UNIT-I

INTRODUCTION TO IMAGE PROCESSING

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, pels 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 application for processing digital images.

Applications:

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

(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.

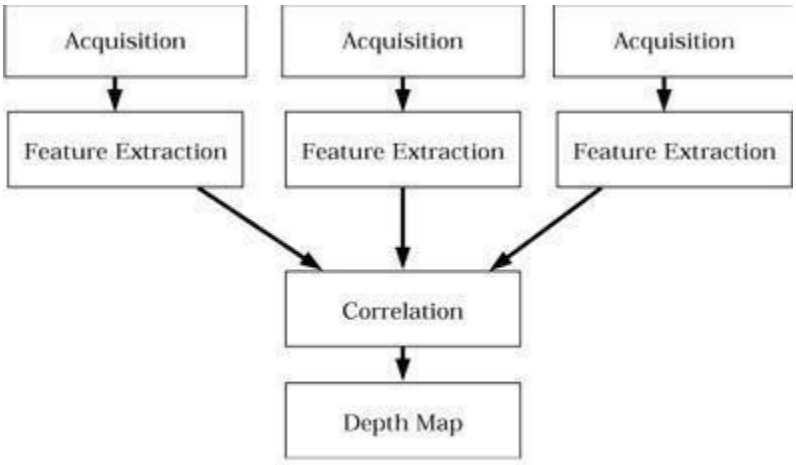


Figure 1: Flow Chart

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

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

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 Spectrum:

(i) Structure of Human eye:

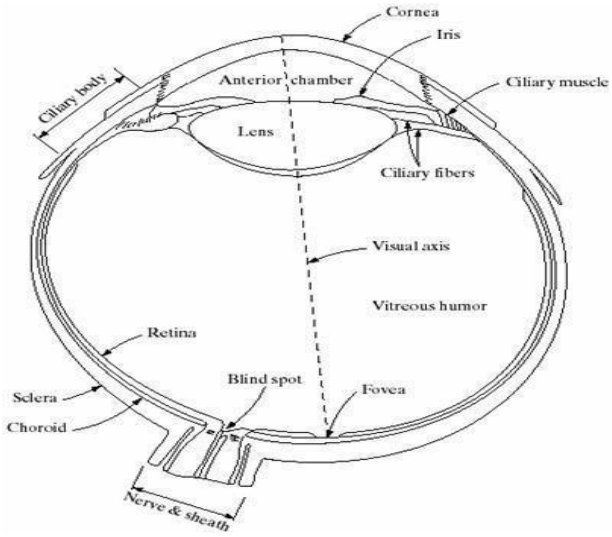
The eye is nearly a sphere with average approximately 20 mm diameter. The eye is enclosed with three membranes

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

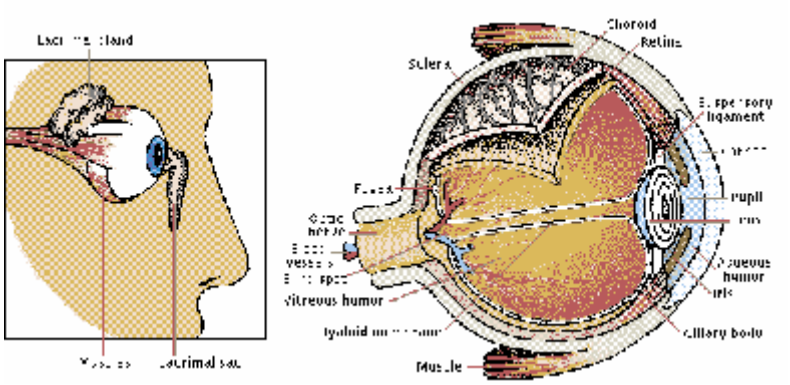


(c) Retina – it is innermost membrane of the eye. When the eye is properly focused, light from an object outside the eye is imaged on the retina. There are various light receptors over the surface of the retina The two major classes of the receptors are-

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 photonic or bright light vision.

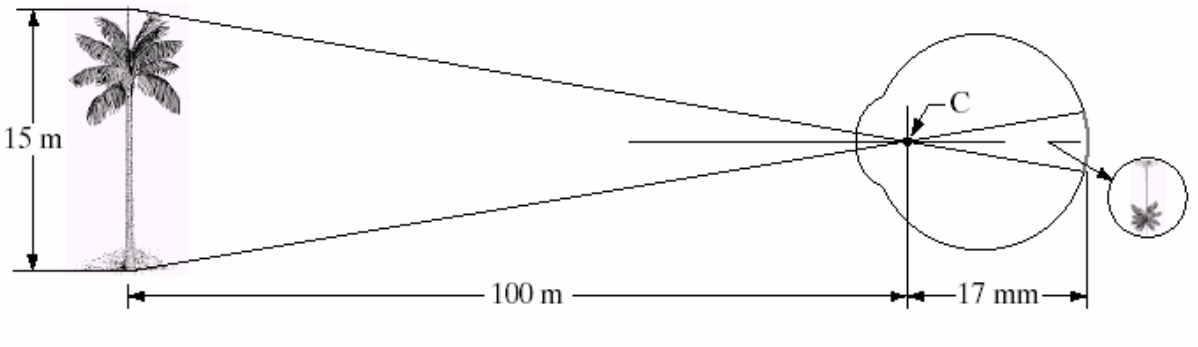
2) Rods – these are very much in number from 75 to 150 million and are distributed over the entire retinal surface. The large area of distribution and the fact that several roads are connected to a single nerve give a general overall picture of the field of view. They are not

involved in the color vision and are sensitive to low level of illumination. Rod vision is called is isotopic or dim light vision. The absent of reciprocators is called blind spot.



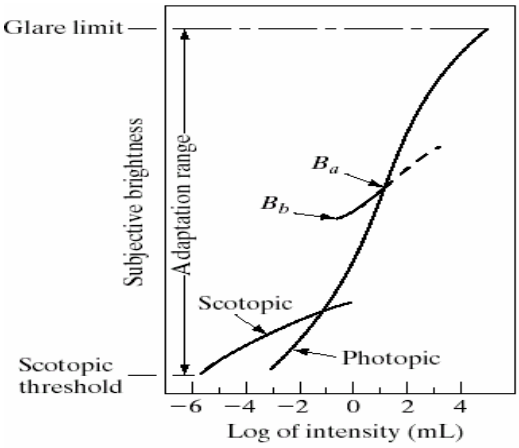
(ii) Image formation in the eye:

The major difference between the lens of the eye and an ordinary optical lens in that the former is flexible. The shape of the lens of the eye is controlled by tension in the fiber of the ciliary body. To focus on the distant object the controlling muscles allow the lens to become thicker 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 nearly object. The lens is most srongly 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.

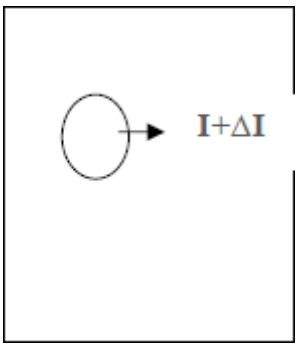


(iii) Brightness adaption and discrimination:

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 isotopic 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 overcall 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 , Ba 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 Bb , at and below which all stimuli are perceived as indistinguishable blacks. The upper portion of the curve is not actually restricted. Whole simply raise the adaptation level higher than Ba. 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 ΔI in the form of a short duration flash that appears as circle in the center of the uniformly illuminated field. If ΔI is not bright enough, the subject cannot see any perceivable changes.

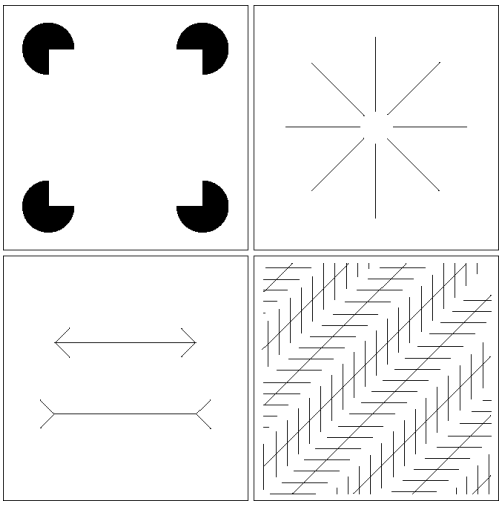


As ΔI gets stronger the subject may indicate of a perceived change. ΔIc is the increment of illumination discernible 50% of the time with background illumination I. Now, ΔIc /I is

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".

(iv) Optical Illusion:

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



Fundamental steps involved in Image processing:

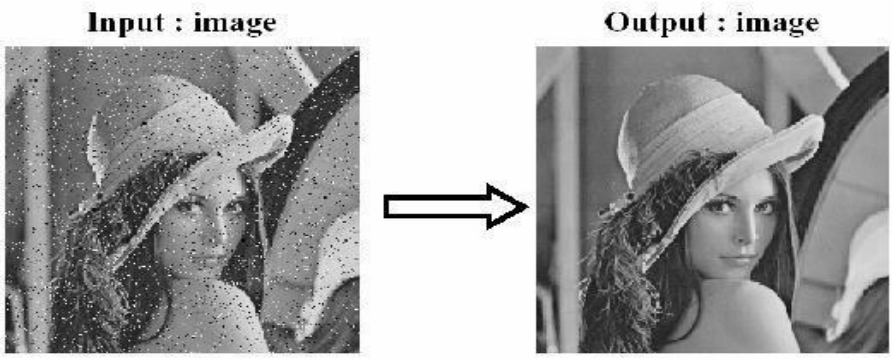
There are two categories of the steps involved in the image processing –

(1) Methods whose outputs are input are images.

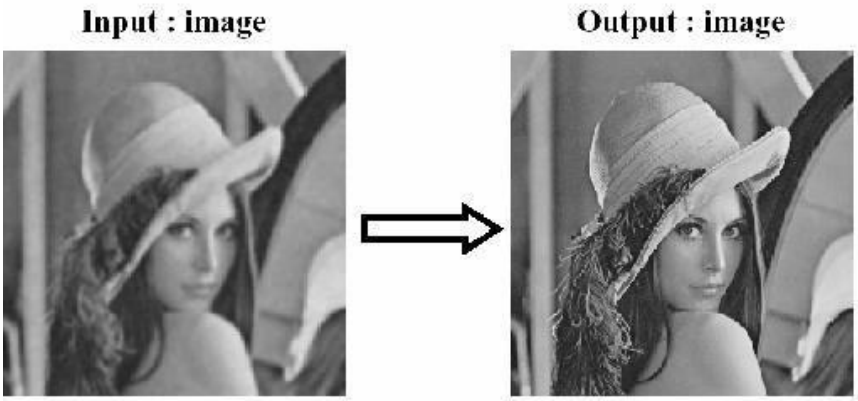
(2) Methods whose outputs are attributes extracted from those images.

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 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.

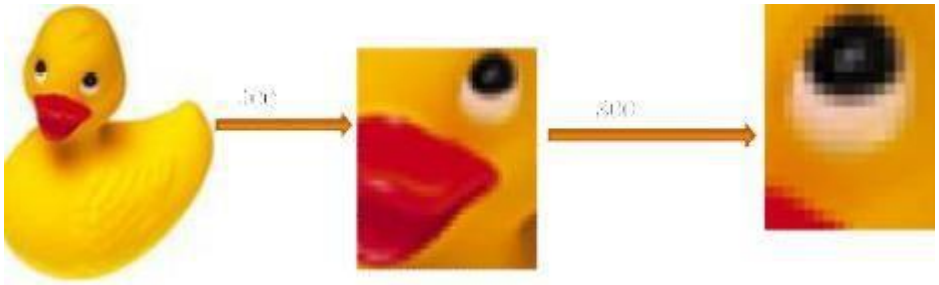


<table>
<tr><td>Color Image Processing</td><td>Wavelets & Multiresolution Processing</td><td>Image Compression</td><td>Morphological Image Processing</td></tr>
<tr><td>Image Restoration</td><td colspan="2" rowspan="3">Knowledge Base</td><td>Image Segmentation</td></tr>
<tr><td>Image Enhancement</td><td>Representation and description</td></tr>
<tr><td>Image Acquisition</td><td>Objects recognition</td></tr>
</table>

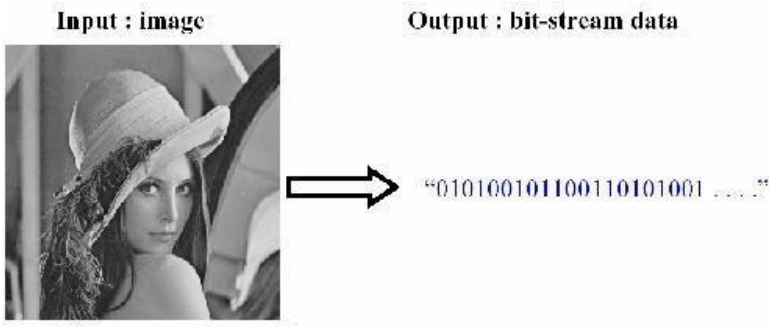
Fig: Fundamental Steps in DIP

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 to 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 use artificial intelligence of software.

Knowledge base:

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge base. This knowledge may be as simple as detailing regions of an image where the information of the interest in known to be located. Thus limiting search that has to be conducted in seeking the information. The knowledge base also can be quite complex such interrelated list of all major possible defects in a materials inspection problems 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 o<f(x,y) <co

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

(a) The amount of the source illumination incident on the scene being viewed.

(b) The amount of the source illumination reflected back by the objects in the scene

These are called illumination and reflectance components and are denoted by i (x,y) an r (x,y) respectively.

The functions combine as a product to form f(x,y). We call the intensity of a monochrome image at any coordinates (x,y) the gray level (l) of the image at that point l= f (x, y.)

$$L_{min} \leq l \leq L_{max}$$

L_{min} is to be positive and Lmax must be finite

$$L_{min} = i_{min}\, r_{min}$$

$$L_{max} = i_{max}\, r_{max}$$

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

Image Sampling And Quantization:

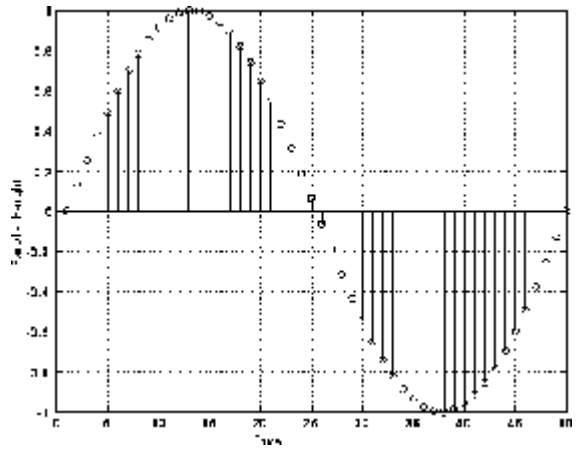
To create a digital image, we need to convert the continuous sensed data into digital from. This involves two processes – sampling and quantization. An image may be continuous with respect to the x and y coordinates and also in amplitude. To convert it into digital form we have to sample the function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

Digitalizing the coordinate values is called sampling.

Digitalizing the amplitude values is called quantization.

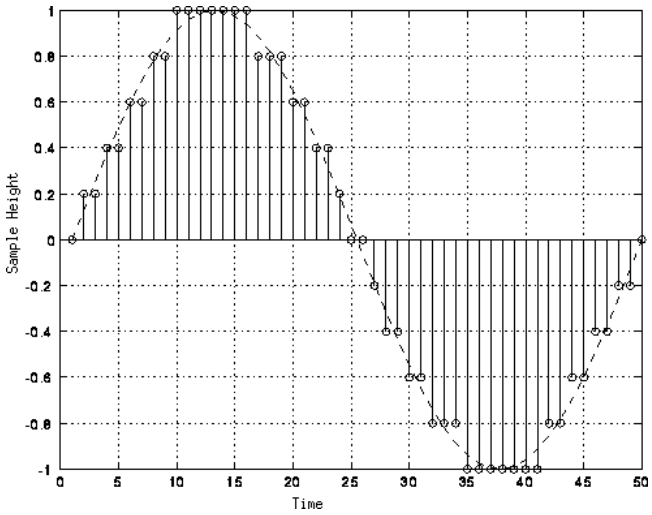
There is a continuous the image along the line segment AB.

To simple this function, we take equally spaced samples along line AB. The location of each samples is given by a vertical tick back (mark) in the bottom part. The samples are shown as block squares superimposed on function the set of these discrete locations gives the sampled function.



In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So we divide the gray level scale into eight discrete levels ranging from

eight level values. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment it made depending on the vertical proximity of a simple to a vertical tick mark.

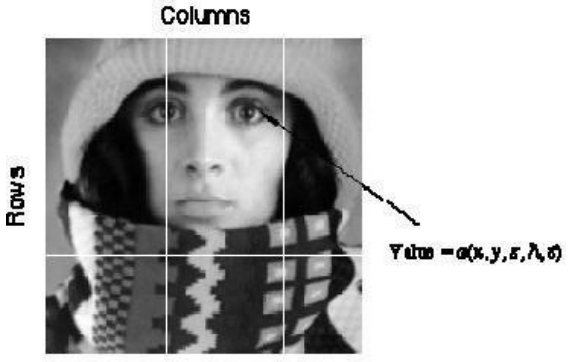


Starting at the top of the image and covering out this procedure line by line produces a two dimensional digital image.

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. The mathematics of that sampling process will be described in subsequent Chapters. For now we will look at some basic definitions associated with the digital image. The effect of digitization is shown in figure.

The 2D continuous image f(x,y) is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates (m,n) with m=0,1,2..N-1 and n=0,1,2…N-1 is f(m,n). In fact, in most cases, is actually a function of many variables including depth, color and time (t).



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

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 orgin become 9X,y) =(o,o) The next Coordinates value along the first signify the iamge along the first row. it does not mean that these are the actual values of physical coordinates when the image was sampled.

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

Thus the right side of the matrix represents a digital element, pixel or pel. The matrix can be represented in the following form as well. The sampling process may be viewed as partitioning the xy plane into a grid with the coordinates of the center of each grid being a pair of elements from the Cartesian products Z2 which is the set of all ordered pair of elements (Zi, Zj) with Zi and Zj 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, the and a digital image become a 2D function whose coordinates and she amplitude value are integers. Due to processing storage and hardware consideration, the number gray levels typically is an integer power of 2.

$$L=2^k$$

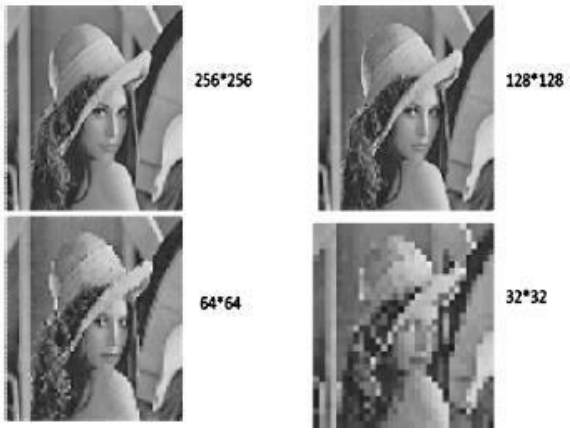
Then, the number, b, of bites required to store a digital image is B=M *N* k

When M=N, the equation become $b=N^2*k$

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

Spatial and Gray level resolution:

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

Relationship between pixels:

(i) Neighbor 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 called the 4-neighbours of p is denoted by $n_4(p)$,Each pixel is a unit distance from (x,y) and some of the neighbors of P lie outside the digital image of (x,y) is on the border if the image . The four diagonal neighbor of P have coordinated

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

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

(ii) Adjacency:

Let v be the set of gray –level values used to define adjacency, in a binary image, v={1} if we are reference to adjacency of pixel with value. Three types of adjacency

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

(i) Q is in n_4 (p) or

(ii) Q is in n_d (q) and the set $N_4(p)$ U $N_4(q)$ has no pixel whose values are from V

(iii) 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

$$D\ [p.q] \geq O\ \{D[p.q] = O \text{ iff } p=q\}$$

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

$$D\ [p.q] \geq O\ \{D[p.q]+D(q,z)$$

The Education Distance between p and is defined as

$$De\ (p,q) = I_y - t\ I$$

The D_4 Education Distance between p and is defined as

$$De\ (p,q) = I_y - t\ I$$

Image sensing and Acquisition:

The types of images in which we are interested are generated by the combination of an

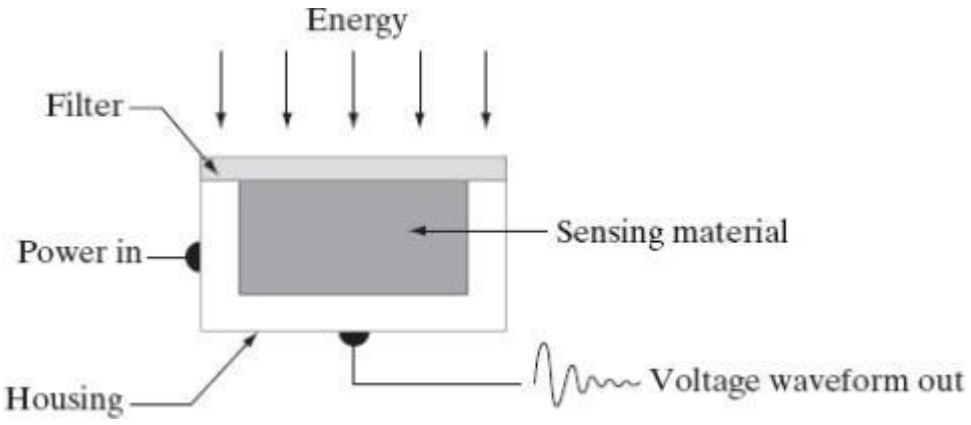


Fig:Single Image sensor

Fig: Line Sensor

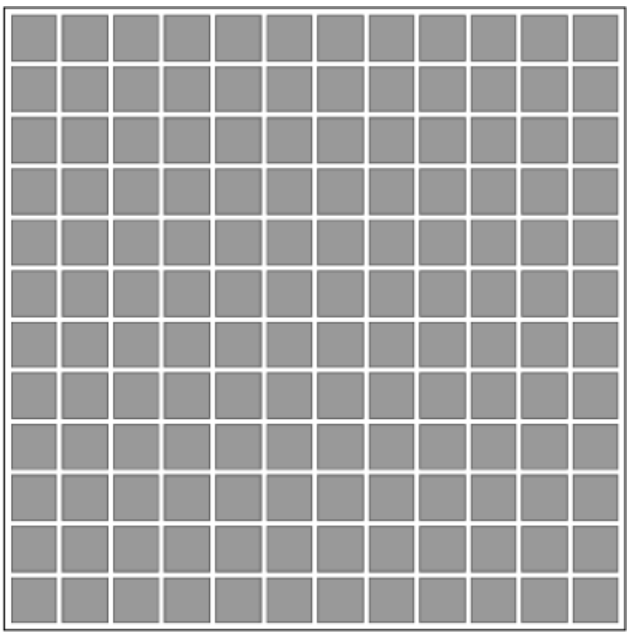


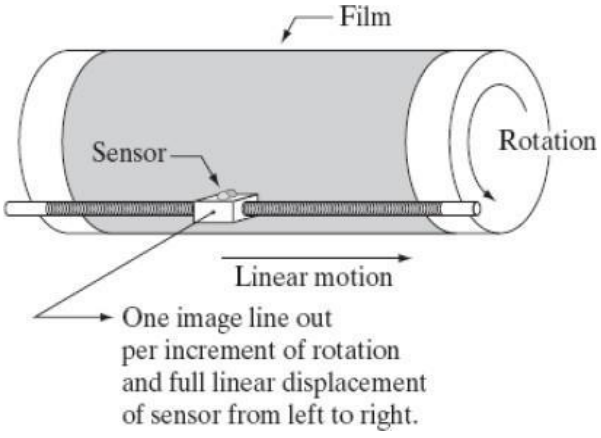
Fig: Array sensor

"illumination" source and the reflection or absorption of energy from that source by the elements of the "scene" being imaged. We enclose *illumination* and *scene* in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene. For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern. Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock

formations, or a human brain. We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light reflected from a planar surface. An example in the second category is when X-rays pass through a patient's body for thepurpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response. In this section, we look at the principal modalities for image sensing and generation.

(i) Image Acquisition using a Single sensor:

The components of a single sensor. Perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform is proportional to light. The use of a filter in front of a sensor improves selectivity. For example, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.

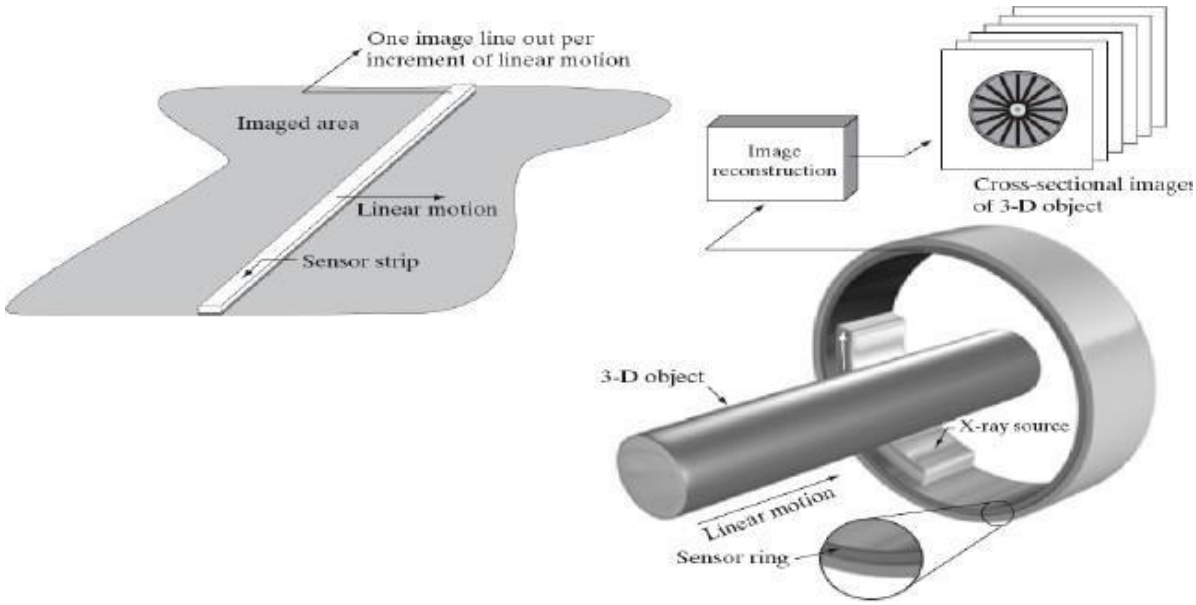


In order to generate a 2-D image using a single sensor, there has to be relative displacements in both the x- and y-directions between the sensor and the area to be imaged. Figure shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Since mechanical motion can be controlled with high precision, this method is an inexpensive (but

slow) way to obtain high-resolution images. Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as *microdensitometers*.

(ii) Image Acquisition using a Sensor strips:

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



(iii) Image Acquisition using a Sensor Arrays:

The individual sensors arranged in the form of a 2-D array. Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras. A typical sensor for these cameras is a

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

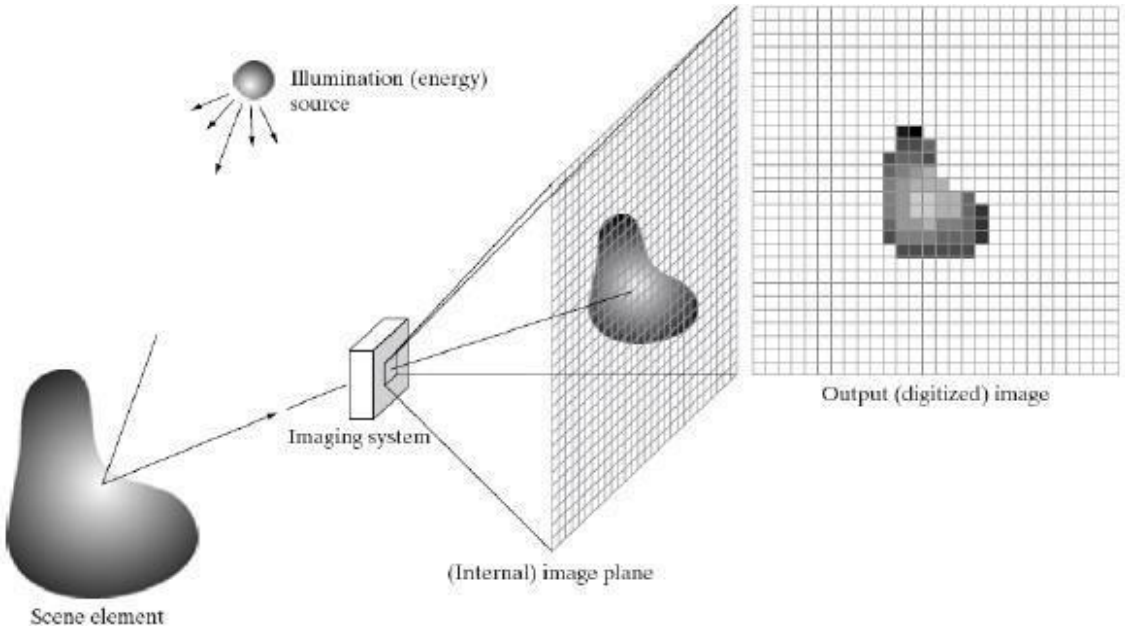
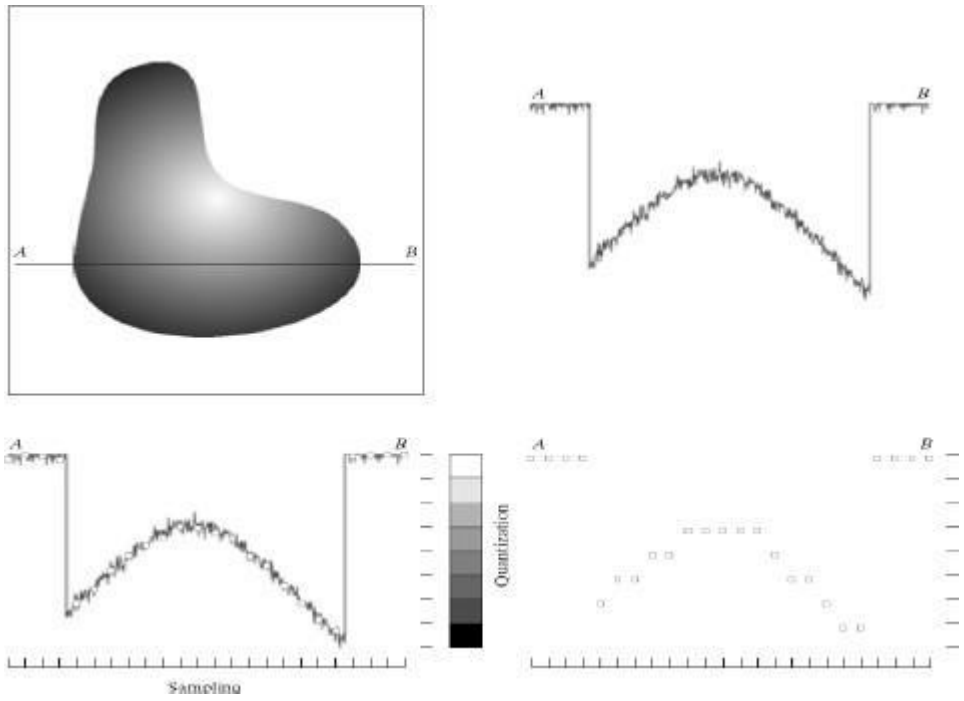


Image sampling and Quantization:

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: *sampling* and *quantization*. A continuous image, f(x, y), that we want to convert to digital form. An image may be continuous with respect to the x- and y-

coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called *sampling*. Digitizing the amplitude values is called *quantization*.



Digital Image representation:

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

- digital photographs
- satellite images
- radiological images (x-rays, mammograms)
- binary images, fax images, engineering drawings

Computer graphics, CAD drawings, and vector graphics in general are not considered in this course even though their reproduction is a possible source of an image. In fact, one goal of intermediate level image processing may be to reconstruct a model (e.g. vector representation) for a given digital image.

Digitization:

Digital image consists of $N \times M$ pixels, each represented by k bits. A pixel can thus have $2k$ different values typically illustrated using a different shades of gray, see Figure . In practical applications, the pixel values are considered as integers varying from 0 (black pixel) to 2^k-1 (white pixel).

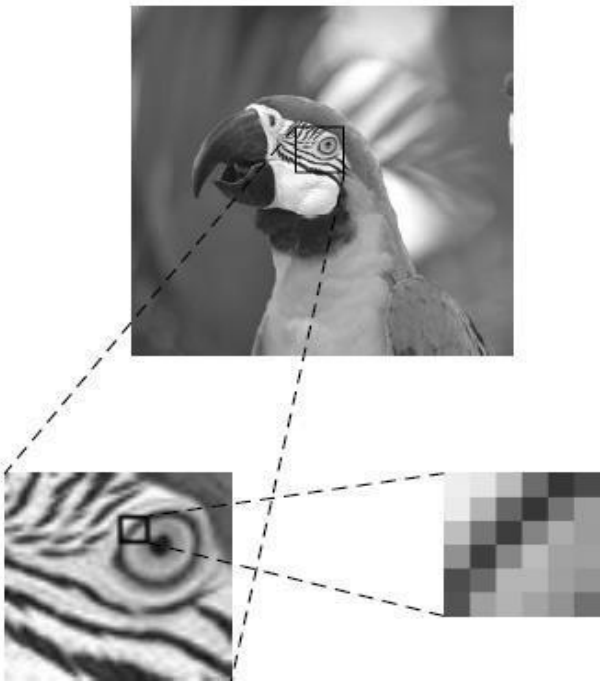


Fig: Example of a digital Image

The images are obtained through a digitization process, in which the object is covered by a two-dimensional sampling grid. The main parameters of the digitization are:

- *Image resolution*: the number of samples in the grid.
- *pixel accuracy*: how many bits are used per sample.

These two parameters have a direct effect on the image quality but also to the storage size of the image (Table 1.1). In general, the quality of the images increases as the resolution and the bits per pixel increase. There are a few exceptions when reducing the number of bits increases the image quality because of increasing the contrast. Moreover, in an image with a very high resolution only very few gray-levels are needed. In some applications it is more important to have a high resolution for detecting details in the image whereas in other applications the number of different levels (or colors) is more important for better outlook of the image. To sum up, if we have a certain amount of bits to allocate for an image, it makes difference how to choose the digitization parameters.

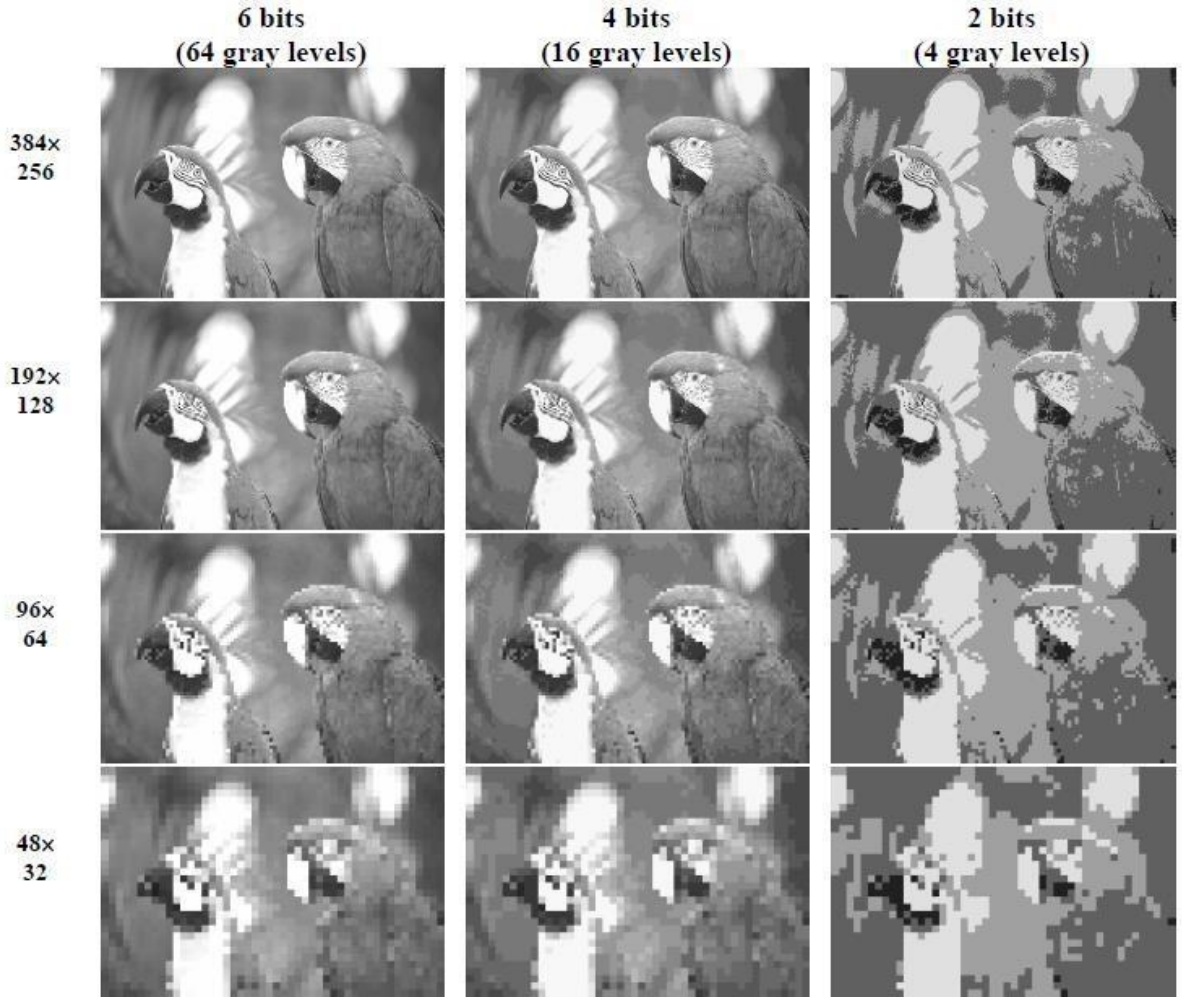


Fig: Effect of resolution and pixel accuracy to image quality

Table Memory consumption (bytes) with the different parameter values.

Resolution	Bits per pixel: 1	2	4	6	8
32 × 32	128	256	512	768	1,024
64 × 64	512	1,024	2,048	3,072	4,096
128 × 128	2,048	4,096	8,192	12,288	16,384
256 × 256	8,192	16,384	32,768	49,152	65,536
512 × 512	32,768	65,536	131,072	196,608	262,144
1024 × 1024	131,072	262,144	524,288	786,432	1,048,576

The properties of human eye imply some upper limits. For example, it is known that the human eye can observe at most one thousand different gray levels in ideal conditions, but in any practical situations 8 bits per pixel (256 gray level) is usually enough. The required levels decreases even further as the resolution of the image increases. In a laser quality printing, as in this lecture notes, even 6 bits (64 levels) results in quite satisfactory result. On the other hand, if the application is e.g. in medical imaging or in cartography, the visual quality is not

the primary concern. For example, if the pixels represent some physical measure and/or the image will be analyzed by a computer, the additional accuracy may be useful. Even if human eye cannot detect any differences, computer analysis may recognize the difference. The requirement of the spatial resolution depends both on the usage of the image and the image content. If the default printing (or display) size of the image is known, the scanning resolution can be chosen accordingly so that the pixels are not seen and the image appearance is not jagged (blocky). However, the final reproduction size of the image is not always known but images are often achieved just for "later use". Thus, once the image is digitized it will most likely (according to Murphy's law) be later edited and enlarged beyond what was allowed by the original resolution. The image content sets also some requirements to the resolution. If the image has very fine structure exceeding the sampling resolution, it may cause so-called aliasing effect where the digitized image has patterns that does not exists in the original.

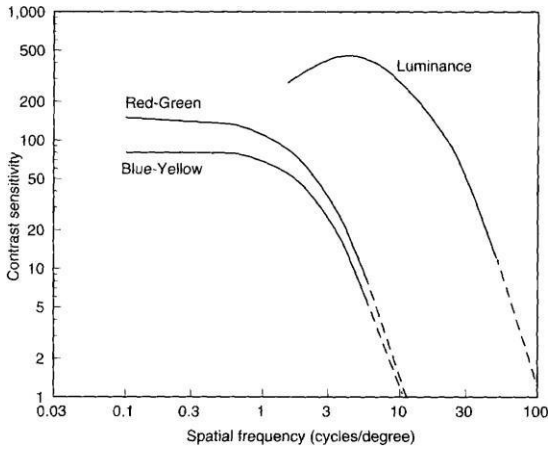


Fig: Sensitivity of the eye to the intensity changes

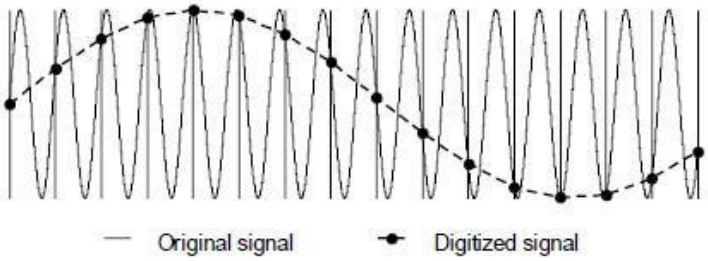


Figure : Aliasing effect for 1-dimensional signal.

Figure : Aliasing effect (Moiré) for an image.

Image processing Techniques:

(i) Point Operations: map each input pixel to output pixel intensity according to an intensity transformation. A simple linear point operation which maps the input gray level f(m,n) to an output gray level g(m,n)is given by:

$$g(m, n) = af(m, n) + b$$

Where *a* and *b* are chosen to achieve a desired intensity variation in the image. Note that the output g(m.n) here depends only on the input f(m,n).

(ii) Local Operations: determine the output pixel intensity as some function of a relatively small neighborhood of input pixels in the vicinity of the output location. A general linear operator can be expressed as weighted of picture elements within a local neighborhood *N*.

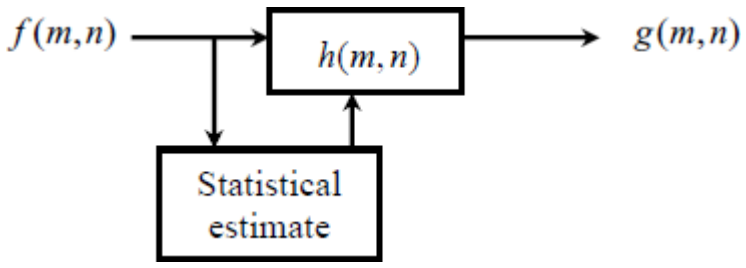
$$g(m,n) = \sum_{k,l \in N} a_{k,l} f(m-k, n-l)$$

Simple local smoothing (for noise reduction) and sharpening (for deploring or edge enhancement) operators can be both linear and non-linear.

(iii) Global Operations: the outputs depend on all input pixels values. If linear, global operators can be expressed using two-dimensional convolution.

$$g(m,n) = f(m,n) * h(m,n) = \sum_{k,l \in N} h(k,l) f(m-k, n-l)$$

(iv) Adaptive Filters: whose coefficients depend on the input image



(v) Non-Linear Filters:

- Median/order statistics
- Non-linear local operations
- Homomorphic filters

$$g(m,n) = \prod_{k,l \in N} [f(m-k, n-l)]^{h(k,l)}$$

In addition to enhancement and restoration, image processing generally includes issues of representations, spatial sampling and intensity quantization, compression or coding and

segmentation. As part of computer vision, image processing leads to feature extraction and pattern recognition or scene analysis.

UNIT–II

IMAGE

TRANSFORMS

1. UNITARY TRANSFORMS

1.1 One dimensional signals

For a one dimensional sequence $\{f(x), 0 \le x \le N-1\}$ represented as a vector $\underline{f} = [f(0)\, f(1) \ldots f(N-1)]^T$ of size N, a transformation may be written as

$$\underline{g} = \underline{T} \cdot \underline{f} \Rightarrow g(u) = \sum_{x=0}^{N-1} T(u,x) f(x), 0 \le u \le N-1$$

where $g(u)$ is the transform (or transformation) of $f(x)$, and $T(u,x)$ is the so called **forward transformation kernel**. Similarly, the inverse transform is the relation

$$f(x) = \sum_{u=0}^{N-1} I(x,u) g(u),\ 0 \le x \le N-1$$

or written in a matrix form

$$\underline{f} = \underline{I} \cdot \underline{g} = \underline{T}^{-1} \cdot \underline{g}$$

where $I(x,u)$ is the so called **inverse transformation kernel**.

If

$$\underline{I} = \underline{T}^{-1} = \underline{T}^{*T}$$

the matrix $\underline{T}$ is called **unitary**, and the transformation is called unitary as well. It can be proven that the columns (or rows) of an $N \times N$ unitary matrix are orthonormal and therefore, form a complete set of **basis vectors** in the $N-$dimensional vector space.

In that case

$$\underline{f} = \underline{T}^{*T} \cdot \underline{g} \Rightarrow f(x) = \sum_{u=0}^{N-1} T^*(u,x) g(u)$$

The columns of $\underline{T}^{*T}$, that is, the vectors $\underline{T}_u^* = [T^*(u,0)\, T^*(u,1) \ldots T^*(u,N-1)]^T$ are called the **basis vectors** of $\underline{T}$.

1.2 Two dimensional signals (images)

As a one dimensional signal can be represented by an orthonormal set of **basis vectors**, an image can also be expanded in terms of a discrete set of **basis arrays** called basis images through a **two dimensional (image) transform**.

For an $N \times N$ image $f(x,y)$ the forward and inverse transforms are given below

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

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

where, again, $T(u,v,x,y)$ and $I(x,y,u,v)$ are called the **forward and inverse transformation kernels**, respectively.

The forward kernel is said to be **separable** if

$$T(u,v,x,y) = T_1(u,x) T_2(v,y)$$

It is said to be **symmetric** if T_1 is functionally equal to T_2 such that

$$T(u,v,x,y) = T_1(u,x)T_1(v,y)$$

The same comments are valid for the inverse kernel.

If the kernel $T(u,v,x,y)$ of an image transform is separable and symmetric, then the transform $g(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} T(u,v,x,y)f(x,y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} T_1(u,x)T_1(v,y)f(x,y)$ can be written in matrix form as follows

$$\underline{g} = \underline{T}_1 \cdot \underline{f} \cdot \underline{T}_1^T$$

where $\underline{f}$ is the original image of size $N \times N$, and $\underline{T}_1$ is an $N \times N$ transformation matrix with elements $t_{ij} = T_1(i,j)$. If, in addition, $\underline{T}_1$ is a unitary matrix then the transform is called **separable unitary** and the original image is recovered through the relationship

$$\underline{f} = \underline{T}_1^{*T} \cdot \underline{g} \cdot \underline{T}_1^*$$

1.3 Fundamental properties of unitary transforms

1.3.1 The property of energy preservation

For a unitary transformation

$$\underline{g} = \underline{T} \cdot \underline{f}$$

and

$$\underline{g}^{*T} = (\underline{T}^* \cdot \underline{f}^*)^T = \underline{f}^{*T} \cdot \underline{T}^{*T}$$

and therefore, by using the relation $\underline{T}^{-1} = \underline{T}^{*T}$ we have that

$$\underline{g}^{*T}\,\underline{g} = (\underline{f}^{*T} \cdot \underline{T}^{*T})(\underline{T} \cdot \underline{f}) = \underline{f}^{*T} \cdot (\underline{T}^{*T}\,\underline{T}) \cdot \underline{f} = \underline{f}^{*T}\,\underline{f} \Rightarrow \|\underline{g}\|^2 = \|\underline{f}\|^2$$

Thus, a unitary transformation preserves the signal energy. This property is called energy preservation property.

This means that every unitary transformation is simply a rotation of the vector $\underline{f}$ in the N-dimensional vector space.

For the 2-D case the energy preservation property is written as

$$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} |f(x,y)|^2 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} |g(u,v)|^2$$

1.3.2 The property of energy compaction

Most unitary transforms pack a large fraction of the energy of the image into relatively few of the transform coefficients. This means that relatively few of the transform coefficients have significant values and these are the coefficients that are close to the origin (small index coefficients).

This property is very useful for compression purposes.

2. THE TWO DIMENSIONAL FOURIER TRANSFORM

2.1 Continuous space and continuous frequency

The Fourier transform is extended to a function $f(x,y)$ of two variables. If $f(x,y)$ is continuous and integrable and $F(u,v)$ is integrable, the following Fourier transform pair exists:

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

$$f(x,y)=\frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\int F(u,v)e^{j2\pi(ux+vy)}dudv$$

In general $F(u,v)$ is a complex-valued function of two real frequency variables u,v and hence, it can be written as:

$$F(u,v)=R(u,v)+jI(u,v)$$

The amplitude spectrum, phase spectrum and power spectrum, respectively, are defined as follows.

$$|F(u,v)|=\sqrt{R^2(u,v)+I^2(u,v)}$$

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

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

2.2 Discrete space and continuous frequency

For the case of a discrete sequence $f(x,y)$ of infinite duration we can define the 2-D discrete space Fourier transform pair as follows

$$F(u,v)=\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}f(x,y)e^{-j(xu+vy)}$$

$$f(x,y)=\frac{1}{(2\pi)^2}\int_{u=-\pi}^{\pi}\int_{v=-\pi}^{\pi}F(u,v)e^{j(xu+vy)}dudv$$

$F(u,v)$ is again a complex-valued function of two real frequency variables u,v and it is periodic with a period $2\pi\times 2\pi$, that is to say $F(u,v)=F(u+2\pi,v)=F(u,v+2\pi)$

The Fourier transform of $f(x,y)$ is said to converge uniformly when $F(u,v)$ is finite and

$$\lim_{N_1\to\infty}\lim_{N_2\to\infty}\sum_{x=-N_1}^{N_1}\sum_{y=-N_2}^{N_2}f(x,y)e^{-j(xu+vy)}=F(u,v) \text{ for all } u,v.$$

When the Fourier transform of $f(x,y)$ converges uniformly, $F(u,v)$ is an analytic function and is infinitely differentiable with respect to u and v.

2.3 Discrete space and discrete frequency: The two dimensional Discrete Fourier Transform (2-D DFT)

If $f(x,y)$ is an $M\times N$ array, such as that obtained by sampling a continuous function of two dimensions at dimensions M and N on a rectangular grid, then its two dimensional Discrete Fourier transform (DFT) is the array given by

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

$u=0,\ldots,M-1,\ v=0,\ldots,N-1$

and the inverse DFT (IDFT) is

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

When images are sampled in a square array, $M=N$ and

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

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

It is straightforward to prove that the two dimensional Discrete Fourier Transform is separable, symmetric and unitary.

2.3.1 Properties of the 2-D DFT

Most of them are straightforward extensions of the properties of the 1-D Fourier Transform. Advise any introductory book on Image Processing.

2.3.2 The importance of the phase in 2-D DFT. Image reconstruction from amplitude or phase only.

The Fourier transform of a sequence is, in general, complex-valued, and the unique representation of a sequence in the Fourier transform domain requires both the phase and the magnitude of the Fourier transform. In various contexts it is often desirable to reconstruct a signal from only partial domain information. Consider a 2-D sequence $f(x,y)$ with Fourier transform $F(u,v)=\Im\{f(x,y)\}$ so that

$$F(u,v)=\Im\{f(x,y\}=|F(u,v)|e^{j\phi_f(u,v)}$$

It has been observed that a straightforward signal synthesis from the Fourier transform phase $\phi_f(u,v)$ alone, often captures most of the intelligibility of the original image $f(x,y)$ (**why?**). A straightforward synthesis from the Fourier transform magnitude $|F(u,v)|$ alone, however, does not generally capture the original signal's intelligibility. The above observation is valid for a large number of signals (or images). To illustrate this, we can synthesise the phase-only signal $f_p(x,y)$ and the magnitude-only signal $f_m(x,y)$ by

$$f_p(x,y)=\Im^{-1}\left[1e^{j\phi_f(u,v)}\right]$$

$$f_m(x,y)=\Im^{-1}\left[|F(u,v)|e^{j0}\right]$$

An experiment which more dramatically illustrates the observation that phase-only signal synthesis captures more of the signal intelligibility than magnitude-only synthesis, can be performed as follows.

Consider two images $f(x,y)$ and $g(x,y)$. From these two images, we synthesise two other images $f_1(x,y)$ and $g_1(x,y)$ by mixing the amplitudes and phases of the original images as follows:

$$f_1(x,y)=\Im^{-1}\left[|G(u,v)|e^{j\phi_f(u,v)}\right]$$

$$g_1(x,y)=\Im^{-1}\left[|F(u,v)|e^{j\phi_g(u,v)}\right]$$

In this experiment $f_1(x,y)$ captures the intelligibility of $f(x,y)$, while $g_1(x,y)$ captures the intelligibility of $g(x,y)$

3. THE DISCRETE COSINE TRANSFORM (DCT)

3.1 One dimensional signals

This is a transform that is similar to the Fourier transform in the sense that the new independent variable represents again frequency. The DCT is defined below.

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

with $a(u)$ a parameter that is defined below.

$$a(u) = \begin{cases} \sqrt{1/N} & u = 0 \\ \sqrt{2/N} & u = 1,\ldots,N-1 \end{cases}$$

The inverse DCT (IDCT) is defined below.

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

3.2 Two dimensional signals (images)

For 2-D signals it is defined as

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

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

$a(u)$ is defined as above and $u, v = 0,1,\ldots,N-1$

3.3 Properties of the DCT transform

- The DCT is a real transform. This property makes it attractive in comparison to the Fourier transform.
- The DCT has excellent energy compaction properties. For that reason it is widely used in image compression standards (as for example JPEG standards).
- There are fast algorithms to compute the DCT, similar to the FFT for computing the DFT.

4. WALSH TRANSFORM (WT)

4.1 One dimensional signals

This transform is slightly different from the transforms you have met so far. Suppose we have a function $f(x), x = 0,\ldots,N-1$ where $N = 2^n$ and its Walsh transform $W(u)$.

If we use binary representation for the values of the independent variables x and u we need n bits to represent them. Hence, for the binary representation of x and u we can write:
$(x)_{10} = (b_{n-1}(x)b_{n-2}(x)\ldots b_0(x))_2$, $(u)_{10} = (b_{n-1}(u)b_{n-2}(u)\ldots b_0(u))_2$
with $b_i(x)$ 0 or 1 for $i = 0,\ldots,n-1$.

Example
If $f(x), x = 0,\ldots,7,$ (8 samples) then $n = 3$ and for $x = 6$, $6 = (110)_2 \Rightarrow b_2(6) = 1, b_1(6) = 1, b_0(6) = 0$

We define now the 1-D Walsh transform as

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left[\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}\right] \text{ or}$$

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}$$

The array formed by the Walsh kernels is again a symmetric matrix having orthogonal rows and columns. Therefore, the Walsh transform is and its elements are of the form $T(u,x) = \prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$. You can immediately observe that $T(u,x) = -1$ or 1 depending on the values of $b_i(x)$ and $b_{n-1-i}(u)$. If the Walsh transform is written in a matrix form

$$\underline{W} = \underline{T}\cdot\underline{f}$$

the rows of the matrix $\underline{T}$ which are the vectors $[T(u,0)\,T(u,1)\ldots T(u,N-1)]$ have the form of square waves. As the variable u (which represents the index of the transform) increases, the corresponding square wave's "frequency" increases as well. For example for $u = 0$ we see that $(u)_{10} = (b_{n-1}(u)b_{n-2}(u)\ldots b_0(u))_2 = (00\ldots0)_2$ and hence, $b_{n-1-i}(u) = 0$, for any i. Thus, $T(0,x) = 1$ and $W(0) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)$. We see that the first element of the Walsh transform in the mean of the original function $f(x)$ (the DC value) as it is the case with the Fourier transform.

The inverse Walsh transform is defined as follows.

$$f(x) = \sum_{u=0}^{N-1} W(u)\left[\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}\right] \text{ or}$$

$$f(x) = \sum_{u=0}^{N-1} W(u)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}$$

4.2 Two dimensional signals

The Walsh transform is defined as follows for two dimensional signals.

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\left[\prod_{i=0}^{n-1}(-1)^{(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}\right] \text{ or}$$

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

The inverse Walsh transform is defined as follows for two dimensional signals.

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}W(u,v)\left[\prod_{i=0}^{n-1}(-1)^{(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}\right] \text{ or}$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}W(u,v)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

4.3 Properties of the Walsh Transform

- Unlike the Fourier transform, which is based on trigonometric terms, the Walsh transform consists of a series expansion of basis functions whose values are only -1 or 1 and they have the form of square waves. These functions can be implemented more efficiently in a digital environment than the exponential basis functions of the Fourier transform.
- The forward and inverse Walsh kernels are identical except for a constant multiplicative factor of $\frac{1}{N}$ for 1-D signals.
- The forward and inverse Walsh kernels are identical for 2-D signals. This is because the array formed by the kernels is a symmetric matrix having orthogonal rows and columns, so its inverse array is the same as the array itself.
- The concept of frequency exists also in Walsh transform basis functions. We can think of frequency as the number of zero crossings or the number of transitions in a basis vector and we call this number **sequency**. The Walsh transform exhibits the property of energy compaction as all the transforms that we are currently studying. (**why**?)
- For the fast computation of the Walsh transform there exists an algorithm called **Fast Walsh Transform (FWT).** This is a straightforward modification of the FFT. Advise any introductory book for your own interest.

5. HADAMARD TRANSFORM (HT)

5.1 Definition

In a similar form as the Walsh transform, the 2-D Hadamard transform is defined as follows.

Forward

$$H(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\left[\prod_{i=0}^{n-1}(-1)^{(b_i(x)b_i(u)+b_i(y)b_i(v))}\right],\ N=2^n \text{ or}$$

$$H(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_i(u)+b_i(y)b_i(v))}$$

Inverse

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}H(u,v)\left[\prod_{i=0}^{n-1}(-1)^{(b_i(x)b_i(u)+b_i(y)b_i(v))}\right] \text{ etc.}$$

5.2 Properties of the Hadamard Transform

- **Most of the comments made for Walsh transform are valid here.**
- The Hadamard transform differs from the Walsh transform only in the order of basis functions. The order of basis functions of the Hadamard transform **does not** allow the fast computation of it by using a straightforward modification of the FFT. An extended version of the Hadamard transform is the **Ordered Hadamard Transform** for which a fast algorithm called **Fast Hadamard Transform (FHT)** can be applied.
- An important property of Hadamard transform is that, letting H_N represent the matrix of order N, the recursive relationship is given by the expression

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

6. KARHUNEN-LOEVE (KLT) or HOTELLING TRANSFORM

The Karhunen-Loeve Transform or KLT was originally introduced as a series expansion for continuous random processes by Karhunen and Loeve. For discrete signals Hotelling first studied what was called a method of principal components, which is the discrete equivalent of the KL series expansion. Consequently, the KL transform is also called the Hotelling transform or the method of principal components. **The term KLT is the most widely used**.

Consider a population of random column vectors of the form

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

The **mean vector** of the population is defined as

$$\underline{m}_x = E\{x\}$$

The operator E refers to the expected value of the population, calculated theoretically using the probability density functions (pdf) of the elements x_i.

The covariance matrix of the population is defined as

$$\underline{C}_x = E\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\}$$

The operator E is now calculated theoretically using the probability density functions (pdf) of the elements x_i and the joint probability density functions between the elements x_i and x_j.

Because $\underline{x}$ is n-dimensional, $\underline{C}_x$ and $(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T$ are matrices of order $n \times n$. The element c_{ii} of $\underline{C}_x$ is the variance of x_i, and the element c_{ij} of $\underline{C}_x$ is the covariance between the elements x_i and x_j. If the elements x_i and x_j are uncorrelated, their covariance is zero and, therefore, $c_{ij} = c_{ji} = 0$. The covariance matrix $\underline{C}_x$ can be written as follows.

$$\underline{C}_x = E\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\} = E\{(\underline{x} - \underline{m}_x)(\underline{x}^T - \underline{m}_x^T)\} = E\{\underline{x}\underline{x}^T - \underline{x}\underline{m}_x^T - \underline{m}_x\underline{x}^T + \underline{m}_x\underline{m}_x^T\}$$

It can be easily shown that

$$\underline{x}\underline{m}_x^T = \underline{m}_x \underline{x}^T$$

Therefore,

$$E\{\underline{x}\underline{x}^T - \underline{x}\underline{m}_x^T - \underline{m}_x\underline{x}^T + \underline{m}_x\underline{m}_x^T\} = E\{\underline{x}\underline{x}^T - \underline{m}_x\underline{x}^T - \underline{m}_x\underline{x}^T + \underline{m}_x\underline{m}_x^T\} = E\{\underline{x}\underline{x}^T - 2\underline{m}_x\underline{x}^T + \underline{m}_x\underline{m}_x^T\}$$
$$= E\{\underline{x}\underline{x}^T\} - E\{2\underline{m}_x\underline{x}^T\} + E\{\underline{m}_x\underline{m}_x^T\}$$

Since the vector $\underline{m}_x$ and the matrix $\underline{m}_x\underline{m}_x^T$ contain constant quantities, we can write

$$E\{\underline{x}\underline{x}^T - 2\underline{m}_x\underline{x}^T + \underline{m}_x\underline{m}_x^T\} = E\{\underline{x}\underline{x}^T\} - 2\underline{m}_x E\{\underline{x}^T\} + \underline{m}_x\underline{m}_x^T$$

Knowing that

$$E\{\underline{x}^T\} = \underline{m}_x^T$$

we have

$$\underline{C}_x = E\{\underline{x}\underline{x}^T\} - 2\underline{m}_x E\{\underline{x}^T\} + \underline{m}_x\underline{m}_x^T = E\{\underline{x}\underline{x}^T\} - 2\underline{m}_x\underline{m}_x^T + \underline{m}_x\underline{m}_x^T \Rightarrow$$
$$\underline{C}_x = E\{\underline{x}\underline{x}^T\} - \underline{m}_x\underline{m}_x^T$$

For M vectors from a random population, where M is large enough, the mean vector $\underline{m}_x$ and the covariance matrix $\underline{C}_x$ can be approximately calculated from the available vectors by using the following relationships where all the expected values are approximated by summations

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

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

Very easily it can be seen that $\underline{C}_x$ is real and symmetric. Let $\underline{e}_i$ and λ_i, $i = 1,2,\ldots,n$, be a set of orthonormal eigenvectors and corresponding eigenvalues of $\underline{C}_x$, arranged in descending order so that $\lambda_i \geq \lambda_{i+1}$ for $i = 1,2,\ldots,n-1$. Suppose that $\underline{e}_i$ are column vectors.

Let $\underline{A}$ be a matrix whose rows are formed from the eigenvectors of $\underline{C}_x$, ordered so that the first row of $\underline{A}$ is the eigenvector corresponding to the largest eigenvalue, and the last row the eigenvector corresponding to the smallest eigenvalue. Therefore,

$$\underline{A} = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_n^T \end{bmatrix} \text{ and } \underline{A}^T = \begin{bmatrix} \underline{e}_1 & \underline{e}_2 & \cdots & \underline{e}_n \end{bmatrix}$$

Suppose that $\underline{A}$ is a transformation matrix that maps the vectors $\underline{x}$ into vectors $\underline{y}$ by using the following transformation

$$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$$

The above transform is called the **Karhunen-Loeve** or **Hotelling** transform. The mean of the $\underline{y}$ vectors resulting from the above transformation is zero, since

$$E\{\underline{y}\} = E\{\underline{A}(\underline{x} - \underline{m}_x)\} = \underline{A}E\{\underline{x} - \underline{m}_x\} = \underline{A}(E\{\underline{x}\} - \underline{m}_x) = \underline{A}(\underline{m}_x - \underline{m}_x) = \underline{0} \Rightarrow$$
$$\underline{m}_y = \underline{0}$$

The covariance matrix of the $\underline{y}$ vectors is

$$\underline{C}_y = E\{(\underline{y} - \underline{m}_y)(\underline{y} - \underline{m}_y)^T\} = E\{\underline{y}\underline{y}^T\}$$

Using the relationships

$$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$$

$$\underline{y}^T = [\underline{A}(\underline{x} - \underline{m}_x)]^T = (\underline{x} - \underline{m}_x)^T \underline{A}^T$$

we get

$$\underline{y}\underline{y}^T = \underline{A}(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T \underline{A}^T \Rightarrow$$

$$E\{\underline{y}\underline{y}^T\} = E\{\underline{A}(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T \underline{A}^T\} = \underline{A}E\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\}\underline{A}^T$$

$$\underline{C}_y = \underline{A}\underline{C}_x\underline{A}^T$$

$$\underline{C}_x\underline{A}^T = \underline{C}_x[\underline{e}_1 \quad \underline{e}_2 \quad \dots \quad \underline{e}_n] = [\lambda_1\underline{e}_1 \quad \lambda_2\underline{e}_2 \quad \dots \quad \lambda_n\underline{e}_n]$$

$$\underline{C}_y = \underline{A}\underline{C}_x\underline{A}^T = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_n^T \end{bmatrix} [\lambda_1\underline{e}_1 \quad \lambda_2\underline{e}_2 \quad \dots \quad \lambda_n\underline{e}_n]$$

Because $\underline{e}_i$ is a set of orthonormal eigenvectors we have that:

$$\underline{e}_i^T\underline{e}_i = 1, i = 1, \dots, n$$

$$\underline{e}_i^T\underline{e}_j = 1, i, j = 1, \dots, n$$

and therefore, $\underline{C}_y$ is a diagonal matrix whose elements along the main diagonal are the eigenvalues of $\underline{C}_x$

$$\underline{C}_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

The off-diagonal elements of the covariance matrix of the population of vectors $\underline{y}$ are 0, and therefore, the elements of the $\underline{y}$ vectors are uncorrelated.

Lets try to reconstruct any of the original vectors $\underline{x}$ from its corresponding $\underline{y}$. Because the rows of $\underline{A}$ are orthonormal vectors we have

$$\underline{A}\underline{A}^T = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_n^T \end{bmatrix} [\underline{e}_1 \quad \underline{e}_2 \quad \dots \quad \underline{e}_n] = I$$

with I the unity matrix. Therefore, $\underline{A}^{-1} = \underline{A}^T$, and any vector $\underline{x}$ can by recovered from its corresponding vector $\underline{y}$ by using the relation

$$\underline{x} = \underline{A}^T\underline{y} + \underline{m}_x$$

Suppose that instead of using all the eigenvectors of $\underline{C}_x$ we form matrix $\underline{A}_K$ from the K eigenvectors corresponding to the K largest eigenvalues,

$$\underline{A}_K = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_K^T \end{bmatrix}$$

yielding a transformation matrix of order $K \times n$. The $\underline{y}$ vectors would then be K dimensional, and the reconstruction of any of the original vectors would be approximated by the following relationship

$$\hat{\underline{x}} = \underline{A}_K^T \underline{y} + \underline{m}_x$$

The mean square error between the perfect reconstruction $\underline{x}$ and the approximate reconstruction $\hat{\underline{x}}$ is given by the expression

$$e_{ms} = \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{K} \lambda_j = \sum_{j=K+1}^{n} \lambda_j .$$

By using $\underline{A}_K$ instead of $\underline{A}$ for the KL transform we achieve compression of the available data.

6.2 Properties of the Karhunen-Loeve transform

Despite its favourable theoretical properties, the KLT is not used in practice for the following reasons.

- Its basis functions depend on the covariance matrix of the image, and hence they have to recomputed and transmitted for every image.
- Perfect decorrelation is not possible, since images can rarely be modelled as realisations of ergodic fields.
- There are no fast computational algorithms for its implementation.

UNIT-III

IMAGE ENHANCEMENT

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 board categories.

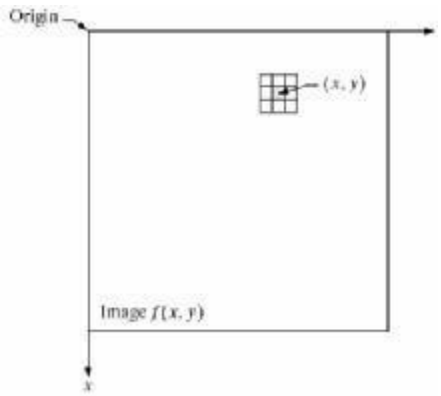
1. Spatial domain methods
2. Frequency domain methods

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

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

Where f(x,y)- input image, T- operator on f, defined over some neighborhood of f(x,y)
and 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 IXI. 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)

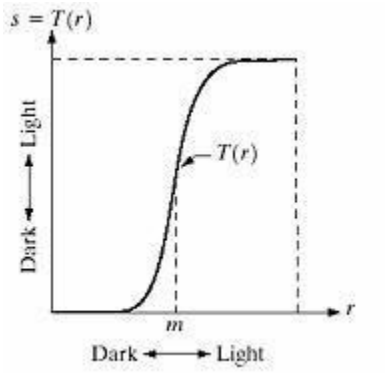
Because enhancement at any point in an image deepens only on the gray level at that point, technique in this category are referred to as point processing.

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

Point Processing:

(i) Contract stretching:

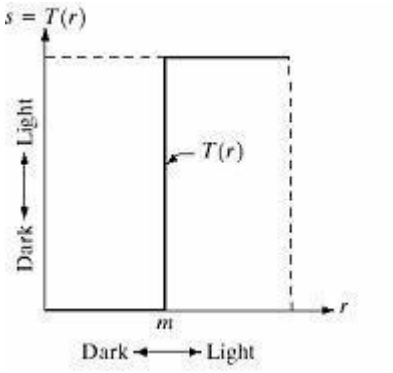
It produces an image of higher contrast than the original one. The operation is performed by darkening the levels below m and brightening the levels above m in the original image.



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.

(ii) 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.

(i) Image Negative:

The negative of in image with gray level in the range [0, l, -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.

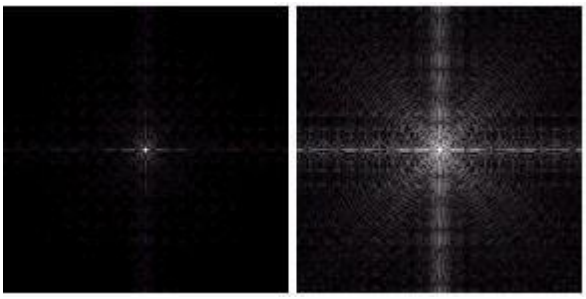
(ii) Log transformations:

The general form of the log transformation is **s= c log(1+r)**

Where c- constant and $r \geq o$

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 transformations 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



(iii) Power Law Transformation:

Power law transformations has the basic form

$$S=cr^{y}$$

Where c and y are positive constants.

Power law curves with fractional values of y 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 y.

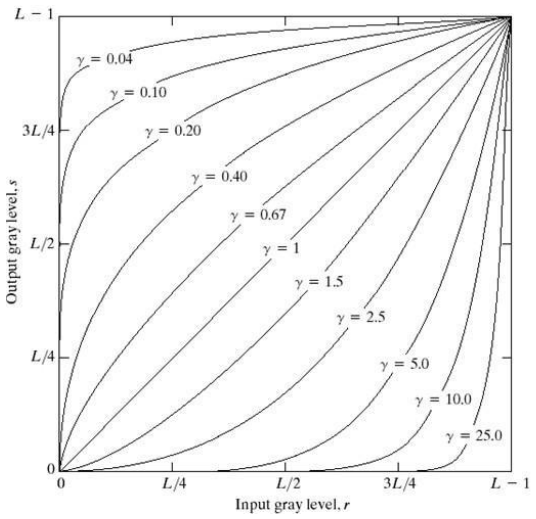


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

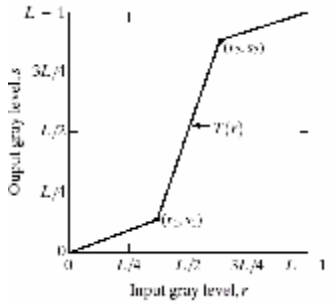
A variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power law response phenomenon is called gamma correction. For eg-CRT devices have intensity to voltage response that is a power function. Gamma correction is important 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 contract 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.

(i) 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 (r1,s1) and (r2,s2) control the shape of the curve

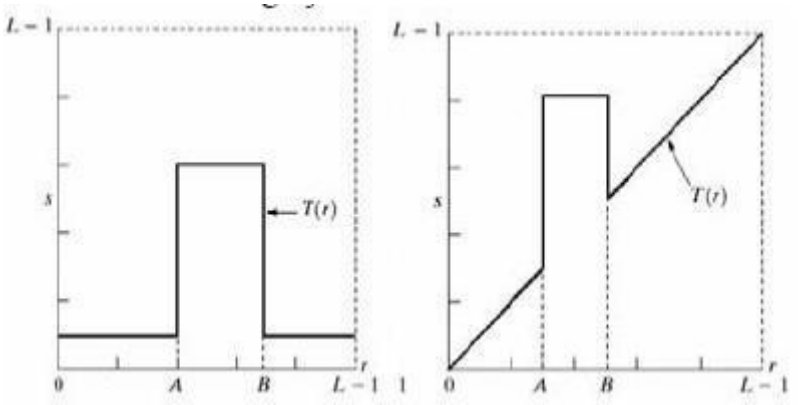
a) If r1=r2 and s1=s2, the transformation is a linear function that deduces no change in gray levels.

b) If r1=s1, s1=0 , and s2=L-1, then the transformation become a thresholding function that creates a binary image

c) Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray value of the output image thus effecting its contract.

Generally r1≤ r2 and s1≤ s2 so that the function is single valued and monotonically increasing.

(ii) 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.

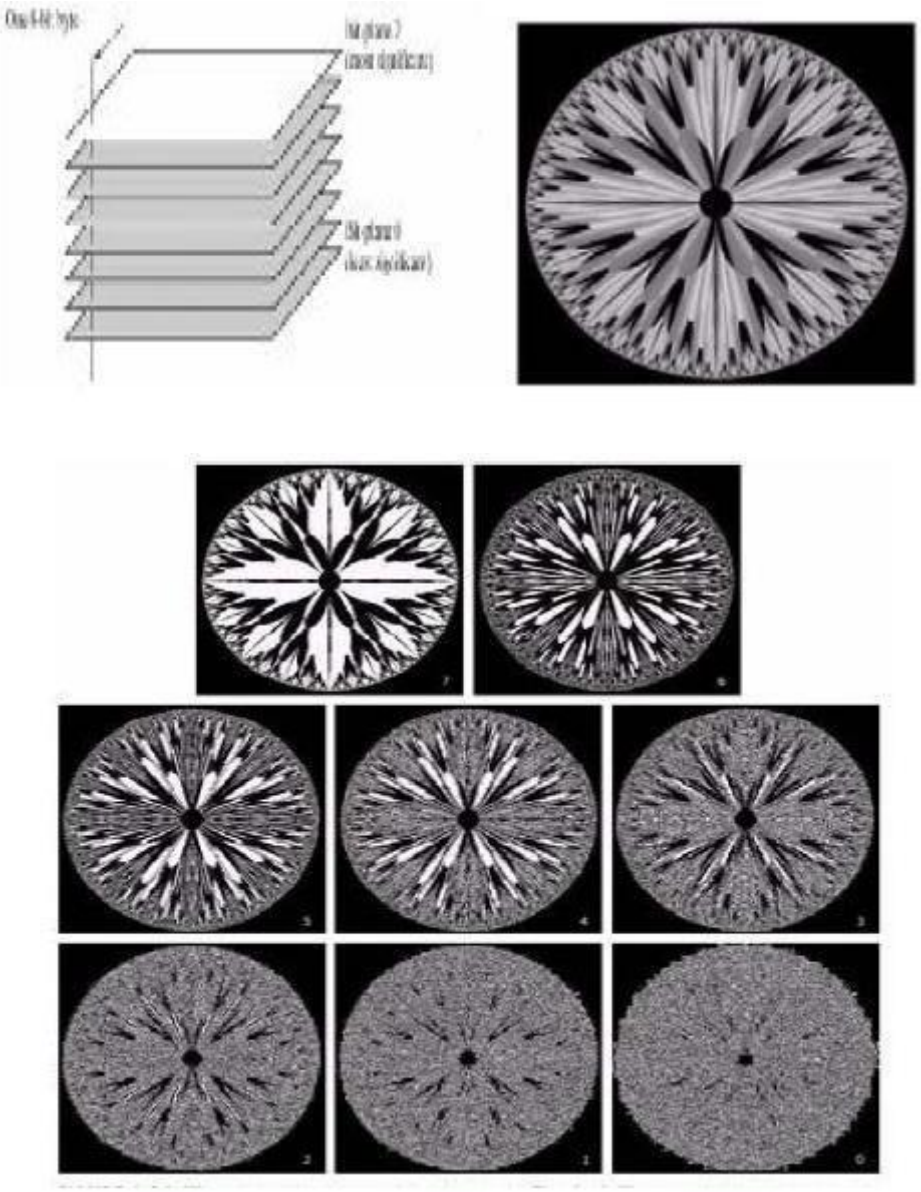
There are two ways of doing this-

(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.

(iii) 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

$$H(rk)=nk$$

where rk is the kth gray level and nk is the number of pixels in the image having the level rk..

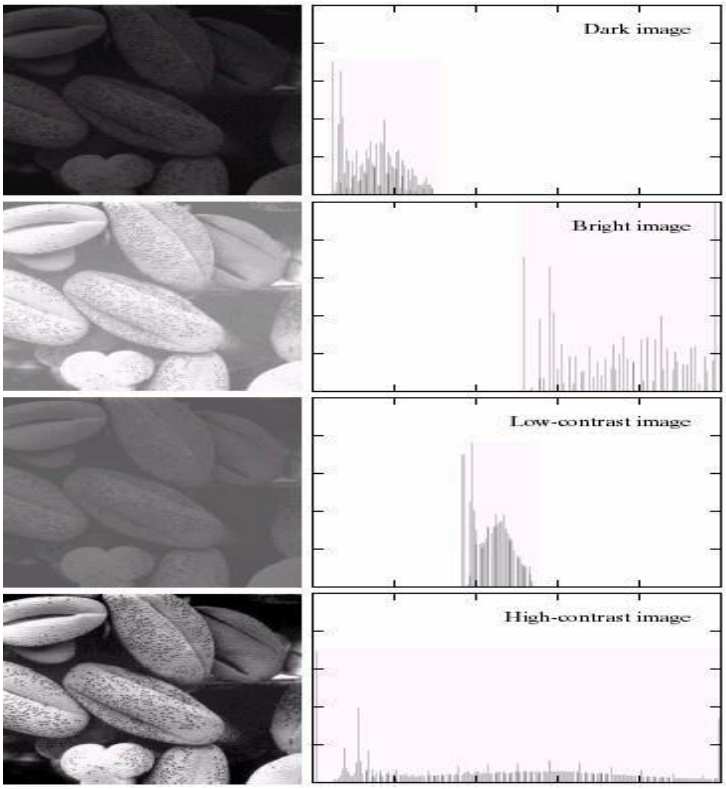
A normalized histogram is given by the equation

$$p(rk)=nk/n \text{ for } k=0,1,2,\ldots..,L-1$$

P(rk) gives the estimate of the probability of occurrence of gray level rk.

The sum of all components of a normalized histogram is equal to 1.

The histogram plots are simple plots of H(rk)=nk versus rk.



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 baised 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 repressing black and r=1 representing white. The transformation function is of the form

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

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

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

Thus the PDF of the transformed variable s is the determined by the gray levels PDF of the input image and by the chosen transformations function.

A transformation function of a particular importance in image processing

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

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 Ps we may get

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

An important point here is that Tr depends on Pr(r) but the resulting Ps(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 rk in an image as approximated

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

N is the total number of the pixels in an image.

nk is the number of the pixels that have gray level rk.

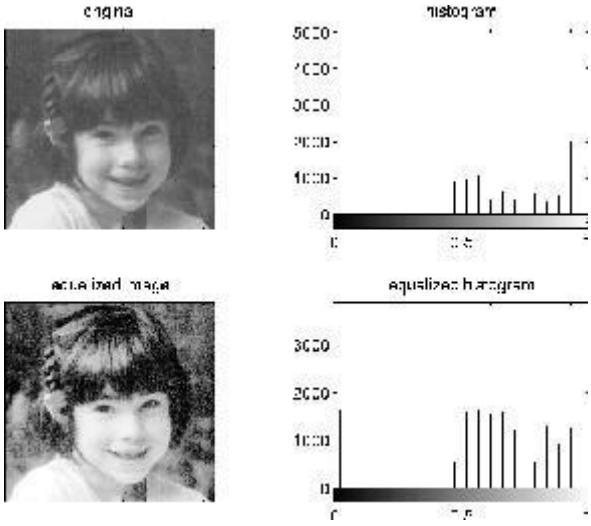
L is the total number of possible gray levels in the image.

The discrete transformation function is given by

$$s_k = T(r_k) = \sum_{i=0}^{k} \frac{n_i}{N}$$

$$= \sum_{i=0}^{k} P_r(r_i).$$

Thus a processed image is obtained by mapping each pixel with levels rk in the input image into a corresponding pixel with level sk in the output image. A plot of Pr (rk) versus rk is called a histogram. The transformation function given by the above equation is the 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 use to generate a processed image that has a specified histogram is called histogram matching or histogram specification.

Algorithm:

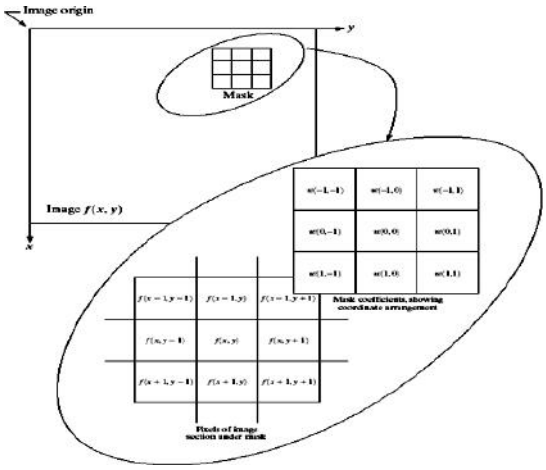
1. Compute sk=Pf (k), k = 0, …, L-1, the cumulative normalized histogram of f .

2. Compute G(k), k = 0, …, L-1, the transformation function, from the given histogram hz 3. Compute G-1(sk) for each k = 0, …, L-1 using an iterative method (iterate on z), or in effect, directly compute G-1(Pf (k))

4. Transform f using G-1(Pf (k)) .

Basic filtering through the enhancement:

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 pixel. 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 linaer 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 liner filtering with the filter mask at point (x,y) in the image is

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

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 it centered at (x,y) when the computation of sum of products takes place. For a mask of size MxN we assume m=2a+1 and n=2b+1, where a and b are nonnegative integers. It shows that

all the masks are of add size. In the general liner filtering of an image of size f of size M*N with a filter mask of size m*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, M-1 and y=0,1,2 ,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= W,Z,+W2, Z2 +….+ Wmn Zmn

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 form 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 o's & or padding by replicating rows and columns. The padding is removed at the end of the process.

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN:

Basics 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

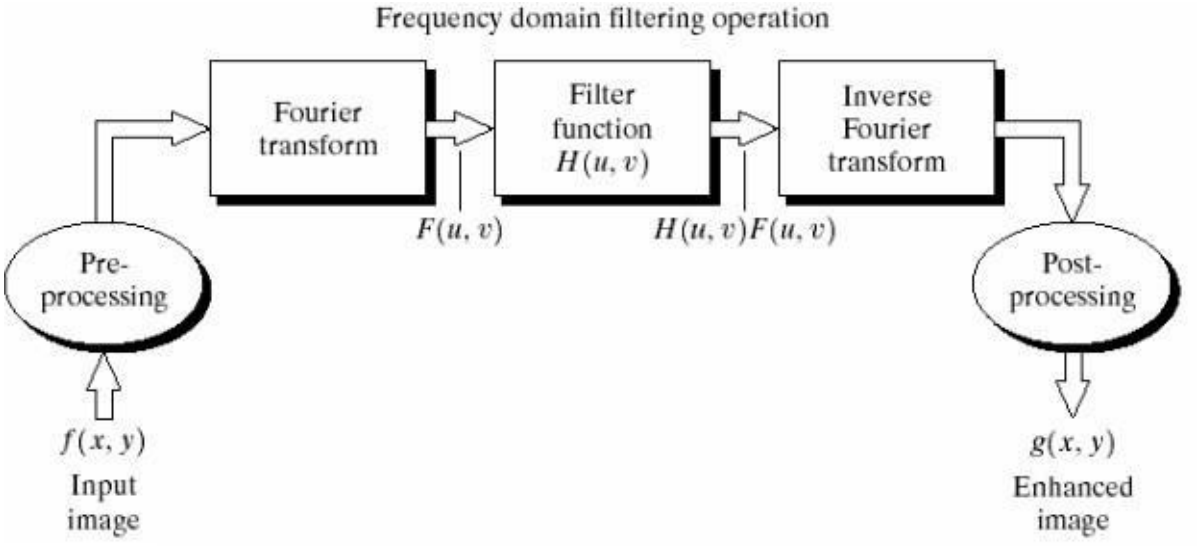
ii) Compute F(u,v), Fourier Transform of the image

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.

Image smoothing:

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

(i) Ideal Low pass filter:

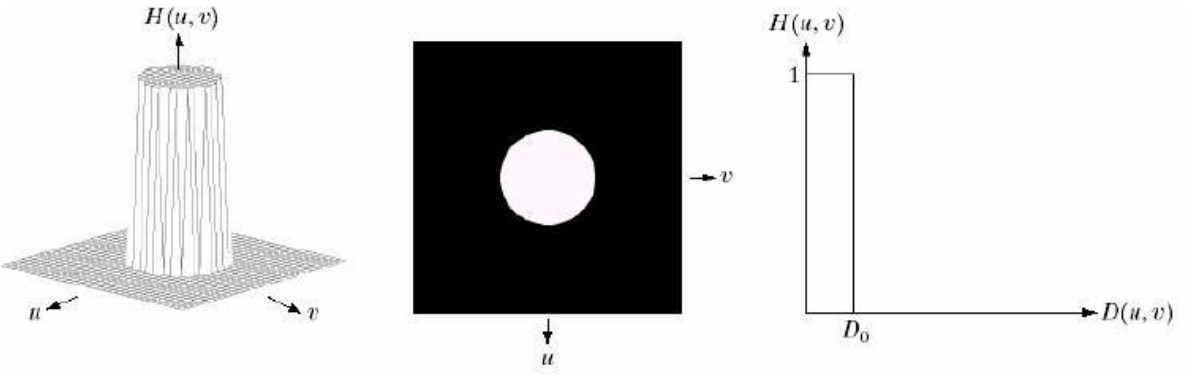
It is the simplest of all the three filters. It cuts of all high frequency component of the Fourier transform that are at a distance greater that a specified distance D0 form the origin of the transform. it is called a two – dimensional ideal low pass filter (ILPF) and has the transfer function

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

Where D0 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*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

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

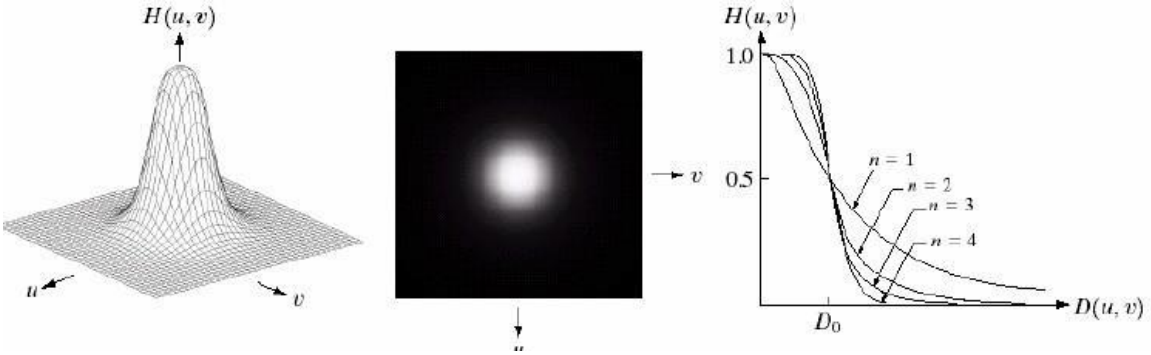
Because it is ideal case. So all frequency inside the circle are passed without any attenuation where as all frequency outside the circle are completely attenuated. For an ideal low pass filter cross section, the point of transition between H (u,v) =1 and H(u,v)=0 is called of the " cut of frequency".



(ii) 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 Do 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).

(iii) 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

σ = D0- specified cut off frequency

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

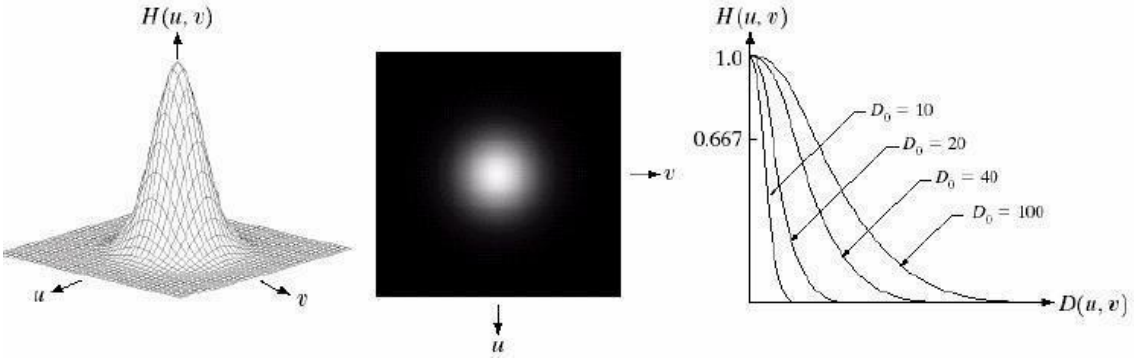


Image Sharpening:

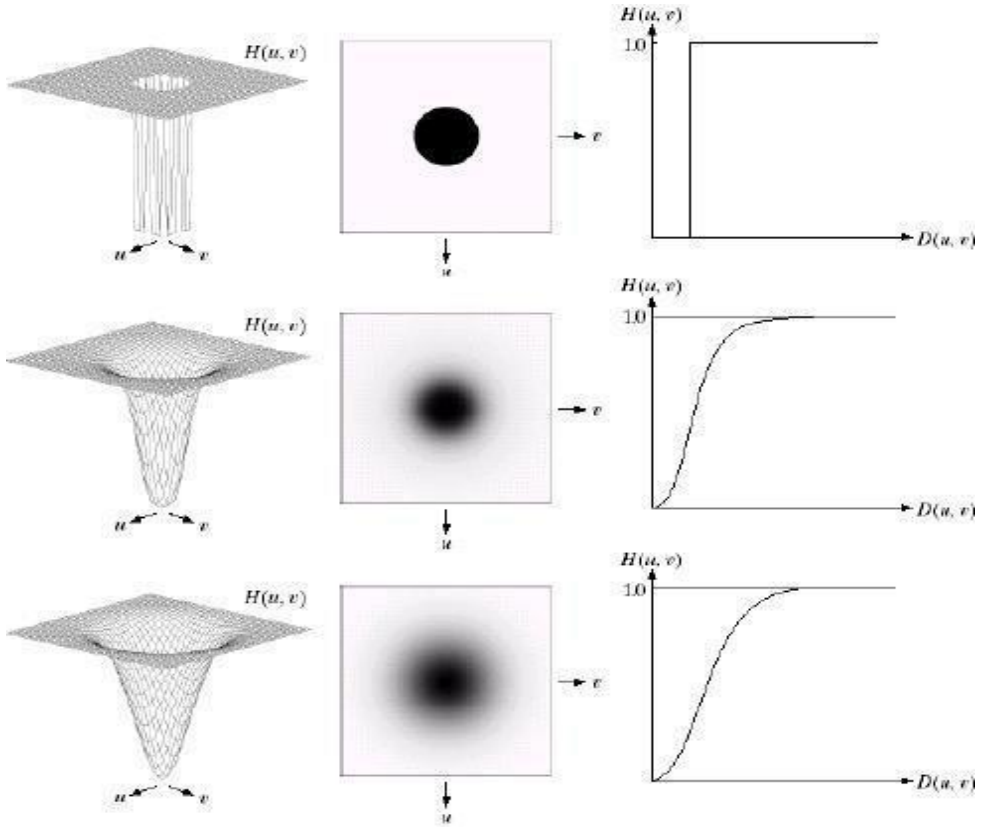


Image sharpening can be achieved by a high pass filtering process, which attenuates the low frequency components without disturbing high-frequency information. These are radially symmetric and completely specified by a cross section. If we have the transfer function of a low pass filter the corresponding high pass filter can be obtained using the equation

$$Hhp\ (u, v) = 1 - Hlp\ (u, v)$$

(i) Ideal High pass filter:

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

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

(ii) Butterworth High pass filter:

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

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

(iii) Guassian 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 no uniform illumination in an image. Pixel intensities in an image represent the light reflected from the corresponding points in the objects. As per as image model, image f(z,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).r(x, y) --------- (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 equation 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)) not equal to F[i(x, y)].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,

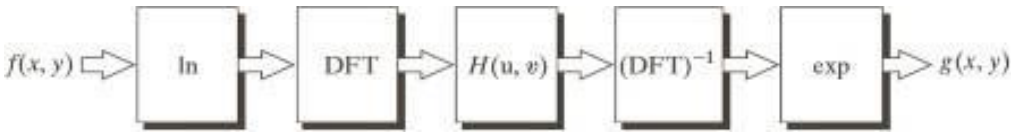
F[ln f(x,y)} = F[ln i(x, y)} + F[ln r(x, y)].

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

f' (x , y) = i' (x , y) + r'(x, y)



Color Image enhancement:

Color of an object is determined by the nature of the light reflected from it. When a beam of sunlight passes through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As shown in figure, the color spectrum may be divided into six broad regions: violet, blue, green,yellow, orange, and red. When viewed in full color no color in the spectrum ends abruptly, but rather each color blends smoothly into the next.

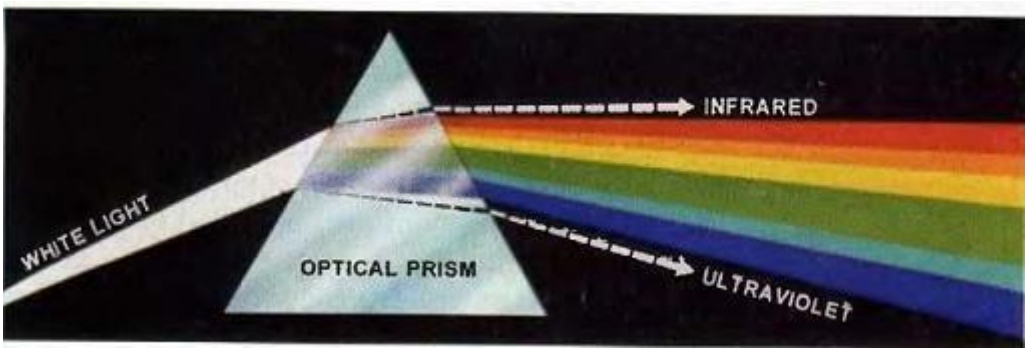


Fig: Color spectrum seen by passing white light through a prism.

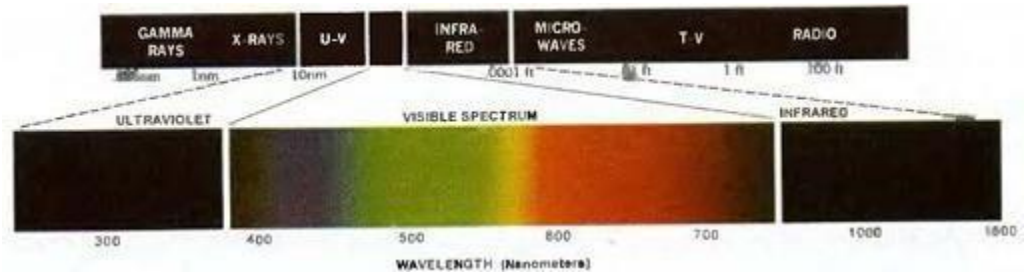


Fig: Wavelength comprising the visible range of electromagnetic spectrum

As illustrated in Figure, visible light is composed of a relatively narrow band of frequencies in the electromagnetic spectrum. A body that reflects light that is balanced in all visible wavelengths appears white to the observer. However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths. Characterization of light is central to the science of color. If the light is achromatic (void of color), its only attribute is its intensity, or amount. Achromatic light is what viewers see on a black and white television set. Three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, andbrightness.

Radiance:

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

Luminance:

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

Brightness:

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

Cones are the sensors in the eye responsible for color vision. Detailed experimental evidence has established that the 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue.

Approximately65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue (but the blue cones are the most sensitive).

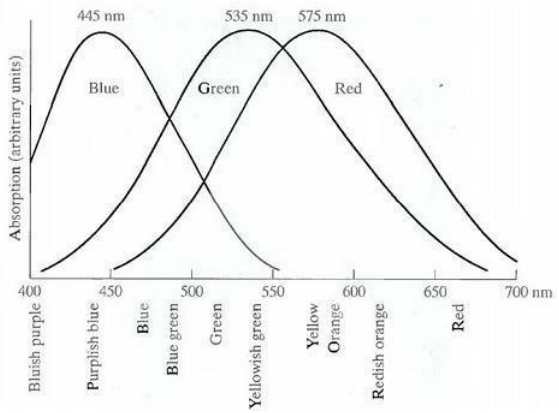


Fig: Absorption of light by RGB cones in human eye

Figure shows average experimental curves detailing the absorption of light by the red, green, and blue cones in the eye. Due to these absorption characteristics of the human eye, colors arc seen as variable combinations of the so- called primary colors red (R), green (G), and blue (B).The primary colors can be added to produce the secondary colors of light --magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries, or a secondary with its opposite primary color, in the right intensities produces white light. The characteristics generally used to distinguish one color from another are brightness, hue, and saturation. Brightness embodies the chromatic notion of intensity. Hue is an attribute associated with the dominant wavelength in a mixture of light waves. Hue represents dominant color as perceived by an observer. Saturation refers to the relative purity or the amount of white light mixed with a hue. The pure spectrum colors are fully saturated. Colors such as pink(red and white) and lavender (violet and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light-added. Hue and saturation taken together are called chromaticity, and. therefore, a color may be characterized by its brightness and chromaticity.

UNIT-IV

IMAGE RESTORATION & IMAGE SEGMENTATION

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, or

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

Degradation Model:

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

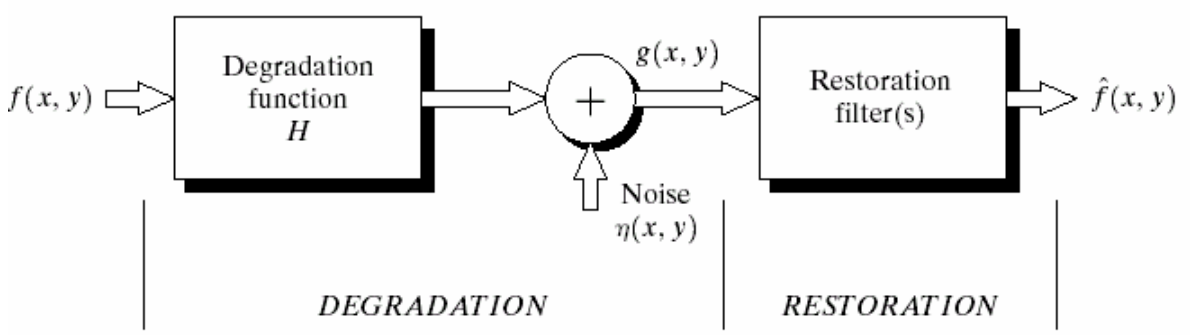
$$g(x,y)=f(x,y)*h(x,y)+\eta(x,y)$$

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 inversing the image degradation process, i.e.,

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

where 1/H(u,v)is the inverse filter, and G(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 and the unknown noise source. 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:

. The noise model is spatial invariant, i.e., independent of spatial location.

. The noise model is uncorrelated with the object function.

(i) Gaussian Noise:

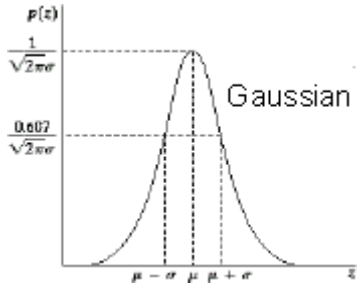
These noise models are used frequently in practices because of its tractability in both spatial and frequency domain. The PDF of Gaussian random variable, z is given by

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

z= gray level

μ= mean of average value of z

σ= standard deviation



(ii) Rayleigh Noise:

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

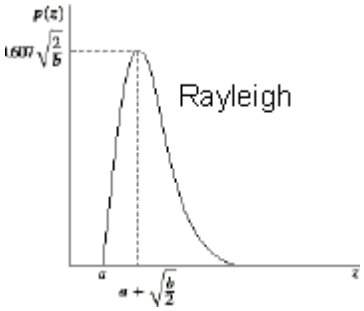
$$p(z) = \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}}, \quad \text{for } z \geq a$$

The mean and variance of this density is

Mean/variance

$$\mu = a + \sqrt{\pi b / 4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$



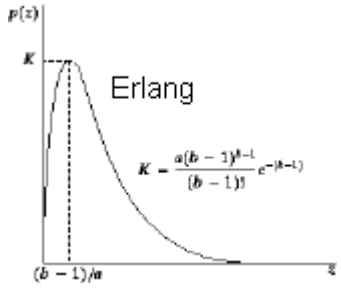
(iii) Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this noise is

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



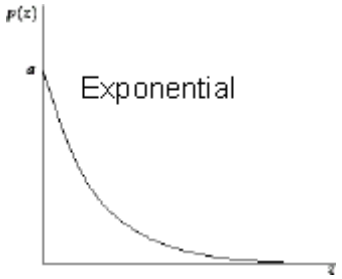
Its shape is similar to Rayleigh disruption. This equation is referred to as gamma density it is correct only when the denominator is the gamma function.

(iv) 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



(v) Uniform Noise:

The PDF of uniform noise is given by

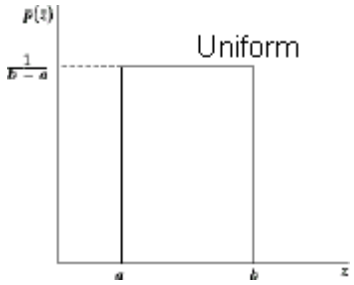
$$p(z) = \begin{cases} \frac{1}{(b-a)} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean and variance of this noise is

Mean/variance

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$



(vi) Impulse (salt & pepper) Noise:

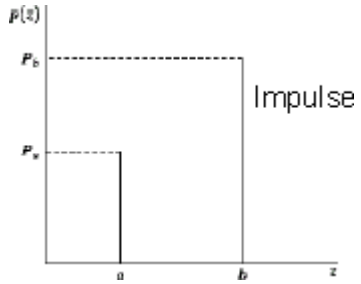
In this case, the noise is signal dependent, and is multiplied to the image.

The PDF of bipolar (impulse) noise is given by

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

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.

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

or

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

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

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

(i) Mean Filter:

(a) Arithmetic Mean filter:

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

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

This operation can be using a convolution mask in which all coefficients have value 1/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 .This will resulted in a smoothing effect in the image.

(b) Geometric Mean filter:

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

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

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

(c)Harmonic Mean filter:

The harmonic mean filtering operation is given by the expression

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

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

(d) 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.

(e) 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.

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

The original of the pixel is included in the computation of the median of the filter are quite possible because for certain types of random noise, the provide excellent noise reduction capabilities with considerably less blurring then smoothing filters of similar size. These are effective for bipolar and unipolor impulse noise.

(e) Max and Min filter:

Using the l00th 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 0th percentile filter is min filter

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

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

(f) Midpoint filter:

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

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

It comeliness 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-

(i) Band Reject Filters:

It removes a band of frequencies about the origin of the Fourier transformer.

(ii) Ideal Band reject Filter:

An ideal band reject filter is given by the expression

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

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

W- the width of the band

Do- the radial center of the frequency rectangle.

(iii) Butterworth Band reject Filter:

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

(iv) 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.

(v) Band pass Filter:

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

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

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

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 f of the uncorrupted image f such that the mean sequence error between then is minimized.

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

This error measure is given by

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

Where e( ) 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 ..... is given by the expression

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

Product of a complex quantity with its conjugate is equal to the magnitude of …… 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) H(u,v)

Sn(u,v)=IN(u,v)I2- power spectrum of the noise

Sf(u,v)=IF(u,v)2- power spectrum of the underrated image

H(u,v)=Fourier transformer of the degraded function

G(u,v)=Fourier transformer of the degraded image

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

Mean square error in statistical form can be approved 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 during the transform of the degraded image G(u,v) by the degradation function.

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

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

It shows an interesting result that even if we know the depredation function we cannot recover the underrated 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 SEGMENTATION:

If an image has been preprocessed appropriately to remove noise and artifacts, segmentation is often the key step in interpreting the image. Image segmentation is a process in which regions or features sharing similar characteristics are identified and grouped together. Image segmentation may use statistical classification, thresholding, edge detection, region detection, or any combination of these techniques. The output of the segmentation step is usually a set of classified elements, Most segmentation techniques are either region-based or edge based.

(i) Region-based techniques rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles.

(ii) Edge-based techniques rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions. Segmentation is a process of extracting and representing information from an image is to group pixels together into regions of similarity. Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consists of : (a)Intensity values from original images, or computed values based on an image operator (b)Textures or patterns that are unique to each type of region (c)Spectral profiles that provide multidimensional image data Elaborate systems may use a combination of these properties to segment images, while

simpler systems may be restricted to a minimal set on properties depending of the type of data available.

Categories of Image Segmentation Methods

· Clustering Methods Level Set Methods

· Histogram-Based Methods Graph portioning methods

· Edge Detection Methods Watershed Transformation

· Region Growing Methods Neural Networks Segmentation

· Model based Segmentation/knowledge-based segmentation - involve active shape and appearance models, active contours and deformable templates.

· Semi-automatic Segmentation - Techniques like Livewire or Intelligent Scissors are used in this kind of segmentation.

(i) Pixel based approach:

Gray level thresholding is the simplest segmentation process. Many objects or image regions are characterized by constant reflectivity or light absorption of their surface. Thresholding is computationally inexpensive and fast. Thresholding can easily be done in real time using specialized hardware. Complete segmentation can result from thresholding in simple scenes.

$$R = \bigcup_{i=1}^{S} R_i \qquad R_i \cap R_j = \emptyset \qquad i \neq j$$

Search all the pixels f(i,j) of the image f. An image element g(i,j) of the segmented image is an object pixel if f(i,j) >= T, and is a background pixel otherwise.

$$\begin{aligned} g(i,j) &= 1 \quad \text{for } f(i,j) \geq T \\ &= 0 \quad \text{for } f(i,j) < T \end{aligned}$$

Correct threshold selection is crucial for successful threshold segmentation. Threshold selection can be interactive or can be the result of some threshold detection method.

(a) Multi Level Thresholding:

The resulting image is no longer binary

$$\begin{aligned} g(i,j) &= 1 \quad \text{for } f(i,j) \in D_1 \\ &= 2 \quad \text{for } f(i,j) \in D_2 \\ &= 3 \quad \text{for } f(i,j) \in D_3 \\ &= 4 \quad \text{for } f(i,j) \in D_4 \\ &\cdots \\ &= n \quad \text{for } f(i,j) \in D_n \\ &= 0 \quad \text{otherwise} \end{aligned}$$

(b) Local Thresholding:

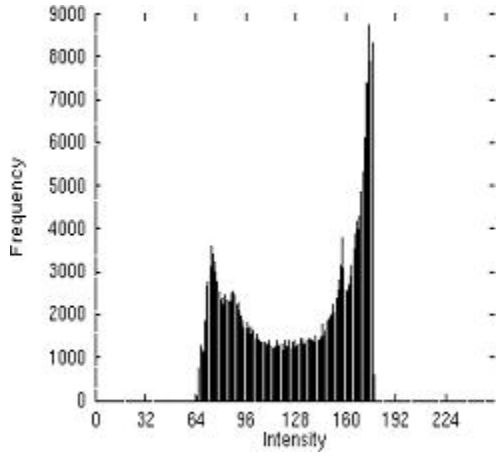
It is successful only under very unusual circumstances. Gray level variations are likely due to non-uniform lighting, non-uniform input device parameters or a number of other factors.

$$T=T(f)$$

(c) Thresholding detection method:

If some property of an image after segmentation is known a priori, the task of threshold selection is simplified, since the threshold is chosen to ensure this property is satisfied. A printed text sheet may be an example if we know that characters of the text cover 1/p of the sheet area.

- **P-type thresholding**
  - choose a threshold T (based on the image histogram) such that 1/p of the image area has gray values less than T and the rest has gray values larger than T.
  - in text segmentation, prior information about the ratio between the sheet area and character area can be used.
  - if such a priori information is not available - another property, for example the average width of lines in drawings, etc. can be used - the threshold can be determined to provide the required line width in the segmented image.
- **More methods of thresholding are**
  - based on histogram shape analysis
  - bimodal histogram - if objects have approximately the same gray level that differs from the gray level of the background



- **Mode Method:** find the highest local maxima first and detect the threshold as a minimum between them. To avoid detection of two local maxima belonging to the same global maximum, a minimum distance in gray levels between these maxima is usually required or techniques to smooth histograms are applied.

(ii) Region based segmentation:

(a) Region based growing segmentation:

Homogeneity of regions is used as the main segmentation criterion in region growing. The criteria for homogeneity:

•graylevel

•color

•texture

•shape

• model

The basic purpose of region growing is to segment an entire image R into smaller sub-images, Ri, i=1,2,….,N. which satisfy the following conditions:

$$R = \bigcup_{i=1}^{N} R_i; R_1 \bigcap R_j = \Phi, i \neq j$$
$$H(R_i) = True; i = 1,2,..., N;$$
$$H(R_i \bigcup R_j) = False, i \neq j;$$

(b) Region Splitting:

The basic idea of region splitting is to break the image into a set of disjoint regions, which are coherent within themselves:

• Initially take the image as a whole to be the area of interest.

• Look at the area of interest and decide if all pixels contained in the region satisfy some *similarity constraint.*

• If TRUE then the area of interest corresponds to an entire region in the image.

• If FALSE split the area of interest (usually into four equal subareas) and consider each of the sub-areas as the area of interest in turn.

• This process continues until no further splitting occurs. In the worst case this happens when the areas are just one pixel in size.

If only a splitting schedule is used then the final segmentation would probably contain many neighboring regions that have identical or similar properties. We need to merge these regions.

(c)Region merging:

The result of region merging usually depends on the order in which regions are merged. The simplest methods begin merging by starting the segmentation using regions of 2x2, 4x4 or 8x8 pixels. Region descriptions are then based on their statistical gray level properties. A region description is compared with the description of an adjacent region; if they match, they

are merged into a larger region and a new region description is computed. Otherwise regions are marked as non-matching. Merging of adjacent regions continues between all neighbors, including newly formed ones. If a region cannot be merged with any of its neighbors, it is marked `final' and the merging process stops when all image regions are so marked. Merging Heuristics:

• Two adjacent regions are merged if a significant part of their common boundary consists of weak edges

• Two adjacent regions are also merged if a significant part of their common boundary consists of weak edges, but in this case not considering the total length of the region borders.

Of the two given heuristics, the first is more general and the second cannot be used alone because it does not consider the influence of different region sizes.

Region merging process could start by considering

• small segments (2*2, 8*8) selected a priori from the image

• segments generated by thresholding

• regions generated by a region splitting module

The last case is called as “Split and Merge” method. Region merging methods generally use similar criteria of homogeneity as region splitting methods, and only differ in the direction of their application.

(d) Split & Merge:

To illustrate the basic principle of split and merge methods, let us consider an imaginary image.

• Let I denote the whole image shown in Fig. (a)

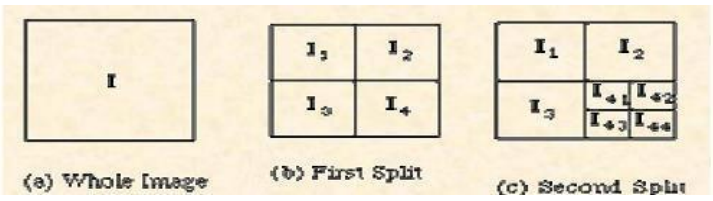
• Not all the pixels in Fig (a) are similar. So the region is split as in Fig. (b).

• Assume that all pixels within each of the regions I1, I2 and I3 are similar, but those in I4 are not.

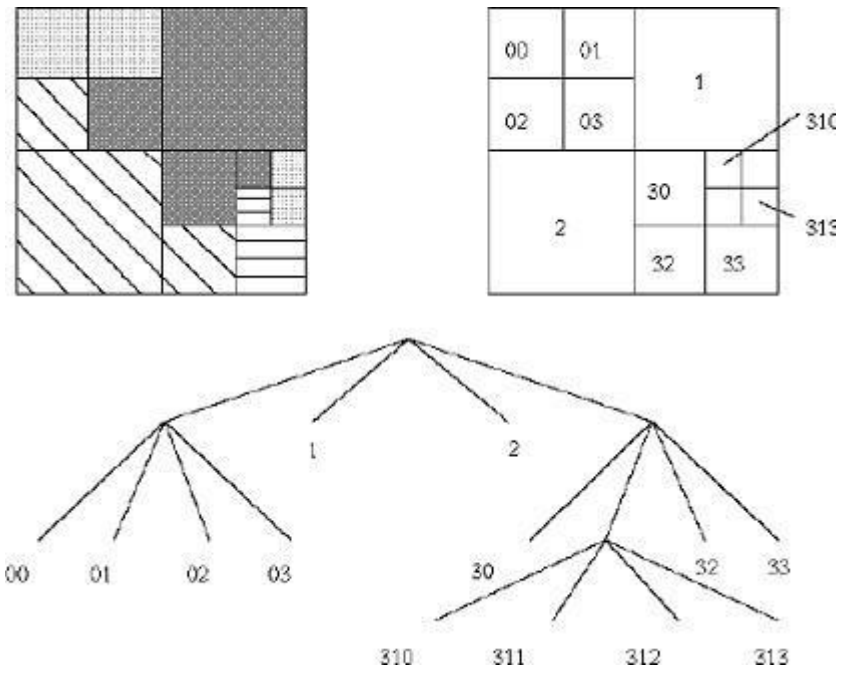
• Therefore I4 is split next, as shown in Fig. (c).

• Now assume that all pixels within each region are similar with respect to that region, and that after comparing the split regions, regions I43 and I44 are found to be identical.

• These pair of regions is thus merged together, as in shown in Fig. (d)



A combination of splitting and merging may result in a method with the advantages of both the approaches. Split-and-merge approaches work using pyramid image representations. Regions are square-shaped and correspond to elements of the appropriate pyramid level. If any region in any pyramid level is not homogeneous (excluding the lowest level), it is split into four sub-regions -- these are elements of higher resolution at the level below. If four regions exist at any pyramid level with approximately the same value of homogeneity measure, they are merged into a single region in an upper pyramid level. We can also describe the splitting of the image using a tree structure, called a modified *quad tree*. Each non-terminal node in the tree has at most four descendants, although it may have less due to merging. Quad tree decomposition is an operation that subdivides an image into blocks that contain "similar" pixels. Usually the blocks are square, although sometimes they may be rectangular. For the purpose of this demo, pixels in a block are said to be "similar" if the range of pixel values in the block are not greater than some threshold. Quad tree decomposition is used in variety of image analysis and compression applications. An unpleasant drawback of segmentation quad trees, is the square region shape assumption. It is not possible to merge regions which are not part of the same branch of the segmentation tree. Because both split-and-merge processing options are available, the starting segmentation does not have to satisfy any of the homogeneity conditions. The segmentation process can be understood as the construction of a segmentation quad tree where each leaf node represents a homogeneous region. Splitting and merging corresponds to removing or building parts of the segmentation quad tree.



(e) Region growing:

Region growing approach is the opposite of the split and merges approach:

• An initial set of small areas is iteratively merged according to similarity constraints.

• Start by choosing an arbitrary *seed pixel* and compare it with neighboring pixels

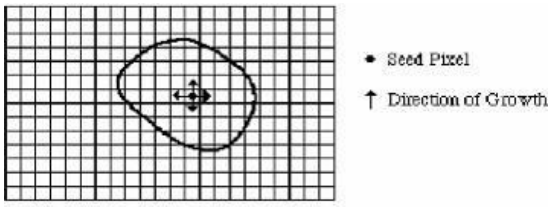
• Region is *grown* from the seed pixel by adding in neighboring pixels that are similar, increasing the size of the region.

• When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again.

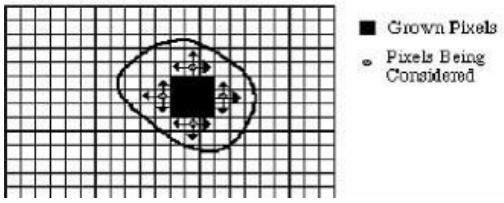
• This whole process is continued until all pixels belong to some region.

• A *bottom up* method.

Region growing methods often give very good segmentations that correspond well to the observed edges.



(a) Start of Growing a Region



However starting with a particular seed pixel and letting this region grow completely before trying other seeds biases the segmentation in favour of the regions which are segmented first. This can have several undesirable effects:

• Current region dominates the growth process -- ambiguities around edges of adjacent regions may not be resolved correctly.

• Different choices of seeds may give different segmentation results.

• Problems can occur if the (arbitrarily chosen) seed point lies on an edge.

To counter the above problems, *simultaneous region growing* techniques have been developed.

• Similarities of neighboring regions are taken into account in the growing process.

• No single region is allowed to completely dominate the proceedings.

• A number of regions are allowed to grow at the same time.

• Similar regions will gradually coalesce into expanding regions.

• Control of these methods may be quite complicated but efficient methods have been developed.

• Easy and efficient to implement on parallel computers.

(iii) Line & Edge detection:

(a) Edge detection:

Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image information.

- **Canny edge detection:**

It is optimal for step edges corrupted by white noise. Optimality related to three criteria

- detection criterion ... important edges should not be missed, there should be no spurious responses
- localization criterion ... distance between the actual and located position of the edge should be minimal
- one response criterion ... minimizes multiple responses to a single edge (also partly covered by the first criterion since when there are two responses to a single edge one of them should be considered as false)

Canny's edge detector is based on several ideas:

1) The edge detector was expressed for a 1D signal and the first two optimality criteria. A closed form solution was found using the calculus of variations.

2) If the third criterion (multiple responses) is added, the best solution may be found by numerical optimization. The resulting filter can be approximated effectively with error less than 20% by the first derivative of a Gaussian smoothing filter with standard deviation; the reason for doing this is the existence of an effective implementation.

3) The detector is then generalized to two dimensions. A step edge is given by its position, orientation, and possibly magnitude (strength).

- Suppose G is a 2D Gaussian and assume we wish to convolute the image with an operator Gn which is a first derivative of G in the direction n.

$$G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla G$$

The direction n should be oriented perpendicular to the edge

- this direction is not known in advance
- however, a robust estimate of it based on the smoothed gradient direction is available
- if g is the image, the normal to the edge is estimated as

$$\mathbf{n} = \frac{\nabla(G * g)}{|\nabla(G * g)|}$$

The edge location is then at the local maximum in the direction n of the operator G_n convoluted with the image g

$$\frac{\partial}{\partial \mathbf{n}} G_n * g = 0$$

- Substituting in equation for G_n from equation, we get

$$\frac{\partial^2}{\partial \mathbf{n}^2} G * g = 0$$

- This equation shows how to find local maxima in the direction perpendicular to the edge; this operation is often referred to as **non-maximum suppression.**
  - As the convolution and derivative are associative operations in equation
  - first convolute an image g with a symmetric Gaussian G
  - then compute the directional second derivative using an estimate of the direction n
  - strength of the edge (magnitude of the gradient of the image intensity function g) is measured as

$$|G_n * g| = |\nabla(G * g)|$$

4) Spurious responses to the single edge caused by noise usually create a so called 'streaking' problem that is very common in edge detection in general.

- Output of an edge detector is usually thresholded to decide which edges are significant.

- Streaking means breaking up of the edge contour caused by the operator fluctuating above and below the threshold.
- Streaking can be eliminated by thresholding with hysteresis
    - If any edge response is above a high threshold, those pixels constitute definite output of the edge detector for a particular scale.
    - Individual weak responses usually correspond to noise, but if these points are connected to any of the pixels with strong responses they are more likely to be actual edges in the image.
    - Such connected pixels are treated as edge pixels if their response is above a low threshold.
    - The low and high thresholds are set according to an estimated signal to noise ratio.

5) The correct scale for the operator depends on the objects contained in the image.

- The solution to this unknown is to use multiple scales and aggregate information from them.
- Different scale for the Canny detector is represented by different standard deviations of the Gaussians.
- There may be several scales of operators that give significant responses to edges (i.e., signal to noise ratio above the threshold); in this case the operator with the smallest scale is chosen as it gives the best localization of the edge.
- Feature synthesis approach.
    - All significant edges from the operator with the smallest scale are marked first.
    - Edges of a hypothetical operator with larger are synthesized from them (i.e., a prediction is made of how the large should perform on the evidence gleaned from the smaller).
    - Then the synthesized edge response is compared with the actual edge response for larger.
    - Additional edges are marked only if they have significantly stronger response than that predicted from synthetic output.
- This procedure may be repeated for a sequence of scales, a cumulative edge map is built by adding those edges that were not identified at smaller scales.

Algorithm: Canny edge detector:

1. Repeat steps (2) till (6) for ascending values of the standard deviation .
2. Convolve an image g with a Gaussian of scale .
3. Estimate local edge normal directions n for each pixel in the image.
4. Find the location of the edges (non-maximal suppression).
5. Compute the magnitude of the edge
6. Threshold edges in the image with hysteresis to eliminate spurious responses.
7. Aggregate the final information about edges at multiple scale using the `feature synthesisapproach.

(b) Edge Operators:

Since edges consist of mainly high frequencies, we can, in theory, detect edges by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. In practice, edged detection is performed in the spatial domain, because it is computationally less expensive and often yields better results. `Since edges correspond to strong illumination gradients, we can highlight them by calculating the derivatives of the image. We can see that the position of the edge can be estimated with the maximum of the 1st derivative or with the zero-crossing of the 2nd derivative. Therefore we want to find a technique to calculate the derivative of a two-dimensional image. For a discrete one dimensional function *f(i)*, the first derivative can be approximated by

$$\frac{d\,f(i)}{d(i)} = f(i+1) - f(i)$$

Calculating this formula is equivalent to convolving the function with [-1 1]. Similarly the 2nd derivative can be estimated by convolving *f(i)* with [1 -2 1]. Different edge detection kernels which are based on the above formula enable us to calculate either the 1st or the 2nd derivative of a two-dimensional image. There are two common approaches to estimate the 1st derivative in a two-dimensional image, Prewitt compass edge detection and *gradient edge detection*.Prewitt compass edge detection involves convolving the image with a set of (usually 8) kernels, each of which is sensitive to a different edge orientation. The kernel producing the maximum response at a pixel location determines the edge magnitude and orientation. Different sets of kernels might be used: examples include Prewitt, Sobel, Kirsch and Robinson kernels. Gradient edge detection is the second and more widely used technique. Here, the image is

convolved with only two kernels, one estimating the gradient in the *x*-direction, *Gx*, the other the gradient in the *y*-direction, *Gy*. The absolute gradient magnitude is then given by

$$|G| = \sqrt{Gx^2 + Gy^2}$$

and is often approximated with

$$|G| = |Gx| + |Gy|$$

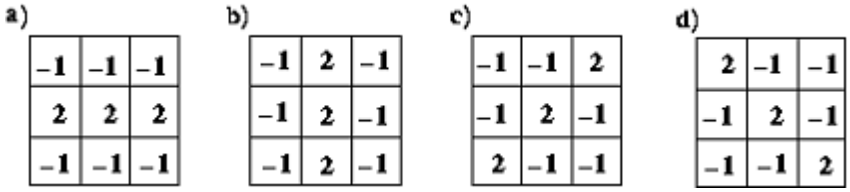
In many implementations, the gradient magnitude is the only output of a gradient edge detector, however the edge orientation might be calculated with The most common kernels used for the gradient edge detector are the Sobel, Roberts Cross and Prewitt operators. After having calculated the magnitude of the 1st derivative, we now have to identify those pixels corresponding to an edge. The easiest way is to threshold the gradient image, assuming that all pixels having a local gradient above the threshold must representanedge.Analternative technique is to look for local maxima in the gradient image, thus producing one pixel wide edges. A more sophisticated technique is used by the Canny edge detector. It first applies a gradient edge detector to the image and then finds the edge pixels using *non-maximal suppression* and *hysteresis tracking*.

An operator based on the 2nd derivative of an image is the Marr edge detector, also known as *zero crossing detector*. Here, the 2nd derivative is calculated using a Laplacian of aussian (LoG) filter. The Laplacian has the advantage that it is an isotropic measure of the 2nd derivative of an image, *i.e.* the edge magnitude is obtained independently from the edge orientation by convolving the image with only one kernel. The edge positions are then given by the zero-crossings in the LoG image. The scale of the edges which are to be detected can be controlled by changing the variance of the Gaussian. A general problem for edge detection is its sensitivity to noise, the reason being that calculating the derivative in the spatial domain corresponds to accentuating high frequencies and hence magnifying noise. This problem is addressed in the Canny and Marr operators by convolving the image with a smoothing operator (Gaussian) before calculating the derivative.

(c) Line detection:

While edges (I. E. boundaries between regions with relatively distinct graylevels) are by far the most common type of discontinuity in an image, instances of thin lines in an image occur frequently enough that it is useful to have a separate mechanism for detecting them. A convolution based technique can be used which produces an image description of the thin

lines in an input image. Note that the Hough transform can be used to detect lines; however, in that case, the output is a P ARA ME TRIC description of the lines in an image. The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width *n*, at a particular orientation ɵ. Figure below shows a collection of four such kernels, which each respond to lines of single pixel width at the particular orientation shown.



Four line detection kernels which respond maximally to horizontal, vertical and oblique (+45 and - 45 degree) single pixel wide lines. These masks above are tuned for light lines against a dark background, and would give a big negative response to dark lines against a light background. If we are only interested in detecting dark lines against a light background, then we should negate the mask values. Alternatively, we might be interested in either kind of line, in which case, we could take the absolute value of the convolution output.

If R_i denotes the response of kernel I, we can apply each of these kernels across an image, and for any particular point, if for all that point is more likely to contain a line whose orientation (and width) corresponds to that of kernel I. One usually thresholds to eliminate weak lines corresponding to edges and other features with intensity gradients which have a different scale than the desired line width. In order to find complete lines, one must join together line fragments, ex: with an edge tracking operator.

(d) Corner detection:

Input to the corner detector is the gray-level image. Output is an image in which values are proportional to the likelihood that the pixel is a corner. The Moravec detector is maximal in pixels with high contrast. These points are on corners and sharp edges.

$$MO(i,j) = \frac{1}{8} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} |g(k,l) - g(i,j)|$$

Using ZH Operator,

The image function f is approximated in the neighborhood of the pixel (i,j) by a cubic polynomial with coefficients c_k. This is a cubic facet model. The ZH operator estimates the corner strength based on the coefficients of the cubic facet model.

$$g(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$

$$ZH(i,j) = \frac{-2\left(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4\right)}{\left(c_2^2 + c_3^2\right)^{\frac{3}{2}}}$$

UNIT-V

IMAGE COMPRESSION

Introduction:

In recent years, there have been significant advancements in algorithms and architectures for the processing of image, video, and audio signals. These advancements have proceeded along several directions. On the algorithmic front, new techniques have led to the development of robust methods to reduce the size of the image, video, or audio data. Such methods are extremely vital in many applications that manipulate and store digital data. Informally, we refer to the process of size reduction as a compression process. We will define this process in a more formal way later. On the architecture front, it is now feasible to put sophisticated compression processes on a relatively low-cost single chip; this has spurred a great deal of activity in developing multimedia systems for the large consumer market.

One of the exciting prospects of such advancements is that multimedia information comprising image, video, and audio has the potential to become just another data type. This usually implies that multimedia information will be digitally encoded so that it can be manipulated, stored, and transmitted along with other digital data types. For such data usage to be pervasive, it is essential that the data encoding is standard across different platforms and applications. This will foster widespread development of applications and will also promote interoperability among systems from different vendors. Furthermore, standardization can lead to the development of cost-effective implementations, which in turn will promote the widespread use of multimedia information. This is the primary motivation behind the emergence of image and video compression standards.

Background:

Compression is a process intended to yield a compact digital representation of a signal. In the literature, the terms *source coding*, *data compression*, *bandwidth compression*, and *signal compression* are all used to refer to the process of compression. In the cases where the signal is defined as an image, a video stream, or an audio signal, the generic problem of compression is to minimise the bit rate of their digital representation. There are many applications that benefit when image, video, and audio signals are available in compressed form. Without compression, most of these applications would not be feasible!

Example 1: Let us consider **facsimile image transmission**. In most facsimile machines, the document is scanned and digitised. Typically, an 8.5x11 inches page is scanned at 200 dpi;

thus, resulting in 3.74 Mbits. Transmitting this data over a low-cost 14.4 kbits/s modem would require 5.62 minutes. With compression, the transmission time can be reduced to 17 seconds. This results in substantial savings in transmission costs.

Example 2: Let us consider a video-based CD-ROM application. **Full-motion video**, at 30 fps and a 720 x 480 resolution, generates data at 20.736 Mbytes/s. At this rate, only 31 seconds of video can be stored on a 650 MByte CD-ROM. Compression technology can increase the storage capacity to 74 minutes, for VHS-grade video quality.

Image, video, and audio signals are amenable to compression due to the factors below.

• There is considerable statistical redundancy in the signal.

1. Within a single image or a single video frame, there exists significant correlation among neighbor samples. This correlation is referred to as *spatial correlation.*

2. For data acquired from multiple sensors (such as satellite images), there exists significant correlation amongst samples from these sensors. This correlation is referred to as *spectral correlation.*

3. For temporal data (such as video), there is significant correlation amongst samples in different segments of time. This is referred to as *temporal correlation.*

• There is considerable information in the signal that is irrelevant from a perceptual point of view.

• Some data tends to have high-level features that are redundant across space and time; that is, the data is of a fractal nature.

Application	Data Rate	
	Uncompressed	Compressed
Voice 8 ksamples/s, 8 bits/sample	64 kbps	2-4 kbps
Slow motion video (10fps) framesize 176x120, 8bits/pixel	5.07 Mbps	8-16 kbps
Audio conference 8 ksamples/s, 8 bits/sample	64 kbps	16-64 kbps
Video conference (15fps) framesize 352x240, 8bits/pixel	30.41 Mbps	64-768 kbps
Digital audio 44.1 ksamples/s, 16 bits/sample	1.5 Mbps	1.28-1.5 Mbps
Video file transfer (15fps) framesize 352x240, 8bits/pixel	30.41 Mbps	384 kbps
Digital video on CD-ROM (30fps) framesize 352x240, 8bits/pixel	60.83 Mbps	1.5-4 Mbps
Broadcast video (30fps) framesize 720x480, 8bits/pixel	248.83 Mbps	3-8 Mbps
HDTV (59.94 fps) framesize 1280x720, 8bits/pixel	1.33 Gbps	20 Mbps

For a given application, compression schemes may exploit any one or all of the above factors to achieve the desired compression data rate.

There are many applications that benefit from data compression technology. Table shown above lists a representative set of such applications for image, video, and audio data, as well as typical data rates of the corresponding compressed bit streams. Typical data rates for the uncompressed bit streams are also shown.

In the following figure, a systems view of Image compression process is depicted.

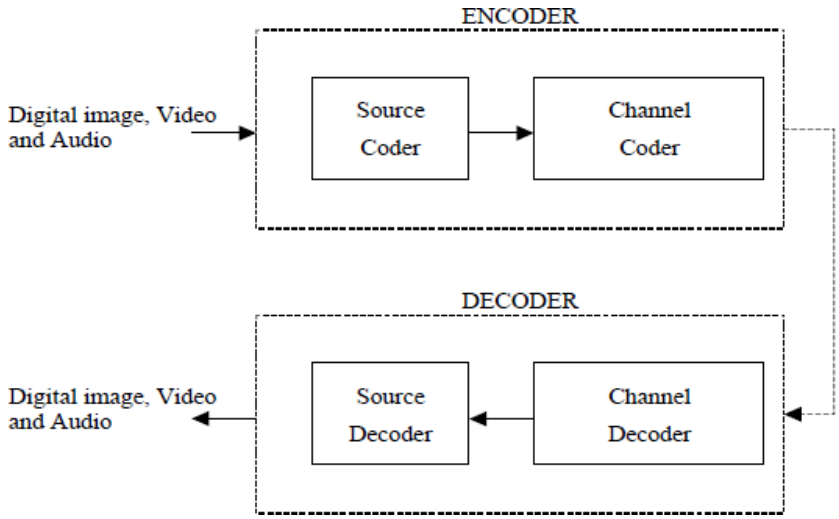


Fig: Generic compression system

The core of the encoder is the source coder. The source coder performs the compression process by reducing the input data rate to a level that can be supported by the storage or transmission medium. The bit rate output of the encoder is measured in bits per sample or bits per second. For image or video data, a pixel is the basic element; thus, bits per sample is also referred to as bits per pixel or bits per pel. In the literature, the term *compression ratio*, denoted as c_r , is also used instead of *bit rate* to characterise the capability of the compression system. An intuitive definition of c_r is

c_r =(Source coder input size) / (Source coder output size)

This definition is somewhat ambiguous and depends on the data type and the specific compression method that is employed. For a still-image, size could refer to the bits needed to represent the entire image. For video, size could refer to the bits needed to represent one frame of video. Many compression methods for video do not process each frame of video, hence, a more commonly used notion for size is the bits needed to represent one second of

video. In a practical system, the source coder is usually followed by a second level of coding: the channel coder. The channel coder translates the compressed bit stream into a signal suitable for either storage or transmission. In most systems, source coding and channel coding are distinct processes. In recent years, methods to perform combined source and channel coding have also been developed. Note that, in order to reconstruct the image, video, or audio signal, one needs to reverse the processes of channel coding and source coding. This is usually performed at the decoder.

From a system design viewpoint, one can restate the compression problem as a bit rate minimization problem, where several constraints may have to be met, including the following:

- **Specified level of signal quality**. This constraint is usually applied at the decoder.
- **Implementation complexity**. This constraint is often applied at the decoder, and in some instances at both the encoder and the decoder.
- **Communication delay**. This constraint refers to the end to end delay, and is measured from the start of encoding a sample to the complete decoding of that sample.

Note that, these constraints have different importance in different applications. For example, in a two-way teleconferencing system, the communication delay might be the major constraint, whereas, in a television broadcasting system, signal quality and decoder complexity might be the main constraints.

Types of compression:

There are two types og mage compression namely

- Lossless compression
- Lossy compression

(i) Lossless compression:

In many applications, the decoder has to reconstruct without any loss the original data. For a lossless compression process, the reconstructed data and the original data must be identical in value for each 5 and every data sample. This is also referred to as a reversible process. In lossless compression, for a specific application, the choice of a compression method involves a trade-off along the three dimensions depicted in Figure that is, coding efficiency, coding complexity, and coding delay.

Coding Efficiency:

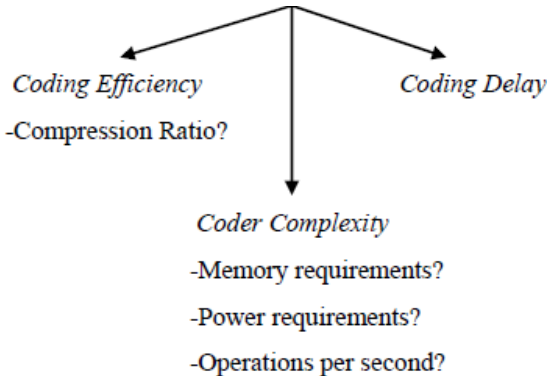
This is usually measured in bits per sample or bits per second (bps). Coding efficiency is usually limited by the information content or *entropy* of the source. In intuitive terms, the entropy of a source X provides a measure for the "randomness" of X. From a compression theory point of view, sources with large entropy are more difficult to compress (for example, random noise is very hard to compress).

Coding Complexity :

The complexity of a compression process is analogous to the computational effort needed to implement the encoder and decoder functions. The computational effort is usually measured in terms of memory requirements and number of arithmetic operations. The operations count is characterized by the term millions of operations per second and is often referred to as MOPS. Here, by operation, we imply a basic arithmetic operation that is supported by the computational engine. In the compression literature, the term MIPS (millions of instructions per second) is sometimes used. This is specific to a computational engine's architecture; thus, in this text we refer to coding complexity in terms of MOPS. In some applications, such as portable devices, coding complexity may be characterized by the power requirements of a hardware implementation.

Coding Delay:

A complex compression process often leads to increased coding delays at the encoder and the decoder. Coding delays can be alleviated by increasing the processing power of the computational engine; however, this may be impractical in environments where there is a power constraint or when the underlying computational engine cannot be improved. Furthermore, in many applications, coding delays have to be constrained; for example, in interactive communications. The need to constrain the coding delay often forces the compression system designer to use a less sophisticated algorithm for the compression processes.



From this discussion, it can be concluded that these trade-offs in coding complexity, delay, and efficiency are usually limited to a small set of choices along these axes. In a subsequent section, we will briefly describe the trade-offs within the context of specific lossless compression methods.

(ii) Lossy compression:

The majority of the applications in image or video data processing do not require that the reconstructed data and the original data are identical in value. Thus, some amount of loss is permitted in the reconstructed data. A compression process that results in an imperfect reconstruction is referred to as a lossy compression process. This compression process is irreversible. In practice, most irreversible compression processes degrade rapidly the signal quality when they are repeatedly applied on previously decompressed data.

The choice of a specific lossy compression method involves trade-offs along the four dimensions shown in Figure. Due to the additional degree of freedom, namely, in the signal quality, a lossy compression process can yield higher compression ratios than a lossless compression scheme.

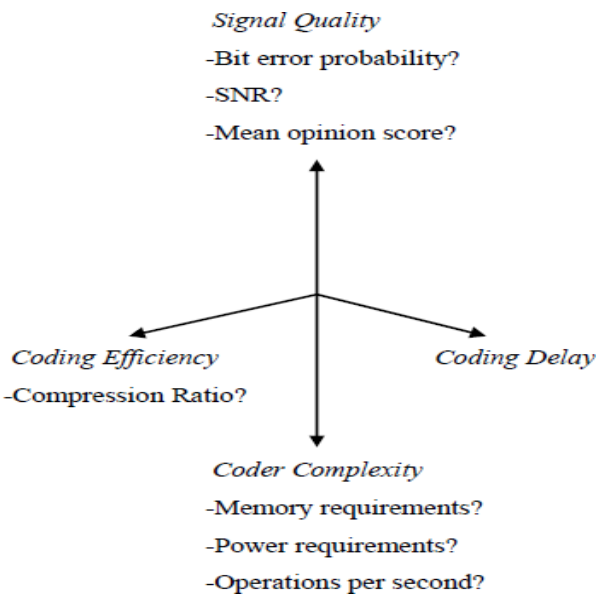
Signal Quality: This term is often used to characterize the signal at the output of the decoder. There is no universally accepted measure for signal quality.

One measure that is often cited is the signal to noise ratio, which can be expressed as SNR.

The noise signal energy is defined as the energy measured for a hypothetical signal that is the difference between the encoder input signal and the decoder output signal. Note that, *SNR* as defined here is given in decibels (dB). In the case of images or video, *PSNR* (peak signal-to-noise ratio) is used instead of *SNR* . The calculations are essentially the same as in the case of *SNR* , however, in the numerator, instead of using the encoder input signal one uses a hypothetical signal with a signal strength of 255 (the maximum decimal value of an unsigned 8-bit number, such as in a pixel). High *SNR* or *PSNR* values do not always correspond to signals with perceptually high quality. Another measure of signal quality is the mean opinion score, where the performance of a compression process is characterized by the subjective quality of the decoded signal. For instance, a five point scale such as *very annoying, annoying, slightly annoying, perceptible but not annoying,* and *imperceptible* might be used to characterize the impairments in the decoder output.

In either lossless or lossy compression schemes, the quality of the input data affects the compression ratio. For instance, acquisition noise, data sampling timing errors, and even the analogue-to-digital conversion process affects the signal quality and reduces the spatial and

temporal correlation. Some compression schemes are quite sensitive to the loss in correlation and may yield significantly worse compression in the presence of noise.



Issues in compression method selection:

In this chapter, we have introduced some fundamental concepts related to image, video, and audio compression. When choosing a specific compression method, one should consider the following issues:

- Lossless or lossy. This is usually dictated by the coding efficiency requirements. Coding efficiency. Even in a lossy compression process, the desirable coding efficiency might not be achievable. This is especially the case when there are specific constraints on output signal quality.
- Variability in coding efficiency. In some applications, large variations in coding efficiency among different data sets may not be acceptable.
- Resilience to transmission errors. Some compression methods are more robust to transmission errors than others. If retransmissions are not permitted, then this requirement may impact on the overall encoder- decoder design.
- Complexity trade-offs. In most implementations, it is important to keep the overall encoder-decoder complexity low. However, certain applications may require only a low decoding complexity.
- Nature of degradations in decoder output. Lossy compression methods introduce artifacts in the decoded signal. The nature of artifacts depends on the compression method that is employed. The degree to which these artifacts are judged also varies

from application to application. In communication systems, there is often an interplay between the transmission errors and the coding artifacts introduced by the coder. Thus, it is important to consider all types of error in a system design.

- Data representation. In many applications, there is a need to support two decoding phases. In the first phase, decoding is performed to derive an intelligible signal; this is the case in data browsing. In the second phase, decoding is performed to derive a higher quality signal. One can generalise this notion to suggest that some applications require a hierarchical representation of the data. In the compression context, we refer to such compression schemes as *scalable compression methods.* The notion of scalability has been adopted in the compression standards.
- Multiple usage of the encoding-decoding tandem. In many applications, such as video editing, there is a need to perform multiple encode-decode operations using results from a previous encode-decode operation. This is not an issue for lossless compression; however, for lossy schemes, resilience to multiple encoding-decoding cycles is essential.
- Interplay with other data modalities, such as audio and video. In a system where several data modalities have to be supported, the compression methods for each modality should have some common elements. For instance, in an interactive videophone system, the audio compression method should have a frame structure that is consistent with the video frame structure. Otherwise, there will be unnecessary requirements on buffers at the decoder and a reduced tolerance to timing errors.
- Interworking with other systems. In a mass-market environment, there will be multiple data modalities and multiple compression systems. In such an environment, transcoding from one compression method to another may be needed. For instance, video editing might be done on a frame by frame basis; hence, a compression method that does not exploit temporal redundancies might be used here. After video editing, there might be a need to broadcast this video. In this case, temporal redundancies can be exploited to achieve a higher coding efficiency. In such a scenario, it is important to select compression methods that support transcoding from one compressed stream format to another. Interworking is important in many communications environments as well.

The Source coder:

In this course we are interested in exploring various compression techniques referring to the **source coder** only, where the image compression takes place. A general procedure for image data compression is shown in the following block diagram.

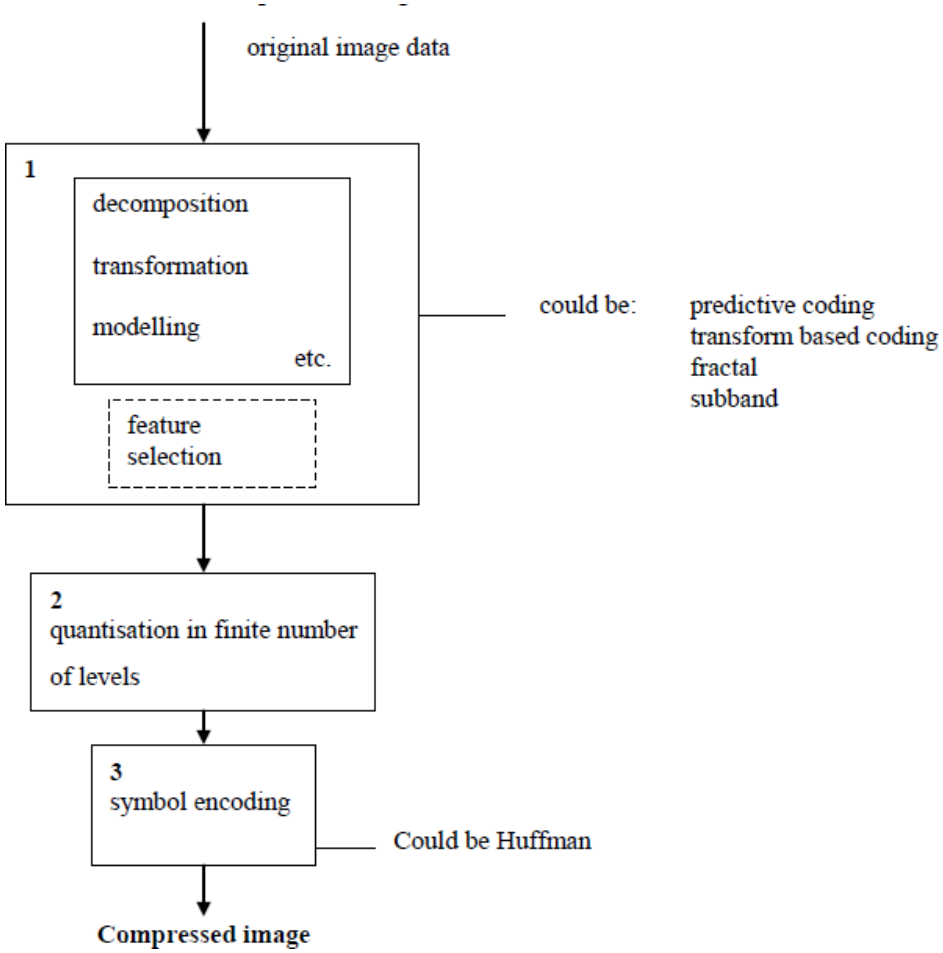


Image Formats and compression standards:

(a)Lossless compression:

Lossless compression refers to compression methods for which the original uncompressed data set can be recovered exactly from the compressed stream. The need for lossless compression arises from the fact that many applications, such as the compression of digitized medical data, require that no loss be introduced from the compression method. Bitonal image transmission via a facsimile device also imposes such requirements. In recent years, several compression standards have been developed for the lossless compression of such images. We discuss these standards later. In general, even when lossy compression is allowed, the overall compression scheme may be a combination of a lossy compression process followed by a lossless compression process. Various image, video, and audio compression standards follow this model, and several of the lossless compression schemes used in these standards are

described in this section. The general model of a lossless compression scheme is as depicted in the following figure.

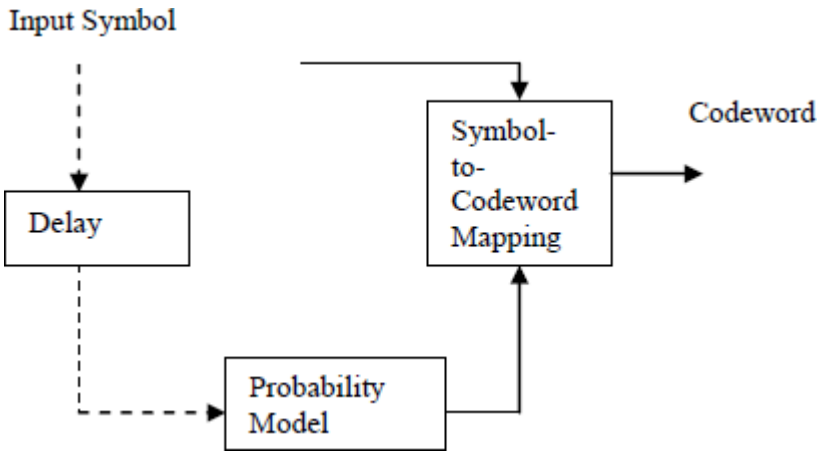


Fig: A generic model of lossless compression

Given an input set of symbols, a modeler generates an estimate of the probability distribution of the input symbols. This probability model is then used to map symbols into code words. The combination of the probability modeling and the symbol-to-codeword mapping functions is usually referred to as **entropy coding**. The key idea of entropy coding is to use short code words for symbols that occur with high probability and long code words for symbols that occur with low probability.

The probability model can be derived either from the input data or from a priori assumptions about the data. Note that, for decodability, the same model must also be generated by the decoder. Thus, if the model is dynamically estimated from the input data, causality constraints require a delay function between the input and the modeler. If the model is derived from a priori assumptions, then the delay block is not required; furthermore, the model function need not have access to the input symbols. The probability model does not have to be very accurate, but the more accurate it is, the better the compression will be. Note that, compression is not always guaranteed. If the probability model is wildly inaccurate, then the output size may even expand. However, even then the original input can be recovered without any loss.

Decompression is performed by reversing the flow of operations shown in the above Figure. This decompression process is usually referred to as **entropy decoding**.

(i)Differential coding:

Another preprocessing technique that improves the compression ratio is differential coding.

Differential coding skews the symbol statistics so that the resulting distribution is more amenable to compression. Image data tend to have strong inter-pixel correlation. If, say, the pixels in the image are in the order $x_1, x_2, \ldots x_N$, then instead of compressing these pixels, one might process the sequence of differentials $y_i = x_i - x_{i-1}$, where $i = 0,1,2 \ldots N-1$, and $x_0 = 0$. In compression terminology, $i\ y$ is referred to as the **prediction residual** of $i\ x$. The notion of compressing the prediction residual instead of $i\ x$ is used in all the image and video compression standards. For images, a typical probability distribution for $i\ x$ and the resulting distribution for $i\ y$ are shown in Figure.

Let symbol S_i have a probability of occurrence P_i. From coding theory, the ideal symbol-to codeword mapping function will produce a codeword requiring $\log (1/2P_i)$ bits. A distribution close to uniform for P_i, such as the one shown in the left plot of Figure, will result in codewords that on the average require eight bits; thus, no compression is achieved. On the other hand, for a skewed probability distribution, such as the one shown in the right plot of Figure, the **symbol-to-codeword mapping function** can on the average yield codewords requiring less than eight bits per symbol and thereby achieve compression.

We will understand these concepts better in the following Huffman encoding section.

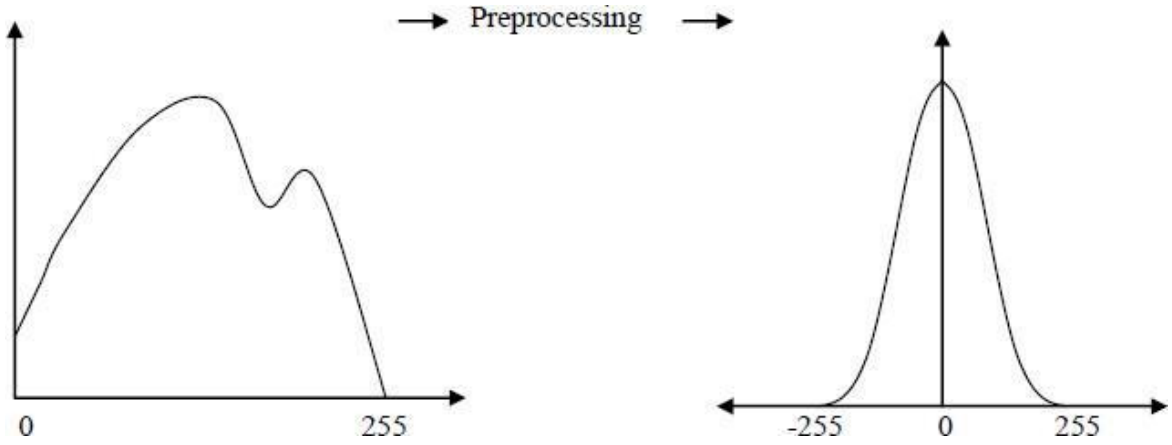


Fig: Typical distribution of pixel values for x_i and y_i.

(ii)Huffman coding:

In 1952, D. A. Huffman developed a code construction method that can be used to perform lossless compression. In Huffman coding, the modeling and the symbol-to-codeword mapping functions of Figure 1.1 are combined into a single process. As discussed earlier, **the input data are partitioned into a sequence of symbols** so as to facilitate the modeling process. In most image and video compression applications, the size of the alphabet

composing these symbols is restricted to at most 64000 symbols. The Huffman code construction procedure evolves along the following parts:

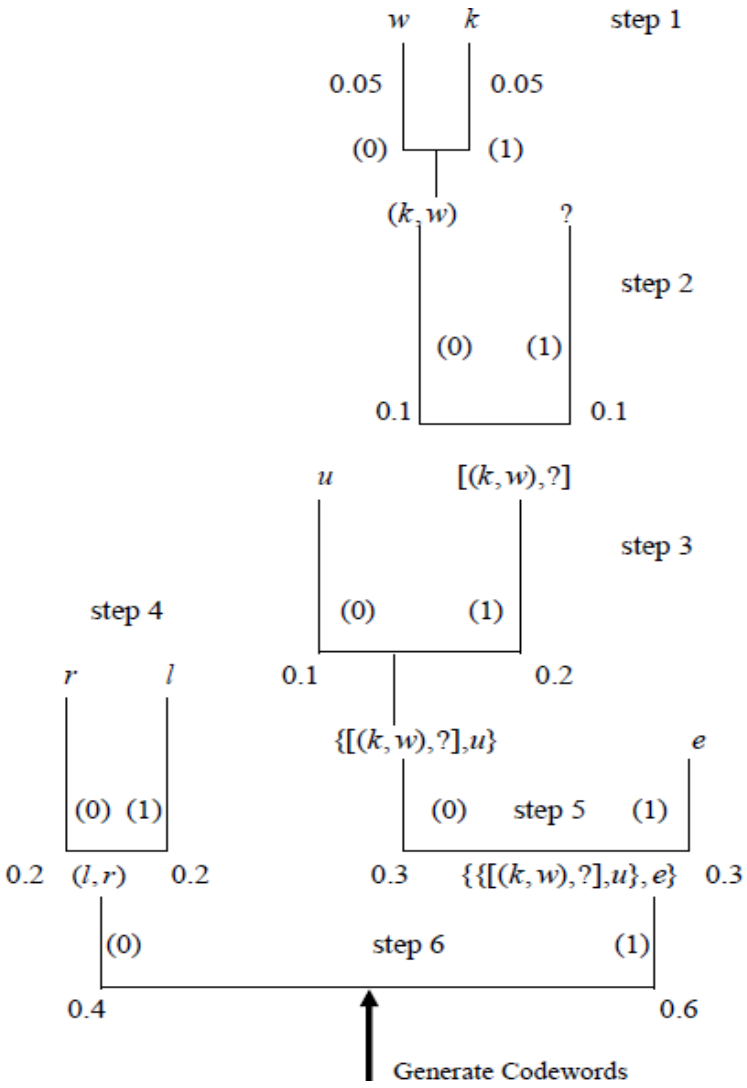
1. Order the symbols according to their probabilities. For Huffman code construction, the frequency of occurrence of each symbol must be known a priori. In practice, the frequency of occurrence can be estimated from a training set of data that is representative of the data to be compressed in a lossless manner. If, say, the alphabet is composed of *N* distinct symbols $s_1,s_2,s_3\ldots.s_{N-1}$ and the probabilities of occurrence are $p_1,p_2\ldots p_{N-1}$, then the symbols are rearranged so that.

2. Apply a contraction process to the two symbols with the smallest probabilities. Suppose the two symbols are $s_1..s_N$. We replace these two symbols by a hypothetical symbol, say, $H_{N-1}=(s_{N-1},s_N)$that has a probability of occurrence *N N p* □ *p* □1 . Thus, the new set of symbols has N-1 members $s_1,s_2\ldots s_{N-1}$

3. We repeat the previous part 2 until the final set has only one member.

The recursive procedure in part 2 can be viewed as the construction of a binary tree, since at each step we are merging two symbols. At the end of the recursion process all the symbols *N s1,s2..sN-1* will be leaf nodes of this tree. The codeword for each symbol S_i is obtained by traversing the binary tree from its root to the leaf node corresponding to S_i

We illustrate the code construction process with the following example depicted in Figure. The input data to be compressed is composed of symbols in the alphabet *k, l,u,w,e, r,*? . First we sort the probabilities. In Step 1, we merge the two symbols *k* and *w* to form the new symbol (*k,w*) . The probability of occurrence for the new symbol is the sum of the probabilities of occurrence for *k* and *w.* We sort the probabilities again and perform the merge on the pair of least frequently occurring symbols which are now the symbols (*k,w*) and ? . We repeat this process through Step 6. By visualizing this process as a binary tree as shown in this figure and traversing the process from the bottom of the tree to the top, one can determine the code words for each symbol. For example, to reach the symbol *u* from the root of the tree, one traverses nodes that were assigned the bits 1,0 and 0 . Thus, the codeword for *u* is 100.



	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
k 0.05	*e* 0.3	*e* 0.3	*e* 0.3	*e* 0.3	(*l*,*r*) 0.4	{{[(*k*,*w*),?],*u*},*e*} 0.6
l 0.2	*l* 0.2	*l* 0.2	*l* 0.2	{[(*k*,*w*),?],*u*} 0.3	*e* 0.3	(*l*,*r*) 0.4
u 0.1	*r* 0.2	*r* 0.2	*r* 0.2	*l* 0.2	{[(*k*,*w*),?],*u*} 0.3	
w 0.05	*u* 0.1	*u* 0.1	[(*k*,*w*),?] 0.2	*r* 0.2		
e 0.3	? 0.1	? 0.1	*u* 0.1			

r 0.2	*k* 0.05	(*k*,*w*) 0.1				
? 0.1	*w* 0.05					

Symbol	Probability	Codeword
k	0.05	10101
l	0.2	01
u	0.1	100
w	0.05	10100
e	0.3	11
r	0.2	00
?	0.1	1011

Fig: An example for Huffman coding

In this example, the average codeword length is 2.6 bits per symbol. In general, the average codeword length is defined as

$$l_{avg} = \sum l_i p_i$$

where l_i is the codeword length (in bits) for the codeword corresponding to symbol s_i . The average codeword length is a measure of the compression ratio. Since our alphabet has seven symbols, a fixed-length coder would require at least three bits per codeword. In this example, we have reduced the representation from three bits per symbol to 2.6 bits per symbol; thus, the corresponding compression ratio can be stated as 3/2.6=1.15. For the lossless compression of typical image or video data, compression ratios in excess of two are hard to come by.

Standards of Lossless compression:

Standards related to the coding and transmission of signals over public telecommunication channels are developed under the auspices of the telecommunication standardization sector of the International Telecommunication Union (ITU-T). This sector was formerly known as the CCITT. The first standards for lossless compression were developed for facsimile applications. Scanned images used in such applications are bitonal, that is, the pixels take on one of two values, black or white, and these values are represented with one bit per pixel.

(iii)Run-length coding:

In every bitonal image there are large regions that are either all white or all black. For instance,in Figure, we show a few pixels of a line in a bitonal image. Note that, the six contiguous pixels of the same color can be described as a run of six pixels with value 0 . Thus, if each pixel of the image is remapped from say, its (position, value) to a **run** and

value, then a more compact description can be obtained. In our example, no more than four bits are needed to describe the six-pixel run. In general, for many document type images, significant compression can be achieved using such preprocessing. Such a mapping scheme is referred to as a run-length coding scheme.

The combination of a run-length coding scheme followed by a Huffman coder forms the basis of the image coding standards for facsimile applications. These standards include the following:

- ITU-T Rec. T.4 (also known as Group 3). There are two coding approaches within this standard.

1. Modified Huffman (MH) code. The image is treated as a sequence of scanlines, and a runlength description is first obtained for each line. A Huffman code is then applied to the (run, value) description. A separate Huffman code is used to distinguish between black and white runs, since the characteristics of these runs are quite different. The Huffman code table is static: that is, it does not change from image to image. For error-detection purposes, after each line is coded, an EOL (end of line) codeword is inserted.

   2. Modified Read (MR) code. Here, pixel values in a previous line are used as predictors for the current line. This is followed by a run-length description and a static Huffman code as in the MH code. An EOL codeword is also used. To prevent error propagation, MR coding is mixed with MH coding periodically.

- ITU-T Rec. T.6 (also known as Group 4). The coding technique used here is referred to as a Modified Modified Read (MMR) code. This code is a simplification of the MR code, wherein the error-protection mechanisms in the MR code are removed so as to improve the overall compression ratio.

These compression standards yield good compression (20:1 to 50:1) for business-type scanned documents. For images composed of natural scenes and rendered as bitonal images using a halftoning technique the compression ratio is severely degraded. In Figure shown, the image on the left possesses characteristics representative of business document images whereas the image on the right is a typical halftone image. The former is characterized by long runs of black or white, and the static Huffman code in the facsimile compression

standards is matched to these run-lengths. In the latter image, the run-lengths are relatively short, spanning only one to two pixels and the static Huffman code is not matched to such runs. An adaptive arithmetic coder is better suited for such images.

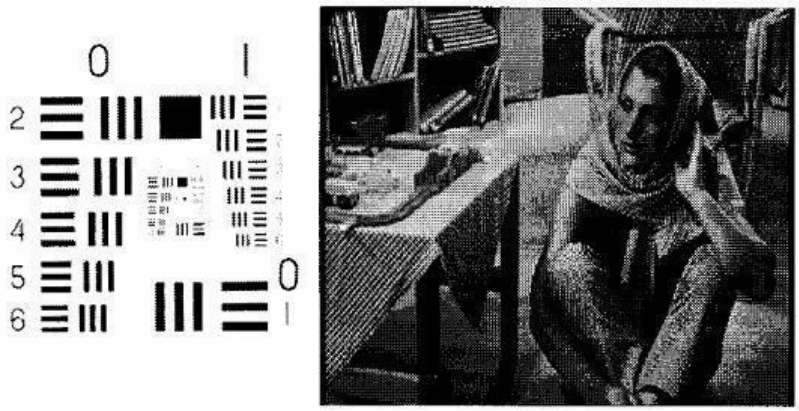


Fig: Typical Bitional Image

The JBIG Standard:

Recently, a compression standard was developed to efficiently compress halftone as well as business document type images. This is the JBIG (Joint Binary Image Experts Group) compression standard. Its standard nomenclature is ISO/IEC IS 11544, ITU-T Rec. T.82. The JBIG compression standard consists of a modeler and an arithmetic coder. The modeler is used to estimate the symbol probabilities that are then used by the arithmetic coder. **The JBIG is out of the scope of this course.** In Tables shown below, we provide a simple complexity analysis between the JBIG coding scheme and the ITU-T Rec. T.6 facsimile compressionstandard.

	JBIG-Baselayer	**ITU-T Rec. T.6**
Complexity Parameters	Three-line template, AT-max=16	2-D runlength, Huffman code
Memory	1589 bytes	1024 bytes
Buffer	Three scanlines	Two scanlines
Operations	Add, shift	Add, shift, compare

Compression		
Halftone Image	5.2:1	1.5:1
Letter Image	48:1	33.3:1

Table: Comparative analysis between JBIG and the ITU-T Rec. T.6 facsimile compression standards

We also provide compression ratios for two typical images: a 202 Kbyte halftone image and a 1 Mbyte image, primarily comprised of text. The latter image is referred to as letter in the table. For business-type documents, JBIG yields 20 percent to 50 percent more compression than the facsimile compression standards ITU-T Rec. T.4 and Rec.

T.6. For halftone images, compression ratios with JBIG are two to five times more than those obtained with the facsimile compression standards. However, software implementations of JBIG compression on a general purpose computer are two to three times slower than implementations of the ITU-T Rec. T.4 and T.6 standards. The JBIG standard can also handle grayscale images by processing each bit-plane of a grayscale image as separate bitonal images.

The Lossless JPEG Standard:

Most people know JPEG as a transform-based lossy compression standard. JPEG (Joint Photographic Experts Group), like JBIG, has been developed jointly by both the ITU-T and the ISO. We will describe this standard in greater detail in a subsequent section; however, here, we describe briefly the lossless mode of compression supported within this standard. The lossless compression method within JPEG is fully independent from transform-based coding. Instead, it uses differential coding to form prediction residuals that are then coded with either a Huffman coder or an arithmetic coder. As explained earlier, the prediction residuals usually have a lower entropy; thus, they are more amenable to compression than the original image pixels. In lossless JPEG, one forms a prediction residual using previously encoded pixels in the current line and/or the previous line. The prediction residual for pixel in Figure is r = y-x.

where can be any of the following functions:

y =0
y = a
y = b
y = c
y = a + b − c
y = a (+b − c) / 2
y = b (+a − c) / 2
y (=a + b) / 2

Note that, pixel values at pixel positions, and, are available to both the encoder and the *X y* decoder prior to processing. The particular choice for the function is defined in the scan header of the compressed stream so that both the encoder and the decoder use identical functions. Divisions by two are computed by performing a one-bit right shift.

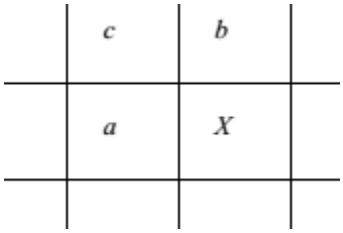


Fig: Lossless JPEG Prediction kernel

The prediction residual is computed modulo 2. This residual is not directly Huffman coded. Instead, it is expressed as a pair of symbols: the category and the magnitude. The first symbol represents the number of bits needed to encode the magnitude. Only this value is Huffman coded. The magnitude categories for all possible values of the prediction residual are shown in Table. If, say, the *X* prediction residual for is 42, then from Table 4.2 we determine that this value belongs to category 6; that is, we need an additional six bits to uniquely determine the value 42. The prediction residual is then mapped into the two-tuple (6, 6-bit code for 42). Category 6 is Huffman coded, and the compressed representation for the prediction residual consists of this Huffman codeword followed by the 6-bit representation for the magnitude. In general, if the value of the residual is positive, then the code for the magnitude is its direct binary representation. If the residual is negative, then the code for the magnitude is the one's complement of its absolute value. Therefore, code words for negative residual always start wish a zero bit.

Fundamental Coding Theorem:

At various points in their study of the image coding literature readers will come across with the R(D) application of rate-distortion theory, or its equivalent, distortion-rate theory. The significant introduction by Shannon (1959) of a fidelity criterion into his work on coding theory led to an extensive body of work which sought to characterize the relationship between **coding rate** and **measured distortion** in a consistent way. *R*(*D*) Briefly, the general form of the curve is as shown in the following figure where, as expected, for smaller levels of distortion we require a higher coding rate. If the attainable level of distortion is no smaller than *D*max then no information need be sent anyway. For an initially analogue input signal, as the distortion falls to zero the quantisation intervals must have a width which tends to zero and so the rate curve moves towards infinity (dotted curve in Figure). For a discrete signal we know that we can encode at a rate equivalent to the entropy and incur [*R*=(0) *H*] zero distortion, at least in principle . We may therefore set our operating point anywhere

≤0 $D \leq D$max within the region . Deviations between the results achieved with practical algorithms and the theory are to be expected, and are usually due to lack of knowledge of the true source $R(D)$ distribution (and the associated relation) and an inability to allocate fractional numbers of bits to encoded symbols (some kinds of block schemes, vector quantisation, for example, allow this to be done).

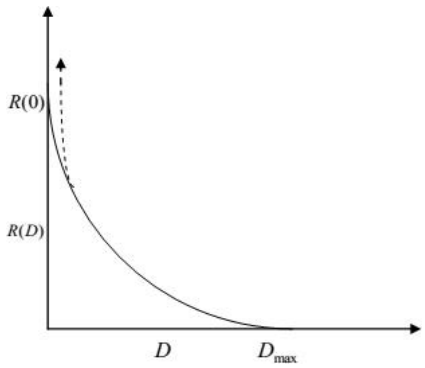


Fig: Rate-distortion relationship. For a discrete signal zero distortion coding is *R*=(0) achieved when the source entropy. For a continuous source zero distortion implies that the rate rises without limit (---).