DIGITAL IMAGE PROCESSING

LECTURE NOTES

B.E (IVYEAR)

Prepared by

Dr.S.Vijayaraghavan

Assistant Professor-ECE

SCSVMV Deemed University, Kanchipuram

DIGITAL IMAGE PROCESSING VII-Semester

Pre-requisite: Basic knowledge of Signals & Systems, Digital Signal Processing and Digital Design

L	T	P	C
4	1	0	4

OBJECTIVES:

- To learn digital image fundamentals.
- To be exposed to simple image processing techniques.
- To be familiar with image compression and segmentation techniques
- To represent image in form of features.

UNIT - I DIGITAL IMAGE FUNDAMENTALS

Introduction – Origin – Steps in Digital Image Processing – Components – Elements of Visual Perception – Image Sensing and Acquisition – Image Sampling and Quantization – Relationships between pixels - color models.

UNIT - II IMAGE ENHANCEMENT

Spatial Domain: Gray level transformations – Histogram processing – Basics of Spatial Filtering–Smoothing and Sharpening Spatial Filtering – Frequency Domain: Introduction to Fourier Transform– Smoothing and Sharpening frequency domain filters – Ideal, Butterworth and Gaussian filters.

UNIT - III IMAGE RESTORATION AND SEGMENTATION

Noise models – Mean Filters – Order Statistics – Adaptive filters – Band reject Filters – Band pass Filters – Notch Filters – Optimum Notch Filtering – Inverse Filtering – Wiener filtering Segmentation: Detection of Discontinuities– Edge Linking and Boundary detection – Region based segmentation- Morphological processing- erosion and dilation.

UNIT - IV WAVELETS AND IMAGE COMPRESSION

Wavelets – Sub band coding – Multi-resolution expansions - Compression: Fundamentals – Image Compression models – Error Free Compression – Variable Length Coding – Bit-Plane Coding – Lossless Predictive Coding – Lossy Compression – Lossy Predictive Coding – Compression Standards.

UNIT - V IMAGE REPRESENTATION AND RECOGNITION

Boundary representation – Chain Code – Polygonal approximation, signature, boundary segments – Boundary description – Shape number – Fourier Descriptor, moments-Regional Descriptors – Topological feature, Texture - Patterns and Pattern classes - Recognition based on matching.

OUTCOMES:

At the end of the course, the student should be able to:

- ✓ Understand the image enhancement techniques
- ✓ Understand the concept of restoration and segmentation
- ✓ Understand wavelets and image compression

TEXT BOOK:

1. Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Third Edition, Pearson Education, 2010.

REFERENCES:

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB", Third Edition Tata Mc Graw Hill Pvt. Ltd., 2011.
2. Anil Jain K. "Fundamentals of Digital Image Processing", PHI Learning Pvt. Ltd., 2011.
3. Willliam K Pratt, "Digital Image Processing", John Willey, 2002.
4. Malay K. Pakhira, "Digital Image Processing and Pattern Recognition", First Edition, PHI Learning Pvt. Ltd., 2011

UNIT-1
DIGITAL IMAGE FUNDAMENTALS

LEARNING OBJECTIVES:

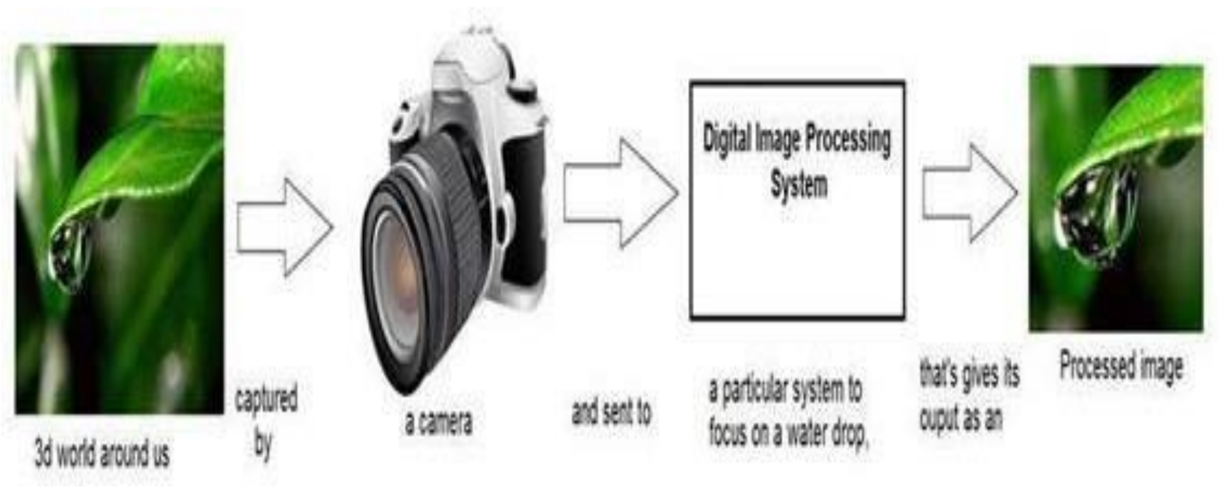
This unit provides an overview of the image –processing system which includes various elements like image sampling, quantization, Basic steps in image processing, image formation, storage and display. After completing this unit, the reader is expected to be familiar with the following concepts:

1. Image sampling
2. Image sensors
3. Different steps in image processing
4. Image formation

DIGITAL IMAGE FUNDAMENTALS:

The field of digital image processing refers to processing digital images by means of digital computer. 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 digitalimage.

An image is a two-dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene. An image is a projection of a 3- D scene into a 2D projection plane.


3d world around us
captured by
a camera
and sent to
Digital Image Processing System
a particular system to focus on a water drop,
that's gives its ouput as an
Processed image

An image may be defined as a two-dimensional function ***f(x,y),*** where x and y are spatial (plane) coordinates, and the amplitude to ***f*** at any pair of coordinates **(x,y)** is called the intensity of the image at thatpoint. The term **gray level**is used often to refer to the intensity of monochrome images.
Color images are formed by a combination of individual 2-D images. For example, the RGB color system, a color image consists of three (red, green and blue) individual component images. For this reason, many of the techniques developed for monochrome images can be extended to color images by processing the three component images individually.

An image may be continuous with respect to the x- and y- coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized.

APPLICATIONS OF DIGITAL IMAGE PROCESSING:

Since digital image processing has very wide applications and almost all of the technical fields are impacted by DIP, we will just discuss some of the major applications of DIP.
Digital image processing has a broad spectrum of applications, such as

1. Remote sensing via satellites and otherspacecrafts
2. Image transmission and storage for businessapplications
3. Medicalprocessing
4. RADAR (Radio Detection and Ranging)
5. SONAR (Sound Navigation and Ranging)
6. Acoustic Image Processing (The study of underwater sound is known as Underwater Acousticsor HydroAcoustics)
7. Robotics and automated inspection of industrial parts

Images acquired by satellites are useful in trackingof

1. Earthresources
2. Geographical mapping
3. Prediction of agriculturalcrops
4. Urban growth and weathermonitoring
5. Flood and fire control and many other environmentalapplications

Space image applicationsinclude:

1. Recognition and analysis of objects contained in images obtained from deep space-probemissions.
2. Image transmission and storage applications occur in broadcasttelevision
3. Teleconferencing
4. Transmission of facsimile images (Printed documents and graphics) for office automation
5. Communication over computer networks

6. Closed-circuit television-based security monitoring systemsand
7. In militarycommunications

Medicalapplications:

1. Processing of chest X-rays
2. Cineangiograms
3. Projection images of trans axial tomographyand
4. Medical images that occur in radiology nuclear magneticresonance (NMR)
5. Ultrasonicscanning

IMAGE PROCESSING TOOLBOX (IPT):

It is a collection of functions that extend the capability of the MATLAB numeric computing environment. These functions, and the expressiveness of the MATLAB language, make many image-processing operations easy to write in a compact, clear manner, thus providing an ideal software prototyping environment for the solution of image processingproblem.

COMPONENTS OF IMAGE PROCESSING SYSTEM:

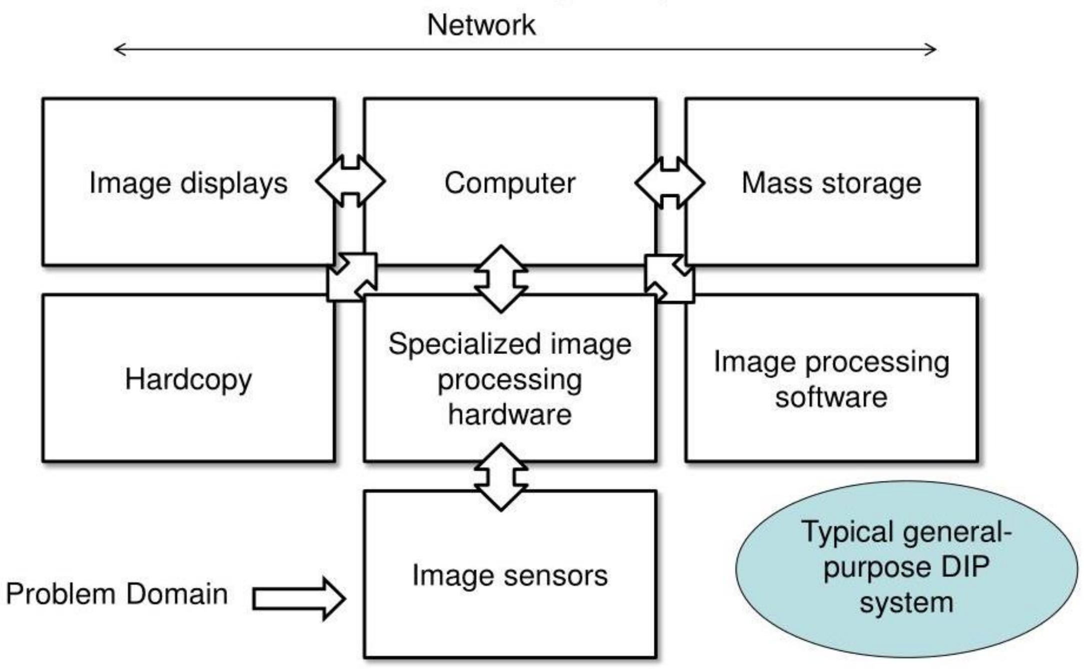

Network
Image displays
Computer
Mass storage
Hardcopy
Specialized image processing hardware
Image processing software
Problem Domain
Image sensors
Typical general-purpose DIP system

Fig: Components of Image processing System

Image Sensors: With reference to sensing, two elements are required to acquire digital image. The first is a physical device that is sensitive to the energy radiated by the object we wish to image and second is specialized image processing hardware.

Specialize Image Processing Hardware: It consists of the digitizer just mentioned, plus hardware that performs other primitive operations such as an arithmetic logic unit, which performs arithmetic such addition and subtraction and logical operations in parallel onimages.

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

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

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

- Short term storage for use duringprocessing
- On line storage for relatively fastretrieval
- Archival storage such as magnetic tapes anddisks

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

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

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

FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:

There are two categories of the steps involved in the image processing –

1. Methods whose outputs are input areimages.
2. Methods whose outputs are attributes extracted from thoseimages.

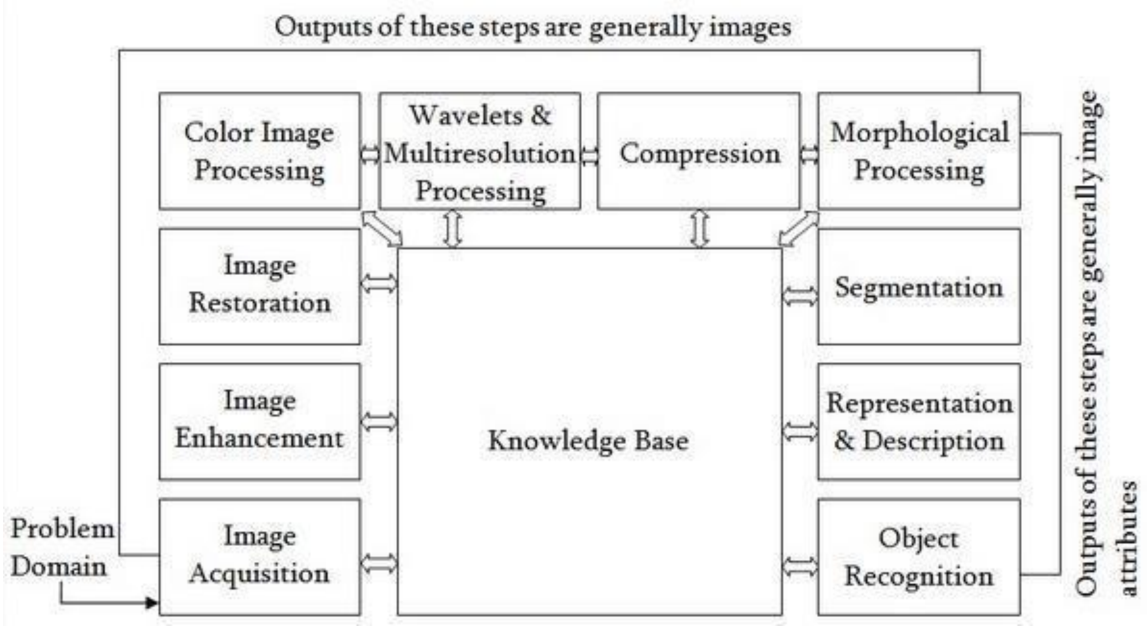


Fig: Fundamental Steps in Digital Image Processing

Image Acquisition: It could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves processing such scaling.

Image Enhancement: It is among the simplest and most appealing areas of digital image processing. The idea behind this is to bring out details that are obscured or simply to highlight certain features of interest in image. Image enhancement is a very subjective area of imageprocessing.

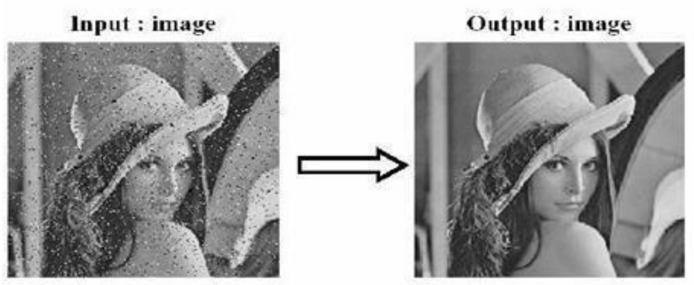
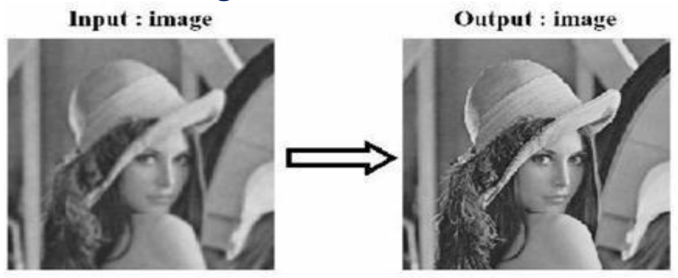


Image Restoration: It deals with improving the appearance of an image. It is an objective approach, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding what constitutes a “good” enhancement result.



Color Image Processing: It is an area that is been gaining importance because of the use of digital images over the internet. Color image processing deals with basically color models and their implementation in image processingapplications.

Wavelets and Multiresolution Processing: These are the foundation for representing image in various degrees of resolution.

Compression: It deals with techniques reducing the storage required to save an image, or the bandwidth required to transmit it over the network. It has to major approaches

1. Lossless Compression
2. Lossy Compression

Morphological Processing: It deals with tools for extracting image components that are useful in the representation and description of shape and boundary of objects. It is majorly used in automated inspection applications.

Representation and Description: It always follows the output of segmentation step that is, raw pixel data, constituting either the boundary of an image or points in the region itself. In either case converting the data to a form suitable for computer processing is necessary.

Recognition: It is the process that assigns label to an object based on its descriptors. It is the last step of image processing which use artificial intelligence of software.

Knowledge Base: Knowledge about a problem domain is coded into an image processing system in the form of a knowledge base. This knowledge may be as simple as detailing regions of an image where the information of the interest in known to be located. Thus, limiting search that has to be conducted in seeking the information. The knowledge base also can be quite complex such interrelated list of all major possible defects in a materials inspection problem or an image database containing high resolution satellite images of a region in connection with change detection application.

Simple Image Model: An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be nonzero and finite; that is o<f(x,y) <co The function f(x,y) may be characterized by two components- The amount of the source illumination incident on the scene beingviewed.
The amount of the source illumination reflected back by the objects in the scene These are called illumination and reflectance components and are denoted by i(x,y) an r (x,y) respectively.
The functions combine as a product to form f(x,y). We call the intensity of a monochrome image at any coordinates (x,y) the gray level (l) of the image at that point l= f (x, y.)
L min ≤ l ≤ L_{max} $L_{min}$is to be positive and L_{max} must be finite

$$L_{min} = i_{min} r_{min}$$
$$L_{max} = i_{max} r_{max}$$

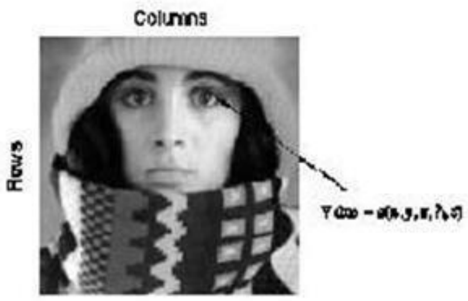
The interval [L_{min}, L_{max}] is called gray scale. Common practice is to shift this interval numerically to the interval [0, L-l] where l=0 is considered black and l= L-1 is considered white on the gray scale. All intermediate values are shades of gray of gray varying from black towhite.

SAMPLING AND QUANTIZATION:

To create a digital image, we need to convert the continuous sensed data into digital from. This involves two processes – sampling and quantization. An image may be continuous with respect to the x and y coordinates and also in amplitude. To convert it into digital form we have to sample the function in both coordinates and in amplitudes.
Digitalizing the coordinate values is called **Sampling.** Digitalizing the amplitude values is called **Quantization**. There is a continuous the image along the line segment AB. To simple this function, we take equally spaced samples along line AB. The location of each samples is given by a vertical tick back (mark) in the bottom part. The samples are shown as block squares superimposed on function the set of these discrete locations gives the sampled function.

In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So, we divide the gray level scale into eight discrete levels ranging from eight level values. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment it made depending on the vertical proximity of a simple to a vertical tick mark. Starting at the top of the image and covering out this procedure line by line produces a two-dimensional digitalimage.

Digital ImageDefinition: A digital image f(m,n) described in a 2D discrete space is derived from an analog image f(x,y) in a 2D continuous space through a sampling process that is frequently referred to as digitization. The mathematics of that sampling process will be described in subsequent Chapters. For now, we will look at some basic definitions associated with the digital image. The effect of digitization is shown in figure.
The 2D continuous image f(x,y) is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates (m,n) with m=0,1,2..N-1 and n=0,1,2…N-1 is f(m,n). In fact, in most cases, is actually a function of many variables including depth, color and time (t).



There are three types of computerized processes in the processing of image
Low level Process: These involve primitive operations such as image processing to reduce noise, contrast enhancement and image sharpening. These kinds of processes are characterized by fact the both inputs and output areimages.
Mid-levelImage Processing: It involves tasks like segmentation, description of those objects to reduce them to a form suitable for computer processing, and classification of individual objects. The inputs to the process are generally images but outputs are attributes extracted from images.
High level Processing: It involves “making sense” of an ensemble of recognized objects, as in image analysis, and performing the cognitive functions normally associated with vision.

Representing Digital Images: The result of sampling and quantization is matrix of real numbers. Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N Columns. The values of the coordinates (x,y) now become discrete quantities thus the value of the coordinates at origin become (x,y) = (0,0) The next Coordinates value along the first signify the image along the first row. it does not mean that these are the actual values of physical coordinates when the image was sampled. Thus, the right side of the matrix represents a digital element, pixel or pel. The matrix can be represented in the following form as well. The sampling process may be viewed as partitioning the XY plane into a grid with the coordinates of the center of each grid being a pair of elements from the Cartesian products Z_2 which is the set of all ordered pair of elements (Z_i, Z_j) with Z_i and Z_j being integers from Z.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,N-1) \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Hence f(x,y) is a digital image ifgray level (that is, a real number from the set of real number R) to each distinct pair of coordinates (x,y). This functional assignment is the quantization process. If the gray levels are also integers, Z replaces R, the and a digital image become a 2D function whose coordinates and she amplitude value are integers. Due to processing storage and hardware consideration, the number gray levels typically are an integer power of 2.

$$L=2^k$$

Then, the number 'b' of bites required to store a digital image is

$$b=M*N*k$$

When M=N, the equation become

$$b=N^2*k$$

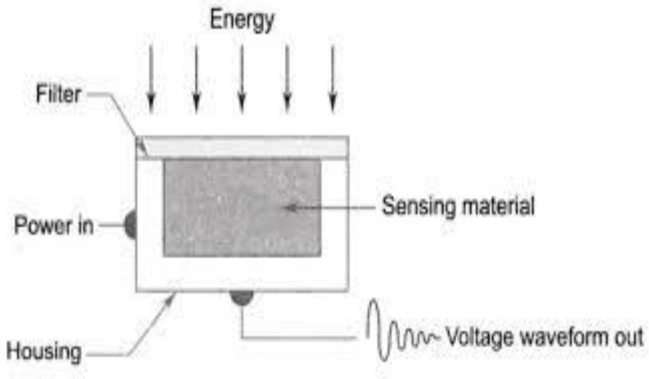
When an image can have 2^k gray levels, it is referred to as "k- bit". An image with 256 possible gray levels is called an "8- bit image" ($256=2^8$).

Spatial and Gray level Resolution:

Spatial resolution is the smallest discernible details are an image. Suppose a chart can be constructed with vertical lines of width w with the space between the also having width W, so a line pair consists of one such line and its adjacent space thus. The width of the line pair is 2w and there is 1/2w line pair per unit distance resolution is simply the smallest number of discernible line pair unitdistance.

Gray levels resolution refers to smallest discernible change in gray levels. Measuring discernible change in gray levels is a highly subjective process reducing the number of bits R while repairing the spatial resolution constant creates the problem of false contouring.It is caused by the use of an insufficient number of gray levels on the smooth areas of the digital image. It is called so because the rides resemble top graphics contours in a map. It is generally quite visible in image displayed using 16 or less uniformly spaced gray levels.

Image Sensing and Acquisition: The types of images in which we are interested are generated by the combination of an "**Illumination**" source and the reflection or absorption of energy from that source by the elements of the "scene" being imaged. We enclose *illumination* and *scene* in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene.



For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern. Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rockformations,orahumanbrain.Wecouldevenimageasource,suchasacquirin gimages of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects.

An example in the first category is light reflected from a planar surface. An example in the second category is when X-rays pass through a patient's body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light.

Electron microscopy and some applications of gamma imaging use this approach. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response. In this section, we look at the principal modalities for image sensing andgeneration.

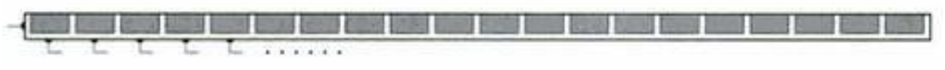


Fig: Line Sensor

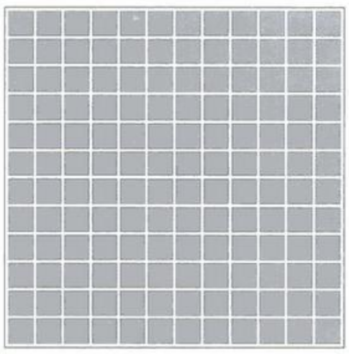


Fig: Single Image Sensor

The components of a single sensor, perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform isproportionaltolight.Theuseofafilterinfrontofasensorimprovesselectivity.F orexample, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.

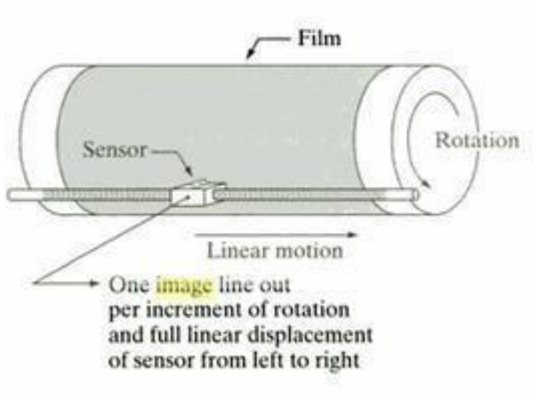

Film
Sensor
Rotation
Linear motion
One image line out per increment of rotation and full linear displacement of sensor from left to right

Fig: Array sensor Image Acquisition using a Single Sensor

In order to generate a 2-D image using a single sensor, there has to be relative displacements in both the x- and y-directions between the sensor and the area to be imaged. Figure shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Since mechanical motion can be controlled with high precision, this method is an inexpensive (but slow) way to obtain high-resolution images. Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as microdensitometers.

Image Acquisition using a Sensor Strips:

A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction. This is the type of arrangement used in most flatbed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged. One dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project area to be scanned onto the sensors. Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional ("slice") images of 3-Dobjects.

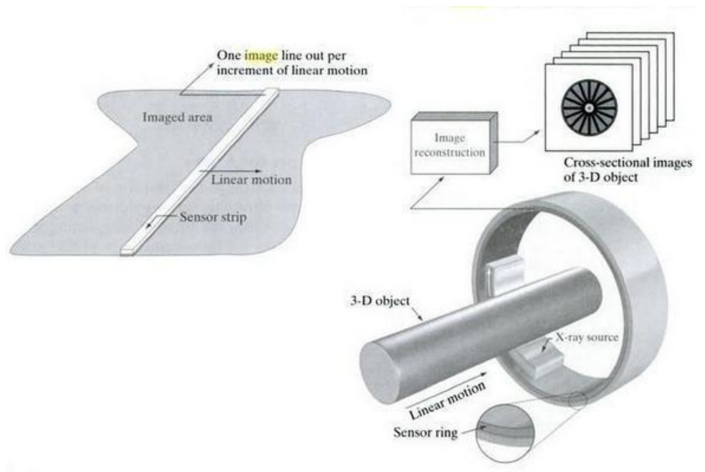
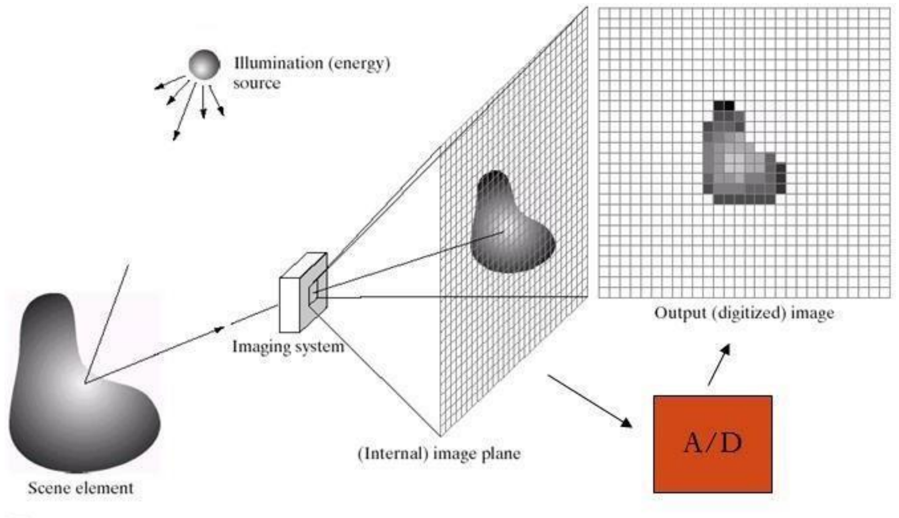


Fig: Image Acquisition using linear strip and circular strips

Image Acquisition using a Sensor Arrays:



a b c d e

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

The individual sensors arranged in the form of a 2-D array. Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras. A typical sensor for these cameras is a CCD array, which can be manufactured with a broad range of sensing properties and can be packaged in rugged arrays of elements or more. CCD sensors are used widely in digital cameras and other light sensing instruments. The response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor, a property that is used in astronomical and other applications requiring low noise images. Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or even hours. The two dimensional, its key advantage is that a complete image can be obtained by focusing the energy pattern onto the surface of the array. Motion obviously is not necessary, as is the case with the sensor arrangements This figure shows the energy from an illumination source being reflected from a scene element, but, as mentioned at the beginning of this section, the energy also could be transmitted through the scene elements. The first function performed by the imaging system is to collect the incoming energy and focus it onto an image plane. If the illumination is light, the front end of the imaging system is a lens, which projects the viewed scene onto the lens focal plane. The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital andanalog circuitry sweeps these outputs and convert them to a video signal, which is then digitized by another section of the imaging system.

Image sampling and Quantization:

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes.

1. Sampling and
2. Quantization

A continuous image, f(x, y), that we want to convert to digital form. An image may be continuous with respect to the x- and y- coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude.

Digitizing the coordinate values is called **Sampling**. Digitizing the amplitude values is called **Quantization**.

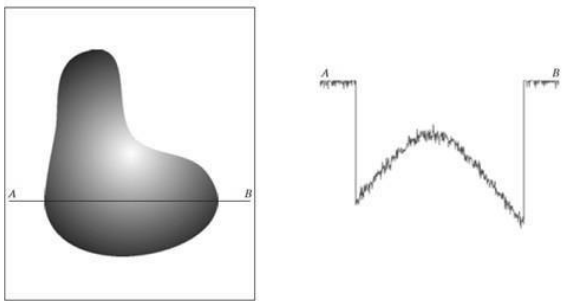


Fig: Sampling

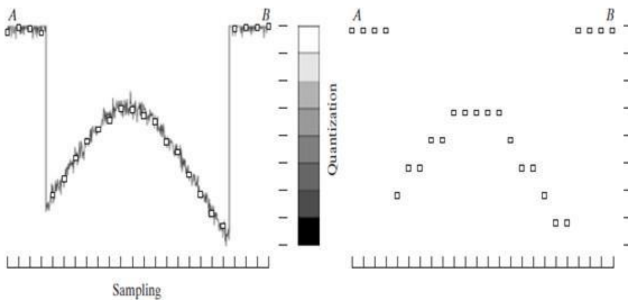


Fig: Quantization

Digital Image Representation:

Digital image is a finite collection of discrete samples (pixels) of any observable object. The pixels represent a two- or higher dimensional "view" of the object, each pixel having its own discrete value in a finite range. The pixel values may represent the amount of visible light, infra-red light, absorption of x-rays, electrons, or any other measurable value such as ultrasound wave impulses. The image does not need to have any visual sense; it is sufficient that the samples form a two-dimensional spatial structure that may be illustrated as an image. The images may be obtained by a digital camera, scanner, electron microscope, ultrasound stethoscope, or any other optical or non-optical sensor.

Examples of digital image areDigitalphotographs, Satelliteimages, Radiological images (x-rays, mammograms), Binary images, Fax images, Engineeringdrawings, Computer graphics, CAD drawings, and vector graphics in general are not considered in this course even though their reproduction is a possible source of an image. In fact, one goal of intermediate level image processing may be to reconstruct a model (Eg: Vector Representation) for a given digital image.

Relationship between Pixels:

We consider several important relationships between pixels in a digital image.

Neighbors of a Pixel:

A pixel *p* at coordinates (*x,y*) has four horizontal and verticalneighbors whose coordinates are givenby:

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

	(x, y-1)	
(x-1, y)	*P* (x,y)	(x+1, y)
	(x, y+1)	

This set of pixels, called the 4-neighbors or p, is denoted by $N_4(p)$. Each pixel is one unit distance from (x,y) and some of the neighbors of p lie outside the digital image if (x,y) is on the border of the image. The four diagonal neighbors of p have coordinates and are denoted by $N_D(p)$.

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

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

These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by $N_8(p)$.

(x-1, y+1)	(x, y-1)	(x+1, y-1)
(x-1, y)	*P (x,y)*	(x+1, y)
(x-1, y-1)	(x, y+1)	(x+1, y+1)

As before, some of the points in $N_D(p)$ and $N_8(p)$ fall outside the image if (x,y) is on the border of the image.

Adjacency and Connectivity:

Let v be the set of gray –level values used to define adjacency, in a binary image, V={1}. In a gray-scale image, the idea is the same, but Vtypically contains more elements, for example, V= {180, 181, 182, …,200}.

If the possible intensity values 0 – 255, Vset can be any subset of these 256 values. if we are reference to adjacency of pixel with value.

Three types of Adjacency:

4- Adjacency – two pixel P and Q with value from V are 4 –adjacency if A is in the setN4(P)

8- Adjacency – two pixel P and Q with value from V are 8 –adjacency if A is in the set N8(P)

M-adjacency –two pixel P and Q with value from V are m – adjacency if (i) Q is in N4(p) or
(ii) Q is in ND(q) and the
(iii) Set $N_4(p) \cap N_4(q)$ has no pixel whose values are fromV.
Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency isused.
Forexample:

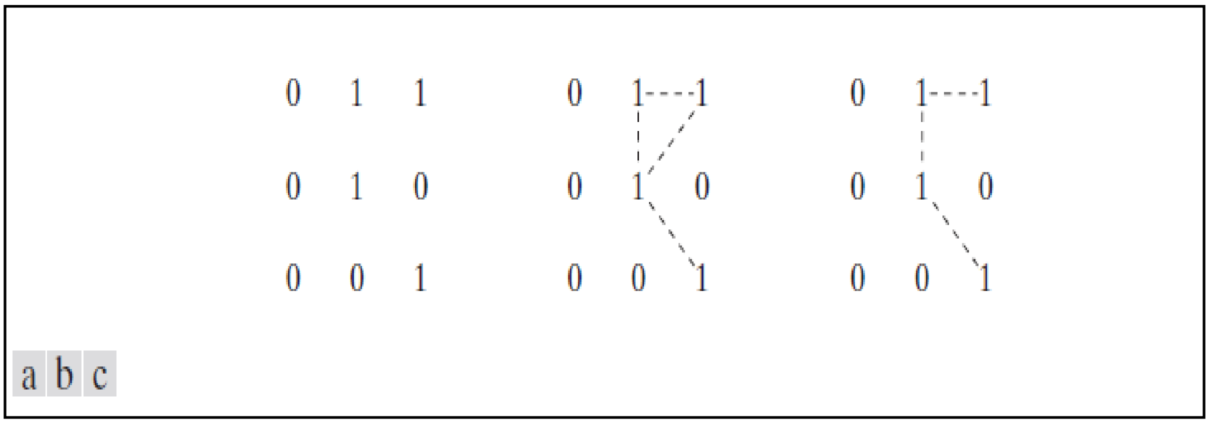


Fig: (a) Arrangement of pixels
(b) pixels that are 8-adjacent (shown dashed) to the center pixel
(c) **m**-adjacency

Types of Adjacency:
In this example, we can note that to connect between two pixels (finding a path between twopixels):
In 8-adjacency way, you can find multiple paths between twopixels
While, in m-adjacency, you can find only one path between twopixels
So, m-adjacency has eliminated the multiple path connection that has been generated by the8-adjacency.
Two subsets *S1* and *S2* are adjacent, if some pixel in *S1* is adjacent to some pixel in *S2*. Adjacent means, either 4-, 8- orm-adjacency.

Digital Path:
A digital path (or curve) from pixel p with coordinate (x,y) to pixel q with coordinate (s,t) is a sequence of distinct pixels with coordinates (x0,y0), (x_1,y_1), ..., (x_n, y_n) where $(x_0,y_0) = (x,y)$ and $(x_n, y_n) = (s,t)$ and pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$, n is the length of thepath.
If $(x_0,y_0) = (x_n, y_n)$, the path isclosed.
We can specify 4, 8or m-paths depending on the type of adjacency specified.

Return to the previousexample:

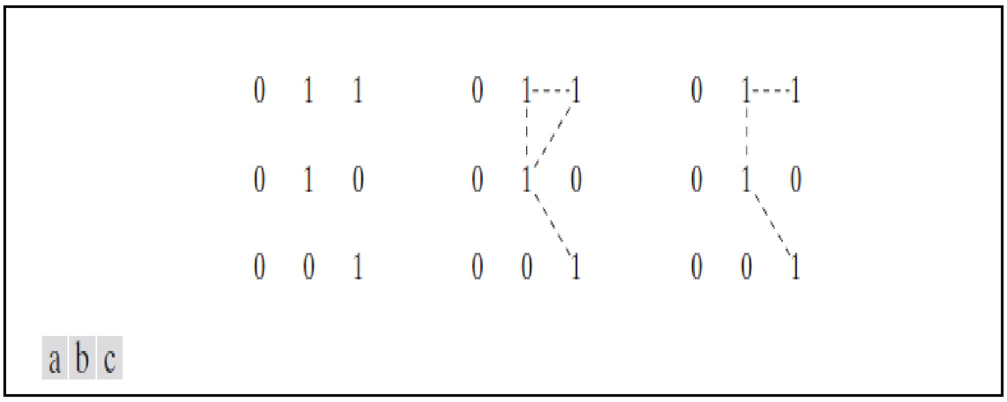


Fig:(a) Arrangement of pixels
(b) pixels that are 8-adjacent (shown dashed) to the center pixel
(c) m-adjacency

In figure (b) the paths between the top right and bottom right pixels are 8-paths. And the path between the same 2 pixels in figure (c) is m-path

Connectivity:

Let S represent a subset of pixels in an image, two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels inS.

For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S. If it only has one connected component, then set S is called a connectedset.

Region and Boundary:

Region: Let R be a subset of pixels in an image, we call R a region of the image ifRis a connected set.

Boundary: The boundary (also called border or contour) of a region R is the set of pixels in the region that have one or more neighbors that are not inR.

If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns in the image. This extra definition is required because an image has no neighbors beyond its borders. Normally, when we refer to a region, we are referring to subset of an image, and any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the regionboundary.

Distance Measures:

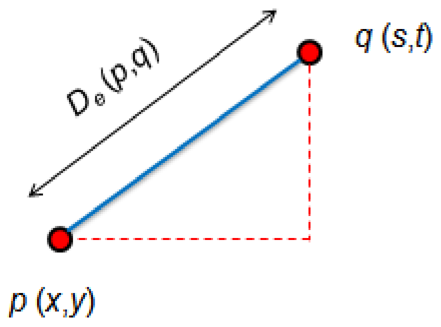
For pixel p,q and z with coordinate (x.y) ,(s,t) and (v,w) respectively D is a distance function or metric if

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

The Euclidean Distance between p and q is definedas:

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

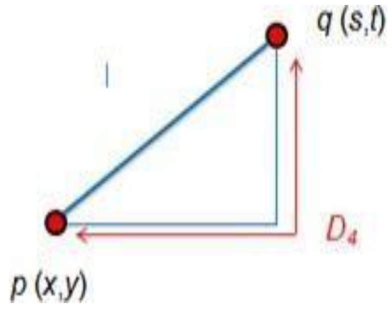
Pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius „ r „centered at (x,y)



The D_4distance (also called **city-block distance**) between p and q is definedas:

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

Pixels having a *D4* distance from (x,y), less than or equal to some value r form a Diamond centered at (x,y)



Example:

The pixels with distance $D_4 \leq$ 2 from (x,y) form the following contours of constant distance.

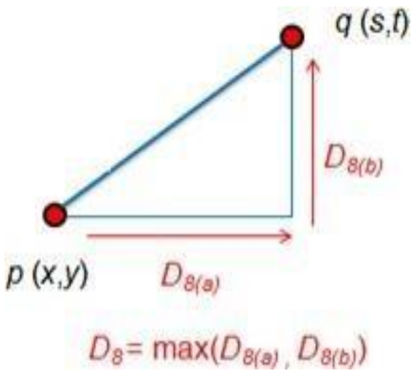
The pixels with D_4= 1 are the 4-neighbors of (x,y)

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

The D_8distance (also called **chessboard distance**) between p and q is definedas:

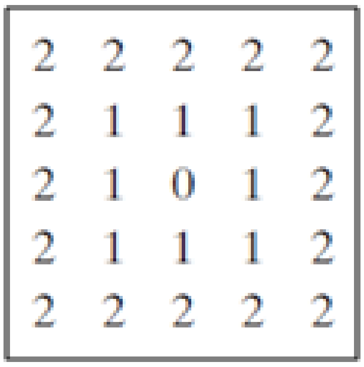
$$D_8(p,q) = \max(|\ x - s\ |, |\ y - t\ |)$$

Pixels having a D8 distance from (x,y), less than or equal to some value r form a square Centered at (x,y).



Example:

D_8distance ≤ 2 from (x,y) form the following contours of constant distance.



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

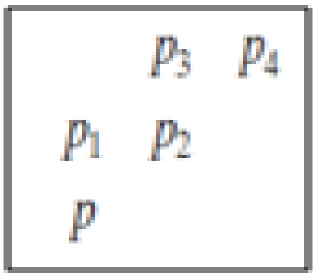
D_mDistance:

It is defined as the shortest m-path between the points.In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

Example:

Consider the following arrangement of pixels and assume that p, p2, and p4 have value 1 and that p1 and p3 can have can have a value of 0 or 1 Suppose that we consider the adjacency of pixels values 1 (i.e. V ={1})

Now, to compute the $D_m$between points p and p4



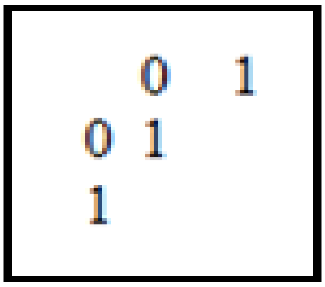
	p_3	p_4
p_1	p_2	
p		

Here we have 4 cases:

Case1: If p1 =0 and p3 = 0

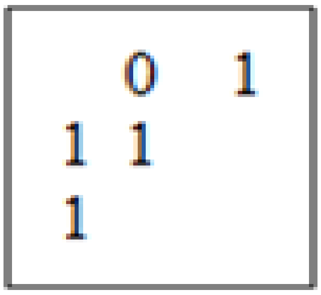
The length of the shortest m-path (the Dm distance) is 2 (p, p2, p4)

Case2: If p1 =1 and p3 = 0

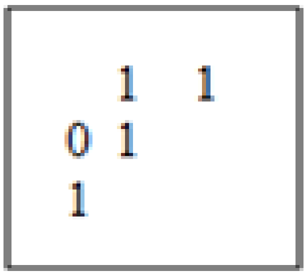


now, p1 and p will no longer be adjacent (see m-adjacency definition) then, the length of the shortest path will be 3 (p, p1, p2, p4)

Case3: If p1 =0 and p3 = 1

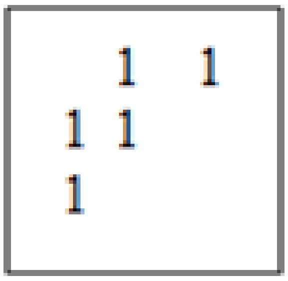


The same applies here, and the shortest –m-path will be 3 (p, p2, p3, p4)



Case4: If p1 =1 and p3 = 1

The length of the shortest m-path will be 4 (p, p1, p2, p3, *p4*)



UNIT -2
IMAGE ENHANCEMENT

Learning Objectives:

Image enhancement techniques are designed to improve the quality of an image as perceived by a human being. Image enhancement can be performed both in the spatial as well as in the frequency domain. After reading this chapter, the reader should have a basic knowledge about the following concepts:

1. Image enhancement in spatial and frequency domain
2. Point operations and mask operations in spatial domain
3. Different types of gray –level transformation
4. Histogram and histogram equalization
5. Frequency domain filtering

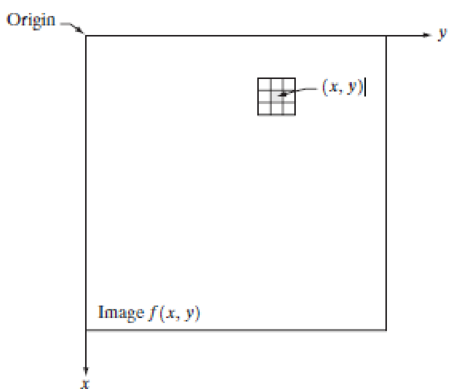
Introduction:

Image enhancement approaches fall into two broad categories: spatial domain methods and frequency domain methods. The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.

Frequency domain processing techniques are based on modifying the Fourier transform of an image. Enhancing an image provides better contrast and a more detailed image as compare to non-enhanced image. Image enhancement has very good applications. It is used to enhance medical images, images captured in remote sensing, images from satellite etc. As indicated previously, the term spatial domain refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression.

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

where f(x, y) is the input image, g(x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y).



The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular sub image area centered at (x, y), as Fig. 2.1 shows. The center of the sub image is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.

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

$$s = T(r)$$

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".

For example, if T(r) has the form shown in Fig. 2.2(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 m are compressed by the transformation function into a narrow range of s, toward black. The opposite effect takes place for values of r above m.

In the limiting case shown in Fig. 2.2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.
One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig. 2.1, in which the values of the mask coefficients determine the nature of the process, such as image sharpening. Enhancement techniques based on this type of approach often are referred to as mask processing or filtering.

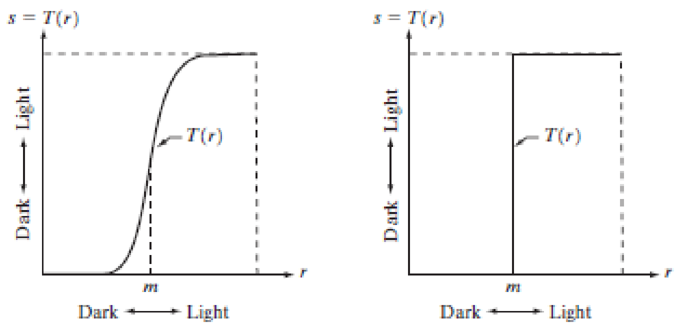


Fig. 2.2 Gray level transformation functions for contrast enhancement

Image enhancement can be done through gray level transformations which are discussed below.

BASIC GRAY LEVEL TRANSFORMATIONS:

1. Imagenegative
2. Logtransformations
3. Power lawtransformations
4. Piecewise-Linear transformationfunctions

LINEAR TRANSFORMATION:

First, we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transformation has been discussed in ourtutorial of image transformation, but a brief description of this transformation has been given here.

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. It has been shown below:

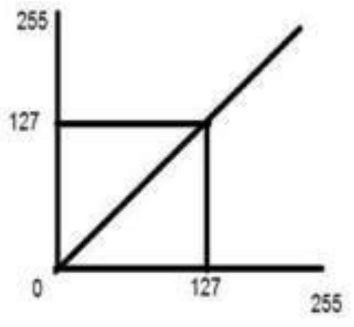


Fig. Linear transformation between input and output

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

IMAGE NEGATIVE:

The image negative with gray level value in the range of [0, L-1] is obtained by negative transformation given by S = T(r) or

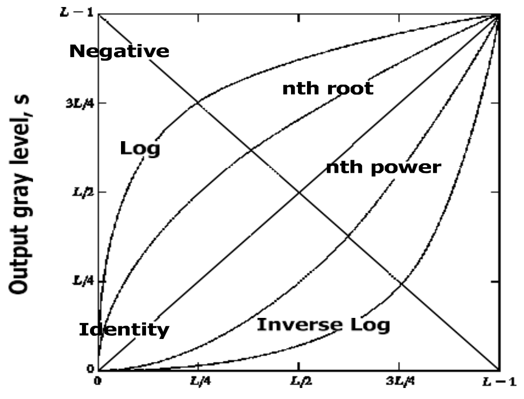
$$S = L - 1 - r$$

Where r= gray level value at pixel (x,y)

L is the largest gray level consists in the image

It results in getting photograph negative. It is useful when for enhancing white details embedded in dark regions of the image.

The overall graph of these transitions has been shown below.



Input gray level, r

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

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 are256. 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.

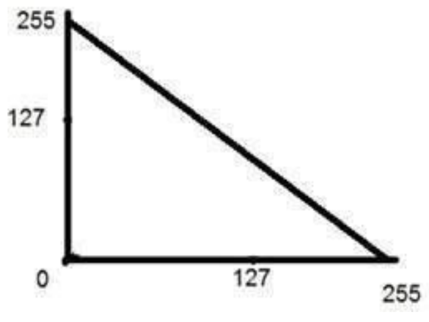


Fig. Negative transformations

LOGARITHMIC TRANSFORMATIONS:

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

LOG TRANSFORMATIONS:

The log transformations can be defined by this formula

$$S = c \log(r + 1).$$

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.

ANOTHER WAY TO REPRESENT LOG TRANSFORMATIONS: Enhance details in the darker regions of an image at the expense of detail in brighter regions.

$$T(f) = C * \log (1+r)$$

HereC is constantand r≥ 0

The shape of the curve shows that this transformation maps the narrow range of low gray level valuesintheinputimageintoa widerrangeofoutputimage.
Theoppositeis trueforhighlevelvaluesofinputimage.

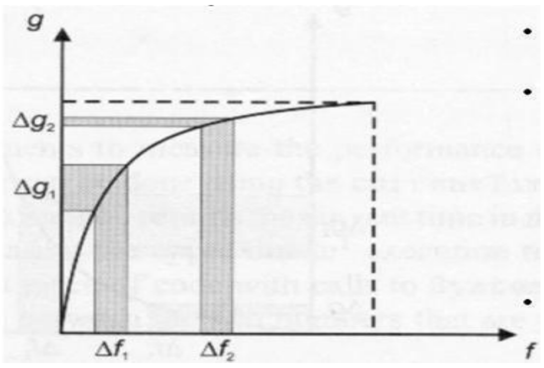


Fig. Log Transformation Curve input vs output

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, where c and g are positive constants. Sometimes Eq. (6) is written as

$$S = C\ (r+\varepsilon)\ \ \gamma$$

to account for an offset (that is, a measurable output when the input is zero). Plots of s versus r for various values of γ are shown in Fig. 2.10. As in the case of the log transformation, 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. Unlike the log function, however, we notice here a family of possible transformation curves obtained simply by varying γ.

In Fig that curves generated with values of γ>1 have exactly the opposite effect as those generated with values of γ<1. Finally, we Note that Eq. (6) reduces to the identity transformation when c=γ=1.

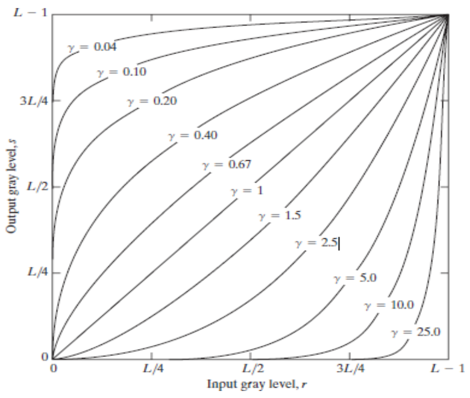


Fig. 2.13 Plot of the equation S = Crγ for various values of γ (C =1 in all cases)

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.
Varying gamma (γ) obtains family of possible transformation curves

$$S = C^* r \gamma$$

Here C and γ are positive constants. Plot of S versus r for various values of γ is γ > 1 compresses dark values and expands bright valuesγ < 1 (similar to Log transformation) but expands dark values Compresses bright values
When C = γ = 1, it reduces to identity transformation.

CORRECTING GAMMA:

$$S=Cr \gamma$$

$$S=Cr\ (1/2.5)$$

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

PIECEWISE-LINEAR TRANSFORMATION FUNCTIONS:

A complementary approach to the methods discussed in the previous three sections is to use piecewise linear functions. The principal advantage of piecewise linear functions over the types of functions we have discussed thus far is that the form of piecewise functions can be arbitrarily complex.The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

Contrast Stretching: One of the simplest piecewise linear functions is a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamicrange in the imaging sensor, or even wrong setting of a lens aperture during image acquisition.

$$S= T(r)$$

Figure x(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function. If r1=s1 and r2=s2, the transformation is a linear function that produces No changes in gray levels. If r1=r2, s1=0and s2= L-1, the transformation Becomes a thresholding function that creates a binary image, as illustrated in fig. 2.2(b).

Intermediate values of ar1, s1b and ar2, s2b produce various degrees of spread in the gray levels of the output image, thus affecting its contrast. In general, r1≤ r2 and s1 ≤ s2 is assumed so that the function is single valued and Monotonically increasing.

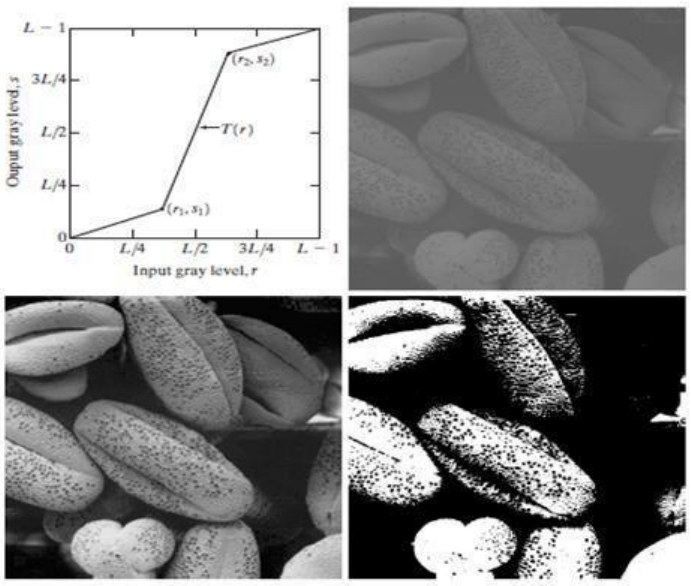


Fig. x Contrast Stretching.
(a) Form of transformation function
(b) A low-contrast stretching.
(c) Result of contrast stretching
(d) Result of thresholding

Figure x(b) shows an 8-bit image with low contrast.
Fig. x(c) shows the result of contrast stretching, obtained by setting $(r_1, s_1) = (r_{min}, 0)$ and $(r2, s2) = (r_{max}, L-1)$ where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range[0, L-1].

Finally, Fig. x(d) shows the result of using the thresholding function defined previously,with r1=r2=m, the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

Gray-level Slicing: Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.There are several ways of doing level slicing, but most of them are variations of two basic themes. One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.

This transformation, shown in Fig. y(a), produces a binary image. The second approach, based on the transformation shown in Fig y(b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure y (c) shows a gray-scale image, and Fig. y(d) shows the result of using the transformation in Fig. y(a). Variations of the two transformations shown in Fig. are easy to formulate.

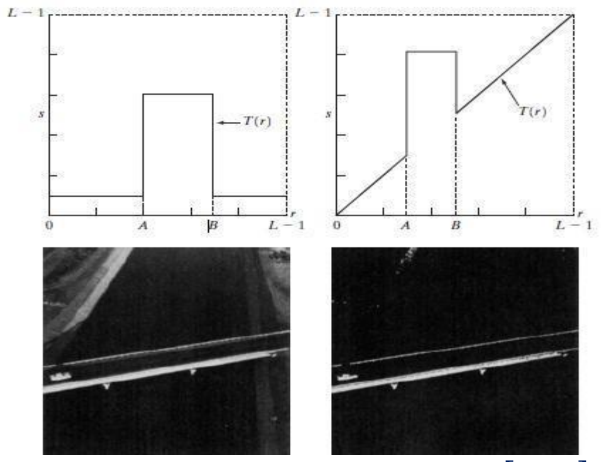


Fig. y (a)This transformation highlights range [A,B] of gray levels and reduces all others to a constant level
(b) This transformation highlights range [A,B] but preserves all other levels.
An image. (d) Result of using the transformation in (a).

BIT-PLANE SLICING:

Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for the most significant bit. In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.

Figure 3.12 illustrates these ideas, and Fig. 3.14 shows the various bit planes for the image shown in Fig. 3.13. Note that the higher-order bits

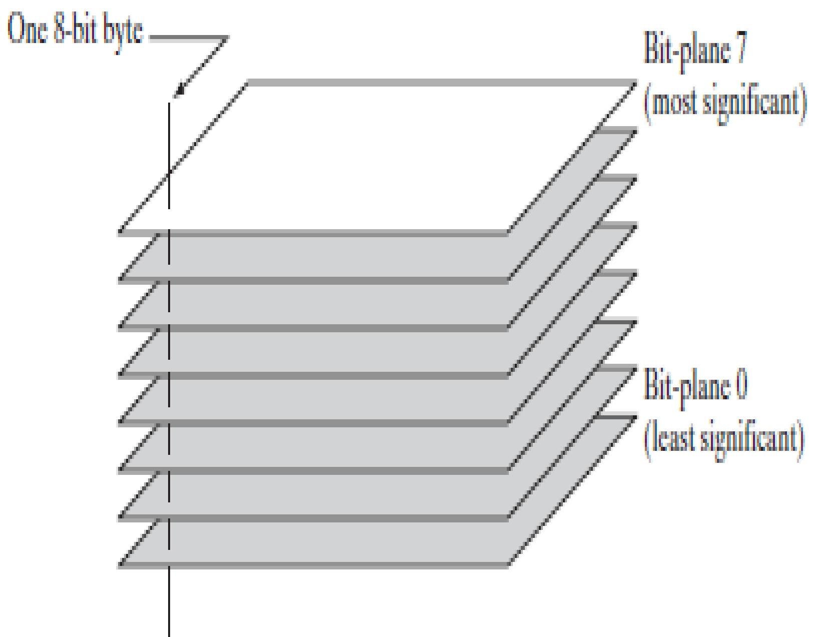


FIGURE Bit-plane representation of an 8-bit image.

(especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

In terms of bit-plane extraction for an 8-bit image, it is not difficult to show that the (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation function that (1) maps all levels in the image between 0 and 127 to one level (for example, 0); and (2) maps all levels between 129 and 255 to another (for example, 255).The binary image for bit-plane 7 in Fig. 3.14 was obtained in just this manner. It is left as an exercise(Problem 3.3) to obtain the gray-level transformation functions that would yield the other bit planes.

HISTOGRAM PROCESSING:

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function of the form

$$H(r_k)=n_k$$

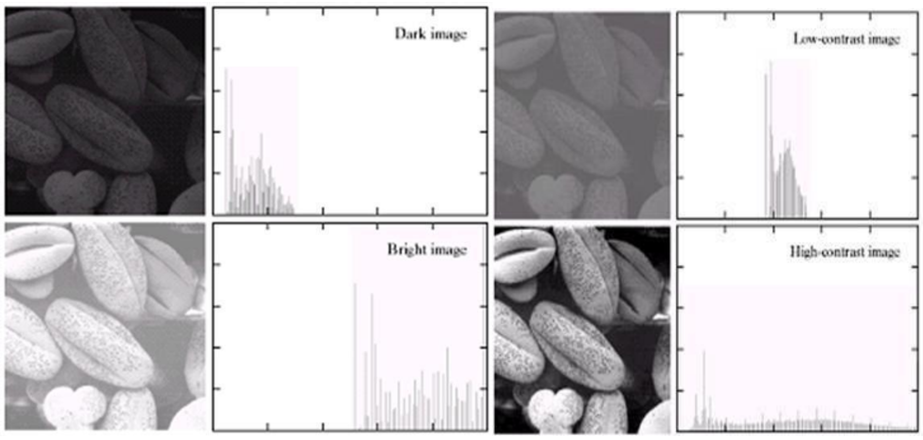
where r_k is the kth gray level and n_k is the number of pixels in the image having the level r_k. A normalized histogram is given by the equation

$$P(r_k)=n_k/n \text{ for } k=0,1,2,\ldots..,L-1$$

$P(r_k)$ gives the estimate of the probability of occurrence of gray level r_k. The sum of all components of a normalized histogram is equal to 1.
The histogram plots are simple plots of $H(r_k)=n_k$ versus r_k.

In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale. In case of bright image, the histogram components are biased towards the high side of the gray scale. The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.

The components of the histogram in the high contrast image cover a broad range of the gray scale. The net effect of this will be an image that shows a great deal of gray levels details and has high dynamic range.



HISTOGRAM EQUALIZATION:

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would beskewed towards the lower end of the grey scale and all the image detail are compressed into the dark end of the histogram. If we could stretch out the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.

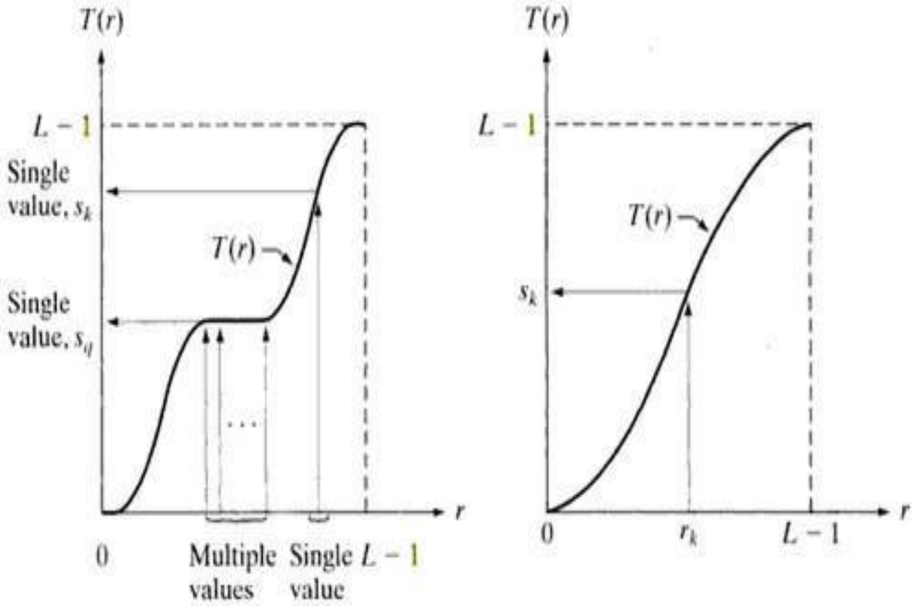
Let there be a continuous function with r being gray levels of the image to be enhanced. The range of r is [0, 1] with r=0 repressing black and r=1 representing white. The transformation function is of the form

S=T(r) where 0<r<1

It produces a level s for every pixel value r in the original image.

a b

FIGURE
(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.



The transformation function is assumed to fulfill two condition T(r) is single valued and monotonically increasing in the internal 0<T(r)<1 for 0<r<1.The transformation function should be single valued so that the inverse transformations should exist. Monotonically increasing condition preserves the increasing order from black to white in the output image. The second conditions guarantee that the output gray levels will be in the same range as the input levels. The gray levels of the image may be viewed as random variables in the interval [0.1]. The most fundamental descriptor of a random variable is its probability density function (PDF) $P_r(r)$ and $P_s(s)$ denote the probability density functions of random variables r and s respectively. Basic results from an elementary probability theory states that if $P_r(r)$ and T_r are known and T-1(s) satisfies conditions (a), then the probability density function $P_s(s)$ of the transformed variable is given by the formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Thus, the PDF of the transformed variable s is the determined by the gray levels PDF of the input image and by the chosen transformations function. A transformation function of a particular importance in image processing

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

This is the cumulative distribution function of r.
L is the total number of possible gray levels in the image.

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN:

Blurring/Noise Reduction: Noise characterized by sharp transitions in image intensity. Such transitions contribute significantly to high frequency components of Fourier transform. Intuitively, attenuating certain high frequency components result in blurring and reduction of image noise.

IDEAL LOW-PASS FILTER:

Cuts off all high-frequency components at a distance greater than a certain distance from origin (cutoff frequency).

$$H(u,v) = 1, \text{ if } D(u,v) \leq D0\ 0, \text{ if } D(u,v) > D0$$

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

$$D(u,v) = [(u-P/2)2 + (v-Q/2)2]\ 1/2$$

Whereas P and Q are the padded sizes from the basic equations
Wraparound error in their circular convolution can be avoided by padding these functions with zeros

Visualization: Ideal Low Pass Filter:
Aa shown in figure below

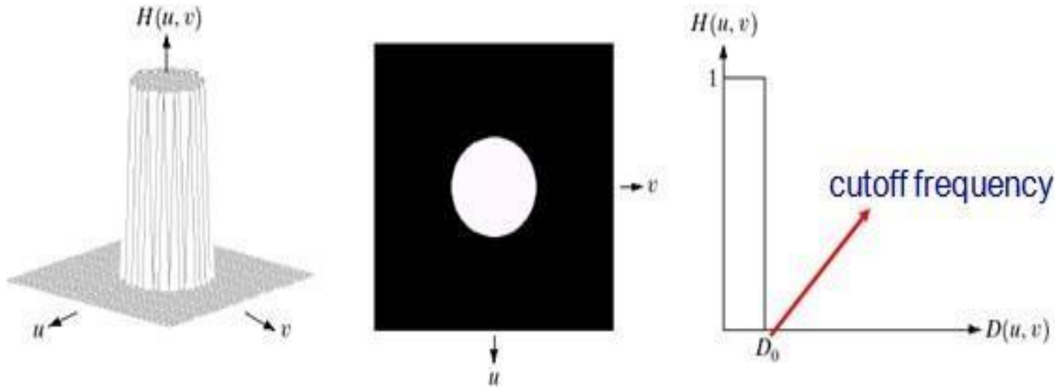


Fig: Ideal Low Pass Filter 3-D view and 2-D view and line graph

EFFECT OF DIFFERENT CUTOFF FREQUENCIES:
Fig.below(a) Test pattern of size 688x688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160 and 460 with respect to the full- size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8 and 99.2% of the padded image power respectively.

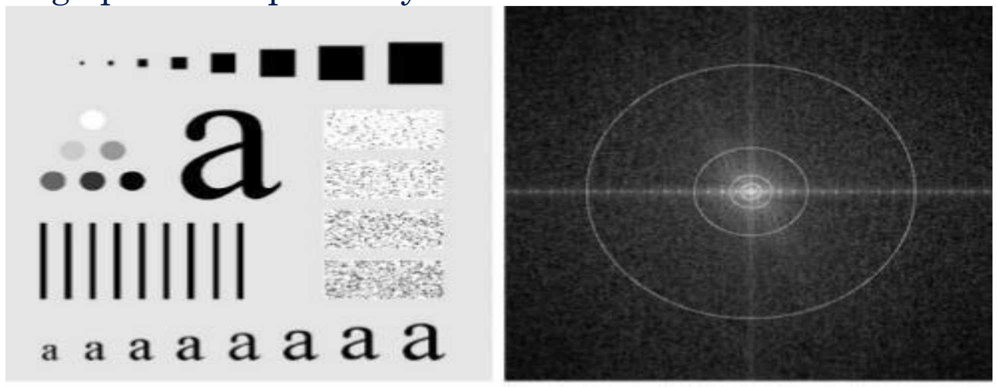


Fig: (a) Test pattern of size 688x688 pixels (b) its Fourier spectrum

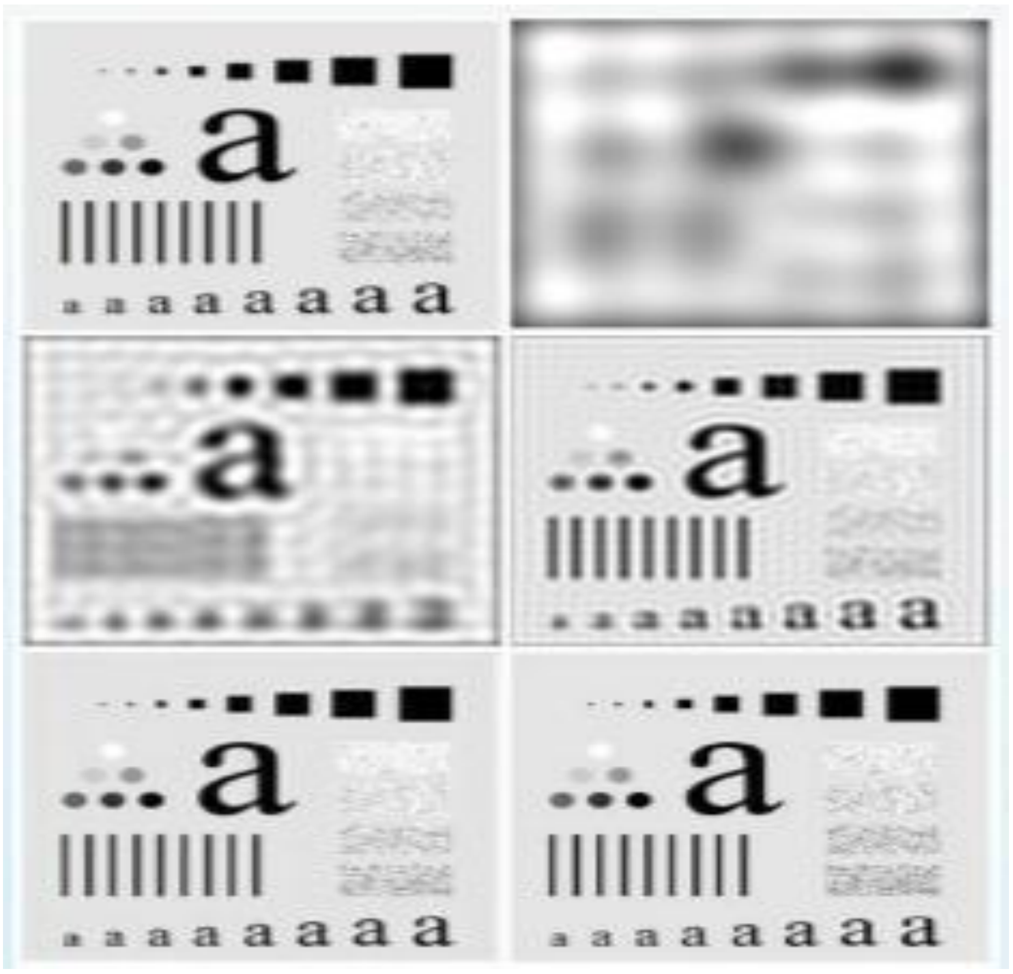


Fig: (a) original image, (b)-(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160 and 460, as shown in fig.2.2.2(b). The power removed by these filters was 13, 6.9, 4.3, 2.2 and 0.8% of the total, respectively.

As the cutoff frequency decreases,image becomes more blurred, Noiseincreases. Analogous to larger spatial filter sizes the severe blurring in this image is a clear indication that most of the sharp detail information in the picture is contained in the 13% power removed by the filter. As the filter radius is increases less and less power is removed, resulting in less blurring. Fig. (c) throughare characterized by **"Ringing"**, which becomes finer in texture as the amount of high frequency content removed decreases.

WHY IS THERE RINGING?

Ideal low-pass filter function is a rectangular function.

The inverse Fourier transform of a rectangular function is a sinc function.

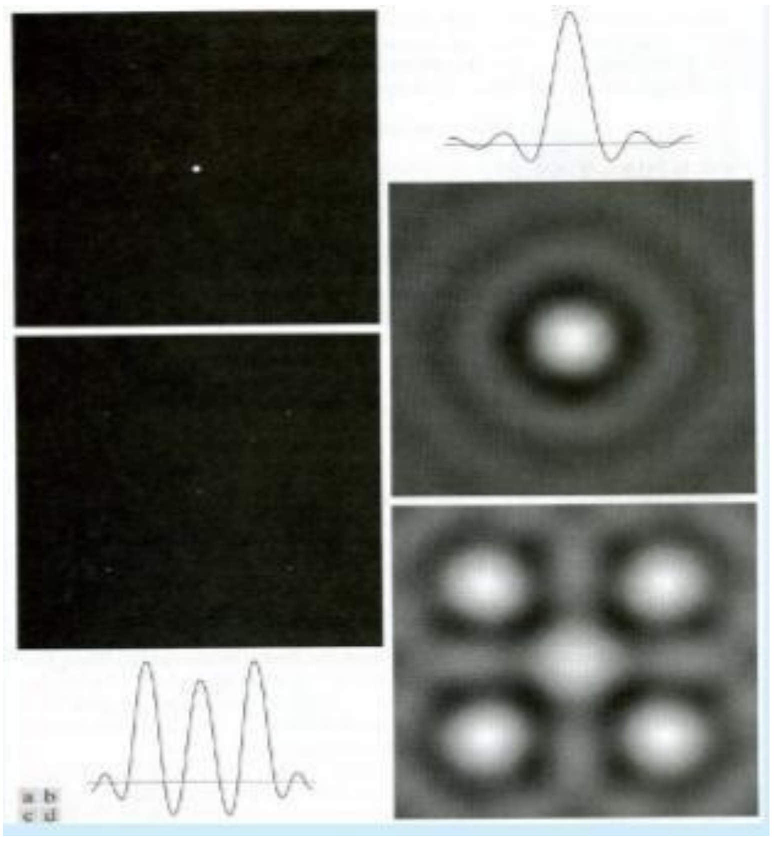


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity profiles through the center of the filters (the size of all cases is 1000x1000 and the cutoff frequency is 5)

Observe how ringing increases as a function of filter order.

BUTTERWORTH LOW-PASS FILTER:

Transferfunction of a Butterworth lowpass filter (BLPF) of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as

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

-

Transfer function does not have sharp discontinuity establishing cutoff between passed and filtered frequencies.

Cut off frequency D_0 defines point at which $H(u,v) = 0.5$

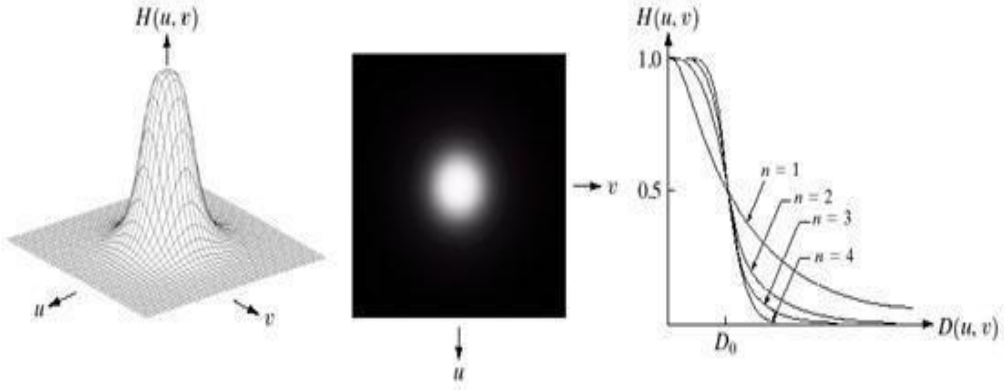


Fig. (a) perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c)Filter radial cross sections of order 1 through 4.

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that gives a clear cutoff between passed and filtered frequencies.

BUTTERWORTH LPF OF DIFFERENTFREQUENCIES:

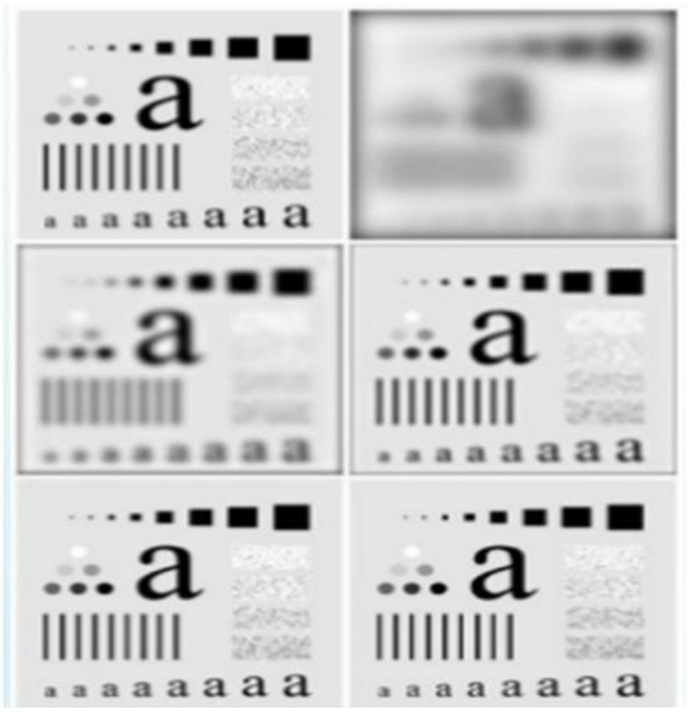


Fig. (a) Original image. (b)-(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii

Fig. shows the results of applying the BLPF of eq. to fig.(a), with n=2 and D0 equal to the five radii in fig.(b) for the ILPF, we note here a smooth transition in blurring as a function of increasing cutoff frequency. Moreover, no ringing is visible in any of the images processed with this particular BLPF, a fact attributed to the filter’s smooth transition between low and high frequencies.

A BLPF of order 1 has no ringing in the spatial domain. Ringing generally is imperceptible in filters of order 2, but can become significant in filters of higher order.Figure shows a comparison between the spatial representation of BLPFs of various orders (using a cutoff frequency of 5 in all cases). Shown also is the intensity profile along a horizontal scan line through the center of each filter. The filter of order 2 does show mild ringing and small negative values, but they certainly are less pronounced than in the ILPF. A butter worth filters of order 20 exhibits characteristics similar to those of the ILPF (in the limit, both filters are identical).

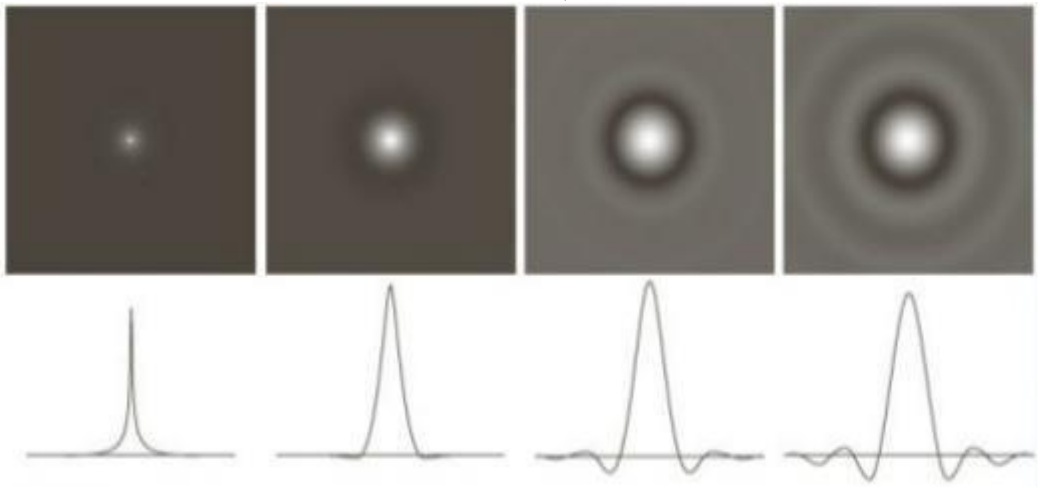


Fig.2.2.7 (a)-(d) Spatial representation of BLPFs of order 1, 2, 5 and 20 and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 x 1000 and the cutoff frequency is 5) Observe how ringing increases as a function of filter order.

GAUSSIAN LOWPASS FILTERS:
The form of these filters in two dimensions is given by

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

This transfer function is smooth, like Butterworth filter.
Gaussian in frequency domain remains a Gaussian in spatial domain
Advantage: No ringing artifacts.

Where D0 is the cutoff frequency. When D(u,v) = D0, the GLPF is down to 0.607 of its maximum value. This means that a spatial Gaussian filter, obtained by computing the IDFT of above equation., will have no ringing.

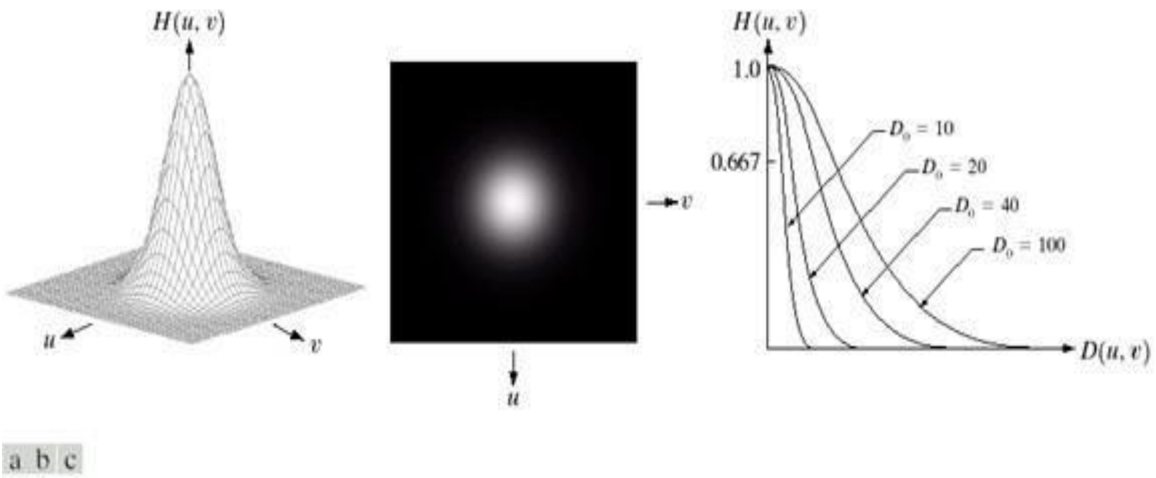


Figure shows a perspective plot, image display and radial cross sections of a GLPF function.

Fig. (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

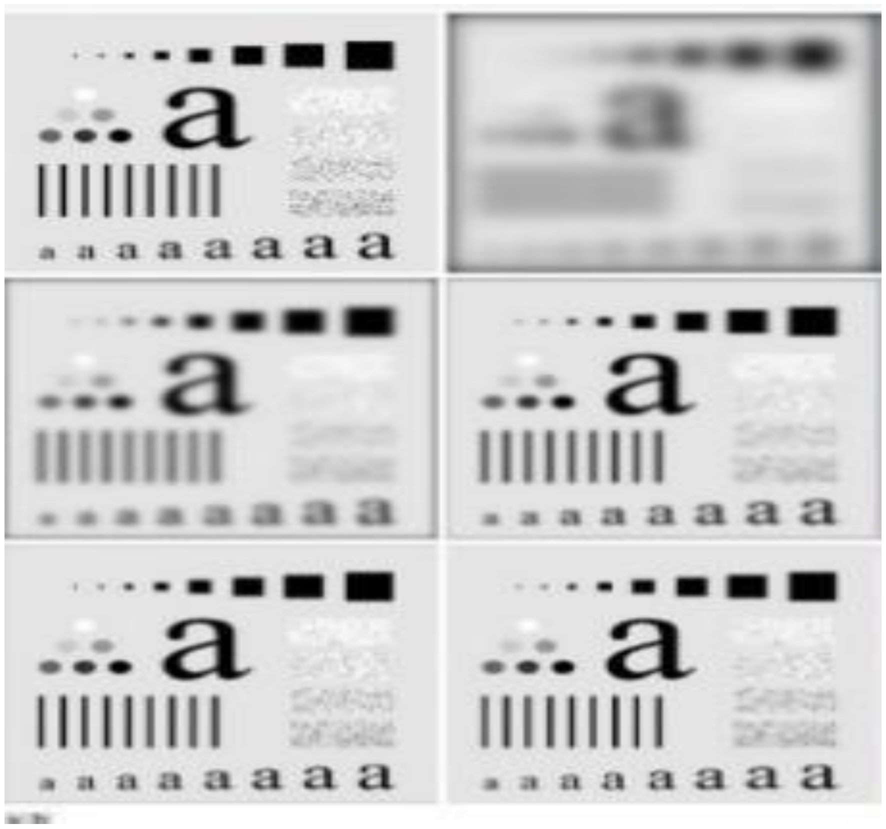


Fig.(a) Original image. (b)-(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in fig.2.2.2. compare with fig.2.2.3 and fig.2.2.6

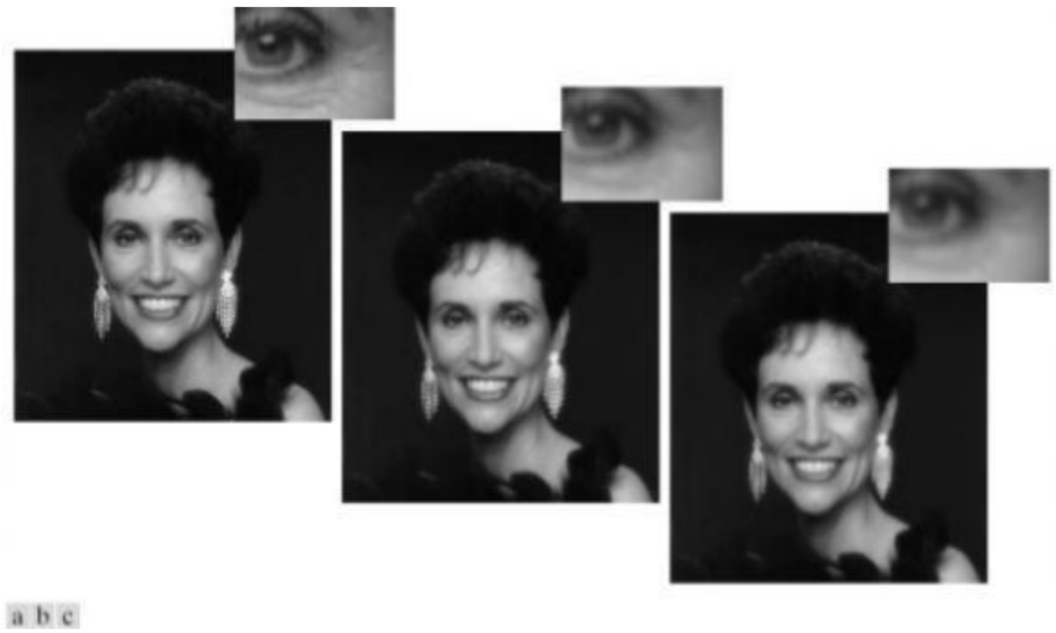


Fig. (a) Original image (784x 732 pixels)
(b) Result of filtering using a GLPF with D0 = 100
(c) Result of filtering using a GLPF with D_0 = 80. Note the reduction in fine skin lines in the magnified sections in (b) and (c)

Fig. shows an application of lowpass filtering for producing a smoother, softer- looking result from a sharp original. For human faces, the typical objective is to reduce the sharpness of fine skin lines and small blemished.

IMAGE SHARPENING USING FREQUENCY DOMAIN FILTERS:

An image can be smoothed by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in intensities are associated with high-frequency components, image sharpening can be achieved in the frequency domain by high pass filtering, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.
The filter function H(u,v) are understood to be discrete functions of size PxQ; that is the discrete frequency variables are in the range

$$u = 0,1, 2\ldots\ldots.,P-1 \text{ and}$$
$$v = 0,1,2,\ldots\ldots.,(Q-1)$$

The meaning of sharpening isedges and fine detail characterized by sharp transitions in image intensity. Such transitions contribute significantly to high frequency components of Fourier transform. Intuitively, attenuating certain low frequency components and preserving high frequency components result in sharpening.Intended goal is to do the reverse operation of low-pass filters.
When low-pass filter attenuated frequencies, high-pass filter passes them.

When high-pass filter attenuates frequencies, low-pass filter passes them. A high pass filter is obtained from a given low pass filter using the equation.

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

Where $H_{lp}(u,v)$ is the transfer function of the low-pass filter. That is when the low- pass filter attenuates frequencies, the high-pass filter passed them, and vice-versa.

We consider ideal, Butter-worth, and Gaussian high-pass filters. As in the previous section, we illustrate the characteristics of these filters in both the frequency and spatial domains. Fig shows typical 3-D plots, image representations and cross sections for these filters. As before, we see that the Butter-worth filter represents a transition between the sharpness of the ideal filter and the broad smoothness of the Gaussian filter. Figdiscussed in the sections the follow, illustrates what these filters look like in the spatial domain. The spatial filters were obtained and displayed by using the procedure used.

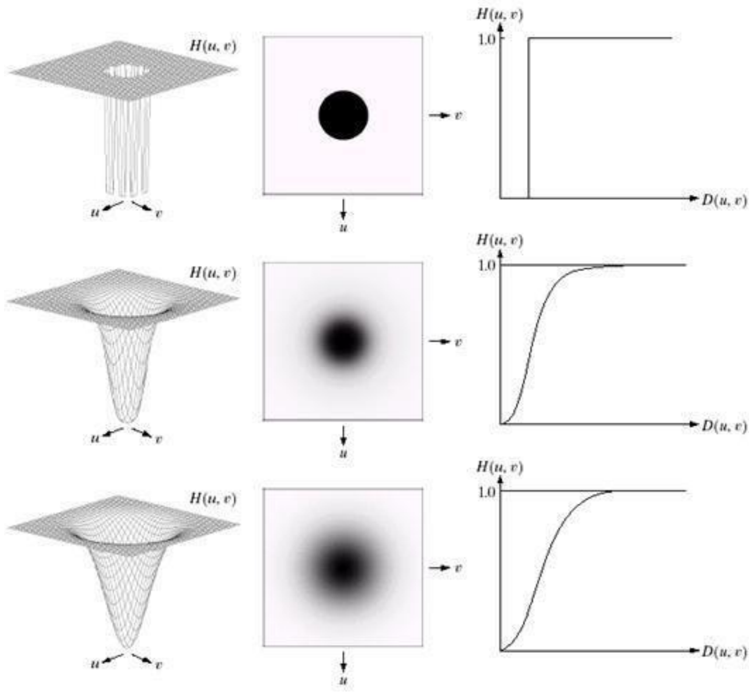


Fig: Top row: Perspective plot, image representation, and cross section of a typical ideal high-pass filter. Middle and bottom rows: The same sequence for typical butter-worth and Gaussian high-pass filters.

IDEAL HIGH-PASS FILTER:

A 2-D ideal high-pass filter (IHPF) is defined as

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

Where D_0 is the cutoff frequency and $D(u,v)$ is given by eq. As intended, the IHPF is the opposite of the ILPF in the sense that it sets to zero all frequencies inside a circle ofradius D_0 while passing, without attenuation, all frequencies outside the circle. As in case of the ILPF, the IHPF is not physically realizable.

SPATIAL REPRESENTATION OF HIGHPASS FILTERS:

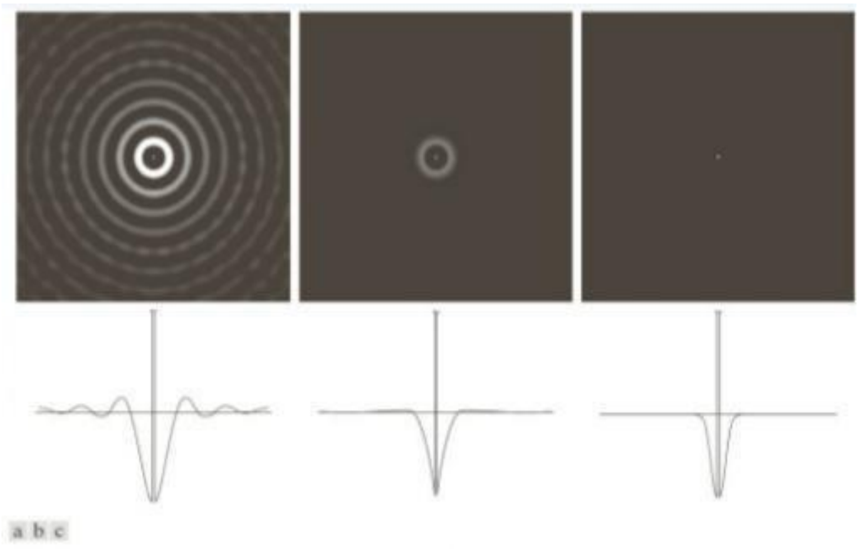


Fig Spatial representation of typical (a) ideal (b) Butter-worth and (c) Gaussian frequency domain high-pass filters, and corresponding intensity profiles through their centers.

We can expect IHPFs to have the same ringing properties as ILPFs. This is demonstrated clearly in Fig. which consists of various IHPF results using the original image in Fig.(a) with D_0 set to 30, 60, and 160 pixels, respectively. The ringing in Fig. (a) is so severe that it produced distorted, thickened object boundaries (E.g., look at the large letter “a”). Edges of the top three circles do not show well because they are not as strong as the other edges in the image (the intensity of these three objects is much closer to the background intensity, giving discontinuities of smaller magnitude).

FILTERED RESULTS OF IHPF:

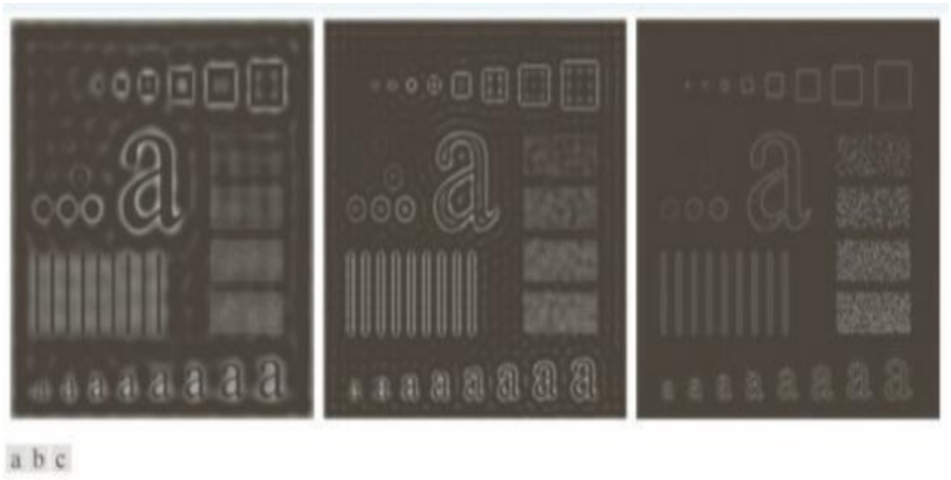


Fig. Results of high-pass filtering the image in Fig.(a) using an IHPF with D_0 = 30, 60, and 160

The situation improved somewhat with $D_0 = 60$. Edge distortion is quite evident still, but now we begin to see filtering on the smaller objects. Due to the now familiar inverse relationship between the frequency and spatial domains, we know that the spot size of this filter is smaller than the spot of the filter with $D_0 = 30$. The result for $D_0 = 160$ is closer to what a high-pass filtered image should look like. Here, the edges are much cleaner and less distorted, and the smaller objects have been filtered properly.Of course, the constant background in all images is zero in these high-pass filtered images because high pass filtering is analogous to differentiation in the spatial domain.

BUTTER-WORTH HIGH-PASS FILTERS:

A 2-D Butter-worth high-pass filter (BHPF) of order n and cutoff frequency D0 is defined as

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

Where D(u,v) is given by Eq.(3). This expression follows directly from Eqs.(3) and (6). The middle row of Fig.2.2.11. shows an image and cross section of the BHPF function.Butter-worth high-pass filter to behave smoother than IHPFs. Fig.2.2.14. shows the performance of a BHPF of order 2 and with D0 set to the same values as in Fig.2.2.13. The boundaries are much less distorted than in Fig.2.2.13. even for the smallest value of cutoff frequency.

FILTERED RESULTS OF BHPF:

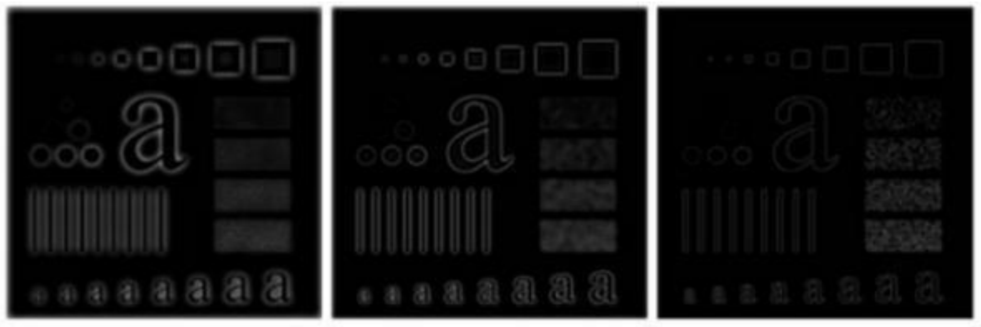

a b c

Fig. Results of high-pass filtering the image in Fig.2.2.2(a) using a BHPF of order 2 with D0 = 30, 60, and 160 corresponding to the circles in

Fig.2.2.2(b). These results are much smoother than those obtained with an IHPF

GAUSSIAN HIGH-PASS FILTERS:

The transfer function of the Gaussian high-pass filter (GHPF) with cutoff frequency locus at a distance D0 from the center of the frequency rectangle is given by

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

Where D(u,v) is given by Eq.(4). This expression follows directly from Eqs.(2) and (6). The third row in Fig.2.2.11. shows a perspective plot, image and cross section of the GHPF function. Following the same format as for the BHPF, we show in Fig.2.2.15. comparable results using GHPFs. As expected, the results obtained are more gradual than with the previous two filters.

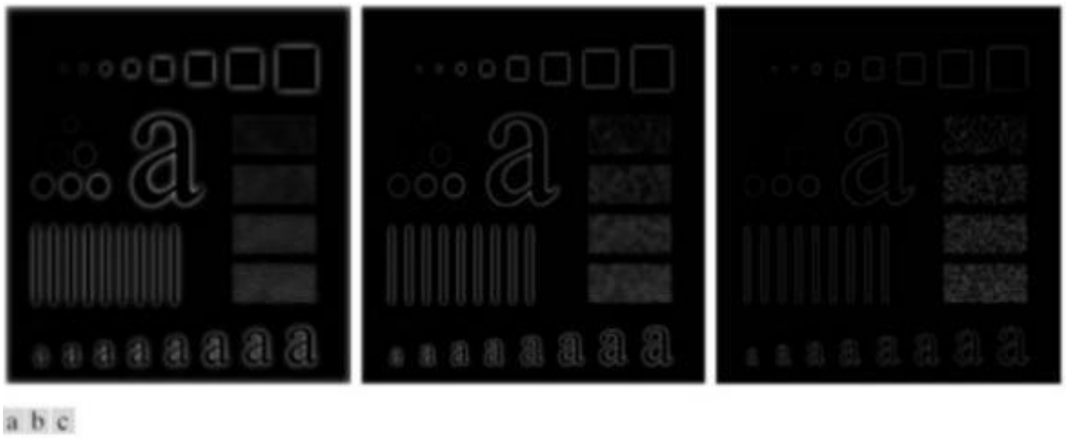


Fig. Results of high-pass filtering the image in fig.(a) using a GHPF with D_0 = 30, 60 and 160, corresponding to the circles in Fig.(b)

UNIT-3
IMAGE RESTORATION AND SEGMENTATION

Learning Objectives:

The goal of image restoration is to reconstruct the original scene from a degraded observation. After reading this unit, the reader should be familiar with the following concepts:

1. Different types of image degradation
2. Linear image –restoration techniques
3. Nonlinear image restoration techniques

IMAGE RESTORATION:

Restoration improves image in some predefined sense. It is an objective process. Restoration attempts to reconstruct an image that has been degraded by using a priori knowledge of the degradation phenomenon. These techniques are oriented toward modeling the degradation and then applying the inverse process in order to recover the original image. Restoration techniques are based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding what constitutes a “good” enhancement result. Image Restoration refers to a class of methods that aim to remove or reduce the degradations that have occurred while the digital image was being obtained. All-natural images when displayed have gone through some sort of degradation:

1. During display mode
2. Acquisition mode, or
3. Processing mode
4. Sensor noise
5. Blur due to camera mis focus
6. Relative object-camera motion
7. Random atmospheric turbulence
8. Others

DEGRADATION MODEL:

Degradation process operates on a degradation function that operates on an input image with an additive noise term. Input image is represented by using the notation f(x,y), noise term can be represented as η(x,y).These two terms when combined gives the result as g(x,y). If we are given g(x,y), some knowledge about the degradation function H or J and some knowledge about the additive noise teem η(x,y), the objective of restoration is to obtain an estimate f'(x,y) of the original image. We want the estimate to be as close as possible to the original image. The more we know about h and η , the closer f(x,y) will be to f'(x,y). If it is a linear position invariant process, then degraded image is given in the spatial domain by

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

where h(x,y) is spatial representation of degradation function and symbol * represents convolution. In frequency domain we may write this equation as

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

The terms in the capital letters are the Fourier Transform of the corresponding terms in the spatial domain.

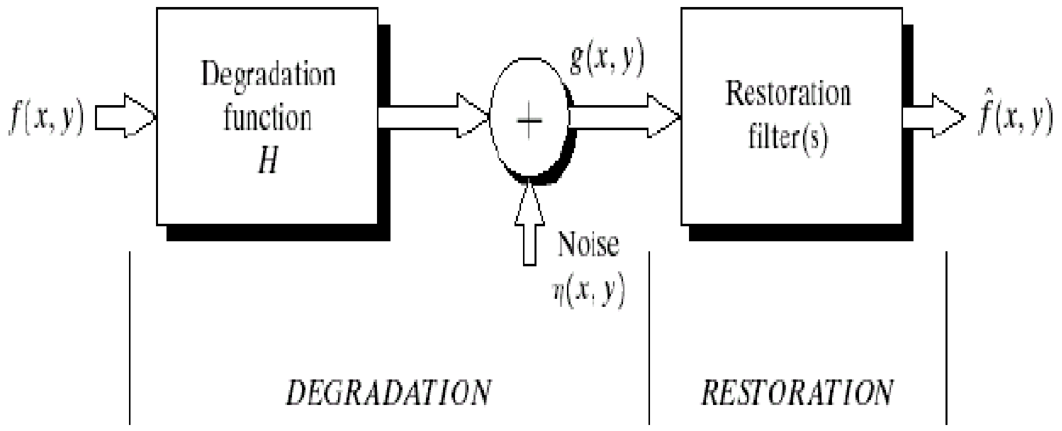


Fig: A model of the image Degradation / Restoration Process

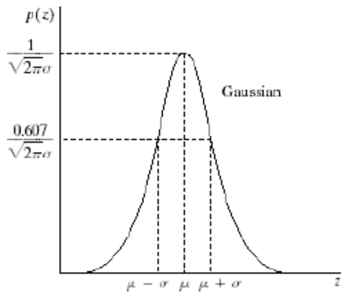
NOISE MODELS:

The principal source of noise in digital images arises during image acquisition and /or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted during transmission principally due to interference in the channels used for transmission. Since main sources of noise presented in digital images are resulted from atmospheric disturbance and image sensor circuitry, following assumptions can be made i.e. the noise model is spatial invariant (independent of spatial location). The noise model is uncorrelated with the object function.

Gaussian Noise:

These noise models are used frequently in practices because of its tractability in both spatial and frequency domain. The PDF of Gaussian random variable is as follows, where z represents the gray level, μ= mean of average value 0

$$p_z(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$



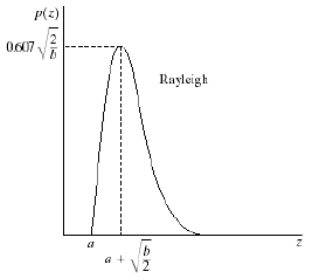
Rayleigh Noise:

Unlike Gaussian distribution, the Rayleigh distribution is no symmetric. It is given by the formula.

$$p_z(z) = \begin{cases} \dfrac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$$

The mean and variance of this density is

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



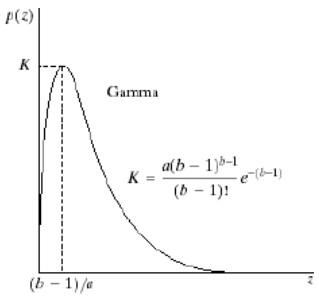
Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

$$\text{mean}: \mu = \frac{b}{a} \quad \text{variance}: \sigma^2 = \frac{b}{a^2}$$



Its shape is similar to Rayleigh disruption. This equation is referred to as gamma density it is correct only when the denominator is the gamma function.

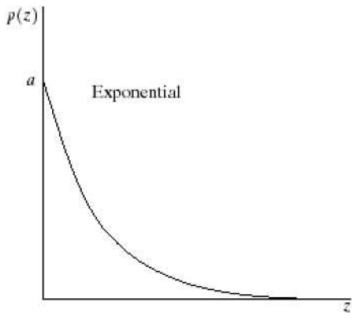
Exponential Noise:

Exponential distribution has an exponential shape. The PDF of exponential noise is given as

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

Where a>0. The mean and variance of this density are given by

$$m = \frac{1}{a}, \quad \sigma^2 = \frac{1}{a^2}$$



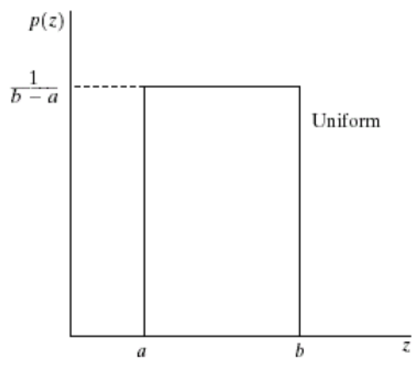
Uniform Noise:

The PDF of uniform noise is given by

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

The mean and variance of this noise is

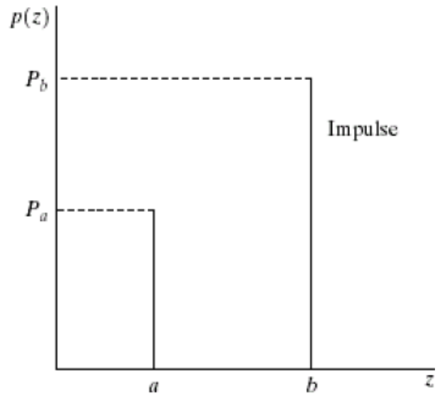
$$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$$



Impulse (Salt &Pepper) Noise:
In this case, the noise is signal dependent, and is multiplied to the image. The PDF of bipolar (impulse) noise is given by
If b>a, gray level b will appear as a light dot in image. Level a will appear like a dark dot.

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

$b > a$



Restoration in the Presence of Noise Only- Spatial Filtering:
When the only degradation present in an image is noise, i.e.

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

or

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

The noise terms are unknown so subtracting them from g(x,y) or G(u,v) is not a realistic approach. In the case of periodic noise it is possible to estimate N(u,v) from the spectrum G(u,v).So, N(u,v) can be subtracted

from G(u,v) to obtain an estimate of original image. Spatial filtering can be done when only additive noise is present.
The following techniques can be used to reduce the noise effect:

Mean Filter:

(a)Arithmetic Mean filter:

It is the simplest mean filter. Let Sxy represents the set of coordinates in the sub image of size m*n centered at point (x,y). The arithmetic mean filter computes the average value of the corrupted image g(x,y) in the area defined by Sxy. The value of the restored image f at any point (x,y) is the arithmetic mean computed using the pixels in the region defined by Sxy.

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

This operation can be using a convolution mask in which all coefficients have value 1/mm A mean filter soothes local variations in image Noise is reduced as a result of blurring. For every pixel in the image, the pixel value is replaced by the mean value of its neighboring pixels with a weight. This will be resulted in a smoothing effect in the image.

Geometric Mean Filter:

An image restored using a geometric mean filter is given by the expression

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

Here, each restored pixel is given by the product of the pixel in the sub image window, raised to the power 1/mm. A geometric mean filter but it to loosen image details in the process.

Harmonic Mean Filter:

The harmonic mean filtering operation is given by the expression

$$\hat{f}(x,y)=\sum_{(s,t)\in Sxy} g(s,t)^{Q+1} \Big/ \sum_{(s,t)\in Sxy} g(s,t)^{Q}$$

The harmonic mean filter works well for salt noise but fails for pepper noise. It does well with Gaussian noise also.

Order Statistics Filter:

Order statistics filters are spatial filters whose response is based on ordering the pixel contained in the image area encompassed by the filter.

The response of the filter at any point is determined by the ranking result.

Median filter:
It is the best order statistic filter; it replaces the value of a pixel by the median of gray levels in $f(x,y) = \underset{(s,t)\in Sxy}{\text{median}}\{g(s,t)\}$ pixel.The original of the pixel is included in the computation of the median of the filter are quite possible because for certain types of random noise, the provide excellent noise reduction capabilities with considerably less blurring then smoothing filters of similar size. These are effective for bipolar and unipolar impulse noise.

Max and Min Filter:
Using the 100th percentile of ranked set of numbers is called the max filter and is given by the equation.
It is used for finding the brightest point in an image. Pepper noise in the image has very low values, it is reduced by max filter using the max selection process in the sublimated area sky. The 0th percentile filter is

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

min filter.

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

This filter is useful for flinging the darkest point in image. Also, it reduces salt noise of the min operation.

Midpoint Filter:
The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by its comeliness the order statistics and averaging. This filter works best for randomly distributed noise like Gaussian or uniform noise.

$$\hat{f}(x,y) = \left(\max_{(s,t)\in Sxy}\{g(s,t)\} + \min_{(s,t)\in Sxy}\{g(s,t)\} \right) \Big/ 2$$

Periodic Noise by Frequency Domain Filtering:
These types of filters are used for this purpose.
Band Reject Filters: It removes a band of frequencies about the origin of the Fourier transformer.
Ideal Band reject Filter: An ideal band reject filter is given by the

$$H(u,v) = \begin{cases} 1 & \text{if} \quad D(u,v) < D_0 - W/2 \\ 0 & \text{if} \quad D_0 - W/2 \le D(u,v) \le D_0 + W/2 \\ 1 & \text{if} \quad D(u,v) > D_0 + W/2 \end{cases}$$

expression
D(u,v)- the distance from the origin of the centered frequency rectangle, W- the width of the band and D_o- the radial center of the frequency rectangle.

Butterworth Band Reject Filter:

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

Gaussian Band Reject Filter:

$$H(u,v) = 1 - \exp\left[-\frac{1}{2} \left(\frac{D^2(u,v) - D_0^2}{D(u,v)W} \right)^2 \right]$$

These filters are mostly used when the location of noise component in the frequency domain is known. Sinusoidal noise can be easily removed by using these kinds of filters because it shows two impulses that are mirror images of each other about the origin. Of the frequency transform.

Band Pass Filter:

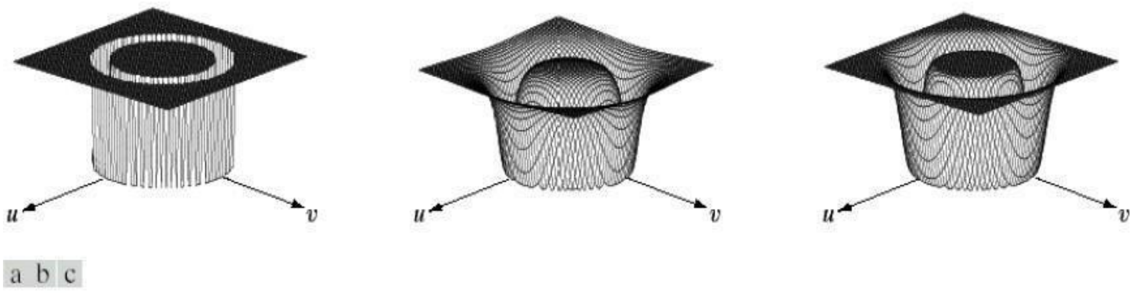


FIGURE From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

The function of a band pass filter is opposite to that of a band reject filter It allows a specific frequency band of the image to be passed and blocks the rest of frequencies. The transfer function of a band pass filter can be obtained from a corresponding band reject filter with transfer function Hbr(u,v) by using the equation

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

These filters cannot be applied directly on an image because it may remove too much details of an image but these are effective in isolating the effect of an image of selected frequency bands.

Notch Filters:

A notch filter rejects (or passes) frequencies in predefined neighborhoods

$$D_1(u,v)=\sqrt{(u-M/2-u_0)^2+(v-N/2-v_0)^2}$$
$$D_2(u,v)=\sqrt{(u-M/2+u_0)^2+(v-N/2+v_0)^2}$$

about a center frequency.Due to the symmetry of the Fourier transform notch filters must appear in symmetric pairs about the origin.The transfer function of an ideal notch reject filter of radius D_0 with centers a (u_0,v_0) and by symmetry at $(-u_0\ ,\ v_0)$ is

Ideal, Butterworth, Gaussian Notch Filters:

$$H(u,v)=\begin{cases}0 & \text{if} \quad D_1(u,v)\le D_0 \text{ or } D_2(u,v)\le D_0\\ 1 & \text{otherwise}\end{cases}$$

$$H(u,v)=1\Bigg/\left[1+\left(\frac{D_0^2}{D_1(u,v)D_2(u,v)}\right)^n\right]$$

$$H(u,v)=1-\exp\left[-\frac{1}{2}\left(\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right)\right]$$

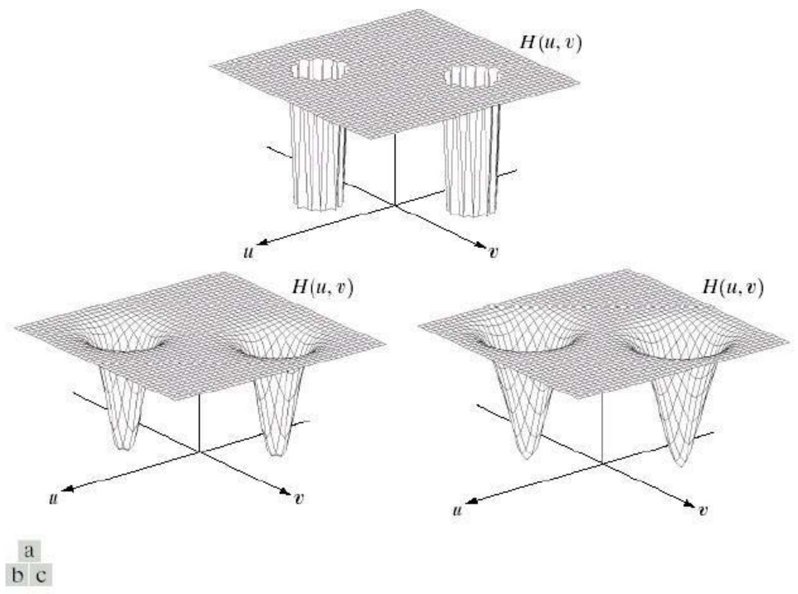


FIGURE Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Inverse Filtering:

The simplest approach to restoration is direct inverse filtering where we complete anestimate $\hat{F}(u,v)$ of the transform of the original image simply by dividing the transform of the degraded image G(u,v) by degradation

$$\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}$$

function H(u,v)

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

We know that

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

Therefore

From the above equation we observe that we cannot recover the undegraded image exactly because N(u,v) is a random function whose Fourier transform is not known. One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin.We know that H(0,0) is equal to the average values of h(x,y). By Limiting the analysis to frequencies near the origin we reduse the probability of encountering zero values.

Minimum Mean Square Error (Wiener) Filtering:

The inverse filtering approach has poor performance. The wiener filtering approach uses the degradation function and statistical characteristics of noise into the restoration process.The objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized.

The error measure is given by

$$e^2 = E\{[f(x) - \hat{f}(x)]^2$$

Where E{.} is the expected value of the argument.

We assume that the noise and the image are uncorrelated one or the other has zero mean.The gray levels in the estimate are a linear function of the levels in the degraded image.

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

Here, H(u,v)= degradation function,

H*(u,v)=complex conjugate of H(u,v)

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

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

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

The power spectrum of the undegraded image is rarely known. An approach used frequently when these quantities are not known or cannot be estimated then the expression used is

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

Where K is a specified constant.

Constrained Least Squares Filtering:

The wiener filter has a disadvantage that we need to know the power spectra of the undegraded image and noise. The constrained least square filtering requires only the knowledge of only the mean and variance of the noise. These parameters usually can be calculated from a given degraded image this is the advantage with this method. This method produces an optimal result. This method requires the optimal criteria which is important we express thein vector-matrix form

$$\mathbf{g} = \mathbf{Hf} + \boldsymbol{\eta}$$

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

The optimality criteria for restoration is based on a measure of smoothness, such as the second derivative of an image (Laplacian).
The minimum of a criterion function C defined as

$$C = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\nabla^2 f(x,y)\right]^2$$

Subject to the constraint

$$\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\|^2 = \|\boldsymbol{\eta}\|^2$$

Where $\|\mathbf{w}\|^2 \triangleq \mathbf{w}^T\mathbf{w}$ is a Euclidean vector norm $\hat{\mathbf{f}}$ is estimate of the undegraded image, ∇^2 is Laplacian operator.
The frequency domain solution to this optimization problem is given by

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

Where γ is a parameter that must be adjusted so that the constraint is satisfied. P(u,v) is the Fourier transform of the Laplacian operator

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

IMAGE SEGMENTATION:

Edge Detection:

Edge detection is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction, which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities.

Motivation: Canny edge detection applied to a photograph

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

1. Discontinuities in depth,
2. Discontinuities in surface orientation,
3. Changes in material properties and
4. Variations in scene illumination

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image.

If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. However, it is not always possible to obtain such ideal edges from real life images of moderate complexity. Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, missing edge segments as well as false edges notcorresponding to interesting phenomena in the image – thus complicating the subsequent task of interpreting the image data.

Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques. During recent years, however, substantial (and successful) research has also been made on computer vision methods that do not explicitly rely on edge detection as a pre-processing step.

Edge Properties:

The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent. A viewpoint independent edge typically reflects inherent properties of the three-dimensional objects, such as surface markings and surface shape. A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects occluding one another.

A typical edge might for instance be the border between a block of red color and a block of yellow. In contrast a line(as can be extracted by a ridge detector) can be a small number of pixels of a different color on an otherwise unchanging background. For a line, there may therefore usually be one edge on each side of the line.

A Simple Edge Model:

Although certain literature has considered the detection of ideal step edges, the edges obtained from natural images are usually not at all ideal step edges. Instead they are normally affected by one or several of the following effects:

1. Focal blur caused by a finite depth-of-field and finite point spread function.
2. Penumbral blur caused by shadows created by light sources of non-zero radius.
3. Shading at a smooth object

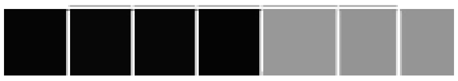
A number of researchers have used a Gaussian smoothed step edge (an error function) as the simplest extension of the ideal step edge model for modeling the effects of edge blur in practical applications. Thus, a one-dimensional image f which has exactly one
edge placed at x = 0 may be modeled as:

$$f(x) = \frac{I_r - I_l}{2}\left(\operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + 1\right) + I_l.$$

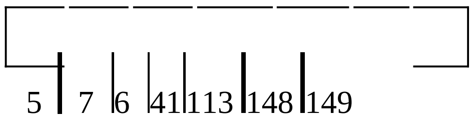
At the left side of the edge, the intensity is $I_l = \lim_{x\to-\infty} f(x)$, and right of the edge it is $I_r = \lim_{x\to\infty} f(x)$. The scale parameter σ is called the blur scale of the edge.

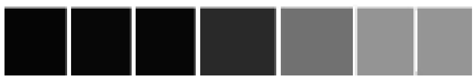
Why edge detection is a non-trivial task?

To illustrate why edge detection is not a trivial task, let us consider the problem of detecting edges in the following one-dimensional signal. Here, we may intuitively say that there should be an edge between the 4th and 5th pixels.



If the intensity difference were smaller between the 4th and the 5th pixels and if the intensity differences between the adjacent neighboring pixels were higher, it would not be as easy to say that there should be an edge in the corresponding region. Moreover, one could argue that this case is one in which there are several edges.





Hence, to firmly state a specific threshold on how large the intensity change between two neighboring pixels must be for us to say that there should be an edge between these pixels is not always simple. Indeed, this is one of the reasons why edge detection may be a non-trivial problem unless the objects in the scene are particularly simple and the illumination conditions can be well controlled (see for example, the edges extracted from the image with the girl above).

Approaches to Edge Detection:

There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression.

As a pre-processing step to edge detection, a smoothing stage, typically Gaussian smoothing, is almost always applied (see also noise reduction). The edge detection methods that have been published mainly differ in the types of smoothing filters that are applied and the way the measures of edge strength are computed. As many edge detection methods rely on the computation of image gradients, they also differ in the types of filters used for computing gradient estimates in the x- and y-directions.

Canny Edge Detection:

John Canny considered the mathematical problem of deriving an optimal smoothing filter given the criteria of detection, localization and minimizing multiple responses to a single edge. He showed that the optimal filter given these assumptions is a sum of four exponential terms. He also showed that this filter can be well approximated by first-order derivatives of Gaussians. Canny also introduced the notion of non-maximum suppression, which means that given the pre-smoothing filters, edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. Looking for the zero crossing of the 2nd derivative along the gradient direction was first proposed by Haralick. It took less than two decades to find a modern geometric variational meaning for that operator that links it to the Marr-Hildreth (zero crossing of the Laplacian) edge detector. That observation was presented by Ron Kimmel and Alfred Bruckstein.

Although his work was done in the early days of computer vision, the Canny edge detector (including its variations) is still a state-of-the-art edge detector. Unless the preconditions are particularly suitable, it is hard to find an edge detector that performs significantly better than the Canny edge detector.

The Canny-Deriche detector was derived from similar mathematical criteria as the Canny edge detector, although starting from a discrete viewpoint and then leading to a set of recursive filters for image smoothing instead of exponential filters or Gaussian filters.The differential edge detector described below can be seen as a reformulation of Canny's method from the viewpoint of differential invariants computed from a scale-space representation leading to a number of advantages in terms of both theoretical analysis and sub-pixel implementation.

Other First-Order Methods:

For estimating image gradients from the input image or a smoothed version of it, different gradient operators can be applied. The simplest approach is to use central differences:

$$L_x(x,y) = -1/2 \cdot L(x-1,y) + 0 \cdot L(x,y) + 1/2 \cdot L(x+1,y)$$
$$L_y(x,y) = -1/2 \cdot L(x,y-1) + 0 \cdot L(x,y) + 1/2 \cdot L(x,y+1),$$

corresponding to the application of the following filter masks to the image data:

$$L_x = \begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix} * L \quad \text{and} \quad L_y = \begin{bmatrix} +1/2 \\ 0 \\ -1/2 \end{bmatrix} * L.$$

$$L_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * L \quad \text{and} \quad L_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * L.$$

The well-known and $\theta = \operatorname{atan2}(L_y, L_x).$ earlier Sobel operator is based on the following filters:

Given such estimates of first- order derivatives, the gradient magnitude is then computed as:

while the gradient orientation can be estimated as

$$|\nabla L| = \sqrt{L_x^2 + L_y^2}$$

Other first-order difference operators for estimating image gradient have been proposed in the Prewitt operator and Roberts cross.

Thresholding and Linking:

Once we have computed a measure of edge strength (typically the gradient magnitude), the next stage is to apply a threshold, to decide whether edges are present or not at an image point. The lower the threshold, the more edges will be detected, and the result will be increasingly susceptible to noise and detecting edges of irrelevant features in the image. Conversely a high threshold may miss subtle edges, or result in fragmented edges.

If the edge thresholding is applied to just the gradient magnitude image, the resulting edges will in general be thick and some type of edge thinning post-processing is necessary. For edges detected with non-maximum suppression however, the edge curves are thin by definition and the edge pixels can be linked into edge polygon by an edge linking (edge tracking) procedure. On a discrete grid, the non-maximum suppression stage can be implemented by estimating the gradient direction using first-order derivatives, then rounding off the gradient direction to multiples of 45 degrees, and finally comparing the values of the gradient magnitude in the estimated gradient direction.

A commonly used approach to handle the problem of appropriate thresholds for thresholding is by using thresholding with hysteresis. This method uses multiple thresholds to find edges. We begin by using the upper threshold to find the start of anedge. Once we have a start point, we then trace the path of the edge through the image pixel by pixel, marking an edge whenever we are above the lower threshold. We stop marking our edge only when the value falls below our lower threshold. This approach makes the assumption that edges are likely to be in continuous curves, and allows us to follow a faint section of an edge we have previously seen, without meaning that every noisy pixel in the image is marked down as an edge. Still, however, we have the problem of choosing appropriate thresholding parameters, and suitable thresholding values may vary over the image.

Edge Thinning:

Edge thinning is a technique used to remove the unwanted spurious points on the edge of an image. This technique is employed after the image has been filtered for noise (using median, Gaussian filter etc.), the edge operator has been applied (like the ones described above) to detect the edges and after the edges have been smoothed using an appropriate threshold value. This removes all the unwanted points and if applied carefully, results in one-pixel thick edge elements.

Advantages:

1) Sharp and thin edges lead to greater efficiency in object recognition.
2) If you are using Hough transforms to detect lines and ellipses then thinning could give much better results.
3) If the edge happens to be boundary of a region then, thinning could easily give the image parameters like perimeter without much algebra.

There are many popular algorithms used to do this, one such is described below:

1. Choose a type of connectivity, like 8, 6 or 4.
2. 8 connectivity is preferred, where all the immediate pixels surrounding a particular pixel are considered.
3. Remove points from North, south, east and west.
4. Do this in multiple passes, i.e. after the north pass, use the same semi processed image in the other passes and so on.

Remove a point if:

1. The point has no neighbors in the North (if you are in the north pass, and respective directions for other passes.)
2. The point is not the end of a line. The point is isolated.
3. Removing the points will not cause to disconnect its neighbors in any way.
4. Else keep the point. The number of passes across direction should be chosen according to the level of accuracy desired.

Second-Order approaches to Edge Detection:

Some edge-detection operators are instead based upon second-order derivatives of the intensity. This essentially captures the rate of change in the intensity gradient. Thus, in the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient.

The early Marr-Hildreth operator is based on the detection of zero-crossings of the Laplacian operator applied to a Gaussian-smoothed image. It can be shown, however, that this operator will also return false edges corresponding to local minima of the gradient magnitude. Moreover, this operator will give poor localization at curved edges. Hence, this operator is today mainly of historical interest.

Differential Edge Detection:

A more refined second-order edge detection approach which automatically detects edges with sub-pixel accuracy, uses the following differential approach of detecting zero- crossings of the second-order directional derivative in the gradient direction. Following the differential geometric way of expressing the requirement of non-maximum suppression proposed by Lindeberg, let us introduce at every image point a localcoordinate system (u,v), with the v-direction parallel to the gradient direction.

Assuming that the image has been pre-smoothed by Gaussian smoothing and a scale-spacerepresentation L(x,y;t) at scale t has been computed, we can require that the gradient magnitude of the scale-space representation, which is equal to the first-order directionalderivative in the v-direction Lv, should have its first order directional derivative in the v- direction equal to zero

$$\partial_v(L_v) = 0$$

while the second-order directional derivative in the v-direction of L_v should

$$\partial_{vv}(L_v) \leq 0.$$

be negative, i.e.,

Written out as an explicit expression in terms of local partial derivatives L_x, L_y ... L_{yyy}, this edge definition can be expressed as the zero-crossing curves of the differential invariant

that satisfy a sign-condition on the following differential invariant

$$L_v^2 L_{vv} = L_x^2\, L_{xx} + 2\, L_x\, L_y\, L_{xy} + L_y^2\, L_{yy} = 0,$$

$$L_v^3 L_{vvv} = L_x^3\, L_{xxx} + 3\, L_x^2\, L_y\, L_{xxy} + 3\, L_x\, L_y^2\, L_{xyy} + L_y^3\, L_{yyy} \leq 0$$

where L_x, L_y ... L_{yyy} denote partial derivatives computed from a scale-spacerepresentation L obtained by smoothing the original image with a Gaussian kernel. In this way, the edges will be automatically obtained as continuous curves with subpixel accuracy. Hysteresis thresholding can also be applied to these differential and subpixel edge segments.

In practice, first-order derivative approximations can be computed by central differences as described above, while second-order derivatives can be computed from the scale-space representation L according to:

$$L_{xx}(x,y) = L(x-1,y) - 2L(x,y) + L(x+1,y).$$
$$L_{xy}(x,y) = (L(x-1,y-1) - L(x-1,y+1) - L(x+1,y-1) + L(x+1,y+1))/4,$$
$$L_{yy}(x,y) = L(x,y-1) - 2L(x,y) + L(x,y+1).$$

corresponding to the following filter masks:

$$L_{xx} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} * L \quad \text{and} \quad L_{xy} = \begin{bmatrix} -1/4 & 0 & 1/4 \\ 0 & 0 & 0 \\ 1/4 & 0 & -1/4 \end{bmatrix} * L \quad \text{and} \quad L_{yy} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Higher-order derivatives for the third-order sign condition can be obtained in an analogous fashion.

Phase Congruency-based Edge Detection:

A recent development in edge detection techniques takes a frequency domain approach to finding edge locations. Phase congruency (also known as phase coherence) methods attempt to find locations in an image where all sinusoids in the frequency domain are in phase. These locations will generally correspond to the location of a perceived edge, regardless of whether the edge is represented by a large change in intensity in the spatial domain. A key benefit of this technique is that it responds strongly to Mach bands, and avoids false positives typically found around roof edges. A roof edge, is a discontinuity in the first order derivative of a grey-level profile.

Thresholding:

Thresholding is the simplest method of image segmentation. From agrayscale image, thresholding can be used to create binary images. During the thresholding process, individual pixels in an image are marked as object pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as ―background pixels otherwise. This convention isknown as **Threshold Above**. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside (Shapiro, et al. 2001:83). Typically, an object pixel is given a value of ―1‖ while a background pixel is given a value of ―0. ‖ Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels.

Threshold Selection:

The key parameter in the thresholding process is the choice of the threshold value (or values, as mentioned earlier). Several different methods for choosing a threshold exist; users can manually choose a threshold value, or a thresholding algorithm can compute a value automatically, which is known as automatic thresholding (Shapiro, et al. 2001:83). A simple method would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average.

In a noiseless image with uniform background and object values, the mean or median will work well as the threshold, however, this will generally not be the case. A more sophisticated approach might be to create a histogram of the image pixel intensitiesand use the valley point as the threshold. The histogram approach assumes that there is some average value for the background and object pixels, but that the actual pixel values have some variation around these average values. However, this may be computationally expensive, and image histograms may not have clearly defined valley points, often making the selection of an accurate threshold difficult.

One method that is relatively simple, does not require much specific knowledge of the image, and is robust against image noise, is the following iterative method:

- An initial threshold (T) is chosen, this can be done randomly or according to any other method desired.

- The image is segmented into object and background pixels as described above, creating two sets:

$$G1 = \{f(m,n):f(m,n)>T\} \text{ (object pixels)}$$
$$G2 = \{f(m,n):f(m,n)\ T\} \text{ (background pixels)}$$

Note: f(m,n) is the value of the pixel located in the mth column, nth row

The average of each set is computed.

$$m_1 = \text{average value of } G1$$
$$m_2 = \text{average value of } G2$$

A new threshold is created that is the average of m1 and m2

$$T = (m_1 + m_2)/2$$

Go back to step two, now using the new threshold computed in step four, keep repeating until the new threshold matches the one before it (i.e. until convergence has been reached).This iterative algorithm is a special one-dimensional case of the k-means clustering algorithm, which has been proven to converge at a local minimum—meaning that a different initial threshold may give a different final result.

Adaptive Thresholding:

Thresholding is called adaptive thresholding when a different threshold is used for different regions in the image. This may also be known as local or dynamic thresholding (Shapiro, et al. 2001:89).Mples are clustered in two parts as background and foreground (object), or alternately are modeled as a mixture of two Gaussiansentropy-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.object attribute-based methods search a measure of similarity between the gray- level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc. Spatial methods [that] use higher-order probability distribution and/or correlation between pixels. Local methods adapt the threshold value on each pixel to the local image characteristics.

Multiband Thresholding:

Colour images can also be thresholders. One approach is to designate a separate threshold for each of the RGB components of the image and then combine them with an AND operation. This reflects the way the camera works and how the data is stored in the computer, but it does not correspond to the way that people recognize color. Therefore, the HSL and HSV color models are more often used. It is also possible to use the CMYK color model (Pham et al., 2007).

Region Growing:

Region growing is a simple region-based image segmentation method. It is also classified as a pixel-based image segmentation method since it involves the selection of initial seed points.This approach to segmentation examines neighboring pixels of initial ―seed points‖ and determines whether the pixel neighbors should be added to the region. The process is iterated on, in the same manner as general data clustering algorithms.

Region-based Segmentation:

The main goal of segmentation is to partition an image into regions. Some

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

segmentation methods such as "Thresholding" achieve this goal by looking for the boundaries between regions based on discontinuities in gray levels or color properties. Region-basedsegmentation is a technique for determining the region directly. The basic formulation for Region-Based Segmentation is:

(b)Ri is a connected region, i = 1, 2, ...,n

(d)P(Ri) = TRUE for i = 1,2,...,n.

$$(c)\ R_i \bigcap R_j = \varnothing \text{ for all } i = 1, 2, ..., n.$$

$$(e) P(R_i \bigcup R_j) = FALSE \text{ for any adjacent region } R_i \text{ and } R_j.$$

Region-based Segmentation:

The main goal of segmentation is to partition an image into regions. Some segmentation methods such as "Thresholding" achieve this goal by looking for the boundaries between regions based on discontinuities in gray levels or color properties. Region-basedsegmentation is a technique for determining the region directly. The basic formulation for Region-Based Segmentation is:

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

(b)Ri is a connected region, i = 1, 2, ...,n

$$(c)\ R_i \bigcap R_j = \varnothing \text{ for all } i = 1, 2, ..., n.$$

(d)P(Ri) = TRUE for i = 1,2,...,n.

$$(e) P(R_i \bigcup R_j) = FALSE \text{ for any adjacent region } R_i \text{ and } R_j.$$

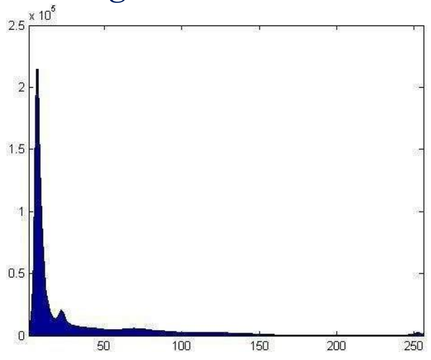
P(Ri) is a logical predicate defined over the points in set P(R_k) andisthe null set, means that the segmentation must be complete; that is, every pixel must be in a regionrequires that points in a region must be connected in some predefined senseindicates that the regions must be disjointdeals

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 gray levelindicates that region Ri and Rj are different in the sense of predicate P.

Basic Concept of Seed Points:

The first step in region growing is to select a set of seed points. Seed point selection is based on some user criterion (for example, pixels in a certain gray-level range, pixels evenly spaced on a grid, etc.). The initial region begins as the exact location of these seeds.The regions are then grown from these seed points to adjacent points depending on a region membership criterion. The criterion could be, for example, pixel intensity, gray level texture, or color. Since the regions are grown on the basis of the criterion, the image information itself is important. For example, if the criterion were a pixel intensity threshold value, knowledge of the histogram of the image would be of use, as one could use it to determine a suitable threshold value for the region membership criterion.

There is a very simple example followed below. Here we use 4-connected neighborhood to grow from the seed points. We can also choose 8-connected neighborhood for our pixel's adjacent relationship. And the criteria we make here is the same pixel value. That is, we keep examining the adjacent pixels of seed points. If they have the same intensity value with the seed points, we classify them into the seed points. It is an iterated process until there are no change in two successive iterative stages. Of course, we can make other criteria, but the main goal is to classify the similarity of the image into regions.



Some Important Issues:

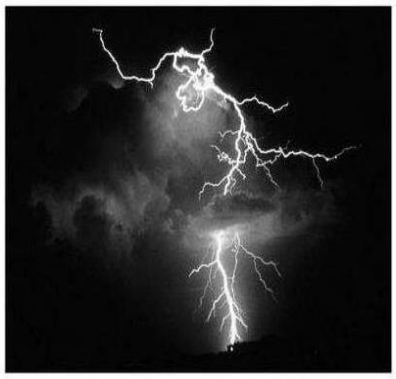
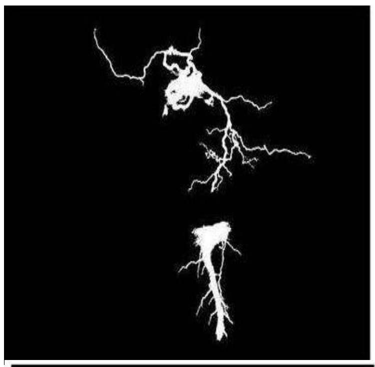
Then we can conclude several important issues about region growing. The suitable selection of seed points is important.The selection of seed points is depending on the users. For example, in a gray-level lightning image, we may want to segment the lightning from the background. Then probably, we can examine the histogram and choose the seed points from the highest range of it.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.The value, **“minimum area threshold”**, no region in region growing method result will be smaller than this threshold in the segmented image.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.

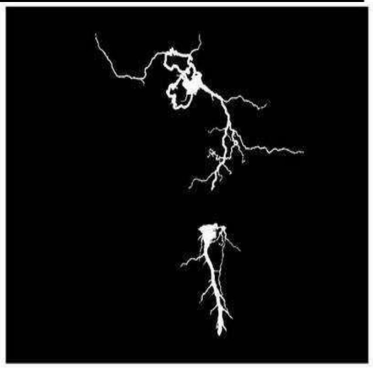
The criteria of similarities or so-called homogeneity we choose are also important. It usually depends on the original image and the segmentation result we want.Here are some criteria we often useGray level(average intensity or variance), color, and texture or shape.

Simulation Examples:

Here we show a simple example for region growing.Figure. 1 is the original image which is a gray-scale lightning image. The gray-scale value of this image is from 0 to 255. The purpose we apply region growing on this image is that we want to mark the strongest lightning part of the image and we

Figures 1, 2 3 and 4



also want the result is connected without being split apart. Therefore, we choose the points having the highest gray-scale value which is 255 as the seed points showed in the Figure. 2.

After determining the seed points, we have to determine the range of threshold. Always keeps in mind that the purpose we want to do is to mark the strongest light in the image. The third figure is the region growing result from choosing the threshold between 225 and the value of seed points (which is 255). It means we only want to mark out the points whose gray-scale values are above 225.If we make the range of threshold wider, we will get a result having a bigger area of the lightning region show as the Figure. 3 and the Figure. 4.

We can observe the difference between the last two figures which have different threshold value showed above. Region growing provides the ability for us to separate the part we want connected.As we can see in Figure. 3 to Figure. 5, the segmented result in this example are seed-oriented connected. That means the result grew from the same seed points are the same regions. And the points will not be grown without connected with the seed points in the beginning. Therefore, we can mention that there are still lots of points in the original image having the gray-scale value above 155 which are not marked in Figure. 5. This characteristic ensures the reliability of the segmentation and provides the ability to resist noise. For this example, this characteristic prevents us marking out the non-lightning part in the image because the lightning is always connected as one part.The advantages and disadvantages of region growing. 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 determine the seed points and the criteria we want to make.
5. We can choose the multiple criteria at the same time.
6. It performs well with respect to noise.

Disadvantages

1. The computation is consuming, no matter the time or power.
2. Noise or variation of intensity may result in holes or over segmentation.
3. This method may not distinguish the shading of the real images.

We can conquer the noise problem easily by using some mask to filter the holes or outlier. Therefore, the problem or noise actually does not exist. In conclusion, it is obvious that the most serious problem of region growing is the power and time consuming.

Unit IV
IMAGE COMPRESSION

LEARNING OBJECTIVES:

The rapid growth of digital imaging applications, including desktop publishing, multimedia, teleconferencing and high –definition television has increased the need for effective and standard image –compression techniques. After completing this unit, the reader is expected to be familiar with the following concepts:

1. Need for image compression
2. Lossless and lossy image compression
3. Spatial domain and frequency domain image compression
4. Wavelet based compression
5. Image compression

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 where 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

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

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
where C_R, commonly called the compression ratio, is

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

For the case $n_2 = n_1$, $C_R = 1$ and $R_D = 0$, indicating that (relative to the second data set) the first representation of the information contains no redundant data. When $n_2 << n_1$, C_R tends to ∞ and R_D tends to 1, implying significant compression and highly redundant data. Finally, when $n_2 >> n_1$, C_R tends to 0 and R_D tends to ∞, 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, C_R and R_D 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.

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

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

probability pr (rk).

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 representeach value of r_k is l (r_k), then the average number of bits required to represent each pixel is

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

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

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 isMNL_{avg}.

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.

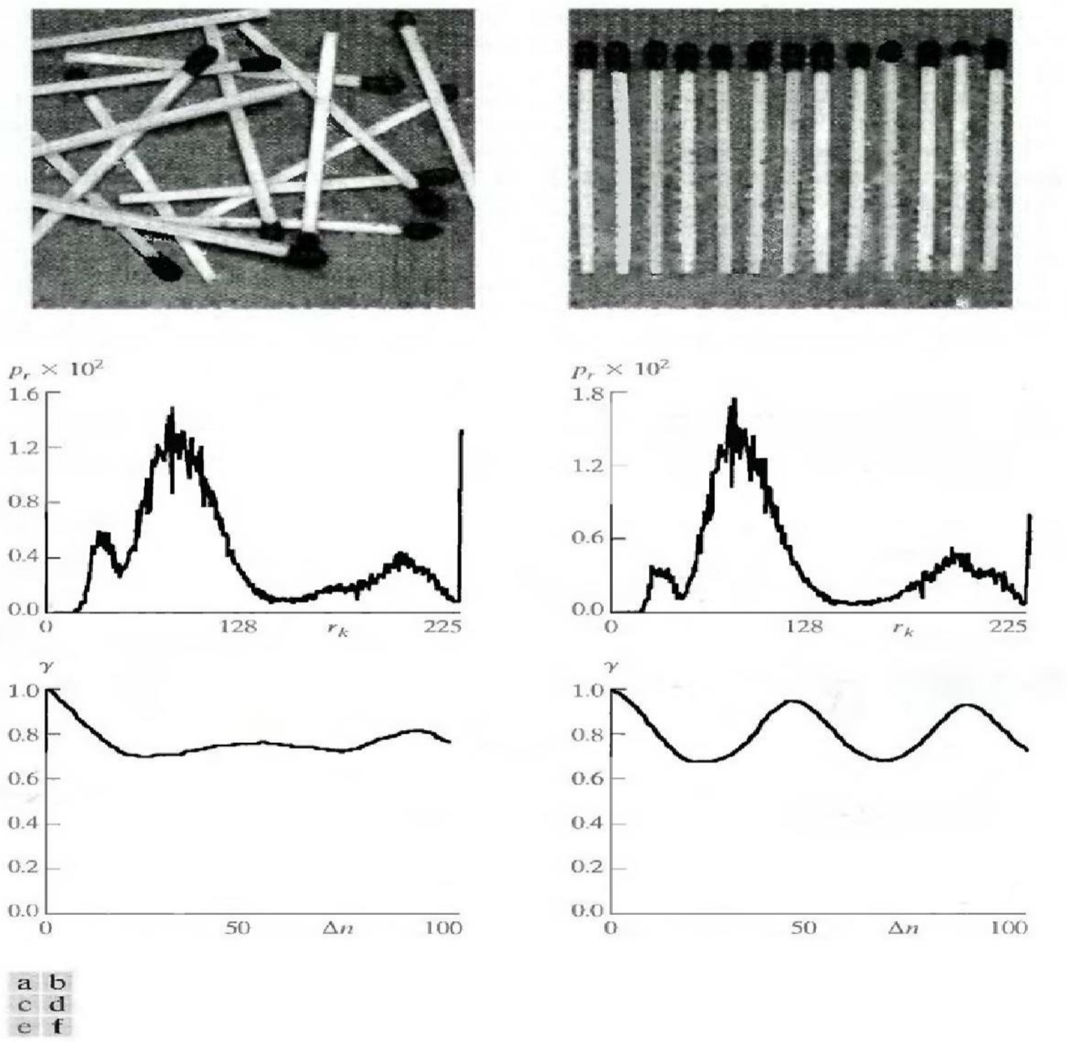


Fig. 5.1 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

Figures 5.1(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 n. Of course, n 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 n 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 psychovisual 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 psychovisual 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.
fidelity criterion.The removal of psychovisual redundant data results in a loss of real or quantitative visual information. Because information of interest may be lost, a repeatable or reproducible means of quantifying the nature and extent of information loss is highly desirable. Two general classes of criteria are used as the basis for such an assessment:
Objective fidelity criteria and

SUBJECTIVE FIDELITY CRITERIA:

When the level of information loss can be expressed as a function of the original or input image and the compressed and subsequently decompressed output image, it is said to be based on an objective fidelity criterion. A good example is the root-mean-square (rms) error between an input and output image. Let f(x, y) represent an input image and let f(x, y) denote an estimate or approximation of f(x, y) that results from compressing and subsequently decompressing the input.

For any value of x and y, the error e(x, y) between f (x, y) and f^ (x, y) can be defined as

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

so that the total error between the two images is

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x, y) - f(x, y)]$$

where the images are of size M X N. The root-mean-square error, erms, between f(x, y) and f^(x,y) then is the square root of the squared error averaged over the M X N array, or

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

A closely related objective fidelity criterion is the mean-square signal-to-noise ratio of the compressed-decompressed image. If f^ (x, y) is considered to be the sum of the original image f(x,y) and a noise signal e(x, y), the mean-square signal-to-noise ratio of the output image, denoted

$$SNR_{\text{ms}} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x, y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x, y) - f(x, y)]^2}.$$

SNR$_{\text{rms}}$, is

The rms value of the signal-to-noise ratio, denoted SNR$_{\text{rms}}$, is obtained by taking the square root of Eq. above.

Although objective fidelity criteria offer a simple and convenient mechanism for evaluating information loss, most decompressed images ultimately are viewed by humans. Consequently, measuring image quality by the subjective evaluations of a human observer often is more appropriate. This can be accomplished by showing a "typical" decompressed image to an appropriate cross section of viewers and averaging their evaluations. The evaluations may be made using an absolute rating scale or by means of side-by-side comparisons of f(x, y) and f^(x, y).

IMAGE COMPRESSION MODELS:

Fig. 3.1 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.

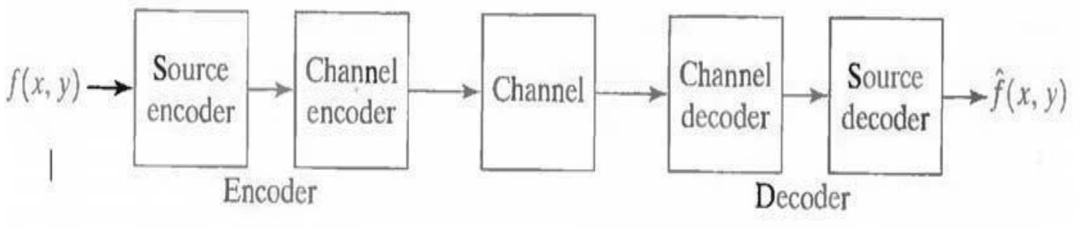


Fig.A general compression system model

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.

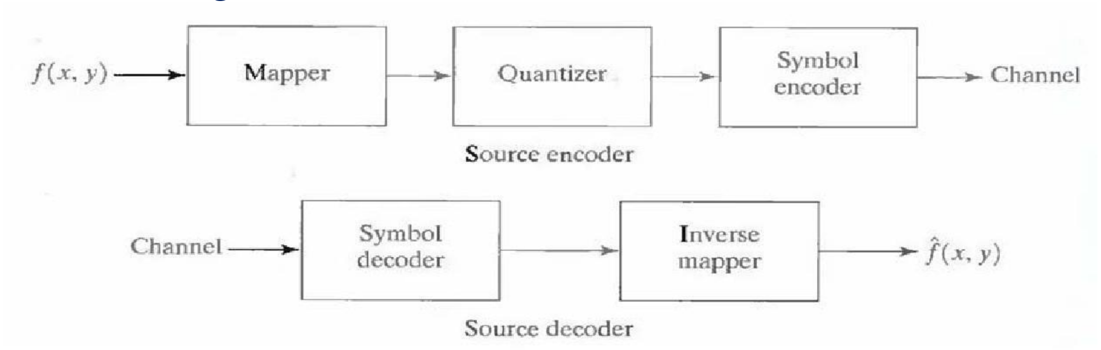


Fig.(a) Source encoder and (b) source decoder model

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. (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(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 inFig(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(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(b).

THE CHANNEL ENCODER AND DECODER:

The channel encoder and decoder play an important role in the overall encoding-decoding process when the channel of Fig. 3.1 is noisy or prone to error. They are designed to reduce the impact of channel noise by inserting a controlled form of redundancy into the source encoded data. As the output of the source encoder contains little redundancy, it would be highly sensitive to transmission noise without the addition of this "Controlled Redundancy."

One of the most useful channels encoding techniques was devised by R. W. Hamming (Hamming [1950]). It is based on appending enough bits to the data being encoded to ensure that some minimum number of bits must change between valid code words. Hamming showed, for example, that if 3 bits of redundancy are added to a 4-bit word, so that the distance between any two valid code words is 3, all single-bit errors can be detected and corrected. (By appending additional bits of redundancy,multiple-bit errors can be detected and corrected.) The 7-bit Hamming (7, 4) code word h_1, h_2, h_3…., h_6, h_7 associated with a 4-bit

$$\begin{array}{ll}
h_1 = b_3 \oplus b_2 \oplus b_0 & h_3 = b_3 \\
h_2 = b_3 \oplus b_1 \oplus b_0 & h_5 = b_2 \\
h_4 = b_2 \oplus b_1 \oplus b_0 & h_6 = b_1 \\
 & h_7 = b_0
\end{array}$$

binary number b_3,b_2,b_1,b_0 is

where denotes the exclusive OR operation. Note that bits h_1, h_2, and h_4 are even- parity bits forthe bit fields b_3 b_2 b_0, $b_3b_1b_0$, and $b_2b_1b_0$, respectively. (Recall that a string of binary bits has even parity if the number of bits with a value of 1 is even.) To decode a Hamming encoded result, the channel decoder must check the encoded value for odd parity over the bit fields in which even parity was previously established. A single-bit error is indicated by a nonzero parity word c4c2c1, where

$$c_1 = h_1 \oplus h_3 \oplus h_5 \oplus h_7$$
$$c_2 = h_2 \oplus h_3 \oplus h_6 \oplus h_7$$
$$c_4 = h_4 \oplus h_5 \oplus h_6 \oplus h_7.$$

If a nonzero value is found, the decoder simply complements the code word bit position indicated by the parity word. The decoded binary value is then extracted from the corrected code word as$h_3h_5h_6h_7$.

VARIABLE-LENGTH CODING:

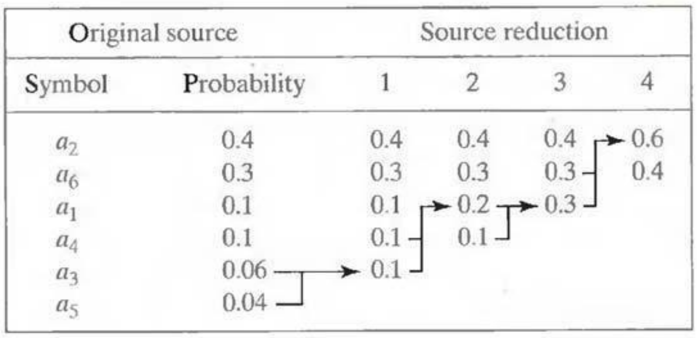
The simplest approach to error-free image compression is to reduce only coding redundancy. Coding redundancy normally is present in any natural binary encoding of the gray levels in an image. It can be eliminated by coding the gray levels. To do so requires construction of a variable- length code that assigns the shortest possible code words to the most probable gray levels. Here, we examine several optimal and near optimal techniques for constructing such a code. These techniques are formulated in the language of information theory. In practice, the source symbols may be either the gray levels of an image or the output of a gray-level mapping operation (pixel differences, run lengths, and so on).

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



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				

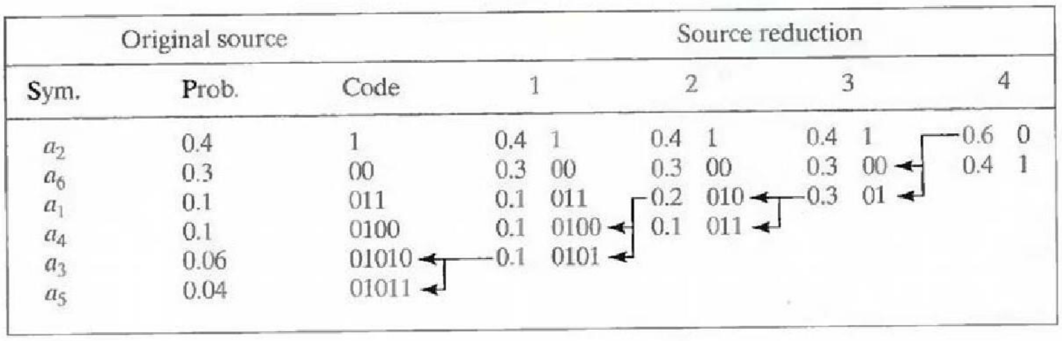
symbols, and a 0 and 1 are arbitrarily appended to each to distinguish them from each other.

Fig. Huffman source reductions.

Fig. Huffman code assignment procedure.

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.The average length of this code is

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



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								

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. 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 a_3. The next valid code is 011, which corresponds to symbol a_1. Continuing in this manner reveals the completely decoded message to be $a_3a_1a_2a_2a_6$.

ARITHMETIC CODING:
Unlike the variable-length codes described previously, arithmetic coding generates non-block 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.

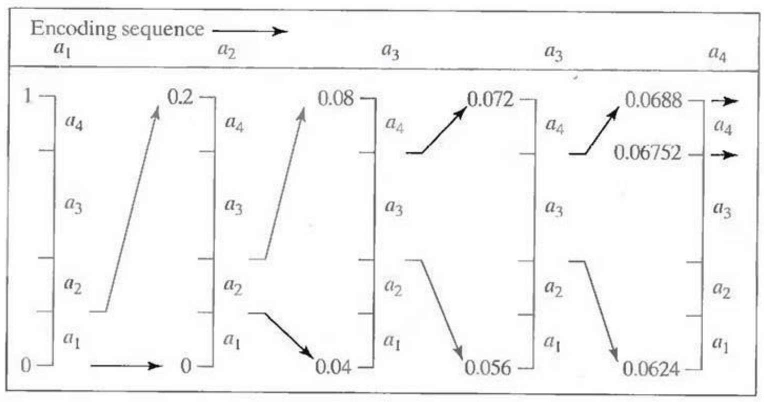
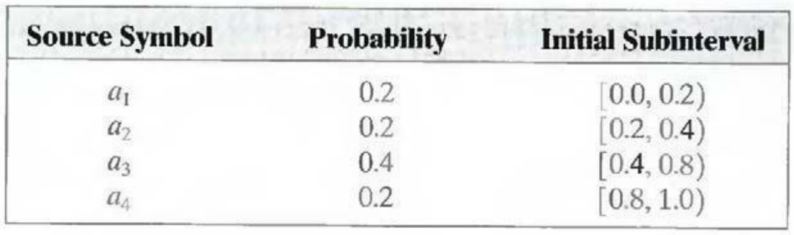


Fig.Arithmetic Coding Procedure

Figure illustrates the basic arithmetic coding process. Here, a five-symbol sequence ormessage, $a_1a_2a_3a_4$, 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 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. [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: Arithmetic coding example

In this manner, symbol a_2 narrows the subinterval to [0.04, 0.08), a_3 further narrows it to [0.056, 0.072), and so on. The final message symbol, which must be reserved as a special end-of-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. 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.

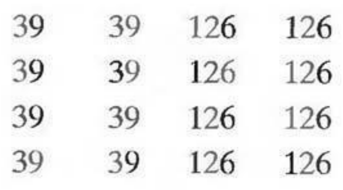
In practice, two factors cause coding performance to fall short of the bound: (1) the addition of the end-of-message indicator that is needed to separate one message from an-other; and (2) the use of finite precision arithmetic. Practical implementations of arithmetic coding address the latter problem 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.

LZW CODING:

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, gray-level 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 thedictionary is an important system parameter. If it is too small, the detection of matching gray-level

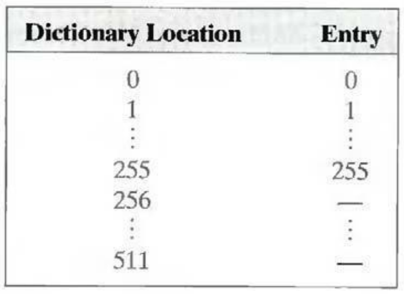
sequences will be less likely; if it is too large, the size of the code words will adversely affect compression performance.
Consider the following 4 x 4, 8-bit image of a vertical edge:



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

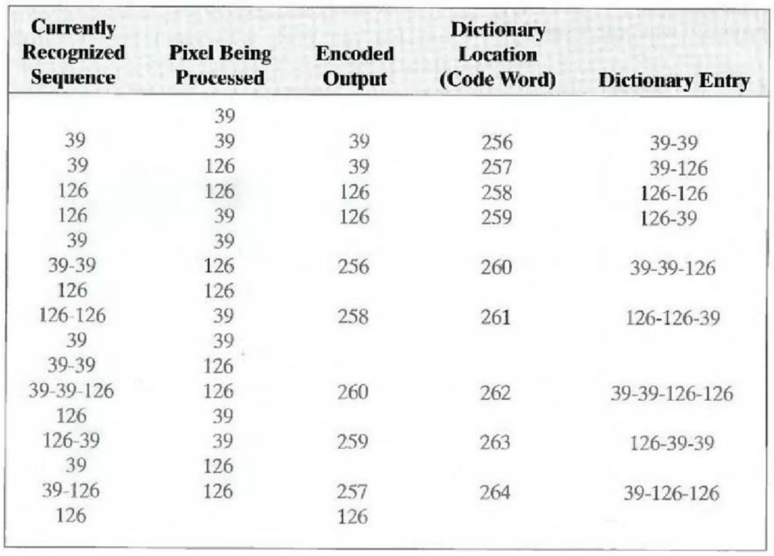
Table:Details the steps involved in coding its 16 pixels.

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: 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 isto 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.

BIT-PLANE CODING:

An 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.

BIT-PLANE DECOMPOSITION:

The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0.$$

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. The zeroth-orderbit plane is generated by collecting the a0 bits of each pixel, while the (m - 1) st -order bit plane contains the a_{m-1}, bits or coefficients. In general, each bit plane is numbered from 0 to m-1 and is constructed by setting its pixels equal to the values of the appropriate bits or polynomial coefficients from each pixel in the original image.

The inherent disadvantage of this approach is that small changes in gray level can have a significant impact on the complexity of the bit planes. If a pixel of intensity 127 (01111111) is adjacent to a pixel of intensity 128 (10000000), for instance, every bit plane will contain a corresponding 0 to 1 (or 1 to 0) transition. For example, as the most significant bits of the two binary codes for 127 and 128 are different, bit plane 7 will contain a zero-valued pixel next to a pixel of value 1, creating a 0 to 1 (or 1 to 0) transition at that point.

An alternative decomposition approach (which reduces the effect of small gray-level variations) is to first represent the image by an m-bit Gray code. The m-bit Gray code $g_{m-1} \ldots g_2 g_1 g_0$ that corresponds to the polynomial in Eq. above can be computed from

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2$$
$$g_{m-1} = a_{m-1}.$$

Here, denotes the exclusive OR operation. This code has the unique property that successive code words differ in only one-bit position. Thus, small changes in gray level are less likely to affect all m bit planes. For instance, when gray levels 127 and 128 are adjacent, only the 7th bit plane will contain a 0 to 1 transition, because the Gray codes that correspond to 127 and 128 are 11000000 and 01000000, respectively.

LOSSLESS PREDICTIVE CODING:

The error-free compression approach 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 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 eachsuccessive pixel of the input image, denoted fn, is introduced to the encoder, the predictor generates the anticipated value of that pixel based on some number of past inputs. The output ofthe predictor is then rounded to the nearest integer, denoted f^n and used to form the difference or prediction error which is coded using a variable-length code (by the symbol encoder) to generate the next element of the compressed data stream.

$$e_n = f_n - \hat{f}_n,$$

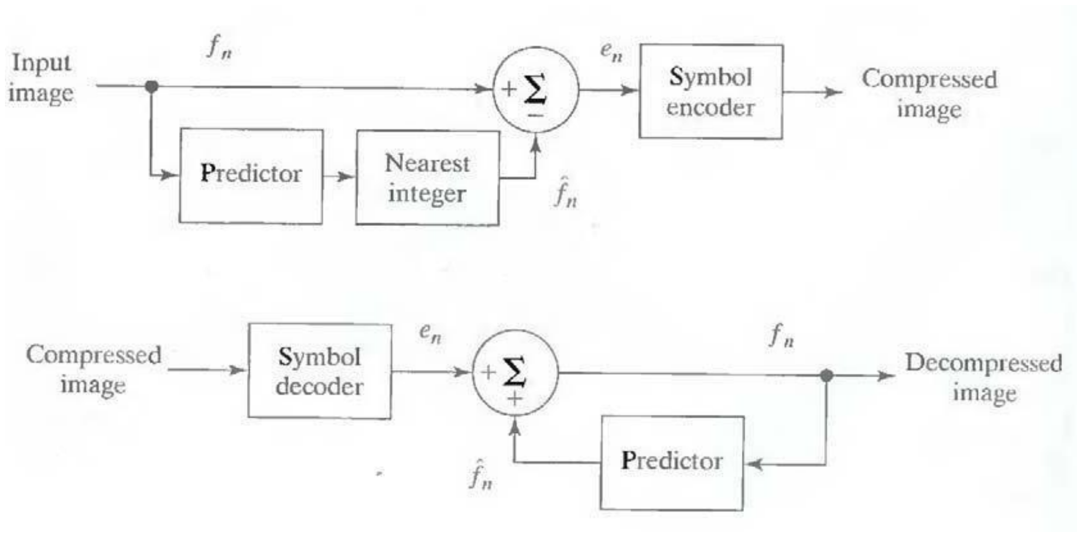


Fig. A lossless predictive coding model: (a) encoder; (b) decoder

The decoder of Fig. 8.1 (b) reconstructs en from the received variable-length code words and performs the inverse operation

$$f_n = e_n + \hat{f}_n.$$

Various local, global, and adaptive methods can be used to generate f^n. In most cases, however, the prediction is formed by a linear combination of m previous pixels. That is,

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

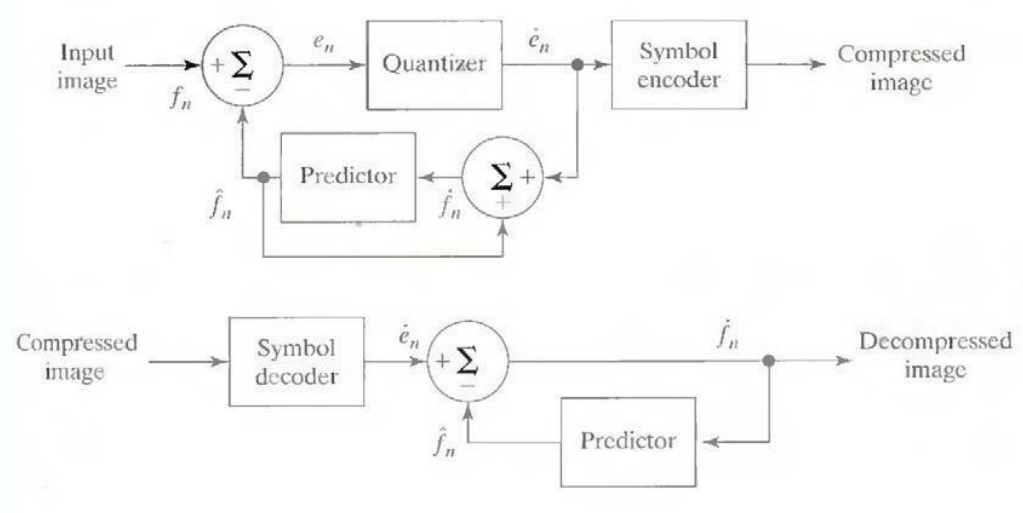
where m is the order of the linear predictor, round is a function used to denote the rounding ornearest integer operation, and the α_i, for i = 1,2,..., m are prediction coefficients. In raster scan applications, the subscript n indexes the predictor outputs in accordance with their time of occurrence. That is, fn, f^n and en in Eqns. above could be replaced with the more explicit notation f (t), f^(t), and e (t), where t represents time. In other cases, n is used as an index on the spatial coordinates and/or frame number (in a time sequence of images) of an image. In 1-D linear predictive coding, for example, Eq. above can be written as

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

where each subscripted variable is now expressed explicitly as a function of spatial coordinates x and y. The Eq. indicates that the 1-D linear prediction f(x, y) is a function of the previous pixels on the current line alone. In 2-D predictive coding, the prediction is a function of the previous pixels in a left-to-right, top-to-bottom scan of an image. In the 3-D case, it is based on these pixels and the previous pixels of preceding frames. Equation above cannot be evaluated for the first m pixels of each line, so these pixels must be coded by using other means (such as a Huffman code) and considered as an overhead of the predictive coding process. A similar comment applies to the higher-dimensional cases.

LOSSY PREDICTIVE CODING:

In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig. shows, the quantizer, which 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.



It mapsthe prediction error into a limited range of outputs, denoted e^n which establish the amount of compression and distortion associated with lossy predictive coding.

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

In order to accommodate the insertion of the quantization step, the error-free encoder of figure must be altered so that the predictions generated by the encoder and decoder are equivalent. As Fig.9 (a) shows, this is accomplished by placing the lossy encoder's predictor within feedback loop, where its input, denoted f˙n, is generated as a function of past predictions and the corresponding quantized errors. That is,

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

This closed loop configuration prevents error buildup at the decoder's output. Note from Fig. (b) that the output of the decoder also is given by the above Eqn.

OPTIMAL PREDICTORS:

The optimal predictor used in most predictive coding applications minimizes the encoder's mean- square prediction error

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

subject to the constraint that

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

and

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

That is, the optimization criterion is chosen to minimize the mean-square prediction error, thequantization error is assumed to be negligible (e˙n ≈ en), and the prediction is constrained to a linear combination of m previous pixels.1 These restrictions are not essential, but they simplify the analysis considerably and, at the same time, decrease the computational complexity of the predictor. The resulting predictive coding approach is referred to as differential pulse code modulation (**DPCM**)

TRANSFORM CODING:

All the predictive coding techniques operate directly on the pixels of an

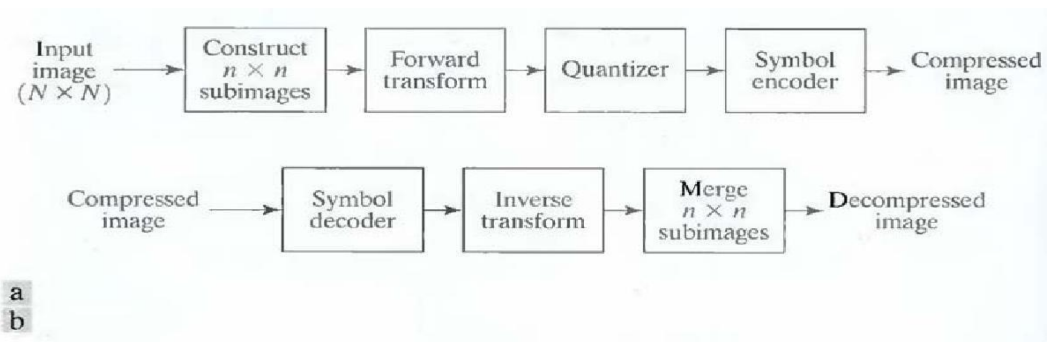


image and thus are spatial domain methods. In this coding, we consider compression techniques that are based on modifying the transform of an image. In transform coding, a reversible, linear transform (such as the

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

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.

Figure 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: sub image decomposition, transformation, quantization, and coding. An N X N input image first is subdivided into sub images of size n X n, which are thentransformed to generate (N/n) 2 sub images transform arrays, each of size n X n.

The goal of thetransformation process is to decorrelate the pixels of each sub image, or to pack as much information as possible into the smallest number of transform coefficients. 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 sub image 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 toDigital Image Processinglocal image content, called adaptive transform coding, or fixed for all sub images, called nonadaptive transform coding.

WAVELET CODING:

The wavelet coding is based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. If the transform's basis functions—in this case wavelets—pack most of the important visual information into a small number of coefficients, the remaining coefficients can be quantized coarsely or truncated to zero with little image distortion.

Figure shows a typical wavelet coding system. To encodea 2^J X 2image, an analyzing wavelet, Ψ, and minimum decomposition level, J - P, are selected and used to compute the image's discrete wavelet transform. If the wavelet has a complimentary scaling function φ, the fast wavelet transform can be used. In either case, the computed transform converts a large portion of the original image to horizontal, vertical, and diagonal decomposition coefficients with zero mean and Laplacian-like distributions.

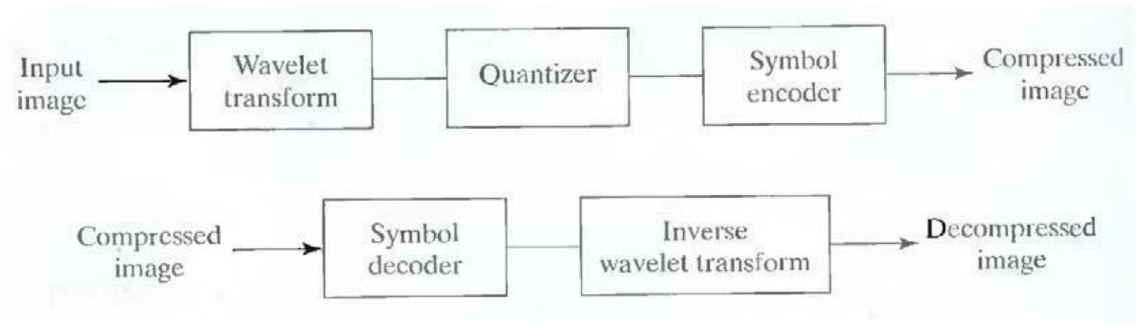


Fig. A wavelet coding system: (a) encoder; (b) decoder.

Since many of the computed coefficients carry little visual information, they can be quantized and coded to minimize inter- coefficient and coding redundancy. Moreover, the quantization can be adapted to exploit any positional correlation across the P decomposition levels. One or more of the lossless coding methods, including run-length, Huffman, arithmetic, and bit-plane coding, can be incorporated into the final symbol coding step. Decoding is accomplished by inverting the encoding operations—with the exception of quantization, which cannot be reversed exactly.

The principal difference between the wavelet-based system and the transform coding system is the omission of the transform coder's sub image processing stages.Because wavelet transforms are both computationally efficient and inherently local (i.e., their basis functions are limited in duration), subdivision of the original image is unnecessary.

Unit V
IMAGE REPRESENTATION AND RECOGNITION

LEARNING OBJECTIVES:

Image representation is the process of generating descriptions from the visual contents of an image. After reading this unit, the reader should be familiar with the following concepts:

1. Boundary representation
2. Chain Code
3. Descriptors
4. Pattern Recognition

BOUNDARY REPRESENTATION:

Models are a more explicit representation than CSG.The object is represented by a complicated data structure giving information about each of the object's faces, edges and vertices and how they are joined together.

Appears to be a more natural representation for Vision since surface information is readily available.The description of the object can be into two parts:

Topology: Itrecords the connectivity of the faces, edges and vertices by means of pointers in the data structure.

Geometry: Itdescribes the exact shape and position of each of the edges, faces and vertices.The geometry of a vertex is just its position in space as given by its (x,y,z) coordinates.Edges may be straight lines, circular arcs, etc.A face is represented by some description of its surface (algebraic or parametric forms used).

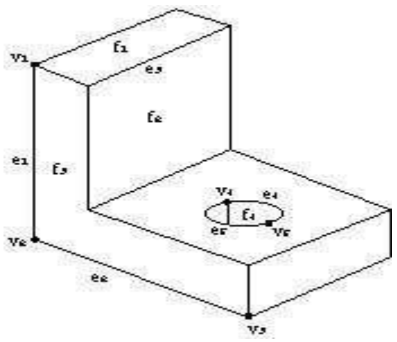


Fig. Faces, edges and vertices

CHAIN CODE:

A chain code is a lossless compression algorithm for monochrome images. The basic principle of chain codes is to separately encode each connected component, or "blot", in the image. For each such region, a point on the boundary is selected and its coordinates are transmitted. The encoder then moves along the boundary of the image and, at each step, transmits a symbol representing the direction of this movement. This continues until the encoder returns to the starting position, at which point the blot has been completely described, and encoding continues with the next blot in the image.

This encoding method is particularly effective for images consisting of a reasonable number of large connected components.Some popular chain codes include the Freeman Chain Code of Eight Directions (FCCE), Vertex Chain Code (VCC), Three Orthogonal symbol chain code (3OT) and Directional Freeman Chain Code of Eight Directions (DFCCE).

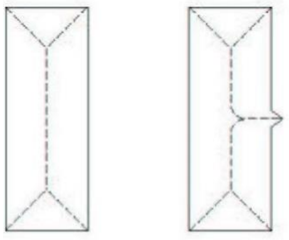
A related blot encoding method is crack code Algorithms exist to convert between chain code, crack code, and run-length encoding.

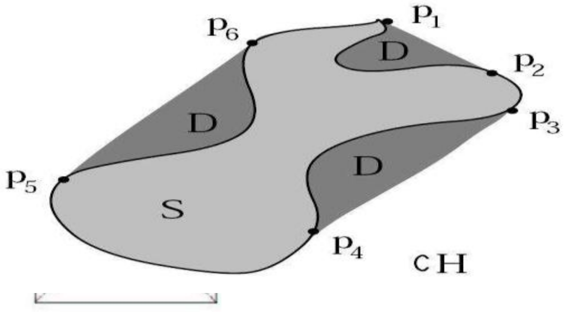
1. Let i=3
2. Mark the points in the object that have furthest distance from each other with p and p2
3. Connect the points in the order they are numbered with lines
4. For each segment in the polygon, find the point on the perimeter between the point that have furthest distance to the polygonal line-segment. If this distance Is larger than a threshold, mark the point with a label pi
5. Renumber the points so that they are consecutive
6. Increase i
7. If no points have been added break, otherwise go to 3

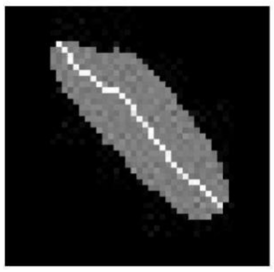
The convex hull H of a set S is defined as the smallest convex set that contains S. We define the set

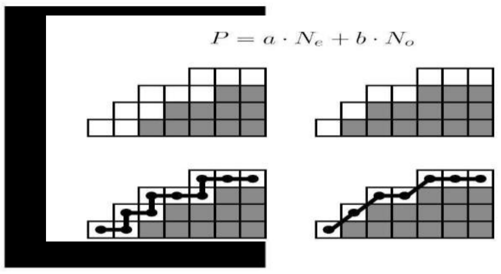
$$D= H \setminus S$$

The points that have neighbors from the sets D, H and CH is called p. These points are used for representation of the object. To limit the number of points found it is possible to smooth the edge of the objectWe define an object’s skeleton or medial axel as the points in the object that have several nearest neighbors on the edge of the object









To find the skeleton by using the definition is very costly, since it involves calculating the distance between all points in the object to all points on the perimeter.
Usually some iterative method is used to remove perimeter pixels until the skeleton alone remains.
In order to achieve this the following rules must be adhered to:

1. Erosion of object must be kept to a minimum
2. Connected components must not be broken
3. End points must not be removed

The purpose with Description is to Quantify a representation of an object. This implies that we instead of talking about the areas in the image we can talk about their properties such as length, curviness and so on.

One of the simplest descriptions is the length P of the perimeter of an object.
The obvious measure of perimeter – length is the number of edge pixels. That is, pixel that do belong to the object, but have a neighbor that belong to the object, but have a neighbor that belong to the background.
A more precise measure is to assume that each pixel center is a corner in a polygon.
The length of the perimeter is given by

$$P = aN_e + bN_o$$

Intuitively we would like to set a=1 and b=
It is however possible to show that the length of the perimeter will be slightly overestimated with these values.
The optimal weights for a and b (in least square sense) will depend on the curviness of the perimeter.
If the perimeter is straight line (!?) a=0.948 and b=1.343 will be optimal.
If it is assumed that the direction of two consecutive line-segments is uncorrelated a=0.948 and b=1.343 will be optimal.
The diameter of an object is defined as

Where D is a metric
The line that passes through the points Pi and Pjthat defines the diameter is called the main axis of the $\sqrt{2}$ object.
The curviness of perimeter can be obtained by calculating the angle between two consecutive line-segments of the polygonal approximation
The curvature at point Pjof the curve

$\{p_i\}_{i=1}^{N}$ is also given by

$$c = ||p_{j-1} - 2p_j + p_{j+1}||^2$$

where $p_i \in \mathbb{R}^2$

$$Diam(B) = \max_{i,j}[D(p_i, p_j)]$$

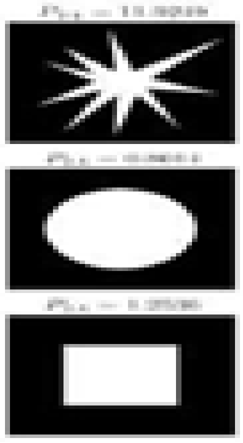
We can approximate the area of an object with the number of pixels belonging to the object.
More accurate measures are, however obtained by using a polygonal approximation. The area of a polygon segment (with one corner in the origin) is given by the area of the entire polygon is then

$$A = \frac{1}{2}\sum_{i=1}^{N_b} x_i y_{i+1} - x_{i+1} y_i$$

A circle of radius r has the area A=пr2 and the length of the perimeter is P=2 пr. So, by defining the quotient

$$P_{2A} = \frac{P^2}{4\pi A}$$

We have a measurement that is 1, if the object is a circle. The larger the measurement the less circle-like is the object

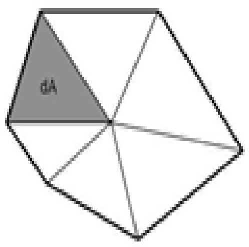


The measure of the shape of the object can be obtained according to:

1. Calculate the chain code for the object
2. Calculate the difference code for the chain code
3. Rotate the code so that it is minimal
4. This number is called the shape number of the object
5. The length of the number is called the order of the shape

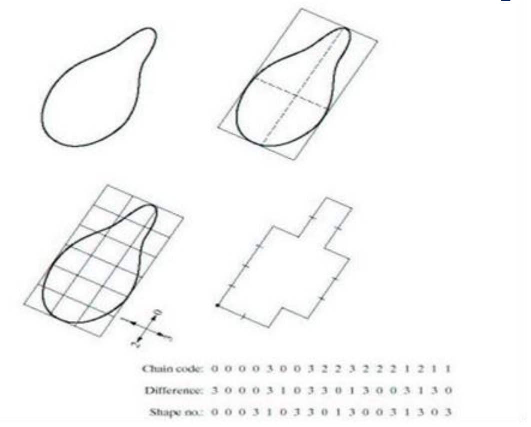
The quotient between the length of the main axis and the largest width orthogonal to the main axis is called the eccentricity of the object. Given a order of the shape we can find the rectangle that best approximates the eccentricity of an object. The rectangle is rotated so that its main axis coincides with one of the objects

$$dA = x_1y_1 - \frac{1}{2}x_1y_1 - \frac{1}{2}x_2y_2 + \frac{1}{2}(x_2 - x_1)(y_1 - y_2)$$
$$= \frac{1}{2}(x_1y_2 - x_2y_1)$$



The rectangle defines a resampling grid.

The shape number is then calculated on the re-sampled object.



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

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

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

Suppose we have a perimeter of length N made out of coordinates (xi, yi)
We then define the functions x(k) = x_k and y(k) = y_k and

$$s(k) = x(k) + jy(k)$$

For k= 0, 1, ……., (N-1)
The discrete Fourier transform of s(k) is

$$a(u) = \frac{1}{N}\sum_{k=0}^{N-1} s(k)e^{-j2\pi uk/N}$$

The complex coefficients a(u) is called Fourier Descriptors of the object
By using M<=N coeffiencts in the reconstruction of the object an approximation is obtained

$$\hat{s}(k) = \sum_{u=0}^{M-1} a(u)e^{j2\pi uk/N}$$

Observe that it is still N points in the countour but that we use M frequencies to construct the object.

TEXTURES:

"Texture" is an ambiguous word and in the context of texture synthesis may have one of the following meanings. In common speech, "texture" used as a synonym for "surface structure". Texture has been described by five different properties in the psychology of perception: coarseness, contrast, directionality, line-likeness and roughness.

In 3D computer graphics, a texture is a digital image applied to the surface of a three-dimensional model by texture mapping to give the model a more realistic appearance. Often, the image is a photograph of a "real" texture, such as wood grain.In image processing, every digital image composed of repeated elements is called a "texture." Texture can be arranged along a spectrum going from stochastic to regular:

Stochastic Textures: Texture images of stochastic textures look like noise: colour dots that are randomly scattered over the image, barely specified by the attributes minimum and maximum brightness and average colour. Many textures look like stochastic textures when viewed from a distance. An example of a stochastic texture is roughcast.
Structured textures. These textures look like somewhat regular patterns. An example of a structured texture is a stonewall or a floor tiled with paving stones.

Visual Descriptors: In computer vision, visual descriptors or image descriptors are descriptions of the visual features of the contents in images, videos, algorithms, or applications that produce such descriptions. They describe elementary characteristics such as the shape, the color, the texture or the motion, among others.

Introduction: As a result of the new communication technologies and the massive use of Internet in our society, the amount of audio-visual information available in digital format is increasing considerably. Therefore, it has been necessary to design some systems that allow us to describe the content of several types of multimedia information in order to search and classify them.

The audio-visual descriptors are in charge of the contents description. These descriptors have a good knowledge of the objects and events found in a video, image or audio and they allow the quick and efficient searches of the audio-visual content.This system can be compared to the search engines for textual contents. Although it is certain, that it is relatively easy to find text with a computer, is much more difficult to find concrete audio and video parts.

For instance, imagine somebody searching a scene ofa happy person. The happiness is a feeling and it is not evident its shape, color and texture description in images.The description of the audio-visual content is not a superficial task and it is essential for the effective use of this type of archives. The standardization system that deals with audio-visual descriptors is the MPEG-7 (Motion Picture Expert Group - 7).

Types of Visual Descriptors:
Descriptors are the first step to find out the connection between pixels contained in a digital image and what humans recall after having

observed an image or a group of images after some minutes.
Visual descriptors are divided in two main groups:
General Information Descriptors: They contain low level descriptors which give a description about color, shape, regions, textures and motion.
Specific Domain Information Descriptors: They give information about objects and events in the scene. A concrete example would be face recognition.General information descriptors consist of a set of descriptors that covers different basic and elementary features like: color, texture, shape, motion, location and others. This description is automatically generated by means of signal processing.

COLOR: The most basic quality of visual content. Five tools are defined to describe color. The three first tools represent the color distribution and the last ones describe the color relation between sequences or group of images:
1. Dominant Color Descriptor (DCD)
2. Scalable Color Descriptor (SCD)
3. Color Structure Descriptor (CSD)
4. Color Layout Descriptor (CLD)

Group of Frame (GoF) or Group-of-Pictures (GoP):

TEXTURE: Also, an important quality in order to describe an image. The texture descriptors characterize image textures or regions. They observe the region homogeneity and the histograms of these region borders. The set of descriptors is formed by:
1. Homogeneous Texture Descriptor (HTD)
2. Texture Browsing Descriptor (TBD)
3. Edge Histogram Descriptor (EHD)

SHAPE: Contains important semantic information due to human ‘s ability to recognize objects through their shape. However, this information can only be extracted by means of a segmentation similar to the one that the human visual system implements. Nowadays, such a segmentation system is not available yet, however there exists a serial of algorithms which are considered to be a goodapproximation. These descriptors describe regions, contours and shapes for 2D images and for 3D volumes. The shape descriptors are the following ones:
1. Region-based Shape Descriptor (RSD)
2. Contour-based Shape Descriptor (CSD)
3. 3-D Shape Descriptor (3-D SD)

MOTION: Defined by four different descriptors which describe motion in video sequence. Motion is related to the objects motion in the sequence and to the camera motion. This last information is provided by the capture device, whereas the rest is implemented by means of image processing. The descriptor set is the following one:

1. Motion Activity Descriptor (MAD)
2. Camera Motion Descriptor (CMD)
3. Motion Trajectory Descriptor (MTD)
4. Warping and Parametric Motion Descriptor (WMD and PMD)

LOCATION: Elements location in the image is used to describe elements in the spatial domain. In addition, elements can also be located in the temporal domain:

1. Region Locator Descriptor (RLD)
2. Spatio Temporal Locator Descriptor (STLD)

Specific Domain Information Descriptors: These descriptors, which give information about objects and events in the scene, are not easily extractable, even more when the extraction is to be automatically done. Nevertheless, they can be manually processed.
As mentioned before, face recognition is a concrete example of an application that tries to automatically obtain this information.

Descriptors Applications:
Among all applications, the most important ones are:

- Multimedia documents search engines and classifiers.
- **Digital Library:**visual descriptors allow a very detailed and concrete search of any video or image by means of different search parameters. For instance, the search of films where a known actor appears, the search of videos containing the Everest mountain, etc.
- Personalized electronic news service.
- Possibility of an automatic connection to a TV channel broadcasting a soccer match, for example, whenever a player approaches the goal area.
- Control and filtering of concrete audio-visual contents, like violent or pornographic material. Also, authorization for some multimedia contents.

What is Pattern Recognition?
Pattern recognition is the process of recognizing patterns by using

machine learning algorithm. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation. One of the important aspects of the pattern recognition is its application potential.

Examples: Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis. In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves classification and cluster of patterns.In classification, an appropriate class label is assigned to a pattern based on an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.Clustering generated a partition of the data which helps decision making, the specific decision-making activity of interest to us. Clustering is used in an unsupervised learning.Features may be represented as continuous, discrete or discrete binary variables. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristics of the object.

Example: consider our face then eyes, ears, nose etc are features of the face.
A set of features that are taken together, forms the features vector. Example: In the above example of face, if all the features (eyes, ears, nose etc) taken together then the sequence is feature vector ([eyes, ears, nose]). Feature vector is the sequence of a features represented as a d-dimensional column vector. In case of speech, MFCC (Mel frequency Cepstral Coefficient) is the spectral features of the speech. Sequence of first 13 features forms a feature vector.

Pattern recognition possesses the following features:
- Pattern recognition system should recognize familiar pattern quickly and accurate
- Recognize and classify unfamiliar objects
- Accurately recognize shapes and objects from different angles
- Identify patterns and objects even when partly hidden
- Recognize patterns quickly with ease, and with automaticity

PatternRecognition System: Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms.

- In Pattern Recognition, pattern is comprising of the following two fundamental things:
- Collection of observations
- The concept behind the observation

FeatureVector: The collection of observations is also known as a feature vector. A feature is a distinctive characteristic of a good or service that sets it apart from similar items. Feature vector is the combination of n features in n-dimensional column vector.The different classes may have different features values but the same class always has the same features value

Classifier and Decision Boundaries:

In a statistical-classification problem, a decision boundary is a hypersurface that partitions the underlying vector space into two sets. A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous.Classifier is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points.Classifier is used to partition the feature space into class-labeled decision regions.

While Decision Boundaries are the borders between decisionregions.

Components in Pattern Recognition System:

Pattern recognition systems can be partitioned into components. There are five typical components for various pattern recognition systems. These are as following:

Sensor: A sensor is a device used to measure a property, such as pressure, position, temperature, or acceleration, and respond with feedback.

A Preprocessing Mechanism: Segmentation is used and it is the process of partitioning a data into multiple segments. It can also be defined as the technique of dividing or partitioning a data into parts called segments.

A Feature Extraction Mechanism: Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. It can be manual or automated.

A Description Algorithm: Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform “most likely” matching of the inputs, taking into account their statistical variation

A Training Set: Training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs.

MULTIPLE CHOICE QUESTIONS

1. To convert a continuous sensed data into Digital form, which of the following is required?
   a) Sampling
   b) Quantization
   c) Both Sampling and Quantization
   d) Neither Sampling nor Quantization

2. For a continuous image f(x, y), Quantization is defined as
   a) Digitizing the coordinate values
   b) Digitizing the amplitude values
   c) All of the mentioned
   d) None of the mentioned

3. What is pixel?
   a) Pixel is the elements of a digital image
   b) Pixel is the elements of an analog image
   c) Pixel is the cluster of a digital image
   d) Pixel is the cluster of an analog image

4. Which is a color attribute that describes a pure color?
   a) Saturation
   b) Hue
   c) Brightness
   d) Intensity

5. Which is a color attribute that describes a pure color?
   a) Saturation
   b) Hue
   c) Brightness
   d) Intensity

6. A typical size comparable in quality to monochromatic TV image is of size
   a) 256 X 256
   b) 512 X 512
   c) 1920 X 1080
   d) 1080 X 1080

7. The number of grey values is integer powers of:
   a) 4
   b) 2
   c) 8
   d) 1

8. The section of the real plane spanned by the coordinates of an image is called the ___________
   a) Spatial Domain
   b) Coordinate Axes
   c) Plane of Symmetry
   d) None of the Mentioned

9. The procedure done on a digital image to alter the values of its individual pixels is
   a) Neighborhood Operations
   b) Image Registration
   c) Geometric Spatial Transformation
   d) Single Pixel Operation

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

11. How many numbers of steps are involved in image processing?
   a) 10
   b) 9
   c) 11
   d) 12

12. After digitization process a digital image with M rows and N columns have to be positive and for the number, L, max gray levels i.e. an integer power of 2 for each pixel. Then, the number b, of bits required to store a digitized image is:
   a) b=M*N*k
   b) b=M*N*L
   c) b=M*L*k
   d) b=L*N*k

13. In digital image of M rows and N columns and L discrete gray levels, calculate the bits required to store a digitized image for M=N=32 and L=16.
   a) 16384
   b) 4096
   c) 8192
   d) 512

14. The quality of a digital image is well determined by ___________
   a) The number of samples
   b) The discrete gray levels
   c)All of the mentioned
   d) None of the mentioned

15. The most familiar single sensor used for Image Acquisition is
   a) Microdensitometer
   b) Photodiode
   c)CMOS
   d) None of the Mentioned

16. The difference is intensity between the highest and the lowest intensity levels in an image is ___________
   a) Noise
   b) Saturation
   c)Contrast
   d) Brightness

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

18. What is the general form of representation of power transformation?
a) s=crγ
b) c=srγ
c) s=rc
d) s=rcγ

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

a. s=clog10(1/r)
b. s=clog10(1+r)
c. s=clog10(1*r)
d. s=clog10(1-r)

20. Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range [0, L-1]?
a) s=L+1-r
b) s=L+1+r
c) s=L-1-r
d) s=L-1+r

21. What is the full form for PDF, a fundamental descriptor of random variables i.e. gray values in an image?
a) Pixel distribution function
b) Portable document format
c) Pel deriving function
d) Probability density function

22. What is the output of a smoothing, linear spatial filter?
a) Median of pixels
b) Maximum of pixels
c) Minimum of pixels
d) Average of pixels

23. Which of the following is the disadvantage of using smoothing filter?
a) Blur edges
b) Blur inner pixels
c) Remove sharp transitions
d) Sharp edges

24. Which of the following comes under the application of image blurring?
a) Object detection
b) Gross representation
c) Object motion
d) Image segmentation

25. Which of the following derivatives produce a double response at step changes in gray level?
a) First order derivative

b) Third order derivative
c) Second order derivative
d) First and second order derivatives

26. What is the name of process used to correct the power-law response phenomena?
a) Beta correction
b) Alpha correction
c) Gamma correction
d) Pie correction

27. Which of the following transformation function requires much information to be specified at the time of input?
a) Log transformation
b) Power transformation
c) Piece-wise transformation
d) Linear transformation

28. In which type of slicing, highlighting a specific range of gray levels in an image often is desired?
a) Gray-level slicing
b) Bit-plane slicing
c) Contrast stretching
d) Byte-level slicing

29. In general, which of the following assures of no ringing in the output?
a) Gaussian Lowpass Filter
b) Ideal Lowpass Filter
c) Butterworth Lowpass Filter
d) All of the mentioned

30. The lowpass filtering process can be applied in which of the following area(s)?
a) The field of machine perception, with application of character recognition
b) In field of printing and publishing industry
c) In field of processing satellite and aerial images
d) All of the mentioned

31. The function of filters in Image sharpening in frequency domain is to perform reverse operation of which of the following Lowpass filter?

a) Gaussian Lowpass filter
b) Butterworth Lowpass filter
c) Ideal Lowpass filter
d) None of the Mentioned

32. Which of the following is a second-order derivative operator?
a) Histogram
b) Laplacian
c) Gaussian
d) None of the mentioned

33.Dark characteristics in an image are better solved using ___________
a) Laplacian Transform
b) Gaussian Transform
c) Histogram Specification
d) Power-law Transformation

34. _____________ is used to detect diseases such as bone infection and tumors.
a) MRI Scan
b) PET Scan
c) Nuclear Whole-Body Scan
d) X-Ray

35. An alternate approach to median filtering is ______________
a) Use a mask
b) Gaussian filter
c) Sharpening
d) Laplacian filter

36.Sobel is better than prewet in image
a) Sharpening
b) Blurring
c) Smoothing
d) Contrast

37.If the inner region of the object is textured then approach, we use is
a) Discontinuity
b) Similarity
c) Extraction
d) Recognition

38.Gradient magnitude images are more useful in
a) Point Detection
b) Line Detection
c) Area Detection
d) Edge Detection

39.Diagonal lines are angles at
a) 0
b) 30
c) 45
d) 90

40.Transition between objects and background shows
a) Ramp Edges
b) Step Edges
c) Sharp Edges
d) Both A And B

41.For edge detection we use
a) First Derivative
b) Second Derivative
c) Third Derivative

42. Sobel gradient is not that good for detection of
a) Horizontal Lines
b) Vertical Lines
c) Diagonal Lines
d) Edges

43. Method in which images are input and attributes are output is called
a) Low Level Processes
b) High Level Processes
c) Mid-Level Processes
d) Edge Level Processes

44. Computation of derivatives in segmentation is also called
a) Spatial Filtering
b) Frequency Filtering
c) Low Pass Filtering
d) High Pass Filtering

45. Transition of intensity takes place between
a) Adjacent Pixels
b) Near Pixels
c) Edge Pixels
d) Line Pixels

46. Discontinuity approach of segmentation depends upon
a) Low Frequencies
b) Smooth Changes
c) Abrupt Changes
d) Contrast

47. Two regions are said to be adjacent if their union forms
a) Connected Set
b) Boundaries
c) Region
d) Image

47. Example of similarity approach in image segmentation is
a) Edge Based Segmentation
b) Boundary Based Segmentation
c) Region Based Segmentation
d) Both A And B

48. LOG stands for
a) Laplacian of Gaussian
b) Length of Gaussian
c) Laplacian of Gray Level
d) Length of Gray level

49.Double line effect is produced by
a) First Derivative
b) Second Derivative
c) Third Derivative
d) Both A And B

50.Intersection between zero intensity and extreme of second derivative is called
a) Discontinuity
b) Similarity

c) Continuity
d) Zero Crossing

51. DWT stands for
a) Discrete Wavelet Transform
b) Discrete Wavelet Transformation
c) Digital Wavelet Transform
d) Digital Wavelet Transformation

52. Discarding every sample is called
a) Up Sampling
b) Filtering
c) Down Sampling A
d) Blurring

53. High contrast images are considered as
a) Low Resolution
b) High Resolution
c) Intense
d) Blurred

54. In multiresolution processing * represents the
a) Complete Conjugate Operation
b) Complex Conjugate Operation
c) Complete Complex Operation
d) Complex Complex Operation

55.Representing image in more than one resolution is
a) Histogram
b) Image Pyramid
c) Local Histogram
d) Equalized Histogram

56.Subband of input image, showing dD(m,n) is called
a) Approximation
b) Vertical Detail
c) Horizontal Detail
d) Diagonal Detail

57. Which of the following measures are not used to describe a region?
a) Mean and median of grey values

b) Minimum and maximum of grey values
c) Number of pixels alone A
d) Number of pixels above and below mean

58. Which of the following techniques is based on the Fourier transform?
a) Structural
b) Spectral
c) Statistical
d) Topological

59. Which of the following of a boundary is defined as the line perpendicular to the major axis?
a) Equilateral axis
b) Equidistant axis
c) Minor axis A
d) Median axis

60. Which of the following techniques of boundary descriptions have the physical interpretation of boundary shape?
a) Fourier transform
b) Statistical moments
c) Laplace transform
d) Curvature

REFERENCES

1. Rafael C. Gonzales, Richard E. Woods, “Digital Image Processing”, Third Edition, Pearson Education, 2010.
2. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing Using MATLAB”, Third Edition Tata Mc Graw Hill Pvt. Ltd., 2011.
3. Anil Jain K. “Fundamentals of Digital Image Processing”, PHI Learning Pvt. Ltd., 2011.ACC.NO: B130746
4. Willliam K Pratt, “Digital Image Processing”, John Willey, 2002.
5. Malay K. Pakhira, “Digital Image Processing and Pattern Recognition”, First Edition, PHI Learning Pvt. Ltd., 2011.