CHAPTER 2

Digital Image Processing
2.1 INTRODUCTION

The real history of what we call digital goes back to the famous poem, 'Paul Revere’s ride' instructing the army men with two lanterns as on-on, on-of, of-of, conveying the message for situation. This can be considered as being the first digital signal (11, 10, 00). When America was a rebellious colony of England, Paul Revere was assigned the job of notifying the minute men in the country side if the British left Boston to attack and by what means they were coming. We may recall long fellow's poem, Paul Revere's Ride. Here are a few stanzas:

Listen, my children, and you shall hear
Of the midnight ride of Paul Revere,
On the eighteenth of April, in seventy five,
Hardly a man is now alive
Who remembers that famous day and year
He said to his friend, “If the British march
By land or sea from the town tonight,
Hang a lantern aloft in the belfry arch
Of the North Church as a signal light,
One, if by land, and two, if by sea,
And I on the opposite shore will be,
Read to ride and spread the alarm
Through every Middlesex village and farm,
For the country folk to be up and arm

One can notice here that, 'One', if by land, and 'Two', if by sea, is a digital signal message (Peter Norton, 2002). Revere's friend Robert Newman, the Sexton of Old North Church, spied on the British, then lit a lantern and hung it in the belfry arch of the church Revere, waiting on the other side of the harbor so that he had a head start on the British troops, saw the signal and began his famous ride. (Unfortunately, he was immediately
arrested and the news was delivered instead by one William Dawes; so much for legends.)

The history of the digital image processing is as old as the origin of sophisticated digital computing machines. With the advent of the third generation computer around 1960s, the area of digital image processing and analysis has emerged as a subject of interdisciplinary study and research in the fields of physics, biomedicines, engineering, meteorology, remote sensing, chemistry, space science, forensic, statistics, agriculture and of course computer science. The 2-D and 3-D image and signal processing and analysis today form a major area of research and development in the broad fields of pattern recognition, computer vision, machine learning and also of artificial intelligence and neural networking. On of the motivation behind this spurt of activity in this field is the need felt by people to communicate with computers in a natural mode of communication in respect of diverse applications.

One of the first application with digital images was in the newspaper industry, when pictures were first sent by the submarine cable between London and New York. Introduction of Bartlane cable picture transmission system in the early 1920s reduced the time required to transfer a picture across Atlantic from more than a week to less than three hours. Specialized printing equipments coded pictures for cable transmission and then reconstructed them at the receiving end. This mode of telegraphic picture transmission was well in custom in the 1921. The early Bartlane systems were capable of coding images in five distinct gray levels. The capability was increased to 15 levels in 1929.

The history of digital image processing is intimately tied to the development of digital computer. In fact the digital images require so much storage and computational
power that the progress of the digital image processing has been dependent on computing devices and of supporting technologies that include data storage, display and transmission. The idea of a computer goes back to the invention of the abacus in Asia Minor, more than 5000 years ago. More recently, there were developments in the past two centuries that are the foundation of what we call a computer today. However, the basis of what we call a modern digital computer dates back to only the 1940s with the introduction by John Von Neumann of two key concepts: (1) A memory to hold the program and data (2) Conditional branching. These two ideas are the basic foundation of central processing unit. Starting with Van Neumann, there were a series of key advances that led to computers powerful enough to be used for digital image processing. Briefly these advances may be summarized as: (1) The invention of transistor by Bell Laboratories in 1948; (2) The development of the high-level programming languages like COBOL (Common Business-Oriented Language) and FORTRAN (formula translation) in 1950s and 1960s respectively; (3) The development of Integrated Circuit (IC) in 1958 at Texas Instruments; (4) The development of operating system in the early 1960s; (5) The development of microprocessor by Intel in the early 1970s; (6) Introduction of first personal computer by IBM in 1981; (7) Increasing miniaturization, high component density, ultra high processing speed (32-bit Pentium by Intel and 64-bit Athlon by Advance Micro Devices) and surprisingly decreasing cost.

The first computer powerful enough to carry out meaningful image processing tasks appeared in the early 1960s. The birth of what we call digital image processing today can be traced to the availability of those machines and the onset of the space program during that period. It took the combination of those two developments to bring into focus the potential of digital image processing concepts. Work on using computer techniques for improving images from a space probe began at the Jet Propulsion Laboratory (Pasadena,
California) in 1964 when a picture of the moon transmitted by Ranger-7 was processed by computer to correct various types of distortions inherent in the onboard television camera. The imaging lessons learned with Ranger-7 served as the bases for improved methods used to restore and enhance images from surveyor missions to the moon, the mariner series of flyby missions to Mars, the Apollo manned flights to the moon, and others.

The image processing laboratory (IPL) at NASA's JPL began in 1966 to retrieve and process video images from spacecrafts. Specialized software called VICAR (video image communication and retrieval) was developed to process the images on IBM-360 computer. VICAR was written by Howard Frieden, Bob Nathan, Stan Bressler, John Campbell, Tom King and Ed Efron. In the mid of 1970s, time-sharing was introduced in IPL when IBM TSO was released. In the late 1970s, PDP-11 peripheral processors were added to manage image displays. The VICAR algorithm continued to be developed and improved. This phase of IPL supported the Ranger, Surveyor, Mariner (2, 4, 6, 7, 9, 10), Viking (1, 2), Voyager (Jupiter, Saturn). In 1979, IPL merged with MTIS and continued to support real-time and systematic production for Voyager.

In parallel with the space applications, digital image processing techniques began in the late 1960s and early 1970s to be used in medical imaging, remote earth resource observations and astronomy. The invention in the early 1970s of computerized axial tomography (CAT, CT) is one of the most important events in the application of image processing in the medical diagnosis. CT is a process in which a ring of detectors encircles the object (or patient) and an X-ray source concentric with the detector ring rotates about the object. A slice of the object is scanned. Combination of a number of such slices gives 3-D impression.
The motivations in the field of DIP became much more important in the early 1980s, when fifth generation computer systems/knowledge based computing systems (FGCS/KBCS) were launched in different countries like Japan, USA, Europe and also in India, with the objective of designing and making automation that can carry out certain tasks as we human being perform. As a result a new generation of applications are in the market or, are being planned not only in respect of robotics but also in respect of areas like office automation, biomedical engineering, industrial automation, meteorological predictions, high energy physics, environment and urban planning, oil and natural gas exploration, fingerprint processing, forensic investigations, restoration of images suffering from photometric and geometric distortions, multimedia and computerized video editing and video conferencing apart from several very sophisticated military applications. The result of these efforts established image processing and computer vision as one of the fastest growing technology worldwide.

2.2 IMAGE ENHANCEMENT IN SPATIAL DOMAIN

"Every image needs enhancement"

This statement reflects the importance and relevance of digital image enhancement. All kinds of raster formats whether true images or spectrographs, electrographs, sonograph, radiograph, always are in subject to enhance. The field of digital image enhancement extends from remotely sensed satellite images, telescopic space probes, medical, pathological, food quality control, forensic and microprocessor circuit design and inherent component defects and malfunctioning detection, apart from conventional photo-processing. Further the term ‘Every’ implies whether the image, unprocessed or processed, it requires some further enhancing treatments, as per the desire of a particular
image analyst. A given and processed image is said enhanced only with respect to a specific information extraction and a particular user. Thus no image is ideal or perfect.

Defects and distortions are always persisting in every format of raster kind due to inherent functional limitations of the imaging and pre-imaging tools. What actually done in digital image enhancement is to improve and upgrade the visual appearance of an image to extract maximum and optimum informations.

Vision is the most the advanced in our senses, so it is not surprising that images play the single most important role in human perception. However, unlike human who are limited to the visible band of EM spectrum, imaging mechanisms span almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that human are not accustomed to associating with images. These include ultra-sound, electron microscopy and computer-generated images. Thus digital image processing encompasses a wide and varied fields of applications.

There is no general agreement among analysts regarding where image processing stops and other related areas, such as image analysis and computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input and output of a process are images. We believe this to be a limiting and somewhat artificial boundary. For example, under this definition, even the trivial task of computing the average intensity of an image would not be considered as an image processing operation. On the other hand, there are fields such as computer vision whose ultimate goal is to use computers to emulate human vision, including learning and being able to make interfaces and take actions based on visual inputs.
The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application [25]. The word specific is important, because it establishes at the outset that image enhancing techniques are very much problem oriented. A method that is quite useful for enhancing X-ray images can result garbage to pictures of Mars transmitted by a space probe. Regardless the method used, however, image enhancement is one of the most interesting and visually appealing area of DIP. There is no general theory of image enhancement. When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works [26]. Visual evaluation of image quality is a highly subjective process, thus making the definition of a ‘good image’ an elusive standard by which to compare algorithm performance. When the problem is one of processing images for machine perception, the evaluation task is somewhat easier, i.e. the best image processing method would be the one yielding the best machine recognition [27].

Spatial domain (SD) refers to the aggregate of pixels composing an image. SD methods are procedures that operate directly on pixels and can be denoted as
\[ g(x, y) = T[f(x, y)], \]
where \( f(x, y) \) the input image, \( g(x, y) \) 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 \( (x, y) \). The centre of the subimage is moved from pixel to pixel and the operator \( T \) is applied at each location. The process utilizes only the pixels in the area spanned by the neighborhood. The simplest form of \( T \) is when the neighborhood is of size \( |x| \) (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 transformation function as \( s = T(r) \), \( r \) the input gray-level, and \( s \) the output gray-level. Larger neighborhoods allow considerably more flexibility. One of the principal approach in this formulation is based on the use of so-called masks. A mask is a

44
small (say, 3x3 pixels) 2-D array in which the values of the mask coefficients determine the nature of the process, such as image sharpening. Enhancement operations based on this approach often are referred to as mask processing or filtering.

2.3 BASIC GRAY LEVEL TRANSFORMATIONS

Image gray-level transformations are among the simplest of all image enhancement techniques and are given by a generalized formula as \( s = T(r) \). Since we are dealing with digital quantities, values of the transformation function typically are stored in a 1-D array and the mappings from \( r \) to \( s \) are implemented via table lookups [28]. Three basic types of functions often used for image enhancement: linear (negative and identity), logarithmic (log and inverse log), power-law (\( n^{th} \) power, \( n^{th} \) root), and logic operators (AND, OR, NOT).

The negative of an image in the gray-level range \([0 - L - 1]\) is obtained by a transformation: \( s = L - 1 - r \). Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray details embedded in dark region, especially when the dark regions are predominant (recall that “it makes all the differences whether one sees darkness through light or brightness through a shadow”, David Lindsay).

The general form of the log transform is given by \( s = c \log (1 + r) \), \( c \) being a constant and \( r \geq 0 \). This transform maps a narrow range of low gray-levels in the input image into a wider range of output gray levels. Converse is true for higher values of input levels.

The basic form of power-law transform is \( s = cr^\gamma \) which is generalized as \( s = c(r + \varepsilon)^\gamma \), \( \varepsilon \) being an offset factor (i.e., a measurable output without an input). For
fractional values of $\gamma$, the effect is just like a log transform. Curves generated with values of $\gamma>1$ have exactly the opposite effect as those generated for $\gamma<1$. When $c = \gamma = 1$, the transform is identity (no change in gray levels).

Logic operations involving images are performed on a pixel-by-pixel basis between two or more images, with exception of NOT operation which is performed on a single image. The logic operators AND, OR, and NOT are known as being functionally complete because any type of complex logic operation may be realized in terms of these three operators. In the realm of logic operators, the pixels are processed as strings to binary numbers, i.e. for an 8-bit pixel (a string of 8 binary digits), say 10011010, a not operator yields 01100101 (just like a negative transform). The logic operators are basically used for masking; that is, for selecting a subimage from a given image, which is useful in area-of-interest processing.

In the category of arithmetic operations, substraction and addition are of particular interest in enhancement. Substraction or difference between two images $f(x, y)$ and $h(x, y)$ is given by an output image as $g(x, y) = f(x, y) - h(x, y)$. The key usefulness of substraction is the enhancement of difference between two images (mainly of the same region, location or area). One of the most commercially successful and beneficial uses of image substraction is in the area of medical imaging called mask mode radiography in which the propagation of contrast medium is analyzed by a sequence of substracted images. Image addition and subsequent averaging is a major tool for additive noise removal. A noisy image can be modeled as $g(x, y) = f(x, y) + \eta(x, y)$, where the additive noise $\eta(x, y)$ is uncorrelated and has zero average value. If such types of $K$ images are averaged then $\sigma_{g(x,y)} = \sigma_{\eta(x,y)} / \sqrt{K}$, where $\sigma_{g(x,y)}$ is the variance (or standard deviation) of output. It is clear that as $K$ (the number of images) is increased, the variability (or
noise) is decreased [29]. An important application of image averaging is in the field of astronomy where imaging with very low light level is routine, causing sensor noise frequently to render single images virtually useless for analysis.

Piecewise linear transformations are basically employed for contrast stretching and gray-level slicing. The principal advantage of piecewise linear functions is that the form of piecewise functions can be arbitrarily complex. In fact, a practical implementation of some important transformations can be formulated only as piecewise functions. Contrast stretching is the simplest piecewise linear function. The idea behind contrast stretching is to increase the dynamic range of the gray-levels in an image. A typical transformation used for contrast stretching is shown in the Fig. (2.1). The locations of points \((r_1, s_1)\) and \((r_2, s_2)\) control the shape of the transformation. If \(r_1 = s_1\) and \(r_2 = s_2\), the transformation is linear function that produces no change in gray-level. If \(r_1 = r_2, s_1 = 0\) and \(s_2 = L-1\), the transformation becomes a thresholding function. Highlighting a specific range of gray-levels in an image is done by gray-level slicing. It is of particular importance in enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images. The procedure is depicted in Fig. (2.2).

![Fig. 2.1 A typical transformation function for contrast stretching](image)

![Fig. 2.2 Transformation function for gray-level slicing](image)
2.4 HISTOGRAM PROCESSING

The histogram of a digital image with gray-level range \([0, L-1]\) is a discrete function
\[ h(r_k) = n_k, \]
where \(r_k\) is the \(k^{th}\) gray-level and \(n_k\) is the number of pixels in the image
having gray-levels \(r_k\). It is common practice to normalize the histogram by dividing each
of its value by total number of pixels in one image i.e. a normalized histogram is given by

\[ P(r_k) = \frac{n_k}{n}, \]

where \(P(r_k)\) gives approximate estimate of probability of occurrence of
gray-level \(r_k\) \([30]\). Histograms are the basis of numerous SD processing techniques. In
addition to providing useful image statistics, the information inherent in the histogram
also is quite useful in image compression and segmentation. Histograms are simple to
calculate in software and also lend themselves to economic hardware implementations,
thus making them a popular tool for real-time image processing. There are two main
histogram treatment techniques: Histogram Equalization (HE) and Histogram
Specification (HS).

In many situations it is desirable to modify the contrast of an image so that is
histogram matches a preconceived shape, other than a simple closed form mathematical
modification of the original version. A particular and important modified shape is the
uniform histogram in which, in principle, each bar has the same height. Such histogram
has associated with it an image that utilizes the available brightness levels equally and
thus should give a display in which there is good representation of details at all brightness
values. In practice a perfectly uniform histogram can not be realized for image. Thus the
method of producing a uniform /quasi-uniform histogram is known as HE. The process of
HE is formulated as

\[ y = f(x) = \frac{(L - 1)}{N} \int h_i(x) \, dx, \]

where \(h_i(x)\) is the histogram function of the original image, \(L\) is the brightness values, and \(N\) the total number of pixels. Thus,
the HE therefore is the integral of the original histogram function times a scaling factor.
The integral is just the continuous cumulative histogram. In case of the imagery with quantized brightness values this can be replaced by a discrete version as

\[ y = \left\lfloor \frac{(L-1)/A} \right\rfloor \sum_{i=0}^{L-1} h_i(x), \]

which will produce the discrete equivalent of a uniform probability density function i.e. a uniform histogram. This equation has a general tendency of spreading the histogram of image so that the levels of the histogram-equalized image will span the full dynamic range of gray-scale.

Frequently it is desirable to match the histogram of one image to that of another image to make the apparent distribution of brightness values in the two images as close as possible. This would be a necessary step, for example, when a pair of contiguous images are to be joined to form a mosaic. Matching their histograms will minimize the brightness value variations across the join [31]. In another case, it might be desirable to match the histogram of an image to a pre-specified shape, other than the uniform distribution. For example, it is often found of value in photointerpretation to have an image whose histogram is a Gaussian function. The overall treatment is what we call histogram specification (HS) or matching. Suppose it is desired to match the histogram of a given image, \( h_i(x) \), to the histogram \( h_d(y) \) (which could be a pre-specified mathematical expression or the histogram a second image). The first step is to equalize \( h_i(x) \) by HE process to obtain an intermediate histogram \( h^*(z) \) which is then modified to the desired shape \( h_d(y) \) by following mapping procedure using \( z = f(x) \) as translation function:

\[ y = g^{-1}(z), \quad z = f(x) \text{ or } y = g^{-1}(f(x)). \]

### 2.5 SPATIAL DOMAIN FILTERING

Some neighborhood operations work with the values of image pixels in the neighborhood and the corresponding values of a subimage that has the same dimension as
the neighborhood. The subimage is called a filter, mask, kernel, template, or window. The values in a filter subimage are referred to as coefficients, rather than pixels. The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called FD [32]. The process of filtering consists simply of moving a filter mask from point to point in an image. At each point \((x, y)\), the response of the filter is calculated using a predefined relationship. For a linear spatial filter, the response is given by a sum of products. For a 3×3 mask, the response \(R\) of a linear operation with centre of the mask at \((x, y)\) is 
\[
R = \sum w_i f_i ,
\]
where \(w_i\) are filter weights. The general form of linear filtering of an image of size \(MN\) with a mask of size \(mn\) is 
\[
g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t),
\]
where \(a = (m-1)/2, b = (n-1)/2\). The SD filters are broadly categorized as smoothing spatial filters and sharpening spatial filters, so are the FD filters.

The response of a smoothing, linear spatial filter is simply the average of pixels contained in the neighborhood of the filter mask. These filters are sometimes called averaging filters (or low pass filters in FD). A smoothing filter operates by replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined, by the filter mask. This process results in an image with reduced sharp transition in gray levels. The general implementation for filtering an \(MN\) image with a weighted averaging filter of size \(mn\) is expressed as:

\[
g(x, y) = \left[ \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t), \right] / \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t), \text{ with } a = (m-1)/2 \text{ and } b = (n-1)/2, x = 0, 1, 2... M-1, y =0, 1, 2... N-1.\]

Smoothing filters are used for blurring and for noise reduction. Blurring is useful in processing steps, such as removal of small
details from an image prior to object (large) extraction and bridging of small gaps in lines or curves.

The principal objective of sharpening spatial filters is to highlight fine details in an image or to enhance details that have been blurred, either in error or as a natural effect of image acquisition [33]. The concept of sharpening filters is based on first and second order derivatives. The derivatives of a digital function are characterized in terms of differences. There are various ways to define these differences. However, we require that any definition we use for a first derivative: must be zero in a flat segment (areas of constant gray-levels), must be non-zero at the onset of a gray-level step or ramp, and must be non-zero along a ramp. Similarly, any definition of a second derivative: must be zero in flat areas, must be non-zero at the onset and end of a gray-level step or ramp, and must be zero along ramps of constant slope. For 1-D digital function \( f(x) \), the discrete form of first derivative is given as \( \frac{df}{dx} = f(x + 1) - f(x) \). Similarly, the second derivative of the same function in 1-D is given as \( \frac{d^2f}{dx^2} = f(x + 1) + f(x - 1) - 2f(x) \).

These two expressions completely satisfy the aforementioned conditions. For a 2-D (image) function \( f(x, y) \), first and second derivatives are implemented as Gradients and Laplacian operators respectively [34].

Laplacian is the simplest isotropic derivative operator (Rosenfeld and Kak, 1982), whose operation on a 2-D discrete function \( f(x, y) \) is given as:

\[
\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}.
\]

But for a 2-D discrete function we have:

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

\[
\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y), \text{ So we get that}
\]
\[ \nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y). \]

This is the discrete version of Laplacian which can be implemented via a 3x3 kernel with coefficients of various terms in Laplacian expression as weights. Addition or subtraction of Laplacian from the original image yields a sharpened result as 
\[ g(x, y) = [f(x, y) - \nabla^2 f(x, y)], \text{if the central weight of Laplacian is negative, and} \]
\[ [f(x, y) + \nabla^2 f(x, y)], \text{if the same is positive.} \]

First derivatives in DIP are implemented using the magnitude of the gradient. For a function \( f(x, y) \), the gradient of \( f \) at coordinates \( (x, y) \) is defined as a 2-D column vector by the expression 
\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}^T, \text{ } \tau \text{ being the transpose.} \]

The magnitude of this vector is given by 
\[ |\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \]

It can be seen that components of gradient vector are linear, but the magnitude is non-linear. On the other hand, the partial derivatives are not isotropic, but the magnitude of gradient vector is. The computational burden of implementation of gradient operator in this present form is not trivial, and it is common practice to approximate the magnitude of the gradient by using the absolute values instead of squares and square roots i.e. 
\[ |\nabla f| = |G_x| + |G_y|. \]

The gradient operator highlights the discontinuities in a particular direction, specified by the arrangement and signs of mask weight.

In the category of non-linear, filters, order-statistics filters enjoy a central position in SD filtering. The response of these filters is based on ordering (ranking) the pixels in the image area encompassed by the filter mask, and then replacing the central pixel with the value determined by ranking result. Median filter; if the ranking result is median, Max filter; if highest value in the ranking order is taken, and min filter; if the lowest value in

52
the ranking order is considered, are some most widely used order statistics filters. Median filter is the most potent tool for the removal of impulse noise or salt-and-pepper noise.

2.6 IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

In his famous book ‘La Théorie analytique de la Chaleur,’ the Frenchman, Jean Baptiste Joseph Fourier established that any function that periodically repeats itself can be expressed as a sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient, the result is what we call a Fourier series. Even, functions that are not periodic, but whose area under the curve is finite, can be expressed as the integral of sines and/or cosines multiplied by a weighing function. The formulation in this case is the Fourier transform and its utility is even greater than the Fourier series in most of practical problems. Apart from that, both representations share the important characteristics that a function, expressed in either a Fourier series or transform, can be reconstructed (recovered) completely via an inverse process with no loss of information. This facilitates to allow work in ‘Fourier domain’ and then return to original domain without sacrificing any information [35]. The application of Fourier initial ideas was in the field of heat diffusion, where they allowed the formulation of differential equations representing heat flow in such a way that solution could be obtained first time. The advent of digital computation and the discovery of Fast Fourier transform (FFT), algorithm revolutionized the field of signal processing. Fourier techniques provide a meaningful and practical way to study and implement a host of image enhancement approaches.
2.6.1 1-D and 2-D Discrete Fourier Transform

DFT of a 1-D discrete function \( f(x), x = 0, 1, 2, \ldots M - 1 \) is given by

\[
F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi xu/M}, \text{ for } u = 0, 1, 2, \ldots M-1.
\]

For a given \( F(u) \), the original function \( f(x) \) may be retrieved via an inverse transform as

\[
f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi xu/M}, \text{ for } x = 0, 1, 2, \ldots M-1.
\]

These two expressions constitute a DFT pair. Taking into consideration that \( e^{j\theta} = \cos \theta + j \sin \theta \), we can express the above expressions as

\[
F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) [\cos 2\pi xu/M - j \sin 2\pi xu/M].
\]

Thus, we can notice that each term of the DFT is composed of the sum of all values of the function \( f(x) \), multiplied by the sines and cosines of various frequencies. The domain (values of \( u \)) over which the values of \( F(u) \) range, is appropriately called frequency domain, because \( u \) determines the frequency of the components of transform. The 2-D counterparts of above expressions may be written as:

\[
F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}, \text{ for } u = 0, 1, 2, \ldots M-1, \text{ and } v = 0, 1, 2, \ldots N-1,
\]

and

\[
f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}, \text{ for } x = 0, 1, 2, \ldots M-1, \text{ and } y = 0, 1, 2, \ldots N-1.
\]

As we see that the components of DFT are complex, it is convenient to express it in terms of magnitude, popularly known as Fourier Spectrum and is given by

\[
F(u, v) = [R(u, v) + I(u, v)]^{1/2}, \text{ with } R(u, v) \text{ and } I(u, v) \text{ being real and imaginary parts of } F(u, v).
\]
2.6.2 Basic Properties of Frequency Domain

Each term of \( F(u, v) \) contains all values of \( f(x, y) \), modified by the values of the exponential terms. Thus, with the exception of trivial cases, it is usually impossible to make direct associations between specific components of an image and its transform. However, some general statements can be made about the relationship between the frequency components of DFT and spatial characteristics of an image. For instance, since frequency is directly related to the rate of change in gray-levels, it is not difficult intuitively to associate frequencies in the DFT with patterns of intensity variations in an image function. As we move away from the origin of the transform (Fourier spectrum), the low frequencies correspond to slowly varying components of an image (i.e. the smooth gray-level variations on walls and floor of a room). As we move further away from the origin, the higher frequencies begin to correspond to faster and faster gray-level changes in the image, which is due to edges and other components of an image characterized by abrupt changes in gray-levels, such as noise. The idea underlying the DFT is that the gray-scale values forming a single-band image can be viewed as a 3-D intensity surface, with the rows and columns defining two axes and the gray level value at each pixel giving the third (z) dimension. A series of waveforms of increasing frequency are fitted to this intensity surface and the information associated with each such waveform is evaluated. The DFT, therefore, provide details of (i) The frequency of each of the scale components of the image and (ii) The proportion of information associated with each frequency component. Here, the frequency is defined in terms of cycles/basic interval, where the basic interval in the across-row direction is the number of pixels on the scan line and in the down-column direction it is the number of scan lines. Frequency could be expressed in terms of meters by dividing the magnitude of the basic interval by cycles per basic interval. Thus, if the basic interval is 512 pixels each 20 meters wide,
then the wavelength of the fifth harmonic component would be \((512 \times 20)/5\) or 2048 meters. The first scale component, conventionally labeled zero, is simply the mean gray-level of the pixels making up the image. The remaining scale components have increasing frequencies starting with 1 cycle/basic interval, then 2, 3,... \(n/2\) cycle/basic interval, where \(n\) is the number of pixels or scan lines in the basic interval [36].

### 2.7 FREQUENCY DOMAIN FILTERING

Filtering in the FD consists of following step: (1) Multiply the image function \(f(x, y)\) by \((-1)^{x+y} = e^{i\pi(x+y)}\), to centre the transform, (2) Compute \(F(u, v)\), the DFT of image from (1), (3) Multiply \(F(u, v)\), by a filter transform function \(H(u, v)\), (4) Compute the inverse DFT, (5) Retrieve the real part, (6) Again multiply it by \((-1)^{x+y}\) to cancel out the previous term [37]. The basic filter operation in FD is given by \(G(u, v) = H(u, v) \cdot F(u, v)\).

The multiplication of \(H\) and \(F\) involves 2-D functions and is defined on an element-by-element basis. Smoothing and sharpening filters in FD are referred to as lowpass and highpass filters, both types are categorized in Ideal, Butterworth, and Gaussian.

The Ideal lowpass filter (ILPF) is the simplest we can envision, that cuts off all high frequency components of DFT that are at a distance greater than a specified distance \(D_0\) from the origin of centered transform. The transfer function of an ILPF is \(H(u, v) = 1\) if \(D(u, v) \leq D_0\), and 0 if \(D(u, v) \geq D_0\). For a \(MN\) image the \(D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}\) for a centered transform. In ILPF the value of \(D_0\) is called cut-off frequency. The sharp cut-off frequencies of an ILPF can not be realized in electronic components, but they can certainly be implemented in a computer.

The filter transfer function of a Butterworth lowpass filter (BLPF) is \(H(u, v) = 1/[1+(D(u, v)/D_0)^{2n}]\), where \(n\) is the order of the filter. Unlike the ILPF, the
BLPF transfer function does not have a sharp discontinuity that establishes a clear cut-off between passed and filtered frequencies. In fact a BLPF may vary from ideal response to Gaussian depending on its order (for higher orders it tends to be ideal and for lower order it tends to be Gaussian).

Gaussian lowpass filter (GLPF) is characterized by a filter transfer function

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

where \( \sigma \) is the measure of the spread of Gaussian curve (or standard deviation). It is often convenient to put \( \sigma = D_0 \) (cut-off frequency). The Gaussian filters are very convenient to deal with both in SD and FD because of the important property that Fourier transform of a Gaussian function is also Gaussian, and so its spatial counterpart can be readily procured.

A sharpening FD filter can be obtained from corresponding smoothing counterpart by a simple relation \( H_{hp}(u, v) = 1 - H_{lp}(u, v) \). In view of this the filter transfer functions of IHPS, BHPF, and GHPF would be as

\[ H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0, \text{ and } 1 \text{ if } D(u, v) > 0 \end{cases}, \quad H(u, v) = 1/[1+(D_0/D(u, v))^2\sigma^1], \quad \text{and} \quad H(u, v) = 1-e^{-\sigma^2(u,v)/2D_0^1}, \]

respectively, where all the symbols are in the usual meanings.

2.8 CORRESPONDENCE BETWEEN SD AND FD

The most fundamental relationship between the SD and FD is established by well-known result called the convolution theorem (CT). Convolution is the process by which we move a mask from pixel to pixel in an image, and compute a predefined quantity at each pixel position. The discrete convolution of two functions, \( f(x, y) \) and \( h(x, y) \) of size \( MN \) is given by expression

\[ f(x, y) * h(x, y) = (1/MN) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x-m, y-n), \]

where ‘*’ is the sign for convolution operation. The implementation of this equation
involves: (i) Flipping one function about the origin, (ii) Shifting that function with respect to the other by changing the values of \((x, y)\), and (iii) Computing a sum of products over all values of \(m\) and \(n\) for each displacement \((x, y)\) which are integer increments, that stop when the functions no longer overlap. If \(F(u, v)\) and \(H(u, v)\) denote the DFT of \(f(x, y)\) and \(h(x, y)\), respectively then the CT states that \(f(x, y) * h(x, y) \Leftrightarrow F(u, v) \cdot H(u, v)\) and \(f(x, y) \cdot h(x, y) \Leftrightarrow F(u, v) \cdot H(u, v)\). In other words CT states that convolution in SD is equivalent to multiplication in FD and vice versa. Thus, the FD may be viewed as a “laboratory” in which we take advantage of the correspondence between the frequency content and image appearance. Some enhancement tasks that would be exceptionally difficult or impossible to formulate in SD, become almost trivial in FD [38]. Once we have selected a specific filter via experimentation in FD, the actual implementation is usually done in SD [39].

2.9 SEGMENTATION AND THRESHOLDING

Most of the DIP algorithms perform segmentation as a first step towards producing the description. Here, input and output are images, but output is an abstract representation of the input. Segmentation techniques basically divide the SD, on which the image is defined, into meaningful parts or regions. The meaningful region may be a complete object or a part of it. The segmentation algorithms try to make systematic use of some physically measured image features, but its performance is measured based on the meaning associated with the extracted regions [40]. So segmentation is a psychophysical problem. Secondly, the requirement that the extracted region must have some meaning with respect to the given scene in it, makes the problem mathematically ill-posed. As a result, there exists no general algorithm for segmentation. Users have to chose or develop a segmentation algorithm suitable for the problem in hand. So, all the segmentation
algorithms are *ad hoc* in nature. A segmentation algorithm should satisfy two approaches: (1) Homogeneity property in an image feature(s) over a large region, and (2) Detecting abrupt changes in image feature(s) within a small neighborhood (Chanda, 1988). Fu and Mui (1981) and Pal and Pal (1993) have presented good surveys on segmentation methodologies.

### 2.9.1 Thresholding

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in the applications of image segmentation. Suppose that the gray-level histogram shown in Fig.2.3 (a) corresponds to an image \( f(x, y) \), composed of light objects on a dark background, in such a way that object and background pixels have gray-levels grouped into two dominant modes (two peaks). One obvious way to extract the objects from the background is to select a threshold \( T \) that separates these modes. Then any point \( (x, y) \) for which \( f(x, y) > T \) is called an object point; otherwise the point is called a background point. This approach is used in the present study. In the next figure (Fig.2.3 (b)) a slightly more general case is approached, where three dominant modes, characterize the image histogram, depicting, say, two types of light objects in a dark background. Here, the approach is multi-level thresholding that classifies a point \( (x, y) \) as belonging to one object class if \( T_1 < f(x, y) \leq T_2 \), to the other object class if \( f(x, y) > T_2 \), and to the background if \( f(x, y) \leq T_1 \). In general, segmentation problems requiring multiple thresholds are best-solved using region growing methods [41]. Thus, thresholding may be viewed as an operation that involves tests against a function \( T \) of the form \( T = T [x, y, p(x, y), f(x, y)] \), where \( f(x, y) \) is the gray-level at point \( (x, y) \), and \( p(x, y) \) denotes some local property of this point (e.g. the average gray-level of a neighborhood centered on \( (x, y) \)). In this way, a thresholded image \( g(x, y) \) is defined as
\[ g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T, \\ 0 & \text{if } f(x, y) \leq T \end{cases} \] Thus, pixels labeled 1 (or any other convenient gray-level) correspond to objects, whereas pixels labeled 0 correspond to the background. When \( T \) depends only on \( f(x, y) \) (that is, only on gray-level values) the threshold is called global. On the other hand, it is called local if it depends on both \( f(x, y) \) and \( p(x, y) \). If, in addition, \( T \) depends on the spatial coordinates \( x \) and \( y \), it is called dynamic or adaptive.

Fig. 2.3 Gray level histograms
(a) One type of light object in a dark background
(b) Two types of light objects in a dark background