ARULMIGU PALANIANDAVR ARTS COLLEGE FOR WOMEN

(Autonomous)

(Re-Accredited with 'B++' Grade by NAAC 3rd Cycle)

Run by Arulmigu Dhandayuthapani Swamy Thirukoil, H.R & C.E Dept. Government of Tamil Nadu

A Government Aided College - Affiliated to Mother Teresa Women's University, Kodaikanal

CHINNAKALAYAMPUTHUR(PO), PALANI - 624615

DEPARTMENT OF COMPUTER SCIENCE

AND

COMPUTER APPLICATION

LEARNING RESOURCE

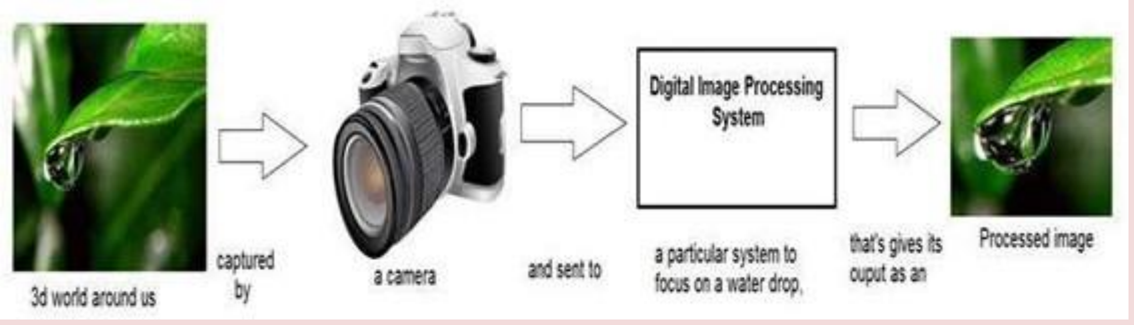
DIGITAL IMAGE FUNDAMENTALS & IMAGE TRANSFORMS

DIGITAL IMAGE FUNDAMENTALS:

The field of digital image processing refers to processing digital images by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels and pixels. Pixel is the term used most widely to denote the elements of digital image.

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

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 of the image at that point.



The term ***gray level*** is used often to refer to the intensity of monochrome images. Color images are formed by a combination of individual 2-D images.

For example: The RGB color system, a color image consists of three (red, green and blue) individual component images. For this reason many of the techniques developed for monochrome images can be extended to color images by processing the three component images individually.

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

APPLICATIONS OF DIGITAL IMAGE PROCESSING

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

Digital image processing has a broad spectrum of applications, such as

- Remote sensing via satellites and other spacecrafts
- Image transmission and storage for business applications
- Medical processing,
- RADAR (Radio Detection and Ranging)
- SONAR(Sound Navigation and Ranging) and
- Acoustic image processing (The study of underwater sound is known as underwater acoustics or hydro acoustics.)
- Robotics and automated inspection of industrial parts.

Images acquired by satellites are useful in tracking of

- Earth resources;
- Geographical mapping;
- Prediction of agricultural crops,
- Urban growth and weather monitoring
- Flood and fire control and many other environmental applications.

Space image applications include:

- Recognition and analysis of objects contained in images obtained from deep space-probe missions.
- Image transmission and storage applications occur in broadcast television
- Teleconferencing
- Transmission of facsimile images(Printed documents and graphics) for office automation

Communication over computer networks

- Closed-circuit television based security monitoring systems and
- In military communications.

Medical applications:

- Processing of chest X- rays
- Cineangiograms
- Projection images of transaxial tomography and
- Medical images that occur in radiology nuclear magnetic resonance(NMR)
- Ultrasonic scanning

IMAGE PROCESSING TOOLBOX (IPT) is a collection of functions that extend the capability of the MATLAB numeric computing environment. These functions, and the expressiveness of the MATLAB language, make many image-processing operations easy to write in a compact, clear manner, thus providing a ideal software prototyping environment for the solution of image processing problem.

Components of Image processing System:

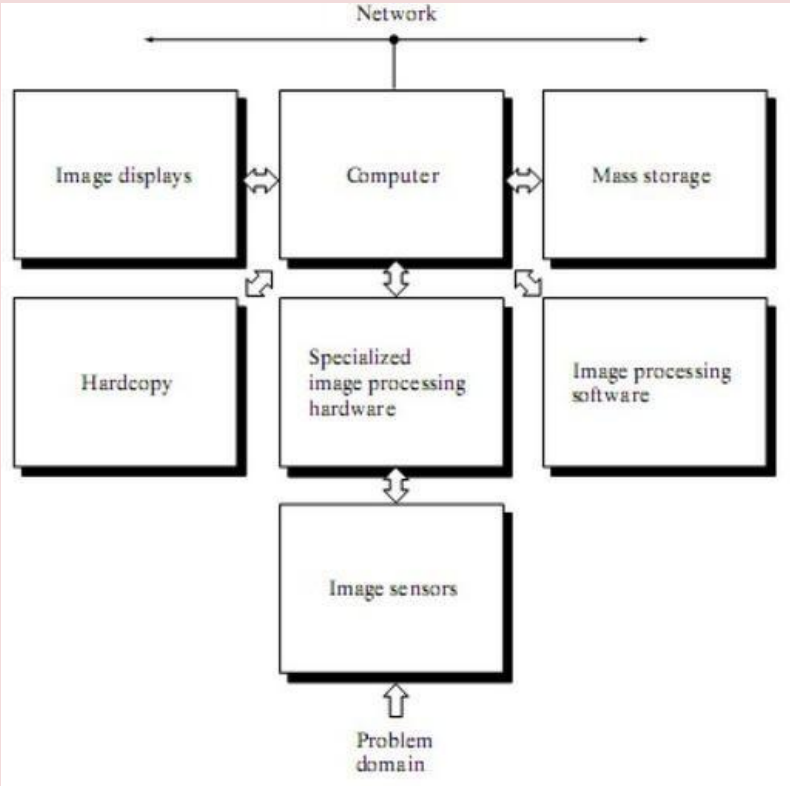


Figure : Components of Image processing System

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

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

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

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

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

i) Short term storage for use during processing

ii) On line storage for relatively fast retrieval

iii) Archival storage such as magnetic tapes and disks

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

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

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

Fundamental Steps in Digital Image Processing:

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

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

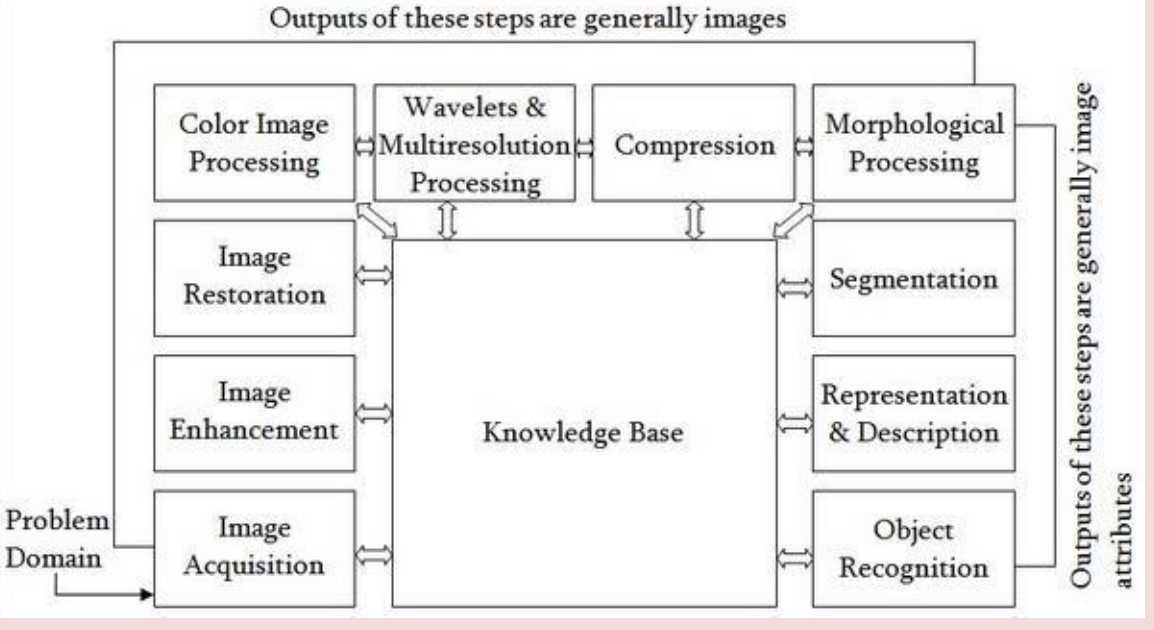


Fig: Fundamental Steps in Digital Image Processing

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

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

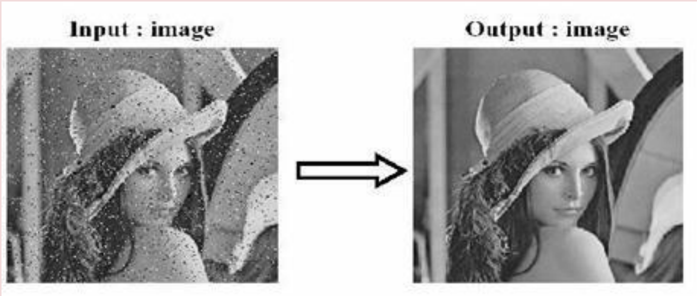
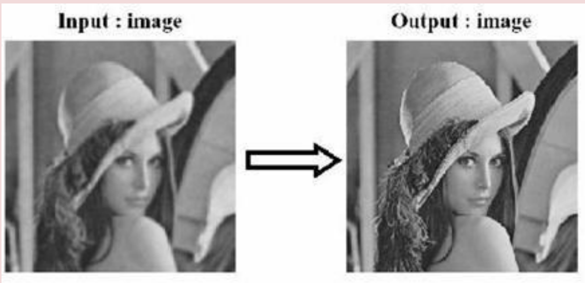


Image Restoration: It deals with improving the appearance of an image. It is an objective approach, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding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



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

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

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

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

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

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

Knowledge base:

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

A Simple Image Model:

An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be nonzero and finite; that is o<f(x,y) <co The function f(x,y) may be characterized by two components- The amount of the source illumination incident on the scene being viewed.

(a) The amount of the source illumination reflected back by the objects in the scene These are called illumination and reflectance components and are denoted by i(x,y) an r (x,y) respectively.

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

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

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

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

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

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

SAMPLING AND QUANTIZATION:

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

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

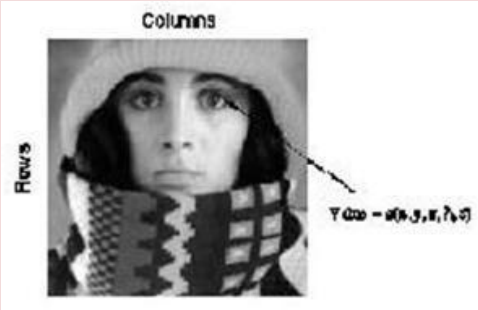
In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So we divide the gray level scale into eight discrete levels ranging from eight level values. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment it made depending on the vertical proximity of a simple to a vertical tick mark.

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

Digital Image definition:

A digital image f(m,n) described in a 2D discrete space is derived from an analog image f(x,y) in a 2D continuous space through a sampling process that is frequently referred to as digitization. The mathematics of that sampling process will be described in subsequent Chapters. For now we will look at some basic definitions associated with the digital image. The effect of digitization is shown in figure.

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



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

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

2) Mid level image processing - it involves tasks like segmentation, description of those objects to reduce them to a form suitable for computer processing, and classification of individual objects. The inputs to the process are generally images but outputs are attributes extracted from images.

3) High level processing – It involves "making sense" of an ensemble of recognized objects, as in image analysis, and performing the cognitive functions normally associated with vision.

Representing Digital Images:

The result of sampling and quantization is matrix of real numbers. Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N Columns. The values of the coordinates (x,y) now become discrete quantities thus the value of the coordinates at orgin become 9X,y) =(o,o) The next Coordinates value along the first signify the iamge along the first row. it does not mean that these are the actual values of physical coordinates when the image was sampled.

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

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

$$L=2^k$$

Then, the number, b, of bites required to store a digital image is B=M *N* k When M=N, the equation become $b=N^2*k$

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

Spatial and Gray level resolution:

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

Gray levels resolution refers to smallest discernible change in gray levels. Measuring discernible change in gray levels is a highly subjective process reducing the number of bits R while repairing the spatial resolution constant creates the problem of false contouring.

It is caused by the use of an insufficient number of gray levels on the smooth areas of the digital image . It is called so because the rides resemble top graphics contours in a map. It is generally quite visible in image displayed using 16 or less uniformly spaced gray levels.

Image sensing and Acquisition:

The types of images in which we are interested are generated by the combination of an "illumination" source and the reflection or absorption of energy from that source by the elements of the "scene" being imaged. We enclose *illumination* and *scene* in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene. For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern. Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock formations, or a human brain. We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light reflected from a planar surface. An example in the second category is when X-rays pass through a patient‟s body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response. In this section, we look at the principal modalities for image sensing and generation.

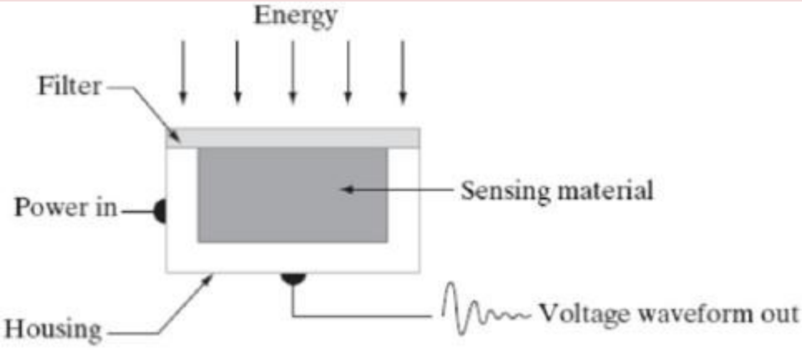


Fig:Single Image sensor

Fig: Line Sensor

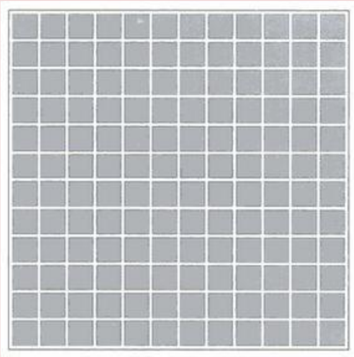
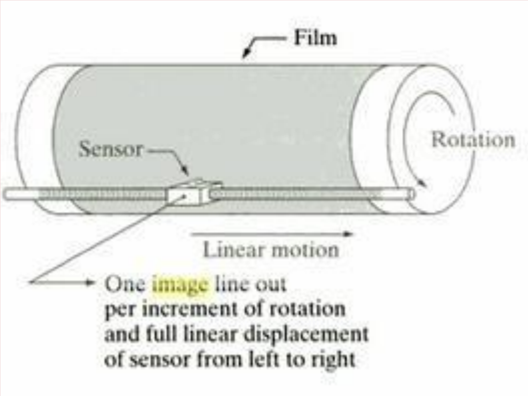


Fig: Array sensor

Image Acquisition using a Single sensor:

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



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

Image Acquisition using a Sensor strips:

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

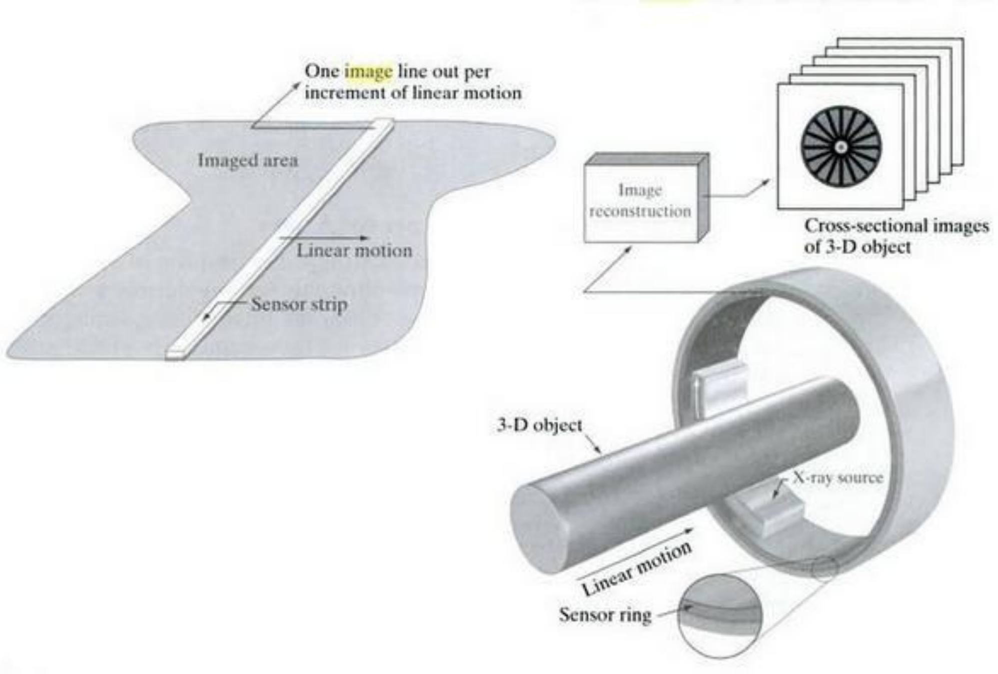


Fig: Image Acquisition using linear strip and circular strips.

Image Acquisition using a Sensor Arrays:

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

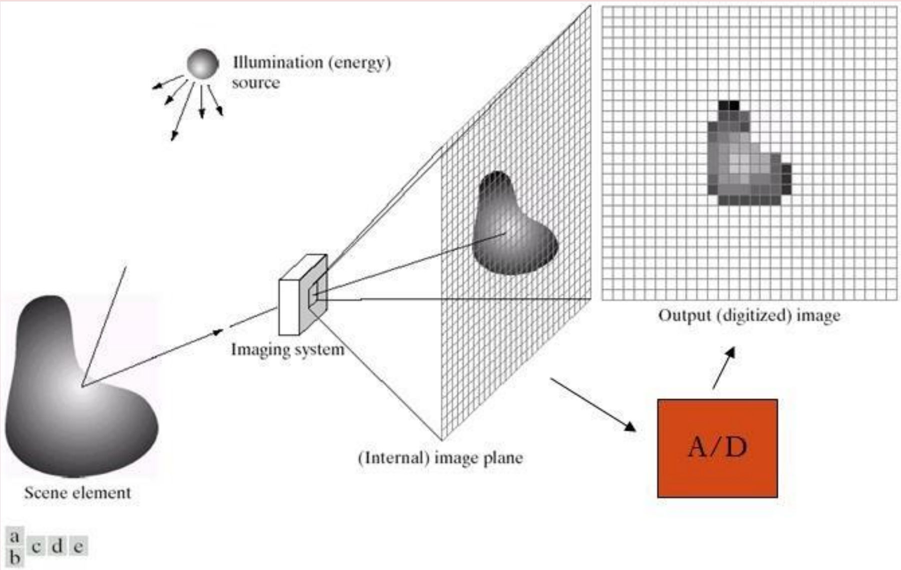
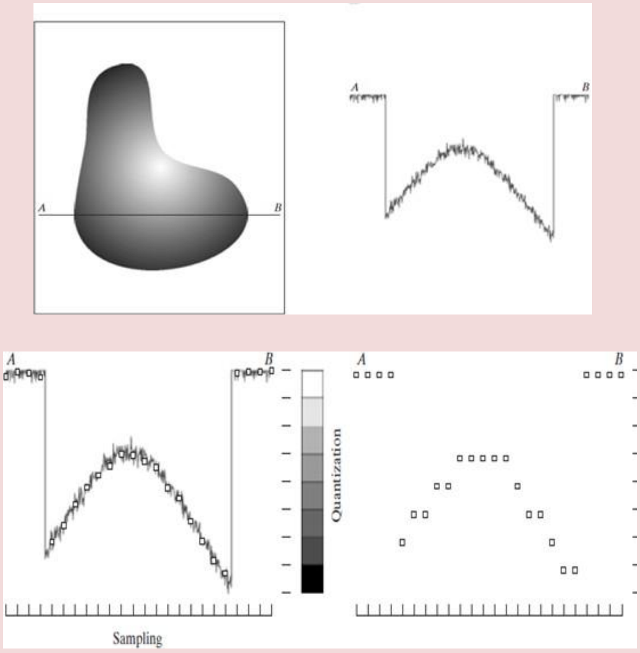


FIGURE An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image sampling and Quantization:

To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: *sampling* and *quantization*. A continuous image, f(x, y), that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called *sampling*. Digitizing the amplitude values is called *quantization*.



Digital Image representation:

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

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

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

RELATIONSHIP BETWEEN PIXELS:

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

NEIGHBORS OF A PIXEL

- A pixel *p* at coordinates (*x*,*y*) has four *horizontal* and *vertical* neighbors whose coordinates are given by:

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

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

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

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

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

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

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

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

ADJACENCY AND CONNECTIVITY

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

If the possible intensity values 0 – 255, *V* set can be any subset of these 256 values.

if we are reference to adjacency of pixel with value.

Three types of adjacency

- 4- Adjacency – two pixel P and Q with value from V are 4 –adjacency if A is in the set $N_4(P)$
- 8- Adjacency – two pixel P and Q with value from V are 8 –adjacency if A is in the set $N_8(P)$
- M-adjacency –two pixel P and Q with value from V are m – adjacency if (i) Q is in $N_4(p)$ or (ii) Q is in $N_D(q)$ and the set $N_4(p) \cap N_4(q)$ has no pixel whose values are from V.
- Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.
- For example:

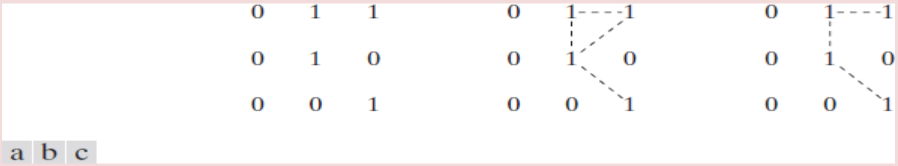


Fig:1.8(a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) ***m***-adjacency.

Types of Adjacency:

- In this example, we can note that to connect between two pixels (finding a path between two pixels):
  - In 8-adjacency way, you can find multiple paths between two pixels
  - While, in m-adjacency, you can find only one path between two pixels
- So, m-adjacency has eliminated the multiple path connection that has been generated by the 8-adjacency.
- Two subsets S_1 and S_2 are adjacent, if some pixel in S_1 is adjacent to some pixel in S_2. Adjacent means, either 4-, 8- or m-adjacency.

A Digital Path:

- A digital path (or curve) from pixel p with coordinate (x,y) to pixel q with coordinate (s,t) is a sequence of distinct pixels with coordinates (x_0,y_0), (x_1,y_1), …, (x_n, y_n) where $(x_0,y_0) = (x,y)$ and $(x_n, y_n) = (s,t)$ and pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$
- n is the length of the path
- If $(x_0,y_0) = (x_n, y_n)$, the path is closed.

We can specify 4-, 8- or m-paths depending on the type of adjacency specified.

- Return to the previous example:

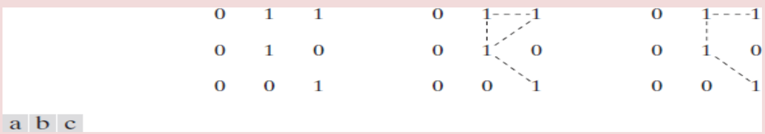


Fig:1.8 (a) Arrangement of pixels; (b) pixels that are 8-adjacent(shown dashed) to the center pixel; (c) m-adjacency.

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

Connectivity:

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

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

Region and Boundary:

- REGION: Let *R* be a subset of pixels in an image, we call *R* a region of the image if *R* is a connected set.
- BOUNDARY: The *boundary* (also called *border* or *contour*) of a region *R* is the set of pixels in the region that have one or more neighbors that are not in *R*.

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

DISTANCE MEASURES:

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

D [p.q] ≥ O {D[p.q] = O iff p=q}

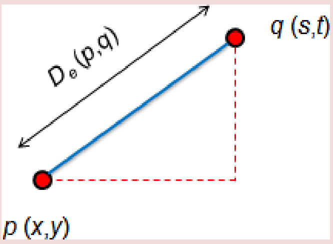
D [p.q] = D [p.q] and

D [p.q] ≥ O {D[p.q]+D(q,z)

- The ***Euclidean Distance*** between *p* and *q* is defined as:

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

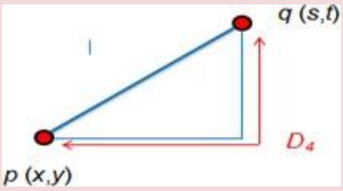
Pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius „ r „centered at (x,y)



- The *D_4 distance* (also called ***city-block distance***) between *p* and *q* is defined as:

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

Pixels having a D_4 distance from (x,y), less than or equal to some value r form a Diamond centered at (x,y)



Example:

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

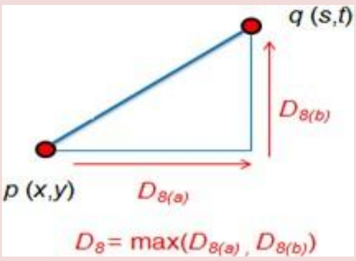
The pixels with $D_4 = 1$ are the 4-neighbors of (x,y)

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

- The D_8 *distance* (also called **chessboard distance**) between p and q is defined as:

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

  Pixels having a D_8 distance from (x,y), less than or equal to some value r form a square Centered at (x,y).



Example:

D_8 distance ≤ 2 from (x,y) form the following contours of constant distance.

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

- **D_m distance:**

  It is defined as the shortest m-path between the points.

In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

- Example:

Consider the following arrangement of pixels and assume that p, p_2, and p_4 have value 1 and that p_1 and p_3 can have can have a value of 0 or 1 Suppose that we consider the adjacency of pixels values 1 (i.e. $V = \{1\}$)

$$\begin{array}{|ccc|} \hline & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \\ \hline \end{array}$$

Now, to compute the D_m between points p and p_4

Here we have 4 cases:

Case1: If p_1 =0 and p_3 = 0

The length of the shortest m-path

(the D_m *distance) is 2 (p, p* p_2, p_4*)*

$$\begin{array}{|ccc|} \hline & 0 & 1 \\ 0 & 1 & \\ 1 & & \\ \hline \end{array}$$

Case2: If p_1 =1 and p_3 = 0

now, p_1 *and p will no longer be adjacent (see m-adjacency definition)*

then, the length of the shortest

path will be 3 (p, p_1, p_2, p_4)

$$\begin{array}{|ccc|} \hline & 0 & 1 \\ 1 & 1 & \\ 1 & & \\ \hline \end{array}$$

Case3: If p_1 =0 and p_3 = 1

The same applies here, and the shortest –m-path will be 3 (p, p_2, p_3, p_4)

$$\begin{array}{|ccc|} \hline & 1 & 1 \\ 0 & 1 & \\ 1 & & \\ \hline \end{array}$$

Case4: If p_1 =1 and p_3 = 1

The length of the shortest m-path will be 4 (p, p_1 , p_2, p_3, p_4)

$$\begin{array}{|ccc|} & 1 & 1 \\ 1 & 1 & \\ 1 & & \end{array}$$

IMAGE TRANSFORMS:

2-D FFT:

2D Discrete Fourier Transform

The independent variable (t,x,y) is discrete

$$F_r = \sum_{k=0}^{N_0-1} f[k] e^{-jr\Omega_0 k}$$

$$f_{N_0}[k] = \frac{1}{N_0} \sum_{r=0}^{N_0-1} F_r e^{jr\Omega_0 k}$$

$$\Omega_0 = \frac{2\pi}{N_0}$$

$$F[u,v] = \sum_{i=0}^{N_0-1} \sum_{k=0}^{N_0-1} f[i,k] e^{-j\Omega_0(ui+vk)}$$

$$f_{N_0}[i,k] = \frac{1}{N_0^{\,2}} \sum_{u=0}^{N_0-1} \sum_{v=0}^{N_0-1} F[u,v] e^{j\Omega_0(ui+vk)}$$

$$\Omega_0 = \frac{2\pi}{N_0}$$

Properties

- Linearity $\quad af(x,y)+bg(x,y) \Leftrightarrow aF(u,v)+bG(u,v)$
- Shifting $\quad f(x-x_0, y-x_0) \Leftrightarrow e^{-j2\pi(ux_0+vy_0)}F(u,v)$
- Modulation $\quad e^{j2\pi(u_0x+v_0y)} f(x,y) \Leftrightarrow F(u-u_0, v-v_0)$
- Convolution $\quad f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$
- Multiplication $\quad f(x,y)g(x,y) \Leftrightarrow F(u,v) * G(u,v)$
- Separability $\quad f(x,y) = f(x)f(y) \Leftrightarrow F(u,v) = F(u)F(v)$

Separability

1. Separability of the 2D Fourier transform
   – 2D Fourier Transforms can be implemented as a sequence of 1D Fourier Transform operations performed *independently* along the two axis

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

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}e^{-j2\pi vy}dxdy=\int_{-\infty}^{\infty} e^{-j2\pi vy}dy\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}dx=$$

$$=\int_{-\infty}^{\infty} F(u,y)e^{-j2\pi vy}dy=F(u,v)$$

2D DFT ⇒ 1D DFT along the rows → 1D DFT along the cols

Separability

- Separable functions can be written as $f(x,y)=f(x)g(y)$

2. The FT of a separable function is the product of the FTs of the two functions

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

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x)g(y)e^{-j2\pi ux}e^{-j2\pi vy}dxdy=\int_{-\infty}^{\infty} g(y)e^{-j2\pi vy}dy\int_{-\infty}^{\infty} h(x)e^{-j2\pi ux}dx=$$

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

$$f(x,y)=h(x)g(y)\Rightarrow F(u,v)=H(u)G(v)$$

WALSH TRANSFORM:

We define now the 1-D Walsh transform as follows:

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

The above is equivalent to:

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

The transform kernel values are obtained from:

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

Therefore, the array formed by the Walsh matrix is a real symmetric matrix. It is easily shown that it has orthogonal columns and rows

1-D Inverse Walsh Transform

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

The above is again equivalent to

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

The array formed by the inverse Walsh matrix is identical to the one formed by the forward Walsh matrix apart from a multiplicative factor N.

2-D Walsh Transform

We define now the 2-D Walsh transform as a straightforward extension of the 1-D transform:

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

•The above is equivalent to:

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

Inverse Walsh Transform

We define now the Inverse 2-D Walsh transform. It is identical to the forward 2-D Walsh transform

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

•The above is equivalent to:

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

HADAMARD TRANSFORM:

We define now the 2-D Hadamard transform. It is similar to the 2-D Walsh transform.

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

The above is equivalent to:

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

We define now the Inverse 2-D Hadamard transform. It is identical to the forward 2-D Hadamard transform.

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

The above is equivalent to:

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

DISCRETE COSINE TRANSFORM (DCT) :

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

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

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

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

where

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

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

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

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

where

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

The basic operation of the DCT is as follows:

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

DISCRETE WAVELET TRANSFORM (DWT):

There are many discrete wavelet transforms they are Coiflet, Daubechies, Haar, Symmlet etc.

Haar Wavelet Transform

The Haar wavelet is the first known wavelet. The Haar wavelet is also the simplest possible wavelet. The Haar Wavelet can also be described as a step function f(x) shown in Eq

$$f(x) = \begin{cases} 1 & 0 \le x < 1/2, \\ -1 & 1/2 \le x < 1, \\ 0 & otherwise. \end{cases}$$

Each step in the one dimensional Haar wavelet transform calculates a set of wavelet coefficients (Hi-D) and a set of averages (Lo-D). If a data set s_0, s_1,…, s_{N-1} contains N elements, there will be N/2 averages and N/2 coefficient values. The averages are stored in the lower half of the N element array and the coefficients are stored in the upper half.

The Haar equations to calculate an average (a_i) and a wavelet coefficient (c_i) from the data set are shown below Eq

$$a_i = \frac{s_i + s_{i+1}}{2} \qquad c_i = \frac{s_i - s_{i+1}}{2}$$

In wavelet terminology the Haar average is calculated by the scaling function. The coefficient is calculated by the wavelet function.

Two-Dimensional Wavelets

The two-dimensional wavelet transform is separable, which means we can apply a one-dimensional wavelet transform to an image. We apply one-dimensional DWT to all rows and then one-dimensional DWTs to all columns of the result. This is called the standard decomposition and it is illustrated in figure 4.8.

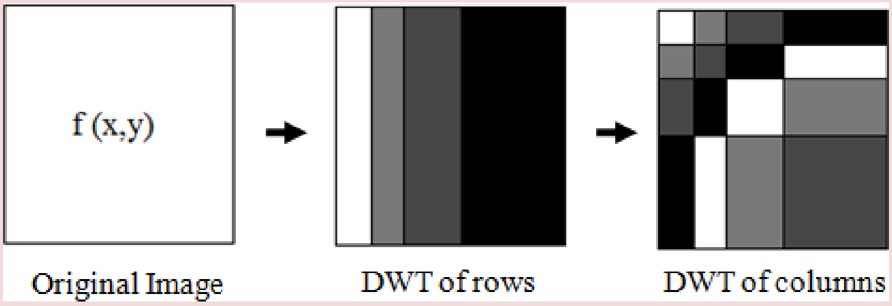


Figure The standard decomposition of the two-dimensional DWT.

We can also apply a wavelet transform differently. Suppose we apply a wavelet transform to an image by rows, then by columns, but using our transform at one scale only. This technique will produce a result in four quarters: the top left will be a half-sized version of the image and the other quarter‟s high-pass filtered images. These quarters will contain horizontal, vertical, and diagonal edges of the image. We then apply a one-scale DWT to the top-left quarter, creating smaller images, and so on. This is called the nonstandard decomposition, and is illustrated in figure 4.9.

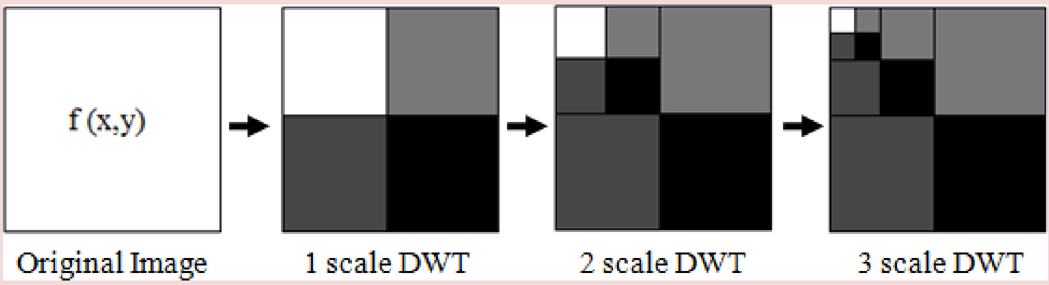


Figure 4.9 The nonstandard decomposition of the two-dimensional DWT.

Steps for performing a one-scale wavelet transform are given below:

Step 1: Convolve the image rows with the low-pass filter.

Step 2 : Convolve the columns of the result of step 1 with the low-pass filter and rescale this to half its size by sub-sampling.

Step 3 : Convolve the result of step 1 with high-pass filter and again sub-sample to obtain an image of half the size.

Step 4 : Convolve the original image rows with the high-pass filter.

Step 5: Convolve the columns of the result of step 4 with the low-pass filter and recycle this to half its size by sub-sampling.

Step 6 :Convolve the result of step 4 with the high-pass filter and again sub-sample to obtain an image of half the size.

At the end of these steps there are four images, each half the size of original. They are

1. The low-pass / low-pass image (LL), the result of step 2,
2. The low-pass / high-pass image (LH), the result of step 3,
3. The high-pass / low-pass image (HL), the result of step 5, and
4. The high-pass / high-pass image (HH), the result of step 6

These images can be placed into a single image grid as shown in the figure 4.10.

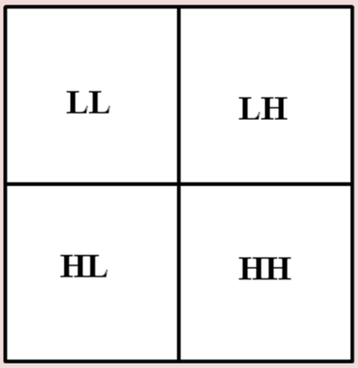


Figure 4.10 the one-scale wavelet transforms in terms of filters.

Figure 4.11 describes the basic dwt decomposition steps for an image in a block diagram form. The two-dimensional DWT leads to a decomposition of image into four components CA, CH, CV and CD, where CA are approximation and CH, CV, CD are details in three orientations (horizontal, vertical, and diagonal), these are same as LL, LH, HL, and HH. In these coefficients the watermark can be embedded.

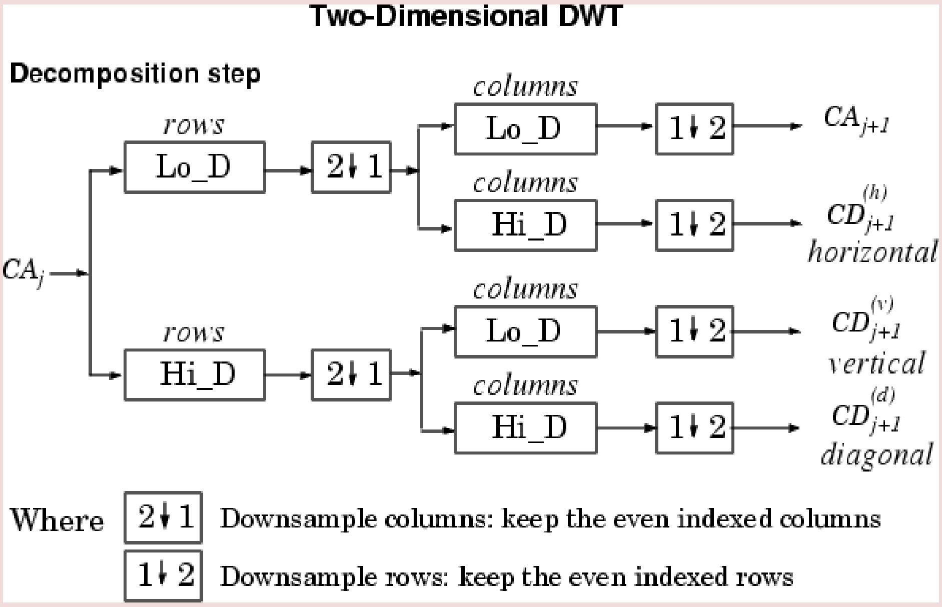


Figure 4.11 DWT decomposition steps for an image.

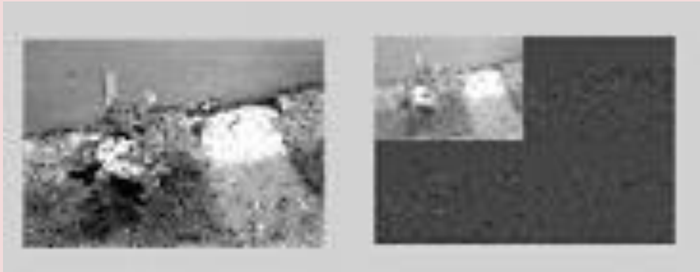


Figure Original image and DWT decomposed image.

An example of a discrete wavelet transform on an image is shown in Figure above. On the left is the original image data, and on the right are the coefficients after a single pass of the wavelet transform. The low-pass data is the recognizable portion of the image in the upper left corner. The high-pass components are almost invisible because image data contains mostly low frequency information.

IMAGE ENHANCEMENT

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

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

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

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

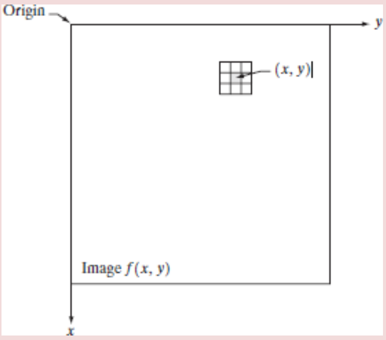


Fig.: 3x3 neighborhood about a point (x,y) in an image.

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

where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of „r" to each value of „s".

For example, if T(r) has the form shown in Fig. 2.2(a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation function into a narrow range of s, toward black.The opposite effect takes place for values of r above m.

In the limiting case shown in Fig. 2.2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.

One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig. 2.1, in which the values of the mask coefficients determine the nature of the process, such as image sharpening. Enhancement techniques based on this type of approach often are referred to as mask processing or filtering.

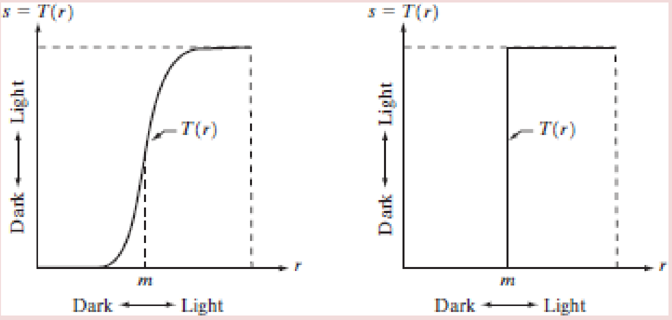


Fig. 2.2 Gray level transformation functions for contrast enhancement.

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

BASIC GRAY LEVEL TRANSFORMATIONS:

- Image negative
- Log transformations
- Power law transformations
- Piecewise-Linear transformation functions

LINEAR TRANSFORMATION:

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

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

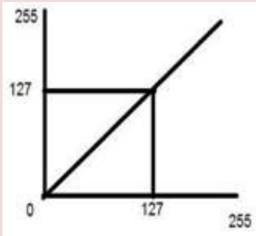


Fig. Linear transformation between input and output.

NEGATIVE TRANSFORMATION:

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

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

$$S = L - 1 - r$$

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

L is the largest gray level consists in the image

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

The overall graph of these transitions has been shown below.

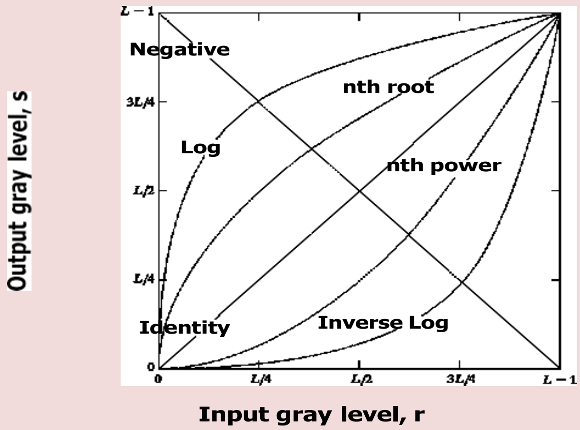


Fig. Some basic gray-level transformation functions used for image enhancement.

In this case the following transition has been done.

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

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

$$S = 255 - r$$

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

It has been shown in the graph below.

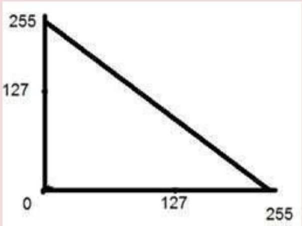


Fig. Negative transformations.

LOGARITHMIC TRANSFORMATIONS:

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

LOG TRANSFORMATIONS:

The log transformations can be defined by this formula

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

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

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

An another way of representing LOG TRANSFORMATIONS: Enhance details in the darker regions of an image at the expense of detail in brighter regions.

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

- Here C is constant and $r \geq 0$.

- The shape of the curve shows that this transformation maps the narrow range of low gray level values in the input image into a wider range of output image.
- The opposite is true for high level values of input image.

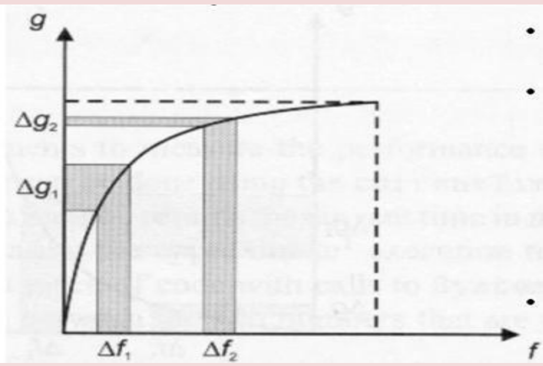


Fig. log transformation curve input vs output

POWER – LAW TRANSFORMATIONS:

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

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

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

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

where c and g are positive constants. Sometimes Eq. (6) is written as $S = C(r+\varepsilon)^{\gamma}$ to account for an offset (that is, a measurable output when the input is zero). Plots of s versus r for various values of γ are shown in Fig. 2.10. As in the case of the log transformation, power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels. Unlike the log function, however, we notice here a family of possible transformation curves obtained simply by varying γ.

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

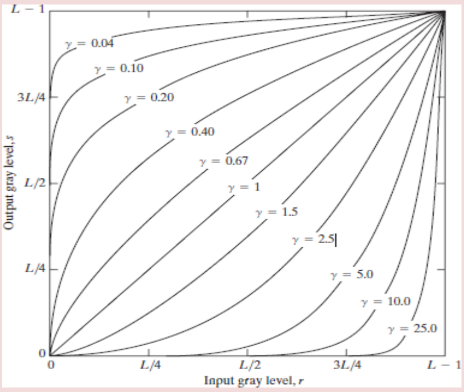


Fig. 2.13 Plot of the equation $S = cr^{\gamma}$ for various values of γ (c =1 in all cases).

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

Varying gamma (γ) obtains family of possible transformation curves $S = C * r^{\gamma}$

Here C and γ are positive constants. Plot of S versus r for various values of γ is

$\gamma > 1$ compresses dark values

Expands bright values

$\gamma < 1$ (similar to Log transformation)

Expands dark values

Compresses bright values

When $C = \gamma = 1$, it reduces to identity transformation.

CORRECTING GAMMA:

$s=cr^{\gamma}$

$s=cr^{(1/2.5)}$

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

Piecewise-Linear Transformation Functions:

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

The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

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

S=T(r)

Figure x(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation

Function. If r1=s1 and r2=s2, the transformation is a linear function that produces No changes in gray levels. If r1=r2, s1=0and s2= L-1, the transformation Becomes a thresholding function that creates a binary image, as illustrated In fig. 2.2(b).

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

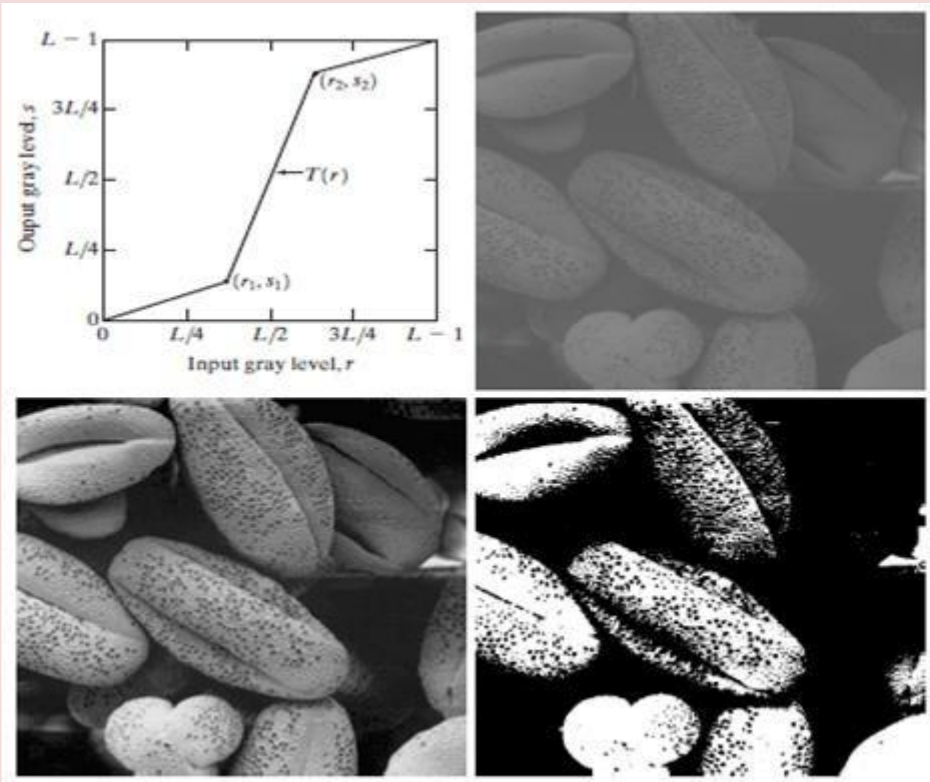


Fig. x Contrast stretching. (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding (original image courtesy of Dr.Roger Heady, Research School of Biological Sciences, Australian National University Canberra Australia.

Figure x(b) shows an 8-bit image with low contrast. Fig. x(c) shows the result of contrast stretching, obtained by setting (r1, s1)=(rmin, 0) and (r2, s2)=(r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively.Thus, the transformation function stretched the levels linearly from their original range to the full range

[0, L-1]. Finally, Fig. x(d) shows the result of using the thresholding function defined previously,

with r1=r2=m, the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.

There are several ways of doing level slicing, but most of them are variations of two basic themes.One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.

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

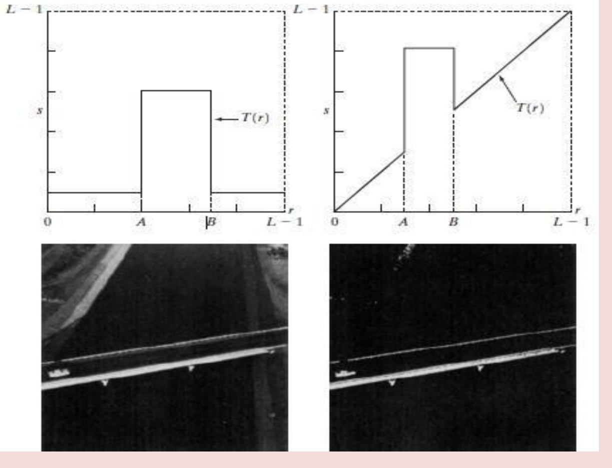


Fig. y (a)This transformation highlights range [A,B] of gray levels and reduces all others to a constant level (b) This transformation highlights range [A,B] but preserves all other levels. (c) An image . (d) Result of using the transformation in (a).

BIT-PLANE SLICING:

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

bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.

Figure 3.12 illustrates these ideas, and Fig. 3.14 shows the various bit planes for the image shown in Fig. 3.13. Note that the higher-order bits (especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

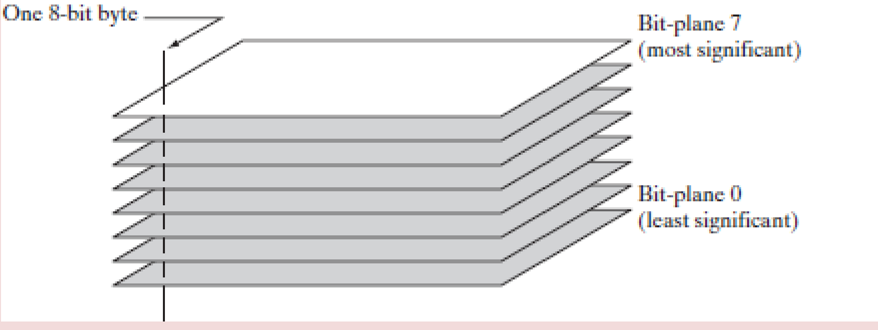


FIGURE
Bit-plane representation of an 8-bit image.

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

(Problem 3.3) to obtain the gray-level transformation functions that would yield the other bit planes.

Histogram Processing:

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

$$H(r_k)=n_k$$

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

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

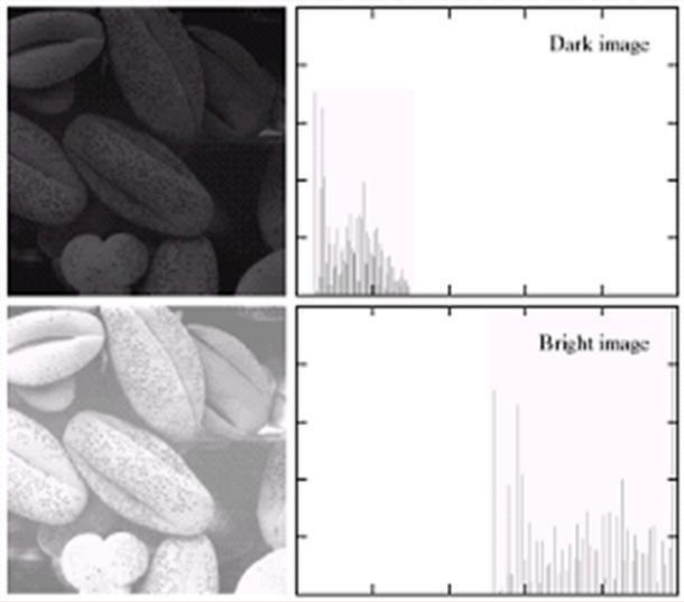
P(rk) gives the estimate of the probability of occurrence of gray level rk.

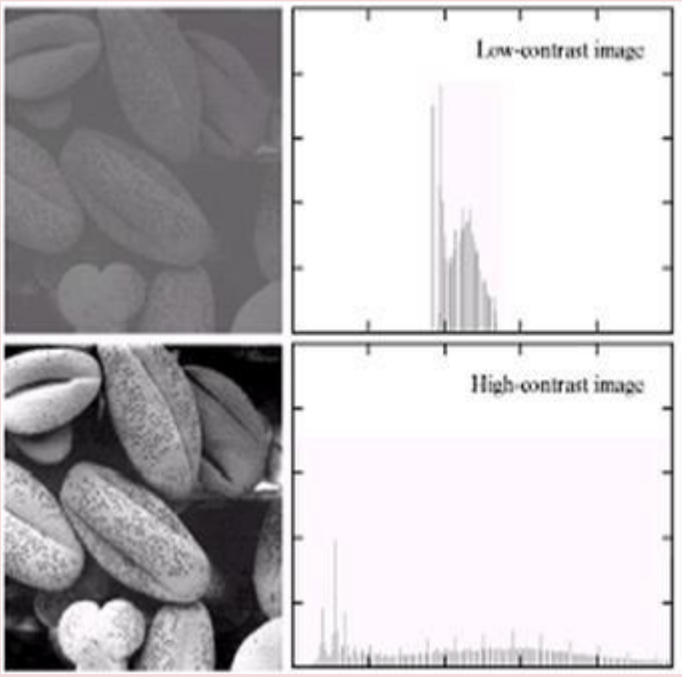
The sum of all components of a normalized histogram is equal to 1.

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

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

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





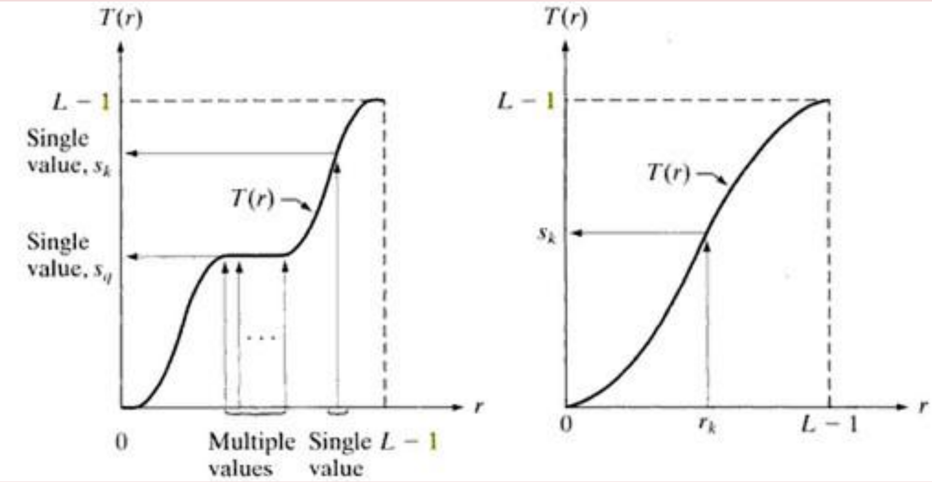
Histogram Equalization:

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

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

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

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



a b

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

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

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

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

A transformation function of a particular importance in image processing

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

This is the cumulative distribution function of r.

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

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

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

IDEAL LOW-PASS FILTER:

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

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

Where D0 is a positive constant and D(u,v) is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle; that is

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

Where as P and Q are the padded sizes from the basic equations

Wraparound error in their circular convolution can be avoided by padding these functions with zeros,

VISUALIZATION: IDEAL LOW PASS FILTER:

Aa shown in fig.below

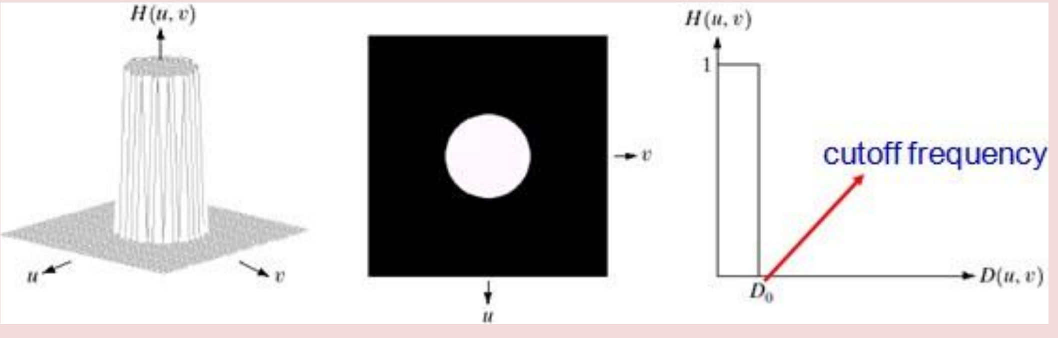


Fig: ideal low pass filter 3-D view and 2-D view and line graph.

EFFECT OF DIFFERENT CUTOFF FREQUENCIES:

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

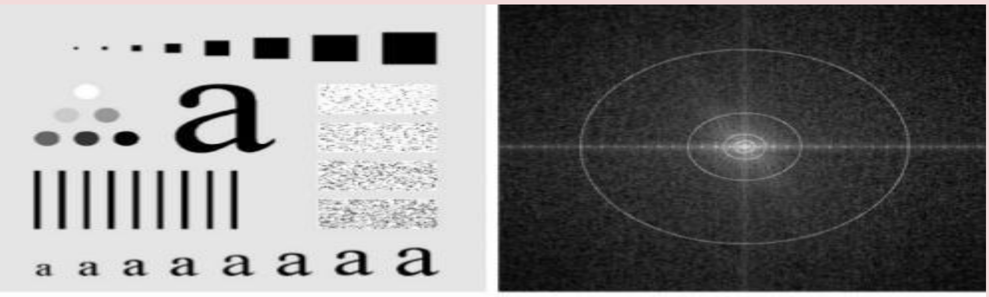


Fig: (a) Test patter of size 688x688 pixels (b) its Fourier spectrum

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

As the cutoff frequency decreases,

- image becomes more blurred
- Noise becomes increases
- Analogous to larger spatial filter sizes

The severe blurring in this image is a clear indication that most of the sharp detail information in the picture is contained in the 13% power removed by the filter. As the filter radius is increases less and less power is removed, resulting in less blurring. Fig. (c) through (e) are characterized by “ringing” , which becomes finer in texture as the amount of high frequency content removed decreases.

WHY IS THERE RINGING?

Ideal low-pass filter function is a rectangular function

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

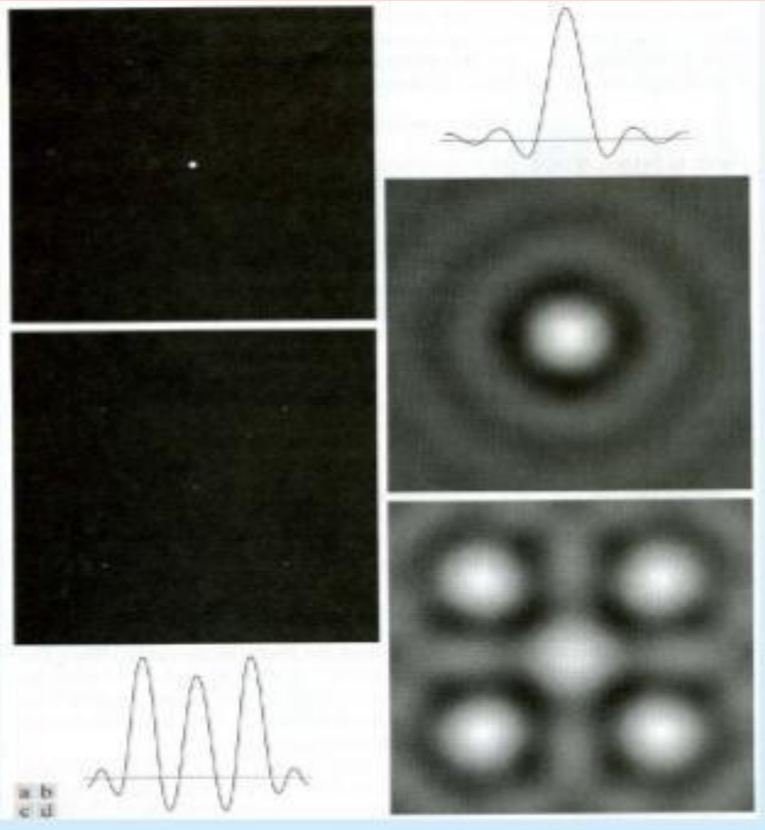


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity profiles through the center of the filters(the size of all cases is 1000x1000 and the cutoff frequency is 5), observe how ringing increases as a function of filter order.

BUTTERWORTH LOW-PASS FILTER:

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

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

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

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

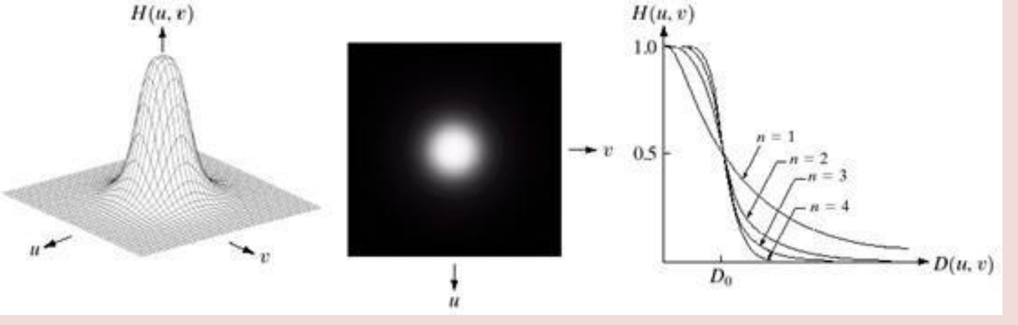


Fig. (a) perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c)Filter radial cross sections of order 1 through 4.

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

BUTTERWORTH LOW-PASS FILTERS OF DIFFEREN T FREQUENCIES:

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

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

A BLPF of order 1 has no ringing in the spatial domain. Ringing generally is imperceptible in filters of order 2, but can become significant in filters of higher order.

Fig.shows a comparison between the spatial representation of BLPFs of various orders (using a cutoff frequency of 5 in all cases). Shown also is the intensity profile along a horizontal scan line through the center of each filter. The filter of order 2 does show mild ringing and small negative values, but they certainly are less pronounced than in the ILPF. A butter worth filter of order 20 exhibits characteristics similar to those of the ILPF (in the limit, both filters are identical).

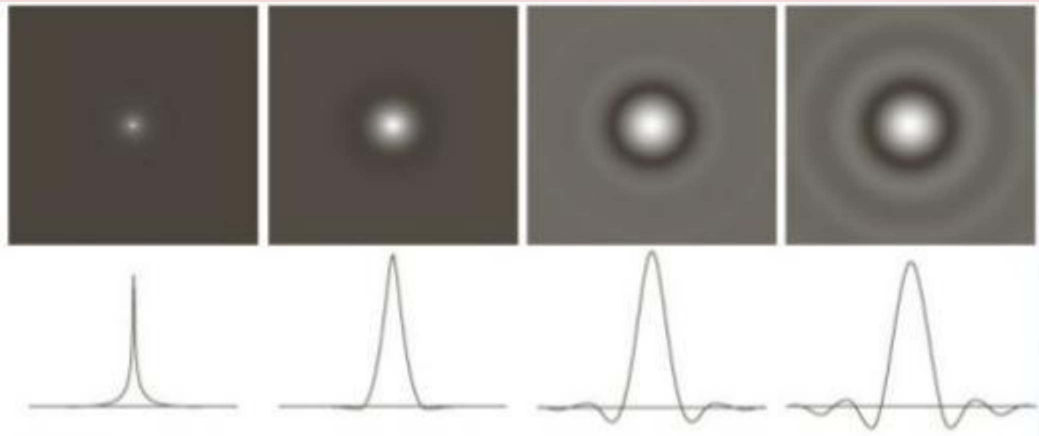


Fig.2.2.7 (a)-(d) Spatial representation of BLPFs of order 1, 2, 5 and 20 and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 x 1000 and the cutoff frequency is 5) Observe how ringing increases as a function of filter order.

GAUSSIAN LOWPASS FILTERS:

The form of these filters in two dimensions is given by

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

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

Where D_0 is the cutoff frequency. When $D(u,v) = D_0$, the GLPF is down to 0.607 of its maximum value. This means that a spatial Gaussian filter, obtained by computing the IDFT of above equation., will have no ringing. Fig..shows a perspective plot, image display and radial cross sections of a GLPF function.

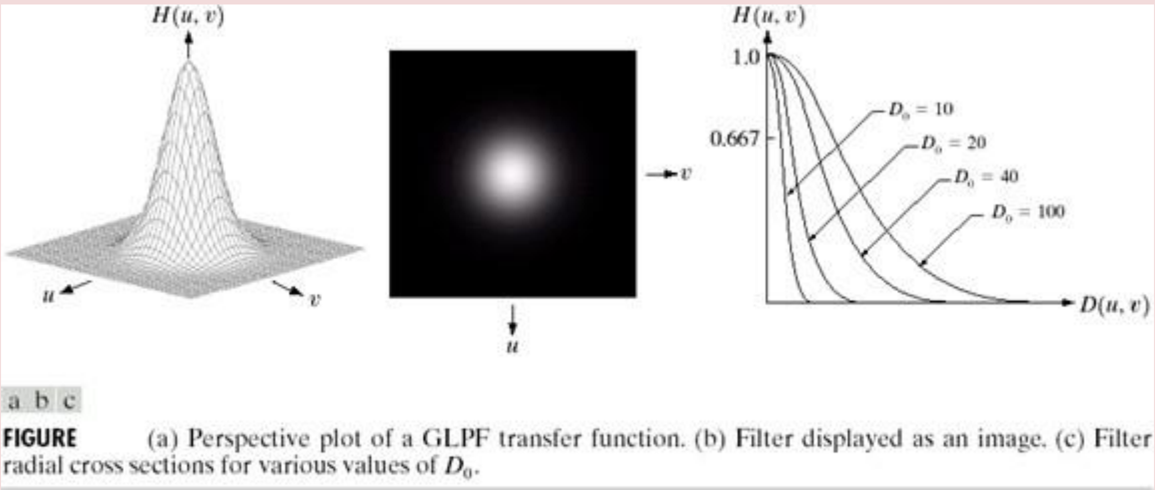


Fig. (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c). Filter radial cross sections for various values of D_0

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

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

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

IMAGE SHARPENING USING FREQUENCY DOMAIN FILTERS:

An image can be smoothed by attenuating the high-frequency components of its Fourier transform. Because edges and other abrupt changes in intensities are associated with high-frequency components, image sharpening can be achieved in the frequency domain by high pass filtering, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

The filter function $H(u,v)$ are understood to be discrete functions of size PxQ; that is the discrete frequency variables are in the range $u = 0,1,2,\ldots\ldots.P-1$ and $v = 0,1,2,\ldots\ldots.Q-1$.

The meaning of sharpening is

- Edges and fine detail characterized by sharp transitions in image intensity
- Such transitions contribute significantly to high frequency components of Fourier transform
- Intuitively, attenuating certain low frequency components and preserving high frequency components result in sharpening.

Intended goal is to do the reverse operation of low-pass filters

- When low-pass filter attenuated frequencies, high-pass filter passes them

- When high-pass filter attenuates frequencies, low-pass filter passes them.

A high pass filter is obtained from a given low pass filter using the equation.

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

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

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

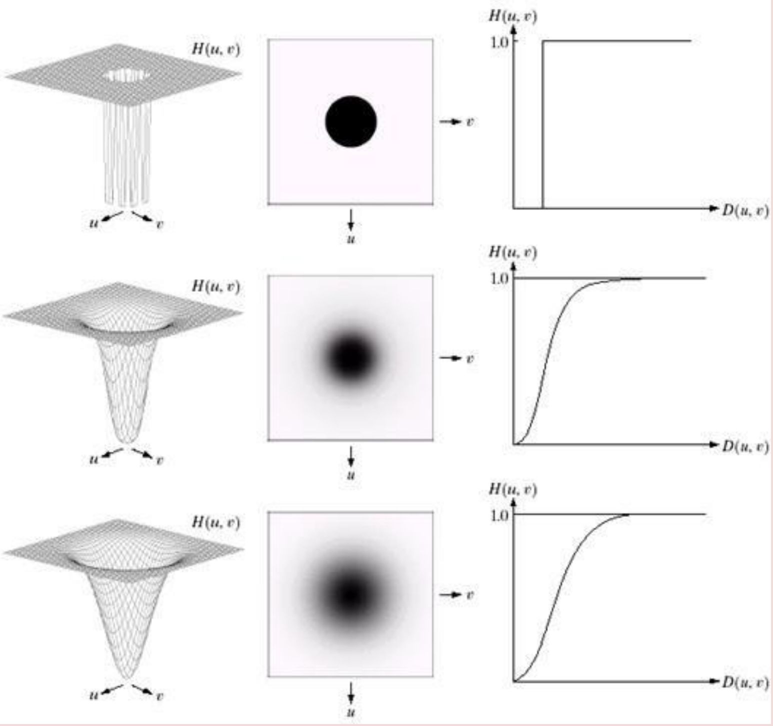


Fig: Top row: Perspective plot, image representation, and cross section of a typical ideal high-pass filter. Middle and bottom rows: The same sequence for typical butter-worth and Gaussian high-pass filters.

IDEAL HIGH-PASS FILTER:

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

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

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

SPATIAL REPRESENTATION OF HIGHPASS FILTERS:

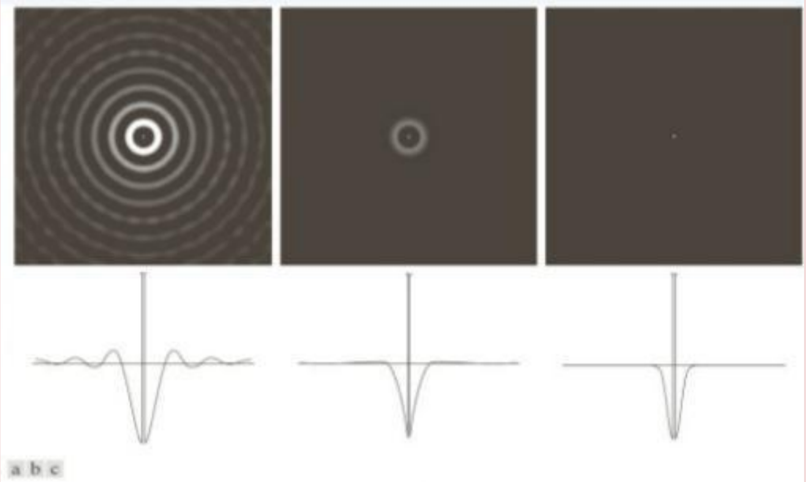


Fig.. Spatial representation of typical (a) ideal (b) Butter-worth and (c) Gaussian frequency domain high-pass filters, and corresponding intensity profiles through their centers.

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

FILTERED RESULTS: IHPF

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

The situation improved somewhat with D0 = 60. Edge distortion is quite evident still, but now we begin to see filtering on the smaller objects. Due to the now familiar inverse relationship between the frequency and spatial domains, we know that the spot size of this filter is smaller than the spot of the filter with D0 = 30. The result for D0 = 160 is closer to what a high-pass filtered image should look like. Here, the edges are much cleaner and less distorted, and the smaller objects have been filtered properly.

Of course, the constant background in all images is zero in these high-pass filtered images because highpass filtering is analogous to differentiation in the spatial domain.

BUTTER-WORTH HIGH-PASS FILTERS:

A 2-D Butter-worth high-pass filter (BHPF) of order n and cutoff frequency D_0 is defined as

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

Where D(u,v) is given by Eq.(3). This expression follows directly from Eqs.(3) and (6). The middle row of Fig.2.2.11. shows an image and cross section of the BHPF function.

Butter-worth high-pass filter to behave smoother than IHPFs. Fig.2.2.14.shows the performance of a BHPF of order 2 and with D0 set to the same values as in Fig.2.2.13. The boundaries are much less distorted than in Fig.2.2.13. even for the smallest value of cutoff frequency.

FILTERED RESULTS: BHPF:

Fig. Results of high-pass filtering the image in Fig.2.2.2(a) using a BHPF of order 2 with D_0 = 30, 60, and 160 corresponding to the circles in Fig.2.2.2(b). These results are much smoother than those obtained with an IHPF.

GAUSSIAN HIGH-PASS FILTERS:

The transfer function of the Gaussian high-pass filter(GHPF) with cutoff frequency locus at a distance D_0 from the center of the frequency rectangle is given by

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

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

FILTERED RESULTS:GHPF:

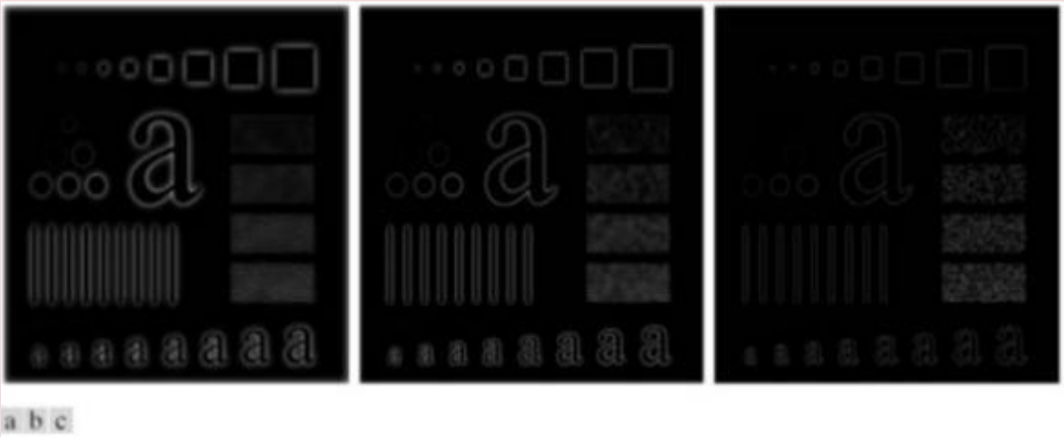


Fig. Results of high-pass filtering the image in fig.(a) using a GHPF with D0 = 30, 60 and 160, corresponding to the circles in Fig.(b).

IMAGE RESTORATION

IMAGE RESTORATION:

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

- During display mode
- Acquisition mode, or
- Processing mode
    - Sensor noise
    - Blur due to camera mis focus
    - Relative object-camera motion
    - Random atmospheric turbulence
- Others

Degradation Model:

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

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

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

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

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

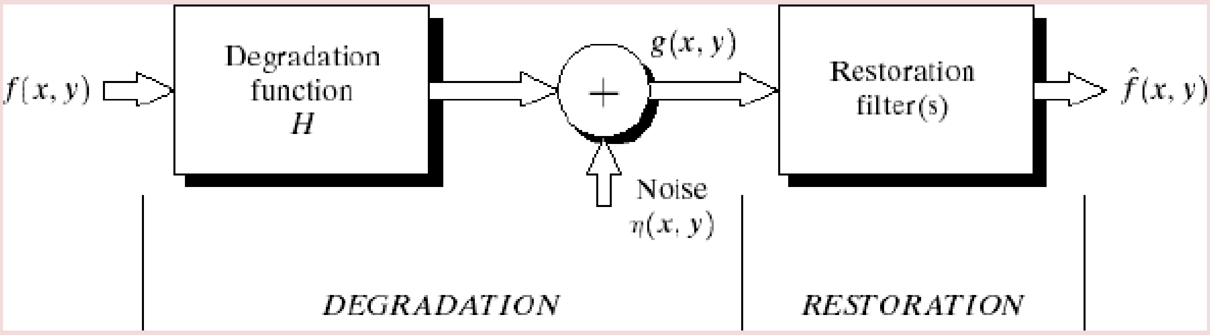


Fig: A model of the image Degradation / Restoration process

Noise Models:

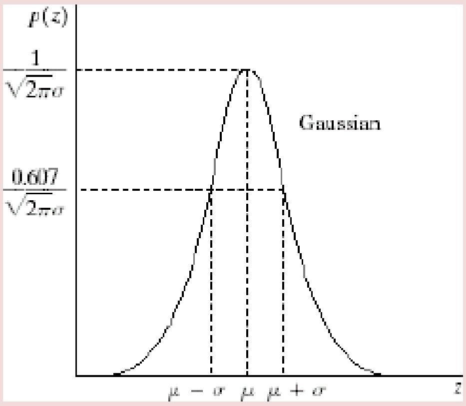
The principal source of noise in digital images arises during image acquisition and /or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted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

Gaussian Noise:

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

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

Where z represents the gray level, μ= mean of average value of z, σ= standard deviation.



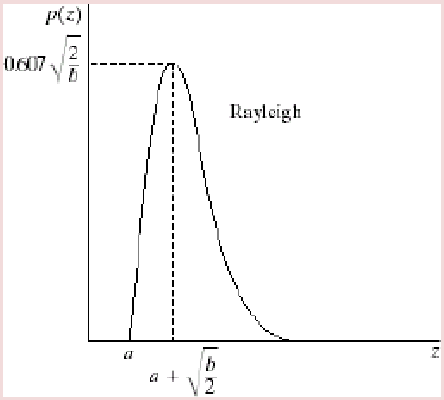
Rayleigh Noise:

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

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

The mean and variance of this density is

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



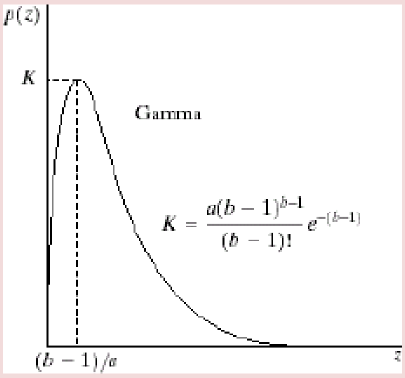
(iii) Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

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



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

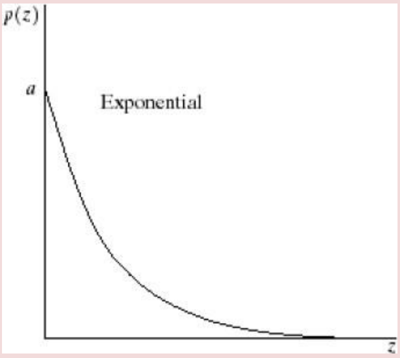
(iv) Exponential Noise:

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

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

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

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



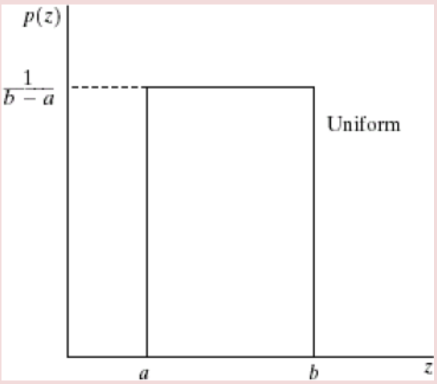
(v) Uniform Noise:

The PDF of uniform noise is given by

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

The mean and variance of this noise is

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



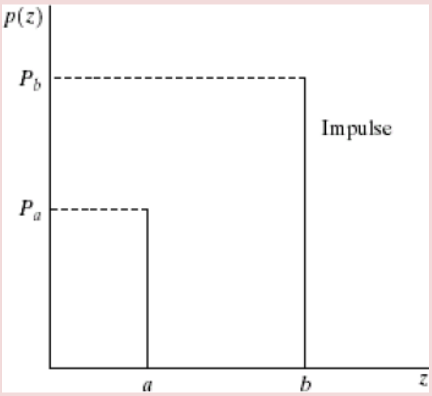
(vi) Impulse (salt & pepper) Noise:

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

The PDF of bipolar (impulse) noise is given by

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

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



Restoration in the presence of Noise only- Spatial filtering:

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

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

or

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

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

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

i) **Mean Filter:**

ii) **(a)Arithmetic Mean filter:**

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

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

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

(b) Geometric Mean filter:

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

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

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

(c) Harmonic Mean filter:

The harmonic mean filtering operation is given by the expression

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

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

(d) Order statistics filter:

Order statistics filters are spatial filters whose response is based on ordering the pixel contained in the image area encompassed by the filter. The response of the filter at any point is determined by the ranking result.

(e) Median filter:

It is the best order statistic filter; it replaces the value of a pixel by the median of gray levels in the Neighborhood of the pixel.

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

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

(e) Max and Min filter:

Using the l00th percentile of ranked set of numbers is called the max filter and is given by the equation

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

It is used for finding the brightest point in an image. Pepper noise in the image has very low values, it is reduced by max filter using the max selection process in the sublimated area sky. The 0th percentile filter is min filter.

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

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

(f) Midpoint filter:

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

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

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

Periodic Noise by Frequency domain filtering:

These types of filters are used for this purpose-

Band Reject Filters:

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

Ideal Band reject Filter:

An ideal band reject filter is given by the expression

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

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

W- the width of the band

Do- the radial center of the frequency rectangle.

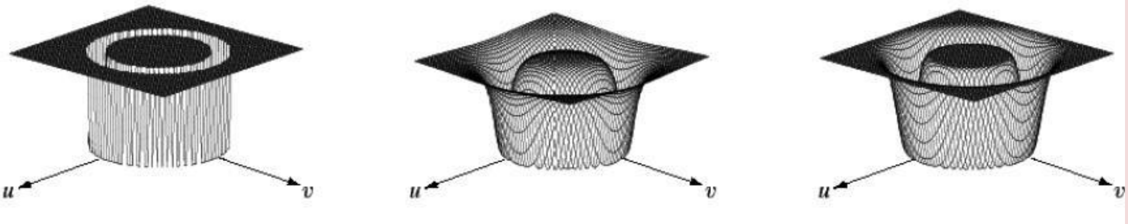
Butterworth Band reject Filter:

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

Gaussian Band reject Filter:

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

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



a b c

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

Band pass Filter:

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

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

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

Notch Filters:

A notch filter rejects (or passes) frequencies in predefined neighborhoods about a center frequency.

Due to the symmetry of the Fourier transform notch filters must appear in symmetric pairs about the origin.

The transfer function of an ideal notch reject filter of radius D_0 with centers a (u_0, v_0) and by symmetry at $(-u_0, v_0)$ is

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

Ideal, butterworth, Gaussian notch filters

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

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

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

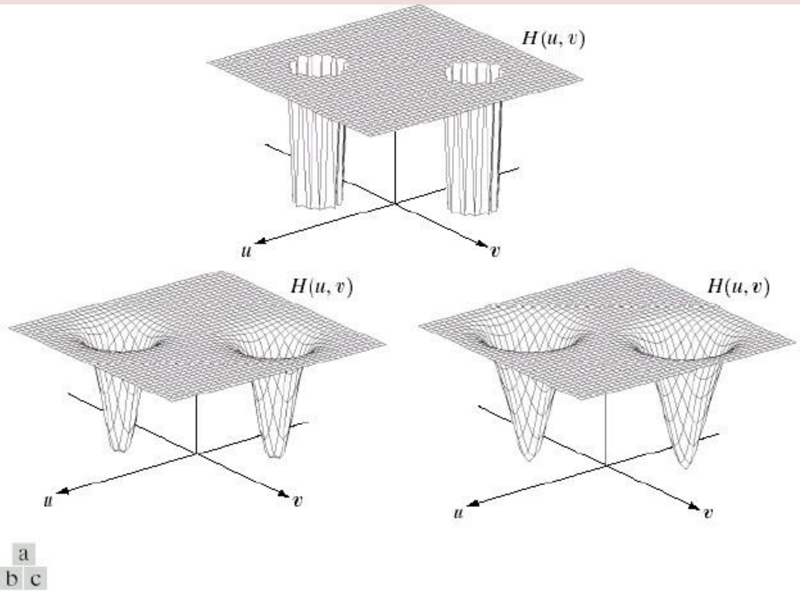


FIGURE Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Inverse Filtering:

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

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

We know that

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

Therefore

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

From the above equation we observe that we cannot recover the undegraded image exactly because N(u,v) is a random function whose Fourier transform is not known.

One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin.

We know that H(0,0) is equal to the average values of h(x,y).

By Limiting the analysis to frequencies near the origin we reduse the probability of encountering zero values.

Minimum mean Square Error (Wiener) filtering:

The inverse filtering approach has poor performance. The wiener filtering approach uses the degradation function and statistical characteristics of noise into the restoration process.

The objective is to find an estimate $\hat{f}$ of the uncorrupted image *f* such that the mean square error between them is minimized.

The error measure is given by

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

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

We assume that the noise and the image are uncorrelated one or the other has zero mean.

The gray levels in the estimate are a linear function of the levels in the degraded image.

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

$$= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

$$= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

Where H(u,v)= degradation function

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

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

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

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

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

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

Where K is a specified constant.

Constrained least squares filtering:

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

$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y)$ in vector-matrix form

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

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

The minimum of a criterion function C defined as

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

Subject to the constraint

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

Where $\|\mathbf{w}\|^2 \triangleq \mathbf{w}^T\mathbf{w}$ is a euclidean vector norm $\hat{\mathbf{f}}$ is estimate of the undegraded image. ∇^2 is laplacian operator.

The frequency domain solution to this optimization problem is given by

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

Where γ is a parameter that must be adjusted so that the constraint is satisfied.

P(u,v) is the Fourier transform of the laplacian operator

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