Digital Image Processing:

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used to denote the elements of a digital image.

There are three types of computerized processes in this continuum: low-, mid-, and high-level processes. Low level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. Mid-level processing on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects). Finally, higher-level processing involves "making sense" of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with vision and, in addition, encompasses processes that extract attributes from images, up to and including the recognition of individual objects. As a simple illustration to clarify these concepts, consider the area of automated analysis of text. The processes of acquiring an image of the area containing the text, preprocessing that image, extracting (segmenting) the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters are in the scope of what we call digital image processing.

Representing Digital Images:

We will use two principle ways to represent digital images. 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. For notational clarity and convenience, we shall use integer values for these discrete coordinates. Thus, the values of the coordinates at the origin are (x, y) = (0, 0). The next coordinate values along the first row of the image are represented as (x, y) = (0, 1). It is important to keep in mind that the notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure.1 shows the coordinate convention used.

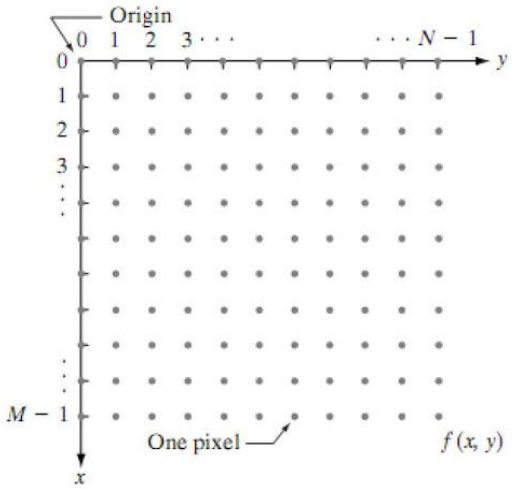


Fig 1 Coordinate convention used to represent digital images

The notation introduced in the preceding paragraph allows us to write the complete M*N digital image in the following compact matrix form:

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

The right side of this equation is by definition a digital image. Each element of this matrix array is called an image element, picture element, pixel, or pel.

Sampling and Quantization

The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

Sampling and Quantization:

The basic idea behind sampling and quantization is illustrated in Fig.2.1. Figure 2.1(a) shows 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”.

The one-dimensional function shown in Fig.2.1 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig. 2.1(a).The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig.2.1 (c).The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of Fig. 2.1 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray

levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig.2.1 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

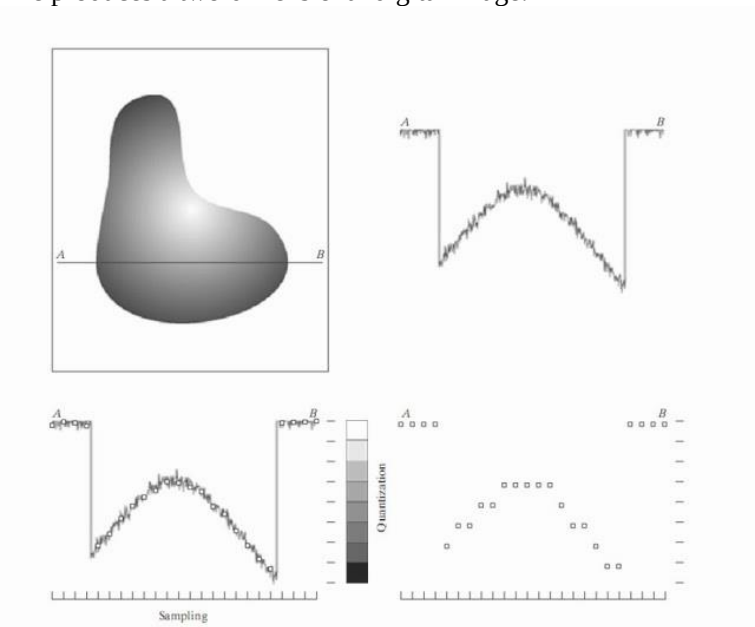
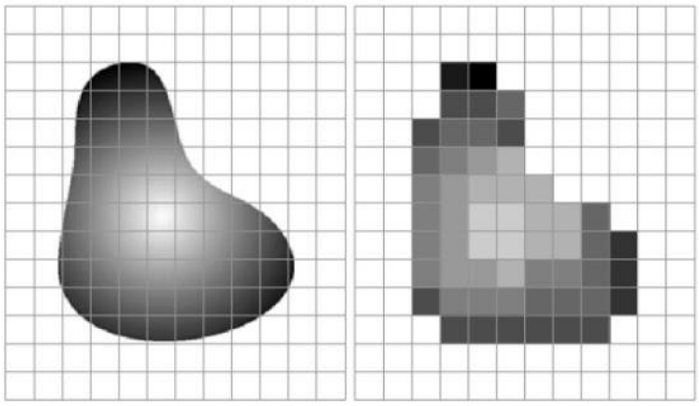


Fig.2.1. Generating a digital image (a) Continuous image (b) A scan line from A to Bin the continuous image, used to illustrate the concepts of sampling and quantization (c) Sampling and quantization. (d) Digital scan line

when a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Figure 2.2 illustrates this concept. Figure 2.2 (a) shows a continuous image projected onto the plane of an array sensor. Figure 2.2 (b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization.



a b

Fig.2. (a) Continuous image projected onto a sensor array (b) Result of image sampling and quantization.

Relationships between pixels

i) Neighbors of a pixel **ii) Connectivity**

iii) Distance measures **iv) Arithmetic& logical operations**

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). This set of pixels, called the 4-neighbors of p, is denoted by N_4 (p). Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) and are denoted by N_D (p). These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N_8 (p). 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.

Connectivity:

Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To establish if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal). For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are said to be connected only if they have the same value.

Let V be the set of gray-level values used to define adjacency. In a binary image, V={1} if we are referring to adjacency of pixels with value 1. In a grayscale image, the idea is the same, but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible gray-level values 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency:

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

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

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

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

(ii) q is in N_D (p) and the set has no pixels whose values are from V.

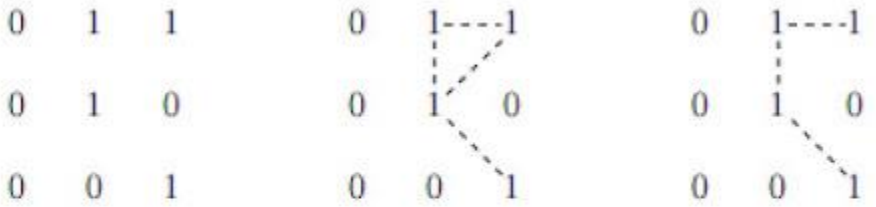
Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used. For example, consider the pixel arrangement shown in Fig.3 (a) for V= {1}.The three pixels at the top of Fig.3 (b) show multiple (ambiguous) 8-adjacency, as indicated by the dashed lines. This ambiguity is removed by using m-adjacency, as shown in Fig. 3 (c).Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to some pixel in S_2. It is understood here and in the following definitions that adjacent means 4-, 8-, or m-adjacent. A (digital) path (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates (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_iy_i) and (x_{i-1},y_{i-1}) are adjacent for $1<=i<=n$. In this case, n is the length of the path. If $(x_0, y_0) = (x_n, y_n)$, the path is a closed path. We can

define 4-, 8-, or m-paths depending on the type of adjacency specified. For example, the paths shown in Fig. 3 (b) between the northeast and southeast points are 8-paths, and the path in Fig. 3 (c) is an m-path. Note the absence of ambiguity in the m-path. 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.

Let R be a subset of pixels in an image. We call R a region of the image if R is a connected set. 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 (which we recall is a rectangular set of pixels), then its boundary is defined as the set of pixels in the first and last rows and columns of the image. This extra definition is required because an image has no neighbors beyond its border. Normally, when we refer to a region, we are referring to a subset



a b c

Fig.3 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency 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 pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a distance function or metric if

(a) $D(p, q) \geq 0 \quad (D(p, q) = 0 \quad \text{iff} \quad p = q)$,
(b) $D(p, q) = D(q, p)$, and
(c) $D(p, z) \leq D(p, q) + D(q, z)$.

The **Euclidean distance** between p and q is defined as

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

For this distance measure, the 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 **D4 distance (also called city-block distance)** between p and q is defined as

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

In this case, the pixels having a D_4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D4 distance. 2 from (x, y) (the center point) form the following contours of constant distance:

		2		
	2	**1**	**2**	
2	**1**	**0**	**1**	**2**
	2	**1**	**2**	
		2		

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

The **D8 distance (also called chessboard distance)** between p and q is defined as

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

In this case, the pixels with D8 distance from(x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D8 distance ≤ 2 from(x, y) (the center point) 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

The pixels with D_8=1 are the 8-neighbors of (x, y). Note that the D_4 and D_8 distances between p and q are independent of any paths that might exist between the points because these distances involve only the coordinates of the points. If we elect to consider m-adjacency, however, the Dm distance between two points 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. For instance, 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 a value of 0 or 1:

	p_3	p_4
p_1	p_2	
p		

Suppose that we consider adjacency of pixels valued 1 (i.e. = {1}). If p_1 and p_3 are 0, the length of the shortest m-path (the Dm distance) between p and p_4 is 2. If p_1 is 1, then p_2 and p will no longer be m-adjacent (see the definition of m-adjacency) and the length of the shortest m-path becomes 3 (the path goes through the points $pp_1p_2p_4$). Similar comments apply if p_3 is 1 (and p_1 is 0); in this case, the length of the shortest m-path also is 3. Finally, if both p_1 and p_3 are 1 the length of the shortest m-path between p and p_4 is 4. In this case, the path goes through the sequence of points **p p_1 p_2 p_3 p_4.**

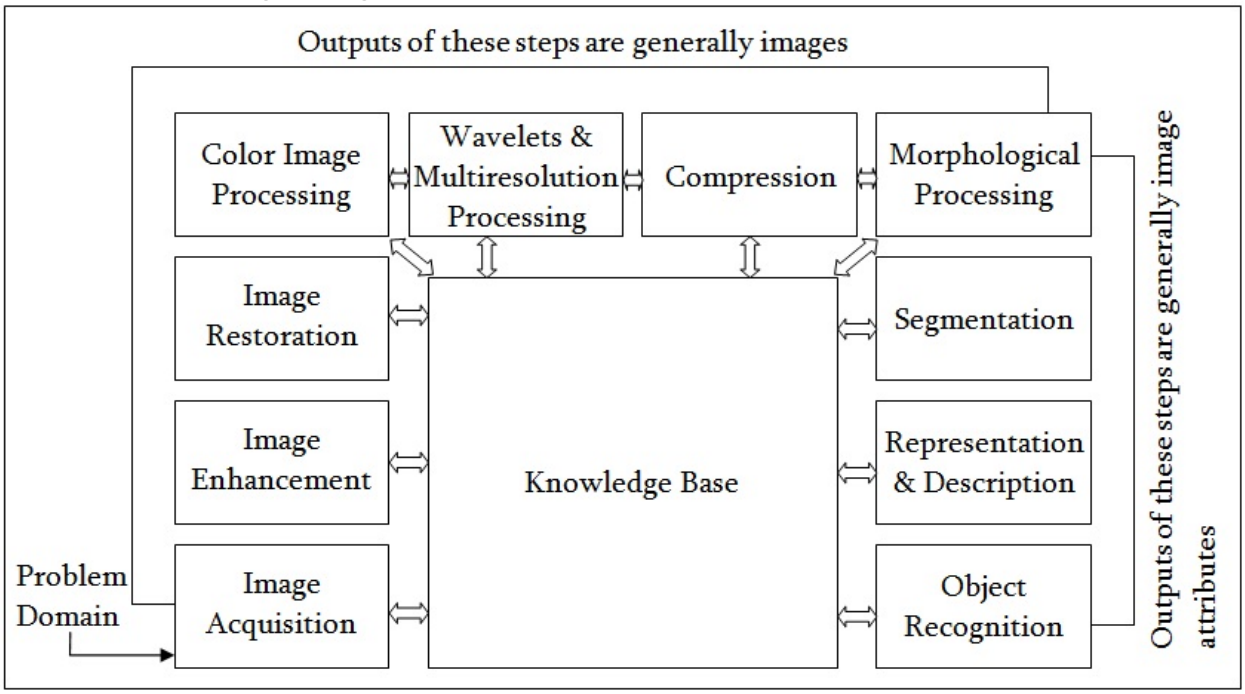
Fundamental steps in digital image processing

Ans: Image acquisition is the first process shown in Fig.4. Note that acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

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

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



Wavelets are the foundation for representing images in various degrees of resolution. Compression, as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard

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

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

Representation and Description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object Recognition is the process that assigns a label (e.g., "vehicle") to an object based on its descriptors. We conclude our coverage of digital image processing with the development of methods for recognition of individual objects.

Applications of Digital Image Processing

Ans: Digital Image Processing has very wide applications and almost all of the technical fields are impacted. The major applications of Digital Image Processing are

- Image sharpening and restoration
- Medical field
- Remote sensing
- Transmission and encoding
- Machine/Robot vision
- Color processing
- Pattern recognition
- Video processing
- Microscopic Imaging
- Others

Image sharpening and restoration

Image sharpening and restoration refers here to process images that have been captured from the modern camera to make them a better image or to manipulate those images in way to achieve desired result. This includes Zooming, blurring , sharpening , gray scale to color conversion, detecting edges and vice versa , Image retrieval and Image recognition.

Medical field
The common applications of DIP in the field of medical is

- Gamma ray imaging
- PET scan
- X Ray Imaging
- Medical CT
- UV imaging

Remote sensing
The area of the earth is scanned by a satellite or from a very high ground and then it is analyzed to obtain information about it. One particular application of digital image processing in the field of remote sensing is to detect infrastructure damages caused by an earthquake.
As it takes longer time to grasp damage, even if serious damages are focused on. Since the area effected by the earthquake is sometimes so wide, that it not possible to examine it with human eye in order to estimate damages. Even if it is, then it is very hectic and time consuming procedure. So a solution to this is found in digital image processing. An image of the effected area is captured from the above ground and then it is analyzed to detect the various types of damage done by the earthquake.
The key steps include in the analysis are

- The extraction of edges
- Analysis and enhancement of various types of edges

Transmission and encoding
The very first image that has been transmitted over the wire was from London to New York via a submarine cable. The picture that was sent took three hours to reach from one place to another. Today we are able to see live video feed, or live CCTV footage from one continent to another with just a delay of seconds. It means that a lot of work has been done in this field too. This field does not only focus on transmission, but also on encoding. Many different formats have been developed for high or low bandwidth to encode photos and then stream it over the internet or etc.

Machine/Robot vision
Apart from the many challenges that a robot face today, one of the biggest challenge still is to increase the vision of the robot. Make robot able to see things, identify them, identify the hurdles etc. Much work has been contributed by this field and a complete other field of computer vision has been introduced to work on it.

- **Hurdle detection:** Hurdle detection is one of the common tasks that has been done through image processing, by identifying different type of objects in the image and then calculating the distance between robot and hurdles.

- **Line follower robot:** Most of the robots today work by following the line and thus are called line follower robots. This helps a robot to move on its path and perform some tasks. This has also been achieved through image processing.

Color processing

Color processing includes processing of colored images and different color spaces that are used. For example RGB color model, YCbCr, HSV. It also involves studying transmission, storage, and encoding of these color images.

Pattern recognition

Pattern recognition involves study from image processing and from various other fields that includes machine learning (a branch of artificial intelligence). In pattern recognition, image processing is used for identifying the objects in an image and then machine learning is used to train the system for the change in pattern. Pattern recognition is used in computer aided diagnosis, recognition of handwriting, recognition of images etc

Video processing

A video is nothing but just the very fast movement of pictures. The quality of the video depends on the number of frames/pictures per minute and the quality of each frame being used. Video processing involves noise reduction, detail enhancement , motion detection , frame rate conversion , aspect ratio conversion , color space conversion etc.

Formation of an Image:

We denote images by two-dimensional functions of the form f (x,y). The value of f at coordinates (x, y) is a positive scalar quantity whose physical meaning is determined by the source of the image. The function f(x, y) must be nonzero and finite $0 < f(x, y) < \infty$.
The function f (x, y) may be characterized by two components:

(1) The amount of source illumination incident on the scene being viewed;
(2) The amount of illumination reflected by the objects in the scene.

These are called illumination and reflectance components and are denoted by i(x, y) and r(x, y). These two functions combine to form f (x, y):

$$f(x, y) = i(x, y) * r(x, y)$$

Where $0 < i(x, y) < \infty$ and $0 < r(x, y) < 1$

which means that reflectance is bounded by 0 (total absorption) and 1 (total reflectance).

Different components used in digital image processing system.

Figure 7 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is discussed in the following paragraphs, starting with image sensing.

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. One example of how an ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle.

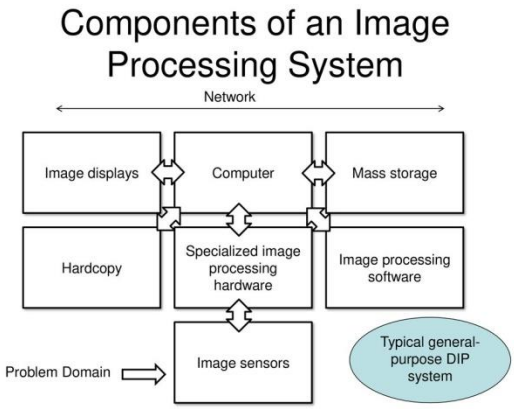


Fig.7. Components of a general purpose Image Processing System

The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, some times specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

Mass storage capability is a must in image processing applications. An image of size 1024*1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images,

providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories:
(1) short-term storage for use during processing,
(2) on-line storage for relatively fast re-call, and
(3) archival storage, characterized by infrequent access. Storage is measured in bytes (eight bits), Kbytes (one thousand bytes), Mbytes (one million bytes), Gbytes (meaning giga, or one billion, bytes), and T bytes (meaning tera, or one trillion, bytes).

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers, that store one or more images and can be accessed rapidly, usually at video rates (e.g., at 30 complete images per second).The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig.7.Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by massive storage requirements but
infrequent need for access. Magnetic tapes and optical disks housed in “jukeboxes” are the usual media for archival applications.

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

Networking 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 is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient. Fortunately, this situation is improving quickly as a result of optical fiber and other broadband technologies.

Imaging Geometry

Geometric operation transforms image *I* to new image *I'* by modifying **coordinates of image pixels**

$$I(x, y) \rightarrow I'(x', y')$$

Intensity value originally at (x,y) moved to new position (x',y')

- Filters, point operations change intensity
- Pixel position (and geometry) unchanged
- Geometric operations: change image geometry
  - **Examples:** translating, rotating, scaling an image
- Example applications of geometric operations:
- Zooming images, windows to arbitrary size
- Computer graphics: deform textures and map to arbitrary surfaces

Translation: (shift) by a vector (d_x, d_y)

$$\begin{matrix} T_x: x' = x + d_x \\ T_y: y' = y + d_y \end{matrix} \quad or \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

Scaling: (contracting or stretching) along x or y axis by a factor s_x or s_y

$$\begin{matrix} T_x: x' = s_x . x \\ T_y: y' = s_y . y \end{matrix} \quad or \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix}$$

Shearing: along x and y axis by factor b_x and b_y

$$\begin{matrix} T_x: x' = x + b_x . y \\ T_y: y' = y + b_y . x \end{matrix} \quad or \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix}$$

Rotation: the image by an angle α

$$\begin{matrix} T_x: x' = x . cos\ \alpha - y . \sin \alpha \\ T_y: y' = x . \sin \alpha + y . \cos \alpha \end{matrix} \quad or \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\ \alpha & -\sin \alpha \\ \sin \alpha & cos\ \alpha \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix}$$

Image warping: we can use a function to select which pixel somewhere else in the image to look up

For example: apply function on both texel coordinates (x, y)

$x = x + y * \sin(p * x)$

Twirl

Notation: Instead using texture colors at (x',y'), use texture colors at twirled (x,y) location

- Rotate image by angle $\boldsymbol{\alpha}$ at center or anchor point (x_c,y_c)
- Increasingly rotate image as radial distance $\boldsymbol{r}$ from center increases (up to $\boldsymbol{r_{max}}$)
- Image unchanged outside radial distance $\boldsymbol{r_{max}}$

$$T_x^{-1}: x = \begin{cases} x_c + r.\cos\beta & for\ r \le r_{max} \\ x' & for\ r > r_{max} \end{cases}$$

$$T_y^{-1}: y = \begin{cases} y_c + r.\sin\beta & for\ r \le r_{max} \\ y' & for\ r > r_{max} \end{cases}$$

With

$$d_x = x' - x_c, \qquad r = \sqrt{d_x^2 + d_y^2},$$

$$d_y = y' - y_c, \qquad \beta = Arctan(d_x.d_y) + \alpha.\left(\frac{r_{max} - r}{r_{max}}\right)$$

Module II

Image Transformations:

2-D DCT for N=4

Ans: The kernel for 2D DCT is

$$F(k,l) = \propto (k) \propto (l) \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} f(m,n) cos\left[\frac{(2m+1)\pi k}{2N}\right] cos\left[\frac{(2n+1)\pi l}{2N}\right]$$

Where

$$\propto (k) = \begin{cases} \sqrt{\frac{1}{N}} & if\ k = 0 \\ \sqrt{\frac{2}{N}} & if\ k \neq 0 \end{cases}$$

$$\propto (l) = \begin{cases} \sqrt{\frac{1}{N}} & if\ k = 0 \\ \sqrt{\frac{2}{N}} & if\ k \neq 0 \end{cases}$$

The inverse 2D DCT transform kernel is

$$f(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \propto (k) \propto (l) F(k,l) cos\left[\frac{(2m+1)\pi k}{2N}\right] cos\left[\frac{(2n+1)\pi l}{2N}\right]$$

By using separability property the N=4 the DCT Matrix can be computed using1D DCT kernel

Can be written as

$$X(k) = \propto (k) \sum_{n=0}^{N-1} x[n] \cos\left[\frac{(2n+1)\pi k}{2N}\right], where\ 0 \leq k \leq N-1$$

$$\propto (k) = \begin{cases} \sqrt{\frac{1}{N}} & if\ k = 0 \\ \sqrt{\frac{2}{N}} & if\ k \neq 0 \end{cases}$$

N=4

$$X(k) = \propto (k) \sum_{n=0}^{3} x[n] \cos\left[\frac{(2n+1)\pi k}{2X4}\right], where\ 0 \leq k \leq 3$$

If k=0

$$X(0) = \propto (0) \sum_{n=0}^{3} x[n] \cos\left[\frac{(2n+1)\pi X0}{2X4}\right]$$

$$X(0) = \sqrt{\frac{1}{4}} X \sum_{n=0}^{3} x[n] \cos[0]$$

$$X(0) = \frac{1}{2}x(0) + \frac{1}{2}x(1) + \frac{1}{2}x(2) + \frac{1}{2}x(3) - - - - - -(1)$$

If k=1

$$X(1) = \propto (1) \sum_{n=0}^{3} x[n] \cos\left[\frac{(2n+1)\pi X1}{2X4}\right]$$

$$X(1) = \sqrt{\frac{2}{4}} X \sum_{n=0}^{3} x[n] \cos\left[\frac{(2n+1)\pi}{8}\right]$$

$$X(1) = \sqrt{\frac{1}{2}} X \sum_{n=0}^{3} x[n] \cos\left[\frac{(2n+1)\pi}{8}\right]$$

$$X(1) = \sqrt{\frac{1}{2}}\left\{x(0)Xcos\left(\frac{\pi}{8}\right) + x(1)Xcos\left(\frac{3\pi}{8}\right) + x(2)Xcos\left(\frac{5\pi}{8}\right) + x(3)Xcos\left(\frac{7\pi}{8}\right)\right\}$$

$$X(1) = \sqrt{\frac{1}{2}}\{x(0)X0.9239 + x(1)X0.3827 + x(2)X(-0.3827) + x(3)X(-0.9239)\}$$

$$X(1) = 0.6532\,x(0) + 0.2706x(1) - 0.2706\,x(2) - 0.6532\,x(3) - - - -(2)$$

If k=2

$$X(2) = \propto (2)\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)\pi X2}{2X4}\right]$$

$$X(2) = \sqrt{\frac{2}{4}}X\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)2\pi}{8}\right]$$

$$X(2) = \sqrt{\frac{1}{2}}X\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)2\pi}{8}\right]$$

$$X(2) = \sqrt{\frac{1}{2}}\left\{x(0)Xcos\left(\frac{2\pi}{8}\right) + x(1)Xcos\left(\frac{6\pi}{8}\right) + x(2)Xcos\left(\frac{10\pi}{8}\right) + x(3)Xcos\left(\frac{14\pi}{8}\right)\right\}$$

$$X(2) = \sqrt{\frac{1}{2}}\left\{x(0)X\sqrt{\frac{1}{2}} - x(1)X\sqrt{\frac{1}{2}} - x(2)X\sqrt{\frac{1}{2}} + x(3)X\sqrt{\frac{1}{2}}\right\}$$

$$X(2) = 0.5\,x(0) - 0.5x(1) - 0.5\,x(2) + 0.5\,x(3) - - - -(3)$$

If k=3

$$X(3) = \propto (3)\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)\pi X3}{2X4}\right]$$

$$X(3) = \sqrt{\frac{2}{4}}X\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)3\pi}{8}\right]$$

$$X(3) = \sqrt{\frac{1}{2}}X\sum_{n=0}^{3} x[n]\cos\left[\frac{(2n+1)3\pi}{8}\right]$$

$$X(3) = \sqrt{\frac{1}{2}}\left\{x(0)Xcos\left(\frac{3\pi}{8}\right) + x(1)Xcos\left(\frac{9\pi}{8}\right) + x(2)Xcos\left(\frac{15\pi}{8}\right) + x(3)Xcos\left(\frac{21\pi}{8}\right)\right\}$$

$$X(3) = \sqrt{\frac{1}{2}}\{x(0)X0.3827 + x(1)X(-0.9239) + x(2)X0.9239 + x(3)X(-0.3827)\}$$

$$X(3) = 0.2706x(0) - 0.6533x(1) + 0.6533\,x(2) - 0.2706\,x(3) - - - -(4)$$

Collecting the coefficients x(0),x(1), x(2), x(3) from X(0), X(1), X(2), X(3), we get

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

Separability Property of 2D-DFT

The separability property of 2D-DFT states that, the discrete Fourier transform pair can be expressed in the separable forms. i.e.,

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} \exp\left[-\frac{j2\pi ux}{N}\right] \sum_{y=0}^{N-1} f(x,y) \exp\left[-\frac{j2\pi vy}{N}\right]$$

for u, v = 0, 1, 2, ….,N-1 and

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1} \exp\left[\frac{j2\pi ux}{N}\right] \sum_{v=0}^{N-1} F(u,v) \exp\left[\frac{j2\pi vy}{N}\right]$$

The principal advantage of the separability property is that F(u,v) or f(x,y) can be obtained in two steps by successive applications of the 1-D Fourier transform or its inverse. This advantage becomes evident.

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v) exp\left[-\frac{j2\pi ux}{N}\right]$$

$$F(x,v) = N\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y) exp\left[-\frac{j2\pi vy}{N}\right]\right]$$

For each value of x, the expression inside the brackets in above equation is a 1-D transform, with frequency values v = 0, 1, . . ., N-1. Therefore the 2-D function f(x, v) is obtained by taking a transform along each row of f(x, y) and multiplying the result by N. The desired result, F(u, v), is then obtained by taking a transform along each column of F(x, v).

Walsh Transform of 2X2 Image

Consider the Image Matrix is F = $\begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix}$

The Walsh transform for N=2 is

$$W_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$T=WFW^T$

$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & -1 \\ 6 & 2 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} 9 & 1 \\ -3 & -3 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} 10 & 8 \\ 6 & 0 \end{bmatrix}$$

Hence the Walsh Transformed image is $T = \begin{bmatrix} 5 & 4 \\ -3 & 0 \end{bmatrix}$

KL Transform

The **KL Transform** is also known as Hotelling Transform. It is not having any mathematical structure.

Step 1: Formulation of vectors from given matrix

Given Image $X = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}$

$\boldsymbol{x_0} = \begin{bmatrix} \mathbf{4} \\ \mathbf{-1} \end{bmatrix}, \boldsymbol{x_1} = \begin{bmatrix} \mathbf{-2} \\ \mathbf{3} \end{bmatrix}$

Step 2: Determination of Covariance Matrix

The formula to find the covariance of the image is

$$cov(X) = E[X.X^T] - \bar{x}.\bar{x}^T$$

$\bar{x}$ indicates the mean value of x0 and x1

It can be written as
Mean of the input image

$$(\bar{x}) = \frac{1}{M}\sum_{k=0}^{M-1} x_k$$

$$= \frac{1}{2}\{x_0 + x_1\} = \frac{1}{2}\left\{\begin{bmatrix} 4 \\ -1 \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix}\right\}$$

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

$$\bar{x}.\bar{x}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}[1 \quad 1] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$E[XX^T] = \frac{1}{2}\sum_{i=0}^{2-1} x_i x_i^T$$

$$= \frac{1}{2}\{x_0 x_0^T + x_1 x_1^T\}$$

$$= \frac{1}{2}\left\{\begin{bmatrix} 4 \\ -1 \end{bmatrix}[4 \quad -1] + \begin{bmatrix} -2 \\ 3 \end{bmatrix}[-2 \quad 3]\right\} = \frac{1}{2}\left\{\begin{bmatrix} 16 & -4 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 4 & -6 \\ -6 & 9 \end{bmatrix}\right\}$$

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

$$cov(X) = E[XX^T] - \bar{x}.\bar{x}^T$$

$$= \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix}$$

The Covariance of image X is $= \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix}$

Step 3: Determination of Eigen Values of the covariance image

$$|cov(X) - \lambda I| = 0$$

$$\left|\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right| = 0$$

$$\begin{vmatrix} 1-\lambda & -2 \\ -2 & -\lambda \end{vmatrix} = 0$$

$$==> \lambda^2 - \lambda - 4 = 0$$

The λ values are λ_0=2.5615, λ_1=-1.5615

Step 4: Determination of Eigen vectors

$$(cov(X) - \lambda I)A = 0$$

If λ_0=2.5615 then $\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} 2.5615 & 0 \\ 0 & 2.5615 \end{bmatrix}\begin{bmatrix} a_{00} \\ a_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\begin{bmatrix} 1.5615 & -2 \\ -2 & 2.5615 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

By solving a_{00}=-1.2808, a_{01}=1

Similarly λ_1=-1.5615 then $\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} -1.5615 & 0 \\ 0 & -1.5615 \end{bmatrix} \begin{bmatrix} a_{01} \\ a_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\begin{bmatrix} 2.5615 & -2 \\ -2 & 1.5615 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

By solving a_{01}=0.7808, a_{11}=1

Step 5: Normalization of the Eigen vectors

$$\frac{A_0}{\|A_0\|} = \frac{1}{\sqrt{a_{00}^2 + a_{01}^2}} \begin{bmatrix} a_{00} \\ a_{01} \end{bmatrix} = \frac{1}{\sqrt{1.2808^2 + 1^2}} \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.7882 \\ 0.6154 \end{bmatrix}$$

similarly

$$\frac{A_1}{\|A_1\|} = \frac{1}{\sqrt{a_{10}^2 + a_{11}^2}} \begin{bmatrix} a_{10} \\ a_{11} \end{bmatrix} = \frac{1}{\sqrt{0.7808^2 + 1^2}} \begin{bmatrix} 0.7808 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.6154 \\ 0.7882 \end{bmatrix}$$

Step 6: Formation of KL Transformation Matrix

$$A = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}$$

According orthogonality principle $AA^T = AA^{-1} = I$

The obtained matrix is satisfying above condition

Step 7: KL Transformation is applied to input image

$$Y_0 = A[x_0] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix}$$

$$Y_0 = A[x_1] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 3.4224 \\ 1.1338 \end{bmatrix}$$

Then $Y = \begin{bmatrix} -3.7682 & 3.4224 \\ 1.6734 & 1.1338 \end{bmatrix}$

Step 8: Reconstruction of input values from the transformed coefficients

$$X = T^T Y$$

$$X = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -3.7682 & 3.4224 \\ 1.6734 & 1.1338 \end{bmatrix}$$

$$X = \begin{bmatrix} 3.9999 & -1.9999 \\ -1 & 2.9999 \end{bmatrix}$$

Slant Transform matrix for the order N=4

1) The Slant Transform is real and orthogonal.

2) The Slant Transform is a fast transform, which will be implemented on $O(N \log_2 N)$ operations on an N X 1 vector

3) It has very good to excellent energy compaction for images.

$$S_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$S_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ a_2 & b_2 & -a_2 & b_2 \\ 0 & 1 & 0 & -1 \\ -b_2 & a_2 & b_2 & a_2 \end{bmatrix} \begin{bmatrix} S_1 & S_1 \\ S_1 & S_1 \end{bmatrix}$$

$$a_1 = 1$$

$$b_n = (1 + 4{a_{n-1}}^2)^{\frac{1}{2}}$$

$$a_n = 2 b_n a_{n-1}$$

Which also

$$N = 2^n, \qquad a_{n+1} = \left(\frac{3N^2}{4N^2 - 1} \right)^{1/2}$$

$$b_{n+1} = \left(\frac{N^2 - 1}{4N^2 - 1}\right)^{1/2}$$

$$S_2 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 0 & 1 & 0 & -1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix}$$

Hadamard Transform and Walsh Transform Matrix for the order N=4

The 1-D Hadamard transform is defines as

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

The inverse Hadamard transform is

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

The higher order Hadamard transformation matrices can be formulated using a simple recursive relationship.

The Hadamard matrix of lowest order, N=2 is

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

If H_N represent the matrix of order N, the recursive relationship is given by the expression

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} H_{2N}\ is\ the\ Hadamard\ matrix\ of\ order\ 2N$$

Thus H_4 matrix can be formed as follows

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Walsh Transform

For N = 4; n = 2.

The basis function h(x,u) is found from

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

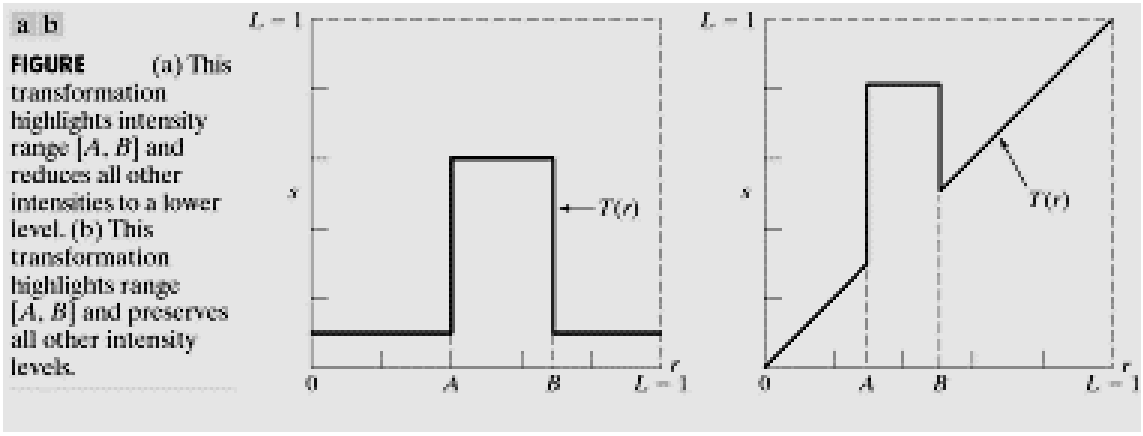
x = 00, 01, 10, 11

u = 00, 01, 10, 11

Gray level slicing and bit plane slicing

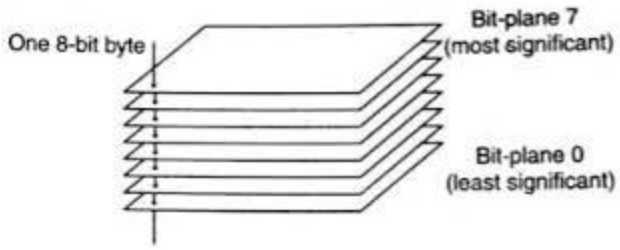
Gray-level slicing:

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. 1.4 (a), produces a binary image. The second approach, based on the transformation shown in Fig. 1.4 (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure 1.4(c) shows a gray-scale image, and Fig. 1.4 (d) shows the result of using the transformation in Fig. 1.4 (a).Variations of the two transformations shown in Fig. 1.4 are easy to formulate.



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 illustrates these ideas, and Figure shows the various bit planes for the image shown in Figure 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 analysing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.



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

Image Enhancement in Frequency Domain

The frequency domain methods of image enhancement are based on convolution theorem. This is represented as,

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

Where $g(x, y)$ = Resultant image

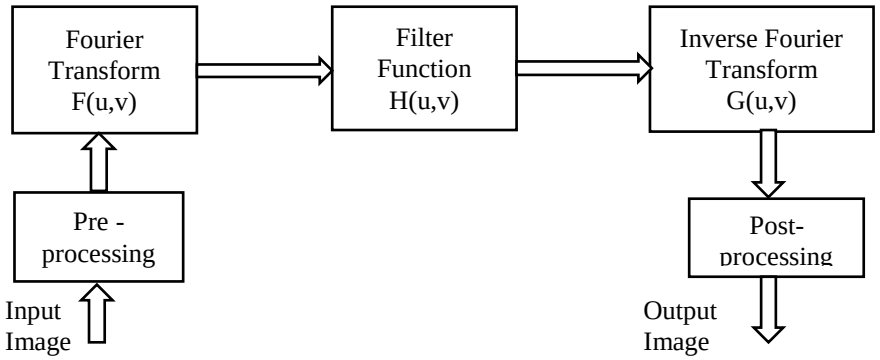
$h(x, y)$ = Position invariant operator

$f(x, y)$= Input image

The Fourier transform representation of equation above is,

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

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



1. Given an input image f(x, y) of size M × N, compute padding parameters as

P = 2M, Q = 2N

2. Form a padded image fp(x, y) of size P × Q by appending the necessary number of zeros to f(x, y)

3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center the transform

4. Compute DFT of the image F(u, v)

5. Multiply F(u, v) by a filter function H(u, v)

H(u, v) is a symmetric function of size P × Q with center at (P/2, Q/2)

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

Multiplication of H and F is only between corresponding elements

6. Compute inverse DFT of the result as

$$g_p(x, y) = \text{inverse transform } [G(u, v)]$$

7. Obtain the real part of the result
8. Multiply the result by $(-1)^{x+y}$
9. Obtain the final processed results g(x, y) by extracting the M × N region from the top, left quadrant of $g_p(x, y)$.

Derivative filters:

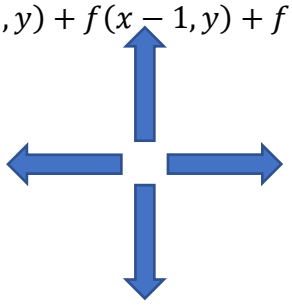
The 1st -order derivative is nonzero along the entire ramp, while the 2nd -order derivative is nonzero only at the onset and end of the ramp. The first order derivatives are providing the thick edge, the second order derivative make thin edge. The response at and around the point is much stronger for the 2nd - than for the 1st order derivative.

Shown by Rosenfeld and Kak[1982] that the simplest isotropic derivative operator is the Laplacian is defined as

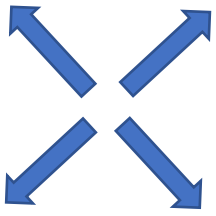
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

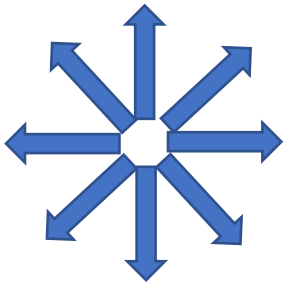
0	1	0
1	-4	1
0	1	0



1	0	1
0	-4	0
1	0	1



1	1	1
1	-8	1



1	1	1

Principle of high pass and high boost filtering methods.

All the filtered images have one thing in common: Their average background intensity has been reduced to near black. This is due to the fact that the high pass filters we applied to those images eliminate the zero-frequency component of their Fourier transforms. In fact, enhancement using the Laplacian does precisely this, by adding back the entire image to the filtered result. Sometimes it is advantageous to increase the contribution made by the original image to the overall filtered result. This approach, called high-boost filtering, is a generalization of unsharp masking. Unsharp masking consists simply of generating a sharp image by subtracting from an image a blurred version of itself. Using frequency domain terminology, this means obtaining a high pass-filtered image by subtracting from the image a lowpass-filtered version of itself. That is

$$f_{hp}(x,y) = f(x,y) - f_{lp}(x,y)$$

High-boost filtering generalizes this by multiplying f (x, y) by a constant A > 1:

$$f_{hb} = Af(x,y) - f_{hp}(x,y)$$

Thus, high-boost filtering gives us the flexibility to increase the contribution made by the image to the overall enhanced result. This equation may be written as

$$f_{hb} = (A-1)f(x,y) + f(x,y) - f_{lp}(x,y)$$

$$f_{hb} = (A-1)f(x,y) + f_{hp}(x,y)$$

This result is based on a high pass rather than a lowpass image. When A = 1, high-boost filtering reduces to regular high pass filtering. As A increases past 1, the contribution made by the image itself becomes more dominant.

$$F_{hp}(u,v) = F(u,v) - F_{lp}(u,v)$$

$$F_{lp}(u,v) = F(u,v)H_{lp}(u,v)$$

where H_{lp} is the transfer function of a lowpass filter. Therefore, unsharp masking can be implemented directly in the frequency domain by using the composite filter,

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

Similarly, high-boost filtering can be implemented with the composite filter

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

with A > 1. The process consists of multiplying this filter by the (centred) transform of the input image and then taking the inverse transform of the product. Multiplication of the real part of this result by $(-l)^{x+y}$ gives us the high-boost filtered image f_{hb} (x, y) in the spatial domain.

Compute the median value of the marked pixels using 3X3 mask and replace them with calculated values, From the image

$$f(x,y) = \begin{bmatrix} 4 & 20 & 12 & \mathbf{25} & 30 & 10 \\ 5 & 18 & 25 & 30 & 16 & 9 \\ 5 & \mathbf{14} & 4 & 20 & 17 & 15 \\ 15 & 13 & 14 & 12 & 25 & \mathbf{7} \end{bmatrix}$$

Ans:

$\begin{bmatrix} 5 & 18 & 25 \\ 5 & 14 & 4 \\ 15 & 13 & 14 \end{bmatrix}$ Place these values in ascending or descending order

~~4~~,~~5~~,~~5~~,~~13~~,14,~~14~~,~~15~~,~~18~~,~~25~~

The median value is 14.

$\begin{bmatrix} 0 & 0 & 0 \\ 12 & 25 & 30 \\ 25 & 30 & 16 \end{bmatrix}$ Place these values in ascending or descending order

~~0~~,~~0~~,~~0~~,~~12~~,16,~~25~~,~~25~~,~~30~~,~~30~~

The median value is 16

$\begin{bmatrix} 17 & 15 & 0 \\ 25 & 7 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ Place these values in ascending or descending order

~~0~~,~~0~~,~~0~~,~~0~~,0,~~7~~,~~15~~,~~17~~,~~25~~

The median value is 0

Then

$$f(x,y) = \begin{bmatrix} 4 & 20 & 12 & \mathbf{16} & 30 & 10 \\ 5 & 18 & 25 & 30 & 16 & 9 \\ 5 & \mathbf{14} & 4 & 20 & 17 & 15 \\ 15 & 13 & 14 & 12 & 25 & \mathbf{0} \end{bmatrix}$$

Apply the 3X3 lowpass spatial filtering mask for the below image and explain the smoothing spatial filtering

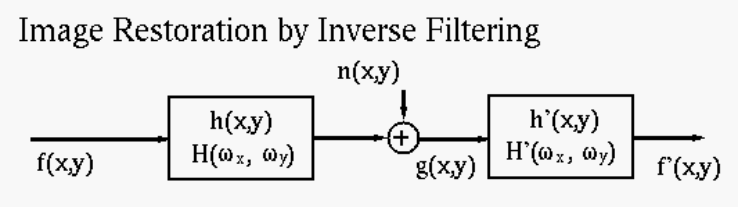
Image: $f(x,y) = \begin{bmatrix} 3 & 4 & 7 & 5 & 8 \\ 3 & 0 & 9 & 0 & 0 \\ 0 & 1 & 1 & 12 & 0 \\ 7 & 2 & 0 & 0 & 1 \\ 0 & 3 & 0 & 3 & 0 \end{bmatrix}$,

$$and\ w(x,y) = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Answer refer class notes

Inverse filtering:

The purpose of image restoration is to estimate or recover the scene without image degradation or distortion caused by non-ideal image system (e.g. the optics of the camera system). Inverse filtering is one of the techniques used for image restoration to obtain a recovered image $f'(x,y)$from the image data $g(x,y)$so that $f(x,y) = f'(x,y)$ in the ideal situation $n(x,y) = 0$ and $h'(x,y) * h(x,y) = \delta(x,y)$ or $H(\omega_x, \omega_y)H'(\omega_x, \omega_y) = 1$.



The camera system for image acquisition can be modelled mathematically by

$$g(x,y) = \int_0^T \int \int_{-\infty}^{\infty} h(x,y,x',y',t) f(x',y',t) dx' dy' dt + n(x,y)$$

where T is the exposure time, $n(x,y)$ is some additive noise, and $h(x,y,x',y',t)$ is a function characterizing the distortion introduced by the imaging system, caused by, for example, limited aperture, out of focus, random atmospheric turbulence, and/or relative motion. If the system is ideal, spatial and time invariant, and noise-free, i.e.,

$$h(x,y,x',y',t) = \delta(x - x', y - y')$$

then the imaging process becomes

$$g(x,y) = \int_0^T f(x,y,t)dt$$

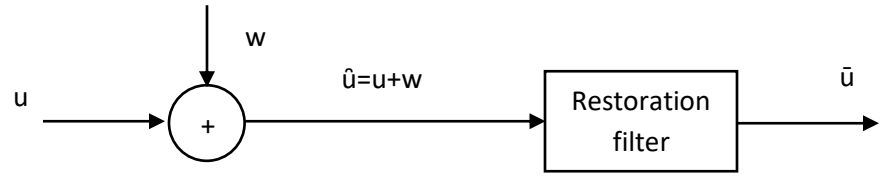
If the signal is also time invariant (a stationary scene), i.e., $f(x,y,t) = f(x,y)$, the image obtained is simply

$$g(x,y) = T\ f(x,y)$$

The inverse filtering is very sensitive to noise presence and difficult it built.

Image Degradation/Restoration Process.

In practice, an image may be degraded by various types and forms of noise. However, the most common type of noise is the additive one. As the following figure shows, the degradation process is modelled as an additive noise term, 'w' which operates on an input image, 'u', to produce a degraded image 'û'. Given this noisy observation, along with some knowledge of the additive noise term, the restoration technique yields an estimate 'û, of the original image. The denoised estimate is desired to be as close as possible to original image 'u'.



Noise Model The principal source of noise in digital images arises during image acquisition (digitization) or transmission. The performance of imaging sensors is affected by a variety of factors, such as the environmental conditions during image acquisition, and by the quality of the sensing elements themselves. For instance, in acquiring images with a camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are also corrupted during transmission principally due to interference in the channel used for transmission. For example, an image transmitted using a wireless network might be corrupted as a result of lighting or other atmospheric disturbance.

Gaussian

Noise Gaussian noise is statistical noise that has its probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution. In other words, the values that the noise can take on are Gaussian-distributed. A special case is white Gaussian

noise, in which the values at any pairs of times are statistically independent (and uncorrelated). In applications, Gaussian noise is most commonly used as additive white noise to yield additive white Gaussian noise. Distortion due to AWGN can be caused by poor quality image acquisition, images observed in a noisy environment or noise inherent in communication channels. However, because of its mathematical tractability in both the spatial and the frequency domains, Gaussian noise models are used frequently in practice. More specifically the noise, 'w', is assumed to be an additive wide-sense stationary (WSS) white Gaussian noise process with zero mean and constant variance σ_w^2 which is formed independently of the original noise-free image. Thus, if an image 'I' of size M X N pixels is defined by its gray-level function, u= $[u]_{m,n}$,m, n =1,2,3,...,M, N, then the noisy image Î is defined by the noisy gray-level function û= $[\hat{u}]_{m,n}$, m, n =1,2,3,...,M, N, as

$$\hat{u}=u+w$$

u u w ,m,n 1,2,3,...,M,N ˆ

where w ,m,n 1,2,3,...,M,N m,n are independent and identically distributed (iid) Gaussian random samples with zero mean and variance σ_w^2, that is $w_{m,n} \sim N(0, \sigma_w^2)$ for m,n =1,2,...,M, N

Salt and Pepper Noise

Salt and pepper noise is an impulse type of noise, which is also referred to as intensity spikes. This is caused generally due to errors in data transmission. It has only two possible values, a and b which represent dark and bright respectively. The probability of each is typically less than 0.1. The corrupted pixels are set alternatively to the minimum or to the maximum value, giving the image a "salt and pepper" like appearance. Unaffected pixels remain unchanged. For an 8-bit image, the typical value for pepper noise is 0 and for salt noise 255. The salt and pepper noise is generally caused by malfunctioning of pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process.

Speckle Noise

Speckle noise affects all coherent imaging systems including medical ultrasound. Within each resolution cell a number of elementary scatters reflect the incident wave towards the sensor. The backscattered coherent waves with different phases undergo a constructive or a destructive interference in a random manner. The acquired image is thus corrupted by a random granular pattern, called speckle that hinders the interpretation of the image content. Speckle noise is a multiplicative noise. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise.

Constrained Restoration (Weiner Filtering)

In constrained restoration, some a priori knowledge of the noise is used to constrain the least square computation. The restoration is carried out using the method of Lagrange multipliers.

Wiener filtering is a statistical method for constrained restoration. It is based on the correlation matrices of the image and the noise. When there is no noise, the Wiener filter degrades to the ideal inverse filter.

The constrained least square restoration only requires knowledge of the noise's mean and variance, but the restoration is optimal for each image.

Wiener Filtering is also known as Mean Squared Error Filtering. It incorporates the both degradation function and statistical characteristics of noise. Here the assumption will be made as the noise and the image are uncorrelated. The MSE (Mean Square Error) is minimized as

$$e = \sum_x \sum_y (f(x,y) - \hat{f}(x,y))^2$$

unknown original — after Wiener filtering

$$e = MN \sum_x \sum_y |f(x,y) - \hat{f}(x,y)|^2$$

$$= \sum_u \sum_v |F(u,v) - \hat{F}(u,v)|^2 \qquad \text{Parseval's Theorem}$$

$$= \sum_u \sum_v |F(u,v) - [F(u,v)H(u,v) + N(u,v)]W(u,v))|^2$$

Unknown original — Corrupted original — Wiener filter

$$e = \sum_u \sum_v |F(u,v)[1 - H(u,v)W(u,v)] - N(u,v)W(u,v)|^2$$

$$= \sum_u \sum_v |F(u,v)[1 - H(u,v)W(u,v)]|^2 + |N(u,v)W(u,v)|^2$$

$$= \sum_u \sum_v |F(u,v)|^2 |1 - H(u,v)W(u,v)|^2 + |N(u,v)|^2 |W(u,v)|^2$$

$$\frac{\partial e}{\partial W(u,v)} = 0 \quad \Rightarrow \quad W(u,v)$$

$$\frac{\partial e}{\partial W(u,v)} = |F|^2[2(1 - W^*H^*)(-H)] + |N|^2[2W^*]$$

$$\frac{\partial e}{\partial W(u,v)} = 0 \quad \Rightarrow \quad W^*(u,v) = \frac{|F(u,v)|^2 H(u,v)}{|H(u,v)|^2 |F(u,v)|^2 + |N(u,v)|^2}$$

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}} = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}}$$

$$W(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{1}{\mathrm{SNR}}}$$

Unconstrained Restoration (Inverse Filtering)

In unconstrained restoration, no a priori knowledge about the noise is assumed. The restoration is carried out in a least squares sense of the estimation error.

Inverse filtering is a commonly used restoration approach, which can be implemented in the Fourier domain.

Removal of the blur caused by uniform linear motion is a typical application of the techniques for the unconstrained restoration in a closed form.

With the estimated degradation function H(u,v)

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

Unknown noise

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

Estimate of original image

Differences between Image Enhancement and Image Restoration.

(i) Image enhancement techniques are heuristic procedures designed to manipulate an image in order to take advantage of the psychophysical aspects of the human system. Whereas image

restoration techniques are basically reconstruction techniques by which a degraded image is reconstructed by using some of the prior knowledge of the degradation phenomenon.

(ii) Image enhancement can be implemented by spatial and frequency domain technique, whereas image restoration can be implement by frequency domain and algebraic techniques.

(iii) The computational complexity for image enhancement is relatively less when compared to the computational complexity for image restoration, since algebraic methods requires manipulation of large number of simultaneous equation. But, under some condition computational complexity can be reduced to the same level as that required by traditional frequency domain technique.

(iv) Image enhancement techniques are problem oriented, whereas image restoration techniques are general and are oriented towards modelling the degradation and applying the reverse process in order to reconstruct the original image.

(v) Masks are used in spatial domain methods for image enhancement, whereas masks are not used for image restoration techniques.

(vi) Contrast stretching is considered as image enhancement technique because it is based on the pleasing aspects of the review, whereas removal of' image blur by applying a deblurring function is considered as an image restoration technique.