Semester – V

17CSU502B DIGITAL IMAGE PROCESSING 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int : **40** Ext : **60** Total: **100**

SCOPE

The objectives of this course are tomake the students learn the fundamental theories and techniques of digital image processing, cover the fundamental concepts of visual perception and image acquisition, basic techniques of image manipulation, segmentation and coding, and a preliminary understanding of Computer Vision.

OBJECTIVES

- To perform image manipulations and analysis in many different fields.
- To provide students with the ability to apply knowledge of computing, mathematics, science and engineering to solve problems in multidisciplinary research.

UNIT-I

Introduction: Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization. Spatial Domain Filtering: Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

UNIT-II

Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.

UNIT-III:

Image Restoration, Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections, Image Compression-Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding.

UNIT – IV

FAX compression (CSUITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation

Wavelet based Image Compression: Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT),Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking

UNIT-V

Morphological Image Processing: Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons,pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion. Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding,Otsu's method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm, Use of motion in segmentation

Suggested Readings

1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.
2. Jain, A. K. (1989). Fundamentals of Digital image Processing. New Delhi: Prentice Hall of India.
3. Castleman, K. R. (1996). Digital Image Processing. New Delhi: Pearson Education.
4. Schalkoff. (1989). Digital Image Processing and Computer Vision. New York: John Wiley and Sons.
5. Rafael, C. Gonzalez., Richard, E. Woods.,& Steven Eddins. (2004). Digital Image Processing using MATLAB. New Delhi: Pearson Education.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2017 onwards)
DEPARTMENT OF COMPUTER SCIENCE, CA & IT

SUBJECT : **DIGITAL IMAGE PROCESSING**

SEMESTER : **V** **L T P C**

SUBJECT CODE: **17CSU502B** **CLASS** : **III B.Sc.CS - A** **4 0 0 4**

LECTURE PLAN

UNIT-I

S.No	Lecture Duration (Hours)	Topics to be Covered	Support Materials
1	1	Introduction: Light, Brightness adaption and discrimination	T1 : Pg 43-45, 39-42
2	1	Pixels, coordinate conventions	T1: 68, W1
3	1	Imaging Geometry, Perspective Projection	W1
4	1	Spatial Domain Filtering, sampling and quantization.	T1: 144, T1: 52-54
5	1	Spatial Domain Filtering: Intensity transformations, contrast stretching	T1: 107-115
6	1	Histogram equalization, Correlation and convolution	T1: 122-126, T1: 146-149
7	1	Smoothing filters	T1: 152-156
8	1	Sharpening filters, gradient and Laplacian.	T1: 157-160, T1:165-168
9	1	Recapitulation and Discussion of Important Questions	
		Total hours planned for Unit 1: 9 hrs	

UNIT-II

S.No	Lecture Duration (Hours)	Topics to be Covered	Support Materials
1	1	Hotelling Transform	W2
2	1	Fourier Transforms and properties	T1: 211-213
3	1	FFT (Decimation in Frequency and Decimation in Time Techniques)	T1: 299
4	1	Convolution	T1: 209-210
5	1	Correlation	T1: 210-211
6	1	2-D sampling	T1: 227
7	1	Discrete Cosine Transform	W3
8	1	Frequency domain filtering.	269-294
9	1	Recapitulation and Discussion of Important Questions	
		Total hours planned for Unit 2: 9 hrs	

UNIT-III

S.No	Lecture Duration (Hours)	Topics to be Covered	Support Materials
1	1	Image Restoration, Basic Framework, Interactive Restoration	**T1: 312-313**
2	1	Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization	T1: 322-330, T1: 656 T1: 313-319
3	1	Noise restoration filters, Adaptive filters, Linear Position invariant degradations, Estimation of Degradation functions, Restoration from projections	T1: 322-338, T1: 343-345, T1: 346-347, T1: 362-381
4	1	Image Compression-Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source	T1: 525-528 T1: 528-529 T1: 536-538 T1: 531-533
5	1	Shannon's 1st Theorem, Huffman Coding Arithmetic Coding,	T1: 542, T1: 548
6	1	Golomb Coding, LZW coding, Transform Coding	T1: 544-547, 551-552, 566-583
7	1	Sub-image size selection blocking artifacts	T1: 562-566
8	1	DCT implementation using FFT	T1: 604-613,

		Run length coding.	T1: 553-558
9	1	Recapitulation and Discussion of Important Questions	
		Total hours planned for Unit 3: 9 hrs	

UNIT – IV

S.No	**Lecture Duration (Hours)**	**Topics to be Covered**	**Support Materials**
1	1	FAX compression (CSUITT Group-3 and Group-4)	**W4, T1:555-559**
2	1	Symbol-based coding, JBIG-2, Bit-plane encoding	**T1: 559-562**
3	1	Bit-allocation, Zonal Coding, Threshold Coding, JPEG	**T1: 574-578, T1:579-583**
4	1	Lossless predictive coding, Lossy predictive coding, Motion Compensation	**T1: 584-588, T1:596-602, T1: 589-591**
5	1	Wavelet based Image Compression: Expansion of functions, Multi-resolution analysis	**T1: 477-478**
6	1	Scaling functions, MRA refinement equation, Wavelet series expansion	**T1: 479, T1: 486**
7	1	Discrete Wavelet Transform (DWT),Continuous Wavelet Transform, Fast Wavelet Transform	**T1: 488-490,T1:491-492, T1: 493-500**
8	1	2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking	**T1:501-509, T1: 614-620**
9	1	Recapitulation and Discussion of Important Questions	
		Total hours planned for Unit 4: 9 hrs	

UNIT-V

S.No	**Lecture Duration (Hours)**	**Topics to be Covered**	**Support Materials**
1	1	Morphological Image Processing: Basics	**T1: 627**
2	1	SE Erosion, Dilation, Opening, Closing	**T1:630-635, T1: 635-640**
3	1	Hit-or-Miss Transform,	**T1: 640-641**

		Boundary Detection, Hole filling	**T1: 642-643**
4	1	Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion	**T1: 645-656**
5	1	Reconstruction by dilation and erosion. Image Segmentation: Boundary detection based techniques	**T1: 665-667,** **T1: 689-691**
6	1	Point line detection, Edge detection, Edge linking, local processing, regional processing	**T1: 692-714** **T1: 725-737**
7	1	Hough transform, Thresholding, Iterative thresholding, Otsu's method	**T1: 738-746**
8	1	Region-based segmentation, Watershed algorithm, Use of motion in segmentation	**T1: 763-768,** **T1: 774-775,** **T1: 778-784**
9	1	Recapitulation and Discussion of Important Questions	
10	1	Discussion of Previous ESE Question Papers	
11	1	Discussion of Previous ESE Question Papers	
12	1	Discussion of Previous ESE Question Papers	
		Total hours planned for Unit 5: 12 hrs	
		Total hours planned : 48 hrs	

TEXTBOOK:

1. **[T1] :** Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.

SUGGESTED READINGS:

2. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.
3. Jain, A. K. (1989). Fundamentals of Digital image Processing. New Delhi: Prentice Hall of India.
4. Castleman, K. R. (1996). Digital Image Processing. New Delhi: Pearson Education.
5. Schalkoff. (1989). Digital Image Processing and Computer Vision. New York: John Wiley and Sons.
6. Rafael, C. Gonzalez., Richard, E. Woods.,& Steven Eddins. (2004). Digital Image Processing using MATLAB. New Delhi: Pearson Education.

WEBSITES:

W1: http://www.cse.psu.edu/~rtc12/CSE486/lecture12.pdf
W2 : www.cs.umu.se/kurser/TDBC30/VT04/material/lect14_kap_11.pdf
W3: https://www.cs.cf.ac.uk/Dave/Multimedia/node231.html
W4: http://cs.haifa.ac.il/~nimrod/Compression/Image/I1fax2009.pdf

UNIT-I

SYLLABUS

Introduction: Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization. Spatial Domain Filtering: Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

INTRODUCTION

Interest in digital image processing methods stems from two principal application areas:

- improvement of pictorial information for human interpretation;
- processing of image data for storage, transmission, and representation for autonomous machine perception.

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 intensity 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 called *picture elements*, *image elements*, *pels*, and *pixels*. *Pixel* is the term used most widely to denote the elements of a digital image.

LIGHT:

Light is a particular type of electromagnetic radiation that can be sensed by the human eye.

Electromagnetic spectrum:

In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As in Fig. the range of colors we perceive in visible light represents a very small portion of the electromagnetic spectrum. On one end of the spectrum are radio waves with wavelengths billions of times longer than those of visible light. On the other end of the spectrum are gamma rays with wavelengths millions of times smaller than those of visible light. The electromagnetic spectrum can be expressed in terms of wavelength, frequency, or energy.

Wavelength(λ) and frequency(v) are related by the expression

$$\lambda = \frac{c}{\nu}$$

where c is the speed of light The energy of the various components of the electromagnetic spectrum is given by the expression (2.2-2) where h is Planck's constant.

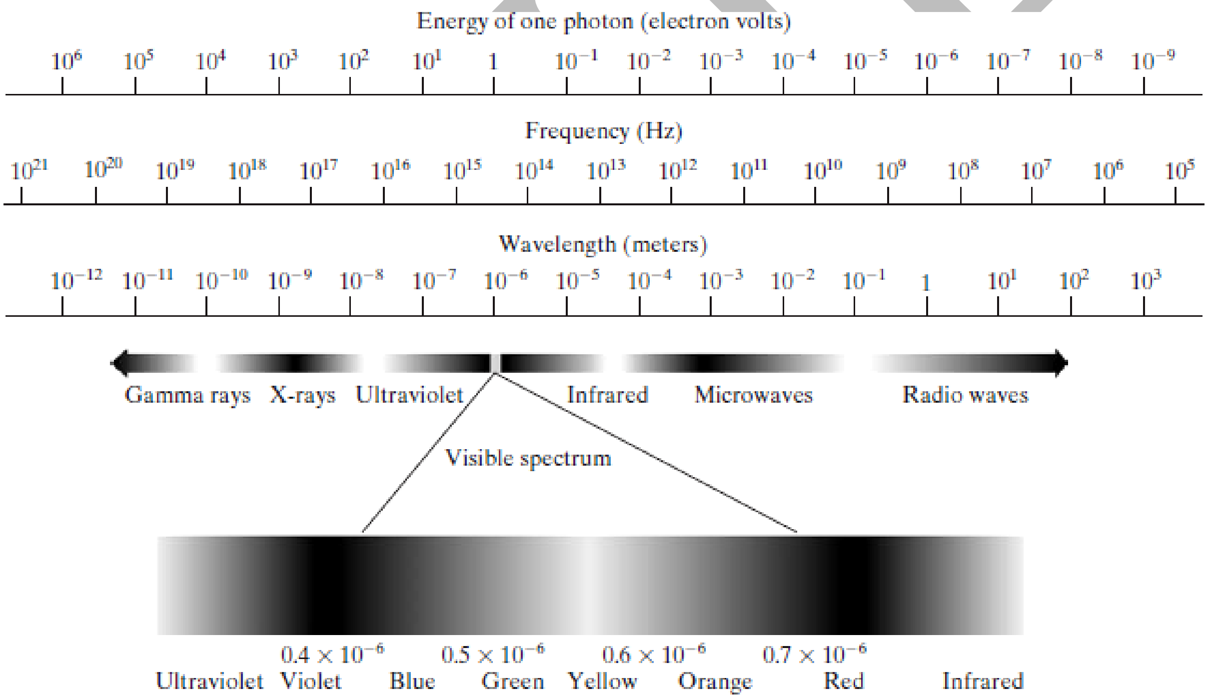


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

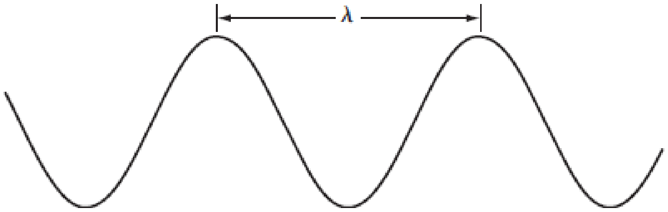
Electromagnetic waves can be conceptualized as propagating sinusoidal waves of varying wavelengths, or they can be thought of as a stream of massless particles, each traveling in a wavelike pattern and moving at the speed of light.

Each massless particle contains a certain amount (or bundle) of energy. Each bundle of energy is called a *photon*.

We see from Eq. (2.2-2) that energy is proportional to frequency, so the higher-frequency (shorter wavelength) electromagnetic phenomena carry more energy per photon. Thus, radio waves have photons with low energies, microwaves have more energy than radio waves, infrared still more, then visible, ultraviolet, X-rays, and finally gamma rays, the most energetic of all. This is the reason why gamma rays are so dangerous to living organisms.

If spectral bands are grouped according to energy per photon, we obtain the spectrum shown in Fig. 1.5, ranging from gamma rays (highest energy) at one end to radio waves (lowest energy) at the other.

FIGURE 2.11
Graphical representation of one wavelength.



Units of measurements

- Frequency is measured in Hertz (Hz)
- Energy is measured in electron-volt

- **Photon**
- Massless particles whose stream in a sinusoidal wave pattern forms energy
- Energy is directly proportional to frequency
- Higher frequency energy particles carry more energy
- Radio waves have less energy while gamma rays have more energy, making gamma rays more dangerous to living organisms

The visible band of the electromagnetic spectrum spans the range from approximately (violet) to about (red). For convenience, the color spectrum is divided into six broad regions: violet, blue, green, yellow, orange, and red. No color (or other component of the electromagnetic spectrum) ends abruptly, but rather each range blends smoothly into the next, as shown in Fig. 2.10.

The colors that humans perceive in an object are determined by the nature of the light *reflected* from the object. A body that reflects light relatively balanced in all visible wavelengths appears white to the observer. However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths.

Light that is void of color is called *monochromatic* (or *achromatic*) *light*.

The only attribute of monochromatic light is its *intensity* or amount.

Because the intensity of monochromatic light is perceived to vary from black to grays and finally to white, the term *gray level* is used commonly to denote monochromatic intensity. We use the terms *intensity* and *gray level* interchangeably in subsequent discussions. The range of measured values of monochromatic light from black to white is usually called the *gray scale*, and monochromatic images are frequently referred to as *gray-scale images*.

Chromatic* (*color*) *light spans the electromagnetic energy spectrum from approximately 0.43 to 0.79 µm as noted previously.

In addition to frequency, three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, and brightness.

Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W).

***Luminance*,** measured in lumens (lm), gives a measure of the amount of energy an observer *perceives* from a light source.

For example, light emitted from a source operating in the far infrared region of the spectrum could have significant energy (radiance), but an observer would hardly perceive it; its luminance would be almost zero.

Brightness is a subjective descriptor of light perception that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

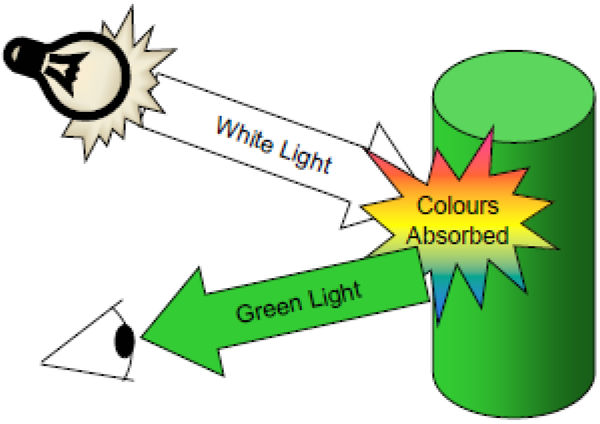
- At the short wavelength end of the electromagnetic spectrum, we have gamma rays and X-rays. Gamma radiation is important for medical and astronomical imaging, and for imaging radiation in nuclear environments. Hard (high-energy) X-rays are used in industrial applications. Chest and dental X-rays are in the lower energy (soft) end of the X-ray band.

- The soft X-ray band transitions into the far ultraviolet light region, which in turn blends with the visible spectrum at longer wavelengths.

- Moving still higher in wavelength, we encounter the infrared band, which radiates heat, a fact that makes it useful in imaging applications that rely on "heat signatures."
- The part of the infrared band close to the visible spectrum is called the *near-infrared* region.
- The opposite end of this band is called the *far-infrared* region.
- This latter region blends with the microwave band. This band is well known as the source of energy in microwave ovens, but it has many other uses, including communication and radar.
- Finally, the radio wave band encompasses television as well as AM and FM radio. In the higher energies, radio signals emanating from certain stellar bodies are useful in astronomical observations.

Light is just a particular part of the electromagnetic spectrum that can be sensed by the human eye

The electromagnetic spectrum is split up according to the wavelengths of different forms of energy. The colours that we perceive are determined by the nature of the light reflected from an object.



For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object

_ Visible spectrum

– 0:43_m (violet) to 0:79_m (red)
– VIBGYOR regions
– Colors are perceived because of light reflected from an object
– Absorption vs reflectance of colors
_ An object appears white because it reflects all colors equally

_ Achromatic or monochromatic light
– No color in light
– Amount of energy describes intensity
_ Quantified by gray level from black through various shades of gray to white
– Monochrome images also called gray scale images

_ Chromatic light
– Spans the energy spectrum from 0.43 to 0.79 _m
– Described by three basic quantities: radiance, luminance, brightness
– Radiance
_ Total amount of energy flowing from a light source
_ Measured in Watts

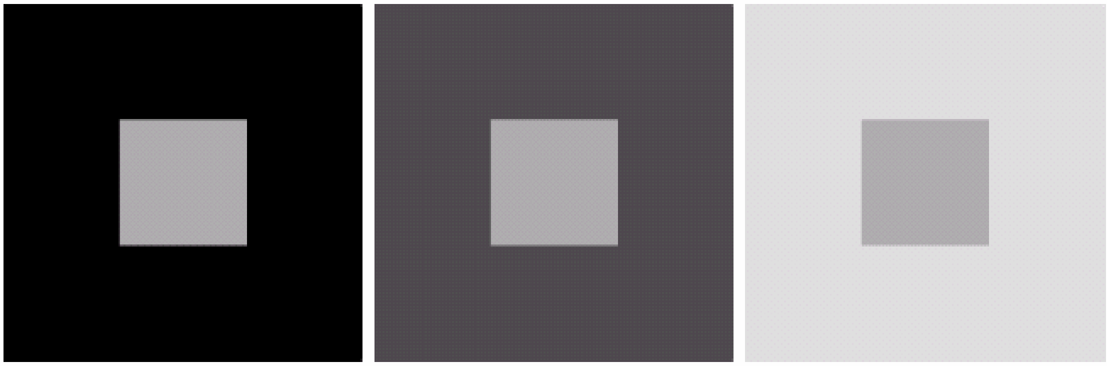
– Luminance
_ Amount of energy perceived by an observer from a light source
_ Measured in lumens (lm)

– Brightness
_ Subjective descriptor of light perception
_ Achromatic notion of intensity
_ Key factor in describing color sensation

BRIGHTNESS ADAPTATION AND DISCRIMINATION

Because digital images are displayed as a discrete set of intensities, the eye's ability to discriminate between different intensity levels is an important consideration in presenting image processing results.

The human visual system can perceive approximately 1010 different light intensity levels However, at any one time we can only discriminate between a much smaller number – *brightness adaptation* Similarly, the *perceived intensity* of a region is related to the light intensities of the regions surrounding it



An example of *simultaneous contrast*

Brightness adaptation and discrimination

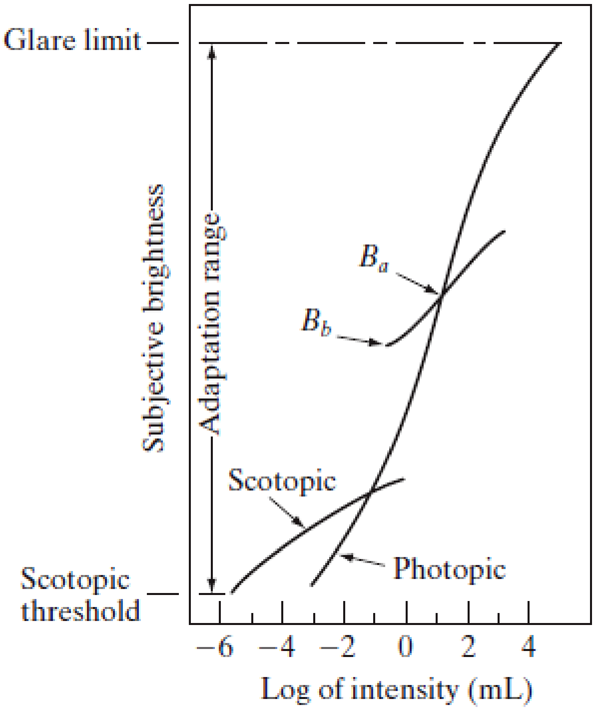
- Digital images displayed as a discrete set of intensities
- Range of human eye is about 10^{10} different light intensity levels, from scotopic threshold to the glare limit
- Subjective brightness (as perceived by humans) is a logarithmic function of light intensity incident on the eye
- Visual system cannot operate over the enormous range simultaneously

Brightness adaptation

- Change in overall sensitivity of perceived brightness
- Number of distinct intensity level that can be perceived simultaneously is small compared to number of levels that can be perceived
- Brightness adaptation level current sensitivity level of the visual system

The range of light intensity levels to which the human visual system can adapt is enormous—on the order of — from the scotopic threshold to the glare limit.

Experimental evidence indicates that *subjective brightness* (intensity as *perceived* by the human visual system) is a logarithmic function of the light intensity incident on the eye.



In fig a plot of light intensity versus subjective brightness, illustrates this characteristic.The long solid curve represents the range of intensities to which the visual system can adapt. In photopic vision alone, the range is about The transition from scotopic to photopic vision is gradual over the approximate range from 0.001 to 0.1 milliambert (to in the log scale), as the double branches of the adaptation curve in this range show.

The essential point in interpreting the impressive dynamic range depicted in Fig. 2.4 is that the visual system cannot operate over such a range *simultaneously.* Rather, it accomplishes this large variation by changing its overall sensitivity, a phenomenon known as *brightness adaptation.*

The total range of distinct intensity levels the eye 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 in Fig. 2.4. 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 at and below which all stimuli are perceived as indistinguishable blacks.

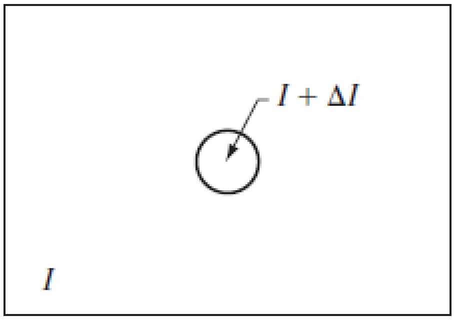
The upper portion of the curve is not actually restricted but, if extended too far, loses its meaning because much higher intensities would simply raise the adaptation level higher than The ability of the eye to discriminate between *changes* in light intensity at any specific adaptation level is also of considerable interest.

A classic experiment used to determine the capability of the human visual system for brightness discrimination consists of having a subject look at a flat, uniformly illuminated area large enough to occupy the entire field of view.

This area typically is a diffuser, such as opaque glass, that is illuminated from behind by a light source whose intensity, *I*, can be varied. To this field is added an increment of illumination, in the form of a short-duration flash that appears as a circle in the center of the uniformly illuminated field, as Fig. 2.5 shows

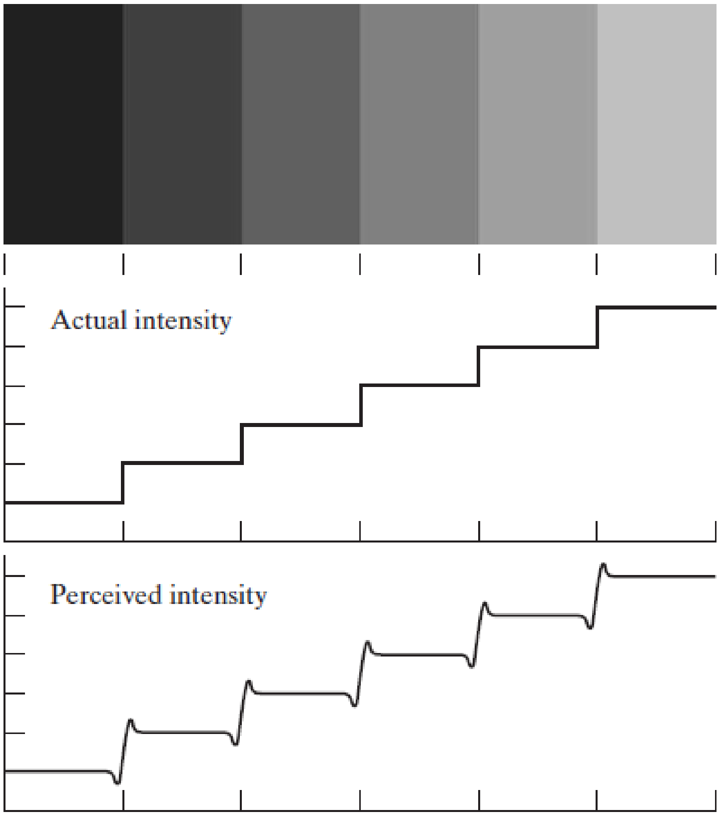
Weber ratio

FIGURE 2.5 Basic experimental setup used to characterize brightness discrimination.



- Measure of contrast discrimination ability
- Background intensity given by I
- Increment of illumination for short duration at intensity ΔI (Fig 2.5)
- ΔIc is the increment of illumination when the illumination is visible half the time against background intensity I
- Weber ratio is given by ΔIc/I
- A small value of ΔIc/I implies that a small percentage change in intensity is visible, representing good brightness
- discrimination
- A large value of ΔIc/I implies that a large percentage change is required for discrimination, representing poor
- brightness discrimination
- Typically, brightness discrimination is poor at low levels of illumination and improves at higher levels of background illumination

Mach bands



a
b
c

FIGURE 2.7 Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

- Brightness pattern near the boundaries shown in stripes of constant intensity (Figure 2.7)
- The bars themselves are useful for calibration of display equipment
  - **Simultaneous contrast**
- A region's perceived brightness does not depend simply on intensity
- Lighter background makes an object appear darker while darker background makes the same object appear brighter (Fig 2.8)
- The second phenomenon, called *simultaneous contrast*, is related to the fact that a region's perceived brightness does not depend simply on its intensity, as Fig. 2.8 demonstrates. All the center squares have exactly the same intensity.
- However, they appear to the eye to become darker as the background gets lighter. A more familiar example is a piece of paper that seems white when lying on a desk, but can appear totally black when used to shield the eyes while looking directly at a bright sky.

FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

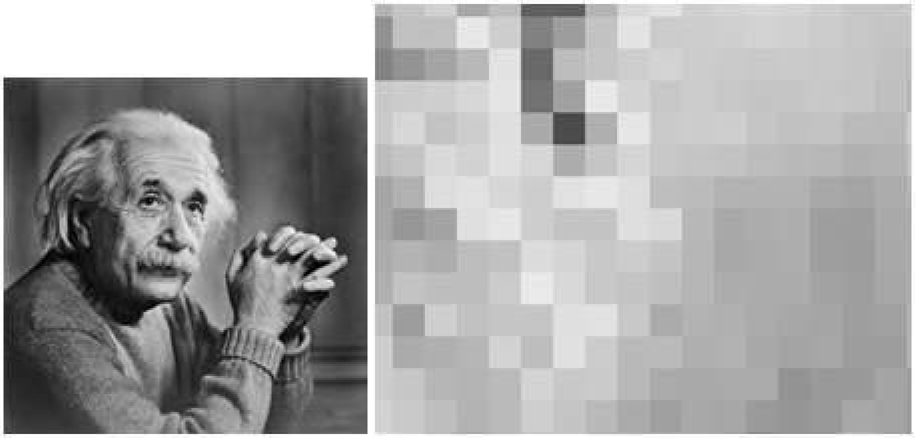
PIXEL

Pixel is the smallest element of an image. Each pixel correspond to any one value. In an 8-bit gray scale image, the value of the pixel between 0 and 255. The value of a pixel at any point correspond to the intensity of the light photons striking at that point. Each pixel store a value proportional to the light intensity at that particular location.

PEL

A pixel is also known as PEL. You can have more understanding of the pixel from the pictures given below.

In the above picture, there may be thousands of pixels, that together make up this image. We will zoom that image to the extent that we are able to see some pixels division. It is shown in the image below.

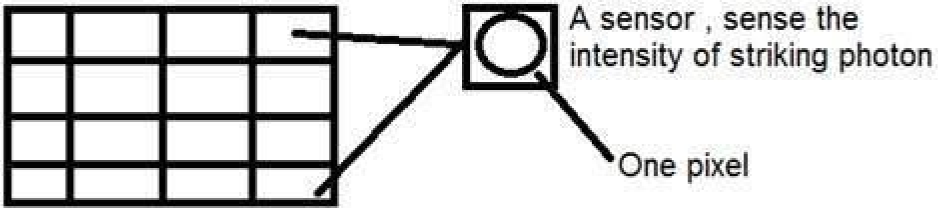


Relationship with CCD array

We have seen that how an image is formed in the CCD array. So a pixel can also be defined as

The smallest division the CCD array is also known as pixel.

Each division of CCD array contains the value against the intensity of the photon striking to it. This value can also be called as a pixel.



Calculation of total number of pixels

We have define an image as a two dimensional signal or matrix. Then in that case the number of PEL would be equal to the number of rows multiply with number of columns.

This can be mathematically represented as below:

Total number of pixels = number of rows (X) number of columns

Or we can say that the number of (x,y) coordinate pairs make up the total number of pixels.

We will look in more detail in the tutorial of image types, that how do we calculate the pixels in a color image.

Gray level

The value of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level.

We will see in more detail about the value of the pixels in the image storage and bits per pixel tutorial, but for now we will just look at the concept of only one pixel value.

Pixel value.(0)

As it has already been define in the beginning of this tutorial, that each pixel can have only one value and each value denotes the intensity of light at that point of the image.

We will now look at a very unique value 0. The value 0 means absence of light. It means that 0 denotes dark, and it further means that when ever a pixel has a value of 0, it means at that point, black color would be formed.

Have a look at this image matrix

0	0	0
0	0	0
0	0	0

Now this image matrix has all filled up with 0. All the pixels have a value of 0. If we were to calculate the total number of pixels form this matrix, this is how we are going to do it.

Total no of pixels = total no. of rows X total no. of columns

= 3 X 3

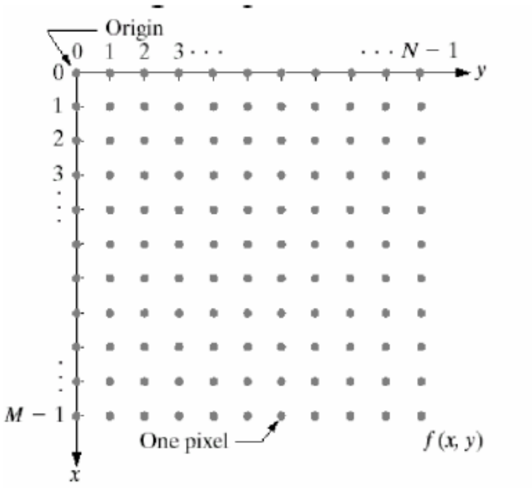
= 9.

It means that an image would be formed with 9 pixels, and that image would have a dimension of 3 rows and 3 column and most importantly that image would be black.

The resulting image that would be made would be something like this

Now why is this image all black. Because all the pixels in the image had a value of 0.

Digital Image Representation



Neighbors of a Pixel

A pixel *p* at coordinates 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 Each pixel is a unit distance from , and some of the neighbor locations of *p* lie outside the digital image if is on the border of the image.

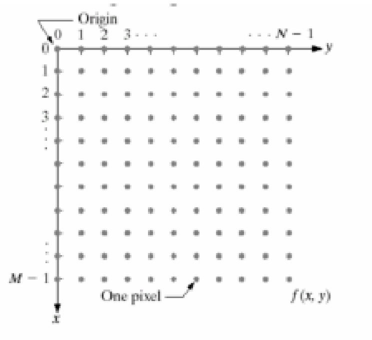
The four *diagonal* neighbors of *p* have coordinates

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

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 neighbor locations in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of the image.

Some Basic Relationships Between Pixels

Neighbors of a Pixel



$$N_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

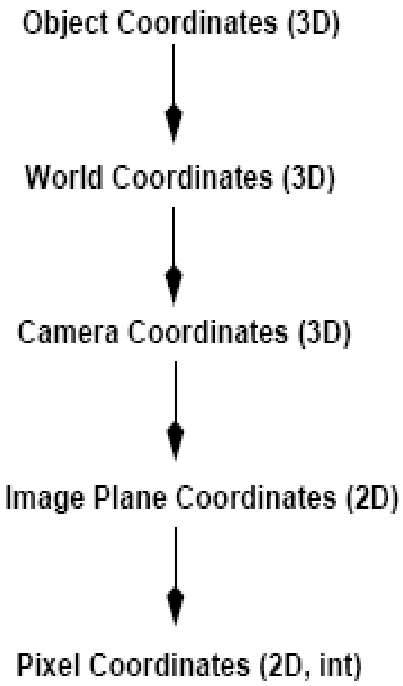
$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1).$

$$N_D = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

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

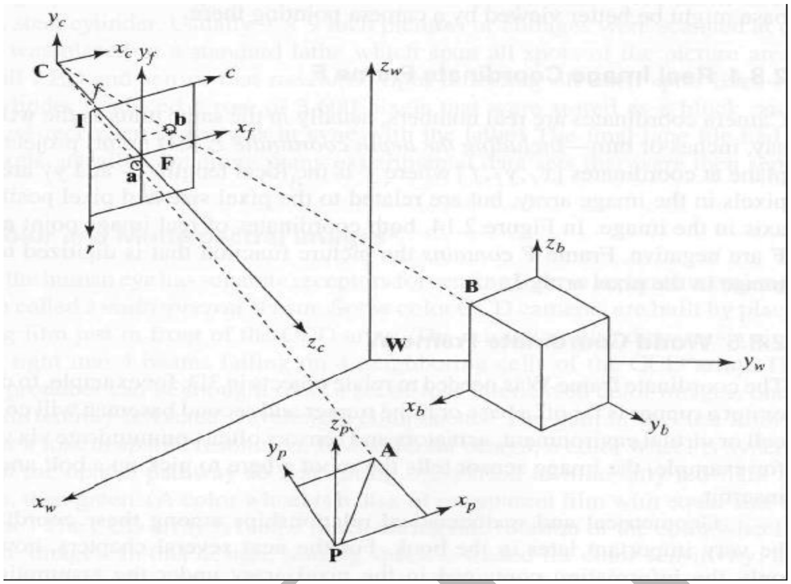
$$N_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

COORDINATE CONVENTIONS



(1) Object Coordinate Frame

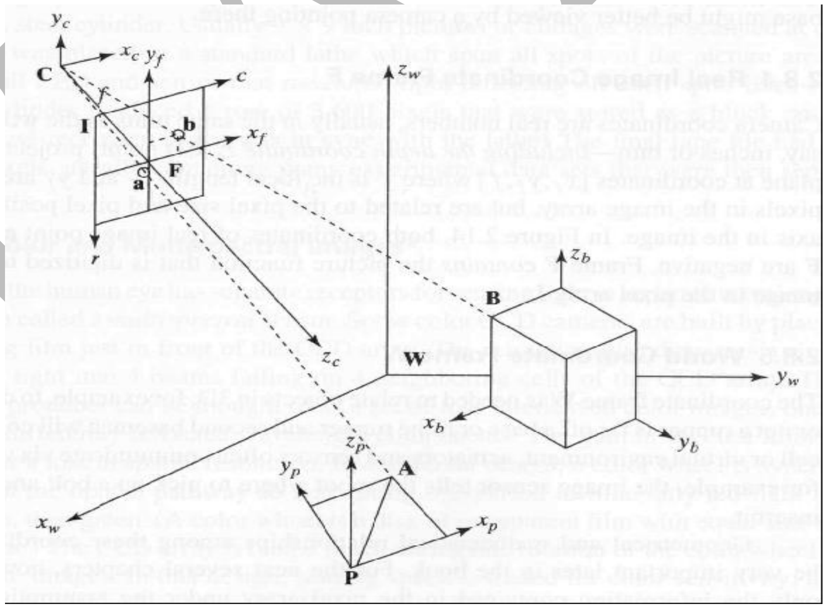
- 3D coordinate system: (x_b, y_b, z_b)
- Useful for modeling objects (i.e., check if a particular hole is in proper position relative to other holes)
- Object coordinates do not change regardless how the object is placed in the scene.



***Our notation:* $(X_o, Y_o, Z_o)^T$**

(2) World Coordinate Frame

- 3D coordinate system: (x_w, y_w, z_w)
- Useful for interrelating objects in 3D



***Our notation:* $(X_w, Y_w, Z_w)^T$**

(3) Camera Coordinate Frame

- 3D coordinate system: (x_c, y_c, z_c)
- Useful for representing objects with respect to the location of the camera.

Our notation: $(X_c, Y_c, Z_c)^T$

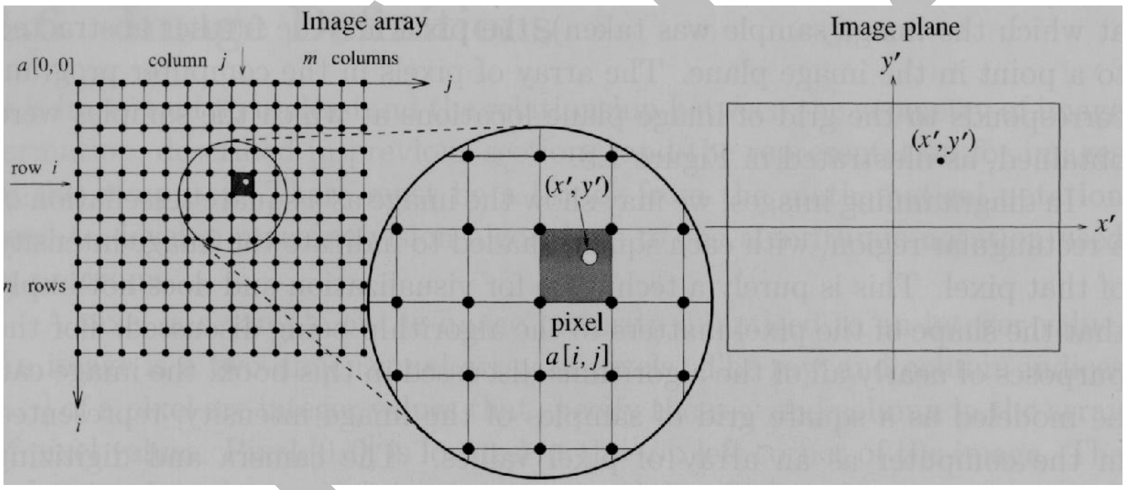
(4) Image Plane Coordinate Frame (i.e., CCD plane)

- 2D coordinate system: (x_f, y_f)
- Describes the coordinates of 3D points projected on the image plane.

  Our notation: $(x, y)^T$

(5) Pixel Coordinate Frame

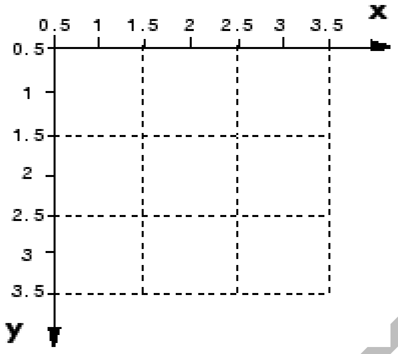
- 2D coordinate system: (c, r)
- Each pixel in this frame has integer pixel coordinates.



- **In general, the world and camera coordinate systems are not aligned.**

Spatial Coordinates

Spatial coordinates enable you to specify a location in an image with greater granularity than pixel coordinates. Such as, in the pixel coordinate system, a pixel is treated as a discrete unit, uniquely identified by an integer row and column pair, such as (3,4). In the spatial coordinate system, locations in an image are represented in terms of partial pixels, such as (3.3, 4.7).

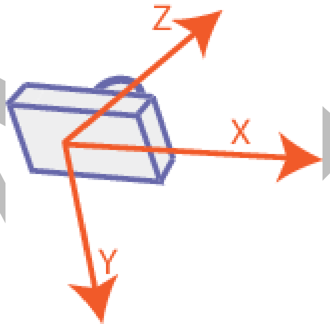


3-D Coordinate Systems

When you reconstruct a 3-D scene, you can define the resulting 3-D points in one of two coordinate systems. In a camera-based coordinate system, the points are defined relative to the center of the camera. In a calibration pattern-based coordinate system, the points are defined relative to a point in the scene.

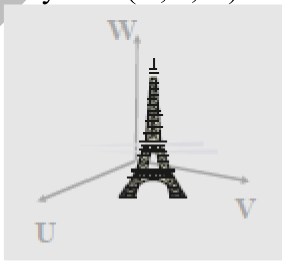
Camera-Based Coordinate System

Points represented in a camera-based coordinate system are described with the origin located at the optical center of the camera.



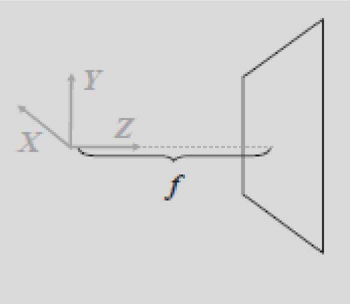
IMAGING GEOMETRY

Object of Interest in World Coordinate System (U,V,W)

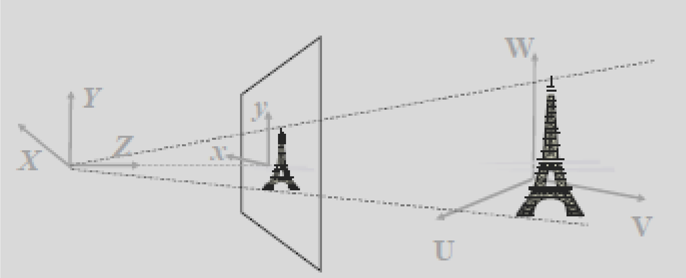


Camera Coordinate System (X,Y,Z).

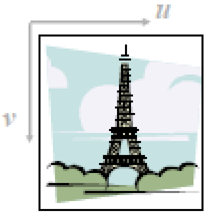
- Z is optic axis
- Image plane located f units out along optic axis
- f is called focal length

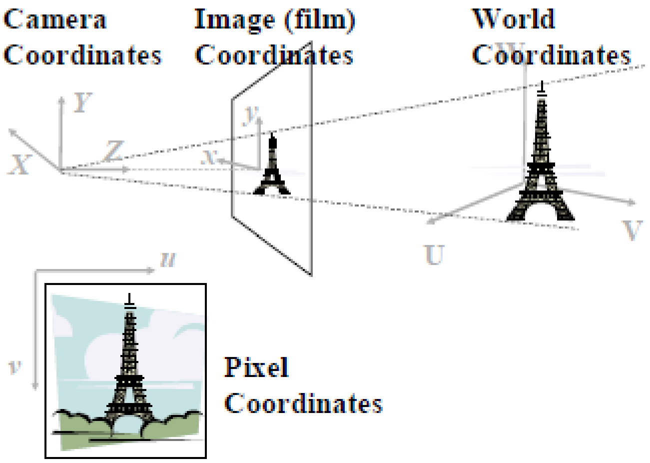


Forward Projection onto image plane.
3D (X,Y,Z) projected to 2D *(x,y)*



Our image gets digitized into pixel coordinates *(u,v)*

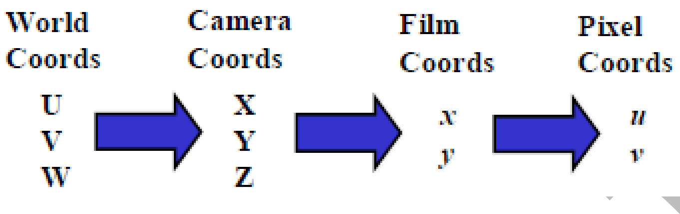




BASIC PERSPECTIVE PROJECTION

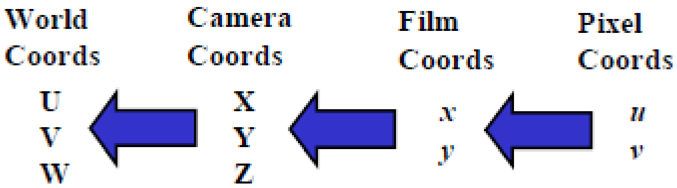
Forward Projection

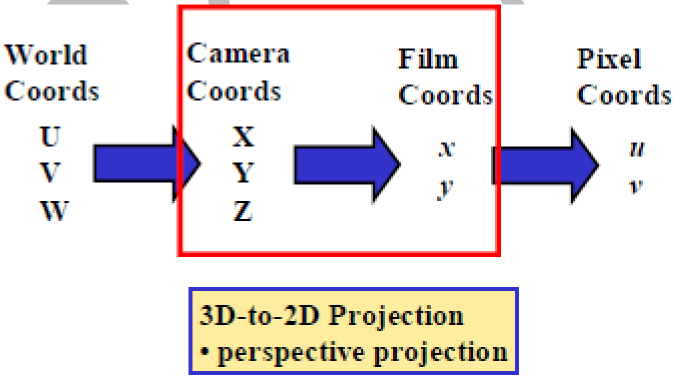
We want a mathematical model to describe how 3D World points get projected into 2D Pixel coordinates.



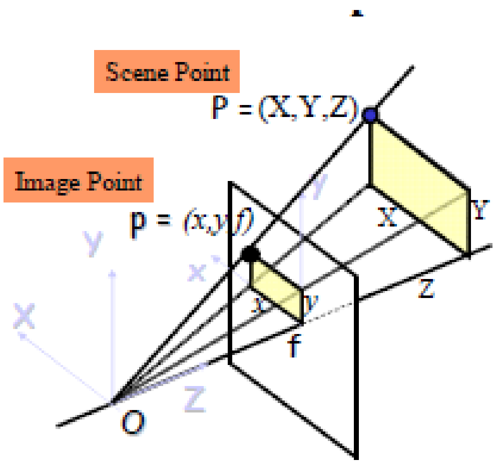
Backward Projection

Note, much of vision concerns trying to derive backward projection equations to recover 3D scene structure from images (via stereo or motion)





Basic Perspective Projection



Perspective Projection Eqns

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

Homogeneous Coordinates

Represent a 2D point (x,y) by a 3D point (x’,y’,z’) by adding a “fictitious” third coordinate.

By convention, we specify that given (x’,y’,z’) we can recover the 2D point (x,y) as

$$x = \frac{x'}{z'} \quad y = \frac{y'}{z'}$$

Note: (x,y) = (x,y,1) = (2x, 2y, 2) = (k x, ky, k)
for any nonzero k (can be negative as well as positive)

Perspective Matrix Equation

$$\begin{aligned} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z} \end{aligned} \quad \Longleftrightarrow \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

SAMPLING AND QUANTIZATION

In order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude.

In order to create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done:

Sampling

- **Digitizing the coordinate values is called *sampling.***

Quantization

- **Digitizing the amplitude values is called *quantization***

Sampling : related to coordinates values

Quantization : related to intensity values

The one-dimensional function shown in Fig. 2.16(b) is a plot of amplitude (gray level) values of the continuous image along the line segment *AB* in Fig. 2.16(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. 2.16(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.

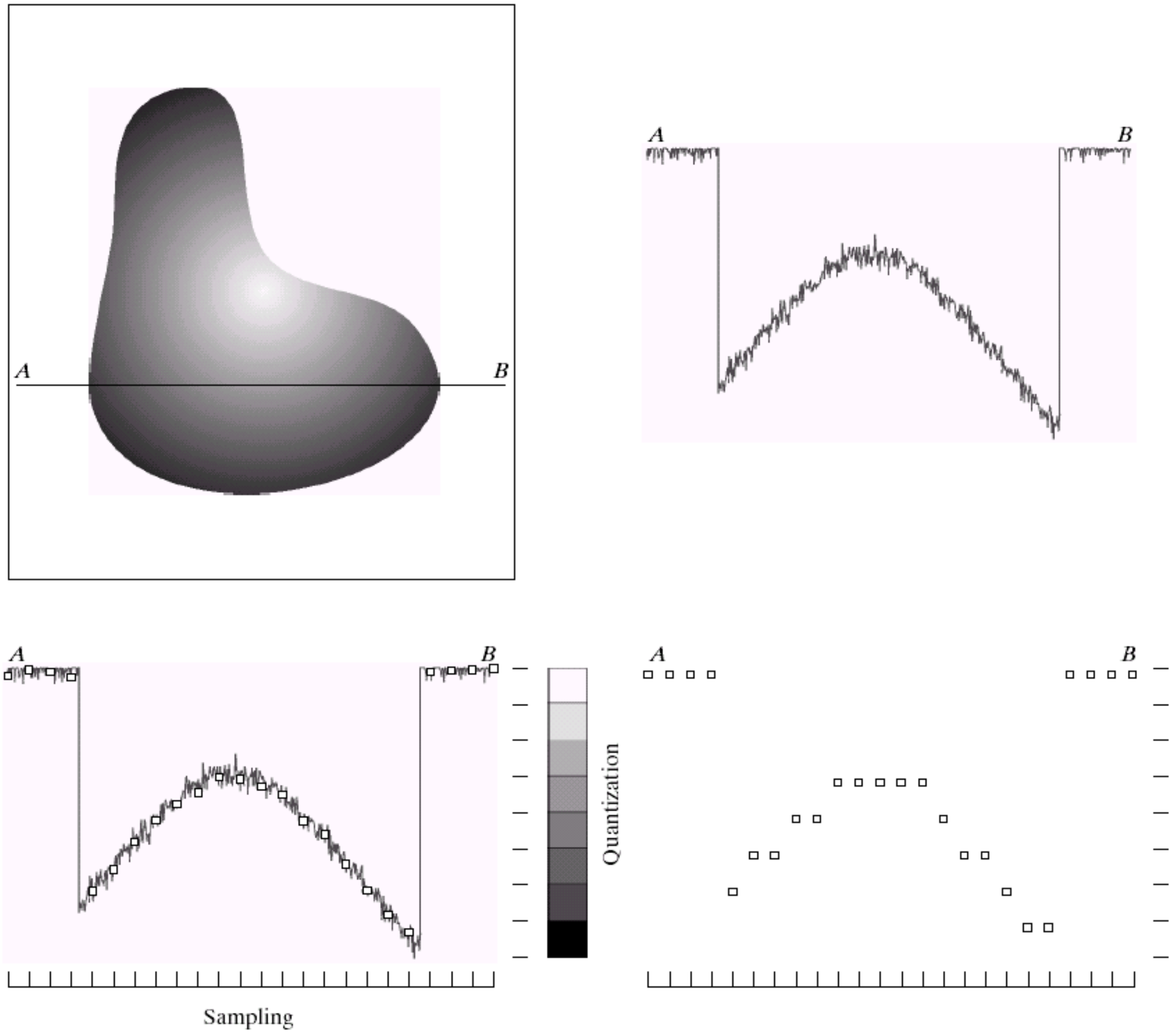
However, the values of the samples still span (vertically) a continuous range of gray-level values.

In order to form a digital function, the gray-level values also must be converted *(quantized)* into discrete quantities.

The right side of Fig. 2.16(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. 2.16(d).

Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.



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

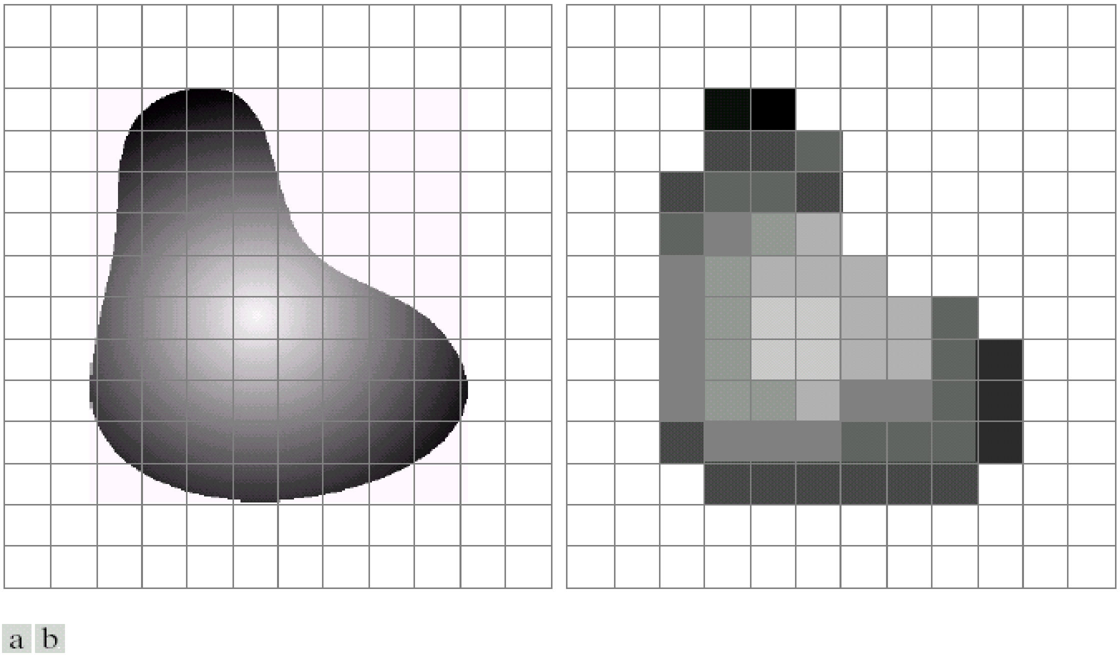


FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Spatial Domain Filtering

Filtering is a technique for modifying or enhancing an image. Spatial domain operation or filtering (processed value for the current pixel depends on both itself and surrounding pixels).

Hence Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel.

- ► **Spatial domain**

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

- ► **Transform domain**

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

Spatial Domain Process

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

$$f(x,y): \text{input image}$$

$$g(x,y): \text{output image}$$

$$T: \text{an operator on } f \text{ defined over}$$
$$\quad \text{a neighborhood of point } (x,y)$$

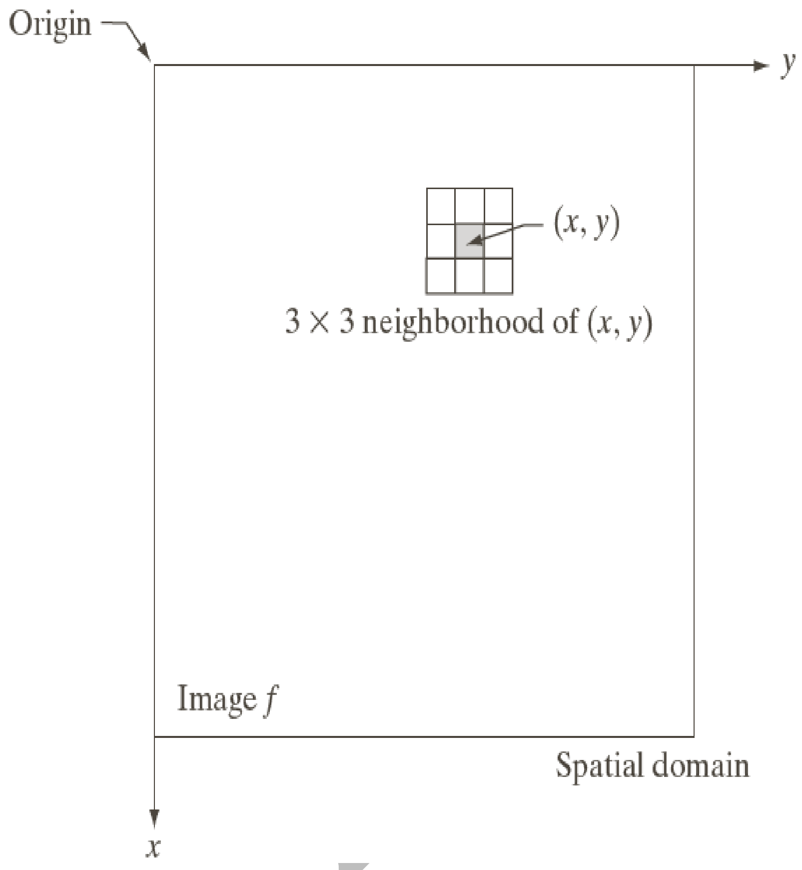


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Some neighborhood operations work with the values of the image pixels in the neighborhood *and* the corresponding values of a subimage that has the same dimensions as the neighborhood. The subimage is called a *filter, mask, kernel, template,* or *window,* with the first three terms being the most prevalent terminology. The values in a filter subimage are referred to as *coefficients,* rather than pixels.

INTENSITY TRANSFORMATIONS

Image enhancement

Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. Image enhancement has very applications. It is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c

The transformation function has been given below

$$s = T(r)$$

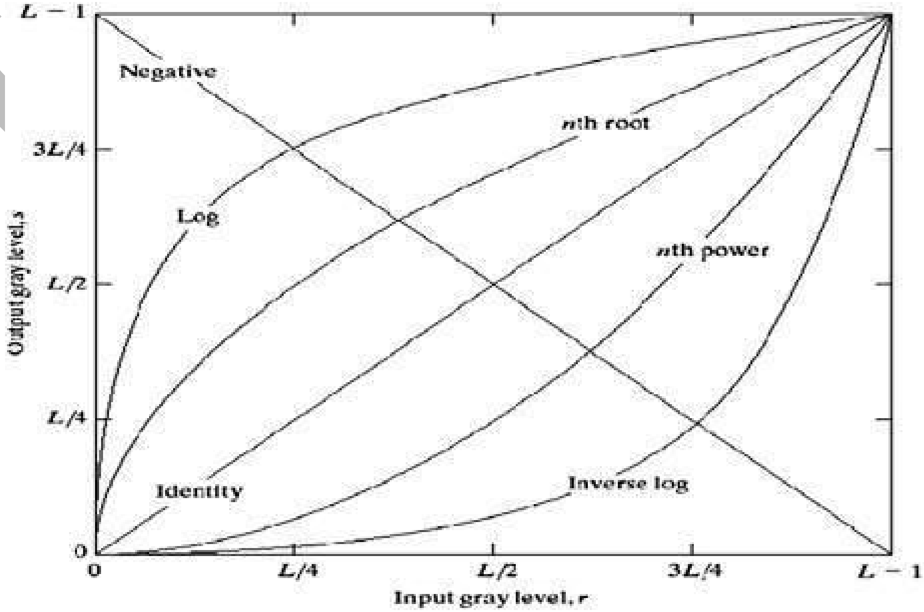
where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of r to each value of s. Image enhancement can be done through gray level transformations which are discussed below.

Gray level transformation

There are three basic gray level transformation.

- **Linear**
- **Logarithmic**
- **Power – law**

The overall graph of these transitions has been shown below.



Linear transformation

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation.

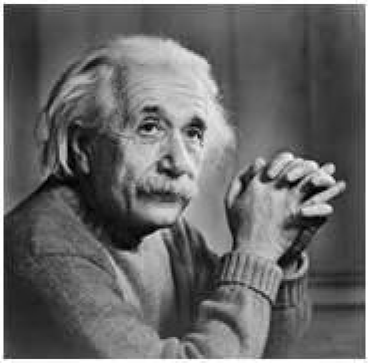
Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation.

Negative transformation

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.

The result is somewhat like this.

Input Image



Output Image

In this case the following transition has been done.

$$s = (L - 1) - r$$

since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this

$$s = 255 – r$$

So each value is subtracted by 255 and the result image has been shown above. So what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

It has been shown in the graph below.

Logarithmic transformations

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

Log transformation

The log transformations can be defined by this formula

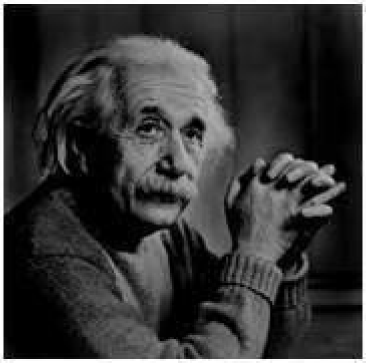
$$\mathbf{s = c \log(r + 1).}$$

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.

The value of c in the log transform adjust the kind of enhancement you are looking for.

Input Image



Log Tranform Image

The inverse log transform is opposite to log transform.

Power – Law transformations

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

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

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Correcting gamma.

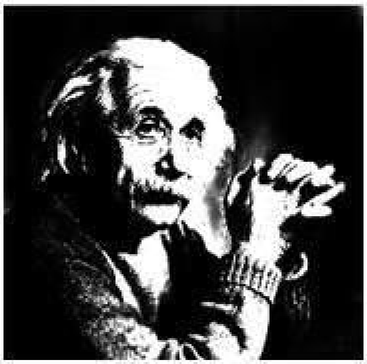
s=cr^γ

s=cr^(1/2.5)

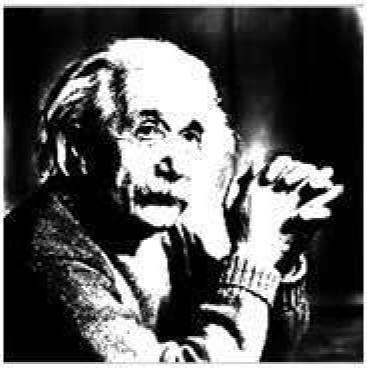
The same image but with different gamma values has been shown here.

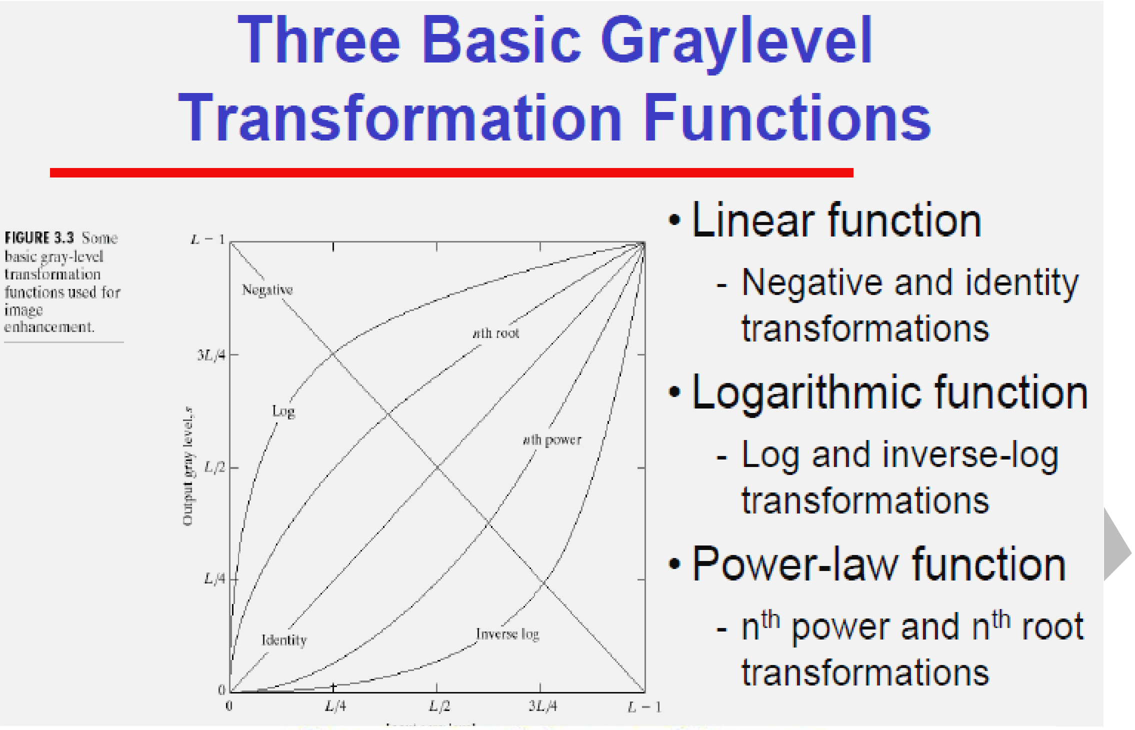
For example

Gamma = 10



Gamma = 8





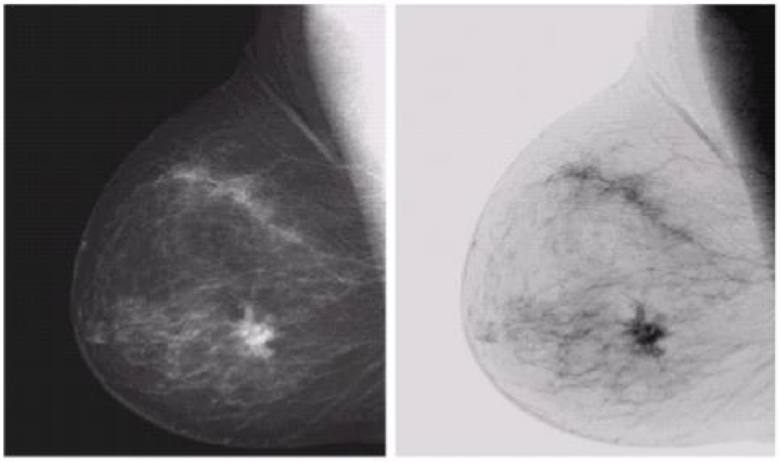
Three Basic Graylevel Transformation Functions

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

- Linear function
  - Negative and identity transformations
- Logarithmic function
  - Log and inverse-log transformations
- Power-law function
  - n^{th} power and n^{th} root transformations

Image Negatives

- Negative transformation : $s = (L-1) - r$
- Reverses the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when black area is large.

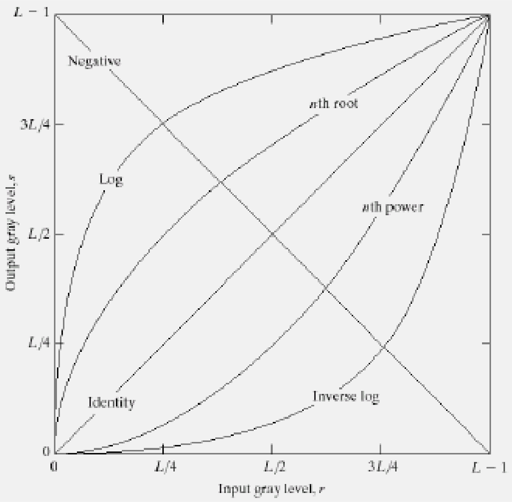


a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Log Transformations

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



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

- c is constant and $r \geq 0$
- Log curve maps a narrow range of low graylevels in input image into a wider range of output levels.
- Expands range of dark image pixels while shrinking bright range.
- Inverse log expands range of bright image pixels while shrinking dark range.

Power-Law Transformations

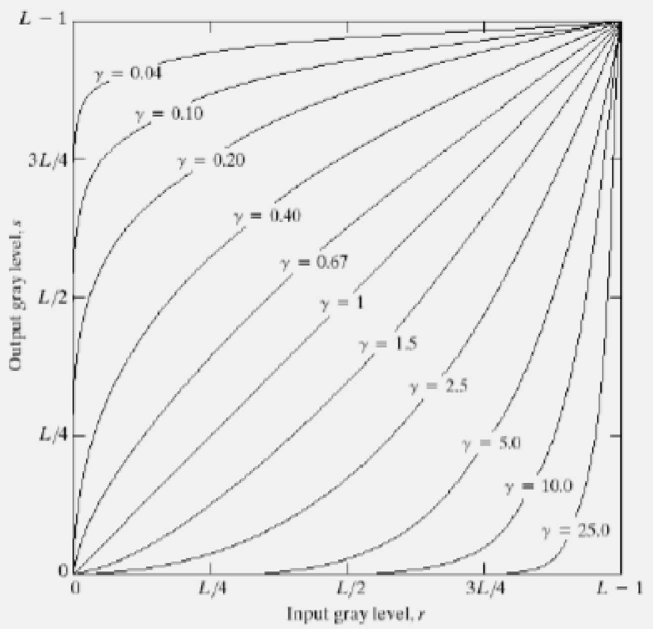


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

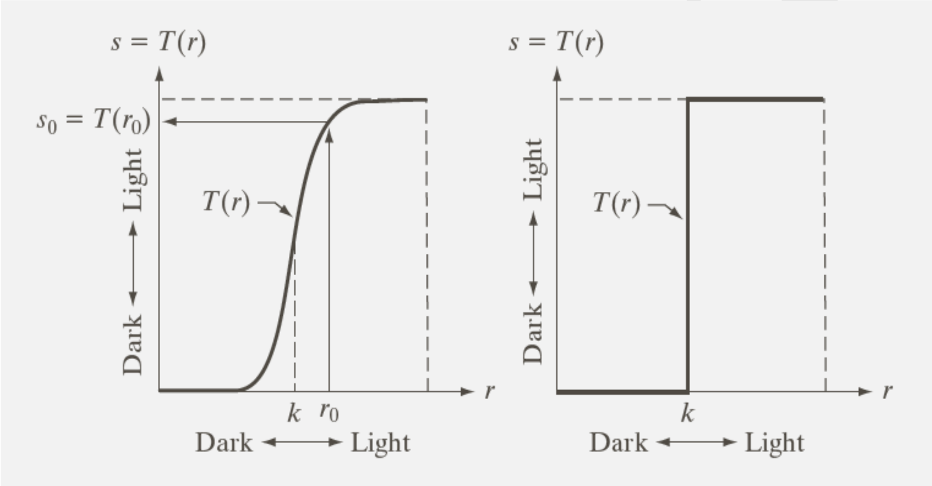
$$s = cr^{\gamma}$$

- c and γ are positive constants
- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- $c = \gamma = 1$ ⇨ identity function

Contrast Stretching

- **Contrast Stretching**
  — Expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

- **Intensity-level Slicing**
  — Highlighting a specific range of intensities in an image often is of interest.



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

For example, if *T(r)* has the form shown in Fig. 3.2(a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and Brightening the levels above m in the original image.

In this technique, known as *contrast stretching,* the values of *r* below k are compressed by the transformation function into a narrow range of *s,* toward black. The opposite effect takes place for values of *r* above k.

In the limiting case shown in Fig. 3.2(b), *T(r)* produces a two-level (binary) image. A mapping of this form is called a *thresholding* function. Some fairly simple, yet powerful, processing approaches can be formulated with gray-level transformations. Because enhancement at any point in an image depends only on the gray level at that point.

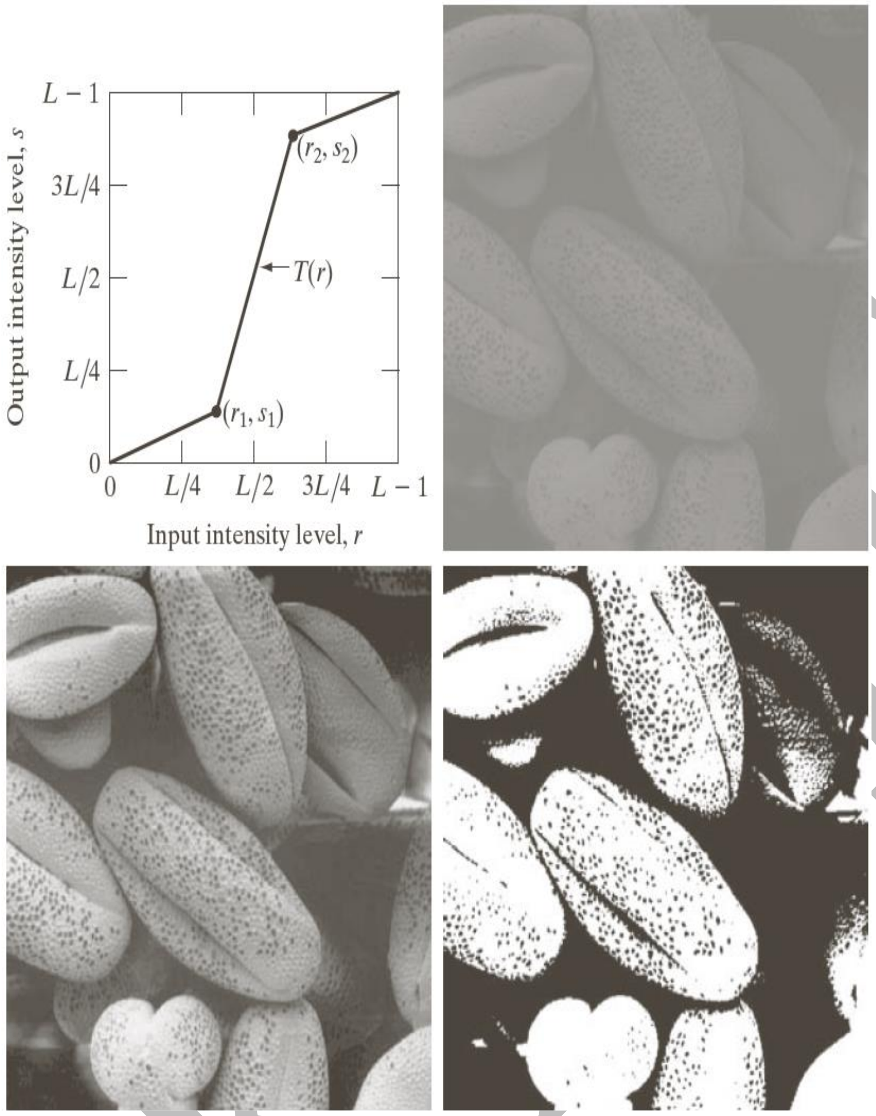
Contrast stretching
One of the simplest piecewise linear functions is a contrast-stretching transformation.

Low-contrast images can result from poor illumination. lack of dy**namic range in the imaging sensor or even wrong setting of a lens aperture** during image acquisition. The idca behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.

Figure 3.10(a) shows a typical transformation used for contrast stretching.
The locations of points *(r1,s1)* and *(r2,s2)* control the shape of the transformation in this category often are referred to as *point* function.

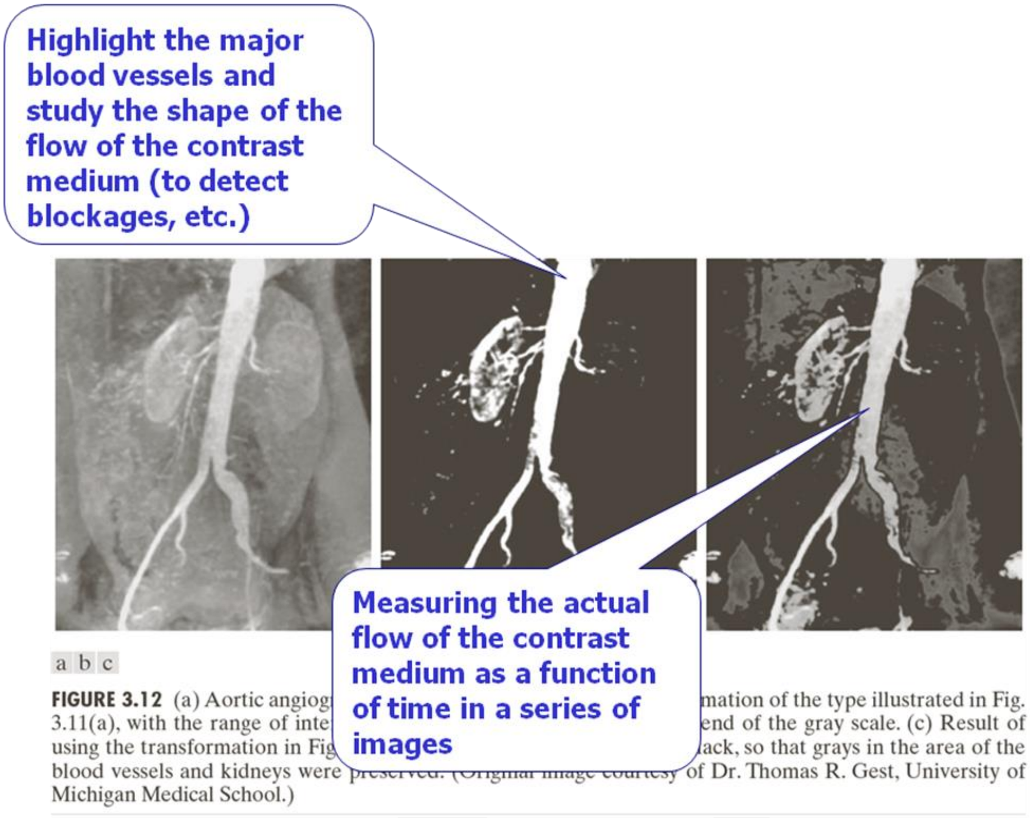
If *rl* = s1 and *r2* = *S2* the transformation is a linear function that produces no changes in gray levels. If *rl* = *r2*' *s1* = 0 and s2 = *L* - 1, the transformation becomes a *thresholding function* that creates a binary image, as illustrated in Fig. 3.2(b).

Intermediate values of (r1,s1) and (r2,s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.



a b
c d

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



Highlight the major blood vessels and study the shape of the flow of the contrast medium (to detect blockages, etc.)

Measuring the actual flow of the contrast medium as a function of time in a series of images

FIGURE 3.12 (a) Aortic angiog mation of the type illustrated in Fig. 3.11(a), with the range of inte end of the gray scale. (c) Result of using the transformation in Fig ack, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

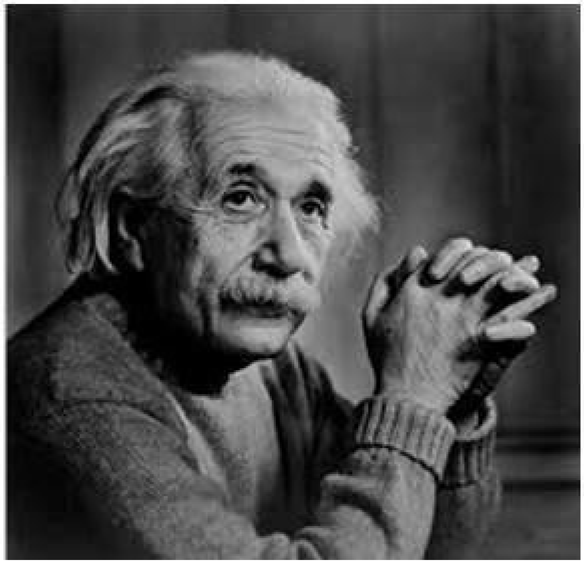
There are two methods of enhancing contrast. The first one is called Histogram stretching that increase contrast. The second one is called Histogram equalization that enhance contrast and it has been discussed in our tutorial of histogram equalization.

Before we will discuss the histogram stretching to increase contrast, we will briefly define contrast.

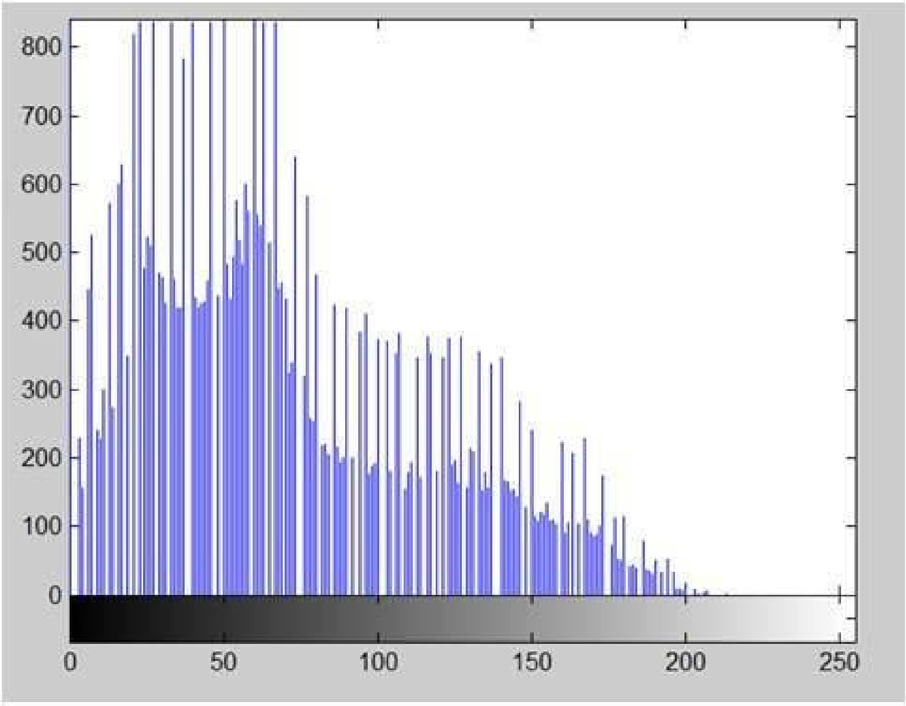
Contrast

Contrast is the difference between maximum and minimum pixel intensity.

Consider this image.



The histogram of this image is shown below.



Now we calculate contrast from this image.

Contrast = 225.

Now we will increase the contrast of the image.

HISTOGRAM PROCESSING

Histogram

Histogram is nothing but a graph that shows frequency of occurrence of data. Histograms has many use in image processing, out of which we are going to discuss one user here which is called histogram sliding.

Brightness

Brightness is a relative term. Brightness can be defined as intensity of light emit by a particular light source.

Contrast

Contrast can be defined as the difference between maximum and minimum pixel intensity in an image.

Using Histogram Statistics for Image Enhancement

A histogram is a graph. A graph that shows frequency of anything. Usually histogram have bars that represent frequency of occurring of data in the whole data set.

A Histogram has two axis the x axis and the y axis.

The x axis contains event whose frequency you have to count.

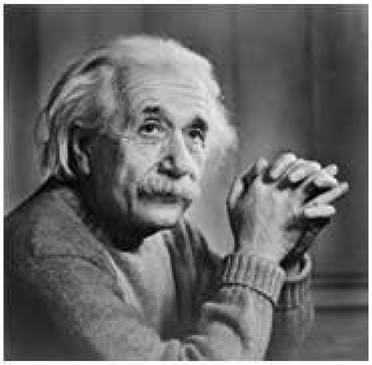
The y axis contains frequency.

HISTOGRAM EQUALIZATION

Histogram equalization is used to enhance contrast. It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased.

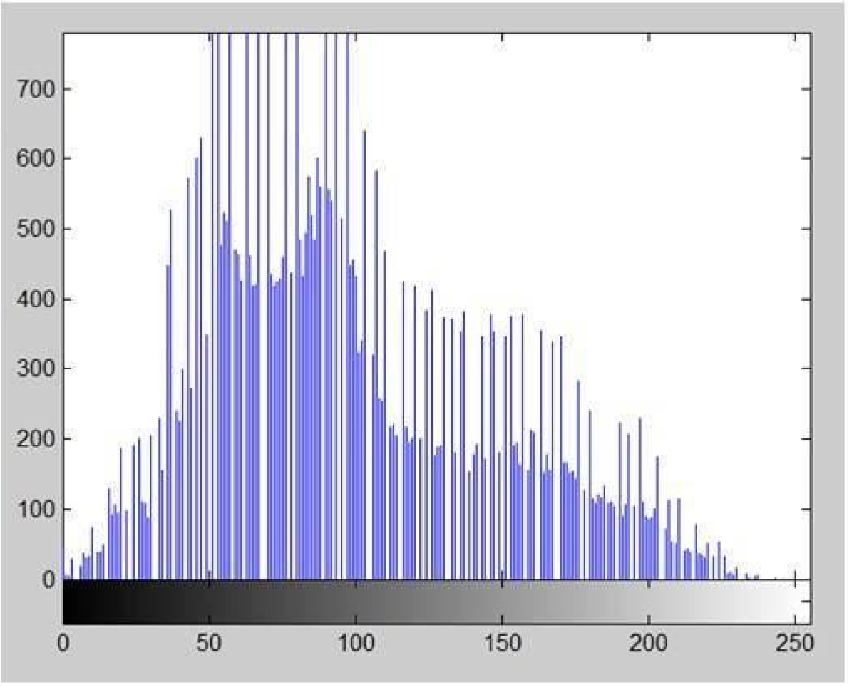
Lets start histogram equalization by taking this image below as a simple image.

Image



Histogram of this image

The histogram of this image has been shown below.



Now we will perform histogram equalization to it.

Histogram $h(r_k) = n_k$

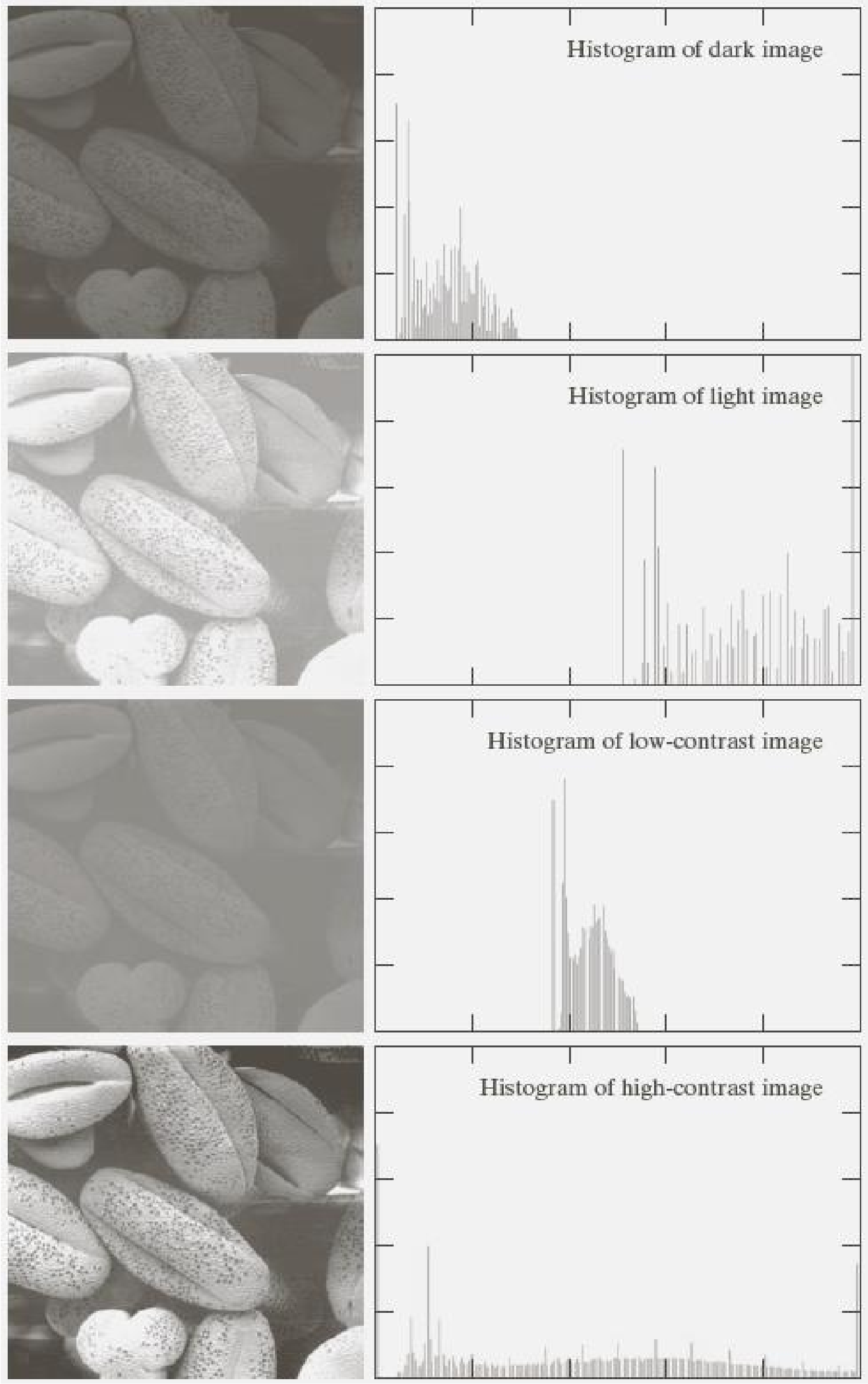
r_k is the k^{th} intensity value

n_k is the number of pixels in the image with intensity r_k

Normalized histogram $p(r_k) = \frac{n_k}{MN}$

n_k: the number of pixels in the image of size $M \times N$ with intensity r_k

The horizontal axis of each histogram plot corresponds to gray level values, r_k. The vertical axis corresponds to values of $h(r_k) = n_k$ or $p(r_k) = n_k/n$ if the values are normalized. Thus, as indicated previously, these histogram plots are simply plots of $h(r_k) = n_k$ versus r_k or $p(r_k) = n_k/n$ versus r_k.

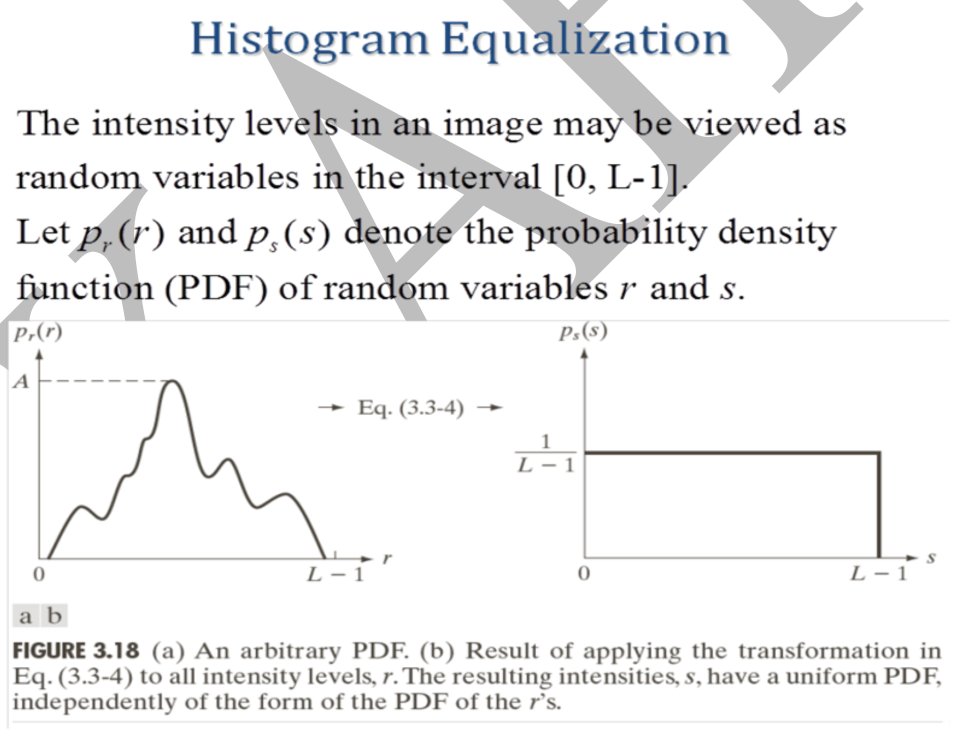


Histogram Equalization Image

- We note in the **dark image** that the components of the histogram are concentrated on the low (dark) side of the gray scale.
- Similarly, the components of the histogram of the **bright image** are biased toward the high side of the gray scale.
- An image with **low contrast** has a histogram that will be narrow and will be centered toward the middle of the gray scale.
- For a **monochrome image** this implies a dull washed-out gray look.
- Finally. we see that the components of the histogram in the **high-contrast image** cover a broad range of the gray scale and, further. that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others.
- Intuitively, it is reasonable to conclude that an image whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly. will have an appearance of high contrast and will exhibit a large variety of gray tones.

Histogram Equalization

- **Histogram equalization is a technique for adjusting image intensities to enhance contrast.**



Histogram Equalization

The intensity levels in an image may be viewed as random variables in the interval [0, L-1].

Let $p_r(r)$ and $p_s(s)$ denote the probability density function (PDF) of random variables r and s.

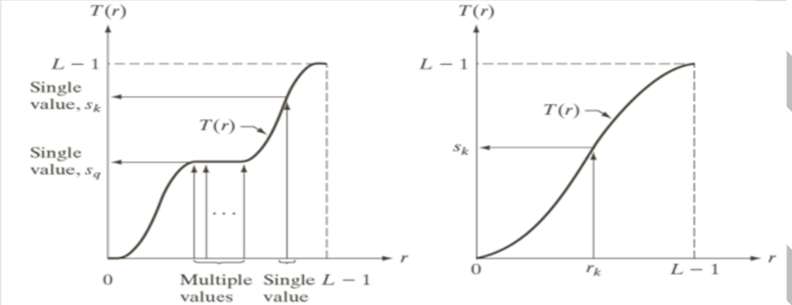
FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization

$$s = T(r) \qquad 0 \le r \le L-1$$

a. T(r) is a strictly monotonically increasing function in the interval $0 \le r \le L-1$;

b. $0 \le T(r) \le L-1 \quad \text{for} \quad 0 \le r \le L-1$.



a b

FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Spatial Filtering (cont'd)

- Spatial filtering is defined by:
  (1) A neighborhood
  (2) An operation that is performed on the pixels inside the neighborhood

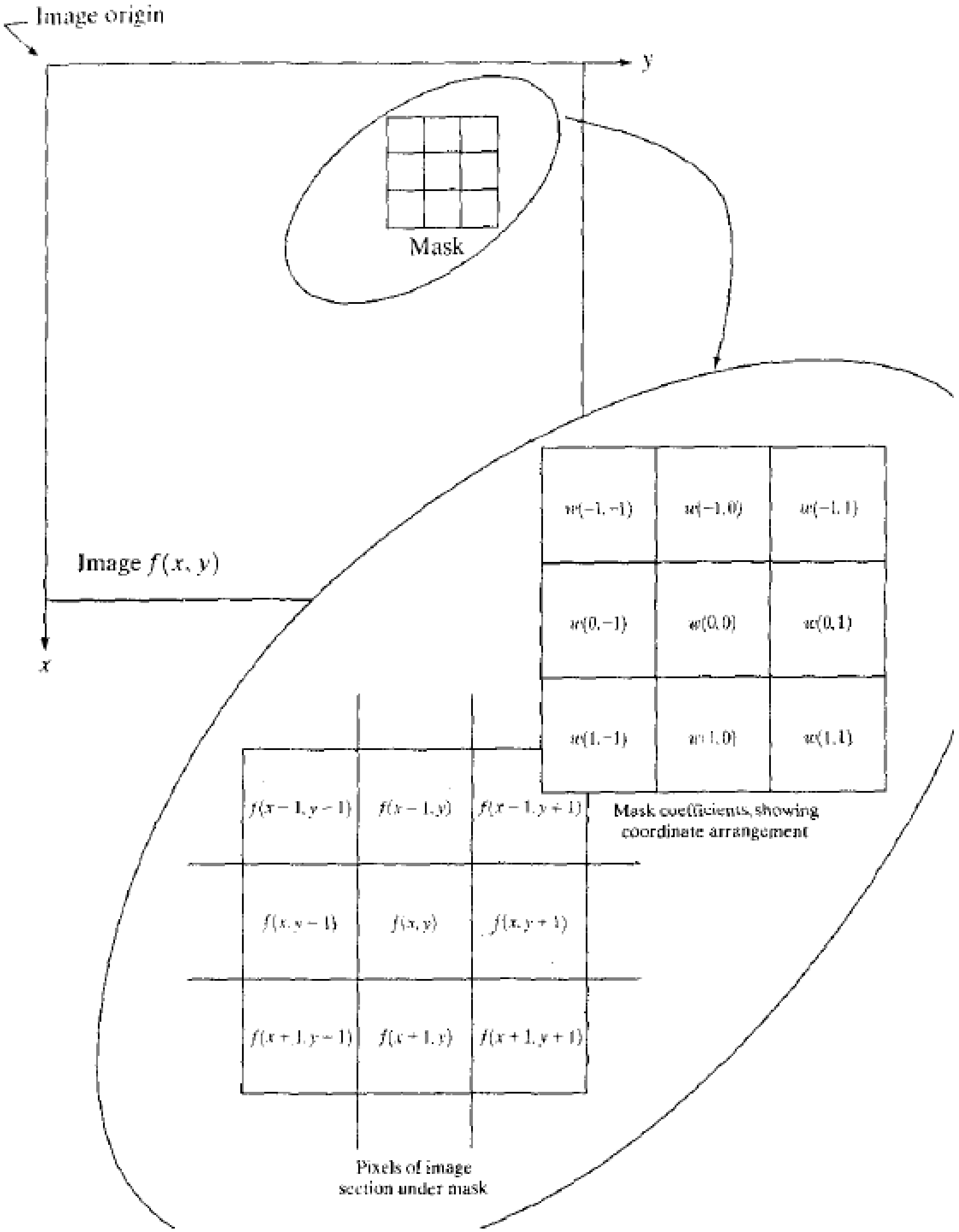


Image origin
y
Mask
Image $f(x, y)$
x
w(−1, −1)
w(−1, 0)
w(−1, 1)
w(0, −1)
w(0, 0)
w(0, 1)
w(1, −1)
w(1, 0)
w(1, 1)
Mask coefficients, showing coordinate arrangement
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image section under mask

When interest lies on the response, R, of an $m \times n$ mask at any point (x, y), and not on the mechanics of implementing mask convolution, it is common practice to simplify the notation by using the following expression:

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} \\ &= \sum_{i=1}^{mn} w_i z_i \end{aligned} \tag{3.5-2}$$

where the w's are mask coefficients, the z's are the values of the image gray levels corresponding to those coefficients, and mn is the total number of coefficients in the mask. For the 3×3 general mask shown in Fig. 3.33 the response at any point (x, y) in the image is given by

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \ldots w_9 z_9 \\ &= \sum_{i=1}^{9} w_i z_i . \end{aligned} \tag{3.5-3}$$

FIGURE 3.33 Another representation of a general 3×3 spatial filter mask.

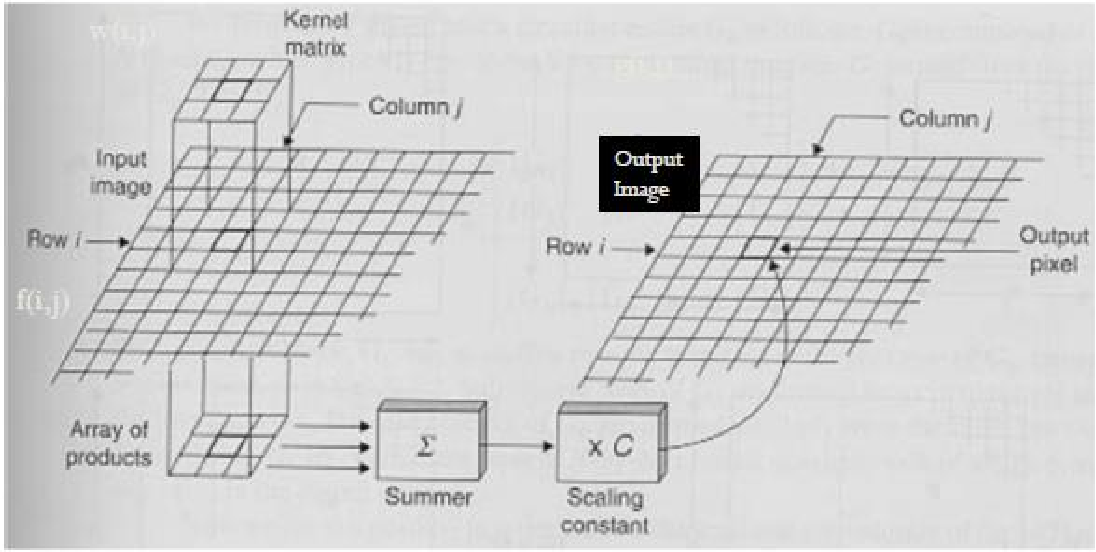
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

CORRELATION AND CONVOLUTION

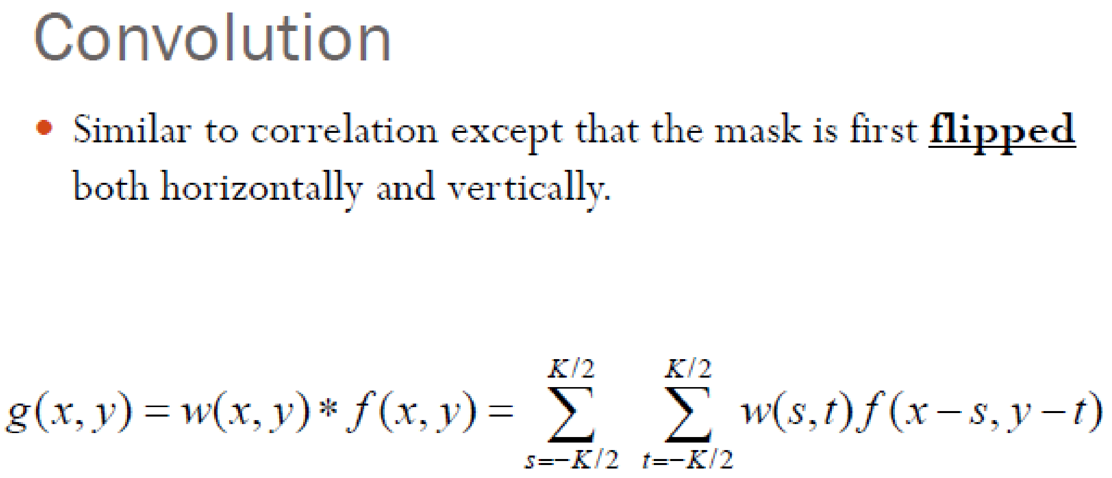
Linear Spatial Filtering Methods

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

Correlation



$$g(x,y) = w(x,y) \bullet f(x,y) = \sum_{s=-K/2}^{K/2} \sum_{t=-K/2}^{K/2} w(s,t) f(x+s, y+t)$$



Convolution

- Similar to correlation except that the mask is first **flipped** both horizontally and vertically.

$$g(x,y) = w(x,y) * f(x,y) = \sum_{s=-K/2}^{K/2} \sum_{t=-K/2}^{K/2} w(s,t) f(x-s, y-t)$$

Convolution can achieve something, that the previous two methods of manipulating images can't achieve. Those include the blurring, sharpening, edge detection, noise reduction e.t.c.

It can be represented as.

It can be mathematically represented as two ways

g(x,y) = h(x,y) * f(x,y)

It can be explained as the "mask convolved with an image".

Or

g(x,y) = f(x,y) * h(x,y)

It can be explained as "image convolved with mask".

There are two ways to represent this because the convolution operator(*) is commutative. The h(x,y) is the mask or filter.

What is mask?

Mask is also a signal. It can be represented by a two dimensional matrix. The mask is usually of the order of 1x1, 3x3, 5x5, 7x7 . A mask should always be in odd number, because other wise you cannot find the mid of the mask. Why do we need to find the mid of the mask. The answer lies below, in topic of, how to perform convolution?

How to perform convolution?

In order to perform convolution on an image, following steps should be taken.

- Flip the mask (horizontally and vertically) only once
- Slide the mask onto the image.
- Multiply the corresponding elements and then add them
- Repeat this procedure until all values of the image has been calculated.

Example of convolution

Let's perform some convolution. Step 1 is to flip the mask.

Mask

Let's take our mask to be this.

1	2	3
4	5	6
7	8	9

Flipping the mask horizontally

3	2	1
6	5	4
9	8	7

Flipping the mask vertically

9	8	7
6	5	4
3	2	1

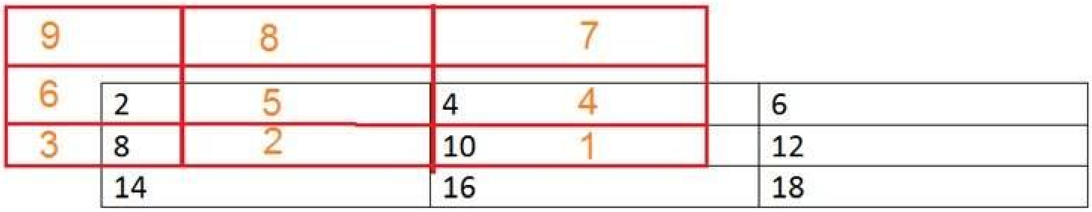
Image

Let's consider an image to be like this

2	4	6
8	10	12
14	16	18

Convolution

Convolving mask over image. It is done in this way. Place the center of the mask at each element of an image. Multiply the corresponding elements and then add them , and paste the result onto the element of the image on which you place the center of mask.



The box in red color is the mask, and the values in the orange are the values of the mask. The black color box and values belong to the image. Now for the first pixel of the image, the value will be calculated as

First pixel = (5*2) + (4*4) + (2*8) + (1*10)

= 10 + 16 + 16 + 10

= 52

Place 52 in the original image at the first index and repeat this procedure for each pixel of the image.

Spatial Convolution

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

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

Spatial Correlation

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

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

Spatial Filtering

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

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

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

Linear vs Non-Linear Spatial Filtering Methods

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

w1	w2	w3
w4	w5	w6
w7	w8	w9

`z5' = R = w1z1 + w2z2 + ... + z9w9`

- Methods that do not satisfy the above property are called non-linear.
  - e.g.,

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

What is a mask

A mask is a filter. Concept of masking is also known as spatial filtering. Masking is also known as filtering. In this concept we just deal with the filtering operation that is performed directly on the image.

A sample mask has been shown below

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

What is filtering

The process of filtering is also known as convolving a mask with an image. As this process is same of convolution so filter masks are also known as convolution masks.

How it is done

The general process of filtering and applying masks is consists of moving the filter mask from point to point in an image. At each point (x,y) of the original image, the response of a filter is calculated by a pre defined relationship. All the filters values are pre defined and are a standard.

Types of filters

Generally there are two types of filters. One is called as linear filters or smoothing filters and others are called as frequency domain filters.

Why filters are used?

Filters are applied on image for multiple purposes. The two most common uses are as following:

- Filters are used for Blurring and noise reduction
- Filters are used for edge detection and sharpness

Blurring and noise reduction

Filters are most commonly used for blurring and for noise reduction. Blurring is used in pre processing steps, such as removal of small details from an image prior to large object extraction.

Masks for blurring

The common masks for blurring are.

- **Box filter**
- **Weighted average filter**

In the process of blurring we reduce the edge content in an image and try to make the transitions between different pixel intensities as smooth as possible.

Noise reduction is also possible with the help of blurring.

Edge Detection and sharpness

Masks or filters can also be used for edge detection in an image and to increase sharpness of an image.

Smoothing filters

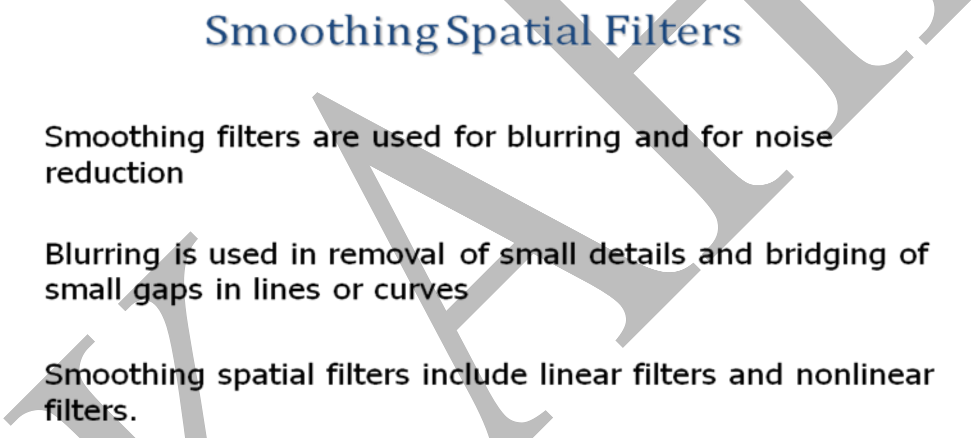
Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction. and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

Smoothing Linear Filters

The output (response) of a smoothing. linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called *averaging filters* also are referred to a *lowpass fillers.*

The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask. this process results in an image with reduced "sharp" transitions in gray levels.

However. edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels. so averaging filters have the undesirable side effect that they blur edges



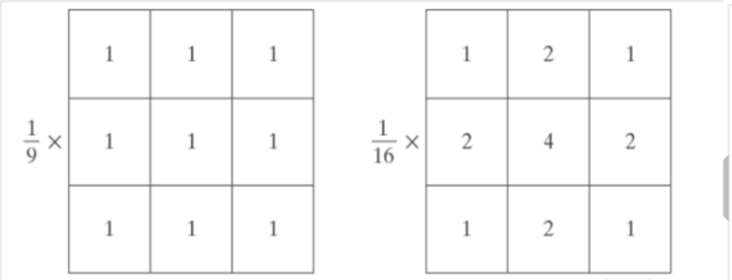
Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction

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

Smoothing spatial filters include linear filters and nonlinear filters.

Two Smoothing Averaging Filter Masks



a b

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

Box filter :
Figure 3.32 shows two 3 X 3 smoothing filters. Use of thefirst filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask into Eq. (3.5-3):

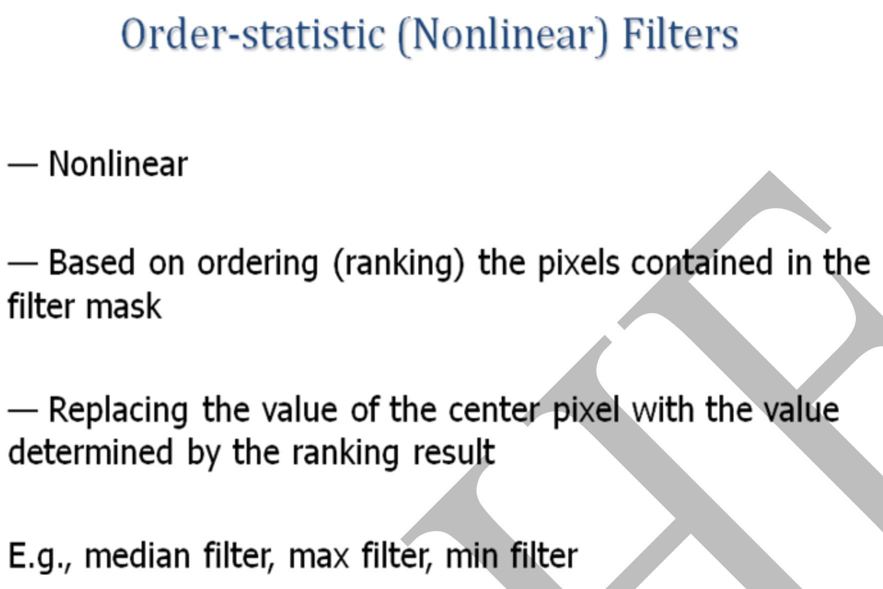
$$R = \frac{1}{9} \sum_{i=1}^{9} z_i,$$

which is the average of the gray levels of the pixels in the 3 x 3 neighborhood defined by the mask. Note that, instead of being 1/9, the coefficients of the filter are all 1's. The idea here is that it is computationally more efficient to have coefficients valued 1. At the end of the filtering process the entire image is divided by 9. An m X n mask would have a normalizing constant equal to 1/*mn.*

A spatial averaging filter in which all coefficients are equal is sometimes called a *box filter.*

Weighted average :
The second mask yields a so-called *weighted average,* terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.

Non-linear Spatial Filters



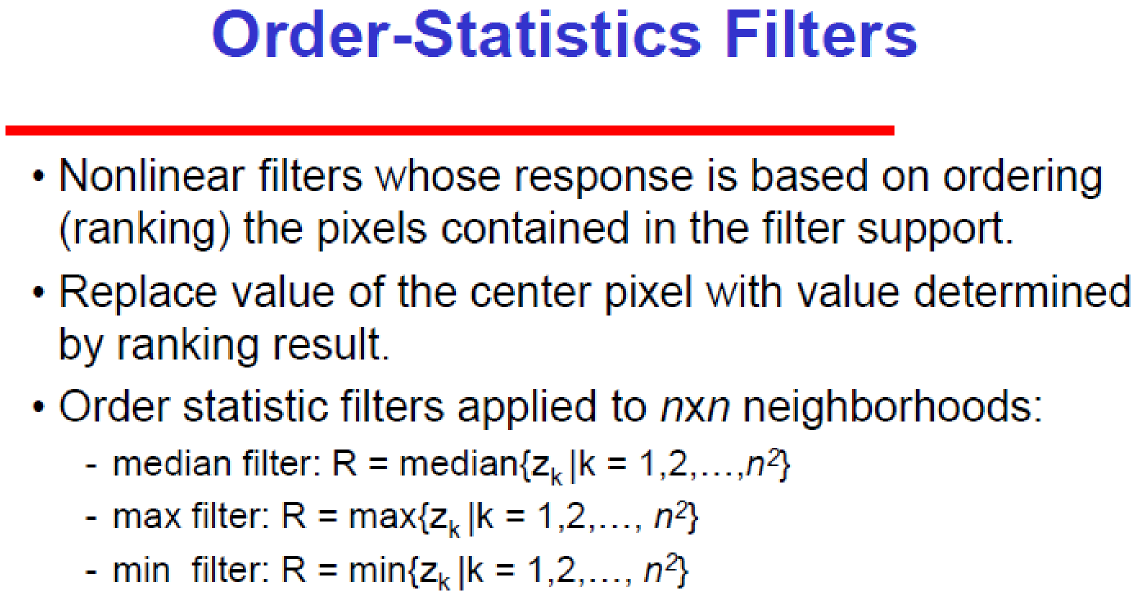
Order-statistic (Nonlinear) Filters

— Nonlinear

— Based on ordering (ranking) the pixels contained in the filter mask

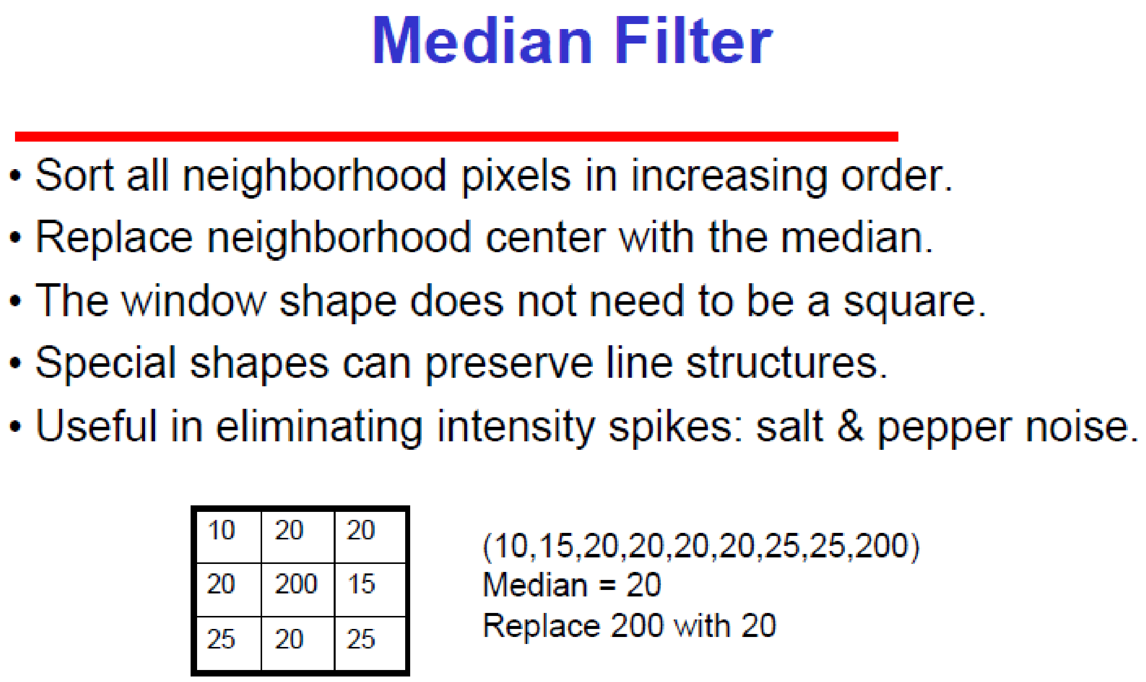
— Replacing the value of the center pixel with the value determined by the ranking result

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



Order-Statistics Filters

- Nonlinear filters whose response is based on ordering (ranking) the pixels contained in the filter support.
- Replace value of the center pixel with value determined by ranking result.
- Order statistic filters applied to *nxn* neighborhoods:
  - median filter: $R = \text{median}\{z_k | k = 1,2,\ldots,n^2\}$
  - max filter: $R = \max\{z_k | k = 1,2,\ldots, n^2\}$
  - min filter: $R = \min\{z_k | k = 1,2,\ldots, n^2\}$



Median Filter

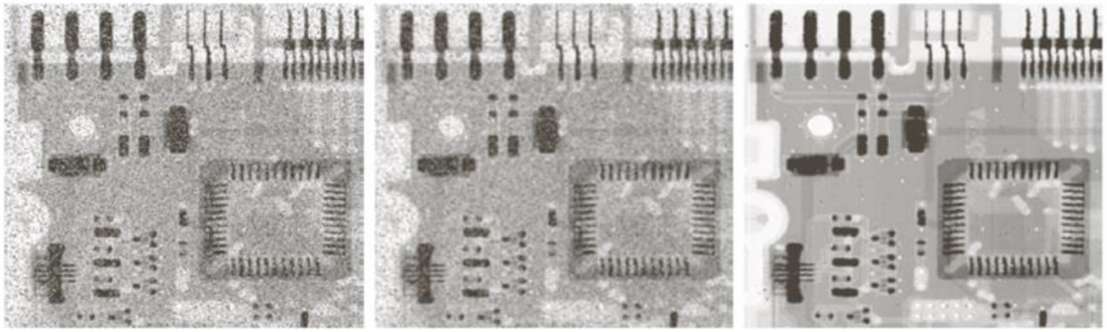
- Sort all neighborhood pixels in increasing order.
- Replace neighborhood center with the median.
- The window shape does not need to be a square.
- Special shapes can preserve line structures.
- Useful in eliminating intensity spikes: salt & pepper noise.

10	20	20
20	200	15
25	20	25

(10,15,20,20,20,20,25,25,200)
Median = 20
Replace 200 with 20

Min Filter	The intensity of the pixel at the center of the mask is replaced by the minimum intensity value of any pixel within the mask. This filter is used to find the dark points in an image.
Max Filter	The intensity of the pixel at the center of the mask is replaced by the maximum intensity value of any pixel within the mask. This filter is used to find the bright points in an image.

Example: Use of Median Filtering for Noise Reduction



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening Spatial Filters :

The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.

In the last section, we saw that image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood.

Since averaging is analogous to integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation.

This, in fact, is the case, and the discussion in this section deals with various ways of defining and implementing operators for sharpening by digital differentiation.

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

Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values.

Looking some of the fundamental properties of these derivatives in a digital context. To simplify the explanation; we focus attention On one-dimensional derivatives. In particular, we are interested in the behavior of these derivatives in areas of constant gray level (flat segments), at the onset and end of discontinuities (step and ramp discontinuities), and along gray-level ramps. These types of discontinuities can be used to model noise points, lines, and edges in an image. The behavior of derivatives during transitions into and out of these image features also is of interest.

The derivatives of a digital function are defined in terms of differences.

There are various ways to define these differences. However, we require that any definition we use for a first derivative (1) must be zero in flat segments (areas of constant gray-level values); (2) must be nonzero at the onset of a gray-level step or ramp; and (3) must be nonzero along ramps.

Similarly, any definition of a second derivative (1) must be zero in flat areas; (2) must be nonzero at the onset and end of a gray-level step or ramp; and (3) must be zero along ramps of constant slope. Since we are dealing with digital quantities whose values are finite, the maximum possible gray-level change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.

A basic definition of the first-order derivative of a one-dimensional function *f(x)* is the difference

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

Similarly, we define a second-order derivative as the difference

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

It is easily verified that these two definitions satisfy the conditions stated previously regarding derivatives of the first and second order.

(I) First -order derivatives generally produce thicker edges in an image and use of first derivatives in image processing. is for edge extraction.
(2) Second-order derivatives have a stronger response to fine detail, such as thin lines and isolated points.
(3) First order derivatives generally have a stronger response to a gray-level step.
(4) Second-order derivatives produce a double response at step changes in gray level.

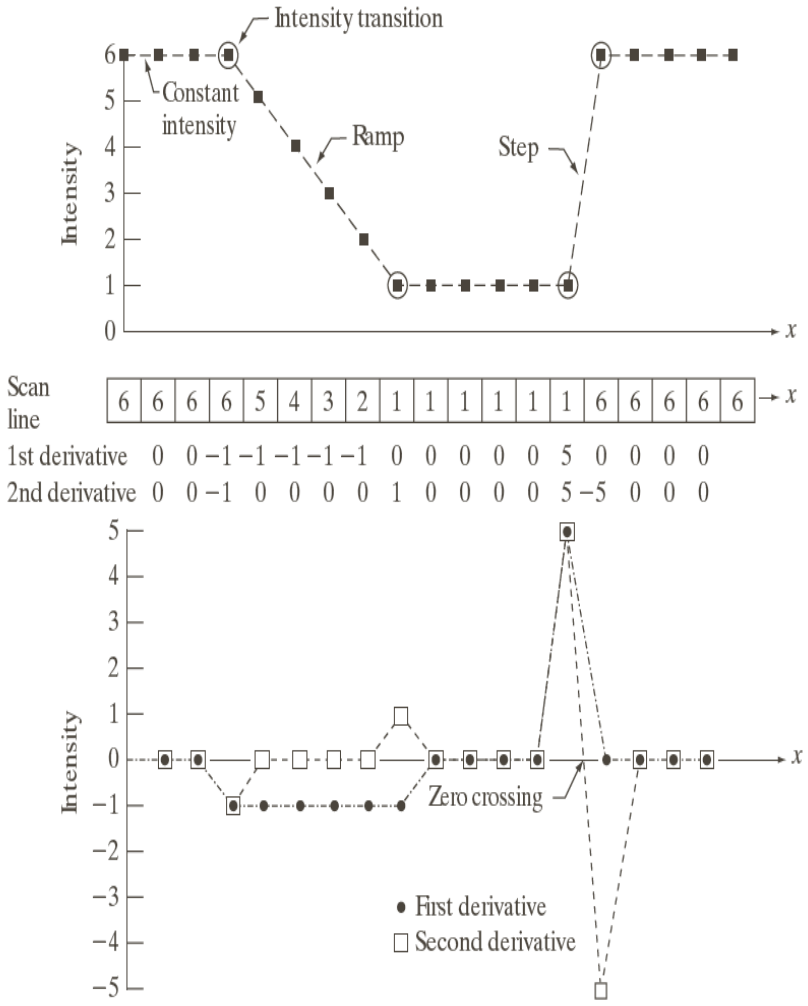


FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Use of Second Derivatives for Enhancement-The Laplacian

The use of two-dimensional, second order derivatives for image enhancement. The approach basically consists of defining a discrete formulation of the second-order derivative and then constructing a filter mask based on that formulation.

We are interested in *isotropic* filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied. In other words, isotropic filters are *rotation invariant,* in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result.

Development of the method

It can be shown (Rosenfeld and Kak [1982]) that the simplest isotropic derivative operator is the *Laplacian,* which, for a function (image) *f(x, y)* of two variables, is defined as

Because derivatives of any order are linear operations, the Laplacian is a linear operator.

In order to be useful for digital image processing, this equation needs to be expressed in discrete form. There are several ways to define a digital Laplacian using neighborhoods.
The definition of the digital second derivative given in that section is one of the most used. Taking into account that we now have two variables, we use the following notation for the partial second-order derivative in the x-direction:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \qquad (3.7\text{-}2)$$

and, similarly in the y-direction, as

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \qquad (3.7\text{-}3)$$

The digital implementation of the two-dimensional Laplacian in Eq. (3.7-1) is obtained by summing these two components

$$\nabla^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\right] - 4f(x, y). \qquad (3.7\text{-}4)$$

This equation can be implemented using the mask shown in Fig. 3.39(a), which gives an isotropic result for rotations in increments of 90_0

Sharpening Spatial Filters: Foundation

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

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

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

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

Sharpening Spatial Filters: Laplace Operator

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

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

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

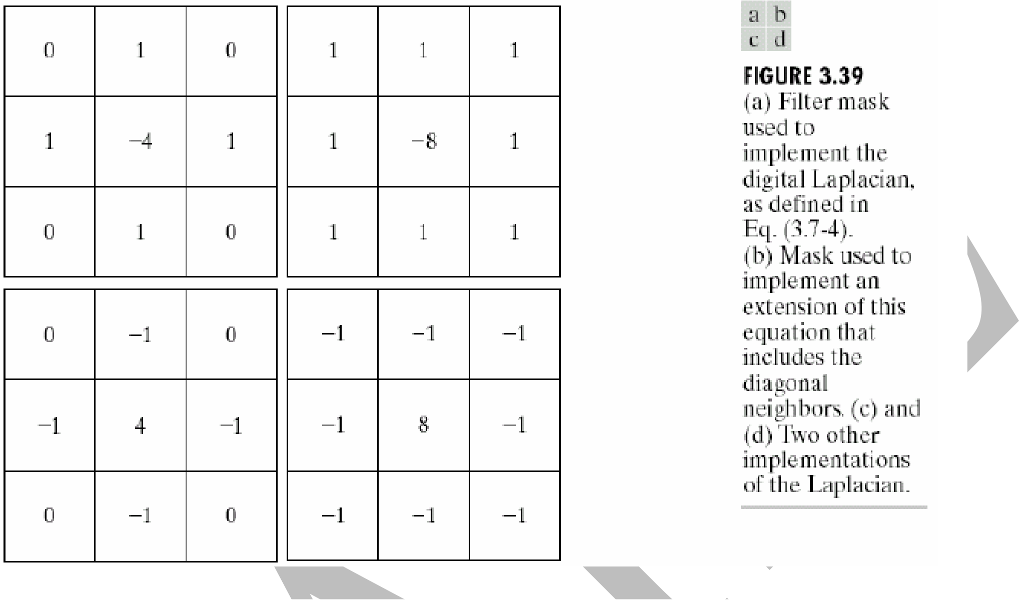
$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

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

Laplacian Mask

Isotropic result for rotations in increments of 90°

Isotropic result for rotations in increments of 45°



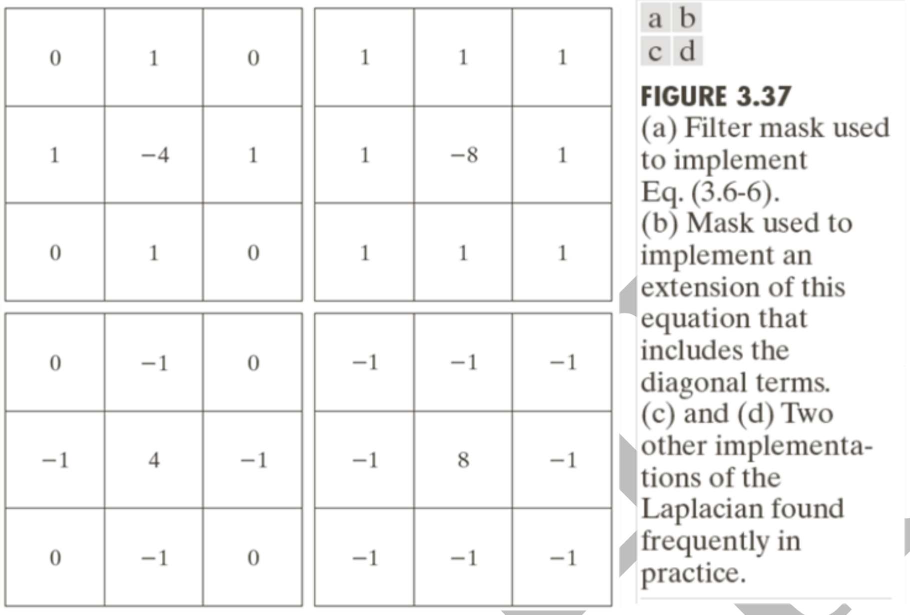
0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

Effect of Laplacian Operator

- Since the Laplacian is a derivative operator
  - it highlights graylevel discontinuities in an image
  - it deemphasizes regions with slowly varying gray levels
- The Laplacian tends to produce images that have
  - grayish edge lines and other discontinuities all superimposed on a dark featureless background

Sharpening Spatial Filters: Laplace Operator



0	1	0
1	−4	1
0	1	0

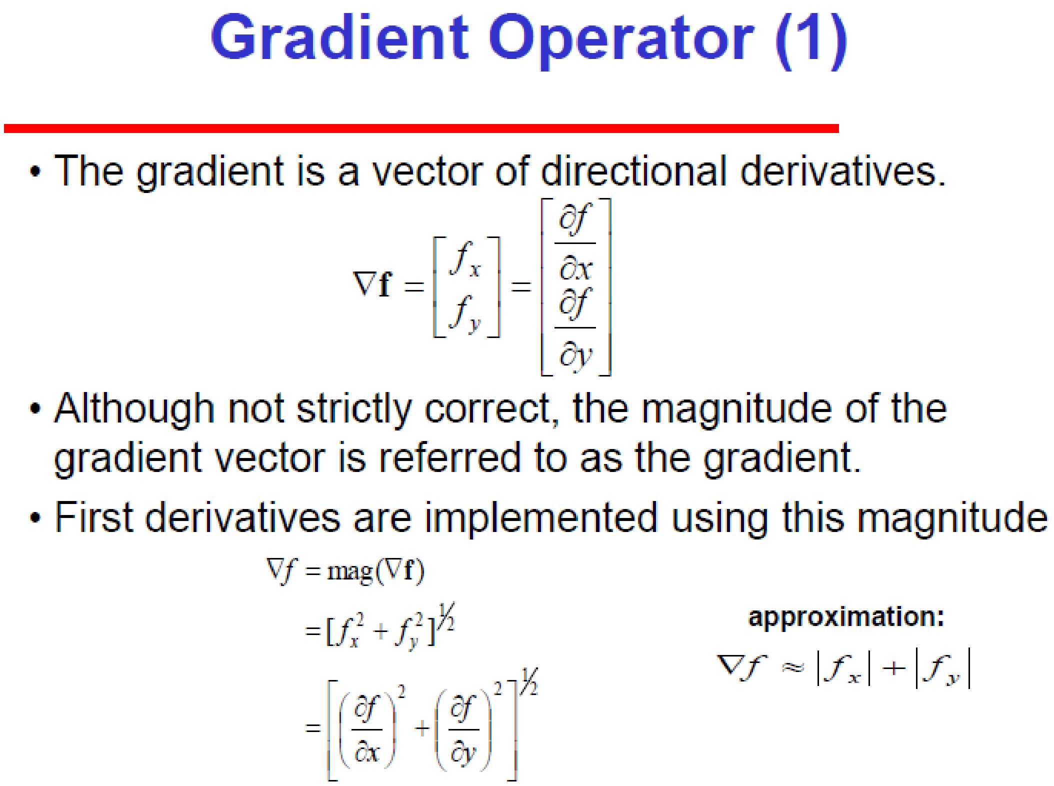
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.



Gradient Operator (1)

- The gradient is a vector of directional derivatives.

$$\nabla \mathbf{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Although not strictly correct, the magnitude of the gradient vector is referred to as the gradient.
- First derivatives are implemented using this magnitude

$$\nabla f = \text{mag}(\nabla \mathbf{f})$$
$$= [f_x^2 + f_y^2]^{1/2}$$
$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

approximation:

$$\nabla f \approx |f_x| + |f_y|$$

3.7.3 Use of First Derivatives for Enhancement—The Gradient

First derivatives in image processing are implemented using the magnitude of the gradient. For a function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as the two-dimensional column *vector*

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{3.7-12}$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned} \tag{3.7-13}$$

The components of the gradient vector itself are linear operators, but the magnitude of this vector obviously is not because of the squaring and square root operations. On the other hand, the partial derivatives in Eq. (3.7-12) are not rotation invariant (isotropic), but the magnitude of the gradient vector is. Although it is not strictly correct, the magnitude of the gradient vector often is referred to as the *gradient*. In keeping with tradition, we will use this term in the following discussions, explicitly referring to the vector or its magnitude only in cases where confusion is likely.

Sharpening Filters: Derivatives (cont'd)

- The gradient is a **vector** which has magnitude and direction:

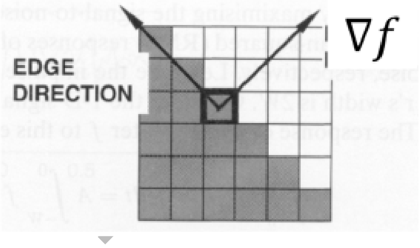
$$magnitude(grad(f)) = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2} \quad \text{or} \quad |\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|$$

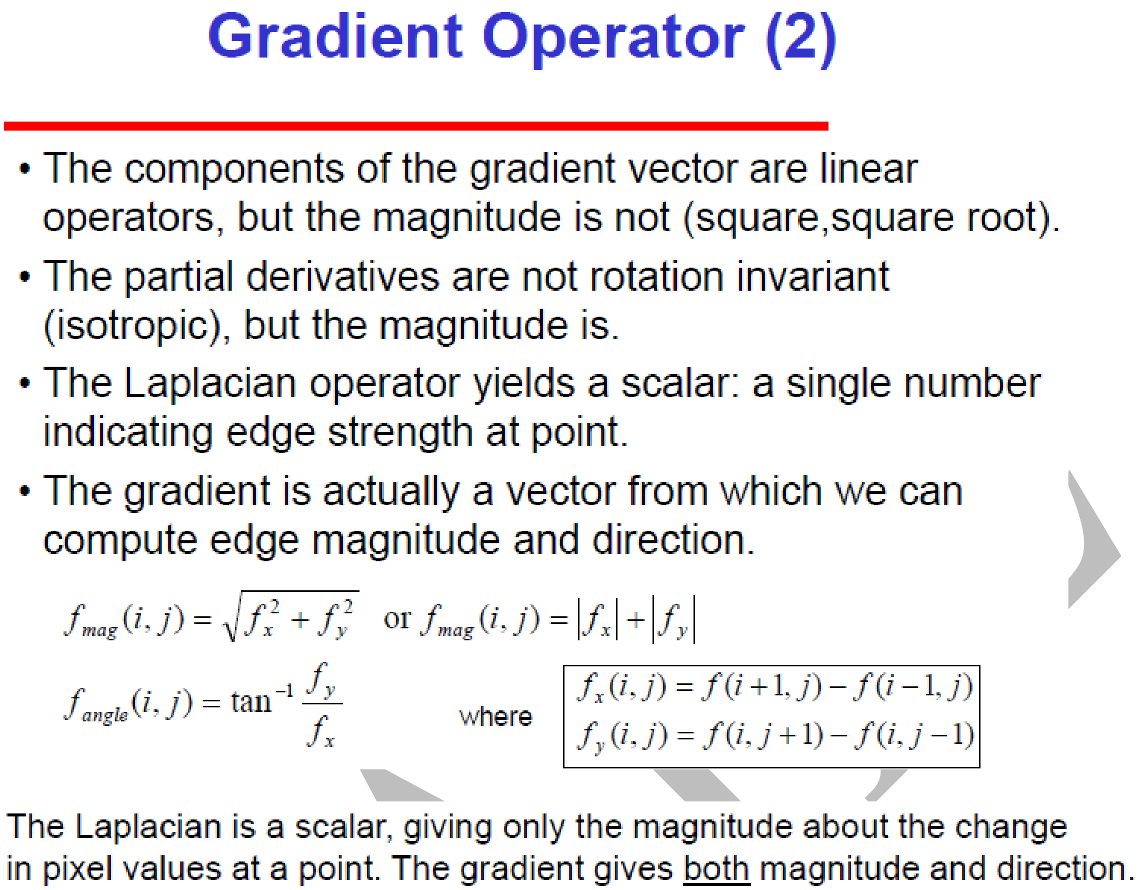
$$direction(grad(f)) = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$$

(approximation)

Sharpening Filters: Derivatives (cont'd)

- **Magnitude:** provides information about edge strength.
- **Direction:** perpendicular to the direction of the edge.





Gradient Operator (2)

- The components of the gradient vector are linear operators, but the magnitude is not (square,square root).
- The partial derivatives are not rotation invariant (isotropic), but the magnitude is.
- The Laplacian operator yields a scalar: a single number indicating edge strength at point.
- The gradient is actually a vector from which we can compute edge magnitude and direction.

$$f_{mag}(i,j) = \sqrt{f_x^2 + f_y^2} \quad \text{or} \quad f_{mag}(i,j) = |f_x| + |f_y|$$

$$f_{angle}(i,j) = \tan^{-1}\frac{f_y}{f_x} \qquad \text{where} \quad \boxed{\begin{array}{l} f_x(i,j) = f(i+1,j) - f(i-1,j) \\ f_y(i,j) = f(i,j+1) - f(i,j-1) \end{array}}$$

The Laplacian is a scalar, giving only the magnitude about the change in pixel values at a point. The gradient gives both magnitude and direction.

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III BSC CS COURSE NAME: DIGITAL IMAGE PROCESSING
COURSE CODE: 16CSU502B BATCH-2016-2019

UNIT 1

SNO	QUESTIONS	CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	ANSWER
1	The spatial coordinates of a digital image (x,y) are proportional to:	Position	Brightness	Contrast	Noise	Brightness
2	Among the following image processing techniques which is fast, precise and flexible.	Optical	Digital	Electronic	Photographic	Digital
3	An image is considered to be a function of a(x,y), where a represents:	Height of image	Width of image	Amplitude of image	Resolution of image	Amplitude of image
4	What is pixel?	Pixel is the elements of a digital image	Pixel is the elements of an analog image	Pixel is the cluster of a digital image	Pixel is the cluster of an analog image	Pixel is the elements of a digital image
5	The range of values spanned by the gray scale is called __________	Dynamic range	Band range	Peak range	Resolution range	Dynamic range
6	Which is a colour attribute that describes a pure colour?	Saturation	Hue	Brightness	Intensity	Hue
7	Which gives a measure of the degree to which a pure colour is diluted by white light?	Saturation	Hue	Intensity	Brightness	Saturation
8	Which means the assigning meaning to a recognized object.	Interpretation	Recognition	Acquisition	Segmentation	Interpretation
9	A typical size comparable in quality to monochromatic TV image is of size.	256 X 256	512 X 512	1920 X 1080	1080 X 1080	512 X 512
10	To convert a continuous sensed data into Digital form, which of the following is required?	Sampling	Quantization	Both Sampling and Quantization	Neither Sampling nor Quantization	Both Sampling and Quantization
11	To convert a continuous image f(x, y) to digital form, we have to sample the function in __________	Coordinates	Amplitude`	All of the mentioned	None of the mentioned	All of the mentioned

12	For a continuous image f(x, y), how could be Sampling defined?	Digitizing the coordinate values	Digitizing the amplitude values	All of the mentioned	None of the mentioned	Digitizing the coordinate values
13	For a continuous image f(x, y), Quantization is defined as	Digitizing the coordinate values	Digitizing the amplitude values	All of the mentioned	None of the mentioned	Digitizing the amplitude values
14	The quality of a digital image is well determined by ___________	The number of samples	The discrete gray levels	All of the mentioned	None of the mentioned	All of the mentioned
15	Assume that an image f(x, y) is sampled so that the result has M rows and N columns. If the values of the coordinates at the origin are (x, y) = (0, 0), then the notation (0, 1) is used to signify :	Second sample along first row	First sample along second row	First sample along first row	Second sample along second row	Second sample along first row
16	The resulting image of sampling and quantization is considered a matrix of real numbers. By what name(s) the element of this matrix array is called __________	Image element or Picture element	Pixel or Pel	All of the mentioned	None of the mentioned	All of the mentioned
17	A continuous image is digitised at _______ points.	random	vertex	contour	sampling	sampling
18	The transition between continuous values of the image function and its digital equivalent is called ______________	Quantisation	Sampling	Rasterisation	None of the Mentioned	Quantisation
19	Images quantised with insufficient brightness levels will lead to the occurrence of ____________	Pixillation	Blurring	False Contours	None of the Mentioned	False Contours
20	The smallest discernible change in intensity level is called ____________	Intensity Resolution	Contour	Saturation	Contrast	Intensity Resolution
21	What is the tool used in tasks such as zooming, shrinking, rotating, etc.?	Sampling	Interpolation	Filters	None of the Mentioned	Interpolation
22	The type of Interpolation where for each new location the intensity of the immediate pixel is assigned is ___________	bicubic interpolation	cubic interpolation	bilinear interpolation	nearest neighbour interpolation	nearest neighbour interpolation

23	The type of Interpolation where the intensity of the FOUR neighbouring pixels is used to obtain intensity a new location is called ___________	cubic interpolation	nearest neighbour interpolation	bilinear interpolation	bicubic interpolation	nearest neighbour interpolation
24	Dynamic range of imaging system is a ratio where the upper limit is determined by	Saturation	Noise	Brightness	Contrast	Saturation
25	For Dynamic range ratio the lower limit is determined by	Saturation	Brightness	Noise	Contrast	Noise
26	Quantitatively, spatial resolution cannot be represented in which of the following ways	line pairs	pixels	dots	none of the Mentioned	none of the Mentioned
27	Of the following, _________ has the maximum frequency.	UV Rays	Gamma Rays	Microwaves	Radio Waves	Gamma Rays
28	In the Visible spectrum the ______ colour has the maximum wavelength.	Violet	Blue	Red	Yellow	Red
29	Wavelength and frequency are related as : (c = speed of light)	c = wavelength / frequency	frequency = wavelength / c	wavelength = c * frequency	c = wavelength * frequency	c = wavelength * frequency
30	Electromagnetic waves can be visualised as a	sine wave	cosine wave	tangential wave	None of the mentioned	sine wave
31	How is radiance measured?	lumens	watts	armstrong	hertz	watts
32	Which of the following is used for chest and dental scans?	Hard X-Rays	Soft X-Rays	Radio waves	Infrared Rays	Soft X-Rays
33	Which of the following is impractical to measure?	Frequency	Radiance	Luminance	Brightness	Brightness
34	Massless particle containing a certain amount of energy is called _________	Photon	Shell	Electron	None of the mentioned	Photon
35	What do you mean by achromatic light?	Chromatic light	Monochromatic light	Infrared light	Invisible light	Monochromatic light
36	Which of the following embodies the achromatic notion of intensity?	Luminance	Brightness	Frequency	Radiance	Brightness
37	Image processing approaches operating directly on pixels of input image work directly in ____________	Transform domain	Spatial domain	Inverse transformation	None of the Mentioned	Spatial domain

38	What is the output of a smoothing, linear spatial filter?	Median of pixels	Maximum of pixels	Minimum of pixels	Average of pixels	Average of pixels
39	Which of the following in an image can be removed by using smoothing filter?	Smooth transitions of gray levels	Smooth transitions of brightness levels	Sharp transitions of gray levels	Sharp transitions of brightness levels	Sharp transitions of gray levels
40	Which of the following is the disadvantage of using smoothing filter?	Blur edges	Blur inner pixels	Remove sharp transitions	Sharp edges	Blur edges
41	Which of the following comes under the application of image blurring?	Object detection	Gross representation	Object motion	Image segmentation	Gross representation
42	Which of the following filters response is based on ranking of pixels?	Nonlinear smoothing filters	Linear smoothing filters	Sharpening filters	Geometric mean filter	Nonlinear smoothing filters
43	Median filter belongs to which category of filters?	Linear spatial filter	Frequency domain filter	Order static filter	Sharpening filter	Order static filter
44	Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?	s=L+1-r	s=L+1+r	s=L-1-r	s=L-1+r	s=L-1-r
45	What is the name of process used to correct the power-law response phenomena?	Beta correction	Alpha correction	Gamma correction	Pie correction	Gamma correction
46	Which of the following transformation function requires much information to be specified at the time of input?	Log transformation	Power transformation	Piece-wise transformation	Linear transformation	Piece-wise transformation
47	In which type of slicing, highlighting a specific range of gray levels in an image often is desired?	Gray-level slicing	Bit-plane slicing	Contrast stretching	Byte-level slicing	Gray-level slicing
48	The process of using known data to estimate values at unknown locations is called	Acquisition	Interpolation	Pixelation	None of the Mentioned	Interpolation
49	Which of the following is NOT an application of Image Multiplication?	Shading Correction	Masking	Pixelation	Region of Interest operations	Pixelation

50	The procedure done on a digital image to alter the values of its individual pixels is _________	Neighbourhood Operations	Image Registration	Geometric Spacial Transformation	Single Pixel Operation	Single Pixel Operation
51	______________ is the effect caused by the use of an insufficient number of intensity levels in smooth areas of a digital image.	Gaussian smooth	Contouring	False Contouring	Interpolation	False Contouring
52	The difference is intensity between the highest and the lowest intensity levels in an image is ____________	Noise	Saturation	Contrast	Brightness	Contrast
53	The most familiar single sensor used for Image Acquisition is __________	Microdensitometer	Photodiode	CMOS	None of the Mentioned	Photodiode
54	What is the first and foremost step in Image Processing?	Image restoration	Image enhancement	Image acquisition	Segmentation	Image acquisition
55	What is the basis for numerous spatial domain processing techniques?	Transformations	Scaling	Histogram	None of the Mentioned	Histogram
56	In _______ image we notice that the components of histogram are concentrated on the low side on intensity scale.	bright	dark	colourful	All of the Mentioned	dark
57	Histogram Equalisation is mainly used for _________________	Image enhancement	Blurring	Contrast adjustment	None of the Mentioned	Image enhancement
58	What is brightness adaptation	Changing the eye's overall sensitivity	Changing the eye's imaging ability	Adjusting the focal length	Transition from scotopic to photopic vision	Changing the eye's overall sensitivity
59	Using gray-level transformation, the basic function linearity deals with which of the following transformation?	log and inverse-log transformations	negative and identity transformations	nth and nth root transformations	All of the mentioned	negative and identity transformations
60	Nearest neighbor Interpolation has an undesirable feature, that is _________	Aliasing effect	False contouring effect	Ridging effect	Checkerboard effect	Checkerboard effect

UNIT-II

SYLLABUS

Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.

HOTELLING TRANSFORM

The Hotelling transform is a linear transformation of a set of n dimensional vectors that decorrelates the n coordinates. When applied to an 2-dimensional image, the transformed image will be aligned along its principal axis.

The Hotelling transform can be used to improve the robustness to distortion of the image due to rotation and displacement. The improvement though, would increase the running time of the image query.

This transformation is a combination of displacement and rotation of the object. The centroid of the image(the average of the x and y coordinates of all pixels) is shifted to the origin and the image is rotated by an angle that minimizes its moment of inertia[4]. Geometrically the transformed image will be oriented in the direction in which it seems to be the most 'elongated'.

In image recognition this transformation is helpful, since the identity of the object is not known and aligning the image with its principal axis can help remove the effects of translation and rotation in the analysis.

First we describe the Hotelling transform. Given a set k of n-dimensional column vectors: $X_1, X_2, ..., X_k$,the covariance matrix of the vector population is given by:

$$C_x = E[(X - m_x)(X - m_x)^T]$$

where m_x is the mean vector of the population.

C_x is a nxn real symmetric matrix whose c_{ij} entry equals the covariance between the i^{th} and the j^{th} coordinates of the vector population. If $c_{ij} = 0$, then the i^{th} and the j^{th} coordinates are decorrelated. When $c_{ij} > 0$ or $c_{ij} < 0$ then the there is a positive or a negative correlation between the i^{th} and the j^{th} coordinates, respectively.

The Hotelling transform maps the given vector population, X, into a vector population Y (that consists of k, n-dimensional vectors) such that $m_y = 0$ and C_y, the covariance of the new vector population is a nxn diagonal matrix and therefore the i^{th} and the j^{th} coordinates of the new vector population are decorrelated for all $i \neq j$. To see the relevance of the Hotelling transform to aligning 2-dimensional image with its principal axis, consider an image composed of mxn pixels . If the (x, y) coordinates of each pixel is presented as a 2-dimensional vector, the image then is represented by a set of mxn, 2-dimensional, vectors. When the Hotelling transform is applied to this vector population, the mean vector of the new vector population, $m_y = 0$, and its covariance will be diagonal, which means that the x and y coordinates are decorrelated.

Geometrically this would mean that the centroid of the original image was shifted to the origin and the image was rotated so that its principle axis is aligned with the X-axis.

The basic principle of hotelling transform is the statistical properties of vector representation. Consider a population of random vectors of the form,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

And the mean vector of the population is defined as the expected value of x i.e.,

$$m_x = E\{x\}$$

The suffix m represents that the mean is associated with the population of x vectors. The expected value of a vector or matrix is obtained by taking the expected value of each element. The covariance matrix C_x in terms of x and m_x is given as

$$C_x = E\{(x-m_x)\ (x-m_x)^T\}$$

T denotes the transpose operation. Since, x is n dimensional, $\{(x-m_x)\ (x-m_x)^T\}$ will be of n x n dimension. The covariance matrix is real and symmetric. If elements xi and xj are uncorrelated, their covariance is zero and, therefore, $c_{ij} = c_{ji} = 0$.

For M vector samples from a random population, the mean vector and covariance matrix can be approximated from the samples by

$$\mathbf{m}_x = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k$$

and

$$\mathbf{C}_x = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T.$$

Fourier Transforms and properties:

FREQUENCY DOMAIN ANALYSIS:

Till now, all the domains in which we have analyzed a signal, we analyze it with respect to time. But in frequency domain we don't analyze signal with respect to time, but with respect of frequency.

Difference between spatial domain and frequency domain

In spatial domain, we deal with images as it is. The value of the pixels of the image change with respect to scene. **Whereas in frequency domain, we deal with the rate at which the pixel values are changing in spatial domain.**

For simplicity, Let's put it this way.

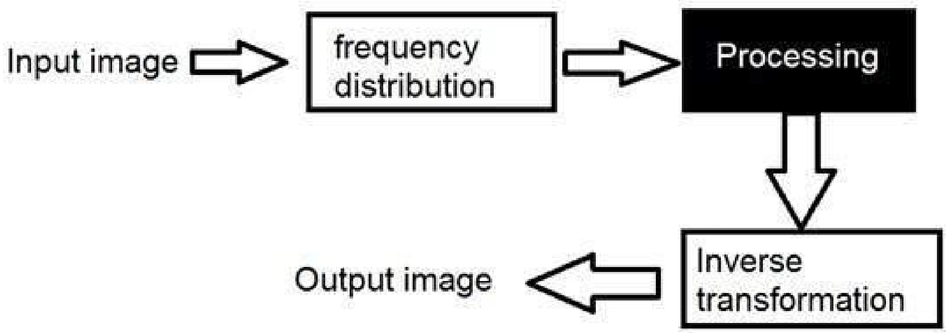
Spatial domain

In simple spatial domain, we directly deal with the image matrix. Whereas in frequency domain, we deal an image like this.

Frequency Domain

We first transform the image to its frequency distribution. Then our black box system perform whatever processing it has to performed, and the output of the black box in this case is not an image, but a transformation. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain.

It can be pictorially viewed as



Transformation

A signal can be converted from time domain into frequency domain using mathematical operators called transforms. There are many kind of transformation that does this. Some of them are given below.

- Fourier Series
- Fourier transformation
- Laplace transform
- Z transform

Frequency components

Any image in spatial domain can be represented in a frequency domain. But what do this frequencies actually mean.

We will divide frequency components into two major components.

High frequency components

High frequency components correspond to edges in an image.

Low frequency components

Low frequency components in an image correspond to smooth regions.

Fourier

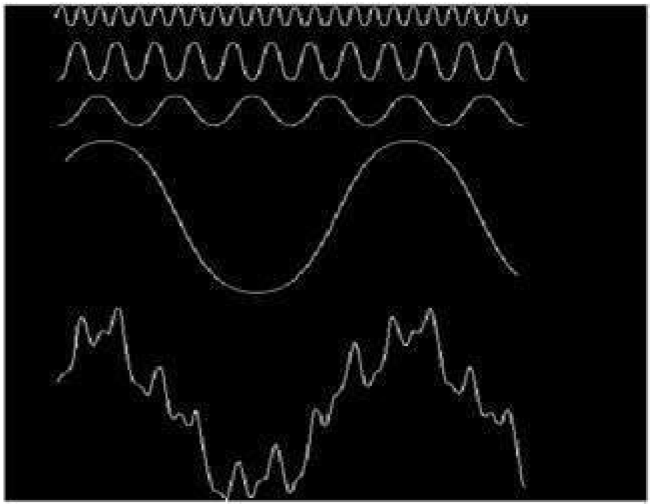
Fourier was a mathematician in 1822. He give Fourier series and Fourier transform to convert a signal into frequency domain.

Fourier Series

Fourier series simply states that, periodic signals can be represented into sum of sines and cosines when multiplied with a certain weight. It further states that periodic signals can be broken down into further signals with the following properties.

- The signals are sines and cosines
- The signals are harmonics of each other

It can be pictorially viewed as



In the above signal, the last signal is actually the sum of all the above signals. This was the idea of the Fourier.

How it is calculated

Since as we have seen in the frequency domain, that in order to process an image in frequency domain, we need to first convert it using into frequency domain and we have to take inverse of the output to convert it back into spatial domain.

That's why both Fourier series and Fourier transform has two formulas. One for conversion and one converting it back to the spatial domain.

Fourier series

The Fourier series can be denoted by this formula.

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

The inverse can be calculated by this formula.

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

FOURIER TRANSFORM:

The Fourier transform simply states that that the non periodic signals whose area under the curve is finite can also be represented into integrals of the sines and cosines after being multiplied by a certain weight.

The Fourier transform has many wide applications that include, image compression (e.g JPEG compression), filtering and image analysis.

Difference between Fourier series and transform

Although both Fourier series and Fourier transform are given by Fourier, but the difference between them is Fourier series is applied on periodic signals and Fourier transform is applied for non periodic signals

Which one is applied on images

Now the question is that which one is applied on the images, the Fourier series or the Fourier transform. Well, the answer to this question lies in the fact that what images are.

Images are non – periodic. And since the images are non periodic, so Fourier transform is used to convert them into frequency domain.

Discrete fourier transform

Since we are dealing with images, and in fact digital images, so for digital images we will be working on discrete fourier transform

Properties of Fourier Transform:

- **Spatial Frequency**
- **Magnitude**
- **Phase**
- The spatial frequency directly relates with the brightness of the image.
- The magnitude of the sinusoid directly relates with the contrast. Contrast is the difference between maximum and minimum pixel intensity.
- Phase contains the color information.
- A 2D signal can also be complex and thus written in terms of its magnitude and phase.
- If a 2D signal is real and even, then the Fourier transform is real and even.

- The Fourier and the inverse Fourier transforms are linear operations.

The formula for 2 dimensional discrete Fourier transform is given below.

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

The discrete Fourier transform is actually the sampled Fourier transform, so it contains some samples that denotes an image. In the above formula f(x,y) denotes the image, and F(u,v) denotes the discrete Fourier transform. The formula for 2 dimensional inverse discrete Fourier transform is given below.

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

The inverse discrete Fourier transform converts the Fourier transform back to the image.

Fourier proved that any periodic function can be expressed as the sum of sinusoids of different frequencies, each multiplied by a different coefficient. →*Fourier series*

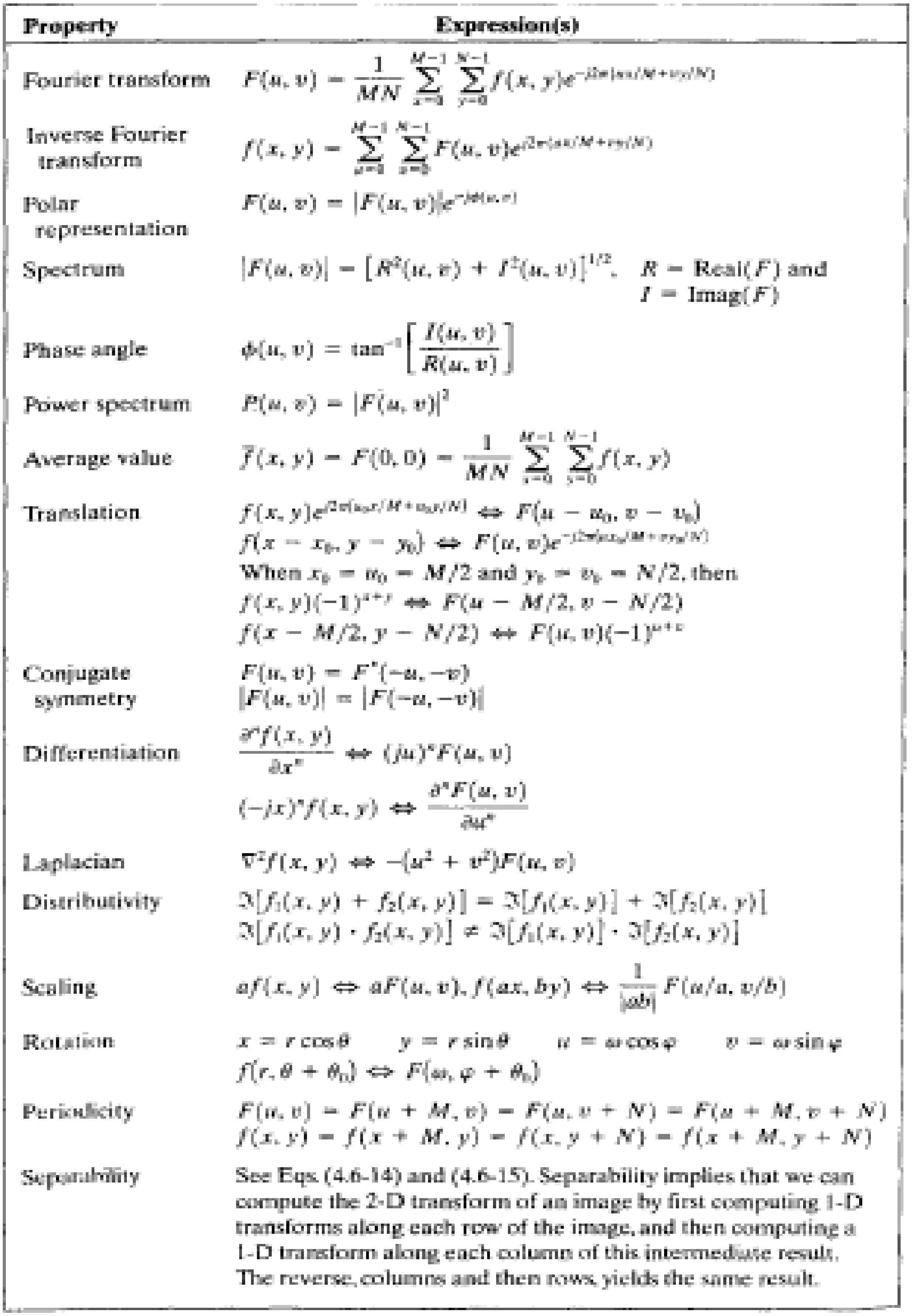
•Even aperiodic functions (whose area under the curve is finite) can be expressed as the integral of sinusoids multiplied by a weighting function. →*Fourier transform*

The Fourier transform is more useful than the Fourier series in most practical problems since it handles signals of finite duration.

•The Fourier transform takes us between the spatial and frequency domains.

•It permits for a dual representation of a signal that is amenable for filtering and analysis.

•Revolutionized the field of signal processing.



Property	**Expression(s)**
Fourier transform	$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u,v) = \|F(u,v)\|e^{-j\phi(u,v)}$
Spectrum	$\|F(u,v)\| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
Power spectrum	$P(u,v) = \|F(u,v)\|^2$
Average value	$\bar{f}(x,y) = F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
Translation	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$
Conjugate symmetry	$F(u,v) = F^*(-u,-v)$ $\|F(u,v)\| = \|F(-u,-v)\|$
Differentiation	$\frac{\partial^n f(x,y)}{\partial x^n} \Leftrightarrow (ju)^n F(u,v)$ $(-jx)^n f(x,y) \Leftrightarrow \frac{\partial^n F(u,v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x,y) \Leftrightarrow -(u^2+v^2)F(u,v)$
Distributivity	$\Im[f_1(x,y) + f_2(x,y)] = \Im[f_1(x,y)] + \Im[f_2(x,y)]$ $\Im[f_1(x,y)\cdot f_2(x,y)] \neq \Im[f_1(x,y)]\cdot\Im[f_2(x,y)]$
Scaling	$af(x,y) \Leftrightarrow aF(u,v), f(ax,by) \Leftrightarrow \frac{1}{\|ab\|}F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$
Periodicity	$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N)$ $f(x,y) = f(x+M,y) = f(x,y+N) = f(x+M,y+N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

FFT (FAST FOURIER TRANSFORM):

- Fast fourier transform proposed by Cooley and Tukey in 1965.
- The fast fourier transform is a highly efficient procedure for computing the DFT of a finite series and requires less number of computations than that of direct evaluation of DFT.
- The FFT is based on decomposition and breaking the transform into smaller transforms and combining them to get the total transform.

Computing the l-D Fourier transform of M points using directly requires on the order of $M2$ multiplication/addition operations. The FFT accomplishes the same task on the order of M log2 M operations. If, for example, $M = 1024$. the brute-force method will require approximately 10^6 operations. while the FFT will require approximately 10^4 operations.

A fourier transform is an useful analytical tool that is important for many fields of application in the digital signal processing.

- In describing the properties of the fourier transform and inverse fourier transform, it is quite convenient to use the concept of time and frequency.
- In image processing applications it plays a critical role.

Discrete Fourier Transform

Discrete Fourier Transform (**DFT**) is the discrete version of the Fourier Transform (FT) that transforms a signal (or discrete sequence) from the time domain representation to its representation in the frequency domain. Whereas, **Fast Fourier Transform** (**FFT**) is any efficient algorithm for calculating the **DFT**.

What is FFT?

- **The fast fourier is an algorithm used to compute the DFT. It makes use of the symmetry and periodicity properties to effectively reduce the DFT computation time.**
- It is based on the fundamental principle of decomposing the computation of DFT of a sequence of length N into successively smaller DFT.

FFT algorithm provides speed increase factors, when compared with direct computation of the DFT.

- The number of multiplications and additions required to compute N-point DFT using radix-2 FFT are Nlog2N and N/2log2N respectively.

FFT Algorithms

- There are basically two types of FFT algorithms.
- They are:

1. Decimation in Time

2. Decimation in frequency

Decimation in time

- DIT algorithm is used to calculate the DFT of a N-point sequence.
- The idea is to break the N-point sequence into two sequences, the DFTs of which can be obtained to give the DFT of the original N-point sequence.
- Initially the N-point sequence is divided into N/2-point sequences Xe(n) and Xo(n) , which have even and odd numbers of x(n) respectively.

The N/2-point DFTs of these two sequences are evaluated and combined to give the N-point DFT.

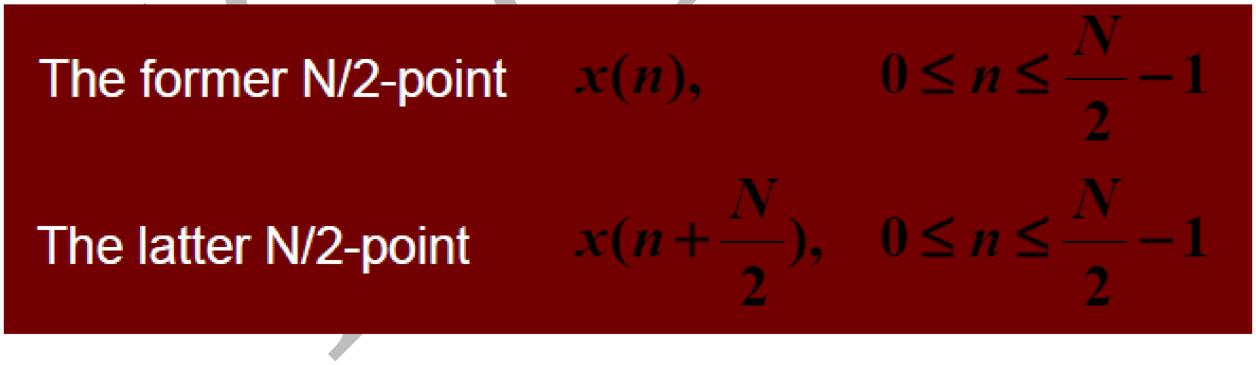
- Similarly the N/2-point DFTs can be expressed as a combination of N/4-point DFTs.
- This process is continued until we are left with two point DFT.
- This algorithm is called decimation-in-time because the sequence X(n) is often split into smaller sequences.

Decimation-In-Frequency

- It is a popular form of FFT algorithm.
- In this the output sequence x(k) is divided into smaller and smaller subsequences, that is why the name decimation in frequency,
- Initially the input sequence x(n) is divided into two sequences x1(n) and x2(n) consisting of the first n/2 samples of x(n) and the last n/2 samples of x(n) respectively

Algorithm principle

- To divide N-point sequence x(n) into two N/2-point Sequence



The former N/2-point $x(n), \quad 0 \le n \le \frac{N}{2} - 1$

The latter N/2-point $x(n + \frac{N}{2}), \quad 0 \le n \le \frac{N}{2} - 1$

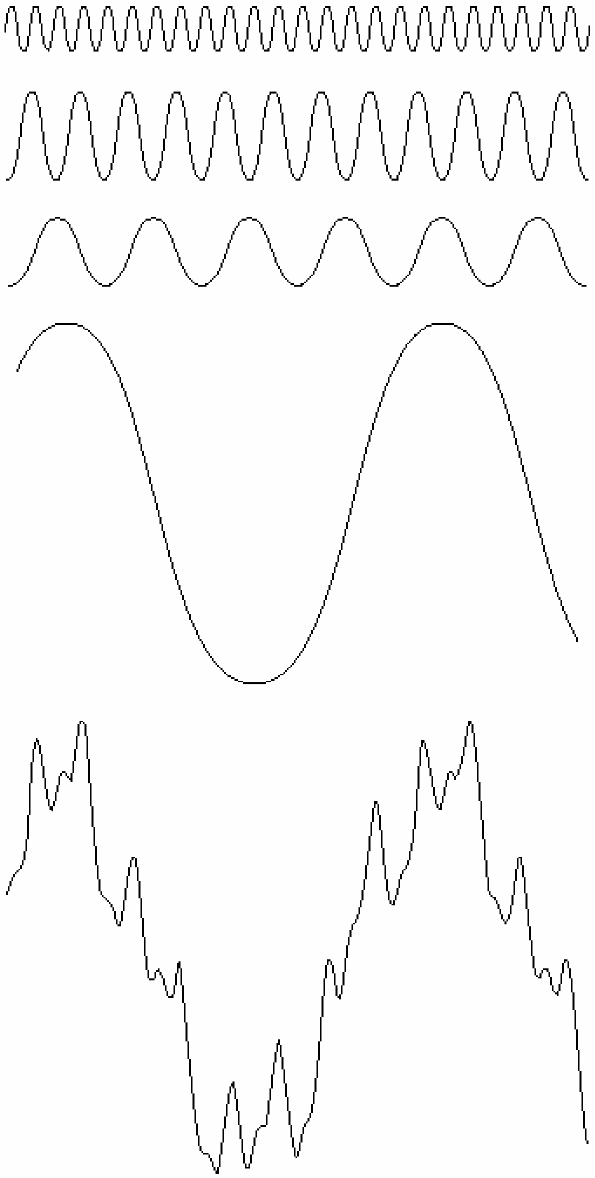
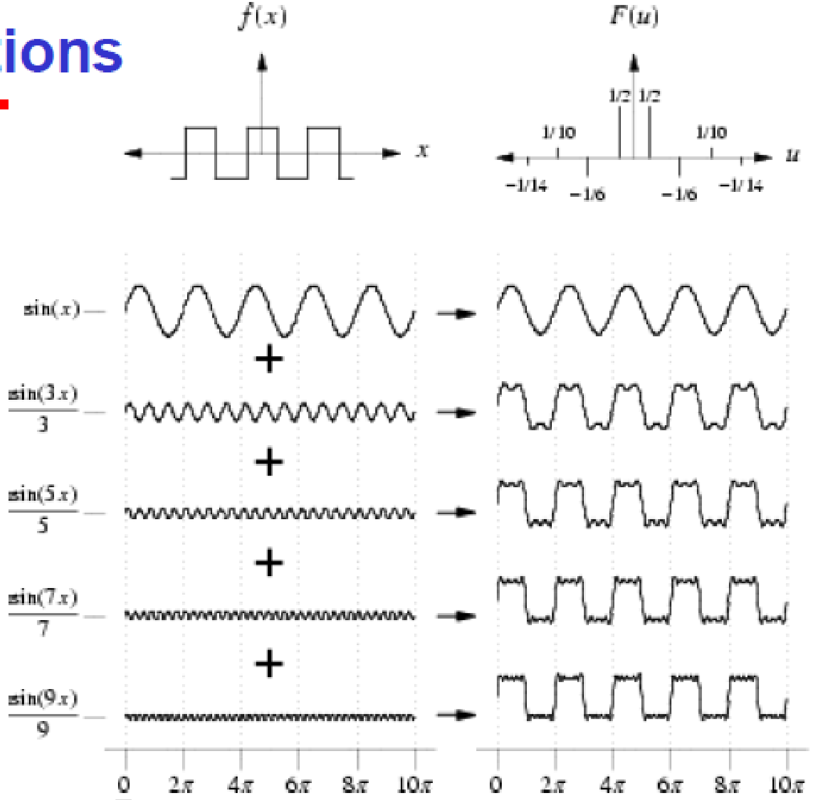


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

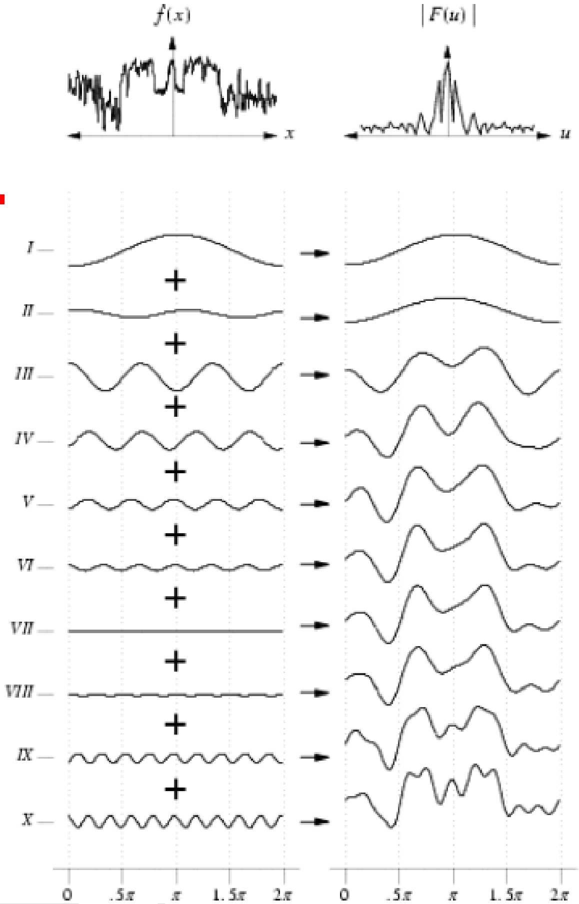
A glass prism is a physical device that separates light into various color components, each depending on its wavelength (or frequency) content.

•The Fourier transform is a mathematical prism that separates a function into its frequency components.

Fourier Series for Periodic Functions



Fourier Transform for Aperiodic Functions



Fourier analysis: determine amplitude & phase shifts

- Fourier synthesis: add scaled and shifted sinusoids together
- Fourier transform pair:

$$\text{Forward F.T.} \quad F(u) = \int_{-\infty} f(x) e^{-i2\pi ux} dx$$

$$\text{Inverse F.T.} \quad f(x) = \int_{-\infty}^{\infty} F(u) e^{+i2\pi ux} dx$$

where $i = \sqrt{-1}$, and

$$\underset{\substack{\uparrow \\ \text{Euler's formula}}}{e^{\pm i2\pi ux}} = \cos(2\pi ux) \pm i \sin(2\pi ux) \leftarrow \text{complex exponential at freq. } u$$

Fourier Coefficients

- Fourier coefficients *F(u)* specify, for each frequency *u*, the amplitude and phase of each complex exponential.
- *F(u)* is the frequency spectrum.
- *f(x)* and *F(u)* are two equivalent representations of the same signal.

$$F(u) = R(u) + iI(u)$$

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \leftarrow \text{ magnitude spectrum; aka Fourier spectrum}$$

$$\Phi(u) = \tan^{-1} \frac{I(u)}{R(u)} \leftarrow \text{phase spectrum}$$

$$\begin{aligned} P(u) &= | F(u) |^2 \\ &= R^2(u) + I^2(u) \leftarrow \text{ spectral density} \end{aligned}$$

Fourier Series (1)

For periodic signals, we have the Fourier series:

$$f(x)=\sum_{n=-\infty}^{n=\infty}c(nu_0)e^{i2\pi nu_0x} \text{ where } c(nu_0) \text{ is the } n^{th} \text{ Fourier coefficient}$$

$$c(nu_0)=\frac{1}{x_0}\int_{-x_0/2}^{x_0/2}f(x)e^{-i2\pi nu_0x}dx$$

That is, the periodic signal contains all the frequencies that are harmonics of the fundamental frequency.

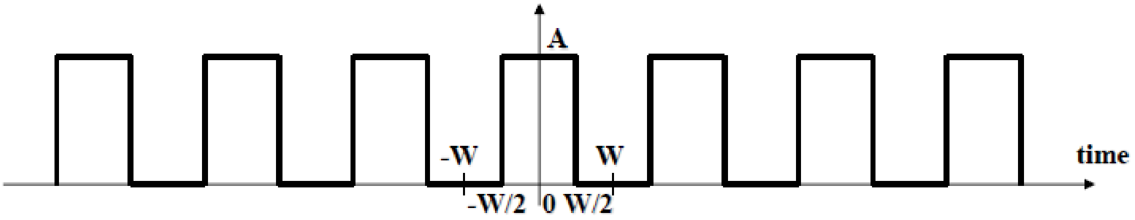
Fourier Series (3)

- The Fourier transform is applied for aperiodic signals.
- It is represented as an integral over a continuum of frequencies.
- The Fourier Series is applied for periodic signals.
- It is represented as a summation of frequency components that are integer multiples of some fundamental frequency.

Example

Ex : Rectangular Pulse Train

$$f(x) = \begin{cases} A & |x| < \frac{W}{2} \\ 0 & |x| > \frac{W}{2} \end{cases} \quad \text{in interval [-W/2, W/2]}$$



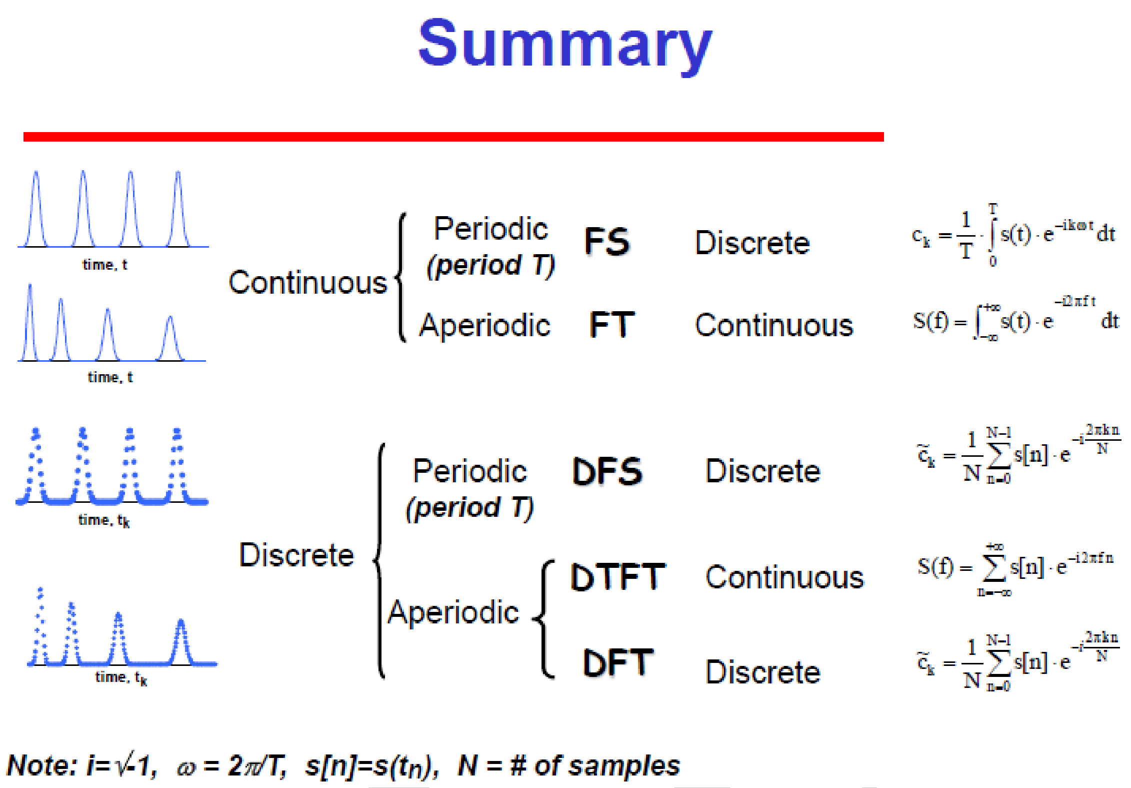
Discrete Fourier Transform

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-i2\pi\frac{ux}{N}} \qquad \text{forward DFT}$$

$$f(x) = \sum_{u=0}^{N-1} F(u)e^{+i2\pi\frac{ux}{N}} \qquad \text{inverse DFT}$$

for $0 \le u \le N-1$ and $0 \le x \le N-1$ where *N* is the number of equi-spaced input samples.

The 1/N factor can be in front of *f(x)* instead.



Summary

Continuous	Periodic *(period T)*		FS	Discrete	$c_k = \frac{1}{T}\cdot\int_0^T s(t)\cdot e^{-ik\omega t}dt$
	Aperiodic		FT	Continuous	$S(f) = \int_{-\infty}^{+\infty} s(t)\cdot e^{-i2\pi f t}dt$
Discrete	Periodic *(period T)*		DFS	Discrete	$\tilde{c}_k = \frac{1}{N}\sum_{n=0}^{N-1} s[n]\cdot e^{-i\frac{2\pi kn}{N}}$
	Aperiodic		DTFT	Continuous	$S(f) = \sum_{n=-\infty}^{+\infty} s[n]\cdot e^{-i2\pi fn}$
			DFT	Discrete	$\tilde{c}_k = \frac{1}{N}\sum_{n=0}^{N-1} s[n]\cdot e^{-i\frac{2\pi kn}{N}}$

***Note:** $i=\sqrt{-1}$, $\omega = 2\pi/T$, $s[n]=s(t_n)$, N = # of samples*

CONVOLUTION THEOREM

Convolution

- Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels.

The matrix of weights is called the convolution kernel, also known as the filter. – A convolution kernel is a correlation kernel that has been rotated 180 degrees.

1. Rotate the convolution kernel 180 degrees about its center element.
2. Slide the center element of the convolution kernel so that it lies on top of the (I,k) element of f.
3. Multiply each weight in the rotated convolution kernel by the pixel of f underneath. Sum the individual products from step 3

Correlation

- The operation called correlation is closely related to convolution.

In correlation, the value of an output pixel is also computed as a weighted sum of neighboring pixels.

• The difference is that the matrix of weights, in this case called the correlation kernel, is not rotated during the computation.

• 1. Slide the center element of the correlation kernel so that lies on top of the (2,4) element of f.

2. Multiply each weight in the correlation kernel by the pixel of A underneath.

3. Sum the individual products from step 2.

The relationship between the spatial domain and the frequency domain can be established by convolution theorem.

The convolution theorem can be represented as.

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

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

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

It can be stated as the convolution in spatial domain is equal to filtering in frequency domain and vice versa.

Correlation

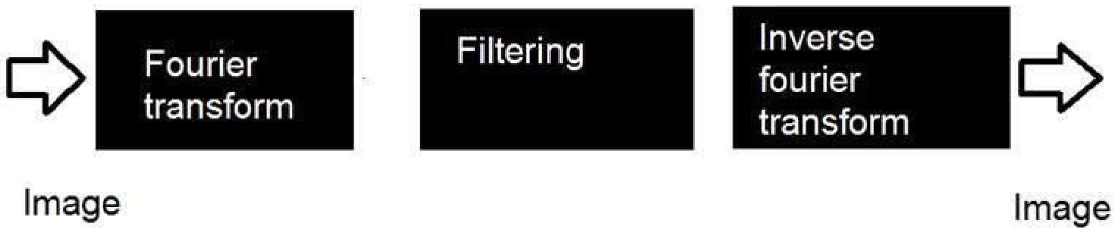
Given the similarity of convolution and correlation, it is not surprising that there is a *correlation theorem,* analogous to the convolution theorem. Let $F(u, v)$ and $H(u, v)$ denote the Fourier transforms of $f(x, y)$ and $h(x, y)$, respectively.

One-half of the correlation theorem states that spatial correlation. $f(x, y)$ o $h(x, y)$, and the frequency domain product, $F^*(u, v)H(u, v)$, constitute a Fourier transform pair. This result, formally stated as

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

indicates that correlation in the spatial domain can be obtained by taking the inverse Fourier transform of the product $F^*(u, v)H(u, v)$, where F^* is the complex conjugate of F.

The filtering in frequency domain can be represented as following:



The steps in filtering are given below.

- At first step we have to do some pre – processing an image in spatial domain, means increase its contrast or brightness
- Then we will take discrete Fourier transform of the image
- Then we will center the discrete Fourier transform, as we will bring the discrete Fourier transform in center from corners
- Then we will apply filtering, means we will multiply the Fourier transform by a filter function
- Then we will again shift the DFT from center to the corners
- Last step would be take to inverse discrete Fourier transform, to bring the result back from frequency domain to spatial domain
- And this step of post processing is optional, just like pre processing , in which we just increase the appearance of image.

Filters

The concept of filter in frequency domain is same as the concept of a mask in convolution.

After converting an image to frequency domain, some filters are applied in filtering process to perform different kind of processing on an image. The processing include blurring an image, sharpening an image e.t.c.

The common type of filters for these purposes are:

- Ideal high pass filter

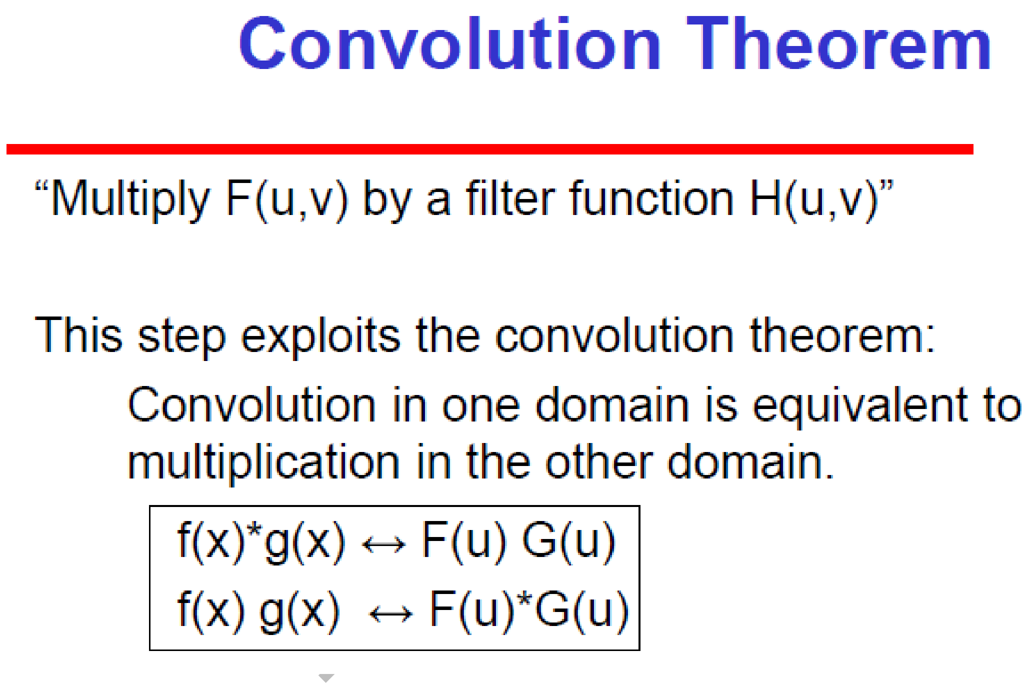
- Ideal low pass filter
- Gaussian high pass filter
- Gaussian low pass filter

The Fourier transform is more useful than the Fourier series in most practical problems since it handles signals of finite duration.

•The Fourier transform takes us between the spatial and frequency domains.

•It permits for a dual representation of a signal that is amenable for filtering and analysis.

•Revolutionized the field of signal processing.



Convolution Theorem

“Multiply F(u,v) by a filter function H(u,v)”

This step exploits the convolution theorem:

Convolution in one domain is equivalent to multiplication in the other domain.

$$\boxed{\begin{array}{l} f(x)*g(x) \leftrightarrow F(u)\,G(u) \\ f(x)\,g(x) \leftrightarrow F(u)*G(u) \end{array}}$$

Frequency Bands

- Low frequencies: general graylevel appearance over smooth areas: slowly varying grayscales.
- High frequencies: responsible for detail, such as edges and noise.
- A filter that attenuates high frequencies while "passing" low frequencies is a *lowpass filter*.
- A filter that attenuates low frequencies while "passing" high frequencies is a *highpass filter*.
- Lowpass filters blur images.
- Highpass filters highlight edges.

Correlation

- Identical to convolution except that kernel is not flipped.
- Kernel is now referred to as the template.
- It is the pattern that we seek to find in the larger image.

$$f(x,y)\circ h(x,y) \Leftrightarrow F^{*}(u,v)H(u,v)$$

Example (2)

Reference image

Template

Correlation image

Peak corresponds to best match location

2D Sampling Theory – A Road Map •

Statement: information is lost during sampling

- Aim: reconstruct original image from sampled image errorless (or what are the conditions for errorless sampling?)
- Sampled image is the product of the image and array of s (mathematical convenience) or apertures. Alternatively, employ the convolution of image transform and impulse response (leads to simpler analysis)
- Resulting in replicated (periodic) spectrum
- Uniqueness – reconstruction of spectrum from sampled spectrum is equivalent to reconstruction of image from sampled image
- Thus, extract replication of the sampled image around the origin (low-pass filtering) and take IFT

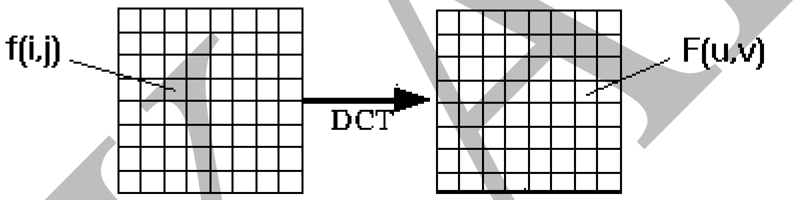
- Conditioned on the image being band-limited and sampling rate complies with Nyquist rate
- Summary: an image sampled at uniform spacing can be completely recovered from the sample values if sampled in Nyquist rates

Sampling Theorem

- A signal can be reconstructed from its samples, if the original signal has no frequencies above 1/2 the sampling frequency - Shannon
- The minimum sampling rate for bandlimited function is called "Nyquist rate"

The Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain (Fig 7.8).



DCT Encoding

The general equation for a 1D (N data items) DCT is defined by the following equation:

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \Lambda(i).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] f(i)$$

and the corresponding ***inverse*** 1D DCT transform is simple $F^{-1}(u)$, i.e.:

where

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

The general equation for a 2D (*N* by *M* image) DCT is defined by the following equation:

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i).\Lambda(j).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$

and the corresponding ***inverse*** 2D DCT transform is simple $F^{-1}(u,v)$, i.e.:

where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

The basic operation of the DCT is as follows:

- The input image is N by M;
- f(i,j) is the intensity of the pixel in row i and column j;
- F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.
- For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.
- Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.
- The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level;
- 8 bit pixels have levels from 0 to 255.
- Therefore an 8 point DCT would be:

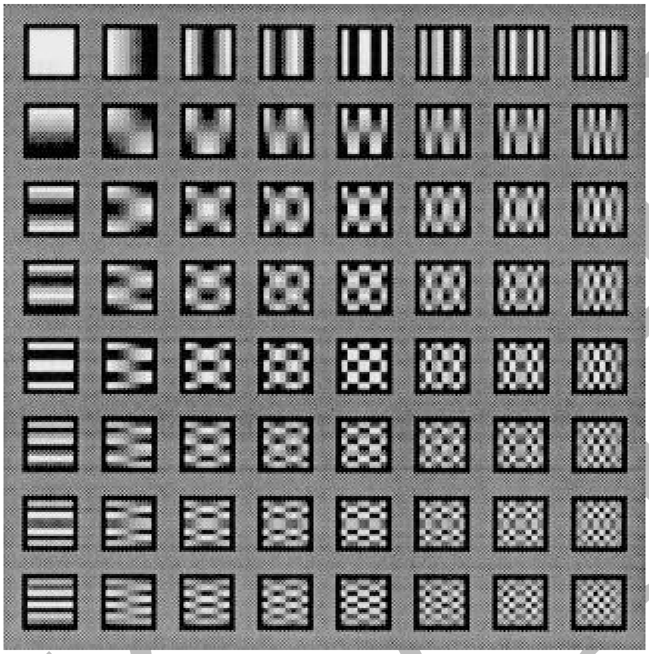
where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

Question: What is F[0,0]?

answer: They define DC and AC components.

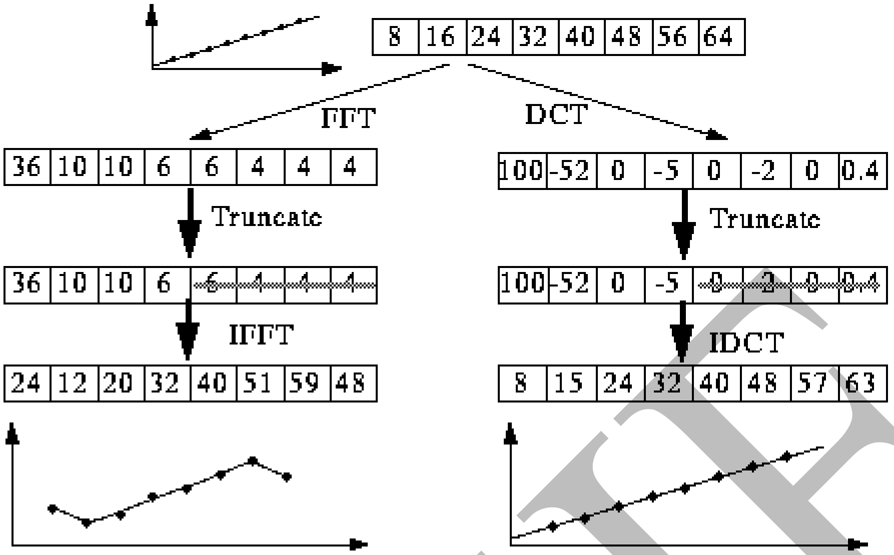
- The output array of DCT coefficients contains integers; these can range from -1024 to 1023.
- It is computationally easier to implement and more efficient to regard the DCT as a set of **basis functions** which given a known input array size (8 x 8) can be precomputed and stored. This involves simply computing values for a convolution mask (8 x8 window) that get applied (sum values x pixel the window overlap with image apply window accros all rows/columns of image). The values as simply calculated from the DCT formula. The 64 (8 x 8) DCT basis functions are illustrated in Fig 7.9.



DCT basis functions

- Why DCT not FFT?

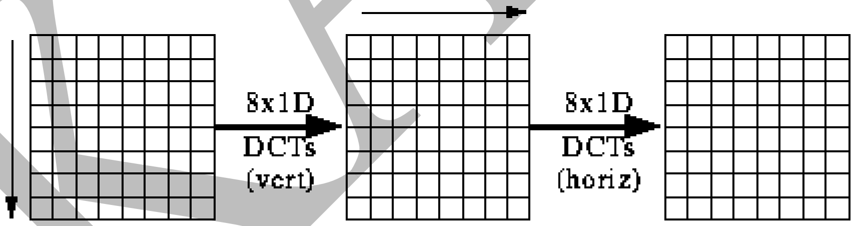
DCT is similar to the Fast Fourier Transform (FFT), but can approximate lines well with fewer coefficients (Fig 7.10)



DCT/FFT Comparison

- Computing the 2D DCT
  - Factoring reduces problem to a series of 1D DCTs (Fig 7.11):
    - apply 1D DCT (Vertically) to Columns
    - apply 1D DCT (Horizontally) to resultant Vertical DCT above.
    - or alternatively Horizontal to Vertical.

The equations are given by:



  - Most software implementations use fixed point arithmetic. Some fast implementations approximate coefficients so all multiplies are shifts and adds.

FREQUENCY DOMAIN FILTERING:

- **Low frequencies in the Fourier transform are responsible for the general gray-level appearance of an image over smooth areas, while high frequencies are responsible for detail, such as edges and noise.**

- **These ideas are discussed in more detail in the sections that follow, but it will be instructive to complement our illustration of the notch filter with an example of filters in these other two categories.**

- **A filter that attenuates high frequencies while "passing" low frequencies is called a *lowpass filter.***
- **A filter that has the opposite characteristic is appropriately called a *highpass filter***

The slowest varying frequency component ($u=v=0$) corresponds to the average gray level of an image. As we move away from the origin of the transform, the low frequencies correspond to the slowly varying components of an image.

In an image of a room, for example, these might correspond to smooth gray-level variations on the walls and door.

As we move further away from the origin, the higher frequencies begin to correspond to faster and faster gray level changes in the image. These arc the edges of objects and other components of an image characterized by abrupt changes in gray level. such as noise.

Basics of filtering in the frequency domain

Filtering in the frequency domain is straightforward. It consists of the following steps:
1. Multiply the input image by $(-1)^{x+y}$ to center the transform, as indicated in Eq. (4.2-21).
2. Compute $F(u, v)$, the DFT of the image from (1).
3. Multiply $F(u, v)$ by a *filter* function $H(u, v)$.
4. Compute the inverse DFf of the result in (3).
5. Obtain the real part of the result in (4).
6. Multiply the result in (5) by $(-1)^{x+y}$.

The reason that $H(u, v)$ is called a *filter* (the term *filter transfer function* also is used commonly) is because it suppresses certain frequencies in the transform while leaving others unchanged.

Basic Steps

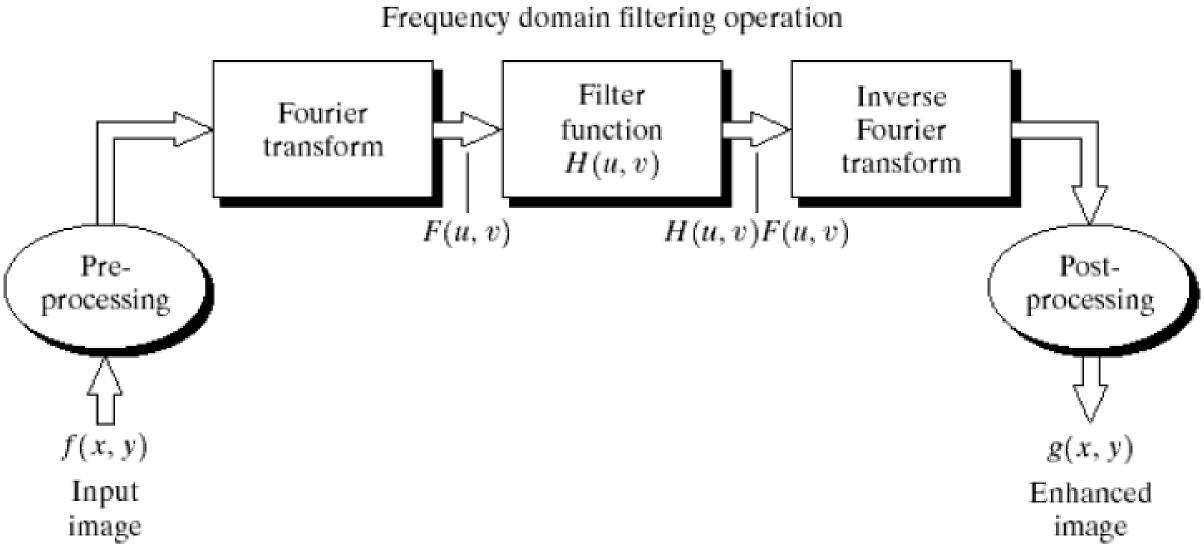


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Smoothing Frequency-Domain Filters

Edges and other sharp transitions (such as noise) in the gray levels of an image contribute significantly to the high-frequency content of its Fourier transform. Hence smoothing (blurring) is achieved in the frequency domain by attenuating a specified range of high-frequency components in the transform of a given image.

Our basic "model" for filtering in the frequency domain is given by:

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

where *F(u, v)* is the Fourier transform of the image to be smoothed. The objective is to select a filter transfer function *H(u. v)* that yields *G(u, v·)* by attenuating the high-frequency components of *F(u, v)*.

Types:

Ideal, Butterworth, and Gaussian filters.

These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions. The Butterworth filter has a parameter. called the filter *order.*

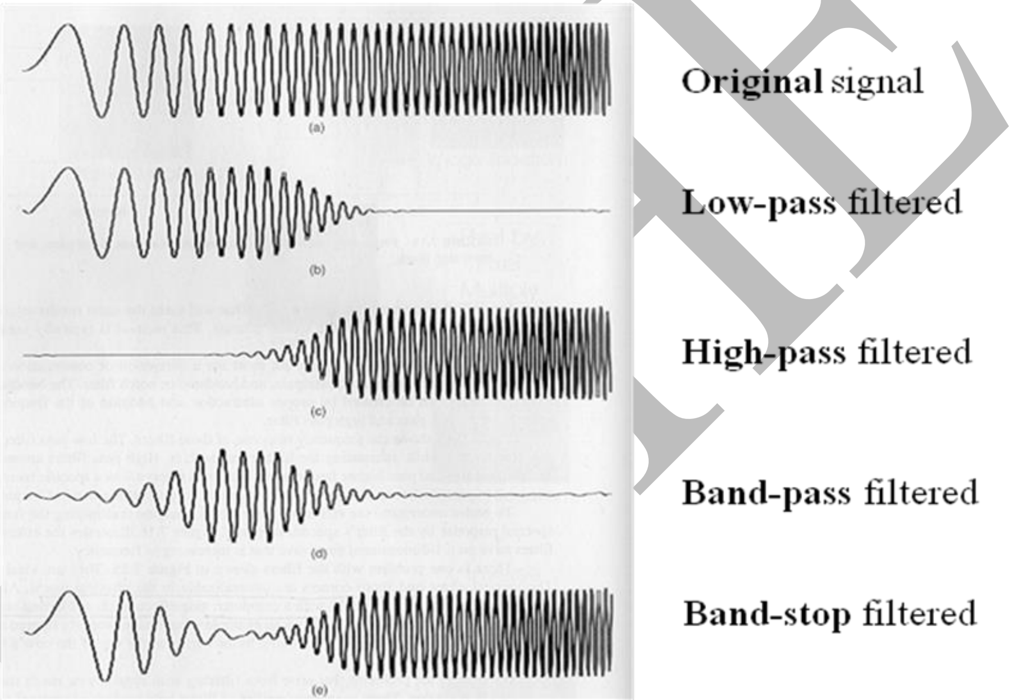
- Typically, filters are classified by examining their properties in the frequency domain:

(1) Low-pass

(2) High-pass

(3) Band-pass

(4) Band-stop



Low Pass (LP) Filters

- Ideal low-pass filter (ILPF)
- Butterworth low-pass filter (BLPF)
- Gaussian low-pass filter (GLPF)

Lowpass and Highpass Filters

Lowpass filter

Highpass filter

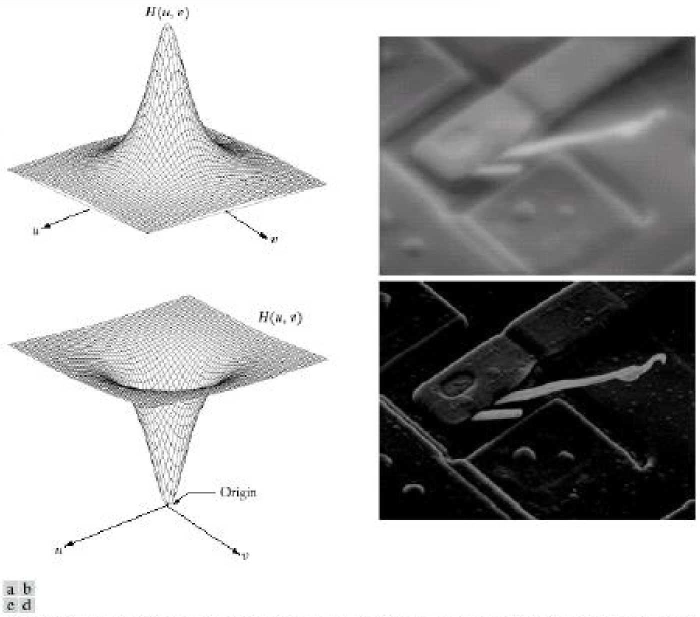
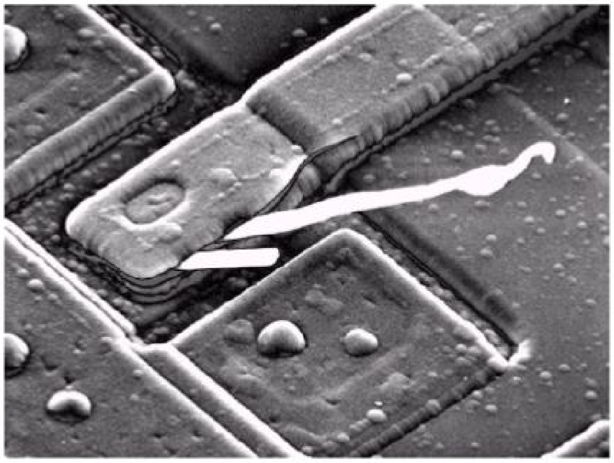


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

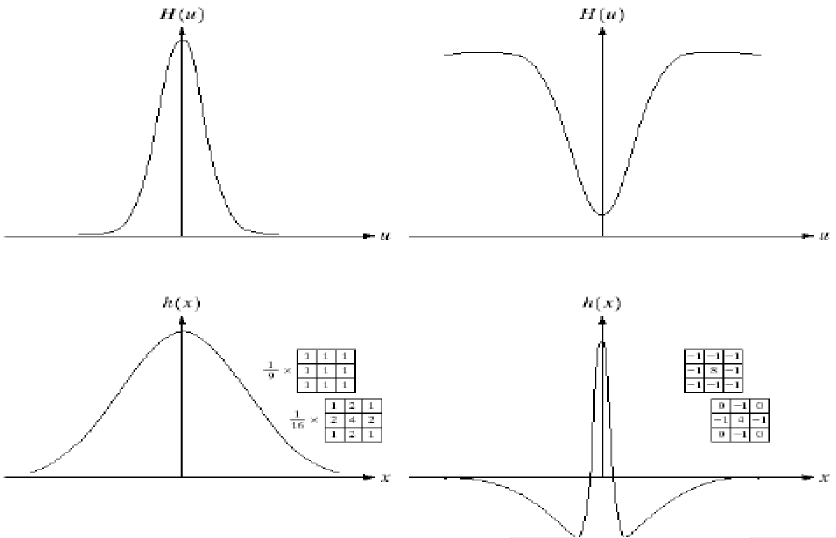
Improved Highpass Output

- Add constant to filter so that it will not eliminate F(0,0)

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).



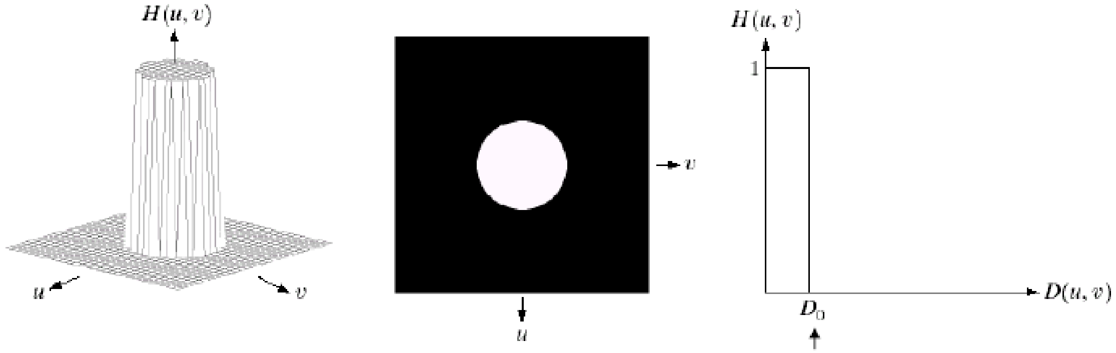
Corresponding Filters in the Spatial and Frequency Domains



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.

Ideal Lowpass Filter



cutoff frequency

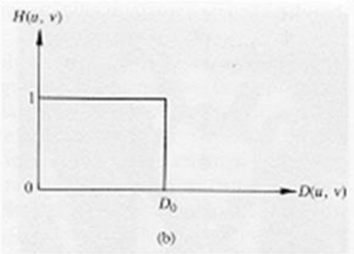
a b c

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

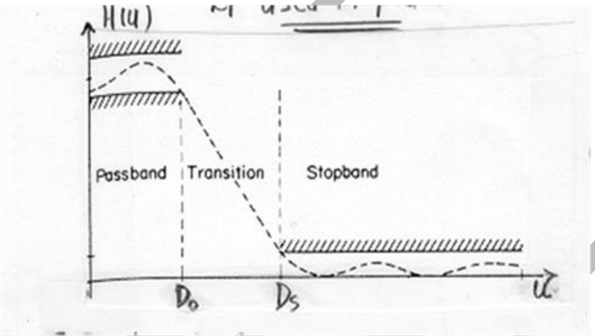
Low-pass (LP) filtering

- Preserves low frequencies, attenuates high frequencies.

Ideal



In practice



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

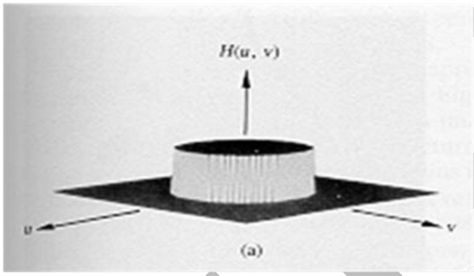
D_0: cut-off frequency

where D_0 is a specified nonnegative quantity, and D(u,v) is the distance from point *(u, v)* to the center of the frequency rectangle. If the image in question is of size *M* x *N,* we know that its transform also is of this size, so the center of the frequency rectangle is at (u, *v)* = *(M*/2, *N*/2) due to the fact that the transform has been centered.

Lowpass (LP) filtering (cont'd)

- In 2D, the cutoff frequencies are specified by a circle.

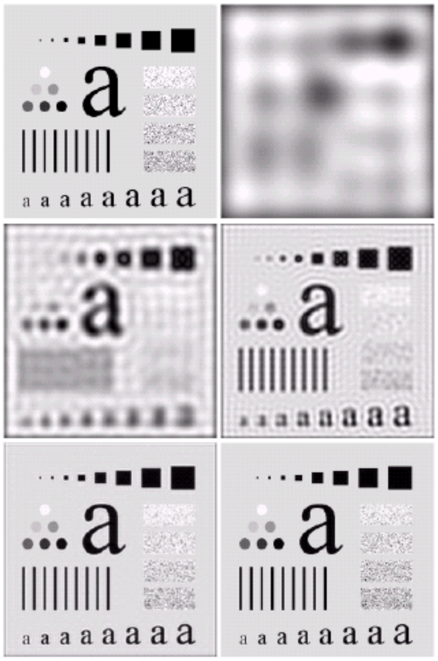
Ideal



$$H(u,v) = \begin{cases} 1 & \text{if } u^2 + v^2 \le D_0{}^2 \\ 0 & otherwise \end{cases}$$

Varying ILPF Cutoff Frequencies

- Most sharp detail in this image is contained in the 8% power removed by filter.
- Ringing behavior is a characteristic of ideal filters.
- Little edge info contained in upper 0.5% of spectrum power in this case.



a b
c d
e f

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.

ILPF Ringing

- Ringing in spatial filter has two characteristics: dominant component at origin and concentric circular components.
- Center component responsible for blurring.
- Concentric components responsible for ringing.
- Radius of center comp. and #circles/unit distance are inversely proportional to cutoff frequency.

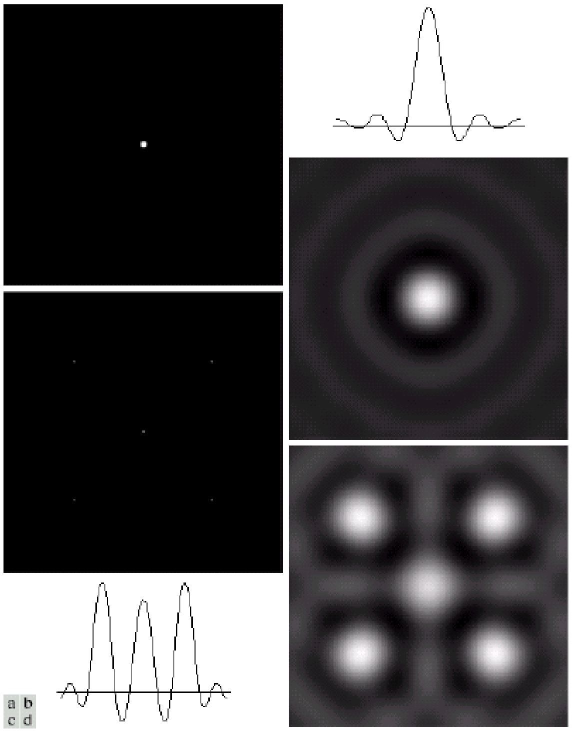


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Butterworth Lowpass Filter (BLPF)

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

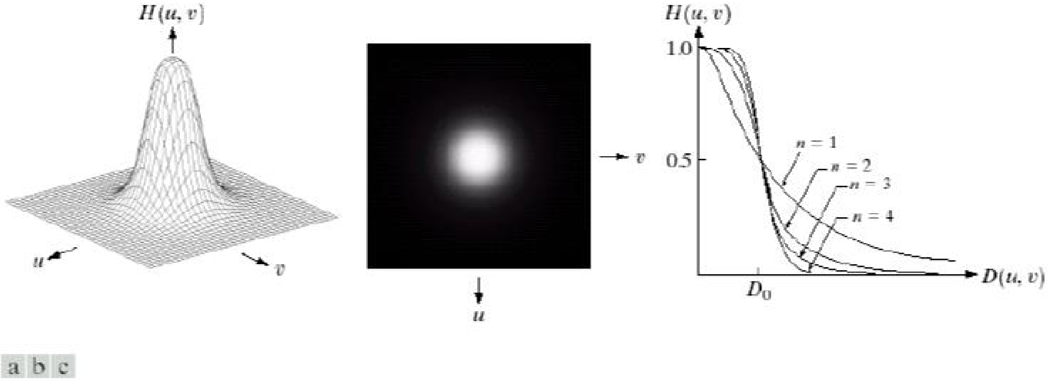
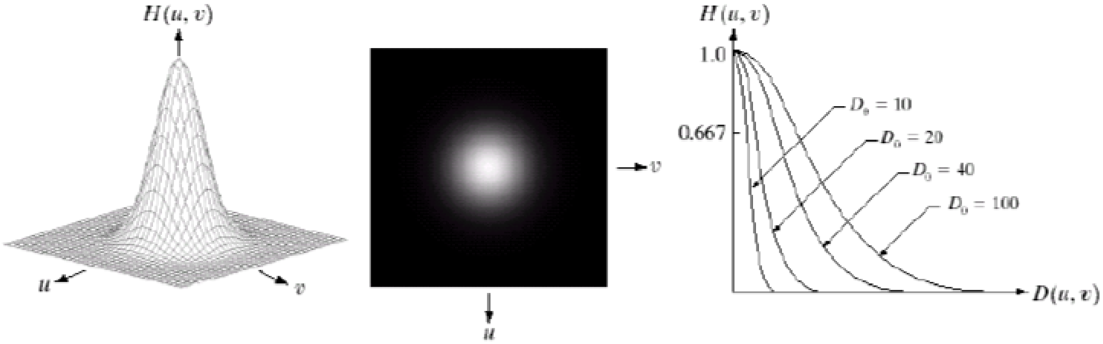


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Unlike the ILPE the BLPF transfer function docs not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies.

Gaussian Lowpass Filter (GLPF)

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \longleftarrow \sigma = D_0 = \text{cutoff freq}$$



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.

D(u, v) is the distance from the origin of the Fourier transform. which we assume has been shifted to the center of the frequency rectangle

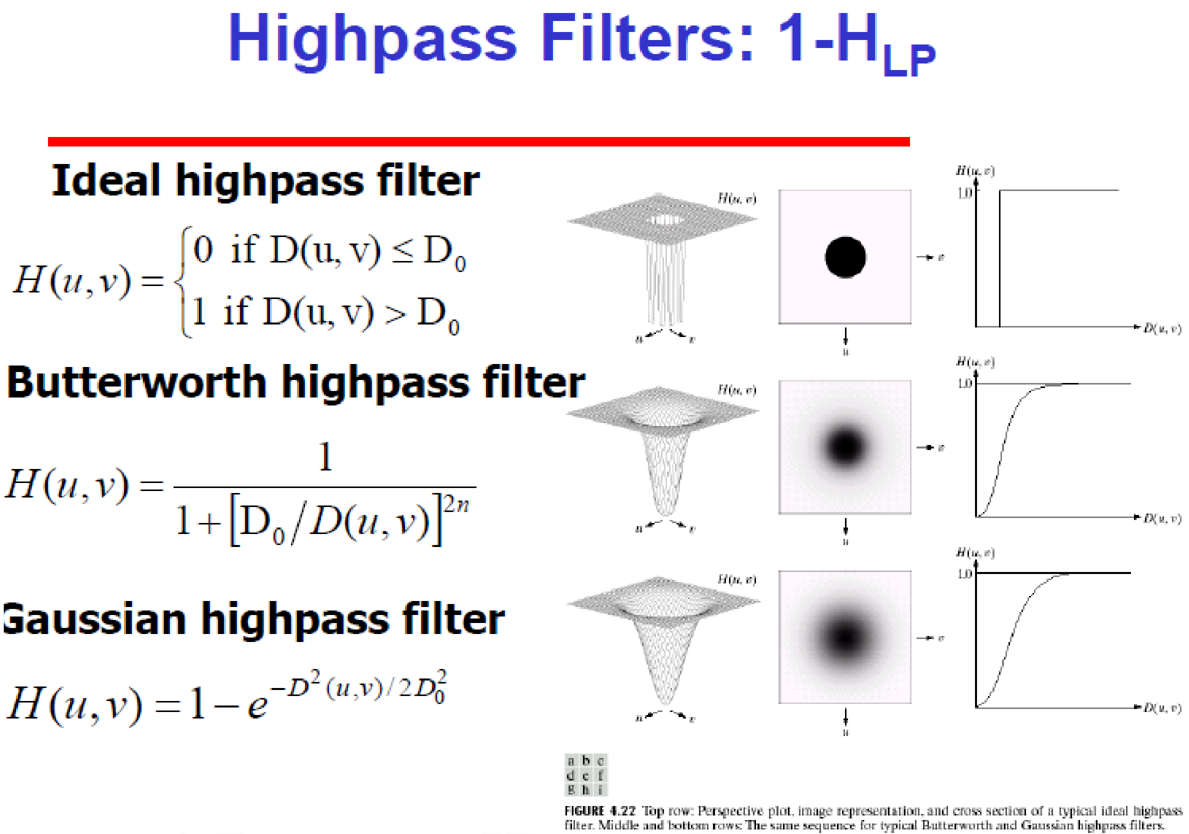
Sharpening Frequency Domain Filters

In the previous section we showed that an image can be blurred by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in gray levels are associated with high-frequency components, image sharpening can be achieved in the frequency domain by a *highpass filtering* process, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

Perform precisely the reverse operation of the ideal lowpass filters discussed in the previous section, the transfer function of the highpass filters discussed in this section can be obtained using the relation

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

where $H_{lp}'(u, v)$ is the transfer function of the corresponding lowpass filter. That is, when the lowpass filter attenuates frequencies, the highpass filter passes them, and vice versa.



Highpass Filters: 1-H_{LP}

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

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

Gaussian highpass filter

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

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.

Ideal Highpass Filters

As intended. this filter is the opposite of the ideal lowpass filter in the sense that it sets to zero all frequencies inside a circle of radius D_0 while passing, without attenuation, all frequencies outside the circle

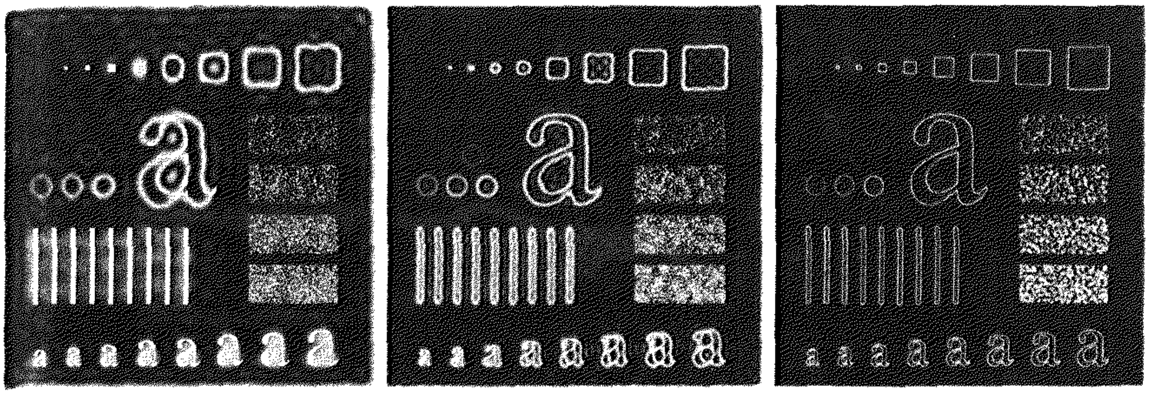

a b c

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

As in the ease of lowpass filters, we. can expect Butterworth highpass fillers to behave smoother than IHPFs. The performance of a BHPE of order 2 and with D_{II} set to the same values as in Fig. 4.24, is shown in Fig. 4.25. The boundaries are much less distorted than in Fig. 4.24. even for the smallest value of cutoff frequency. Since the center spot sizes of the IHPF and the BHPF are similar [sec Figs. 4.23(a) and (b)].the performance of the two filters in terms of filtering the smaller objects is comparable. The transition into higher values of cutoff frequencies is much smoother with the BIIPF

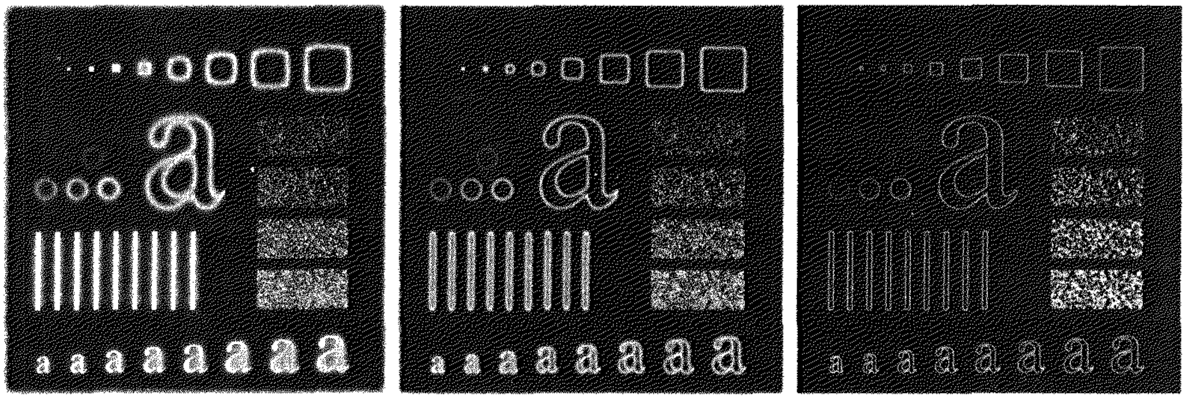
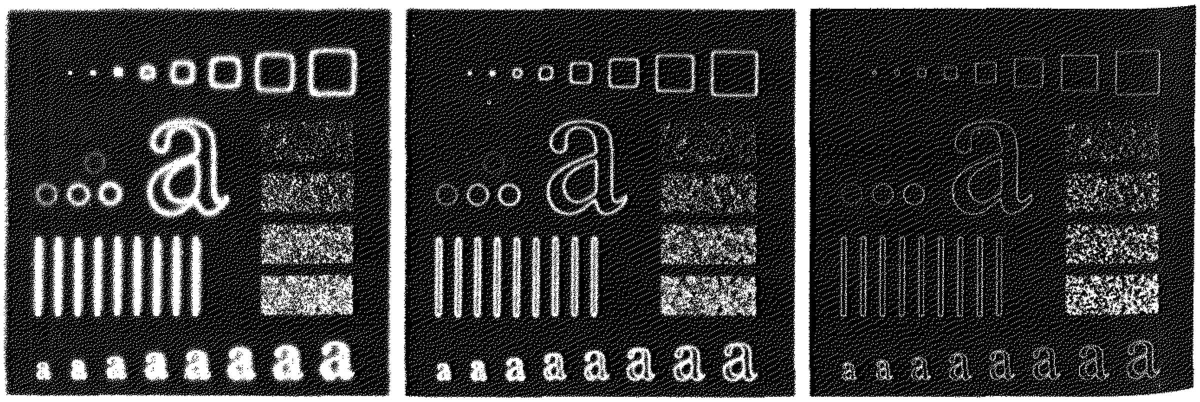

a b c

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 smoothe.: than those obtained with an ILPF.

Gaussian Highpass Filters

As expected. The results obtained are smoother than with the previous two filters. Even the filtering of the smaller objects and thin bars is cleaner with the Gaussian filter.



a b c

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.

Unsharp Masking, High-Boost Filtering, and High-Frequency Emphasis Filtering

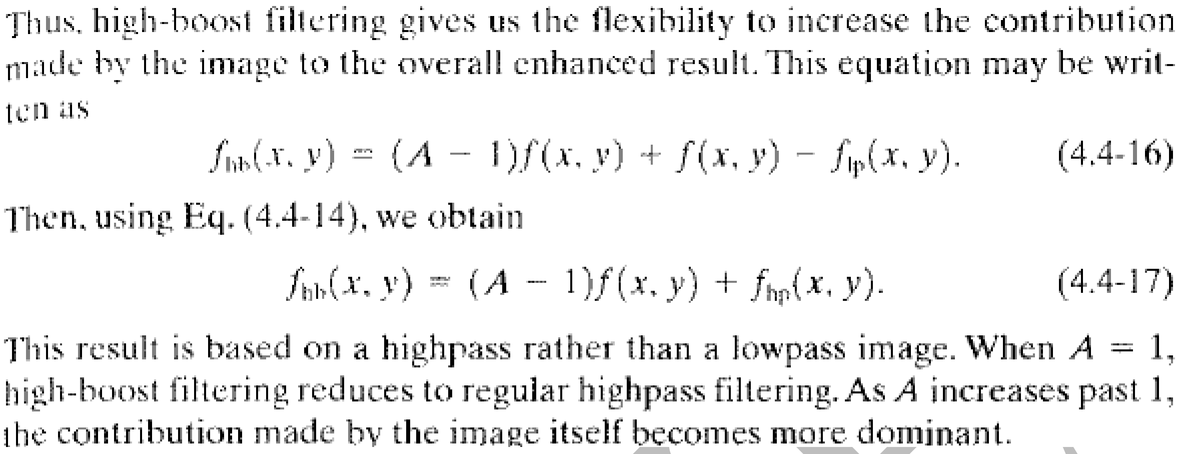
All the filtered images in Sections 4.4.1 through 4.4.3 have one thing in common: Their average background intensity has been reduced to near black. This is due to the fact that the highpass filters we applied to those images eliminate the zero-frequency component of their Fourier transforms (see the discussion in Section 4.2.3 regarding this phenomenon). As discussed in Section 3.7.2, the solution to this problem consists of adding a portion of the image back to the filtered result. In fact, enhancement using the Laplacian does precisely this, by adding back the entire image to the filtered result. Sometimes it is advantageous to increase the contribution made by the original image to the overall filtered result. This approach. called *high-boost filtering.* is a generalization of *unsharp masking.*

These concepts were introduced in Section 3.7.2. We repeat them here using frequency domain concepts and notation. Unsharp masking consists simply of generating a sharp image by subtracting from an image a blurred version of itself. Using frequency domain terminology. this means obtaining a highpass filtered image by subtracting from the image a lowpass-filtered version of itself. That is.

$$f_{\text{hp}}(x, y) = f(x, y) - f_{\text{lp}}(x, y). \tag{4.4-14}$$

High-boost filtering generalizes this by multiplying $f(x, y)$ by a constant $A \geq 1$:

$$f_{\text{hb}} = Af(x, y) - f_{\text{lp}}(x, y). \tag{4.4-15}$$



Thus, high-boost filtering gives us the flexibility to increase the contribution made by the image to the overall enhanced result. This equation may be written as

$$f_{hb}(x, y) = (A - 1)f(x, y) + f(x, y) - f_{lp}(x, y). \qquad (4.4\text{-}16)$$

Then, using Eq. (4.4-14), we obtain

$$f_{hb}(x, y) = (A - 1)f(x, y) + f_{hp}(x, y). \qquad (4.4\text{-}17)$$

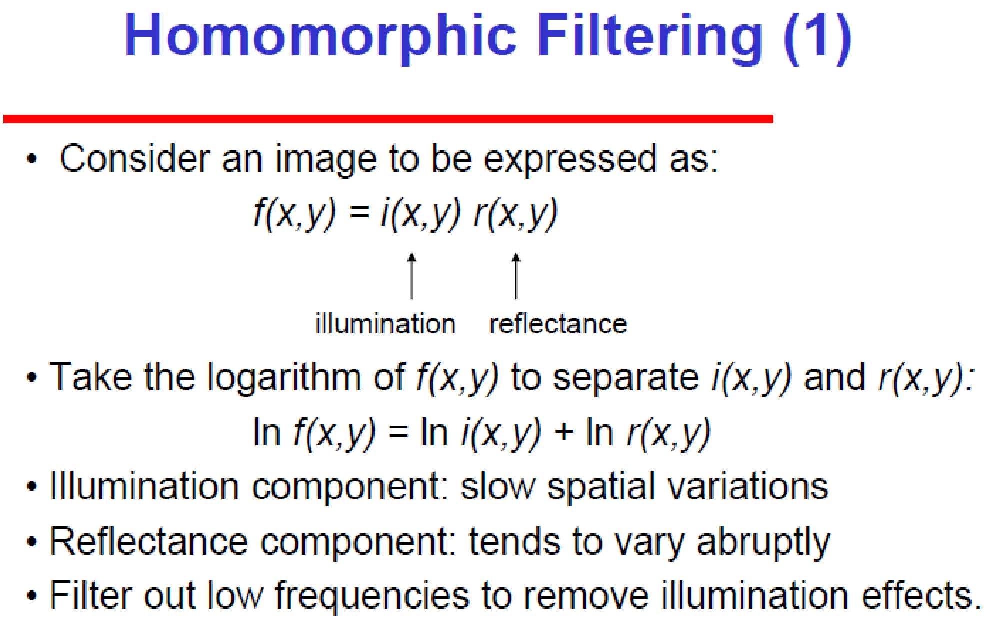
This result is based on a highpass rather than a lowpass image. When $A = 1$, high-boost filtering reduces to regular highpass filtering. As A increases past 1, the contribution made by the image itself becomes more dominant.

Homomorphic Filtering

The illumination-reflectance model introduced in Section 2.3..+ can be used to develop a frequency domain procedure for improving the appearance of an image by simultaneous gray-level range compression and contrast enhancement.

From the discussion in Section 2.3.4. an image *f(x. y)* can be expressed as the product of illumination and reflectance components:

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



Homomorphic Filtering (1)

- Consider an image to be expressed as:

$$f(x,y) = \underset{\text{illumination}}{\underset{\uparrow}{i(x,y)}}\ \underset{\text{reflectance}}{\underset{\uparrow}{r(x,y)}}$$

- Take the logarithm of *f(x,y)* to separate *i(x,y)* and *r(x,y)*:

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

- Illumination component: slow spatial variations
- Reflectance component: tends to vary abruptly
- Filter out low frequencies to remove illumination effects.

Homomorphic Filtering (2)

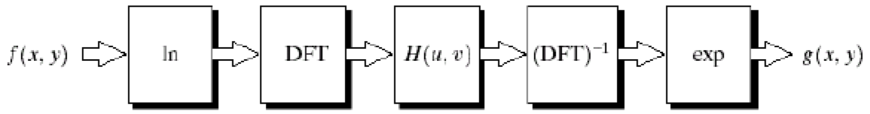


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

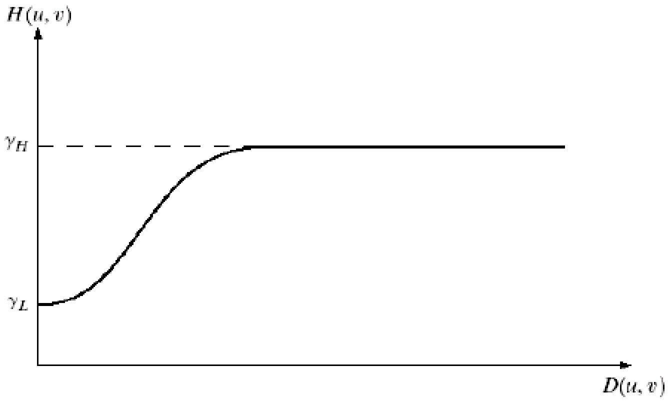


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

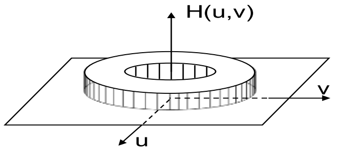
Band Pass Filtering

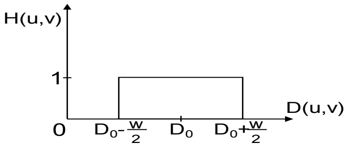
$$H(u,v) = \begin{cases} 0 & D(u,v) \leq D_0 - \frac{w}{2} \\ 1 & D_0 - \frac{w}{2} \leq D(u,v) \leq D_0 + \frac{w}{2} \\ 0 & D(u,v) > D_0 + \frac{w}{2} \end{cases}$$

$D(u,v) = \sqrt{u^2 + v^2}$

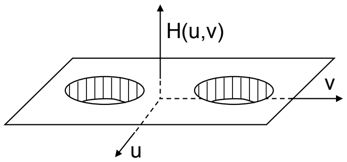
D_0 = cut off frequency

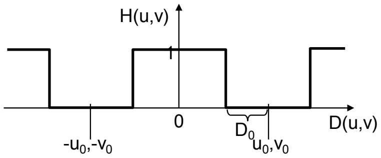
w = band width





Band Rejection Filtering





$$H(u,v) = \begin{cases} 0 & D_1(u,v) \leq D_0 \ \textbf{or} \ D_2(u,v) \leq D_0 \\ 1 & \textbf{otherwise} \end{cases}$$

$D_1(u,v) = \sqrt{(u-u_0)^2 + (v-v_0)^2}$

$D_2(u,v) = \sqrt{(u+u_0)^2 + (v+v_0)^2}$

D_0 = local frequency radius

u_0, v_0 = local frequency coordinates

KARPAGAM ACADEMY OF HIGHER EDUCATION

Enable | Enlighten | Enrich
KARPAGAM
ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act, 1956)

CLASS: III BSC CS COURSE NAME: DIGITAL IMAGE PROCESSING
COURSE CODE: 16CSU502B BATCH-2016-2019

UNIT II

SNO	QUESTIONS	CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	ANSWER
1	What is the name of the filter that is used to turn the average value of a processed image zero?	Unsharp mask filter	Notch filter	Zero-phase-shift-filter	None of the mentioned	Notch filter
2	Which of the following filter(s) attenuates high frequency while passing low frequencies of an image?	Unsharp mask filter	Lowpass filter	Zero-phase-shift filter	All of the mentioned	Lowpass filter
3	Which of the following filter(s) attenuates low frequency while passing high frequencies of an image?	Unsharp mask filter	Highpass filter	Zero-phase-shift filter	All of the mentioned	Highpass filter
4	Which of the following filter have a less sharp detail than the original image because of attenuation of high frequencies?	Highpass filter	Lowpass filter	Zero-phase-shift filter	None of the mentioned	Lowpass filter
5	The feature(s) of a highpass filtered image is/are ____________	Have less gray-level variation in smooth areas	Emphasized transitional gray-level details	An overall sharper image	All of the mentioned	All of the mentioned
6	A spatial domain filter of the corresponding filter in frequency domain can be obtained by applying which of the following operation(s) on filter in frequency domain?	Fourier transform	Inverse Fourier transform	None of the mentioned	All of the mentioned	Inverse Fourier transform
7	A frequency domain filter of the corresponding filter in spatial domain can be obtained by applying which of the following operation(s) on filter in spatial domain?	Fourier transform	Inverse Fourier transform	None of the mentioned	All of the mentioned	Fourier transform

8	Which of the following filtering is done in frequency domain in correspondence to lowpass filtering in spatial domain?	Gaussian filtering	Unsharp mask filtering	High-boost filtering	None of the mentioned	Gaussian filtering
9	Which of the following is/are considered as type(s) of lowpass filters?	Ideal	Butterworth	Gaussian	All of the mentioned	All of the mentioned
10	Which of the following lowpass filters is/are covers the range of very sharp filter function?	Ideal lowpass filters	Butterworth lowpass filter	Gaussian lowpass filter	All of the mentioned	Ideal lowpass filters
11	Which of the following lowpass filters is/are covers the range of very smooth filter function?	Ideal lowpass filters	Butterworth lowpass filter	Gaussian lowpass filter	All of the mentioned	Gaussian lowpass filter
12	The characteristics of the lowpass filter h(x, y) is/are_________	Has a dominant component at origin	Has a concentric, circular components about the center component	All of the mentioned	None of the mentioned	All of the mentioned
13	In frequency domain terminology, which of the following is defined as "obtaining a highpass filtered image by subtracting from the given image a lowpass filtered version of itself"?	Emphasis filtering	Unsharp masking	Butterworth filtering	None of the mentioned	Unsharp masking
14	Which of the following is/ are a generalized form of unsharp masking?	Lowpass filtering	High-boost filtering	Emphasis filtering	All of the mentioned	High-boost filtering
15	Subtracting Laplacian from an image is proportional to which of the following?	Unsharp masking	Box filter	Median filter	None of the mentioned	Unsharp masking
16	A First derivative in image processing is implemented using which of the following given operator(s)?	Magnitude of Gradient vector	The Laplacian	All of the mentioned	None of the mentioned	Magnitude of Gradient vector
17	What is the sum of the coefficient of the mask defined using gradient?	1	-1	0	None of the mentioned	0
18	Gradient is used in which of the following area(s)?	To aid humans in detection of defects	As a preprocessing step for automated inspections	All of the mentioned	None of the mentioned	All of the mentioned

19	Gradient have some important features. Which of the following is/are some of them?	Enhancing small discontinuities in an otherwise flat gray field	Enhancing prominent edges	All of the mentioned	None of the mentioned	All of the mentioned
20	An image has significant edge details. Which of the following fact(s) is/are true for the gradient image and the Laplacian image of the same?	The gradient image is brighter than the Laplacian image	The gradient image is brighter than the Laplacian image	Both the gradient image and the Laplacian image has equal values	None of the mentioned	The gradient image is brighter than the Laplacian image
21	Fourier transform of unit impulse at origin is _____________	undefined	infinity	1	0	1
22	Continuous functions are sampled to form a ____________	Fourier series	Fourier transform	fast Fourier series	digital image	digital image
23	The common example of 2D interpolation is image	enhancement	sharpening	blurring	resizing	resizing
24	2D Fourier transform and its inverse are infinitely	aperiodic	periodic	linear	non linear	periodic
25	Shrinking of image can be done using ______________	pixel replication	bicubic interpolation	bilinear interpolation	row column deletion	bilinear interpolation
26	Sum of many infinitely many periodic impulses is called ___________	aperiodic impulse	periodic impulse	impulse train	summation	impulse train
27	To reduce the effect of aliasing high frequencies are __________	attenuated	accentuated	reduced	removed	attenuated
28	If f(x,y) is imaginary, then its Fourier transform is __________	conjugate symmetry	hermition	antihermition	symmetry	antihermition
29	The product of two even or two odd functions is ___________	even	odd	prime	aliasing	even
30	Unit impulse at every point other than 0 is ________	undefined	infinity	1	0	0
31	The sum of cosines and sines coefficient multiplied is __________	Fourier series	Fourier transform	fast Fourier series	fast Fourier transform	Fourier series
32	A continuous band limited function can be recovered with no error if sampled intervals are less than 1/2umax is the statement of	2D sampling series	3D sampling theorem	1D sampling theorem	2D sampling theorem	2D sampling theorem

33	The impulse S(t,v) is function having _______	one variable	two variables	three variables	four variables	two variables
34	Any function whose Fourier transform is zero for frequencies outside the finite interval is called _________	high pass function	low pass function	band limited function	band pass function	band limited function
35	Fourier transform's domain is ______	frequency domain	spatial domain	Fourier domain	time domain	frequency domain
36	Giving one period of the periodic convolution is called _________	periodic convolution	aperiodic convolution	correlation	circular convolution	circular convolution
37	The continuous variables in 2D transform pair are interpreted as _________	continuous frequency variables	spatial variables	continuous spatial variables	discrete spatial variables	continuous spatial variables
38	Fourier transform of two continuous functions, that are inverse of each other is called ___________	Fourier series pair	Fourier transform pair	Fourier series	Fourier transform	Fourier transform pair
39	Forward and inverse Fourier transforms exist for the samples having values _________	integers	infinite	finite	discrete	finite
40	The greater, the values of continuous variables, the spectrum of Fourier transform will be _________	contracted	expanded	discrete	continuous	contracted
41	Low pass filters are used for image ________	contrast	sharpening	blurring	resizing	blurring
42	Fourier stated that the periodic function is expressed as sum of _________	sine	cosine	tangent	Both A and B	Both A and B
43	The effect caused by under sampling is called __________	smoothing	sharpening	summation	aliasing	aliasing
44	Product of two functions in spatial domain is what, in frequency domain	correlation	convolution	Fourier transform	fast Fourier transform	convolution
45	Digitizing the coordinate values is called _________	quantization	sampling	Fourier transform	discrete Fourier transform	sampling
46	Shrinking of image is viewed as _________	under sampling	over sampling	critical sampling	nyquist sampling	under sampling
47	High pass filters are used for image _________	contrast	sharpening	blurring	resizing	sharpening

48	The Fourier transform is named after French mathematician _________	joseph Fourier	john Fourier	sean Fourier	jay Fourier	joseph Fourier
49	Which of the following is the primary objective of sharpening of an image?	Blurring the image	Highlight fine details in the image	Increase the brightness of the image	Decrease the brightness of the image	Highlight fine details in the image
50	In spatial domain, which of the following operation is done on the pixels in sharpening the image?	Integration	Average	Median	Differentiation	Differentiation
51	In which of the following cases, we wouldn't worry about the behaviour of sharpening filter?	Flat segments	Step discontinuities	Ramp discontinuities	Slow varying gray values	Slow varying gray values
52	Which of the following is the valid response when we apply a first derivative?	Non-zero at flat segments	Zero at the onset of gray level step	Zero in flat segments	Zero along ramps	Zero in flat segments
53	Which of the following is not a valid response when we apply a second derivative?	Zero response at onset of gray level step	Nonzero response at onset of gray level step	Zero response at flat segments	Nonzero response along the ramps	Nonzero response at onset of gray level step
54	If f(x,y) is an image function of two variables, then the first order derivative of a one dimensional function, f(x) is:	f(x+1)-f(x)	f(x)-f(x+1)	f(x-1)-f(x+1)	f(x)+f(x-1)	f(x+1)-f(x)
55	What is the thickness of the edges produced by first order derivatives when compared to that of second order derivatives?	Finer	Equal	Thicker	Independent	Thicker
56	Which of the following derivatives produce a double response at step changes in gray level?	First order derivative	Third order derivative	Second order derivative	First and second order derivatives	Second order derivative
57	The domain that refers to image plane itself and the domain that refers to Fourier transform of an image is/are : _________	Spatial domain in both	Frequency domain in both	Spatial domain and Frequency domain respectively	Frequency domain and Spatial domain respectively	Spatial domain and Frequency domain respectively
58	What is accepting or rejecting certain frequency components called as?	Filtering	Eliminating	Slicing	None of the Mentioned	Filtering

59	What is the process of moving a filter mask over the image and computing the sum of products at each location called as?	Convolution	Correlation	Linear spatial filtering	Non linear spatial filtering	Correlation
60	Convolution and Correlation are functions of _____________	Distance	Time	Intensity	Displacement	Displacement

UNIT-III:

Image Restoration, Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections, Image Compression-Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding.

IMAGE RESTORATION:

Image restoration Image restoration techniques aim at modelling a degradation corrupting the image and inverting this degradation to correct the image so that it is as close as possible to the original.

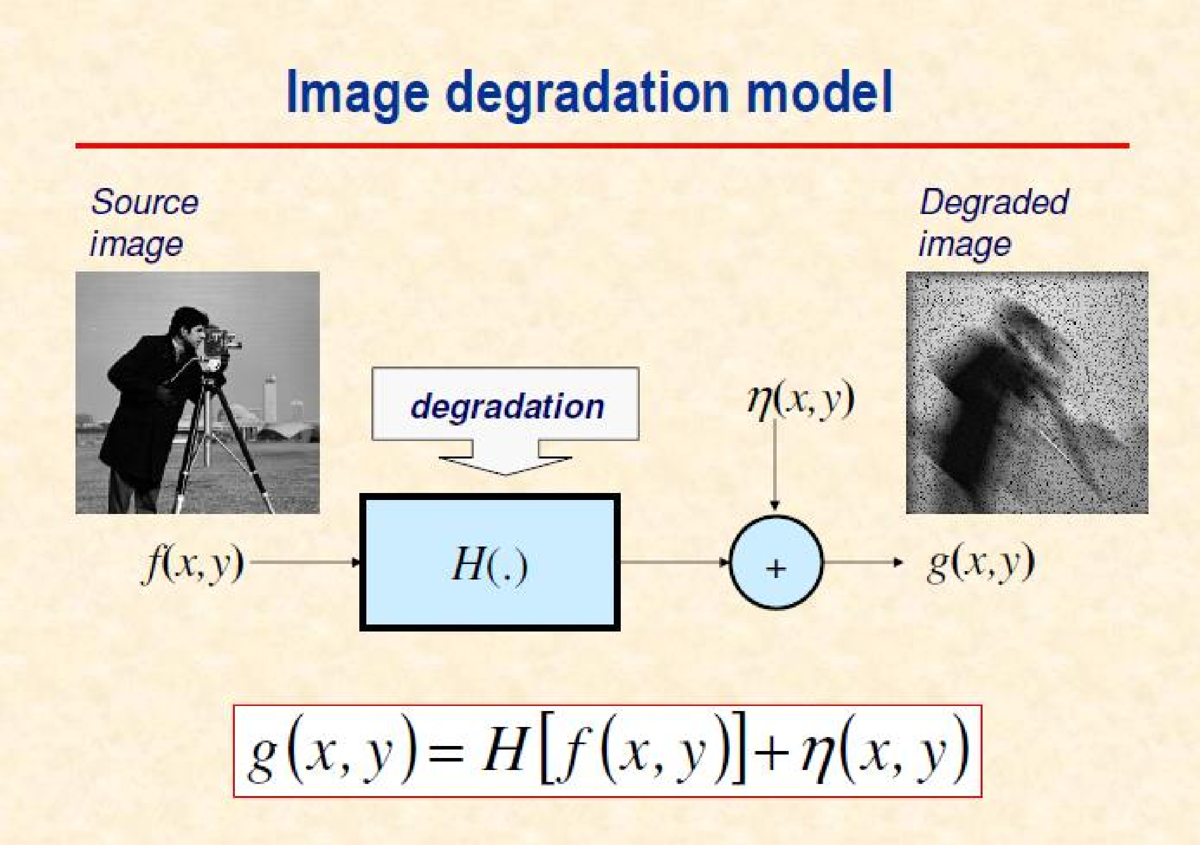


Image degradation model

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

Image Restoration is the operation of taking a corrupt/noisy **image** and estimating the clean, original **image**. Corruption may come in many forms such as motion blur, noise and camera mis-focus. Image restoration is performed by reversing the process that blurred the image and such is performed by imaging a point source and use the point source image, which is called the Point Spread Function (PSF) to restore the image information lost to the blurring process.

The objective of image restoration techniques is to reduce noise and recover resolution loss. Image processing techniques are performed either in the image domain or the frequency domain. The most straightforward and a conventional technique for image restoration is deconvolution, which is performed in the frequency domain and after computing the Fourier transform of both the image and the PSF and undo the resolution loss caused by the blurring factors.

This deconvolution technique, because of its direct inversion of the PSF which typically has poor matrix condition number, amplifies noise and creates an imperfect deblurred image. Also, conventionally the blurring process is assumed to be shift-invariant. Hence more sophisticated techniques, such as regularized deblurring, have been developed to offer robust recovery under different types of noises and blurring functions.

Image restoration is the process of recovering an image that has been degraded by using a priori knowledge of the degradation phenomenon. Restoration techniques involves modeling of the degradation function and applying the inverse process to recover the original image. This process is processed in two domains: spatial domain and frequency domain.

BASIC FRAMEWORK - INTERACTIVE RESTORATION

The Fig. 6.3 shows, the degradation process is modeled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y). Given g(x, y), some knowledge about the degradation function H, and some knowledge about the additive noise term η(x, y), the objective of restoration is to obtain an estimate f(x, y) of the original image. the estimate should be as close as possible to the original input image and, in general, the more we know about H and η, the closer f(x, y) will be to f(x, y).

The degraded image is given in the spatial domain by

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

where h (x, y) is the spatial representation of the degradation function and, the symbol * indicates convolution. Convolution in the spatial domain is equal to multiplication in the frequency domain, hence

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

where the terms in capital letters are the Fourier transforms of the corresponding terms in above equation.

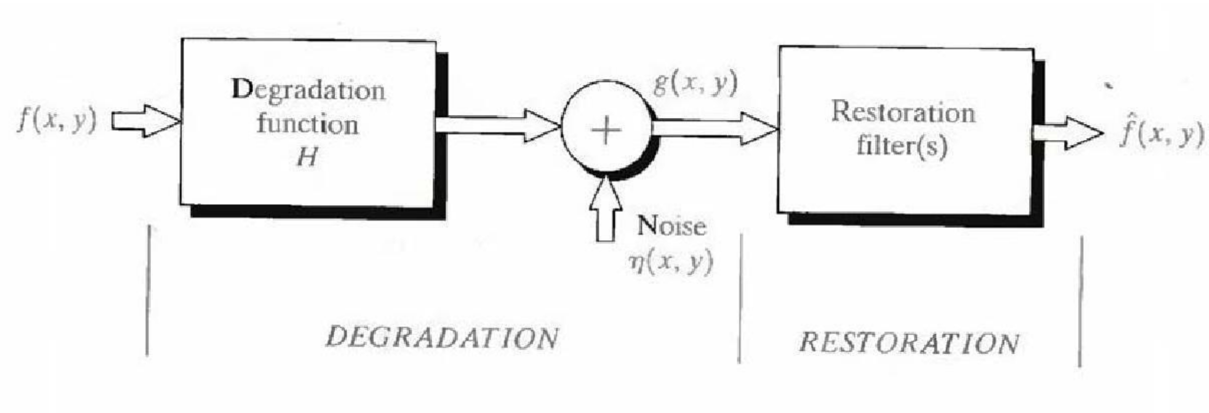


Fig. 6.3 model of the image degradation/restoration process.

GEOMETRIC TRANSFORMATIONS

In terms of digital image processing, a geometric transformation consists of two basic operations: (1) a *spatial transformation,* which defines the "rearrangement" of pixels on the image plane; and (2) *gray-level interpolation,* which deals with the assignment of gray levels to pixels in the spatially transformed image

5.11.1 Spatial Transformations

Suppose that an image f with pixel coordinates (x, y) undergoes geometric distortion to produce an image g with coordinates (x', y'). This transformation may be expressed as

$$x' = r(x, y) \tag{5.11-1}$$

and

$$y' = s(x, y) \tag{5.11-2}$$

where $r(x, y)$ and $s(x, y)$ are the spatial transformations that produced the geometrically distorted image $g(x', y')$. For example, if $r(x, y) = x/2$ and $s(x, y) = y/2$, the "distortion" is simply a shrinking of the size of $f(x, y)$ by one-half in both spatial directions.

If $r(x, y)$ and $s(x, y)$ were known analytically, recovering $f(x, y)$ from the distorted image $g(x', y')$ by applying the transformations in reverse might be possible theoretically. In practice, however, formulating a single set of analytical functions $r(x, y)$ and $s(x, y)$ that describe the geometric distortion process over the entire image plane generally is not possible. The method used most frequently to overcome this difficulty is to formulate the spatial relocation of pixels by the use of *tiepoints*, which are a subset of pixels whose location in the input (distorted) and output (corrected) images is known precisely.

Figure 5.32 shows quadrilateral regions in a distorted and corresponding corrected image. The vertices of the quadrilaterals are corresponding tiepoints.

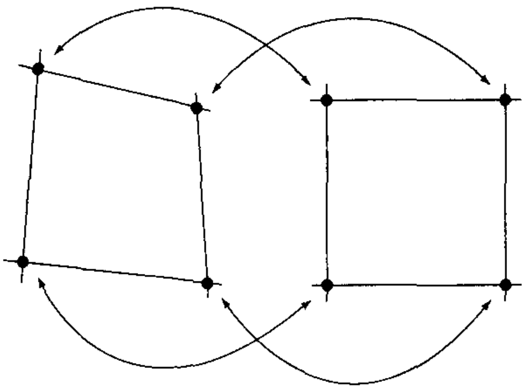


FIGURE 5.32 Corresponding tiepoints in two image segments.

Gray-Level Interpolation

The method discussed in the preceding section steps through integer values of the coordinates (x, y) to yield the restored image $\hat{f}(x, y)$. However, depending on the values of the coefficients c_i, Eqs. (5.11-5) and (5.11-6) can yield noninteger val-

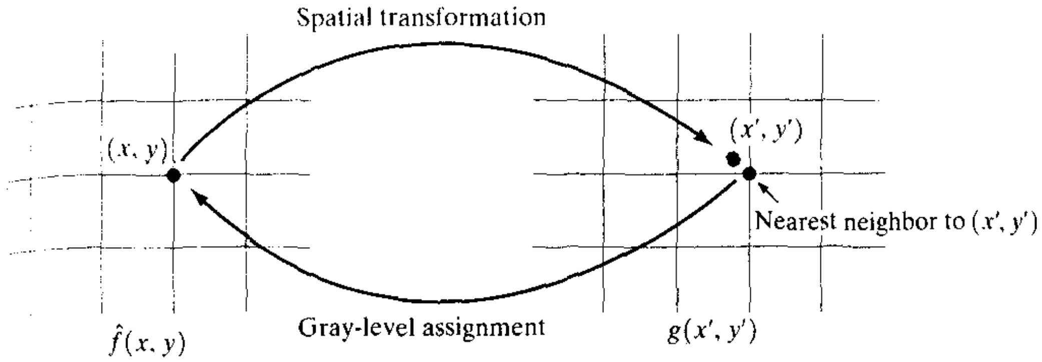
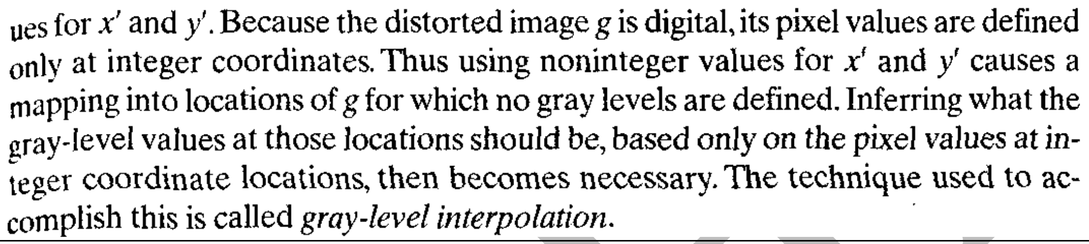


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.



ues for x' and y'. Because the distorted image g is digital, its pixel values are defined only at integer coordinates. Thus using noninteger values for x' and y' causes a mapping into locations of g for which no gray levels are defined. Inferring what the gray-level values at those locations should be, based only on the pixel values at integer coordinate locations, then becomes necessary. The technique used to accomplish this is called *gray-level interpolation*.

- Geometric transforms permit the elimination of geometric distortion that occurs when an image is captured.
- An example is an attempt to match remotely sensed images of the same area taken after one year, when the more recent image was probably not taken from precisely the same position.
- To inspect changes over the year, it is necessary first to execute a geometric transformation, and then subtract one image from the other.

- To inspect changes over the year, it is necessary first to execute a geometric transformation, and then subtract one image from the other.

- A geometric transform is a vector function T that maps the pixel (x,y) to a new position (x',y').

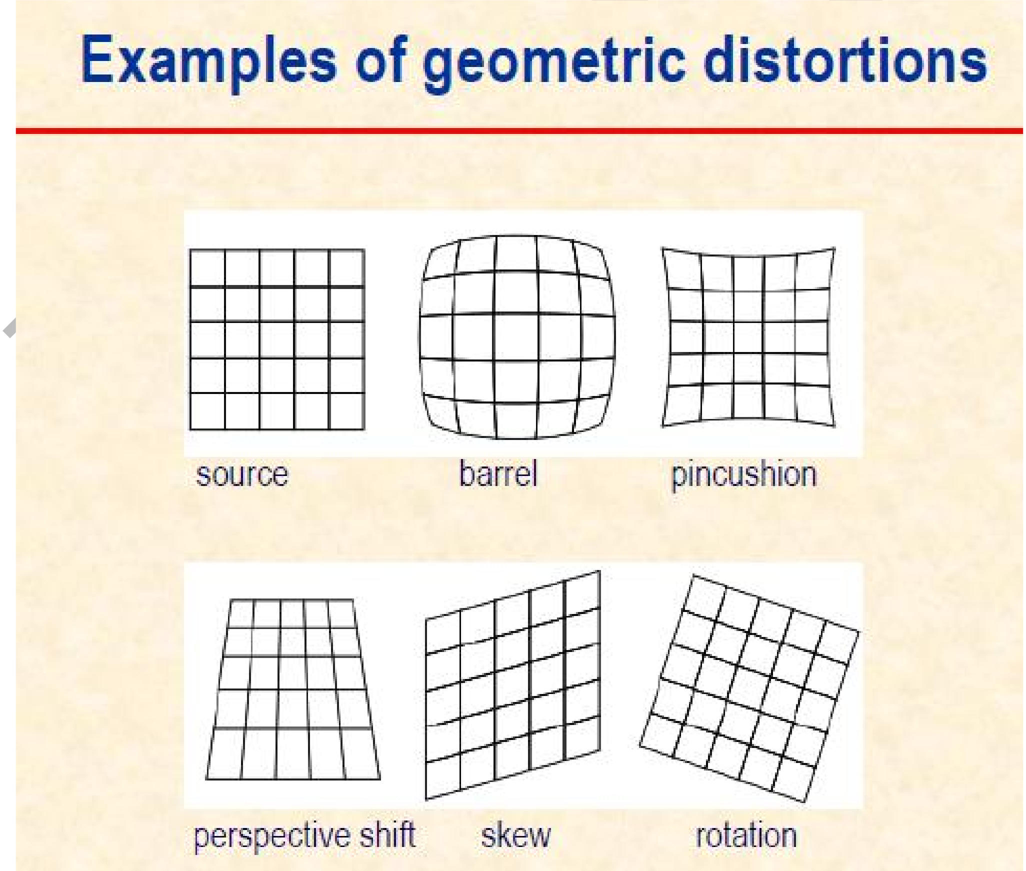
$$X' = T_x (x,y) \text{ , } y' = T_x(x,y)$$

- The transformation equations are either known in advance or can be determined from known original and transformed images.
- Several pixels in both images with known correspondence are used to derive the unknown transformation.

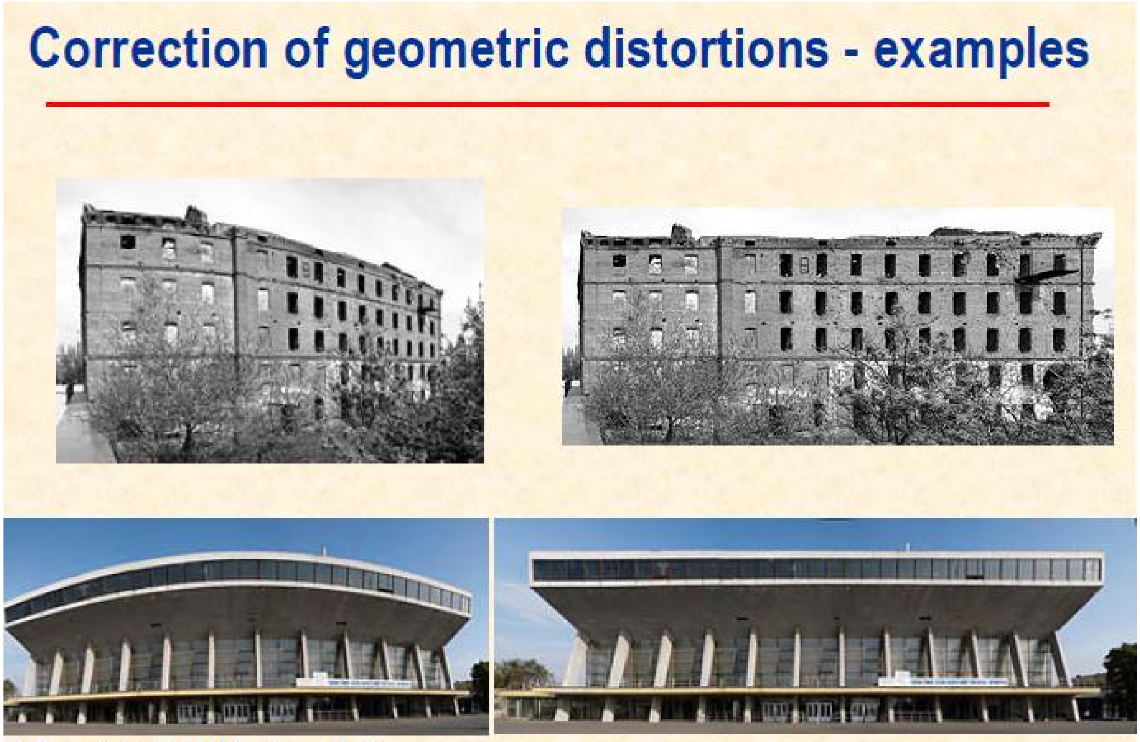
- A geometric transform consists of two basic steps ...

- 1. determining the pixel co-ordinate transformation
    - mapping of the co-ordinates of the input image pixel to the point in the output image.
        - the output point co-ordinates should be computed as continuous values (real numbers) as the position does not necessarily match the digital grid after the transform.

- 2. finding the point in the digital raster which matches the transformed point and determining its brightness.
- brightness is usually computed as an interpolation of the brightnesses of several points in the neighborhood.



Examples of geometric distortions



Correction of geometric distortions - examples

IMAGE MORPHING:

- Morphing is an image processing technique used for the metamorphosis from one image to another. The idea is to get a sequence of intermediate images which when put together with the original images would represent the change from one image to the other. The simplest method of transforming one image into another is to cross-dissolve between them.

IMAGE RESTORATION TECHNIQUES

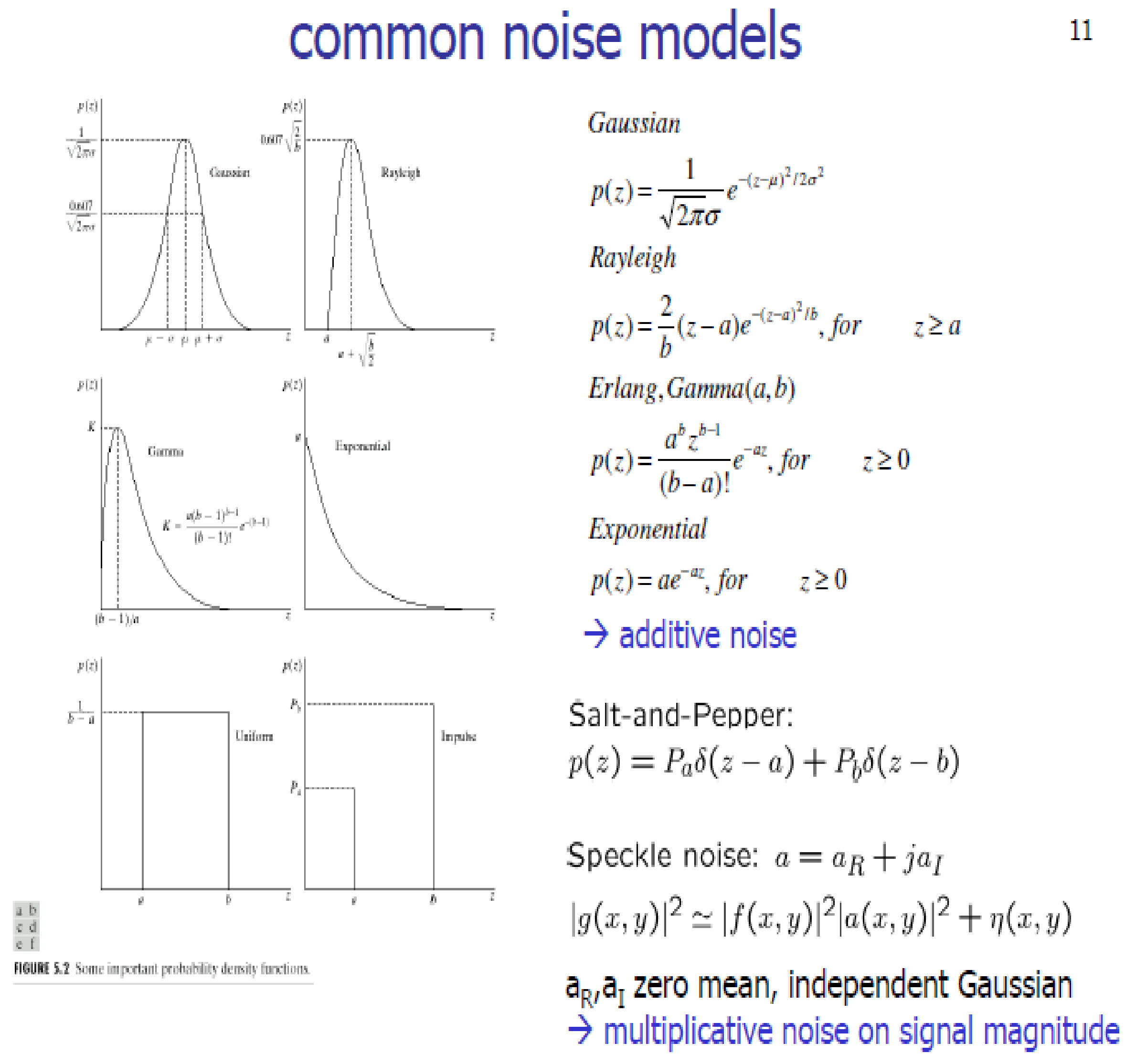
Image restoration is concerned with the reconstruction or estimation of the uncorrupted image from a blurred and noisy one. So, image restoration techniques are mainly categorized for three purposes such as

- restoration from noise,
- linear degradation and
- geometric distortion .

Usage of image restoration techniques has improved enormously since the beginning of the digital image restoration era. Now, different techniques are developed for restoration as shown in figure

NOISE MODELS

In order to restore an image we need to know about the degradation functions. Different models for the noise are described in this section. The set of noise models are defined by specific probability density functions (PDFs). Some commonly found noise models and their corresponding PDFs are given below.



common noise models

FIGURE 5.2 Some important probability density functions.

Gaussian

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

Rayleigh

$$p(z)=\frac{2}{b}(z-a)e^{-(z-a)^2/b}, for \quad z\ge a$$

Erlang, Gamma(a,b)

$$p(z)=\frac{a^b z^{b-1}}{(b-a)!}e^{-az}, for \quad z\ge 0$$

Exponential

$$p(z)=ae^{-az}, for \quad z\ge 0$$

→ additive noise

Salt-and-Pepper:

$$p(z)=P_a\delta(z-a)+P_b\delta(z-b)$$

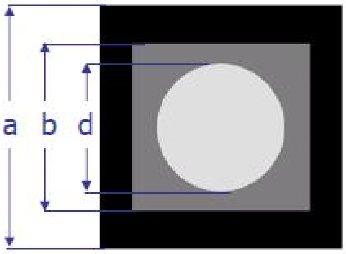
Speckle noise: $a=a_R+ja_I$

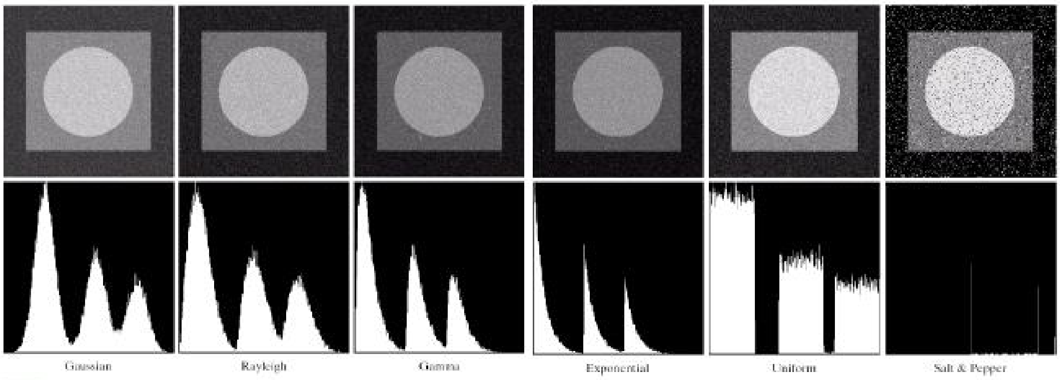
$$|g(x,y)|^2\simeq|f(x,y)|^2|a(x,y)|^2+\eta(x,y)$$

a_R, a_I zero mean, independent Gaussian

→ multiplicative noise on signal magnitude

the visual effects of noise





a b c
d e f

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

g h i
j k l

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

recovering from noise

- overall process
  Observe and estimate noise type and parameters → apply optimal (spatial) filtering (if known) → observe result, adjust filter type/parameters …

- Example noise-reduction filters [G&W 5.3]
  - Mean/median filter family
  - Adaptive filter family
  - Other filter family
    - e.g. Homomorphic filtering for multiplicative noise [G&W 4.9.6, Jain 8.13]

Periodic noise

- Arise from electrical or electromechanical interference during image acquisition
- Spatial dependence
- Observed in the frequency domain

Estimation of noise parameters

- Periodic noise
  - Observe the frequency spectrum
- Random noise with unknown PDFs
  - Case 1: imaging system is available
    - Capture images of "flat" environment
  - Case 2: noisy images available
    - Take a strip from constant area
    - Draw the histogram and observe it
    - Measure the mean and variance

Spatial filters for de-noising additive noise

- Mean filters
- Order-statistics filters
- Adaptive filters

Mean filters

- Arithmetic mean

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

Window centered at (x,y)

- Geometric mean

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

Mean filters (cont.)

- Harmonic mean filter

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

- Contra-harmonic mean filter

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

Q=-1, harmonic

Q=0, airth. mean

Q=+, ?

Order-statistics filters

- Based on the ordering(ranking) of pixels
  - Suitable for unipolar or bipolar noise (salt and pepper noise)
- Median filters
- Max/min filters
- Midpoint filters
- Alpha-trimmed mean filters

Order-statistics filters

- Median filter

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

- Max/min filters

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

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

Order-statistics filters (cont.)

- Midpoint filter

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

- Alpha-trimmed mean filter
  - Delete the d/2 lowest and d/2 highest gray-level pixels

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

Middle (mn-d) pixels

Periodic noise reduction

- Bandreject filters
- Bandpass filters
- Notch filters
- Optimum notch filtering

Bandreject filters

* Reject an isotropic frequency

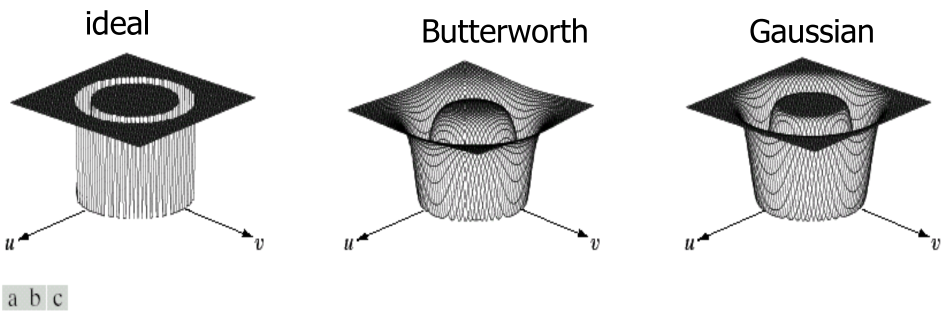
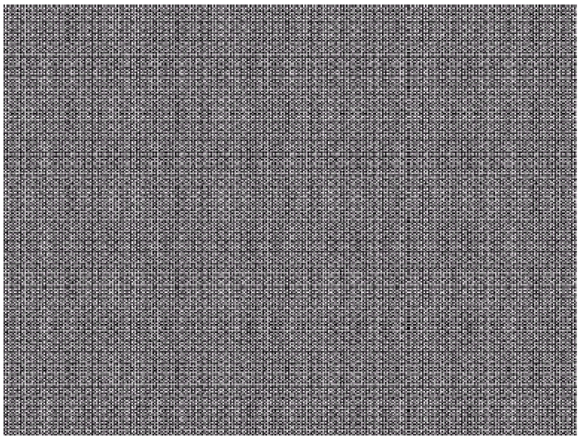


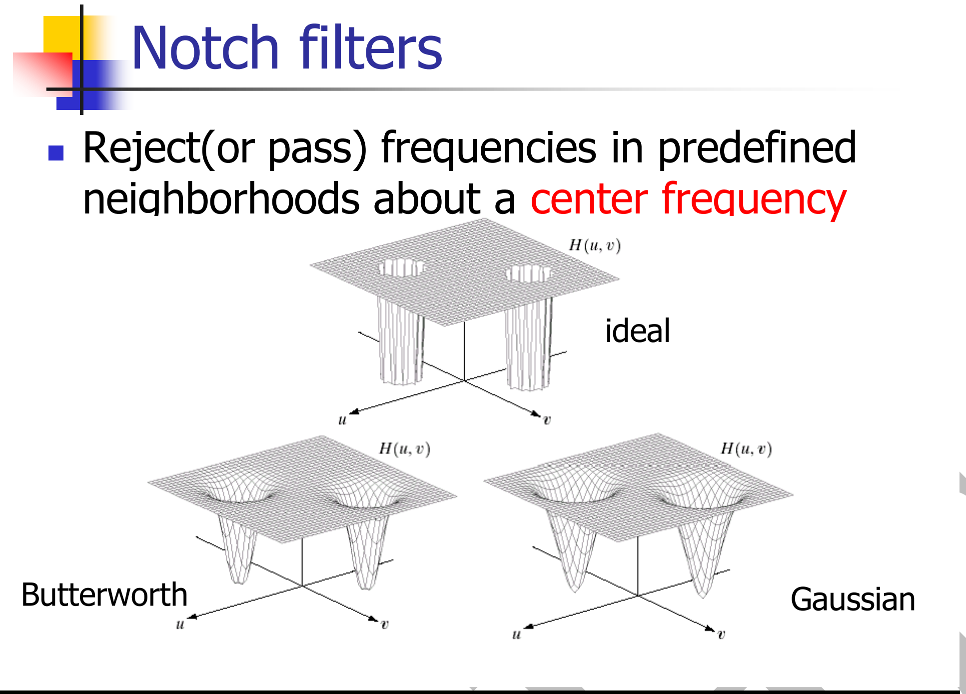
FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandpass filters

- $H_{bp}(u,v)=1- H_{br}(u,v)$



$$\Im^{-1}\left\{G(u,v)H_{bp}(u,v)\right\}$$



Notch filters

- Reject(or pass) frequencies in predefined neighborhoods about a center frequency

ADAPTIVE FILTERS

An **adaptive filter** is a system with a linear **filter** that has a transfer function controlled by variable parameters and a **means** to adjust those parameters according to an optimization algorithm.

Mean and variance are two important statistical measures on which the adaptive filtering is depends upon. For example if the local variance is high compared to the overall image variance, the filter should return a value close to the present value. Because high variance is usually associated with edges and edges should be preserved.

Noise reduction is the process of removing **noise** from a signal. All signal processing devices, both analog and digital, have traits that make them susceptible to **noise**.

Adaptive filters

- Adapted to the behavior based on the statistical characteristics of the image inside the filter region S_{xy}
- Improved performance v.s increased complexity
- Example: Adaptive local noise reduction filter

Adaptive local noise reduction filter

- Simplest statistical measurement
  - Mean and variance
- Known parameters on local region S_{xy}
  - g(x,y): noisy image pixel value
  - σ^2_η: noise variance (assume known a prior)
  - m_L : local mean
  - σ^2_L : local variance

LINEAR, POSITION INVARIANT DEGRADATIONS:

Linear, position-invariant degradation

Properties of the degradation function H

- Linear system
  - $H[af_1(x,y)+bf_2(x,y)]=aH[f_1(x,y)]+bH[f_2(x,y)]$
- Position(space)-invariant system
  - $H[f(x,y)]=g(x,y)$
  - $\Leftrightarrow H[f(x-\alpha, y-\beta)]=g(x-\alpha, y-\beta)$
- c.f. 1-D signal
  - LTI (linear time-invariant system)

Linear, Position-Invariant Degradations

The input-output relationship in Fig. 5.1 before the restoration stage is expressed as

$$g(x, y) = H[f(x, y)] + \eta(x, y). \tag{5.5-1}$$

For the moment, let us assume that $\eta(x, y) = 0$ so that $g(x, y) = H[f(x, y)]$. Based on the discussion in Section 2.6, H is *linear* if

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)] \tag{5.5-2}$$

where a and b are scalars and $f_1(x, y)$ and $f_2(x, y)$ are any two input images.

If $a = b = 1$, Eq. (5.5-2) becomes

$$H[f_1(x, y) + f_2(x, y)] = H[f_1(x, y)] + H[f_2(x, y)] \tag{5.5-3}$$

which is called the property of *additivity*. This property simply says that, if H is a linear operator, the response to a sum of two inputs is equal to the sum of the two responses.

With $f_2(x, y) = 0$, Eq. (5.5-2) becomes

$$H[af_1(x, y)] = aH[f_1(x, y)] \tag{5.5-4}$$

which is called the property of *homogeneity*. It says that the response to a constant multiple of any input is equal to the response to that input multiplied by the same constant. Thus a linear operator possesses both the property of additivity and the property of homogeneity.

An operator having the input-output relationship $g(x, y) = H[f(x, y)]$ is said to be *position* (or *space*) *invariant* if

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta) \tag{5.5-5}$$

for any $f(x, y)$ and any α and β. This definition indicates that the response at any point in the image depends only on the *value* of the input at that point, not on its *position*.

With a slight (but equivalent) change in notation in the definition of the discrete impulse function in Eq. (4.2-33), $f(x, y)$ can be expressed in terms of a continuous impulse function:

$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)\, d\alpha\, d\beta. \tag{5.5-6}$$

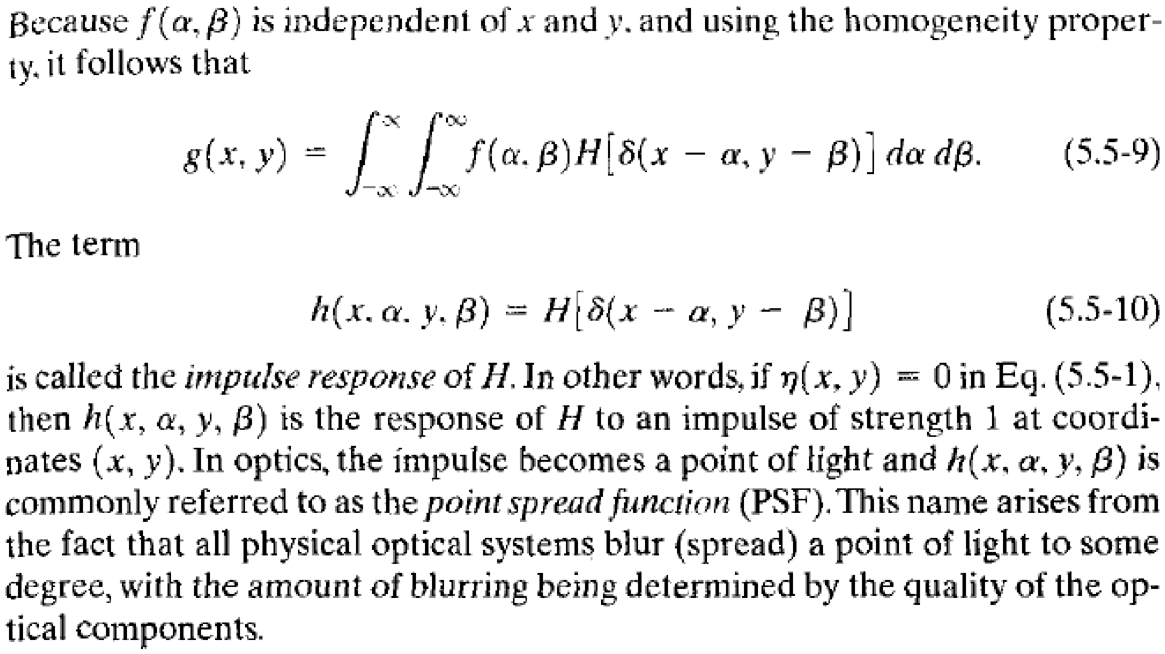
This, in fact, is the *definition* using continuous variables of a unit impulse located at coordinates (x, y).

Assume again for a moment that $\eta(x, y) = 0$. Then, substitution of Eq. (5.5-6) into Eq. (5.5-1) results in the expression

$$g(x, y) = H[f(x, y)] = H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)\delta(x - \alpha, y - \beta)\, d\alpha\, d\beta\right]. \tag{5.5-7}$$

If H is a linear operator and we extend the additivity property to integrals, then

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H[f(\alpha, \beta)\delta(x - \alpha, y - \beta)]\, d\alpha\, d\beta. \tag{5.5-8}$$



Because $f(\alpha, \beta)$ is independent of x and y, and using the homogeneity property, it follows that

$$g(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)H\left[\delta(x - \alpha, y - \beta)\right] d\alpha\, d\beta. \qquad (5.5\text{-}9)$$

The term

$$h(x, \alpha, y, \beta) = H\left[\delta(x - \alpha, y - \beta)\right] \qquad (5.5\text{-}10)$$

is called the *impulse response* of H. In other words, if $\eta(x, y) = 0$ in Eq. (5.5-1), then $h(x, \alpha, y, \beta)$ is the response of H to an impulse of strength 1 at coordinates (x, y). In optics, the impulse becomes a point of light and $h(x, \alpha, y, \beta)$ is commonly referred to as the *point spread function* (PSF). This name arises from the fact that all physical optical systems blur (spread) a point of light to some degree, with the amount of blurring being determined by the quality of the optical components.

ESTIMATING THE DEGRADATION FUNCTION

The process of
restoring an image by using a degradation function that has been estimated in some way sometimes is called *blind deconvolution,*

- **Estimation by Image observation**
- **Estimation by experimentation**
- **Estimation by modeling**

Estimation by Image Observation

Suppose that we are given a degraded image without any knowledge about the degradation function H. One way to estimate this function is to gather information from the image itself. For example, if the image is blurred, we can look at a small section of the image containing simple structures, like part of an object and the background. In order to reduce the effect of noise in our observation, we

would look for areas of strong signal content. Using sample gray levels of the object and background, we can construct an unblurred image of the same size and characteristics as the observed subimage. Let the observed subimage be denoted by gJx, *y).* and let the constructed subimage which in reality is our estimate of the original image in that area) be denoted by *I,(x, y).* Then, assuming that the effect of noise is negligible because of our choice of a strong-signal area, it follows from Eq. (5.5-17) that

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)}. \tag{5.6-1}$$

From the characteristics of this function we then deduce the complete function $H(u, v)$ by making use of the fact that we are assuming position invariance. For example, suppose that a radial plot of $H_s(u, v)$ turns out to have the shape of Butterworth lowpass filter. We can use that information to construct a function $H(u, v)$ on a larger scale, but having the same shape.

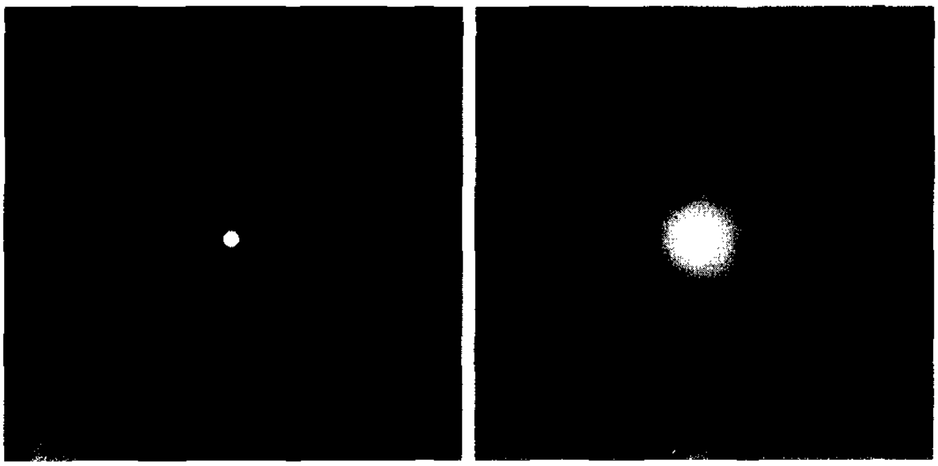
5.6.2 Estimation by Experimentation

If equipment similar to the equipment used to acquire the degraded image is available, it is possible in principle to obtain an accurate estimate of the degradation. Images similar to the degraded image can be acquired with various system settings until they are degraded as closely as possible to the image we wish to restore. Then the idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings. As noted in Section 5.5, a linear, space-invariant system is described completely by its impulse response.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise. Then, recalling that the Fourier transform of an impulse is a constant, it follows from Eq. (5.5-17) that

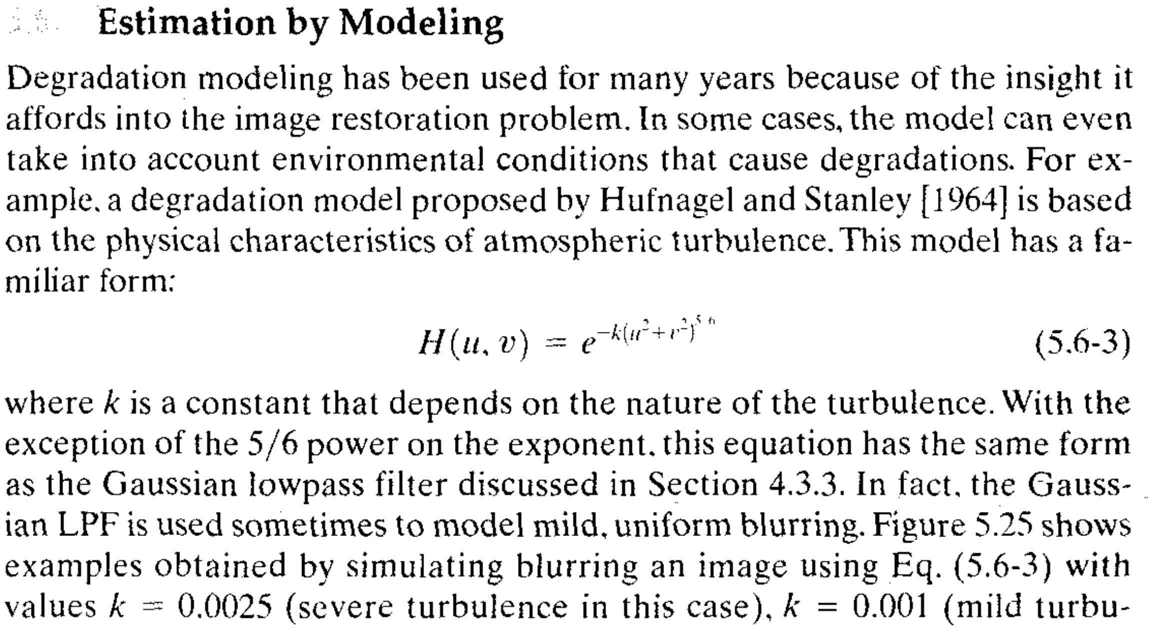
$$H(u, v) = \frac{G(u, v)}{A} \tag{5.6-2}$$

where, as before, $G(u, v)$ is the Fourier transform of the observed image and A is a constant describing the strength of the impulse. Figure 5.24 shows an example.



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

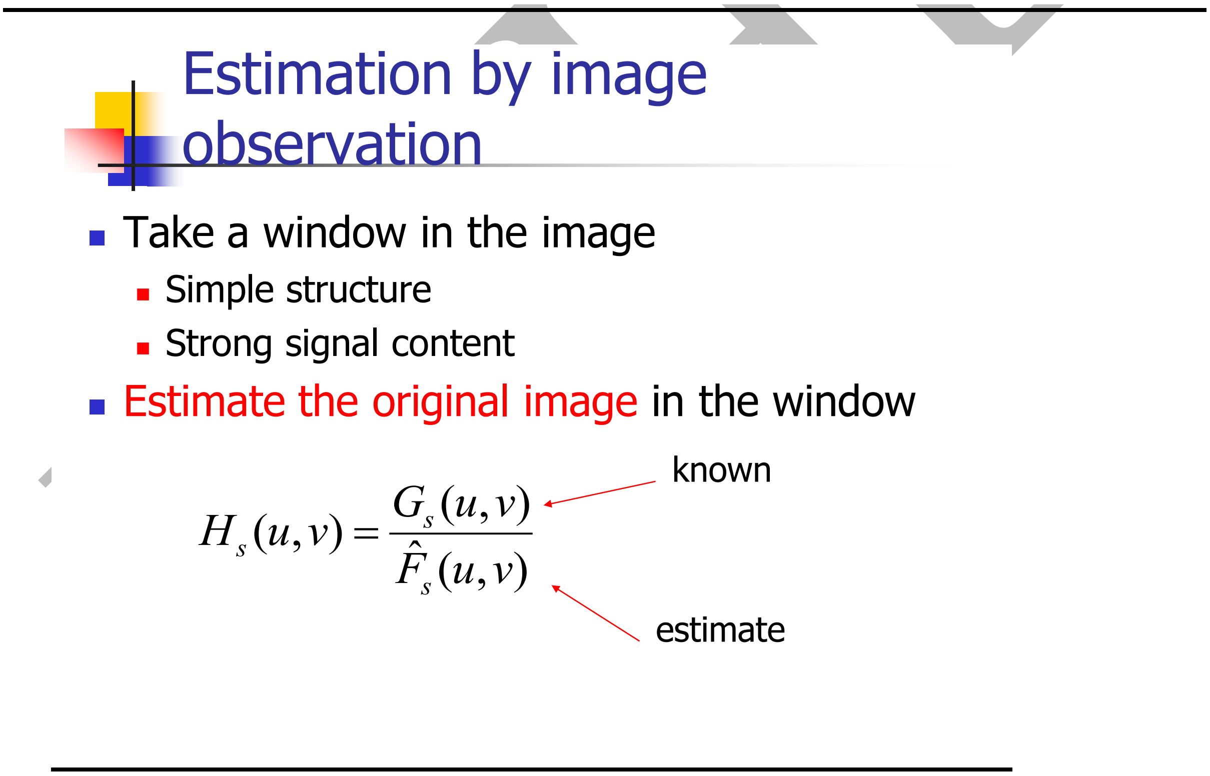


Estimation by Modeling

Degradation modeling has been used for many years because of the insight it affords into the image restoration problem. In some cases, the model can even take into account environmental conditions that cause degradations. For example, a degradation model proposed by Hufnagel and Stanley [1964] is based on the physical characteristics of atmospheric turbulence. This model has a familiar form:

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}} \qquad (5.6\text{-}3)$$

where k is a constant that depends on the nature of the turbulence. With the exception of the 5/6 power on the exponent, this equation has the same form as the Gaussian lowpass filter discussed in Section 4.3.3. In fact, the Gaussian LPF is used sometimes to model mild, uniform blurring. Figure 5.25 shows examples obtained by simulating blurring an image using Eq. (5.6-3) with values $k = 0.0025$ (severe turbulence in this case), $k = 0.001$ (mild turbu-



Estimation by image observation

- Take a window in the image
  - Simple structure
  - Strong signal content
- Estimate the original image in the window

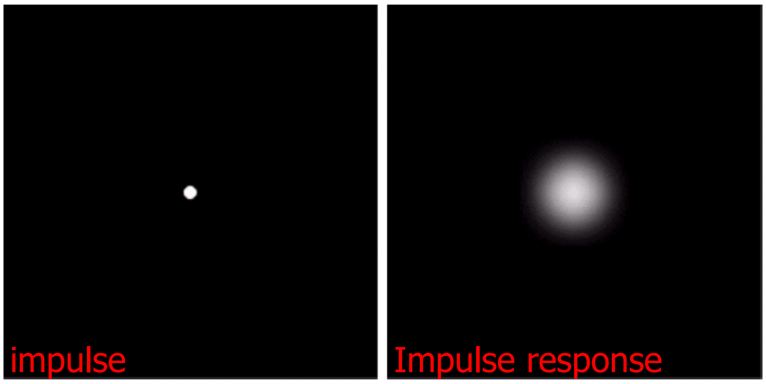
$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

known

estimate

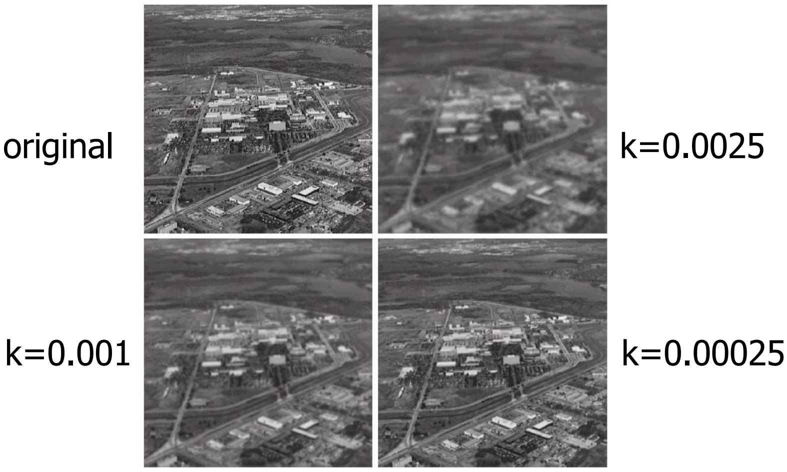
Estimation by experimentation

- If the image acquisition system is ready
- Obtain the impulse response



Estimation by modeling (1)

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



Estimation by modeling (2)

- Derive a mathematical model
- Ex. Motion of image

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

Planar motion

Fourier transform

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

Inverse filtering

- With the estimated degradation function H(u,v)

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

Unknown noise

$$=> \quad \hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}=F(u,v)+\frac{N(u,v)}{H(u,v)}$$

Estimate of original image

Problem: 0 or small values

Sol: limit the frequency around the origin

IMAGE COMPRESSION:

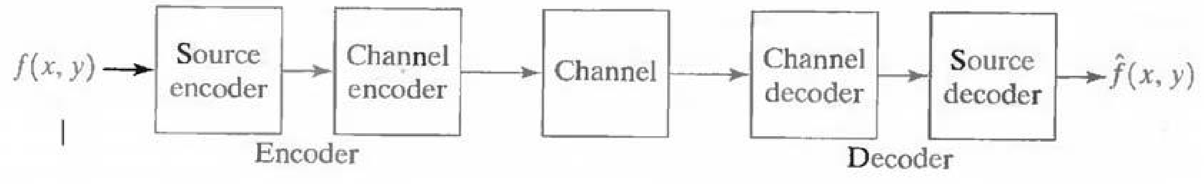
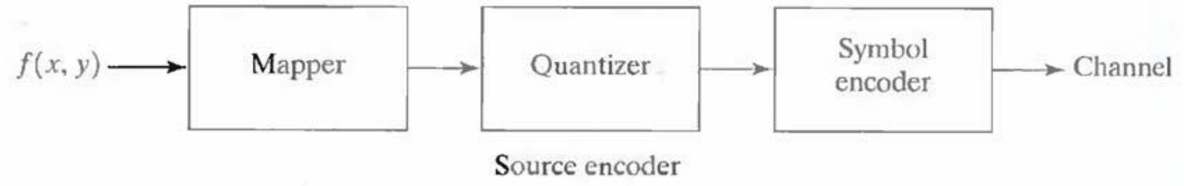


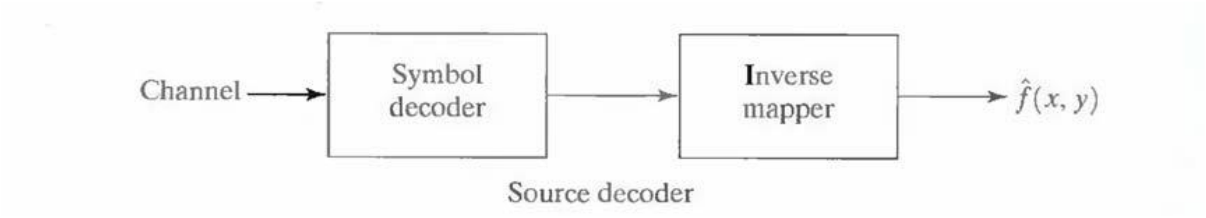
Fig.3.1 A general compression system model

Fig. shows, a compression system consists of two distinct structural blocks: an encoder and a decoder. An input image f(x, y) is fed into the encoder, which creates a set of symbols from the input data. After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image f^(x, y) is generated. In general, f^(x, y) may or may not be an exact replica of f(x, y). If it is, the system is error free or information preserving; if not, some level of distortion is present in the reconstructed image. Both the encoder and decoder shown in Fig. 3.1 consist of two relatively independent functions or subblocks. The encoder is made up of a source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder's output. As would be expected, the decoder includes a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free (not prone to error), the channel encoder and decoder are omitted, and the general encoder and decoder become the source encoder and decoder, respectively.

THE SOURCE ENCODER AND DECODER:

The source encoder is responsible for reducing or eliminating any coding, interpixel, or psychovisual redundancies in the input image. The specific application and associated fidelity requirements dictate the best encoding approach to use in any given situation. Normally, the approach can be modeled by a series of three independent operations. As Fig. 3.2 (a) shows, each operation is designed to reduce one of the three redundancies. Figure 3.2 (b) depicts the corresponding source decoder. In the first stage of the source encoding process, the mapper transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image.





a
b

Fig.3.2 (a) Source encoder and (b) source decoder model

Run-length coding is an example of a mapping that directly results in data compression in this initial stage of the overall source encoding process. The representation of an image by a set of transform coefficients is an example of the opposite case. Here, the mapper transforms the image into an array of coefficients, making its interpixel redundancies more accessible for compression in later stages of the encoding process.

The second stage, or quantizer block in Fig. 3.2 (a), reduces the accuracy of the mapper's output in accordance with some preestablished fidelity criterion. This stage reduces the psychovisual redundancies of the input image. This operation is irreversible. Thus it must be omitted when error-free compression is desired.

In the third and final stage of the source encoding process, the symbol coder creates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code. The term symbol coder distinguishes this coding operation from the overall source encoding process. In most cases, a variable-length code is used to represent the mapped and quantized data set.

It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy. The operation, of course, is reversible. Upon completion of the symbol coding step, the input image has been processed to remove each of the three redundancies.

Figure 3.2(a) shows the source encoding process as three successive operations, but all three operations are not necessarily included in every compression system. Recall, for example, that the quantizer must be omitted when error-free compression is desired. In addition, some compression techniques normally are modeled by merging blocks that are physically separate in Fig. 3.2(a).

In the predictive compression systems, for instance, the mapper and quantizer are often represented by a single block, which simultaneously performs both operations. The source decoder shown in Fig. 3.2(b) contains only two components: a symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model shown in Fig. 3.2(b).

Data Redundancy

The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. A clear distinction must be made between data and information. They are not synonymous. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information. Such might be the case, for example, if a long-winded individual and someone who is short and to the point were to relate the same story.

Here, the information of interest is the story; words are the data used to relate the information. If the two individuals use a different number of words to tell the same basic story, two different versions of the story are created, and at least one includes nonessential data. That is, it contains data (or words) that either provide no relevant information or simply restate that which is already known. It is thus said to contain data redundancy.

Data redundancy is a central issue in digital image compression. It is not an abstract concept but a mathematically quantifiable entity. If n1 and n2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy RD of the first data set (the one characterized by n1) can be defined as

$$R_D = 1 - \frac{1}{C_R}$$

where CR , commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2}.$$

For the case n2 = n1, CR = 1 and RD = 0, indicating that (relative to the second data set) the first
representation of the information contains no redundant data. When n2 << n1, CR -> ∞ and RD->1, implying significant compression and highly redundant data. Finally, when n2 >> n1
,

CR-> 0 and RD ->∞, indicating that the second data set contains much more data than the original representation. This, of course, is the normally undesirable case of data expansion. In general, CR and RD lie in the open intervals (0,∞) and (-∞, 1), respectively. A practical compression ratio, such as 10 (or 10:1), means that the first data set has 10 information carrying

units (say, bits) for every 1 unit in the second or compressed data set. The corresponding redundancy of 0.9 implies that 90% of the data in the first data set is redundant.

TYPES OF REDUNDANCIES

In digital image compression, three basic data redundancies can be identified and exploited: **coding redundancy**, **interpixel redundancy**, and **psychovisual redundancy**. Data compression is achieved when one or more of these redundancies are reduced or eliminated.

Coding Redundancy:

In this, we utilize formulation to show how the gray-level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it. Let us assume, once again, that a discrete random variable rk in the interval [0, 1] represents the gray levels of an image and that each rk occurs with probability pr (rk).

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

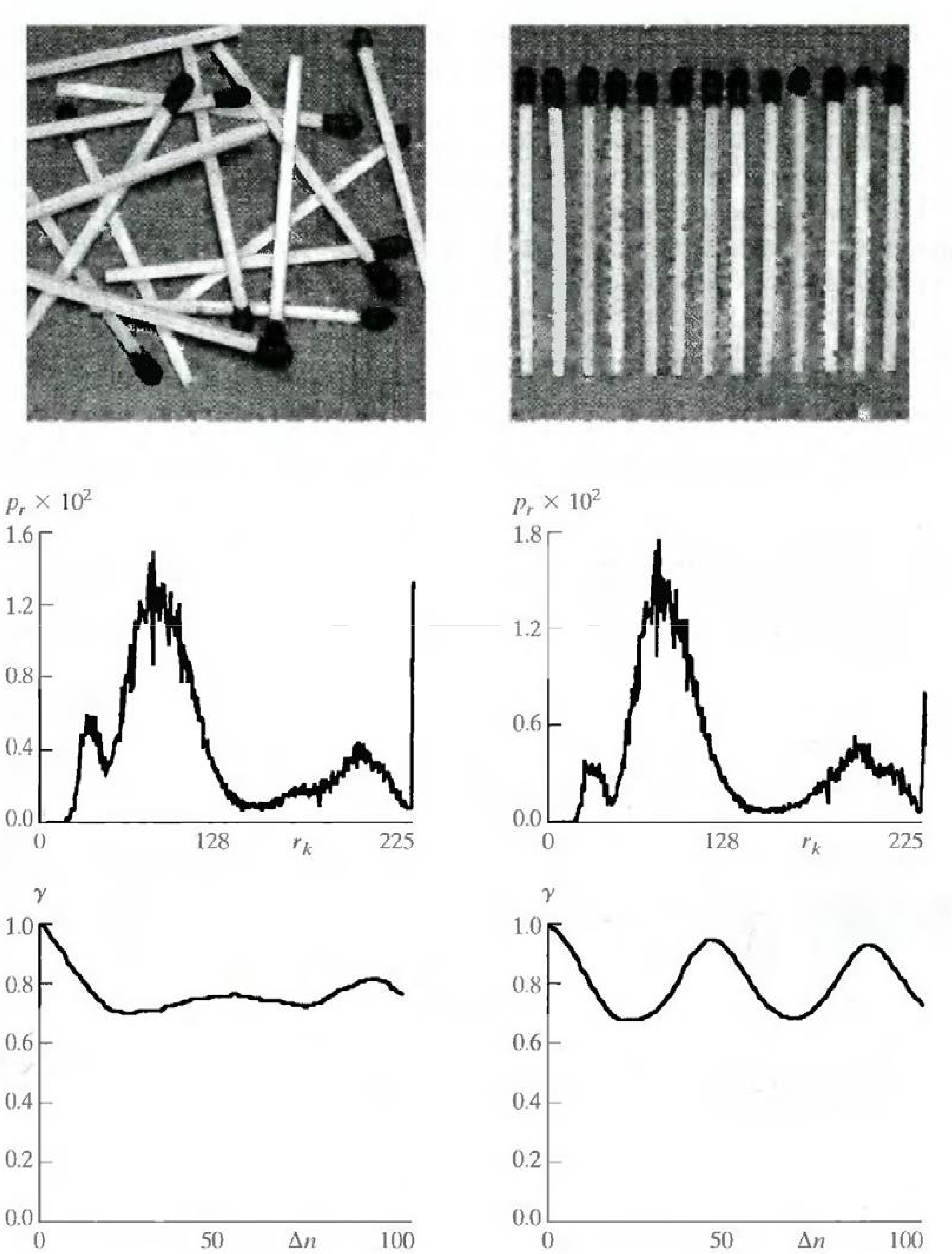
where L is the number of gray levels, nk is the number of times that the kth gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of rk is l (rk), then the average number of bits required to represent each pixel is

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k)p_r(r_k).$$

That is, the average length of the code words assigned to the various gray-level values is found by summing the product of the number of bits used to represent each gray level and the probability that the gray level occurs. Thus the total number of bits required to code an M X N image is MNLavg.

Interpixel Redundancy:

Consider the images shown in Figs. 1.1(a) and (b). As Figs. 1.1(c) and (d) show, these images have virtually identical histograms. Note also that both histograms are trimodal, indicating the presence of three dominant ranges of gray-level values. Because the gray levels in these images are not equally probable, variable-length coding can be used to reduce the coding redundancy that would result from a straight or natural binary encoding of their pixels. The coding process, however, would not alter the level of correlation between the pixels within the images. In other words, the codes used to represent the gray levels of each image have nothing to do with the correlation between pixels. These correlations result from the structural or geometric relationships between the objects in the image.



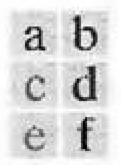


Fig.1.1 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

Figures 1.1(e) and (f) show the respective autocorrelation coefficients computed along one line of each image.

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}$$

Where

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n).$$

The scaling factor in Eq. above accounts for the varying number of sum terms that arise for each integer value of Δn. Of course, Δn must be strictly less than N, the number of pixels on a line. The variable x is the coordinate of the line used in the computation. Note the dramatic difference between the shape of the functions shown in Figs. 1.1(e) and (f). Their shapes can be qualitatively related to the structure in the images in Figs. 1.1(a) and (b).This relationship is particularly noticeable in Fig. 1.1 (f), where the high correlation between pixels separated by 45 and 90 samples can be directly related to the spacing between the vertically oriented matches of Fig. 1.1(b). In addition, the adjacent pixels of both images are highly correlated. When Δn is 1, γ is 0.9922 and 0.9928 for the images of Figs. 1.1 (a) and (b), respectively. These values are typical of most properly sampled television images.

These illustrations reflect another important form of data redundancy—one directly related to the interpixel correlations within an image. Because the value of any given pixel can be reasonably predicted from the value of its neighbors, the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names, including spatial redundancy, geometric redundancy, and interframe redundancy, have been coined to refer to these interpixel dependencies. We use the term interpixel redundancy to encompass them all.

In order to reduce the interpixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "nonvisual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove interpixel redundancy) are referred to as mappings. They are called reversible mappings if the original image elements can be reconstructed from the transformed data set.

Psychovisual Redundancy:

The brightness of a region, as perceived by the eye, depends on factors other than simply the light reflected by the region. For example, intensity variations (Mach bands) can be perceived in an area of constant intensity. Such phenomena result from the fact that the eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psychovisually redundant. It can be eliminated without significantly impairing the quality of image perception.

That psychovisual redundancies exist should not come as a surprise, because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings.

The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Psychovisual redundancy is fundamentally different from the redundancies discussed earlier. Unlike coding and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as quantization.

This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. As it is an irreversible operation (visual information is lost), quantization results in lossy data compression.

LOSSY AND LOSSLESS COMPRESSION:

Data compression implies sending or storing a smaller number of bits. Although many methods are used for this purpose, in general these methods can be divided into two broad categories: **lossless** and **lossy** methods.

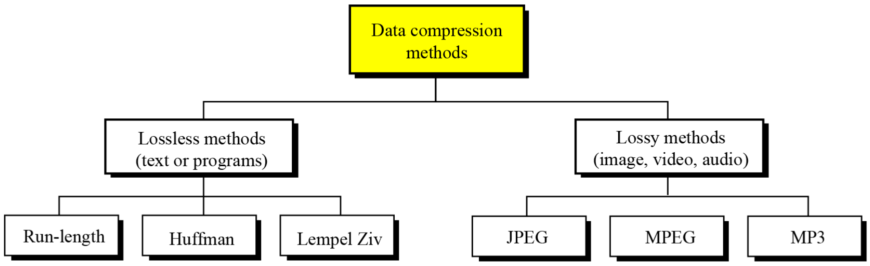


Figure 15.1 **Data compression methods**

15.3

LOSSLESS COMPRESSION

- In **lossless** data compression, the integrity of the data is preserved.
- The original data and the data after compression and decompression are exactly the same because, in these methods, the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process.
- Redundant data is removed in compression and added during decompression. Lossless compression methods are normally used when we cannot afford to lose any data.

LOSSY COMPRESSION METHODS

- Our eyes and ears cannot distinguish subtle changes. In such cases, we can use a lossy data compression method.
- These methods are cheaper—they take less time and space when it comes to sending millions of bits per second for images and video.
- Several methods have been developed using lossy compression techniques. **JPEG (Joint Photographic Experts Group)** encoding is used to compress pictures and graphics, **MPEG (Moving Picture Experts Group)** encoding is used to compress video, and **MP3 (MPEG audio layer 3)** for audio compression.

ENTROPY OF AN INFORMATION SOURCE:

Information entropy is a concept from information theory. It tells how much information there is in an event. In general, the more uncertain or random the event is, the more information it will contain. More clearly stated, information is a decrease in uncertainty or entropy. The concept of information entropy was created by mathematician Claude Shannon.

Information and its relationship to entropy can be modeled by:

$$R = H(x) - Hy(x)$$

"The conditional entropy Hy(x) will, for convenience, be called the equivocation. It measures the average ambiguity of the received signal."[1]

The "average ambiguity" or Hy(x) meaning uncertainty or entropy. H(x) represents information. R is the received signal.

It has applications in many areas, including lossless data compression, statistical inference, cryptography, and sometimes in other disciplines as biology, physics or machine learning.

The **information gain** is a measure of the probability with which a certain result is expected to happen. In the context of a coin flip, with a 50-50 probability, the entropy is the highest value of 1. It does not involve information gain because it does not incline towards a specific result more than the other. If there is a 100-0 probability that a result will occur, the entropy is 0.

SHANNON'S 1ST THEOREM:

The noiseless coding theorem

When both the information channel and communication system are error free, the principal function of the communication system is to represent the source as compactly as possible. Under these circumstances, the *noiseless coding theorem*, also called *Shannon's first theorem* (Shannon [1948]), defines the minimum average code word length per source symbol that can be achieved.

A source of information with finite ensemble $(A, \mathbf{z})$ and statistically independent source symbols is called a *zero-memory* source. If we consider its output to be an n-tuple of symbols from the source alphabet (rather than a single symbol), the source output then takes on one of J^n possible values, denoted α_i, from the set of all possible n element sequences $A' = \{\alpha_1, \alpha_2, \ldots, \alpha_{J^n}\}$. In other words, each α_i (called a *block random variable*) is composed of n symbols from A. (The notation A' distinguishes the set of block symbols from A, the set of single symbols.) The probability of a given α_i is $P(\alpha_i)$, ...which is related to the single-symbol probabilities $P(a_j)$ by

$$P(\alpha_i) = P(a_{j1})P(a_{j2})\cdots P(a_{jn}) \tag{8.3-14}$$

where subscripts $j1, j2, \ldots, jn$ are used to index the n symbols from A that make up an α_i. As before, the vector $\mathbf{z}'$ (the prime is added to indicate the use of the block random variable) denotes the set of all source probabilities $\{P(\alpha_1), P(\alpha_2), \ldots, P(\alpha_{J^n})\}$, and the entropy of the source is

$$H(\mathbf{z}') = -\sum_{i=1}^{J^n} P(\alpha_i) \log P(\alpha_i).$$

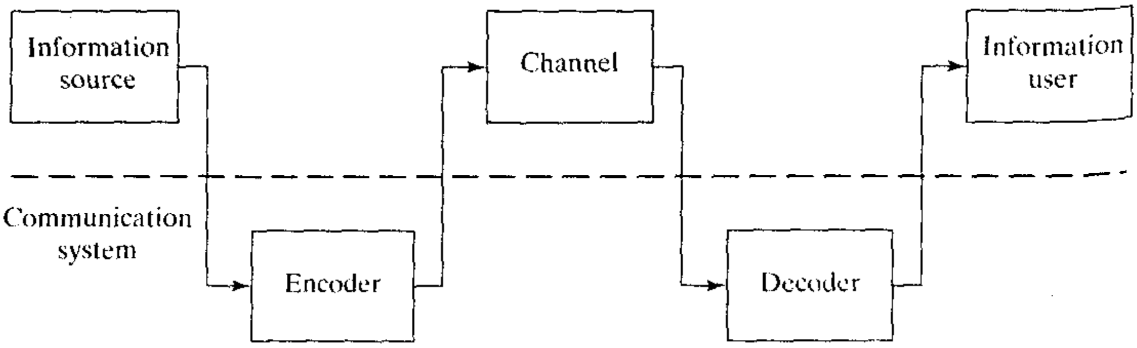


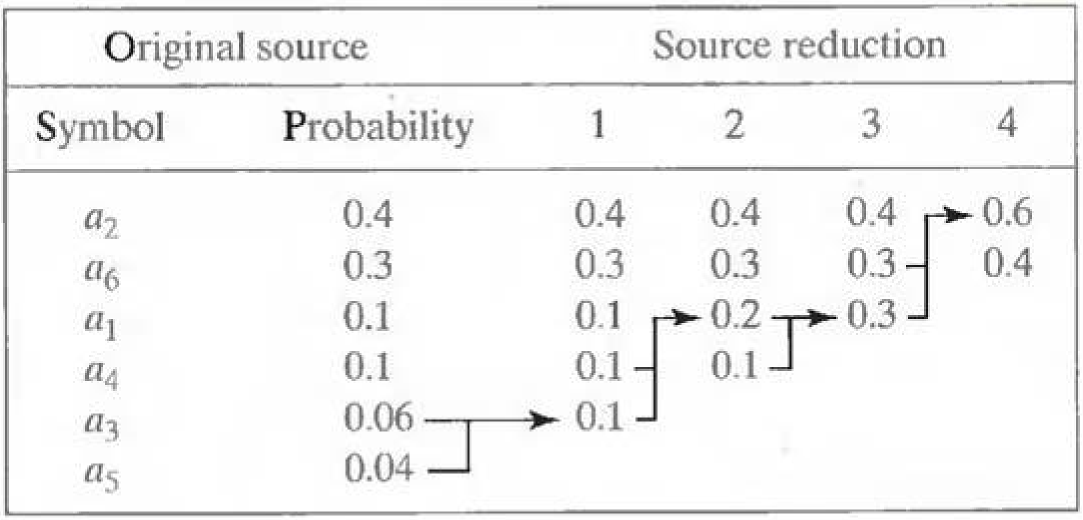
FIGURE 8.9 A communication system model.

HUFFMAN CODING:

The most popular technique for removing coding redundancy is due to Huffman (Huffman [1952]). When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of the noiseless coding theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time. The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction.

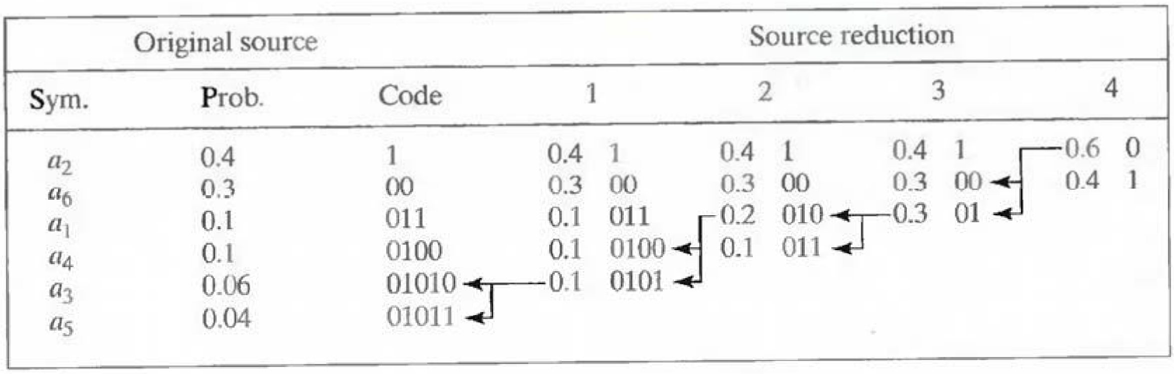
Figure 4.1 illustrates this process for binary coding (K-ary Huffman codes can also be constructed). At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values. To form the first source reduction, the bottom two probabilities, 0.06 and 0.04, are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This process is then repeated until a reduced source with two symbols (at the far right) is reached.

The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As Fig. 4.2 shows, these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrarily



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				

Fig.4.1 Huffman source reductions.



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								

Fig.4.2 Huffman code assignment procedure.

appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final code appears at the far left in Fig. 4.2. The average length of this code is

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

and the entropy of the source is 2.14 bits/symbol. The resulting Huffman code efficiency is 0.973.

Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time. After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. For the binary code of Fig. 4.2, a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a3 .The next valid code is 011, which corresponds to symbol a1. Continuing in this manner reveals the completely decoded message to be a3a1a2a2a6.

ARITHMETIC CODING:

Unlike the variable-length codes described previously, arithmetic coding generates nonblock codes. In arithmetic coding, which can be traced to the work of Elias, a one-to-one correspondence between source symbols and code words does not exist. Instead, an entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units (say, bits) required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. Because the technique does not require, as does Huffman's approach, that each source symbol translate into an integral number of code symbols (that is, that the symbols be coded one at a time), it achieves (but only in theory) the bound established by the noiseless coding theorem.

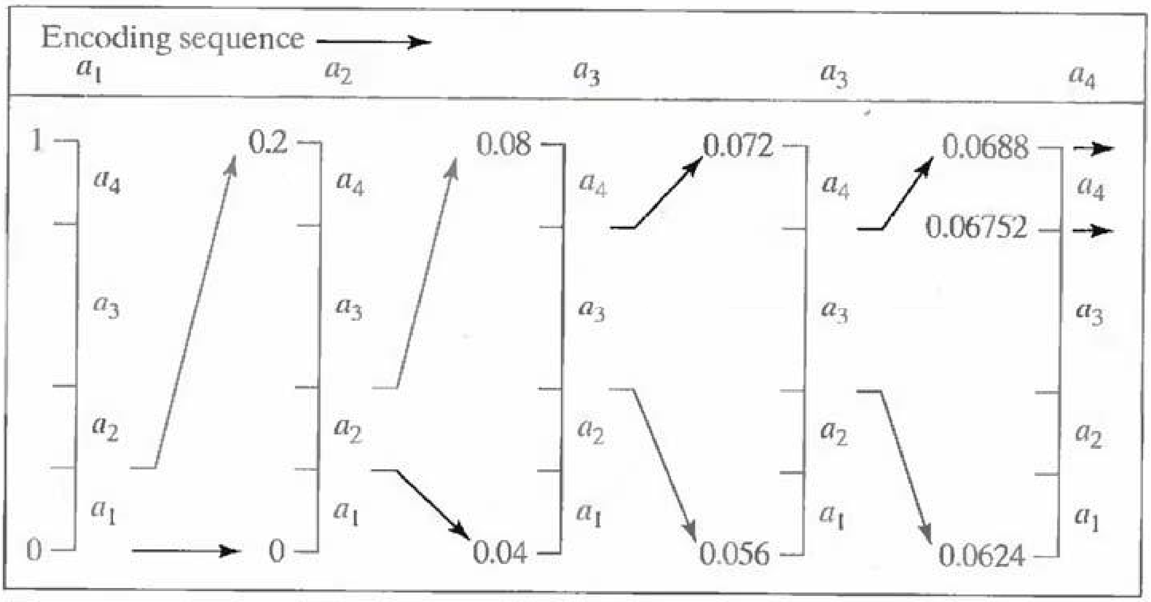
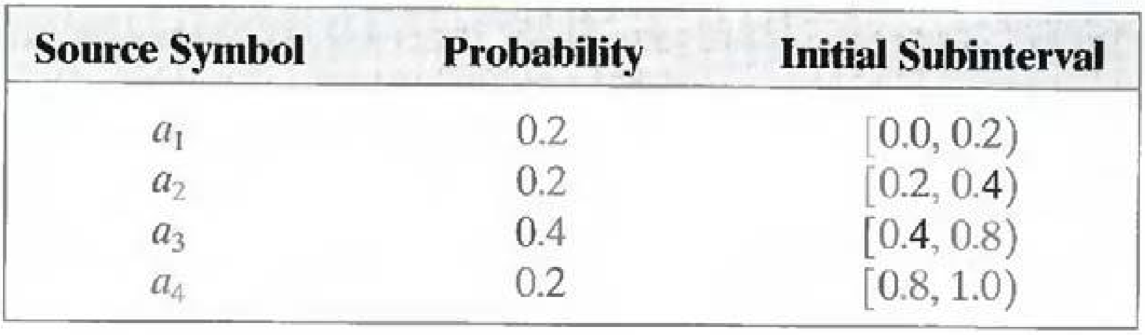


Fig.5.1 Arithmetic coding procedure

Figure 5.1 illustrates the basic arithmetic coding process. Here, a five-symbol sequence or message, a1a2a3a3a4, from a four-symbol source is coded. At the start of the coding process, the message is assumed to occupy the entire half-open interval [0, 1). As Table 5.2 shows, this interval is initially subdivided into four regions based on the probabilities of each source symbol. Symbol ax, for example, is associated with subinterval [0, 0.2). Because it is the first symbol of the message being coded, the message interval is initially narrowed to [0, 0.2). Thus in Fig. 5.1 [0, 0.2) is expanded to the full height of the figure and its end points labeled by the values of the narrowed range. The narrowed range is then subdivided in accordance with the original source symbol probabilities and the process continues with the next message symbol.



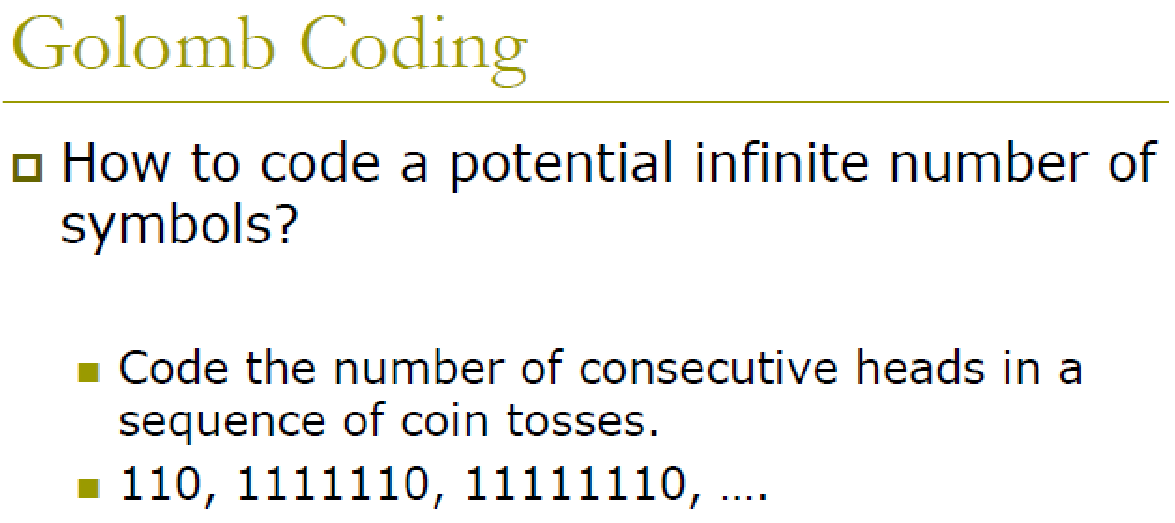
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Table 5.1 Arithmetic coding example

In this manner, symbol a2 narrows the subinterval to [0.04, 0.08), a3 further narrows it to [0.056, 0.072), and so on. The final message symbol, which must be reserved as a special end-offor message indicator, narrows the range to [0.06752, 0.0688). Of course, any number within this subinterval—for example, 0.068—can be used to represent the message. In the arithmetically coded message of Fig. 5.1, three decimal digits are used to represent the five-symbol message.

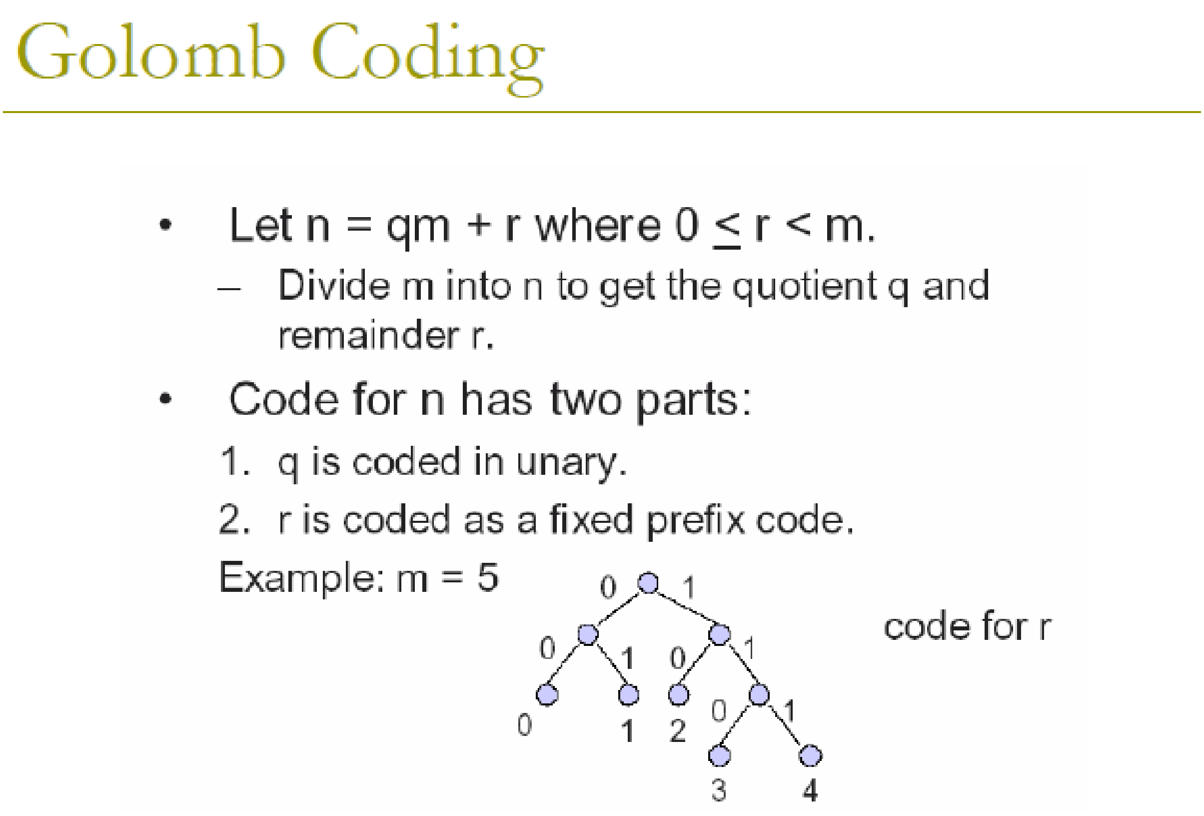
This translates into 3/5 or 0.6 decimal digits per source symbol and compares favorably with the entropy of the source, which is 0.58 decimal digits or 10-ary units/symbol. As the length of the sequence being coded increases, the resulting arithmetic code approaches the bound established by the noiseless coding theorem.

In practice, two factors cause coding performance to fall short of the bound: (1) the addition of the end-of-message indicator that is needed to separate one message from another; and (2) the use of finite precision arithmetic. Practical implementations of arithmetic coding address the latter problem by introducing a scaling strategy and a rounding strategy (Langdon and Rissanen [1981]). The scaling strategy renormalizes each subinterval to the [0, 1) range before subdividing it in accordance with the symbol probabilities. The rounding strategy guarantees that the truncations associated with finite precision arithmetic do not prevent the coding subintervals from being represented accurately.



Golomb Coding

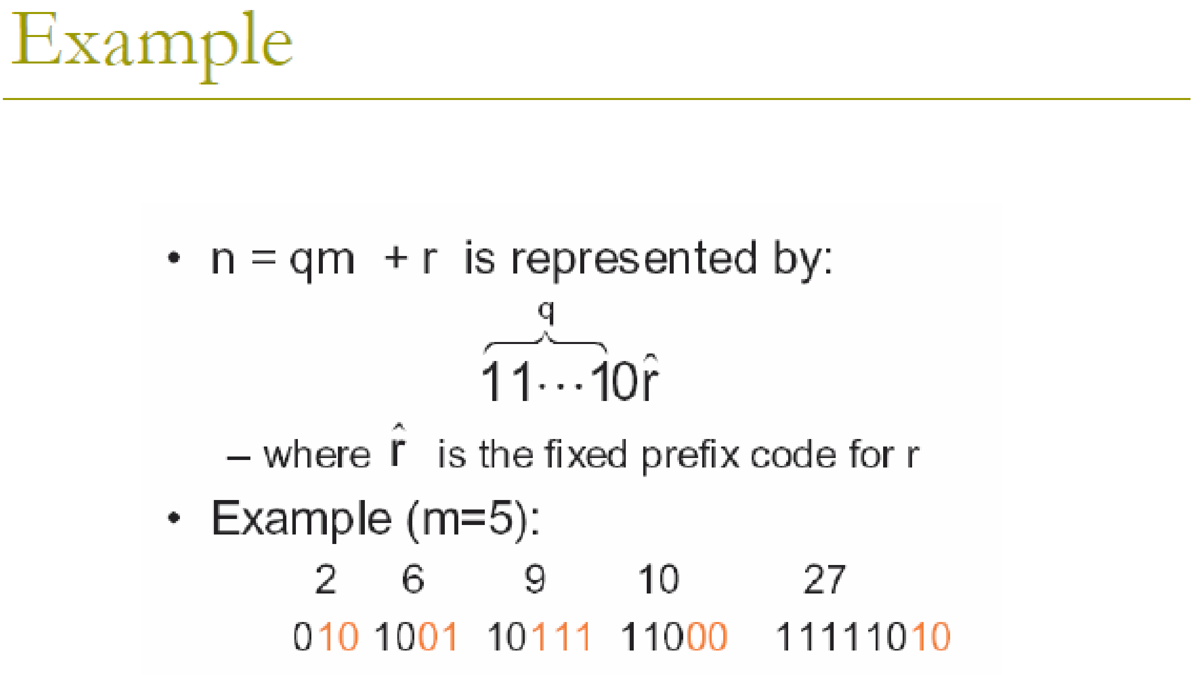
- How to code a potential infinite number of symbols?
  - Code the number of consecutive heads in a sequence of coin tosses.
  - 110, 1111110, 11111110, ...



Golomb Coding

- Let $n = qm + r$ where $0 \leq r < m$.
  - Divide m into n to get the quotient q and remainder r.
- Code for n has two parts:
  1. q is coded in unary.
  2. r is coded as a fixed prefix code.

Example: m = 5



Example

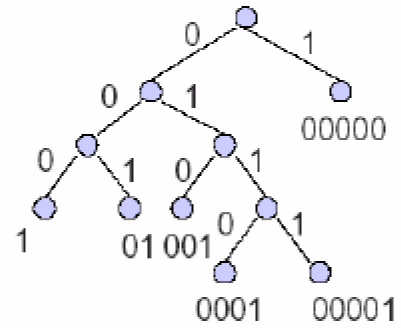
- $n = qm + r$ is represented by:

$$\overbrace{11\cdots1}^{q}0\hat{r}$$

  - where $\hat{r}$ is the fixed prefix code for r
- Example (m=5):

2	6	9	10	27
010	1001	10111	11000	11111010

Another Way of Looking at Golomb Code (m=5)



input	output
00000	1
00001	0111
0001	0110
001	010
01	001
1	000

LZW CODING:

Lempel Ziv (LZ) encoding is an example of a category of algorithms called *dictionary-based* encoding. The idea is to create a dictionary (a table) of strings used during the communication session. If both the sender and the receiver have a copy of the dictionary, then previously-encountered strings can be substituted by their index in the dictionary to reduce the amount of information transmitted.

In this phase there are two concurrent events: building an indexed dictionary and compressing a string of symbols. The algorithm extracts the smallest substring that cannot be found in the dictionary from the remaining uncompressed string. It then stores a copy of this substring in the dictionary as a new entry and assigns it an index value. Compression occurs when the substring, except for the last character, is replaced with the index found in the dictionary. The process then inserts the index and the last character of the substring into the compressed string.

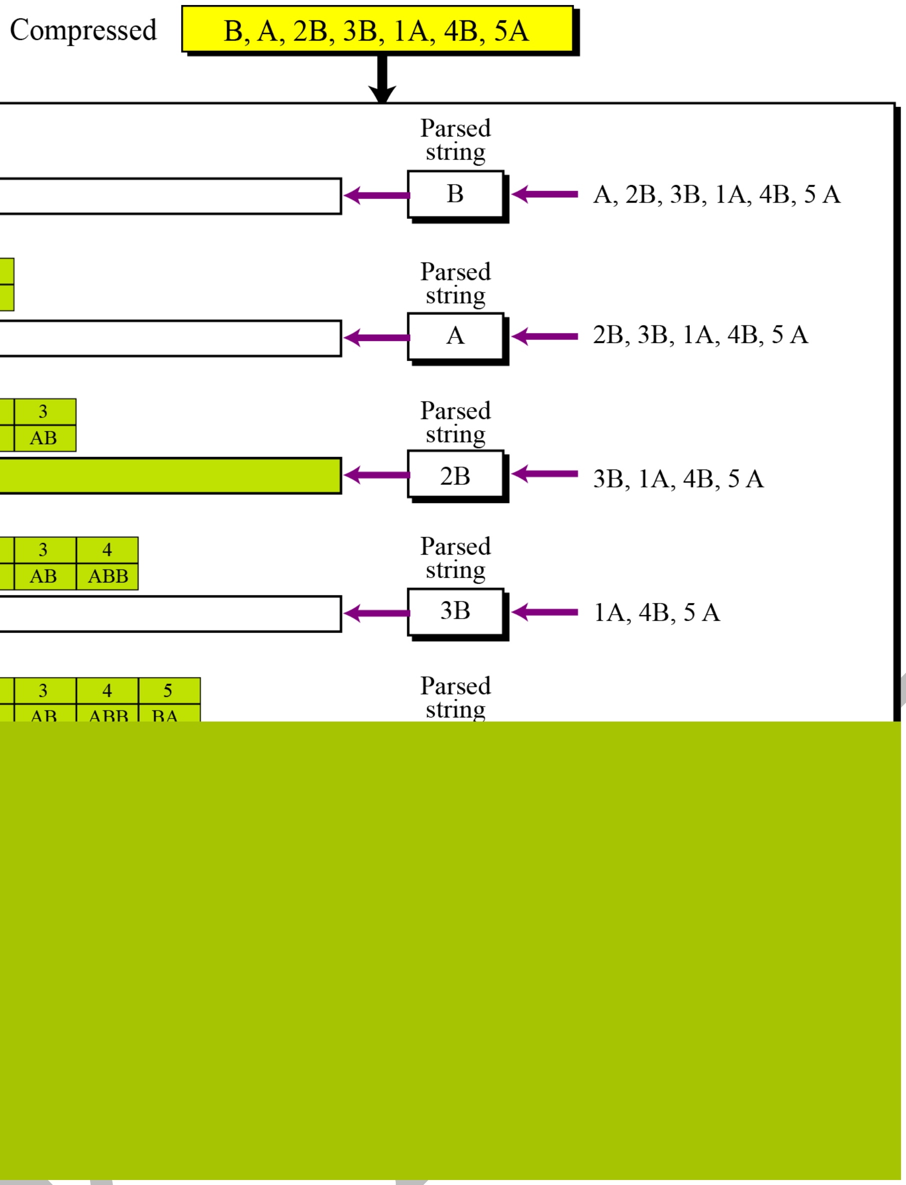


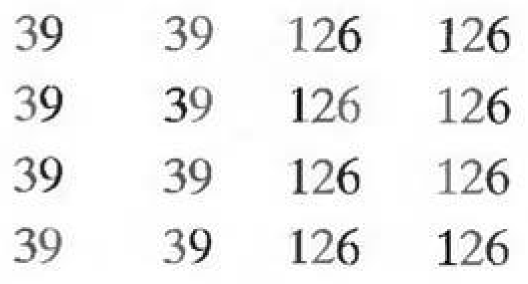
Figure 15.9 An example of Lempel Ziv decoding

The technique, called Lempel-Ziv-Welch (LZW) coding, assigns fixed-length code words to variable length sequences of source symbols but requires no a priori knowledge of the probability of occurrence of the symbols to be encoded. LZW compression has been integrated into a variety of mainstream imaging file formats, including the graphic interchange format (GIF), tagged image file format (TIFF), and the portable document format (PDF).

LZW coding is conceptually very simple (Welch [1984]). At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit monochrome images, the first 256 words of the dictionary are assigned to the gray values 0, 1, 2..., and 255. As the encoder sequentially examines the image's pixels, graylevel sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations.

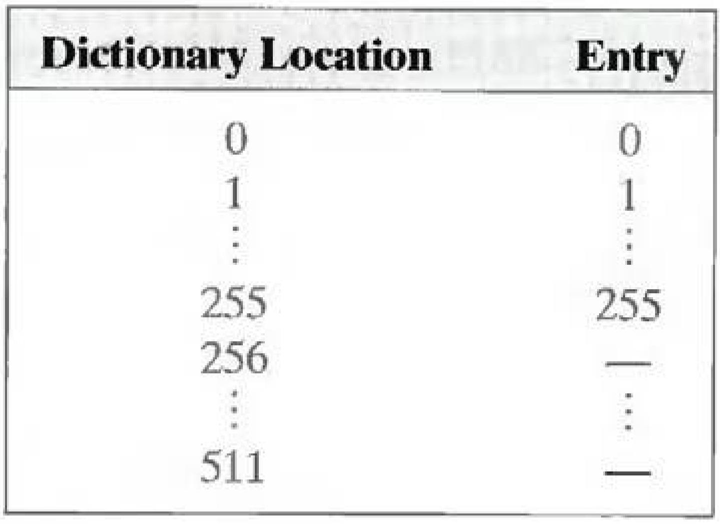
If the first two pixels of the image are white, for instance, sequence “255-255” might be assigned to location 256, the address following the locations reserved for gray levels 0 through 255. The next time that two consecutive white pixels are encountered, code word 256, the address of the location containing sequence 255-255, is used to represent them. If a 9-bit, 512-word dictionary is employed in the coding process, the original (8 + 8) bits that were used to represent the two pixels are replaced by a single 9-bit code word. Cleary, the size of the dictionary is an important system parameter. If it is too small, the detection of matching graylevel sequences will be less likely; if it is too large, the size of the code words will adversely affect compression performance.

Consider the following 4 x 4, 8-bit image of a vertical edge:



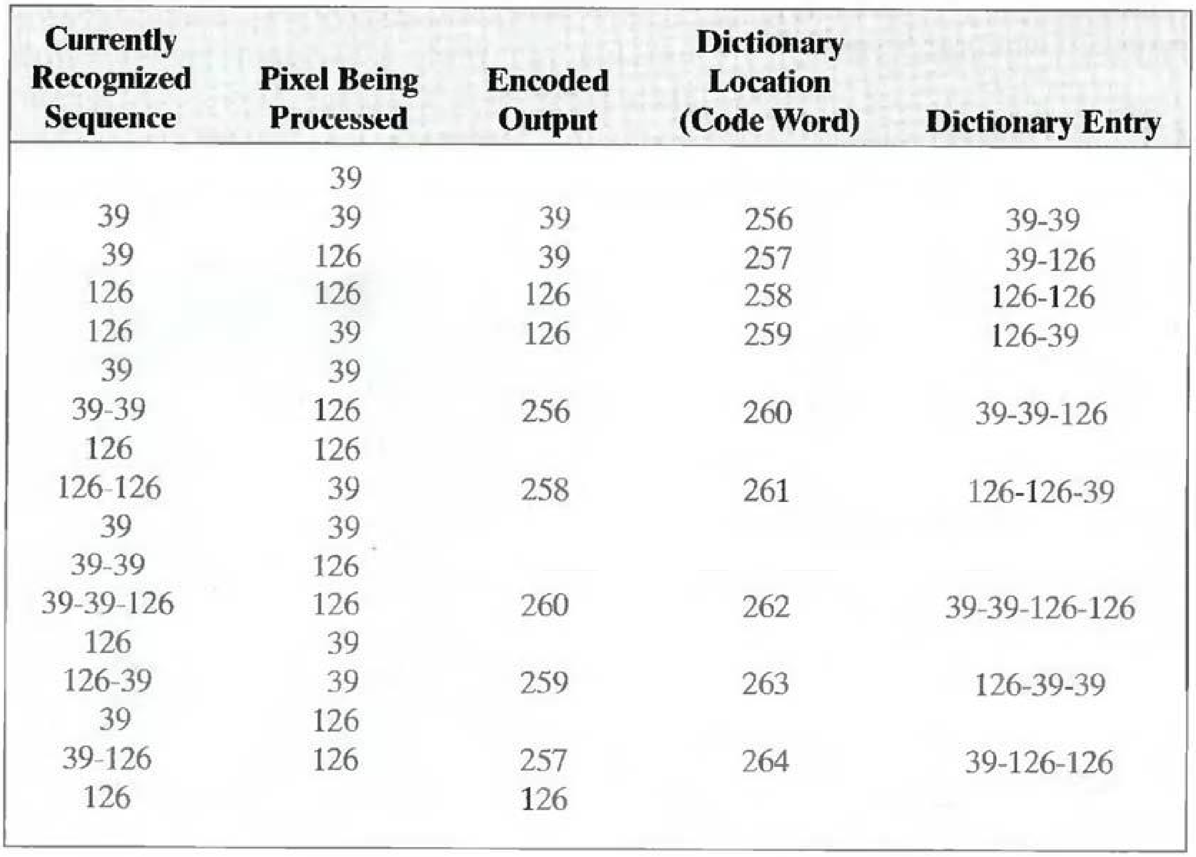
39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Table 6.1 details the steps involved in coding its 16 pixels. A 512-word dictionary with the following starting content is assumed:



Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Locations 256 through 511 are initially unused. The image is encoded by processing its pixels in a left-to-right, top-to-bottom manner. Each successive gray-level value is concatenated with a variable—column 1 of Table 6.1 —called the "currently recognized sequence." As can be seen, this variable is initially null or empty. The dictionary is searched for each concatenated sequence and if found, as was the case in the first row of the table, is replaced by the newly concatenated and recognized (i.e., located in the dictionary) sequence. This was done in column 1 of row 2.



Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Table 6.1 LZW coding example

No output codes are generated, nor is the dictionary altered. If the concatenated sequence is not found, however, the address of the currently recognized sequence is output as the next encoded value, the concatenated but unrecognized sequence is added to the dictionary, and the currently recognized sequence is initialized to the current pixel value. This occurred in row 2 of the table.

The last two columns detail the gray-level sequences that are added to the dictionary when scanning the entire 4 x 4 image. Nine additional code words are defined. At the conclusion of coding, the dictionary contains 265 code words and the LZW algorithm has successfully identified several repeating gray-level sequences—leveraging them to reduce the original 128-bit image lo 90 bits (i.e., 10 9-bit codes). The encoded output is obtained by reading the third column from top to bottom. The resulting compression ratio is 1.42:1.

A unique feature of the LZW coding just demonstrated is that the coding dictionary or code book is created while the data are being encoded. Remarkably, an LZW decoder builds an identical decompression dictionary as it decodes simultaneously the encoded data stream. .

Although not needed in this example, most practical applications require a strategy for handling dictionary overflow. A simple solution is to flush or reinitialize the dictionary when it becomes full and continue coding with a new initialized dictionary. A more complex option is to monitor compression performance and flush the dictionary when it becomes poor or unacceptable. Alternately, the least used dictionary entries can be tracked and replaced when necessary.

Transform Coding:

All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods. In this coding, we consider compression techniques that are based on modifying the transform of an image. In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded. For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion. A variety of transformations, including the discrete Fourier transform (DFT), can be used to transform the image data.

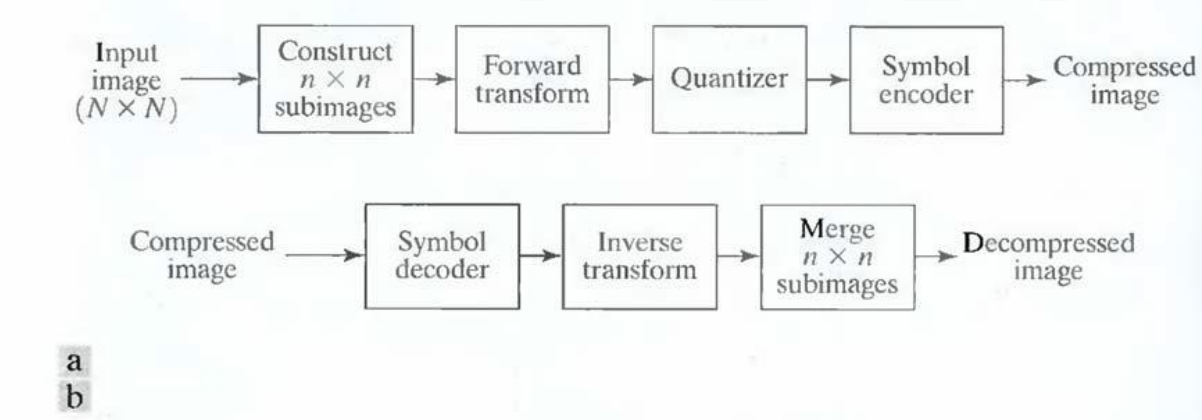
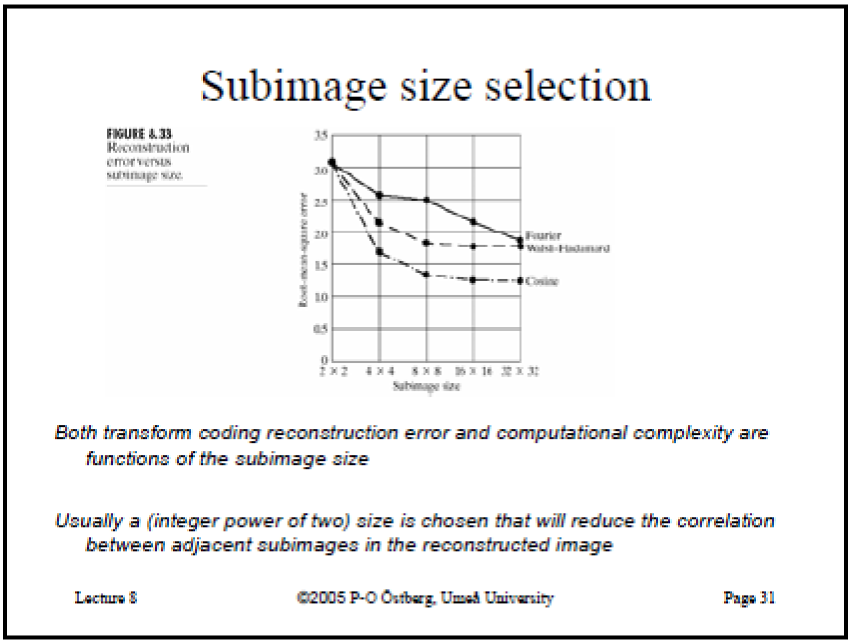


Fig. 10 A transform coding system: (a) encoder; (b) decoder.

Figure 10 shows a typical transform coding system. The decoder implements the inverse sequence of steps (with the exception of the quantization function) of the encoder, which performs four relatively straightforward operations: subimage decomposition, transformation, quantization, and coding.

An N X N input image first is subdivided into subimages of size n X n, which are then transformed to generate (N/n) 2 subimage transform arrays, each of size n X n. The goal of the transformation process is to decorrelate the pixels of each subimage, or to pack as much information as possible into the smallest number of transform coefficients. The quantization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information. These coefficients have the smallest impact on reconstructed subimage quality.

The encoding process terminates by coding (normally using a variable-length code) the quantized coefficients. Any or all of the transform encoding steps can be adapted to local image content, called adaptive transform coding, or fixed for all subimages, called nonadaptive transform coding.



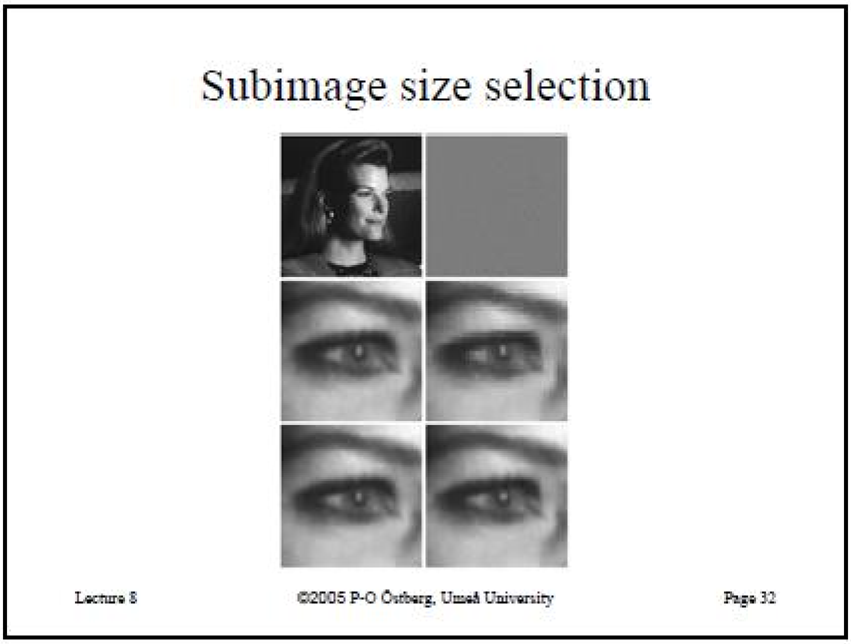
Subimage size selection

FIGURE 8.33 Reconstruction error versus subimage size.

Both transform coding reconstruction error and computational complexity are functions of the subimage size

Usually a (integer power of two) size is chosen that will reduce the correlation between adjacent subimages in the reconstructed image

Lecture 8 ©2005 P-O Östberg, Umeå University Page 31



Subimage size selection

Lecture 8 ©2005 P-O Östberg, Umeå University Page 32

BLOCKING ARTIFACTS

Block boundary artefacts[edit]

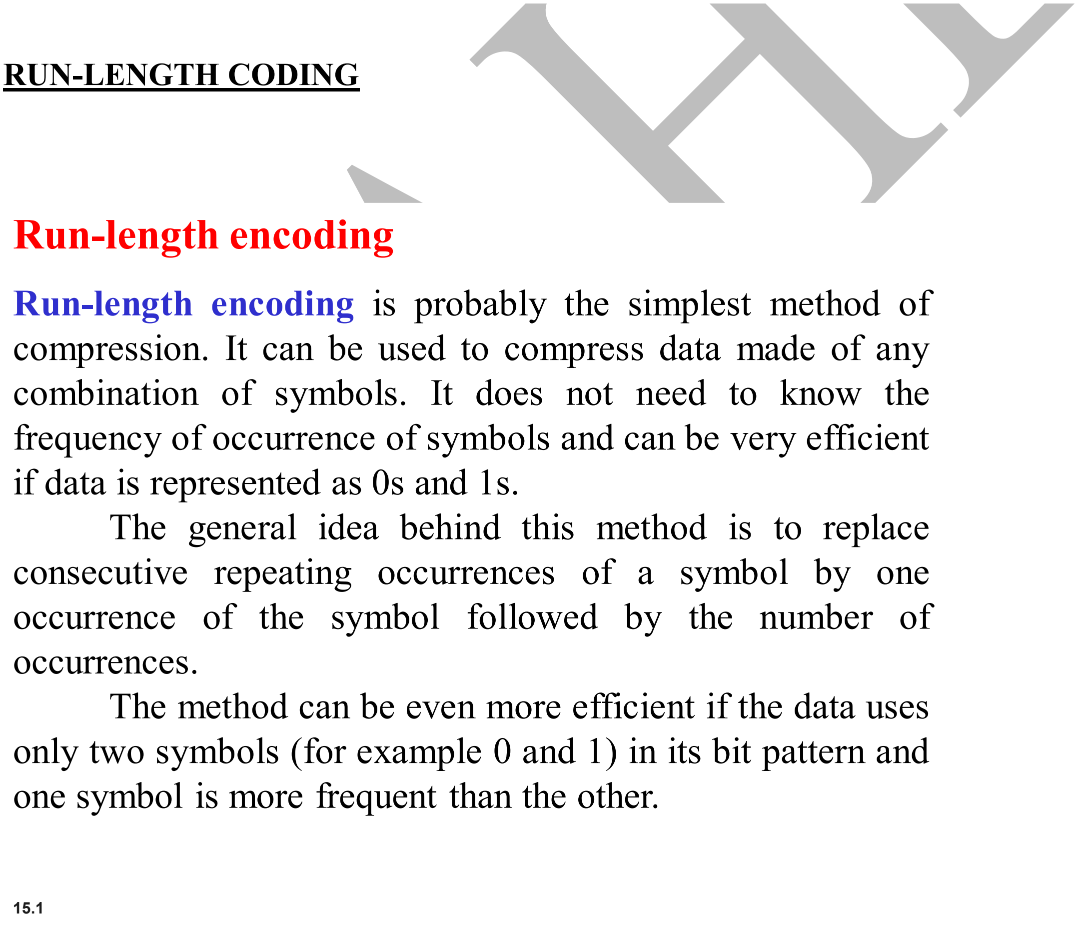
Block coding artefacts in a JPEG image. Flat blocks are caused by coarse quantization. Discontinuities at transform block boundaries are visible.

At low bit rates, any lossy block-based coding scheme introduces visible artefacts in pixel blocks and at block boundaries. These boundaries can be transform block boundaries, prediction block boundaries, or both, and may coincide with macroblock boundaries. The term *macroblocking* is commonly used regardless of the artefact's cause. Other names include tiling,[5] mosaicing, pixelating, quilting, and checkerboarding.

Block-artefacts are a result of the very principle of block transform coding. The transform (for example the discrete cosine transform) is applied to a block of pixels, and to achieve lossy compression, the transform coefficients of each block are quantized. The lower the bit rate, the more coarsely the coefficients are represented and the more coefficients are quantized to zero. Statistically, images have more low-frequency than high-frequency content, so it is the low-frequency content that remains after quantization, which results in blurry, low-resolution blocks. In the most extreme case only the DC-coefficient, that is the coefficient which represents the average color of a block, is retained, and the transform block is only a single color after reconstruction.

Because this quantization process is applied individually in each block, neighboring blocks quantize coefficients differently. This leads to discontinuities at the block boundaries. These are most visible in flat areas, where there is little detail to mask the effect.

RUN-LENGTH CODING



Run-length encoding

Run-length encoding is probably the simplest method of compression. It can be used to compress data made of any combination of symbols. It does not need to know the frequency of occurrence of symbols and can be very efficient if data is represented as 0s and 1s.

The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.

The method can be even more efficient if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other.

15.1

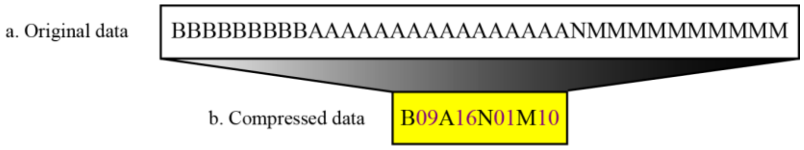


Figure 15.2 **Run-length encoding example**

15.6

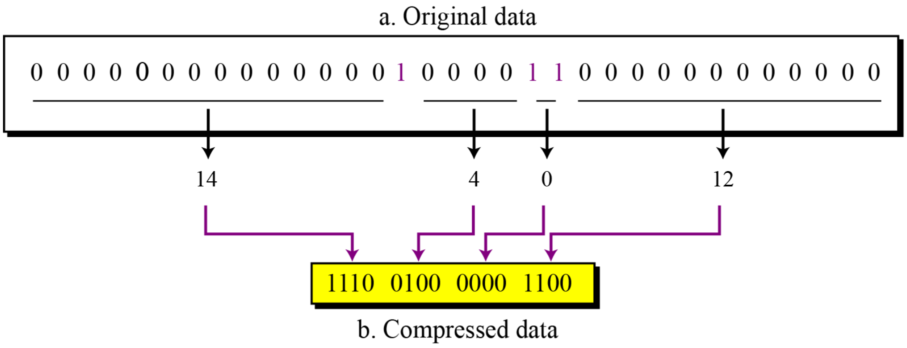


Figure 15.3 **Run-length encoding for two symbols**

15.7

KARPAGAM ACADEMY OF HIGHER EDUCATION
CLASS: III BSC CS COURSE NAME: DIGITAL IMAGE PROCESSING
COURSE CODE: 16CSU502B BATCH-2016-2019

UNIT III

SNO	QUESTIONS	CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	ANSWER
1	Image restoration is to improve the _______________of the image.	quality	noise	intensity	colour	quality
2	_________ is the operation of taking a corrupt/noisy image and estimating the clean, original image.	Image Restoration	Image Restoration	Image Restoration	Image Restoration	Image Restoration
3	The _________is responsible for reducing or eliminating any coding, interpixel, or psychovisual redundancies in the input image.	source encoder	quantizer	mapper	decoder	source encoder
4	Due to uniform linear motion image is_________________	blurred	sharpened	smoothened	blurred and smoothened	blurred and smoothened
5	Blur is characterized by the _______________ response of the system	filter	noise	impulse	image	impulse
6	Gaussian noise is referred to as __________	red noise	black noise	white noise	normal noise	normal noise
7	Convolution in spatial domain is multiplication in ___________	frequency domain	time domain	spatial domain	plane	frequency domain
8	Linear functions possesses the property of _____________	additivity	homogeneity	multiplication	Both A and B	Both A and B
9	PDF in image processing is called ______________	probability degraded function	probability density function	probabilistic degraded function	probabilistic density function	probability density function
10	Filter that replaces the pixel value with the medians of intensity levels is ___________	arithmetic mean filter	geometric mean filter	median filter	sequence mean filter	median filter
11	The purpose of restoration is to gain ________	degraded image	original image	pixels	coordinates	original image
12	Degraded image is produced using degradation process and __________	additive noise	destruction	pixels	coordinates	additive noise
13	Degraded image is given in a __________	frequency domain	time domain	spatial domain	plane	spatial domain

14	Digitizing the image intensity amplitude is called __________	sampling	quantization	framing	Both A and B	quantization
15	Compressed image can be recovered back by ____________	image enhancement	image decompression	image contrast	image equalization	image decompression
16	Digital video is sequence of ___________	pixels	frames	matrix	coordinates	frames
17	Image compression comprised of _____________	encoder	decoder	frames	encoder and decoder	encoder and decoder
18	Information is the ______________	data	meaningful data	raw data	Both A and B	meaningful data
19	Coding redundancy works on _____________	pixels	matrix	intensity	coordinates	intensity
20	Sequence of code assigned is called _____________	code word	word	byte	nibble	code word
21	Every run length pair introduce new _____________	pixels	matrix	frames	intensity	intensity
22	If the pixels are reconstructed without error mapping is said to be _____________	reversible	irreversible	temporal	facsimile	reversible
23	If the $P_{(E)} = 1$, it means event _____________	does not occur	always occur	no probability	normalization	always occur
24	The basic idea behind Huffman coding is to _____________	expand data by using fewer bits to encode more frequently occuring characters	compress data by using fewer bits to encode fewer frequently occuring characters	compress data by using more bits to encode more frequently occuring characters	compress data by using fewer bits to encode more frequently occuring characters	compress data by using fewer bits to encode more frequently occuring characters
25	Huffman coding is an encoding algorithm used for _____________	lossy data compression	lossless data compression	files greater than 1 Mbit	broadband systems	lossless data compression
26	A Huffman encoder takes a set of characters with fixed length and produces a set of characters of _____________	constant length	fixed length	random length	variable length	variable length
27	A Huffman code: A = 1, B = 000, C = 001, D = 01 ,P(A) = 0.4, P(B) = 0.1, P(C) = 0.2, P(D) = 0.3, The average number of bits per letter is	8.0 bit	2.1 bit	2.0 bit	1.9 bit	1.9 bit
28	The idea with wavelets is to represent a complicated function by _____________	square functions	lines	simple basic functions	sinus functions	simple basic functions
29	Down sampling is to make a digital image file smaller by _____________	removing pixels	removing noise	adding pixels	adding noise	removing pixels
30	Without losing quality, JPEG-2000 can achieve compression ratios of _____________	200:01:00	2000:01:00	20:01	2:01	200:1

31	The best visual compression quality is achieved using ______________	Wavelets	Fourier transform	Dolby	DCT	Wavelets
32	In the coding redundancy technique we use ______________	fixed length code	variable length code	byte	Both A and B	Both A and B
33	Morphology refers to ______________	pixels	matrix	frames	shape	shape
34	FAX is an abbreviation of ______________	fast	female	feminine	facsimile	facsimile
35	Source of information depending on finite no of outputs is called ______________	markov	finite memory source	zero source	Both A and B	Both A and B
36	Types of data redundancy are ______________	1	2	3	4	3
37	Information per source is called ______________	sampling	quantization	entropy	normaliza	entropy
38	Image with very high quality is considered as ______________	good	fair	bad	excellent	excellent
39	Range [0, L-1], where L is the ______________	no of levels	length	no of intensity levels	low quality	no of intensity levels
40	Compression is done for saving ______________	storage	bandwidth	money	Both A and B	Both A and B
41	System of symbols to represent event is called ______________	storage	word	code	nibble	Ccode
42	In the image MxN, M is ______________	rows	column	level	intensity	rows
43	In the image MxN, N is ______________	rows	column	level	intensity	column
44	Inferior image is the image having ______________	low definition	high definition	intensity	coordinates	low definition
45	Histogram equalization refers to image ______________	sampling	quantization	framing	normalization	normalization
46	The simple way to compression is removing	data	**redundant data**	information	meaningful data	**redundant data**
47	Shannons theorem is also called	**noiseless coding theorem**	noisy coding theorem	coding theorem	noiseless theorem	**noiseless coding theorem**
48	Information lost when expressed mathematically is called	markov	finite memory source	**fidelity criteria**	noiseless theorem	**fidelity criteria**
49	Error of the image is referred to as	pixels	matrix	frames	**noise**	**noise**
50	_______ and ________ are two important statistical measures on which the adaptive filtering depends upon.	Mean and variance	Sum and average	Mean and median	Max and Min	Mean and variance
51	In ________ data compression, the integrity of the data is preserved	lossless	lossy	image	text	lossless

52	When coding the symbols of an information source individually, ___________ yields the smallest possible number of code symbols per source symbol.	Huffman coding	Arithmetic coding	LZW coding	Transform coding	Huffman coding
53	___________ is an example of a category of algorithms called dictionary-based encoding.	Huffman coding	Arithmetic coding	Lempel Ziv (LZ) encoding	Transform coding	Lempel Ziv (LZ) encoding
54	_________is associated with the representation of information.	Coding redundancy	Interpixel redundancy	Psychovisual redundancies	Temporal Redundancy	Coding redundancy
55	___________, which deals with the assignment of gray levels to pixels in the spatially transformed image	gray-level interpolation	a spatial transformation	linear transformation	nonlinear transformation	gray-level interpolation
56	__________is due to the correlation between the neighboring pixels in an image.	Interpixel redundancy	Interpixel redundancy	Psychovisual redundancies	Temporal Redundancy	Interpixel redundancy
57	In ____________, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded.	Huffman coding	Arithmetic coding	Lempel Ziv (LZ) encoding	Transform coding	Transform coding
58	_________ rejects or pass frequencies in predefined neighborhoods about a center frequency	Adaptive filter	Notch filter	Mean filter	Max filter	Notch filter
59	_________ exist because human perception does not involve quantitative analysis of every pixel or luminance value in the image.	Psychovisual redundancies	Interpixel redundancy	Psychovisual redundancies	Temporal Redundancy	Psychovisual redundancies
60	__________, which defines the "rearrangement" of pixels on the image plane	a spatial transformation	gray-level interpolation	linear transformation	nonlinear transformation	a spatial transformation

UNIT – IV

FAX compression (CSUITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation
Wavelet based Image Compression: Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT),Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking

FAX COMPRESSION (CSUITT GROUP-3 AND GROUP-4)

Fax Group is an encoding format used for fax transmission. There are two types: Fax Group 3, also known as G3, and Fax Group 4, also known as G4. Fax Group 3 and 4 are two of the encoding formats for Tagged Image File Format (TIFF) files. The more commonly used format, Fax Group 3, is Recommendation T.4 of the CCIT , now known as the ITU-T (for Telecommunication Standardization Sector of the International Telecommunications Union).

Fax Group 3 supports one-dimensional image compression of black and white images, On a standard fax machine, Fax Group 3 uses redundancy reduction to enhance speed and is able to transmit a page in one minute or less. Fax Group 3 can achieve compression ratios of 10:1 for office documents and 15:1 for engineering drawings with a resolution of 200 dots per inch (dpi).

Less frequently used, Fax Group 4 (G4) is ITU-T Recommendation T.6 and supports two-dimensional image compression, compressing the line width as well as the line length. Fax Group 4 can achieve compression ratios of 15:1 for office documents and 20:1 for engineering drawings with a resolution of 400 dpi. Unlike Fax Group 3, Fax Group 4 can use Integrated Services Digital Network (ISDN) for transmission.

CCITT Group 4 compression, also referred to as **G4** or **Modified Modified READ** (MMR), is a lossless method of image compression used in Group 4 fax machines defined in the ITU-T T.6 fax standard. It is only used for bitonal (black and white) images. Group 4 compression is based on the Group 3 two-dimensional compression scheme (G3-2D), also known as Modified READ, which is in turn based on the Group 3 one-dimensional compression scheme (G3), also known as Modified Huffman coding. Group 4 compression is available in many proprietary image file formats as well as standardized formats such as TIFF, CALS, CIT (Intergraph Raster Type 24) and the PDF document format.

G4 offers a small improvement over G3-2D by removing the end of line (EOL) codes. G3 and G4 compression both treat an image as a series of horizontal black strips on a white page. Better compression is achieved when there are fewer unique black dots/lines on the page. Both G3-2D and G4 add a two dimensional feature to achieve greater compression by taking advantage of vertical symmetry. A worst-case image would be an alternating pattern of single-pixel black and white dots offset by one pixel on even/odd lines. G4 compression would actually increase the file size on this type of image. G4 typically achieves a 20:1 compression ratio. For an 8.5"×11" page scanned at 200 DPI, this equates to a reduction from 467.5 kB to 23.4 kB (95% compression ratio).

Group 3 & Group 4

GROUP 3

•MODIFIED HUFFMAN METHOD (MHM) – Unidimensional coding method based on the coding of the length of alternate black and white pixel runs using Huffman coding.

GROUP 4 (Also Group 3 Options)

•MODIFIED READ METHOD (MRM) – Bidimensional coding method based on the coding of the variations of the positions of tone transition pixels (black-white or white-black) in relation to the previous line; unidimensional coding may be used every k lines.

•MODIFIED-MODIFIED READ METHOD (MMRM) – Similar to MRM but without periodic unidimensional coding.

JBIG-2

JBIG2 is an image compression standard for bi-level images, developed by the Joint Bi-level Image Experts Group. It is suitable for both lossless and lossy compression. According to a press release[1] from the Group, in its lossless mode JBIG2 typically generates files 3–5 times smaller than Fax Group 4 and 2–4 times smaller than JBIG, the previous bi-level compression standard released by the Group. JBIG2 has been published in 2000 as the international standard ITU T.88,[2] and in 2001 as ISO/IEC 14492.[3]

JBIG • Lossless Compression. • Progressive Coding. • Sequential Coding. • Arithmetic Encoder/Decoder. • Resolution Reduction Algorithm • (optional, can be replaced).

Resolution Reduction Technique • Creates low resolution images. • Combines decimation and filtering in one action. • Uses 9 high resolution and 3 low resolution pixels to determine color of target pixel. • Preserves gray-levels achieved with halftoning.

JBIG Pro's and Con's ☺ Progressive (for binary images). ☺ Better compression for images with up to ☺ 6 bit/pixel (Vs. JPEG) ☺ Better compression than G3 and G4. / Slow and complicated (Vs. JPEG, G.3/4) / Consumes memory resources and needs frame buffers.

Compression Comparison (Cont'd) Typical results for: • Scanned text and line drawings: JBIG ~20-25% better than G4 • Computer generated line-drawing: JBIG ~75% better than G4 • Scanned dither halftones images: JBIG ~85%-90% better than G4

Functionality

Ideally, a JBIG2 encoder will segment the input page into regions of text, regions of halftone images, and regions of other data. Regions that are neither text nor halftones are typically compressed using a context-dependent arithmetic coding algorithm called the MQ coder. Textual regions are compressed as follows: the foreground pixels in the regions are grouped into symbols. A dictionary of symbols is then created and encoded, typically also using context-dependent arithmetic coding, and the regions are encoded by describing which symbols appear where. Typically, a symbol will correspond to a character of text, but this is not required by the compression method.

For lossy compression the difference between similar symbols (e.g., slightly different impressions of the same letter) can be neglected; for lossless compression, this difference is taken into account by compressing one similar symbol using another as a template. Halftone images may be compressed by reconstructing the grayscale image used to generate the halftone and then sending this image together with a dictionary of halftone patterns.[4] Overall, the algorithm used by JBIG2 to compress text is very similar to the JB2 compression scheme used in the DjVu file format for coding binary images.

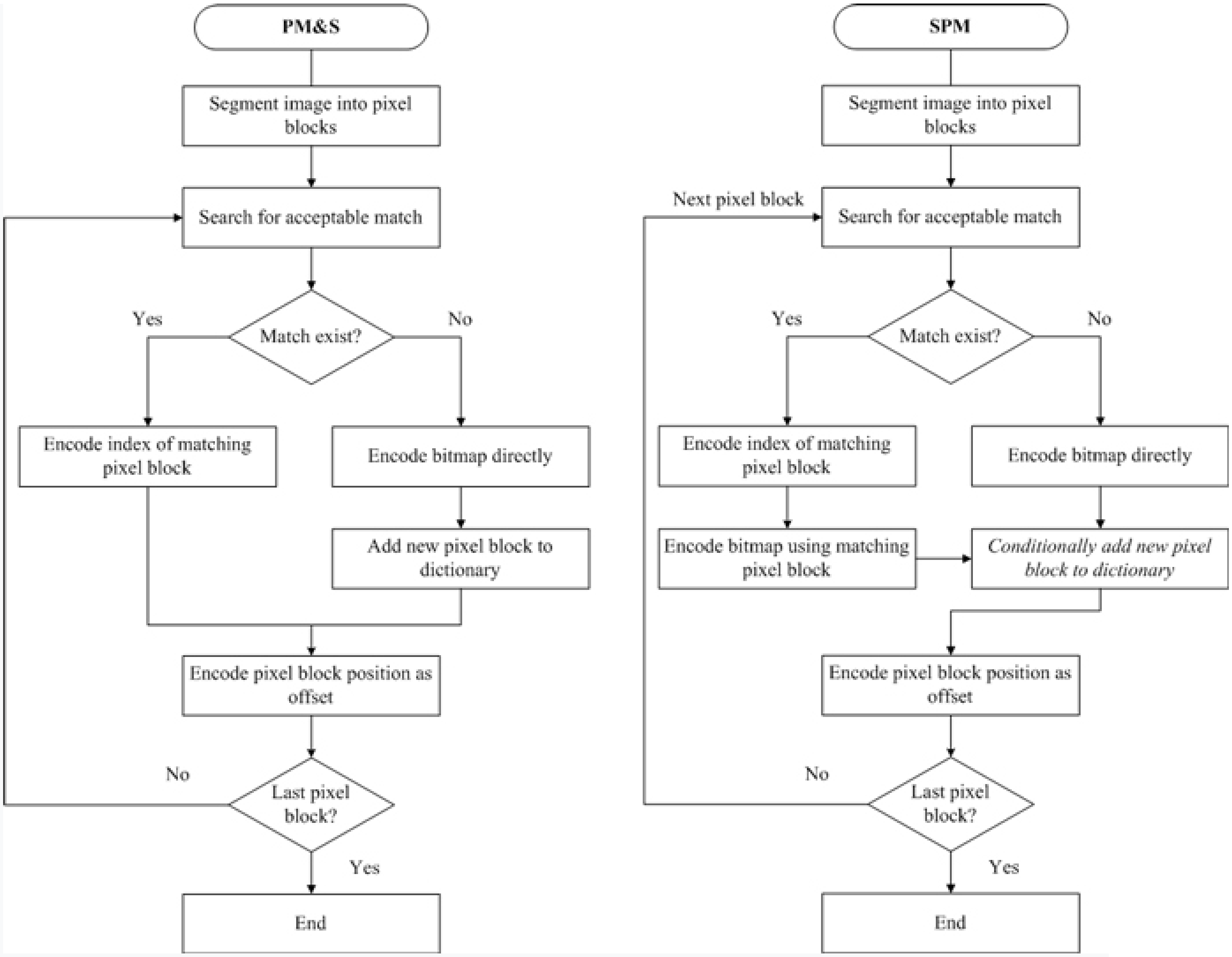
PDF files versions 1.4 and above may contain JBIG2-compressed data. Open-source decoders for JBIG2 are jbig2dec,[5] the java-based jbig2-imageio[6] and the decoder found in versions 2.00 and above of xpdf. An open-source encoder is jbig2enc.[7]

Technical details

Typically, a bi-level image consists mainly of a large amount of textual and halftone data, in which the same shapes appear repeatedly. The bi-level image is segmented into three regions: text, halftone, and generic regions. Each region is coded differently and the coding methodologies are described in the following passage.

Text image data

Text coding is based on the nature of human visual interpretation. A human observer cannot tell the difference between two instances of the same characters in a bi-level image even though they may not exactly match pixel by pixel. Therefore, only the bitmap of one representative character instance needs to be coded instead of coding the bitmaps of each occurrence of the same character individually. For each character instance, the coded instance of the character is then stored into a "symbol dictionary".[8] There are two encoding methods for text image data: pattern matching and substitution (PM&S) and soft pattern matching (SPM). These methods are presented in the following subsections.[9]



Block diagrams of (left) pattern matching and substitution method and (right) soft pattern matching method

Pattern matching and substitution

After performing image segmentation and match searching, and if a match exists, we code an index of the corresponding representative bitmap in the dictionary and the position of the character on the page. The position is usually relative to another previously coded character. If a match is not found, the segmented pixel block is coded directly and added into the dictionary. Typical procedures of pattern matching and substitution algorithm are displayed in the left block diagram of the figure above. Although the method of PM&S can achieve outstanding compression, substitution errors could be made during the process if the image resolution is low.

Soft pattern matching

In addition to a pointer to the dictionary and position information of the character, refinement data is also required because it is a crucial piece of information used to reconstruct the original character in the image. The deployment of refinement data can make the character-substitution error mentioned earlier highly unlikely. The refinement data contains the current desired character instance, which is coded using the pixels of both the current character and the matching character in the dictionary. Since it is known that the current character instance is highly correlated with the matched character, the prediction of the current pixel is more accurate.

Halftones

Halftone images can be compressed using two methods. One of the methods is similar to the context-based arithmetic coding algorithm, which adaptively positions the template pixels in order to obtain correlations between the adjacent pixels. In the second method, descreening is performed on the halftone image so that the image is converted back to grayscale. The converted grayscale values are then used as indexes of fixed-sized tiny bitmap patterns contained in a halftone bitmap dictionary. This allows decoder to successfully render a halftone image by presenting indexed dictionary bitmap patterns neighboring with each other.

Arithmetic entropy coding

All three region types including text, halftone, and generic regions may all use arithmetic coding. JBIG2 specifically uses the MQ coder.

BIT-PLANE CODING

Another effective technique for reducing an image's interpixel redundancies is to process the image's bit planes individually. The technique, called *bit-plane coding,* IS based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images and compressing each binary image via one of several well-known binary compression methods. In this section, we describe the most popular decomposition approaches and review several of the more commonly used compression methods. Bit-plane decomposition
The gray levels of an *m-bit* gray-scale Image can be represented m the form of
the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \qquad + a_1 2^1 + a_0 2^0. \qquad (8.4\text{-}2)$$

Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the *m* coefficients of the polynomial into *m* 1-bit *bit planes.*

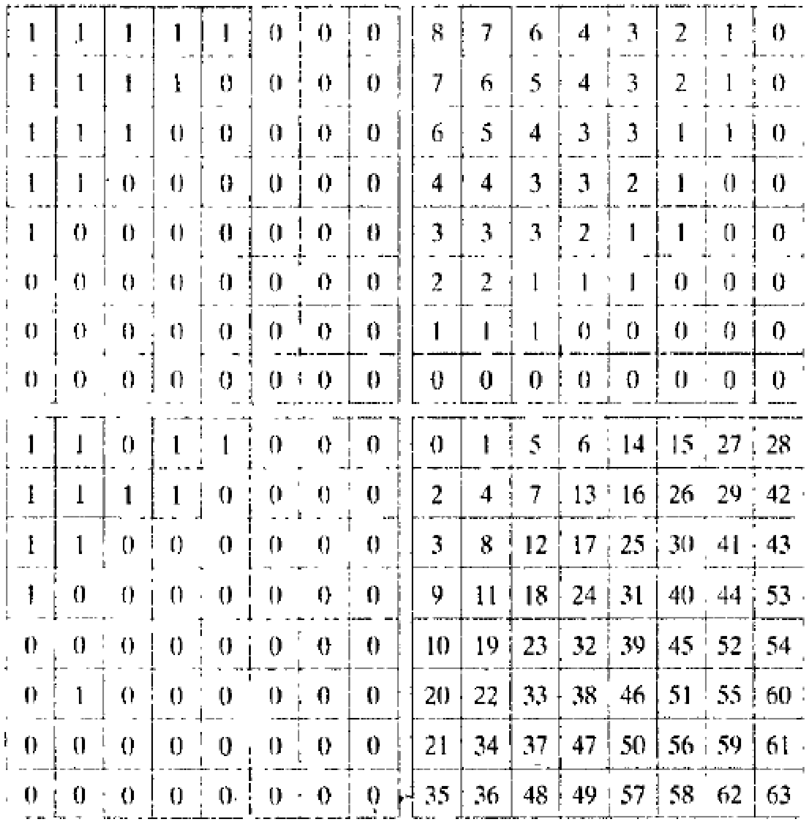
BIT ALLOCATION

The reconstruction error associated with the truncated series expansion of Eq. (8.5-38) is a function of the number and relative importance of the transform coefficients that are discarded, as well as the precision that is used to represent the retained coefficients. In most transform coding systems, the retained coefficients are selected [that is, the masking function of Eq. (8.5-37) is constructed] on the basis of maximum variance, called *zonal coding,* or on the basis of maximum magnitude, called *threshold coding.* The overall process of truncating, quantizing, and coding the coefficients of a transformed subimage is commonly called *bit allocation.*

Zonal Coding

Zonal coding implementation Zonal coding is based on the information theory concept of viewing information as uncertainty. Therefore the transform coefficients of maximum variance carry the most image information and should be retained in the coding process. The variances themselves can be calculated directly from the ensemble of *(N/ n?* transformed subimage arrays. as in the preceding example, or based on an assumed image model (say. a Markov autocorrelation function). In either case, the zonal sampling process can be viewed, in accordance with Eq. (8.5-38). as multiplying each *T(II, v)* by the corresponding element in a *zonal mask,* which is constructed by placing a 1 in the locations of maximum variance and a 0 in all other locations. Coefficients of maximum variance usually are located around the origin of an image transform. resulting in the typical zonal mask shown in Fig. 8.36(a).

The coefficients retained during the zonal sampling process must be quantized and coded, so zonal masks are sometimes depicted showing the number of bits used to code each coefficient [Fig. 8.36(b)]. In most cases, the coefficients are allocated the same number of bits, or some fixed number of bits is distributed among them unequally. In the first case, the coefficients generally are normalized by their standard deviations and uniformly quantized. In the second case. a quantizer. such as an optimal Lloyd-Max quantizer. is designed for each coefficient. To construct the required quantizers, the zeroth or dc coefficient normally is modeled by a Rayleigh density function, whereas the remaining coefficients are modeled by a Laplacian or Gaussian density.[†] The number of quantization levels (and thus the number of bits) allotted to each quantizer is made proportional to $\log_2 \sigma^2_{T(u,v)}$. This allocation is consistent with rate distortion theory, which indicates that a Gaussian random variable of variance σ^2 cannot be represented by less than $\frac{1}{2}\log_2(\sigma^2/D)$ bits and be reproduced with mean-square error less than D (see Problem 8.11). The intuitive conclusion is that the information content of a Gaussian random variable is proportional to $\log_2(\sigma^2/D)$. Thus the retained coefficients in Eq. (8.5-38)—which (in the context of the current discussion) are selected on the basis of maximum variance—should be assigned bits in proportion to the logarithm of the coefficient variances.



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

a b
c d

FIGURE 8.36 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.

Threshold Coding

Thresholding Coding Implementation Zonal coding usually is implemented by using a single fixed mask for all subimages. Threshold coding. however. is inherently adaptive in the sense that the location of the transform coefficients retained for each subimage vary from one subimage to another. In fact. threshold coding is the adaptive transform coding approach most often used in practice because of its computational simplicity. The underlying concept is that. for an\' subimage. the transform coefficients of largest magnitude make the most significant contribution to reconstructed subimage quality. as demonstrated in the last example.

Because the locations of the maximum coefficients vary from one subimage to another. the elements of Y(u,v*)T(u. v)* normally are reordered (in a predefined manner) to form a I-D. run-length coded sequence. Figure *8.3h(c)* shows a *typical threshold mask* for one subimage of a hypothetical image. This mask provides a convenient way to visualize the threshold coding process for the corresponding subimage. as well as to mathematically describe the process using Eq. (8.5-38).

When the mask is applied [via Eq. (8.5-38)] to the subimage for which it was derived. and the resulting *n* x *n* array is reordered to form an n'-element coefficient sequence in accordance with the zigzag ordering pattern of Fig. 8.34(d). the reordered 1-0 sequence contains several long runs of O's [the zigzag pattern becomes evident by starting at 0 in Fig. 8.36(d) and following the numbers in sequence]. These runs normally are run-length coded. The nonzero or retained coefficients. corresponding to the mask locations that contain a L are represented using one of the variable-length codes of Section 8.4.

There are three basic ways to threshold a transformed subimage or. stated differently. to create a subimage threshold masking function of the form given in Eq. (8.5-37): (l) A single global threshold can be applied to all subimages: (2) a different threshold can be used for each subimage: or (3) the threshold can be varied as a function of the location of each coefficient within the subimage. **In** the first approach. the level of compression differs from image to image. depending on the number of coefficients that exceed the global threshold.

JPEG

Image compression

Image compression is the method of data compression on digital images.

The main objective in the image compression is:

- Store data in an efficient form
- Transmit data in an efficient form

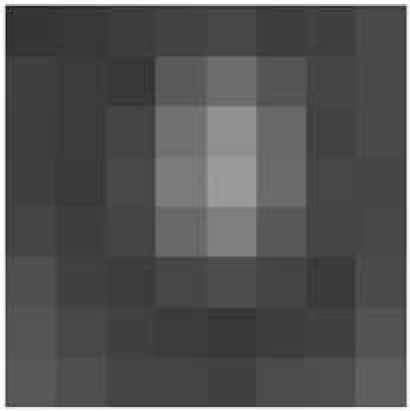
Image compression can be lossy or lossless.

JPEG compression

JPEG stands for Joint photographic experts group. It is the first interanational standard in image compression. It is widely used today. It could be lossy as well as lossless . But the technique we are going to discuss here today is lossy compression technique.

How jpeg compression works

First step is to divide an image into blocks with each having dimensions of 8 x8.



Let’s for the record, say that this 8x8 image contains the following values.

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

The range of the pixels intensities now are from 0 to 255. We will change the range from -128 to 127.

Subtracting 128 from each pixel value yields pixel value from -128 to 127. After subtracting 128 from each of the pixel value, we got the following results.

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Now we will compute using this formula.

$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$$

The result comes from this is stored in let's say A(j,k) matrix.

There is a standard matrix that is used for computing JPEG compression, which is given by a matrix called as Luminance matrix.

This matrix is given below

$$Q_{j,k} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Applying the following formula

$$B_{j,k} = \text{round}\left(\frac{A_{j,k}}{Q_{j,k}}\right)$$

We got this result after applying.

$$B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now we will perform the real trick which is done in JPEG compression which is ZIG-ZAG movement. The zig zag sequence for the above matrix is shown below. You have to perform zig zag until you find all zeroes ahead. Hence our image is now compressed.

$$B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Summarizing JPEG compression

The first step is to convert an image to Y'CbCr and just pick the Y' channel and break into 8 x 8 blocks. Then starting from the first block, map the range from -128 to 127. After that you have to find the discrete Fourier transform of the matrix. The result of this should be quantized. The last step is to apply encoding in the zig zag manner and do it till you find all zero.

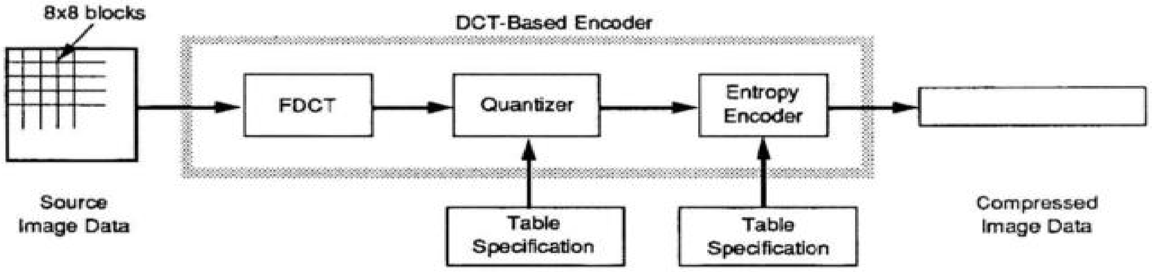
Save this one dimensional array and you are done.

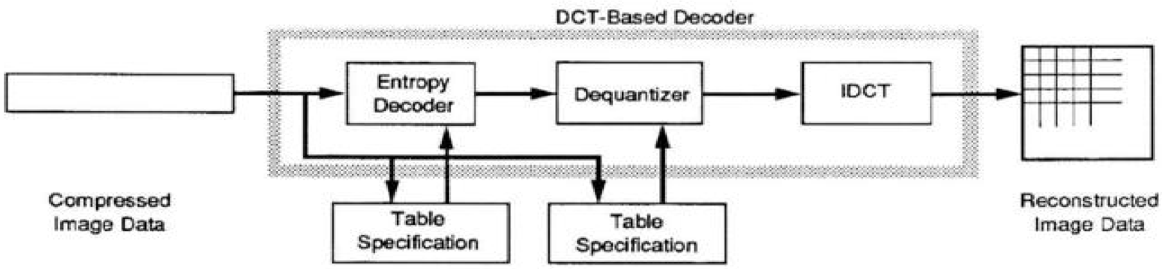
IMAGE COMPRESSION

- **Image compression** is a method to reduce the redundancies in image representation in order to decrease data storage requirements (Gonzalez and Woods, 2013).
- It is a technique used to compress an image without visually reducing the quality of the image itself.
- Data vs. Information.
- The goal of these processes is to represent an image with the same quality level, but in a more solid form.
- The large storage requirement of multimedia data.

The video or image files consume large amount of data and it always required very high bandwidth networks in transmission as well as communication costs.

- Reduce the data storage and maintain the visual image quality (Gonzalez, Woods and Eddins, 2017).
- Increase the speed of transmission by using the repetition property of data.
- The goal of these processes is to represent an image with the same quality level, but in a more solid form.





There are two types of compression methods:

- Lossy image compression

- Lossless image compression

LOSSY COMPRESSION

Unlike the error-free approaches outlined in the previous section, lossy encoding is based on the concept of compromising the accuracy of the reconstructed image m exchange for increased compression. If the resulting distortion (which mayor may not be visually apparent) can be tolerated, the increase m compression can be significant. **In** fact, many lossy encoding techniques are capable of reproducing recognizable monochrome images from data that have been compressed by more than 100: 1 and images that are virtually indistinguishable from the originals at 10: 1 to 50: 1. Error-free encoding of monochrome images, however, seldom results in more than a 3: 1 reduction in data. As indicated in Section 8.2, the principal difference between these two approaches is the presence or absence of the quantizer block of FIg. 8.6.

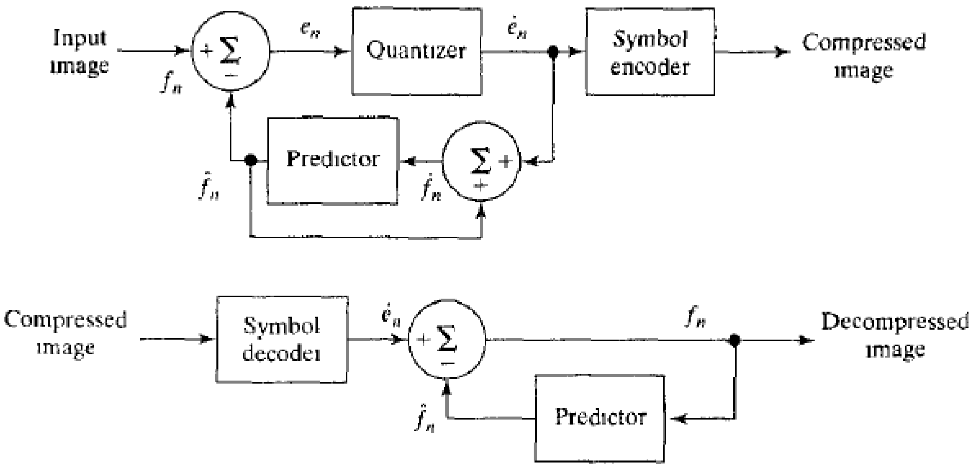
LOSSY PREDICTIVE CODING

- Lossy image compression methods are required to achieve high compression ratios for complex images.
- An image reconstructed Lossy compression can be performed in both spatial or transform domains.
- The process of quantization-dequantization introduces loss in the reconstructed image and is inherently responsible for the "lossy" nature of the compression scheme.
- The quantized transform coefficient is computed by

$$T_{pq} = Round\left(\frac{B_{pq}}{Q_{pq}}\right)$$

Where is the frequency image signals at coordinates (*i*,*j*) in the *k* block.

we add a quantizer to the model introduced in Section 8.4.4 and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig. 8.21 shows, the quantizer, which absorbs the nearest integer function of the error-free encoder, IS inserted between the symbol



a
b

FIGURE 8.21 A lossy predictive coding model: (a) encoder and (b) decoder.

encoder and the point at which the prediction error is formed. It maps the predictIOn error into a limited range of outputs, denoted *en,* which establish the amount of compression and distortion associated with lossy predictive coding. In order to accommodate the insertion of the quantization step, the error-free encoder of Fig. 8.19(a) must be altered so that the predictions generated by the encoder and decoder are equivalent. As Fig. 8.21(a) shows, this is accomplished by placing the lossy encoder's predictor within a feedback loop, where its input, denoted *f n'* is generated as a function of past predictions and the corresponding quantized errors. That is,

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

where f*n* is as defined in Section 8 4.4. This closed loop configuration prevents error buildup at the decoder's output. Note from Fig. 8.21(b) that the output of the decoder also is given by Eq. (8.5-1)

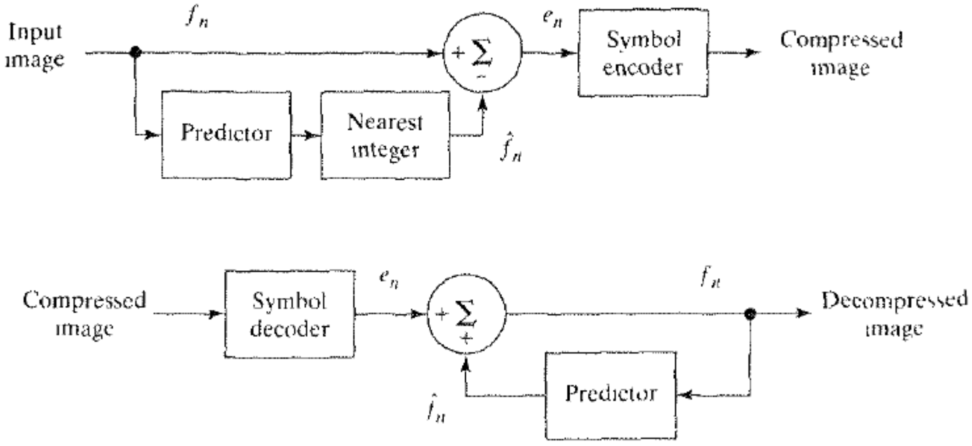
LOSSLESS PREDICTIVE CODING

Lossless compression methods are needed in some digital imaging applications, such as: medical images, x-ray images, etc.

LOSSLESS PREDICTIVE CODING

Let us now turn to an error-free compression approach that does not require decomposition of an image into a collection of bit planes The approach, commonly referred to as *lossless predictive coding,* IS based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding *only* the new information in each pixel The *new information* of a pixel is defined as the difference between the actual and predicted value of that pixel Figure 8.19 shows the basic components of a lossless predictive coding system. The system consists of an encoder and a decoder, each containing an identical *predictor.* As each successive pixel of the input image, denoted *f",* is introduced to the encoder, the predictor generates the anticipated value of that pixel based on some number of past inputs. The output of the predictor IS then rounded to the nearest integer, denoted *fn'* and used to form the difference' or *prediction error*

$$e_n = f_n - \hat{f}_n,$$



a
b
FIGURE 8.19 A lossless predictive coding model: (a) encoder, (b) decoder

which is coded using a variable-length code (by the symbol encoder) to generate the next element of the compressed data stream. The decoder of Fig. 8.19(b) reconstructs e_n from the received variable-length code words and performs the inverse operation

$$f_n = e_n + \hat{f}_n \tag{8.4-6}$$

Various local, global, and adaptive (see Section 8.5.1) methods can be used to generate $\hat{f}_n$. In most cases, however, the prediction is formed by a linear combination of m previous pixels. That is,

$$\hat{f}_n = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right] \tag{8.4-7}$$

where m is the order of the linear predictor, round is a function used to denote the rounding or nearest integer operation, and the α_i for $i = 1, 2, \ldots, m$ are prediction coefficients. In raster scan applications, the subscript n indexes the predictor out-

Lossless compression techniques: run-length coding, Huffman coding, lossless predictive coding etc.

Run-Length Encoding of AC Coefficients

The frequency image signals after quantization process consist of many zeros. Then, a special condition as known end-of-block (EOB) is applied to get an efficient in the entropy code.
A symbol of EOB indicates that the rest of the coefficients in the block are zero. Next, run-length encoding exploits the repeating frequency image signals as the symbols in the sequence a set of the AC coefficients.
The output of run-length encoding represents a sequence value with the consecutive repetition as symbols and the length of occurrence of the symbols

Huffman Coding

- Huffman coding is a famous method that uses variable-length codes (VLC) tables for compressing data (Jayaraman, Veerakumar, and Esakkirajan, 2017).
- Given a set of data symbols and their probabilities, the method creates a set of variable-length codeword with the shortest average length and assigns them to the symbols.

- Achieve minimal redundancy subject to the constraint that the source symbols be coded one at a time.
- Sorting symbols in descending probabilities is the key in the step of source reduction.
- The codeword assignment is not unique. Exchange the labeling of “0” and “1” at any node of binary codeword tree would produce another solution that equally works well.

Only works for a source with finite number of symbols (otherwise, it does not know where to start).

MOTION COMPENSATION

Motion compensation is an algorithmic technique used to predict a frame in a video, given the previous and/or future frames by accounting for motion of the camera and/or objects in the video. It is employed in the encoding of video data for video compression, for example in the generation of MPEG-2 files. Motion compensation describes a picture in terms of the transformation of a reference picture to the current picture. The reference picture may be previous in time or even from the future. When images can be accurately synthesised from previously transmitted/stored images, the compression efficiency can be improved.

Functionality

Motion compensation exploits the fact that, often, for many frames of a movie, the only difference between one frame and another is the result of either the camera moving or an object in the frame moving. In reference to a video file, this means much of the information that represents one frame will be the same as the information used in the next frame.

Using motion compensation, a video stream will contain some full (reference) frames; then the only information stored for the frames in between would be the information needed to transform the previous frame into the next frame.

Illustrated example

The following is a simplistic illustrated explanation of how motion compensation works. Two successive frames were captured from the movie Elephants Dream. As can be seen from the images, the bottom (motion compensated) difference between two frames contains significantly less detail than the prior images, and thus compresses much better than the rest. Thus the information that is required to encode compensated frame will be much smaller than with the difference frame. This also means that it is also possible to encode the information using difference image at a cost of less compression efficiency but by saving coding complexity without motion compensated coding; as a matter of fact that motion compensated coding (together with motion estimation, motion compensation) occupies more than 90% of encoding complexity.

MPEG

In MPEG, images are predicted from previous frames (P frames) or bidirectionally from previous and future frames (B frames). B frames are more complex because the image sequence must be transmitted and stored out of order so that the future frame is available to generate the B frames.[1]

After predicting frames using motion compensation, the coder finds the residual, which is then compressed and transmitted.

Global motion compensation

In global motion compensation, the motion model basically reflects camera motions such as:

- Dolly - moving the camera forward or backward
- Track - moving the camera left or right
- Boom - moving the camera up or down
- Pan - rotating the camera around its Y axis, moving the view left or right
- Tilt - rotating the camera around its X axis, moving the view up or down
- Roll - rotating the camera around the view axis

It works best for still scenes without moving objects.

Block motion compensation

In **block motion compensation** (BMC), the frames are partitioned in blocks of pixels (e.g. macroblocks of 16×16 pixels in MPEG). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by a *motion vector*.

To exploit the redundancy between neighboring block vectors, (e.g. for a single moving object covered by multiple blocks) it is common to encode only the difference between the current and previous motion vector in the bit-stream. The result of this differencing process is mathematically equivalent to a global motion compensation capable of panning. Further down the encoding pipeline, an entropy coder will take advantage of the resulting statistical distribution of the motion vectors around the zero vector to reduce the output size.

Variable block-size motion compensation

Variable block-size motion compensation (VBSMC) is the use of BMC with the ability for the encoder to dynamically select the size of the blocks. When coding video, the use of larger blocks can reduce the number of bits needed to represent the motion vectors, while the use of smaller blocks can result in a smaller amount of prediction residual information to encode.

Overlapped block motion compensation

Overlapped block motion compensation (OBMC) is a good solution to these problems because it not only increases prediction accuracy but also avoids blocking artifacts. When using OBMC, blocks are typically twice as big in each dimension and overlap quadrant-wise with all 8 neighbouring blocks.

Quarter Pixel (QPel) and Half Pixel motion compensation

In motion compensation, quarter or half samples are actually interpolated sub-samples caused by fractional motion vectors. Based on the vectors and full-samples, the sub-samples can be calculated by using bicubic or bilinear 2-D filtering. See subclause 8.4.2.2 "Fractional sample interpolation process" of the H.264 standard.

WAVELET BASED IMAGE COMPRESSION:

Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves, called *wavelets,* of varying frequency *and limited duration.* This allows them to provide the equivalent of a musical score for an image, revealing not only what notes (or frequencies) to play but also when to play them.

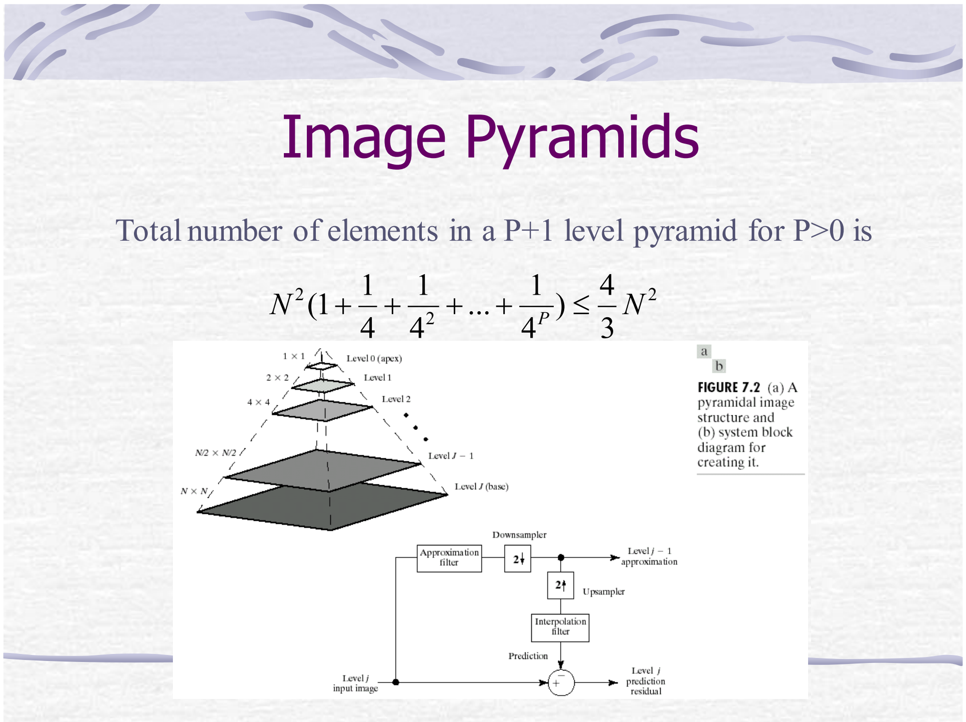
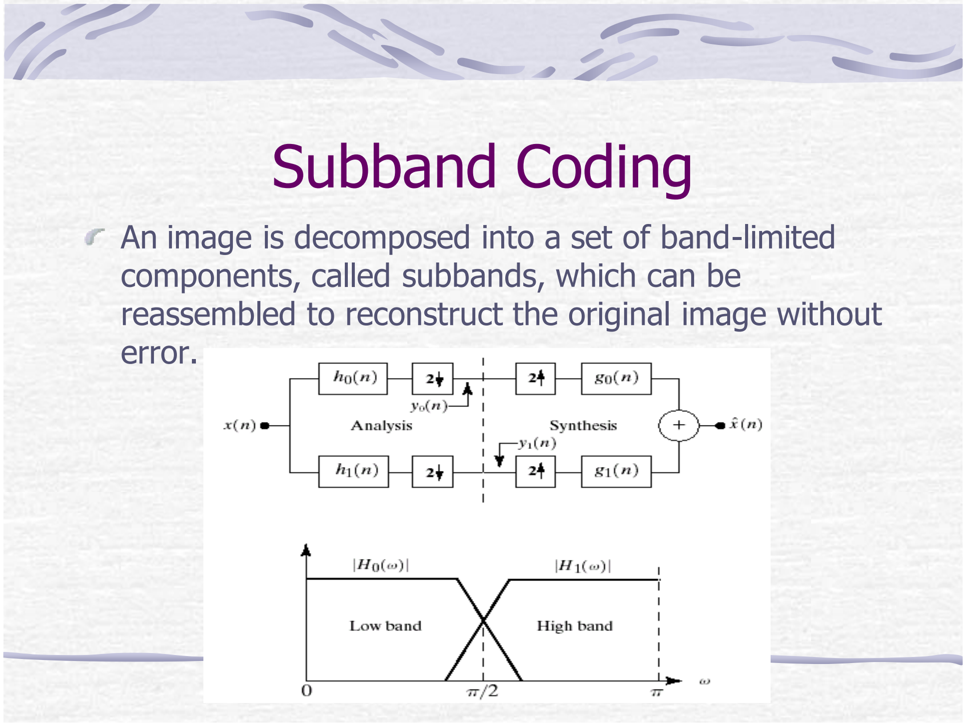


Image Pyramids

Total number of elements in a P+1 level pyramid for P>0 is

$$N^2(1+\frac{1}{4}+\frac{1}{4^2}+...+\frac{1}{4^P}) \le \frac{4}{3}N^2$$

FIGURE 7.2 (a) A pyramidal image structure and (b) system block diagram for creating it.



Subband Coding

- An image is decomposed into a set of band-limited components, called subbands, which can be reassembled to reconstruct the original image without error.

The Haar Transform

- Oldest and simplest known orthonormal wavelets.
- T=HFH where

  F: NXN image matrix,

  H: NxN transformation matrix.
- Haar basis functions $h_k(z)$ are defined over the continuous, closed interval [0,1] for k=0,1,..N-1 where $N=2^n$.

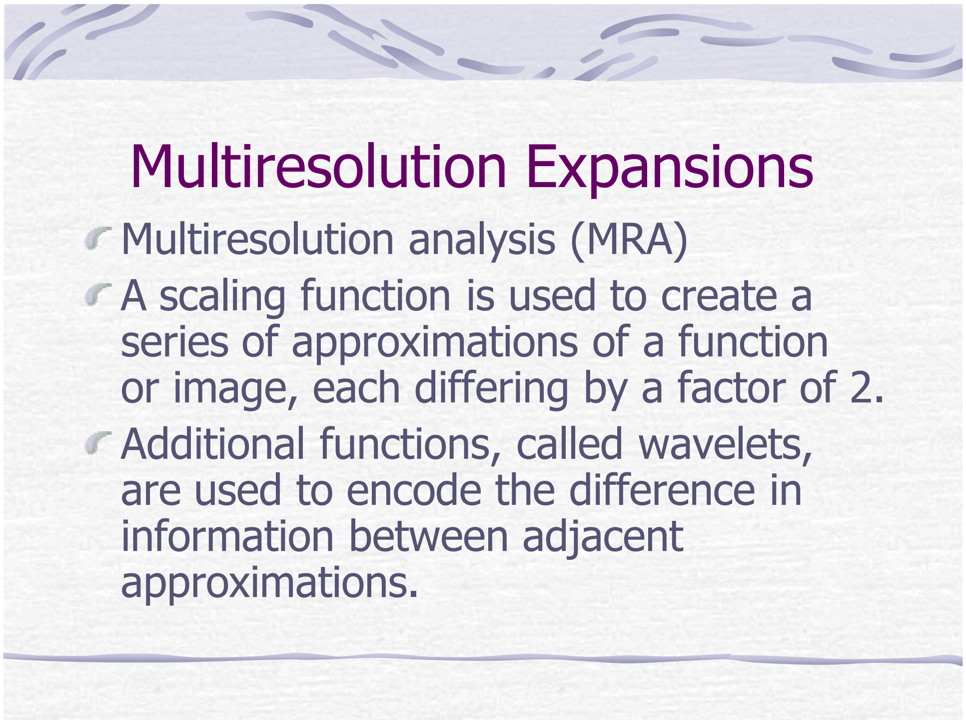
Haar Basis Functions

$$k = 2^p + q - 1 \quad \text{where } 0 \le p \le n-1,$$

$$q = 0 \text{ or } 1 \text{ for } p = 0, 1 \le q \le 2^p \text{ for } p \neq 0$$

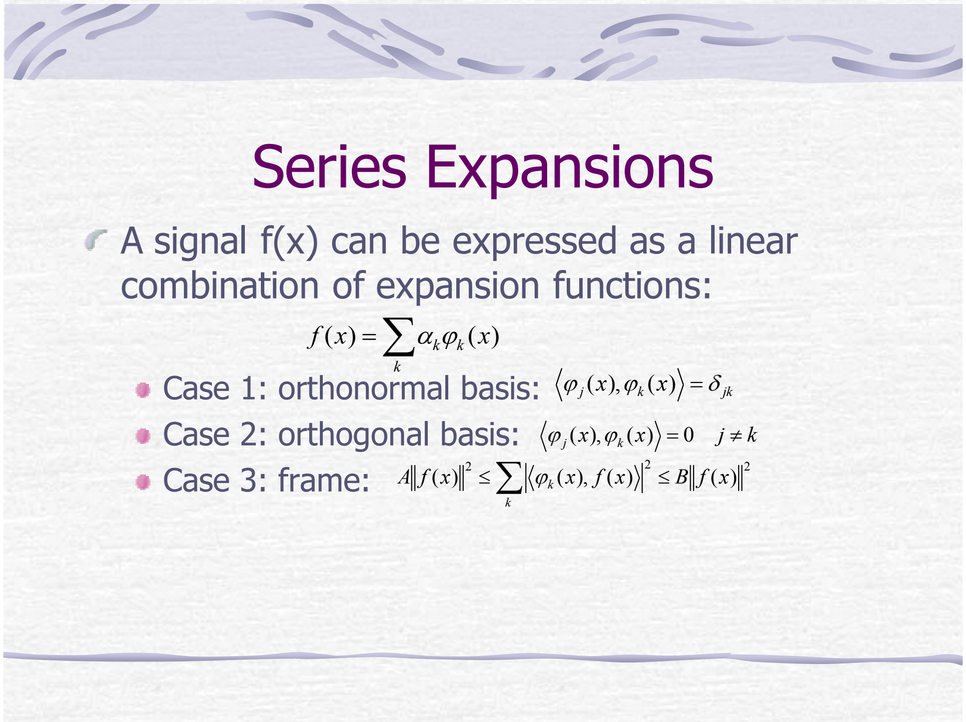
$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \; z \in [0,1]$$

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



Multiresolution Expansions

- Multiresolution analysis (MRA)
- A scaling function is used to create a series of approximations of a function or image, each differing by a factor of 2.
- Additional functions, called wavelets, are used to encode the difference in information between adjacent approximations.



Series Expansions

- A signal f(x) can be expressed as a linear combination of expansion functions:

$$f(x) = \sum_k \alpha_k \varphi_k(x)$$

  - Case 1: orthonormal basis: $\langle \varphi_j(x), \varphi_k(x) \rangle = \delta_{jk}$
  - Case 2: orthogonal basis: $\langle \varphi_j(x), \varphi_k(x) \rangle = 0 \quad j \neq k$
  - Case 3: frame: $A\|f(x)\|^2 \leq \sum_k |\langle \varphi_k(x), f(x) \rangle|^2 \leq B\|f(x)\|^2$

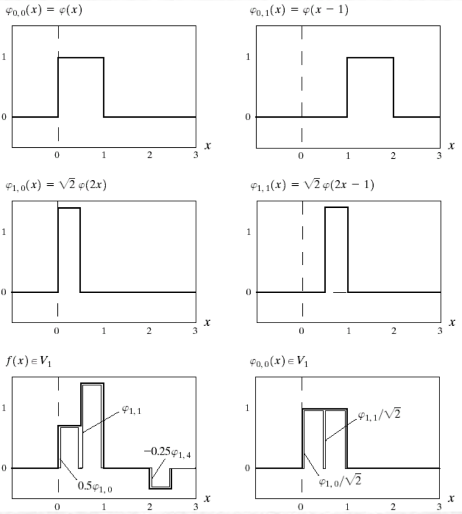
Scaling Functions

- Consider the set of expansion functions composed of integer translations and binary scaling of the real, square-integrable function, $\varphi(x)$,i.e.,

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k)$$

- By choosing φ wisely, $\{\varphi_{j,k}(x)\}$ can be made to span $L^2(\mathbf{R})$

Haar Scaling Function



a b
c d
e f

FIGURE 7.9 Haar scaling functions in V_0 in V_1.

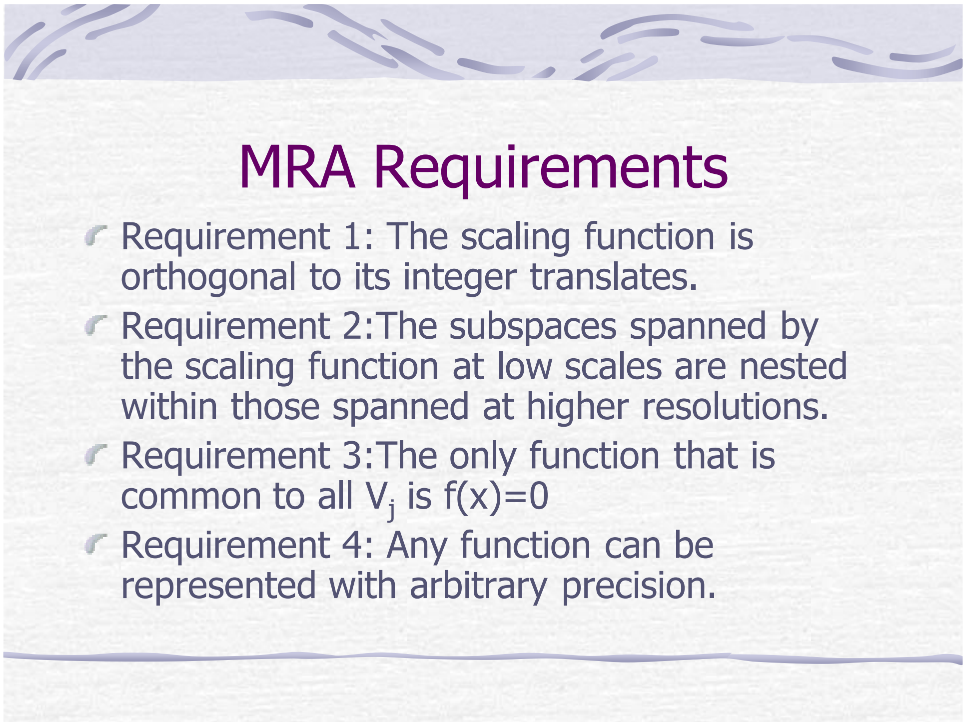
EXPANSION OF FUNCTIONS

MULTIRESOLUTION EXPANSIONS

The previous section introduced three well-known imaging techniques that played an important role in the development of a unique mathematical theory called multiresolution analysis (MRA).ln MRA, a *scaling function* is used to create a series of approximations of a function or image, each differing by a factor of 2 from its nearest neighboring approximations. Additional functions, called *wavelets,* are then used to encode the difference in information between adjacent approximations.

Unlike DCT s and DFT s, which use sinusoidal waves as basis functions, this new variety of transformations use small waves of varying frequency and of limited extent, known as wavelets as basis. The wavelets can be scaled and shifted to analyze the spatial frequency contents of an image at different resolutions and positions. A wavelet can therefore perform analysis of an image at multiple resolutions, making it an effective tool in *multi-resolution analysis* of images. Furthermore, wavelet analysis performs what is known as *space-frequency localization* so that at any specified location in space, one can obtain its details in terms of frequency. It is like placing a magnifying glass above a photograph to explore the details around a specific location. The magnifying glass can be moved up or down to vary the extent of magnification, that is, the level of details and it can be slowly panned over the other locations of the photograph to extract those details.

It is our common observation that the level of details within an image varies from location to location. Some locations contain significant details, where we require finer resolution for analysis and there are other locations, where a coarser resolution representation suffices. A multi-resolution representation of an image gives us a complete idea about the extent of the details existing at different locations from which we can choose our requirements of desired details. Multi-resolution representation facilitates efficient compression by exploiting the redundancies across the resolutions. Wavelet transforms is one of the popular, but not the only approach for multi-resolution image analysis. One can use any of the signal processing approaches to sub-band coding, such as Quadrature Mirror Filters (QMF) in multi-resolution analysis.



MRA Requirements

- Requirement 1: The scaling function is orthogonal to its integer translates.
- Requirement 2:The subspaces spanned by the scaling function at low scales are nested within those spanned at higher resolutions.
- Requirement 3:The only function that is common to all V_j is f(x)=0
- Requirement 4: Any function can be represented with arbitrary precision.

Scaling Functions

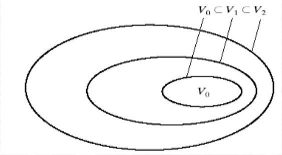
Now consider the set of expansion functions composed of integer translations and binary scalings of the real, square-integrable function *'P(x);* that is, the set *{φ.k (x)}* where

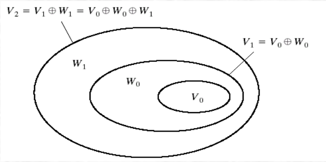
$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k) \tag{7.2-10}$$

for all $j, k \in \mathbf{Z}$ and $\varphi(x) \in L^2(\mathbf{R})$.‡ Here, k determines the position of $\varphi_{j,k}(x)$ along the x-axis, j determines $\varphi_{j,k}(x)$'s width—how broad or narrow it is along the x-axis—and $2^{j/2}$ controls its height or amplitude. Because the shape of $\varphi_{j,k}(x)$ changes with j, $\varphi(x)$ is called a *scaling function*. By choosing $\varphi(x)$ wisely, $\{\varphi_{j,k}(x)\}$ can be made to span $L^2(\mathbf{R})$, the set of all measurable, square-integrable functions.

WAVELET TRANSFORMS IN ONE DIMENSION

Wavelet Functions





Wavelet Functions

- A wavelet function, $\psi(x)$, together with its integer translates and binary scalings, spans the difference between any two adjacent scaling subspace, V_j and V_{j+1}.

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$$

$$\psi\ (x) = \sum_n h_\psi(n)\sqrt{2}\varphi(2x-n) \qquad h_\psi(n) = (-1)^n h_\varphi(1-n)$$

Haar Wavelet Functions

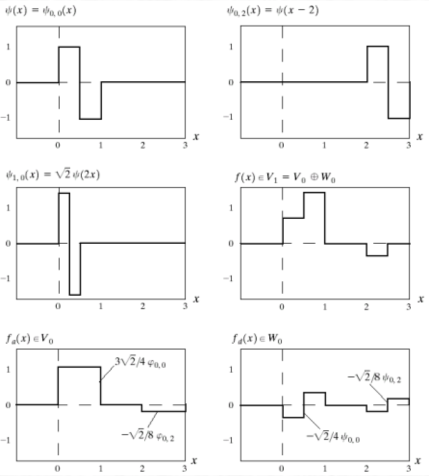


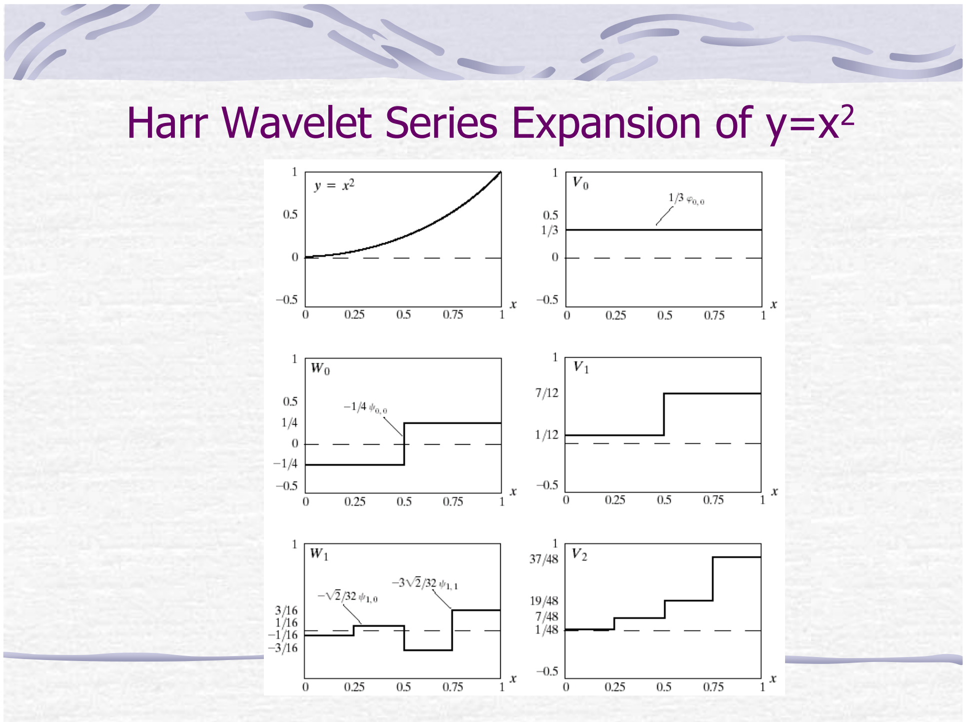
FIGURE 7.12 Haar wavelet functions in W_0 and W_1.

Wavelet Series Expansion

$$f(x) = \sum_{k} c_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_{k} d_j(k)\psi_{j,k}(x)$$

$$c_{j_0}(k) = \int f(x)\varphi_{j_0,k}(x)dx$$

$$d_j(k) = \int f(x)\psi_{j,k}(x)dx$$



Harr Wavelet Series Expansion of $y=x^2$

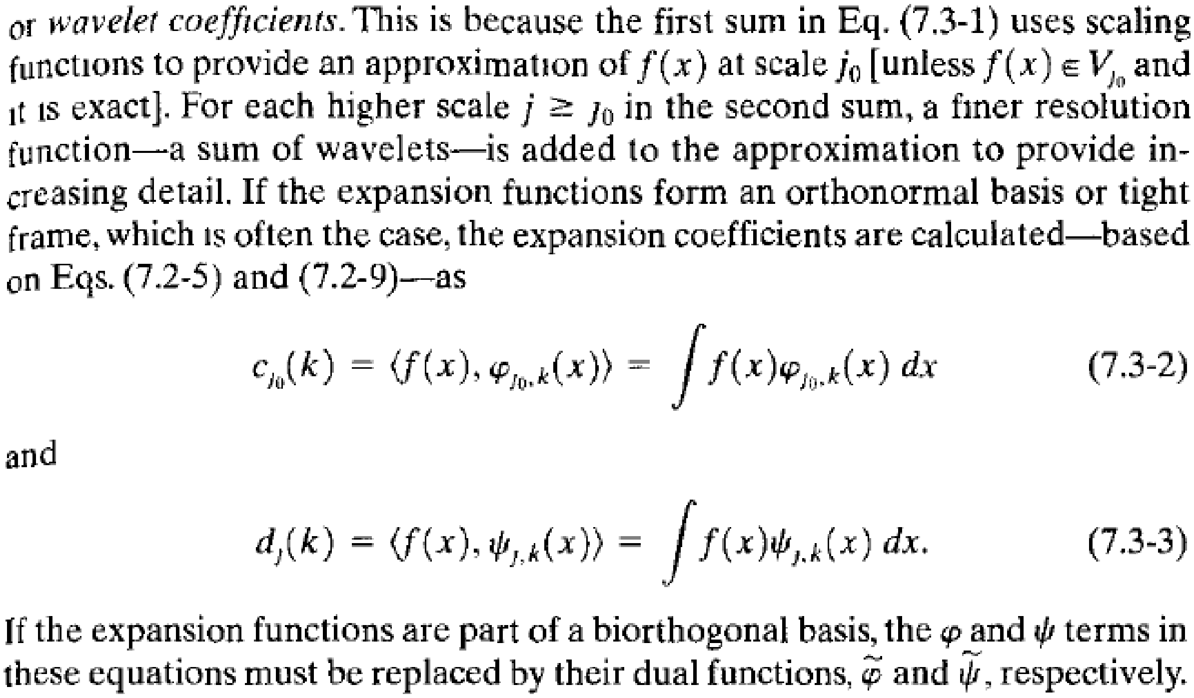
We can now formally define several closely related wavelet transformations: the generalized *wavelet series expansion,* the *discrete wavelet transform,* and the *continuous wavelet transform.* Their counterparts in the Fourier domain are the Fourier series expansion, the discrete Fourier transform, and the integral Fourier transform, respectively. **In** Section 7.4, we will define a computationally efficient implementation of the discrete wavelet transform called the *fast wavelet transform.*

THE WAVELET SERIES EXPANSIONS

We begin by defining the *wavelet series expansion* of function $f(x) \in L^2(\mathbf{R})$ relative to wavelet $\psi(x)$ and scaling function $\varphi(x)$. In accordance with Eq. (7.2-27), we can write

$$f(x) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\psi_{j,k}(x) \qquad (7.3\text{-}1)$$

where j_0 is an arbitrary starting scale and the $c_{j_0}(k)$'s and $d_j(k)$'s are relabeled α_k's from Eqs. (7.2-12) and (7.2-21), respectively. The $c_{j_0}(k)$'s are normally called the *approximation* or *scaling coefficients*; the $d_j(k)$'s are referred to as the *detail*



or *wavelet coefficients*. This is because the first sum in Eq. (7.3-1) uses scaling functions to provide an approximation of $f(x)$ at scale j_0 [unless $f(x) \in V_{j_0}$ and it is exact]. For each higher scale $j \geq j_0$ in the second sum, a finer resolution function—a sum of wavelets—is added to the approximation to provide increasing detail. If the expansion functions form an orthonormal basis or tight frame, which is often the case, the expansion coefficients are calculated—based on Eqs. (7.2-5) and (7.2-9)—as

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x)\varphi_{j_0,k}(x)\,dx \qquad (7.3\text{-}2)$$

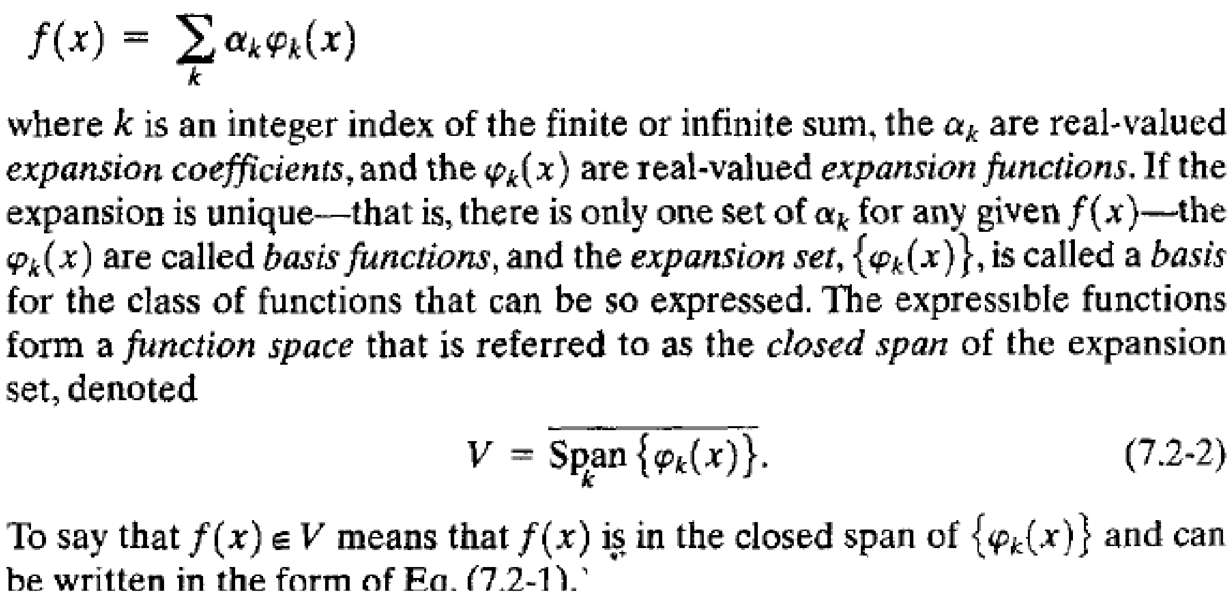
and

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x)\psi_{j,k}(x)\,dx. \qquad (7.3\text{-}3)$$

If the expansion functions are part of a biorthogonal basis, the φ and ψ terms in these equations must be replaced by their dual functions, $\tilde{\varphi}$ and $\tilde{\psi}$, respectively.

SERIES EXPANSIONS

A signal or *functionf(x)* can often be better analyzed as a linear combination of expansion functions



$$f(x) = \sum_k \alpha_k \varphi_k(x)$$

where k is an integer index of the finite or infinite sum, the α_k are real-valued *expansion coefficients*, and the $\varphi_k(x)$ are real-valued *expansion functions*. If the expansion is unique—that is, there is only one set of α_k for any given $f(x)$—the $\varphi_k(x)$ are called *basis functions*, and the *expansion set*, $\{\varphi_k(x)\}$, is called a *basis* for the class of functions that can be so expressed. The expressible functions form a *function space* that is referred to as the *closed span* of the expansion set, denoted

$$V = \overline{\operatorname*{Span}_k \{\varphi_k(x)\}}. \qquad (7.2\text{-}2)$$

To say that $f(x) \in V$ means that $f(x)$ is in the closed span of $\{\varphi_k(x)\}$ and can be written in the form of Eq. (7.2-1).`

THE DISCRETE WAVELET TRANSFORM

Like the Fourier series expansion, the wavelet series expansion of the previous section maps a function of a continuous variable into a sequence of coefficients If the function being expanded is a sequence of numbers. like samples of a continuous function *(x),* the resulting coefficients are called the *discrete wareler*

transform (DWT) of $f(x)$. For this case, the series expansion defined in Eqs (7.3-1) through (7.3-3) becomes the DWT transform pair

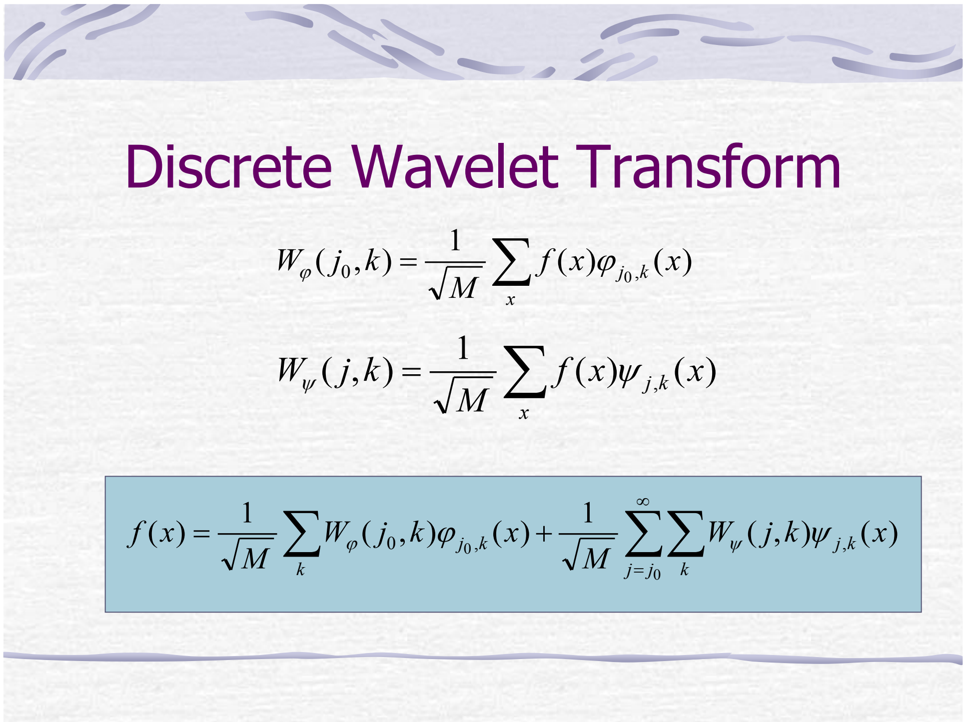
$$W_{\varphi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\varphi_{j_0,k}(x) \tag{7.3-5}$$

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\psi_{j,k}(x) \tag{7.3-6}$$

for $j \geq j_0$ and

$$f(x) = \frac{1}{\sqrt{M}} \sum_{k} W_{\varphi}(j_0, k)\varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k} W_{\psi}(j, k)\psi_{j,k}(x). \tag{7.3-7}$$

Here, $f(x)$, $\varphi_{j_0,k}(x)$, and $\psi_{j,k}(x)$ are functions of the discrete variable $x = 0, 1, 2, \ldots, M - 1$. For example, $f(x) = f(x_0 + x\Delta x)$ for some x_0, Δx, and $x = 0, 1, 2, \ldots, M - 1$. Normally, we let $j_0 = 0$ and select M to be a power of 2 (i.e., $M = 2^J$) so that the summations are performed over $x = 0, 1, 2, \ldots, M - 1$, $j = 0, 1, 2, \ldots, J - 1$, and $k = 0, 1, 2, \ldots, 2^j - 1$. For Haar wavelets, the discretized scaling and wavelet functions employed in the transform (i.e., the basis functions) correspond to the rows of the $M \times M$ Haar transformation matrix of Section 7.1.3. The transform itself is composed of M coefficients, the minimum scale is 0, and the maximum scale is $J - 1$. For reasons noted in Section 7.3.1 and illustrated in Example 7.6, the coefficients defined in Eqs. (7.3-5) and (7.3-6) are usually called *approximation* and *detail coefficients*, respectively.



Discrete Wavelet Transform

$$W_{\varphi}(j_0,k)=\frac{1}{\sqrt{M}}\sum_{x} f(x)\varphi_{j_0,k}(x)$$

$$W_{\psi}(j,k)=\frac{1}{\sqrt{M}}\sum_{x} f(x)\psi_{j,k}(x)$$

$$\boxed{f(x)=\frac{1}{\sqrt{M}}\sum_{k} W_{\varphi}(j_0,k)\varphi_{j_0,k}(x)+\frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_{k} W_{\psi}(j,k)\psi_{j,k}(x)}$$

THE CONTINUOUS WAVELET TRANSFORM

The natural extension of the discrete wavelet transform is the *continuous wavelet transform* (CWT), which transforms a continuous function into a highly redundant function of two continuous variables-translation and scale. The resulting transform is easy to interpret and valuable for time-frequency analysis. Although our interest is in discrete images, it is covered here for completeness. The continuous wavelet transform of a continuous, square-integrable function, f(x,y), relative to a real-valued wavelet, $\psi(x)$. is

$$W_\psi(s, \tau) = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)\, dx \tag{7.3-8}$$

where

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x - \tau}{s}\right) \tag{7.3-9}$$

and s and τ are called *scale* and *translation* parameters, respectively. Given $W_\psi(s, \tau)$, $f(x)$ can be obtained using the *inverse continuous wavelet transform*

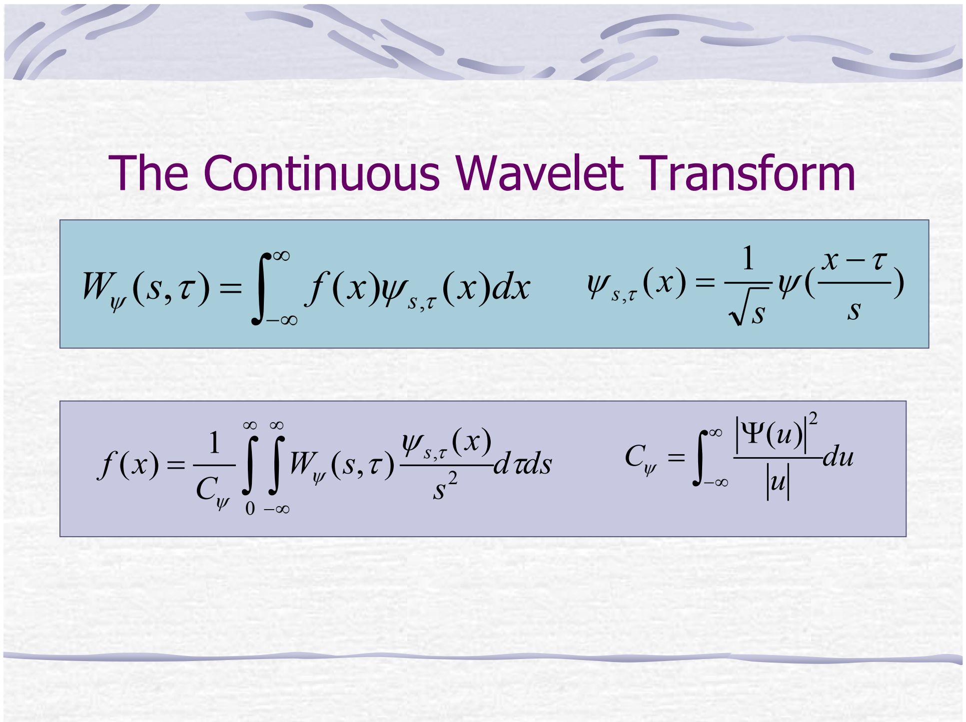
$$f(x) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty} W_\psi(s, \tau)\frac{\psi_{s,\tau}(x)}{s^2}\, d\tau\, ds \tag{7.3-10}$$

where

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(u)|^2}{|u|}\, du \tag{7.3-11}$$

and $\Psi(u)$ is the Fourier transform of $\psi(x)$. Equations (7.3-8) through (7.3-11) define a reversible transformation as long as the so-called *admissibility criterion*, $C_\psi < \infty$, is satisfied (Grossman and Morlet [1984]). In most cases, this simply means that $\Psi(0) = 0$ and $\Psi(u) \to 0$ as $u \to \infty$ fast enough to make $C_\psi < \infty$.

The preceding equations are reminiscent of their discrete counterparts—Eqs. (7.2-19), (7.3-1), (7.3-3), (7.3-6), and (7.3-7). The following similarities should be noted:



The Continuous Wavelet Transform

$$W_{\psi}(s,\tau)=\int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)dx \qquad \psi_{s,\tau}(x)=\frac{1}{\sqrt{s}}\psi(\frac{x-\tau}{s})$$

$$f(x)=\frac{1}{C_{\psi}}\int_{0}^{\infty}\int_{-\infty}^{\infty} W_{\psi}(s,\tau)\frac{\psi_{s,\tau}(x)}{s^{2}}d\tau ds \qquad C_{\psi}=\int_{-\infty}^{\infty}\frac{\left|\Psi(u)\right|^{2}}{\left|u\right|}du$$

THE FAST WAVELET TRANSFORM

The *fast wavelet transform* (FWT) is a computationally efficient implementation of the discrete wavelet transform (DWT) that exploits a surprising but fortunate relationship between the coefficients of the DWT at adjacent scales.

Consider again the multiresolution refinement equation

$$\varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x - n). \tag{7.4-1}$$

Scaling x by 2^j, translating it by k, and letting $m = 2k + n$ gives

$$\begin{aligned}\varphi(2^j x - k) &= \sum_n h_\varphi(n)\sqrt{2}\varphi(2(2^j x - k) - n) \\ &= \sum_m h_\varphi(m - 2k)\sqrt{2}\varphi(2^{j+1}x - m).\end{aligned} \tag{7.4-2}$$

Note that scaling vector h_φ can be thought of as the "weights" used to expand $\varphi(2^j x - k)$ as a sum of scale $j + 1$ scaling functions. A similar sequence of operations—beginning with Eq. (7.2-28)—provides an analogous result for $\psi(2^j x - k)$. That is,

$$\psi(2^j x - k) = \sum_m h_\psi(m - 2k)\sqrt{2}\varphi(2^{j+1}x - m) \tag{7.4-3}$$

where scaling vector $h_\varphi(n)$ in Eq. (7.4-2) is replaced by wavelet vector $h_\psi(n)$ in Eq. (7.4-3).

Wavelet Transforms in Two Dimensions

The one-dimensional transforms of the previous sections are easily extended to two-dimensional functions like images. In two dimensions, a two-dimensional scaling function, $\varphi(x, y)$, and three two-dimensional wavelets, $\psi^H(x, y)$, $\psi^V(x, y)$, and $\psi^D(x, y)$, are required. Each is the product of a one-dimensional scaling function φ and corresponding wavelet ψ. Excluding products that produce one-dimensional results, like $\varphi(x)\psi(x)$, the four remaining products produce the *separable* scaling function

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{7.5-1}$$

and separable, "directionally sensitive" wavelets

$$\psi^H(x, y) = \psi(x)\varphi(y) \tag{7.5-2}$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \tag{7.5-3}$$

$$\psi^D(x, y) = \psi(x)\psi(y). \tag{7.5-4}$$

These wavelets measure functional variations—intensity or gray-level variations for images—along different directions: ψ^H measures variations along columns (for example, horizontal edges), ψ^V responds to variations along rows (like vertical edges), and ψ^D corresponds to variations along diagonals. The directional sensitivity is a natural consequence of the separability imposed by

Eqs. (7.5-2) to (7.5-4); it does not increase the computational complexity of the two-dimensional transform discussed in this section.

Given separable two-dimensional scaling and wavelet functions, extension of the one-dimensional DWT to two dimensions is straightforward. We first define the scaled and translated basis functions:

$$\varphi_{j,m,n}(x, y) = 2^{j/2}\varphi(2^j x - m, 2^j y - n), \tag{7.5-5}$$

$$\psi^i_{j,m,n}(x, y) = 2^{j/2}\psi^i(2^j x - m, 2^j y - n), \qquad i = \{H, V, D\} \tag{7.5-6}$$

where index i identifies the directional wavelets in Eqs. (7.5-2) to (7.5-4). Rather than an exponent, i is a superscript that assumes the values H, V, and D. The discrete wavelet transform of function $f(x, y)$ of size $M \times N$ is then

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\varphi_{j_0,m,n}(x, y) \tag{7.5-7}$$

$$W^i_\psi(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\psi^i_{j,m,n}(x, y) \qquad i = \{H, V, D\}. \tag{7.5-8}$$

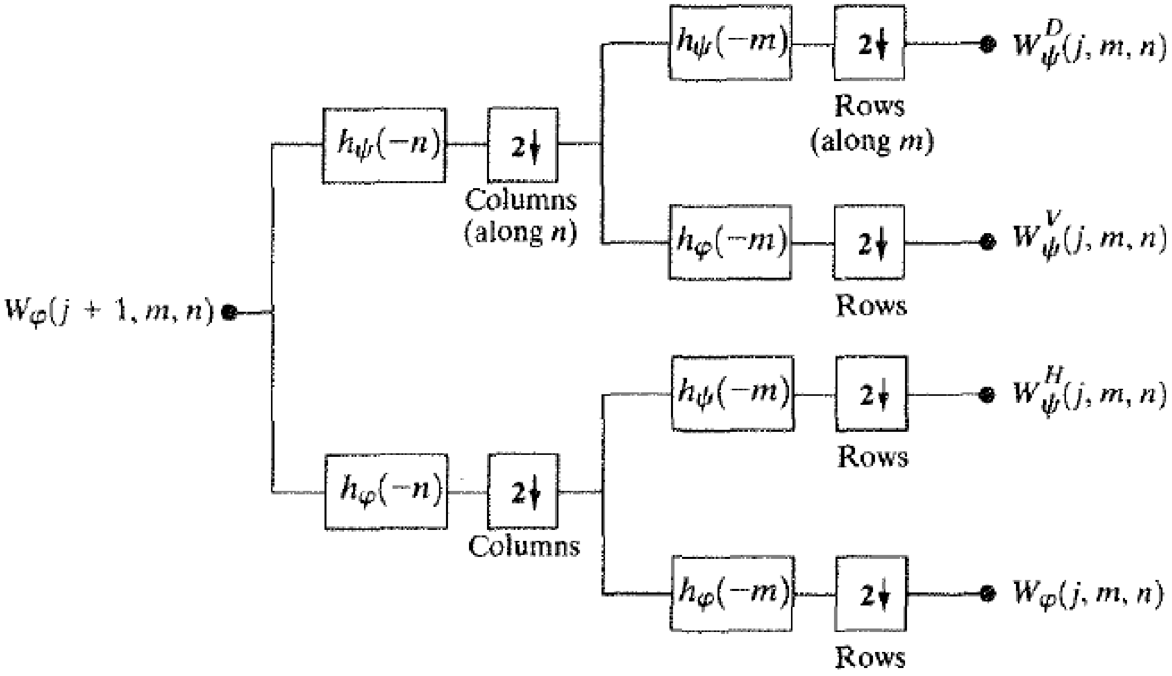
As in the one-dimensional case, j_0 is an arbitrary starting scale and the $W_\varphi(j_0, m, n)$ coefficients define an approximation of $f(x, y)$ at scale j_0. The $W^i_\psi(j, m, n)$ coefficients add horizontal, vertical, and diagonal details for scales $j \geq j_0$. We normally let $j_0 = 0$ and select $N = M = 2^J$ so that $j = 0, 1, 2, \ldots, J - 1$ and $m, n = 0, 1, 2, \ldots, 2^j - 1$. Given the W_φ and W^i_ψ of Eqs. (7.5-7) and (7.5-8), $f(x, y)$ is obtained via the inverse discrete wavelet transform

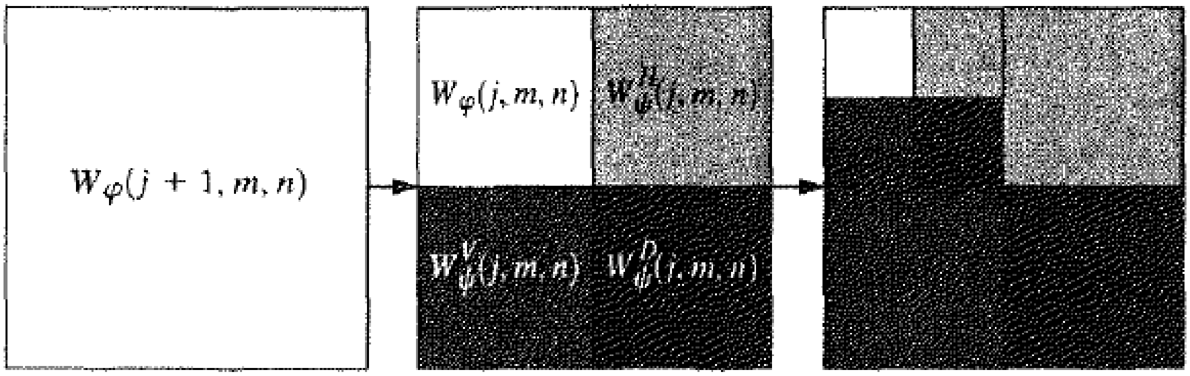
$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n)\varphi_{j_0,m,n}(x, y) \tag{7.5-9}$$

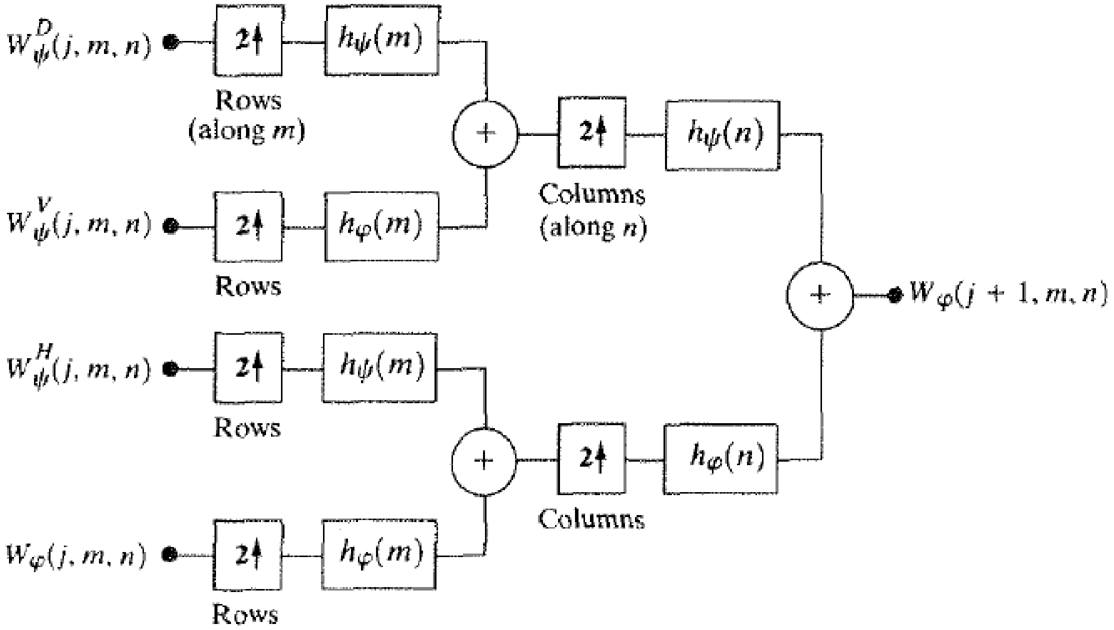
$$+ \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W^i_\psi(j, m, n)\psi^i_{j,m,n}(x, y).$$

Like the one-dimensional discrete wavelet transform, the two-dimensional DWT can be implemented using digital filters and downsamplers. With separable two-dimensional scaling and wavelet functions, we simply take the one-dimensional FWT of the rows of $f(x, y)$, followed by the one-dimensional FWT of the resulting columns. Figure 7.22(a) shows the process in block diagram form. Note that, like its one-dimensional counterpart in Fig. 7.15, the two-dimensional FWT "filters" the scale $j + 1$ approximation coefficients to construct the scale j approximation and detail coefficients. In the two-dimensional case, however, we get three sets of detail coefficients—the horizontal, vertical, and diagonal details.

The single-scale filter bank of Fig. 7.22(a) can be "iterated" (by tying the approximation output to the input of another filter bank) to produce a P scale transform in which scale $j = J - 1, J - 2, \ldots, J - P$. As in the one-dimensional case, image $f(x, y)$ is used as the $W_\varphi(J, m, n)$ input. Convolving its rows with $h_\varphi(-n)$ and $h_\psi(-n)$ and downsampling its columns, we get two subimages whose horizontal resolutions are reduced by a factor of 2. The highpass or detail component characterizes the image's high-frequency information with vertical orientation; the lowpass, approximation component contains its low-frequency, vertical information. Both subimages are then filtered columnwise and downsampled to yield four quarter-size output subimages— W_φ, W^H_ψ, W^V_ψ, and W^D_ψ.


$h_\psi(-m)$
$2\downarrow$
$W_\psi^D(j, m, n)$
Rows
(along m)
$h_\psi(-n)$
$2\downarrow$
Columns
(along n)
$h_\varphi(-m)$
$2\downarrow$
$W_\psi^V(j, m, n)$
Rows
$W_\varphi(j + 1, m, n)$
$h_\psi(-m)$
$2\downarrow$
$W_\psi^H(j, m, n)$
Rows
$h_\varphi(-n)$
$2\downarrow$
Columns
$h_\varphi(-m)$
$2\downarrow$
$W_\varphi(j, m, n)$
Rows


$W_\varphi(j + 1, m, n)$
$W_\varphi(j, m, n)$
$W_\psi^H(j, m, n)$
$W_\psi^V(j, m, n)$
$W_\psi^D(j, m, n)$



a
b
c

FIGURE 7.22 The two-dimensional fast wavelet transform: (a) the analysis filter bank; (b) the resulting decomposition; and (c) the synthesis filter bank.

These subimages, which are shown in the middle of Fig. 7.22(b), are the inner products of $f(x, y)$ and the two-dimensional scaling and wavelet functions in Eqs. (7.5-1) through (7.5-4), followed by downsampling by two in each dimension. Two iterations of the filtering process produces the two-scale decomposition at the far right of Fig. 7.22(b).

Figure 7.22(c) shows the synthesis filter bank that reverses the process described above. As would be expected, the reconstruction algorithm is similar to the one-dimensional case. At each iteration, four scale j approximation and detail subimages are upsampled and convolved with two one-dimensional filters—one operating on the subimages' columns and the other on its rows. Addition of the results yields the scale $j + 1$ approximation, and the process is repeated until the original image is reconstructed.

We conclude the section with two examples that demonstrate the usefulness of wavelets in image processing. As in the Fourier domain, the basic approach is-to

1. Compute the two-dimensional wavelet transform of an image.
2. Alter the transform.
3. Compute the inverse transform.

Because the DWT's scaling and wavelet vectors are used as lowpass and highpass filters. most Fourier based filtering techniques have an equivalent "wavelet domain" counterpart.

The general wavelet-based procedure for *denoising* the image (i.e., suppressing the noise part) is as follows:

1. Choose a wavelet (e.g., Haar, symlet, ...) and number of levels or scales, *P,* for the decomposition. Then compute the FWT of the noisy image.

2. Threshold the detail coefficients-That is, select and apply a threshold to the detail coefficients from scales *J* - 1 to *J* - *P.* This can be accomplished by *hard thresholding,* which means setting to zero the elements whose absolute values are lower than the threshold, or by *soft thresholding,* which involves first setting to zero the elements whose absolute values are lower than the threshold and then scaling the nonzero coefficients toward zero. Soft thresholding eliminates the discontinuity (at the threshold) that is inherent in hard thresholding.

3. Perform a wavelet reconstruction based on the original approximation coefficients at level *J* - *P* and the modified detail coefficients for level.

JPEG 2000

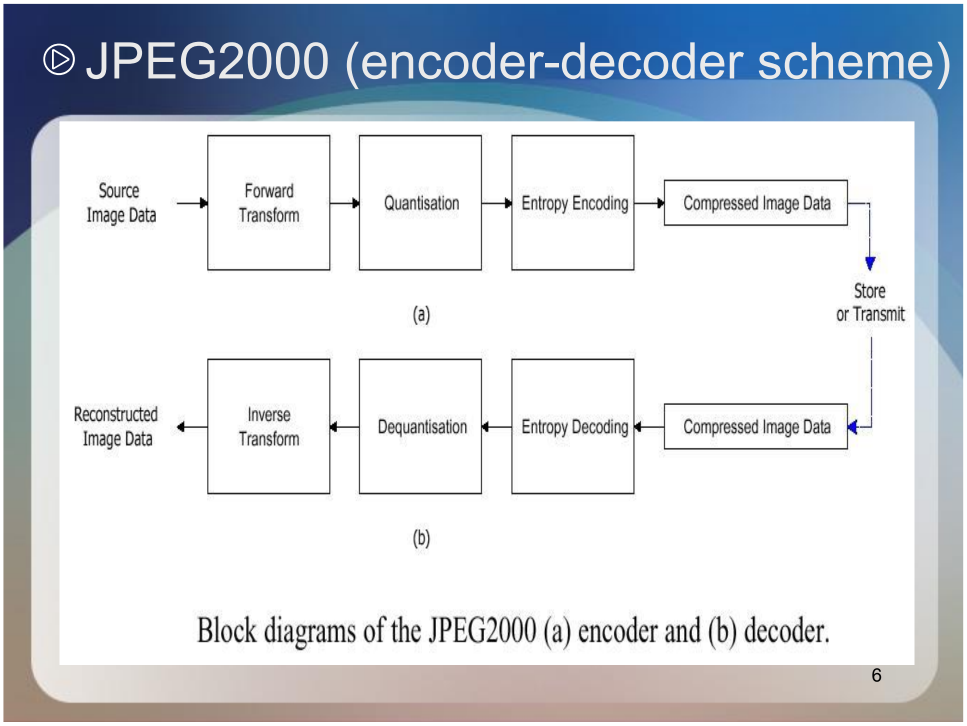
Although not yet formally adopted, JPEG 2000 extends the initial JPEG standard to provide increased flexibility in both the compression of continuous tone still images and access to the compressed data. For example, portions of a JPEG 2000 compressed image can be extracted for retransmission, storage, display, and/or editing. The standard is based on the wavelet coding techniques of Section 8.5.3. Coefficient quantization is adapted to individual scales and subbands and the quantized coefficients are arithmetically coded on a bit-plane basis (see Section *804).* Using the notation of the standard. an image is encoded as follows (ISO/IEC [2000]).

The first step of the encoding process is to DC level shift the samples of the Ssiz-bit unsigned image to be coded by subtracting *28"z-I.If* the image has more than one *component-like* the red, green, and blue planes of a color imageeach component is individually shifted. If there are exactly three components, they may be optionally decorrelated using a reversible or nonreversible linear combination of the components. The *irreversible component transform* of the standard, for example, is

$$
\begin{aligned}
Y_0(x, y) &= 0.299I_0(x, y) + 0.587I_1(x, y) + 0.114I_2(x, y) \\
Y_1(x, y) &= -0.16875I_0(x, y) - 0.33126I_1(x, y) + 0.5I_2(x, y) \quad (8.6\text{-}1) \\
Y_2(x, y) &= 0.5I_0(x, y) - 0.41869I_1(x, y) - 0.08131I_2(x, y)
\end{aligned}
$$

where I_0,.I_1 and I_2 are the level-shifted input components and Yo. YI , and Y2 are the corresponding decorrelated components. If the input components are the red, green, and blue planes of a color image, Eq. (8.6-1) approximates the *R'C'B'* to *Y'ChC,* color video transform (Poynton [1996]).' The goal of the transformation is to improve compression efficiency; transformed components YI and Y2 are difference images whose histograms are highly peaked around zero.

After the image has been level shifted and optionally decorrelated. its components are optionally divided into *tiles.* Tiles are rectangular arrays of pixels that contain the same relative portion of all components. Thus, the tiling process creates *tile components* that can be extracted and reconstructed independently, providing a simple mechanism for accessing and/or manipulating a limited re' gion of a coded image.
The one-dimensional discrete wavelet transform of the rows and columns of each tile component is then computed.



JPEG2000 (encoder-decoder scheme)

Source Image Data → Forward Transform → Quantisation → Entropy Encoding → Compressed Image Data

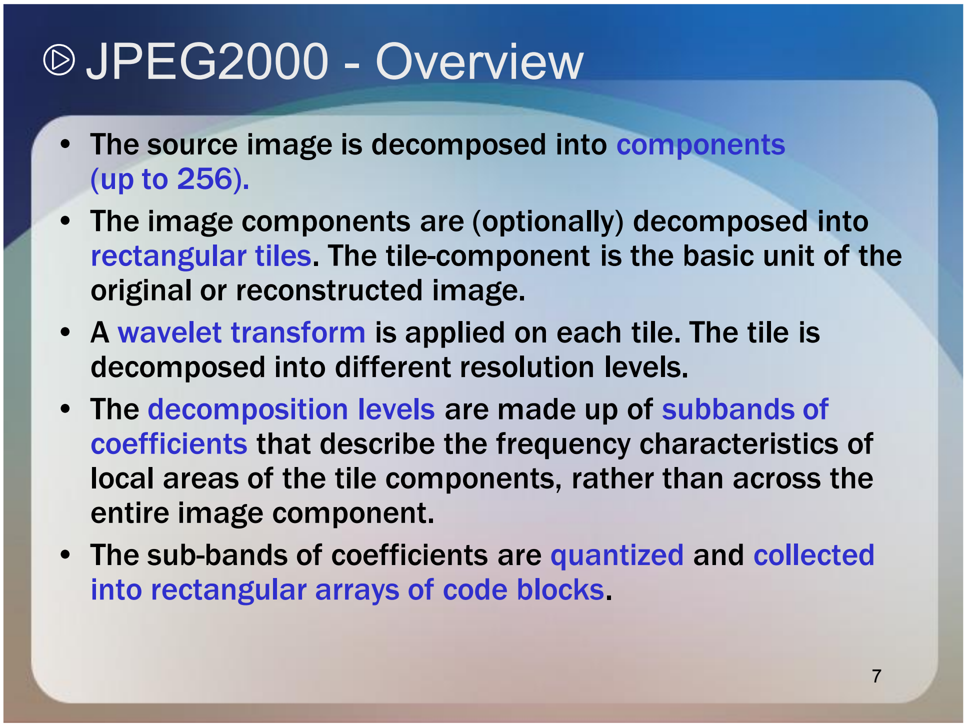
(a)

Store or Transmit

Reconstructed Image Data ← Inverse Transform ← Dequantisation ← Entropy Decoding ← Compressed Image Data

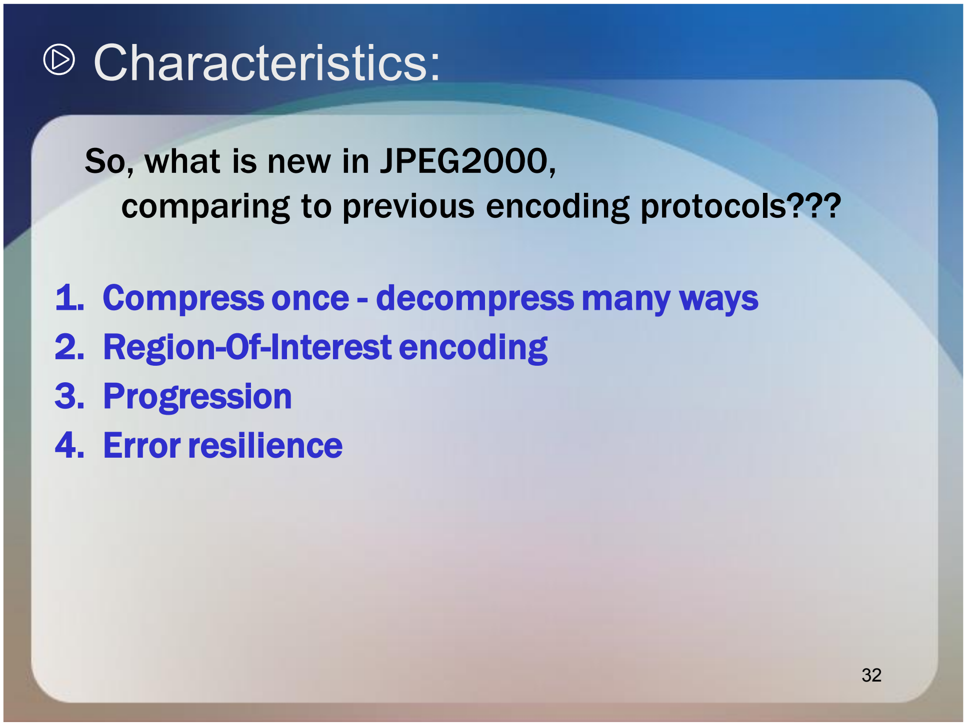
(b)

Block diagrams of the JPEG2000 (a) encoder and (b) decoder.



JPEG2000 - Overview

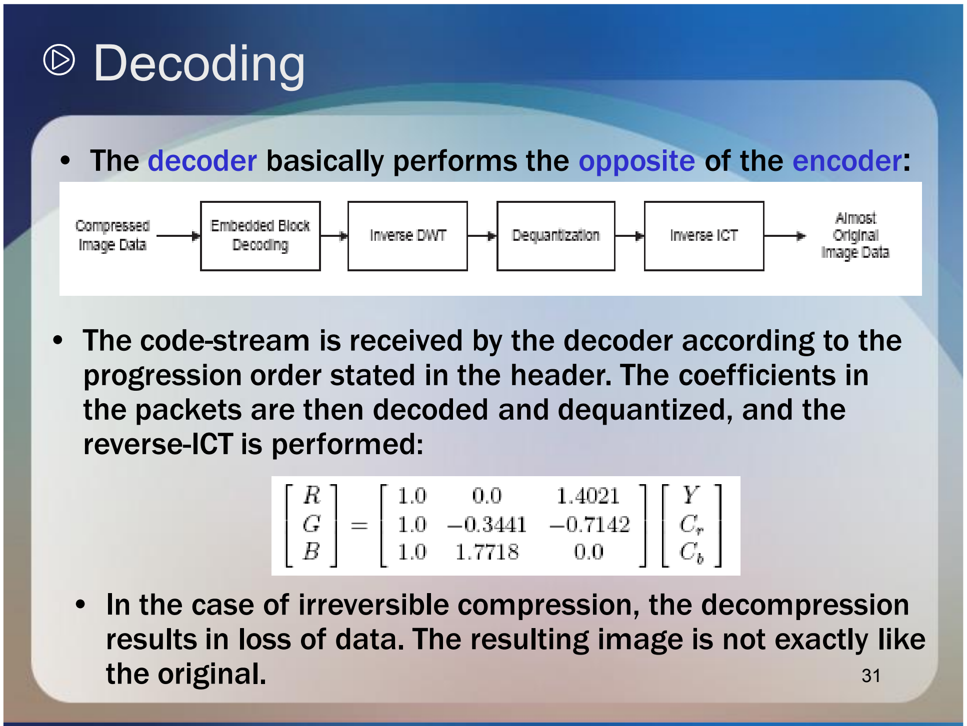
- The source image is decomposed into components (up to 256).
- The image components are (optionally) decomposed into rectangular tiles. The tile-component is the basic unit of the original or reconstructed image.
- A wavelet transform is applied on each tile. The tile is decomposed into different resolution levels.
- The decomposition levels are made up of subbands of coefficients that describe the frequency characteristics of local areas of the tile components, rather than across the entire image component.
- The sub-bands of coefficients are quantized and collected into rectangular arrays of code blocks.



Characteristics:

So, what is new in JPEG2000,
comparing to previous encoding protocols???

1. **Compress once - decompress many ways**
2. **Region-Of-Interest encoding**
3. **Progression**
4. **Error resilience**



Decoding

- The decoder basically performs the opposite of the encoder:

Compressed Image Data → Embedded Block Decoding → Inverse DWT → Dequantization → Inverse ICT → Almost Original Image Data

- The code-stream is received by the decoder according to the progression order stated in the header. The coefficients in the packets are then decoded and dequantized, and the reverse-ICT is performed:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.4021 \\ 1.0 & -0.3441 & -0.7142 \\ 1.0 & 1.7718 & 0.0 \end{bmatrix} \begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix}$$

- In the case of irreversible compression, the decompression results in loss of data. The resulting image is not exactly like the original.

JPEG2000 - Markets & Applications

- Internet
- Mobile
- Printing
- Scanning
- Digital Photography
- Remote Sensing
- Facsimile
- Medical
- Digital Libraries
- E-Commerce

DIGITAL IMAGE WATERMARKING

- Watermark--an invisible signature embedded inside an image to show authenticity or proof of ownership
- Discourage unauthorized copying and distribution of images over the internet

- Ensure a digital picture has not been altered
- Software can be used to search for a specific watermark

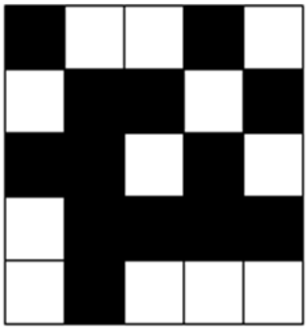
Types of Watermarking:
The digital watermarking is of four types and they are as follows:

- Visible
- Invisible
- Fragile
- Public

1. **Visible watermarking:** the visible watermarking have the visible message or the logo of the company which represents the ownership of the message, removing of the logo and meddling of the logo breaks the law of copyright.
2. **Invisible watermarking:** the invisible watermarking cannot be seen on the original images and the picture looks like an original image though it has the watermark. The invisible watermark can be extracted with the detection procedure or with any watermark extraction method.
3. **Fragile watermark:** the fragile watermarks can be demolished by the data manipulation and these are also called as tamper proof watermark. The fragile watermark has the ability to detect the modifications in the signal and also recognizes the place where the modifications have occurred and also the signal before the change.
4. **Public watermark:** The public watermark does not have the protection and these can be read by everyone by availing the unique algorithm.

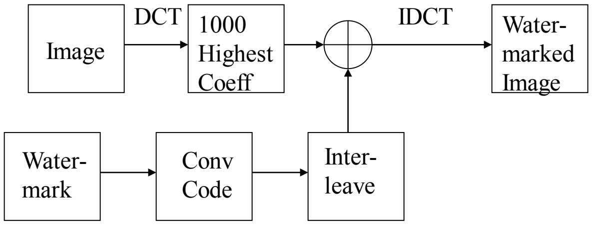
Watermark Properties

- Watermark should appear random, noise-like sequence
- Appear Undetectable
- Good Correlation Properties
  High correlation with signals similar to watermark
  Low correlation with other watermarks or random noise
- Common sequences
  A) Normal distribution
  B) m-sequences



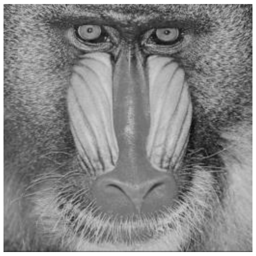
$$W=\begin{bmatrix}1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

Setup-Watermark Embedding

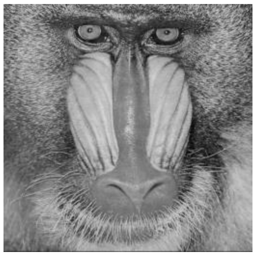


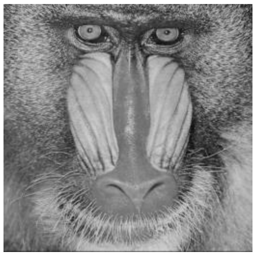
- •DC Component Excluded for 1000 Highest Coefficients
- •Interleaving prevents burst errors
- •Watermarked Image Similar to original image
- •Without coding, ignore Conv Code and Interleave block

Original Image



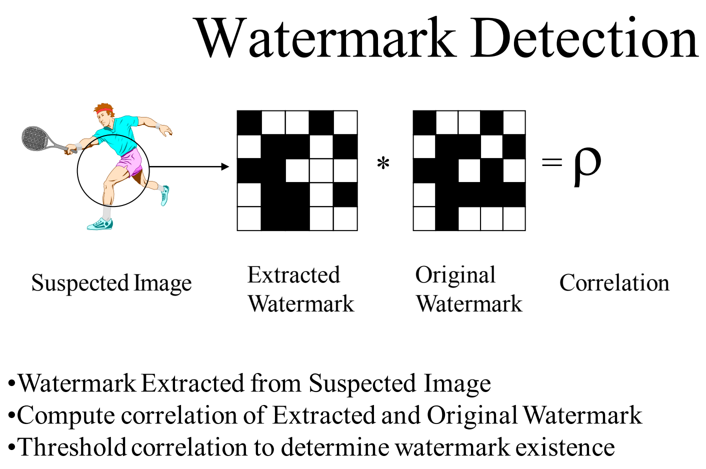
Watermarked Image, No Coding





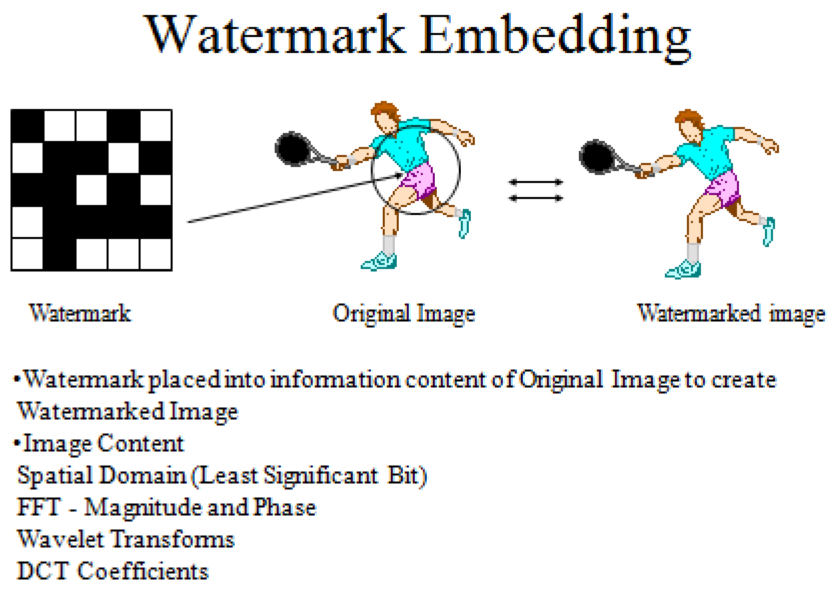
Watermarked Image with Coding

- •512x512 “Mandrill” Image
- •See Handout
- •Both watermarks imperceptible
- •Alterations to original image difficult to notice



Watermark Detection

- Watermark Extracted from Suspected Image
- Compute correlation of Extracted and Original Watermark
- Threshold correlation to determine watermark existence

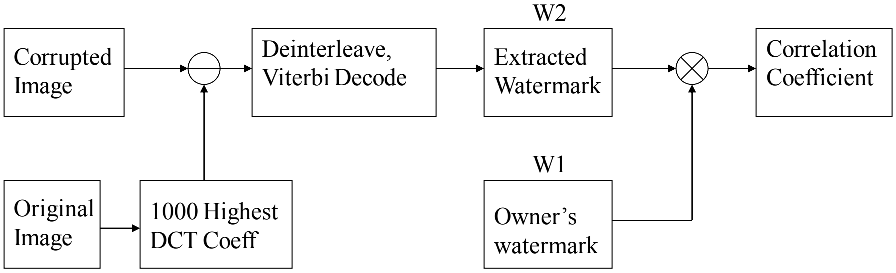


Watermark Embedding

- Watermark placed into information content of Original Image to create Watermarked Image
- Image Content

Spatial Domain (Least Significant Bit)
FFT - Magnitude and Phase
Wavelet Transforms
DCT Coefficients

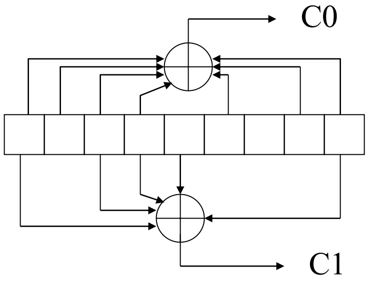
Watermark Detection



- •For no coding, deinterleave and decode block ignored
- •ρ=E[W1*W2]/{ E[W1^2]E[W2^2]}
- •If W1=W2 then ρ=1
- •if W1 and W2 are independent, then ρ=0 if E[W1]=0
- •Corruptions are additive noise, JPEG Compression Image scaling, and UnZign

Convolutional Codes

Input=[...1011010101100000000]
G0 = [1 1 1 1 0 1 0 1 1]
G1 = [1 0 1 1 1 0 0 0 1]



- •Output C0 = conv(G0,Input); Output C1=conv(G1,Input)
- •Convolutional code implemented using linear shift registers
- •Adds redundancy for error-correction
- •Encoding/Decoding well researched
- •Good coding performance, very popular

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III BSC CS COURSE NAME: DIGITAL IMAGE PROCESSING
COURSE CODE: 16CSU502B BATCH-2016-2019

UNIT IV

SNO	QUESTIONS	CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	ANSWER
1	___________ is an encoding format used for fax transmission.	Fax Group	ITU-T	JPEG	TIFF	Fax Group
2	JBIG stands for ____________	Joint Bi-level Image Experts Group	Joint Bit Image Experts Group	Joint Bi-level Image Encoding Group	Joint Bit Image Encoding Group	Joint Bi-level Image Experts Group
3	____________ is an effective technique for reducing an image's interpixel redundancies is to process the image's bit planes individually	Huffman coding	bit-plane coding	Arithmetic coding	Golomb coding	bit-plane coding
4	The overall process of truncating, quantizing, and coding the coefficients of a transformed subimage is commonly called ___________	bit-plane coding	zonal coding	threshold coding	bit allocation	bit allocation
5	______________ is based on the information the	Zonal coding	bit-plane coding	Arithmetic coding	threshold coding	Zonal coding
6	JPEG stands for _______________	Joint photographic experts group	Joint photographic encoding group	Joint phase experts group	Joint photo encode group	Joint photographic experts group
7	is a method to reduce the redundancies in image representation in order to decrease data storage requirements.	Image enhancement	Image compression	Image segmentation	Image restoration	Image compression
8	__________ consume large amount of data and requires very high bandwidth networks in transmission	video or image files	email	file attachment	text	video or image files
9	___________supports one-dimensional image compression of black and white images, on a standard fax machine.	Fax Group 5	Fax Group 4	Fax Group 3	Fax Group 6	Fax Group 3
10	________________ supports two-dimensional image compression, compressing the line width as well as the line length.	Fax Group 5	Fax Group 4	Fax Group 3	Fax Group 6	Fax Group 4

11	____________ is an algorithmic technique used to predict a frame in a video, given the previous and/or future frames.	Motion compensation	Huffman coding	bit-plane coding	Arithmetic coding	Motion compensation
12	___________ transforms a continuous function into a highly redundant function of two continuous variables-translation and scale.	Discrete wavelet transform	Continuous wavelet transform	Fast Wavelet transform	Fourier transform	Continuous wavelet transform
13	_____________ is based on the concept of compromising the accuracy of the reconstructed image m exchange for increased compression.	lossy encoding	lossless encoding	lzw encoding	transform coding	lossy encoding
14	______________ is error-free compression approach that does not require decomposition of an image into a collection of bit planes	lossy encoding	lossless encoding	lzw encoding	transform coding	lossless encoding
15	An image is decomposed into a set of band-limited components called ___________	subbands	frequency band	bandwidth	bitplane	subbands
16	____________ absorbs the nearest integer function of the error-free encoder, IS inserted between the symbol encoder and the point at which the prediction error is formed.	quantizer	encoder	decoder	predictor	quantizer
17	Orthonormal filter is developed around filter called ___________	up sampling	filtering	Digital segment filtering	prototype	prototype
18	The base of image pyramid contains ___________	low resolution	high resolution	intensity	blurred portion	high resolution
19	FWT stands for ___________	Fast wavelet transformation	Fast wavelet transform	Fourier wavelet transform	Fourier wavelet transformation	Fast wavelet transform
20	Wavelet series equation is the sum of ___________	scaling coefficient	detail coefficient	span coefficient	Both scaling and detail coefficient	Both scaling and detail coefficient
21	DWT stands for ___________	Discrete wavelet transform	Discrete wavelet transformation	Digital wavelet transform	Digital wavelet transformation	Discrete wavelet transform
22	In MRA, a _______ is used to create a series of approximations of a function or image.	scaling function	signal function	series function	wavelet function	scaling function

23	Scaling vectors are taken as ___________	heights	sharpness	intensity	weights	weights
24	MRA equation is also called ___________	modulating equation	FIR filter	dilation equation	span equation	dilation equation
25	Subspaces spanned are nested at ___________	lower scales	higher scales	mid scales	intense scales	higher scales
26	Prediction residual pyramid is computed in ___________	2 steps	3 steps	4 steps	5 steps	3 steps
27	K multiplication constants in digital filters are called ___________	co efficient	multipliers	subtractors	filter coefficients	filter coefficients
28	The size of the base image will be ___________	N-1 x N-1	N+1 x N-1	N-1 x N	N x N	N x N
29	MRA stands for ___________	Multiresolution analysis	Multiresolution assembly	Multiresemble analysis	Multiresemble assembly	Multiresolution analysis
30	Discarding every sample is called ___________	up sampling	filtering	down sampling	blurring	down sampling
31	Images are ___________	1D arrays	2D arrays	3D arrays	4D arrays	2D arrays
32	Filter banks consists of ___________	1 FIR filter	2 FIR filters	3 FIR filters	4 FIR filters	2 FIR filters
33	Neighborhood averaging produces ___________	histogram	pyramids	mean pyramids	equalized histogram	mean pyramids
34	Decomposing image into band limit components is called ___________	low coding	high coding	intense coding	subband coding	subband coding
35	Narrow wavelets represents ___________	sharp details	finer details	blur details	edge details	finer details
36	High contrast images are considered as ___________	low resolution	high resolution	intense	blurred	low resolution
37	Function having compact support is ___________	histogram	pyramids	mean pyramids	haar function	haar function
38	In multiresolution processing * represents the ___________	complete conjugate operation	complex conjugate operation	complete complex operation	complex complex operation	complex conjugate operation
39	The apex of image pyramid contains ___________	low resolution	high resolution	intensity	blurred portion	low resolution
40	Representing image in more than one resolution is ___________	histogram	image pyramid	local histogram	equalized histogram	image pyramid
41	Moving up in pyramid the size ___________	increases	remain same	decreases	blurred	decreases
42	The scaling function is ___________	pentagonal	square	orthogonal	oval	orthogonal

43	CWT stands for ___________	Complete wavelet transform	Complex wavelet transform	Continuous wavelet transform	Continuous wavelet transformation	Continuous wavelet transform
44	Integer wavelet translates are ___________	pentagonal	square	orthogonal	oval	orthogonal
45	Low contrast images are considered as ___________	low resolution	high resolution	intense	blurred	high resolution
46	Processing the image in small parts is ___________	histogram	pyramids	local histogram	equalized histogram	local histogram
47	Function can be represented with ___________	arbitrary precision	filtering	down sampling	prototype	arbitrary precision
48	One that is not a part of digital filter ___________	unit delay	multiplier	subtractor	adder	subtractor
49	The function changing the shape is called ___________	scaling function	shaping function	down sampling	blurring	scaling function
50	Function space is referred to as ___________	open span	fully span	closed span	span	closed span
51	No filtering produces ___________	Gaussian pyramids	pyramids	mean pyramids	subsampling pyramids	subsampling pyramids
52	Diagonally opposed filters is said to be ___________	modulation	multiplier	cross modulation	subband coding	cross modulation
53	The image pyramid contains ___________	j levels	j-1 levels	j+1 levels	n levels	j levels
54	Subband of input image, showing a(m,n) is called ___________	approximation	vertical detail	horizontal detail	diagonal detail	approximation
55	Decomposition in subband coding is performed to	segment image	reconstruct image	blur image	sharpened image	reconstruct image
56	Lowpass Gaussian filtering produces _______	Gaussian pyramids	pyramids	mean pyramids	equalized histogram	Gaussian pyramids
57	_____________ an invisible signature embedde	Watermark	compression	enhancement	restoration	Watermark
58	___________cannot be seen on the original images and the picture looks like an original image though it has the watermark.	invisible watermarking	visible watermarking	fragile watermarking	public watermarking	invisible watermarking
59	__________watermarks can be demolished by the data manipulation and these are also called as tamper proof watermark.	invisible watermarking	visible watermarking	fragile watermarking	public watermarking	fragile watermarking

60	_____________ does not have the protection and these can be read by everyone by availing the unique algorithm.	invisible watermarking	visible watermarking	fragile watermarking	public watermarking	public watermarking

UNIT V

SYLLABUS

Morphological Image Processing: Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion. Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm, Use of motion in segmentation

Introduction to Image Morphing

The problem of creating a smooth transition from one object to another object is called morphing. More specifically, the problem of creating a smooth transition from one image to another image is called image morphing. In other words image morphing can be described as the interpolation from one image to another image. The focus of this thesis is on images and therefore only morphing in two dimensions will be discussed. It is however necessary to state that morphing is not at all restricted to only two dimensions.

The field of morphing has received a lot of attention over the last years and it has reached a state of maturity. Various solutions to address this problem have been submitted, all with their own advantages and disadvantages, but before discussing how it is done it helps to understand what is being done. It is important to note that a morphing sequence consists of two warps (the spatial transformation of the images to align the features specified in both) followed by a blend, as demonstrated in Figure 1.1. Figure 1.2 illustrates an example of how this procedure is used to create an image morph.

IMAGE MORPHING:

Image morphing can be defined as the construction of an image sequence depicting the gradual transition between the two images.

The simplest way to transform one image into another image is to cross-dissolve (better known as “fade”) them. This is achieved by interpolating the color of each pixel over time from the source image to the destination image. However this will not render a very effective visual morph as can be seen in Figure. 2.1. The first image is merely replaced by the second image without any warping (spatial deformation).

A more effective and spectacular method exists and is known as image morphing. Image morphing involves image warping (changing the position of key features in the images) combined with cross-dissolving.

Some Applications of Image Morphing

In the medical profession, with modern CRT or NMR scans, slices of the human body can be imaged and combined into 3D models. The distance between such slices is usually much larger than the spatial resolution within each slice. For rendering (especially direct volume rendering) and surface reconstruction, this is undesirable as these require the volume elements to have edges of the same length. To achieve this some interpolation between slices is necessary.

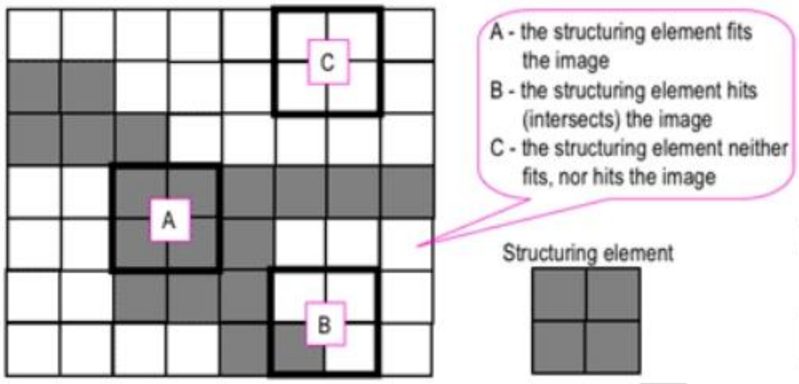
Morphological Dilation and Erosion

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion. This table lists the rules for both dilation and erosion.

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a **structuring element**. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood:

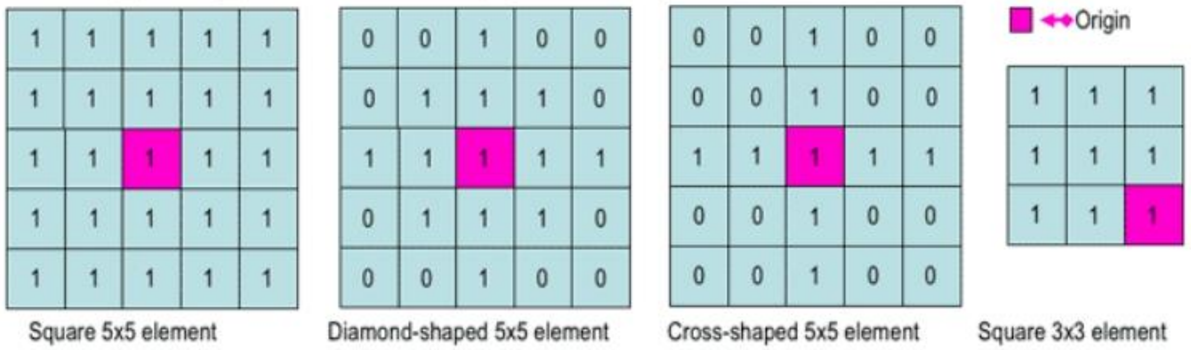


Probing of an image with a structuring element (white and grey pixels have zero and non-zero values, respectively).

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The **structuring element** is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the *size* of the structuring element.
- The pattern of ones and zeros specifies the *shape* of the structuring element.
- An *origin* of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.



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

Square 5x5 element

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

Diamond-shaped 5x5 element

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

Cross-shaped 5x5 element

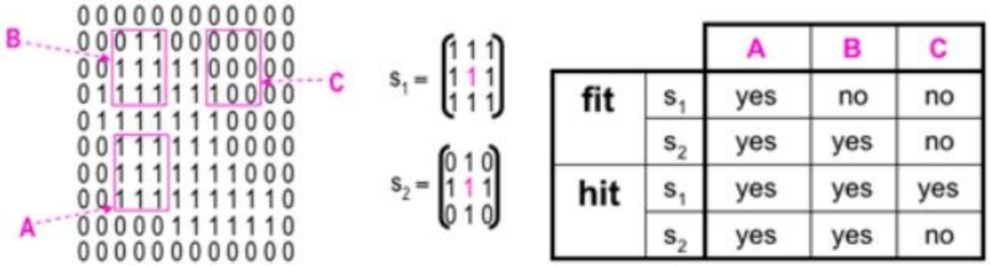
1	1	1
1	1	1
1	1	1

Square 3x3 element

Examples of simple structuring elements.

A common practice is to have odd dimensions of the structuring matrix and the origin defined as the centre of the matrix. Stucturing elements play in moprphological image processing the same role as convolution kernels in linear image filtering.

When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighbourhood under the structuring element. The structuring element is said to **fit** the image if, for each of its pixels set to 1, the corresponding image pixel is also 1. Similarly, a structuring element is said to **hit**, or intersect, an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1.



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	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	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	1	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

$$s_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$s_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

Fitting and hitting of a binary image with structuring elements s_1 and s_2.

Zero-valued pixels of the structuring element are ignored, i.e. indicate points where the corresponding image value is irrelevant.

Fundamental operations

More formal descriptions and examples of how basic morphological operations work are given in the Hypermedia Image Processing Reference (HIPR) developed by Dr. R. Fisher et al. at the Department of Artificial Intelligence in the University of Edinburgh, Scotland, UK.

Erosion and dilation

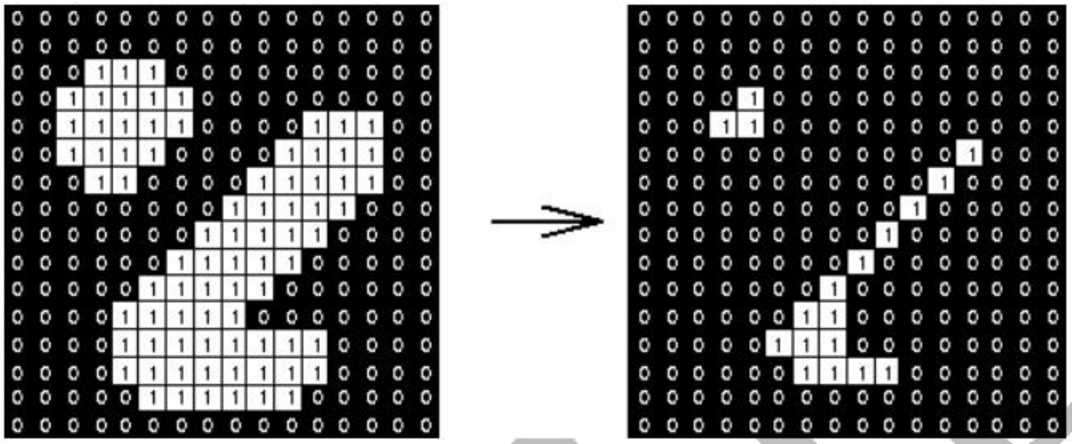
The **erosion** of a binary image *f* by a structuring element *s* (denoted $f \ominus s$) produces a new binary image $g = f \ominus s$ with ones in all locations (x,y) of a structuring element's origin at which that structuring element *s* fits the input image *f*, i.e. $g(x,y) = 1$ is *s* fits *f* and 0 otherwise, repeating for all pixel coordinates (x,y).

Greyscale image

Binary image by thresholding

Erosion: a 2×2 square structuring element

Erosion with small (e.g. 2×2 - 5×5) square structuring elements shrinks an image by stripping away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger, and small details are eliminated:



Erosion: a 3×3 square structuring element (www.cs.princeton.edu/~pshilane/class/mosaic/).

Larger structuring elements have a more pronounced effect, the result of erosion with a large structuring element being similar to the result obtained by iterated erosion using a smaller structuring element of the same shape. If s_1 and s_2 are a pair of structuring elements identical in shape, with s_2 twice the size of s_1, then

$f \ominus s_2 \approx (f \ominus s_1) \ominus s_1$.

Erosion removes small-scale details from a binary image but simultaneously reduces the size of regions of interest, too. By subtracting the eroded image from the original image, boundaries of each region can be found: $b = f - (f \ominus s)$ where f is an image of the regions, s is a 3×3 structuring element, and b is an image of the region boundaries.

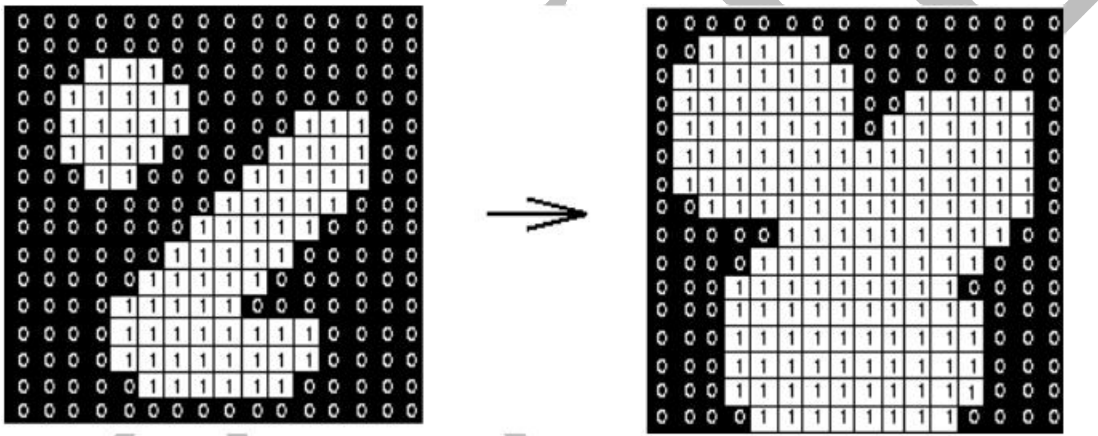
The **dilation** of an image f by a structuring element s (denoted $f \oplus s$) produces a new binary image $g = f \oplus s$ with ones in all locations (x,y) of a structuring element's orogin at which that structuring element s hits the the input image f, i.e. $g(x,y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x,y). Dilation has the opposite effect to erosion -- it adds a layer of pixels to both the inner and outer boundaries of regions.

Binary image

Dilation: a 2×2 square structuring element

http://documents.wolfram.com/applications/digitalimage/UsersGuide/Morphology/ImageProcessing6.3.html

The holes enclosed by a single region and gaps between different regions become smaller, and small intrusions into boundaries of a region are filled in:



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

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

Dilation: a 3×3 square structuring element (www.cs.princeton.edu/~pshilane/class/mosaic/).

Results of dilation or erosion are influenced both by the size and shape of a structuring element. Dilation and erosion are *dual* operations in that they have opposite effects. Let f^c denote the complement of an image *f*, i.e., the image produced by replacing 1 with 0 and vice versa. Formally, the duality is written as

$$f \oplus s = f^c \ominus s_{rot}$$

where s_{rot} is the structuring element *s* rotated by 180°. If a structuring element is symmetrical with respect to rotation, then s_{rot} does not differ from *s*. If a binary image is considered to be a collection of connected regions of pixels set to 1 on a background of pixels set to 0, then erosion is the fitting of a structuring element to these regions and dilation is the fitting of a structuring element (rotated if necessary) into the background, followed by inversion of the result.

Rules for Dilation and Erosion

Operation	**Rule**
Dilation	The value of the output pixel is the *maximum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value1, the output pixel is set to 1.
Erosion	The value of the output pixel is the *minimum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.

The following illustrates the dilation of a binary image. Note how the structuring element defines the neighborhood of the pixel of interest, which is circled. (See Structuring Elements for more information.) The dilation function applies the appropriate rule to the pixels in the neighborhood and assigns a value to the corresponding pixel in the output image. In the figure, the morphological dilation function sets the value of the output pixel to 1 because one of the elements in the neighborhood defined by the structuring element is on.

A- B = {z|(B) -A}

The following illustrates this processing for a grayscale image. The figure shows the processing of a particular pixel in the input image. Note how the function applies the rule to the input pixel's neighborhood and uses the highest value of all the pixels in the neighborhood as the value of the corresponding pixel in the output image.

Structure elements

Morphology involves the use of subimages called as structuring elements. The pixels in a structuring element can have values 0 (black), 1 (white), or may even be don't care (either black or white). The structuring element is used to assess or probe the attributes and properties of the images under study.

The origin of the structuring element is generally taken as the center of the rectangular array which contains the structuring element. However the origin need not be specified as the center. Changing the origin of the structuring element also changes the output of the morphological operations.

- We talk of morphological operations between two image objects.
- The first one is the object/ region under study.

- The second one is an object (a subimage depicting a region) used to probe the first one to identify its structural characteristics.
- All sets are padded with background elements to form a rectangular array or to provide a background border.

The structuring element is also called as a mask or a kernel.

Translation and reflection are set operations which do not involve any structuring element. Translation of a set means that each element of the set is displaced by a fixed translation distance. Reflection of a set means that the coordinate of each pixel will shift to the other side of the axis. So x becomes - x and y becomes - y.

Reflection of a set	
Translation of a set	
Set intersection	This is the traditional intersection of two sets. If the sets indicate image regions, then their intersection would give the region overlap.
Set union	This is the traditional union of two sets. If the sets indicate image regions, then their union would give the aggregate of the two regions.

OPENING AND CLOSING:

Opening and closing are two important operators from mathematical morphology. They are both derived from the fundamental operations of erosion and dilation. Like those operators they are normally applied to binary images, although there are also graylevel versions. The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels. However it is less destructive than erosion in general. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve foreground regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

Very simply, an opening is defined as an erosion followed by a dilation using the same structuring element for both operations. See the sections on erosion and dilation for details of the individual steps. The opening operator therefore requires two inputs: an image to be opened, and a structuring element.

Compound operations

Many morphological operations are represented as combinations of erosion, dilation, and simple set-theoretic operations such as the **complement** of a binary image:

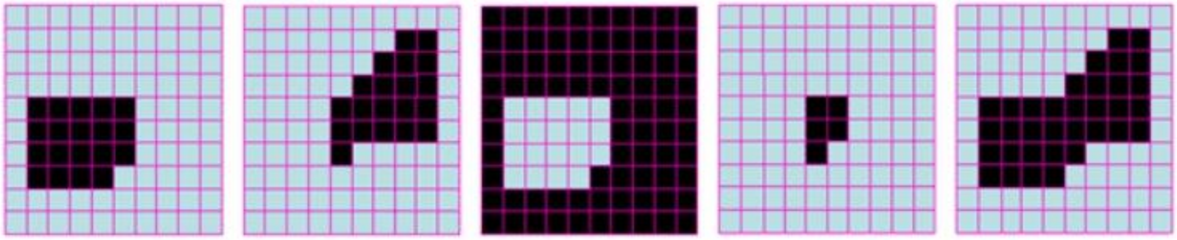
$f_c(x,y) = 1$ if $f(x,y) = 0$, and $f_c(x,y) = 0$ if $f(x,y) = 1$,

the **intersection** $h = f \cap g$ of two binary images f and g:

$h(x,y) = 1$ if $f(x,y) = 1$ and $g(x,y) = 1$, and $h(x,y) = 0$ otherwise,

and the **union** $h = f \cup g$ of two binary images f and g:

$h(x,y) = 1$ if $f(x,y) = 1$ or $g(x,y) = 1$, and $h(x,y) = 0$ otherwise:



Set operations on binary images: from left to right: a binary image f, a binary image g, the complement f_c of f, the intersection $f \cap g$, and the union $f \cup g$.

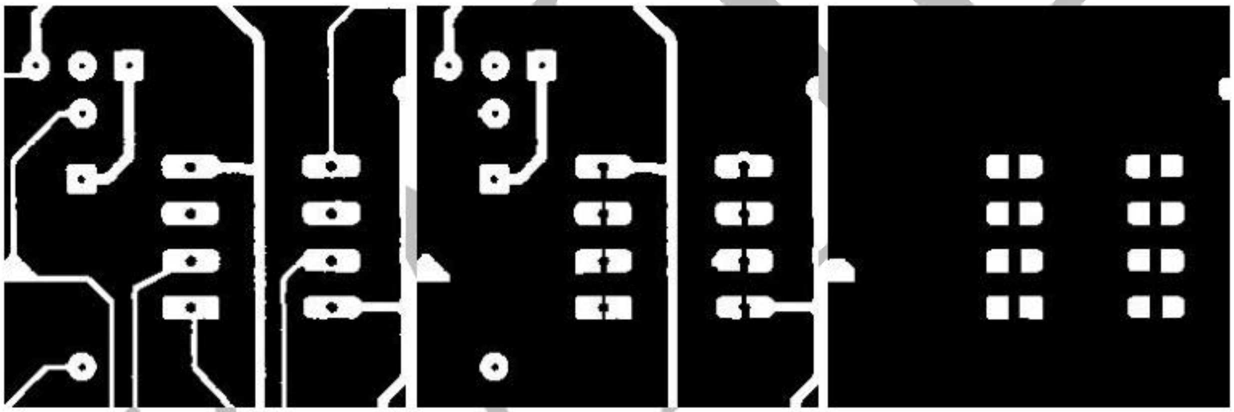
The **opening** of an image f by a structuring element s (denoted by $f \circ s$) is an erosion followed by a dilation:

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


Binary image


Opening: a 2×2 square structuring element

Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation:


Binary image f $f \circ s$ (5×5 square) $f \circ s$ (9×9 square)

Results of opening with a square structuring element

Opening is an **idempotent** operation: once an image has been opened, subsequent openings with the same structuring element have no further effect on that image:

$(f \circ s) \circ s) = f \circ s.$

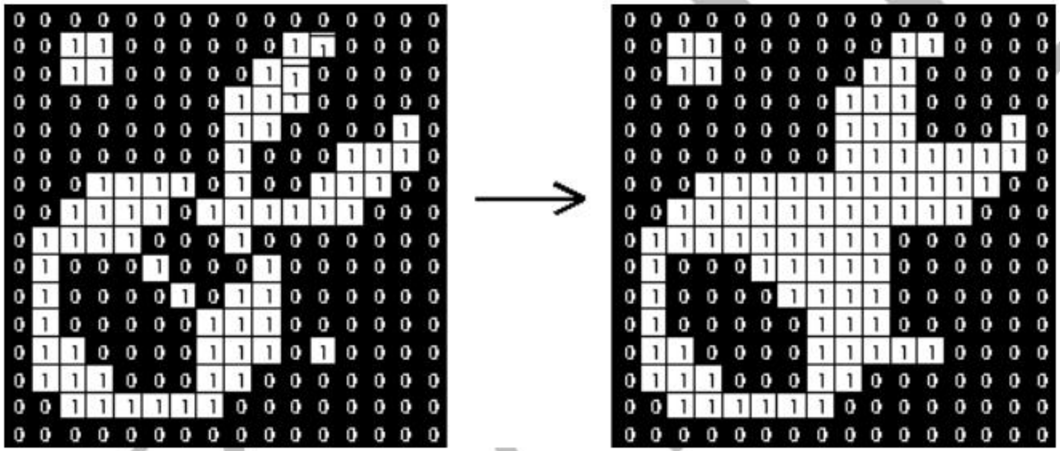
The **closing** of an image f by a structuring element s (denoted by $f \bullet s$) is a dilation followed by an erosion:

$f \bullet s = (f \oplus s_{rot}) \ominus s_{rot}$

Binary image

Closing: a 2×2 square structuring element

In this case, the dilation and erosion should be performed with a rotated by 180° structuring element. Typically, the latter is symmetrical, so that the rotated and initial versions of it do not differ.



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

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

Closing with a 3×3 square structuring element

Closing is so called because it can fill holes in the regions while keeping the initial region sizes. Like opening, closing is idempotent: $(f \bullet s) \bullet s = f \bullet s$, and it is dual operation of opening (just as opening is the dual operation of closing):

$f \bullet s = (f^c \circ s)^c$; $f \circ s = (f^c \bullet s)^c$.

In other words, closing (opening) of a binary image can be performed by taking the complement of that image, opening (closing) with the structuring element, and taking the complement of the result.

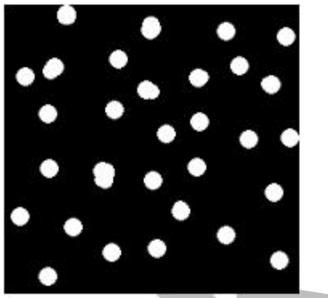
Graylevel opening consists simply of a graylevel erosion followed by a graylevel dilation.

Opening is the dual of closing, i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

As with erosion and dilation, it is very common to use this 3×3 structuring element. The effect in the above figure is rather subtle since the structuring element is quite compact and so it fits into the foreground boundaries quite well even before the opening operation. To increase the effect, multiple erosions are often performed with this element followed by the same number of dilations. This effectively performs an opening with a larger square structuring element.

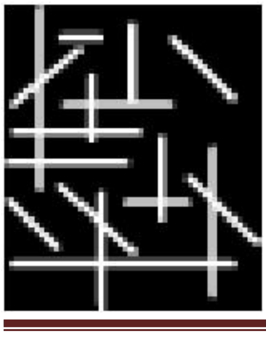
Consider

which is a binary image containing a mixture of circles and lines. Suppose that we want to separate out the circles from the lines, so that they can be counted. Opening with a disk shaped structuring element 11 pixels in diameter gives



Some of the circles are slightly distorted, but in general, the lines have been almost completely removed while the circles remain almost completely unaffected.

The image



Opening can be very useful for separating out particularly shaped objects from the background, but it is far from being a universal 2-D object recognizer/segmenter. For instance if we try and use a long thin structuring element to locate, say, pencils in our image, any one such element will only find pencils at a particular orientation. If it is necessary to find pencils at other orientations then differently oriented elements must be used to look for each desired orientation. It is also necessary to be very careful that the structuring element chosen does not eliminate too many desirable objects, or retain too many undesirable ones, and sometimes this can be a delicate or even impossible balance.

Closing:

Closing is an important operator from the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators it is normally applied to binary images, although there are graylevel versions. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

Closing is opening performed in reverse. It is defined simply as a dilation followed by an erosion using the same structuring element for both operations. See the sections on erosion and dilation for details of the individual steps. The closing operator therefore requires two inputs: an image to be closed and a structuring element.

Graylevel closing consists straightforwardly of a graylevel dilation followed by a graylevel erosion.

Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element.

Closing can sometimes be used to selectively fill in particular background regions of an image. Whether or not this can be done depends upon whether a suitable structuring element can be found that fits well inside regions that are to be preserved, but doesn't fit inside regions that are to be removed.

The image

is an image containing large holes and small holes. If it is desired to remove the small holes while retaining the large holes, then we can simply perform a closing with a disk-shaped structuring element with a diameter larger than the smaller holes, but smaller than the large holes.

The image

is the result of a closing with a 22 pixel diameter disk. Note that the thin black ring has also been filled in as a result of the closing operation.

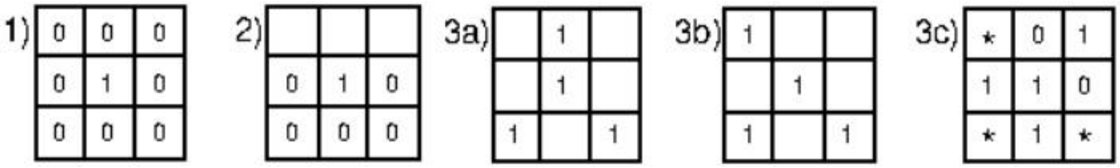
HIT-AND-MISS TRANSFORM

The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image. It is actually the basic operation of binary morphology since almost all the other binary morphological operators can be derived from it. As with other binary morphological operators it takes as input a binary image and a structuring element, and produces another binary image as output.

The hit-and-miss transform is used to look for occurrences of particular binary patterns in fixed orientations. It can be used to look for several patterns (or alternatively, for the same pattern in several orientations as above) simply by running successive transforms using different structuring elements, and then ORing the results together.

The operations of erosion, dilation, opening, closing, thinning and thickening can all be derived from the hit-and-miss transform in conjunction with simple set operations.

Figure illustrates some structuring elements that can be used for locating various binary features.



1)

0	0	0
0	1	0
0	0	0

2)

0	1	0
0	0	0

3a)

	1	
	1	
1		1

3b)

1		
	1	
1		1

3c)

*	0	1
1	1	0
*	1	*

Figure Some applications of the hit-and-miss transform. 1 is used to locate isolated points in a binary image. 2 is used to locate the end points on a binary skeleton Note that this structuring element must be used in all its rotations so four hit-and-miss passes are required. 3a, 3b, and 3c are the kernels used to locate the triple points (junctions) on a skeleton. All three structuring elements must be run in all orientations so twelve hit-and-miss passes are required.

The successful use of the hit-and-miss transform relies on being able to think of a relatively small set of binary patterns that capture all the possible variations and orientations of a feature that is to be located. For features larger than a few pixels across this is often not feasible.

Edge Detection

What are edges

We can also say that sudden changes of discontinuities in an image are called as edges. Significant transitions in an image are called as edges.

Types of edges

Generally edges are of three types:

- Horizontal edges
- Vertical Edges
- Diagonal Edges

Why detect edges

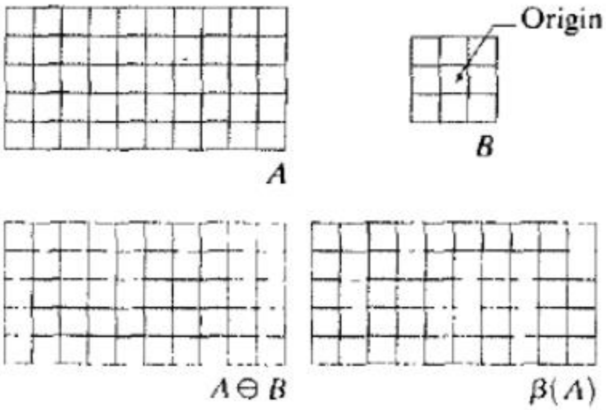
Most of the shape information of an image is enclosed in edges. So first we detect these edges in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer.

BOUNDARY EXTRACTION

The boundary of a set *A*, denoted by (3(*A*), can be obtained by first eroding *A* by *B* and then performing the set difference between *A* and its erosion. That is.

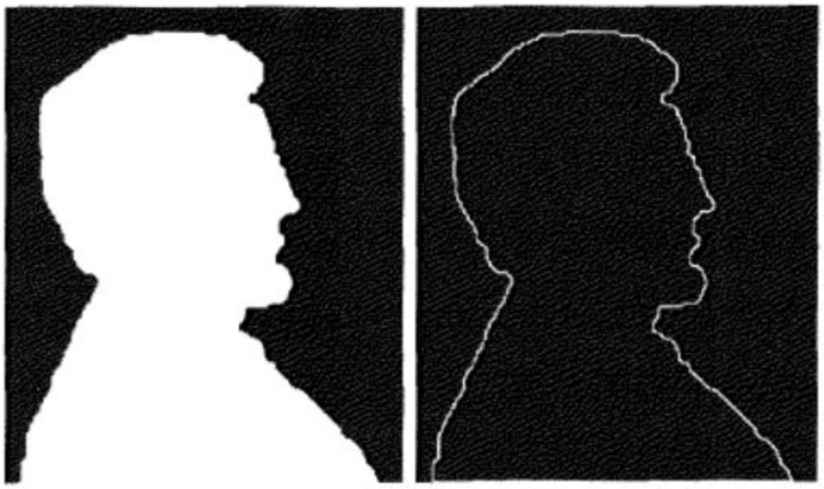
$$\beta(A) = A - (A \ominus B)$$

where *B* is a suitable structuring element.
Figure 9.13 illustrates the mechanics of boundary extraction. It shows a simple binary object, a structuring element *B*, and the result of using Eq. (9.5-1).



a b
c d

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



a b

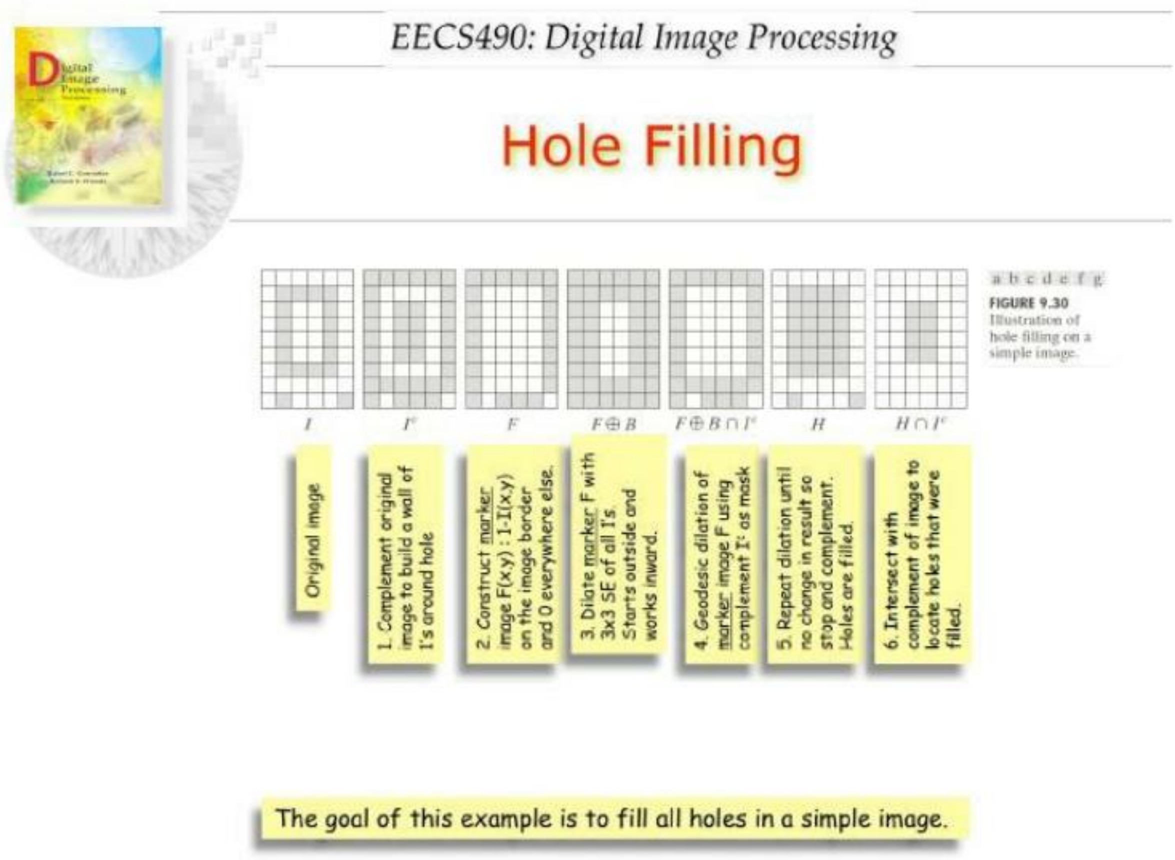
FIGURE 9.14 (a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

Although the structuring element shown in Fig. 9.13(b) is among the most frequently used, it is by no means unique. For example, using a 5 × 5 structuring element of 1's would result in a boundary between 2 and 3 pixels thick. Note that when the origin of B is on the edges of the set, part of the structuring element may be outside the image. The normal treatment of this condition is to assume that values outside the borders of the image are 0.

EXAMPLE 9.5: Boundary extraction by morphological processing.

▨ Figure 9.14 further illustrates the use of Eq. (9.5-1) with the structuring element of Fig. 9.13(b). In this example, binary 1's are shown in white and 0's in black, so the elements of the structuring element, which are 1's, also are treated as white. Because of the structuring element used, the boundary shown in Fig. 9.14(b) is one pixel thick.

REGION FILLING



EECS490: Digital Image Processing

Hole Filling

FIGURE 9.30 Illustration of hole filling on a simple image.

Original image

1. Complement original image to build a wall of 1's around hole

2. Construct marker image F(x,y) : 1-I(x,y) on the image border and 0 everywhere else.

3. Dilate marker F with 3x3 SE of all 1's. Starts outside and works inward.

4. Geodesic dilation of marker image F using complement I: as mask

5. Repeat dilation until no change in result so stop and complement. Holes are filled.

6. Intersect with complement of image to locate holes that were filled.

The goal of this example is to fill all holes in a simple image.

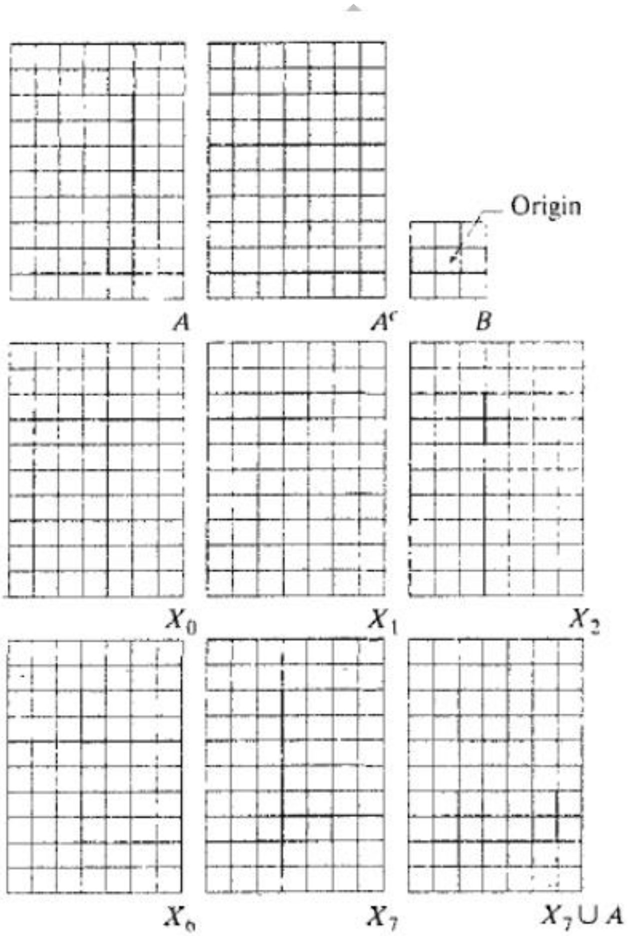
Next we develop a simple algorithm for region filling based on set dilations, complementation, and intersections. In Fig. 9.15 *A* denotes a set containing a subset whose elements are 8-connected boundary points of a region. Beginning with a point *p* inside the boundary. the objective is to fill the entire region with 1's. If we adopt the convention that all nonboundary (background) points are labeled O.then we assign a value of I to *p* to begin. The following procedure then fills the region with l's:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \ldots \qquad (9.5\text{-}2)$$

where $X_0 = p$, and B is the symmetric structuring element shown in Fig. 9.15(c). The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The set union of X_k and A contains the filled set and its boundary.

The dilation process of Eq. (9.5-2) would fill the entire area if left unchecked. However, the intersection at each step with *A′* limits the result to inside the region of interest.This is our first example of how a morphological process can be conditioned to meet a desired property. In the current application, it is

appropriately called *conditional dilation.* The rest of Fig. 9.15 illustrates further the mechanics of Eq. (9.5-2). Although this example has only one subset, the concept clearly applies to any finite number of such subsets, assuming that a point inside each boundary is given.

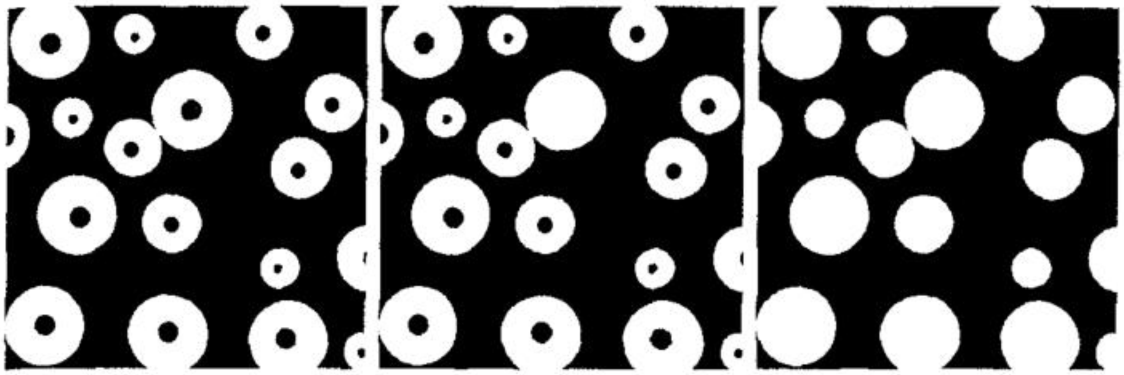


a b c
d e f
g h i

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

EXTRACTION OF CONNECTED COMPONENTS

The concepts of connectivity and connected components were introduced in Section 2.S.2.ln practice. extraction of connected components in a binary image, is central to many. automated image analysis applications. Let *Y* represent a connected component contained in a set *A* and assume that a point p of Y is



a b c

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

known. Then the following iterative expression yields all the elements of Y:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \ldots \tag{9.5-3}$$

where $X_0 = p$, and B is a suitable structuring element, as shown in Fig. 9.17. If $X_k = X_{k-1}$, the algorithm has converged and we let $Y = X_k$.

Equation (9.5-3) is similar in form to Eq. (9.5-2). The only difference is the use of A instead of its complement. This difference arises because all the elements sought (that is, the elements of the connected component) are labeled 1.

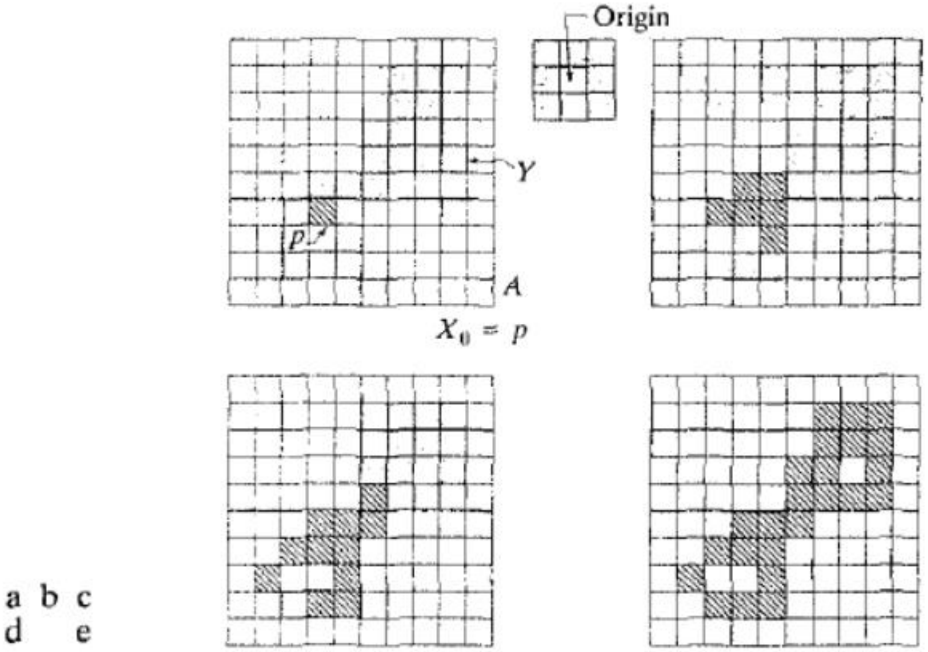


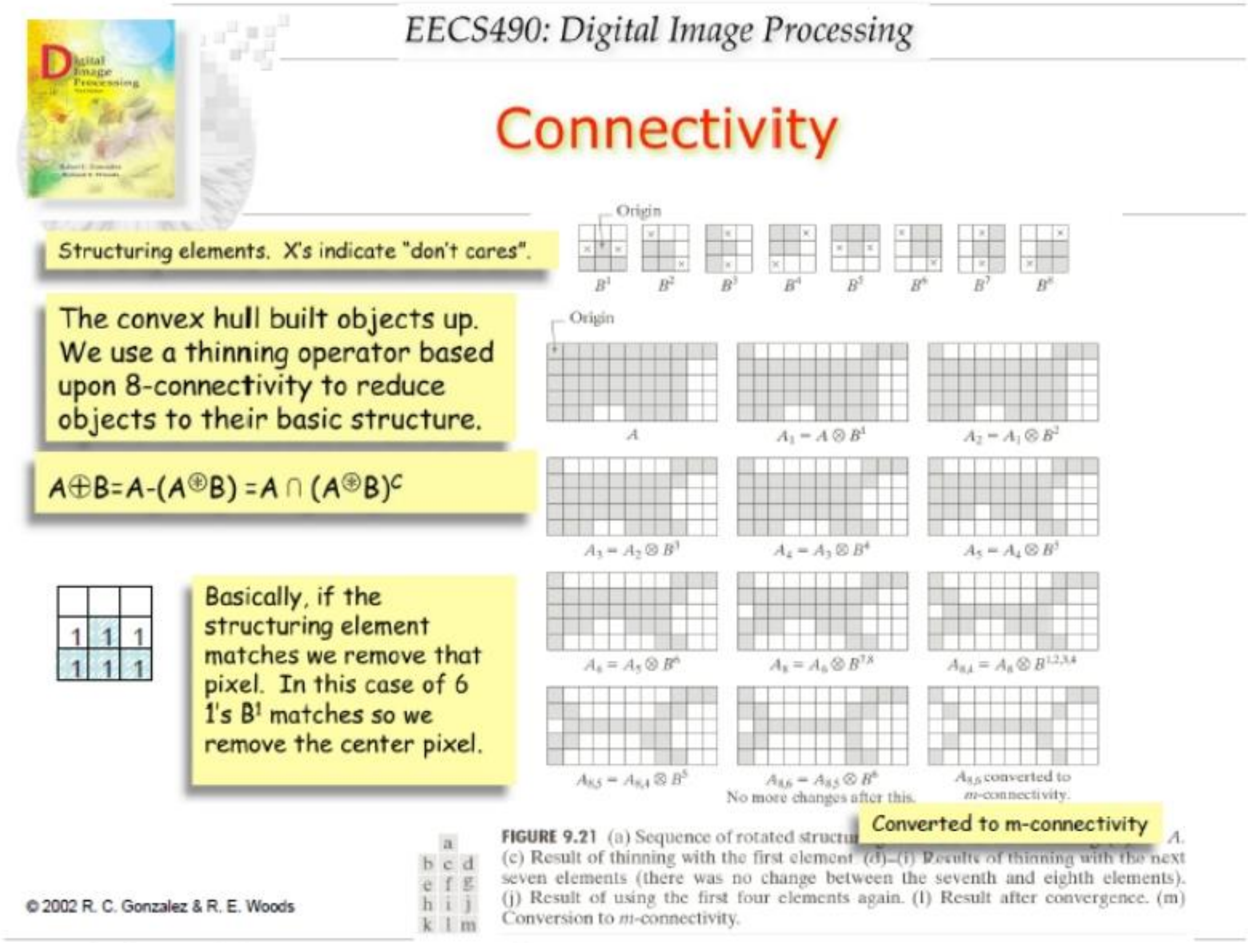
FIGURE 9.17 (a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

The intersection with *A* at each iterative step eliminates dilations centered on elements labeled O. Figure 9.17 illustrates the mechanics of Eq. (9.5-3). Note that the shape of the structuring element assumes 8-connectivity between pixels. As in the region-filling algorithm, the results just discussed are applicable to any finite number of sets of connected components contained in *A*, assuming that a point is known in each connected component.

EXAMPLE :

Using connected components to detect foreign objects in packaged food.

CONVEX HULL



EECS490: Digital Image Processing

Connectivity

Structuring elements. X's indicate "don't cares".

The convex hull built objects up. We use a thinning operator based upon 8-connectivity to reduce objects to their basic structure.

$A \oplus B = A - (A \circledast B) = A \cap (A \circledast B)^C$

1	1	1
1	1	1

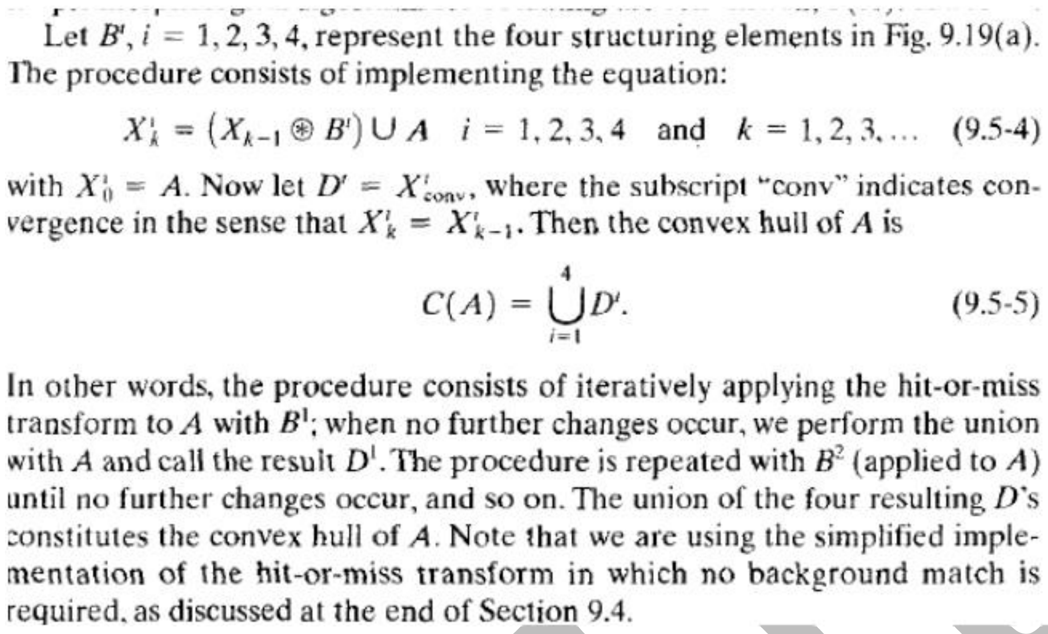
Basically, if the structuring element matches we remove that pixel. In this case of 6 1's B^1 matches so we remove the center pixel.

Converted to m-connectivity

FIGURE 9.21 (a) Sequence of rotated structu A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to m-connectivity.

© 2002 R. C. Gonzalez & R. E. Woods

A set *A* is said to be *convex* if the straight line segment joining any two points in *A* lies entirely within *A*. The *convex hull H* of an arbitrary set S is the smallest convex set containing S. The set difference *H* - S is called the *convex deficiency* of S.As discussed in more detail in Sections 11.1.4 and 11.3.2, the convex hull and convex deficiency are useful for object description. Here, we present a simple morphological algorithm for obtaining the convex hull, C(*A*). of a set *A*.



Let B^i, $i = 1, 2, 3, 4$, represent the four structuring elements in Fig. 9.19(a). The procedure consists of implementing the equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \quad \text{and} \quad k = 1, 2, 3, \ldots \qquad (9.5\text{-}4)$$

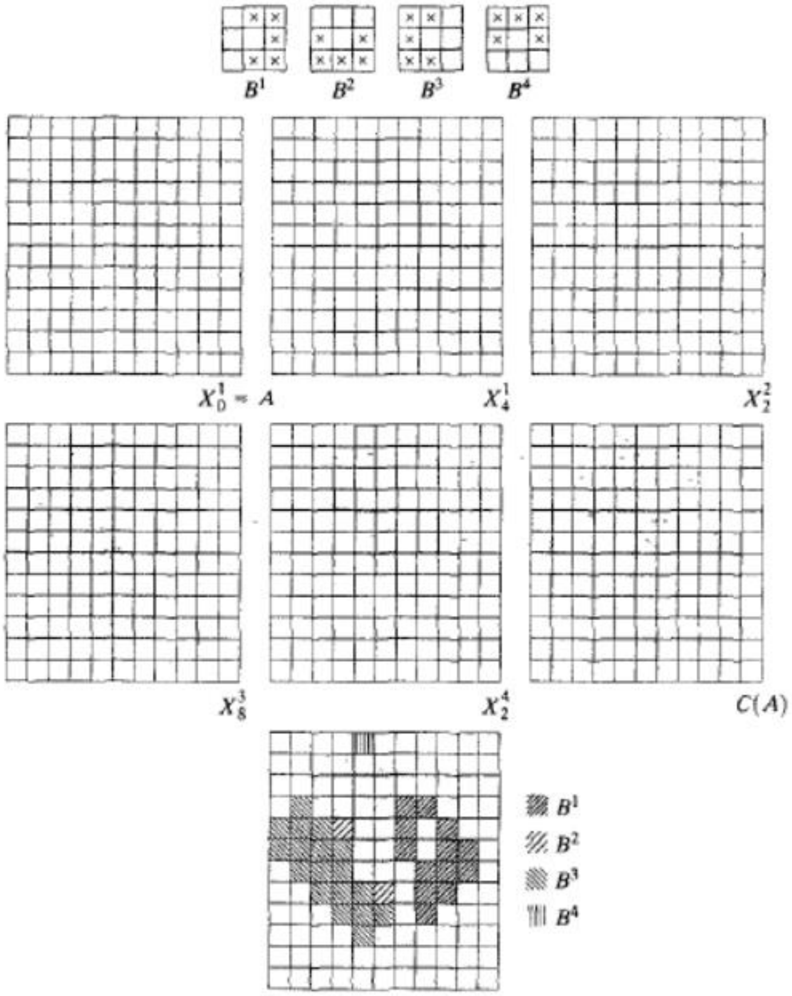
with $X_0^i = A$. Now let $D^i = X_{\text{conv}}^i$, where the subscript "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. Then the convex hull of A is

$$C(A) = \bigcup_{i=1}^{4} D^i. \qquad (9.5\text{-}5)$$

In other words, the procedure consists of iteratively applying the hit-or-miss transform to A with B^1; when no further changes occur, we perform the union with A and call the result D^1. The procedure is repeated with B^2 (applied to A) until no further changes occur, and so on. The union of the four resulting D's constitutes the convex hull of A. Note that we are using the simplified implementation of the hit-or-miss transform in which no background match is required, as discussed at the end of Section 9.4.

Figure 9.19 illustrates the procedure given in Eqs. (9.5-4) and (9.5-5). Figure 9.19(a) shows the structuring elements used to extract the convex hull. The origin of each element is at its center. The X entries indicate "don't care" conditions. This means that a structuring element is said to have found a match in *A* if the 3-by-3 region of *A* under the structuring element mask at that location matches the pattern of the mask. For a particular mask, a pattern match occurs when the center of the 3-by-3 region in *A* is 0, and the three pixels under the shaded mask elements are 1.The values of the other pixels in the 3-by-3 region do not matter. Also, with respect to the notation in Fig. 9.19(a), B' is a clockwise rotation of B' -I by 90°.

Figure 9.19(b) shows a set A for which the convex hull is sought. Starting with $X_0^1 = A$ resulted in the set shown in Fig. 9.19(c) after four iterations of Eq. (9.5-4). Then, letting $X_0^2 = A$ and again using Eq. (9.5-4) resulted in the set



a
b c d
e f g
h

FIGURE 9.19 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

THINNING AND THICKENING

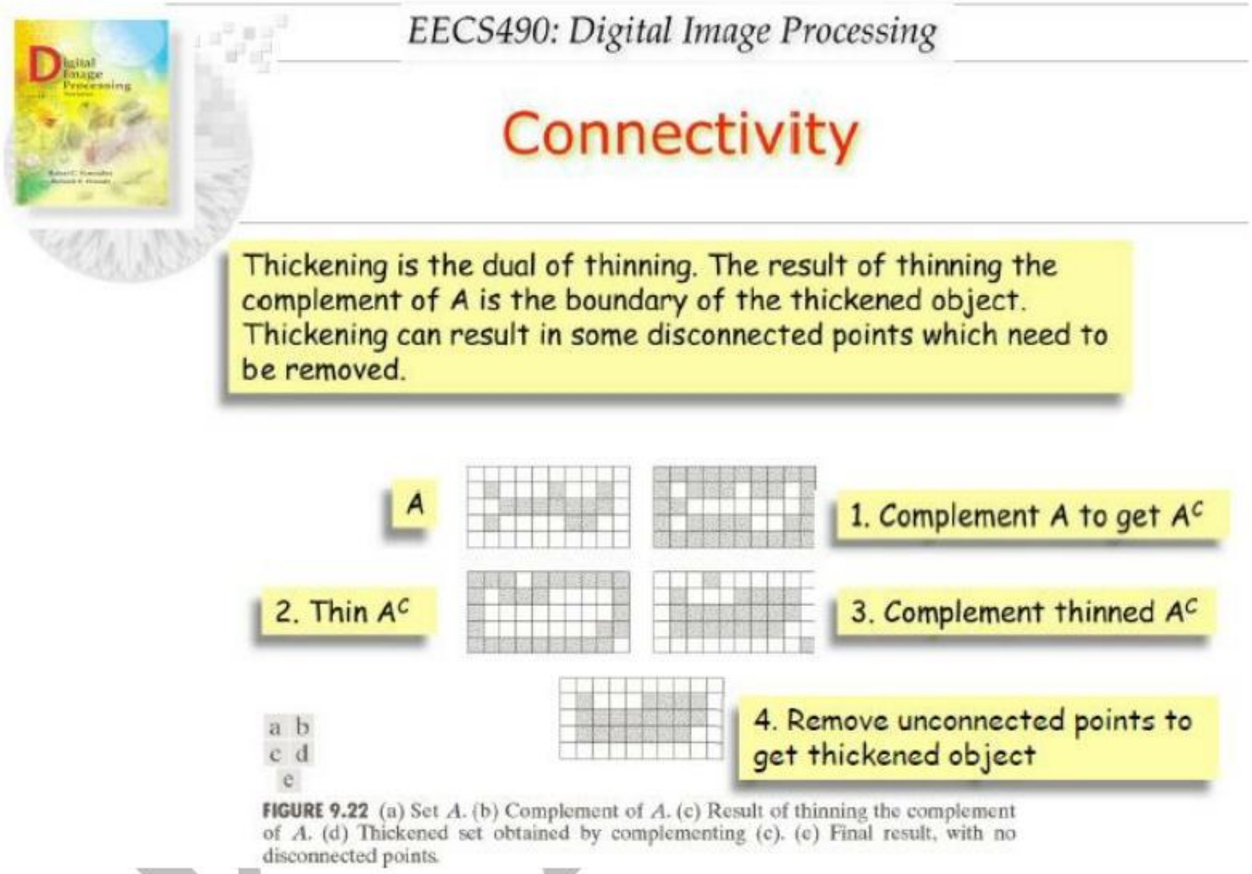
- Thinning is an image-processing operation in which binary valued image regions are reduced to lines

- The purpose of thinning is to reduce the image components to their essential information for further analysis and recognition

- Thickening is changing a pixel from 1 to 0 if any neighbors of the pixel are 1.

- Thickening followed by thinning can be used for filling undesirable holes.

- Thinning followed by thickening is used for determining isolated components and clusters.



EECS490: Digital Image Processing

Connectivity

Thickening is the dual of thinning. The result of thinning the complement of A is the boundary of the thickened object. Thickening can result in some disconnected points which need to be removed.

A

1. Complement A to get A^C

2. Thin A^C

3. Complement thinned A^C

4. Remove unconnected points to get thickened object

FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

Thinning

- Thinning is defined in terms of hit or miss as

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c.$$

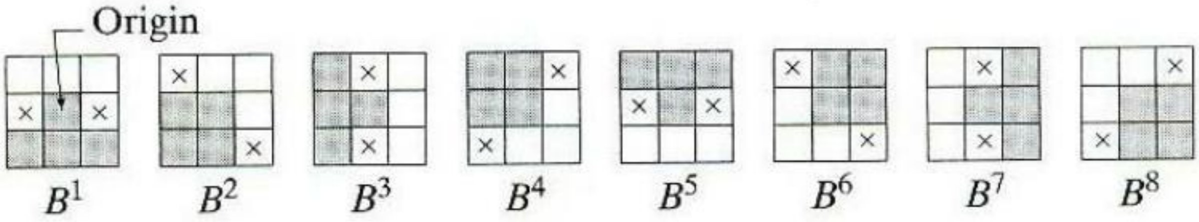
where B is a sequence of structuring elements like

{B} = {B1, B2, B3, …, Bn} and the operation can be given as

$$A \otimes \{B\} = ((\dots((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$$

Thinning

- Sample set of structuring elements



Thickening

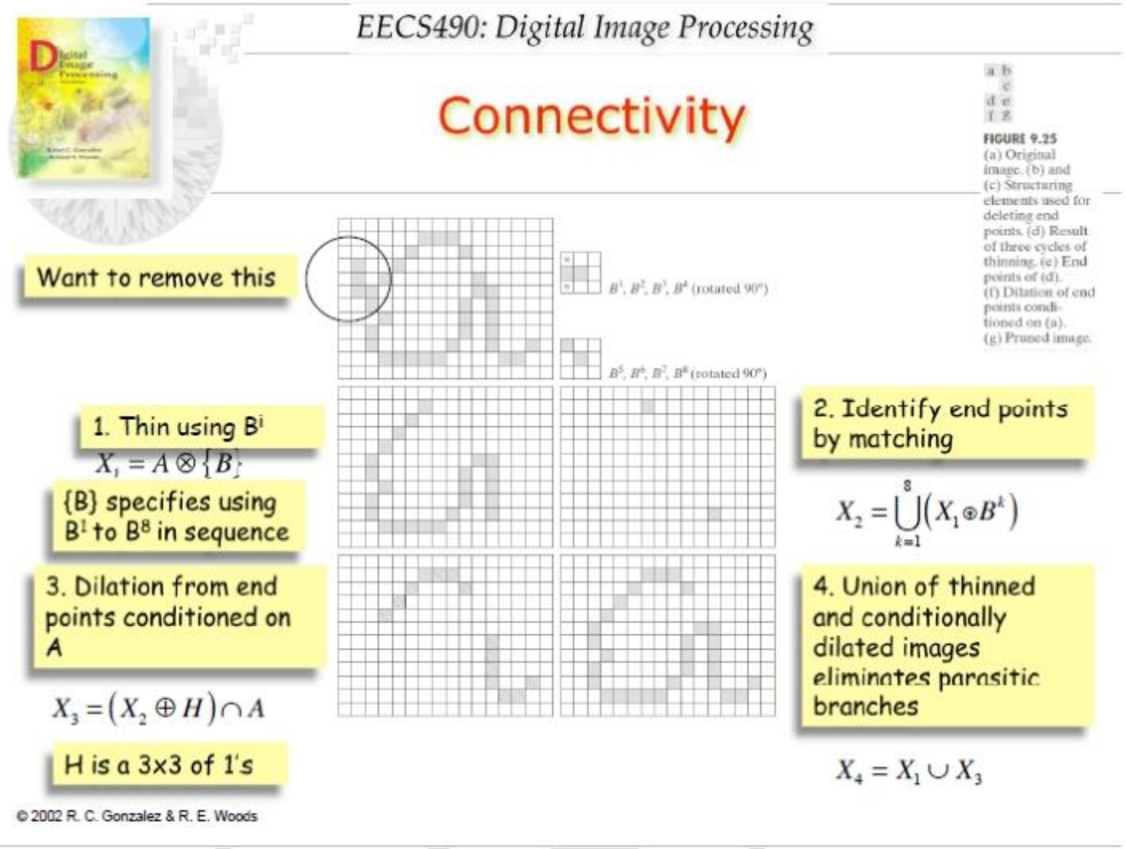
- Thickening is the morphological dual of thinning and defined as

$$A \odot B = A \cup (A \circledast B)$$

Or

$$A \odot \{B\} = ((\dots((A \odot B^1) \odot B^2) \dots) \odot B^n)$$

Example:



EECS490: Digital Image Processing

Connectivity

Want to remove this

FIGURE 9.25 (a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

1. Thin using B^i

$$X_1 = A \otimes \{B\}$$

{B} specifies using B^1 to B^8 in sequence

2. Identify end points by matching

$$X_2 = \bigcup_{k=1}^{8} \left(X_1 \circledast B^k \right)$$

3. Dilation from end points conditioned on A

$$X_3 = \left(X_2 \oplus H \right) \cap A$$

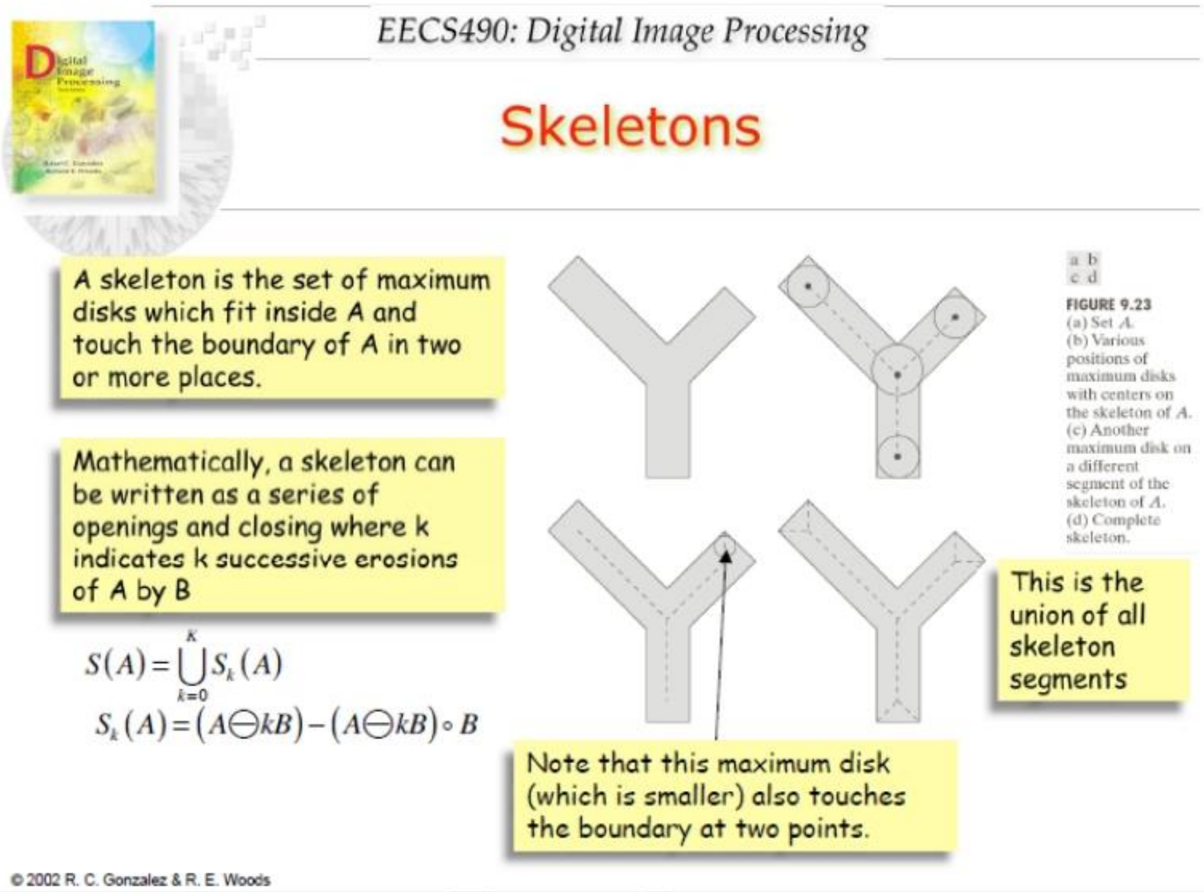
H is a 3x3 of 1's

4. Union of thinned and conditionally dilated images eliminates parasitic branches

$$X_4 = X_1 \cup X_3$$

© 2002 R. C. Gonzalez & R. E. Woods

SKELETON



EECS490: Digital Image Processing

Skeletons

A skeleton is the set of maximum disks which fit inside A and touch the boundary of A in two or more places.

Mathematically, a skeleton can be written as a series of openings and closing where k indicates k successive erosions of A by B

$$S(A)=\bigcup_{k=0}^{K} S_k(A)$$

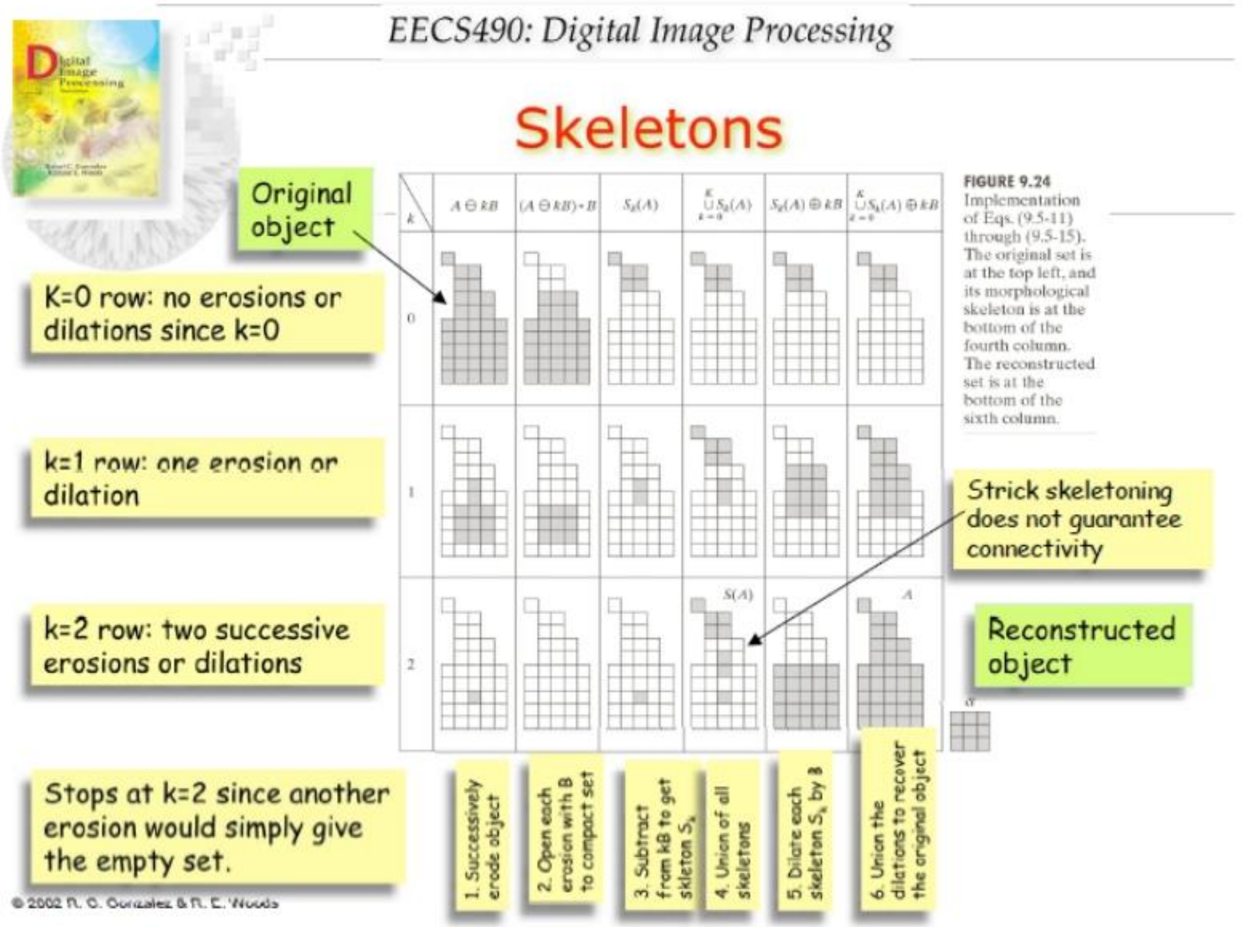
$$S_k(A)=(A\ominus kB)-(A\ominus kB)\circ B$$

FIGURE 9.23 (a) Set A. (b) Various positions of maximum disks with centers on the skeleton of A. (c) Another maximum disk on a different segment of the skeleton of A. (d) Complete skeleton.

This is the union of all skeleton segments

Note that this maximum disk (which is smaller) also touches the boundary at two points.

© 2002 R. C. Gonzalez & R. E. Woods



EECS490: Digital Image Processing

Skeletons

Original object

K=0 row: no erosions or dilations since k=0

k=1 row: one erosion or dilation

k=2 row: two successive erosions or dilations

Stops at k=2 since another erosion would simply give the empty set.

1. Successively erode object
2. Open each erosion with B to compact set
3. Subtract from kB to get skeleton S_k
4. Union of all skeletons
5. Dilate each skeleton S_k by B
6. Union the dilations to recover the original object

FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

Strick skeletoning does not guarantee connectivity

Reconstructed object

© 2002 R. C. Gonzalez & R. E. Woods

□The informal definition of a skeleton is a line representation of an object that is:

□one-pixel thick,

□through the "middle" of the object, and,

□ □preserves the topology of the object.

Skeleton is defined by

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

Where,

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

k is the last iterative step before A erodes to an empty set

$$K = \max\{k \mid (A \ominus kB) \neq \emptyset\}$$

PRUNING

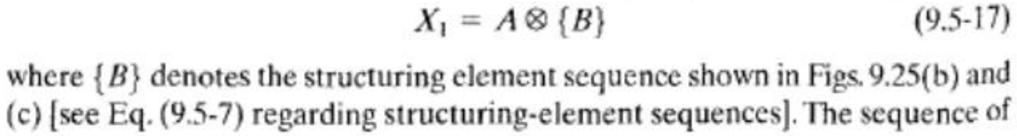
□Thinning and skeletonizing algorithms need a clean-up post-processing

□The following steps are used for pruning:

- Thinning
- Find the end points
- Dilate end points
- Find the union of X1 and X3

Pruning methods are an essential complement to thinning and skeletonizing algorithms because these procedures tend to leave parasitic components that need to he "cleaned up" by post-processing. We begin the discussion with a pruning problem and then develop a morphological solution based on the material introduced in the preceding sections. Thus we take this opportunity to illustrate how to go about solving a problem by combining several of the techniques discussed up to this point.

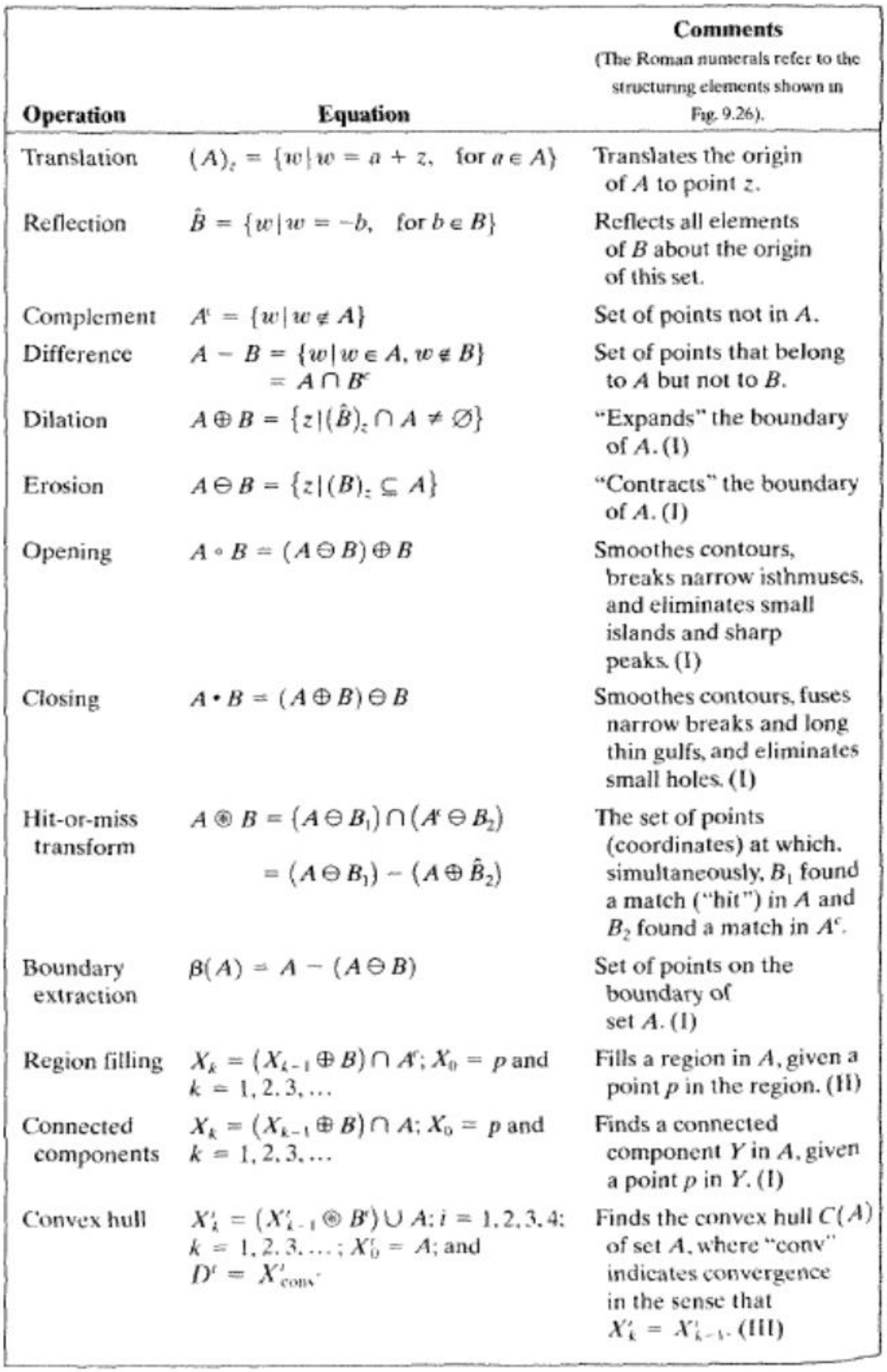
A common approach in the automated recognition of hand-printed characters is to analyze the shape of the skeleton of each character. These skeletons often are characterized by "spurs" (parasitic components). Spurs are caused during erosion by non uniformities in the strokes composing the characters. We develop a morphological technique for handling this problem, starting with the assumption that the length of a parasitic component does not exceed a specified number of pixels. Figure 9.25(a) shows the skeleton of a hand-printed "a." The parasitic component on the leftmost part of the character is illustrative of what we are interested in removing. The solution is based on suppressing a parasitic branch by successively eliminating its end point. Of course, this also shortens (or eliminates) other branches in the character but, in the absence of other structural information, the assumption in this example is that any branch with three or less pixels is to be eliminated. Thinning of an input set *A* with a sequence of structuring elements designed to detect only end points achieves the desired result. That is, let



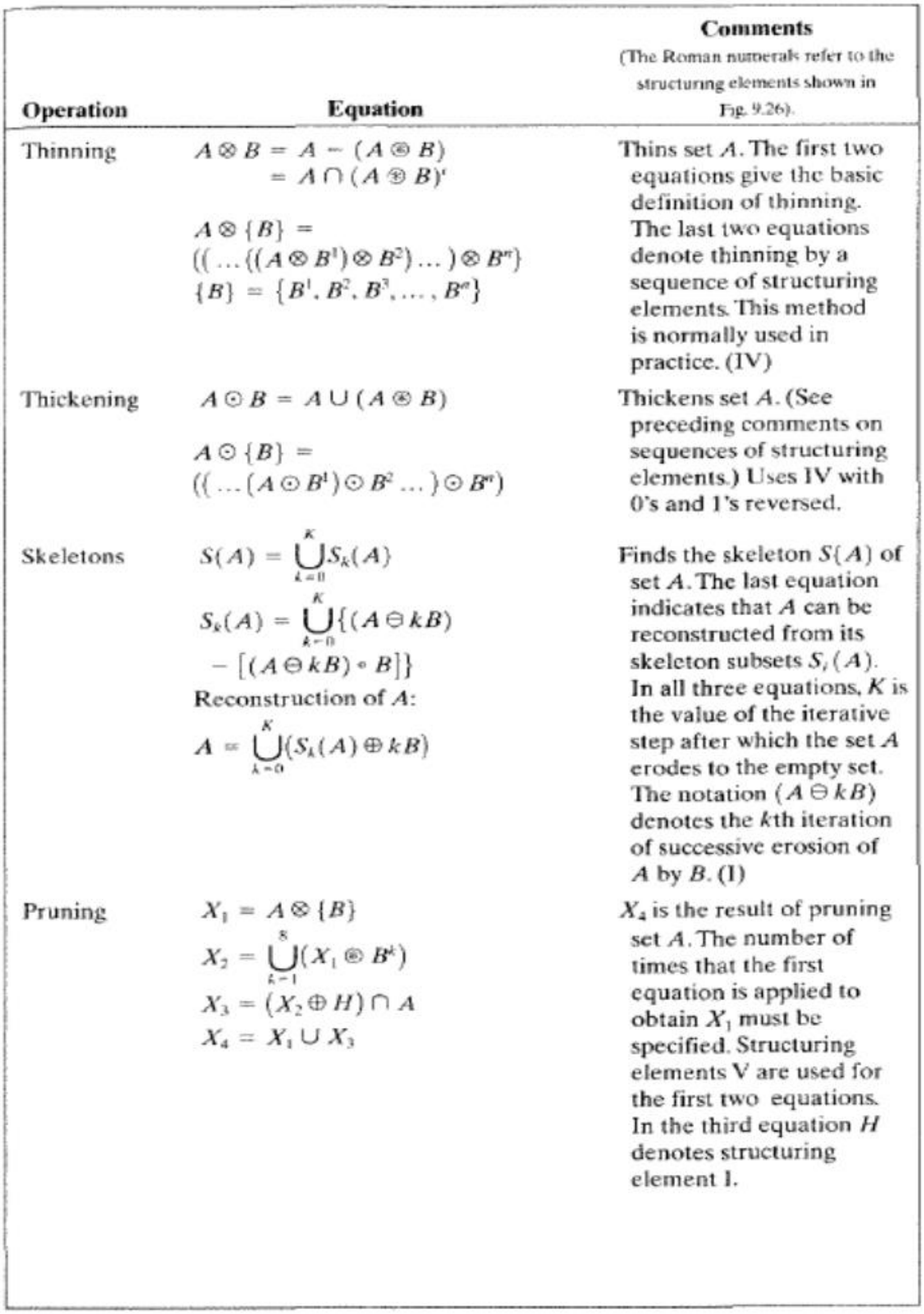
$$X_1 = A \otimes \{B\} \qquad (9.5\text{-}17)$$

where $\{B\}$ denotes the structuring element sequence shown in Figs. 9.25(b) and (c) [see Eq. (9.5-7) regarding structuring-element sequences]. The sequence of

In more complex scenarios. use of Eq. (9.5-19) sometimes picks up the "tips" of some parasitic branches. This condition can occur when the end points of these branches are near the skeleton. Although Eq. (9.5-17) may eliminate them. they can be picked up again during dilation because they are valid points in *A*. Unless entire parasitic elements are picked up again (a rare case if these elements arc short with respect to valid strokes), detecting and eliminating them is easy because they are disconnected regions.



Operation	Equation	**Comments** (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Translation	$(A)_z = \{w \mid w = a + z, \quad \text{for } a \in A\}$	Translates the origin of A to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)
Hit-or-miss transform	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) = (A \ominus B_1) - (A \oplus \hat{B}_2)$	The set of points (coordinates) at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c.
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Region filling	$X_k = (X_{k-1} \oplus B) \cap A^c; X_0 = p$ and $k = 1, 2, 3, \ldots$	Fills a region in A, given a point p in the region. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap A; X_0 = p$ and $k = 1, 2, 3, \ldots$	Finds a connected component Y in A, given a point p in Y. (I)
Convex hull	$X_k^i = (X_{k-1}^i \circledast B^i) \cup A; i = 1, 2, 3, 4;$ $k = 1, 2, 3, \ldots; X_0^i = A$; and $D^i = X^i_{\text{conv}}$.	Finds the convex hull $C(A)$ of set A, where "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. (III)



Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)
Thickening	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $((\ldots(A \odot B^1) \odot B^2 \ldots) \odot B^n)$	Thickens set A. (See preceding comments on sequences of structuring elements.) Uses IV with 0's and 1's reversed.
Skeletons	$S(A) = \bigcup_{k=0}^{K} S_k(A)$ $S_k(A) = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\}$ Reconstruction of A: $A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$	Finds the skeleton $S(A)$ of set A. The last equation indicates that A can be reconstructed from its skeleton subsets $S_i(A)$. In all three equations, K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the kth iteration of successive erosion of A by B. (I)
Pruning	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	X_4 is the result of pruning set A. The number of times that the first equation is applied to obtain X_1 must be specified. Structuring elements V are used for the first two equations. In the third equation H denotes structuring element I.

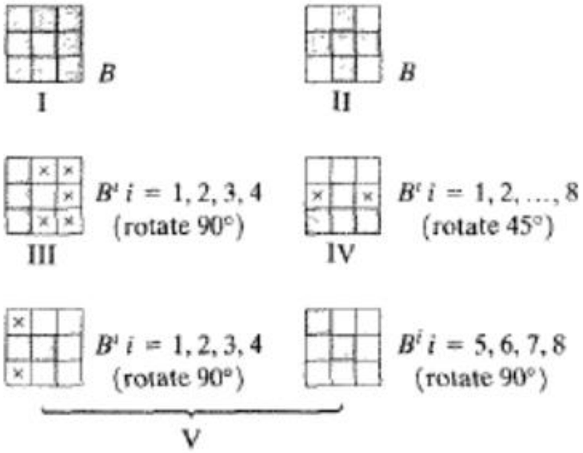
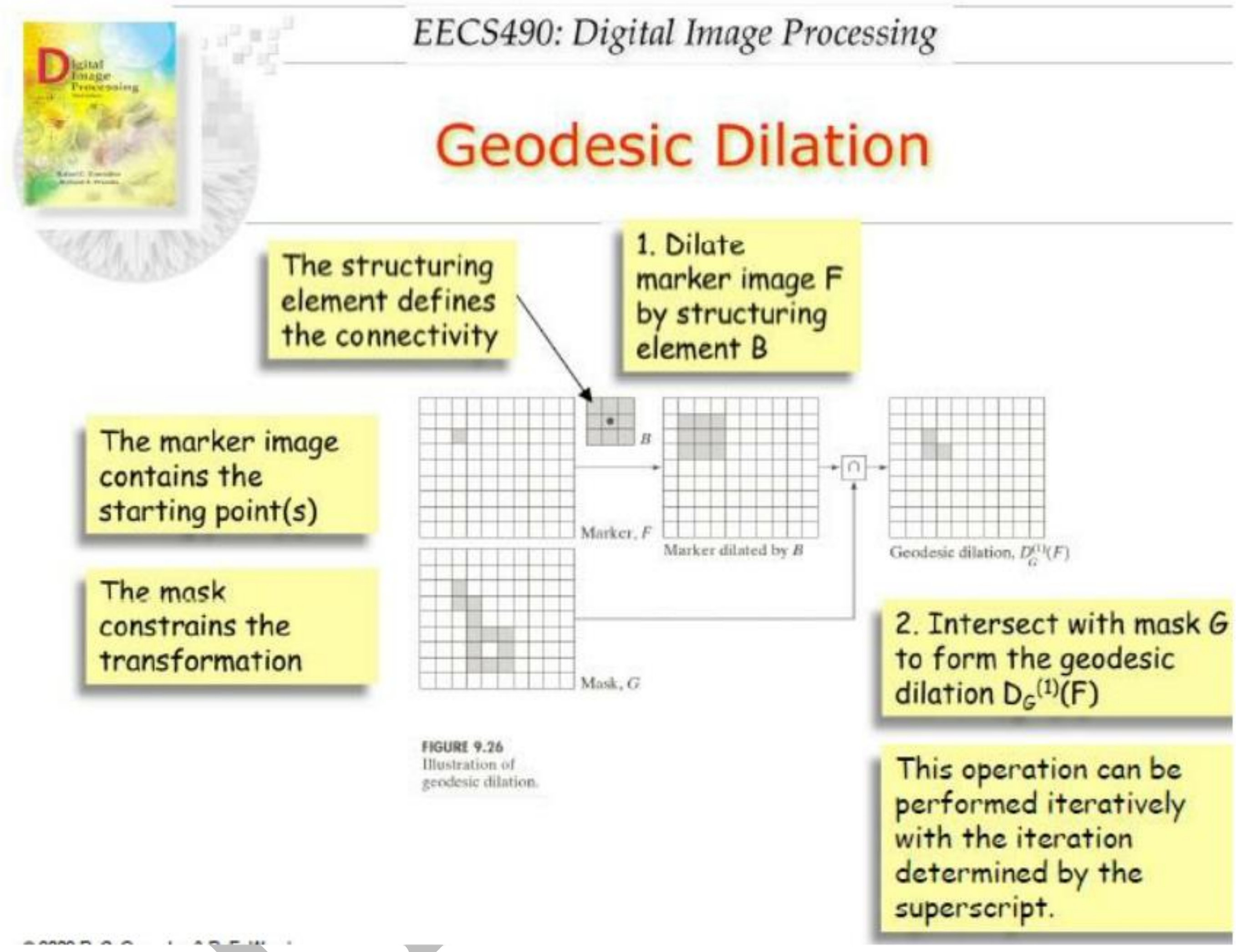


FIGURE 9.26 Five basic types of structuring elements used for binary morphology. The origin of each element is at its center and the ×'s indicate "don't care" values.

GEODESIC DILATION



EECS490: Digital Image Processing

Geodesic Dilation

The structuring element defines the connectivity

1. Dilate marker image F by structuring element B

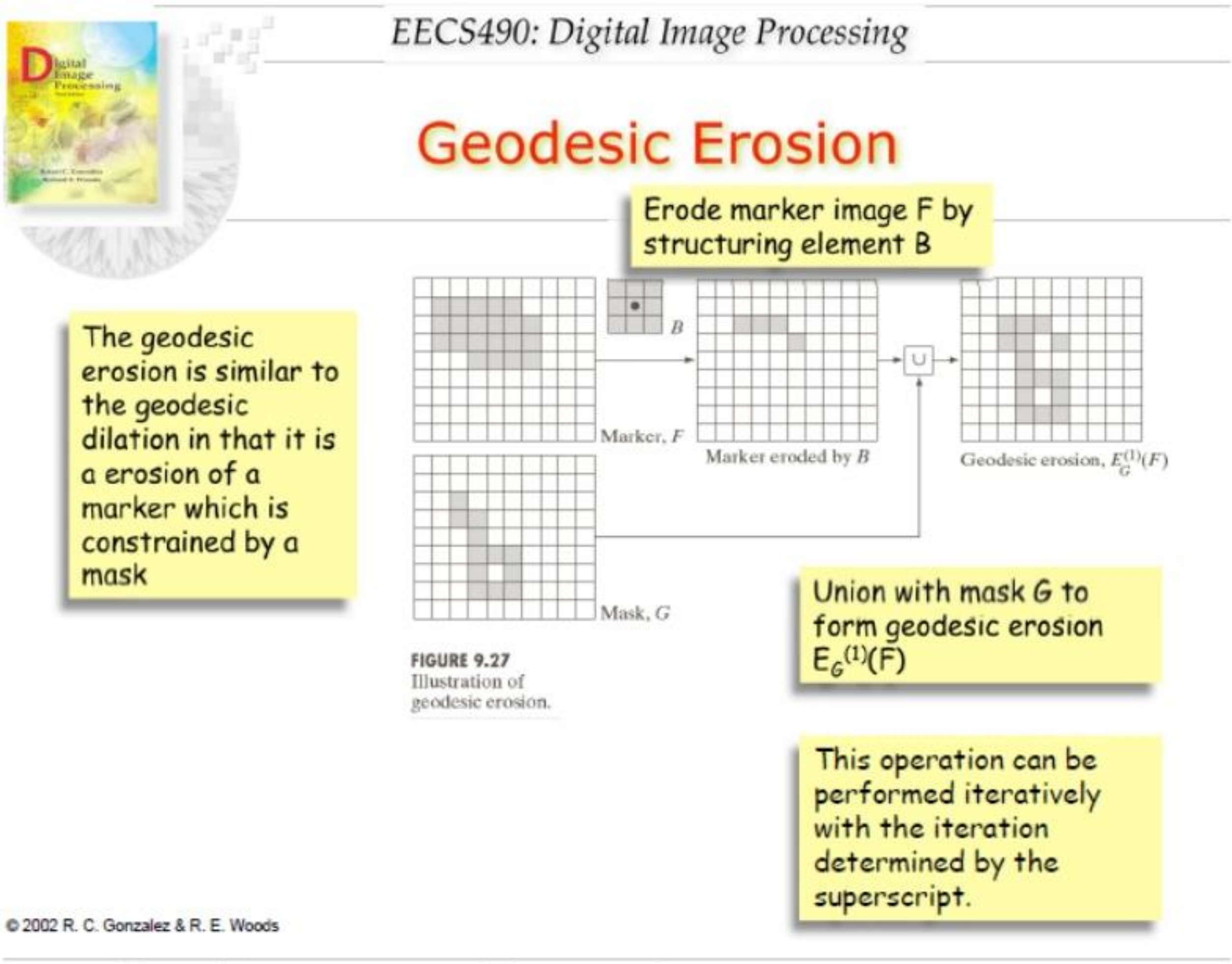
The marker image contains the starting point(s)

The mask constrains the transformation

2. Intersect with mask G to form the geodesic dilation $D_G^{(1)}(F)$

This operation can be performed iteratively with the iteration determined by the superscript.

FIGURE 9.26 Illustration of geodesic dilation.



EECS490: Digital Image Processing

Geodesic Erosion

Erode marker image F by structuring element B

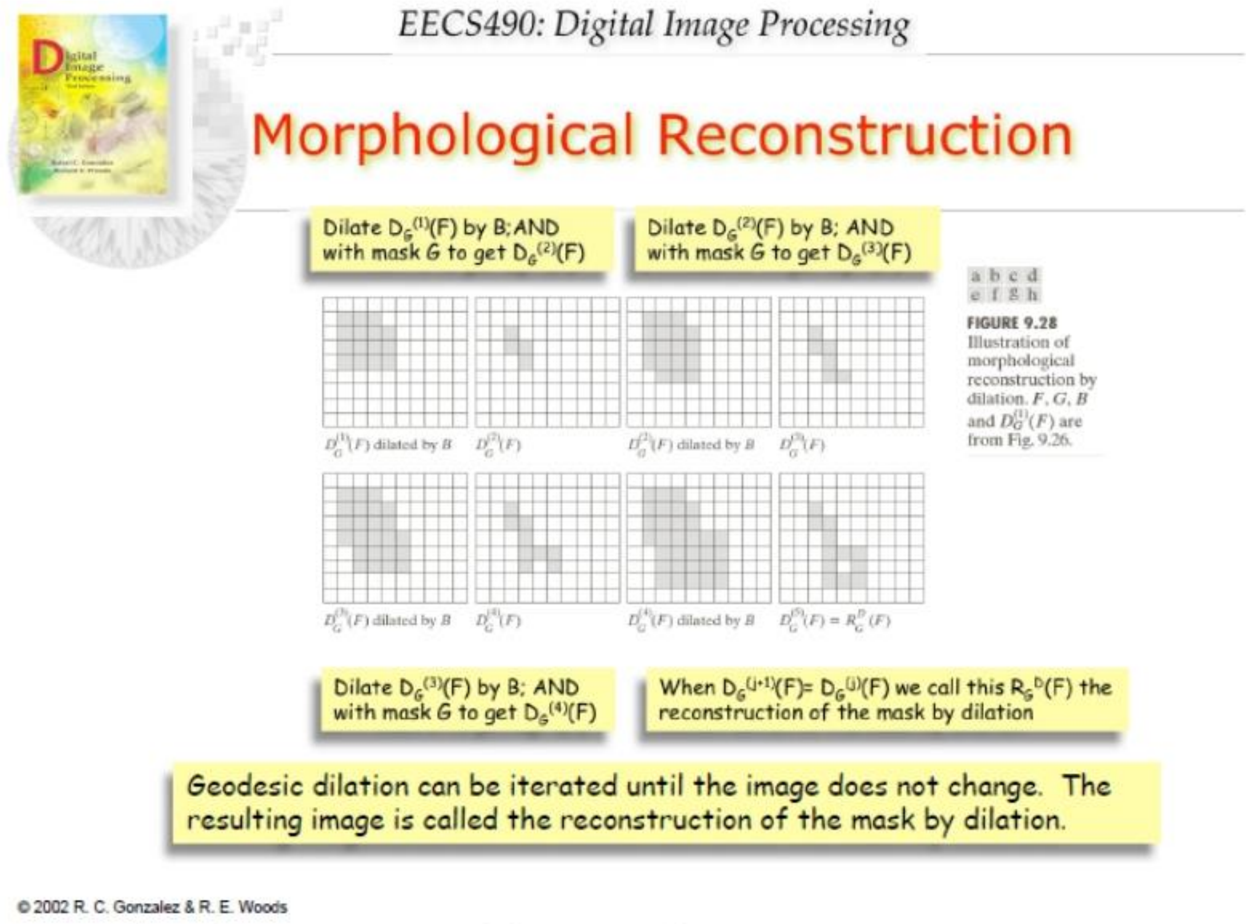
The geodesic erosion is similar to the geodesic dilation in that it is a erosion of a marker which is constrained by a mask

FIGURE 9.27
Illustration of geodesic erosion.

Union with mask G to form geodesic erosion $E_G^{(1)}(F)$

This operation can be performed iteratively with the iteration determined by the superscript.

RECONSTRUCTION BY DILATION AND EROSION



EECS490: Digital Image Processing

Morphological Reconstruction

Dilate $D_G^{(1)}(F)$ by B;AND with mask G to get $D_G^{(2)}(F)$

Dilate $D_G^{(2)}(F)$ by B; AND with mask G to get $D_G^{(3)}(F)$

FIGURE 9.28 Illustration of morphological reconstruction by dilation. F, G, B and $D_G^{(1)}(F)$ are from Fig. 9.26.

Dilate $D_G^{(3)}(F)$ by B; AND with mask G to get $D_G^{(4)}(F)$

When $D_G^{(j+1)}(F) = D_G^{(j)}(F)$ we call this $R_G^D(F)$ the reconstruction of the mask by dilation

Geodesic dilation can be iterated until the image does not change. The resulting image is called the reconstruction of the mask by dilation.

© 2002 R. C. Gonzalez & R. E. Woods

IMAGE SEGMENTATION: BOUNDARY DETECTION BASED TECHNIQUES

Image Segmentation

Segmentation subdivides an image into its constituent regions or objects. The level to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the objects of interest in an application have been isolated. For example, in the automated inspection of electronic assemblies. interest lies in analyzing images of the products with the objective of determining the presence or absence of specific anomalies, such as missing components or broken connection paths. There is no point in carrying segmentation past the level of detail required to identify those elements. Segmentation of nontrivial images is one of the most difficult tasks in image processing.

Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason, considerable care should be taken to improve the probability of rugged segmentation. In some situations, such as industrial inspection applications, at least some measure of control over the environment is possible at times. The experienced image processing system designer invariably pays considerable attention to such opportunities. In other applications, such as autonomous target acquisition, the system designer has no control of the environment. Then the usual approach is to focus on selecting the types of sensors most likely to enhance the objects of interest while diminishing the contribution of irrelevant image detail. A good example is the use of infrared imaging by the military to detect objects with strong heat signatures, such as equipment and troops in motion.

Image segmentation algorithms generally are based on one of two basic properties of intensity values: discontinuity and similarity. In the first category, the approach is to partition an image based on abrupt changes in intensity, such as edges in an image. The principal approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined criteria. Thresholding, region growing, and region splitting and merging are examples of methods in this category.

1. Introduction

· The objective is to subdivide an image into its constituent parts or objects for subsequent processing such as recognition. · It is one of the most important steps leading to the analysis of processed image data.

Complete v.s. patrial segmentation ·
In *complete segmentation,* . Disjoint regions segmented are uniquely corresponding with objects in the input image. . Cooperation with higher processing levels which use specific knowledge of the problem domain is necessary. ·

In *partial segmentation,* . Regions segmented do not correspond directly with image objects. · Totally correct and complete segmentation of complex scenes usually can't be achieved. · A reasonable aim is to use partial segmentation as an input to higher level processing.
Applications: · Simple segmentation problems:
1. Contrasted objects on a uniform background
2. Simple assembly tasks, blood cells, printed characters, etc.

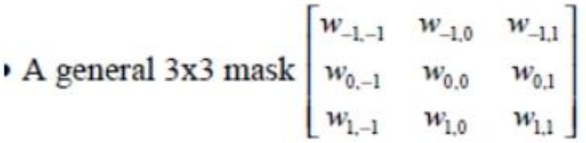
How to achieve segmentation? · Image is divided into separate regions that are homogeneous with respect to a chosen property such as color, brightness, texture, etc. · Segmentation algorithms generally are based on 2 basic properties of gray level values:

1. Discontinuity - isolated points, lines and edges of image.
2. Similarity - thresholding, region growing, region splitting and merging.

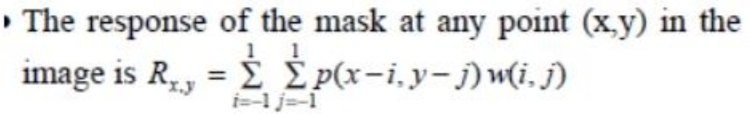
· Segmentation methods:

1. Global approaches such as thresholding
2. Edge-based segmentation 3. Region-based segmentation

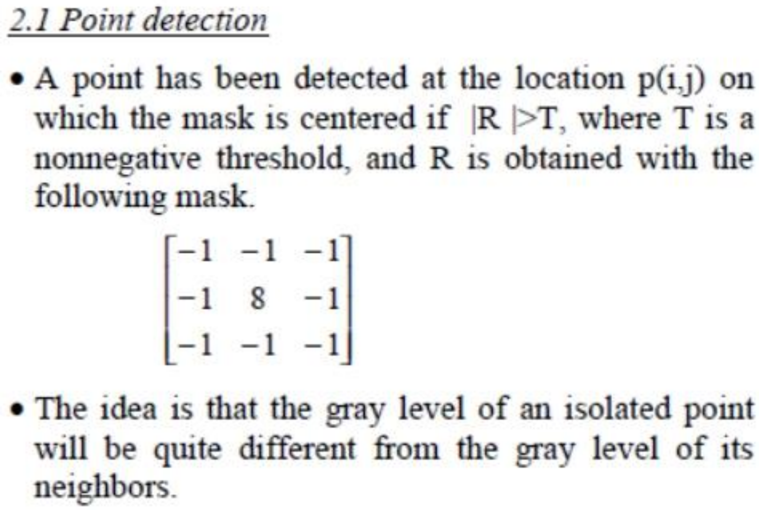
2. *Detection of Discontinuities:* · There are 3 basic types of discontinuities: points, lines and edges. · The detection is based on convoluting the image with a spatial mask.



- A general 3x3 mask $\begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix}$



- The response of the mask at any point (x,y) in the image is $R_{x,y} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} p(x-i, y-j)\, w(i,j)$

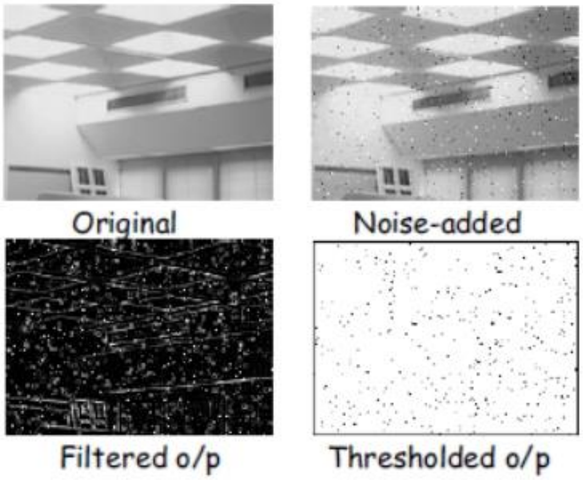


2.1 Point detection

- A point has been detected at the location p(i,j) on which the mask is centered if $|R| > T$, where T is a nonnegative threshold, and R is obtained with the following mask.

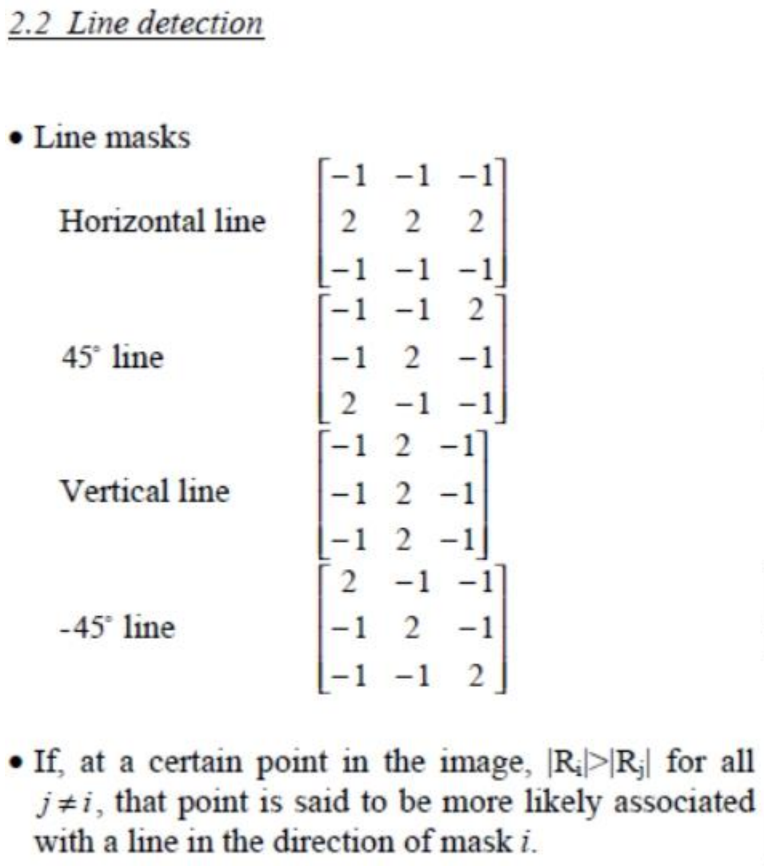
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- The idea is that the gray level of an isolated point will be quite different from the gray level of its neighbors.



Original Noise-added

Filtered o/p Thresholded o/p



2.2 Line detection

- Line masks

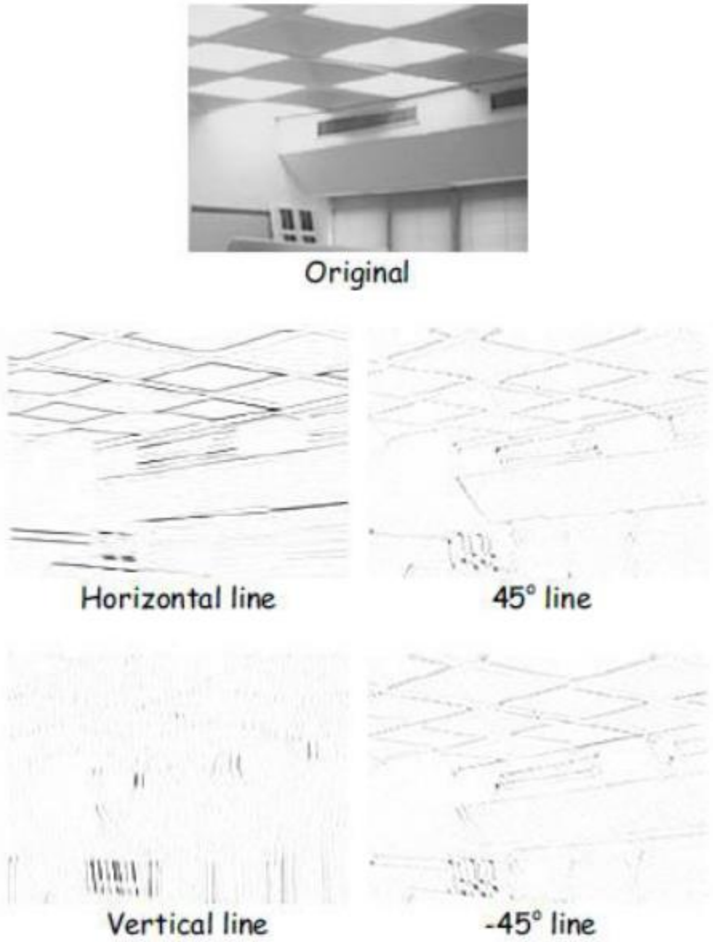
Horizontal line $\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$

45° line $\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$

Vertical line $\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$

-45° line $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

- If, at a certain point in the image, $|R_i|>|R_j|$ for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask *i*.



2.3 Edge detection · It locates sharp changes in the intensity function. · Edges are pixels where brightness changes abruptly. ·

A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function. ·

An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel. ·

Magnitude of the first derivative detects the presence of the edge. · Sign of the second derivative determines whether the edge pixel lies on the dark sign or light side.

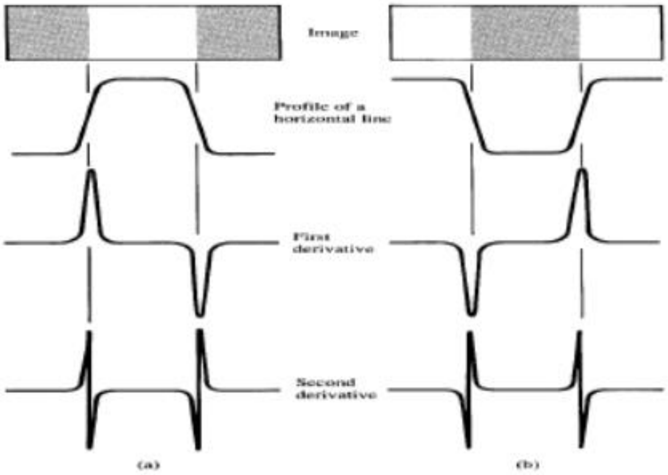


Fig 1. Edge detection by derivative operators: (a) light stripe on a dark background; (b) dark stripe on a light background.

(a) <u>Gradient operator</u>

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

$$\nabla f(x',y') = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}_{(x',y')}$$

- Magnitude of vector $\nabla f(x',y')$:

$$|\nabla f(x',y')| = \left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right)^{\frac{1}{2}} \Bigg|_{(x',y')}$$

- Direction of the vector $\nabla f(x',y')$:

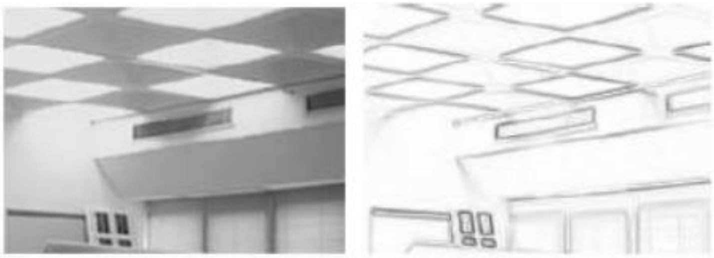
$$\alpha(x',y') = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)\Big|_{(x',y')}$$

- Its magnitude can be approximated in the digital domain in a number of ways, which result in a number of operators such as Roberts, Prewitt and Sobel operators for computing its value.

Sobel operator:

- It provides both a differentiating and a smoothing effect, which is particularly attractive as derivatives typically enhance noise.

$$G_x : \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad G_y : \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Original Processed image

(b) Laplacian Operator

- The Laplacian of a 2D function f(x,y) is a 2nd-order derivative defined as

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

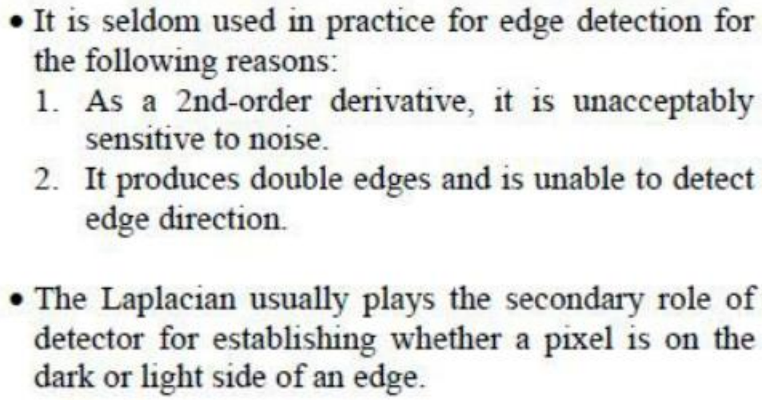
- The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image.

- It can also be implemented in digital form in various ways.

- For a 3x3 region, the mask is given as

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

Original Processed image



- It is seldom used in practice for edge detection for the following reasons:
  1. As a 2nd-order derivative, it is unacceptably sensitive to noise.
  2. It produces double edges and is unable to detect edge direction.
- The Laplacian usually plays the secondary role of detector for establishing whether a pixel is on the dark or light side of an edge.

2.4 Combined Detection:

- Detection of combinations of points, lines and edges can be achieved by using sets of orthogonal masks.
- A set of 9 3X3 masks were proposed by Frei and Chen (1977)

Basis of edge subspace :

$$W_1=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ -1 & -\sqrt{2} & -1\end{bmatrix} \quad W_2=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 0 & -1\\ \sqrt{2} & 0 & -\sqrt{2}\\ 1 & 0 & -1\end{bmatrix}$$

$$W_3=\frac{1}{2\sqrt{2}}\begin{bmatrix}0 & -1 & \sqrt{2}\\ 1 & 0 & -1\\ -\sqrt{2} & 1 & 0\end{bmatrix} \quad W_4=\frac{1}{2\sqrt{2}}\begin{bmatrix}\sqrt{2} & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & -\sqrt{2}\end{bmatrix}$$

Basis of line subspace :

$$W_5=\frac{1}{2}\begin{bmatrix}-1 & 0 & 1\\ 0 & 0 & 0\\ 1 & 0 & -1\end{bmatrix} \qquad W_6=\frac{1}{2}\begin{bmatrix}0 & 1 & 0\\ -1 & 0 & -1\\ 0 & 1 & 0\end{bmatrix}$$

$$W_7=\frac{1}{6}\begin{bmatrix}1 & -2 & 1\\ -2 & 4 & -2\\ 1 & -2 & 1\end{bmatrix} \qquad W_8=\frac{1}{6}\begin{bmatrix}-2 & 1 & -2\\ 1 & 4 & 1\\ -2 & 1 & -2\end{bmatrix}$$

2.5 EDGE LINKING AND BOUNDARY DETECTION ·

The techniques of detecting intensity discontinuities yield pixels lying only on the boundary between regions. · In practice, this set of pixels seldom characterizes a boundary completely because if noise, breaks in boundary from nonuniform illumination, and other effects that introduce spurious intensity discontinuities. ·

Edge detection algorithms are typically followed by linking and other boundary detection procedures designed to assemble edge pixels into meaningful boundaries. (a) *Local processing* ·

Two principal properties used for establishing similarity of edge pixels in this kind of analysis are:

1. The strength of the response of the gradient operator used to produce the edge pixel.
2. The direction of the gradient.

· In a small neighborhood, e.g. 3x3, 5x5, all points with common properties are linked:

- A point (x',y') in the neighborhood of (x,y) is linked to the pixel at (x,y) if both the following magnitude and direction criteria are satisfied.

$$|\nabla f(x',y') - \nabla f(x,y)| \le Threshold\ T_m$$
$$|\alpha(x',y') - \alpha(x,y)| \le Threshold\ T_d$$

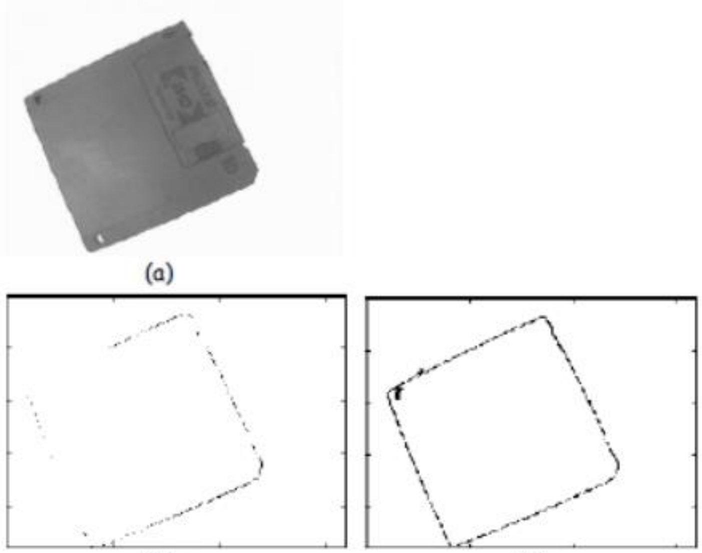

(a)
(b) (c)

Fig 2. (a) Original image; (b) detection result without local processing; (c) detction result with local processing. (Tm = 0.15 x max(|∇f|) and Td = pi/9)

HOUGH TRANSFORM

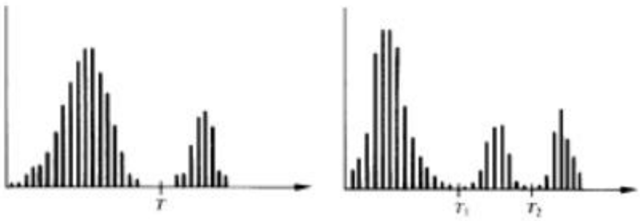
In the edge-linking problem an approach based on the Hough transform is as follows:

1. Compute the gradient of an image and threshold it to obtain a binary image
2. Specify subdivisions in the p0-plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

The concept of continuity in this case usually is based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell. A gap at any point is significant if the distance between that point and its closest neighbor exceeds a certain threshold

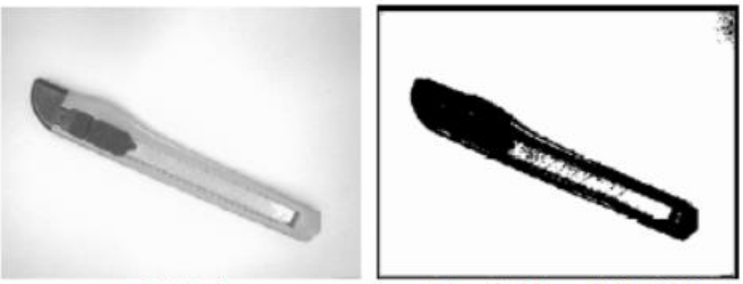
THRESHOLDING

· Thresholding is one of the most important approaches to image segmentation.
· If background and object pixels have gray levels grouped into 2 dominant modes, they can be separated with a threshold easily.

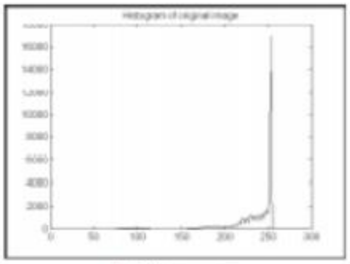


Thresholding may be viewed as an operation that involves tests against a function T of the form T=T[x,y,p(x,y),f(x,y)], where f(x,y) is the gray level of point (x,y), and p(x,y) denotes some local property of this point such as the average gray level of a neighborhood centered on (x,y).

· Special cases:
If T depends on
1. f(x,y) only - global threshold
2. Both f(x,y) & p(x,y) - local threshold
3. (x,y) - dynamic threshold
· Multilevel thresholding is in general less reliable as
it is difficult to establish effective thresholds to
isolate the regions of interest.



Original Threshold result (T=200)



Histogram

Fig 3. Nonadaptive thresholding result

3.1 Adaptive thresholding

· The threshold value varies over the image as a function of local image characteristics.
· Image f is divided into subimages.
· A threshold is determined independently in each subimage.
· If a threshold can't be determined in a subimage, it can be interpolated with thresholds obtained in neighboring subimages.
· Each subimage is then processed with respect to its local threshold.

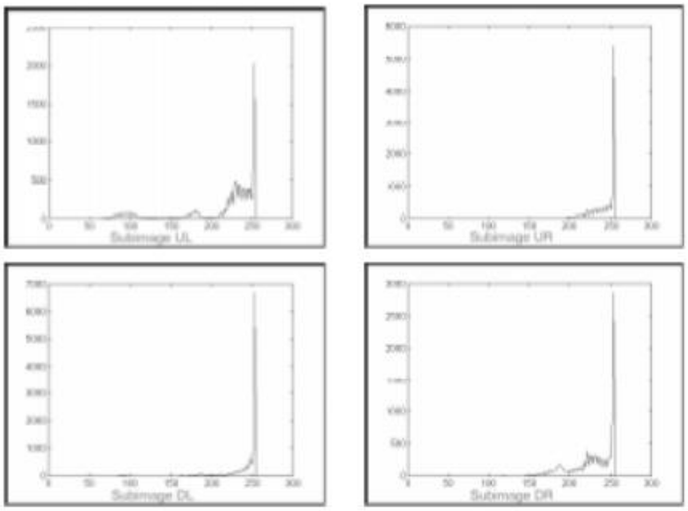


Fig 4. Histogram of the 4 subimages

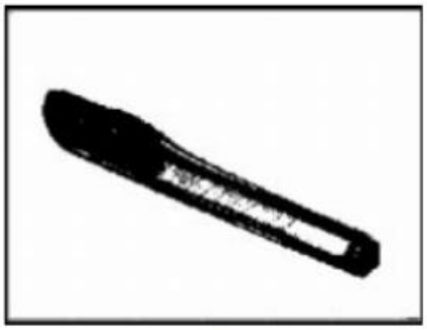


Fig 5. Adaptive thresholding result (T_{UL}=200,T_{UR}=100, T_{DL}=200,T_{DR}=200)

3.2 Threshold selection based on boundary characteristics

· A reliable threshold must be selected to identify the mode peaks of a given histogram.

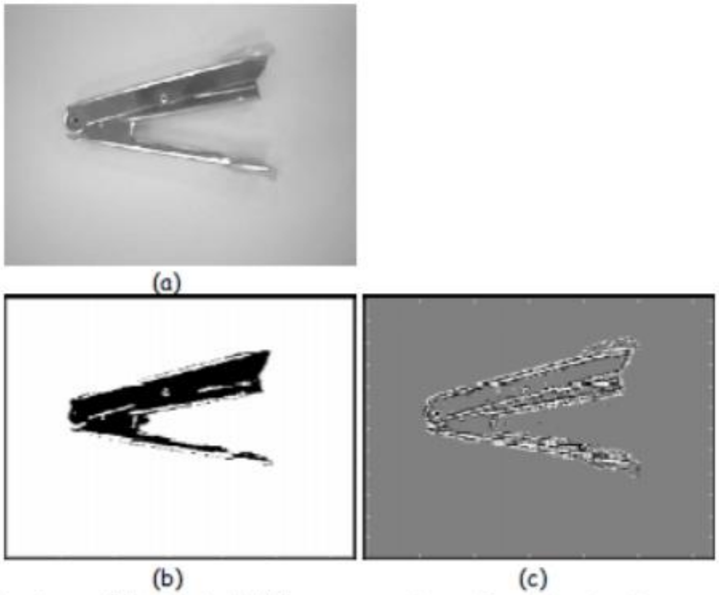
· This capability is very important for automatic threshold selection in situations where image characteristics can change over a broad range of intensity distributions. ·

We can consider only those pixels that lie on or near the boundary between objects and the background such that the associated histogram is well-shaped to provide a good chance for us to select a good threshold. · The gradient can indicate if a pixel is on an edge or not. ·

The Laplacian can tell if a given pixel lies on the dark or light (background or object) side of an edge. · The gradient and laplacian can produce a 3-level Image

$$s(x,y)=\begin{cases}0 & \text{if } G[f(x,y)]<T\\ 1 & \text{if } G[f(x,y)]\geq T \text{ and } L[f(x,y)]\geq 0\\ -1 & \text{if } G[f(x,y)]\geq T \text{ and } L[f(x,y)]<0\end{cases}$$

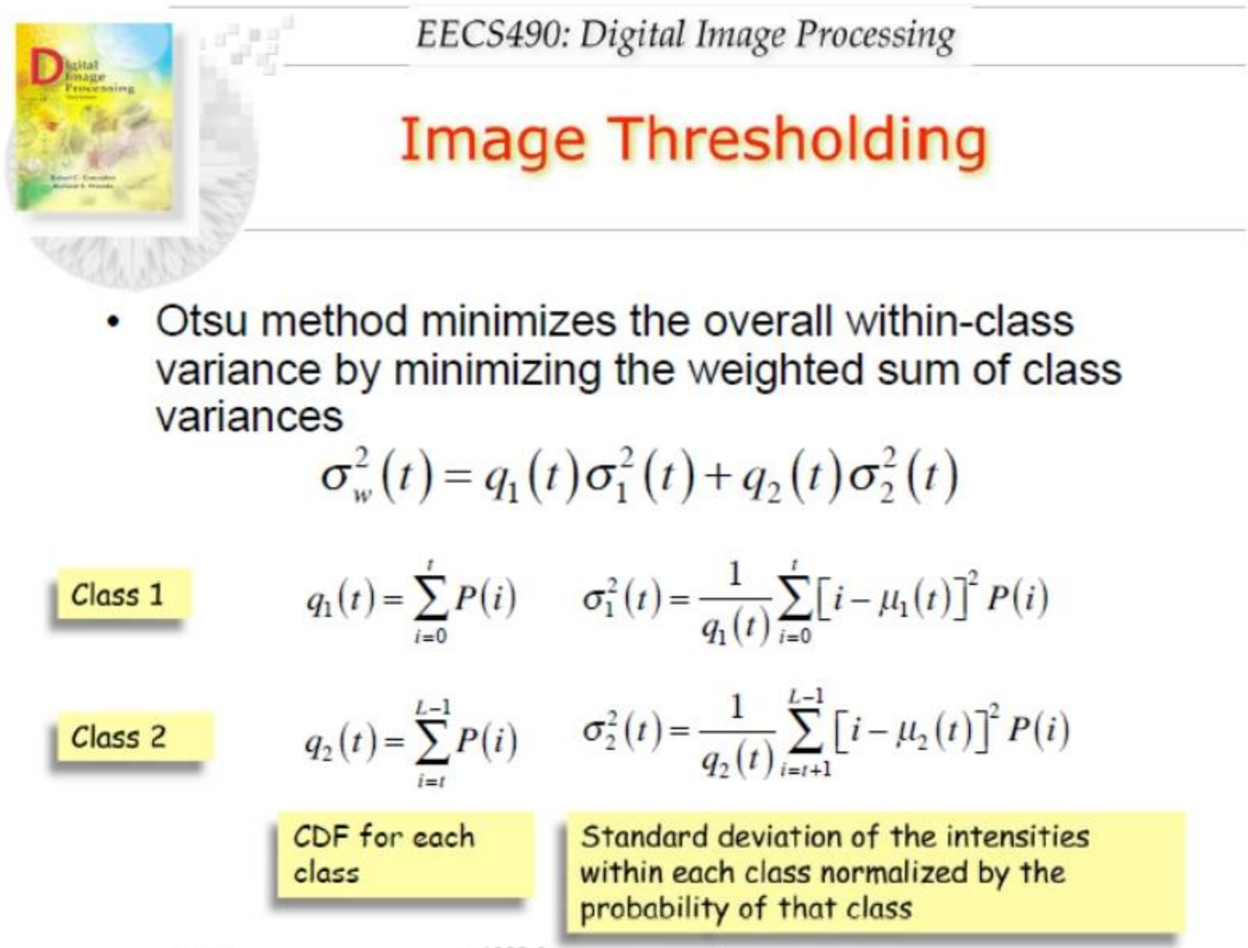
where T is a threshold.



(a)

(b) (c)

Fig 6. (a) Original, (b) processed result without using boundary characteristic and (c) processed result with using boundary characteristic.



EECS490: Digital Image Processing

Image Thresholding

- Otsu method minimizes the overall within-class variance by minimizing the weighted sum of class variances

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Class 1

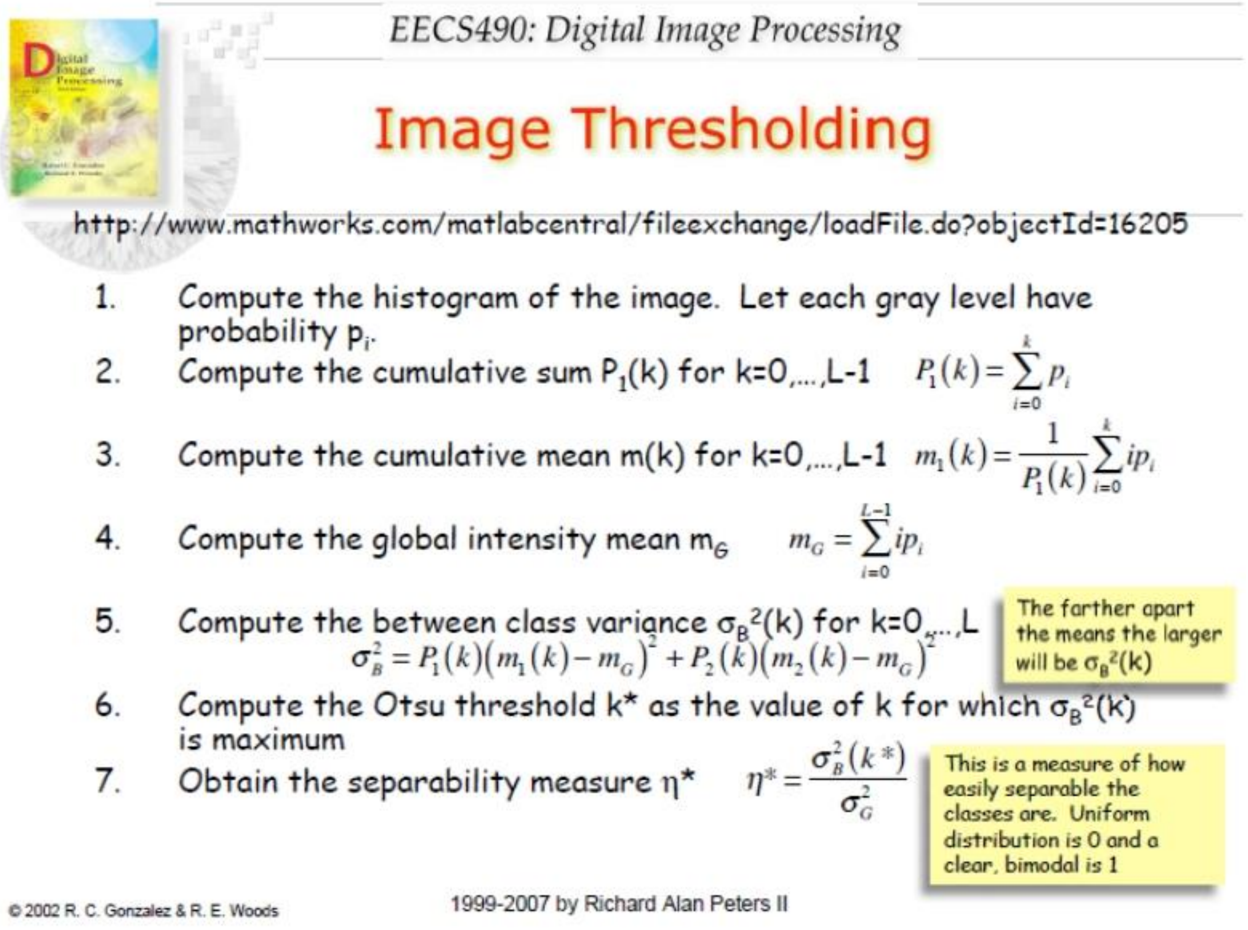
$$q_1(t) = \sum_{i=0}^{t} P(i) \qquad \sigma_1^2(t) = \frac{1}{q_1(t)}\sum_{i=0}^{t}\left[i - \mu_1(t)\right]^2 P(i)$$

Class 2

$$q_2(t) = \sum_{i=t}^{L-1} P(i) \qquad \sigma_2^2(t) = \frac{1}{q_2(t)}\sum_{i=t+1}^{L-1}\left[i - \mu_2(t)\right]^2 P(i)$$

CDF for each class

Standard deviation of the intensities within each class normalized by the probability of that class



EECS490: Digital Image Processing

Image Thresholding

http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=16205

1. Compute the histogram of the image. Let each gray level have probability p_i.
2. Compute the cumulative sum $P_1(k)$ for k=0,…,L-1 $P_1(k)=\sum_{i=0}^{k} p_i$
3. Compute the cumulative mean m(k) for k=0,…,L-1 $m_1(k)=\frac{1}{P_1(k)}\sum_{i=0}^{k} ip_i$
4. Compute the global intensity mean m_G $m_G=\sum_{i=0}^{L-1} ip_i$
5. Compute the between class variance $\sigma_B^2(k)$ for k=0,…,L $\sigma_B^2 = P_1(k)(m_1(k)-m_G)^2+P_2(k)(m_2(k)-m_G)^2$
6. Compute the Otsu threshold k* as the value of k for which $\sigma_B^2(k)$ is maximum
7. Obtain the separability measure η* $\eta^*=\frac{\sigma_B^2(k^*)}{\sigma_G^2}$

The farther apart the means the larger will be $\sigma_B^2(k)$

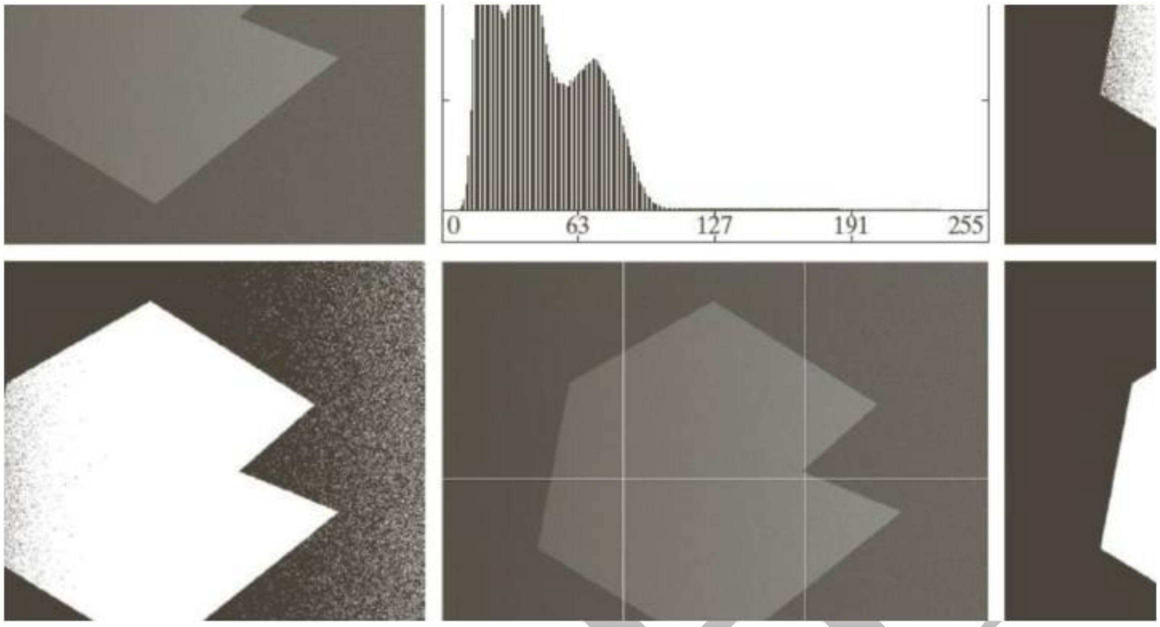
This is a measure of how easily separable the classes are. Uniform distribution is 0 and a clear, bimodal is 1

© 2002 R. C. Gonzalez & R. E. Woods

1999-2007 by Richard Alan Peters II

Variable Thresholding

OtsuImage partitioning.The image is sub-divided and the method is applied to every sub-image. Useful for illumination non-uniformity correction.


Shaded image Histogram

Moving averages.

- Generally used along lines, columns or in zigzag .
- Useful in document image processing.
- Let zk+1 be the intensity of a pixel in the scanning sequence at step k+1.

Segmentation is then performed at each point comparing the pixel value to a fraction of the moving average.

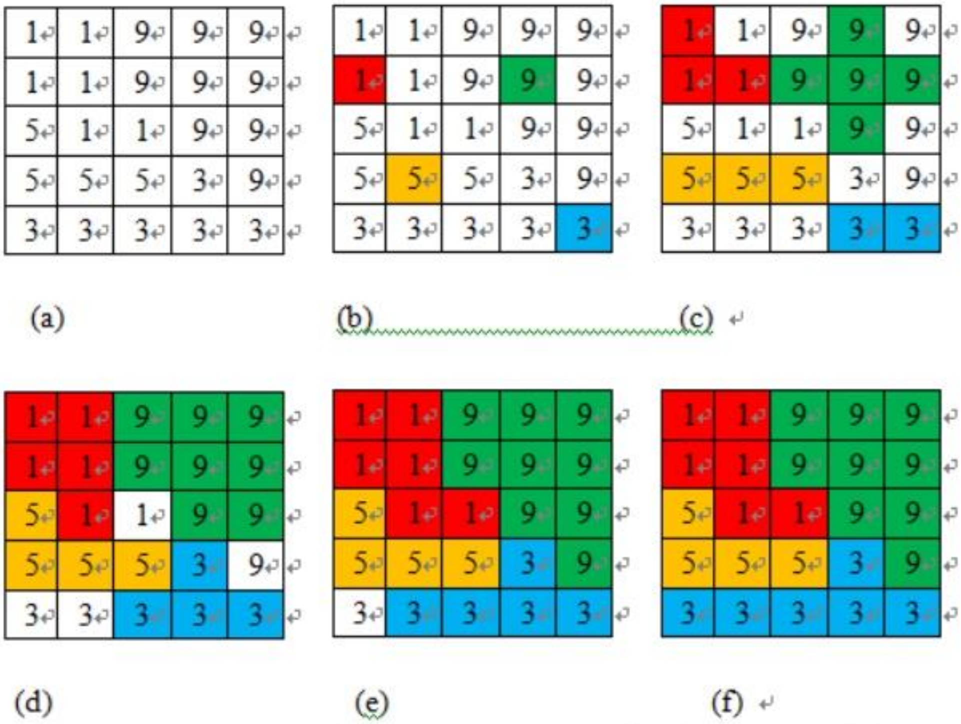
Region Growing

Region growing segmentation is an approach to examine the neighboring pixels of the initial "seed points" and determine if the pixels are added to the seed point or not.

Step1. Selecting a set of one or more starting point (seed) often can be based on the nature of the problem.

Step2. The region are grown from these seed points to adjacent point depending on a threshold or criteria(8-connected) we make.

Step3. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region



1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(a)

1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(b)

1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(c)

1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(d)

1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(e)

1	1	9	9	9
1	1	9	9	9
5	1	1	9	9
5	5	5	3	9
3	3	3	3	3

(f)

Figure Error! No text of specified style in document.**.1 (a)Original image (b)Use step 1 to find seed based on the nature problem.(c) Use Step 2(4-connected here) to growing the region and finding the similar point. (d)(e) repeat Step 2. Until no more pixels satisfy the criteria. (f) The final image.**

Then we can conclude several important issues about region growing：

1. The suitable selection of seed points is important. The selection of seed points is depending on the users.
2. More information of the image is better. Obviously, the connectivity or pixel adjacent information is helpful for us to determine the threshold and seed points.
3. The value, “minimum area threshold”. No region in region growing method result will be smaller than this threshold in the segmented image.
4. The value, “Similarity threshold value“. If the difference of pixel-value or the difference value of average gray level of a set of pixels less than “Similarity threshold value”, the regions will be considered as a same region.
5. The result of an image after region growing still have point’s gray-level higher than the threshold but not connected with the object in image.

We briefly conclude the advantages and disadvantages of region growing.

Advantages：

1. Region growing methods can correctly separate the regions that have the same properties we define.

2. Region growing methods can provide the original images which have clear edges the good segmentation results.
3. The concept is simple. We only need a small numbers of seed point to represent the property we want, then grow the region.
4. We can choose the multiple criteria at the same time.

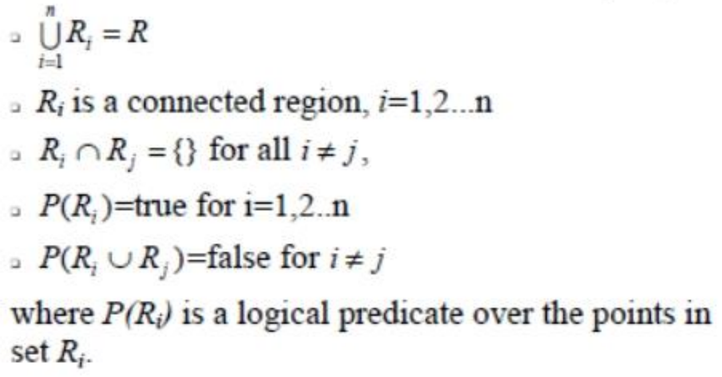
It performs well with respect to noise, which means it has a good shape matching of its result.

Disadvantage :

1. The computation is consuming, no matter the time or power.
2. This method may not distinguish the shading of the real images.

In conclusion, the region growing method has a good performance with the good shape matching and connectivity. The most serious problem of region growing method is the time consuming.

4. *Region-oriented segmentation* · In previous methods, we partition an image into regions by finding boundaries between regions based on intensity discontinuities. · Here, segmentation is accomplished via thresholds based on the distribution of pixel properties, such as intensity or color. · Basic formulation: Let *R* represents the entire image which is partitioned into subregions *R1*, *R2*,...*Rn* such that



- $\bigcup_{i=1}^{n} R_i = R$
- R_i is a connected region, i=1,2...n
- $R_i \cap R_j = \{\}$ for all $i \neq j$,
- $P(R_i)$=true for i=1,2..n
- $P(R_i \cup R_j)$=false for $i \neq j$

where $P(R_i)$ is a logical predicate over the points in set R_i.

· Physical meaning of the formulation: .

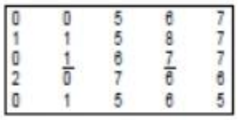
The segmentation must be complete, i.e. every point must be in a region. . Points in a region must be connected. . The regions must be disjoint. .

It deals with the properties that must be satisfied by the pixels in a segmented region - for example *P(Ri)*=true if all pixels in *Ri* have the same intensity. .

Regions *Ri* and *Rj* are different in the sense of predicate *P*.

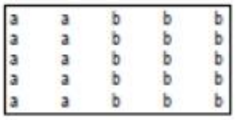
4.1 Region growing by pixel aggregation

- Region growing is a procedure that groups pixels or subregions into larger regions.
- Pixel aggregation starts with a set of "seed" points from those grows by appending to each seed point those neighboring pixels that have similar properties such as gray level, texture and color.



0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Original intensity array



a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

Result of threshold=3

a	a	a	b	b
a	a	a	b	b
a	a	b	b	b
a	a	b	b	b
a	a	a	b	?

Result of Threshold=5.5

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Result of threshold=9

Example of region growing using known starting points

(a) (b) (c)

Fig 7. Original image with seed point, (b) early stage of region growth, (c) final region.

- Problems have to be resolved:
  1. Selection of initial seeds that properly represent regions of interest.
  2. Selection of suitable properties for including points in the various regions during the growing process.
  3. The formulation of stopping rule.

4.2 Region splitting and merging

- 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 stated above.
- A split and merge algorithm is summarized by the following procedure in which, at each step, we:
  1. (1) split into 4 disjointed quadrants any regions R_i where $P(R_i)$=false;
  2. (2) merge any adjacent regions R_j and R_k for which $P(R_i \cup R_j)$=true; and
  3. (3) stop when no further merging or splitting is possible.

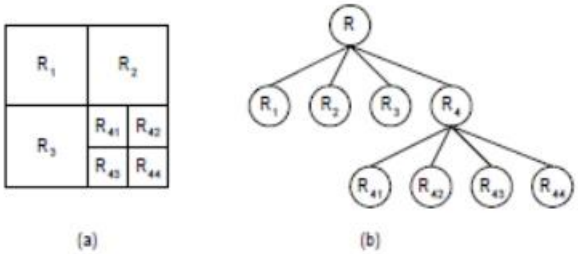
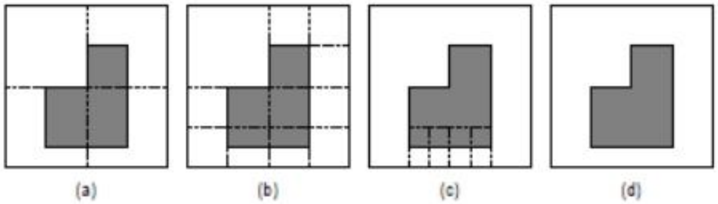


Fig 8. Partitioned image and corresponding quadtree

- Example:



Example of split-and-merge algorithm

(a) The entire image is split into 4 quadrants.

(b) Only the top left region satisfies the predicate so it is not changed, while the other 3 quadrants are split into subquadrants.

(c) At this point several regions can be merged, with the exception of the 2 subquadrants that include the lower part of the object; these do not satisfy the predicate and must be split further.

(a) (b) (c)

Fig 9. (a) Original image, (b) Result of split and merge algorithm; (c) Result of thresholding (b).

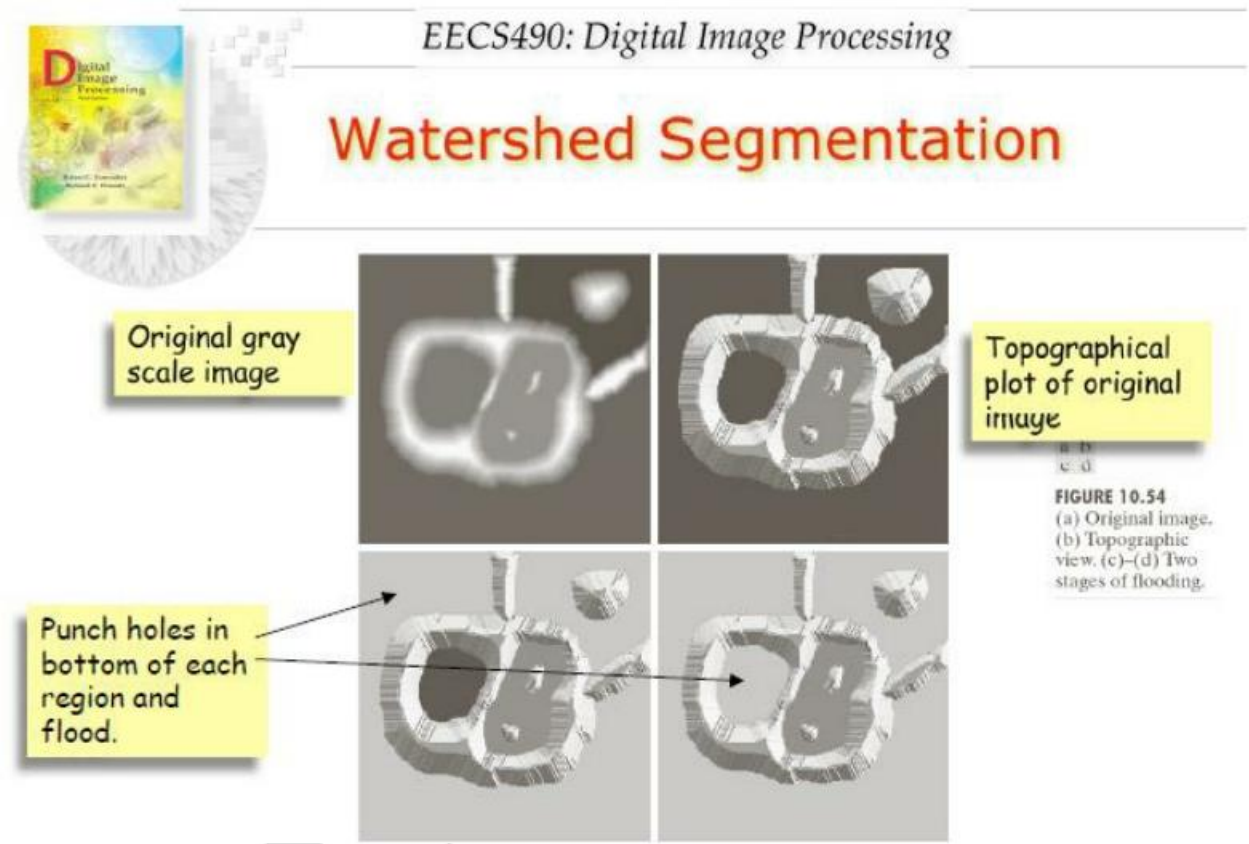
- Image segmentation is a preliminary step in most automatic pictorial pattern-recognition and scene-analysis problems.
- The choice of one segmentation technique over another is dicated mostly by the peculiar characteristics of the problem being considered.

WATERSHED ALGORITHM

The concept of watersheds is based on visualizing an image in three dimensions: two spatial coordinates versus gray levels. In such a "topographic" interpretation, we consider three types of points: (a) points belonging to a regional minimum; (b) points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum; and (c) points at which water would be equally likely to fall to more than one such minimum. For a particular regional minimum, the set of points satisfying condition (b) is called the *catchment basin* or *watershed* of that minimum.

The points satisfying condition (c) form crest lines on the topographic surface and are termed *divide lines* or *watershed lines.* The principal objective of segmentation algorithms based on these concepts is to find the watershed lines. The basic idea is simple: Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging, The flooding will eventually reach a stage when only the tops of the dams are visible above the water line, These dam boundaries correspond to the divide lines of the watersheds,

Therefore, they are the (continuous) boundaries extracted by a watershed segmentation algorithm, These ideas can be explained further with the aid of Fig, 10.44, Figure 10.44(a) shows a simple gray-scale image and Fig, I0.44(b) is a topographic view, in which the height of the "mountains" is proportional to gray-level values in the input image, For ease of interpretation, the backsides of structures are shaded, 111is is not to be confused with gray-level values; only the general topography of the three-dimensional representation is of interest. In order to prevent the rising water from spilling out through the edges of the structure, we imagine the perimeter of the entire topography (image) being enclosed by dams of height greater than the highest possible mountain, whose value is determined by the highest possible gray-level value in the input image,



EECS490: Digital Image Processing

Watershed Segmentation

Original gray scale image

Topographical plot of original image

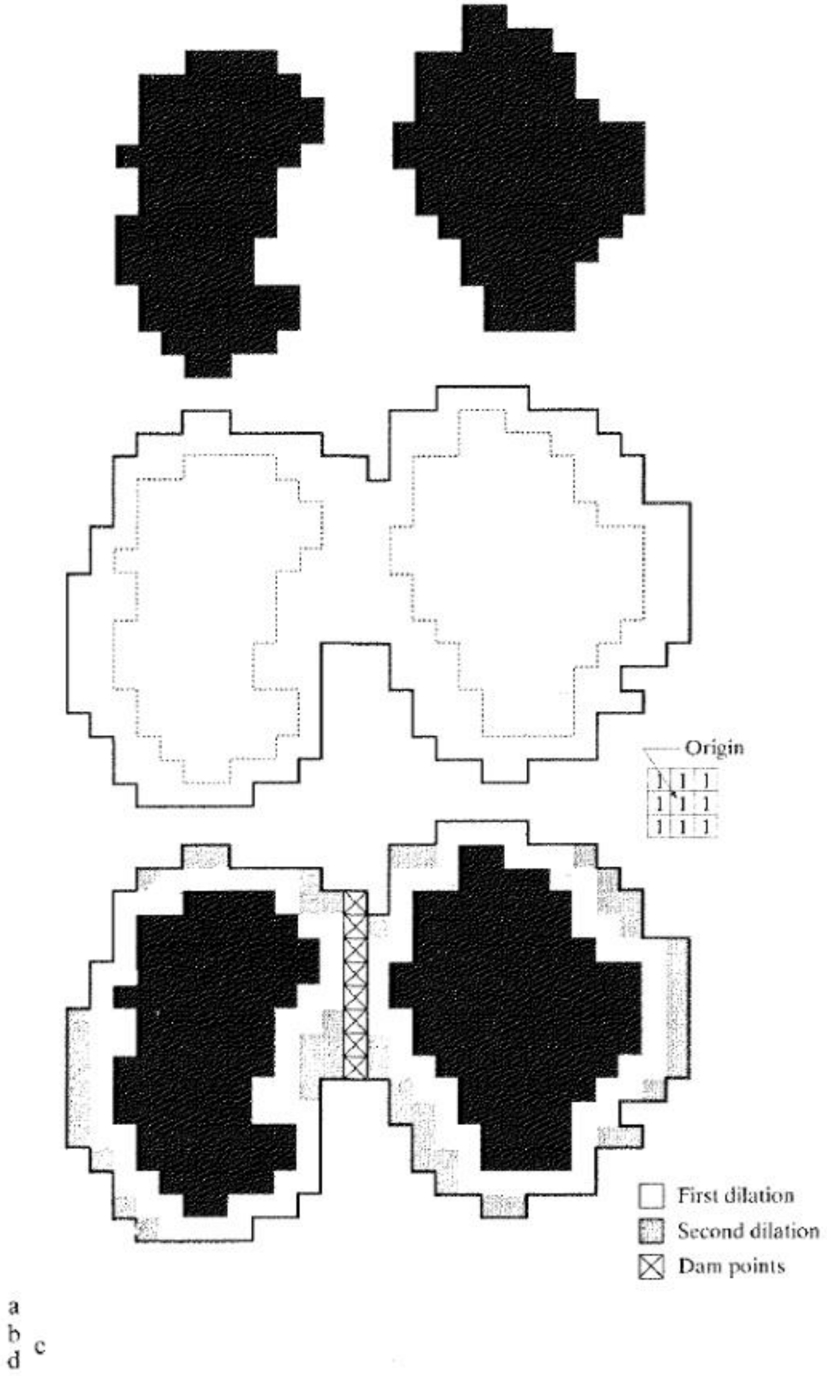
Punch holes in bottom of each region and flood.

FIGURE 10.54 (a) Original image. (b) Topographic view. (c)–(d) Two stages of flooding.

Dam Construction

Before proceeding, let us consider how to construct the dams or watershed lines required by watershed segmentation algorithms. Dam construction is based on binary images, which are members of 2-D integer space Z2 (see Section 2.4.2). The simplest way to construct dams separating sets of binary points is to use morphological dilation (see Section 9.2.1). The basics of how to construct dams using dilation are illustrated in Fig. 10.45. Figure 1O.45(a) shows portions of two catchment basins at flooding step *n* - I and Fig. 1O.45(b) shows the result at the next flooding step, *n*.

The water has spilled from one basin to the other and, therefore, a dam must be built to keep this from happening. In order to be consistent with notation to be introduced shortly, let M1 and *M2* denote the sets of coordinates of points in two regional minima. Then let the set of coordinates of points in the *catchment basin* associated with these two minima at stage *n* - 1 of flooding be denoted by $C_{n-I}(M1)$ and *Cn* - I(*M2*), respectively.



10.5.3 Watershed Segmentation Algorithm

Let $M_1, M_2, \ldots, M_R$ be sets denoting the *coordinates* of the points in the regional minima of an image $g(x, y)$. As indicated at the end of Section 10.5.1, this typically will be a gradient image. Let $C(M_i)$ be a set denoting the coordinates of the points in the catchment basin associated with regional minimum M_i (recall that the points in any catchment basin form a connected component). The notation min and max will be used to denote the minimum and maximum values of $g(x, y)$. Finally, let $T[n]$ represent the set of coordinates (s, t) for which $g(s, t) < n$. That is,

$$T[n] = \{(s, t) \mid g(s, t) < n\}. \qquad (10.5\text{-}1)$$

Geometrically, $T[n]$ is the set of coordinates of points in $g(x, y)$ lying below the plane $g(x, y) = n$.

The topography will be flooded in *integer* flood increments, from $n = \min + 1$ to $n = \max + 1$. At any step n of the flooding process, the algorithm needs to know the number of points below the flood depth. Conceptually, suppose that the coordinates in $T[n]$ that are below the plane $g(x, y) = n$ are "marked" black, and all other coordinates are marked white. Then when we look "down" on the xy-plane at any increment n of flooding, we will see a binary image in which black points correspond to points in the function that are below the plane $g(x, y) = n$. This interpretation is quite useful in helping understand the following discussion.

Let $C_n(M_i)$ denote the set of coordinates of points in the catchment basin associated with minimum M_i that are flooded at stage n. With reference to the discussion in the previous paragraph, $C_n(M_i)$ may be viewed as a binary image given by

$$C_n(M_i) = C(M_i) \cap T[n]. \qquad (10.5\text{-}2)$$

In other words, $C_n(M_i) = 1$ at location (x, y) if $(x, y) \in C(M_i)$ AND $(x, y) \in T[n]$; otherwise $C_n(M_i) = 0$. The geometrical interpretation of this result is straightforward. We are simply using the AND operator to isolate at stage n of flooding the portion of the binary image in $T[n]$ that is associated with regional minimum M_i.

Next, we let $C[n]$ denote the union of the flooded catchment basins portions at stage n:

$$C[n] = \bigcup_{i=1}^{R} C_n(M_i). \tag{10.5-3}$$

Then $C[\max + 1]$ is the union of all catchment basins:

$$C[\max + 1] = \bigcup_{i=1}^{R} C(M_i). \tag{10.5-4}$$

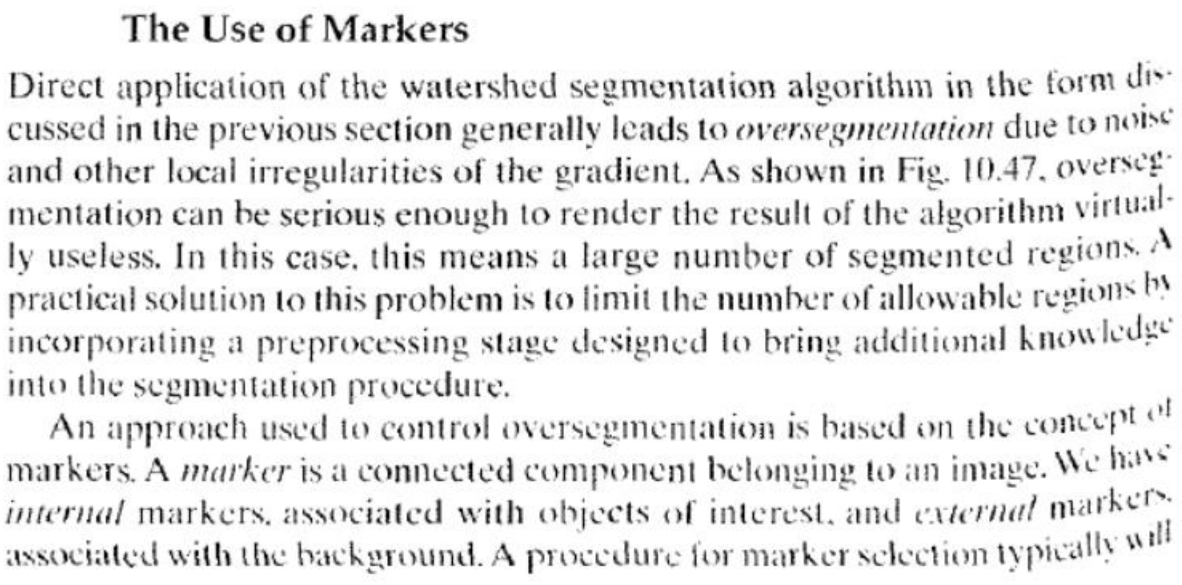
It can be shown (Problem 10.29) that the elements in both $C_n(M_i)$ and $T[n]$ are never replaced during execution of the algorithm, and that the number of elements in these two sets either increases or remains the same as n increases. Thus, it follows that $C[n-1]$ is a subset of $C[n]$. According to Eqs. (10.5-2) and (10.5-3), $C[n]$ is a subset of $T[n]$, so it follows that $C[n-1]$ is a subset of $T[n]$. From this we have the important result that each connected component of $C[n-1]$ is contained in exactly one connected component of $T[n]$.

The algorithm for finding the watershed lines is initialized with $C[\min + 1] = T[\min + 1]$. The algorithm then proceeds recursively, assuming at step n that $C[n-1]$ has been constructed. A procedure for obtaining $C[n]$ from $C[n-1]$ is as follows. Let Q denote the set of connected components in $T[n]$. Then, for each connected component $q \in Q[n]$, there are three possibilities:

(a) $q \cap C[n-1]$ is empty.
(b) $q \cap C[n-1]$ contains one connected component of $C[n-1]$.
(c) $q \cap C[n-1]$ contains more than one connected component of $C[n-1]$.

Construction of $C[n]$ from $C[n-1]$ depends on which of these three conditions holds. Condition (a) occurs when a new minimum is encountered, in which case connected component q is incorporated into $C[n-1]$ to form $C[n]$. Condition (b) occurs when q lies within the catchment basin of some regional minimum, in which case q is incorporated into $C[n-1]$ to form $C[n]$. Condition (c) occurs when all, or part, of a ridge separating two or more catchment basins is encountered. Further flooding would cause the water level in these catchment basins to merge. Thus a dam (or dams if more than two catchment basins are involved) must be built within q to prevent overflow between the catchment basins. As explained in the previous section, a one-pixel-thick dam can be constructed when needed by dilating $q \cap C[n-1]$ with a 3 × 3 structuring element of 1's, and constraining the dilation to q.

Algorithm efficiency is improved by using only values of n that correspond to existing gray-level values in $g(x, y)$; we can determine these values, as well as the values of min and max, from the histogram of $g(x, y)$.



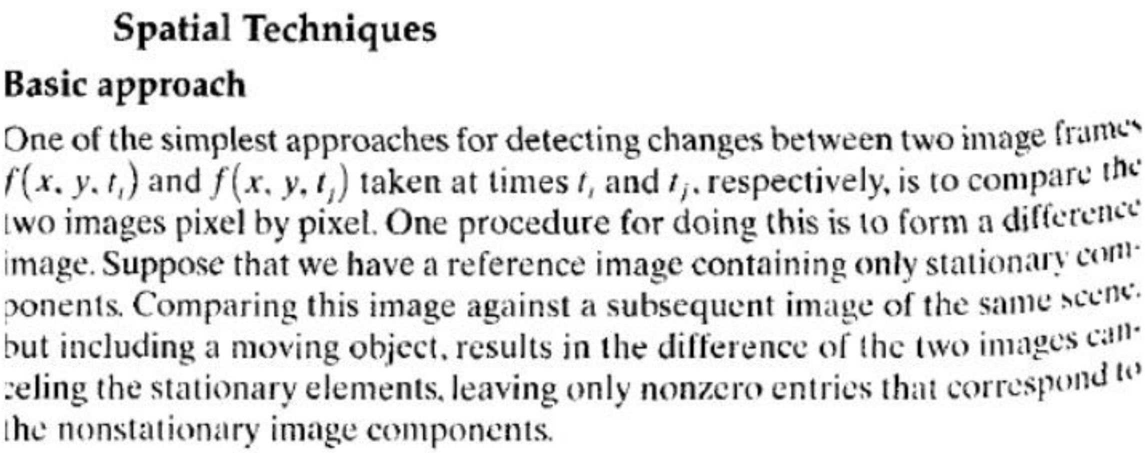
The Use of Markers

Direct application of the watershed segmentation algorithm in the form discussed in the previous section generally leads to *oversegmentation* due to noise and other local irregularities of the gradient. As shown in Fig. 10.47, oversegmentation can be serious enough to render the result of the algorithm virtually useless. In this case, this means a large number of segmented regions. A practical solution to this problem is to limit the number of allowable regions by incorporating a preprocessing stage designed to bring additional knowledge into the segmentation procedure.

An approach used to control oversegmentation is based on the concept of markers. A *marker* is a connected component belonging to an image. We have *internal* markers, associated with objects of interest, and *external* markers, associated with the background. A procedure for marker selection typically will

THE USE OF MOTION IN SEGMENTATION

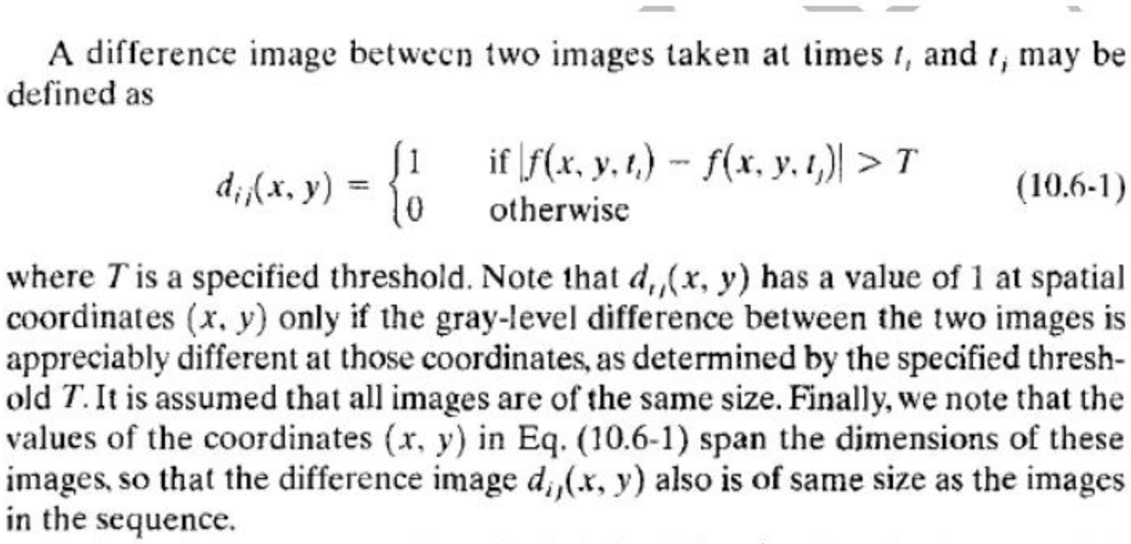
Motion is a powerful cue used by humans and many animals to extract object> of interest from a background of irrelevant detail. In imaging applications. motion arises from a relative displacement between the sensing system and the scene being viewed, such as in robotic applications, autonomous navigation., and dynamic scene analysis. In the following sections we consider the use of motion in segmentation both spatially and in the frequency domain.



Spatial Techniques

Basic approach

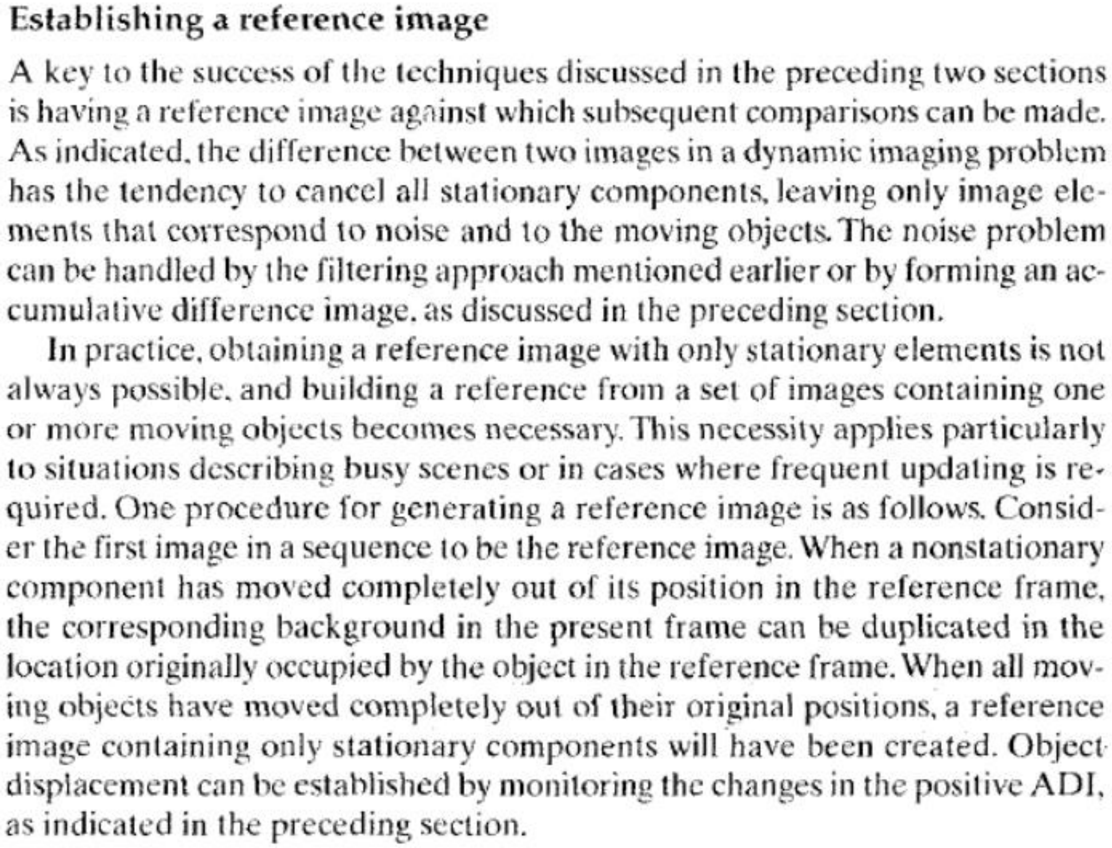
One of the simplest approaches for detecting changes between two image frames $f(x, y, t_i)$ and $f(x, y, t_j)$ taken at times t_i and t_j, respectively, is to compare the two images pixel by pixel. One procedure for doing this is to form a difference image. Suppose that we have a reference image containing only stationary components. Comparing this image against a subsequent image of the same scene, but including a moving object, results in the difference of the two images canceling the stationary elements, leaving only nonzero entries that correspond to the nonstationary image components.



A difference image between two images taken at times t_i and t_j may be defined as

$$d_{ij}(x, y) = \begin{cases} 1 & \text{if } \left|f(x, y, t_i) - f(x, y, t_j)\right| > T \\ 0 & \text{otherwise} \end{cases} \qquad (10.6\text{-}1)$$

where T is a specified threshold. Note that $d_{ij}(x, y)$ has a value of 1 at spatial coordinates (x, y) only if the gray-level difference between the two images is appreciably different at those coordinates, as determined by the specified threshold T. It is assumed that all images are of the same size. Finally, we note that the values of the coordinates (x, y) in Eq. (10.6-1) span the dimensions of these images, so that the difference image $d_{ij}(x, y)$ also is of same size as the images in the sequence.



Establishing a reference image

A key to the success of the techniques discussed in the preceding two sections is having a reference image against which subsequent comparisons can be made. As indicated, the difference between two images in a dynamic imaging problem has the tendency to cancel all stationary components, leaving only image elements that correspond to noise and to the moving objects. The noise problem can be handled by the filtering approach mentioned earlier or by forming an accumulative difference image, as discussed in the preceding section.

In practice, obtaining a reference image with only stationary elements is not always possible, and building a reference from a set of images containing one or more moving objects becomes necessary. This necessity applies particularly to situations describing busy scenes or in cases where frequent updating is required. One procedure for generating a reference image is as follows. Consider the first image in a sequence to be the reference image. When a nonstationary component has moved completely out of its position in the reference frame, the corresponding background in the present frame can be duplicated in the location originally occupied by the object in the reference frame. When all moving objects have moved completely out of their original positions, a reference image containing only stationary components will have been created. Object displacement can be established by monitoring the changes in the positive ADI, as indicated in the preceding section.

Frequency Domain Techniques

In this section we consider the problem of determining motion estimates via a Fourier transform formulation. Consider a sequence *[(x, y, t),* *t* = 0,1, ... , *K* - 1, of *K* digital image frames of size *M* X *N* generated by a stationary camera. We begin the development by assuming that all frames have a homogeneous background of zero intensity. The exception is a single, I-pixel object of unit intensity that is moving with constant velocity. Suppose that for frame one (*t* = 0) the image plane is *projected* onto the *x-axis;* that is, the pixel intensities are summed across the columns. This operation yields a 1-D array with *M* entries that are 0, except at the location where the object is projected.

PART-A (20 X 1 = 20 Marks)
(Answer All The Questions)

PART-B (5 * 2= 10 Marks)
(Answer all the Questions)

1. Define Erosion and Dilation
2. Define Opening and Closing
3. What is convex hull?
4. What is Hole filling and Connected components?
5. Define skeletons and pruning.
6. What is Image Segmentation?
7. Define Hough transform.
8. What is Thresholding?

PART-C (5 * 6=30 Marks)
(Answer all the Questions)

1. Discuss about the Morphological Image Processing methods in detail.
2. Explain about Opening and Closing with a neat diagram
3. Explain about Erosion, Dilation in detail with neat sketch.
4. Explain Hit-or-Miss Transform in detail.
5. Explain about thinning and thickening
6. Describe about Point, line and Edge detection in detail.
7. Discuss about Boundary detection techniques.
8. Illustrate Watershed algorithm with neat diagram.
9. Discuss about Use of motion in segmentation.
10. Describe Thresholding and Iterative thresholding.
11. Explain about Region-based segmentation.

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III BSC CS COURSE NAME: DIGITAL IMAGE PROCESSING
COURSE CODE: 16CSU502B BATCH-2016-2019

UNIT V

SNO	QUESTIONS	CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	ANSWER
1	Closing is represented by __________	A .B	A+B	A-B	AxB	A .B
2	Closing is used for __________	separation	compression	decompression	filling holes	filling holes
3	Erosion is used for object __________	removing lines	producing lines	blurring image	sharpening image	removing lines
4	Reflection is applied on image's __________	x coordinate	y coordinate	z coordinate	Both A and B	Both x and y coordinate
5	Opening with rolling SE __________	sharps	shrinks	smooths	deletes	smooths
6	For line detection we assume that lines are __________	thin	thick	sharp	blur	thin
7	The maximum length of the breaks can gets upto __________	4pixels	3pixels	2pixels	1pixel	2pixels
8	__________	rows	columns	edges	every element	every element
9	Structuring elements usually are __________	square array	circular array	triangular array	rectangular array	rectangular array
10	Duality principle is used when SE is __________	square	symmetric	asymmetric	translated	symmetric
11	Dilation followed by erosion is called __________	opening	closing	blurring	translation	closing
12	Reflection and translation of the image objects are based on __________	pixels	frames	structuring elements	coordinates	structuring elements
13	Opening smooths the image's __________	pixels	lines	contour	boundary	contour
14	Two main operations of morphology are __________	erosion	dilation	set theory	Both erosion and dilation	Both erosion and dilation
15	Structuring elements have origins at __________	top left	top right	center	bottom left	center
16	Structuring element is also called __________	pixels	lines	subimage	noise	subimage
17	With dilation process images get __________	thinner	shrinked	thickened	sharpened	thickened
18	Opening and closing are each others __________	neighbors	duals	centers	corners	duals
19	Fully containment of the SE in an image is required in __________	erosion	dilation	opening	closing	erosion
20	Erosion followed by dilation is called __________	opening	closing	blurring	translation	opening
21	To make the SE rectangular array approach that is used is called __________	padding	logic diagram	set theory	map	padding

22	Hit-or-miss transformation is used for shape ___________	removal	detection	compression	decompression	detection
23	Mathematical morphology is a ___________	set theory	logic diagram	graph	map	set theory
24	Duality principle is valid to involved ___________	one equation	both equations	any equation	Both A and B	both equations
25	Opening is represented by ___________	A o B	A+B	A-B	AxB	A o B
26	Subimages used to probe the image is called ___________	pixels	frames	structuring elements	coordinates	structuring elements
27	Closing produces ___________	narrow breaks	lines	dots	noise	narrow breaks
28	Replacing the object from its origin referred to as ___________	reflection	compression	decompression	translation	translation
29	Dilation is used for ___________	bridging gaps	compression	decompression	translation	bridging gaps
30	For line detection we use mask that is ___________	Gaussian	laplacian	ideal	butterworth	laplacian
31	If the inner region of the object is textured then approach we use is ___________	discontinuity	similarity	extraction	recognition	similarity
32	To avoid the negative values taking absolute values in lapacian image doubles ___________	thickness of lines	thinness of lines	thickness of edges	thinness of edges	thickness of lines
33	Gradient magnitude images are more useful in ___________	point detection	line detection	area detection	edge detection	edge detection
34	Image having gradient pixels is called ___________	sharp image	blur image	gradient image	binary image	gradient image
35	Diagonal lines are angles at ___________	0	30	45	90	45
36	Transition between objects and background shows ___________	ramp edges	step edges	sharp edges	Both ramp and step edges	Both ramp and step edges
37	Horizontal lines are angles at ___________	0	30	45	90	0
38	For edge detection we use ___________	first derivative	second derivative	third derivative	Both A and B	first derivative
39	Sobel gradient is not that good for detection of ___________	horizontal lines	vertical lines	Diagonal lines	edges	Diagonal lines
40	Method in which images are input and attributes are output is called ___________	low level processes	high level processes	mid level processes	edge level processes	mid level processes
41	Image morphology is an important tool in extraction of image ______________	features	colour	intensities	nature	features
42	The difference between the original image and the	higher level gray	low lever gray	boundary	unfilled regions	boundary
43	Top-hat transform is used for _____________	highlighting the right peaks	highlighting the dark peaks	highlighting the bright and dark peaks	highlighting the dark and bright peaks	highlighting the right peaks

44	The theory of mathematical morphology is based on_______________	image size	set theory	probability	correlation	set theory
45	well transform is used for _____________	highlighting the bright peaks	highlighting the dark peaks	highlighting the bright and dark peaks	highlighting the dark and bright peaks	highlighting the dark peaks
46	Thinning operation is used to remove the _________pixels	foreground	back ground	object	image	foreground
47	Morphlogical gradient gives___________	transition from spatial to frequency	transition from dark to bright	transition from frequency to spatial	none	transition from dark to bright
48	Structering element is a ______________	mask	colour	background	pixel	mask
49	_____________is a process of removing of the extra tail pixels in an image	erosion	dilation	hit-miss transform	pruning	pruning
50	Watershed is process of ______________ the object	histogram	locating	transform	highliting	locating
51	_________________is process of partition the digital image in to multiple regions	merging	filling	splitting	transform	splitting
52	_________is set of connected pixel that lie on the boundary between two regions.	point	edge	colour	line	edge
53	The objective of the sharpening filter is _________	highlight the intensity transitions	highlight the low transitions	highlight the bright transitions	highlight the colour transitions	highlight the intensity transitions
54	______________has number of peaks	bimodel histogram	multimodel histogram	histogram	image	multimodel histogram
55	_____________is the starting pixel of region growing process.	seed pixel	base pixel	original pixel	image	seed pixel
56	____________is a deformable model that fits a model for segmenting ROI	tiger	snake	goat	image	snake
57	_______________is the position of sign change of the first derivative among neighboring points	edge	zero-crosing	line	point	zero-crosing
58	__________ has unimodel histogram	one pixel	one peak	one valley	one intensity level	one peak
59	ROI stands for ________________	region of image	region of interest	region of indicator	restoration of image	region of interest
60	The hough transform is used to fit points as __________________	line	edge	curve	ROI	curve