INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal, Hyderabad - 500 043

LECTURE NOTES:

IMAGE PROCESSING(AECC26)

DRAFTED BY :

MS.B LAKSHMI PRASANNA , ASSISTANT PROFESSOR

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

INSTITUTE OF AERONAUTICAL ENGINEERING

February 27, 2024

Contents

List of Figures

Chapter 1

INTRODUCTION

Course Outcomes

After successful completion of this module, students should be able to:

CO 1	**Interpret** the principles and terminology of digital image processing for describing the features of image.	Understand
CO 2	**Make use of** image transform techniques for analyzing images in transformation domain for image pre-processing.	Apply

1.1 Introduction

Basic concept of digital image:

The field of digital image processing refers to processing digital images by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels and pixels. Pixel is the term used most widely to denote the elements of digital image.An image is a two-dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene. An image is a projection of a 3- D scene into a 2D projection plane.

An image may be defined as a two-dimensional function f(x,y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity of the image at that point.

The term gray level is used often to refer to the intensity of monochrome images.

Color images are formed by a combination of individual 2-D images.For example: The RGB color system, a color image consists of three (red, green and blue) individual component images. For this reason many of the techniques developed for monochrome images can be extended to color images by processing the three component images individually.

FIGURE 1.1: Figure Image aquisition

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

1.2 Applications

Since digital image processing has very wide applications and almost all of the technical fields are impacted by DIP, we will just discuss some of the major applications of DIP.
Digital image processing has a broad spectrum of applications, such as
• Remote sensing via satellites and other spacecrafts
• Image transmission and storage for business applications
• Medical processing
• RADAR (Radio Detection and Ranging)
• SONAR(Sound Navigation and Ranging) and
• Acoustic image processing (The study of underwater sound is known as underwater acoustics or hydro acoustics.)
• Robotics and automated inspection of industrial parts. mages acquired by satellites are useful in tracking of
• Earth resources
• Geographical mapping
• Prediction of agricultural crops
• Urban growth and weather monitoring • Flood and fire control and many other environmental applications. Space image applications include
• Recognition and analysis of objects contained in images obtained from deep space-probe missions
• Image transmission and storage applications occur in broadcast television
• Teleconferencing
• Transmission of facsimile images(Printed documents and graphics) for office automation Communication over computer networks
• Closed-circuit television based security monitoring systems and
• In military communications.
• Medical applications:

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

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

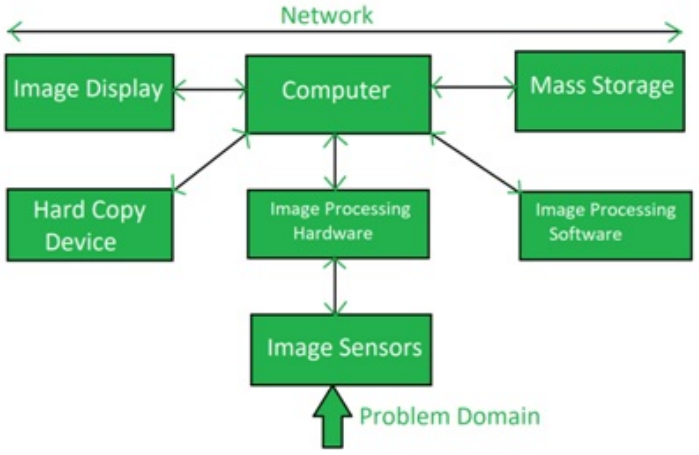


FIGURE 1.2: Components of Image processing System:

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

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

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

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

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

• Short term storage for use during processing
• On line storage for relatively fast retrieval
• Archival storage such as magnetic tapes and disks

Image display: Image displays in use today are mainly color TV monitors. These monitors are driven by the outputs of image and graphics displays cards that are an integral part of computer system.
Hardcopy devices: The devices for recording image includes laser printers, film cameras, heat sensitive devices inkjet units and digital units such as optical and CD ROM disk. Films provide the highest possible resolution, but paper is the obvious medium of choice for written applications.
Networking: It is almost a default function in any computer system in use today because of the large amount of data inherent in image processing applications. The key consideration in image transmission bandwidth.

1.2.1 Fundamental Steps in Digital Image Processing:

There are two categories of the steps involved in the image processing
1. Methods whose outputs are input are images.
2. Methods whose outputs are attributes extracted from those images.

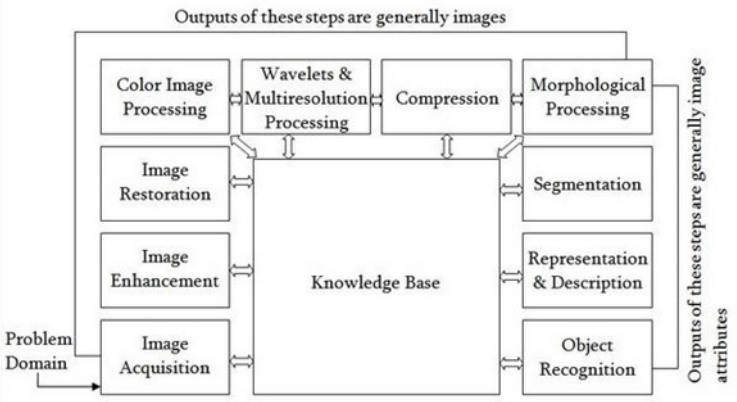


FIGURE 1.3: fundamental steps in image processing

Image acquisition It could be as simple as being given an image that is already in digital form. Generally the image acquisition stage involves processing such scaling.
Image Enhancement It is among the simplest and most appealing areas of digital image processing. The idea behind this is to bring out details that are obscured or simply to highlight certain features of interest in image. Image enhancement is a very subjective area of image processing.

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

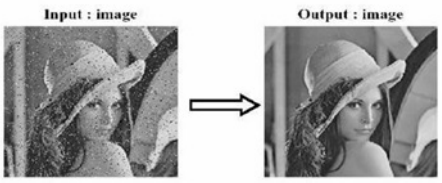


FIGURE 1.4: example for enhancement

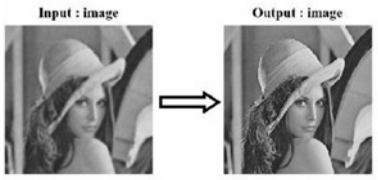


FIGURE 1.5: Example for restoration

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

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

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

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

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

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

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

1.3 Digital image through scanner

Scanner is a device that scans images, printed text, and handwriting etc and converts it to digital form or image. It is so named because the data is converted one line at a time or scanned down the page as the scanning head moves down the page.

FIGURE 1.6: scanner

Components inside a scanner are the following Glass Plate and Cover The glass plate is the transparent plate wherein the original is placed so that the scanner can scan it and the cover keeps out stray light that can affect the accuracy of the scan

Scanning head Scanning head is the most important component because it is the one which does actual scanning. It contains components like

Light source and mirror: It is the bright white light that is used to illuminate the original as it is being scanned and which bounces off the original and reflected off several mirrors

Stabilizer bar: It is a long stainless steel rod that is securely fastened to the case of the scanner and it provides a smooth ride as the scanner scans down the page

CCD (Charge Coupled Device) or CIS (Contact Image Sensor): A CCD array is a device that converts photons into electricity. Any scanner that uses CCD use lens to focus the light coming from the mirrors within the scanning head. Another technology used in some cheaper scanners is CIS wherein the light source is a set of LEDs that runs the length of the glass plate.

iii. Stepper Motor The stepper motor in a scanner moves the scan head down the page during scan cycle and this is often located either on the scan head itself or attached to a belt to drive the scanner head.

1.4 Image formation on analog cameras

In analog cameras, the image formation is due to the chemical reaction that takes place on the strip that is used for image formation.A 35mm strip is used in analog camera. It is denoted in the figure by 35mm film cartridge. This strip is coated with silver halide (a chemical substance).

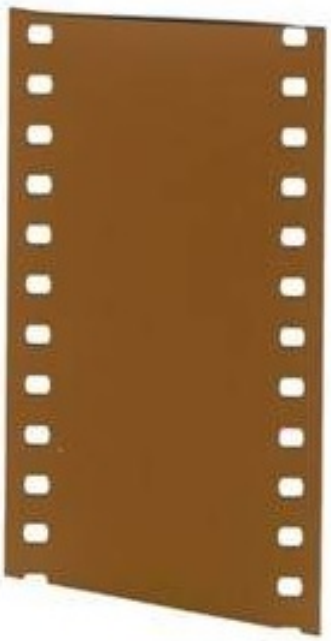


FIGURE 1.7: Analog film

A 35mm strip is used in analog camera. It is denoted in the figure by 35mm film cartridge. This strip is coated with silver halide (a chemical substance).
Light is nothing but just the small particles known as photon particles. So when these photon particles are passed through the camera, it reacts with the silver halide particles on the strip and it results in the silver which is the negative of the image.In order to understand it better, have a look at this equation.
Photons (light particles) + silver halide +Silver Image negative.
This is just the basics, although image formation involves many other concepts regarding the passing of light inside, and the concepts of shutter and shutter speed and aperture and its opening but for now we will move on to the next part. Although most of these concepts have been discussed in our tutorial of shutter and aperture.
This is just the basics, although image formation involves many other concepts regarding the passing of light inside, and the concepts of shutter and shutter speed and aperture and its opening but for now we will move on to the next part. Although most of these concepts have been discussed in our tutorial of shutter and aperture.

1.5 Image formation on digital cameras

In the digital cameras, the image formation is not due to the chemical reaction that takes place; rather it is a bit more complex then this. In the digital camera, a CCD array of sensors is used for the image formation. Image formation through CCD array

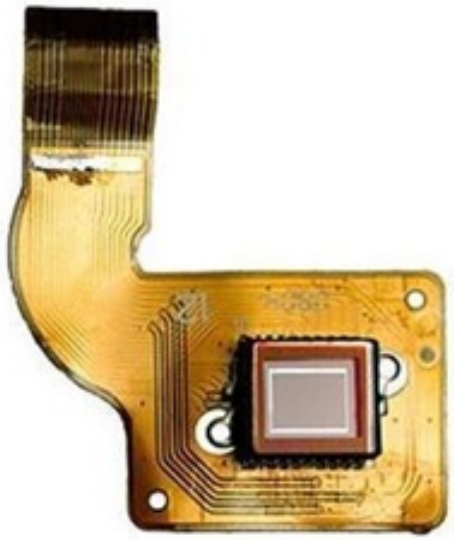


FIGURE 1.8: digital camera CCD

CCD stands for charge-coupled device. It is an image sensor, and like other sensors it senses the values and converts them into an electric signal. In case of CCD it senses the image and converts it into electric signal e.t.c.
This CCD is actually in the shape of array or a rectangular grid. It is like a matrix with each cell in the matrix contains a censor that senses the intensity of photon. Like analog cameras, in the case

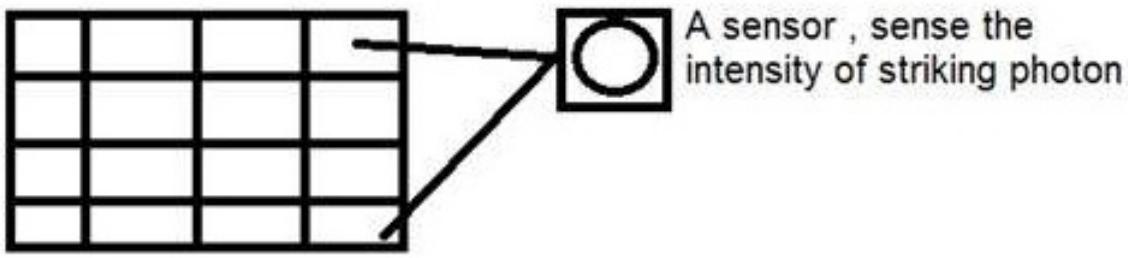


FIGURE 1.9: digital camera CCD

of digital too, when light falls on the object, the light reflects back after striking the object and

allowed to enter inside the camera.
Each sensor of the CCD array itself is an analog sensor. When photons of light strike on the chip, it is held as a small electrical charge in each photo sensor. The response of each sensor is directly equal to the amount of light or (photon) energy striked on the surface of the sensor.
Since we have already define an image as a two dimensional signal and due to the two dimensional formation of the CCD array, a complete image can be achieved from this CCD array.It has limited number of sensors, and it means a limited detail can be captured by it. Also each sensor can have only one value against the each photon particle that strike on it.So the number of photons striking (current) are counted and stored. In order to measure accurately these, external CMOS sensors are also attached with CCD array.

1.6 Sampling and quantization

To create a digital image, we need to convert the continuous sensed data into digital from. This involves two processes – sampling and quantization. An image may be continuous with respect to the x and y coordinates and also in amplitude. To convert it into digital form we have to sample the function in both coordinates and in amplitudes.
Digitalizing the coordinate values is called sampling. Digitalizing the amplitude values is called quantization. There is a continuous the image along the line segment AB. To simple this function, we take equally spaced samples along line AB. The location of each samples is given by a vertical tick back (mark) in the bottom part. The samples are shown as block squares superimposed on function the set of these discrete locations gives the sampled function.
In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So we divide the gray level scale into eight discrete levels ranging from eight level values. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment it made depending on the vertical proximity of a simple to a vertical tick mark.
Starting at the top of the image and covering out this procedure line by line produces a two dimensional digital image.

1.7 Digital Image definition:

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

row and a column is termed a pixel. The value assigned to the integer coordinates (m,n) with m=0,1,2..N-1 and n=0,1,2...N-1 is f(m,n). In fact, in most cases, is actually a function of many variables including depth, color and time (t).

Representing Digital Images

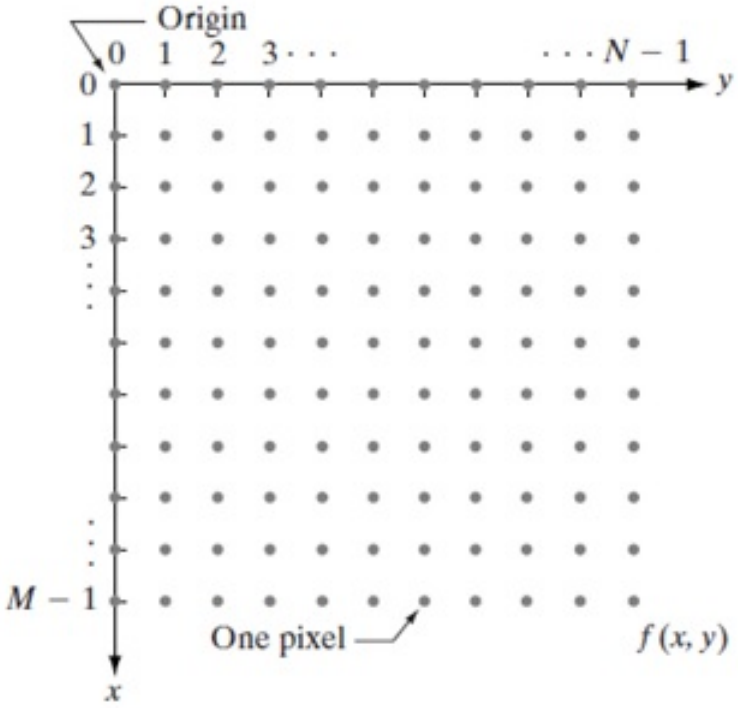


FIGURE 1.10: representation of digital image

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

FIGURE 1.11: representation of digital image

1.8 Spatial and Gray level resolution:

Spatial resolution is the smallest discernible details are an image. Suppose a chart can be constructed with vertical lines of width w with the space between the also having width W, so a line pair consists of one such line and its adjacent space thus. The width of the line pair is 2w and there is 1/2w line pair per unit distance resolution is simply the smallest number of discernible line pair unit distance.
Gray levels resolution refers to smallest discernible change in gray levels. Measuring discernible change in gray levels is a highly subjective process reducing the number of bits R while repairing the spatial resolution constant creates the problem of false contouring.
It is caused by the use of an insufficient number of gray levels on the smooth areas of the digital image . It is called so because the rides resemble top graphics contours in a map. It is generally quite visible in image displayed using 16 or less uniformly spaced gray levels
the gray levels are also integers, Z replaces R, the and a digital image become a 2D function whose coordinates and she amplitude value are integers. Due to processing storage and hardware consideration, the number gray levels typically is an integer power of 2. L=2k
Then, the number, b, of bites required to store a digital image is B=M *N* k When M=N, the equation become b=N2*k
When an image can have 2k gray levels, it is referred to as “k- bit”. An image with 256 possible gray levels is called an “8- bit image” (256=28).

1.9 Image sensing and Acquisition:

The types of images in which we are interested are generated by the combination of an “illumination” source and the reflection or absorption of energy from that source by the elements of the “scene” being imaged. We enclose illumination and scene in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene.
For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray energy. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern.
Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock formations, or a human brain. We could even image a source, such as acquiring images of the sun. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light reflected from a planar surface.
An example in the second category is when X-rays pass through a patient’s body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photo converter (e.g., a phosphor screen), which converts the energy into visible

light. Electron microscopy and some applications of gamma imaging use this approach.
The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response. In this section, we look at the principal modalities for image sensing and generation.

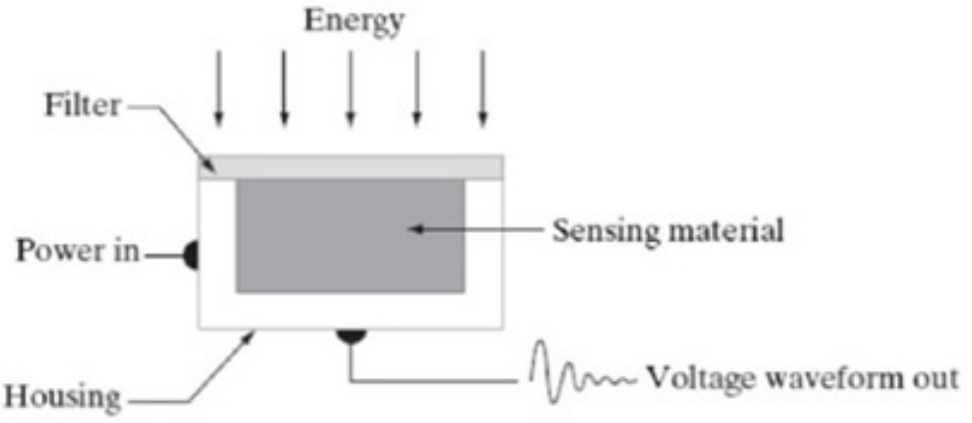


FIGURE 1.12: single image sensor

FIGURE 1.13: line sensor

FIGURE 1.14: Array sensor Image Acquisition using a Single sensor

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

In order to generate a 2-D image using a single sensor, there has to be relative displacements

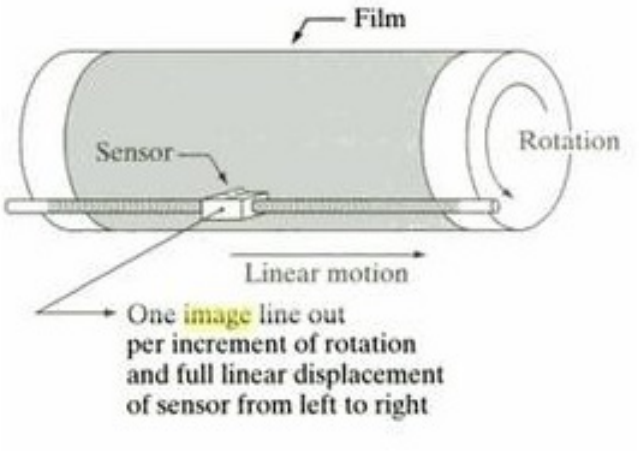


FIGURE 1.15: Array sensor Image Acquisition using a Single sensor

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

1.10 Image Acquisition using a Sensor strips:

A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction. This is the type of arrangement used in most flat bed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that flies at a constant altitude and speed over the geographical area to be imaged. One dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project area to be scanned onto the sensors. Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional ("slice") images of 3-D objects. **Image Acquisition using a Sensor Arrays:** The individual sensors arranged in the form of a 2-D array.

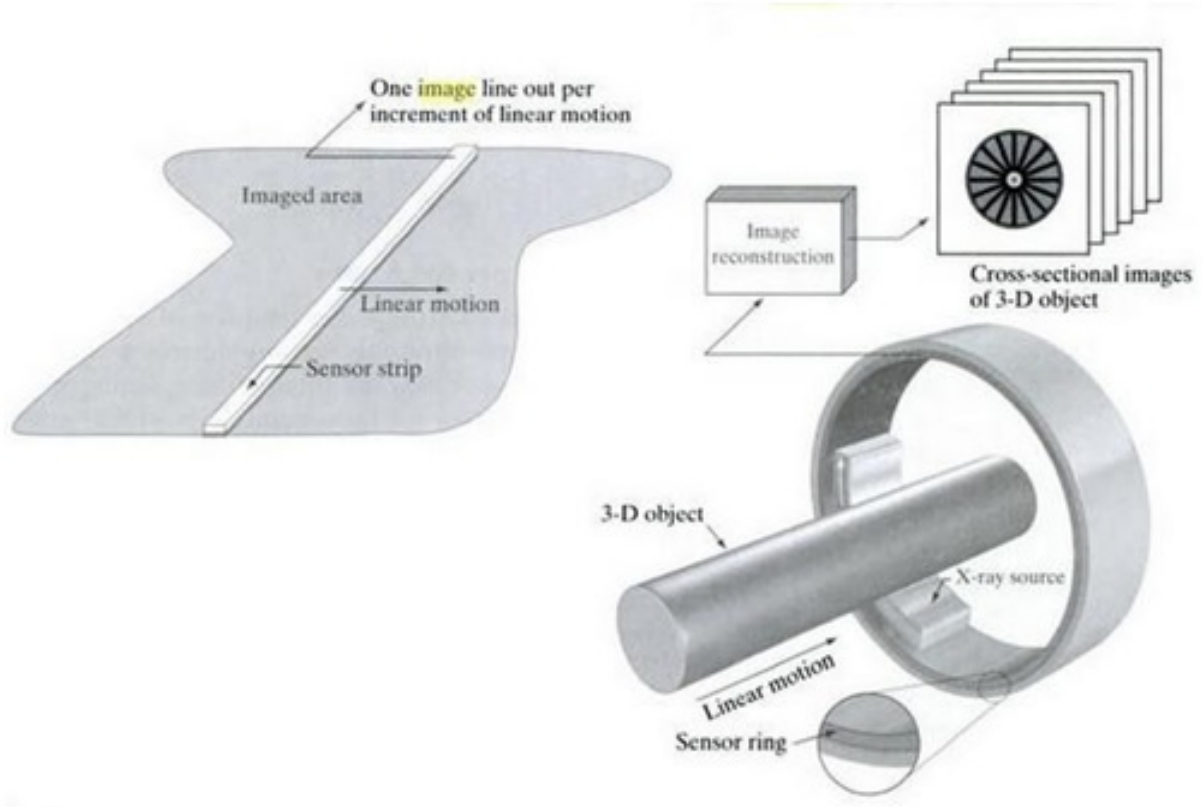


FIGURE 1.16: Image Acquisition using linear strip and circular strips

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

1.11 Image sampling and Quantization:

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

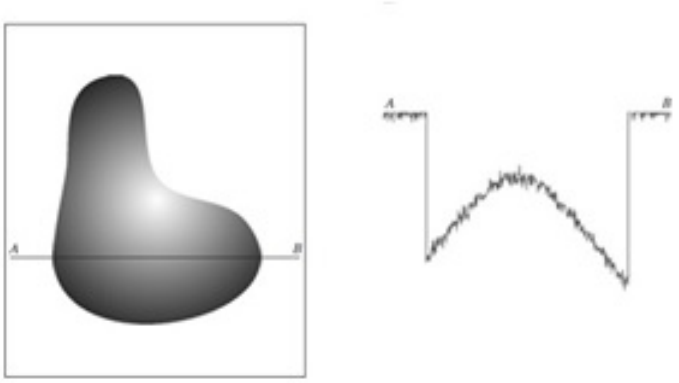


FIGURE 1.17: Image Acquisition using linear strip and circular strips

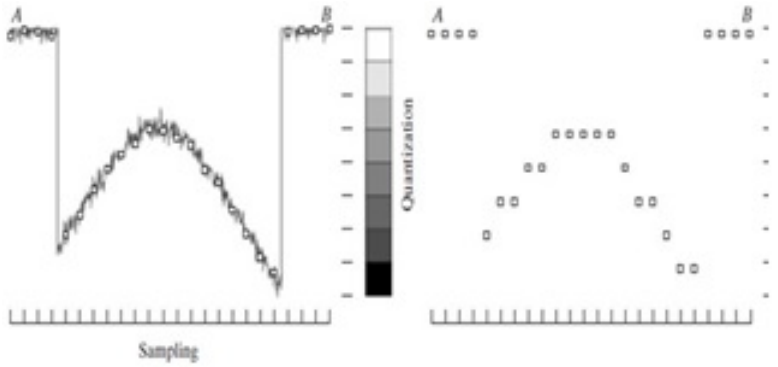


FIGURE 1.18: Image Acquisition using linear strip and circular strips

1.12 Digital Image representation:

Digital image is a finite collection of discrete samples (pixels) of any observable object. The pixels represent a two- or higher dimensional "view" of the object, each pixel having its own discrete value in a finite range. The pixel values may represent the amount of visible light, infra red light, absortation of x-rays, electrons, or any other measurable value such as ultrasound wave impulses. The image does not need to have any visual sense; it is sufficient that the samples form a

two-dimensional spatial structure that may be illustrated as an image. The images may be obtained by a digital camera, scanner, electron microscope, ultrasound stethoscope, or any other optical or non-optical sensor. Examples of digital image are:
• digital photographs
• satellite images
• radiological images (x-rays, mammograms)
• binary images, fax images, engineering drawings
Computer graphics, CAD drawings, and vector graphics in general are not considered in this course even though their reproduction is a possible source of an image. In fact, one goal of intermediate level image processing may be to reconstruct a model (e.g. vector representation) for a given digital image.

1.13 Relationship between pixels:

We consider several important relationships between pixels in a digital image.
Neighbors of a pixel
• A pixel p at coordinates (x,y) has four horizontal and vertical neighbors whose coordinates are given by:
(x+1,y), (x-1, y), (x, y+1), (x,y-1) This set of pixels, called the 4-neighbors or p, is denoted by

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

FIGURE 1.19: N4(P)

N4(p). Each pixel is one unit distance from (x,y) and some of the neighbors of p lie outside the digital image if (x,y) is on the border of the image. The four diagonal neighbors of p have coordinates and are denoted by ND (p).
(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by N8 (p). As before, some of the points in ND (p) and N8 (p) fall outside the image if (x,y) is on the border of the image. **Adjacency and connectivity** Let v be the set of gray –level values used to define adjacency, in a binary image, v=1. In a gray-scale image, the idea is the same, but V typically contains more elements, for example, V = {180, 181, 182, . . . , 200}.

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

FIGURE 1.20: ND(P)

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

FIGURE 1.21: N8(P)

If the possible intensity values 0 – 255, V set can be any subset of these 256 values. if we are reference to adjacency of pixel with value.

Three types of adjacency

• 4- Adjacency – two pixel P and Q with value from V are 4 –adjacency if A is in the set N4(P)

• 8- Adjacency – two pixel P and Q with value from V are 8 –adjacency if A is in the set N8(P)

• • 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: **Types of Adjacency:** • In this example, we can note that to connect between two

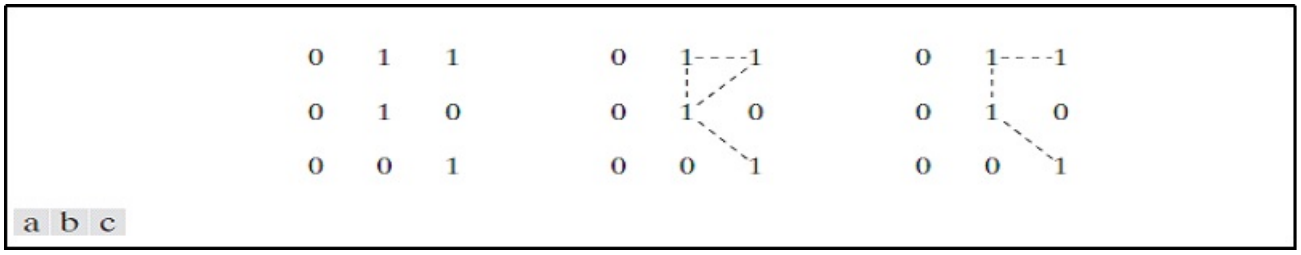


FIGURE 1.22: :(a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency.

pixels (finding a path between two pixels):

– In 8-adjacency way, you can find multiple paths between two pixels

– While, in m-adjacency, you can find only one path between two pixels
• So, m-adjacency has eliminated the multiple path connection that has been generated by the 8-adjacency.
• Two subsets S1 and S2 are adjacent, if some pixel in S1 is adjacent to some pixel in S2. Adjacent means, either 4-, 8- or m-adjacency.

A Digital Path: • n is the length of the path
• If (x0,y0) = (xn, yn), the path is closed.
We can specify 4-, 8- or m-paths depending on the type of adjacency specified.
• Return to the previous example:

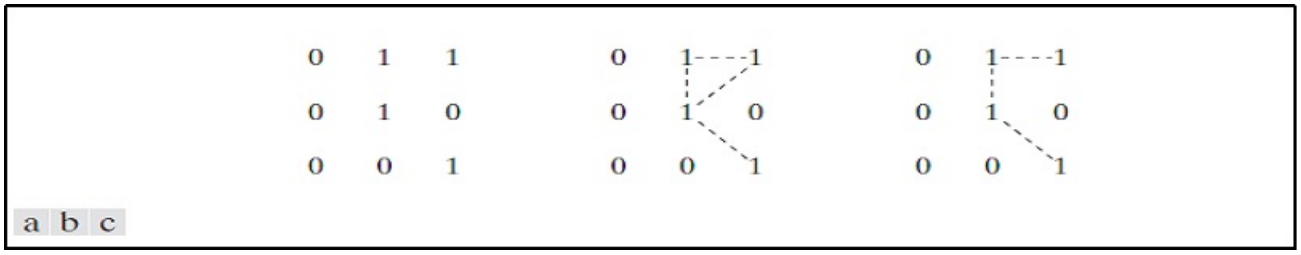


FIGURE 1.23: :(a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m-adjacency.

Connectivity: • Let S represent a subset of pixels in an image, two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels in S.
• For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S. If it only has one connected component, then set S is called a connected set. **Region and Boundary:** REGION: Let R be a subset of pixels in an image, we call R a region of the image if R is a connected set.BOUNDARY: The boundary (also called border or contour) of a region R is the set of pixels in the region that have one or more neighbors that are not in R. If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns in the image. This extra definition is required because an image has no neighbors beyond its borders. Normally, when we refer to a region, we are referring to subset of an image, and any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the region boundary.
DISTANCE MEASURES: For pixel p,q and z with coordinate (x.y) ,(s,t) and (v,w) respectively D is a distance function or metric if

D [p.q] ≥ O {D[p.q] = O iff p=q}

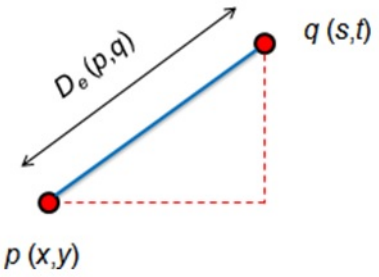
D [p.q] = D [p.q] and

D [p.q] ≥ O {D[p.q]+D(q.z)

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

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

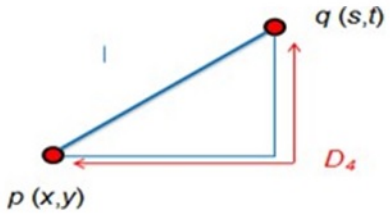
Pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius „ r „centered at (x,y)



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

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

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



Example:

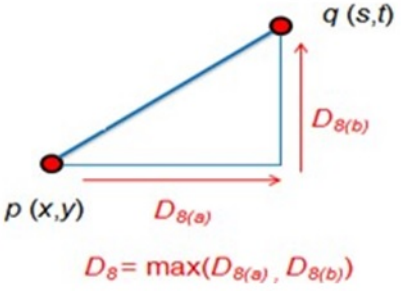
The pixels with distance $D_4 \leq 2$ from (x,y) form the following contours of constant distance.
The pixels with $D_4 = 1$ are the 4-neighbors of (x,y)

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

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

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

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



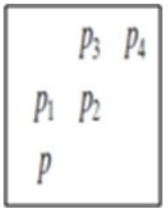
Example:

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

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

Dm distance: It is defined as the shortest m-path between the points.In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

• Example: Consider the following arrangement of pixels and assume that p, p2, and p4 have value 1 and that p1 and p3 can have can have a value of 0 or 1 Suppose that we consider the adjacency of pixels values 1 (i.e. V = {1})



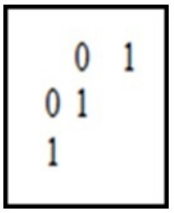
Now, to compute the D_m between points p and p_4

Here we have 4 cases:

Case1: If p_1 =0 and p_3 = 0

The length of the shortest m-path (the D_m *distance) is 2 (p, p_2, p_4)*

Case2: If p_1 =1 and p_3 = 0



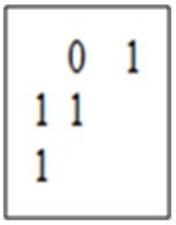
Now, p_1 *and p will no longer be adjacent (see m-adjacency definition)*

Then, the length of the shortest path will be 3 (p, p_1, p_2, p_4)

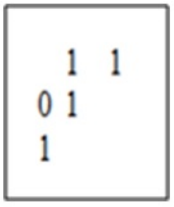
Now, p_1 *and p will no longer be adjacent (see m-adjacency definition)*

Then, the length of the shortest path will be 3 (p, p_1, p_2, p_4)

Case3: If p_1 =0 and p_3 = 1

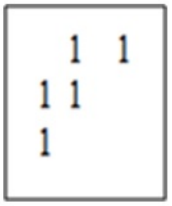


The same applies here, and the shortest –m-path will be 3 (p, p_2, p_3, p_4)



Case4: If p_1 =1 and p_3 = 1

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



1.14 Gray level resolution and quantization:

The quantization will be formally introduced in the next tutorial, but here we are just going to explain the relationship between gray level resolution and quantization.

Gray level resolution is found on the y axis of the signal. In the tutorial of Introduction to signals and system, we have studied that digitizing a an analog signal requires two steps. Sampling and quantization.

Sampling is done on x axis. And quantization is done in Y axis.

FIGURE 1.24: sampling

So that means digitizing the gray level resolution of an image is done in quantization we have introduced quantization in our tutorial of signals and system. We are formally going to relate it with digital images in this tutorial. Let's discuss first a little bit about quantization. **Digitizing a signal** As we have seen in the previous tutorials, that digitizing an analog signal into a digital requires two basic steps. Sampling and quantization. Sampling is done on x axis. It is the conversion of x axis (infinite values) to digital values. The below figure shows sampling of a signal. The concept

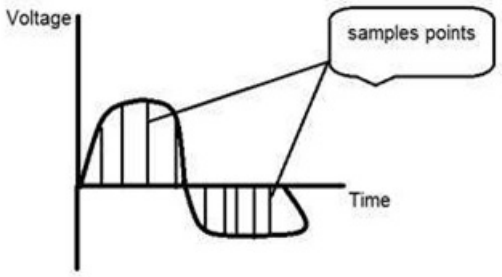


FIGURE 1.25: Sampling with relation to digital images

of sampling is directly related to zooming. The more samples you take, the more pixels, you get. Oversampling can also be called as zooming. This has been discussed under sampling and zooming tutorial.

But the story of digitizing a signal does not end at sampling too, there is another step involved which is known as Quantization. **What is quantization**

Quantization is opposite to sampling. It is done on y axis. When you are quantizing an image, you are actually dividing a signal into quanta (partitions).On the x axis of the signal, are the coordinate values, and on the y axis, we have amplitudes. So digitizing the amplitudes is known as Quantization. Here how it is done You can see in this image, that the signal has been quantified into three different levels. That means that when we sample an image, we actually gather a lot of values, and in quantization, we set levels to these values. This can be more clear in the image below.

In the figure shown in sampling, although the samples has been taken, but they were still spanning vertically to a continuous range of gray level values. In the figure shown above, these vertically

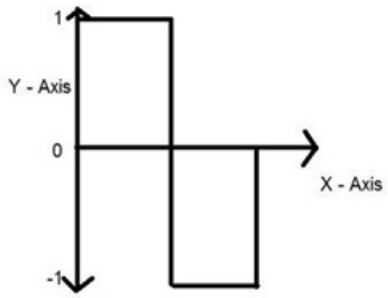


FIGURE 1.26: quantization

ranging values have been quantized into 5 different levels or partitions. Ranging from 0 black to 4 white. This level could vary according to the type of image you want. The relation of quantization with gray levels has been further discussed below. **Relation of Quantization with gray level res-**

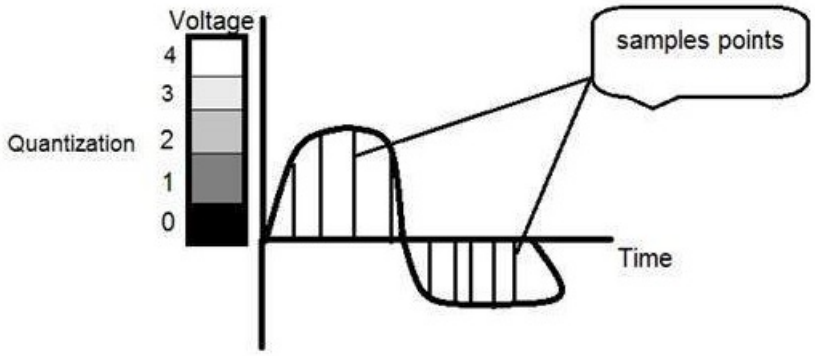


FIGURE 1.27: continues to sampling conversion

olution: The quantized figure shown above has 5 different levels of gray. It means that the image formed from this signal, would only have 5 different colors. It would be a black and white image more or less with some colours of gray. Now if you were to make the quality of the image better, there is one thing you can do here. Which is, to increase the levels or gray level resolution up? If you increase this level to 256, it means you have a gray scale image. Which is far better then simple black and white image?

Now 256, or 5 or whatever level you choose is called gray level. Remember the formula that we discussed in the previous tutorial of gray level resolution which is, $L = 2^k$

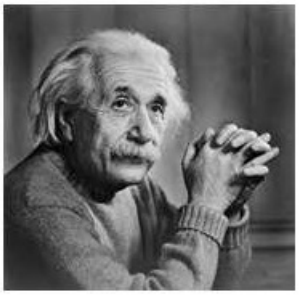
We have discussed that gray level can be defined in two ways. Which were these two.

• Gray level = number of bits per pixel (BPP).(k in the equation)

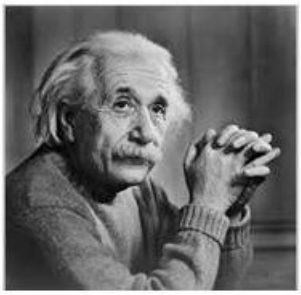
• Gray level = number of levels per pixel.

In this case we have gray level is equal to 256. If we have to calculate the number of bits, we

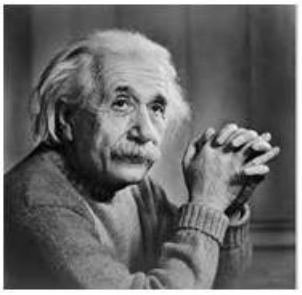
would simply put the values in the equation. In case of 256levels, we have
256 different shades of gray and 8 bits per pixel; hence the image would be a gray scale image.
Reducing the gray level
Now we will reduce the gray levels of the image to see the effect on the image.
For example
Lets say you have an image of 8bpp, that has 256 different levels. It is a gray scale image and the image looks something like this.
256 Gray Levels



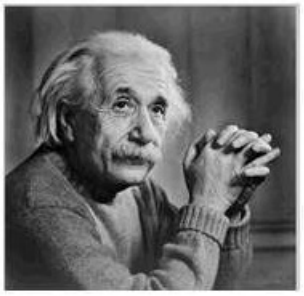
Now we will start reducing the gray levels. We will first reduce the gray levels from 256 to 128.
128 Gray Levels



There is not much effect on an image after decrease the gray levels to its half. Lets decrease some more.
64 Gray Levels

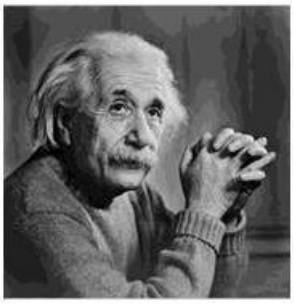


Still not much effect, then lets reduce the levels more.
32 Gray Levels



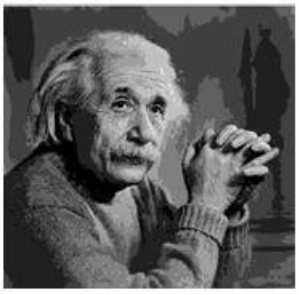
Surprised to see, that there is still some little effect. May be it's due to reason, that it is the picture of Einstein, but let's reduce the levels more

16 Gray Levels

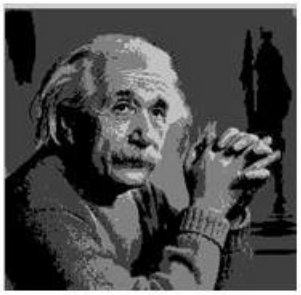


Boom here, we go, the image finally reveals, that it is effected by the levels.

8 Gray Levels



4 Gray Levels



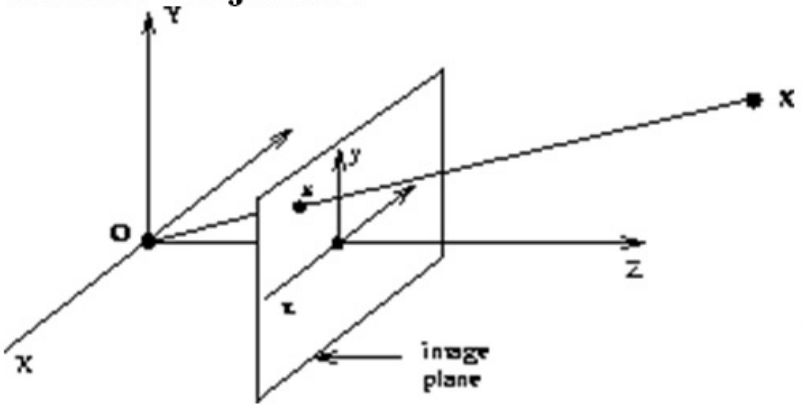
Now before reducing it, further two 2 levels, you can easily see that the image has been distorted badly by reducing the gray levels. Now we will reduce it to 2 levels, which is nothing but a simple black and white level. It means the image would be simple black and white image.

2 Gray Levels

1.15 Imaging geometry

Central Projection Here central projection is represented in the coordinate frame attached to the

Central Projection



$$\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\frac{x}{f} = \frac{X}{Z} \qquad \frac{y}{f} = \frac{Y}{Z}$$

Vector notation:

$$\mathbf{x} = f\frac{\mathbf{X}}{Z}$$

Where $\mathbf{x}$ and $\mathbf{X}$ are 3-vectors, with.

$$\mathbf{x} = (x, y, f)^{\mathsf{T}}$$

camera. Generally, there is not direct access to this camera coordinate frame. Instead, we need to determine the mapping from a world coordinate frame to an image coordinate system

1.16 Introduction

An image transform can be applied to an image to convert it from one domain to another. Viewing an image in domains such as frequency or Hough space enables the identification of features that may not be as easily detected in the spatial domain. ... Discrete Fourier Transform, used in filtering and frequency analysis.
Many times, image processing tasks are best performed in a domain other than the spatial domain.
•Key steps
(1) Transform the image
(2) Carry the task(s) in the transformed domain.

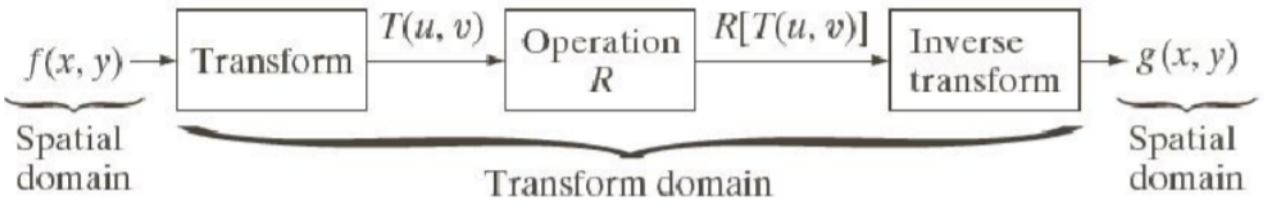


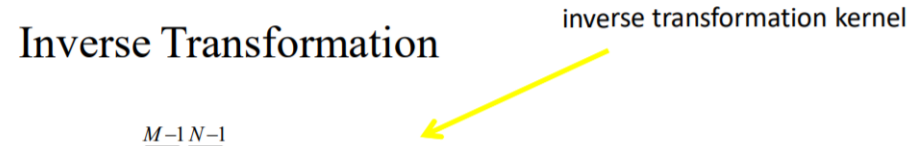
FIGURE 1.28: Image Transformation

Transformation Kernels

- Forward Transformation

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

- Inverse Transformation



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

Kernel Properties

- A kernel is said to be *separable* if:

$$r(x,y,u,v) = r_1(x,u)r_2(y,v)$$

- A kernel is said to be *symmetric* if:

$$r(x,y,u,v) = r_1(x,u)r_1(y,v)$$

1.17 2D-FFT PROPERTIES

2D Discrete Fourier Transform

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

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

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

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

2D Discrete Fourier Transform

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

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

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

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

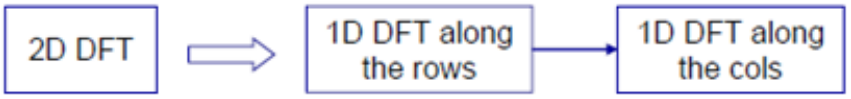
Separability

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

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

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

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



Separability

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

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

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

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

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

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

1.18 Walsh Transform

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

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

The above is equivalent to:

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

The transform kernel values are obtained from:

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

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

1.18.1 1-D Inverse Walsh Transform

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

The above is again equivalent to

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

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

1.18.2 2-D Walsh Transform

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

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

The above is equivalent to:

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

1.18.3 2D inverse Walsh Transform

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

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

The above is equivalent to:

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

1.19 Hadamard Transform:

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

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

The above is equivalent to:

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

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

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

The above is equivalent to:

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

1.20 Discrete cosine transforms (DCT):

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

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

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

and the corresponding

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

inverse 1D DCT transform is simple F-1(u), i.e.: where

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

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

and the

corresponding inverse 2D DCT transform is simple F-1(u,v), i.e.: where The basic operation of

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

the DCT is as follows:

• The input image is N by M;

• f(i,j) is the intensity of the pixel in row i and column j;

• F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.

• For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.

• Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.

• The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level;

• 8 bit pixels have levels from 0 to 255.

• The Haar transform is based on a class of orthogonal matrices whose elements are either 1, –1, or 0 multiplied by powers of 2 . The Haar transform is a computationally efficient transform as the transform of an N-point 2 vector requires only 2(N – 1) additions and N multiplications..

1.21 Haar Transform

Algorithm to Generate Haar Basis The algorithm to generate Haar basis is given below:

Step 1 Determine the order of N of the Haar basis.
Step 2 Determine n where $n = \log_2 N$.
Step 3 Determine p and q.

(i) $0 \le p < n$–1
(ii) If $p = 0$ then $q = 0$ or $q = 1$
(iii) If $p \ne 0$, $1 \le q \le 2^p$

Step 4 Determine k.

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

Step 5 Determine Z.

$$Z \to [0, 1) \Rightarrow \left\{\frac{0}{N}, \quad \frac{1}{N}, \quad \cdots \quad \frac{N-1}{N}\right\}$$

Step 6

$$\text{If} \quad k = 0 \quad \text{then} \quad H(Z) = \frac{1}{\sqrt{N}}$$

Otherwise,

$$H_k(Z) = H_{pq}(Z) = \frac{1}{\sqrt{N}} \begin{cases} +2^{p/2} & \dfrac{(q-1)}{2^p} \le Z < \dfrac{(q-1/2)}{2^p} \\ -2^{p/2} & \dfrac{(q-1/2)}{2^p} \le Z < \dfrac{q}{2^p} \\ 0 & \text{otherwise} \end{cases}$$

1.22 Slant transform

The slant transform was introduced by Enomoto and Shibata as an orthogonal transform containing sawtooth waveforms or 'slant' basis vectors. A slant basis vector that is monotonically decreasing in constant steps from maximum to minimum has the sequency property and has a fast computational algorithm. Let SN denote an N × N slant matrix with N = 2n. Then

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

The S4 matrix is obtained by the following operation:

$$S_4 = \left[\begin{array}{cc:cc} 1 & 0 & 1 & 0 \\ a & b & -a & b \\ \hdashline 0 & 1 & 0 & -1 \\ -b & a & b & a \end{array}\right] \left[\begin{array}{c:c} S_2 & 0 \\ \hdashline 0 & S_2 \end{array}\right]$$

If a = 2b and b = 15, the slant matrix is given by

$$S_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

The sequency of the slant matrix of order four is given below:

$$S_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \begin{array}{l} \textbf{Sequency} \\ \text{Zero sign change} \\ \text{One sign change} \\ \text{Two sign changes} \\ \text{Three sign changes} \end{array}$$

From the sequency property, it is clear that the rows are ordered by the number of sign changes. The slant transform reproduces linear variations of brightness very well. However, its performance at edges is not as optimal as the KLT or DCT. Because of the 'slant' nature of the lower order coefficients, its effect is to smear the edges

1.23 Hotelling Transform:

The KL transform is named after Kari Karhunen and Michel Loeve who developed it as a series expansion method for continuous random processes. Originally, Harold Hotelling studied the discrete formulation of the KL transform and for this reason, the KL transform is also known as the Hotelling transform. The KL transform is a reversible linear transform that exploits the statistical properties of a vector representation. The basic functions of the KL transform are orthogonal eigen vectors of the covariance matrix of a data set. A KL transform optimally decorrelates the input data. After a KL transform, most of the 'energy' of the transform coefficients is concentrated within the first few components. This is the energy compaction property of a KL transform.

1.23.1 Drawbacks of KL Transforms

The two serious practical drawbacks of KL transform are the following:
i. A KL transform is input-dependent and the basic function has to be calculated for each signal model on which it operates. The KL bases have no specific mathematical structure that leads to fast implementations.
ii. The KL transform requires (m2) multiply/add operations. The DFT and DCT require (log2 m) multiplications.

1.23.2 Applications of KL Transforms

(i) Clustering Analysis The KL transform is used in clustering analysis to determine a new coordinate system for sample data where the largest variance of a projection of the data lies on the first axis, the next largest variance on the second axis, and so on. Because these axes are orthogonal, this approach allows for reducing the dimensionality of the data set by eliminating those coordinate axes with small variances. This data-reduction technique is commonly referred as Principle Component Analysis (PCA).
(ii) Image Compression The KL transform is heavily utilised for performance evaluation of compression algorithms since it has been proven to be the optimal transform for the compression of an image sequence in the sense that the KL spectrum contains the largest number of zero-valued coefficients.

Chapter 2

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

Course Outcomes

After successful completion of this module, students should be able to:

CO 3	**Construct** image intensity transformation and filtering techniques for image enhancement in the spatial and frequency domain.	Apply

2.1 Introduction

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

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

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

s = T (r)

where f(x, y) is the input image, g(x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular subimage area centered at (x, y), as shows. The center of the subimage is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.
The simplest form of T is when the neighborhood is of size 1*1 (that is, a single pixel). In this case, g depends only on the value of f at (x, y), and T becomes a gray-level (also called an intensity or mapping) transformation function of the form

s = T (r)
where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of ”r” to each value of ”s”. For example, if T(r) has the form the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation
function into a narrow range of s, toward black.The opposite effect takes place for values of r above m.
In the limiting case shown in Fig. T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.
One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig, in which the values of the mask coefficients determine the nature of the process, such as image sharpening. Enhancement techniques based on this type of approach often are referred to as mask processing or filtering.

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

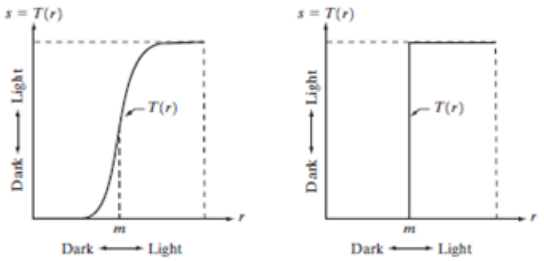


FIGURE 2.1: Gray level transformation functions for contrast enhancement

2.2 Basic gray level transformations:

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

2.2.1 Linear transformation:

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below:

2.2.2 Negative transformation:

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

2.2.3 Image negative:

The image negative with gray level value in the range of [0, L-1] is obtained by negative transformation given by S = T(r) or S = L -1 – r
Where r= gray level value at pixel (x,y)
L is the largest gray level consists in the image It results in getting photograph negative. It is useful when for enhancing white details embedded in dark regions of the image. The overall graph of these transitions has been shown below.

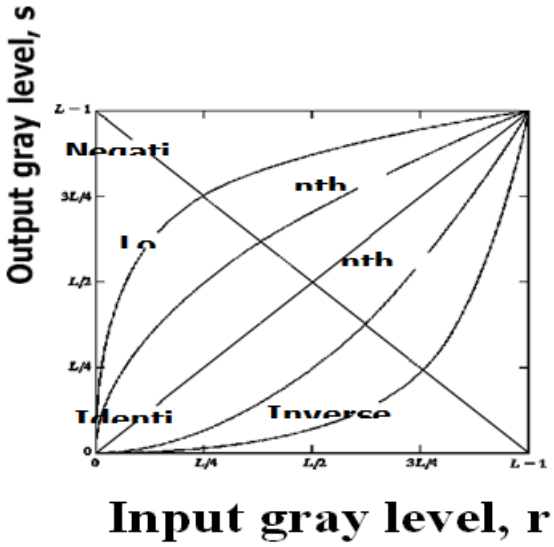


FIGURE 2.2: Some basic gray-level transformation functions used for image enhancement.

In this case the following transition has been done.
s = (L – 1) – r
since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this
s = 255 – r
So each value is subtracted by 255 and the result image has been shown above. So what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.
It has been shown in the graph below.

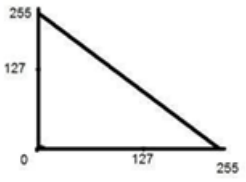


FIGURE 2.3: Negative transformations.

2.2.4 Logarithmic transformations:

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation **Log transformations:** The log transformations can be defined by this formula
s = c log(r + 1).

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1. During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.An another way of representing

2.2.5 Log transformations:

Enhance details in the darker regions of an image at the expense of detail in brighter regions. T(f) = C * log (1+r)

• Here C is constant and r greater than equal to 0 • The shape of the curve shows that this transformation maps the narrow range of low gray level values in the input image into a wider range of output image.

• The opposite is true for high level values of input image.

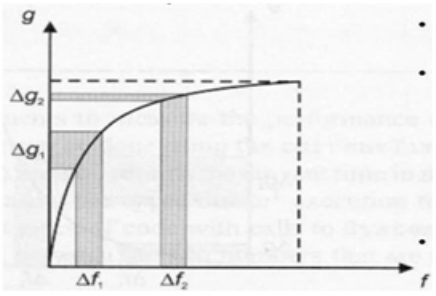


FIGURE 2.4: log transformation curve input vs output

2.2.6 POWER – LAW TRANSFORMATIONS:

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

s=crγ

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

2.2.7 Piecewise-Linear Transformation Functions:

A complementary approach to the methods discussed in the previous three sections is to use piecewise linear functions. The principal advantage of piecewise linear functions over the types of

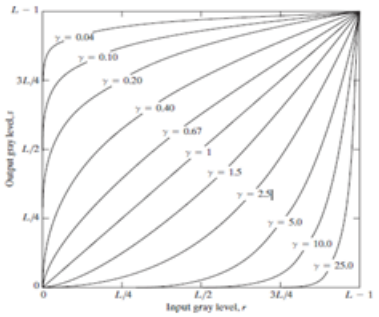


FIGURE 2.5: Plot of the equation S = crγ for various values of γ (c =1 in all cases).

functions we have discussed thus far is that the form of piecewise functions can be arbitrarily complex.

The principal disadvantage of piecewise functions is that their specification requires considerably more user input. Contrast stretching: One of the simplest piecewise linear functions are a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamic angle in the imaging sensor, or even wrong setting of a lens aperture during image acquisition.

S= T(r)

Figure x(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation Function. If r1=s1 and r2=s2, the transformation is a linear function that produces No changes in gray levels. If r1=r2, s1=0and s2= L-1, the transformation Becomes a thresholding function that creates a binary image, Intermediate values of ar1, s1b and ar2, s2b produce various degrees Of spread in the gray levels of the output image, thus affecting its contrast.

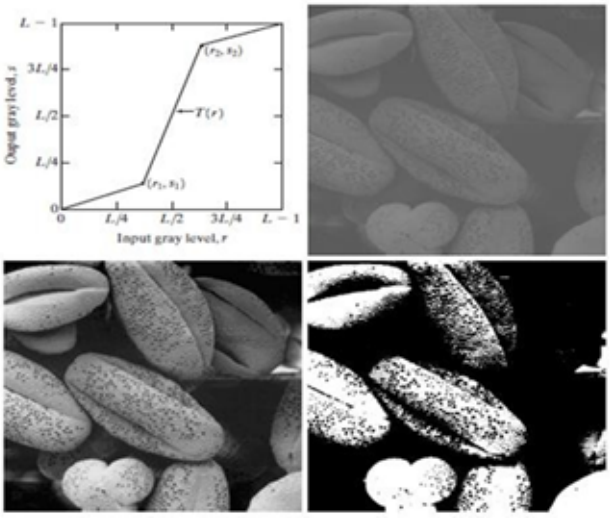


FIGURE 2.6: (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding

Figure x(b) shows an 8-bit image with low contrast. Fig. x(c) shows the result of contrast stretching, obtained by setting (r1, s1)=(rmin, 0) and (r2, s2)=(rmax,L-1) where rmin and rmax denote

the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1] . Finally, Fig. x(d) shows the result of using the thresholding function defined previously, with r1=r2=m, the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

2.2.8 Gray-level slicing:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.
There are several ways of doing level slicing, but most of them are variations of two basic themes .One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.
This transformation, shown in Fig. y(a), produces a binary image. The second approach, based on the transformation shown in Fig .y (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure y (c) shows a gray-scale image, and Fig. y(d) shows the result of using the transformation in Fig. y(a).Variations of the two transformations shown in Fig. are easy to formulate.

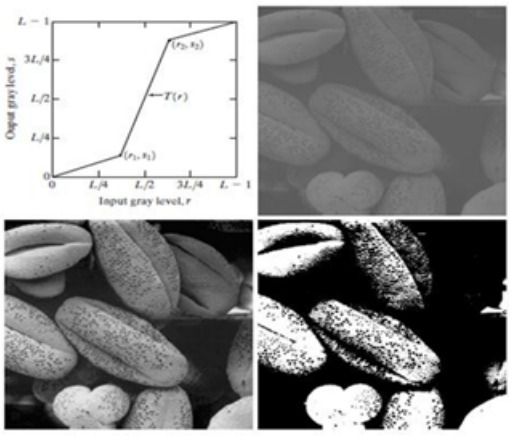


FIGURE 2.7: Gray level sclicing

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

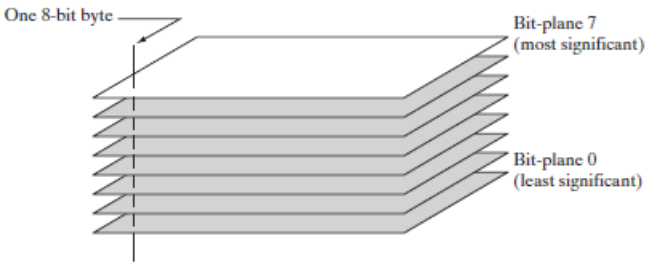


FIGURE Bit-plane representation of an 8-bit image.

FIGURE 2.8: Bit plane representation of 8 bits

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

2.3 Histogram Processing:

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function of the form
H(rk)=nk
where rk is the kth gray level and nk is the number of pixels in the image having the level rk.. A normalized histogram is given by the equation p(rk)=nk/n for k=0,1,2,. . . .,L-1 ,P(rk) gives the estimate of the probability of occurrence of gray level rk. The sum of all components of a normalized histogram is equal to 1.The histogram plots are simple plots of H(rk)=nk versus rk. In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale. In case of bright image the histogram components are baised towards the high side of the gray scale. The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.
The components of the histogram in the high contrast image cover a broad range of the gray scale. The net effect of this will be an image that shows a great deal of gray levels details and has high

dynamic range.

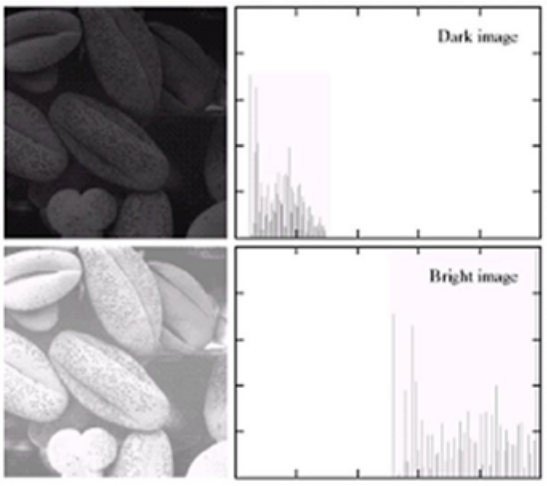


FIGURE 2.9: Histogram Reprasentation of dark and brightness of image

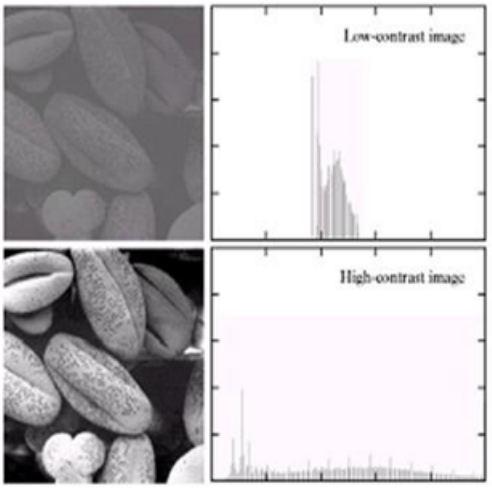


FIGURE 2.10: Histogram Reprasentation low and high contrast of an image

2.4 Image enhancement in frequency domain

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

2.4.1 Ideal low-pass filter:

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

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

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

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

Where as P and Q are the padded sizes from the basic equations
Wraparound error in their circular convolution can be avoided by padding these functions with zeros,

Visualization: ideal low pass filter:
As shown in fig. below

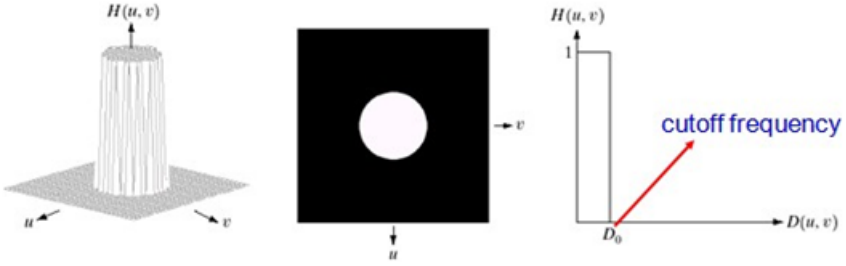


FIGURE 2.11: Fig: ideal low pass filter 3-D view and 2-D view and line graph

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

As the cutoff frequency decreases,
• image becomes more blurred
• Noise becomes increases
• Analogous to larger spatial filter sizes
The severe blurring in this image is a clear indication that most of the sharp detail information in the picture is contained in the 13% power removed by the filter. As the filter radius is increases less and less power is removed, resulting in less blurring. **Why is there ringing?**
Ideal low-pass filter function is a rectangular function The inverse Fourier transform of a rectangular function is a sinc function.

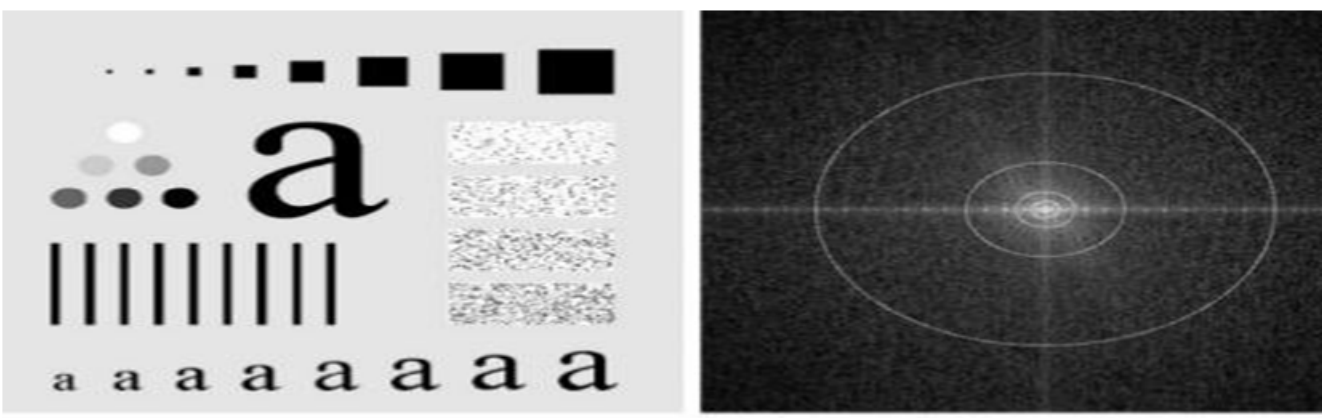


FIGURE 2.12: Fig: i(a) Test patter of size 688x688 pixels (b) its Fourier spectrum

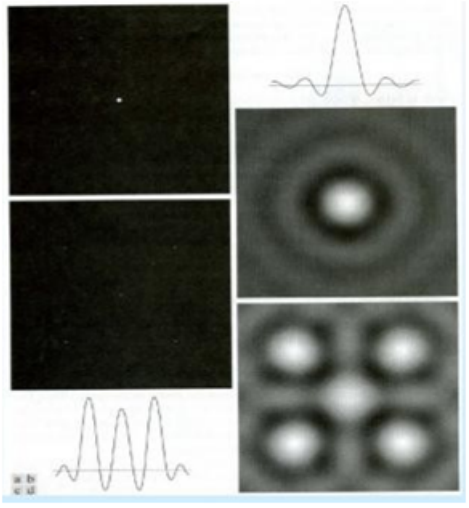


FIGURE 2.13: Fig: (a) original image, (b)-(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160 and 460

2.4.2 Butterworth low-pass filter:

Transfor function of a Butterworth low pass filter (BLPF) of order n, and with cutoff frequency at a distance D0 from the origin, is defined as

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

Transfer function does not have sharp discontinuity establishing cutoff between passed and filtered frequencies.
Cut off frequency D0 defines point at which H (u,v) = 0.5

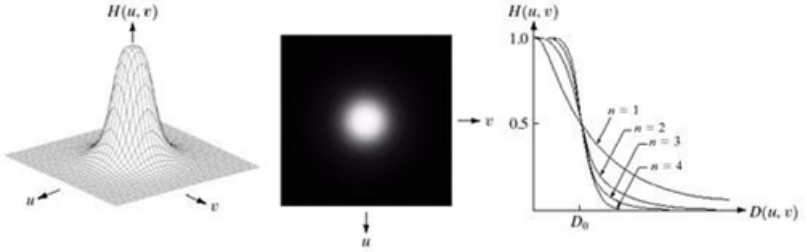


FIGURE 2.14: Fig: (a) perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c)Filter radial cross sections of order 1 through 4

2.4.3 Butterworth low-pass filters of different frequencies:

frequencies at the radii Fig. shows the results of applying the BLPF of eq. to fig.(a), with n=2 and D0 equal to the five radii in fig.(b) for the ILPF, we note here asmooth transition in blurring as a function of increasing cutoff frequency. Moreover, no ringing is visible in any of the images processed with this particular BLPF, a fact attributed to the filter's smooth transition between low and high frequencie A BLPF of order 1 has no ringing in the spatial domain. Ringing generally is imperceptible in filters of order 2, but can become significant in filters of higher orde Fig. Shows a comparison between the spatial representations of BLPFs of various orders (using a cutoff frequency of 5 in all cases). Shown also is the intensity profile along a horizontal scan line through the center of each filter. The filter of order 2 does show mild ringing and small negative values, but they certainly are less pronounced than in the ILPF. A butter worth filter of order 20 exhibits characteristics similar to those of the ILPF (in the limit, both filters are identical).

FIGURE 2.15: Fig: (a) Original image.(b)-(f) Results of filtering using BLPFs of order 2, with cutoff

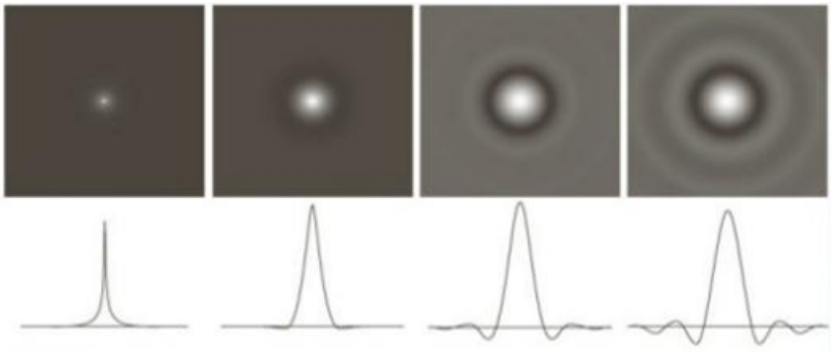


FIGURE 2.16: Fig: (a)-(d) Spatial representation of BLPFs of order 1, 2, 5 and 20 and corresponding intensity profiles through the center of the filters

2.4.4 Gaussian low pass filters:

The form of these filters in two dimensions is given by

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

• This transfer function is smooth, like Butterworth filter.

• Gaussian in frequency domain remains a Gaussian in spatial domain

• Advantage: No ringing artifacts.

Where D0 is the cutoff frequency. When D (u,v) = D0, the GLPF is down to 0.607 of its maximum value. This means that a spatial Gaussian filter, obtained by computing the IDFT of above equation. will have no ringing. Fig. Shows a perspective plot, image display and radial cross sections of a GLPF function.

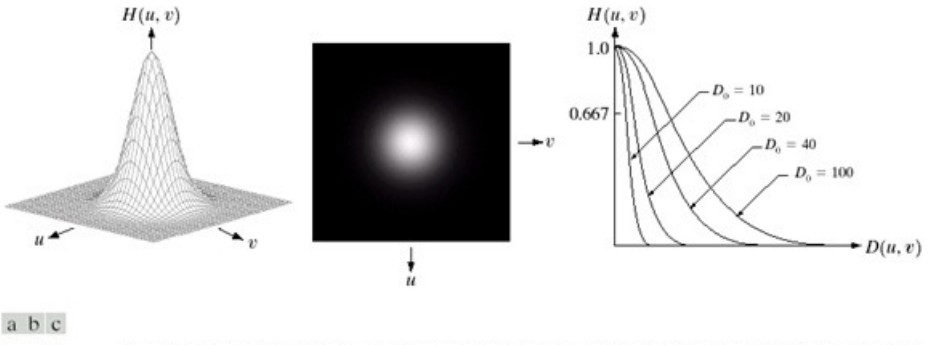


FIGURE 2.17: Fig: Fig. (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c). Filter radial cross sections for various values of D0

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

FIGURE 2.18: Fig:(a) Original image. (b)-(f) Results of filtering using GLPFs with cutoff frequencies at the radii.

FIGURE 2.19: Fig: (a) Original image (784x 732 pixels). (b) Result of filtering using a GLPF with D0 = 100. (c) Result of filtering using a GLPF with D0 = 80. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

2.4.5 Image sharpening using frequency domain filters:

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

The filter function H(u,v) are understood to be discrete functions of size PxQ; that is the discrete

frequency variables are in the range u = 0,1,2,. . . . . .P-1 and v = 0,1,2,. . . . . .Q-1. The meaning of sharpening is

• Edges and fine detail characterized by sharp transitions in image intensity

• Such transitions contribute significantly to high frequency components of Fourier transform

• Intuitively, attenuating certain low frequency components and preserving high frequency components result in sharpening.

• Intended goal is to do the reverse operation of low-pass filters When low-pass filter attenuated frequencies, high-pass filter passes them

• When high-pass filter attenuates frequencies, low-pass filter passes them. A high pass filter is obtained from a given low pass filter using the equation.

H hp (u,v) = 1- Htp (u,v) Where Hlp (u,v) is the transfer function of the low-pass filter. That is when the low- pass filter attenuates frequencies; the high-pass filter passed them, and vice-versa. We consider ideal, Butter-worth, and Gaussian high-pass filters. As in the previous section, we illustrate the characteristics of these filters in both the frequency and spatial domains. Fig. Shows typical 3-D plots, image representations and cross sections for these filters. As before, we see that the Butter-worth filter represents a transition between the sharpness of the ideal filter and the broad smoothness of the Gaussian filter. Fig. discussed in the sections the follow, illustrates what these filters look like in the spatial domain. The spatial filters were obtained and displayed by using the procedure used.

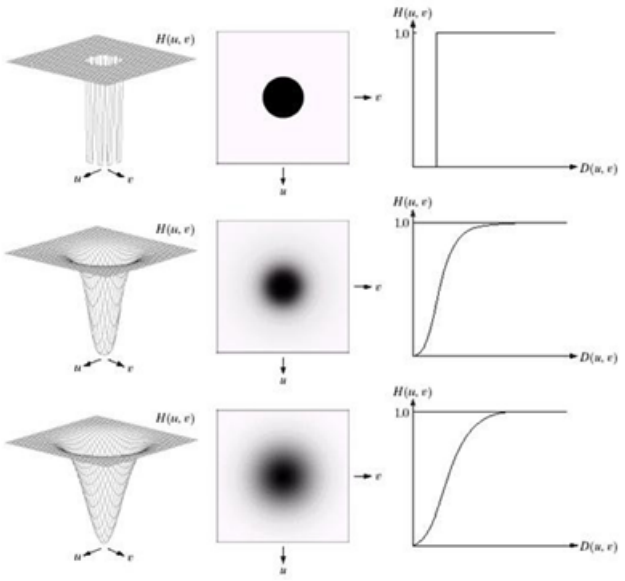


FIGURE 2.20: Fig:Top row: Perspective plot, image representation, and cross section of a typical ideal high-pass filter. Middle and bottom rows

2.4.6 Ideal high-pass filter:

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

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

Where D0 is the cutoff frequency and D(u,v) is given by eq. As intended, the IHPF is the opposite of the ILPF in the sense that it sets to zero all frequencies inside a circle of radius D0 while passing, without attenuation, all frequencies outside the circle. As in case of the ILPF, the IHPF is not physically realizable. **Spatial representation of high pass filters:** We can expect IHPFs to

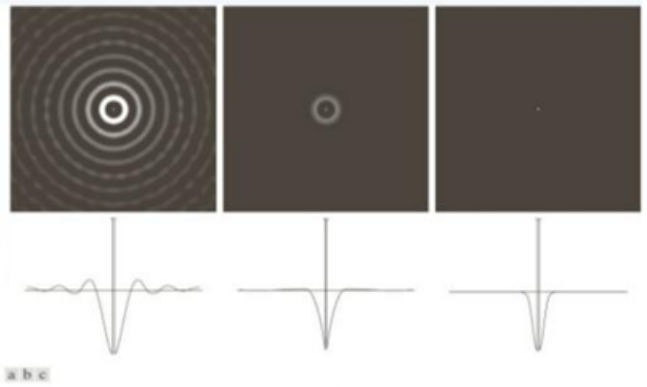


FIGURE 2.21: Fig: Spatial representation of typical (a) ideal (b) Butter-worth and (c) Gaussian frequency domain high-pass filters, and corresponding intensity profiles through their centers.

have the same ringing properties as ILPFs. This is demonstrated clearly in Fig. Which consists of various IHPF results using the original image in Fig.(a) with D0 set to 30, 60,and 160 pixels, respectively. The ringing in Fig. (a) is so severe that it produced distorted, thickened object boundaries (e.g., look at the large letter “a”). Edges of the top three circles do not show well because they are not as strong as the other edges in the image (the intensity of these three objects is much closer to the background intensity, giving discontinuities of smaller magnitude).

Filtered results: IHPF: The situation improved somewhat with D0 = 60. Edge distortion is quite

FIGURE 2.22: Fig: Results of high-pass filtering the image in Fig.(a) using an IHPF with D0 = 30, 60, and 160.

evident still, but now we begin to see filtering on the smaller objects. Due to the now familiar

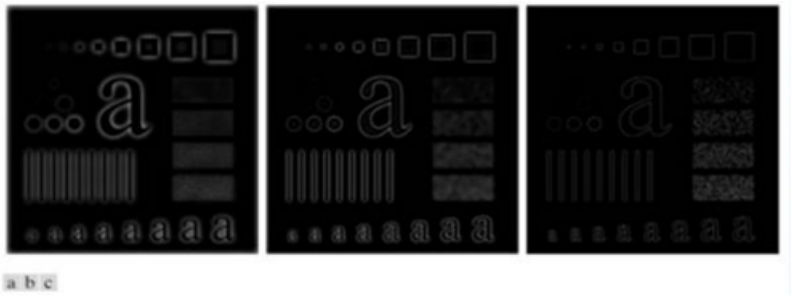
inverse relationship between the frequency and spatial domains, we know that the spot size of this filter is smaller than the spot of the filter with D0 = 30. The result for D0 = 160 is closer to what a high-pass filtered image should look like. Here, the edges are much cleaner and less distorted, and the smaller objects have been filtered properly.Of course, the constant background in all images is zero in these high-pass filtered images because highpass filtering is analogous to differentiation in the spatial domain.

2.4.7 Butter-worth high-pass filters:

A 2-D Butter-worth high-pass filter (BHPF) of order n and cutoff frequency D0 is defined as

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

Where D(u,v) is given by Eq.(3). This expression follows directly from and (6). The middle row of Fig. shows an image and cross section of the BHPF function.
Butter-worth high-pass filter to behave smoother than IHPFs. Figure shows the
performance of a BHPF of order 2 and with D0 set to the same values as in Figure shows The boundaries are much less distorted than in below figure. even for the smallest value of cutoff frequency.
Filtered results: BHPF:



a b c

FIGURE 2.23: Fig: Fig. Results of high-pass filtering the image in above figure (a) using a BHPF of order 2 with D0 = 30, 60, and 160 corresponding to the circles in above figure (b). These results are much smoother than those obtained with an IHPF.

2.4.8 Gaussian high-pass filters:

The transfer function of the Gaussian high-pass filter(GHPF) with cutoff frequency locus at a distance D0 from the center of the frequency rectangle is given by

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

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

Chapter 3

IMAGE RESTORATION AND FILTERING

Course Outcomes

After successful completion of this module, students should be able to:

CO 4	**Analyze** the image restoration in the spatial and frequency domains to deal with noise models for removing degradation from given image.	Analyze

3.1 Introduction

The distortion correction equations yield non integer values for x' and y'. Because the distorted image g is digital, its pixel values are defined only at integer coordinates. Thus using non integer values for x' and y' causes a mapping into locations of g for which no gray levels are defined. Inferring what the gray-level values at those locations should be, based only on the pixel values at integer coordinate locations, and then becomes necessary. The technique used to accomplish this is called gray-level interpolation. The simplest scheme for gray-level interpolation is based on a nearest neighbor approach.This method, also called zero-order interpolation, is illustrated in Fig. 3.1: This figure shows The mapping of integer (x, y) coordinates into fractional coordinates (x', y') by means of following equations

and

x' = c1x + c2y + c3xy + c4

y' = c5x + c6y + c7xy + c8

(A) The selection of the closest integer coordinate neighbor to (x', y'); and (B) The assignment of the gray level of this nearest neighbor to the pixel located at (x, y).

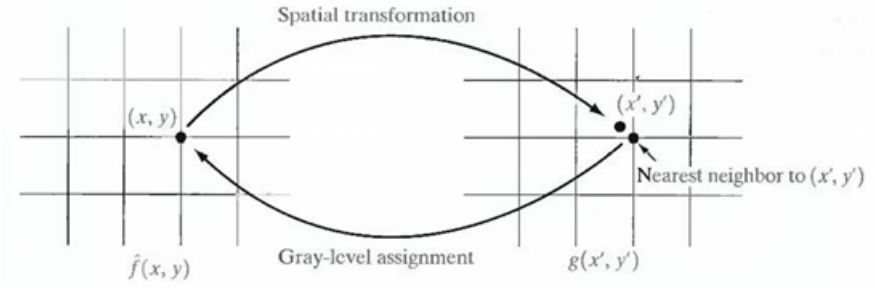


FIGURE 3.1: Gray-level interpolation based on the nearest neighbor concept

Although nearest neighbor interpolation is simple to implement, this method often has the drawback of producing undesirable artifacts, such as distortion of straight edges in images of high resolution. Smoother results can be obtained by using more sophisticated techniques, such as cubic convolution interpolation, which fits a surface of the sin(z)/z type through a much larger number of neighbors (say, 16) in order to obtain a smooth estimate of the gray level at any Desired point. Typical areas in which smoother approximations generally are required include 3-D graphics and medical imaging. The price paid for smoother approximations is additional computational burden. For general-purpose image processing a bilinear interpolation approach that uses the gray levels of the four nearest neighbors usually is adequate. This approach is straightforward. Because the gray level of each of the four integral nearest neighbors of a non integral pair of coordinates (x', y') is known, the gray-level value at these coordinates, denoted v(x', y'), can be interpolated from the values of its neighbors by using the relationship

v (x', y') = ax' + by' + c x' y' + d

where the four coefficients are easily determined from the four equations in four unknowns that can be written using the four known neighbors of (x', y'). When these coefficients have been determined, v(x', y') is computed and this value is assigned to the location in fx, y) that yielded the spatial mapping into location (x', y'). It is easy to visualize this procedure with the aid of Fig.3.1. The exception is that, instead of using the gray-level value of the nearest neighbor to (x', y'), we actually interpolate a value at location (x', y') and use this value for the gray-level assignment at (x, y).

3.2 Wiener filter used for image restoration.

The inverse filtering approach makes no explicit provision for handling noise. This approach incorporates both the degradation function and statistical characteristics of noise into the restoration process. The method is founded on considering images and noise as random processes and the objective is to find an estimate f of the uncorrupted image f such that the mean square error between them is minimized. This error measure is given by

e2 = E (f- f)2

where E • is the expected value of the argument. It is assumed that the noise and the image are

uncorrelated; that one or the other has zero mean; and that the gray levels in the estimate are a linear function of the levels in the degraded image. Based on these conditions, the minimum of the error function is given in the frequency domain by the expression

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v)\\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\\ &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)\end{aligned}$$

begin fleshleft

end fleshleft where we used the fact that the product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. This result is known as the Wiener filter, after N. Wiener [1942], who first proposed the concept in the year shown. The filter, which consists of the terms inside the brackets, also is commonly referred to as the minimum mean square error filter or the least square error filter. The Wiener filter does not have the same problem as the inverse filter with zeros in the degradation function, unless both H(u, v) and S(u, v) are zero for the same value(s) of u and v. The terms in above equation are as follows: H (u, v) = degradation function H(u, v) = complex conjugate of H (u, v)

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

$$S_\eta(u,v) = |N(u,v)|^2 = \text{power spectrum of the noise}$$

$$S_f(u,v) = |F(u,v)|^2 = \text{power spectrum of the un-degraded image.}$$

As before, H (u, v) is the transform of the degradation function and G (u, v) is the transform of the degraded image. The restored image in the spatial domain is given by the inverse Fourier transform of the frequency-domain estimate F (u, v). Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter. When we are dealing with spectrally white noise, the spectrum N(U,V) 2 is a constant, which simplifies things considerably. However, the power spectrum of the undegraded image seldom is known. An approach used frequently when these quantities are not known or cannot be estimated is to approximate the equation

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

as

where K is a specified constant.

3.3 of the Image Degradation/Restoration Process.

The Fig. 3.1shows, the degradation process is modeled as a degradation function that, together with an additive noise term, operates on an input image f(x, y) to produce a degraded image g(x, y). Given g(x, y), some knowledge about the degradation function H, and some knowledge about the additive noise term (x, y), the objective of restoration is to obtain an estimate f(x, y) of the original image. the estimate should be as close as possible to the original input image and, in general, the more we know about H and , the closer f(x, y) will be to f(x, y). The degraded image is given in the spatial domain by

where h (x, y) is the spatial representation of the degradation function and, the symbol indicates convolution. Convolution in the spatial domain is equal to multiplication in the frequency domain, hence G (u, v) = H (u, v) F (u, v) + N (u, v) where the terms in capital letters are the Fourier transforms of the corresponding terms in above equation. The restoration filters used when the

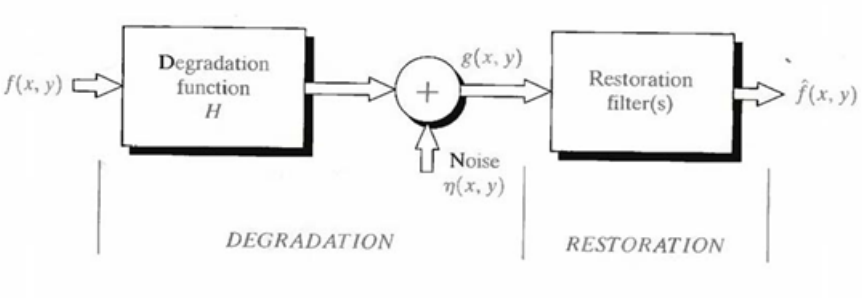


FIGURE 3.2: model of the image degradation/restoration process

image degradation is due to noise only.

If the degradation present in an image is only due to noise, then,

g (x, y) = f (x, y) + (x, y)
G (u, v) = F (u, v) + N (u, v)
The restoration filters used in this case are,

1. Mean filters 2. Order static filters and 3. Adaptive filters sectionMean filters.

There are four types of mean filters. They are

3.3.1 Arithmetic mean filter

This is the simplest of the mean filters. Let Sxy represent the set of coordinates in a rectangular subimage window of size m X n, centered at point (x, y).The arithmetic mean filtering process computes the average value of the corrupted image g(x, y) in the area defined by Sxy. The value of

the restored image f at any point (x, y) is simply the arithmetic mean computed using the pixels in

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

the region defined by Sxy. In other words This operation can be implemented using a convolution mask in which all coefficients have value 1/mn

3.3.2 Geometric mean filter

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

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

Here, each restored pixel is given by the product of the pixels in the subimage window, raised to the power 1/mn. A geometric mean filter achieves smoothing comparable to the arithmetic mean filter, but it tends to lose less image detail in the process.

3.3.3 Harmonic mean filter

The harmonic mean filtering operation is given by the expression The harmonic mean filter works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

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

3.3.4 Contra harmonic mean filter

The contra harmonic mean filtering operation yields a restored image based on the expression where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q, the filter eliminates pepper noise. For negative values of Q it eliminates salt noise. It cannot do both simultaneously. Note that the contra harmonic filter reduces to the arithmetic mean filter if Q = 0, and to the harmonic mean filter if Q = -1.

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

3.4 The Order-Statistic Filters.

There are four types of Order-Statistic filters. They are

3.4.1 Median filter

The best-known order-statistics filter is the median filter, which, as its name implies, replaces the

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

value of a pixel by the median of the gray levels in the neighborhood of that pixel:
The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.

3.4.2 Max and min filters

Although the median filter is by far the order-statistics filler most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called max filter, given by This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area Sxy. The

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

0th percentile filter is the min filter. This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

3.4.3 Midpoint filter

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

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

Note that this filter combines order statistics and averaging. This filter works best for randomly distributed noise, like Gaussian or uniform noise.

3.4.4 Alpha - trimmed mean filter

It is a filter formed by deleting the d/2 lowest and the d/2 highest gray-level values of g(s, t) in the neighborhood Sxy. Let gr (s, t) represent the remaining mn - d pixels. A filter formed by averaging these remaining pixels is called an alpha-trimmed mean filter:

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

3.4.5 The Adaptive Filters.

Adaptive filters are filters whose behavior changes based on statistical characteristics of the image inside the filter region defined by the m X n rectangular window Sxy.

3.4.6 Adaptive, local noise reduction filter:

The simplest statistical measures of a random variable are its mean and variance. These are reasonable parameters on which to base an adaptive filler because they are quantities closely related to the appearance of an image. The mean gives a measure of average gray level in the region over which the mean is computed, and the variance gives a measure of average contrast in that region.

3.5 Adaptive median filter:

The median filter performs well as long as the spatial density of the impulse noise is not large (as a rule of thumb, Pa and Pb less than 0.2). The adaptive median filtering can handle impulse noise

with probabilities even larger than these. An additional benefit of the adaptive median filter is that it seeks to preserve detail while smoothing nonimpulse noise, something that the ”traditional” median filter does not do. The adaptive median filter also works in a rectangular window area Sxy. Unlike those filters, however, the adaptive median filter changes (increases) the size of Sxy during filter operation, depending on certain conditions. The output of the filter is a single value used to replace the value of the pixel at (x, y), the particular point on which the window Sxy is centered at a given time.

Consider the following notation: Consider the following notation:

Z min = minimum gray level value in S xy z max = maximum gray level value in Sxy zmcd = median of gray levels in Sxy
Zxy = gray level at coordinates (x, y) Smax = maximum allowed size of Sxy.

The adaptive median filtering algorithm works in two levels, denoted level A and level B, as follows:
Level A: A1 = zmed - zmin
A2 = zmed - zmax

If $A1 > 0$ AND $A2 < 0$, Go to level B

Else increase the window size

If window size $\leq S_{max}$ repeat level A

Else output z_{xy}

Level B: $B1 = Z_{xy} - Z_{min}$

$B2 = z_{xy} - z_{max}$

If $B1 > 0$ AND $B2 < 0$, output zxy

Else output z_{med}

3.6 Image Formation Model.

An image is represented by two-dimensional functions of the form f(x, y). The value or amplitude of f at spatial coordinates (x, y) is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated from a physical process, its values are

proportional to energy radiated by a physical source (e.g., electromagnetic waves). As a consequence, f(x, y) must be nonzero and finite; that is,

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

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

B) The amount of illumination reflected by the objects in the scene.

Appropriately, these are called the illumination and reflectance components and are denoted by i (x, y) and r (x, y), respectively. The two functions combine as a product to form f (x, y).

$$\mathbf{f(x, y) = i(x, y)\, r(x, y)} \qquad \mathbf{\ldots . (2)}$$

where

$$0 < i(x, y) < \infty \qquad \ldots . (3)$$

and

$$0 < r(x, y) < 1 \qquad \ldots . (4)$$

Equation (4) indicates that reflectance is bounded by 0 (total absorption) and 1 (total reflectance).The nature of i (x, y) is determined by the illumination source, and r (x, y) is determined by the characteristics of the imaged objects. It is noted that these expressions also are applicable to images formed via transmission of the illumination through a medium, such as a chest X-ray.

3.7 Inverse filtering.

The simplest approach to restoration is direct inverse filtering, where F (u, v), the transform of the original image is computed simply by dividing the transform of the degraded image, G (u, v), by

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

The divisions are between individual elements of the functions

But G (u, v) is given by

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

Hence

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

the degradation function b leshleft
leshleft It tells that even if the degradation function is known the undegraded image cannot be recovered [the inverse Fourier transform of F(u, v)] exactly because N(u, v) is a random function whose Fourier transform is not known. If the degradation has zero or very small values, then the ratio N(u, v)/H(u, v) could easily dominate the estimate F(u, v).One approach to get around the zero or small-value problem is to limit the filter frequencies to values near the origin. H (0, 0) is equal to the average value of h(x, y) and that this

is usually the highest value of H (u, v) in the frequency domain. Thus, by limiting the analysis to frequencies near the origin, the probability of encountering zero values is reduced.

3.8 Noise Probability Density Functions.

The following are among the most common PDFs found in image processing applications.

3.8.1 Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called normal) noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best. The PDF of a Gaussian random variable, z, is given by

Where z represents gray level, μ is the mean of average value of z, and a σ is its standard deviation. The standard deviation squared, σ^2, is called the variance of z. A plot of this function is shown in Fig. 5.10. When z is described by Eq. (1), approximately 70% of its values will be in the range [(μ - σ), (μ +σ)], and about 95% will be in the range [(μ - 2σ), (μ + 2σ)].

Rayleigh noise

The PDF of Rayleigh noise is given by

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

The mean and variance of this density are given by

$$\mu = a + f\ \text{Mb/4}$$

$$\sigma^2 = b\ (4 - \Pi)/4$$

Erlang (Gamma) noise

The PDF of Erlang noise is given by

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

where the parameters are such that a > 0, b is a positive integer, and "!" indicates factorial. The mean and variance of this density are given by

$$\mu = b / a$$

$$\sigma^2 = b / a^2$$

Exponential noise

The PDF of exponential noise is given by

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

The mean of this density function is given by

$$\mu = 1 / a$$

$$\sigma^2 = 1 / a^2$$

This PDF is a special case of the Erlang PDF, with b = 1.

Uniform noise

The PDF of uniform noise is given by

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

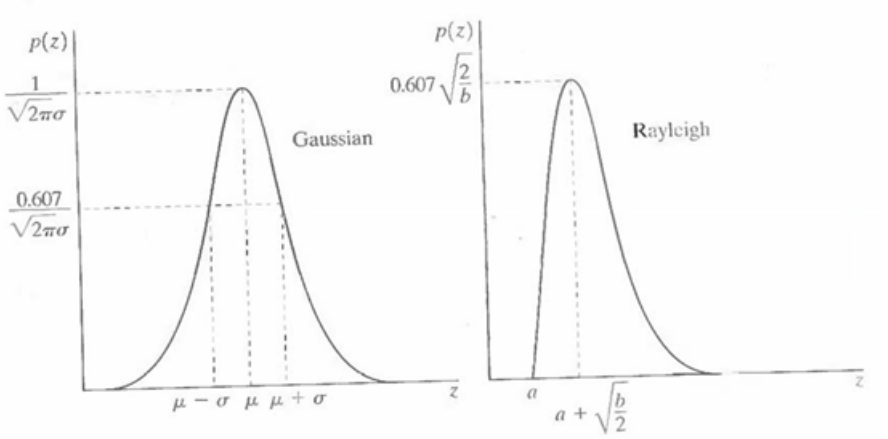
Impulse (salt-and-pe

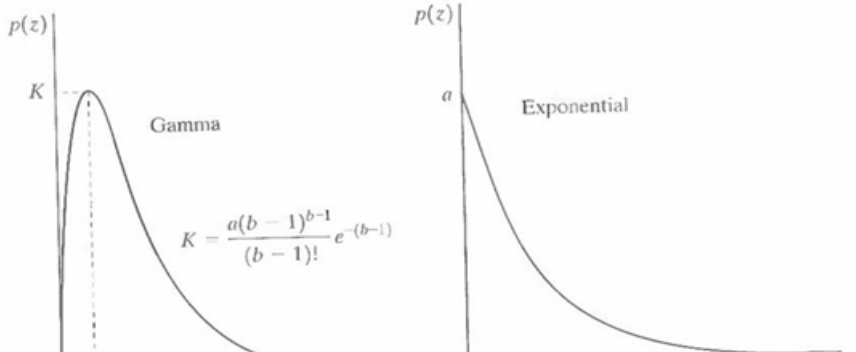
The PDF of (bipo

The mean of this density function is given by

$$\mu = a + b / 2$$

$$\sigma^2 = (b - a)^2 / 12$$





3.8.2 Enumerate the differences between the 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 arrange 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 modeling 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 a image restoration technique.

3.8.3 Iterative nonlinear restoration using the Lucy–Richardson algorithm.

Lucy-Richardson algorithm is a nonlinear restoration method used to recover a latent image which is blurred by a Point Spread Function (psf). It is also known as Richardson-Lucy de-convolution. With as the point spread function, the pixels in observed image are expressed as,

3.8.4 DAMPAR

The DAMPAR parameter is a scalar parameter which is used to determine the deviation of resultant image with the degraded image (g). The pixels which gel deviated from their original value within the DAMPAR, for these pixels iterations are cancelled so as to reduce noise generation and present essential image information.

3.8.5 WEIGHT

WEIGHT parameter gives a weight to each and every pixel. It is array of size similar to that of degraded image (g). In applications where a pixel leads to improper image is removed by assigning it to a weight as 0'. The pixels may also be given weights depending upon the flat-field correction, which is essential according to image array. Weights are used in applications such as blurring with specified psf. They are used to remove the pixels which are pre9ent at the boundary of the image and are blurred separately by psf. If the array size of psf is n x n then the width of weight of border of zeroes being used is ceil (n / 2).

Chapter 4

COLOR IMAGE PROCESSING

Course Outcomes

After successful completion of this module, students should be able to:

CO 5	**Apply** region-based morphological operations and edge-based image segmentation techniques for detection of objects in images to remove the imperfections in the structure of the image.	Apply

4.1 Color models, pseudo color image processing:

First-order derivatives of a digital image are based on various approximations of the 2-D gradient. The gradient of an image f (x, y) at location (x, y) is defined as the vector It is well known from

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

vector analysis that the gradient vector points in the direction of maximum rate of change of f at coordinates (x, y). An important quantity in edge detection is the magnitude of this vector, denoted by Af, where These derivatives can be implemented for an entire image by using the masks shown in Fig. Masks of size 2 X 2 are awkward to implement because they do not have a clear center. An approach using masks of size 3 X 3 is given by Figure: A 3 X 3 region of an image (the z's are gray-level values) and various masks used to compute the gradient at point labeled z5. A weight value of 2 is used to achieve some smoothing by giving more importance to the center

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n).$$

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = [G_x^2 + G_y^2]^{1/2}.$$

$$G_x = (z_9 - z_5)$$

$$G_y = (z_8 - z_6).$$

point. Figures 1.1(f) and (g), called the Sobel operators, and are used to implement these two equations. The Prewitt and Sobel operators are among the most used in practice for computing digital gradients. The

Prewitt masks are simpler to implement than the Sobel masks, but the latter have slightly superior noise-suppression characteristics, an important issue when dealing with derivatives. Note that the coefficients in all the masks shown in Fig sum to 0, indicating that they give a response of 0 in areas of constant gray level, as expected of a derivative operator. The masks just discussed are used to obtain the gradient components Gx and Gy. Computation of the gradient requires that these two components be combined. However, this implementation is not always desirable because of the computational burden required by squares and square roots. An approach used frequently is to approximate the gradient by absolute values: This equation is much more attractive computationally, and it still preserves relative changes in gray levels. However, this is not an issue when masks such as the Prewitt and Sobel masks are used to compute Gx and Gy. It is possible to modify the 3 X 3 masks in Fig. 4.1 so that they have their strongest responses along the diagonal directions. The two additional Prewitt and Sobel masks for detecting discontinuities in the diagonal directions are shown in Fig.

The Laplacian of a 2-D function f(x, y) is a second-order derivative defined as where the z's are defined in Fig. A digital approximation including the diagonal neighbors is given by For a 3 X 3 region, one of the two forms encountered most frequently in practice is

4.1.1 EDGE DETECTION

Intuitively, an edge is a set of connected pixels that lie on the boundary between two regions. Fundamentally, an edge is a "local" concept whereas a region boundary, owing to the way it is

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7).$$

defined, is a more global idea. A reasonable definition of ”edge” requires the ability to measure gray-level transitions in a meaningful way. We start by modeling an edge intuitively. This will lead us to formalism in which ”meaningful” transitions in gray levels can be measured. Intuitively, an ideal edge has the properties of the model shown in Fig.. An ideal edge according to this model is a set of connected pixels (in the vertical direction here), each of which is located at an orthogonal step transition in gray level (as shown by the horizontal profile in the figure).In practice, optics, sampling, and other image acquisition imperfections yield edges that are blurred, with the degree of blurring being determined by factors such as the quality of the image acquisition system, the

$$\nabla f \approx |G_x| + |G_y|.$$

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

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

sampling rate, and illumination conditions under which the image is acquired. As a result, edges are more closely modeled as having a ”ramp like” profile, such as the one shown in Fig. The slope

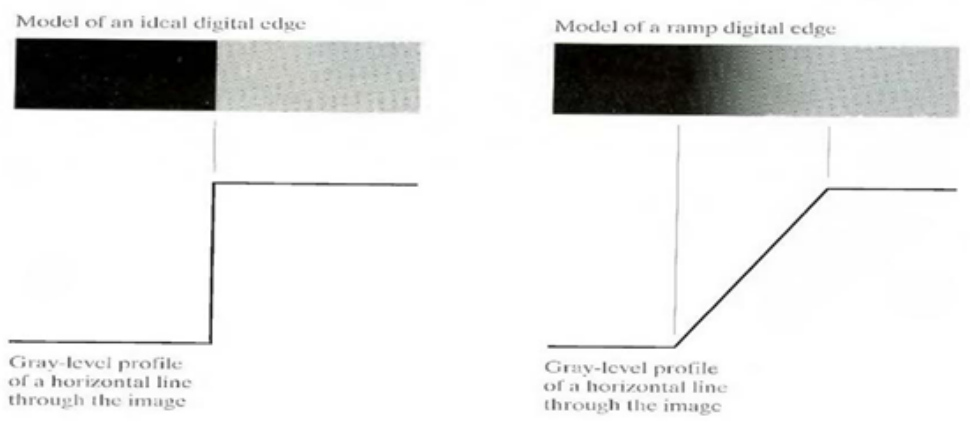


FIGURE 4.1: (a)Model of an ideal digital edge (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

of the ramp is inversely proportional to the degree of blurring in the edge. In this model, we no longer have a thin (one pixel thick) path. Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected. The ”thickness” of the edge is determined by the length of the ramp, as it transitions from an initial to a final gray level. This length is determined by the slope, which, in turn, is determined by the degree of blurring. This makes sense: Blurred edges lend to be thick and sharp edges tend to be thin. Figure (a) shows the image from which the close-up in Fig. (b) was extracted. Figure (b) shows a horizontal gray-level profile of the edge between the two regions. This figure also shows the first and second derivatives of the gray-level profile. The first derivative is positive at the points of transition into and out of the ramp as we move from left to right along the profile; it is constant for points in the ramp; and is zero in areas of constant gray level. The second derivative is positive at the transition associated with the dark side of the edge, negative at the transition associated with the light side of

the edge, and zero along the ramp and in areas of constant gray level. The signs of the derivatives in Fig. (b) Would be reversed for an edge that transitions from light to dark. We conclude from these observations that the magnitude of the first derivative can be used to detect the presence of an edge at a point in an image (i.e. to determine if a point is on a ramp). Similarly, the sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge. We note two additional properties of the second derivative around an edge: A) It produces two values for every edge in an image (an undesirable feature); and B) an imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. This zero-crossing property of the second derivative is quite useful for locating the centers of thick edges.

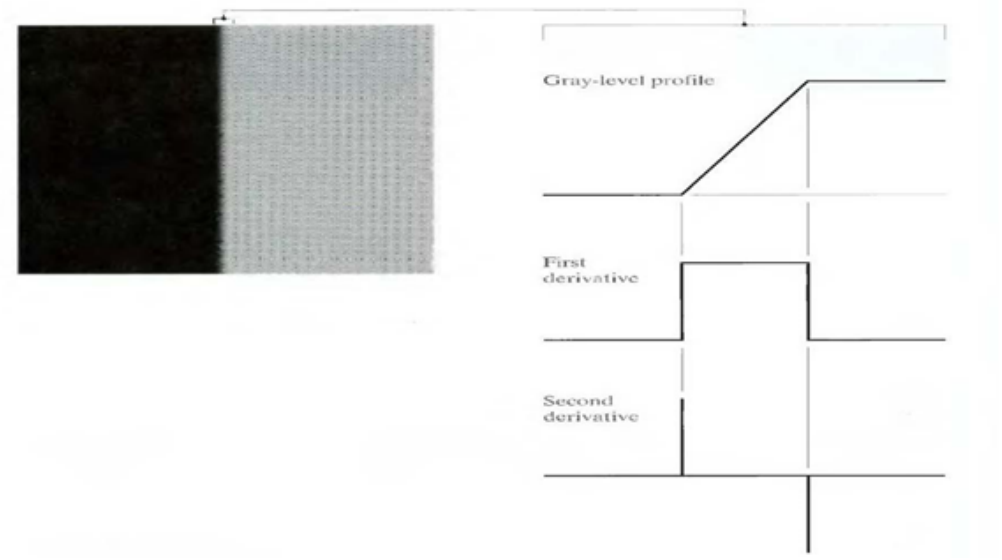


FIGURE 4.2: Figure (a) Two regions separated by a vertical edge (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.

4.1.2 basics of full-color image processing.

The different methods for edge linking are as follows

o Local processing

o Global processing via the Hough Transform

o Global processing via graph-theoretic techniques.

4.1.3 Local Processing.

One of the simplest approaches for linking edge points is to analyze the characteristics of pixels in a small neighborhood (say, 3 X 3 or 5 X 5) about every point (x, y) in an image that has been labeled an edge point. All points that are similar according to a set of predefined criteria are linked, forming an edge of pixels that share those criteria. The two principal properties used for establishing similarity of edge pixels in this kind of analysis are (1) the strength of the response of the gradient operator used to produce the edge pixel; and (2) the direction of the gradient vector. The first property is given by the value of Thus an edge pixel with coordinates (xo, yo) in a predefined neighborhood of (x, y), is similar in magnitude to the pixel at (x, y) if The direction (angle) of

$$\left|\nabla f(x, y) - \nabla f(x_0, y_0)\right| \leq E$$

the gradient vector is given by Eq. An edge pixel at (xo, yo) in the predefined neighborhood of (x, y) has an angle similar to the pixel at (x, y) if where A is a nonnegative angle threshold. The

$$\left|\alpha(x, y) - \alpha(x_0, y_0)\right| < A$$

direction of the edge at (x, y) is perpendicular to the direction of the gradient vector at that point. A point in the predefined neighborhood of (x, y) is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied. This process is repeated at every location in the image. A record must be kept of linked points as the center of the neighborhood is moved from pixel to pixel. A simple bookkeeping procedure is to assign a different gray level to each set of linked edge pixels.

4.1.4 smoothing and sharpening, color segmentation:

In this process, points are linked by determining first if they lie on a curve of specified shape. We now consider global relationships between pixels. Given n points in an image, suppose that we want to find subsets of these points that lie on straight lines. One possible solution is to first find all lines determined by every pair of points and then find all subsets of points that are close to particular lines. The problem with this procedure is that it involves finding n(n - 1)/2 n2 lines and then performing (n)(n(n - l))/2 n3 comparisons of every point to all lines. This approach is computationally prohibitive in all but the most trivial applications. Hough [1962] proposed an alternative approach, commonly referred to as the Hough transform. Consider a point (xi, yi) and the general equation of a straight line in slope-intercept form, yi = a.xi + b. Infinitely many lines pass through (xi, yi) but they all satisfy the equation yi = a.xi + b for varying values of a and b. However, writing this equation as b = -a.xi + yi, and considering the ab- plane (also called

parameter space) yields the equation of a single line for a fixed pair (xi, yi). Furthermore, a second point (xj, yj) also has a line in parameter space associated with it, and this line intersects the line associated with (xi, yi) at (a', b'), where a' is the slope and b' the intercept of the line containing both (xi, yi) and (xj, yj) in the xy-plane. In fact, all points contained on this line have lines in parameter space that intersect at (a', b'). The computational attractiveness of the Hough transform

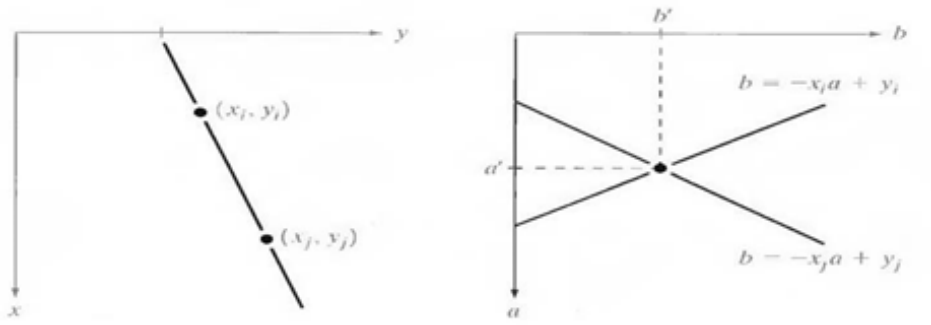


FIGURE 4.3: Figure:(a) xy-plane (b) Parameter space

arises from subdividing the parameter space into so-called accumulator cells, as illustrated where (amax , amin) and (bmax , bmin), are the expected ranges of slope and intercept values. The cell at coordinates (i, j), with accumulator value A(i, j), corresponds to the square associated with parameter space coordinates (ai , bi). Initially, these cells are set to zero. Then, for every point (xk, yk) in the image plane, we let the parameter a equal each of the allowed subdivision values on the fl-axis and solve for the corresponding b using the equation b = - xk a + yk .The resulting b's are then rounded off to the nearest allowed value in the b-axis. If a choice of ap results in solution bq, we let A (p, q) = A (p,q) + 1. At the end of this procedure, a value of Q in A (i, j) corresponds to Q points in the xy- plane lying on the line y = ai x + bj.

The number of subdivisions in the ab-plane determines the accuracy of the co linearity of these points. Note that subdividing the a axis into K increments gives, for every point (xk, yk), K values of b corresponding to the K possible values of a. With n image points, this method involves nK computations. Thus the procedure just discussed is linear in n, and the product nK does not approach the number of computations discussed at the beginning unless K approaches or exceeds n. A problem with using the equation y = ax + b to represent a line is that the slope approaches infinity as the line approaches the vertical. where H is the highest gray-level value in the image (7 in this case), and f(p) and f(q) are the gray-level values of p and q, respectively. By convention, the point p is on the right-hand side of the direction of travel along edge elements. For example, the edge segment (1, 2) (2, 2) is between points (1, 2) and (2, 2) in Fig. 4.3.5 (b). If the direction of travel is to the right, then p is the point with coordinates (2,2) and q is point with coordinates (1, 2); therefore, c (p, q) = 7 - [7- 6] = 6. This cost is shown in the box below the edge segment. If, on the other hand, we are traveling to the left between the same two points, then p is point (1, 2) and q is (2, 2). In this case the cost is 8, as shown above the edge segment in Fig (b). To simplify the discussion, we assume that edges start in the top row and terminate in the last row, so that the

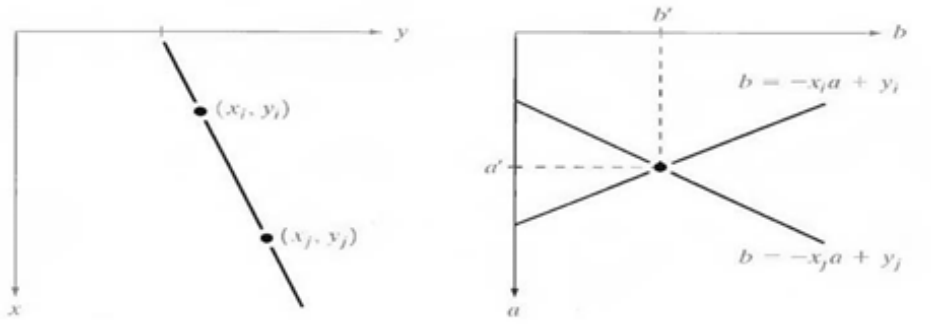


FIGURE 4.4: Figure:(a) xy-plane (b) Parameter space

first element of an edge can be only between points (1, 1), (1, 2) or (1, 2), (1, 3). Similarly, the last edge element has to be between points (3, 1), (3, 2) or (3, 2), (3, 3). Keep in mind that p and q are 4-neighbors, as noted earlier. Figure 3.6 shows the graph for this problem. Each node (rectangle) in the graph corresponds to an edge element from Fig.. An arc exists between two nodes if the two corresponding edge elements taken in succession can be part of an edge.

4.1.5 Noise in color images

In this process we have a global approach for edge detection and linking based on representing edge segments in the form of a graph and searching the graph for low-cost paths that correspond to significant edges. This representation provides a rugged approach that performs well in the presence of noise. We begin the development with some basic definitions. A graph G = (N,U) is

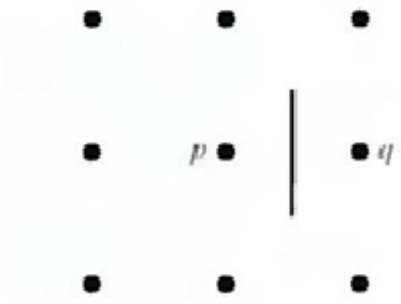
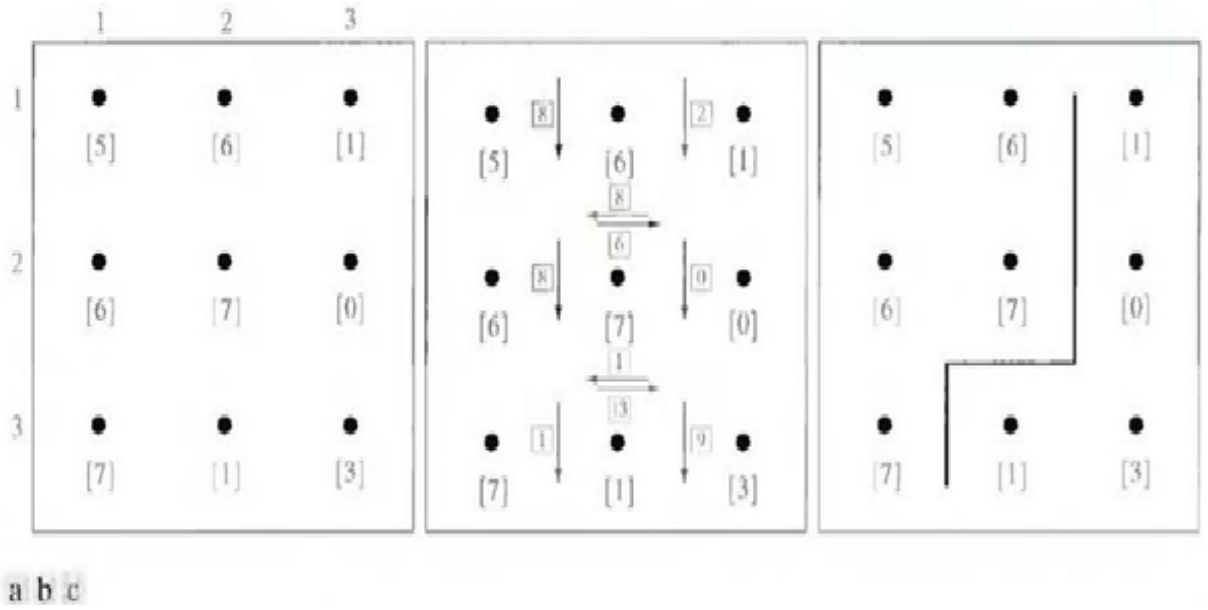


FIGURE 4.5: Edge clement between pixels p and q

a finite, nonempty set of nodes N, together with a set U of unordered pairs of distinct elements of N. Each pair (ni, nj) of U is called an arc. A graph in which the arcs are directed is called a directed graph. If an arc is directed from node ni to node nj, then nj is said to be a successor of the parent node ni. The process of identifying the successors of a node is called expansion of the node. In each graph we define levels, such that level 0 consists of a single node, called the start or root node, and the nodes in the last level are called goal nodes. A cost c (ni, nj) can be associated

with every arc (ni, nj). A sequence of nodes n1, n2... nk, with each node ni being a successor of node ni-1 is called a path from n1 to nk. The cost of the entire path is The following discussion



is simplified if we define an edge element as the boundary between two pixels p and q, such that p and q are 4-neighbors, as Fig.3.4 illustrates. Edge elements are identified by the xy-coordinates of points p and q. In other words, the edge element in Fig. 3.4 is defined by the pairs (xp, yp) (xq, yq). Consistent with the definition an edge is a sequence of connected edge elements. We can illustrate how the concepts just discussed apply to edge detection using the 3 X 3 image shown in Fig. (a). The outer numbers are pixel coordinates and the numbers in brackets represent gray-level

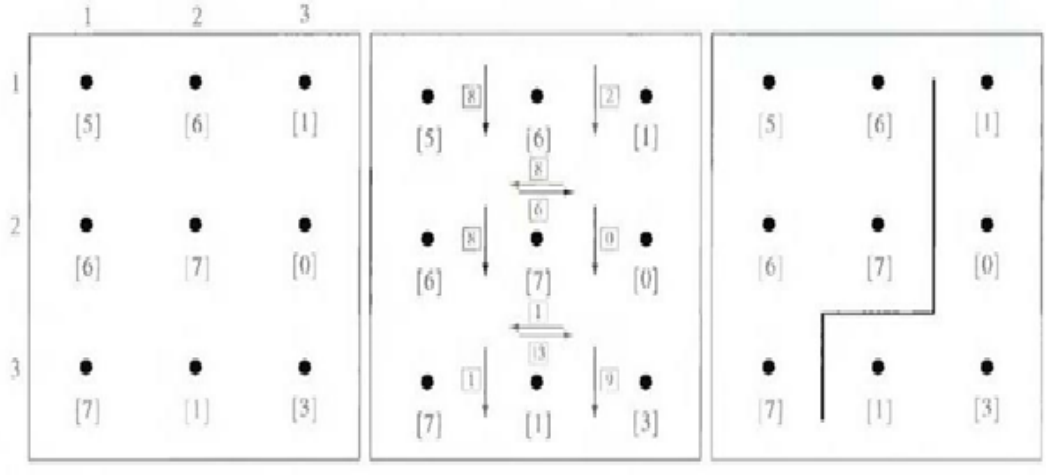


FIGURE 4.6: Figure (a) A 3 X 3 image region, (b) Edge segments and their costs, (c) Edge corresponding to the lowest-cost path in the graph shown in Fig.

values. Each edge element, defined by pixels p and q, has an associated cost, defined as As in

$$c(p,q) = H - [f(p) - f(q)]$$

Fig. (b), the cost of each edge segment, is shown in a box on the side of the arc leading into the corresponding node. Goal nodes are shown shaded. The minimum cost path is shown dashed, and the edge corresponding to this path is shown in Fig. (c).

4.1.6 Image pyramids, sub band coding:

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation. Global Thresholding: The simplest of all thresholding techniques is to partition the image histogram by using a single global threshold, T. Segmentation is then accomplished by scanning the image pixel by pixel and labeling each pixel as object or back-ground, depending on whether the gray level of that pixel is greater or less than the value of T. As indicated earlier, the success of this method depends entirely on how well the histogram can be partitioned. Compute a new threshold value: Repeat steps 2 through 4 until the

$$T = \frac{1}{2}(\mu_1 + \mu_2).$$

difference in T in successive iterations is smaller than a predefined parameter To. When there is reason to believe that the background and object occupy comparable areas in the image, a good initial value for T is the average gray level of the image. When objects are small compared to the area occupied by the background (or vice versa), then one group of pixels will dominate the histogram and the average gray level is not as good an initial choice. A more appropriate initial value for T in cases such as this is a value midway between the maximum and minimum gray levels. The parameter To is used to stop the algorithm after changes become small in terms of this parameter. This is used when speed of iteration is an important issue.

4.1.7 The haar transform.

4.1.8 wavelet transforms in one dimension:

Imaging factors such as uneven illumination can transform a perfectly segmentable histogram into a histogram that cannot be partitioned effectively by a single global threshold. An approach for

handling such a situation is to divide the original image into sub images and then utilize a Different threshold to segment each sub image. The key issues in this approach are how to subdivide the image and how to estimate the threshold for each resulting sub image. Since the threshold used for each pixel depends on the location of the pixel in terms of the sub images, this type of thresholding is adaptive. We illustrate adaptive thresholding with a example. Figure (a) shows the

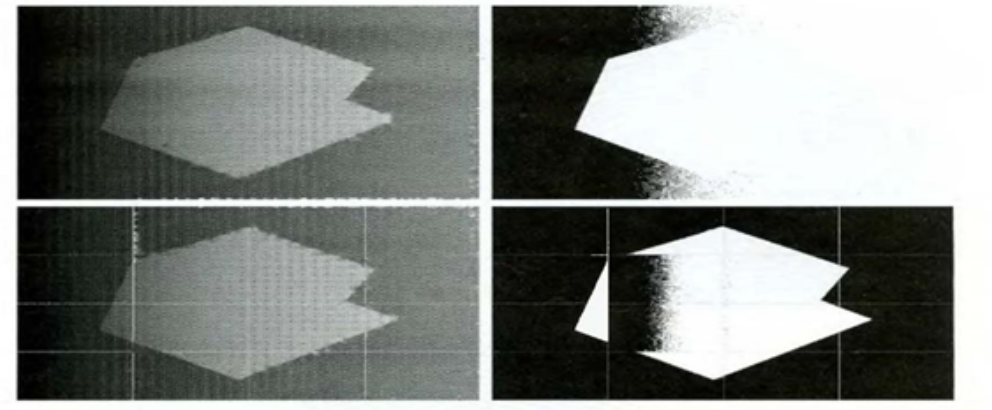


FIGURE 4.7: Figure : (a) Original image, (b) Result of global thresholding. (c) Image subdivided into individual sub images (d) Result of adaptive thresholding.

image, which we concluded could not be thresholded effectively with a single global threshold. In fact, Fig. shows the result of thresholding the image with a global threshold manually placed in the valley of its histogram. One approach to reduce the effect of non-uniform illumination is to subdivide the image into smaller sub-images, such that the illumination of each sub image is approximately uniform. Figure (c) shows such a partition, obtained by subdividing the image into four equal parts, and then subdividing each part by four again. All the sub-images that did not contain a boundary between object and back-ground had variances of less than 75. All sub-images containing boundaries had variances in excess of 100. Each sub-image with variance greater than 100 was segmented with a threshold computed for that sub-image using the algorithm. The initial value for T in each case was selected as the point midway between the minimum and maximum gray levels in the sub-image. All sub-images with variance less than 100 were treated as one composite image, which was segmented using a single threshold estimated using the same algorithm. The result of segmentation using this procedure is shown in Fig. (d). With the exception of two sub-images, the improvement over Fig. (b) is evident. The boundary between object and background in each of the improperly segmented sub-images was small and dark, and the resulting histogram was almost unimodal.

4.1.9 wavelet transforms in two dimensions. Boundary Characteristics for Histogram Improvement and Local Thresholding:

It is intuitively evident that the chances of selecting a ”good” threshold are enhanced considerably if the histogram peaks are tall, narrow, symmetric, and separated by deep valleys. One approach

for improving the shape of histograms is to consider only those pixels that lie on or near the edges between objects and the background. An immediate and obvious improvement is that histograms would be less dependent on the relative sizes of objects and the background. For instance, the histogram of an image composed of a small object on a large background area (or vice versa) would be dominated by a large peak because of the high concentration of one type of pixels. If only the pixels on or near the edge between object and the background were used, the resulting histogram would have peaks of approximately the same height. In addition, the probability that any of those given pixels lies on an object would be approximately equal to the probability that it lies on the back-ground, thus improving the symmetry of the histogram peaks. Finally, as indicated in the following paragraph, using pixels that satisfy some simple measures based on gradient and Laplacian operators has a tendency to deepen the valley between histogram peaks. The principal problem with the approach just discussed is the implicit assumption that the edges between objects and background arc known. This information clearly is not available during segmentation, as finding a division between objects and background is precisely what segmentation is all about. However, an indication of whether a pixel is on an edge may be obtained by computing its gradient. In addition, use of the Laplacian can yield information regarding whether a given pixel lies on the dark or light side of an edge. The average value of the Laplacian is 0 at the transition of an edge, so in practice the valleys of histograms formed from the pixels selected by a gradient/Laplacian criterion can be expected to be sparsely populated. This property produces the highly desirable deep valleys. The gradient Af at any point (x, y) in an image can be found. Similarly, the Laplacian A2f can also be found. These two quantities may be used to form a three-level image, as follows: where the symbols 0, +, and - represent any three distinct gray levels, T is a threshold, and the

$$s(x,y) = \begin{cases} 0 & \text{if } \nabla f < T \\ + & \text{if } \nabla f \geq T \quad \text{and} \quad \nabla^2 f \geq 0 \\ - & \text{if } \nabla f \geq T \quad \text{and} \quad \nabla^2 f < 0 \end{cases}$$

gradient and Laplacian are computed at every point (x, y). For a dark object on a light background, the use of the Eqn. produces an image s(x, y) in which (1) all pixels that are not on an edge (as determined by Af being less than T) are labeled 0; (2) all pixels on the dark side of an edge are labeled +; and (3) all pixels on the light side of an edge are labeled -. The symbols + and - in Eq. above are reversed for a light object on a dark background. Figure 6.1 shows the labeling produced by Eq. for an image of a dark, underlined stroke written on a light background. The information obtained with this procedure can be used to generate a segmented, binary image in which l's correspond to objects of interest and 0's correspond to the background. The transition (along a horizontal or vertical scan line) from a light background to a dark object must be characterized by the occurrence of a - followed by a + in s (x, y). The interior of the object is composed of pixels that are labeled either 0 or +. Finally, the transition from the object back to the background

is characterized by the occurrence of a + followed by a -. Thus a horizontal or vertical scan line containing a section of an object has the following structure:

$$(\cdots)(-,+)(0 \text{ or } +)(+,\ )(\cdots)$$

where $(\ldots)$ represents any combination of +, -, and 0. The innermost parentheses contain object points and are labeled 1. All other pixels along the same scan line are labeled 0, with the exception of any other sequence of (- or +) bounded by (-, +) and (+, -). Figure (a) shows an image of

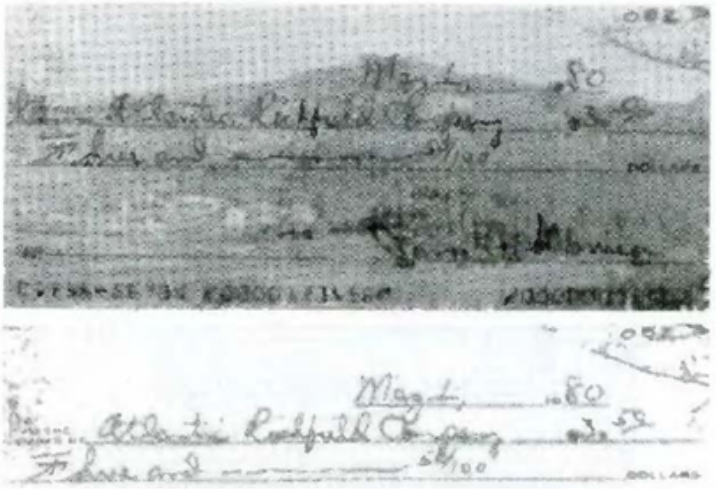


FIGURE 4.8: Figure : (a) Original image, (b) Image segmented by local thresholding.

an ordinary scenic bank check. Figure show the histogram as a function of gradient values for pixels with gradients greater than 5. Note that this histogram has two dominant modes that are symmetric, nearly of the same height, and arc separated by a distinct valley. Finally, Fig. (b) shows the segmented image obtained by with T at or near the midpoint of the valley. Note that this example is an illustration of local thresholding, because the value of T was determined from a histogram of the gradient and Laplacian, which are local properties.

4.1.10 Region based segmentation.

The objective of segmentation is to partition an image into regions. We approached this problem by finding boundaries between regions based on discontinuities in gray levels, whereas segmentation was accomplished via thresholds based on the distribution of pixel properties, such as gray-level values or color.

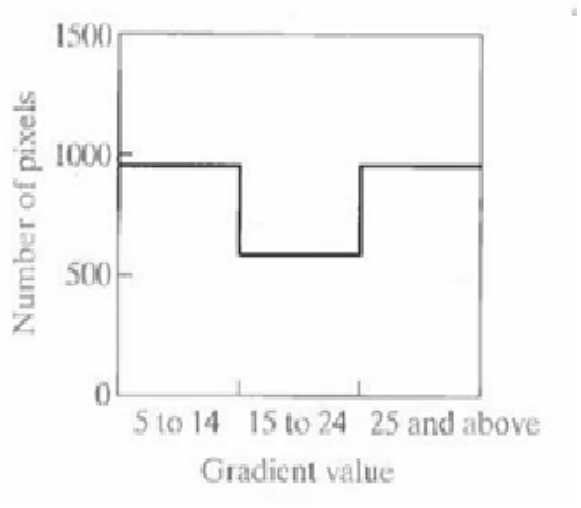
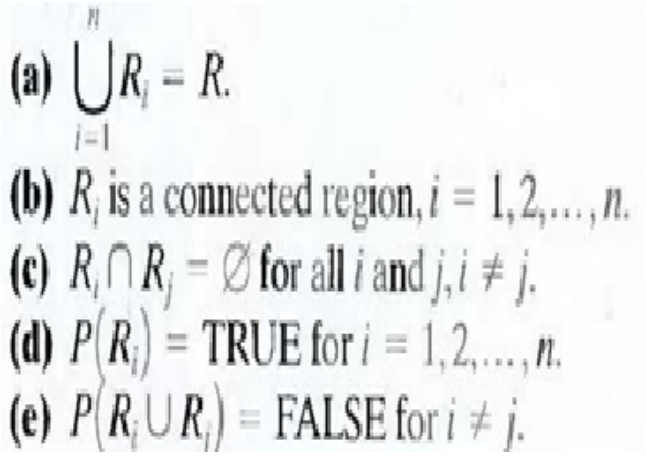


FIGURE 4.9: Figure: Histogram of pixels with gradients greater than 5

4.1.11 Basic Formulation.

Let R represent the entire image region. We may view segmentation as a process that partitions R into n subregions, R1, R2..., Rn, such that Here, P (Ri) is a logical predicate defined over the points



in set Ri and is the null set. Condition (a) indicates that the segmentation must be complete; that is, every pixel must be in a region. Condition (b) requires that points in a region must be connected in some predefined sense. Condition (c) indicates that the regions must be disjoint. Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region—for example P (Ri) =TRUE if all pixels in Ri, have the same gray level. Finally, condition (c) indicates that regions Ri and Rj are different in the sense of predicate P.

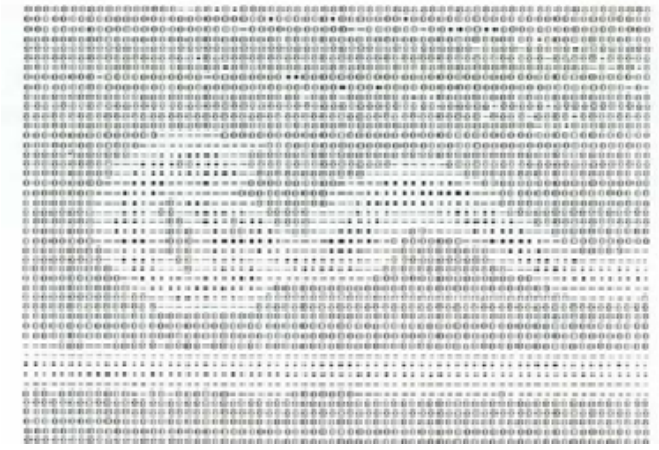


FIGURE 4.10: Signature

4.1.12 Region Growing:

As its name implies, region growing is a procedure that groups pixels or sub regions into larger regions based on predefined criteria. The basic approach is to start with a set of ”seed” points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (such as specific ranges of gray level or color). When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds. The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of color. This problem would be significantly more difficult, or even impossible, to handle without the inherent information available in color images. When the images are monochrome, region analysis must be carried out with a set of descriptors based on gray levels and spatial properties (such as moments or texture). Basically, growing a region should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as gray level, texture, and color, are local in nature and do not take into account the ”history” of region growth. Additional criteria that increase the power of a region- growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the gray level of a candidate and the average gray level of the grown region), and the shape of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available. Figure (a) shows an X-ray image of a weld (the horizontal dark region) containing several cracks and porosities (the bright, white streaks running horizontally through the middle of the image). We wish to use region growing to segment the regions of the weld failures. These segmented features could be used for inspection, for inclusion in a database of historical studies, for controlling an automated welding system, and for other numerous applications. The first order of business is to determine

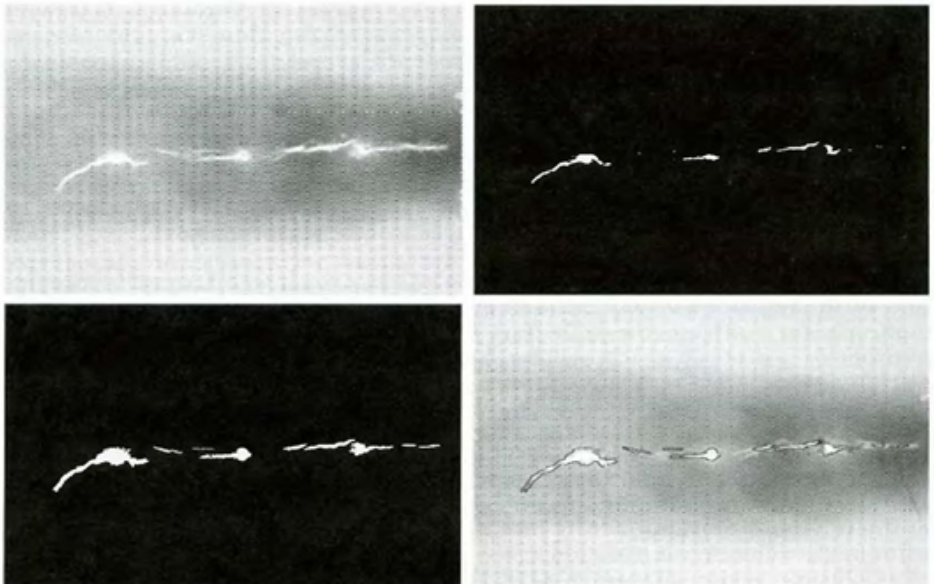


FIGURE 4.11: Figure: (a) Image showing defective welds, (b) Seed points, (c) Result of region growing, (c) Boundaries of segmented ; defective welds (in black).

the initial seed points. In this application, it is known that pixels of defective welds tend to have the maximum allowable digital value B55 in this case). Based on this information, we selected as starting points all pixels having values of 255. The points thus extracted from the original image are shown in Fig. Note that many of the points are clustered into seed regions. The next step is to choose criteria for region growing. In this particular example we chose two criteria for a pixel to be annexed to a region: (1) The absolute gray-level difference between any pixel and the seed had to be less than 65. This number is based on the histogram shown in Fig. and represents the difference between 255 and the location of the first major valley to the left, which is representative of the highest gray level value in the dark weld region. (2) To be included in one of the regions, the pixel had to be 8-connected to at least one pixel in that region. If a pixel was found to be connected to more than one region, the regions were merged. Figure (c) shows the regions that resulted by starting with the seeds in Fig. (b) and utilizing the criteria defined in the previous paragraph. Superimposing the boundaries of these regions on the original image [Fig. (d)] reveals that the region-growing procedure did indeed segment the defective welds with an acceptable degree of accuracy. It is of interest to note that it was not necessary to specify any stopping rules in this case because the criteria for region growing were sufficient to isolate the features of interest.

4.1.13 Region Splitting and Merging:

The procedure just discussed grows regions from a set of seed points. An alternative is to subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions. A split and merge algorithm that iteratively works toward satisfying these constraints is developed. Let R represent the entire image region and select a

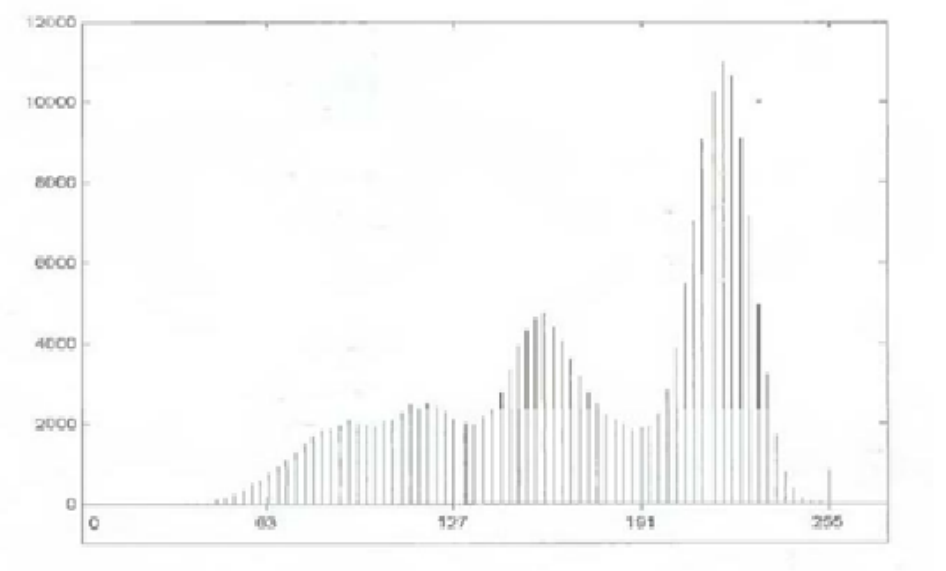


FIGURE 4.12: Region splitting and merging

predicate P. One approach for segmenting R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region Ri, P(Ri) = TRUE. We start with the entire region. If P(R) = FALSE, we divide the image into quadrants. If P is FALSE for any quadrant, we subdivide that quadrant into subquadrants, and so on. This particular splitting technique has a convenient representation in the form of a so-called quadtree (that is, a tree in which nodes have exactly four descendants), as illustrated in Fig.. Note that the root of the tree corresponds to the entire image and that each node corresponds to a subdivision. In this case, only R4 was subdivided further. If only splitting were used, the final partition likely would contain adjacent regions with

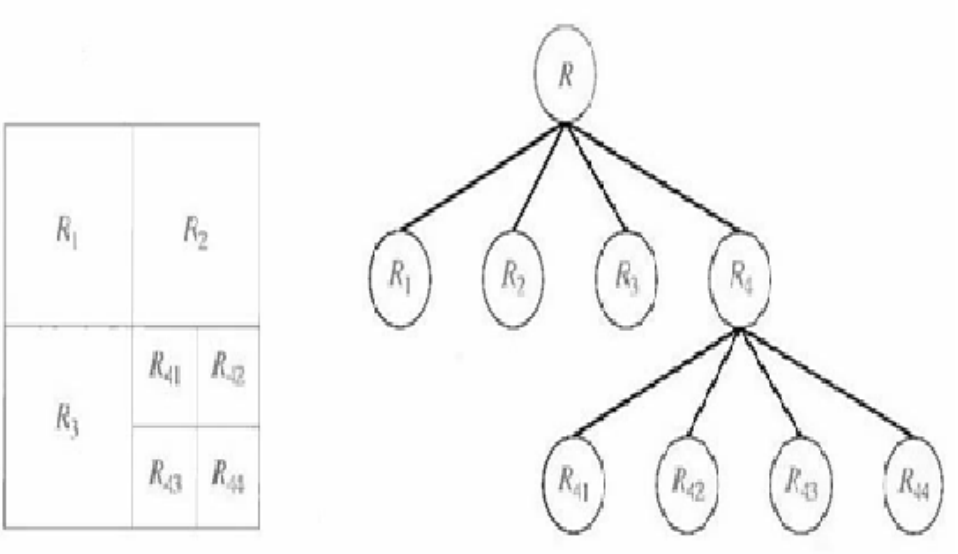


FIGURE 4.13: Figure: (a) Partitioned image (b) Corresponding quadtree.

identical properties. This drawback may be remedied by allowing merging, as well as splitting. Satisfying the constraints, requires merging only adjacent regions whose combined pixels satisfy the predicate P. That is, two adjacent regions Rj and Rk are merged only if P (Rj U Rk) = TRUE. The preceding discussion may be summarized by the following procedure, in which, at any step we

Split into four disjoint quadrants any region Ri, for which P (Ri) = FALSE.

Merge any adjacent regions Rj and Rk for which P (Rj U Rk) =TRUE. Stop when no further merging or splitting is possible. Several variations of the preceding basic theme are possible. For example, one possibility is to split the image initially into a set of blocks. Further splitting is carried out as described previously, but merging is initially limited to groups of four blocks that are descendants in the quadtree representation and that satisfy the predicate P. When no further mergings of this type are possible, the procedure is terminated by one final merging of regions satisfying step 2. At this point, the merged regions may be of different sizes. The principal advantage of this approach is that it uses the same quadtree for splitting and merging, until the final merging step.

Chapter 5

SYSTEM DESIGN TECHNIQUES

Course Outcomes

After successful completion of this module, students should be able to:

CO 6	**Compare** the lossy and lossless compression models for achieving image compression.	Analyze

5.1 Introduction: Image compression and The redundancies in a digital image.

The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. A clear distinction must be made between data and information. They are not synonymous. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information. Such might be the case, for example, if a long-winded individual and someone who is short and to the point where to relate the same story. Here, the information of interest is the story; words are the data used to relate the information. If the two individuals use a different number of words to tell the same basic story, two different versions of the story are created, and at least one includes nonessential data. That is, it contains data (or words) that either provide no relevant information or simply restate that which is already known. It is thus said to contain data redundancy. Data redundancy is a central issue in digital image compression. It is not an abstract concept but a mathematically quantifiable entity. If n1 and n2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy RD of the first data set (the one characterized by n1) can be defined as where CR , commonly called the compression ratio, is In digital image compression, three basic data redundancies can be identified and exploited: coding redundancy, inter pixel redundancy, and psychovisual redundancy. Data compression is achieved

$$R_D = 1 - \frac{1}{C_R}$$

$$C_R = \frac{n_1}{n_2}.$$

when one or more of these redundancies are reduced or eliminated. Coding Redundancy: In this, we utilize formulation to show how the gray-level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it. Let us assume, once again, that a discrete random variable rk in the interval [0, 1] represents the gray levels of an image and that each rk occurs with probability pr (rk). where L is the number

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L - 1$$

of gray levels, nk is the number of times that the kth gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of rk is l (rk), then the average number of bits required to represent each pixel is That is, the average length

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k).$$

of the code words assigned to the various gray-level values is found by summing the product of the number of bits used to represent each gray level and the probability that the gray level occurs. Thus the total number of bits required to code an M X N image is MNLavg.

5.1.1 Inter pixel Redundancy:

Consider the images shown in Figs. (a) and (b). As Figs. (c) and (d) show, these images have virtually identical histograms. Note also that both histograms are trimodal, indicating the presence of three dominant ranges of gray-level values. Because the gray levels in these images are not equally probable, variable-length coding can be used to reduce the coding redundancy that would result from a straight or natural binary encoding of their pixels. The coding process, however, would not alter the level of correlation between the pixels within the images. In other words, the codes

used to represent the gray levels of each image have nothing to do with the correlation between pixels. These correlations result from the structural or geometric relationships between the objects in the image. Figures (e) and (f) show the respective autocorrelation coefficients computed along

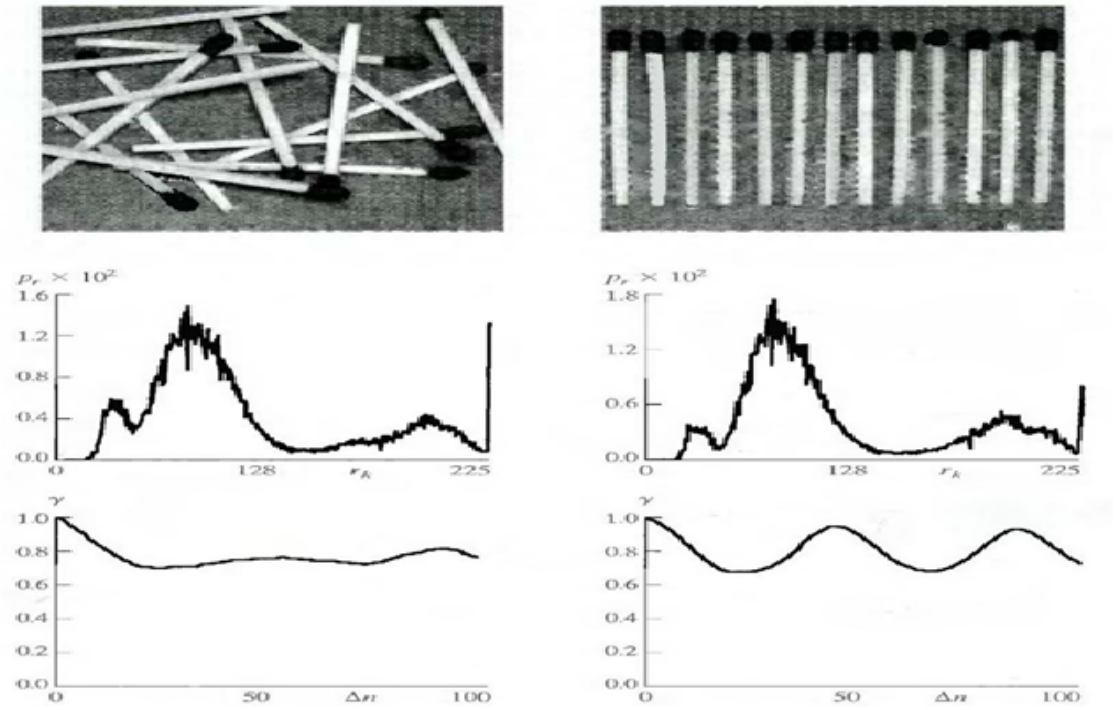


FIGURE 5.1: Figure: Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

one line of each image. The scaling factor in Eq. above accounts for the varying number of sum

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n).$$

terms that arise for each integer value of /n. Of course, /n must be strictly less than N, the number of pixels on a line. The variable x is the coordinate of the line used in the computation. Note the dramatic difference between the shape of the functions shown in Figs. (e) and (f). Their shapes can be qualitatively related to the structure in the images in Figs. (a) and (b).This relationship is particularly noticeable in Fig. (f), where the high correlation between pixels separated by 45 and 90 samples can be directly related to the spacing between the vertically oriented matches of Fig. (b). In addition, the adjacent pixels of both images are highly correlated. When n is 1, is 0.9922 and 0.9928 for the images of Figs. (a) and (b), respectively. These values are typical of most properly sampled television images. These illustrations reflect another important form of data redundancy—one directly related to the inter pixel correlations within an image. Because the value of any given pixel can be reasonably predicted from the value of its neighbors, the information

carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names, including spatial redundancy, geometric redundancy, and inter frame. redundancy, have been coined to refer to these inter pixel dependencies. We use the term inter pixel redundancy to encompass them all. In order to reduce the inter pixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "non visual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove inter pixel redundancy) are referred to as mappings. They are called reversible mappings if the original image elements can be reconstructed from the transformed data set.

5.1.2 Psychovisual Redundancy:

The brightness of a region, as perceived by the eye, depends on factors other than simply the light reflected by the region. For example, intensity variations (Mach bands) can be perceived in an area of constant intensity. Such phenomena result from the fact that the eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psycho visually redundant. It can be eliminated without significantly impairing the quality of image perception. That psycho visual redundancies exist should not come as a surprise, because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Psycho visual redundancy is fundamentally different from the redundancies discussed earlier. Unlike coding and inter pixel redundancy, psycho visual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psycho visually redundant data results in a loss of quantitative information, it is commonly referred to as quantization. This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. As it is an irreversible operation (visual information is lost), quantization results in lossy data compression.

5.1.3 Fidelity criterion.

The removal of psycho visually redundant data results in a loss of real or quantitative visual information. Because information of interest may be lost, a repeatable or reproducible means of quantifying the nature and extent of information loss is highly desirable. Two general classes of criteria are used as the basis for such an assessment:

5.1.4 Objective fidelity criteria and Subjective fidelity criteria.

When the level of information loss can be expressed as a function of the original or input image and the compressed and subsequently decompressed output image, it is said to be based on an objective fidelity criterion. A good example is the root-mean-square (rms) error between an input and output image. Let f(x, y) represent an input image and let f(x, y) denote an estimate or approximation of f(x, y) that results from compressing and subsequently decompressing the input. For any value of x and y, the error e(x, y) between f (x, y) and $\hat{f}(x,y)$ *canbedefinedas*

so that the total error between the two images is Where the images are of size M X N. y) A closely

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

$$e_{rms}=\left[\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y\ 0}^{N\ 1}[\hat{f}(x,y)-f(x,y)]^2\right]^{1/2}$$

related objective fidelity criterion is the mean-square signal-to-noise ratio of the compressed- decompressed image. If $\hat{f}(x,y)$ *isconsideredtobethesumoftheoriginalimage* $f(x,y)$ *andanoisesignale*(x,y), *themean* $-$ *squaresignal* $-$ *to* $-$ *noiseratioofthe outputimage*, *denotedSNRrms*, *isThermsvalueofthesignal* $-$ *to* $-$ *noiseratio*, *denotedSNRrms*, *isobtainedbytakingthesquarerootofEq.above.Althoughobjectivefidelitycriteria*

5.1.5 Image compression models.

a compression system consists of two distinct structural blocks: an encoder and a decoder. An input image f(x, y) is fed into the encoder, which creates a set of symbols from the input data. After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image If it is, the system is error free or information preserving; if not, some level of distortion is present in the reconstructed image. Both the encoder and decoder shown in Fig. consist of two relatively independent functions or sub blocks. The encoder is made up of a source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder's output. As would be expected, the decoder includes a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free (not prone to error), the channel encoder and decoder are omitted, and the general encoder and decoder become the source encoder and decoder, respectively.

5.1.6 The Source Encoder and Decoder:

The source encoder is responsible for reducing or eliminating any coding, inters pixel, or psycho visual redundancies in the input image. The specific application and associated fidelity requirements dictate the best encoding approach to use in any given situation. Normally, the approach can be modeled by a series of three independent operations. As Fig (a) shows, each operation is designed to reduce one of the three redundancies. Figure (b) depicts the corresponding source decoder. In the first stage of the source encoding process, the mapper transforms the input data into a (usually non visual) format designed to reduce inte rpixel redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image. Run-length coding is an example of a mapping that directly results

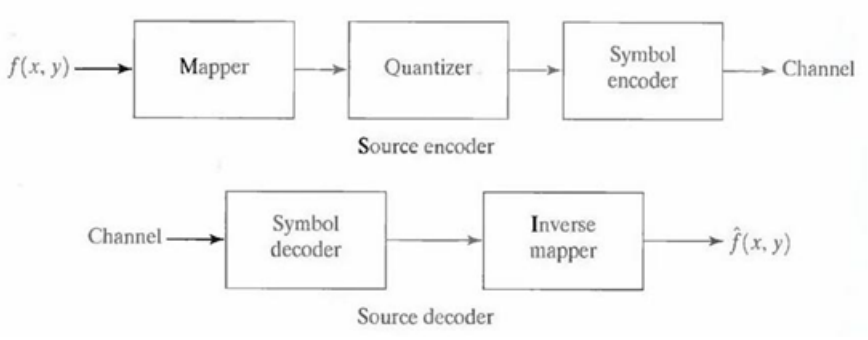


FIGURE 5.3: Figure: (a) Source encoder and (b) source decoder model

in data compression in this initial stage of the overall source encoding process. The representation of an image by a set of transform coefficients is an example of the opposite case. Here, the mapper transforms the image into an array of coefficients, making its inter pixel redundancies more accessible for compression in later stages of the encoding process. The second stage, or quantizer block in Fig. (a), reduces the accuracy of the mapper's output in accordance with some reestablished fidelity criterion. This stage reduces the psycho visual redundancies of the input image. This operation is irreversible. Thus it must be omitted when error-free compression is desired. In the third and final stage of the source encoding process, the symbol coder creates a fixed- or variable-length code to represent the quantizer output and maps the output in accordance with the code. The term symbol coder distinguishes this coding operation from the overall source encoding process. In most cases, a variable-length code is used to represent the mapped and quantized data set. It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy. The operation, of course, is reversible. Upon completion of the symbol coding step, the input image has been processed to remove each of the three redundancies. Figure (a) shows the source encoding process as three successive operations, but all three operations are not necessarily included in every compression system. Recall, for example, that the quantizer must be omitted when error-free compression is desired. In addition, some compression techniques normally are modeled by merging blocks that are physically separate in Fig (a). In the

predictive compression systems, for instance, the mapper and quantizer are often represented by a single block, which simultaneously performs both operations. The source decoder shown in Fig. (b) Contains only two components: a symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model shown in Fig. (b).

5.1.7 The Channel Encoder and Decoder:

The channel encoder and decoder play an important role in the overall encoding-decoding process when the channel of Fig. is noisy or prone to error. They are designed to reduce the impact of channel noise by inserting a controlled form of redundancy into the source encoded data. As the output of the source encoder contains little redundancy, it would be highly sensitive to transmission noise without the addition of this "controlled redundancy." One of the most useful channel encoding techniques was devised by R. W. Hamming (Hamming [1950]). It is based on appending enough bits to the data being encoded to ensure that some minimum number of bits must change between valid code words. Hamming showed, for example, that if 3 bits of redundancy are added to a 4-bit word, so that the distance between any two valid code words is 3, all single-bit errors can be detected and corrected. (By appending additional bits of redundancy, multiple-bit errors can be detected and corrected.) The 7-bit Hamming (7, 4) code word h1, h2, h3. . . ., h6, h7 associated with a 4-bit binary number b3b2b1b0 is Where denotes the exclusive OR operation. Note that bits

$$
\begin{aligned}
h_1 &= b_3 \oplus b_2 \oplus b_0 \qquad & h_3 &= b_3 \\
h_2 &= b_3 \oplus b_1 \oplus b_0 \qquad & h_5 &= b_2 \\
h_4 &= b_2 \oplus b_1 \oplus b_0 \qquad & h_6 &= b_1 \\
& & h_7 &= b_0
\end{aligned}
$$

h1, h2, and h4 are even- parity bits for the bit fields b3 b2 b0, b3b1b0, and b2b1b0, respectively. (Recall that a string of binary bits has even parity if the number of bits with a value of 1 is even.) To decode a Hamming encoded result, the channel decoder must check the encoded value for odd parity over the bit fields in which even parity was previously established. A single-bit error is indicated by a nonzero parity word c4c2c1, where If a nonzero value is found, the decoder simply

$$
\begin{aligned}
c_1 &= h_1 \oplus h_3 \oplus h_5 \oplus h_7 \\
c_2 &= h_2 \oplus h_3 \oplus h_6 \oplus h_7 \\
c_4 &= h_4 \oplus h_5 \oplus h_6 \oplus h_7.
\end{aligned}
$$

complements the code word bit position indicated by the parity word. The decoded binary value is

then extracted from the corrected code word as h3h5h6h7. Method of generating variable length codes with an example.

5.1.8 Variable-Length Coding:

The simplest approach to error-free image compression is to reduce only coding redundancy. Coding redundancy normally is present in any natural binary encoding of the gray levels in an image. It can be eliminated by coding the gray levels. To do so requires construction of a variable-length code that assigns the shortest possible code words to the most probable gray levels. Here, we examine several optimal and near optimal techniques for constructing such a code. These techniques are formulated in the language of information theory. In practice, the source symbols may be either the gray levels of an image or the output of a gray-level mapping operation (pixel differences, run lengths, and so on).

5.1.9 Huffman coding:

The most popular technique for removing coding redundancy is due to Huffman (Huffman [1952]). When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of the noiseless coding theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time.The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. Figure illustrates this process for binary coding (K-ary Huffman codes can also be constructed). At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values. To form the first source reduction, the bottom two probabilities,0.06 and 0.04, are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This process is then repeated until a reduced source with two symbols (at the far right) is reached.

The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As Fig. 4.2 shows, these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrarily Appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final

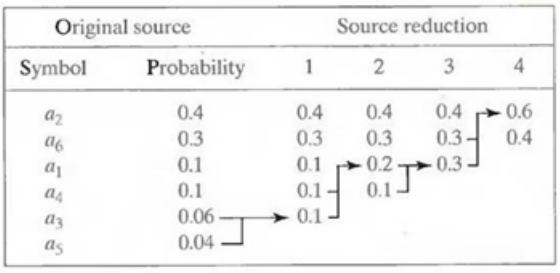


FIGURE 5.4: Figure: Huffman source reductions.

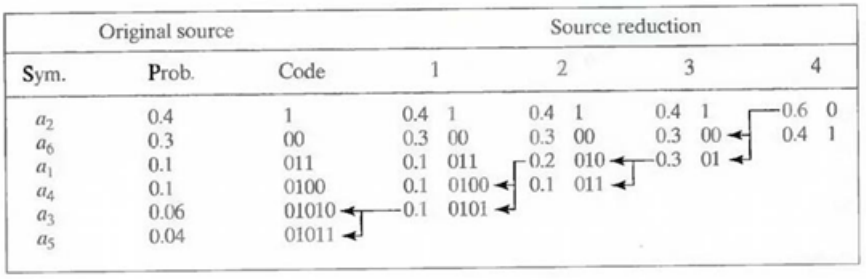


FIGURE 5.5: Figure: Huffman code assignment procedure.

code appears at the far left in Fig. The average length of this code is and the entropy of the source

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$
$$= 2.2 \text{ bits/symbol}$$

is 2.14 bits/symbol. The resulting Huffman code efficiency is 0.973. Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time. After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. For the binary code of Fig., a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a3 .The next valid code is 011, which corresponds to symbol a1. Continuing in this manner reveals the completely decoded message to be a3a1a2a2a6.

5.1.10 Arithmetic encoding process with an example. Arithmetic coding:

Unlike the variable-length codes described previously, arithmetic coding generates non block codes. In arithmetic coding, which can be traced to the work of Elias, a one-to-one correspondence between source symbols and code words does not exist. Instead, an entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units (say, bits) required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. Because the technique does not require, as does Huffman's approach, that each source symbol translate into an integral number of code symbols (that is, that the symbols be coded one at a time), it achieves (but only in theory) the bound established by the noiseless coding theorem. Figure illustrates the basic arithmetic cod-

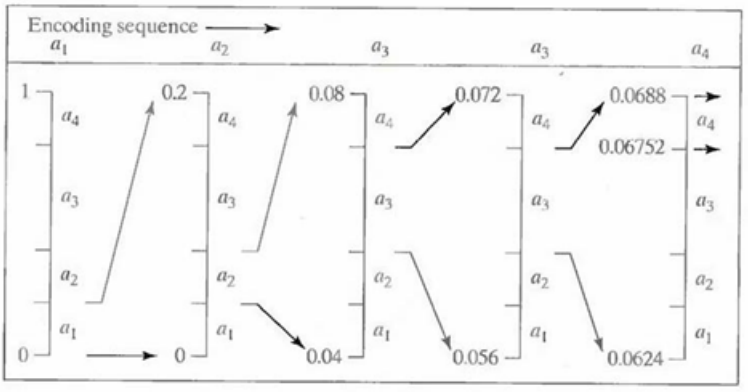


FIGURE 5.6: Figure: Arithmetic coding procedure

ing process. Here, a five-symbol sequence or message, a1a2a3a3a4, from a four-symbol source is coded. At the start of the coding process, the message is assumed to occupy the entire half-open interval [0, 1). As Table shows, this interval is initially subdivided into four regions based on the probabilities of each source symbol. Symbol ax, for example, is associated with subinterval [0, 0.2). Because it is the first symbol of the message being coded, the message interval is initially narrowed to [0, 0.2). Thus in Fig. [0, 0.2) is expanded to the full height of the figure and its end points labeled by the values of the narrowed range. The narrowed range is then subdivided in accordance with the original source symbol probabilities and the process continues with the next message symbol.

5.1.11 LZW coding with an example. LZW Coding:

The technique, called Lempel-Ziv-Welch (LZW) coding, assigns fixed-length code words to variable length sequences of source symbols but requires no a priori knowledge of the probability of

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

FIGURE 5.7: Table Arithmetic coding example

occurrence of the symbols to be encoded. LZW compression has been integrated into a variety of mainstream imaging file formats, including the graphic interchange format (GIF), tagged image file format (TIFF), and the portable document format (PDF). LZW coding is conceptually very simple (Welch [1984]). At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit Monochrome images, the first 256 words of the dictionary are assigned to the gray values 0, 1, 2..., and 255. As the encoder sequentially examines the image's pixels, gray- level sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations. If the first two pixels of the image are white, for instance, sequence —255- 255‖ might be assigned to location 256, the address following the locations reserved for gray levels 0 through 255. The next time that two consecutive white pixels are encountered, code word 256, the address of the location containing sequence 255-255, is used to represent them. If a 9-bit, 512-word dictionary is employed in the coding process, the original (8 + 8) bits that were used to represent the two pixels are replaced by a single 9-bit code word. Cleary, the size of the dictionary is an important system parameter. If it is too small, the detection of matching gray- level sequences will be less likely; if it is too large, the size of the code words will adversely affect compression performance. Consider the following 4 x 4, 8-bit image of a vertical edge:

5.1.12 Concept of bit plane coding method. Bit-Plane Coding:

An effective technique for reducing an image's inter pixel redundancies is to process the image's bit planes individually. The technique, called bit-plane coding, is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images and compressing each binary image via one of several well-known binary compression methods.

5.1.13 Bit-plane decomposition:

The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit bit planes. The zeroth-

order bit plane is generated by collecting the a0 bits of each pixel, while the (m - 1) st-order bit plane contains the am-1, bits or coefficients. In general, each bit plane is numbered from 0 to m-1 and is constructed by setting its pixels equal to the values of the appropriate bits or polynomial coefficients from each pixel in the original image. The inherent disadvantage of this approach is that small changes in gray level can have a significant impact on the complexity of the bit planes. If a pixel of intensity 127 (01111111) is adjacent to a pixel of intensity 128 (10000000), for instance, every bit plane will contain a corresponding 0 to 1 (or 1 to 0) transition. For example, as the most significant bits of the two binary codes for 127 and 128 are different, bit plane 7 will contain a zero-valued pixel next to a pixel of value 1, creating a 0 to 1 (or 1 to 0) transition at that point. An alternative decomposition approach (which reduces the effect of small gray-level variations) is to first represent the image by an m-bit Gray code. The m-bit Gray code gm-1... g2g1g0 that corresponds to the polynomial in Eq. above can be computed from

Here, denotes the exclusive OR operation. This code has the unique property that successive code words differ in only one bit position. Thus, small changes in gray level are less likely to affect all m bit planes. For instance, when gray levels 127 and 128 are adjacent, only the 7th bit plane will contain a 0 to 1 transition, because the Gray codes that correspond to 127 and 128 are 11000000 and 01000000, respectively.

5.1.14 Lossy Predictive Coding:

In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig. shows, the quantizer, which absorbs the nearest integer function of the error-free encoder, is inserted between the symbol encoder and the point at which the prediction error is formed. In order to accommodate

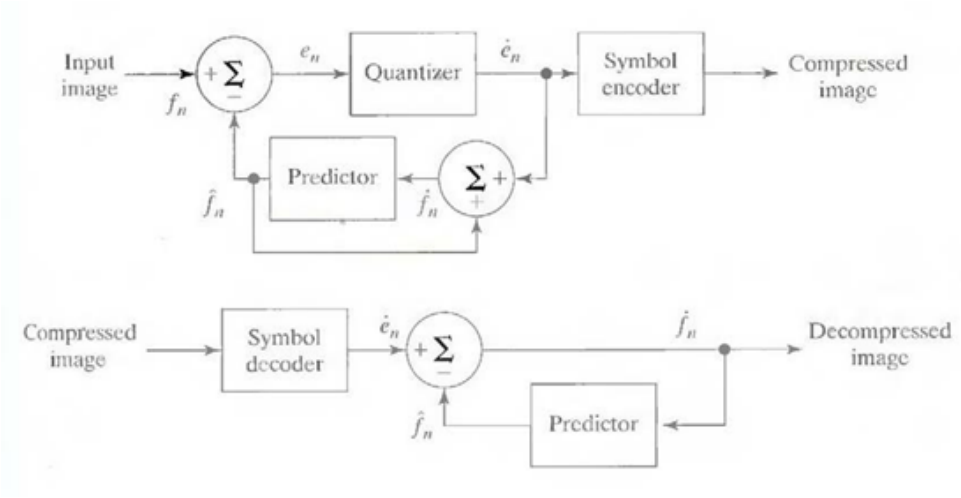


FIGURE 5.8: Figure: A lossy predictive coding model: (a) encoder and (b) decoder.

the insertion of the quantization step, the error-free encoder of figure must be altered so that the

predictions generated by the encoder and decoder are equivalent. As Fig. shows, this is accomplished by placing the lossy encoder's predictor within a feedback loop, where its input, denoted f˙n, is generated as a function of past predictions and the corresponding quantized errors.

5.1.15 Block diagram about transform coding system. Transform Coding:

All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods. In this coding, we consider compression techniques that are based on modifying the transform of an image. In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded. For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion. A variety of transformations, including the discrete Fourier transform (DFT), can be used to transform the image data. Figure shows a typical transform coding system. The decoder implements the inverse

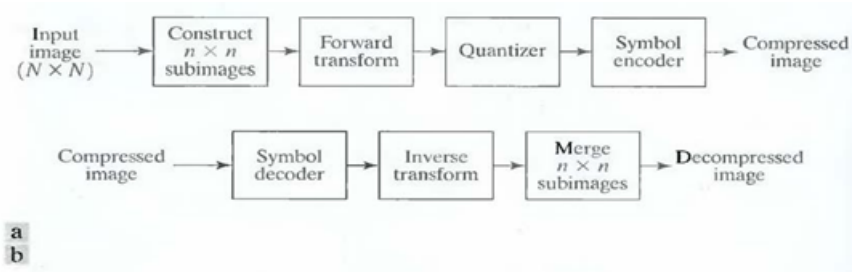


FIGURE 5.9: Figure A transform coding system: (a) encoder; (b) decoder.

sequence of steps (with the exception of the quantization function) of the encoder, which performs four relatively straightforward operations: sub image decomposition, transformation, quantization, and coding. An N X N input image first is subdivided into sub images of size n X n, which are then transformed to generate (N/n) 2 sub image transform arrays, each of size n X n. The goal of the transformation process is to decorrelate the pixels of each sub image, or to pack as much information as possible into the smallest number of transform coefficients. The quantization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information. These coefficients have the smallest impact on reconstructed sub image quality. The encoding process terminates by coding (normally using a variable-length code) the quantized coefficients. Any or all of the transform encoding steps can be adapted to local image content, called adaptive transform coding, or fixed for all sub images, called non adaptive transform coding.