

# LECTURE NOTES

## SUBJECT: DIGITAL IMAGE AND SPEECH PROCESSING

SUBJECT CODE: ECS-702, BRANCH: EL&TCE

### SYLLABUS

#### Module –I Digital Image

(12 hours)

1. Different stages of Image processing & Analysis Scheme. Components of Image Processing System, Multiprocessor Interconnections.
2. A Review of various Mathematical Transforms.
3. Image Formation: Geometric Model, Photometric Model.
4. Image Digitization : A review of Sampling and quantization processes. A digital image.

#### Module – II Image Processing

(12 Hours)

5. Image Enhancement: Contrast Intensification, Smoothing, Image sharpening.
6. Restoration : Minimum Mean - Square Error Restoration by Homomorphic Filtering.
7. Image Compression : Schematic diagram of Data Compression Procedure, Lossless compression - coding.
8. Multivalued Image Processing, Multispectral Image Processing, Processing of color images.

#### Module –III Digital Speech Processing

(8 Hours)

1. The Fundamentals of Digital Speech Processing.  
A Review of Discrete-Time Signal & Systems , the Z-transform, the DFT, Fundamental of Digital Filters, FIR system, IIR Systems.
2. Time -Domain Methods for Speech Processing.  
Time-Dependent Processing of speech, short-time energy and Average Magnitude, Short time Average Zero- Crossing Rate.
3. Digital Representation of speech Waveform  
Sampling speech signals, statistical model, Instantaneous quantization, Instantaneous companding, quantization for optimum SNR, Adaptive quantization, Feed-forward Feedback adaptations.

#### Module –IV Linear Predictive Coding of Speech

(8 Hours)

Block diagram of Simplified Model for Speech Production. Basic Principles of Linear Predictive Analysis- The Auto Correlation Method. The Prediction Error Signal. Digital Speech Processing for Man-Machine Communication by voice. Speaker Recognition Systems- Speaker verification and Speaker Identification Systems.

## **MODULE-1**

### **DIGITAL IMAGE**

#### **INTRODUCTION**

The digital image processing deals with developing a digital system that performs operations on a digital image.

An image is nothing more than a two dimensional signal. It is defined by the mathematical function  $f(x,y)$  where  $x$  and  $y$  are the two co-ordinates horizontally and vertically and the amplitude of  $f$  at any pair of coordinate  $(x, y)$  is called the intensity or gray level of the image at that point.

When  $x$ ,  $y$  and the amplitude values of  $f$  are all finite discrete quantities, we call the image a digital image. The field of image digital image processing refers to the processing of digital image by means of a digital computer.

A digital image is composed of a finite number of elements, each of which has a particular location and values of these elements are referred to as picture elements, image elements and pixels.

#### **Motivation and Perspective**

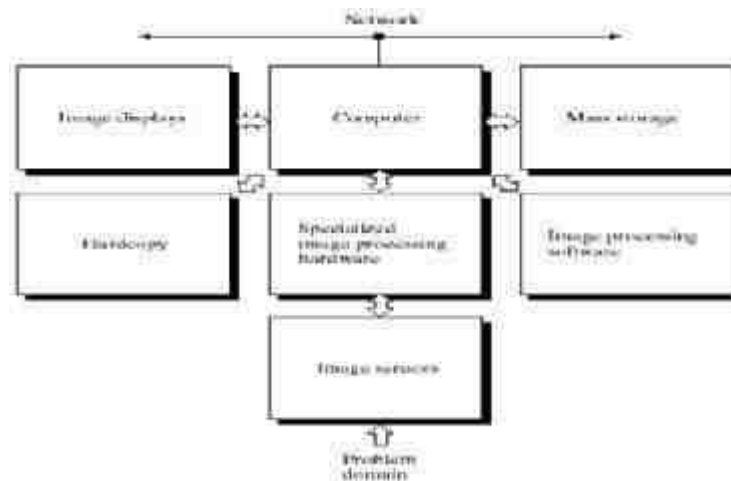
Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output. The most common example is Adobe Photoshop. It is one of the widely used applications for processing digital images.

#### **Applications**

Some of the major fields in which digital image processing is widely used are

1. Gamma Ray Imaging- Nuclear medicine and astronomical observations.
2. X-Ray imaging - X-rays of body.
3. Ultraviolet Band -Lithography, industrial inspection, microscopy, lasers.
4. Visual And Infrared Band - Remote sensing.
5. Microwave Band - Radar imaging.

## Components of Image Processing System



### i) Image Sensors

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

### ii) Specialize image processing hardware

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

### iii) Computer

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

### iv) Software

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

### v) Mass storage

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

- i) Short term storage for use during processing
- ii) On line storage for relatively fast retrieval
- iii) Archival storage such as magnetic tapes and disks

vi) Image displays

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

vii) Hardcopy devices

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

viii) Networking

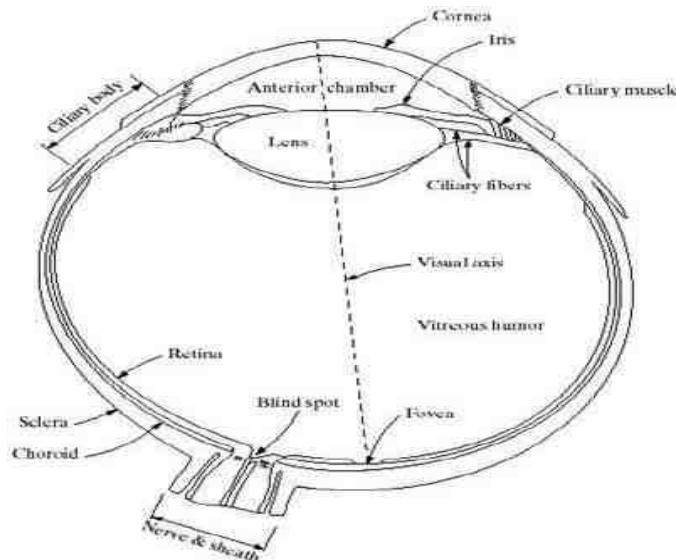
It is almost a default function in any computer system in use today because of the large amount of data inherent in image processing applications. The key consideration in image transmission bandwidth.

## Elements of Visual Perception

### Structure of the human Eye

The eye is nearly a sphere with average approximately 20 mm diameter. The eye is enclosed with three membranes

- a) The cornea and sclera: it is a tough, transparent tissue that covers the anterior surface of the eye. Rest of the optic globe is covered by the sclera
- b) The choroid: It contains a network of blood vessels that serve as the major source of nutrition to the eyes. It helps to reduce extraneous light entering in the eye  
It has two parts
  - (1) Iris Diaphragms- it contracts or expands to control the amount of light that enters the eyes.
  - (2) Ciliary body



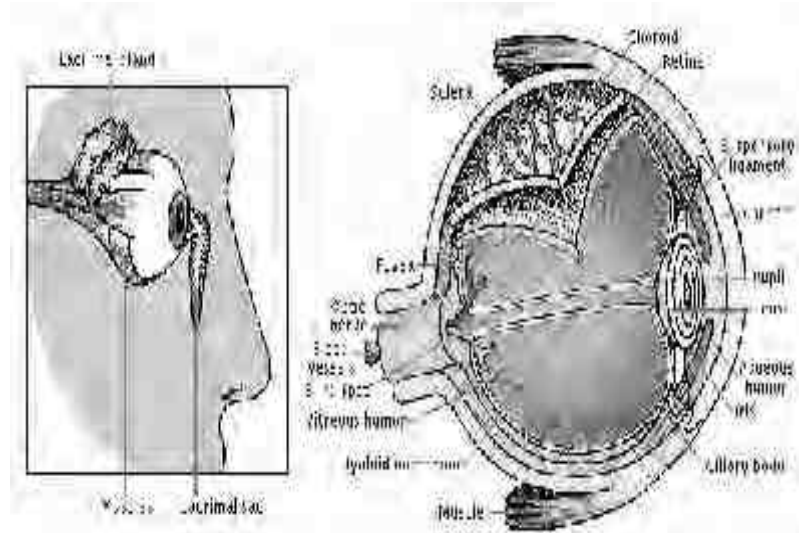
- c) Retina - it is innermost membrane of the eye. When the eye is properly focused, light from an object outside the eye is imaged on the retina. There are various light receptors over the surface of the retina

The two major classes of the receptors are-

- 1) cones- it is in the number about 6 to 7 million. These are located in the central portion of the retina called the fovea. These are highly sensitive to color. Human can resolve fine details with these cones because each one is connected to its own nerve end. Cone vision is called photopic or bright light vision

- 2) Rods - these are very much in number from 75 to 150 million and are distributed over the entire retinal surface. The large area of distribution and the fact that several rods are connected to a single nerve give a general overall picture of the field of view. They are not involved in the color vision and are sensitive to low level of illumination. Rod vision is called is scotopic or dim light vision.

The absent of reciprocators is called blind spot



### Image Formation in the Eye

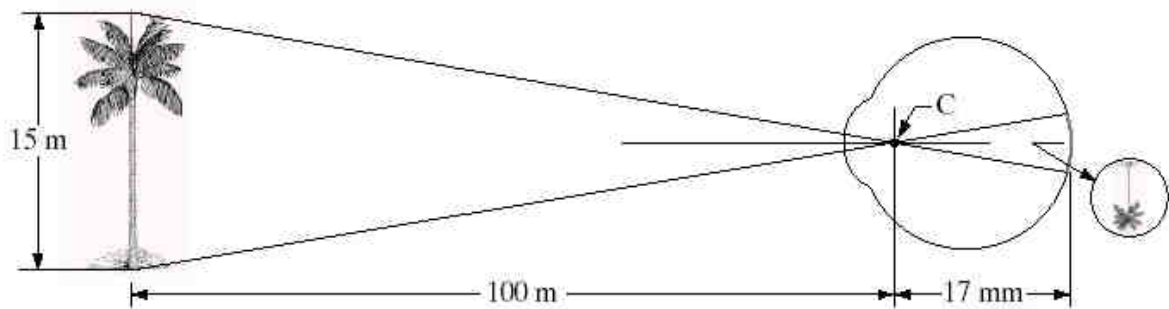
The major difference between the lens of the eye and an ordinary optical lens is that the former is flexible.

The shape of the lens of the eye is controlled by tension in the fibers of the ciliary body. To focus on the distant object the controlling muscles allow the lens to become thinner in order to focus on object near the eye it becomes relatively flattened.

The distance between the center of the lens and the retina is called the focal length and it varies from 17mm to 14mm as the refractive power of the lens increases from its minimum to its maximum.

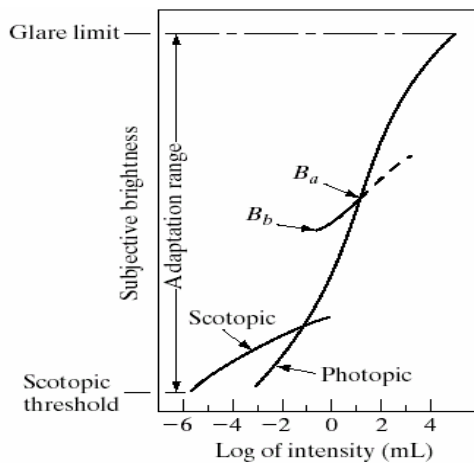
When the eye focuses on an object farther away than about 3m. the lens exhibits its lowest refractive power. When the eye focuses on a nearby object. The lens is most strongly refractive.

The retinal image is reflected primarily in the area of the fovea. Perception then takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that are ultimately decoded by the brain.



## Brightness Adaption and Discrimination

Digital image are displayed as a discrete set of intensities. The range of light intensity levels to which the human visual system can adopt is enormous- on the order of  $10^{10}$  from scotopic threshold to the glare limit. Experimental evidences indicate that subjective brightness is a logarithmic function of the light intensity incident on the eye.

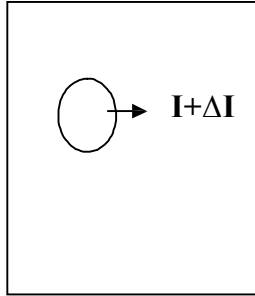


The curve represents the range of intensities to which the visual system can adopt. But the visual system cannot operate over such a dynamic range simultaneously. Rather, it is accomplished by change in its overall sensitivity called brightness adaptation.

For any given set of conditions, the current sensitivity level to which of the visual system is called brightness adoption level,  $B_a$  in the curve. The small intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. It is restricted at level  $B_b$ , at and below which all stimuli are perceived as indistinguishable blacks. The upper portion of the curve is not actually restricted, while simply raise the adaptation level higher than  $B_a$ .

The ability of the eye to discriminate between change in light intensity at any specific adaptation level is also of considerable interest.

Take a flat, uniformly illuminated area large enough to occupy the entire field of view of the subject. It may be a diffuser such as an opaque glass, that is illuminated from behind by a light source whose intensity,  $I$  can be varied. To this field is added an increment of illumination  $\Delta I$  in the form of a short duration flash that appears as circle in the center of the uniformly illuminated field. If  $\Delta I$  is not bright enough, the subject cannot see any perceivable changes.



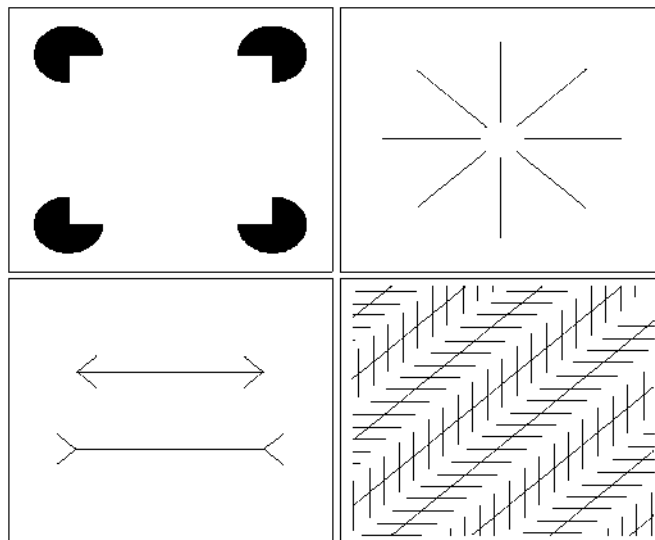
As  $\Delta I$  gets stronger the subject may indicate of a perceived change.  $\Delta I_c$  is the increment of illumination discernible 50% of the time with background illumination  $I$ . Now,  $\Delta I_c / I$  is called the Weber ratio.

Small value means that small percentage change in intensity is discernible representing 'good' brightness discrimination.

Large value of Weber ratio means large percentage change in intensity is required representing 'poor brightness discrimination'.

### Optical illusion

In this the eye fills the non existing information or wrongly pervious geometrical properties of objects.



## Fundamental Steps in Digital Image Processing

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

1. Methods whose outputs are input are images.
2. Methods whose outputs are attributes extracted from those images.

Color Image Processing	Wavelets & Multiresolution Processing	Image Compression	Morphological Image Processing
Image Restoration	Knowledge Base		Image Segmentation
Image Enhancement			Representation and description
Image Acquisition			Objects recognition

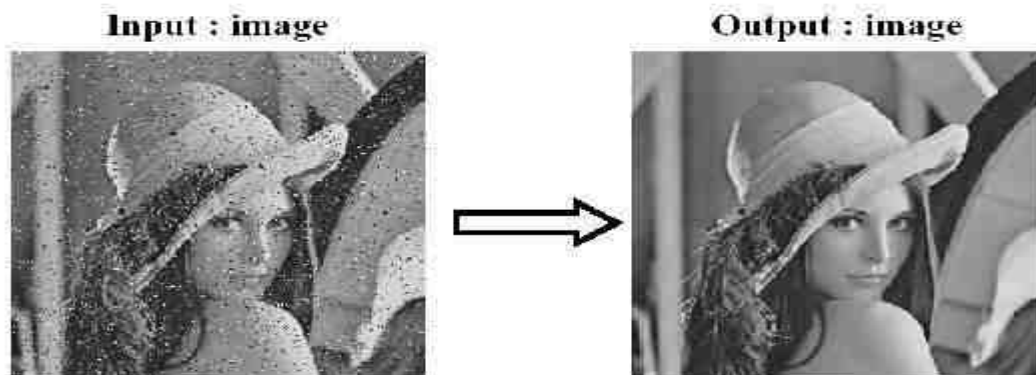
### Fundamental Steps in DIP

#### i) 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 as scaling.

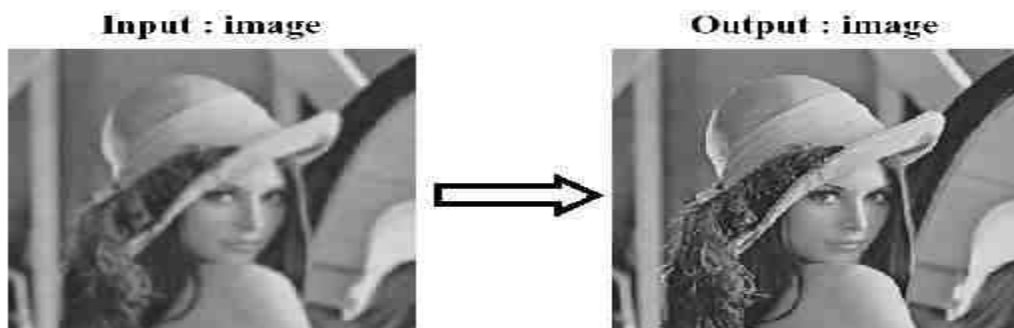
#### ii) Image Enhancement

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 image processing.



#### iii) 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



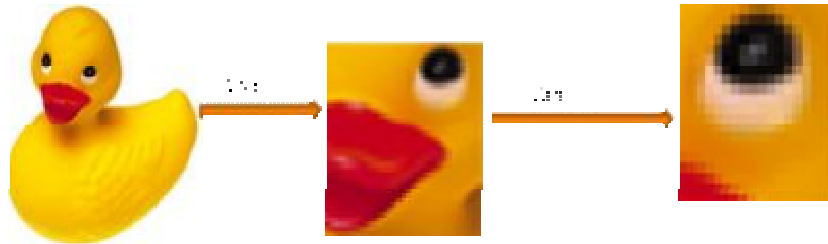


#### iv) Color image processing

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 processing applications.

#### v) Wavelets and Multiresolution Processing

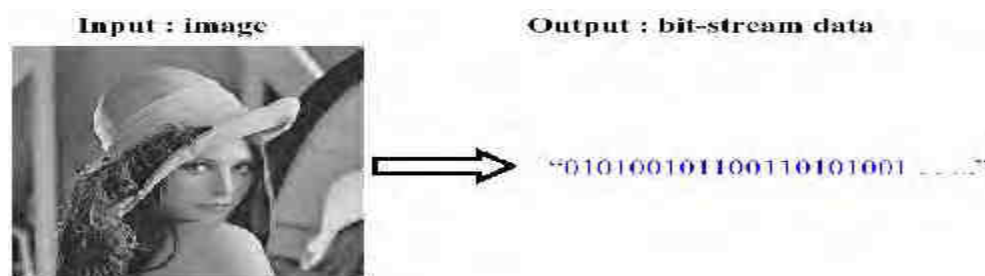
These are the foundation for representing image in various degrees of resolution



#### vi) 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 two major approaches:

- a) Lossless Compression
- b) Lossy Compression



#### vii) 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.

#### viii) 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.

#### ix) Recognition

It is the process that assigns label to an object based on its descriptors. It is the last step of image processing which uses artificial intelligence 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 is 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 as an 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.

## A 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  $0 < f(x, y) < \bar{e}$

The function  $f(x, y)$  may be characterized by two components-

- The amount of the source illumination incident on the scene being viewed.
- 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  $i(x, y)$  and  $r(x, y)$  respectively. The functions combine as a product to form  $f(x, y)$

We call the intensity of a monochrome image at any coordinate  $(x, y)$  the gray level ( $I$ ) of the image at that point  $I = f(x, y)$ ,  $L_{\min} \leq I \leq L_{\max}$

$L_{\min}$  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  $[0, L-1]$  where  $I=0$  is considered black and  $I=L-1$  is considered white on the gray scale. All intermediate values are shades of gray varying from black to white.

## Image Digitization

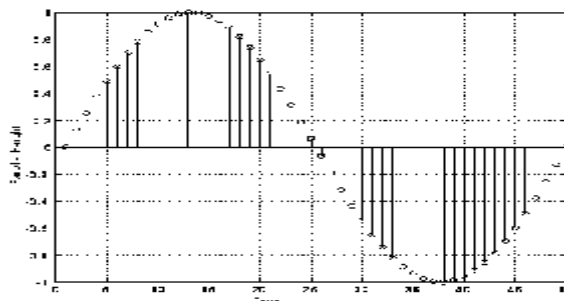
To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes - sampling and quantization. An image may be continuous with respect to the  $x$  and  $y$  coordinates and also in amplitude. To convert it into digital form we have to sample the function in both coordinates and in amplitudes.

**Digitalizing the coordinate values is called sampling**

**Digitalizing the amplitude values is called quantization**

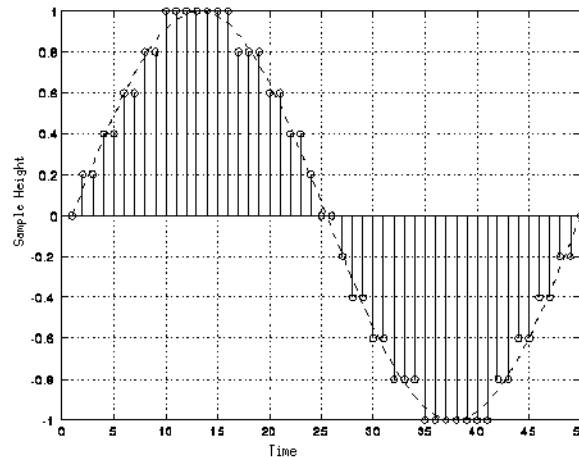
There is a continuous image along the line segment AB.

To sample 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 image, 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 black to white. The vertical tick mark assign the specific value assigned to each of the 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 sample to a vertical tick mark.

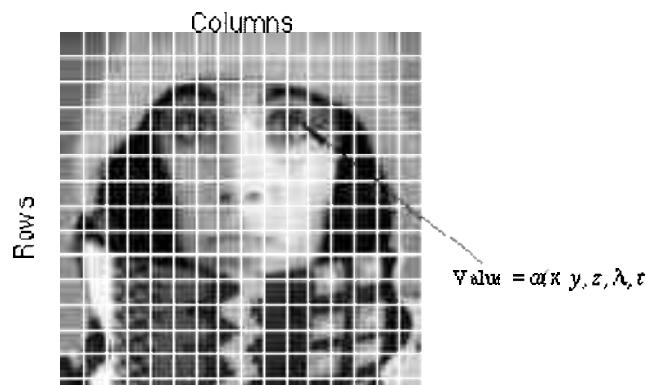


Starting at the top of the image and covering out this procedure line by line produces a two dimensional digital image.

## Digital Image Definition

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. Some basic definitions associated with the digital image are described.

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  $\{m=0,1,2,..., M-1\}$  and  $\{n=0,1,2,...,N-1\}$  is  $f[m,n]$ . In fact, in most cases  $f(x,y)$  is actually a function of many variables including depth ( $d$ ), color ( $\mu$ ) and time ( $t$ ).



There are three types of computerized processes in the processing of image

- 1) Low level process- these involve primitive operations such as image processing to reduce noise, contrast enhancement and image sharpening. These kind of processes are characterized by fact the both inputs and output are images.
- 2) Mid level image 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.
- 3) 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.

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

The sampling process may be viewed as partitioning the  $x$ - $y$  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  $Z$ .

Hence  $f(x,y)$  is a digital image if gray 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  $R$ , and a digital image become a 2D function whose coordinates and the amplitude value are integers.

Due to processing storage and hardware consideration, the number of gray levels typically is an integer power of 2.  $L=2^k$

Then, the number  $b$ , of bits 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 (because  $256=2^8$ ).

## 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 unit distance.



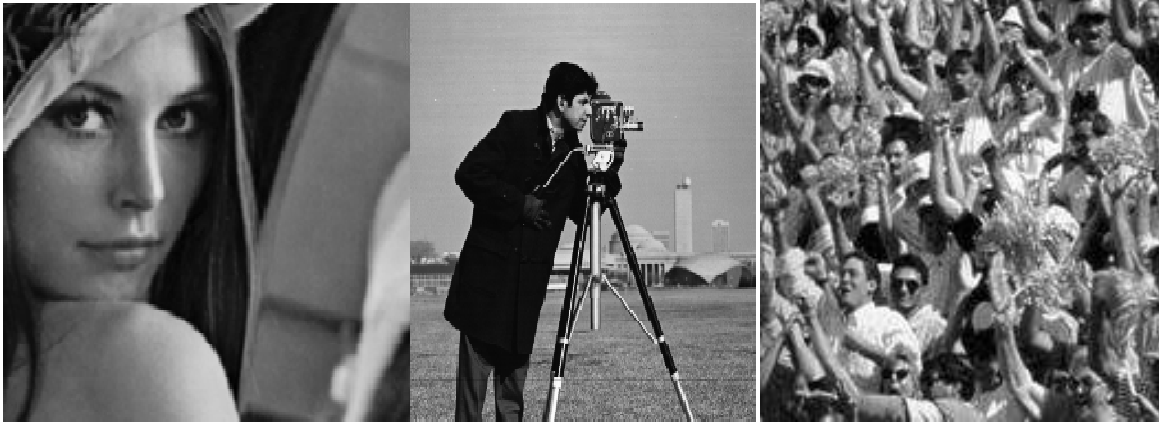
Gray levels resolution refers to smallest discernible change in gray levels.

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.

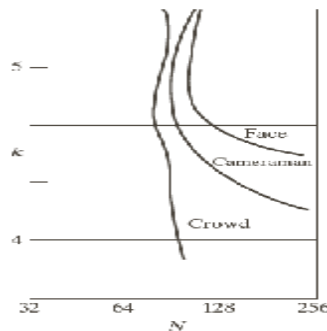


## Iso Preference Curves

To see the effect of varying  $N$  and  $R$  simultaneously. Three pictures are taken having little, mid level and high level of details.



Different images were generated by varying  $N$  and  $k$  and observers were then asked to rank the results according to their subjective quality. Results were summarized in the form of iso-preference curves in the  $N$ - $k$  plane.



The iso-preference curve tends to shift right and upward but their shapes in each of the three image categories are shown in the figure. A shift up and right in the curve simply means large values for  $N$  and  $k$  which implies better picture quality.

The result shows that the iso-preference curve tends to become more vertical as the detail in the image increases. The result suggests that for an image with a large amount of details, only a few gray levels may be needed. For a fixed value of  $N$ , the perceived quality for this type of image is nearly independent of the number of gray levels used.

## Pixel Relationships

### Neighbors of a pixel

A pixel  $p$  at coordinate  $(x,y)$  has four horizontal and vertical neighbor whose coordinate can be given by

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

This set of pixel is called the 4-neighbours of  $p$  and is denoted by  $n_4(p)$ , Each pixel is at a unit distance from  $(x,y)$  and some of the neighbors of  $P$  lie outside the digital image or  $(x,y)$  is on the border of the image .

The four diagonal neighbor of  $P$  have coordinates

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

And are denoted by  $n_d(p)$  these points, together with the 4-neighbours are called 8 - neighbors of  $P$  denoted by  $n_8(p)$

### Adjacency

Let  $V$  be the set of gray-level values used to define adjacency in a binary image, if  $V=\{1\}$  we are referencing to adjacency of pixel with value. Three types of adjacency occurs

4- Adjacency - two pixel  $P$  and  $Q$  with value from  $V$  are 4-adjacency if  $A$  is in the set  $n_4(P)$

8- Adjacency - two pixel  $P$  and  $Q$  with value from  $V$  are 8-adjacency if  $A$  is in the set  $n_8(P)$

M-adjacency - two pixel  $P$  and  $Q$  with value from  $V$  are  $m$ -adjacency if

- $Q$  is in  $n_4(p)$  or
- $Q$  is in  $n_d(q)$  and the set  $N_4(p) \cap N_4(q)$  has no pixel whose values are from  $V$

### 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

$$D[p,q] \geq 0 \quad \{D[p,q] = 0 \text{ iff } p=q\} \quad D$$

$$[p,q] = D[p,q] \text{ and}$$

$$D[p,q] \leq D[p,q] + D(q,z)$$

The Euclidean Distance between  $p$  and is defined as

$$De(p,q) = \sqrt{(x - t)^2 + (y - t)^2}$$

The D4 Education Distance between  $p$  and is defined as

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

## IMAGE ENHANCEMENT IN SPATIAL DOMAIN

### Introduction

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Image enhancement approaches fall into two broad categories

- Ø Spatial domain methods
- Ø Frequency domain methods

The term spatial domain refers to the image plane itself and approaches in this categories are based on direct manipulation of pixel in an image.

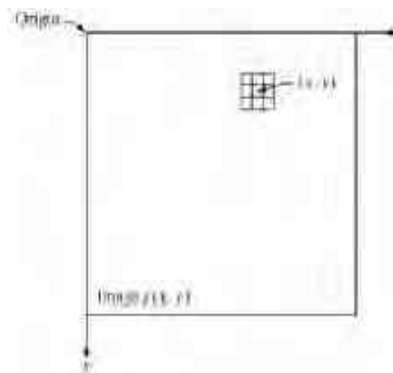
Spatial domain process are denoted by the expression

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

$f(x,y)$ - input image     $T$ - operator on  $f$ , defined over some neighborhood of  $f(x,y)$

$g(x,y)$ -processed image

The neighborhood of a point  $(x,y)$  can be explain by using as square or rectangular sub image area centered at  $(x,y)$ .



The center of sub image is moved from pixel to pixel starting at the top left corner. The operator  $T$  is applied to each location  $(x,y)$  to find the output  $g$  at that location . The process utilizes only the pixel in the area of the image spanned by the neighborhood.

### Basic Gray Level Transformation Functions

It is the simplest form of the transformations when the neighborhood is of size  $1 \times 1$ . In this case  $g$  depends only on the value of  $f$  at  $(x,y)$  and  $T$  becomes a gray level transformation function of the forms

$$S=T(r)$$

$r$ - Denotes the gray level of  $f(x,y)$

$s$ - Denotes the gray level of  $g(x,y)$  at any point  $(x,y)$



Because enhancement at any point in an image deepens only on the gray level at that point, techniques in this category are referred to as point processing.

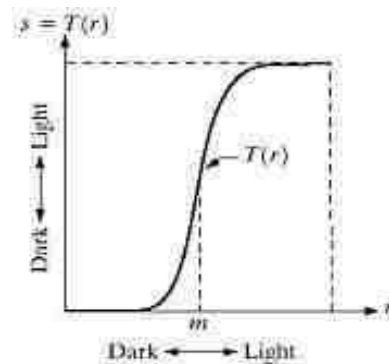
There are basically three kinds of functions in gray level transformation -

## Point Processing

Contract stretching -

It produces an image of higher contrast than the original one.

The operation is performed by darkening the levels below  $m$  and brightening the levels above  $m$  in the original image.

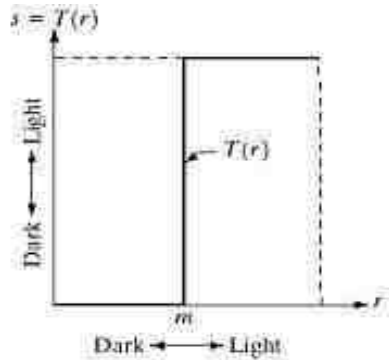


In this technique the value of  $r$  below  $m$  are compressed by the transformation function into a narrow range of  $s$  towards black. The opposite effect takes place for the values of  $r$  above  $m$ .

Thresholding function -

It is a limiting case where  $T(r)$  produces a two levels binary image.

The values below  $m$  are transformed as black and above  $m$  are transformed as white.



## Basic Gray Level Transformation

These are the simplest image enhancement techniques

Image Negative -

The negative of an image with gray level in the range  $[0, 1-1]$  is obtained by using the negative transformation.

The expression of the transformation is

$$s = L - 1 - r$$

Reverting the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is practically suited for enhancing white or gray details embedded in dark regions of an image especially when the black areas are dominant in size.



Log transformations-

The general form of log transform is

$$S = c \log(1 + R)$$

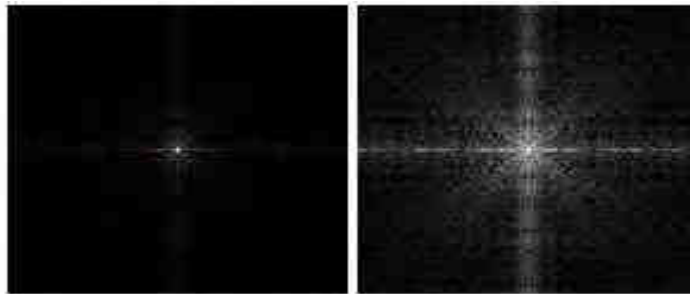
Where  $R \in [0, 255]$

This transformation maps a narrow range of gray level values in the input image into a wider range of output gray levels. The opposite is true for higher values of input levels. We would use this transformation to expand the values of dark pixels in an image while compressing the higher level values. The opposite is true for inverse log transformation.

The log transformation function has an important characteristic that it compresses the dynamic range of images with large variations in pixel values.

Eg-

Fourier spectrum



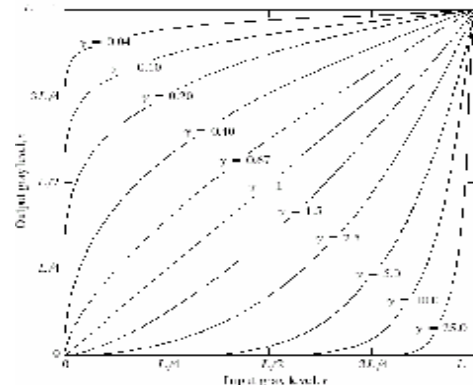
Power law transformation

Power law transformation has the basic function

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

Where  $c$  and  $\gamma$  are positive constants.

Power law curves with fractional values of  $\gamma$  map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input gray levels. We may get various curves by varying values of  $\gamma$ .



A variety of devices used for image capture, printing and display respond according to a power

law. The process used to correct this power law response phenomenon is called gamma correction.

For eg-CRT devices have intensity to voltage response that is a power function

Gamma correction is important if displaying an image accurately on a computer screen is of concern. Images that are not corrected properly can look either bleached out or too dark.

Color phenomenon also uses this concept of gamma correction. It is becoming more popular due to use of images over the internet.

It is important in general purpose contrast manipulation. To make an image black we use  $y > 1$  and  $y < 1$  for white image.

### Piece wise Linear transformation functions-

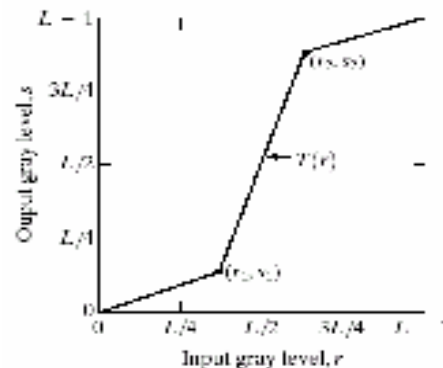
The principal advantage of piecewise linear functions is that these functions can be arbitrarily complex. But their specification requires considerably more user input

#### Contrast Stretching-

It is the simplest piecewise linear transformation function.

We may have various low contrast images and that might result due to various reasons such as lack of illumination, problem in imaging sensor or wrong setting of lens aperture during image acquisition.

The idea behind contrast stretching is to increase the dynamic range of gray levels in the image being processed.



The location of points  $(r_1, s_1)$  and  $(r_2, s_2)$  control the shape of the curve

- If  $r_1 = r_2$  and  $s_1 = s_2$ , the transformation is a linear function that deduces no change in gray levels.
- If  $r_1 = s_1$ ,  $s_1 = 0$ , and  $s_2 = L-1$ , then the transformation become a thresholding function that creates a binary image
- Intermediate values of  $(r_1, s_1)$  and  $(r_2, s_2)$  produce various degrees of spread in the gray value of the output image thus effecting its contract.

Generally  $r_1 \neq r_2$  and  $s_1 \neq s_2$  so that the function is single valued and monotonically increasing

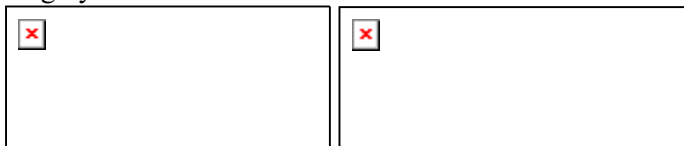
#### Gray Level Slicing-

Highlighting a specific range of gray levels in an image is often desirable

For example when enhancing features such as masses of water in satellite image and enhancing flaws in x- ray images.

There are two ways of doing this-

- (1) One method is to display a high value for all gray level in the range. Of interest and a low value for all other gray level.

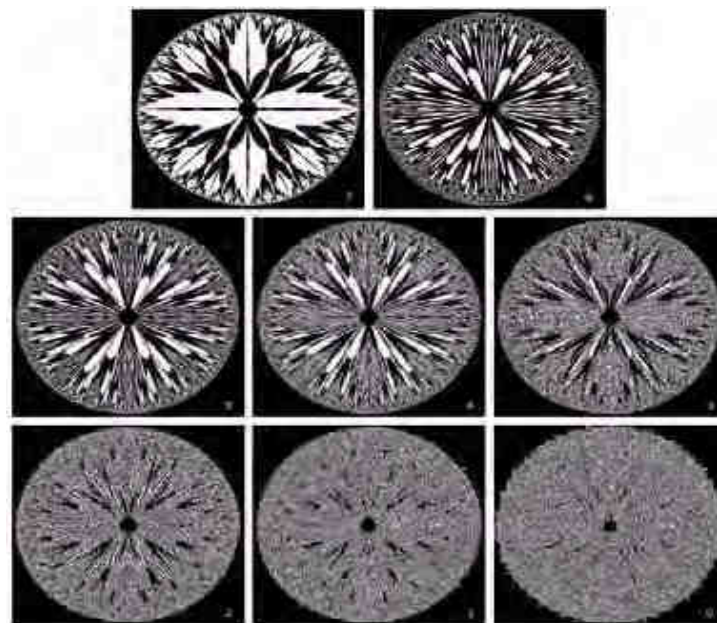
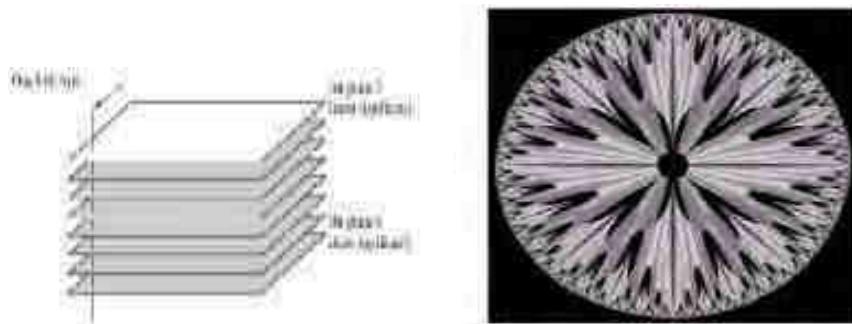


(2) Second method is to brighten the desired ranges of gray levels but preserve the background and gray level tonalities in the image

#### Bit Plane Slicing

Sometimes it is important to highlight the contribution made to the total image appearance by specific bits. Suppose that each pixel is represented by 8 bits.

Imagine that an 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 image and plane 7 contains all the high order bits



High order bits contain the majority of visually significant data and contribute to more subtle details in the image.

Separating a digital image into its bits planes is useful for analyzing the relative importance

played by each bit of the image.

It helps in determining the adequacy of the number of bits used to quantize each pixel. It is also useful for image compression.

### Histogram Processing

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

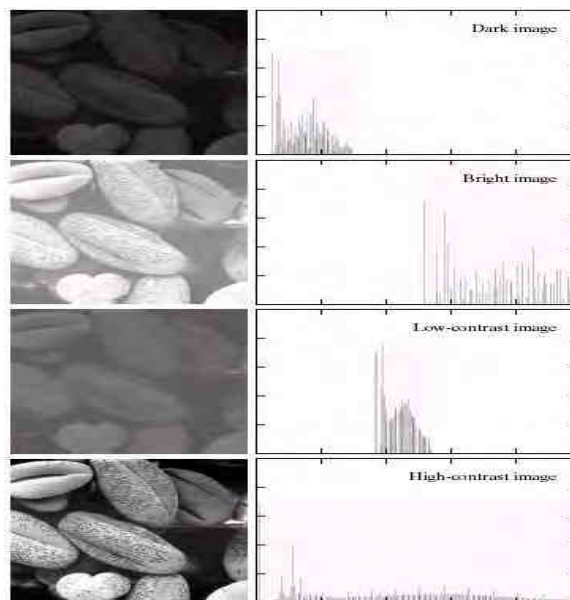
$$H(r_k) = n_k$$

where  $r_k$  is the  $k$ th 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

$$p(r_k) = n_k/n \quad \text{for } k=0, 1, 2, \dots, L-1$$

$P(r_k)$  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 be skewed 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  $r=0$  representing black and  $r=1$  representing white. The transformation function is of the form.

$$S = T(r) \quad \text{where } 0 < r < 1$$

It produces a level  $s$  for every pixel value  $r$  in the original image. The transformation function is assumed to fulfill two conditions:  $T(r)$  is single valued and monotonically increasing in the interval  $[0, 1]$ .

$$0 < T(r) < 1 \text{ for } 0 < r < 1$$

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  $s$  is given by the formula-

$$P_s(s) = P_r(r) \frac{dr}{ds},$$

Thus the PDF of the transformed variable  $s$  is 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

$$s = T(r) = \int_0^r P_r(w) dw.$$

This is the cumulative distribution function of  $r$

Using this definition of  $T$  we see that the derivative of  $s$  with respect to  $r$  is

$$\frac{ds}{dr} = P_r(r).$$

Substituting it back in the expression for  $P_s$  we may get

$$P_s(s) = P_r(r) \frac{1}{P_r(r)} = \frac{1}{1} = 1$$

An important point here is that  $T_r$  depends on  $P_r(r)$  but the resulting  $P_s(s)$  always is uniform, and independent of the form of  $P(r)$ .

For discrete values we deal with probability and summations instead of probability density functions and integrals.

The probability of occurrence of gray levels  $r_k$  in an image as approximated

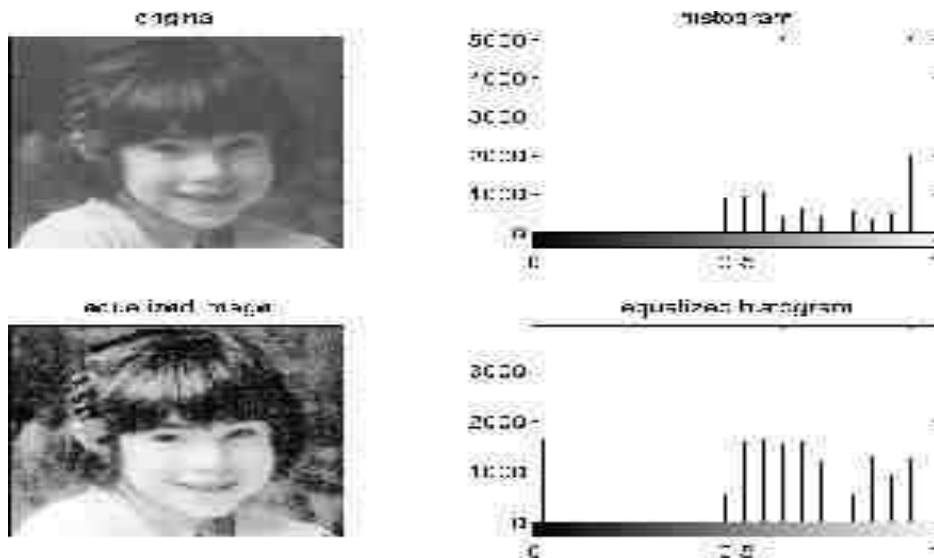
$$P_r(r) = nk/N$$

$N$  is the total number of the pixels in an image.  $nk$  is the number of the pixels that have gray level  $r_k$ .  $L$  is the total number of possible gray levels in the image. The discrete transformation function is given by

$$\begin{aligned} s_k = T(r_k) &= \sum_{i=0}^k \frac{n_i}{N} \\ &= \sum_{i=0}^k P_r(r_i). \end{aligned}$$

Thus a processed image is obtained by mapping each pixel with levels  $r_k$  in the input image into a corresponding pixel with level  $s_k$  in the output image.

A plot of  $P_r(r_k)$  versus  $r_k$  is called a histogram. The transformation function given by the above equation is called histogram equalization or linearization. Given an image the process of histogram equalization consists simple of implementing the transformation function which is based information that can be extracted directly from the given image, without the need for further parameter specification



Equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram. It is a good approach when automatic enhancement is needed.

#### Histogram Matching (Specification)

In some cases it may be desirable to specify the shape of the histogram that we wish the processed image to have.

Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram. Sometimes we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges. The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

#### Algorithm

1. Compute  $s_k = P_f(k)$ ,  $k = 0, \dots, L-1$ , the cumulative normalized histogram of  $f$ .
2. Compute  $G(k)$ ,  $k = 0, \dots, L-1$ , the transformation function, from the given histogram  $h_z$ .
3. Compute  $G^{-1}(s_k)$  for each  $k = 0, \dots, L-1$  using an iterative method (iterate on  $z$ ), or in effect, directly compute  $G^{-1}(P_f(k))$ .
4. Transform  $f$  using  $G^{-1}(P_f(k))$ .

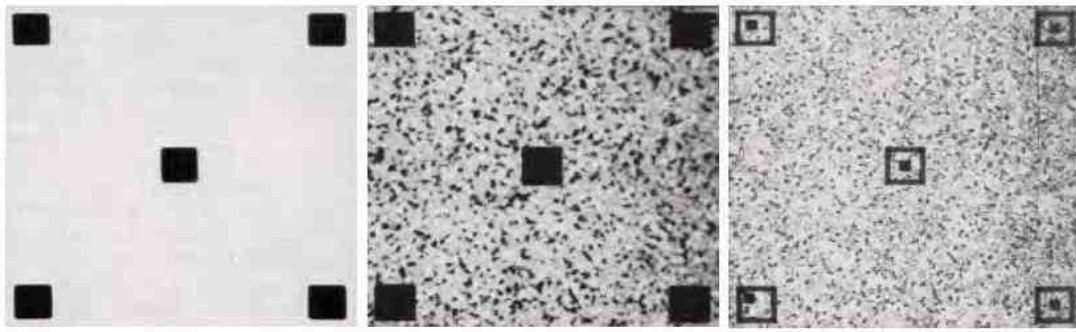
#### Local Enhancement

In earlier methods pixels were modified by a transformation function based on the gray level of an entire image. It is not suitable when enhancement is to be done in some small areas of the image. This problem can be solved by local enhancement where a transformation function is applied only in the neighborhood of pixels in the interested region.

Define square or rectangular neighborhood (mask) and move the center from pixel to pixel.

For each neighborhood

- 1) Calculate histogram of the points in the neighborhood
- 2) Obtain histogram equalization/specification function
- 3) Map gray level of pixel centered in neighborhood
- 4) The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.



### Enhancement Using Arithmetic/Logic Operations

These operations are performed on a pixel by pixel basis between two or more images excluding not operation which is performed on a single image. It depends on the hardware and/or software that the actual mechanism of implementation should be sequential, parallel or simultaneous.

Logic operations are also generally operated on a pixel by pixel basis.

Only AND, OR and NOT logical operators are functionally complete. Because all other operators can be implemented by using these operators.

While applying the operations on gray scale images, pixel values are processed as strings of binary numbers.

The NOT logic operation performs the same function as the negative transformation.

Image Masking is also referred to as region of Interest (RoI) processing. This is done to highlight a particular area and to differentiate it from the rest of the image.

Out of the four arithmetic operations, subtraction and addition are the most useful for image enhancement.

#### Image Subtraction

The difference between two images  $f(x,y)$  and  $h(x,y)$  is expressed as

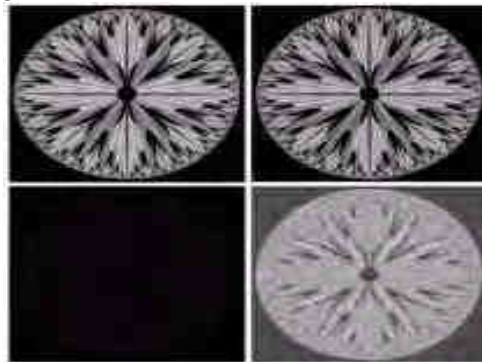
$$g(x,y) = f(x,y) - h(x,y)$$

It is obtained by computing the difference between all pairs of corresponding pixels from  $f$  and  $h$ .

The key usefulness of subtraction is the enhancement of difference between images.

This concept is used in another gray scale transformation for enhancement known as bit plane slicing. The higher order bit planes of an image carry a significant amount of visually relevant detail while the lower planes contribute to fine details.

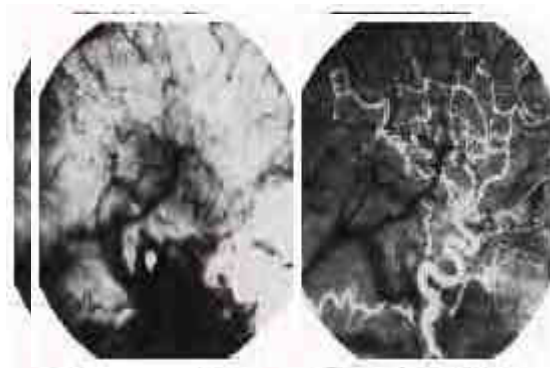
If we subtract the four least significant bit planes from the image the result will be nearly identical but there will be a slight drop in the overall contrast due to less variability in the gray level values of image



The use of image subtraction is seen in medical imaging area named as mask mode radiography. The mask  $h(x,y)$  is an X-ray image of a region of a patient's body this image is captured by using an intensified TV camera located opposite to the x-ray machine then a consistent medium is injected into the patient's blood stream and then a series of images are taken of the region same as  $h(x,y)$ .

The mask is then subtracted from the series of incoming images. This subtraction will give the area which will be the difference between  $f(x,y)$  and  $h(x,y)$  this difference will be given as enhanced detail in the output image.





This procedure produces a movie showing how the contrast medium propagates through various arteries of the area being viewed.

Most of the image in use today is 8-bit image so the values of the image lie in the range 0 to 255. The value in the difference image can lie from -255 to 255. For these reasons we have to do some sort of scaling to display the results.

There are two methods to scale an image

- (i) Add 255 to every pixel and then divide it by 2.

This gives the surety that pixel values will be in the range 0 to 255 but it is not guaranteed whether it will cover the entire 8-bit range or not.

It is a simple method and fast to implement but will not utilize the entire gray scale range to display the results.

- (ii) Another approach is

- (a) Obtain the value of minimum difference
- (b) Add the negative of minimum value to the pixels in the difference image (this will give a modified image whose minimum value will be 0)
- (c) Perform scaling on the difference image by multiplying each pixel by the quantity  $255/\text{max}$ . This approach is complicated and difficult to implement.

Image subtraction is used in segmentation application also

**Image Averaging**

Consider a noisy image  $g(x,y)$  formed by the addition of noise  $n(x,y)$  to the original image  $f(x,y)$

$$g(x,y) = f(x,y) + n(x,y)$$

Assuming that at every point of coordinate  $(x,y)$  the noise is uncorrelated and has zero average value. The objective of image averaging is to reduce the noise content by adding a set of noise images,

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

If an image is formed by image averaging  $K$  different noisy images.

As  $k$  increases the variability (noise) of the pixel value at each location  $(x,y)$  decreases

$E\{g(x,y)\} = f(x,y)$  means that  $g(x,y)$  approaches  $f(x,y)$  as the number of noisy images used in the averaging processes increases

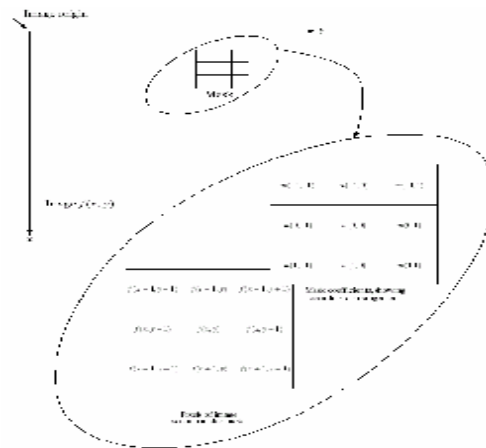
Image averaging is important in various applications such as in the field of astronomy where the images are low light levels

### **Basic of Spatial Filtering**

Spatial filtering is an example of neighborhood operations, in this the operations are done on the values of the image pixels in the neighborhood and the corresponding value of a sub image that has the same dimensions as of the neighborhood

This sub image is called a filter, mask, kernel, template or window; the values in the filter sub image are referred to as coefficients rather than pixels. Spatial filtering operations are performed directly on the pixel values (amplitude/gray scale) of the image

The process consists of moving the filter mask from point to point in the image. At each point  $(x,y)$  the response is calculated using a predefined relationship



For linear spatial filtering the response is given by a sum of products of the filter coefficient and the corresponding image pixels in the area spanned by the filter mask.

The results  $R$  of linear filtering with the filter mask at point  $(x,y)$  in the image is

$$R = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 0)f(x + 1, y) + w(1, 1)f(x + 1, y + 1) + 1)$$

The sum of products of the mask coefficient with the corresponding pixel directly under the mask. The coefficient  $w(0,0)$  coincides with image value  $f(x,y)$  indicating that mask is centered at  $(x,y)$  when the computation of sum of products takes place.

For a mask of size  $M \times N$  we assume  $m=2a+1$  and  $n=2b+1$ , where  $a$  and  $b$  are nonnegative integers. It shows that all the masks are of odd size.

In the general linear filtering of an image of size  $f$  of size  $M \times N$  with a filter mask of size  $m \times m$  is given by the expression.

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

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

To generate a complete filtered image this equation must be applied for  $x=0, 1, 2, \dots, M-1$  and  $y=0, 1, 2, \dots, N-1$ . Thus the mask processes all the pixels in the image.

The process of linear filtering is similar to frequency domain concept called convolution. For this reason, linear spatial filtering often is referred to as convolving a mask with an image. Filter mask are sometimes called convolution mask.

$$R = \sum_{m,n} w_{mn} Z_{mn}$$

Where  $w$ 's are mask coefficients and

$z$ 's are the values of the image gray levels corresponding to those coefficients.

$mn$  is the total number of coefficients in the mask.

An important point in implementing neighborhood operations for spatial filtering is the issue of what happens when the center of the filter approaches the border of the image.

There are several ways to handle this situation.

i) To limit the excursion of the center of the mask to be at distance of less than  $(n-1)/2$  pixels from the border. The resulting filtered image will be smaller than the original but all the pixels will be processed with the full mask.

ii) Filter all pixels only with the section of the mask that is fully contained in the image. It will create bands of pixels near the border that will be processed with a partial mask.

iii) Padding the image by adding rows and columns of 0's & of padding by replicating rows and columns. The padding is removed at the end of the process

Smoothing Spatial Filters

These filters are used for blurring and noise reduction blurring is used in preprocessing steps such as removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves.

### Smoothing Linear Filters

The output of a smoothing linear spatial filter is simply the average of the pixel contained in the neighborhood of the filter mask. These filters are also called averaging filters or low pass filters.

The operation is performed by replacing the value of every pixel in the image by the average of the gray levels in the neighborhood defined by the filter mask. This process reduces sharp transitions in gray levels in the image



A major application of smoothing is noise reduction but because edge are also provided using sharp transitions so smoothing filters have the undesirable side effect that they blur edges . It also removes an effect named as false contouring which is caused by using insufficient number of gray levels in the image.

Irrelevant details can also be removed by these kinds of filters, irrelevant means which are not of our interest.

A spatial averaging filter in which all coefficients are equal is sometimes referred to as a 'box filter'.

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

A weighted average filter is the one in which pixel are multiplied by different coefficients.

### Order Statistics Filter

These are nonlinear spatial filter whose response is based on ordering of the pixels contained in the image area compressed by the filter and the replacing the value of the center pixel with value determined by the ranking result.

The best example of this category is median filter. In this filter the values of the center pixel is replaced by median of gray levels in the neighborhood of that pixel. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters.

These filters are particularly effective in the case of impulse or salt and pepper noise. It is called so because of its appearance as white and black dots superimposed on an image.

The median  $\ell$  of a set of values is such that half the values in the set less than or equal to  $\ell$  and half are greater than or equal to this. In order to perform median filtering at a point in an image,

we first sort the values of the pixel in the question and its neighbors, determine their median and assign this value to that pixel.

We introduce some additional order-statistics filters. Order-statistics filters are spatial filters whose response is based on ordering (ranking) the pixels 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

The best-known order-statistics filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel.

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

The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, the median filter yields excellent results for images corrupted by this type of noise.

#### Max and min filters

Although the median filter is by far the order-statistics filter most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called max filter given by:

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

This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area  $S$ . The 0th percentile filter is the Min filter.

#### Sharpening Spatial Filters

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

The applications of image sharpening range from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.

As smoothing can be achieved by integration, sharpening can be achieved by spatial differentiation. The strength of response of derivative operator is proportional to the degree of discontinuity of the image at that point at which the operator is applied. Thus image differentiation enhances edges and other discontinuities and deemphasizes the areas with slow varying grey levels.

It is a common practice to approximate the magnitude of the gradient by using absolute values instead of square and square roots.

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

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

Similarly we can define a second order derivative as the difference

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

The LAPLACIAN

The second order derivative is calculated using Laplacian. It is simplest isotropic filter. Isotropic filters are the ones whose response is independent of the direction of the image to which the operator is applied.

The Laplacian for a two dimensional function  $f(x,y)$  is defined as

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

Partial second order derivative in the x-direction

And similarly in the y-direction

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

The digital implementation of a two-dimensional Laplacian obtained by summing the two components

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

The equation can be represented using any one of the following masks

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Laplacian highlights gray-level discontinuities in an image and deemphasize the regions of slow varying gray levels. This makes the background a black image. The background texture can be recovered by adding the original and Laplacian images.

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

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

For example



The strength of the response of a derivative operator is propositional to the degree of discontinuity of the image at that point at which the operator is applied. Thus image differentiation enhances eddies and other discontinuities and deemphasizes areas with slowly varying gray levels.

The derivative of a digital function is defined in terms of differences. Any first derivative definition

- (1) Must be zero in flat segments (areas of constant gray level values)
- (2) Must be nonzero at the onset of a gray level step or ramp
- (3) Must be nonzero along ramps.

Any second derivative definition

- (1) Must be zero in flat areas
- (2) Must be nonzero at the onset and end of a gray level step or ramp
- (3) Must be zero along ramps of constant slope .

It is common practice to approximate the magnitude of the gradient by using also lute values instead or squares and square roots.

Roberts Goss gradient operators

For digitally implementing the gradient operators

Let center point,  $5z$  denote  $f(x,y)$ ,  $Z1$  denotes  $f(x-1,y)$  and so on

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

-1	0	0	-1
0	1	1	0

But it different implement even sized mask. So the smallest filter mask is size 3x3 mask is

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

The difference between third and first row a 3x3 mask approximates the derivate in the x-direction and difference between the third and first column approximates the derivative in y-direction.

These masks are called sobel operators.

Unsharp Masking and High Boost Filtering.

Unsharp masking means subtracting a blurred version of an image form the image itself.

Where  $f(x,y)$  denotes the sharpened image obtained by unsharp masking and  $\bar{f}(x,y)$  is a blurred version of  $(x,y)$

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

A slight further generalization of unsharp masking is called high boost filtering. A high boost filtered image is defined at any point  $(x,y)$  as

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

## IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

### Fourier Transform and the Frequency Domain

Any function that periodically reports itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient, this sum is called Fourier series. Even the functions which are non periodic but whose area under the curve if finite can also be represented in such form; this is now called Fourier transform.

A function represented in either of these forms and can be completely reconstructed via an inverse process with no loss of information.

### 1-D Fourier Transformation and its Inverse

If there is a single variable, continuous function  $f(x)$  , then Fourier transformation  $F(u)$  may be given as

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

And the reverse process to recover  $f(x)$  from  $F(u)$  is

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \exp[j2\pi ux] du$$

Fourier transformation of a discrete function of one variable  $f(x)$ ,  $x=0, 1, 2, \dots, m-1$  is given by

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \text{ for } u=0,1,2,\dots,N-1$$

to obtain  $f(x)$  from  $F(u)$

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \text{ for } x=0,1,2,\dots,N-1$$

The above two equations (e) and (f) comprise of a discrete Fourier transformation pair. According to Euler's formula

$$e^{jx} = \cos x + j \sin x$$

Substituting these values to equation (e)

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) [\cos(2\pi ux/N) + j \sin(2\pi ux/N)] \text{ for } u=0,1,2,\dots,N-1$$

Now each of the  $m$  terms of  $F(u)$  is called a frequency component of transformation

The Fourier transformation separates a function into various components, based on frequency components. These components are complex quantities.

$F(u)$  in polar coordinates

$$F(u) = R(u) + jI(u) \quad \text{or} \quad F(u) = |F(u)| e^{j\phi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad \phi(u) = \tan^{-1} \left[ \frac{I(u)}{R(u)} \right]$$

## 2-D Fourier Transformation and its Inverse

The Fourier Transform of a two dimensional continuous function  $f(x,y)$  (an image) of size  $M * N$  is given by

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

Inverse Fourier transformation is given by equation



$$\mathcal{F}^{-1}\{F(u, v)\} = f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[j2\pi(ux + vy)] du dv$$

Where (u,v) are frequency variables.

Preprocessing is done to shift the origin of F(u,v) to frequency coordinate (m/2, n/2) which is the center of the M\*N area occupied by the 2D-FT. It is known as frequency rectangle.

It extends from u = 0 to M-1 and v = 0 to N-1. For this, we multiply the input image by  $(-1)^{x+y}$  prior to compute the transformation

$$\mathcal{F}\{f(x, y) (-1)^{x+y}\} = F(u - M/2, v - N/2)$$

$\mathcal{F}\{ \}$  denotes the Fourier transformation of the argument

Value of transformation at (u,v)=(0,0) is

$$F(0,0) = \frac{1}{MN} \sum_{x,y} f(x,y)$$

Discrete Fourier Transform

✖

✖

Extending it to two variables

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

for  $u=0,1,2,\dots,M-1$   $v=0,1,2,\dots,N-1$

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

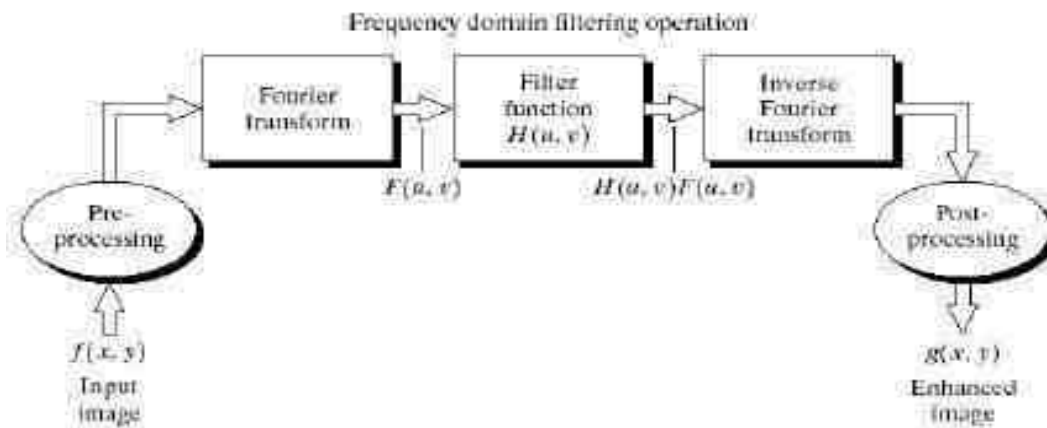
for  $x=0,1,\dots,M-1$   $y=0,1,\dots,N-1$

$$\Delta u = \frac{1}{M\Delta x} \quad \Delta v = \frac{1}{N\Delta y}$$

### Basis of Filtering in Frequency Domain

Basic steps of filtering in frequency Domain

- i) Multiply the input image by  $(-1)^{x+y}$  to centre the transform
- ii) Compute  $F(u,v)$ , Fourier Transform of the image
- iii) Multiply  $f(u,v)$  by a filter function  $H(u,v)$
- iv) Compute the inverse DFT of Result of (iii)
- v) Obtain the real part of result of (iv)
- vi) Multiply the result in (v) by  $(-1)^{x+y}$



$H(u,v)$  called a filter because it suppresses certain frequencies from the image while leaving others unchanged.

#### Filters

Smoothing Frequency Domain Filters

Edges and other sharp transition of the gray levels of an image contribute significantly to the high frequency contents of its Fourier transformation. Hence smoothing is achieved in the frequency domain by attenuation a specified range of high frequency components in the transform of a given image.

Basic model of filtering in the frequency domain is

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

F(u,v) - Fourier transform of the image to be smoothed

Objective is to find out a filter function H (u,v) that yields G (u,v) by attenuating the high frequency component of F (u,v)

There are three types of low pass filters

1. Ideal
2. Butterworth
3. Gaussian

### IDEAL LOW PASS FILTER

It is the simplest of all the three filters

It cuts off all high frequency component of the Fourier transform that are at a distance greater than a specified distance  $D_0$  from the origin of the transform.

it is called a two - dimensional ideal low pass filter (ILPF) and has the transfer function

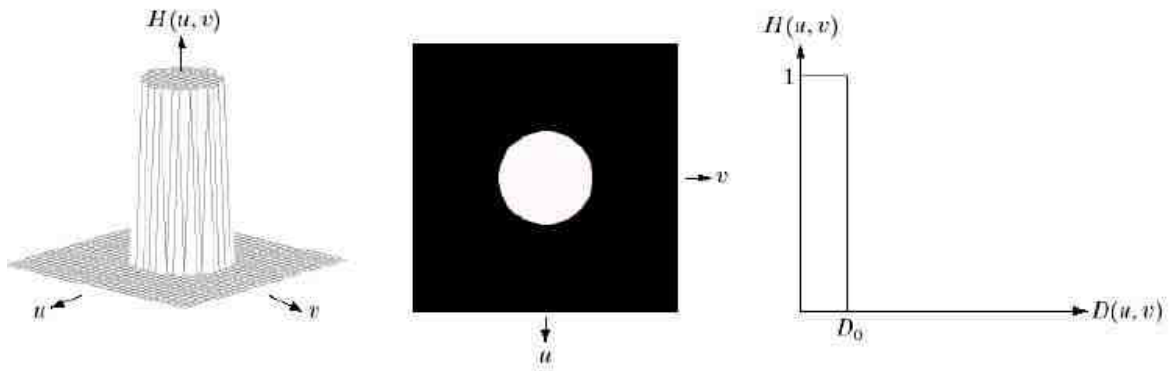
$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

Where  $D_0$  is a specified nonnegative quantity and  $D(u,v)$  is the distance from point (u,v) to the center of frequency rectangle

If the size of image is  $M \times N$ , filter will also be of the same size so center of the frequency rectangle  $(u,v) = (M/2, N/2)$  because of center transform

$$D(u, v) = (u^2 + v^2)^{1/2}$$

ILPF is not suitable for practical usage. But they can be implemented in any computer system



BUTTERWORTH LOW PASS FILTER It has a parameter

called the filter order.

For high values of filter order it approaches the form of the ideal filter whereas for low filter order values it reach Gaussian filter. It may be viewed as a transition between two extremes. The transfer function of a Butterworth low pass filter (BLPF) of order  $n$  with cut off frequency at distance  $D_0$  from the origin is defined as

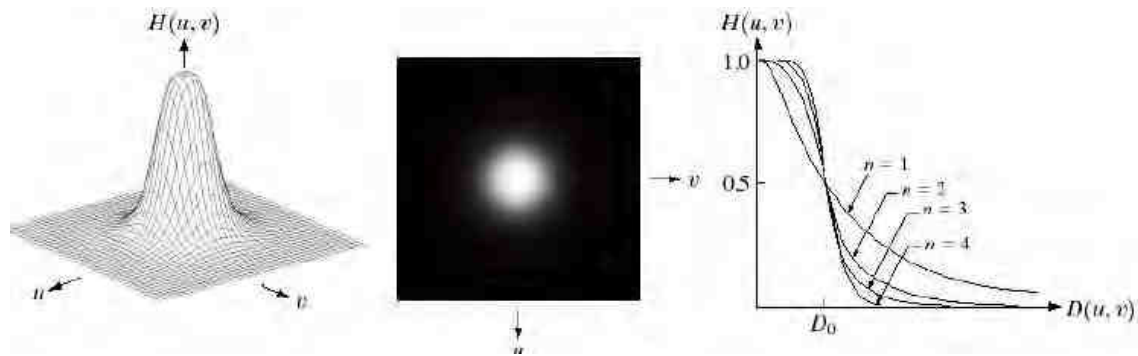
$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$

Most appropriate value of  $n$  is

2.

It does not have sharp discontinuity unlike ILPF that establishes a clear cutoff between passed and filtered frequencies.

Defining a cutoff frequency is a main concern in these filters. This filter gives a smooth transition in blurring as a function of increasing cutoff frequency. A Butterworth filter of order 1 has no ringing. Ringing increases as a function of filter order. (Higher order leads to negative values)



### GAUSSIAN LOW PASS FILTER

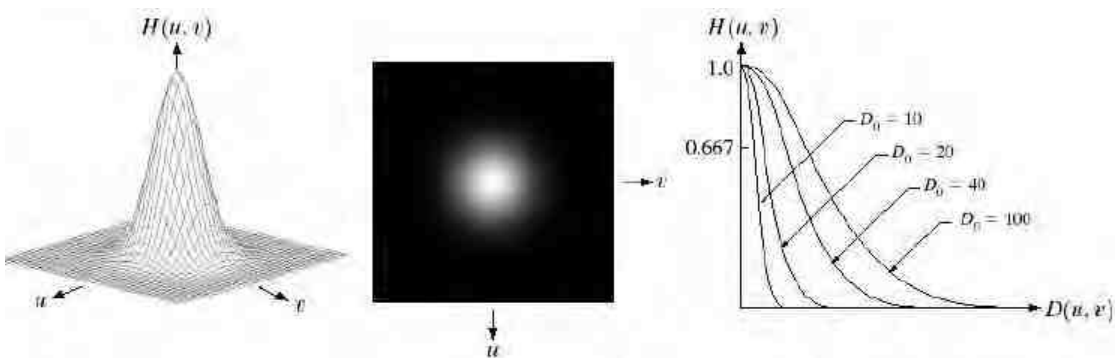
The transfer function of a Gaussian low pass filter is

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

Where:

$D(u, v)$ - the distance of point  $(u, v)$  from the center of the transform  
 $= D_0$ - specified cut off frequency

The filter has an important characteristic that the inverse of it is also Gaussian.



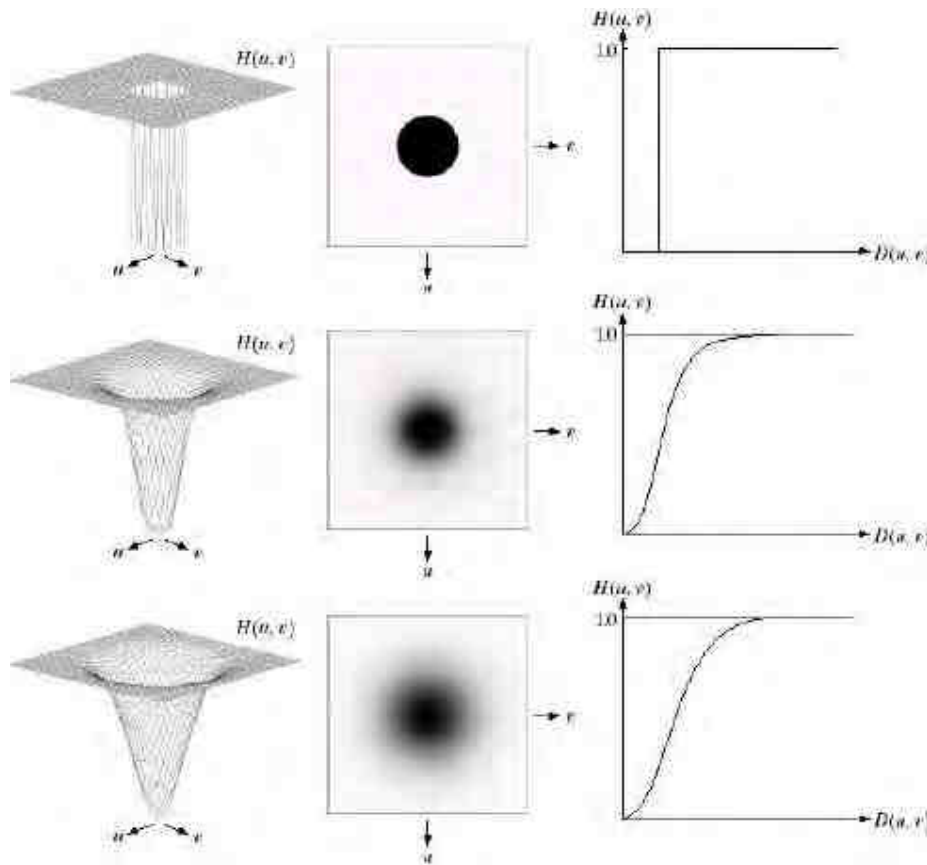
### SHARPENING FREQUENCY DOMAIN FILTERS

Image sharpening can be achieved by a high pass filtering process, which attenuates the low-

frequency components without disturbing high-frequency information. These are radially symmetric and completely specified by a cross section.

If we have the transfer function of a low pass filter the corresponding high pass filter can be obtained using the equation

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



## IDEAL HIGH PASS FILTER

This filter is opposite of the Ideal Low Pass filter and has the transfer function of the form

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

## BUTTERWORTH HIGH PASS FILTER

The transfer function of Butterworth High Pass filter of order  $n$  is given by the equation

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

#### GAUSSIAN HIGH PASS FILTER

The transfer function of a Gaussain High Pass Filter is given by the equation

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

## Homomorphic Filtering

Homomorphic filters are widely used in image processing for compensating the effect of non uniform illumination in an image. Pixel intensities in an image represent the light reflected from the corresponding points in the objects. As per an image model, image  $f(x,y)$  may be characterized by two components: (1) the amount of source light incident on the scene being viewed, and (2) the amount of light reflected by the objects in the scene. These portions of light are called the illumination and reflectance components, and are denoted  $i(x,y)$  and  $r(x,y)$  respectively. The functions  $i(x,y)$  and  $r(x,y)$  combine multiplicatively to give the image function  $f(x,y)$ :

$$f(x,y) = i(x,y) \cdot r(x,y) \quad (1)$$

where  $0 < i(x,y) < a$  and  $0 < r(x,y) < 1$ . Homomorphic filters are used in such situations where the image is subjected to the multiplicative interference or noise as depicted in Eq. 1. We cannot easily use the above product to operate separately on the frequency components of illumination and reflection because the Fourier transform of  $f(x,y)$  is not separable; that is

$$F[f(x,y)] \text{ not equal to } F[i(x,y)] \cdot F[r(x,y)].$$

We can separate the two components by taking the logarithm of the two sides

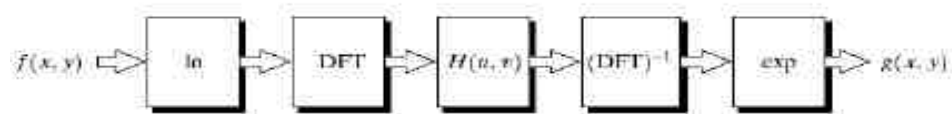
$$\ln f(x,y) = \ln i(x,y) + \ln r(x,y).$$

Taking Fourier transforms on both sides we get,

$$F[\ln f(x,y)] = F[\ln i(x,y)] + F[\ln r(x,y)].$$

that is,  $F(x,y) = I(x,y) + R(x,y)$ , where  $F$ ,  $I$  and  $R$  are the Fourier transforms  $\ln f(x,y)$ ,  $\ln i(x,y)$ , and  $\ln r(x,y)$  respectively. The function  $F$  represents the Fourier transform of the sum of two images: a low-frequency illumination image and a high-frequency reflectance image. If we now apply a filter with a transfer function that suppresses low-frequency components and enhances high-frequency components, then we can suppress the illumination component and enhance the reflectance component. Taking the inverse transform of  $F(x,y)$  and then anti-logarithm, we get

$$\tilde{f}(x,y) = \tilde{i}(x,y) + \tilde{r}(x,y)$$



$$\tilde{f}(x,y) = \tilde{i}(x,y) + \tilde{r}(x,y)$$



# IMAGE RESTORATION

## 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.

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:

- a) During display mode
- b) Acquisition mode.
- c) Processing mode.

The degradations may be due to

- a) Sensor noise
- b) Blur due to camera mis focus
- c) Relative object-camera motion
- d) Random atmospheric turbulence
- e) Others

### A Model of Image Restoration Process

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  $\eta(x,y)$ . 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 term  $\eta(x,y)$ , 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  $\eta$ , 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

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

$h(x,y)$  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.

The image restoration process can be achieved by inverting the image degradation process, i.e.,

Where  $1/H(u,v)$  is the inverse filter, and  $B(u,v)$  is the recovered image. Although the concept is relatively simple, the actual implementation is difficult to achieve, as one requires prior knowledge or identifications of the unknown degradation function  $h(x,y)$  and the unknown noise source  $n(x,y)$ .

In the following sections, common noise models and method of estimating the degradation function are presented

### 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:

- 1The noise model is spatial invariant, i.e., independent of spatial location.
- 2The noise model is uncorrelated with the object function.

### I. Gaussian Noise

These noise models are used frequently in practices because of its tractability in both spatial and frequency domain.

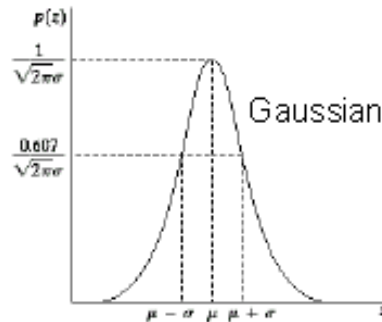
The PDF of Gaussian random variable,  $z$  is given by

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)$$

$z$  = gray level

$\mu$  = mean of average value of  $z$

$\sigma$  = standard deviation



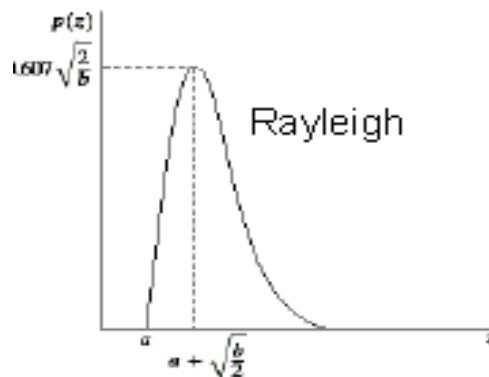
### .2 Rayleigh Noise

Unlike Gaussian distribution, the Rayleigh distribution is not symmetric. It is given by the formula.

$$p(z) = \frac{z}{b^2} \exp\left(-\frac{z^2}{b^2}\right) \quad \text{for } z \geq 0$$

The mean and variance of this density

Mean/variance
$\mu = a + \sqrt{\pi b^2 / 4}$
$\sigma^2 = \frac{b^2(4 - \pi)}{4}$



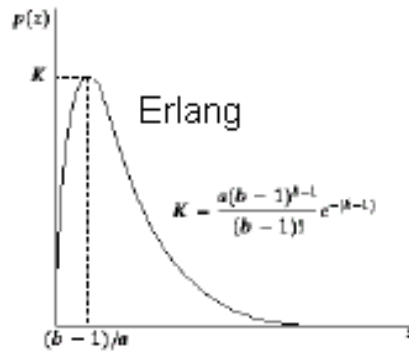
### 3 Erlang (gamma) Noise

The PDF of Erlang noise is given by

$$p(z) = \frac{1}{(k-1)! b^k} z^{k-1} \exp\left(-\frac{z}{b}\right)$$

The mean and variance of this noise is

Mean	$\mu = \frac{b}{a}$
Variance	$\sigma^2 = \frac{b}{a^2}$



Its shape is similar to Rayleigh distribution.

This equation is referred to as gamma density it is correct only when the denominator is the gamma function.

#### 4. Exponential Noise

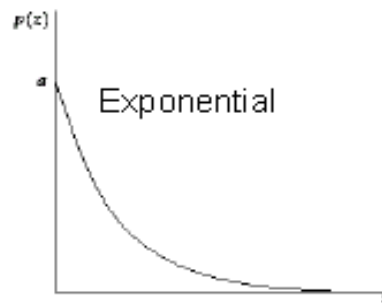
Exponential distribution has an exponential shape.

The PDF of exponential noise is given as

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

Where  $a > 0$

It is a special case of Erlang with  $b=1$



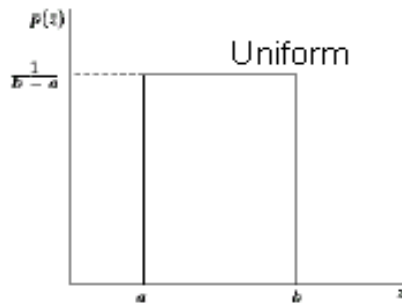
#### 5. Uniform Noise

The PDF of uniform noise is given by

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

The mean of this density function is given by

Mean	$\mu = \frac{a+b}{2}$
Variance	$\sigma^2 = \frac{(b-a)^2}{12}$

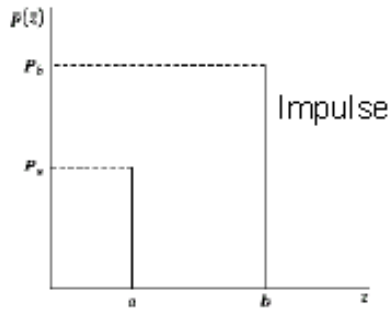


## 6. Impulse (Salt and Pepper )Noise

In this case, the noise is signal dependent, and is multiplied to the image.  
The PDF of bipolar (impulse) noise is given by

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

If  $b > a$ , gray level  $b$  will appear as a light dot in image.  
Level  $a$  will appear like a dark dot.



## Restoration In the Presence of Noise Only-Spatial Filtering

When the only degradation present in an image is noise, i.e.

$$\begin{aligned} g(x,y) &= f(x,y) + n(x,y) \\ \text{or} \\ G(u,v) &= F(u,v) + N(u,v) \end{aligned}$$

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

#### Arithmetic Mean Filter

It is the simplest mean filter. Let  $S_{xy}$  represents the set of coordinates in the sub image of size  $m \times n$  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  $S_{xy}$ . 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  $S_{xy}$ .

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

This operation can be using a convolution mask in which all coefficients have value 1/mn

A mean filter smoothes 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  $1/(MN)$ ). This will result 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 subimage window, raised to the power 1/mn. A geometric mean filter but it loses image details in the process.

Harmonic mean filter

The harmonic mean filtering operation is given by the expression

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

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 the Neighborhood of the pixel.

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

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 than smoothing filters of similar size. These are effective for bipolar and unipolar impulse noise.

Max and Min Filters

Using the 100th percentile of ranked set of numbers is called the max filter and is given by the equation

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

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 0<sup>th</sup> percentile filter is min filter

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

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

a. Midpoint Filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter

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

It combines the order statistics and averaging. This filter works best for randomly distributed noise like Gaussian or uniform noise.

### 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 expression

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

D(u, v)- the distance from the origin of the centered frequency rectangle.

W- the width of the band

D<sub>0</sub>- the radial center of the frequency rectangle.

Butterworth Band reject Filter

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

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 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  $H_{br}(u,v)$  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

This type of filters rejects frequencies in predefined neighborhood above a centre frequency. These filters are symmetric about origin in the Fourier transform. The transfer function of ideal notch reject filter of radius  $D_0$  with centre at  $(u_0, v_0)$  and by symmetry at  $(-u_0, -v_0)$  is

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

Where

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

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

Butterworth notch reject filter of order  $n$  is given by

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

A Gaussian notch reject filter has the formula

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

These filters become high pass rather than suppress. The frequencies contained in the notch areas. These filters will perform exactly the opposite function as the notch reject filter.

The transfer function of this filter may be given as

$$H_{HP}(u,v) = 1 - H_{NR}(u,v)$$

$H_{BP}(u,v)$ - transfer function of the pass filter

$H_{NR}(u,v)$ - transfer function of a notch reject filter

## Minimum Mean Square Error (Wiener) Filtering

This filter incorporates both degradation function and statistical behavior of noise into the restoration process.

The main concept behind this approach is that the images and noise are considered as random variables and the objective is to find an estimate  $\hat{f}$  of the uncorrupted image  $f$  such that the mean sequence error between them is minimized.

$$\hat{f}(x) = \sum_{s=-\infty}^{\infty} h_w(x-s)g(s),$$

This error measure is given by

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

Where  $E(\cdot)$  is the expected value of the argument

Assuming that the noise and the image are uncorrelated (means zero average value) one or other has zero mean values

The minimum error function of the above expression is given in the frequency domain. It is given by the expression.

$$H_w(u, v) = \frac{H^*(u, v) S_g(u, v)}{|H(u, v)|^2 S_g(u, v) + S_m(u, v)} = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_m(u, v) / S_g(u, v)}$$

Product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. This result is known as Wiener Filter. The filter was named so because of the name of its inventor N Wiener. The term in the bracket is known as minimum mean square error filter or least square error filter.

$H^*(u, v)$  - degradation function.

$H^*(u, v)$  - complex conjugate of  $H(u, v)$

$H(u, v)$  - Fourier transform of the degraded image

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

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

$H(u, v)$  - Fourier transform of the degraded function

$G(u, v)$  - Fourier transform of the degraded image

The restored image in the spatial domain is given by the inverse Fourier transform of the frequency domain estimate  $F(u, v)$ .

Mean square error in statistical form can be improved by the function

$$H_w(u, v) = \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K}$$

## Inverse Filtering

It is a process of restoring an image degraded by a degradation function  $H$ . This function can be obtained by any method.

The simplest approach to restoration is direct, inverse filtering.

Inverse filtering provides an estimate  $F(u, v)$  of the transform of the original image simply by dividing the transform of the degraded image  $G(u, v)$  by the degradation function.



It shows an interesting result that even if we know the degradation function we cannot recover the undegraded image exactly because  $N(u,v)$  is not known.

If the degradation value has zero or very small values then the ratio  $N(u,v)/H(u,v)$  could easily dominate the estimate  $F(u,v)$ .

## IMAGE COMPRESSION

### Digital Image Compression

#### Data Compression and Data Redundancy

Data compression is defined as the process of encoding data using a representation that reduces the overall size of data. This reduction is possible when the original dataset contains some type of redundancy. Digital image compression is a field that studies methods for reducing the total number of bits required to represent an image. This can be achieved by eliminating various types of redundancy that exist in the pixel values. In general, three basic redundancies exist in digital images that follow.

**Psycho-visual Redundancy:** It is a redundancy corresponding to different sensitivities to all image signals by human eyes. Therefore, eliminating some less relative important information in our visual processing may be acceptable.

**Inter-pixel Redundancy:** It is a redundancy corresponding to statistical dependencies among pixels, especially between neighboring pixels.

**Coding Redundancy:** The uncompressed image usually is coded with each pixel by a fixed length. For example, an image with 256 gray scales is represented by an array of 8-bit integers. Using some variable length code schemes such as Huffman coding and arithmetic coding may produce compression. There are different methods to deal with different kinds of aforementioned redundancies. As a result, an image compressor often uses a multi-step algorithm to reduce these redundancies.

#### Compression Methods

During the past two decades, various compression methods have been developed to address major challenges faced by digital imaging.<sup>3, 10</sup> These compression methods can be classified broadly into lossy or lossless compression. Lossy compression can achieve a high compression ratio, 50:1 or higher, since it allows some acceptable degradation. Yet it cannot completely recover the original data. On the other hand, lossless compression can completely recover the original data but this reduces the compression ratio to around 2:1. In medical applications, lossless compression has been a requirement because it facilitates accurate diagnosis due to no degradation on the original image. Furthermore, there exist several legal and regulatory issues that favor lossless compression in medical applications.<sup>11</sup>

## Lossy Compression Methods

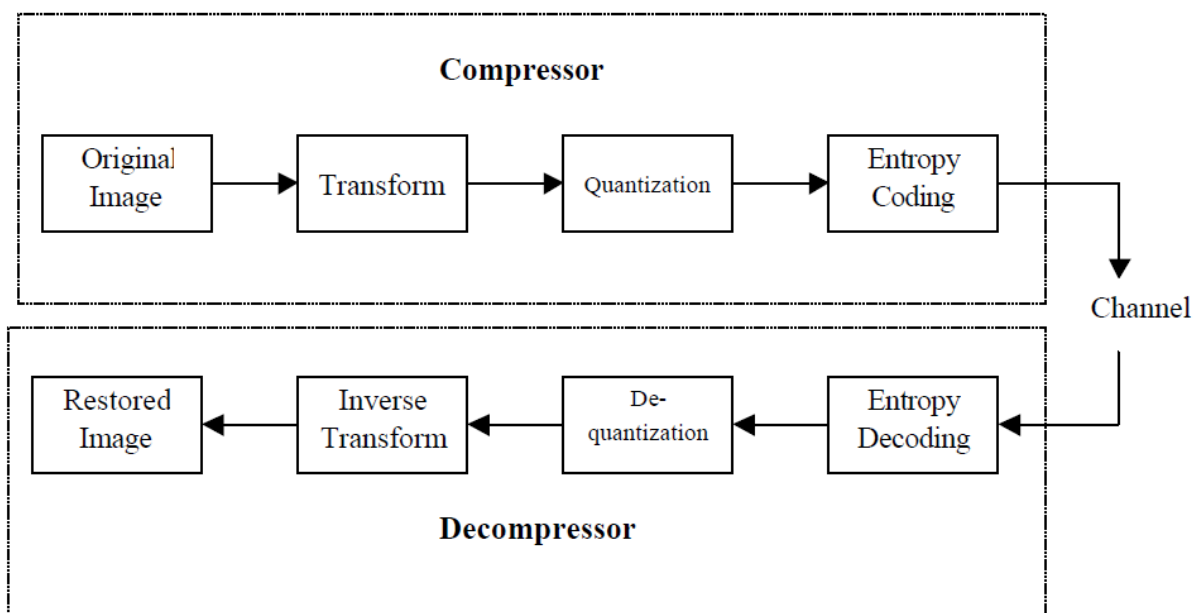


Figure 2.1 Lossy image compression

Generally most lossy compressors (Figure 2.1) are three-step algorithms, each of which is in accordance with three kinds of redundancy mentioned above. Figure 2.1 Lossy image compression The first stage is a transform to eliminate the inter-pixel redundancy to pack information efficiently. Then a quantizer is applied to remove psycho- redundancy to represent the packed information with as few bits as possible. The quantized bits are then efficiently encoded to get more compression from the coding redundancy.

### 2.2.1.1 Quantization

Quantization is a many-to-one mapping that replaces a set of values with only one representative value. Scalar and vector quantization are two basic types of quantization. SQ (scalar quantization) performs many-to-one mapping on each value. VQ (vector quantization) replaces each block of input pixels with the index of a vector in the codebook, which is close to the input vector by using some closeness measurements. The decoder simply receives each index and looks up the corresponding vector in the codebook. Shannon<sup>12</sup> first showed that VQ would result in a lower bit rate than SQ. But VQ suffers from a lack of generality, since the codebook must be trained on some set of initial images. As a result, the design of the codebook will directly affect the bit rate and distortion of the compression. Riskin et. al.<sup>5</sup> presented variable-rate VQ design

and applied it to MR images. Cosman et. al.<sup>13</sup> used similar methods to compress CT and MR chest scans. Xuan et.

al.<sup>14</sup> also used similar VQ techniques to compress mammograms and brain MRI.

### **2.2.1.2 Transform Coding**

Transform coding is a general scheme for lossy image compression. It uses a reversible and linear transform to decorrelate the original image into a set of coefficients in transform domain. The coefficients are then quantized and coded sequentially in transform domain.<sup>7</sup> Numerous transforms are used in a variety of applications. The discrete KLT (Karhunen-Loeve transform), which is based on the Hotelling transform, is optimal with its information packing properties, but usually not practical since it is difficult to compute.<sup>15,16</sup> The DFT (discrete Fourier transform) and DCT (discrete cosine transform) approximate the energy-packing efficiency of the KLT, and have more efficient implementation. In practice, DCT is used by most practical transform systems since the DFT coefficients require twice the storage space of the DCT coefficients.

### **Block Transform Coding**

In order to simplify the computations, block transform coding exploits correlation of the pixels within a number of small blocks that divide the original image. As a result, each block is transformed, quantized and coded separately. This technique, using square 8\*8 pixel blocks and the DCT followed by Huffman or arithmetic coding, is utilized in the ISO JPEG (joint photographic expert group) draft international standard for image compression.<sup>17-19</sup> The disadvantage of this scheme is the blocking (or tiling) artifacts appear at high compression ratios. Since the adoption of the JPEG standard, the algorithm has been the subject of considerable research. Collins et. al.<sup>20</sup> studied the effects of a 10:1 lossy image compression scheme based on JPEG, with modifications to reduce the blocking artifacts. Baskurt et. al.<sup>21</sup> used an algorithm similar to JPEG to compress mammograms with a bit rate as low as 0.27 bpp (bits per pixel) while retaining detection ability of pathologies by radiologists. Kostas et. al.<sup>22</sup> used JPEG modified for use with 12-bit images and custom quantization tables to compress mammograms and chest radiographs. Moreover, the ISO JPEG committee is currently developing a new still-image compression standard called JPEG-2000 for delivery to the marketplace by the end of the year 2000. The new JPEG-2000 standard is based upon wavelet decompositions combined with more powerful quantization and encoding strategies such as embedded quantization and context-based arithmetic. It provides the potential for numerous

advantages over the existing JPEG standard. Performance gains include improved compression efficiency at low bit rates for large images, while new functionalities include multi-resolution representation, scalability and embedded bit stream architecture, lossy to lossless progression, ROI (region of interest) coding, and a rich file format.<sup>23</sup>

### **Full-Frame Transform Coding**

To avoid the artifacts generated by block transforms, full-frame methods, in which the transform is applied to the whole image as a single block, have been investigated in medical imaging research.<sup>24-26</sup> The tradeoff is the increased computational requirements and the appearance of ringing artifacts (a periodic pattern due to the quantization of high frequencies).

Subband coding is one example among full-frame methods. It will produce a number of sub-images with specific properties such as a smoothed version of the original plus a set of images with the horizontal, vertical, and diagonal edges that are missing from the smoothed version according to different frequencies.<sup>27-29</sup> Rompelman<sup>30</sup> applied subband coding to compress 12-bit CT images at rates of 0.75 bpp and 0.625 bpp without significantly affecting diagnostic quality. Recently, much research has been devoted to the DWT (discrete wavelet transform) for subband

coding of images. DWT is a hierarchical subband decomposition particularly suited to image compression.<sup>31</sup> Many different wavelet functions can be applied to different applications. In general, more complicated wavelet functions provide better performance. The wavelet transform can avoid the blocking artifacts presented in block transform methods and allow easy progressive coding due to its multiresolution nature. Bramble et. al.<sup>32</sup> used full-frame Fourier transform compression on 12 bpp digitized hand radiographs at average rates from about 0.75 bpp to 0.1 bpp with no significant degradation in diagnostic quality involving the detection of pathology characterized by a lack of sharpness in a bone edge. However, Cook et. al.<sup>33</sup> investigated the effects of full-frame DCT compression on low-contrast detection of chest lesions and found significant degradation at rates of about 0.75 bpp. These results illustrate that both imaging modality and the task play an important role in determining achievable compression.

### 2.2.2 Lossless Compression Methods

Lossless compressors (Figure 2.2) are usually two-step algorithms. The first step transforms the original image to some other format in which the inter-pixel redundancy is reduced. The second step uses an entropy encoder to remove the coding redundancy. The lossless decompressor is a perfect inverse process of the lossless compressor.

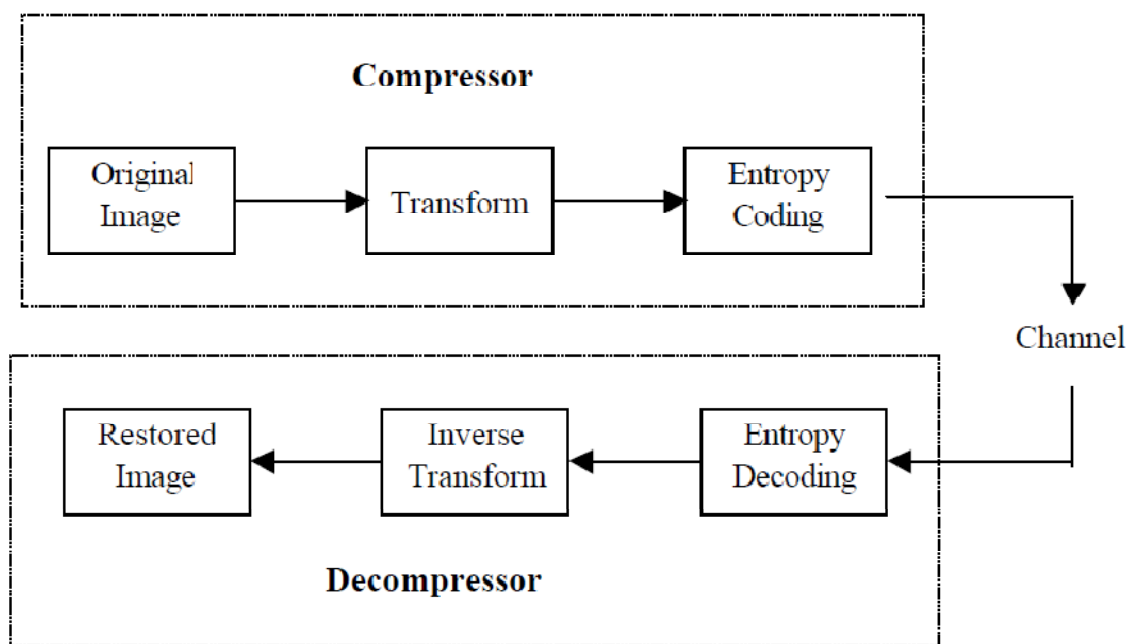


Figure 2.2 Lossless image compression

Typically, medical images can be compressed losslessly to about 50% of their original size. Boncelet et. al.<sup>34</sup> investigated the use of three entropy coding methods for lossless compression with an application to digitized radiographs and found that a bit rate of about 4 to 5 bpp was best. Tavakoli<sup>35, 36</sup> applied various lossless coding techniques to MR images and reported a compression down to about 5 to 6 bpp, with LZ (Lempel-Ziv) coding achieving the best results. Lossless compression works best with decorrelated data. Roose et. al.<sup>5, 37</sup> investigated prediction, linear transformation, and multiresolution methods for decorrelating medical image data before

coding them. The compression result was 3:1 and less than 2:1 for angiograms and MRI respectively. Kuduvalli and Rangayyan<sup>6</sup> studied similar techniques and found linear prediction and interpolation techniques gave the best results with similar compression ratios.

Here, we summarize the lossless compression methods into four categories.

#### **2.2.2.1 Run Length Coding**

Run length coding replaces data by a (length, value) pair, where `value\_` is the repeated value and `length\_` is the number of repetitions. This technique is especially successful in compressing bi-level images since the occurrence of a long run of a value is rare in ordinary gray-scale images. A solution to this is to decompose the gray-scale image into bit planes and compress every bit-plane separately. Efficient run-length coding method<sup>38</sup> is one of the variations of run length coding.

#### **2.2.2.2 Lossless Predictive Coding**

Lossless predictive coding predicts the value of each pixel by using the values of its neighboring pixels. Therefore, every pixel is encoded with a prediction error rather than its original value. Typically, the errors are much smaller compared with the original value so that fewer bits are required to store them.

DPCM (differential pulse code modulation) is a predictive coding based lossless image compression method. It is also the base for lossless JPEG compression. A variation of the lossless predictive coding is the adaptive prediction that splits the image into blocks and computes the prediction coefficients independently for each block to achieve high prediction performance. It can also be combined with other methods to get a hybrid coding algorithm with higher performance.<sup>14, 39</sup>

#### **Entropy Coding**

Entropy represents the minimum size of dataset necessary to convey a particular amount of information. Huffman coding, LZ (Lempel-Ziv) coding and arithmetic coding are the commonly used entropy coding schemes. Huffman coding utilizes a variable length code in which short code words are assigned to more common values or symbols in the data, and longer code words are assigned to less frequently occurring values. Modified Huffman coding<sup>40</sup> and dynamic Huffman coding<sup>41</sup> are two examples among many variations of Huffman's technique. LZ coding replaces repeated substrings in the input data with references to earlier instances of the strings. It often refers to two different approaches to dictionary-based

compression: the LZ77<sup>42</sup> and the LZ78<sup>43</sup>. LZ77 utilizes a sliding window to search for the substrings encountered before and then substitutes them by the (position,length) pair to point back to the existing substring. LZ78 dynamically constructs a dictionary from the input file and then replaces the substrings by the index in the dictionary. Several compression methods, among which LZW (Lempel-Ziv-Welch)<sup>44</sup> is one of the most well known methods, have been developed based on these ideas. Variations of LZ coding are used in the Unix utilities Compress and Gzip. Arithmetic coding<sup>45</sup> represents a message as some finite intervals between 0 and 1 on the real number line. Basically, it divides the intervals between 0 and 1 into a number of smaller intervals corresponding to the probabilities of the message's symbols. Then the first input symbol selects an interval, which is further divided into smaller intervals. The next input symbol selects one of these intervals, and the procedure is repeated. As a result, the selected interval narrows with every symbol, and in the end, any number inside the final interval can be used to represent the message. That is to say, each bit in the output code refines the precision of the value of the input code in the interval. A variation of arithmetic coding is the Q-coder<sup>46</sup>, developed by IBM in the late 1980's. Two references are provided for the latest Q-coder variation.

#### 2.2.2.4 Multiresolution Coding

HINT (hierarchical interpolation)<sup>5, 37</sup> is a multiresolution coding scheme based on sub-samplings. It starts with a low-resolution version of the original image, and interpolates the pixel values to successively generate higher resolutions. The errors between the interpolation values and the real values are stored, along with the initial low-resolution image. Compression is achieved since both the low-resolution image and the error values can be stored with fewer bits than the original image. Laplacian Pyramid<sup>49</sup> is another multiresolution image compression method developed by Burt and Adelson. It successively constructs lower resolution versions of the original image by down sampling so that the number of pixels decreases by a factor of two at each scale. The differences between successive resolution versions together with the lowest resolution image are stored and utilized to perfectly reconstruct the original image. But it cannot achieve a high compression ratio because the number of data values is increased by 4/3 of the original image size. In general, the image is reversibly transformed into a group of different resolution sub-images in multiresolution coding. Usually, it reduces the entropy of the image. Some kinds of tree representation could be used to get more compression by exploiting the tree structure of the multiresolution methods.<sup>50</sup>

### Measurements for Compression Methods

#### Measurements for Lossy Compression Methods

Lossy compression methods result in some loss of quality in the compressed images. It is a tradeoff between image distortion and the compression ratio. Some distortion measurements are often used to quantify the quality of the reconstructed image as well as the compression ratio (the ratio of the size of the original image to the size of the compressed image). The commonly used objective distortion measurements, which are derived from statistical terms, are the RMSE (root mean square error), the NMSE (normalized mean square error) and the PSNR (peak signal-to-noise ratio). These measurements are defined as follows:

$$RMSE = \sqrt{\frac{1}{N * M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i, j) - f'(i, j)]^2}$$
$$NMSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i, j) - f'(i, j)]^2}{[\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i, j)]^2}$$
$$PSNR = 20 * \log_{10} \left( \frac{255}{RMSE} \right)$$

where the images have  $N \times M$  pixels (8 bits per pixel),  $f(i, j)$  represents the original image, and  $\hat{f}(i, j)$  represents the reconstructed image after compression and decompression. Since the images are for human viewing, it leads to subjective measurements based on subjective comparisons to tell how 'good' the decoded image looks to a human viewer. Sometimes, application quality can be used as a measure to classify the usefulness of the decoded image for a particular task such as clinical diagnosis of medical images and meteorological prediction in satellite images and so on. When comparing two lossy coding methods, we may either compare the qualities of images reconstructed at a constant bit rate, or, equivalently, we may compare the bit rates used in two constructions with the same quality, if it is accomplishable.

### **Measurements for Lossless Compression Methods**

Lossless compression methods result in no loss in the compressed images so that it can perfectly restore the original images when applying a reversible process. The frequently used measurement in lossless compression is the compression ratio. This measurement can be misleading, since it depends on the data storage format and sampling density. For instance, medical images containing 12 bits of useful information per pixel are often stored using 16 bpp.

A better measurement of compression is the bit rate due to its independence of the data storage format. A bit rate measures the average number of bits used to represent each pixel of the image in a compressed form. Bit rates are measured in bpp, where a lower bit rate corresponds to a greater amount of compression.

### **Summary**

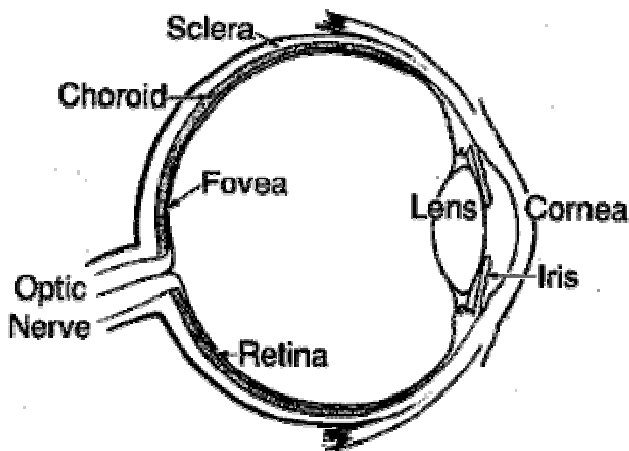
Digital image compression has been the focus of a large amount of research in recent years. As a result, data compression methods grow as new algorithms or variations of the already existing ones are introduced. All these digital image compression methods are concerned with minimization of the amount of information used to represent an image. They are based on the same principles and on the same theoretical compression model, which effectively reduces three types of redundancy, such as psycho-visual, inter-pixel and coding, inherited in gray-level images. However, a 3-D medical image set contains an additional type of redundancy, which is not often addressed by the current compression methods. Several methods that utilize dependencies in all three dimensions have been proposed. Some of these methods used the 3-D DWT in a lossy compression scheme, whereas others used predictive coding in a lossless scheme. In the latest paper,<sup>58</sup> 3D CBEZW (context-based embedded zerotree wavelet) algorithm was proposed to efficiently encode 3-D image data by the exploitation of the dependencies in all dimensions, while enabling lossy and lossless decompression from the same bit stream. In this proposal, we first introduce a new type of redundancy existing among pixels values in all three dimensions from a new point of view and its basic characteristics. Secondly, we propose a novel lossless compression method based on integer wavelet transforms, embedded zerotree and predictive coding to reduce this special redundancy to gain more compression. Thirdly, we expand the proposed compression method to the application of the telemedicine to support the transmission of the ROI without any diagnostic information loss and the simple diagnosis of certain disease such as multiple sclerosis in MR brain images.

# COLOR IMAGE FUNDAMENTALS

## Perception

Many image processing applications are intended to produce images that are to be viewed by human observers. It is therefore important to understand the characteristics and limitations of the human visual system to understand the 'receiver' of the 2D signals. At the outset it is important to realise that (1) human visual system (HVS) is not well understood; (2) no objective measure exists for judging the quality of an image that corresponds to human assessment of image quality, and (3) the typical human observer does not exist. Nevertheless, research in perceptual psychology has provided some important insights into the visual system [stock ham].

### Elements of Human Visual Perception.



### The human eye

The first part of the visual system is the eye. This is shown in figure . Its form is nearly spherical and its diameter is approximately 20 mm. Its outer cover consists of the 'cornea' and 'sclera'

The cornea is a tough transparent tissue in the front part of the eye. The sclera is an opaque membrane, which is continuous with cornea and covers the remainder of the eye. Directly below the sclera lies the 'choroids', which has many blood vessels. At its anterior extreme lies the iris diaphragm. The light enters in the eye through the central opening of the iris, whose diameter varies from 2mm to 8mm, according to the illumination conditions. Behind the iris is the 'lens', which consists of concentric layers of fibrous cells and contains up to 60 to 70% of water. Its operation is similar to that of the man made optical lenses. It focuses the light on the 'retina', which is the innermost membrane of the eye.

Retina has two kinds of photoreceptors: cones and rods. The cones are highly sensitive to color. Their number is 6-7 million and they are mainly located at the central part of the retina. Each cone is connected to one nerve end.

Cone vision is the photopic or bright light vision. Rods serve to view the general picture of the vision field. They are sensitive to low levels of illumination and cannot discriminate colors. This is the scotopic or dim-light vision. Their number is 75 to 150 million and they are distributed over the retinal surface.



Several rods are connected to a single nerve end. This fact and their large spatial distribution explain their low resolution.

Both cones and rods transform light to electric stimulus, which is carried through the optical nerve to the human brain for the high level image processing and perception.

## Model of the Human Eye

Based on the anatomy of the eye, a model can be constructed as shown in Figure(2.2). Its first part is a simple optical system consisting of the cornea, the opening of iris, the lens and the fluids inside the eye. Its second part consists of the retina, which performs the photo electrical transduction, followed by the visual pathway (nerve) which performs simple image processing operations and carries the information to the brain.



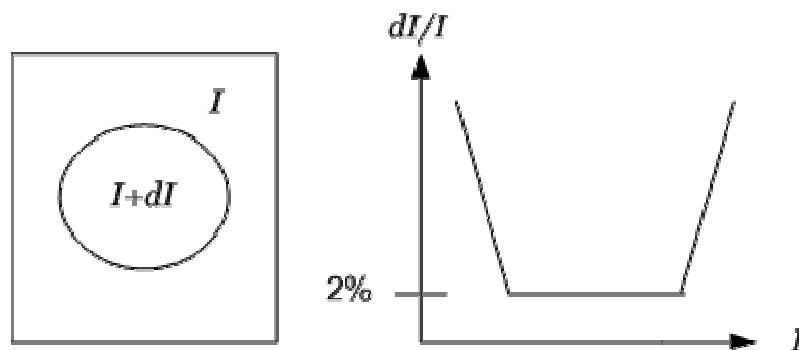
A model of the human eye.

## Image Formation in the Eye.

The image formation in the human eye is not a simple phenomenon. It is only partially understood and only some of the visual phenomena have been measured and understood. Most of them are proven to have non-linear characteristics.

Two examples of visual phenomena are: **Contrast sensitivity** , **Spatial Frequency Sensitivity**

### Contrast sensitivity



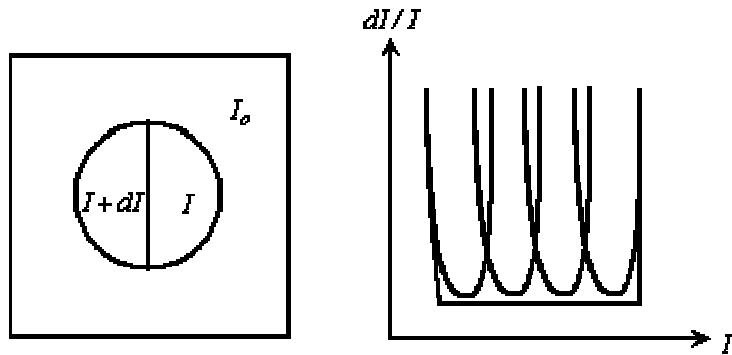
The Weber ratio without background

Let us consider a spot of intensity  $I+dI$  in a background having intensity  $I$ , as is shown in Figure ;  $dI$  is increased from 0 until it becomes noticeable. The ratio  $dI/I$ , called Weber ratio, is nearly constant at about 2% over a wide range of illumination levels, except for very low or very high illuminations, as it is seen in Figure .The range over which the Weber ratio remains constant is reduced considerably, when the experiment of Figure .is considered. In this case, the background has intensity  $I_0$  and two adjacent spots have intensities  $I$  and  $I+dI$ , respectively. The Weber ratio is plotted as a function of the background

intensity in Figure (2.4). The envelope of the lower limits is the same with that of Figure. The derivative of the logarithm of the intensity  $I$  is the Weber ratio:

$$d[\log(I)] = \frac{dI}{I}$$

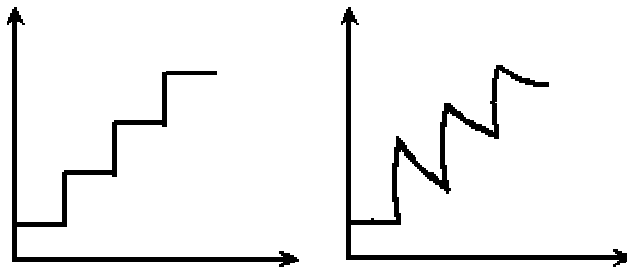
Thus equal changes in the logarithm of the intensity result in equal noticeable changes in the intensity for a wide range of intensities. This fact suggests that the human eye performs a pointwise logarithm operation on the input image.



### The Weber ratio with background

Another characteristic of HVS is that it tends to `overshoot\_ around image edges (boundaries of regions having different intensity). As a result, regions of constant intensity, which are close to edges, appear to have varying intensity. Such an example is shown in Figure .The stripes appear to have varying intensity along the horizontal dimension, whereas their intensity is constant. This effect is called *Mach band effect*. It indicates that the human eye is sensitive to edge information and that it has high-pass characteristics.





**The Mach-band effect:**

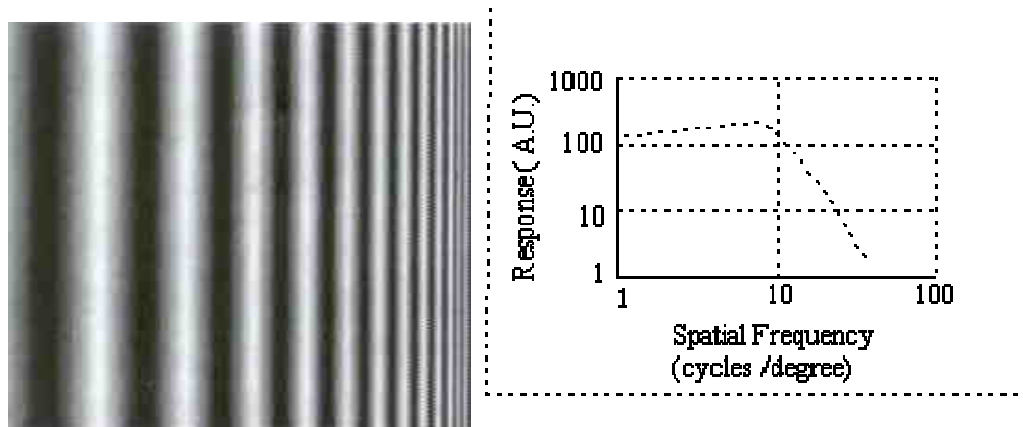
**(a) Vertical stripes having constant illumination;**

**(b) Actual image intensity profile;**

**(c) Perceived image intensity profile.**

### **Spatial Frequency Sensitivity**

If the constant intensity (brightness)  $I_0$  is replaced by a sinusoidal grating with increasing spatial frequency, it is possible to determine the spatial frequency sensitivity. The result is shown in above Figure



**Figure shows Sinusoidal test grating ; spatial frequency sensitivity**

To translate these data into common terms, consider an `ideal` computer monitor at a viewing distance of 50 cm. The spatial frequency that will give maximum response is at 10 cycles per degree. (See figure above) The one degree at 50 cm translates to  $50 \tan(1 \text{ deg.}) = 0.87 \text{ cm}$  on the computer screen. Thus the spatial frequency of maximum response  $f_{max} = 10 \text{ cycles} / 0.87 \text{ cm} = 11.46 \text{ cycles/cm}$  at this viewing distance. Translating this into a general formula gives:

$$f_{\max} = \frac{10}{d * \tan(1^\circ)} = \frac{572.9}{d} \text{ cycles/cm}$$

where  $d$ =viewing distance measured in cm

### Definition of color:

Light is a form of electromagnetic (em) energy that can be completely specified at a point in the image plane by its wavelength distribution. Not all electromagnetic radiation is visible to the human eye. In fact, the entire visible portion of the radiation is only within the narrow wavelength band of 380 to 780 nms. Till now, we were concerned mostly with light intensity, i.e. the sensation of brightness produced by the aggregate of wavelengths. However light of many wavelengths also produces another important visual sensation called `color`. Different spectral distributions generally, but not necessarily, have different perceived color. Thus color is that aspect of visible radiant energy by which an observer may distinguish between different spectral compositions.

A color stimulus therefore specified by visible radiant energy of a given intensity and spectral composition. Color is generally characterised by attaching names to the different stimuli e.g. white, gray, black, red, green, blue. Color stimuli are generally more pleasing to eye than `black and stimuli`. Consequently pictures with color are widespread in TV photography and printing.

Color is also used in computer graphics to add `spice` to the synthesized pictures. Coloring of black and white pictures by transforming intensities into colors (called pseudo colors) has been extensively used by artist's working in pattern recognition. In this module we will be concerned with questions of how to specify color and how to reproduce it. Color specification consists of 3 parts:

- (1) Color matching
- (2) Color differences
- (3) Color appearance or perceived color

## Representation of color for human vision

### Trichromacy of Vision Color Mixture

Let  $S(\lambda)$  denote the spectral power distribution (in watts /m<sup>2</sup>/unit wavelength) of the light emanating from a pixel of the image plane, and  $\lambda$  the wavelength. The human retina contains predominantly three different color receptors (called cones) that are sensitive to 3 overlapping areas of the visible spectrum. The sensitivities of the receptors peak at approximately 445. (Called blue), 535 (called green) and 570 (called red) nanometers.

Each type of receptors integrates the energy in the incident light at various wavelengths in proportion to their sensitivity to light at that wavelength. The three resulting numbers are primarily responsible for color sensation. This is the basis for trichromatic theory of color vision, which states that the color of light entering the eye may be specified by only 3 numbers, rather than a complete function of wavelengths over

the visible range. This leads to significant economy in color specification and reproduction for human viewing. Much of the credit for this significant work goes to the physicist Thomas Young.

The counterpart to trichromacy of vision is the Trichromacy of Color Mixture.

This important principle states that light of any color can be synthesized by an appropriate mixture of 3 properly chosen primary colors.

Maxwell in 1855 showed this using a 3-color projecting system. Several development took place since that time creating a large body of knowledge referred to as colorimetry.

Although trichromacy of color is based on subjective & physiological finding, these are precise measurements that can be made to examine color matches

## Color matching

Consider a bipartite field subtending an angle ( $\angle$ ) of  $2^\circ$  at a viewer's eye. The entire field is viewed against a dark, neutral surround. The field contains the test color on left and an adjustable mixture of 3 suitably chosen primary colors on the right as shown in Figure (2.7).

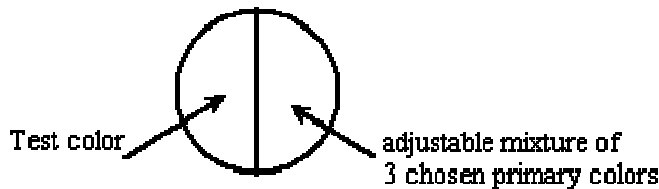


Figure :  $2^\circ$  bipartial field at view's eye

It is found that most test colors can be matched by a proper mixture of 3 primary colors as long as the primary colors are independent. The primary colors are usually chosen as red, green & blue or red, green & violet.

The `tristimulus values` of a test color are the amount of 3 primary colors required to give a match by additive mixture. They are unique within an accuracy of the experiment.

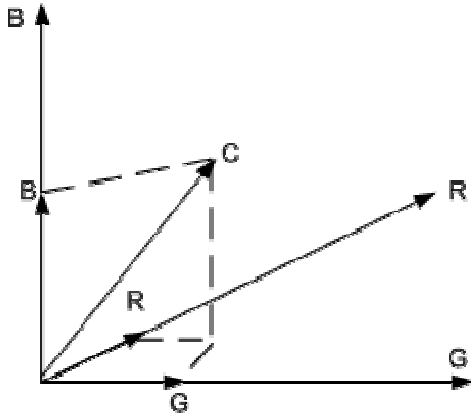
Much of colorimetry is based on experimental results as well as rules attributed to Grassman.

Two important rules that are valid over a large range of observing conditions are `linearity` and `additivity`. They state that,

1) The color match between any two color stimuli holds even if the intensities of the stimuli are increased or decreased by the same multiplying factor, as long as their relative spectral distributions remain unchanged.

As an example, if stimuli  $s_1(\lambda)$  and  $s_2(\lambda)$  match, and stimuli  $s_3(\lambda)$  and  $s_4(\lambda)$  also match, then additive mixtures  $(s_1(\lambda) + s_3(\lambda))$  and  $(s_2(\lambda) + s_4(\lambda))$  will also match.

2) Another consequence of the above rules of Grassman trichromacy is that any four colors cannot be linearly independent. This implies tristimulus value of one of the 4 colors can be expressed as linear combination of tristimulus values of remaining 3 colors.. That is, any color C is specified by its projection on 3-axes R, G, B corresponding to chosen set of primaries. This is shown in Figure 2.8



**Figure:  $R, G, B$  tristimulus space. A color  $C$  is specified by a vector in three-dimensional space with components  $R, G$  and  $B$  (tristimulus values.)**

Consider a mixture of two colors  $S_1$  and  $S_2$  i.e  $S = S_1 + S_2$

If  $S_1$  is specified by  $(R_{s1}, G_{s1}, B_{s1})$  and  $S_2$  is specified by  $(R_{s2}, G_{s2}, B_{s2})$

This implies,  $S$  is specified by  $(R_{s1} + R_{s2}, G_{s1} + G_{s2}, B_{s1} + B_{s2})$

The constraint of color matching experiment is that only non-ve amounts of primary colors can be added to match a test color. In practice this is not sufficient to effect a match. In this case, since negative amounts of primary cannot be produced, a match is made by simple transposition i.e. by adding positive amounts of primary to the test color

$\therefore$  a test color  $S$  might be matched by ,

$$S + 3G = 2R + B$$

or,  $S = 2R - 3G + B$

$\rightarrow$  The negative tristimulus values  $(2, -3, 1)$  present no special problem.

By convention, tristimulus values are expressed in normalized form. This is done by a preliminary color experiment in which left side of the split field shown in Fig (2.7), is allowed to emit light of unit intensity whose spectral distribution is constant wrt  $\lambda$  i.e. (equal energy white  $E$ ). Then the amount of each primary required for a match is taken by definition as one 'unit'.

The amount of primaries for matching other test colors is then expressed in terms of this unit. In practice equal energy white 'E' is matched with positive amounts of each primary.

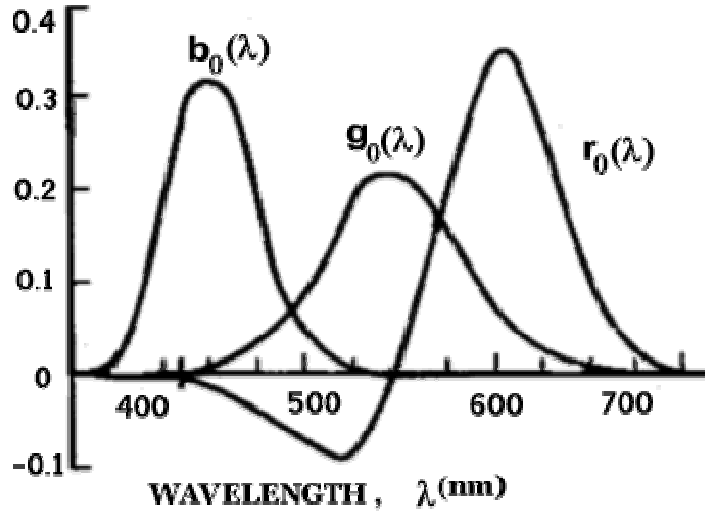


Figure: The color-matching functions for the 2° Standard Observer , using primaries of wavelengths 700(red), 546.1 (green), and 435.8 nm (blue), with units such that equal quantities of the three primaries are needed to match the equal energy white, E .

### Color-Coordinate Systems.

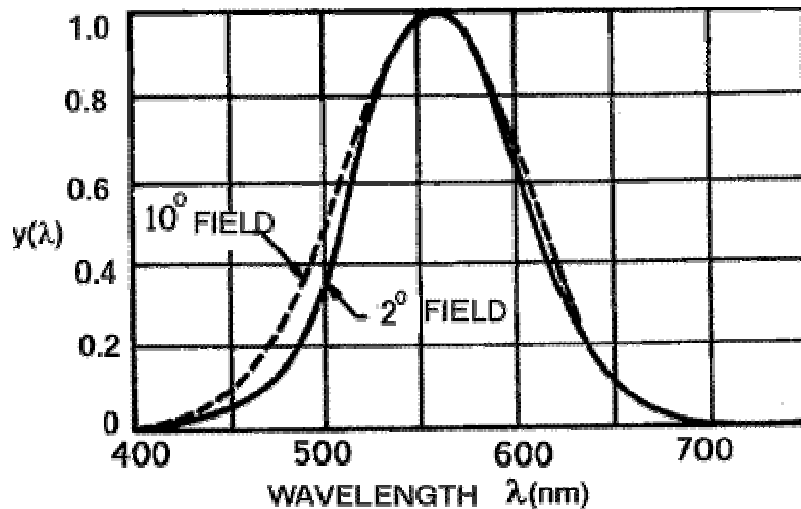
#### CHROMATICITY

Instead of specifying a color by its tristimulus values R, G, B colorimetrists use normalized quantities called chromaticity coordinates. These are expressed by,

$$r = \frac{R}{R+G+B} \quad \text{and} \quad b = \frac{B}{R+G+B} \quad g = G / R+G+B$$

Of course since  $r+g+b=1$ , two chromaticity coordinates are sufficient. This however leaves us with only two pieces of information. The third dimension of color is called the luminance (Y) which may be obtained by a separate match. Luminance is an objective measure of that aspect of radiant energy that produces the sensation of 'brightness'.

Radiation of different wavelengths contributes differently to the sensation of brightness. The relative contribution of monochromatic radiation of a given wavelength to luminance i.e. the brightness sensation is termed as the relative luminous efficiency  $y^{(\lambda)}$  . Since this is obtained by photometric matches i.e. matching of brightness, it is dependent on the condition of observations. Fig (2.11 ) shows the  $y^{(\lambda)}$  vs  $\lambda$  curve for 2° and 10° fields of view.



Both these curves are normalized such that maximum  $y(\lambda)$  is taken to be unity.

The luminance of any given spectral distribution  $S(\lambda)$  is then taken to be

$$\underline{Y} = K_m \int_{\lambda} S(\lambda) y(\lambda) d\lambda \quad \text{Candelas/meter}^2$$

where  $K_m = 680$  lumens/watt. As in color matches, a brightness match is observed between two spectral distributions,  $S_1(\lambda)$  and  $S_2(\lambda)$  if,

$$\int_{\lambda} S_1(\lambda) y(\lambda) d\lambda = \int_{\lambda} S_2(\lambda) y(\lambda) d\lambda$$

It is easy to see that the luminance of the sum of two spectral distributions is the sum of their luminances.

A complete specification of color given by luminance and chromaticities is often used since it is very close to familiar concepts defining perceived color.

### CIE System of Color Specification:

- Standard observer
- CIE primaries.

Another specification of color that is also popular was generated by CIE (commission International de L' Eclairage) an international body of color scientists in 1931.

#### □ Standard Observer



The CIE defined a 'standard observer' by averaging the color matching data of a large number of observers having normal color vision. This standard observed data consists of color matching functions for primary stimuli of wavelengths 700 ( $R_o$ ), 546-1( $G_o$ ) and 435-8( $B_o$ ) nm with units normalized in the standard way i.e. equal amounts of the three primaries are required to match the light from the equal energy illuminant E. Using these curves shown in Fig(2.9) and given the spectral distribution of any color, we can use equation(1) to calculate the tristimulus values required by the standard observer to match that color.

#### □ CIE Primaries:-

CIE defined three new primaries  $x$ ,  $y$ , and  $z$  in which standard observer results can be expressed. It is possible to calculate the amounts of  $X, Y, Z$  needed to match any color, given its tristimulus values corresponding to any other primaries such as  $R_o$ ,  $G_o$  and  $B_o$ . In order to do this, CIE has defined the transformation equations relating two primary systems as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.365 & -0.515 & 0.005 \\ -0.897 & 1.426 & -0.014 \\ -0.468 & 0.089 & 1.009 \end{bmatrix} \begin{bmatrix} R_o \\ G_o \\ B_o \end{bmatrix}$$

#### □ Properties of CIE Coordinate System:-

- (1) The tristimulus values  $X, Y, Z$  are normalized to equal energy white.
- (2) The  $Y$  tristimulus value corresponds to the luminance of the color. The color matching function for  $Y$  is proportional to the relative luminous efficiency shown earlier.
- (3) Unlike  $R, G, B$  system, where sometimes certain tristimulus values must be negative for match, the tristimulus value and the color matching functions in CIE-XYZ system are always positive as shown in Fig (2.12).

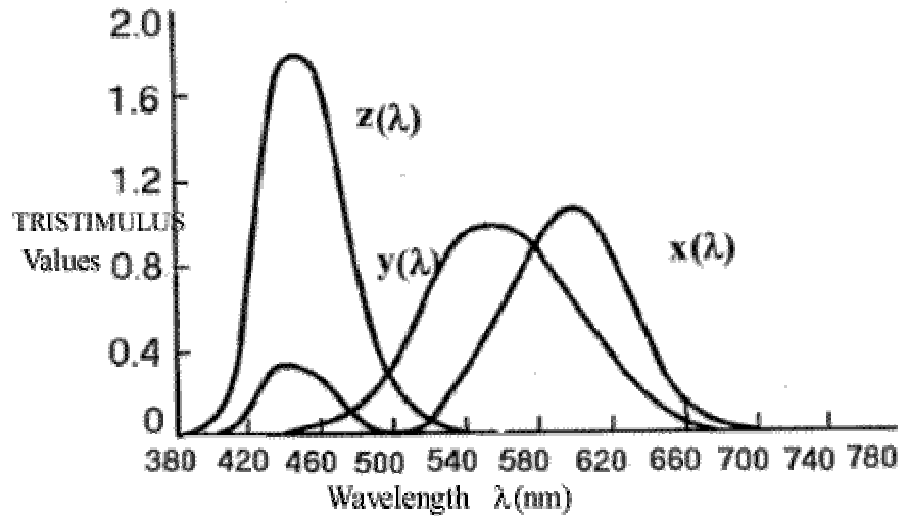


Figure :Color matching functions  $x(\lambda)$ ,  $y(\lambda)$ ,  $z(\lambda)$ , for the 2° Standard Observer

This positivity makes X, Y, Z primaries non-real or imaginary i.e. they cannot be realized by any actual color stimuli. In X, Y, Z tristimulus vector space, the primaries are represented by vectors outside the domain representing real colors. This will be clear from the following section.

### Chromaticity coordinates in CIE-XYZ system.

For tristimulus value X, Y, Z the chromaticity coordinates are given by

$$x = X/X+Y+Z$$

$$y = Y/X+Y+Z$$

$$z = Z/X+Y+Z$$

Thus, a color is specified by two chromaticity coordinates (x, y) and the Y where Y is the luminance and (x, y) can be thought of as color of the stimulus devoid of brightness.

A plot of chromaticity coordinates for the physical colors forms a 'chromaticity diagram'. Two such diagrams are shown in Fig (2.13.) for chromaticities  $(x_0, y_0)$  and (x, y).

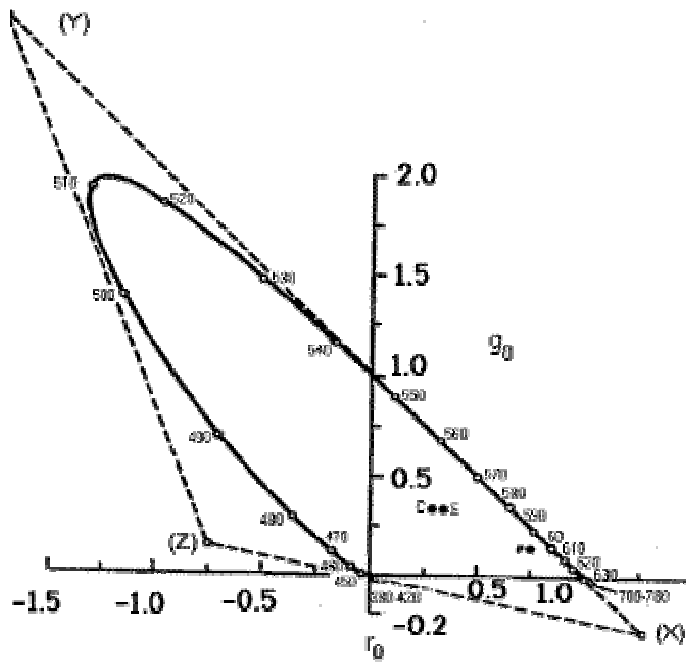


Figure :The  $(r_0, g_0)$  chromaticity diagram for the Standard Observer.

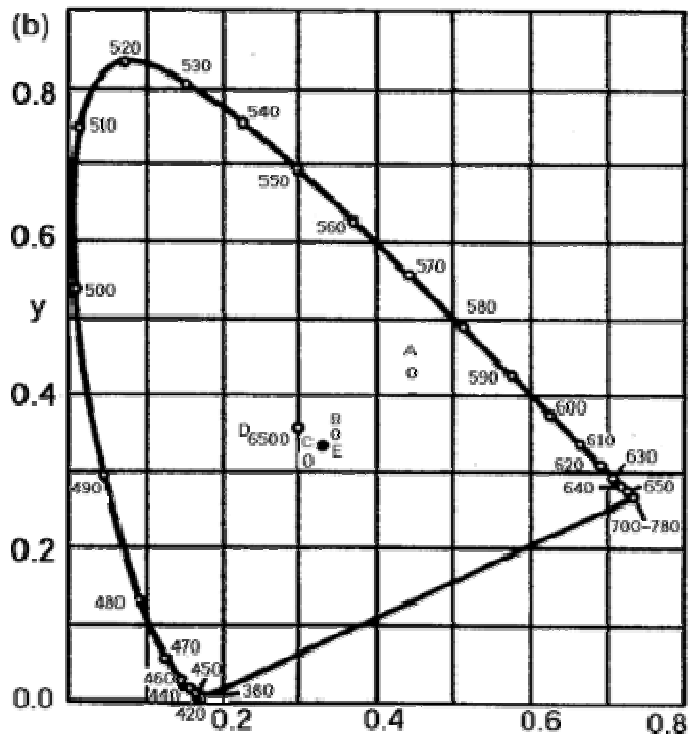


Figure :1931 CIE-xy chromaticity diagram

These chromaticity diagrams also show the chromaticity coordinates of each spectral color. The pure spectral colors are plotted on the elongated horse-shoe shaped curve called 'spectral-locus'. The straight line joining the two extremes of the spectral locus is called the 'line of purples'.

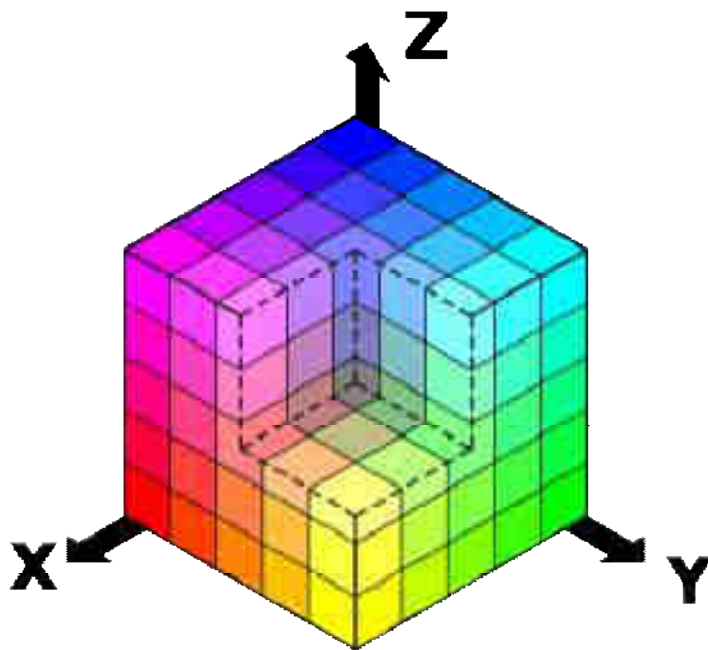
## Color models

A **color model** is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called a color space. This section describes ways in which human color vision can be modeled.

### RGB COLOR MODEL

Media that transmit light (such as television) use additive color mixing with primary colors of red, green, and blue, each of which stimulates one of the three types of the eye's color receptors with as little stimulation as possible of the other two. This is called "RGB" color space. Mixtures of light of these primary colors cover a large part of the human color space and thus produce a large part of human color experiences. This is why color television sets or color computer monitors need only produce mixtures of red, green and blue light. See Additive color.

Other primary colors could in principle be used, but with red, green and blue the largest portion of the human color space can be captured. Unfortunately there is no exact consensus as to what loci in the chromaticity diagram the red, green, and blue colors should have, so the same RGB values can give rise to slightly different colors on different screens.

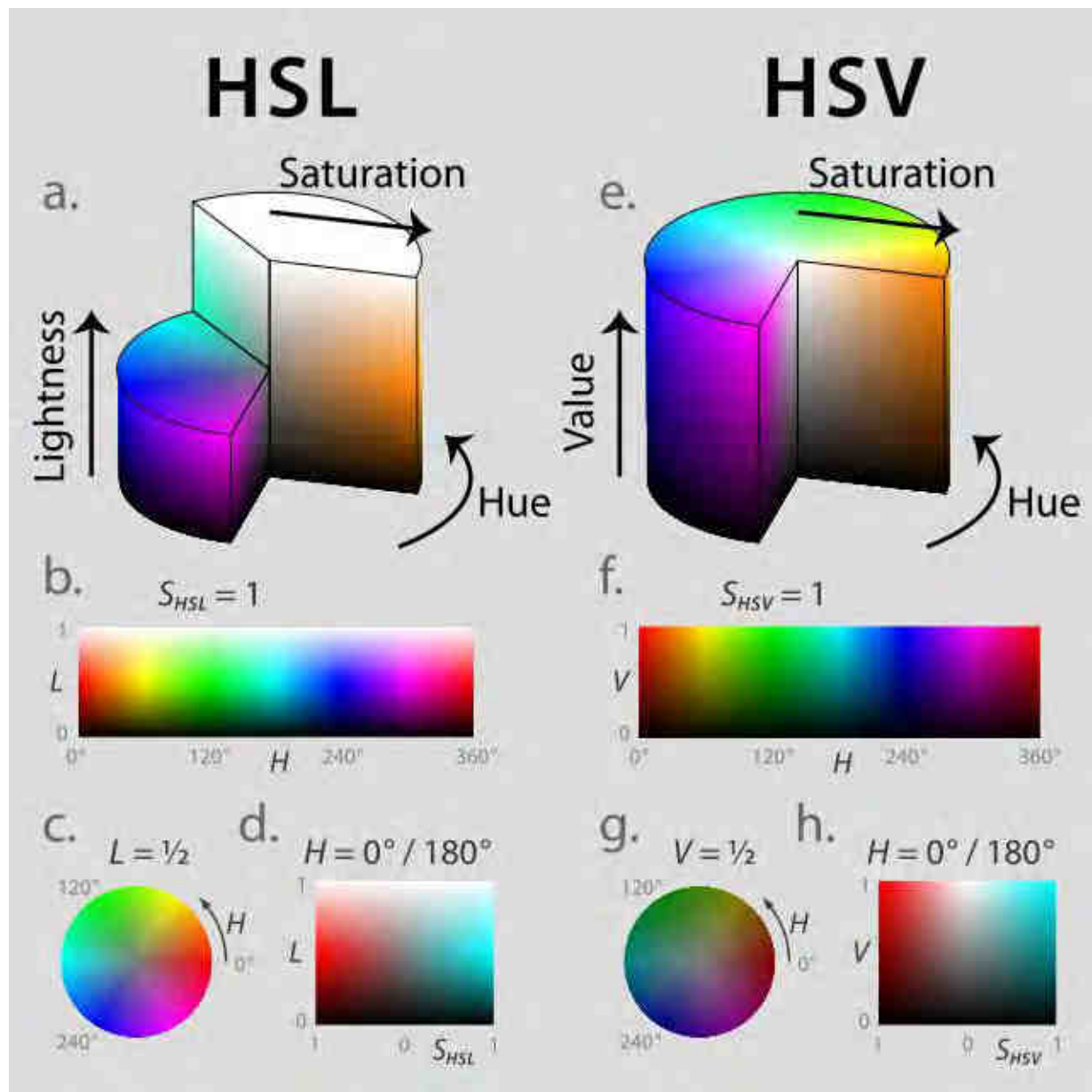


## HSV and HSL representations

Recognizing that the geometry of the RGB model is poorly aligned with the color-making attributes recognized by human vision, computer graphics researchers developed two alternate representations of RGB, HSV and HSL (*hue, saturation, value* and *hue, saturation, lightness*), in the late 1970s. HSV and HSL improve on the color cube representation of RGB by arranging colors of each hue in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. The fully saturated colors of each hue then lie in a circle, a color wheel.

HSV models itself on paint mixture, with its saturation and value dimensions resembling mixtures of a brightly colored paint with, respectively, white and black. HSL tries to resemble more perceptual color models such as NCS or Munsell. It places the fully saturated colors in a circle of lightness  $\frac{1}{2}$ , so that lightness 1 always implies white, and lightness 0 always implies black.

HSV and HSL are both widely used in computer graphics, particularly as color pickers in image editing software. The mathematical transformation from RGB to HSV or HSL could be computed in real time, even on computers of the 1970s, and there is an easy-to-understand mapping between colors in either of these spaces and their manifestation on a physical RGB device.



## CMYK COLOR MODEL

It is possible to achieve a large range of colors seen by humans by combining cyan, magenta, and yellow transparent dyes/inks on a white substrate. These are the *subtractive* primary colors. Often a fourth ink, black, is added to improve reproduction of some dark colors. This is called "CMY" or "CMYK" color space.

The cyan ink absorbs red light but transmits green and blue, the magenta ink absorbs green light but transmits red and blue, and the yellow ink absorbs blue light but transmits red and green. The white substrate reflects the transmitted light back to the viewer. Because in practice the CMY inks suitable for

printing also reflect a little bit of color, making a deep and neutral black impossible, the K (black ink) component, usually printed last, is needed to compensate for their deficiencies. Use of a separate black ink is also economically driven when a lot of black content is expected, e.g. in text media, to reduce simultaneous use of the three colored inks. The dyes used in traditional color photographic prints and slides are much more perfectly transparent, so a K component is normally not needed or used in those media.

## **Color conversion**

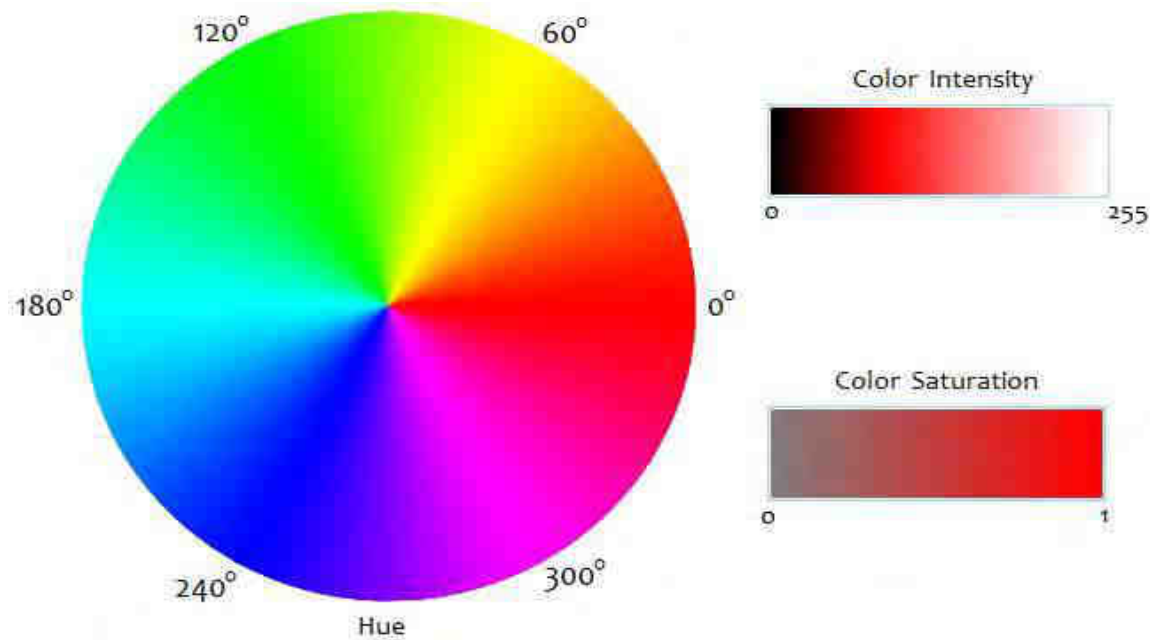
### RGB to HSI, HSI to RGB Conversion

The RGB color model is an additive system in which each color is defined by the amount of red, green, and blue light emitted. In the RGB scheme, colors are represented numerically with a set of three numbers, each of which ranges from 0 to 255. White has the highest RGB value of (255, 255, 255) while black has the lowest value of (0, 0, 0). This is consistent with the additive nature of the RGB system, since white light is the presence of all colors of light, and black is the absence of all light.

There are other three-parameter representations of colors. One such system is the HSI color model, which encodes colors according to their Hue, Saturation, and Intensity. The HSI model is used by some graphics programs and color monitors as an alternative to, or alongside the RGB representation.

In the HSI system, the hue of a color is its angle measure on a color wheel. Pure red hues are  $0^\circ$ , pure green hues are  $120^\circ$ , and pure blues are  $240^\circ$ . (Neutral colors--white, gray, and black--are set to  $0^\circ$  for convenience.) Intensity is the overall lightness or brightness of the color, defined numerically as the average of the equivalent RGB values.

The HSI definition of saturation is a measure of a color's purity/grayness. Purer colors have a saturation value closer to 1, while grayer colors have a saturation value closer to 0. (In other color models, the meanings and mathematical definitions of "saturation" are slightly different. See HSL and HSV color models for comparison.)



### Equations to Convert RGB Values to HSI Values

Suppose R, G, and B are the red, green, and blue values of a color. The HSI intensity is given by the equation

$$I = (R + G + B) / 3.$$

Now let m be the minimum value among R, G, and B. The HSI saturation value of a color is given by the equation

$$S = \begin{cases} 1 - m/I & \text{if } I > 0, \\ 0 & \text{if } I = 0. \end{cases} \quad \text{or} \quad S = \begin{cases} 1 - m/I & \text{if } I > 0, \\ 0 & \text{if } I = 0. \end{cases}$$

To convert a color's overall hue, H, to an angle measure, use the following equations:

$$H = \cos^{-1} \left[ \frac{(R - \frac{1}{2}G - \frac{1}{2}B) / \sqrt{R^2 + G^2 + B^2 - RG - RB - GB}}{1} \right] \quad \text{if } G \leq B, \text{ or}$$

$$H = 360 - \cos^{-1} \left[ \frac{(R - \frac{1}{2}G - \frac{1}{2}B) / \sqrt{R^2 + G^2 + B^2 - RG - RB - GB}}{1} \right] \quad \text{if } B > G,$$

where the inverse cosine output is in degrees.

### Equations to Convert HSI Values to RGB Values

To convert hue, saturation, and intensity to a set of red, green, and blue values, you must first note the value of H. If H = 0, then R, G, and B are given by

$$R = I + S \cos(H/360) \quad G = I + S \sin(H/360) \quad B = I - S$$



$$\begin{array}{lclclcl} G & & = & & I & & - & & IS \\ B & & = & & I & & - & & IS. \end{array}$$

If  $0 < H < 120$ , then

$$\begin{array}{lclclcl} R & & = & & I & & + & & IS \cdot \cos(H) / \cos(60-H) \\ G & & = & & I & & + & & IS \cdot [1 - \cos(H) / \cos(60-H)] \\ B & & = & & I & & - & & IS. \end{array}$$

If  $H = 120$ , then the red, green, and blue values are

$$\begin{array}{lclclcl} R & & = & & I & & - & & IS \\ G & & = & & I & & + & & 2IS \\ B & & = & & I & & - & & IS. \end{array}$$

If  $120 < H < 240$ , then

$$\begin{array}{lclclcl} R & & = & & I & & - & & IS \\ G & & = & & I & & + & & IS \cdot \cos(H-120) / \cos(180-H) \\ B & & = & & I & & + & & IS \cdot [1 - \cos(H-120) / \cos(180-H)]. \end{array}$$

If  $H = 240$  then

$$\begin{array}{lclclcl} R & & = & & I & & - & & IS \\ G & & = & & I & & - & & IS \\ B & & = & & I & & + & & 2IS. \end{array}$$

And if  $240 < H < 360$ , we have

$$\begin{array}{lclclcl} R & & = & & I & & + & & IS \cdot [1 - \cos(H-240) / \cos(300-H)] \\ G & & = & & I & & - & & IS \\ B & & = & & I & & + & & IS \cdot \cos(H-240) / \cos(300-H). \end{array}$$

## PSEUDO IMAGE PROCESSING

To understand false color, a look at the concept behind true color is helpful. An image is called a "*true-color*" image when it offers a natural color rendition, or when it comes close to it. This means that the colors of an object in an image appear to a human observer the same way *as if* this observer were to *directly* view the object: A green tree appears green in the image, a red apple red, a blue sky blue, and so on.<sup>[1]</sup> When applied to black-and-white images, *true-color* means that the perceived lightness of a subject is preserved in its depiction.

A **false-color image** sacrifices natural color rendition (in contrast to a *true-color image*) in order to ease the detection of features that are not readily discernible otherwise – for example the use of near infrared for the detection of vegetation in satellite images.<sup>[1]</sup> While a false-color image can be created using solely the visual spectrum (e.g. to accentuate color differences), typically some or all data used is

from electromagnetic radiation (EM) outside the visual spectrum (e.g. infrared, ultraviolet or X-ray). The choice of spectral bands is governed by the physical properties of the object under investigation.

As the human eye uses three "spectral bands" (see trichromacy for details), three spectral bands are commonly combined into a false-color image. At least two spectral bands are needed for a false-color encoding,<sup>[4]</sup> and it is possible to combine more bands into the three visual RGB bands – with the eye's ability to discern three channels being the limiting factor.<sup>[5]</sup> In contrast, a "color" image made from one spectral band, or an image made from data consisting of non-EM data (e.g. elevation, temperature, tissue type) is a pseudocolor image (see below).

For true color, the RGB channels (red "R", green "G" and blue "B") from the camera are mapped to the corresponding RGB channels of the image, yielding a "RGB" RGB" mapping. For false color this relationship is changed. The simplest false-color encoding is to take an RGB image in the visible spectrum, but map it differently, e.g. "GBR" RGB". For **"traditional false-color" satellite images** of Earth a "NRG" RGB" mapping is used, with "N" being the near-infrared spectral band (and the blue spectral band being unused) – this yields the typical "vegetation in red" false-color images.<sup>[1][6]</sup>

False color is used (among others) for satellite and space images: Examples are remote sensing satellites (e.g. Landsat, see example above), space telescopes (e.g. the Hubble Space Telescope) or space probes (e.g. Cassini-Huygens). Some spacecraft, with rovers (e.g. the Mars Science Laboratory "Curiosity") being the most prominent examples, have the ability to capture *approximate true-color images* as well.<sup>[3]</sup> Weather satellites produce, in contrast the spacecrafts mentioned previously, *grayscale images* from the visible or infrared spectrum

## Color complement

Complementary colors are pairs of colors which, when combined, cancel each other out. This means that when combined, they produce black, or if colored light (rather than pigment) is used, they produce white. When placed next to each other, they create the strongest contrast for those particular two colors. Due to this striking color clash, the term opposite colors is often considered more appropriate than "complementary colors".

The pairs of complementary colors vary depending upon whether the colors are physical (e.g. from pigments), or from light. These change the way in which the color is made, and therefore change the color model which applies. For pigments, subtractive colors apply, so the complementary/opposite color pairs, are red & green, yellow & violet, and blue & orange. In the RGB color model, which applies to colors created by light, such as on computer and television displays, the complementary/opposite pairs are red & cyan, green & magenta, and blue & yellow.

Since color printing ink does not produce color by pigmentation, but instead produces color by masking colors on a white background to reduce light that would otherwise be reflected, the same mix for producing black applies as for light producing white, i.e. the complementary/opposite pairs are red

& cyan, green & magenta, and blue & yellow. The most clashing colors to the eye may still be as for painting

## **The traditional color model**

On the traditional color wheel developed in the 18th century (see 1708 illustration by Boutet below), used by Claude Monet and Vincent van Gogh and other painters, and still used by many artists today, the primary colors were considered to be red, yellow, and blue, and the primary-secondary complementary pairs are red-green, orange-blue, and yellow-violet<sup>[2]</sup> (or yellow-purple in Boutet's color wheel).

In the traditional model, a complementary color pair is made up of a primary color (yellow, blue, or red) and a secondary color (green, violet or orange). For example, yellow is a primary color, and painters can make violet by mixing of red and blue;<sup>[3]</sup> so when yellow and violet paint are mixed, all three primary colors are present. Since paints work by absorbing light, having all three primaries together results in a black or gray color (see subtractive color). In more recent painting manuals, the more precise subtractive primary colors are magenta, cyan, and yellow.<sup>[4]</sup>

Complementary colors can create some striking optical effects. The shadow of an object appears to contain some of the complementary color of the object. For example, the shadow of a red apple will appear to contain a little blue-green. This effect is often copied by painters who want to create more luminous and realistic shadows. Also, if you stare at a square of color for a long period of time (thirty seconds to a minute), and then look at a white paper or wall, you will briefly see an afterimage of the square in its complementary color.

Placed side by side as tiny dots, in partitive color mixing, complementary colors appear gray.<sup>[5]</sup>

## **Colors produced by light**

The RGB color model, invented in the 19th century and fully developed in the 20th century, uses combinations of red, green, and blue light against a black background to make the colors seen on a computer monitor or television screen. In the RGB model, the primary colors are red, green and blue. The complementary primary-secondary combinations are red-cyan, green-magenta, and blue-yellow. In the RGB color model, the light of two complementary colors, such as red and cyan, combined at full intensity, will make white light, since two complementary colors contain light with the full range of the spectrum. If the light is not fully intense, the resulting light will be gray.

In some other color models, such as the HSV color space, the neutral colors (white, greys, and black) lie along a central axis. Complementary colors (as defined in HSV) lie opposite each other on any horizontal cross-section. For example, in the CIE 1931 color space a color of a "dominant" wavelength can be mixed with an amount of the complementary wavelength to produce a neutral color (gray or white)



### Module III

#### **Digital Speech Processing**

#### *A review of digital signals and system*

Signals convey information. Systems transform signals. A signal can be, for example, a sequence of commands or a list of names. We develop models for such signals and the systems that operate on them, such as a system that interprets a sequence of commands from a musician and produces a sound. Mathematically, we model both signals and systems as functions. A **signal** is a function that maps a domain, often time or space, into a range, often a physical measure such as air pressure or light intensity. A **system** is a function that maps signals from its domain' its input signals' into signals in its range' its output signals. Both the domain and the range are sets of signals (**signal spaces**). Thus, systems are functions that operate on functions.

one-sided z-transform equation is given as

$$X(z) = \sum_{m=0}^{\infty} x(m)z^{-m}$$

The two-sided z-transform is defined as

$$X(z) = \sum_{m=-\infty}^{\infty} x(m)z^{-m}$$

#### **The z-Plane and The Unit Circle**

The frequency variables of the Laplace transform  $s = \sigma + j\omega$  and the z-transform  $z = re^{j\phi}$  are complex variables with real and imaginary parts and can be visualised in a two dimensional plane. In the s-plane the vertical  $j\omega$ -axis is the frequency axis, and the horizontal  $\sigma$ -axis gives the exponential rate of decay, or the rate of growth, of the amplitude of the complex sinusoid.

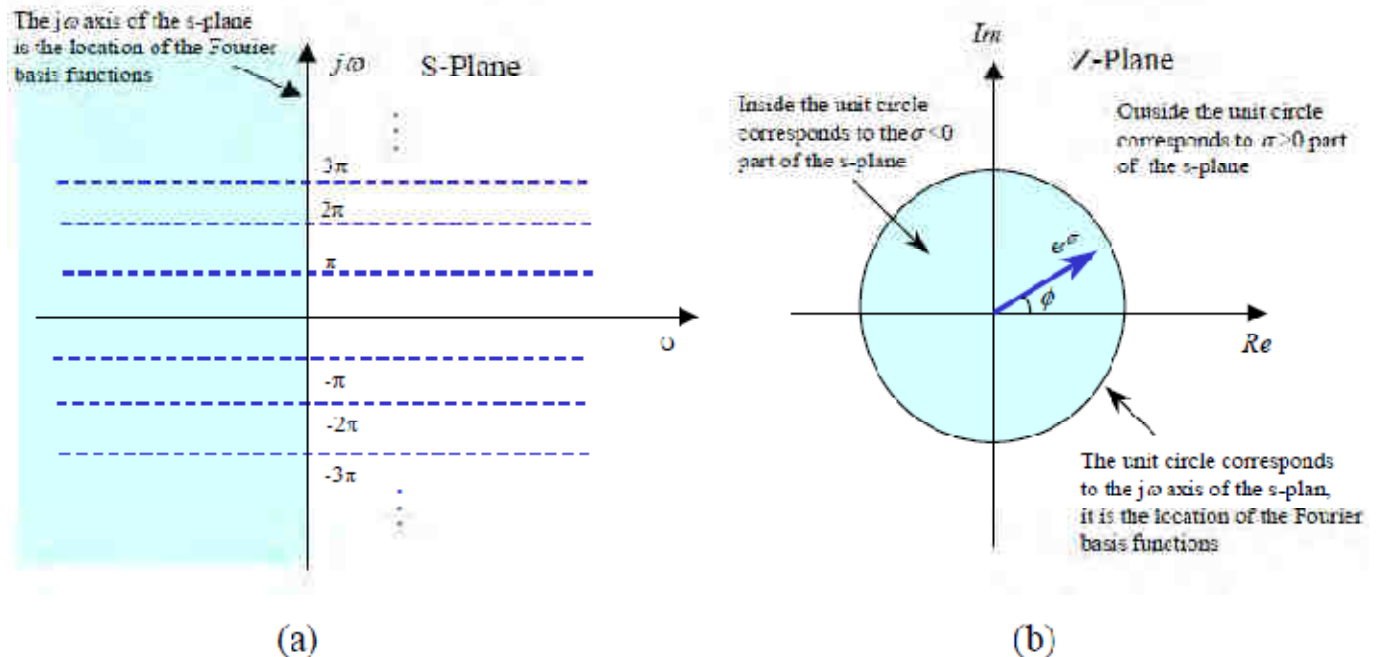
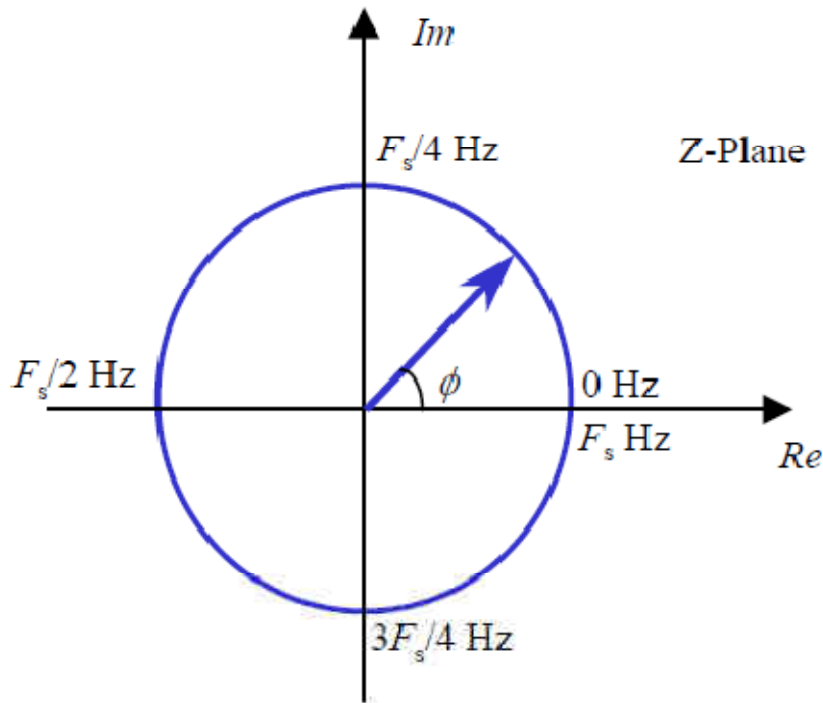


Illustration of (a) the S-plane and (b) the Z-plane

when a signal is sampled in the time domain its Laplace transform, and hence the  $s$ -plane, becomes periodic with respect to the  $j\omega$ -axis. This is illustrated by the periodic horizontal dashed lines in Fig a. Periodic processes can be conveniently represented using a circular polar diagram such as the  $z$ -plane and its associated unit circle. Now imagine bending the  $j\omega$ -axis of the  $s$ -plane of the sampled signal of Fig.a in the direction of the left hand side half of the  $s$ -plane to form a circle such that the points  $(\infty)$  and  $(-\infty)$  meet. The resulting circle is called the *unit circle*, and the resulting diagram is called the  $z$ -plane. The area to the left of the  $s$ -plane, i.e. for  $\sigma < 0$  or  $r = e^{\sigma} < 1$ , is mapped into the area inside the unit circle, this is the region of stable causal signals and systems. The area to the right of the  $s$ -plane,  $\sigma > 0$  or  $r = e^{\sigma} > 1$ , is mapped onto the outside of the unit circle this is the region of unstable signals and systems. The  $j\omega$ -axis, with  $\sigma = 0$  or  $r = e^{\sigma} = 1$ , is itself mapped onto the unit circle line. Hence the Cartesian co-ordinates used in  $s$ -plane for continuous time signals Fig.a, is mapped into a polar representation in the  $z$ -plane for discrete-time signals Fig b.



Above Fig. illustrates that an angle of  $2\pi$  i.e. once round the unit circle, corresponds to a frequency of  $F_s$  Hz where  $F_s$  is the sampling frequency. Hence a frequency of  $f$  Hz corresponds to an angle – given by

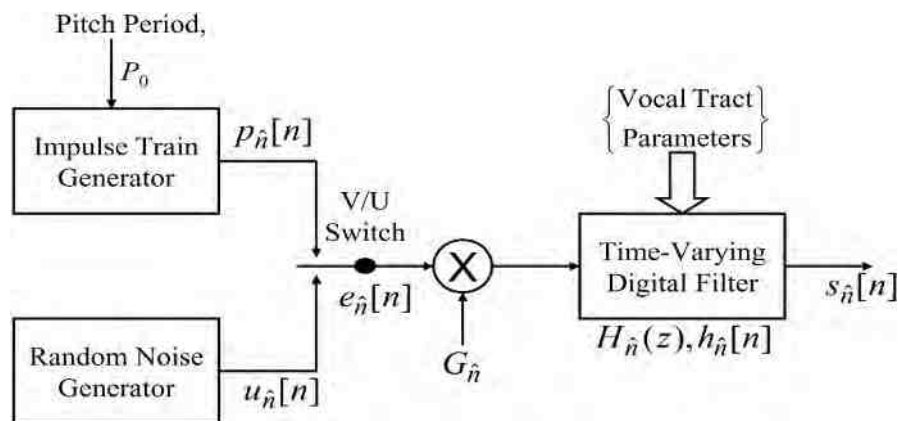
$$\phi = \frac{2\pi}{F_s} f \quad \text{radians}$$

### The Region of Convergence (ROC)

Since the z-transform is an infinite power series, it exists only for those values of the variable  $z$  for which the series converges to a finite sum. The region of convergence (ROC) of  $X(z)$  is the set of all the values of  $z$  for which  $X(z)$  attains a finite computable value.

### Time –Domain Methods for Speech Processing.

Since our goal is to extract parameters of the model by analysis of the speech signal, it is common to assume structures (or representations) for both the excitation generator and the linear system. One such model uses a more detailed representation of the excitation in terms of separate source generators for voiced and unvoiced speech.



(Voiced/unvoiced/system model for a speech signal.)

In this model the unvoiced excitation is assumed to be a random noise sequence, and the voiced excitation is assumed to be a periodic impulse train with impulses spaced by the pitch period ( $P_0$ ) rounded to the nearest sample.<sup>1</sup> The pulses needed to model the glottal flow waveform during voiced speech are assumed to be combined (by convolution) with the impulse response of the linear system, which is assumed to be slowly-time-varying (changing every 50~100 ms or so). By this we mean that over the timescale of phonemes, the impulse response, frequency response, and system function of the system remains relatively constant. For example over time intervals of tens of milliseconds, the system can be described by the convolution expression

$$s_n[n] = \sum_{m=0}^{\infty} h_n[m] e_n[n - m]$$

where the subscript  $\delta n$  denotes the time index pointing to the block of samples of the entire speech signal  $s[n]$  wherein the impulse response  $h_{\delta n}[m]$  applies. We use  $n$  for the time index within that interval, and  $m$  is the index of summation in the convolution sum. In this model, the gain  $G_{\delta n}$  is absorbed into  $h_{\delta n}[m]$  for convenience. To simplify analysis, it is often assumed that the system is an all-pole system with system function of the form:

$$H(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}}$$

For all-pole linear systems, the input and output are related by a difference equation of the form:

$$s[n] = \sum_{k=1}^p a_k s[n-k] + G e[n]$$

Because of the slowly varying nature of the speech signal, it is common to process speech in blocks (also called 'frames') over which the properties of the speech waveform can be assumed to remain relatively constant. This leads to the basic principle of short-time analysis, which is represented in a general form by the equation:

$$X_{\hat{n}} = \sum_{m=-\infty}^{\infty} T\{x[m]w[\hat{n} - m]\}$$

where  $X_{\hat{n}}$  represents the short-time analysis parameter (or vector of parameters) at analysis time  $\hat{n}$ . The operator  $T\{\}$  defines the nature of the short-time analysis function, and  $w[\hat{n} - m]$  represents a time-shifted window sequence, whose purpose is to select a segment of the sequence  $x[m]$  in the neighbourhood of sample  $m = \hat{n}$ .

### Short-Time Energy and Zero-Crossing Rate

Two basic short-time analysis functions useful for speech signals are the short-time energy and the short-time zero-crossing rate. These functions are simple to compute, and they are useful for estimating properties of the excitation function in the model. The short-time energy is defined as

$$E_{\hat{n}} = \sum_{m=-\infty}^{\infty} (x[m]w[\hat{n} - m])^2 = \sum_{m=-\infty}^{\infty} x^2[m]w^2[\hat{n} - m]$$

In this case the operator  $T\{\}$  is simply squaring the windowed samples. In this case,  $E_{\hat{n}} = x^2[n] * h_e[n]$   $n=\hat{n}$ , where the impulse response of the linear filter is  $h_e[n] = w^2[n]$ .

The short-time zero crossing rate is defined as the weighted average of the number of times the speech signal changes sign within the time window. Representing this operator in terms of linear filtering leads to

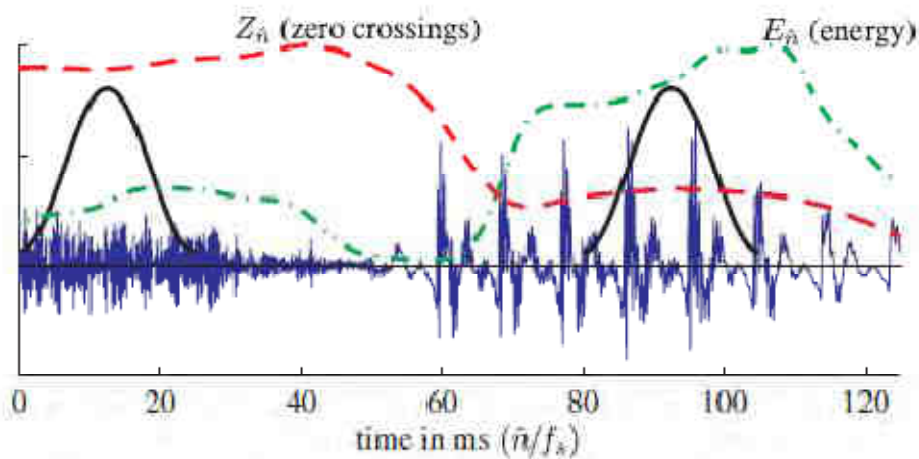
$$Z_{\hat{n}} = \sum_{m=-\infty}^{\infty} 0.5|\text{sgn}\{x[m]\} - \text{sgn}\{x[m - 1]\}|w[\hat{n} - m]$$

where

$$\text{sgn}\{x\} = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0. \end{cases}$$

Since  $0.5|\text{sgn}\{x[m]\} - \text{sgn}\{x[m - 1]\}|$  is equal to 1 if  $x[m]$  and  $x[m - 1]$  have different algebraic signs and 0 if they have the same sign, it follows that  $Z_{\hat{n}}$  in (4.7) is a weighted sum of all the instances of alternating sign (zero-crossing) that fall within the support region of the shifted window  $w[\hat{n} - m]$ .





(Section of speech waveform with short-time energy and zero-crossing rate superimposed.)

## Module IV

### **Digital Speech Processing**

It is the science and technology of the processing of speech signals for different applications in their digital versions. It is a wide field and covers the areas like digital signal processing, digital filtration, speech synthesis, analysis, recognition etc.

### **What is Speech Analysis?**

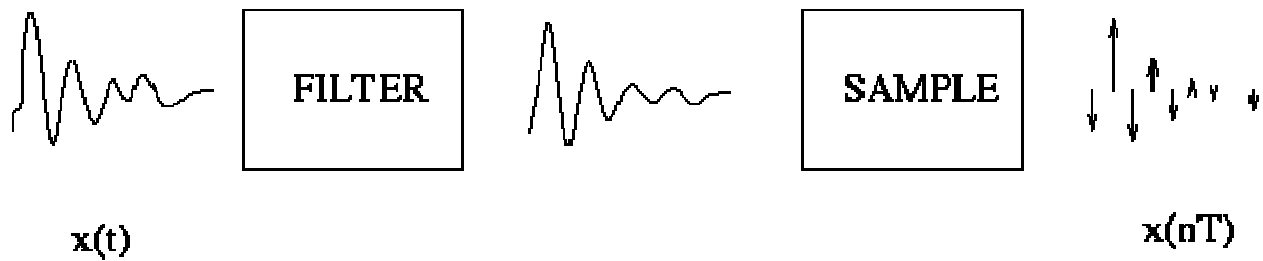
- Analysis of speech sounds taking into consideration their method of production
- The level of processing between the digitized acoustic waveform and the acoustic feature vectors.
- The extraction of "interesting" information as an acoustic vector.

### **Why we should study Speech Processing?**

- In order to process the speech signals we should know their characteristics
- That is possible by study of speech by dissection.
- So this study by parts is known as speech analysis.
- Speech analysis is required for all speech related applications.

## **Sampling theory**

To convert an analogue signal to a digital form it must first be band-limited then sampled:



**Figure :** The digitization process

### Sampling frequency

Signals must be filtered prior to sampling. Theoretically the maximum frequency that can be represented is half the sampling frequency. In practice a higher sample rate is used to allow for non-ideal filters.

The signal is now represented at multiples of the sampling period,  $T$ , as  $s(nT)$  which is also written  $s_n$ .

Telephone speech is sampled at 8 kHz. 16 kHz is generally regarded as sufficient for speech recognition and synthesis. The audio standard is a sample rate of 44.1 kHz (Compact Disc) or 48 kHz (Digital Audio Tape) to represent frequencies up to 20 kHz.

### Waveform coders

- Simple to implement in hardware
- Low delay
- Contain little speech specific information and so are very general
- data rates about 32 kbps

### Pulse Code Modulation (PCM)

- Needs the sampling frequency,  $f_s$ , to be greater than the Nyquist frequency (twice the maximum frequency in the signal)
- For  $n$  bits per sample, the dynamic range is  $\pm 2^{n-1}$  and the quantisation noise is  $1/12$
- Total bit rate:  $n f_s$
- Can use non-uniform quantisation or variable length codes

### Differential Pulse Code Modulation (DPCM)

- Predict the next sample based on the last few *decoded* samples
- Minimize mean squared error of prediction residual - use LP coding
- Good prediction results in a reduction in the dynamic range needed to code the prediction residual and hence a reduction in the bit rate
- Can use non-uniform quantization or variable length codes

## **Adaptive Differential Code Modulation (ADPCM)**

- Speech is quasi-stationary
- Adapt the predictor
- Forward adaptation: send new predictor values
- Backward adaptation: use predictor values computed from recently decoded signal
- Can use non-uniform quantization or variable length codes

## **So what is an acoustic vector?**

A representation of the speech sound at that time (I mean the instant version of speech). For example:

- The short-term power spectra
- A representation of the vocal tract shape
- An estimation of the formant frequencies and bandwidths

These exist as there are limitations on the rate of speech production, (thus the fundamental information transfer rate) and this is less than a general signal at the same sampling rate. Speech analysis deals with time-scales of around 20ms. But normally 10ms or below that gives very accurate output.

## **The problems of speech analysis**

- The assumption that speech is short time stationary
- The formulation of a feature vector representation that captures the important information in the speech signal for future processing
- Speech doesnot have a fixed frequency, so the sampling rate is determined from the highest zero-crossing rate( $f = \text{no of zero crossings}/2$ )

## **Sampling frequency**

Signals must be filtered prior to sampling. Theortically the maximum frequency that can be represented is half the sampling frequency. In practice a higher sample rate is used to allow for non-ideal filters.

Telephone speech is sampled at 8 kHz. 16 kHz is generally regarded as sufficient for speech recognition and synthesis. The audio standard is a sample rate of 44.1 kHz (Compact Disc) or 48 kHz (Digital Audio Tape) to represent frequencies up to 20 kHz.

## **Filters Used for Speech:**

The digital filters used for the processing of speech are of two types:

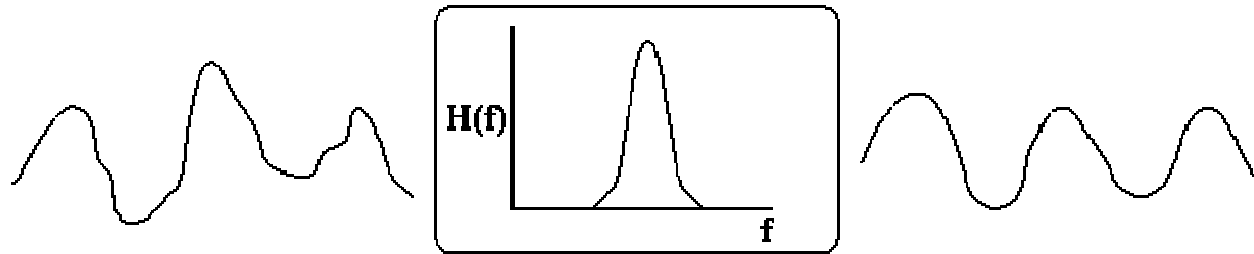
1. FIR or the finite impulse response filters

## 2. IIR or the infinite impulse response filters

### Filter bank Analysis

We can now assemble a set of band pass filters to analyze speech. These need to be covering - that is every frequency is covered by one filter so no information is lost.

An example of one filter has been shown in the figure:



**Figure:** A band pass filter

The output is a waveform, but as the phase is not so important we need the magnitude or the energy - how?

- Rectify and smooth - convenient for hardware implementations
- Square and smooth - gives a better estimation of the power
- Hilbert transforms - generate a signal that is 90 degrees out of phase, then square both signals and add:

If the filter bank is implemented as an FIR filter, a simple transform of the coefficients yields the phase shifted signal

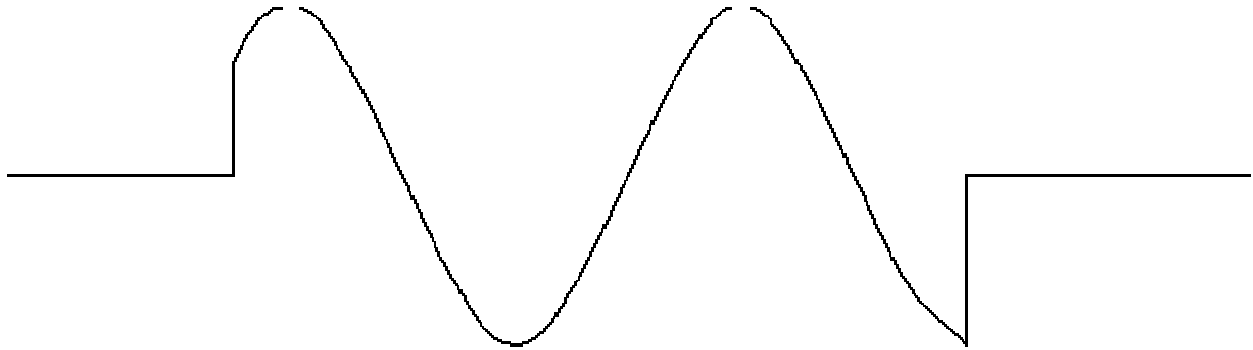
### Windowing

For speech processing we want to assume the signal is short-time stationary and perform a Fourier transform on these small blocks. Solution: multiply the signal by a window function that is zero outside some defined range.

The rectangular window is defined as:

$$W_n = 1, \text{ for } 0 \leq n < N \text{ (where } N \text{ is the window length)}$$
$$= 0, \text{ otherwise}$$

But consider the discontinuities this can generate, as illustrated in the figure below.



**Figure:** A waveform truncated with a rectangular window

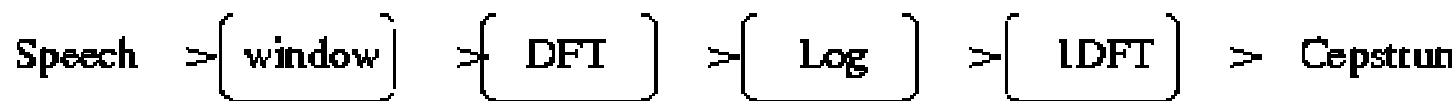
One way to avoid discontinuities at the ends is to taper the signal to zero or near zero and hence reduce the mismatch.

The most common in speech analysis is the Hamming window.

Now the windowed version is quite suitable for the STFT or the Short Time Fourier Transform, because the time is quite small the spectrum can be analyzed properly.

## Cpstral Aalysis:

The following arrangement produces the cepstrum, which is required for the cepstral analysis, shown diagrammatically in the figure below



**Figure:** Cepstral analysis

The reason why the cepstral analysis is required that z-transform based complex analysis is quite suitable from the complex cepstrums.

## Models for Speech Production

A schematic longitudinal cross-sectional drawing of the human vocal tract mechanism is given in figure below. This diagram highlights the essential physical features of human anatomy that enter into the final stages of the speech production process. It shows the vocal tract as a tube of no uniform cross-sectional area that is bounded at one end by the vocal cords and at the other by the mouth opening. This tube serves as an acoustic transmission system for sounds generated inside the vocal tract. For creating nasal sounds like /M/, /N/, or /NG/, a side-branch tube, called the nasal tract, is connected to the main acoustic branch by the trapdoor action of the velum. This branch path radiates sound at the nostrils. The shape (variation of cross-section along the axis) of the vocal tract varies with time due to motions of the lips, jaw, tongue, and velum. Although the

actual human vocal tract is not laid out along a straight line as in figure , this type of model is a reasonable approximation for wavelengths of the sounds in speech. The sounds of speech are generated in the system in several ways. Voiced sounds (vowels, liquids, glides, nasals )

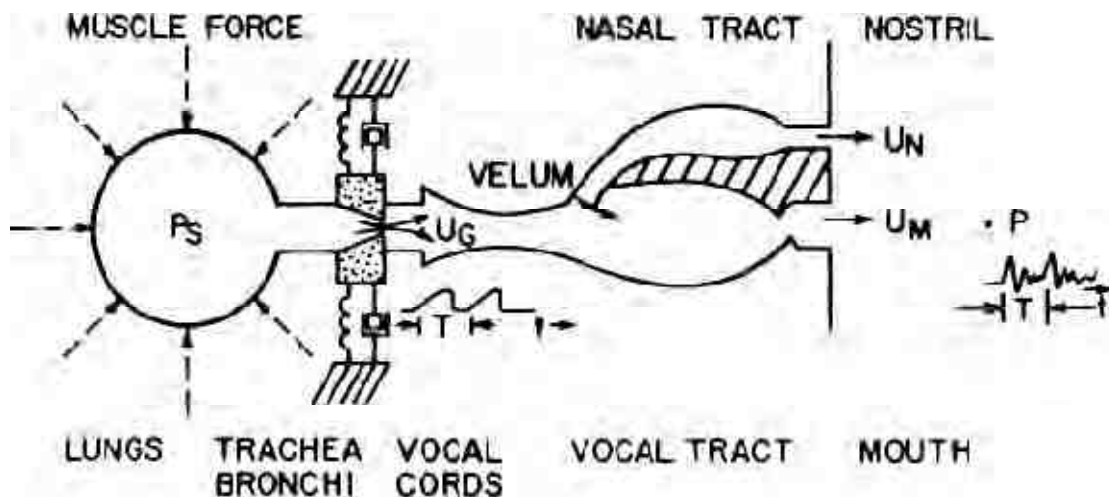
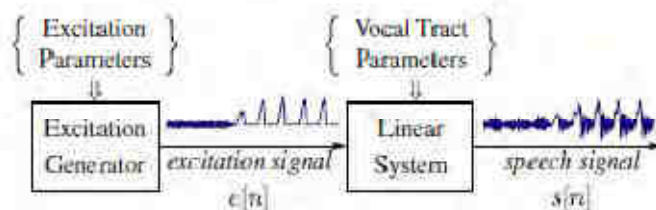


Fig. Schematic model of the vocal tract system. (After Flanagan et al. [35].)

are produced when the vocal tract tube is excited by pulses of air pressure resulting from quasi-periodic opening and closing of the glottal orifice (opening between the vocal cords).

the general character of the speech signal varies at the phoneme rate, which is on the order of 10 phonemes per second, while the detailed time variations of the speech waveform are at a much higher rate. That is, the changes in vocal tract configuration occur relatively slowly compared to the detailed time variation of the speech signal. The sounds created in the vocal tract are shaped in the frequency domain by the frequency response of the vocal tract. The resonance frequencies resulting from a particular configuration of the articulators are instrumental in forming the sound corresponding to a given phoneme. These resonance frequencies are called the *formant frequencies* of the sound . In summary, the fine structure of the time waveform is created by the sound sources in the vocal tract, and the resonances of the vocal tract tube shape these sound sources into the phonemes.



(Source/system model for a speech signal.)

for the most part, it is sufficient to model the production of a sampled speech signal by a discrete-time system model such as the one depicted in above figure. The discrete-time time-

varying linear system simulates the frequency shaping of the vocal tract tube. The excitation generator on the left simulates the different modes of sound generation in the vocal tract. Samples of a speech signal are assumed to be the output of the time-varying linear system. In general such a model is called a *source/system* model of speech production. The short-time frequency response of the linear system simulates the frequency shaping of the vocal tract system, and since the vocal tract changes shape relatively slowly, it is reasonable to assume that the linear system response does not vary over time intervals on the order of 10 ms or so. Thus, it is common to characterize the discrete time linear system by a system function of the form:

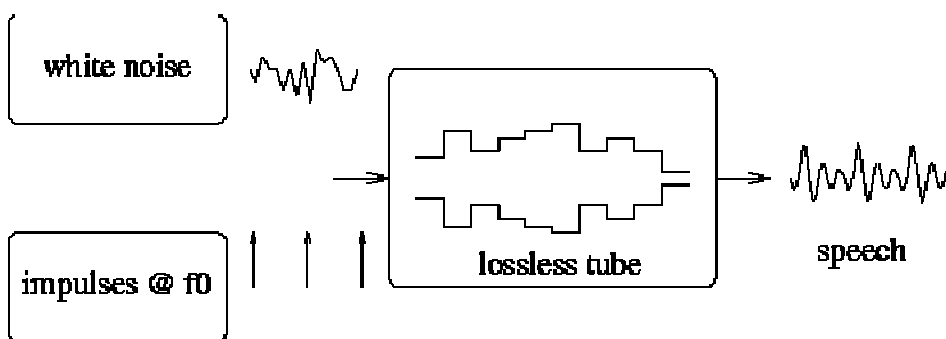
$$H(z) = \frac{\sum_{k=0}^M b_k Z^{-k}}{1 - \sum_{k=1}^N a_k Z^{-k}}$$

where the filter coefficients  $a_k$  and  $b_k$  (labeled as vocal tract parameters) change at a rate on the order of 50–100 times/s. Some of the poles ( $ck$ ) of the system function lie close to the unit circle and create resonances to model the formant frequencies.

## Linear Predictive Analysis

Linear prediction analysis of speech is historically one of the most important speech analysis techniques. The basis is the source-filter model where the filter is constrained to be an all-pole linear filter. This amounts to performing a linear prediction of the next sample as a weighted sum of past samples:

The transfer function of a lossless tube can be described by an all pole model. This is also a reasonable approximation to speech formed by the excitation of the vocal tract by glottal pulses (although the glottal pulses are not spectrally flat).



**Fig:** The lossless tube model of speech production

But:

- The vocal tract is not built of cylinders
- The vocal tract is not lossless

- The vocal tract has a side passage (the nasal cavity)
- fricatives (e.g. /s/ and /sh/) are generated near the lips

Nevertheless, with sufficient parameters the LP model can make a reasonable approximation to the spectral envelope for all speech sounds.

## Parameter estimation

Given  $N$  samples of speech, we would like to compute estimates to  $\alpha_i$  that result in the best fit. One reasonable way to define "best fit" is in terms of mean squared error. These can also be regarded as "most probable" parameters if it is assumed the distribution of errors is Gaussian and a priori there were no restrictions on the values of  $\alpha_i$ .

The error at any time,  $e_n$ , is the difference obtained from the two signals. Refer the book for different methods of parameter estimation. The parameters  $\alpha_i$  and  $\alpha_f$  are from the autocorrelation method of parameter estimation

---

## Practicalities of LPC (As per the Federal Standards)

### Introduction

Linear Predictive Coding (LPC) is one of the most powerful speech analysis techniques, and one of the most useful methods for encoding good quality speech at a low bit rate. It provides extremely accurate estimates of speech parameters, and is relatively efficient for computation. This document describes the basic ideas behind linear prediction, and discusses some of the issues involved in its use.

### Basic Principles

LPC starts with the assumption that the speech signal is produced by a buzzer at the end of a tube. The glottis (the space between the vocal cords) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch). The vocal tract (the throat and mouth) forms the tube, which is characterized by its resonances, which are called *formants*.

LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called *inverse filtering*, and the remaining signal is called the *residue*.

The numbers which describe the formants and the residue can be stored or transmitted somewhere else. LPC synthesizes the speech signal by reversing the process: use the residue to create a source signal, use the formants to create a filter (which represents the tube), and run the source through the filter, resulting in speech.

Because speech signals vary with time, this process is done on short chunks of the speech signal, which are called *frames*. Usually 30 to 50 frames per second give intelligible speech with good compression.



## Estimating the Formants

The basic problem of the LPC system is to determine the formants from the speech signal. The basic solution is a difference equation, which expresses each sample of the signal as a linear combination of previous samples. Such an equation is called a *linear predictor*, which is why this is called Linear Predictive Coding.

The coefficients of the difference equation (the *prediction coefficients*) characterize the formants, so the LPC system needs to estimate these coefficients. The estimate is done by minimizing the mean-square error between the predicted signal and the actual signal.

This is a straightforward problem, in principle. In practice, it involves (1) the computation of a matrix of coefficient values, and (2) the solution of a set of linear equations. Several methods (autocorrelation, covariance, recursive lattice formulation) may be used to assure convergence to a unique solution with efficient computation.

### **Problem: the tube isn't just a tube**

It may seem surprising that the signal can be characterized by such a simple linear predictor. It turns out that, in order for this to work, the tube must not have any side branches. (In mathematical terms, side branches introduce zeros, which require much more complex equations.)

For ordinary vowels, the vocal tract is well represented by a single tube. However, for nasal sounds, the nose cavity forms a side branch. Theoretically, therefore, nasal sounds require a different and more complicated algorithm. In practice, this difference is partly ignored and partly dealt with during the encoding of the residue (see below).

## Encoding the Source

If the predictor coefficients are accurate, and everything else works right, the speech signal can be inverse filtered by the predictor, and the result will be the pure source (buzz). For such a signal, it's fairly easy to extract the frequency and amplitude and encode them.

However, some consonants are produced with turbulent airflow, resulting in a hissy sound (fricatives and stop consonants). Fortunately, the predictor equation doesn't care if the sound source is periodic (buzz) or chaotic (hiss).

This means that for each frame, the LPC encoder must decide if the sound source is buzz or hiss; if buzz, estimate the frequency; in either case, estimate the intensity; and encode the information so that the decoder can undo all these steps. This is how **LPC-10e**, the algorithm described in **federal standard 1015**, works: it uses one number to represent the frequency of the buzz, and the number 0 is understood to represent hiss. **LPC-10e** provides intelligible speech transmission at 2400 bits per second.

### **Problem: the buzz isn't just buzz**

Unfortunately, things are not so simple. One reason is that there are speech sounds which are made with a combination of buzz and hiss sources (for example, the initial consonants in "this zoo" and the middle

consonant in "azure"). Speech sounds like this will not be reproduced accurately by a simple LPC encoder.

Another problem is that, inevitably, any inaccuracy in the estimation of the formants means that more speech information gets left in the residue. The aspects of nasal sounds that don't match the LPC model (as discussed above), for example, will end up in the residue. There are other aspects of the speech sound that don't match the LPC model; side branches introduced by the tongue positions of some consonants, and tracheal (lung) resonances are some examples.

Therefore, the residue contains important information about how the speech should sound, and LPC synthesis without this information will result in poor quality speech. For the best quality results, we could just send the residue signal, and the LPC synthesis would sound great. Unfortunately, the whole idea of this technique is to compress the speech signal, and the residue signal takes just as many bits as the original speech signal, so this would not provide any compression.

## Encoding the Residue

Various attempts have been made to encode the residue signal in an efficient way, providing better quality speech than **LPC-10e** without increasing the bit rate too much. The most successful methods use a *codebook*, a table of typical residue signals, which is set up by the system designers. In operation, the analyzer compares the residue to all the entries in the codebook, chooses the entry which is the closest match, and just sends the *code* for that entry. The synthesizer receives this code, retrieves the corresponding residue from the codebook, and uses that to *excite* the formant filter. Schemes of this kind are called Code Excited Linear Prediction (**CELP**).

For **CELP** to work well, the codebook must be big enough to include all the various kinds of residues. But if the codebook is too big, it will be time consuming to search through, and it will require large codes to specify the desired residue. The biggest problem is that such a system would require a different code for every frequency of the source (pitch of the voice), which would make the codebook extremely large.

This problem can be solved by using two small codebooks instead of one very large one. One codebook is fixed by the designers, and contains just enough codes to represent one pitch period of residue. The other codebook is adaptive; it starts out empty, and is filled in during operation, with copies of the previous residue delayed by various amounts. Thus, the adaptive codebook acts like a variable shift register, and the amount of delay provides the pitch.

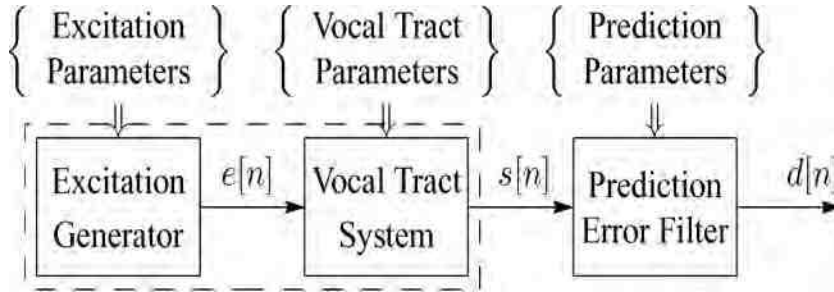
This is the **CELP** algorithm described in **federal standard 1016**. It provides good quality, natural sounding speech at 4800 bits per second.

## Summary

Linear Predictive Coding is a powerful speech analysis technique for representing speech for low bit rate transmission or storage. We hope this tutorial has been informative and helpful. For more information, click on one of the pointers below, or see the texts listed in the References section.

## Linear Predictive Analysis

Linear predictive analysis is one of the most powerful and widely used speech analysis techniques. The importance of this method lies both in its ability to provide accurate estimates of the speech parameters and in its relative speed of computation. The sampled speech signal was modeled as the output of a linear, slowly time-varying system excited by either quasi-periodic impulses (during voiced speech), or random noise (during unvoiced speech).



(Model for linear predictive analysis of speech signals.)

The particular form of the source/system model implied by linear predictive analysis is depicted in above figure, where the speech model is the part inside the dashed box. Over short time intervals, the linear system is described by an all-pole system function of the form:

$$H(z) = \frac{S(z)}{E(z)} = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}}$$

In linear predictive analysis, the excitation is defined implicitly by the vocal tract system model, i.e., the excitation is whatever is needed to produce  $s[n]$  at the output of the system. The major advantage of this model is that the gain parameter,  $G$ , and the filter coefficients  $\{a_k\}$  can be estimated in a very straight forward and computationally efficient manner by the method of linear predictive analysis. For the system of above figure with the vocal tract model, the speech samples  $s[n]$  are related to the excitation  $e[n]$  by the difference equation

$$s[n] = \sum_{k=1}^p a_k s[n-k] + Ge[n]$$

A linear predictor with prediction coefficients,  $a_k$ , is defined as a system whose output is

$$\tilde{s}[n] = \sum_{k=1}^p a_k s[n-k]$$

And the prediction error, defined as the amount by which  $\tilde{s}[n]$  fails to exactly predict sample  $s[n]$ , is

$$d[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^p a_k s[n-k]$$

The basic problem of linear prediction analysis is to determine the set of predictor coefficients  $\{a_k\}$  directly from the speech signal in order to obtain a useful estimate of the time-varying vocal tract system. The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform. The resulting parameters are then *assumed* to be the parameters of the system function  $H(z)$  in the model for production of the given segment of the speech waveform. This process is repeated periodically at a rate appropriate to track the phonetic variation of speech (i.e., order of 50-100 times per second).

### The Autocorrelation Method

Perhaps the most widely used method of linear predictive analysis is called the *autocorrelation method* because the covariance function  $\phi\delta_l[i, k]$  needed reduces to the STACF  $\phi\delta_l[|i - k|]$ . In the autocorrelation method, the analysis segment  $s\delta_l[m]$  is defined as

$$S_{\delta_l}[n] = \begin{cases} S[n+m]w[m] & -M_1 \leq m \leq M_2 \\ 0 & \text{otherwise} \end{cases}$$

where the analysis window  $w[m]$  is used to taper the edges of the segment to zero. Since the analysis segment is defined by the windowing of (6.17) to be zero outside the interval  $-M_1 \leq m \leq M_2$ , it follows that the prediction error sequence  $d\delta_l[m]$  can be nonzero only in the range  $-M_1 \leq m \leq M_2 + p$ . Therefore,  $E\delta_l$  is defined as

$$E_{\delta_l} = \sum_{m=-M_1}^{M_2+p} (d_{\delta_l}[m])^2 = \sum_{m=-\infty}^{\infty} (d_{\delta_l}[m])^2$$

The windowing allows us to use the infinite limits to signify that the sum is over all nonzero values of  $d\delta_l[m]$ . Applying this notion leads to the conclusion that

$$\mathcal{S}_{\delta_l}[i, k] = \sum_{m=-\infty}^{\infty} S_{\delta_l}[m]S_{\delta_l}[m + |i - k|] = \Phi_{\delta_l}[|i - k|]$$

Thus,  $\phi[i, k]$  is a function only of  $|i - k|$ . Therefore, we can replace  $\phi\delta_l[i, k]$  by  $\phi\delta_l[|i - k|]$ , which is the STACF defined as

$$\Phi_{\delta_l}[k] = \sum_{m=-\infty}^{\infty} S_{\delta_l}[m]S_{\delta_l}[m + k] = \Phi_{\delta_l}[-k]$$

The resulting set of equations for the optimum predictor coefficients is therefore

$$\sum_{k=1}^p a_k \Phi_{\delta_l}[i - k] = \Phi_{\delta_l}[i] \quad i = 1, 2, \dots, p$$

## The covariance method

The covariance method uses the real values of covariance coefficients.

Note that:

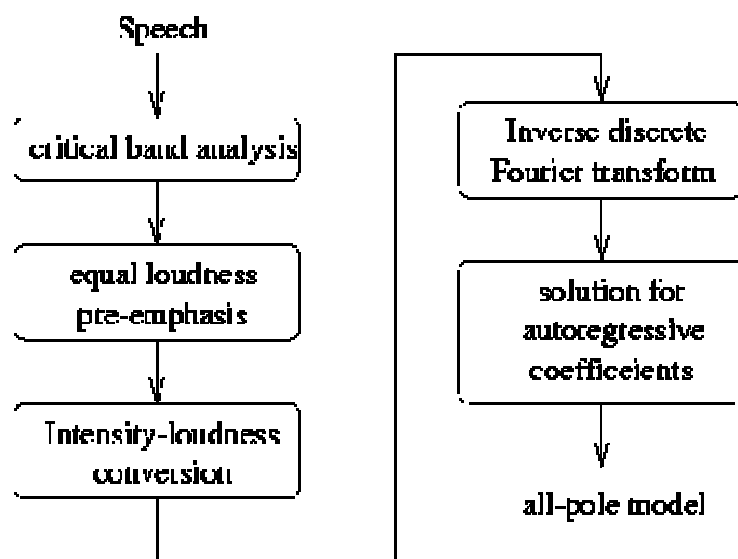
- This method can be used on much smaller sample sequences (as end discontinuities are less of a problem)
- There is no guarantee of stability (but you can check for instability)
- Commonly used in "open" and "closed" phase analysis
- This is just a special case of the general least squares problem.

## The LP spectrum

The transfer function  $1/H(z)$  is a FIR whitening filter for the speech. The frequency response for this can be computed as the Fourier Transform of the filter coefficients, then inverted to give the frequency response of  $H(Z)$ .

## Perceptual Linear Prediction

A combination of DFT and LP techniques is perceptual linear prediction (PLP). It is a modification of normal linear prediction in the design of prediction filter.



**Figure:** Perceptual Linear Prediction in the form of a flow chart

## **Spectral Analysis**

The typical window length is 20ms. For 10 kHz sampling frequency, 200 speech samples are used, padded with 56 zeros, hamming windowed, FFT and converted to a power spectral density. Of course different specifications can be used, but the above one is the most widely used.

## **Speech Coding**

Speech can be coded by various means in the digital format. But the widely used methods are the DPCM, delta modulation (DM), ADM and ADPCM etc.

But as we have seen LPC to be a powerful method, many speech coders are linear prediction vocoders.

## **Linear prediction vocoders**

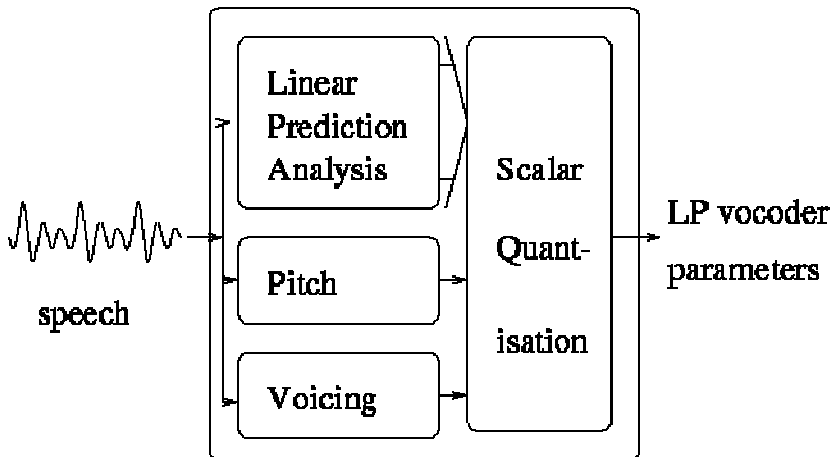
- We employ the source filter model for this.
- For every frame, need to code:
  - A representation of the LP filter
  - Power
  - Degree of voicing
  - Pitch (if voiced)
- Most bits go into the LP parameters

Commonly used representations of the LP parameters:

- LP coefficients: when quantisation is not a problem
- Reflection coefficients: robust but not very efficient
- Line spectral pairs: most efficient

Quantisation:

- Independent: about 50 bits per frame
- Vector quantisation: about 25 bits per frame

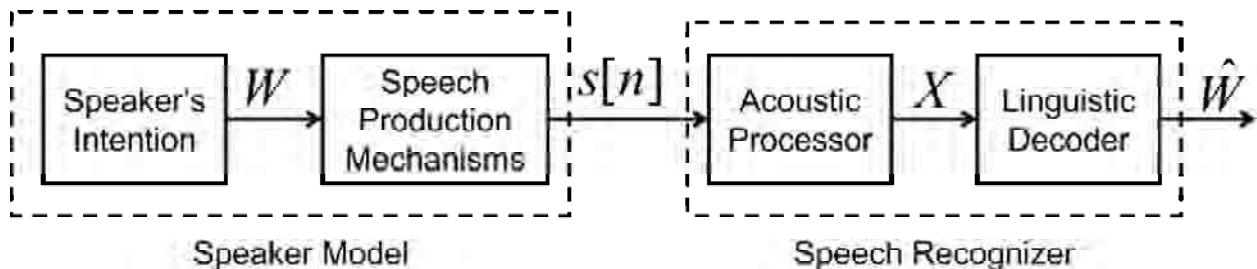


**Figure:** Block Diagram of Simple LPC vocoder

## Speech Recognition

### Speaker Recognition Systems-

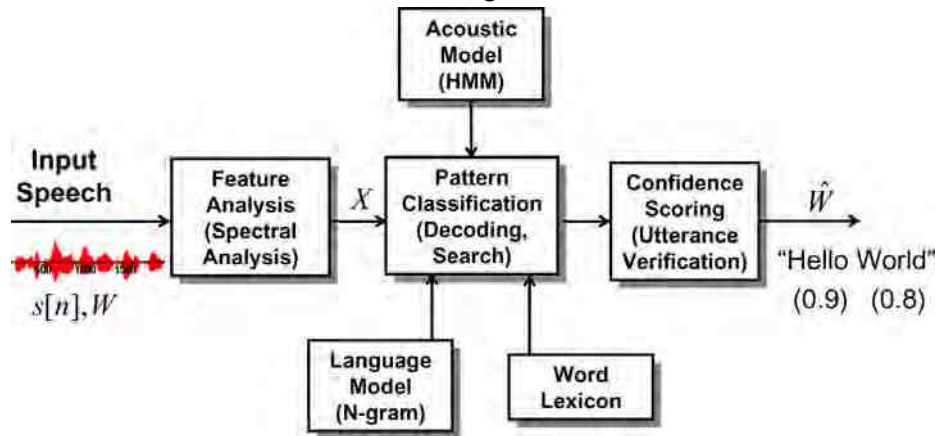
The goal of an ASR system is to accurately and efficiently convert a speech signal into a text message transcription of the spoken words, independent of the device used to record the speech (i.e., the transducer or microphone), the speaker's accent, or the acoustic environment in which the speaker is located (e.g., quiet office, noisy room, outdoors).



(Conceptual model of speech production and speech recognition processes.)

A simple conceptual model of the speech generation and speech recognition processes is given in above figure, which is a simplified version of the speech chain. It is assumed that the speaker intends to express some thought as part of a process of conversing with another human or with a machine. To express that thought, the speaker must compose a linguistically meaningful sentence,  $W$ , in the form of a sequence of words (possibly with pauses and other acoustic events such as uh's, um's, er's etc.). Once the words are chosen, the speaker sends appropriate control signals to the articulatory speech organs which form a speech utterance whose sounds are those required to speak the desired sentence, resulting in the speech waveform  $s[n]$ . We refer to the process of creating the speech waveform from the speaker's intention as the *Speaker Model* since it reflects the speaker's accent and choice of words to express a given thought or request. The processing steps of the Speech Recognizer are shown at the right side of figure and consist of an acoustic processor which analyzes the speech signal and converts it into a set of acoustic (spectral,

temporal) features,  $X$ , which efficiently characterize the speech sounds, followed by a linguistic decoding process which makes a best maximum likelihood estimate of the words of the spoken sentence, resulting in the recognized sentence  $\hat{W}$ .



(Block diagram of an overall speech recognition system)

Above figure shows a more detailed block diagram of the overall speech recognition system. The input speech signal,  $s[n]$ , is converted to the sequence of feature vectors,  $X = \{x_1, x_2, \dots, x_T\}$ , by the feature analysis block (also denoted spectral analysis). The feature vectors are computed on a frame-by-frame basis using various techniques. In particular, the mel frequency cepstrum coefficients are widely used to represent the short-time spectral characteristics. The pattern classification block (also denoted as the decoding and search block) decodes the sequence of feature vectors into a symbolic representation that is the maximum likelihood string,  $\hat{W}$  that could have produced the input sequence of feature vectors. The pattern recognition system uses a set of acoustic models (represented as hidden Markov models) and a word lexicon to provide the acoustic match score for each proposed string. Also, an  $N$ -gram language model is used to compute a language model score for each proposed word string. The final block in the process is a confidence scoring process (also denoted as an utterance verification block), which is used to provide a confidence score for each individual word in the recognized string. Each of the operations in flowchart involves many details and, in some cases, extensive digital computation.

### Hidden Markov model (HMM)-based speech Recognition

As we know the human speech follows the statistical nature, it is always a good thing to put the statistics in the generation and recognition of speech. So when the speech is to be recognised by a statistical means we have to provide a proper model for it. This becomes possible when we use Markov models for this purpose.

So almost all, modern general-purpose speech recognition systems are generally based on HMMs. These are statistical models which output a sequence of symbols or quantities. One possible reason why HMMs are used in speech recognition is that a speech signal could be viewed as a piece-wise stationary signal or a short-time stationary signal. That is, one could assume in a short-time in the range of 10 milliseconds, speech could be approximated as a stationary process. Speech could thus be thought as a Markov model for many stochastic processes (known as **states**).



There are also further reasons why HMMs are popular are because they can be trained automatically and are simple and computationally feasible to use. In speech recognition, to give the very simplest set up possible, the hidden Markov model would output a sequence of n-dimensional real-valued vectors with n around, say, 13, outputting one of these every 10 milliseconds. The vectors, again in the very simplest case, would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short-time window of speech and decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have, in each state, a statistical distribution called a mixture of diagonal covariance Gaussians which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

Of course the above procedure is not the only way we can use the HMMs for the speech recognition. Rather depending on the problem domain we have to choose a proper way of their applications. So HMMs with the help of other optimising tools can be used to recognise the missing data which is required in many complex applications.

Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination hidden Markov model which includes both the acoustic and language model information, or combining it statically beforehand (the finite state transducer, or FST, approach).

### **Dynamic time warping (DTW)-based speech Recognition**

There is an alternative way to recognise the speech as well, by storing them and bringing back in to use when we need them. This process of mating the speech patterns are known as Dynamic time warping or matching of speech patterns with those of the stored versions collected before. Of course this method is an old one and has many limitations. But for small scale applications it can be an economical one.

Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach. Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another they were walking more quickly, or even if there were accelerations and decelerations during the course of one observation. DTW has been applied to video, audio, and graphics as well, any data which can be turned into a linear representation can be analyzed with DTW.

A well known application has been automatic speech recognition, to cope with different speaking speeds. In general, it is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions, i.e. the sequences are "warped" non-linearly to match each other. This sequence alignment method is often used in the context of hidden Markov models.

## Speech Identification

In speech identification a person is singled out from a pool by taking the speech as his or her identifying character. So certain characterising speech templates are stored for this purpose. When it comes for identification the same speech is produced again and the two patterns are compared by some method (Using HMMs or DTW) and then the output is obtained in the form of rejection or acceptance.

## Speech Verification

Speech verification is the verification of a person in terms of his or her claims by taking his or her speech characteristics into consideration. It is similar to that of the speech identification but here the search process is quite small in comparison to that of speech identification.

Speech verification is thus the application of speech recognition to verify the correctness of the pronounced speech. Speech verification doesn't try to decode unknown speech from a huge search space, but instead, knowing the expected speech to be pronounced, it attempts to verify the correctness of the utterance's pronunciation, cadence, pitch, and stress. Pronunciation assessment is the main application of this technology.

## **HMM:**

As we have mentioned before HMMs are nothing but a kind of statistical model which follow the Markov property. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

The **Viterbi algorithm** is a dynamic programming algorithm for finding the most likely sequence of hidden states - called the **Viterbi path** - that results in a sequence of observed events, especially in the context of hidden Markov models.

**Baum-Welch algorithm** is a generalized expectation-maximization (GEM) algorithm. It can compute maximum likelihood estimates and posterior mode estimates for the parameters (transition and emission probabilities) of an HMM, when given only emissions as training data. It is also based on the dynamic programming approach.