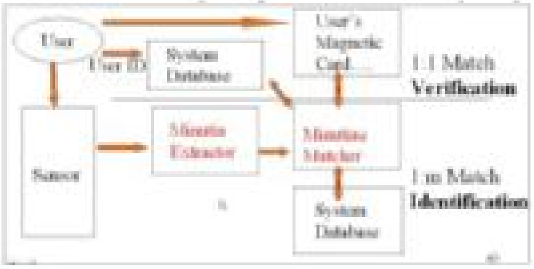


Fig 3.Verification vs. Identification

FINGERPRINT RECOGNITION SYSTEM

- Fingerprint recognition system operates in three stages:

(i) Fingerprint acquiring device

(ii) Minutia extraction and

(iii) Minutia matching

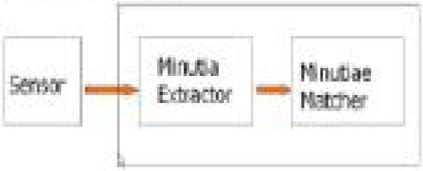


Fig 4. Fingerprint recognitionsystem

1. Fingerprint acquisition:

For fingerprint acquisition, optical or semi-conduct sensors are widely used. They have high efficiency and acceptable accuracy except for some cases that the user's finger is too dirty or dry.

2.Minutia extractor :

To implement a minutia extractor, a three-stage approach is widely used by researchers which are

➢ preprocessing

➢ minutia extraction and

➢ postprocessing stage.

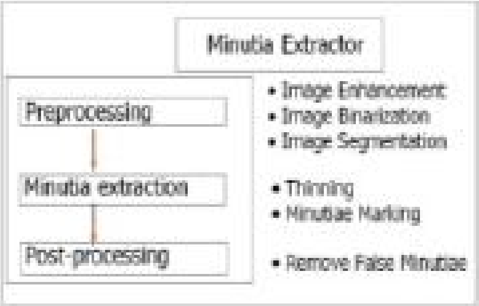


Fig 5.Minutia extractor

- For the fingerprint image preprocessing stage:

➢Image enhancement

➢Image binarization

➢Image segmentation

- The job of minutiae extraction closes down to two operations: Ridge Thinning,Minutiae Marking..

- In post-processing stage, false minutia are removed and bifurcations is proposed to unify terminations and bifurcations.

3.Minutiae Matching:

- Generally, an automatic fingerprint verification is achieved with minutia matching (point pattern matching)instead of a pixel-wise matching or a ridge pattern matching of fingerprint images.
- The minutia matcher chooses any two minutia as a reference minutia pair and then match their associated ridges first.
- If the ridges match well, two fingerprint images are aligned and matching is conducted for all remaining minutia.

ADVANTAGES

○Very high accuracy.

○Easy to use.

○Small storage space required for the biometric template.

DISADVANTAGES

○Dirt , grime and wounds.

○Placement of finger.

○Can be spoofed .

applications

○Banking Security - ATM security,card transaction

○Physical Access Control (e.g. Airport)

○Information System Security

○National ID Systems

○Passport control (INSPASS)

○Prisoner, prison visitors, inmate control

○Voting

○Identification of Criminals

○Identification of missing children

○Secure E-Commerce (Still under research)

IRIS RECOGNITION

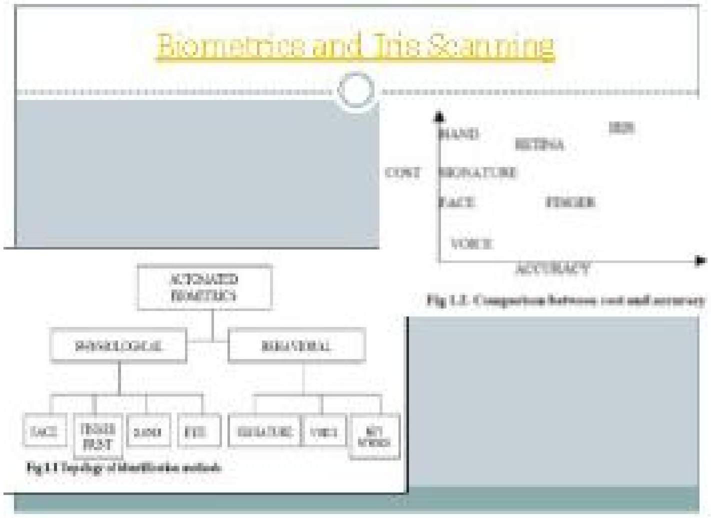
The need for biometrics

As per wikipedia,

"Biometrics consists of methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits"

<u>The need for biometrics</u>

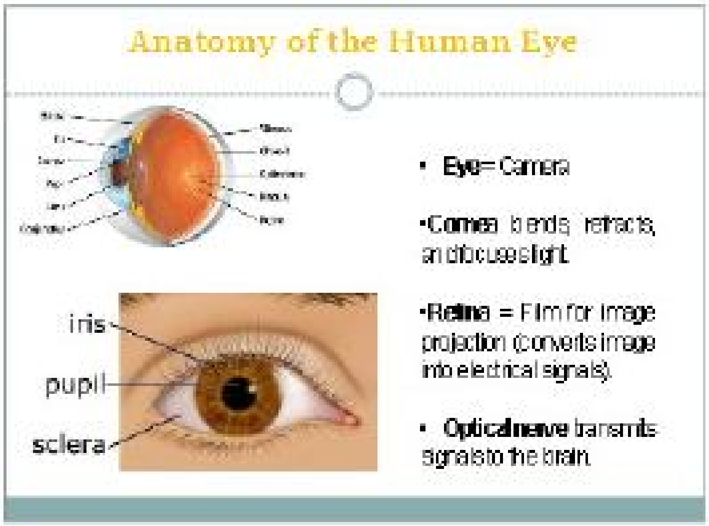
- -> Rapid development in technology
- -> Globalization



<u>Biometrics and Iris Scanning</u>

Fig 1.2. Comparison between cost and accuracy

Fig 1.1 Topology of identification methods



Anatomy of the Human Eye

- **Eye**= Camera

•**Cornea** bends, refracts, and focuses light.

•**Retina** = Film for image projection (converts image into electrical signals).

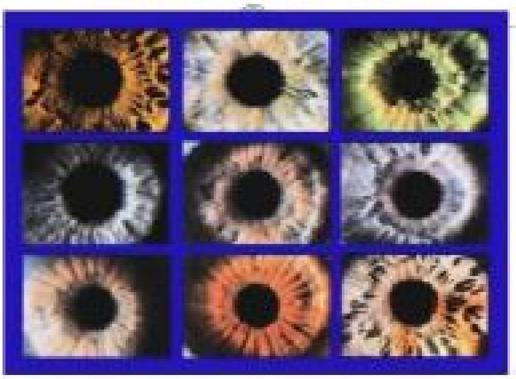
- **Optical nerve** transmits signals to the brain.

What is Iris?

- The coloured ring around the pupil of the eye is called the iris ,like a snowflake.
- Controls light levels inside the eye.
- Tiny muscles that dilate and constrict the pupil size.
- Divides the front of the eye from the back of the eye.
- Color comes from melanin.

brown or black in colour

Individuality of Iris



Left and right eye irises have distinctive pattern.

Characteristics of Iris

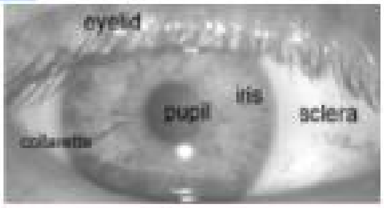
- Has highly distinguishing texture.
- Right eye differs from left eye.
- Twins have different iris texture.
- Iris pattern remains unchanged after the age of two and does not degrade overtime or with the environment.
- Iris patterns are extremely complex than other biometric patterns.

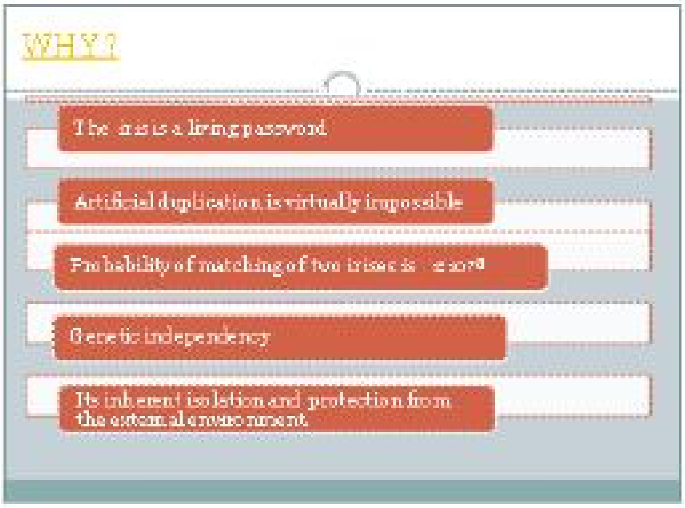
What Is It?

Going the layman way the biometric identification of the iris is called as —IRIS SCANNING‖.

But as per WIKIPEDIA,

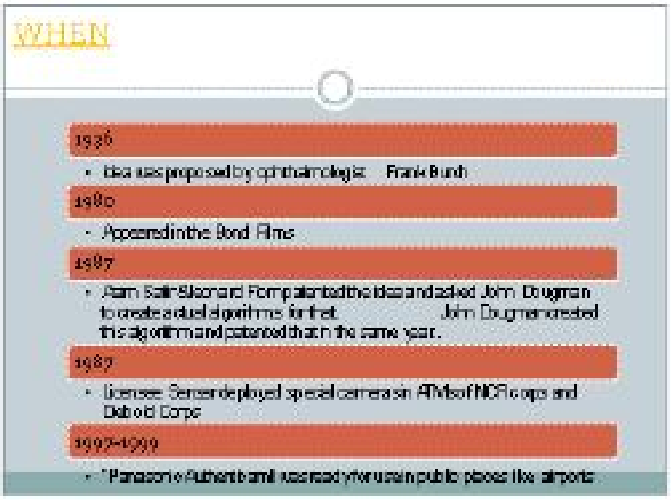
"**Iris recognition** is a method of biometric authentication that uses pattern-recognition techniques based on high-resolution images of the irides of an individual's eyes"





WHY ?

- The iris is a living password
- Artificial duplication is virtually impossible
- Probability of matching of two irises is 1 in 10^{78}
- Genetic independency
- Its inherent isolation and protection from the external environment



WHEN

1936

- Idea was proposed by ophthalmologist Frank Burch

1980

- Appeared in the Bond Films

1987

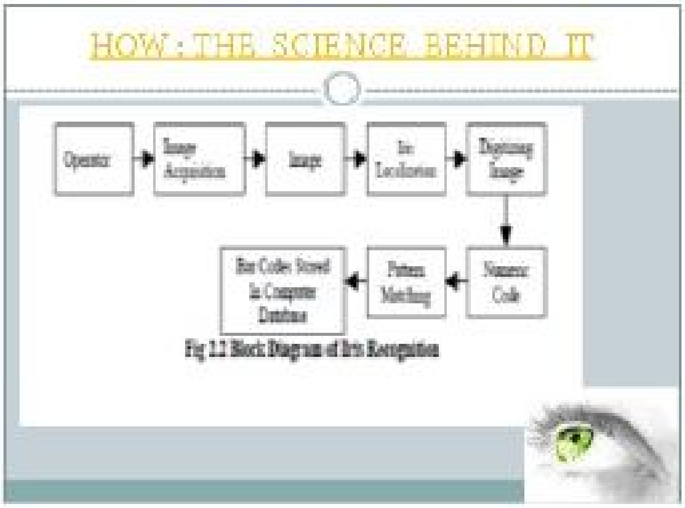
- Aram Safir & Leonard Flom patented the idea and asked John Daugman to create actual algorithms for that. John Daugman created this algorithm and patented that in the same year.

1987

- Licensee Sensar deployed special cameras in ATMs of NCR corps and Diebold Corps

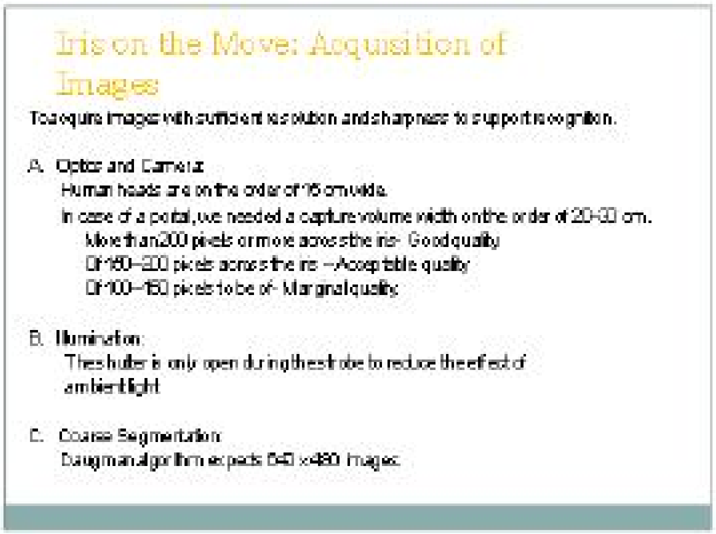
1997-1999

- "Panasonic Authenticam" was ready for use in public places like airports



HOW : THE SCIENCE BEHIND IT

Fig 2.2 Block Diagram of Iris Recognition



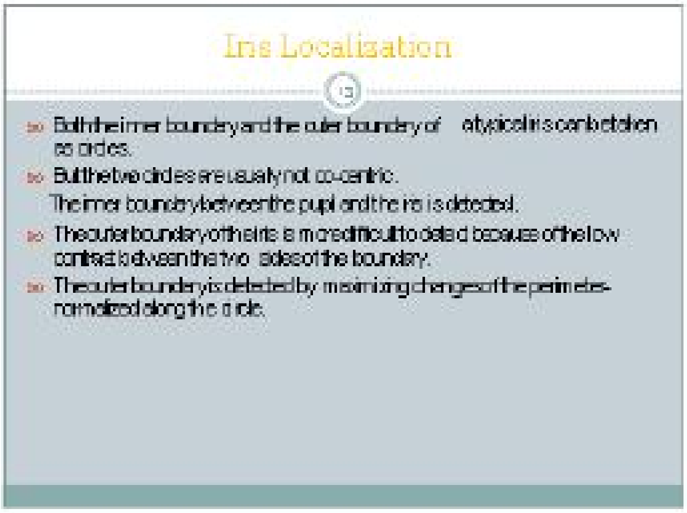
Iris on the Move: Acquisition of Images

To acquire images with sufficient resolution and sharpness to support recognition.

A. Optics and Camera:
   Human heads are on the order of 15 cm wide.
   In case of a portal, we needed a capture volume width on the order of 20-30 cm.
      More than 200 pixels or more across the iris- Good quality
      Of 150–200 pixels across the iris –Acceptable quality
      Of 100–150 pixels to be of- Marginal quality

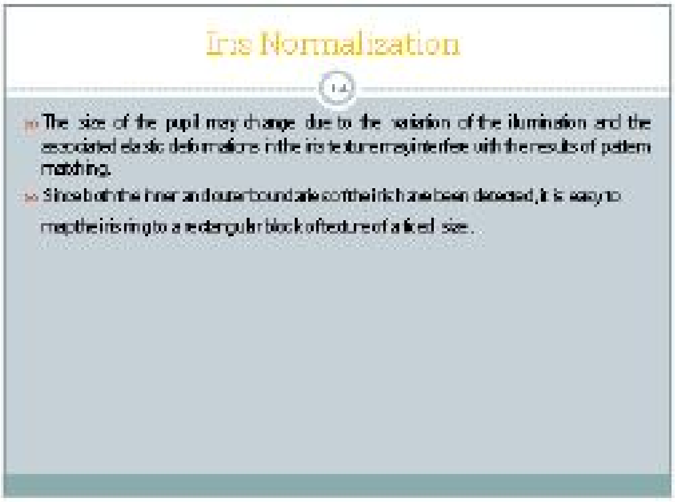
B. Illumination:
   The shutter is only open during the strobe to reduce the effect of ambient light

C. Coarse Segmentation:
   Daugman algorithm expects 640 ×480 images



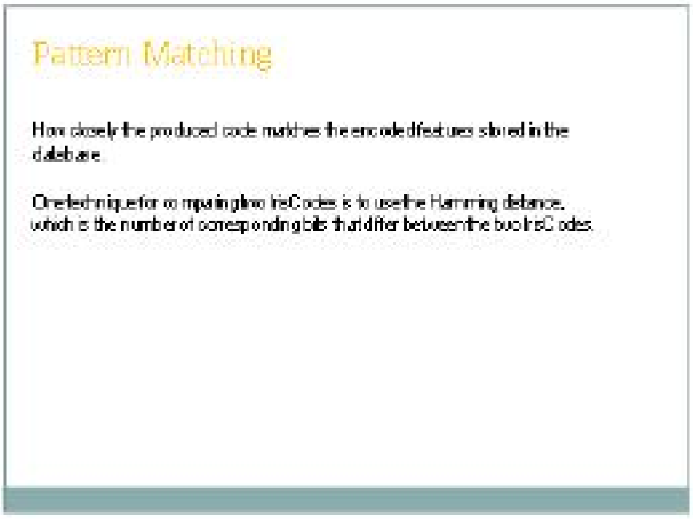
Iris Localization

- Both the inner boundary and the outer boundary of a typical iris can be taken as circles.
- But the two circles are usually not co-centric.
  The inner boundary between the pupil and the iris is detected.
- The outer boundary of the iris is more difficult to detect because of the low contrast between the two sides of the boundary.
- The outer boundary is detected by maximizing changes of the perimeter-normalized along the circle.



Iris Normalization

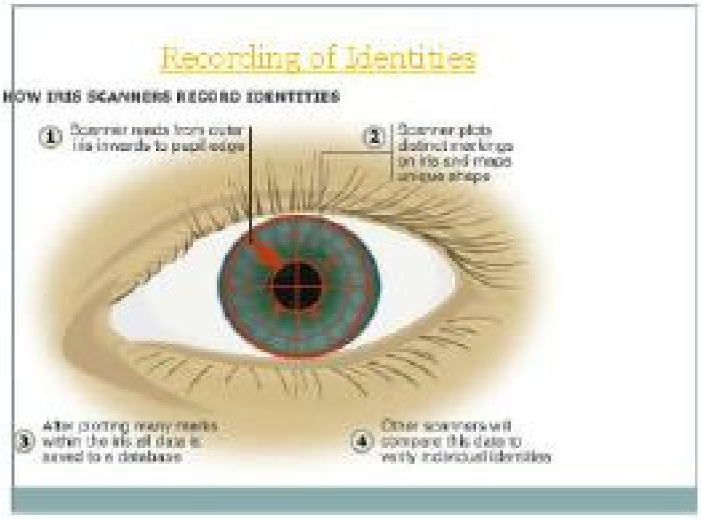
- The size of the pupil may change due to the variation of the illumination and the associated elastic deformations in the iris texture may interfere with the results of pattern matching.
- Since both the inner and outer boundaries of the iris have been detected, it is easy to map the iris ring to a rectangular block of texture of a fixed size.



Pattern Matching

How closely the produced code matches the encoded features stored in the database.

One technique for comparing two IrisCodes is to use the Hamming distance, which is the number of corresponding bits that differ between the two IrisCodes.



Recording of Identities

HOW IRIS SCANNERS RECORD IDENTITIES

1. Scanner reads from outer iris inwards to pupil edge
2. Scanner plots distinct markings on iris and maps unique shape
3. After plotting many marks within the iris all data is saved to a database
4. Other scanners will compare this data to verify individual identities

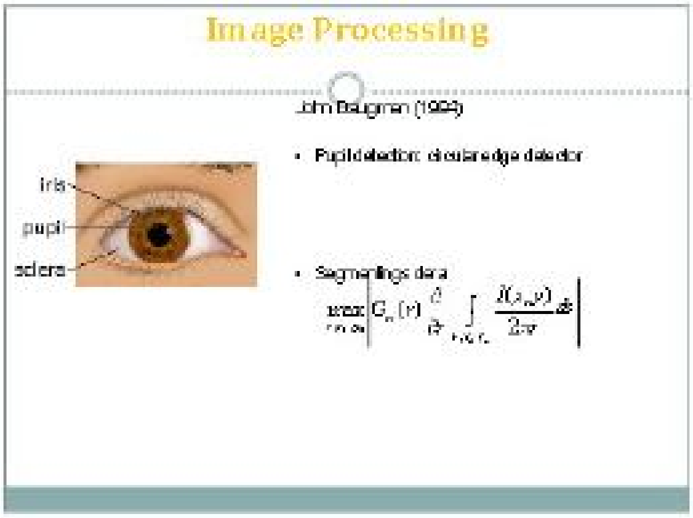
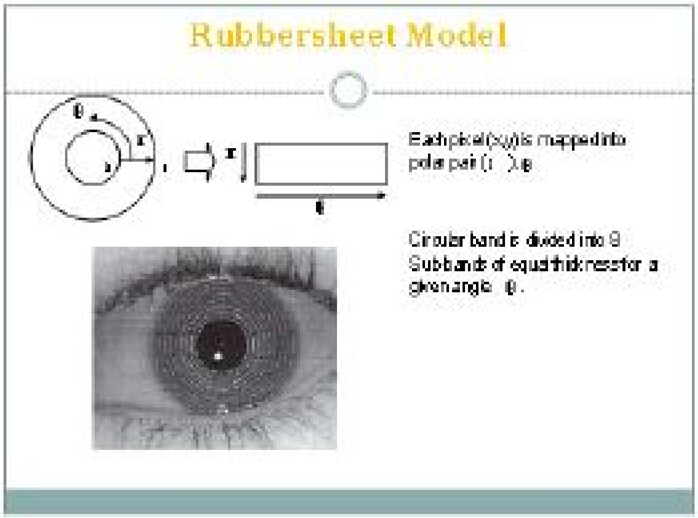


Image Processing

John Daugman (1994)

- Pupil detection: circular edge detector
- Segmenting sclera

$$\max_{(r,x_0,y_0)}\left|G_\sigma(r)*\frac{\partial}{\partial r}\oint_{r,x_0,y_0}\frac{I(x,y)}{2\pi r}ds\right|$$



Rubbersheet Model

Each pixel (x,y) is mapped into polar pair (r)θ

Circular band is divided into 8 Subbands of equal thickness for a given angle θ.

Measure of Performance

- Off line and on-line modes of operation.

Hamming distance is standard measure for comparison of binary strings.

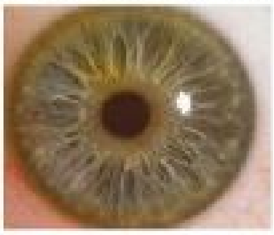
$$D = \frac{1}{n}\sum_{i=1}^{n} x_i \oplus y_i$$

x and y are two IrisCodes

$\oplus$ is the notation for exclusive OR (XOR)

Counts bits that disagree

Observations



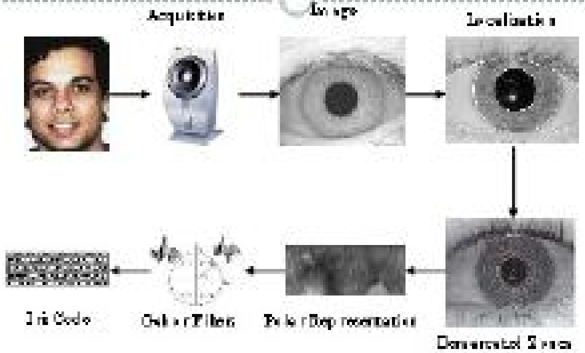
- Two IrisCodes from the same eye form genuine pair => genuine Hamming distance.

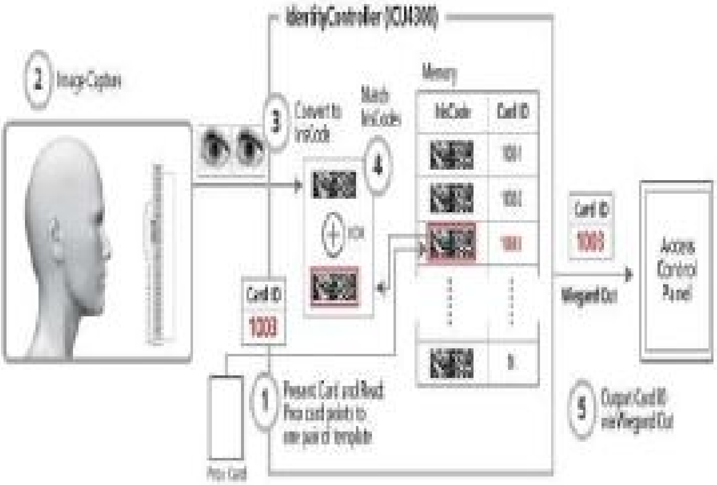
- Two IrisCodes from two different eyes form imposter pair => imposter Hamming distance.

- Bits in IrisCodes are correlated(both for genuine pair and for imposter pair).

- The correlation between IrisCodes from the same eye is stronger.

Iris Recognition System





Imaging Systems

Walkup Cameras

http://www.iridiantech.com

[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]
	[illegible]	[illegible]
	[illegible]	[illegible]
	[illegible]	[illegible]

Panasonic

OKI

Merits

- Highly protected, internal organ of the eye
- Externally visible; patterns imaged from a distance
- Iris patterns possess a high degree of randomness
  .Uniqueness: set by combinatorial complexity
  · Changing pupil size confirms natural physiology
- Limited genetic penetrance of iris patterns
- Patterns apparently stable throughout life.
- A key advantage of iris recognition is its stability, or template longevity, as, barring trauma, a single enrollment can last a lifetime.

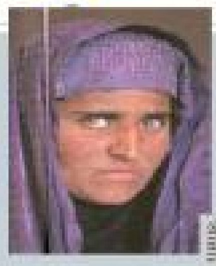
Demerits

- Small target (1 cm) to acquire from a distance (1m)
- Located behind a curved, wet, reflecting surface
- Obscured by eyelashes, lenses, reflections
- Partially occluded by eyelids, often drooping
- Deforms non-elastically as pupil changes size
- Illumination should not be visible or bright.

Applications

- . ATMs
- . Fugitive track record
- .Computer login: The iris as a living password.
- · National Border Controls: The iris as a living password.
- · Ticketless air travel.
- · Premises access control (home, office, laboratory etc.).
- · Driving licenses and other personal certificates.
- · Entitlements and benefits authentication.
- · Forensics, birth certificates, tracking missing or wanted person
- · Credit-card authentication.
- ·Automobile ignition and unlocking; anti-theft devices.
- ·Anti-terrorism (e.g.— suspect Screening at airports)
- · Secure financial transaction (e-commerce, banking).
- · Internet security, control of access to privileged information.

National Geographic: 1984 and 2002



Comparison

Method	Coded Pattern	Misidentification rate	Security	Applications
Iris	Iris pattern	1/1,200,000	High	high-security
Fingerprints	Fingerprints	1/1,000	Medium	Universal
Voice	Voice characteristics	1/30	Low	Telephone service
Signature	Shape of letters, writing Order, pen pressure	1/100	Low	Low-security
Face	Outline, shape & distribution of eyes, nose	1/100	Low	Low-security
Palm	size, length, & thickness hands	1/700	Low	Low-security