Unit - 1: Introduction

Digital Image Processing

An image may be defined as a two-dimensional function, $f(x,y)$, where x and y are spatial (plane) coordinates, and the amplitude of 'f' at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, & discrete quantities we call the image a digital image.

Digital image processing refers to processing digital images by means of digital computer.

Digital image is composed of a finite numbers of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels or pixels.

We can consider three types of computerized processes in image processing : low-level, mid-level and high-level processes.

- Low-level processes :- It involves primitive operations such as image preprocessing to reduce noise, contrast enhancement, & image sharpening. It's both input and output are images.
- Mid-level processes :- It involves tasks such as segmentation, description of those objects to reduce them to a form suitable for computer processing, & classification of individual objects. It's output are attributes extracted from input images.
- High-level processes :- It involves "making sense" of an ensemble of recognized objects, as in image analysis.

Scanned with CamScanner

Representing Digital Images

There are two different ways to represent digital image.

- Co-ordinate convention (or spatial domain)
- Matrix form

Let us 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. The value of coordinates at the origin are $(x,y) = (0,0)$. The next coordinate values at first row of image are $(0,1)$.

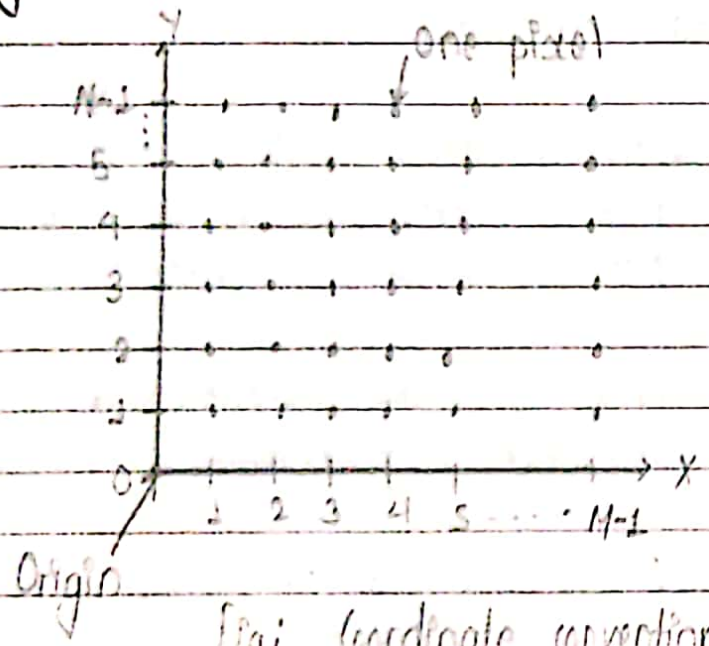


Fig: Coordinate convention

The representation of M×N digital image in matrix form is shown below:-

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

Scanned with CamScanner

Fundamental Steps in Digital Image Processing

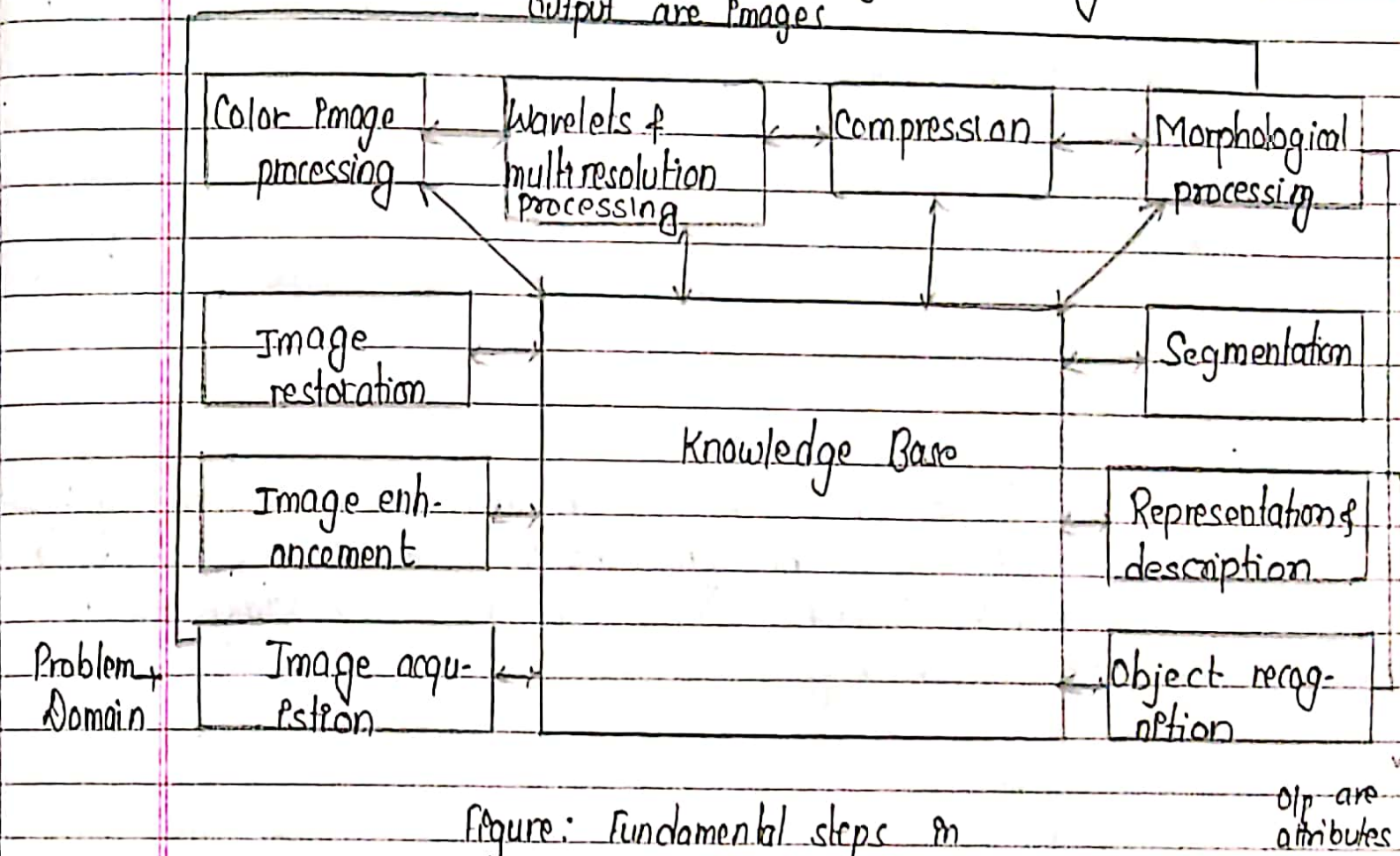


Figure: Fundamental steps in Digital Image Processing.

1. Image acquisition

Image acquisition is the first process acquisition which can be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.

2. Image enhancement

Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. for example: we increase the contrast of an image to make it "look better".

Scanned with CamScanner

3. Image restoration

It is an area that also deals with improving the appearance of an image. Unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. Enhancement is based on human subjective preferences regarding what constitutes a "good" enhancement result.

4. Color image processing

Color image processing is an area that has been gaining importance because of the significant increase in the use of digital images over the Internet. Color is used for extracting features of interest in an image.

5. Wavelets and multiresolution processing

Wavelets are the foundation for representing images in various degrees of resolution. It is used for image data compression & for pyramidal representation.

6. Compressing

It is the technique for reducing the storage required to save an image, or the bandwith required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. Image compression is familiar to most users of computers in the form of image file extensions, such as 'jpg' used in 'JPEG' image compression standard.

Scanned with CamScanner

7. Morphological processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

8. Segmentation

Segmentation procedures partition an image into its constituent parts or objects. Generally, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. In general, the more accurate the segmentation, the more likely recognition is to succeed.

9. Representation and description

It almost always follows the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Boundary representation is appropriate when focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when focus is on internal properties, such as texture or skeletal shape. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

10. Recognition

~~Reg~~ Recognition is the process that assigns a label to an object based on its descriptors. eg:- vehicle, box, board, etc.

Scanned with CamScanner

Applications of Digital Image Processing

1) Image sharpening and restoration
2) Medical Field
3) Robot vision
4) Pattern recognition
5) Video processing
6) Color processing, etc.

Elements of Digital Image Processing systems.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, etc.

First of all, to acquire digital image, we need a physical device that is sensitive to the energy radiated by the object we wish to image. Secondly, a digitizer, is a device for converting the output of the physical sensing device into digital form. Thus, we use digitizer to convert the outputs into digital data.

Specialized image processing hardware usually consists of the digitizer and other hardware such as ALU. This unit performs functions that require fast data throughputs (e.g. digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

The computer in an image processing system is a general purpose computer & can range from a PC to a supercomputer. Almost any well-equipped PC-type machine is suitable for offline image processing tasks.

Scanned with CamScanner

Software for image processing consists of specialized modules that perform specific tasks. A well designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules.

Mass storage capability is a must in image processing application. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. Digital storage for image processing application falls into three principal categories :-

(1) short term storage for use during processing,
(2) on-line storage for relatively fast recall,
(3) archival storage.

Image displays in use today are mainly color TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Sometimes images are implemented in form of headgear containing two small displays embedded in goggles worn by the user to have stereo displays.

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, etc. such as optical and CD ROM disks. Film provides the highest possible resolution, but paper is the obvious medium for written materials.

Networking is almost a default function in any computer system because of the large amount of data inherent in

Scanned with CamScanner

image processing applications, the key consideration in image transmission is bandwidth.

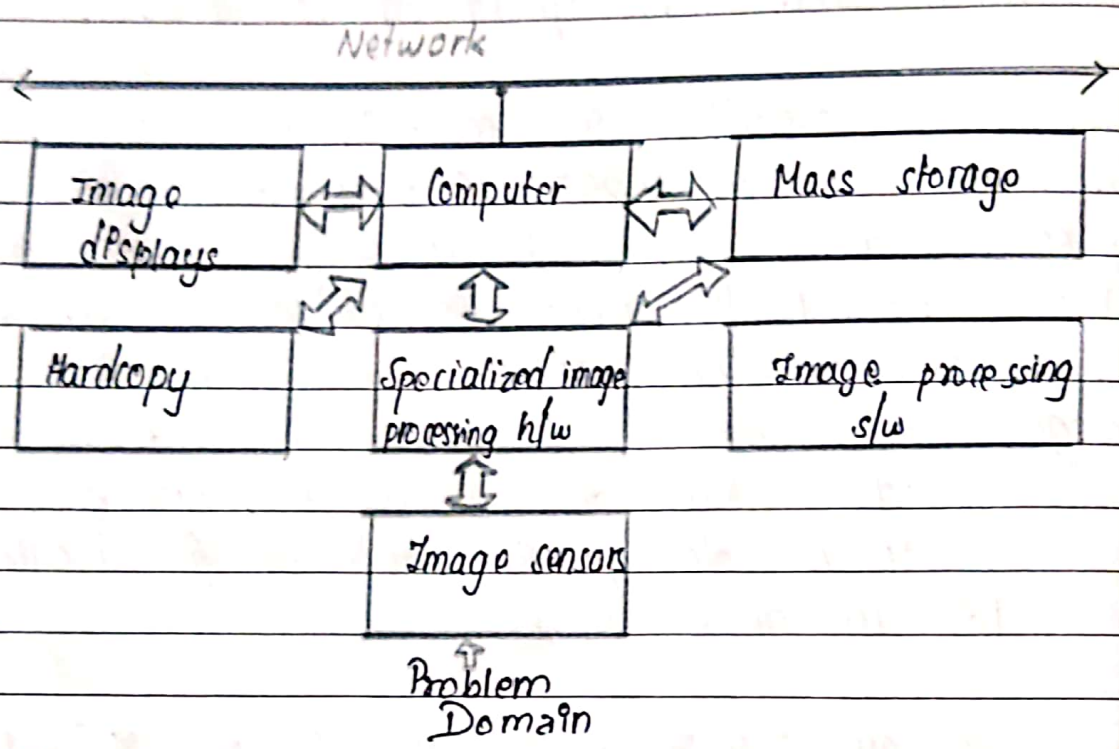


Fig: Elements of Digital Image Processing System.

Elements of Visual Perception

(1) Structure of Human Eye.

→ An eye is nearly spherical shaped, with an average diameter of 20mm (approx.).

→ Three membranes that encloses an eye are :-

1) the cornea and sclera outer cover,
2) the choroid,
3) the retina.

→ The cornea is a tough, transparent tissue that covers

Scanned with CamScanner

the interior surface of the eye.

The sclera is an opaque membrane that encloses the remainder of the optic globe.

→ The choroid lies directly below the sclera, it contains a network of blood vessels that serve as the major source of nutrition to the eye.

→ The choroid is divided into the ciliary body and the iris diaphragm. The later contracts or expands to control the amount of light that enters the eye.

→ The central opening of iris varies in diameter from approximately 2 to 8 mm. The front of iris contains the visible pigment of eye, whereas the back contains a black pigment.

→ The lens is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It contains 60 to 70% water, about 6% fat and more protein than any other tissue in the eye.

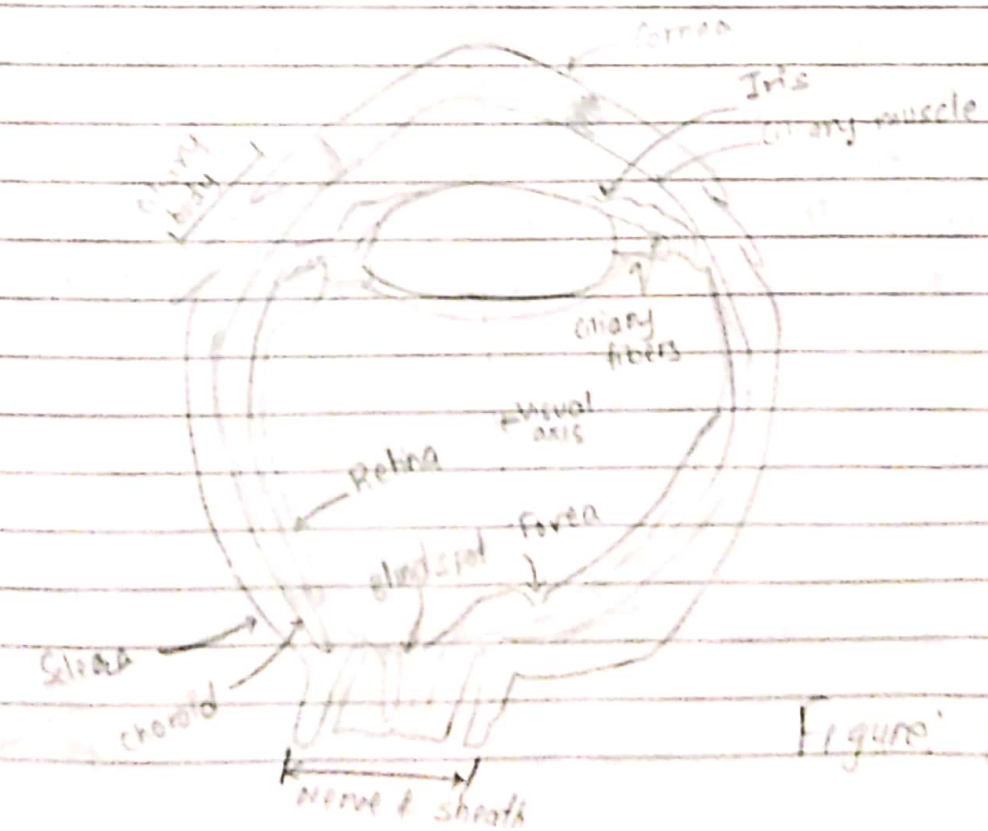


Figure: Human eye

→ The innermost membrane of the eye is retina.

→ When the eye is properly focused, light from an ~~out~~ object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptor over the surface of retina.

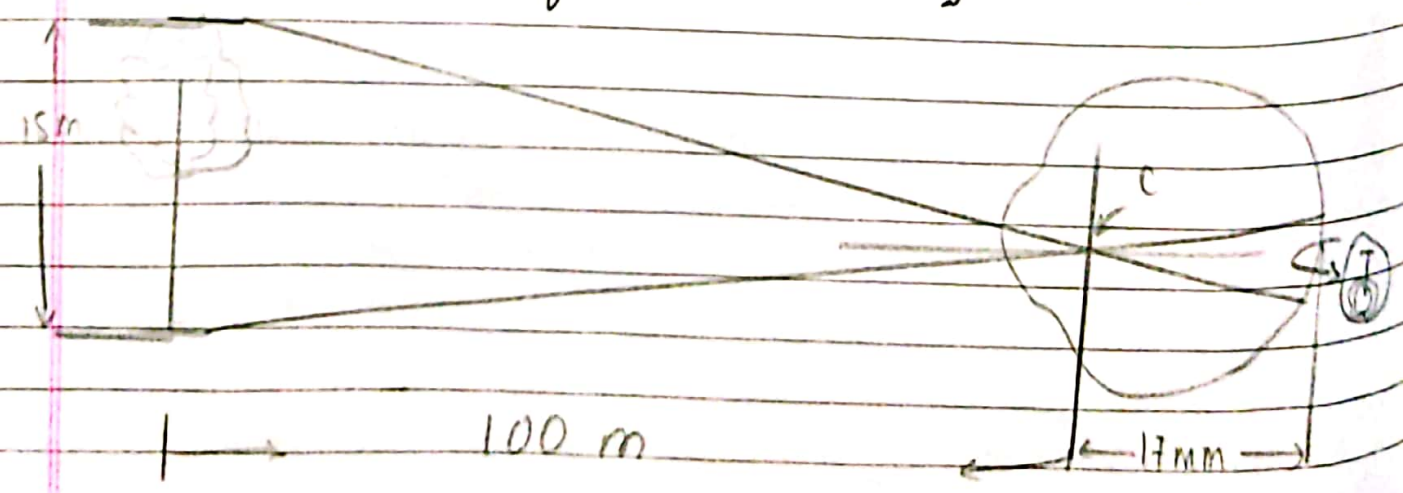
→ There are two classes of receptors: cones and rods, which are located primarily in the central portion of retina, called fovea.

→ Cone vision is called photopic or bright-light vision.

→ Rods yields the overall picture of field of view, they are not involved in color vision and are sensitive to low levels of illumination. Also called dim-light vision.

(2) Image Formation in the Eye

The radius of curvature of the anterior surface of the lens is greater than the radius of its posterior surface. The shape of the lens is controlled by tension in the fibers of the ciliary body. To focus on distant objects, the controlling muscles cause the lens to be relatively flattened. The distance between the center of lens and the retina varies from approximately 17 mm to 14 mm.



(5) Brightness Adaption and Discrimination

Since digital images are displayed as a discrete set of intensities, the eye's ability to discriminate between different intensity levels is an important consideration in presenting image-processing results. The range of light intensity levels to which the human visual system can adapt is enormous - on the order of 10^{10} - from the scotopic threshold to the glare limit.

The subjective brightness (intensity as perciered by human visual system) is a logarithmic function of the light incident on the eye.

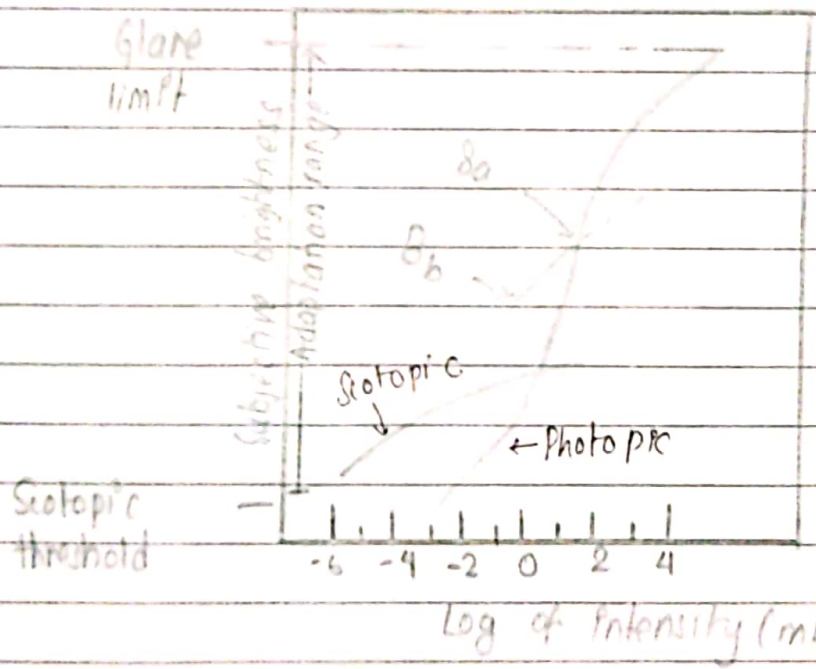


Fig: Range of subjective brightness sensations showing a particular adaptation level

The visual system cannot operate over dynamic range simultaneously. Rather, it accomplishes this large variation by changing in its overall sensitivity, a phenomenon known as brightness adaption.

Sampling and Quantization

There are numerous ways to acquire images, but our objective is to generate digital images from sensed data. The output of most sensors is a continuous voltage waveform whose amplitude and spatial behaviour are related to the physical phenomenon being sensed.

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

i) Sampling
ii) Quantization

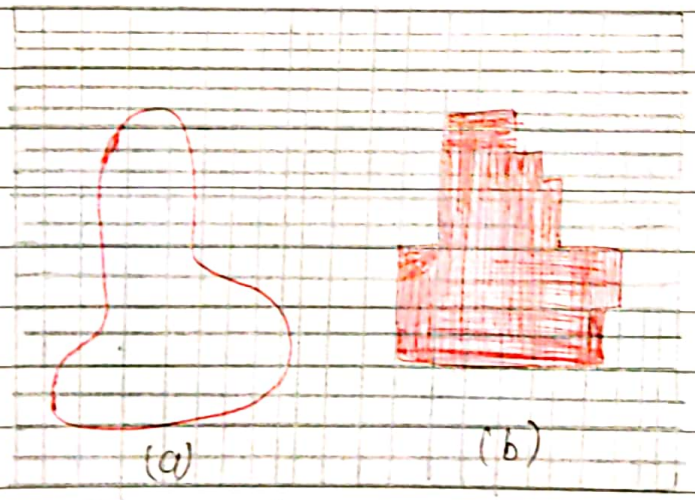


Fig: (a) Continuous Image (b) Result of sampling & quantization

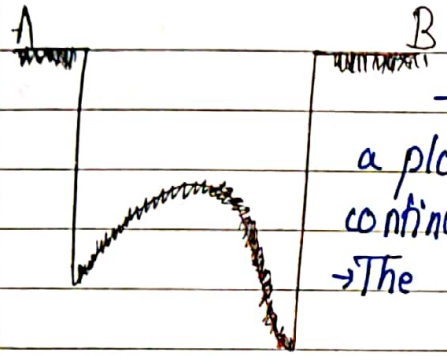
The figure shows a continuous image that we want to convert to digital form. An image may be continuous wrt the x & y coordinates & also in amplitude. To convert it to digital form, we have to sample ~~fn~~ the function in both co-ordinates & in amplitude.

→ Digitizing the co-ordinate values is called sampling.
→ Digitizing the amplitude values is called quantization.

Scanned with CamScanner

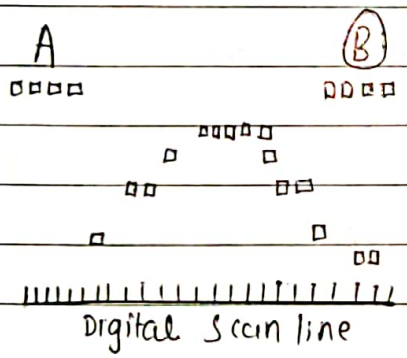
The quality of a digital image is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization.

Representing digital image.

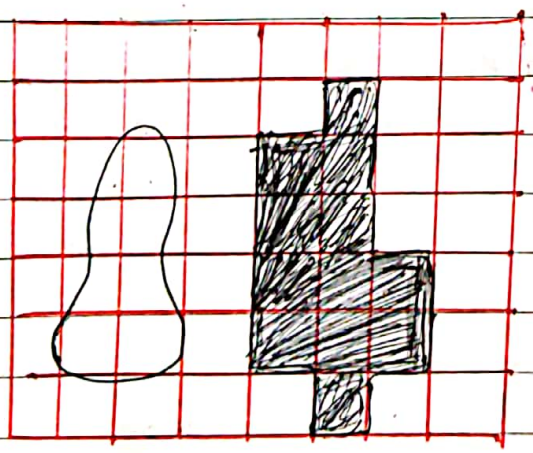
① 

→ The one-dimensional function in figure is a plot of amplitude (intensity level) values of the continuous image along the line segment AB.
→ The random variations are due to image noise.

A scan line from A to B in the continuous image.

② 

→ The continuous intensity level are quantized by assigning one of the 8 values to each sample. The assignment is made depending on the vertical proximity of sample to a vertical tick mark.

③ 

Scanned with CamScanner

Some basic relationships like Neighbours

Neighbours of a Pixel

→ Any pixel $p(x,y)$ has two vertical and two horizontal neighbours given by:

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

⇒ This set of pixels are called the 4-neighbours of p and and is denoted by $N_4(p)$.

⇒ Each of them are at a unit distance from p.

→ The 4 diagonal neighbours of $p(x,y)$ are given by,

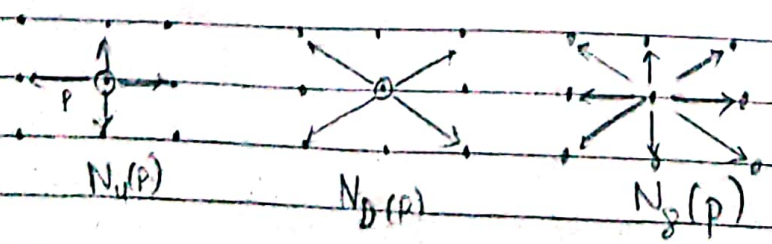
$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

⇒ These set is denoted by $N_D(p)$.

⇒ Each of them are at Euclidean distance of 1.414 from p.

→ The points $N_D(p)$ and $N_4(p)$ are together known as 8-neighbours of the point p, denoted by $N_8(p)$.

→ Some of the points in the N_4, N_D & N_8 may fall outside image when p lies on the border of image.



N_D	N_4	N_D
N_4	p	N_4
N_D	N_4	N_D

$N_8 = (N_4 \cup N_D)$

Scanned with CamScanner

Adjacency and Connectivity

Connectivity between pixels simplifies various digital image concepts like regions and boundaries. To establish connection between two pixels, they must be ~~ne~~ neighbours and their gray level must satisfy a specified criterion of similarity.

For example: In a binary image, two pixels are connected if they are 4-neighbors and have same value (0/1).

Let V be set of gray levels values used to define adjacency, then, we consider the following 3 types of adjacency:-

(a) 4-adjacency : Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

(b) 8-adjacency : Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

(c) m-adjacency : Two pixels p and q with values from V are m-adjacent if :-

→ q is in $N_4(p)$.

→ q is in $N_D(p)$ and the set $[N_4(p) \mid N_4(q)]$ is empty.

Similarly, for connectivity, we consider the following 3 types:-

a) 4-connected ; if q is in set $N_4(p)$.

b) 8-connected, if q is in set $N_8(p)$.

c) m-connected, iff

(i) q is in $N_4(p)$ or

(ii) q is in $N_D(p)$ & the set is empty.

0	1	1
0	1	0
0	0	1

4-connected
m-connected
8-connected

Scanned with CamScanner

Path & Path length

- A path from pixel 'p' with coordinates (x,y) to pixel q with coordinates (s,t) is a sequence of distinct pixels with co-ordinates:

$$(x_0, y_0), (x_1, y_1), (x_2, y_2) \ldots (x_n, y_n).$$

where $(x_0, y_0) = (x, y)$ & $(x_n, y_n) = (s, t)$;

(x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}) ; $1 \le i \le n$

- Here, n is the length of path.

Connected Components

- If p and q are pixels of an image subset S then p is connected to q in S if there is a path from p to q consisting entirely of pixels in S.
- For every pixel 'p' in S, the set of pixels in S that are connected to 'p' is called connected component of S.

Regions and Boundaries

- A subset 'R' of pixel in an image is called a region of image if 'R' is a connected set.
- The boundary of the region R is the set of pixels in the region that have one or more neighbors that are not in 'R'.

Distance measures

Given pixels p, q & z with co-ordinates (x,y), (s,t), (u,v) respectively, the distance function 'D' has following properties:

Scanned with CamScanner

(a) $D(p,q) \geq 0$ $[D(p,q) = 0,$ iff $p = q]$
(b) $D(p,q) = D(q,p)$
(c) $D(p,z) \leq D(p,q) + D(q,z)$

The following are different distance measures:

a) Euclidean Distance :

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

b) City Block Distance

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

c) Chess Board Distance

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

Scanned with CamScanner

Unit : 2
Image Enhancement and Filter in Spatial Domain

Spatial domain enhancement methods

→ Spatial domain techniques are performed to the image plane itself and they are based on direct manipulation of pixels in an image.

→ The operation can be formulated as $g(x,y) = T[f(x,y)]$, where g is the output, f is the input image and T is an operation on f defined over some neighborhood of (x,y).

→ According to the operations on the image pixels, it can be further divided into 2 categories:-

→ Point operations

→ Spatial operations.

I Point Operation

Point operations are zero-memory operations where a given gray level $r \in [0, L]$ is mapped to another gray level $s \in [0, L]$ according to a transformation. The operations deal with pixel intensity values individually. The intensity values are altered ~~using~~ using particular transformation techniques as per the requirement.

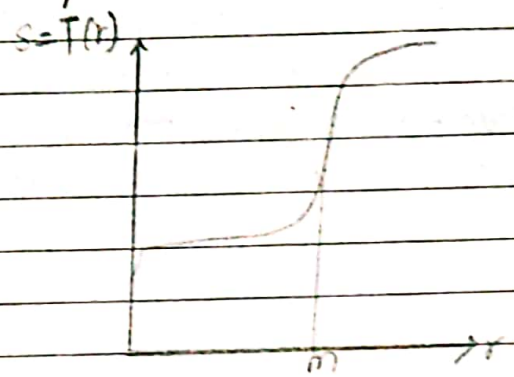
Eg :- Image Negative, Contrast stretching, Thresholding, etc.

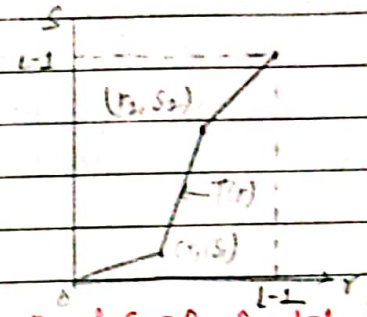
Contrast Stretching

Contrast images occur often due to poor or non uniform lighting conditions, or due to nonlinearity or small dynamic range of the imaging sensor. The result of transformation shown in figure below is to produce an image of higher contrast

Scanned with CamScanner

than the original, by darkening the levels below 'm' and brightening the levels above 'm' in the original image. This technique is contrast stretching.

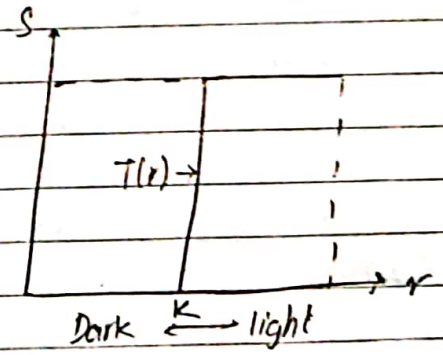




If $r_1 = r_2$ & $s_1 = 0$, $s_2 = L-1$, it gives contrast stretching (threshold)

Thresholding

Extreme contrast stretching yields thresholding. Thresholded image has maximum contrast as it has only black and white gray values.

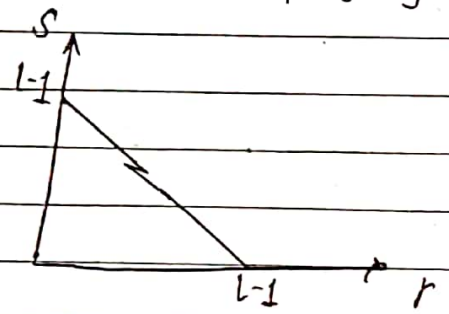


The function T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function. Sometimes in image processing, we need to separate an image into some regions, the process is called segmentation. So, the natural way to segment is thresholding, which is nothing but separation of dark & light regions.

Scanned with CamScanner

Image Negatives

The negative of a digital image is obtained by the transformation function $s = T(r) = L-1-r$ shown in figure below, where L is the number of gray levels. The idea is that the intensity of the output image decreases as the intensity of the input increases. This is useful in numerous applications such as displaying medical images.



Clipping

Image clipping is a useful technique to clip an image from its uninteresting (noisy) background to make it the center of attention for the viewer's eye. Image clipping is the best way to showcase even the most intricate details about the image.

It is a result of capturing or processing an image where the intensity in a certain area falls outside the maximum and minimum intensity which can be represented. In general, clipping is performed with maximum greater (>) or minimum (<) operator.

Example: If we have an image variable and we want to scale it to include only the value greater than or equal to 125 i.e. $x = 125$.

The resulting image variable is created as:-

clipping image = (Input image) > 125

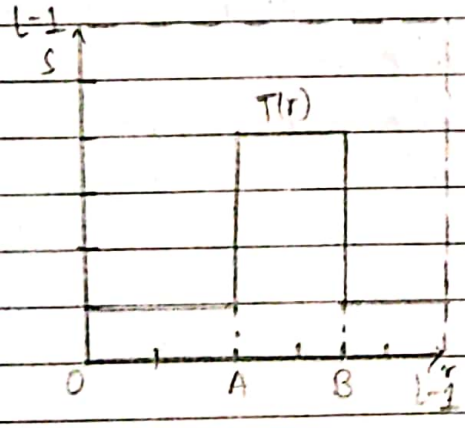
Scanned with CamScanner

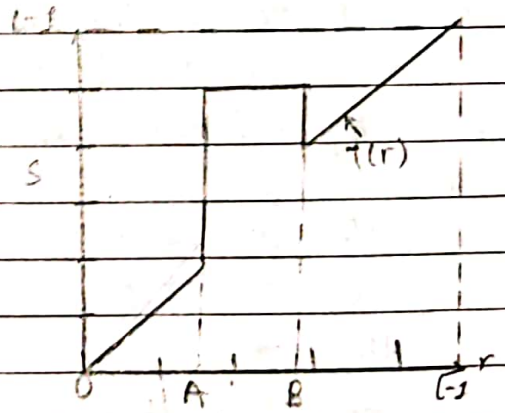
Intensity level slicing

Highlighting a specific range of gray levels in an image often is desired. There are several ways of doing level slicing, but most of them are variations of two basic themes.

One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This process produces a binary image.

The another approach is to brighten the desired range of gray levels but preserve the background and gray level tonalities in the image.





Log transformation

The general form of the log transformation is:

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

where c is the constant and $r \geq 0$.

Log transformation compresses the dynamic range of images with large variations in the pixel values. For eg:- Fourier spectrum.

Scanned with CamScanner

Power law transformation

Power law transformations have the basic form : $s = cr^{\gamma}$ where c and γ are positive constants. Sometimes equations are also written as: $s = c(r+\epsilon)^{\gamma}$.

Bit plane slicing

Bit plane slicing is a method of representing an image with one or more bits of the byte used for each pixel. The three main goals of bit-plane slicing are:-

i) Converting a gray level image to a binary image.
ii) Representing an image with fewer bits and corresponding the image to a smaller size
iii) Enhancing the image by focussing.

110	111	110
000	001	010
110	110	111

Separating the bit planes,

1	1	1
0	0	0
1	1	1

MSB plane

1	1	1
0	0	1
1	1	1

Central bit plane

0	1	0
0	1	0
0	0	1

LSB plane

Scanned with CamScanner

Histogram Processing

Histogram

The histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k) = (n_k)$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k.

The histogram generation process is a systematic counting of the number of pixels of every gray level. The process is implemented by :-

- → iterating through all possible gray level values
- → for each gray level value, counting the number of pixels in the image that have that value.
- → storing the results in a table or displaying in a chart

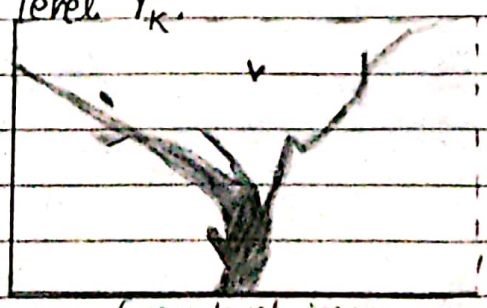
The shape of histogram provides key information on the overall quality of image and the possibility for enhancement.

Normalized Histogram

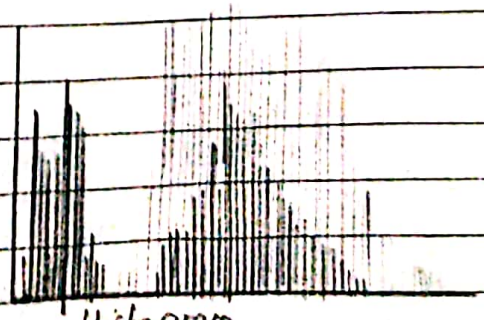
It is a common practise to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by :

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

Here $p(r_k)$ gives an estimate of probability of occurrence of gray level r_k.



Gray level image



Histogram

Scanned with CamScanner

Histogram Modeling

Histogram modeling techniques modify an image so that its histogram has a desired shape. This is useful in stretching the low contrast levels with narrow histograms.

→ **Histogram equalization**

Histogram equalization is an automatic contrast enhancement technique where pixels of entire image are uniformly distributed among all possible gray level.

The basic idea is to find a transformation function such that the histogram of the modified (equalized) image is flat (uniform).

→ In continuous domain, with gray level ranges $[0, L-1]$, the transformation function is:

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

→ In discrete domain, the monotonic property of transformation T implies:

Discrete values:

$$S_k = T(r_k) = (L-1) \sum_{j=0}^{k} P_r(r_j)$$

$$= (L-1) \sum_{j=0}^{k} \frac{n_j}{MN}$$

$$= \frac{L-1}{MN} \sum_{j=0}^{k} n_j \quad , \text{for } k = 0, 1, \ldots L-1$$

Scanned with CamScanner

Example, Suppose that a 3 bit image ($L=8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in following table. Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k.

r_k	n_k	$P_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Now,

$$S_0 = T(r_0) = 7 \sum_{j=0}^{0} P_r(r_j) = 7 \times 0.19 = 1.33 \rightarrow 1$$

$$S_1 = T(r_1) = 7 \sum_{j=0}^{1} P_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \rightarrow 3$$

$$S_2 = T(r_2) = 7 \sum_{j=0}^{2} P_r(r_j) = 7 \times (0.19 + 0.25 + 0.21) = 4.55 \rightarrow 5$$

$$S_3 = T(r_3) = 7 \sum_{j=0}^{3} P_r(r_j) = 7 \times (0.19 + 0.25 + 0.21 + 0.16) = 5.67 \rightarrow 6$$

$$S_4 = T(r_4) = 7 \times (0.19 + 0.25 + 0.21 + 0.16 + 0.08) = 6.03 \rightarrow 6$$

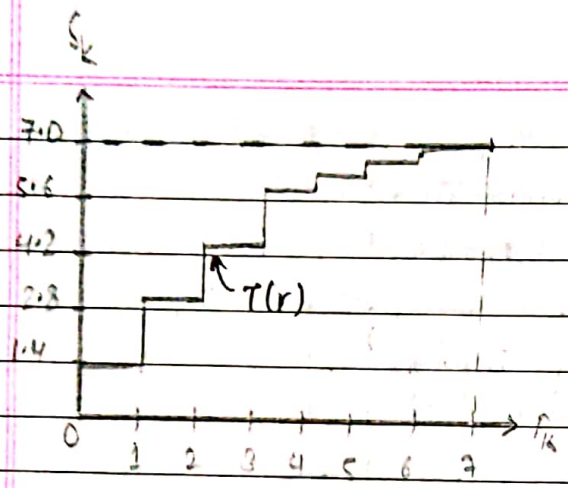
$$S_5 = T(r_5) = 7(0.19 + 0.25 + 0.21 + 0.16 + 0.08 + 0.06) = 6.65 \rightarrow 7$$

$$S_6 = T(r_6) = 7 \times (0.19 + 0.25 + 0.21 + 0.16 + 0.08 + 0.06 + 0.03) = 6.86 \rightarrow 7$$

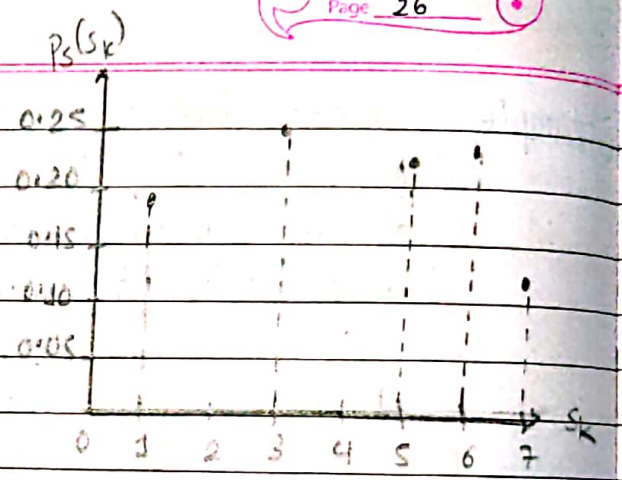
$$S_7 = T(r_7) = 7 \times (0.19 + 0.25 + 0.21 + 0.16 + 0.08 + 0.06 + 0.03 + 0.02) = 7.00$$

Figure →

Scanned with CamScanner



Transformation function



Equalized Histogram

Spatial Operations

Spatial Domain

The pixel representation of the image is called spatial domain. In spatial domain, the operations are directly applied to pixel values.

The value of pixel is in the range 0 to 255.

Spatial Filtering

Spatial filtering is an image processing technique for changing the intensities of a pixel according to the intensities of the neighboring pixels.

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

where $f(x,y)$ = input image, $g(x,y)$ = output image &

T = operator like average, median, etc.

Scanned with CamScanner

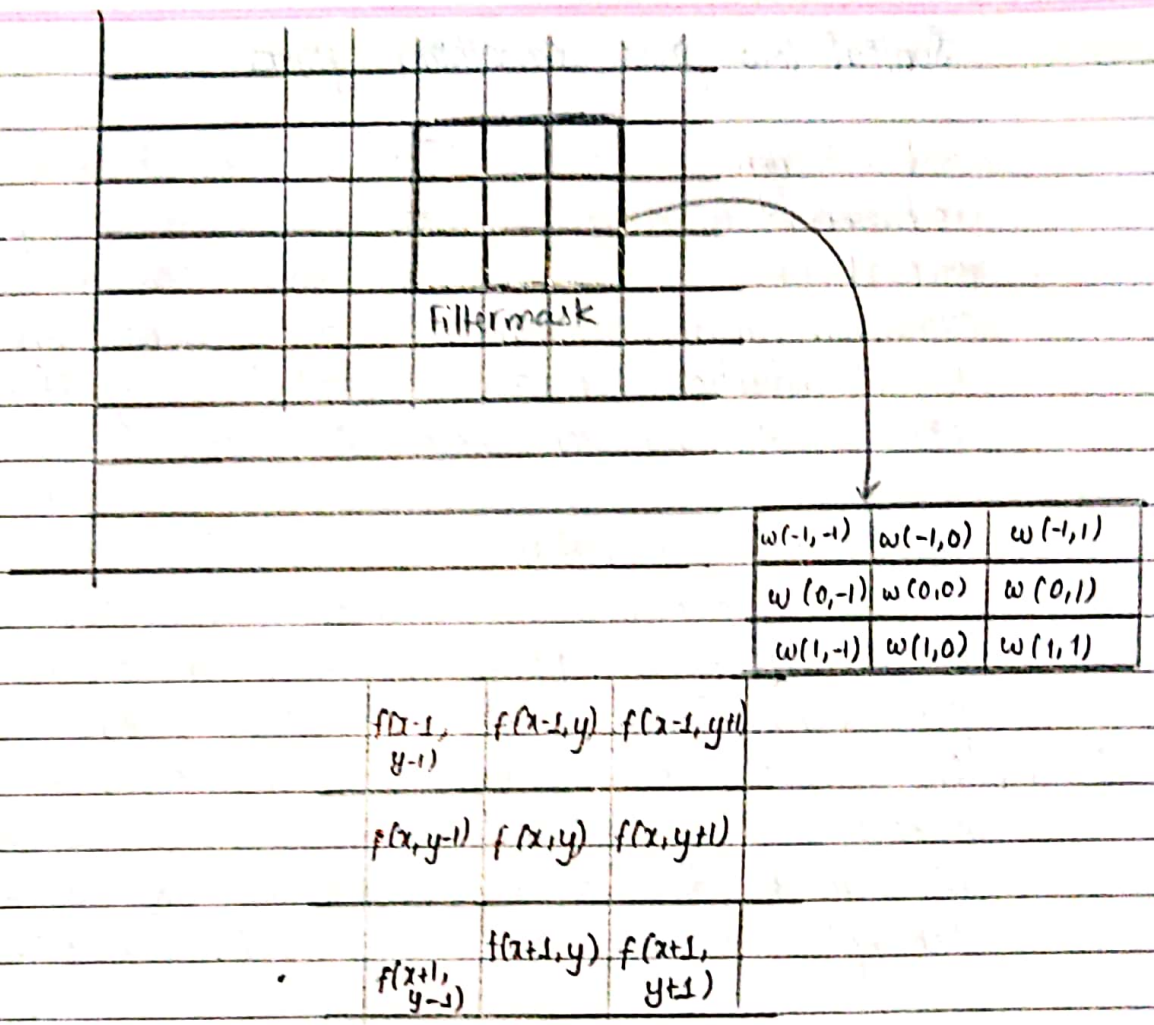


Figure :- Mechanics of Spatial Filtering.

In general, linear filtering of an image 'f' of size MXN with a filter mask of size mxn is given by :-

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

The process of linear filtering is similar to frequency domain concept called convolution. It is often reffered as "convolving a mask with an image."

Scanned with CamScanner

Spatial low pass smoothing filters

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

① Smoothing linear filters

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

Idea :- By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced "sharp" transition in gray level. Eg: Image Averaging

$$\frac{1}{9} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad , \quad \frac{1}{16} * \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

② Order-statistics Filters (Non-linear)

These filters are non-linear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

Eg: Median filter.

Scanned with CamScanner

• Averaging Filters

→ The output of an averaging filter is simply the average of the pixels contained in the neighborhood of the filter mask.

→ The algorithm operates by replacing the values of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

→ Result is reduction of noise.

→ This technique has an undesirable side effect on the edge, i.e. blurs the edge.

→ The general implementation for filtering an MxN image with a weighted averaging filter of size mxn is given by:

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

where $m = 2a+1$, $n = 2b+1$

• Median Filters

→ It is a non-linear digital filtering technique, often used for noise removal.

→ Median filter replaces the intensity value of that pixel by the median of the neighborhood of that pixel.

→ While choosing the neighborhood of a pixel, it should not exceed beyond dimension of the original image.

→ It is best to remove impulse noise (salt and pepper). Impulse has either min value (0) or max value (255).

Scanned with CamScanner

Steps for solving:

1. Write the value in ascending order.
2. Marked value should lie in mid place or at center of ascending order.

Q^n, Compute median value of underlined pixels.

18	22	33	25
34	128	24	172
22	19	32	31

Soln →

Median value for 128

18	22	33
34	128	24
22	19	32

Ans:

18	22	33
34	24	24
22	19	32

In ascending order:

18 19 22 22 24 32 33 34 128

Median value for 24:

22	33	25
128	24	172
19	32	31

Ans:

22	33	25
24	25	172
19	32	31

In ascending order,

19 22 24 24 25 31 32 33 172

Final output image:

18	22	33	25
34	24	25	172
22	19	32	31

Scanned with CamScanner

Median Vs Averaging filters.

1) Median completely discards the spike whereas averaging always responds to all aspects.

2) Median filter preserves discontinuities whereas averaging produces rounding-off effects.

High pass sharpening filters

Sharpening refers to the process of highlighing fine details in an image or to enhance the detail that has been blurred. A image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness. The kernel of high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels.

→ Single value at center is positive.
→ All other values are negative.

Eg:

$$\begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & 8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$$

Properties:

1) It has both positive as well as negative values.
2) The sum of all values in high filter mask is equal to zero.
3) The edge content is increased by derivative mask.
4) As the size of mask grows, more edge content is increased.

Scanned with CamScanner

Example :-

let us take 3*3 matrix as image.

10	20	30
10	120	30.
10	20	30

we take high pass mask,

$1/9$

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

For sharpening, we pad values in Image matrix so as to make 10 (at 1st position) to be at center., we get,

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	20	20	30	30

Now,

calculation for 10,

$= \{(10*8 - 10*(-1) + 10*(-1) + 20*(-1) + 10*(-1) + 20*(-1) + 10*(-1) + 10*(-1) + 120*(-1)\} / 9$

$= -130/9$

$= -14.44$ (negative range so keeping the negative intensity as 0)

Scanned with CamScanner

After calculating 3x3 matrix, their output matrix becomes as below :-

0	0	0
0	88.88	0
0	0	0

High boost filter

→ It is often desirable to emphasize high frequency component representing the image details (by means such as sharpening) without eliminating low frequency components representing the basic form of the signal.

→ In this case, the high boost filter can be used to enhance high frequency component while still keeping the low frequency components.

→ In this technique, we pass some of the background along with high frequency content.

→ To pass some of the background, we multiply the original image with a multiplicative factor A.

Here.

High Pass Filter Low Pass filter

Original img = HPF img + LPF img

LPF = original img − HPF

HBF = A (original img) − LPF

or, HBF = A (original img) − [original img − HPF]

or, HBF = A (original img) − original img + HPF

∴ HBF = original img [A−1] + HPF

If A=1, then, High boost = High pass.

If A>1, original signal is added to High pass result.

Scanned with CamScanner

High frequency Emphasis filtering

The high-frequency emphasis filter helps in the sharpening of an image by emphasizing the edges; since the edges usually consists of a sharp change in intensity levels, they represent the high-frequency spectrum of the image. The high frequency emphasis (HFE) filtering is a kind of 2-D Fourier filtering which would require the following 5 steps to be followed :-

Input image → Fourier transform $F(u,v)$ → Filter function $F^* H_{HFE}$ → Inverse Fourier → final image

Gaussian High pass

$$H = 1 - e^{-D^2/2D_0^2}$$

High Frequency emphasis

$$H_{HFE} = k_1 + k_2 H$$

Steps :-

1. Load image.
2. Compute the Fast Fourier transform and FFT shift of the original image.
3. Compute the HFE filter using a Gaussian High-Pass filter.
4. Apply the HFE filter (by multiplying HFE with the FFT of original image).
5. Perform the inverse Fourier transform and generate an image to view the results.

Scanned with CamScanner

Gradient based filters

Image gradients can be used to extract information from images. Gradient images are created from the original image (generally by convolving with a filter). Each pixel of a gradient image measures the change in intensity of that same point in the original image, in a given direction. To get the full range of direction, gradient images in the x and y directions are computed. One of the most common uses is in edge detection.

Robert Cross Gradient Operators

Robert Cross operator is used in image processing for edge detection. As a differential operator, the idea behind the Robert cross operator is to approximate the gradient of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels.

$$\nabla I(x,y) = G(x,y) = \sqrt{G_x^2 + G_y^2}$$

Prewitt filters

The prewitt operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. Prewitt operator provides two masks one for detecting edges in horizontal direction and another for detecting edges in an vertical direction.

Scanned with CamScanner

Vertical direction

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

Horizontal Direction

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

Sobel filters

The sobel operator is very similar to Prewitt operator. It is also used to detect two kinds of edges in an image:-

→ Vertical direction

→ Horizontal direction.

The major difference is that in Sobel operator, the coefficients of masks are not fixed and they can be adjusted as per requirement unless they donot violate any property of derivative masks.

Vertical mask:

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

Horizontal mask:

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

Derivative operator

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

→ Thus image differentiation enhances edges & noises.

→ It deemphasizes area with slowly varying gray-level values.

Scanned with CamScanner

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

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

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

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

First derivative

- Must be zero in flat segments
- Must be nonzero at the onset of a gray level step or ramp,
- Must be nonzero along ramps.

Second derivative

- Must be zero in flat areas
- Must be nonzero at the onset and end of a gray-level step
- Must be zero along ramps of constant slope

Laplacian filters

The laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection.

The laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian

Scanned with CamScanner

smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together here. The operator normally takes a single gray image as input and produce another grayscale image as output.

→ The laplacian is defined as follows:

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

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x,y) \quad \text{[In x-direction]}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y) \quad \text{[In y-direction]}$$

So, the Laplacian can be given as follows:

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

We can easily build a filter based on this

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

0	1	0
1	-4	1
0	1	0

Image sharpening in the way of using the laplacian:

$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)]$$

where,

$f(x,y)$ = input image,

$g(x,y)$ = sharpened images,

$c = -1$ if $\nabla^2 f(x,y)$ corresponding to 1st two filters

$c = 1$ if either of the other two filters is used

→ There are two types of laplacian filter:
- Positive laplacian operator
- Negative laplacian operator.

* Positive laplacian Operator

0	1	0
1	-4	1
0	1	0

In positive laplacian we have standard mask in which center element of the mask should be negative and corner elements of mask should be zero. Positive laplacian Operator is used to take out outward edges in an image.

* Negative laplacian Operator

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

In negative laplacian operator, we also have a standard mask, in which center element should be positive, all the elements in corner should be zero & rest of all elements in mask should be -1. It is used to take out inward edges in an image.

Examples:

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

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

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

Application
→ Image sharpening
→ Edge detection
→ Blob detection.

Scanned with CamScanner

Magnification

Image magnification ideally is a process which virtually increases image resolution in order to highlight implicit information present in the original image. It can also be referred as scale transformation. Before applying magnification technique, any coded image has to be converted into normal form through decoding process.

(a) Magnification by replication

It is also known as nearest neighbor interpolation. In this method, we just replicate the neighboring pixels.

Working : In this method, we create new pixels from the already given pixels. Each pixel is replicated in this method 'n' times row wise and column wise & we get a zoomed image.

Eg: let us consider an image of 2 rows and 2 columns,

1	2
3	4

- Row wize zooming :

When we zoom it row wise, we just simply copy the rows pixels to its adjacent new cell.

When replicated twice, we get following matrix :-

1	1	2	2
3	3	4	4

- Column wize zooming

The next step is to replicate each of the pixel column wise, that we simply copy the column pixel to its adjacent new column.

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Scanned with CamScanner

The new image has a dimension of :-

(Original image rows * zooming factor, Original image cols* zooming factor)

Advantage

→ Simple to perform (only requires copying the pixels).

Disadvantage

→ Image gets zoomed but output is very blurry.
The more the zooming factor, the more blurry the image is.

(b) Magnification by interpolation.

Image interpolation occurs in all digital photos for eg: in photo enlargement. It happens anytime we resize or remap our image from one pixel grid to another. Image resizing is necessary when we need to increase or decrease the total number of pixels, whereas remapping can occur under a wider variety of scenarios: correcting for lens distortion, changing perspective, & rotating an image.

→ Image will always loose some quality each time interpolation is performed.

Common interpolation algorithms can be grouped into two categories:-

1) Adaptive methods
2) Non - adaptive methods

Adaptive methods change depending on what they are interpolating (sharp edges vs smooth texture), whereas non-adaptive methods treat all pixels equally.

Scanned with CamScanner

Nearest neighbor interpolation

This process requires the least processing time because it only considers one pixel.

- Bilinear interpolation

It considers the closest 2x2 neighborhood of known pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to arrive at its final interpolated value. This results in much smoother looking images than nearest neighbor.

- Bicubic interpolation

Bicubic goes one step beyond bilinear by considering the closest 4x4 neighboorhood of known pixels - for a total of 16 pixels. This method produces noticeably sharper images.

→ It is a standard method in many image editing programs (including Adobe photoshop).

Scanned with CamScanner

Unit - 3

Image Enhancement in the Frequency Domain

Introduction to Fourier Transform and the frequency Domain

In frequency domain, we analyze signal with respect of frequency. The frequency domain methods of image enhancement are based on convolution theorem. This is represented as,

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

where,

$g(x,y)$ = resultant image
$h(x,y)$ = position invariant operator
$f(x,y)$ = Input image

The Fourier transform representation of equation above is

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

The function $H(u,v)$ in equation is called transfer function. It is used to boost the edges of input image $f(x,y)$ to emphasize the high frequency components.

In spatial domain, we deal with the images as it is but in frequency domain, we first transform the image to it's frequency distribution, and perform the desired processing. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain.

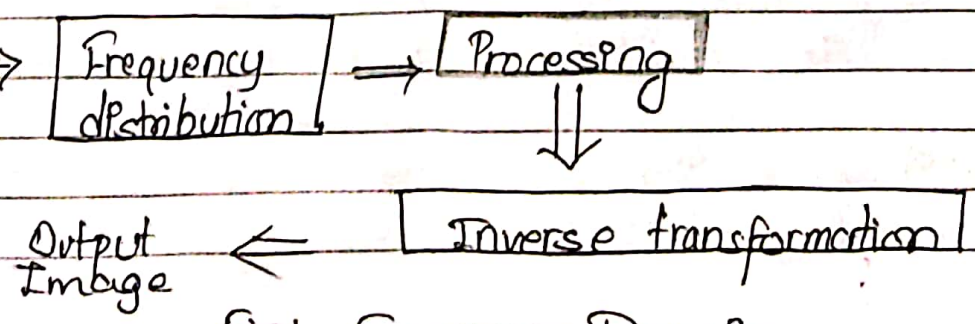


Fig: Frequency Domain

Scanned with CamScanner

Transformation

A signal can be converted from time domain into frequency domain using mathematical operators called transforms. Some of the transformation are:-

- Fourier Series
- Fourier transformation
- laplace transform
- Z transform.

Frequency Components

① High frequency components
→ corresponds to edges in an image

② Low frequency components
→ corresponds to smooth regions in an image.

Fourier transformation

→ Both fourier series and fourier transformation are mathematical tools for representing signal in frequency domain.

→ Fourier series is mainly used for periodic signal and Fourier transform is used for aperiodic signal.

→ The Fourier transform decomposes a waveform - basically any real world waveform, into sinusoids.

Condition for Existence.

(i) $f(t)$ should be absolutely integrable i.e. $\int_{-\infty}^{\infty} |f(t)|\, dt < \infty$

(ii) The function must have finite number of maxima & minima.

(iii) The function must have finite number of discontinuities.

Scanned with CamScanner

Representation

$$x(t) \rightleftharpoons X(j\omega) \text{ or } X(f)$$

$$x(j\omega) = |x(j\omega)| \angle x(j\omega)$$

formula

1) $x(t) \longrightarrow x(j\omega)$

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt \longleftarrow \text{FT}$$

2) $x(j\omega) \longrightarrow x(t)$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(j\omega) \cdot e^{j\omega t} d\omega \longleftarrow \text{IFT}$$

Fourier transform can exist for

- Energy signal (E → finite, P → 0)
- Power signal, (P → finite, E → ∞)
- Impulse related signal.

1-D Continuous Fourier transform

→ Let $f(x)$ be a continuous function of a row value x.

→ The fourier transform of $f(x)$ denoted by

$$x(t) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt \quad \text{where } j = \sqrt{-1}$$

Q A periodic waveform in time domain. Calculate its fourier transform. (1D)

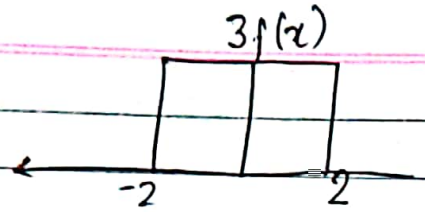
Solution,

Note : $\sin\theta = \frac{1}{2j}\left[e^{j\theta} - e^{-j\theta}\right]$ where, $j = \sqrt{-1}$

$$\begin{cases} f(t) = 3 & -2 \le t \le 2 \\ f(t) = 0 & \text{otherwise} \end{cases}$$

Scanned with CamScanner

$\int e^{-x} = -\frac{1}{x} e^{-x}$



$$\int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt$$

$$= \int_{-2}^{2} 3e^{-j\omega t}\, dt$$

$$= 3\int_{-2}^{2} e^{-j\omega t}\, dt$$

$$= \frac{3}{-j\omega}\left[e^{-j\omega t}\right]_{-2}^{2} = \frac{3}{j\omega}\left[-e^{-2j\omega} + e^{2j\omega}\right]$$

$$= \frac{3}{j\omega}\left[-e^{-2j\omega} + e^{2j\omega}\right]$$

$$= \frac{3}{j\omega}\left[e^{2j\omega} - e^{-2j\omega}\right] \times \frac{2}{2}$$

$$= \frac{6}{\omega} \cdot \frac{1}{2j}\left[e^{2j\omega} - e^{-2j\omega}\right]$$

$$= \frac{6}{\omega} \sin 2\omega$$

→ The fourier transform, $F(u)$ of a single variable continuous function, $f(x)$ is defined by:

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

Given fourier transform of a function $F(u)$, the inverse fourier

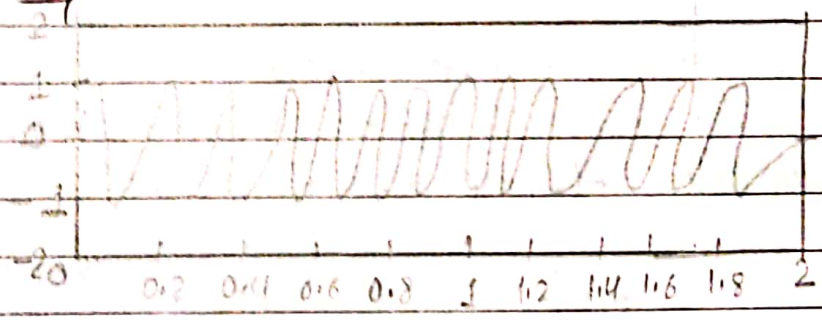
Scanned with CamScanner

transform can be used to obtain. $f(x)$, by:

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

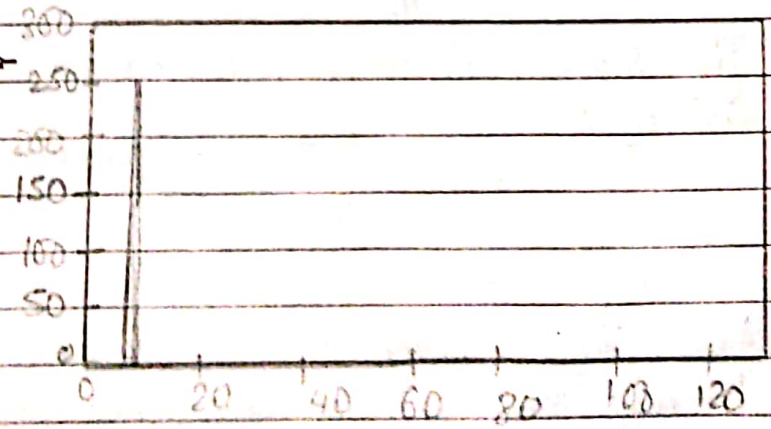
Note: $F(u)$ is the frequency spectrum, where u represents the frequency and $f(x)$ is the signal where x represents time/space
$j = \sqrt{-1}$.

Time/Space domain ⇒

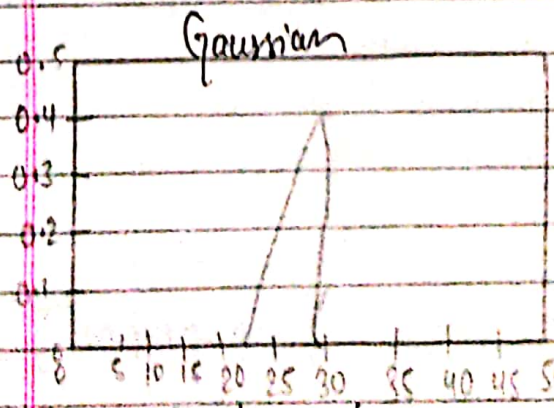


sine wave

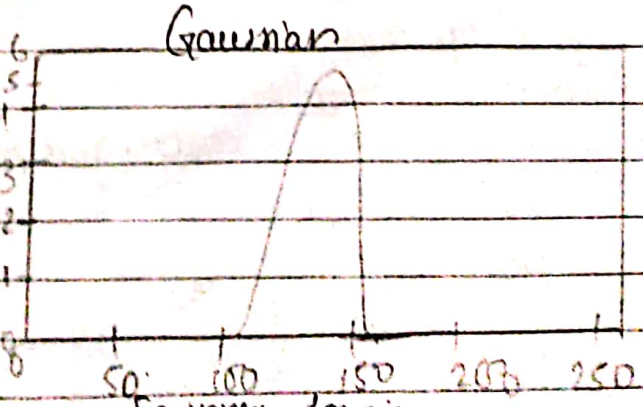
Frequency domain ⇒



Delta function



Time/Space domain



frequency domain

Scanned with CamScanner

2-D Continuous Fourier Transform.

The Fourier transform can be extended to 2 dimensions.

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

and it's Inverse transform is given as:

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

1-D Discrete Fourier Transform.

The fourier transform, $F(u)$, of a discrete function of one variable, $f(x)$, $x = 0, 1, 2, \ldots m-1$ is given by:

$$F(u) = \frac{1}{m} \sum_{x=0}^{m-1} f(x)\, e^{-j2\pi ux/m}$$

The Inverse discrete fourier transform can be used to calculate $f(x)$ by:

$$f(x) = \sum_{u=0}^{m-1} F(u)\, e^{j2\pi ux/m}$$

Note: $F(u)$, which is the Fourier transform of $f(x)$ contains discrete complex quantities and it has the same number of component as $f(x)$.

$$e^{j\theta} = \cos\theta + j\sin\theta$$

Then,

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

Scanned with CamScanner

→ The Fourier Transform generates complex quantities. The magnitude and spectrum of the Fourier transform is given by:

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad ; R(u) = \text{real part}$$
$$I(u) = \text{imaginary part}$$

→ The phase spectrum of the transform refers to the angles between the real & imaginary components & it is denoted by:

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

→ Power spectrum / Spectrum density is defined as the square of Fourier spectrum and denoted by.

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

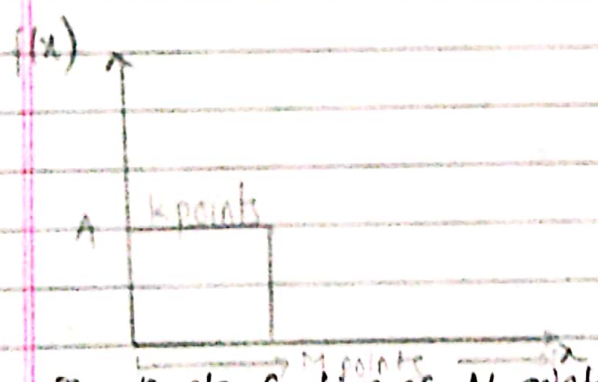


Fig: discrete function of M points

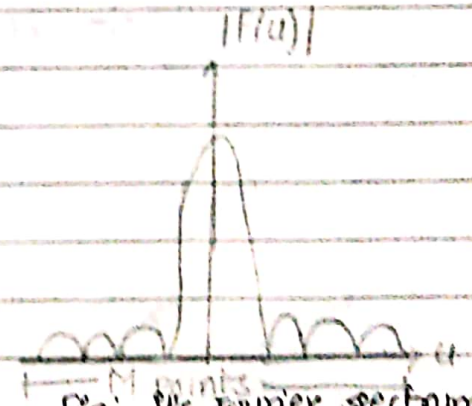


Fig: its Fourier spectrum

2D Discrete Fourier Transform

The Fourier transform of a 2D discrete function (image) $f(x,y)$ of size $M \times N$ is given by:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-2j\pi(ux/M + vy/N)}$$

$u = 0,1,2,\ldots M-1$ & $v = 0,1,2,\ldots N-1$.

Scanned with CamScanner

The inverse 2D discrete fourier transform can be calculated by:

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

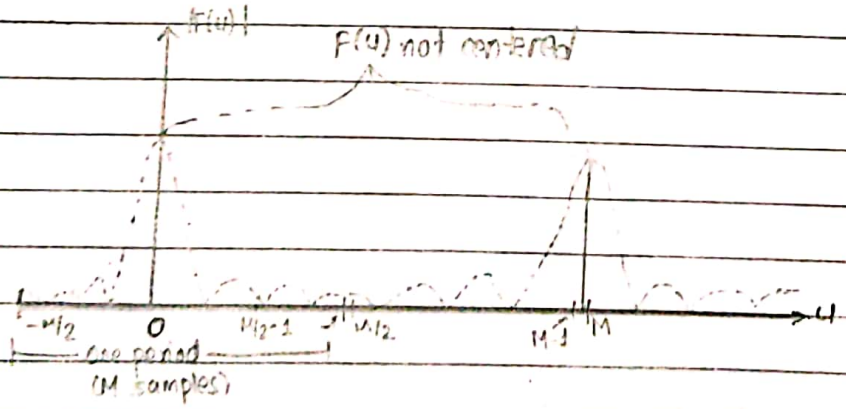
$x = 0, 1, \ldots M-1$ & $y = 0, 1, 2, \ldots N-1$

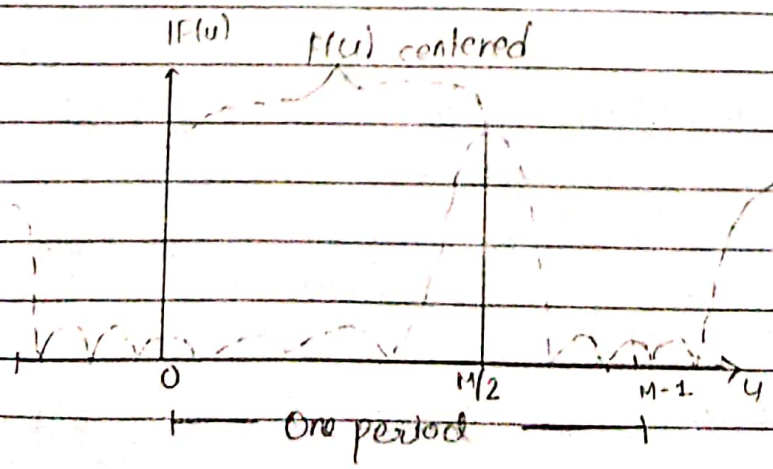
The 2D fourier Spectrum, Phase spectrum & Power Spectrum can be respectively given as:

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

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

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





Scanned with CamScanner

Properties of Fourier Transform

- **Linearity Property**

If $x(t) \xleftrightarrow{F.T} X(\omega)$ and $y(t) \xleftrightarrow{F.T} Y(\omega)$

Then linearity property states that

$$ax(t) + by(t) \xleftrightarrow{F.T} aX(\omega) + bY(\omega)$$

- **Time shifting Property**

If $x(t) \xleftrightarrow{F.T} X(\omega)$, then time shifting property states that

$$x(t-t_0) \xleftrightarrow{F.T} e^{-j\omega t_0} X(\omega)$$

- **Frequency shifting Property**

If $x(t) \xleftrightarrow{F.T} X(\omega)$, then frequency shifting property states that

$$e^{j\omega_0 t} \cdot x(t) \xleftrightarrow{F.T} X(\omega-\omega_0)$$

- **Time Reversal Property**

If $x(t) \xleftrightarrow{F.T} X(\omega)$ then time reversal property states that

$$x(-t) \xleftrightarrow{F.T} X(-\omega)$$

- **Time Scaling Property.**

If $x(t) \xleftrightarrow{F.T} X(\omega)$, then time scaling property states that

$$x(at) \frac{1}{|a|} X \frac{\omega}{a}$$

- **Differentiation & Integration Properties**

If $x(t) \xleftrightarrow{F.T} X(\omega)$, then differentiation property states that,

$$\frac{dx(t)}{dt} \xleftrightarrow{FT} j\omega . X(\omega) \quad , \quad \frac{d^n x(t)}{dt^n} \xleftrightarrow{F.T} (j\omega)^n . X(\omega)$$

& integration property states that:

$$\int x(t)\,dt \xleftrightarrow{F.T} \frac{1}{j\omega} X(\omega) \quad , \quad \iiint\ldots\int x(t)\,dt \xleftrightarrow{FT} \frac{1}{(j\omega)^n} X(\omega)$$

Scanned with CamScanner

- Multiplication and Convolution Properties

If $x(t) \xleftrightarrow{F.T.} X(\omega)$ & $y(t) \xleftrightarrow{F.T} Y(\omega)$

Then multiplication property states that,

$$x(t) \cdot y(t) \xleftrightarrow{F.T} X(\omega) * Y(\omega)$$

and convolution property states that,

$$x(t) * y(t) \xleftrightarrow{F.T} \frac{1}{2\pi} X(\omega) \cdot Y(\omega)$$

Smoothing Frequency Domain Filters

→ Edge and other sharp transition such as noise in the gray level of an image contribute significantly to high frequency content of its Fourier transform.

→ Thus, smoothing may be achieved in the frequency domain by attenuating a specified range of high frequency components in the transform of a given image.

→ Filtering represented as : $F(u,v) = G(u,v) \cdot H(u,v)$.

where $G(u,v)$ is the Fourier Transform of the image to be smoothed and

$H(u,v)$ is transfer function which yields $F(u,v)$ by attenuating the high frequency components of $G(u,v)$.

Scanned with CamScanner

There are three types of low pass filter.

i) Ideal LPF
ii) Butterworth Filter
iii) Gaussian LPF

Hence, an image can be smoothed in the frequency domain by attenuating the high frequency content of its fourier transform. This would be a low pass filter.

Ideal low pass filter

The simplest low pass filter may be a filter which cuts off all high frequency components of the Fourier transform that are at a distance greater than a specified distance D_0 from the origin of the transform. Such type of filter is known as 2D ideal low pass filter. The center coefficient $F(M/2, N/2)$ correlates to lower frequency of the original image. As we move away from the center, in any direction, the fourier coefficients correlates to higher frequencies.

An ~~ideal~~ ideal low pass filter with cut off frequency 'r_0' is given by following relation.

$$H(u,v) = \begin{cases} 1, & \text{if } \sqrt{u^2+v^2} \le r_0 \\ 0, & \text{if } \sqrt{u^2+v^2} > r_0 \end{cases}$$

→ The cut off frequency of the ideal LPF determines the amount of frequency components passed by the filter.

→ Smaller the value of r_0, more the number of image components are eliminated by the filter.

→ In general, the value of r_0 is choosen such that most of the components of interest are passed through while most components of less interest are cut off.

Scanned with CamScanner

Butterworth low pass filter

A two dimensional Butterworth low pass filter has the transfer function:

$$H(u,v) = \frac{1}{1+\left[\frac{\sqrt{u^2+v^2}}{r_0}\right]^{2n}}$$

n: filter order
r_0: cutoff frequency

→ Frequency response does not have a sharp transition as in the ideal LPF.
→ This is more appropriate for image smoothing than the ideal LPF, since this not introduce ringing.
→ The BLPF transfer function does not have a sharp discontinuity that establishes a clear cutoff between passed and filtered frequencies.
→ For filters with smooth transfer functions, defining a cutoff frequency locus at points for which H(u,v) is down to a certain fraction of its maximum value is customary.

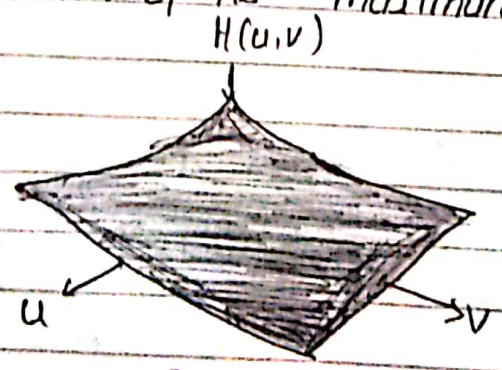


fig: BLPF.

Scanned with CamScanner

Gaussian LPF

The transfer function of a Gaussian lowpass filter is defined as :

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

where,

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

i.e. $D(u,v)$ is the distance from the point (u,v) to the origin of frequency. D_0 is specified non negative quantity.

The parameter D_0 measures the dispersion of the Gaussian curve. Larger the value of D_0, larger the cut off frequency and milder the filtering.

Sharpening frequency Domain filters

An image can be blurred by attenuating high frequency components of its Fourier transform. Since, edge and other abrupt changes in grey level are associated with high frequency components so image sharpening may be achieved by applying high pass filter which attenuates the low frequency components without disturbing high frequency information in the Fourier transform.

The Fourier function of HPF may be obtained by using following relation :-

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

High pass filter are the reverse of low pass filter.

Scanned with CamScanner

Date
Page

Types of HPF

→ Ideal HPF
→ Butterworth HPF
→ Gaussian HPF

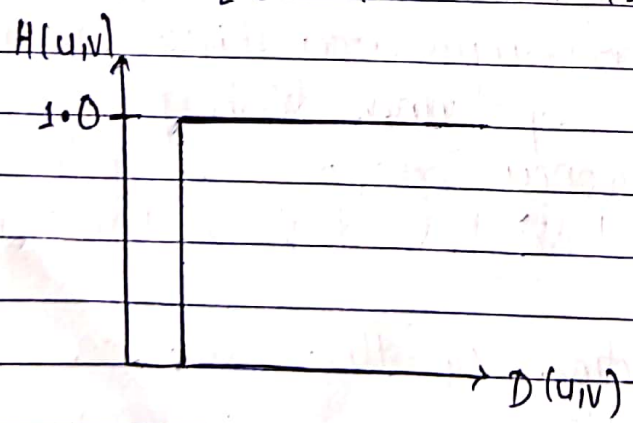
- **Ideal HPF**

A 2D ideal HPF may be described with the help of following equation

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

i.e. D_0 is specified non-negative quantity. $D(u,v)$ is the distance from the point (u,v) to the origin of frequency given as :

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



- **Butterworth HPF**

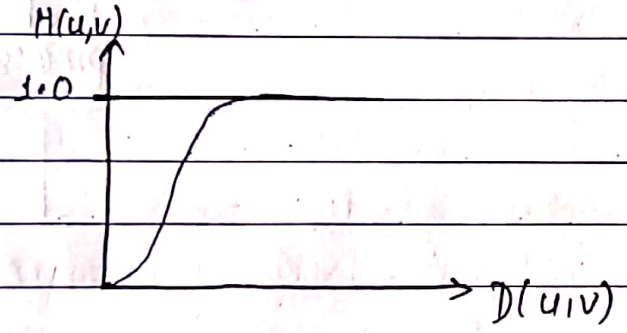
The transfer function of the butterworth HPF of order 'n' with cut off frequency at a distance D_0 from the origin is described with the help of following equation :-

Scanned with CamScanner

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

i.e. D_0 is specified non negative quantity. $D(u,v)$ is the distance from the point (u,v) to the origin of frequency given as

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



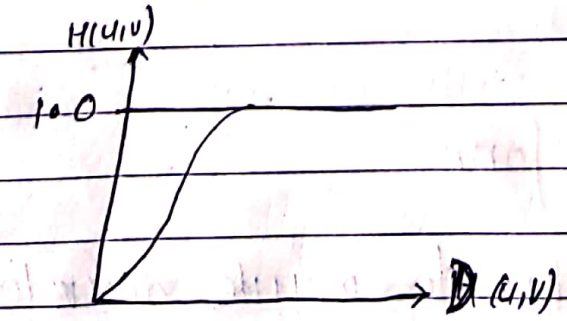
- Gaussian HPF

The transfer function of Gaussian HPF with cutoff frequency at distance D_0 from origin is described with the help of following equation:

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

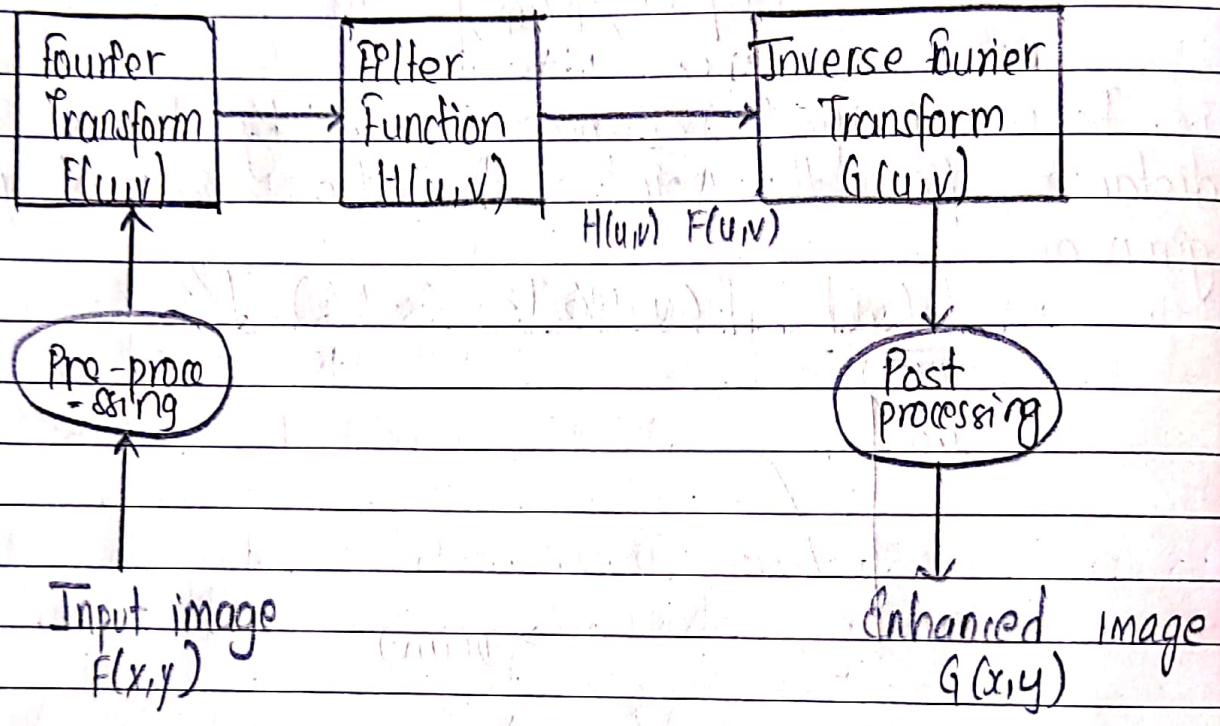
i.e. D_0 is the specified non-negative quantity. $D(u,v)$ is the distance from the point (u,v) to the origin of frequency given as:

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



Scanned with CamScanner

Frequency Domain Enhancement



Steps for filtering in frequency domain

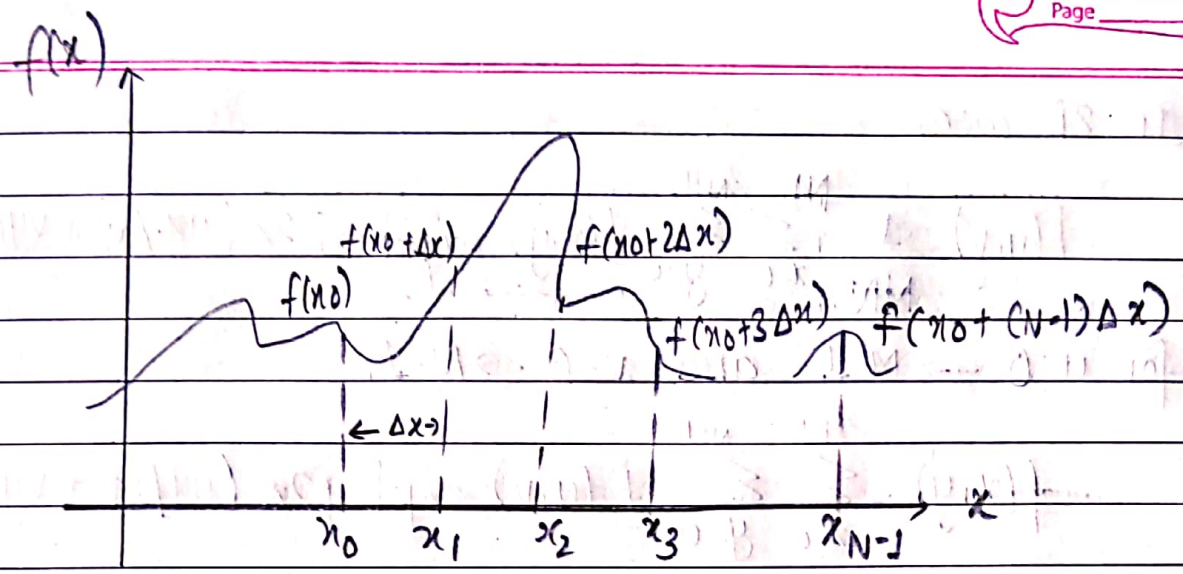
1) $f(x,y)$, dimension $M \times N$
2) $f_p(x,y)$, dimension $P \times Q$, $P = 2M$ and $Q = 2N$.
3) $f_p(x,y) \times (-1)^{x+y}$
4) $H(u,v)$ – filter, $F(x,y)$
   $G(u,v) = H(u,v) \times F(x,y)$
5) $g(x,y) = \mathcal{F}^{-1}[G(u,v)](-1)^{x+y}$

Fast Fourier Transform

→ FFT is an algorithm to compute discrete fourier transform and its inverse.

→ DFT is obtained by decomposing a sequence of values into a components of frequencies $f(u)$.

Scanned with CamScanner



In above figure, $f(x)$ is a continuous function to be changed by taking N- sample.

→ Suppose a continuous function, $f(x)$ is discretized into a sequence.

$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \ldots f(x_0+(N-1)\Delta x)\}$$

by taking N samples, Δx units apart.

→ Let x refer to either a continuous or discrete value by saying

$$f(x) = f(x_0+\Delta x)$$

where $x = 0, 1, \ldots N-1$ and $\{f(0), f(1), \ldots f(N-1)\}$ denotes any N uniformly spaced samples from a corresponding continuous function.

→ The Discrete fourier transform is given by

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

for $u = 0, 1, \ldots N-1$.

$$\Delta u = \frac{1}{N\Delta x}$$

Scanned with CamScanner

→ In 2D case,

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j 2\pi (ux/M + vy/N)]$$

for $u=0 \rightarrow M-1$ and $v=0 \rightarrow N-1$.

$$f(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) \exp[j 2\pi (ux/M + vy/N)]$$

for $x=0 \rightarrow M-1$ & $y=0 \rightarrow N-1$.

→ The discrete function $f(x,y)$ represents samples of the continuous function at $f(x_0 + x\Delta x, y_0 + y\Delta y)$

$$\Delta u = 1/M\Delta x, \quad \Delta v = 1/N\Delta y.$$

Discrete Cosine Transform.

→ DFT is very popular due to its good computational efficiency.
→ DCT is similar to DFT but it concerns only with the real parts of image.
→ Since DCT provides high energy compaction capacity, so this method is widely used in image compression. (eg :- JPEG compression)
→ If compaction is high, we can only transmit or store a few coefficients.
→ The general equation for a 1D DCT is given by the following equation :-

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

where $u = 0$ to $N-1$ & $\alpha(u) = \frac{1}{\sqrt{N}}$ for $u=0$

$= \sqrt{2/N}$ for $u = 1$ to $N-1$

Scanned with CamScanner

The general equation for 1D, IDCT is:-

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

where $x = 0$ to $N-1$.

→ For 2D, DCT is defined as:-

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

where $\alpha(u) = \frac{1}{\sqrt{N}}$ for $u=0$

$= \sqrt{\frac{2}{N}}$ for $u=1$ to $N-1$ and $u,v = 0$ to $N-1$

The general equation for 2D IDCT is:

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

where $x, y = 0, 1, \ldots N-1$

→ Signal energy lies at low frequency in image, it appears in the upper left corner of DCT. Compression can be achieved since the lower right values represent higher frequencies, & generally small enough to be neglected with little visible distortion.

Scanned with CamScanner

→ Also, we can compute 2D DCT using 1D DCT as:

- apply 1D DCT (vertically) to columns
- apply 1D DCT (horizontally) to resultant vertical DCT above
- or alternatively Horizontal to vertical.

Example: Image compression using DCT: JPEG Compression.

Input image (N×N) → Construct n×n subimages → forward transform → Quantizer → Symbol encoder → Compressed image

Compressed Image → Symbol decoder → Inverse transform → Merge n×n sub-images → Decompressed image.

Time Complexity

* DFT

→ The time complexity required for a DFT to perform is equal to the order of N^2 or $O(N^2)$.

→ DFT has less speed than the FFT.

* FFT

→ The time complexity reduces in the case of FFT and becomes equal to $O(N \log N)$.

→ It is the faster version of DFT.

Scanned with CamScanner

Convolution

The convolution of two signals in time domain is equivalent to the multiplication of their spectra in frequency domain.

Therefore, if $x_1(t) \xleftrightarrow{FT} X_1(\omega)$ and $x_2(t) \xleftrightarrow{FT} X_2(\omega)$
Then, according to time convolution property of fourier transform,

$$x_1(t) * x_2(t) \xleftrightarrow{FT} X_1(\omega) * X_2(\omega).$$

Correlation

The correlation is one member of the transform pair

$$Corr(g, h) \longleftrightarrow G(f)\, H^*(f)$$

→ More generally, the RHS of the pair is $G(f) . H(-f)$.
→ Usually g & h are real, so $H(-f) = H^*(f)$

* Multiplying the FT of one function by the complex conjugate of the FT of the other gives the FT of their correlation.

→ This is the Correlation Theorem.

Padding (Zero Padding)

Zero padding is a simple concept; it simply refers to adding zeros to end of a time-domain signal to increase its length. The most common reason to zero pad time-domain data is to make a waveform have a power-of-two number of samples. Thus, it helps in speeding up the processing time.

Hadamard Transform

→ The elements of a basic vectors of a Hadamard transform takes only the binary values +1 & -1 in its kernel matrix.
→ It can be implemented in order of $O[N \log_2 N]$ addition and subtraction.

Scanned with CamScanner

Date
Page

→ Hence, it is very simple & easy to implement but unlike DCT it doesnot have good energy packing capacity.

→ The 1-D Hadamard kernel is given by the relation

$$g(x,u) = \frac{1}{N} (-1)^{\exp\left[\sum_{i=0}^{n-1} b_i(x)\, b_i(u)\right]}$$

where summation in the exponent is performed in modulo 2 arithmetic and $b_k(z)$ is the kth bit in the binary representation of z.

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

where $N = 2^n$ & $u = 0, 1, 2, \ldots N-1$

→ The inverse Hadamard kernel is given by:

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x)\, b_i(u)}$$

→ The 2D kernels are given by:

$$g(x,y,u,v) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} [b_i(x)\, b_i(u) + b_i(y) \cdot b_i(v)]}$$

and

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

→ The 2-D Hadamard transform pair is:-

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

and

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

Scanned with CamScanner

→ The Hadamard transform kernel H_n is a $N*N$ matrix where $N = 2^n$, which can be recursively defined as:

$$H_n = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}$$

Hadamard matrix for $N=2$

$$H_n = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$$

For $N=2$, $N = 2^n$, $n = 1$.

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} H_0 & H_0 \\ H_0 & -H_0 \end{bmatrix}$$

let $H_0 = 1$

then, $H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

For $N=4$, $n=2$

$$H_2 = \left(\frac{1}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\right) \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

for $N=8$, $n=3$

$$H_3 = \frac{1}{2\sqrt{2}} \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Scanned with CamScanner

Haar Transform

→ Haar transform is effective in applications such as signal and image compression in electrical and computer engineering as it provides a simple & computationally efficient approach for analysing the local aspects of a signal.

→ The Haar function $h_k(x)$ are defined on a continuous interval $x \in [0,1]$ and for $k = 0, 1, \ldots N-1$, where $N = 2^n$. The integer k can be uniquely decomposed as.

$$k = 2^p + q - 1$$

where $0 \le p \le n-1$; $q = 0, 1$ for $p = 0$ & $1 \le q \le 2^p$ for $p \ne 0$.

For example, when $N = 4$, $n = 2$ & $k = 0, 1, 2, 3$, we have

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

Representing k by (p,q) the Haar function are defined as

$$h_0(x) = h_{0,0}(x) = 1/\sqrt{N}$$

$$h_k(x) = h_{p,q}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & q-1/2^p \le x \le \frac{q-1/2}{2^p} \\ -2^{p/2} & \frac{q-1/2}{2^p} \le x \le q/2^p \\ 0 & \text{otherwise} \end{cases}$$

and $x \in [0,1]$.

Scanned with CamScanner

Date
Page

Now,

For $k=0$, $p=0$ & $q=0$. $x = 0, 1/4, 2/4, 3/4$.

$$h_0(x) = h_{0,0}(x) = \frac{1}{\sqrt{N}}$$

$$h_{0,0}(0) = h_{0,0}(0) = \frac{1}{\sqrt{4}} = 1/2$$

$$h_{0,0}(1/4) = 1/\sqrt{N} = 1/2$$
$$h_{0,0}(2/4) = 1/\sqrt{N} = 1/2$$
$$h_{0,0}(3/4) = 1/\sqrt{N} = 1/2$$

For $k=1$, $p=0$, $q=1$.

$$h_1(x) = h_{0,1}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & 0 \le x \le 1/2 \\ -2^{p/2} & 1/2 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\therefore h_1(0) = h_{0,1}(0) = \frac{1}{\sqrt{4}} \times 2^{0/2} = \frac{1}{2}$$

$$h_1(1/4) = 1/2$$
$$h_1(2/4) = -1/2$$
$$h_1(3/4) = -1/2$$

for $k=2$, $p=1$, $q=1$

we have,

$$h_2(x) = h_{1,1}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & 0 \le x \le 1/4 \\ -2^{p/2} & 1/4 \le x \le 1/2 \\ 0 & \text{otherwise} \end{cases}$$

$$\therefore h_2(0) = h_{1,1}(0) = \frac{1}{\sqrt{4}} \times 2^{1/2} = \frac{\sqrt{2}}{2}$$

$$h_2(1/4) = h_{1,1}(1/4) = 1/\sqrt{4} \times -2^{1/2} = -\sqrt{2}/2$$
$$h_2(2/4) = h_{1,1}(2/4) = 1/\sqrt{4} \times 0 = 0$$
$$h_2(3/4) = h_{1,1}(3/4) = 1/\sqrt{4} \times 0 = 0$$

Scanned with CamScanner

Date
Page

For $k=3$, $p=1$ and $q=2$, we have,

$$h_3(x) = h_{1,2}(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & 1/2 \le x < 0.75 \\ -2^{p/2} & 0.75 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$h_3(0) = h_{1,2}(0) = \frac{1}{\sqrt{4}} * 0 = 0$$

$$h_3(1/4) = h_{1,2}(1/4) = \frac{1}{\sqrt{4}} * 0 = 0$$

$$h_3(2/4) = \frac{1}{\sqrt{4}} * 2^{1/2} = \frac{\sqrt{2}}{2}$$

$$h_3(3/4) = \frac{1}{\sqrt{4}} * -2^{1/2} = -\frac{\sqrt{2}}{2}$$

The Haar matrix for $N=4$ is as below:-

$1/2$	$1/2$	$1/2$	$1/2$
$1/2$	$1/2$	$-1/2$	$-1/2$
$\sqrt{2}/2$	$-\sqrt{2}/2$	0	0
0	0	$\sqrt{2}/2$	$-\sqrt{2}/2$

$\longrightarrow \frac{1}{2}$

1	1	1	1
1	1	-1	-1
$\sqrt{2}$	$-\sqrt{2}$	0	0
0	0	$\sqrt{2}$	$-\sqrt{2}$

Properties of Fourier Transform

1. Addition theorem $F[f(x,y)+g(x,y)] = F(u,v) + G(u,v)$
2. Shift theorem $F[f(x-a, y-b)] = F(u,v)\, e^{-j2\pi(ua+vb)}$
3. Similarity theorem $F[f(ax, by)] = \frac{1}{|ab|} F(u/a, v/b)$
4. Convolution theorem $F[f(x,y) * g(x,y)] = F(u,v)\, G(u,v)$
5. Rayleigh's theorm $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2\, dx\, dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |F(u,v)|^2\, du\, dv$
6. Evenness & oddness $f_e(x) \Rightarrow f(x) = f(-x)$ & $f_o(x) \Rightarrow f(-x) = -f(x)$
7. Power theorem $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) * g^*(x,y)\, dx\, dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, G^*(u,v)\, du\, dv$

Scanned with CamScanner

Properties of DCT

1) The cosine transform is real and orthogonal.

i.e. $C = C^* \Rightarrow C^{-1} = C^T$

where $C^* =$ conjugate

$C^T =$ transform

2) The cosine transform is not real part of Unitary DFT,

$\therefore T^{-1} = T^{*T}$

3) It is a fast transform method, the cosine transform of N elements can be calculated as $O[N \log_2 N]$.

4) It has excellent energy compaction for highly correlated data.

Properties of Hadamard Transform

1) The Hadamard transform is real, symmetric and orthogonal.

$H = H^* = H^T$.

2) It is fast transform and can be implemented in order of $O[N \log_2 N]$ addition and subtraction.

3) It has only binary values +1 & -1 in its kernel matrix.

4) It has very good energy compaction for highly co-related image.

Properties of Haar Transform

1) It is symmetric, separable unitary transform.

2) It is orthogonal and real.

$\underbrace{T^{-1} = T^T}_{\text{matrix inverse = matrix transpose}}$ $\qquad$ $\underbrace{H_r = H_r^*}_{\text{Conjugate of } H^T = H^T}$

3) It is a fast transform and can be implemented in $O(N)$.

4) It has poor energy compaction property.

5) It exists for $N = 2^n$ where 'n' is an integer.

Scanned with CamScanner

Date
Page

Unit - 4
Image Restoration and Compression

Image Restoration

→ Goal of image restoration technique is to improve an image in some predefined sense.

→ Restoration attempts to reconstruct or recover an image that has been degraded by using a prior knowledge of the degradation phenomenon.

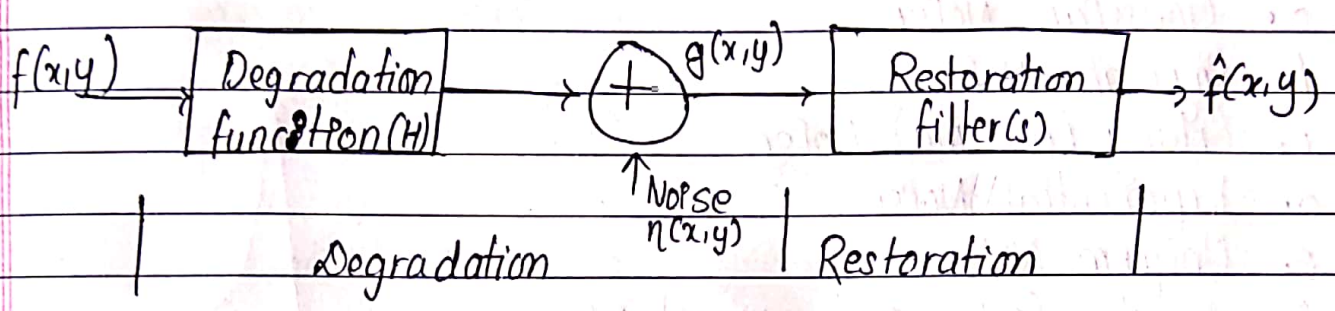
→ Restoration techniques are oriented towards modeling the degradation and applying the inverse process in order to recover the original image.

Differences between Image Restoration and Image Enhancement

- Image restoration differs from image enhancement in that the latter is concerned more with accentuation or extraction of image features rather than restoration of degradations.
- Image restoration problems can be quantified precisely, whereas enhancement criteria are difficult to represent mathematically.
- Eg of enhancement technique :- Contrast stretching
- Eg of image degradation :- image blur.

Scanned with CamScanner

Model of Image Degradation / Restoration Process



→ Image Degradation process for an image uses a degradation function together with an additive noise term.

→ If H is a liner, position-invariant process, then the degraded image is given in the spatial domain by:

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

degradation function | convolution | input image | noise term

In frequency domain,

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

* Noise in Image Degradation

All natural images when displayed have gone through some sort of degradation during display mode, acquisition mode or processing mode. The degradation may be due to sensor noise, blur due to camera misfocus, relative object-camera motion, etc.

* Image noises

Image noise is random variation of brightness or color information in images, and is usually an aspect of electronic noise.

Scanned with CamScanner

Types of Image Noises are as follows:-

a. Gaussian Noise
b. Rayleigh Noise
c. Erlang (Gamma) Noise
d. Exponential Noise
e. Uniform Noise
f. Impulse (salf-and pepper) Noise

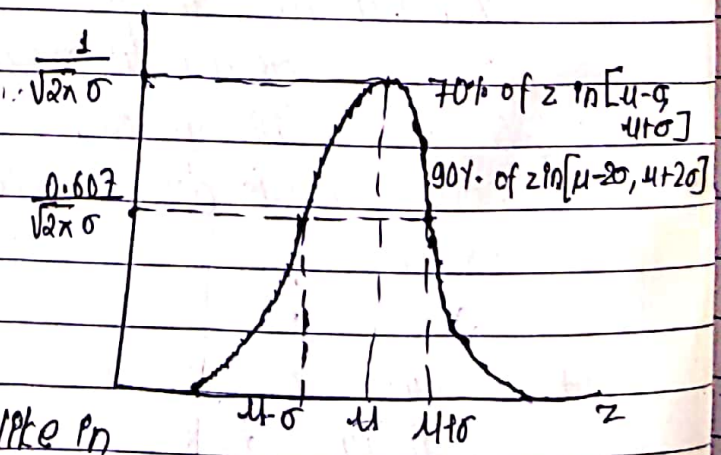
- Gaussian Noise

It is also called as electronic noise because it arises in amplifiers or detectors. Gaussian noise is caused by natural sources such as thermal vibration of atoms and discrete nature of radiation of warm objects. Gaussian noise generally disturbs the gray values in digital images. That is why Gaussian noise model essentially designed and characteristics by its PDF or normalized histogram with respect to gray value. This is given as:

Probability density function (PDF)

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

z: gray level
μ: mean of average value of z
σ: standard deviation of z
σ^2: variance of z



→ Due to this equal randomness the normalized Gaussian noise curve looks like in bell shaped.

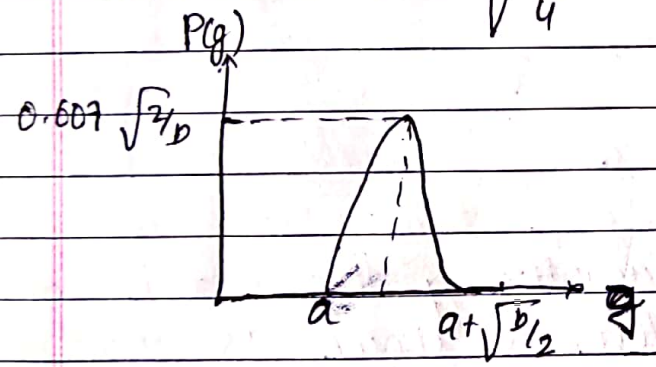
Scanned with CamScanner

- Rayleigh noise

Rayleigh noise presents in radar range images. In rayleigh noise, probability density function is given as:-

$$P(g) = \begin{cases} \frac{2}{b}(g-a)e^{-\frac{(g-a)^2}{b}} & \text{for } g \geq a \\ 0 & \text{for } g < a \end{cases}$$

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



- Erlang Noise

Erlang probability distribution is a two parameter distribution with support $x \in (0, \infty)$. The two parameters are :-

- A positive integer representing 'shape' k.
- A positive real representing 'rate' λ.

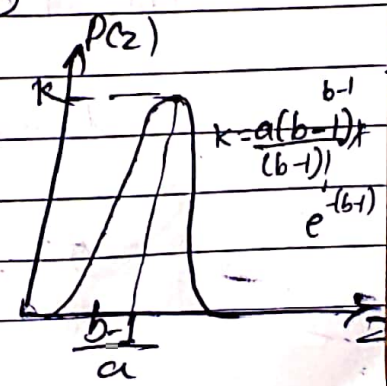
The PDF is given as:

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

$$= 0 \quad \text{, otherwise.}$$

where $a > 0$, b is positive integer.

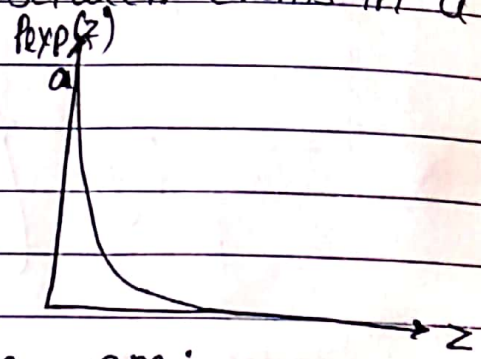
$\mu = b/a$, $\sigma^2 = \frac{b}{a^2}$



Scanned with CamScanner

- Exponential noise

The exponential distribution is the probability distribution that describes the time relation between events in a Poisson process.



The PDF is given as:

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

where $a > 0$. The mean & variance are:

$$\mu = 1/a \qquad \sigma^2 = \frac{1}{a^2}$$

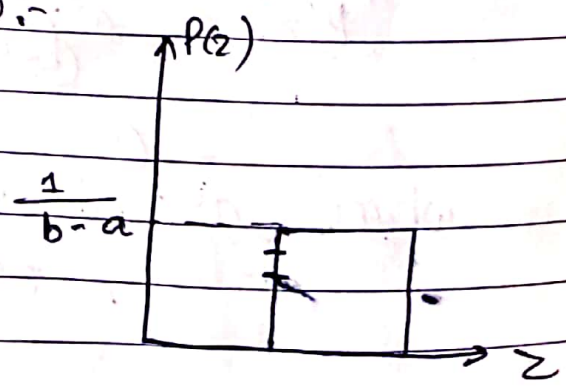
- Uniform noise

The noise caused by quantizing the pixels of image to a number of distinct level is known as quantization noise. It has distribution approximated to uniform distribution. In the uniform noise, the noise are uniformly distributed across a specified range.

The PDF is given as:

$$P(z) = \frac{1}{(b-a)} \quad \text{for } a \leq z \leq b$$
$$= 0 \quad \text{for otherwise}$$

The mean and variance are:-



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

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

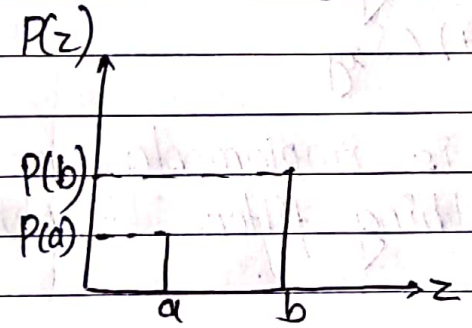
Scanned with CamScanner

- Impulse Noise

It is also known as salt and pepper noise. This is caused due to sharp & sudden disturbances in the image gray values. Its appearance is randomly scattered white or black pixel over the image.

The PDF is given as:-

$$P(z) = P_a \quad \text{for } z = a$$
$$= P_b \quad \text{for } z = b$$
$$= 0 \quad \text{otherwise}$$



Restoration Filters

- Image restoration in presence of noise only

Spatial filtering, which is used for image smoothening and sharpening, can also be used to remove noise.

Spatial filters for Image Restoration

- Mean filters
- Order statistic filters

Scanned with CamScanner

Mean filters

We can use spatial filters of different kinds to remove different kinds of noise.

→ Arithmetic mean filters

The arithmetic mean filter is a very simple one & is calculated as follows :-

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

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

This is implemented as the simple smoothing filter. It blurs the image.

→ Geometric mean

It is calculated as follows :-

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

It achieves similar smoothing to the arithmetic mean, but tends to lose less image detail.

→ Harmonic mean

It is calculated as follows :-

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

It works well for salt noise, but fails for pepper noise.

Scanned with CamScanner

And it also does well for other kinds of noise such as Gaussian noise.

→ Contraharmonic mean

It is calculated as follows:-

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

; when $Q=0 \rightarrow$ arithmetic
$Q=1 \rightarrow$ harmonic

Q is the order of the filter.
Positive values of Q eliminate pepper noise.
Negative values of Q eliminate salt noise.
It cannot eliminate both simultaneously.

Usage :-

- Arithmetic and geometric mean filters are suited for Gaussian or uniform noise.
- Contraharmonic filters are suited for impulse noise.

Order Statistics Filters

1) Median filter

→ It is an example of non-linear filter.
→ It works by sorting the list and finding the median. The center pixel is then replaced by the median value.
→ It is effective in the presence of both bipolar and unipolar impulse noise.

Scanned with CamScanner

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

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

e) Maximum filter

→ It selects the largest value in the sorted list.

→ It is used for removing pepper noise.

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

3) Minimum filter

→ It selects the minimum value in the sorted list.

→ It is used for removing salt noise.

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

4) Midpoint filter

→ This filter selects the midpoint, which is the average of the minimum and maximum values.

→ It is very effective in removing Gaussian noise and uniform noise,

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

5) Alpha-trimmed mean filter

→ It is based on the concept of computation of the average of the pixels that falls within the window.

Scanned with CamScanner

→ It works by deleting the d/2 lowest and the d/2 highest grey level values.

→ It is useful in situations involving multiple types of noise such as a combination of salt and pepper and Gaussian noise.

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

If $d=0 \rightarrow$ arithmetic mean filter

$d = mn-1 \rightarrow$ median filter

6) Trimmed mean filter

→ It works by removing the max and min values and calculating the average of the remaining values.

Band pass and Band Reject Filters

→ Let only a portion of the frequency components pass. They can be constructed by

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

Bandpass filters are widely used in wireless transmitters and receivers. The main function of such a filter in a transmitter is to limit the bandwidth of the output signal to the band allocated for the transmission. This prevents the transmitter from interfering with other stations. In a receiver, a bandpass filter allows signals with an a selected range of frequencies to be heard or decoded, while preventing signals at unwanted frequencies from getting through. A bandpass filter also optimizes the signal-to-noise ratio and sensitivity of a receiver.

Scanned with CamScanner

1) Ideal filter

In this type of filter, frequencies outside of the given range are passed without attenuation and frequencies inside of the given range are blocked. This behaviour ideal filter very sharp.

The centered FFT (Fast fourier Transform) is filtered by the following function, where D_L is the lower bound of the frequency band, D_H is the upper bound of the frequency band, and $D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle:

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

2) Butterworth

Using butterworth band reject filter, frequencies at the center of the frequency band are completely blocked and frequencies at the edge of the band are attenuated by a fraction of the maximum value. The butterworth filter does not have any sharp discontinuities between passed and filtered frequencies.

Note → The default for Bandreject-filter is BUTTERWORTH = 1.

The centered FFT is filtered by one of the following functions, where D_0 is the center of the frequency band, W is the width of the frequency band, $D = D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle, & n is the dimension of the Butterworth filter.

Scanned with CamScanner

low reject $(D_L = 0, D_H < 1)$: $H(u,v) = 1 - \frac{1}{1 + (D/D_H)^{2n}}$

Band reject $(D_L > 0, D_H < 1)$: $H(u,v) = \frac{1}{1 + [(DW)/(D^2 - D_0^2)]^{2n}}$

High reject $(D_L > 0, D_H = 1)$: $H(u,v) = 1 - \frac{1}{1 + \left[\frac{D_L}{D}\right]^{2n}}$

3) Gaussian

In this type of filter, the transition between unfiltered and filtered frequencies is very smooth.

The centered FFT is filtered by one of the following functions, where D_0 is the center of the frequency band, W is the width of the frequency band, and $D = D(u,v)$ is the distance between a point (u,v) in the frequency domain & the center of the frequency rectangle.

Low-reject $(D_L = 0, D_H < 1)$: $H(u,v) = 1 - e^{-D^2/(2D_H^2)}$

Band-reject $(D_L > 0, D_H < 1)$: $H(u,v) = e^{-[(D^2 - D_0^2)/(DW)]^2}$

High-reject $(D_L > 0, D_H = 1)$: $H(u,v) = 1 - e^{-D^2/(2D_L^2)}$

→ Bandreject filter

The bandreject filter function applies a low-reject, high-reject or band-reject filter on a one channel image.

A band reject filter is useful when the general location of the noise in the frequency domain is known. A band reject filter blocks frequencies within the chosen range and lets frequencies outside of the range pass through.

Scanned with CamScanner

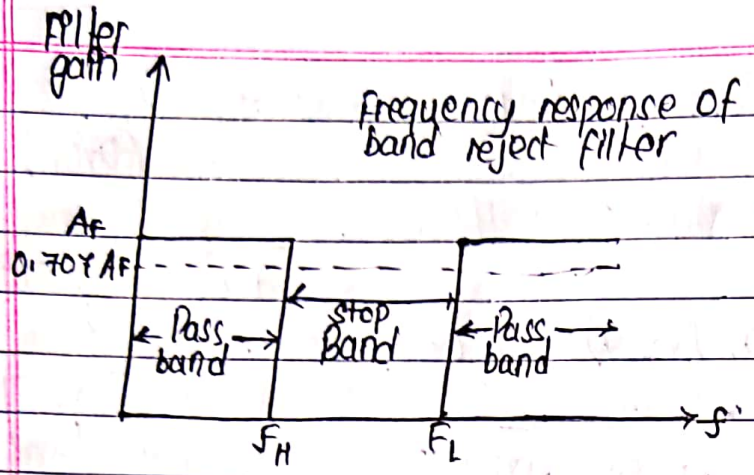


Image Compression

Image compression is the process of encoding or converting an image file in such a way that it consumes less space than the original file. It is a type of compression technique that reduces the size of an image file without affecting or degrading its quality to a greater extent.

- Compression factors
  - → Transmission and storing of large images.
  - → Reduce of baud rate
  - → Reduce of transmission time

Compression Ratio

Compression ratio is defined as number of bits to represent the size of original image to the number of bits to represent the size of compressed image. Compression ratio shows how much time the image has been compressed.

Data Redundancy

Data that provide no relevant information is called redundant data or redundancy. Image compression techniques can be designed by reducing or eliminating the data

Scanned with CamScanner

redundancy. Image coding or compression has a goal to reduce the amount of data by reducing the amount of redundancy.

- Coding Redundancy Basic concept

A natural m-bit coding method assigns m-bit to each graylevel without considering the probability that gray level occurs with. It is very likely to contain coding redundancy.

→ Basic concept: Utilize the probability of occurrance of each gray level (histogram) to determine length of code representing that particular gray level: variable length coding. Assign shorter code words to the gray levels that occur most frequently or vice versa.

- Interpixel Redundancy

It is caused by high interpixel correlations within an image, i.e. gray level of any given pixel can be reasonably predicted from the value of its neighbours (information carried by individual pixels is relatively small).

To reduce interpixel redundancy, mapping is used. The mapping scheme can be selected according to the properties of redundancy.

An example of mapping can be to map pixels of an image: $f(x,y)$ to a sequence of pairs: $(g_1, r_1), (g_2, r_2), \ldots (g_i, r_i)$, where g_i is i^{th} gray level and r_i is run length of the ith run.

- Psychovisual Redundancy

The eye does not respond with equal sensitivity to all

visual information. Certain information has less relative importance than other information in normal visual processing psychovisually redundant (which can be eliminated without significantly impairing the quality of image perception). The elimination of psychovisually redundant data results in a loss of quantitative information lossy data compression method. Image compression methods based on the elimination of psychovisually redundant data (usually called quantization) are usually applied to commercial broadcast TV and similar applications for human visualization.

→ Eg: Improved Grey Scale Quantization.

Huffman Coding Algorithm for Image Compression

Step 1: Build a Huffman tree by sorting the histogram & successively combine the two bins of the lowest value ~~until~~ until only one bin remains.

Step 2: Encode the Huffman tree and save the Huffman tree with the coded value.

Step 3: Encode the residual image.

Example :-

Say → Image size : 10X10 (5 bit image)

Frequency : $a_2 = 40$, $a_6 = 30$, $a_1 = 10$, $a_4 = 10$, $a_3 = 6$, $a_5 = 4$

$$P(a_2) = 40/100 = 0.4$$

Scanned with CamScanner

Symbols (Pixe Intensity levels)	Probabilities (Sorted)	Source Reduction (do till two values are left) (maintain in sorted order as well)			
		1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

(Add 1st & last 2 and update on last of next step and again ~~order last 2 & update on and last of next step simultaneously~~) sort them)

Symbols	Probabilities	Source Reduction			
		1	2	3	4
a_2	0.4 1	0.4	0.4	0.4	0.6 $\boxed{0}$
a_6	0.3 00	0.3	0.3	0.3 00	0.4 $\boxed{1}$
a_1	0.1 011	0.1	0.2 010	0.3 01	
a_4	0.1 0100	0.1 0100	0.1 011		
a_3	0.06 01010	0.1 0101			
a_5	0.04 01011				

Encoded String : 010100111100

Decoding : a_3 a_1 a_2 a_2 a_6

→ Parameters

1. Average length of code

$L_{avg} = 0.4*1 + 0.3*2 + 0.1\times3 + 0.1\times4 + 0.06\times5 + 0.04\times5$
$= 2.2$ bits/symbol

2. Total number of bits to be transmitted

$10 * 10 * L_{avg}$
$= 10 * 10 * 2.2$
$= 220$ bits

3. Entropy
= Sum of (-Probability * log prob)
= 2.13.96

4. How much saved = $\frac{10*10*5 - 10*10*2.2}{10*10*5}$ (5 bits req in question so 10x10x5)
= 0.56
= 56%

Run length Coding

It is the simplest data compression technique. In this technique, runs of data are stored as a single data value and count ~~than~~ the original run.

Run is the sequence of same data or value.
→ It is loseless compression and is reversible.

Drawback
→ If there is no repeating values in the data RLC doubles

Scanned with CamScanner

the amount of number compared with the original data.

$$\underset{\text{no. of bits}=3}{xyz} \xrightarrow{RLC} \underset{\text{no. of bits}=6}{x1y1z1}$$

Eg: If input string is 'WWWWAAADEXXXXXX' then the run length encoding is W4A3D1E1X6.

4-bit Improved Gray Scale Coding : IGS Coding Scheme

Step 1 : Lower 4 bits of the proceeding modified pixel are added to the present pixel.
Step 2 : The new 4 MSB's of the present pixel are taken as the I.G.S code.
Step 3 : Repeat above steps after moving on to a new pixel.
Step 4 : If MSB is 1111, then add 0000 instead of the 4 LSB's of the previous sum.

Example: Construct the I.G.S code of the given grey level data set {100, 110, 124, 124, 130, 110, 200, 210}

Pixel	Gray level (Binary)	Sum (000 000)	IGS code
100	0110 0100	0010 0100	0110
110	0110 1110	0111 0010	0111
124	0111 1100	0111 1110	0111
124	0111 1100	1000 1010	1000
130	1000 0010	1000 1100	1000
200	1100 1000	1101 0100	1101
210	1101 0010	1101 0110	1101

IGS

Bit plane

A bit plane of a digital discrete signal (such as image or sound) is a set of bits corresponding to a given bit position in each of the binary numbers representing the signal. For example, for 16-bit data representation there are 16 bit plane the first bit plane contains the set of the most significant bit, and the 16th bit contains the least significant bit.

It is possible to see that the 1st bit plane gives the roughest but the most critical approximation of values of a medium, and the higher the number of the bit plane the less is it's contribution to the final stage. Thus, adding a bit plane gives a better approximation.

→ Bit plane slicing

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

Since the given image has maximum gray level of 7, it is a 3-bit image:

110	111	110	110	111
000	000	000	001	010
001	001	001	010	011
100	101	101	100	010
110	110	110	111	111

→ MSB Plane

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

→ Center bit plane

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

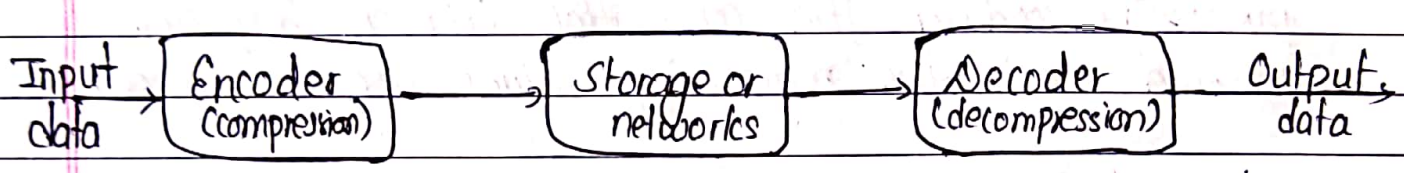
→ LSB plane

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

Scanned with CamScanner

Image Compression Models

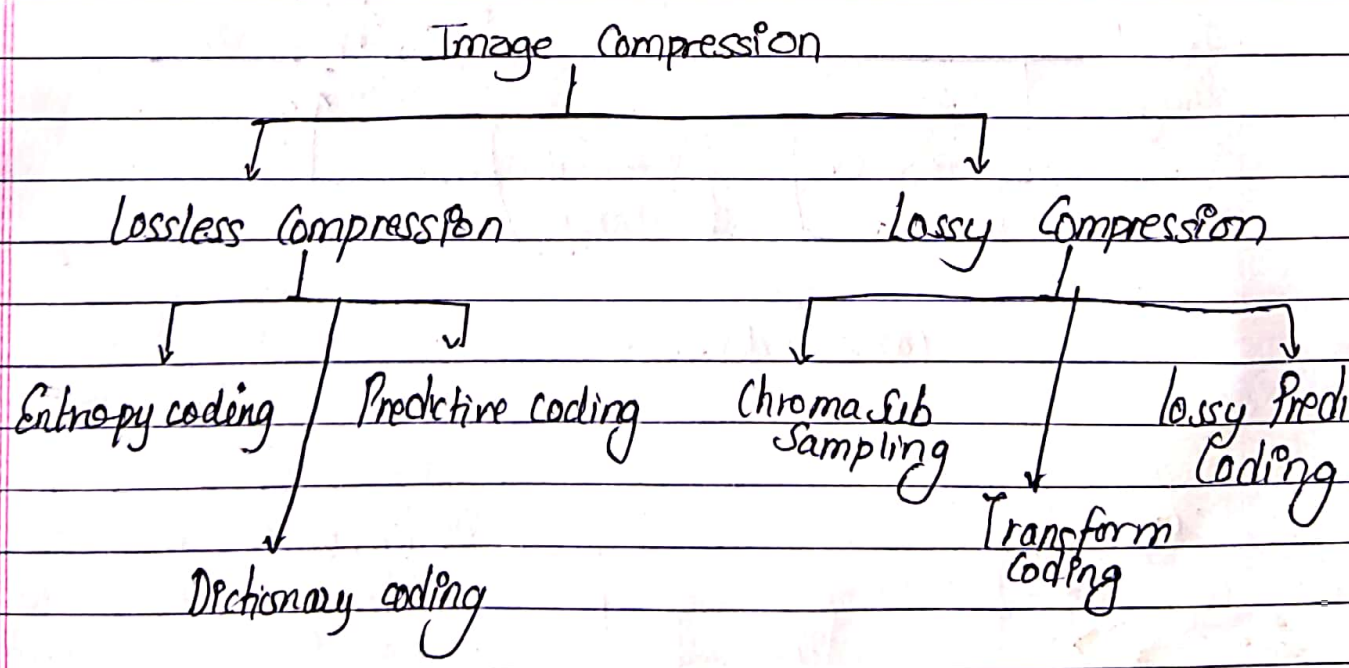
Compression, The process of coding that will effectively reduce the total number of bits needed to represent certain information.



A general data compression scheme.

Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission.

- Lossy compression
- lossless compression



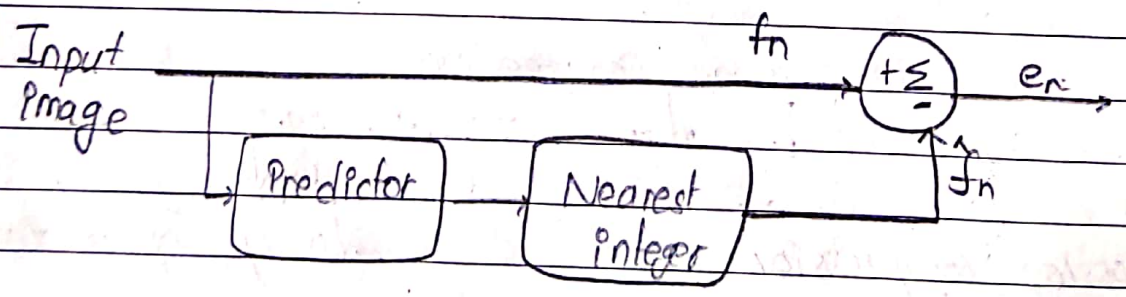
→ Prediction coding

For most images, there are redundancies among adjacent pixel values; that is, adjacent pixels are highly correlated. Thus, a current pixel can be predicted reasonably well based

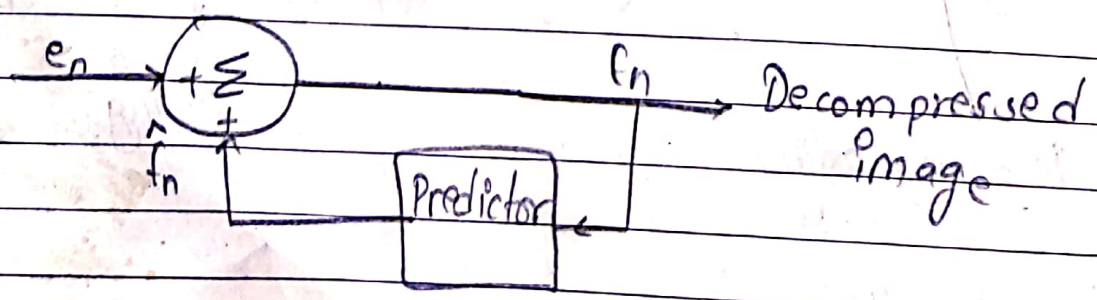
on a neighborhood of pixels. The prediction error, which is obtained by subtracting the prediction from the original pixel, has smaller entropy than the original pixels. Hence, the prediction error can be encoded with fewer bits. In predictive coding, the correlation between adjacent pixels is removed and the remaining values are encoded.

- Lossless Predictive Coding

This approach is based on eliminating the interpixel redundancies of closely spaced pixels by extracting & coding only the new information in each pixel. The new information of a pixel is defined as the difference between the actual & predicted value of that pixel.



(a) Encoder



(b) Decoder

The figure shows the basic components of a lossless predictive coding system. The system consists of an encoder and decoder, each containing an identical predictor. As each successive pixel of input image in the encoder, the predictor generates the anticipated value of that pixel based on some past inputs.

The output of the predictor is then rounded to the nearest integer, denoted $\hat{f}_n$ and is used to form the prediction error (coded with variable-length code) to generate next element of compressed data stream.

- Lossy predictive coding

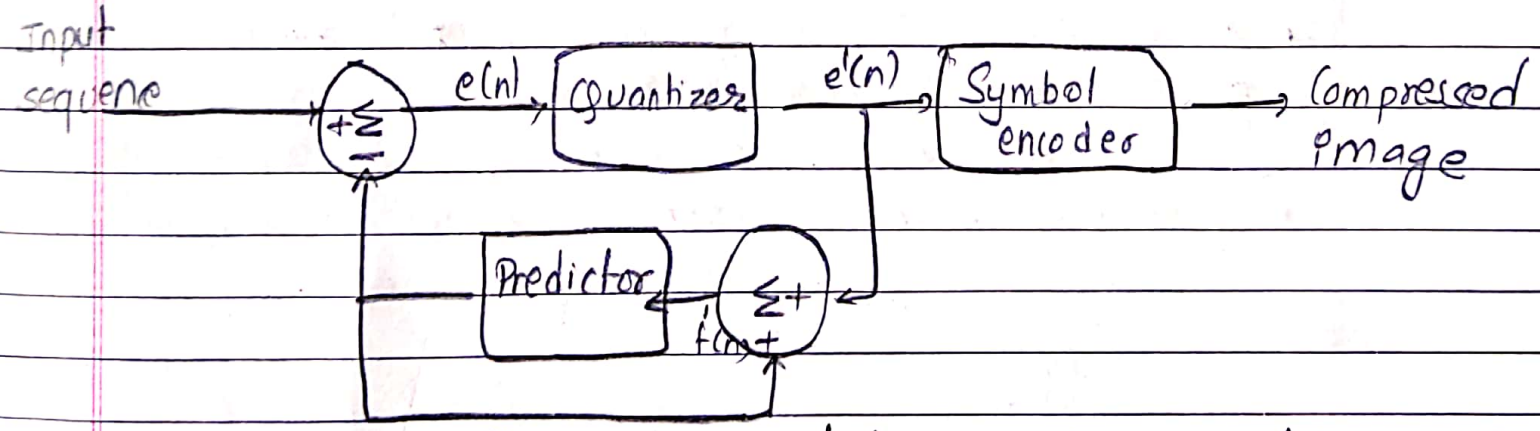


Fig: lossy predictive coding model, encoder.

Quantizer, which replaces the nearest integer function of the error-free encoder, is inserted between the symbol encoder & the point at which the prediction error is formed.

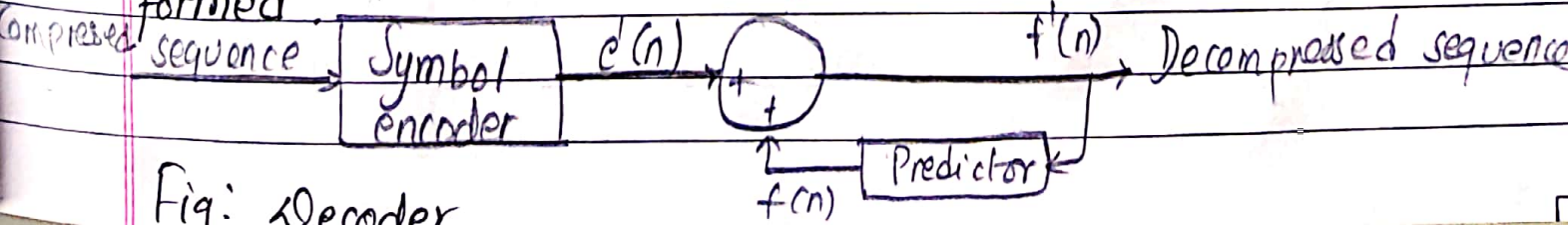


Fig: Decoder

It maps prediction error into a limited range of outputs $e'(n)$ which establishes the amount of compression and distortion associated with lossy predictive coding.

Here,

$$f'(n) = e'(n) + \hat{f}(n)$$

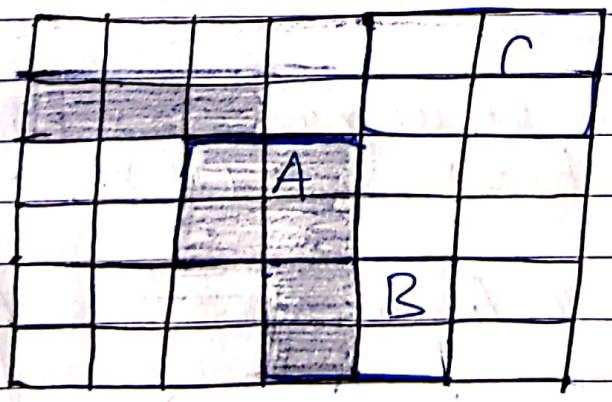
Scanned with CamScanner

Date
Page

Unit : 5
Introduction to Morphological Image Processing

- Introduction

→ Morphological operations apply a structuring element to an input image creating an output image of the same size.
→ The value of each pixel in output image is based on comparision of corresponding pixel in input image with it's neighbors.
→ Morphological operations rely only on relative ordering of pixel values, so it is best suited for binary images.
→ Morphological techniques probe an image with a small shape or template called structuring element.
→ The structuring element is positioned at all possible location in the image & compared with the corresponding neighborhood of pixels.
→ When comparing we check if it fits or hits.



A → fit
B → hit
C → neither hit nor fit.

→ Applications : Boundaries extraction, skeletons, convex hull, morphological filtering, thinning, pruning

Scanned with CamScanner

- **Structuring Element (SE)**

The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one.

→ The matrix dimensions specify the size of the structuring element.

→ The patterns of ones and zeros specifies the shape of SE.

→ An origin of SE is usually one of it's pixels, although generally the origin can be outside the SE.

- **The Hit - or Miss - Transformation**

→ A basic tool for shape detection.

→ Let the origin of each shape be located at its center of gravity.

→ If we want to find the location of a shape (let x) at larger image (let A).

- Let x be enclosed by small window 'W'.
- The local ~~differen~~ background of x w.r.t W is defined as the set difference $(W-x)$.
- Then, apply erosion operation of A by x.
- If any SE matches then a match (hit) is found in A.

Formally,

If B denotes the set composed of x & its background -

$B = (B_1, B_2)$; $B_1 = X$, $B_2 = (W-X)$.

The match of B in A, ~~det~~ denoted as $A \circledast B$ is :-

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

Scanned with CamScanner

• Logic Operations involving Binary Pixels and Images

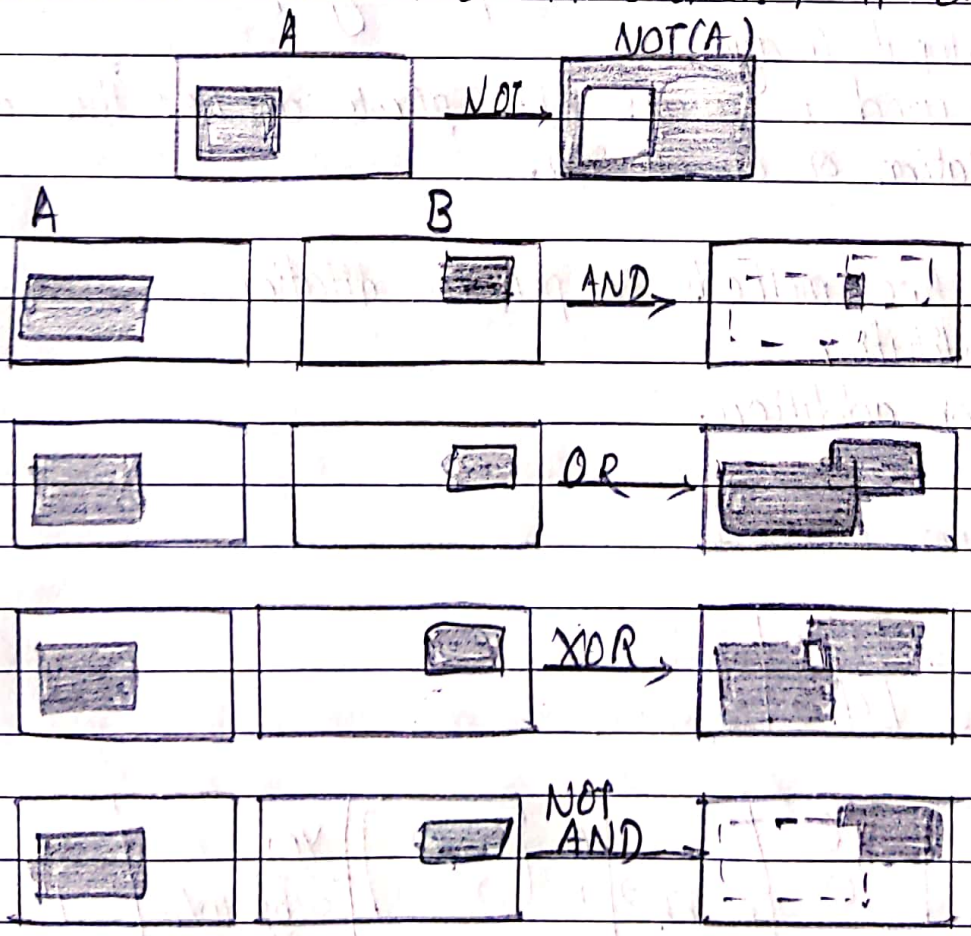
→ The principal logic operations ~~using~~ used in image processing are : AND, OR, NOT(Complement)

→ These operations are functionally complete.

→ Logic operations are performed on a pixel by pixel basis between corresponding pixels (bitwise).

→ Other important logic operations are:-
XOR, NAND.

→ Logic operations are just a private case for a binary set operations, such as : AND - Intersection, OR - union, NOT - complement



Scanned with CamScanner

Types of Morphological Operations

- Morphological Dilation and Erosion

→ Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

→ The number of pixels added or removed from the object in an image depends on the shape & size of SE used to process the image.

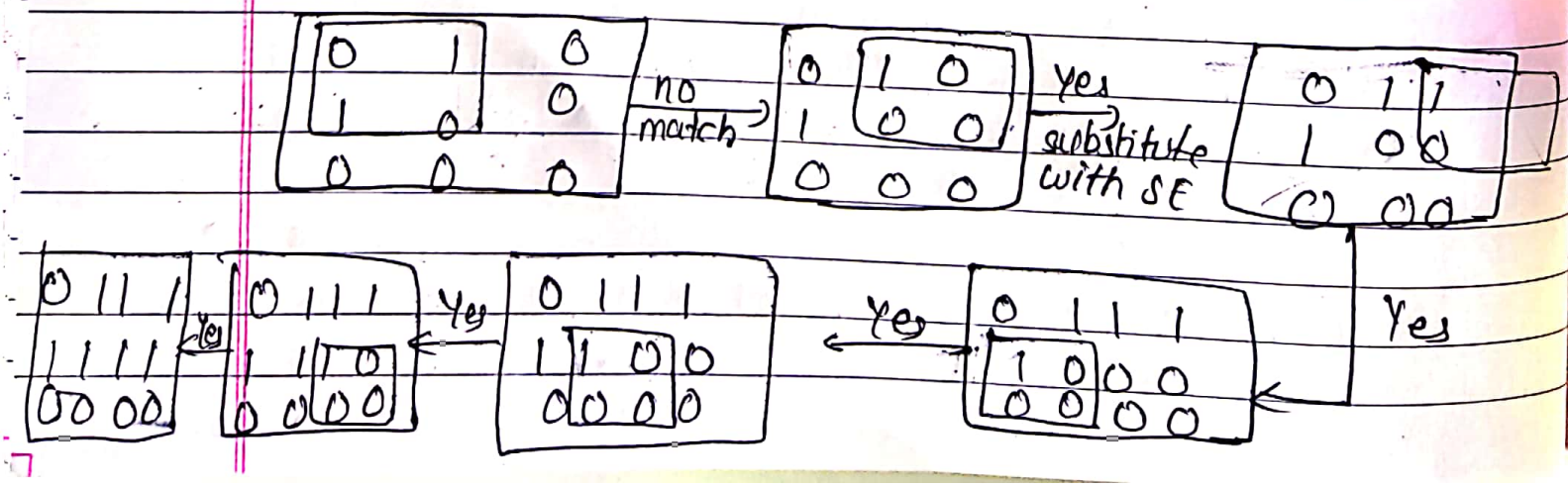
→ In the morphological dilation & erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image.

→ The rule used to process the pixels defines the operation as a dilation or an erosion.

There are two methods to perform dilation
1. Substitution
2. Vector addition.

Example. Substitution method

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



Date
Page

Vector addition

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Position of 1's:

$A = \{(0,1), (1,0)\}$

$B = \{(0,0)\ (0,1)\}$

$(0,0) + (0,1) \longrightarrow (0,1)$

$(0,0) + (1,0) \longrightarrow (1,0)$

$(0,1) + (0,1) \longrightarrow (0,2)$

$(0,1) + (1,0) \longrightarrow (1,1)$

$A \oplus B = \{(0,1), (1,0), (0,2), (1,1)\}$

Characteristics of Dilation

1. Dilation add pixel to the boundary of an object in an image.
2. Number of pixel added depend on shape and size of structuring element.
3. Connects area that are separated.
4. In gray scale image dilation increases the brightness of the object.

Scanned with CamScanner

EROSION Vs. DILATION.

- Erosion

1. Shrinks or thins objects in binary images.
2. Remove image components.
3. Erode away the boundaries of regions of foreground pixels
4. Areas of foreground pixels shrink in size, & holes within those areas become larger.

- Dilation

1. Grows or thickens object in a binary image
2. Bridging gaps
3. Fill small holes of sufficiently small size.

Other operations are :→ Opening
→ Closing

- Opening

The opening of an image f by structuring element s (denoted by to) is an erosion followed by a dilation

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

It is called so because it can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation.

- Closing

The closing of an image f by a structuring elements (denoted by $f \bullet s$) is a dilation followed by an erosion.

Scanned with CamScanner

$$f \bullet s = (f \oplus s_{rot}) \ominus s_{rot}$$

It is called so because it can fill holes in the regions while keeping the initial region sizes.

Basic morphological Algorithms (Applications)

1. Boundary Extraction

First erode A by B, then make set difference between A and the erosion.

The thickness of the contour depends on the size of constructing object (B).

$$\beta(A) = A - (A \ominus B)$$

2. Region Filling

- This algorithm is based on set of dilations, complementations and intersections.
- p is the point inside the boundary (given), with the value of 1.

$$x_k = (x_{k-1} \oplus B) \cap A^c \quad , k = 1, 2, 3, \ldots$$

- The process stops when $x_k = x_{k-1}$
- The result that given by union of A and $X_{(k)}$, is a set contains the filled set and the boundary.

$$X_k = ((X_{k-1}) \oplus B) \cap A^c$$

3. Extraction of Connected Components.

- This algorithm extracts a component by selecting a point on a binary object A.

$$X_k = (X_{k-1} \oplus B) \cap A$$

Scanned with CamScanner

Date
Page

4. Convex Hull

A is said to be convex if a straight line segment joining any two points in A lies entirely within A. The convex hull H of set S is the smallest convex set containing S.

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1,2,3,4 \text{ and } k = 1,2,3,\ldots$$

Let $D^i = X_k^i$. The convex hull is:

$$C(A) = \bigcup_{i=1}^{4} D^i$$

5. Thinning

The thinning of a set A by a structuring element B can be defined by terms of the hit-and-miss transform:

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c.$$

6. Thickening

Thickening is a morphological dual of thinning. Thickening is defined as:

$$A \odot B = A \cup (A \circledast B)$$

7. Skeleton

The skeleton of A is defined by terms of erosions & openings:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

where $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$; B is the S.E. (structuring element)

Scanned with CamScanner