IMAGE AND VIDEO PROCESSING (R20A0425)

Lecture Notes

B.TECH
(IV YEAR – I SEM)
(2024-25)

Prepared by:

Ms. D Asha, Associate Professor

Department of Electronics and Communication Engineering

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution - UGC, Govt. of India)
Recognized under 2(f) and 12 (B) of UGC ACT 1956
(Affiliated to JNTUH, Hyderabad, Approved by AICTE – Accredited by NBA & NAAC–'A' Grade- ISO 9001:2015 Certified)
Maisammaguda, Dhulapally (PostVia.Kompally), Secunderabad–500100, TelanganaState,India

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

IV Year B. Tech. ECE-I Sem **L/T/P/C**
3/-/- 3

CORE ELECTIVE – IV
(R20A0425) IMAGE AND VIDEO PROCESSING

Course Objectives:

1) To understand the basic fundamentals of digital image processing and Image Transforms.
2) To Master the Image Processing Techniques in Spatial Domain and Frequency Domain.
3) To learn the fundamentals of various Image compression models.
4) To understand the Basic Steps of Video Processing
5) To learn the Mathematical and computational skills needed to understand the principal of 2-DMotion Estimation

UNIT I

Fundamentals of Image Processing and Image Transforms:- Digital Image fundamentals, Sampling and quantization of an Image, Relationship between pixels.

Image Transforms: 2-D Discrete Fourier Transform, Properties, Discrete cosine Transform, Hadamard Transform.

UNIT II

Image Processing Techniques: Image Enhancement **:** Spatial Domain methods: Histogram Processing, Fundamentals of Spatial filtering, Smoothing spatial filters, Sharpening Spatial filters. Frequency Domain methods : Basics of filtering in frequency domain, image smoothing, image sharpening. **Image Restoration**: Degradation Model, Inverse Filtering, Least Mean Square Filters, Constrained Least Squares Restoration

UNIT III

Image Compression: Image compression fundamentals – coding Redundancy, spatial and temporal redundancy. Compression models : Lossy and Lossless, Huffmann coding, Arithmetic coding, run length coding, transform coding, predictive coding , JPEG standards

UNIT IV

BASIC STEPS OF VIDEO PROCESSING:

Analog video, Digital video, Time varying image formation model, Geometric image formation, formation, sampling of video signal.

UNIT V

2D Motion Estimation

Optical flow, Pixel based motion estimation, Region based Motion estimation, Multi resolution motionestimation, Application of motion estimation in video coding

TEXT BOOKS:

1) Gonzaleze and Woods ,"Digital Image Processing ", 3rd edition , Pearson

2) Yao wang, Joem Ostarmann and Ya – quin Zhang, "Video processing and communication ",1stedition , PHI

REFERENCE BOOKS

1) M. Tekalp ,"Digital video Processing", Prentice Hall International

Course Outcomes:

1. Understand the basic fundamentals of digital image processing and Image Transforms.
2. Master the Image Processing Techniques in Spatial Domain and Frequency Domain.
3. Learn the fundamentals of various Image compression models.
4. Understand the Basic Steps of Video Processing.
5. Learn the Mathematical and computational skills needed to understand the principle of 2- D MotionEstimation

UNIT-I

Fundamentals of Image Processing and Image Transforms

DIGITAL IMAGE FUNDAMENTALS:

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

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

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

Fig 1.1 Image acquisition and processing

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

For example: The RGB color system, a color image consists of three (red, green and blue) individual component images. For this reason, many of the techniques developed for

monochrome images can be extended to color images by processing the three component images individually.

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

APPLICATIONS OF DIGITAL IMAGE PROCESSING

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

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

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

Images acquired by satellites are useful in tracking of

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

Space image applications include:

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

Communication over computer networks

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

Medical applications:

- Processing of chest X- rays
- Cin angiograms
- Projection images of trans axial tomography and
- Medical images that occur in radiology nuclear magnetic resonance (NMR)
- Ultrasonic scanning

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

Components of Image processing System:

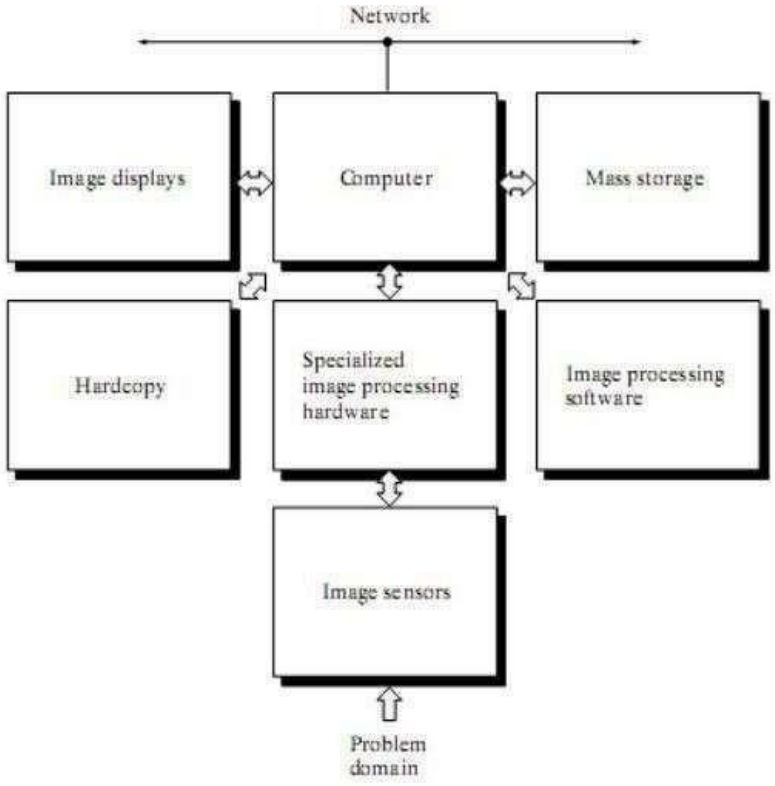


Figure 1.2: Components of Image processing System

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 is used to achieve a required level of performance

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

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

i) Short term storage for use during processing

ii) On line storage for relatively fast retrieval

iii) Archival storage such as magnetic tapes and disks

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

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

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

Fundamental Steps in Digital Image Processing:

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

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

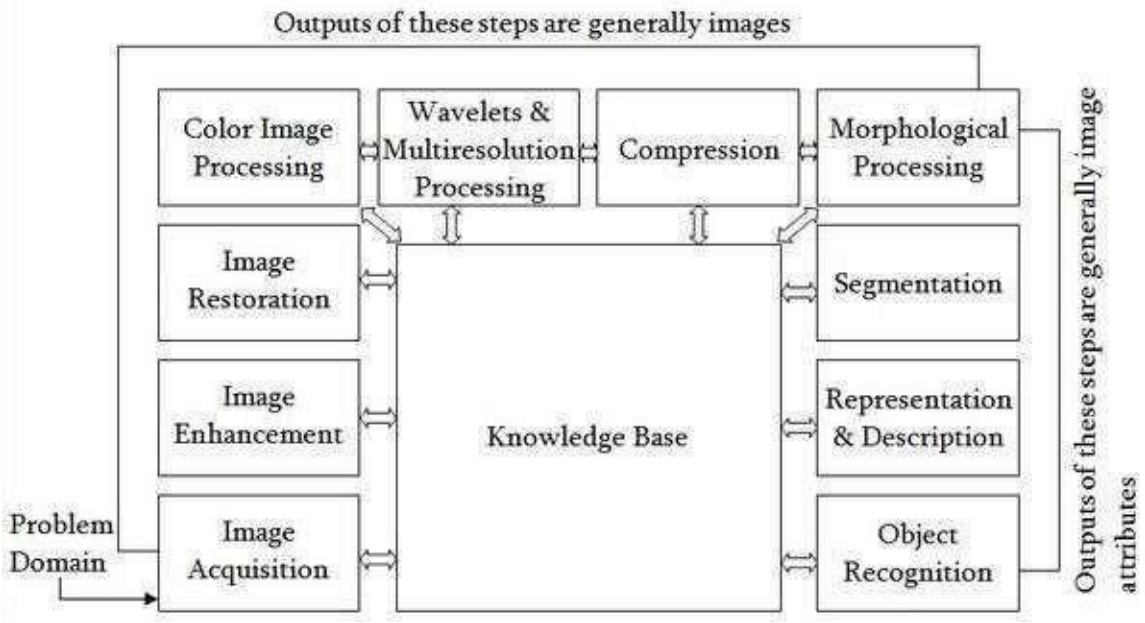


Fig 1.3 : Fundamental Steps in Digital Image Processing

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

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

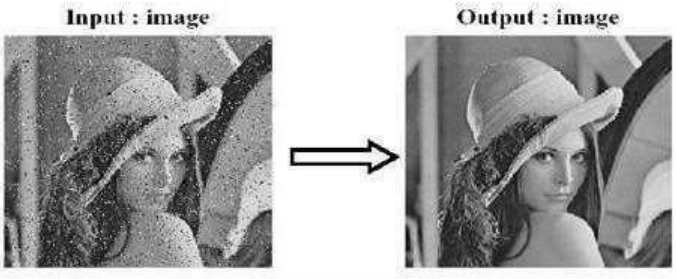
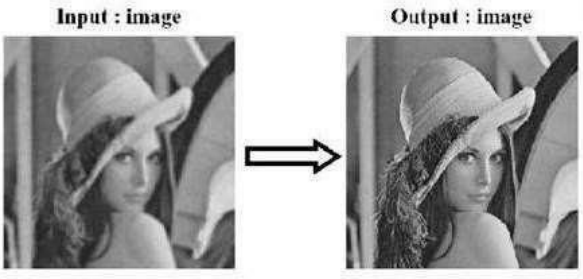


Image Restoration: It deals with improving the appearance of an image. It is an objective approach, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding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



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

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

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

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

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

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

Knowledge base:

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

A Simple Image Model:

An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is

determined by the source of the image. When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be nonzero and finite; that is o<f(x,y) <co The function f(x,y) may be characterized by two components- The amount of the source illumination incident on the scene being viewed.

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

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

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

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

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

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

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

SAMPLING AND QUANTIZATION:

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

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

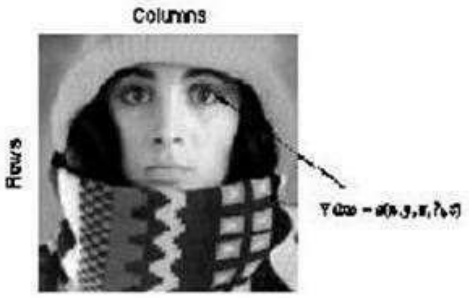
In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So we divide the gray level scale into eight discrete levels ranging from eight level values. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels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

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

Digital Image definition:

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

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



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

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

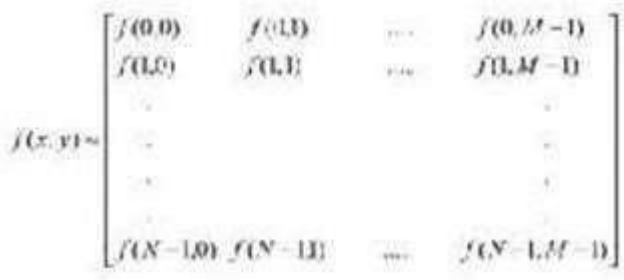
2) Mid-level image processing - it involves tasks like segmentation, description of those objects to reduce them to a form suitable for computer processing, and classification of individual objects. The inputs to the process are generally images but outputs are attributes extracted from images.

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

Representing Digital Images:

The result of sampling and quantization is matrix of real numbers. Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N Columns. The values of

the coordinates (x,y) now become discrete quantities thus the value of the coordinates at origin become (x,y) =(0,0) The next Coordinates value along the first signify the image along the first row. it does not mean that these are the actual values of physical coordinates when the image was sampled.



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

$$L=2^k$$

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

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

Spatial and Gray level resolution:

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

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

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

Image sensing and Acquisition:

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

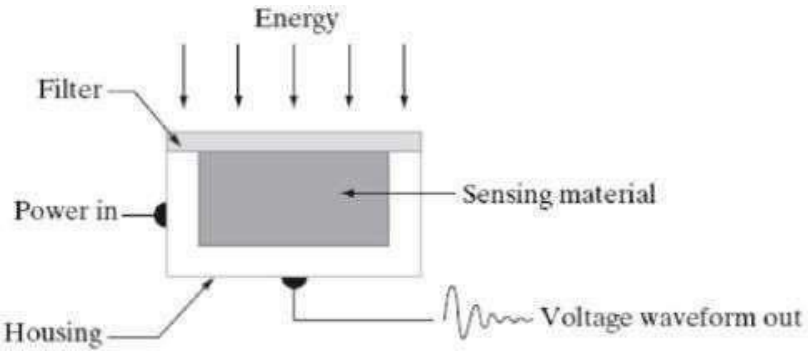


Fig: Single Image sensor

Fig: Line Sensor

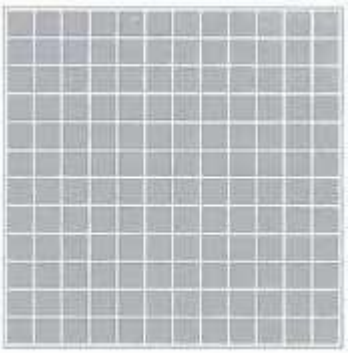
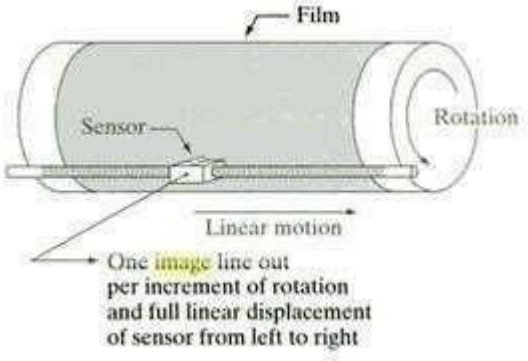


Fig: Array sensor

Image Acquisition using a Single sensor:

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



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

Image Acquisition using a Sensor strips:

A geometry that is used much more frequently than single sensors consist 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 flatbed 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.

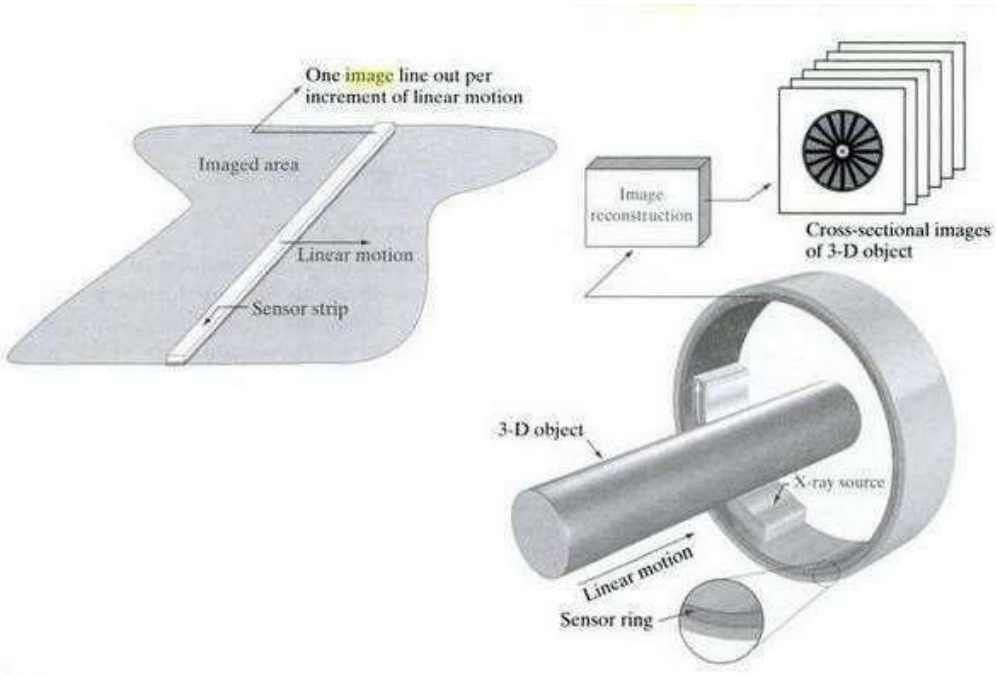
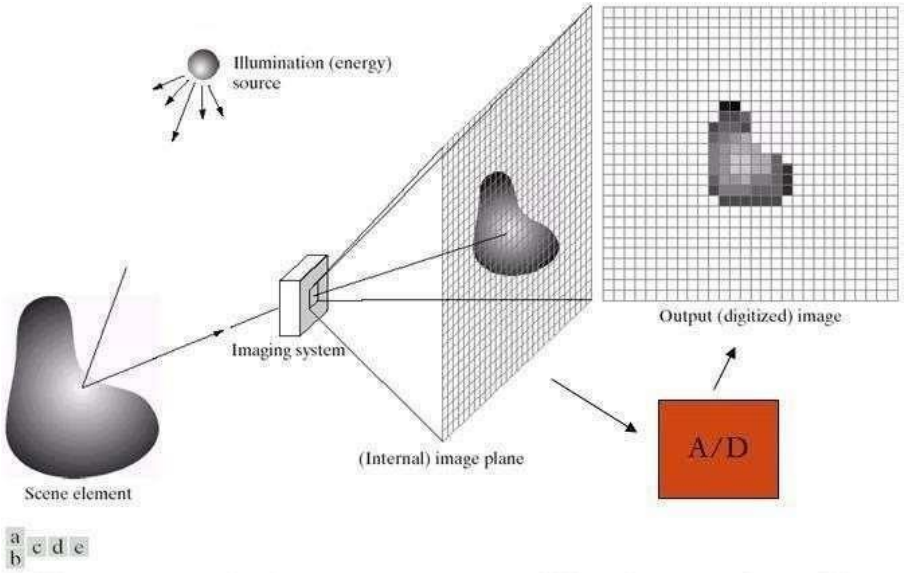


Fig1.4 : Image Acquisition using linear strip and circular strips.

Image Acquisition using a Sensor Arrays:

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

to the integral of the light received at each sensor. Digital and analog circuitry sweep these outputs and convert them to a video signal, which is then digitized by another section of the imaging system.

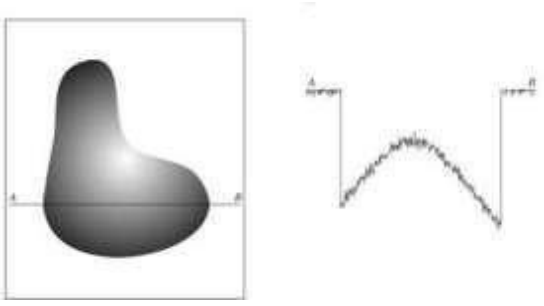


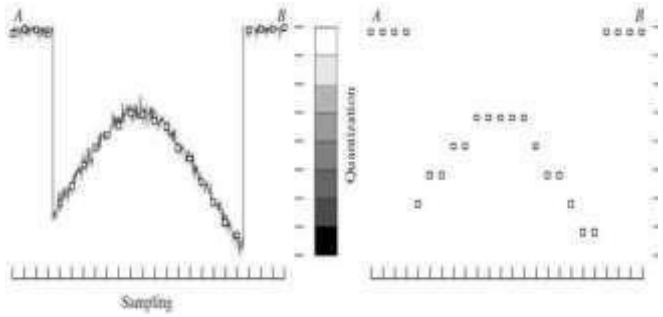
a b c d e

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

Image sampling and Quantization:

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





Digital Image representation:

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

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

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

RELATIONSHIP BETWEEN PIXELS:

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

NEIGHBORS OF A PIXEL

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

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

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

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

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

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

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

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

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

ADJACENCY AND CONNECTIVITY

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

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

if we are reference to adjacency of pixel with value.

Three types of adjacency

- 4- Adjacency – two-pixel P and Q with value from V are 4 –adjacency if A is in the set $N_4(P)$

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

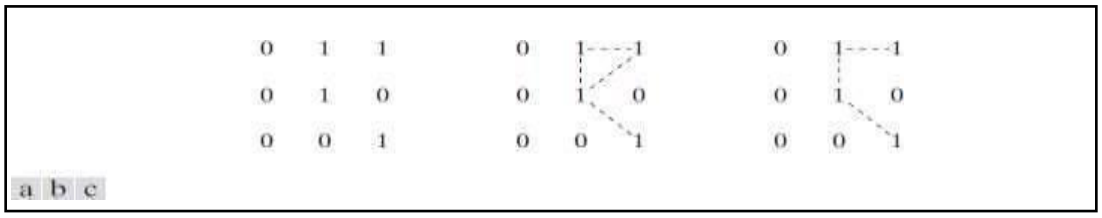


Fig:1.5(a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) ***m***-adjacency.

Types of Adjacency:

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

A Digital Path:

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

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

- Return to the previous example:

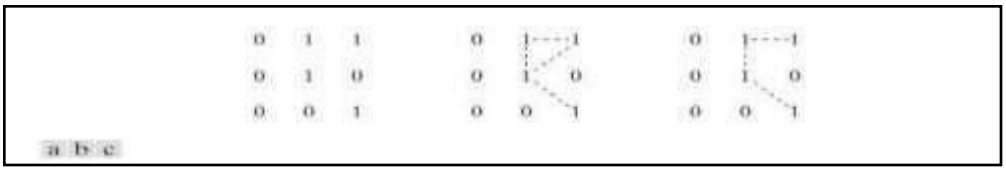


Fig:1.6 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to thecenter pixel; (c) m-adjacency.

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

Connectivity:

- Let *S* represent a subset of pixels in an image, two pixels *p* and *q* are said to be connected in *S* if there exists a path between them consisting entirely of pixels in *S*.
- For any pixel *p* in *S*, the set of pixels that are connected to it in *S* is called a *connected component* of *S*. If it only has one connected component, then set *S* is called a *connected set*.

Region and Boundary:

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

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

DISTANCE MEASURES:

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

D [p.q] ≥ O {D[p.q] = O iff p=q}

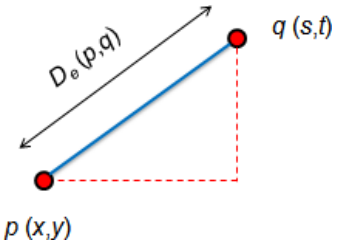
D [p.q] = D [p.q] and

D [p.q] ≥ O {D[p.q]+D(q,z)

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

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

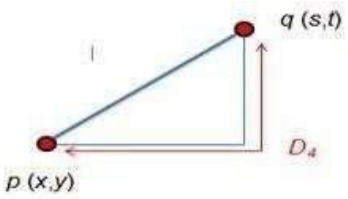
Pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius „ r „centered at (x,y)



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

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

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



Example:

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

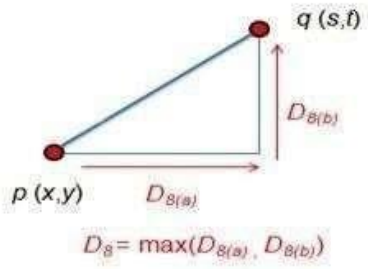
The pixels with $D_4 = 1$ are the 4-neighbors of (x,y)

		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

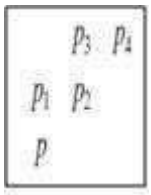
- **D_m distance:**

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

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

- Example:

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



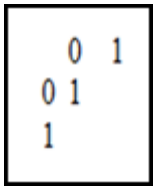
Now, to compute the D_m between points p and p_4

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)*

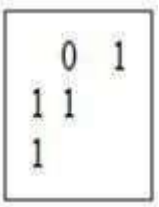


Case2: If p_1 =1 and p_3 = 0

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

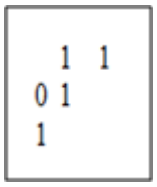
then, the length of the shortest

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



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)

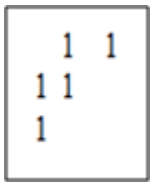
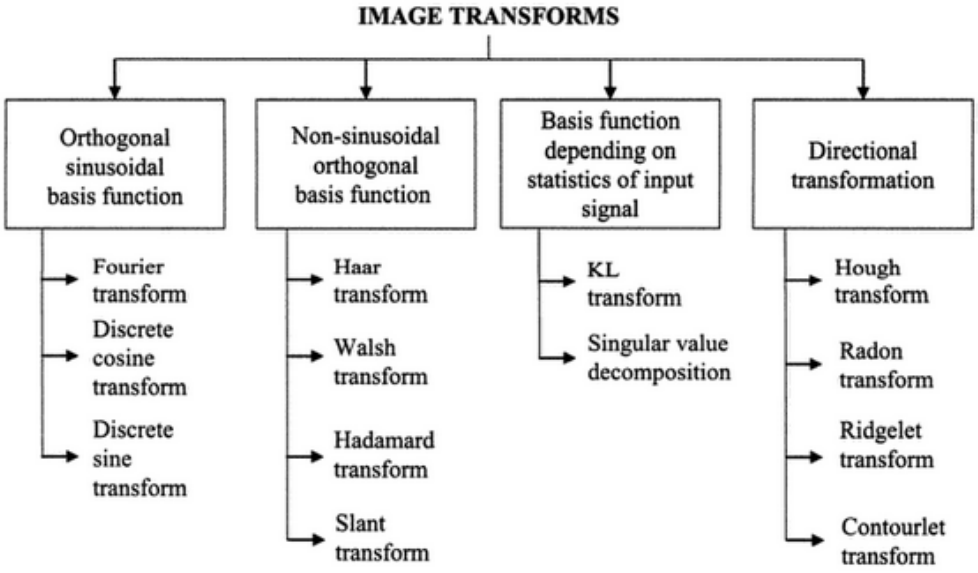


IMAGE TRANSFORMS:



2-D FFT:

2D Discrete Fourier Transform

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

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

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

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

Properties

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

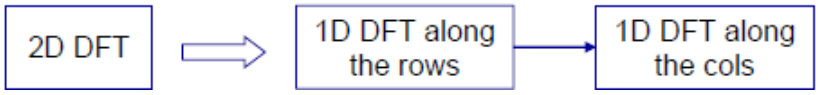
Separability

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

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

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

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



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

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

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

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

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

HADAMARD TRANSFORM:

4.9 HADAMARD TRANSFORM

The Hadamard transform is basically the same as the Walsh transform except the rows of the transform matrix are re-ordered. The elements of the mutually orthogonal basis vectors of a Hadamard transform are either +1 or −1, which results in very low computational complexity in the calculation of the transform coefficients. Hadamard matrices are easily constructed for $N = 2^n$ by the following procedure

The order $N = 2$ Hadamard matrix is given as

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

The Hadamard matrix of order $2N$ can be generated by Kronecker product operation:

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

Substituting $N = 2$ in Eq. (4.59), we get

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right]$$

DISCRETE COSINE TRANSFORM (DCT) :

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

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

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

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

where

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

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

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

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

where

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

The basic operation of the DCT is as follows:

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

UNIT -II

IMAGE PROCESSING TECHNIQUES

IMAGE ENHANCEMENT

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

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

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

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

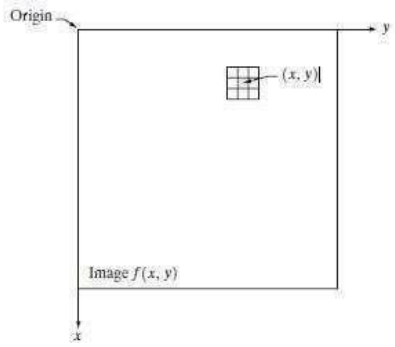


Fig.2.1 : 3x3 neighborhood about a point (x,y) in an image.

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

intensity or mapping) transformation function of the form

$$s = T (r)$$

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

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

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

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

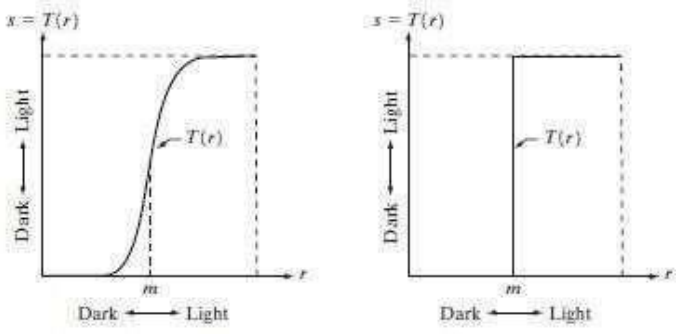


Fig. 2.2 Gray level transformation functions for contrast enhancement.

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

BASIC GRAY LEVEL TRANSFORMATIONS:

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

LINEAR TRANSFORMATION:

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

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

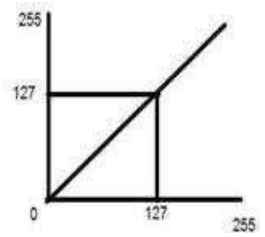


Fig.2.3 Linear transformation between input and output.

NEGATIVE TRANSFORMATION:

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

IMAGENEGATIVE: The image negative with gray level value in the range o f[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 photo graph negative. It is useful when for enhancing white details embedded in dark regions of the image.

The overall graph of these transitions has been shown below.

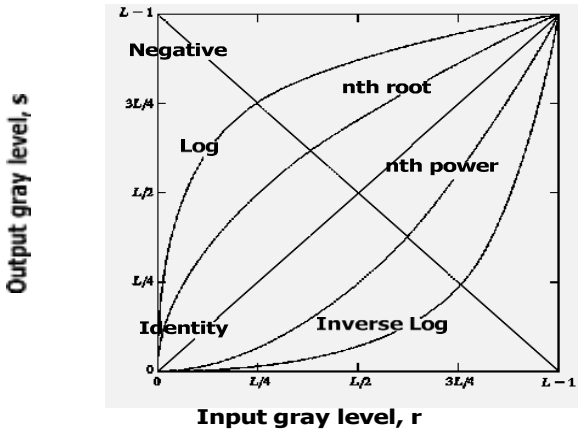


Fig. Some basic gray-level transformation functions used for image enhancement.

In this case the following transition has been done.

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

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

$$S = 255 - r$$

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

It has been shown in the graph below.

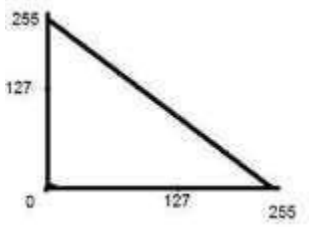


Fig. Negative transformations.

LOGARITHMIC TRANSFORMATIONS:

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

LOG TRANSFORMATIONS:

The log transformations can be defined by this formula

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

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

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

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

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

- Here C is constant and $r \geq 0$.
- Theshapeofthecurveshowsthatthistransformationmapsthenarrowrangeoflowgraylevelvaluesin the input image into a wider range of output image.
- The opposite is true for high level values of input image.

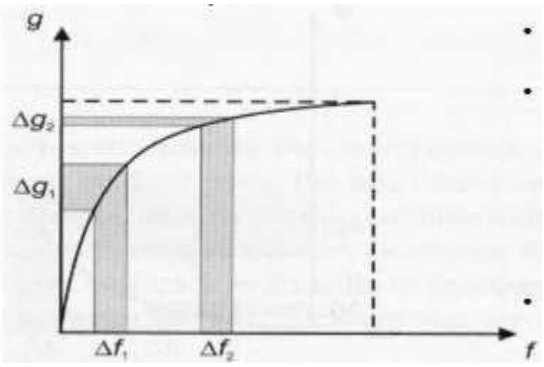


Fig. log transformation curve input vs output

POWER – LAW TRANSFORMATIONS:

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

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

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

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

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

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

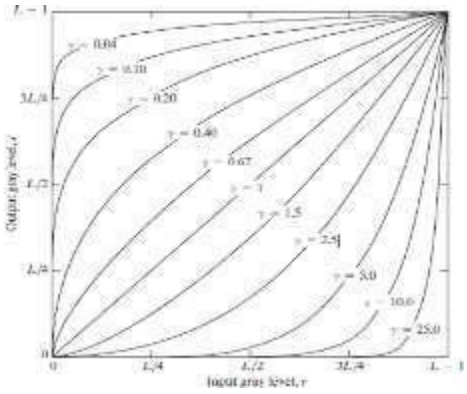


Fig. 2.13 Plot of the equation $S = cr^{\gamma}$ for various values of γ (c =1 in all cases).

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

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

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

γ isγ > 1 compresses dark values

Expands bright values

γ <1(similar to Log transformation)

Expands dark values

Compresses bright values

When C = γ = 1 , it reduces to identity transformation .

CORRECTING GAMMA:

$s=cr^{\gamma}$

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

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

Piecewise-Linear Transformation Functions:

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

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

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

$$S= T(r)$$

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

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

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

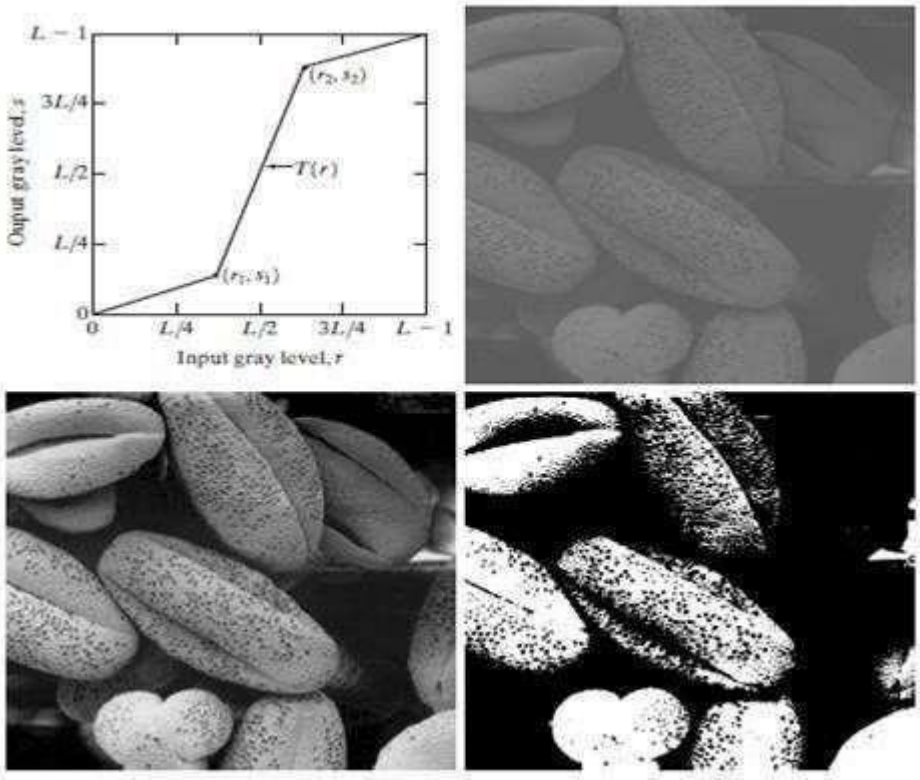


Fig. x Contrast stretching. (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of high contrast stretching. (d) Result of thresholding

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

Gray-level slicing:

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

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

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

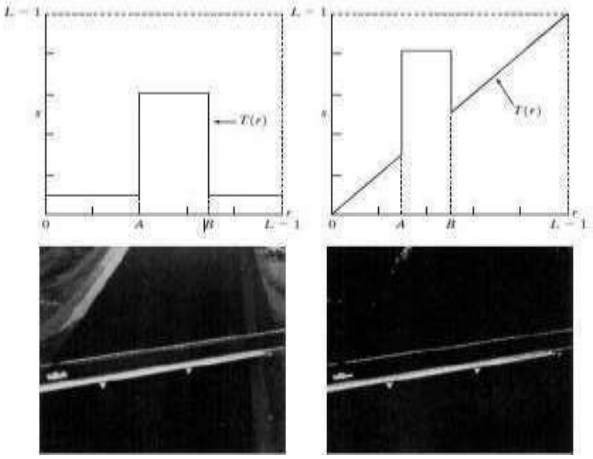


Fig. y (a)This transformation highlights range [A,B] of gray levels and reduces all others to a constant level (b) This transformation highlights range [A,B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

BIT-PLANE SLICING:

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

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

importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

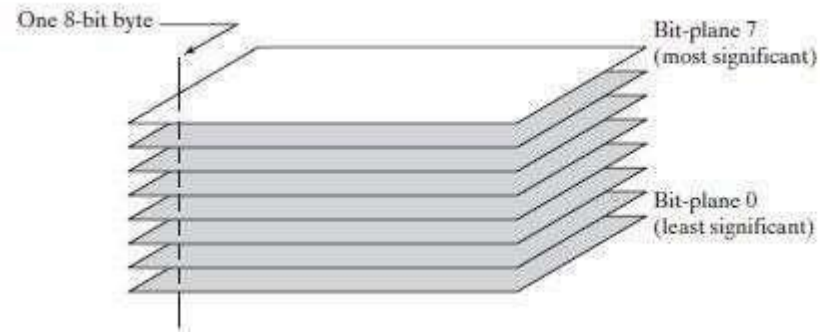


FIGURE Bit-plane representation of an 8-bit image.

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

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

Histogram Processing:

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

$$H(r_k)=n_k$$

where r_k is the k^{th} gray level and n_k is the number of pixels in the image having the level r_k..
A normalized histogram is given by the equation

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

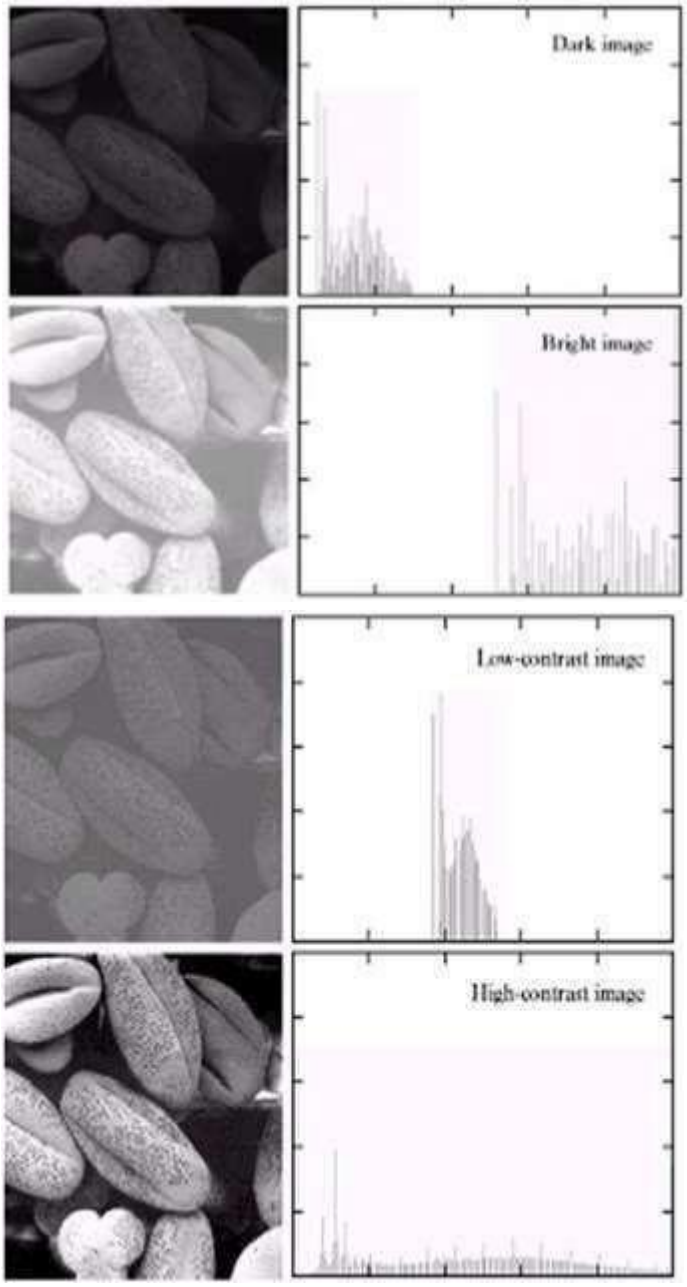
$P(r_k)$ gives the estimate of the probability of occurrence of gray level r_k.

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

The histogram plots are simple plots of $p(r_k)=n_k$ versus r_k.

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 biased towards the high side of the gray scale. The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.

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



Histogram Equalization:

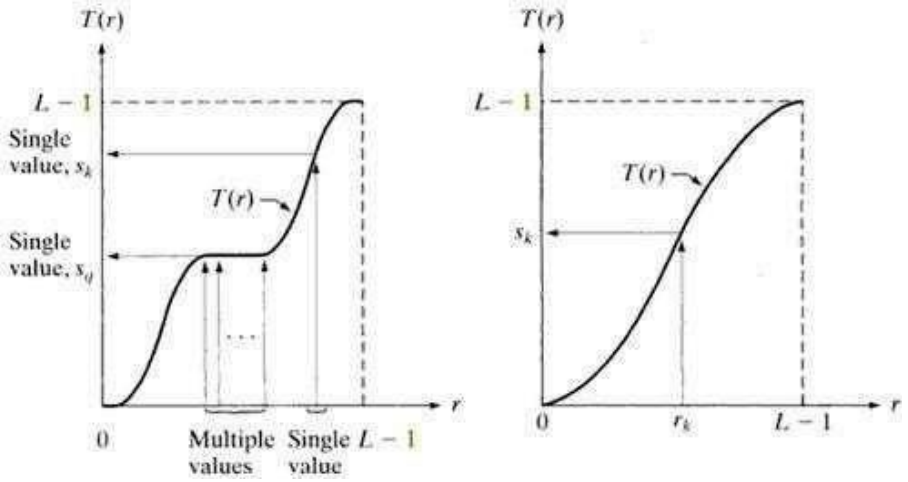
Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail are compressed into the dark

end of the histogram. If we could „stretch out" the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.

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

$$S=T(r) \text{ where } 0<r<1$$

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



a b

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

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

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

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

A transformation function of a particular importance in image processing

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

This is the cumulative distribution function of r.

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

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

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

IDEAL LOW-PASS FILTER:

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

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

$$0, \text{ if } D(u,v) > D_0$$

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

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

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

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

VISUALIZATION: IDEAL LOW PASS FILTER:

Aa shown in fig.below

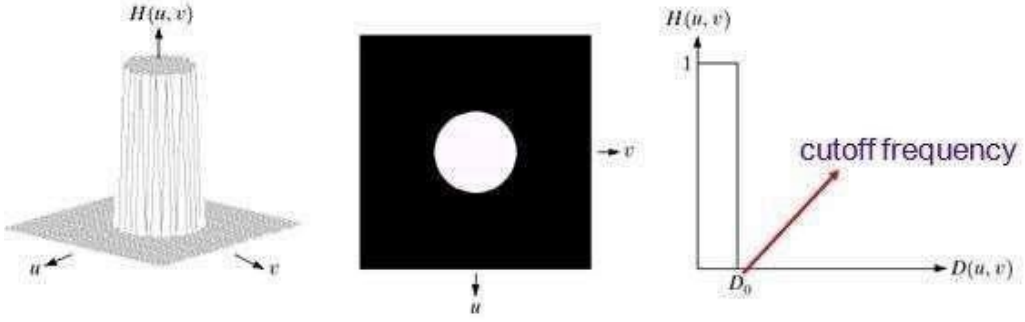


Fig: ideal low pass filter 3-D view and 2-D view and line graph.

EFFECT OF DIFFERENT CUT OFF FREQUENCIES:

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

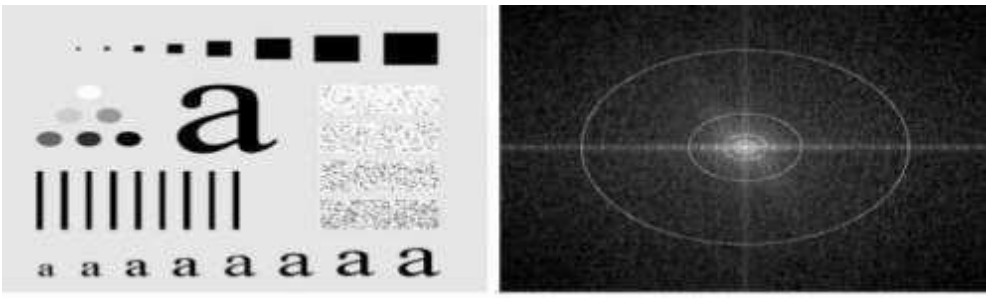


Fig: (a) Test patter of size 688x688 pixels (b) its Fourier spectrum

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

As the cutoff frequency decreases,

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

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

WHY IS THERE RINGING?

Ideal low-pass filter function is a rectangular function

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

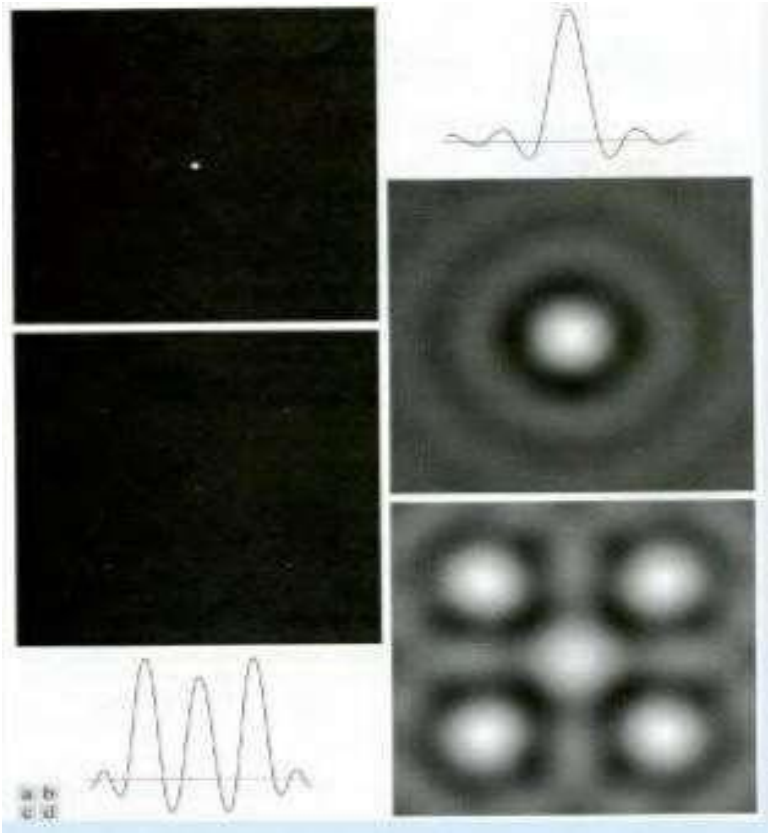


Fig. Spatial representation of ILPFs of order 1 and 20 and corresponding intensity profiles through the center of the filters(the size of all cases is 1000x1000 and the cutoff frequency is 5), observe how ringing increases as a function of filter order.

BUTTERWORTH LOW-PASS FILTER:

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

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

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

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

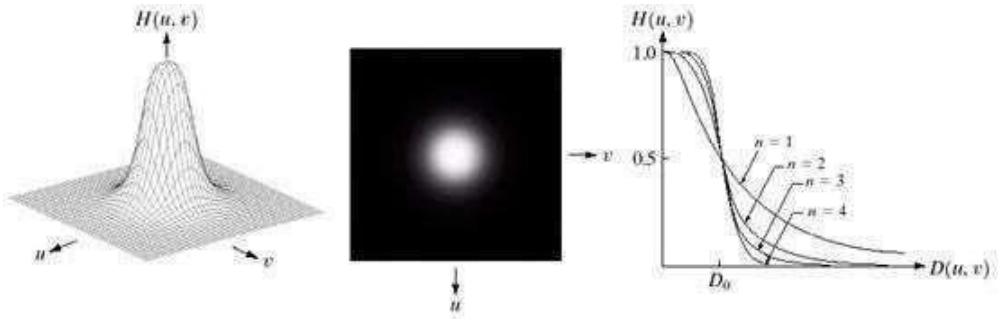


Fig. (a) perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c)Filter radial cross sections of order 1 through 4.

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

BUTTERWORTH LOW-PASS FILTERS OF DIFFEREN T FREQUENCIES:

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

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

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

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

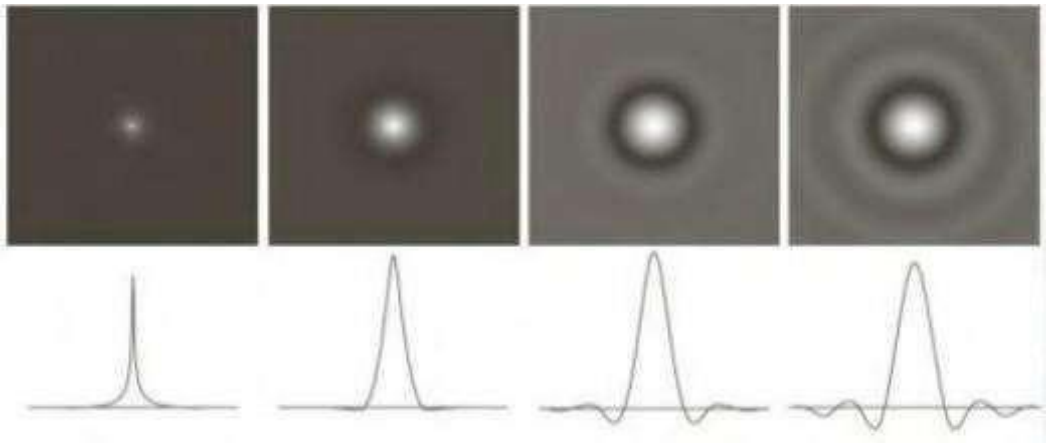


Fig.2.2.7 (a)-(d) Spatial representation of BLPFs of order 1, 2, 5 and 20 and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 x 1000 and the cutoff frequency is 5) Observe how ringing increases as a function of filter order.

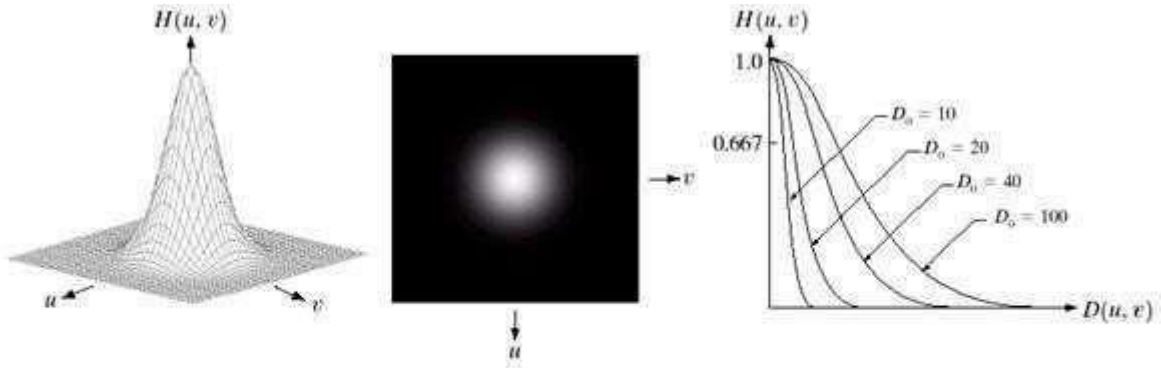
GAUSSIAN LOWPASS FILTERS:

The form of these filters in two dimensions is given by

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

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

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



a b c

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

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

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

a b c

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

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

IMAGE SHARPENING USING FREQUENCY DOMAIN FILTERS:

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

The filter function H(u,v) are understood to be discrete functions of size PxQ; that is the discrete frequency variables are in the range u = 0,1,2,…….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.

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

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

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

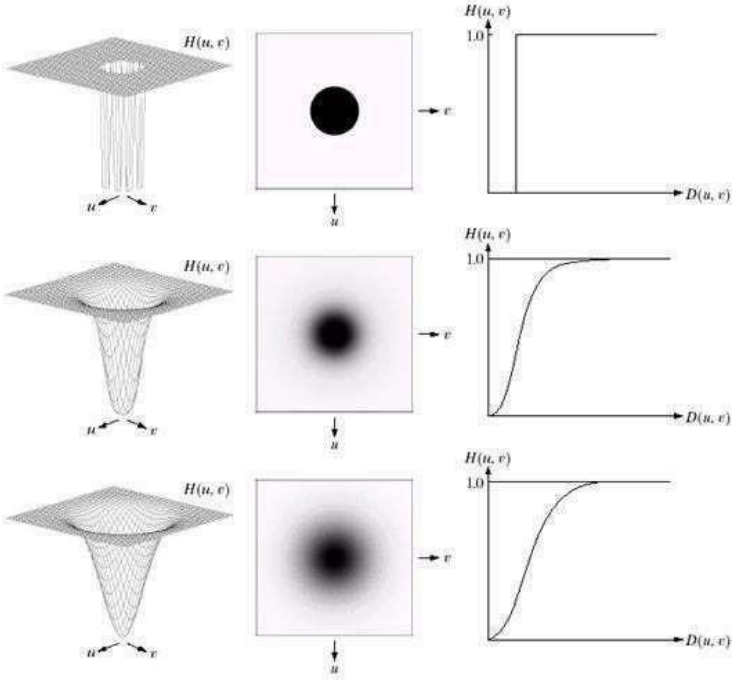


Fig: Top row: Perspective plot, image representation, and cross section of a typical ideal high-pass filter. Middle and bottom rows: The same sequence for typical butter-worth and Gaussian high-pass filters.

IDEAL HIGH-PASS FILTER:

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

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

$$1, \text{ if } D(u,v) > D_0$$

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

SPATIAL REPRESENTATION OF HIGHPASS FILTERS:

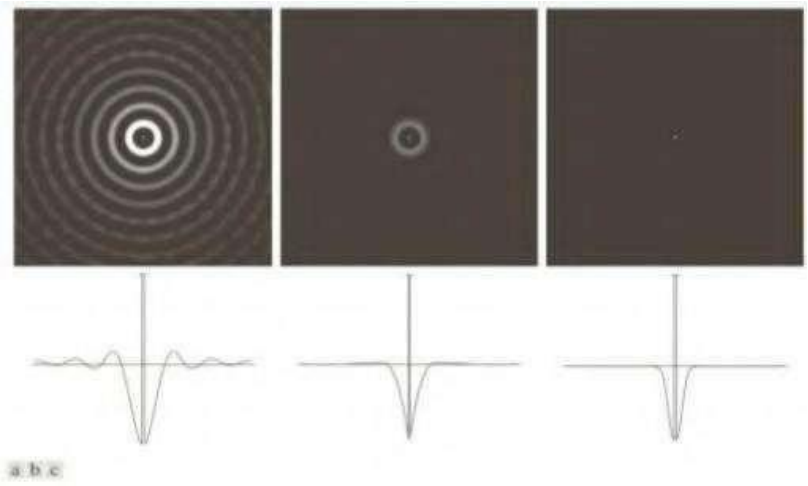


Fig.. Spatial representation of typical (a) ideal (b) Butter-worth and (c) Gaussian frequency domain high-pass filters, and corresponding intensity profiles through their centers.

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

FILTERED RESULTS: IHPF:

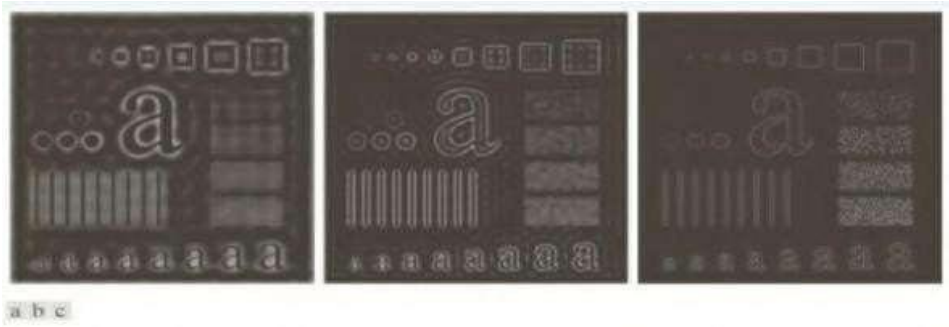


Fig.. Results of high-pass filtering the image in Fig.(a) using an IHPF with D_0 = 30, 60, and 160.

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

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

BUTTER-WORTH HIGH-PASS FILTERS:

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

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

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

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

FILTERED RESULTS: BHPF:

a b c

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

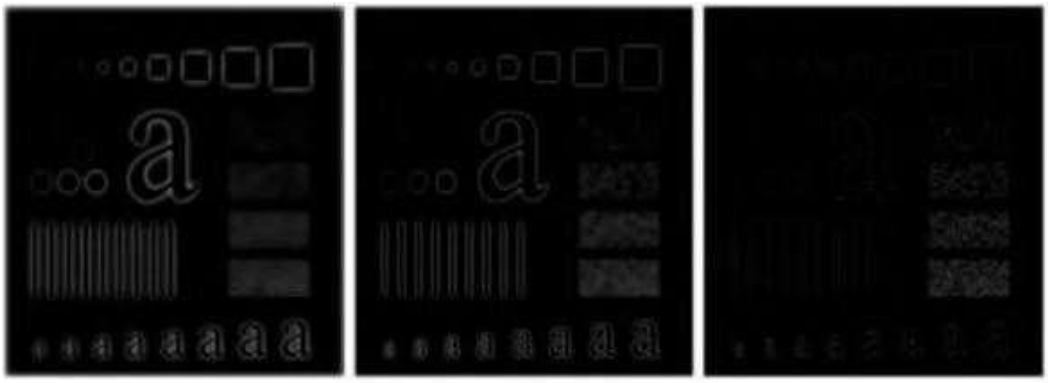
GAUSSIAN HIGH-PASS FILTERS:

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

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

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

FILTERED RESULTS:GHPF:



a b c

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

IMAGE RESTORATION:

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

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

Degradation Model:

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

f'(x,y). If it is a linear position invariant process, then degraded image is given in the spatial domain by

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

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

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

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

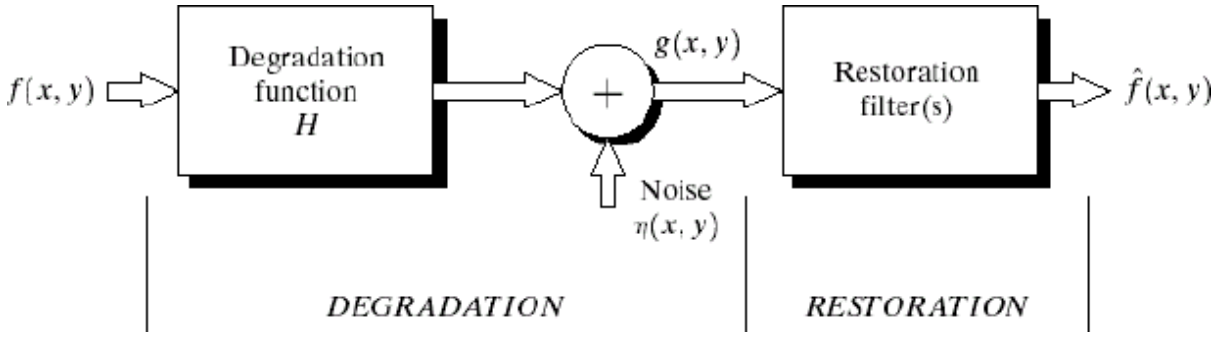


Fig: A model of the image Degradation / Restoration process

Noise Models:

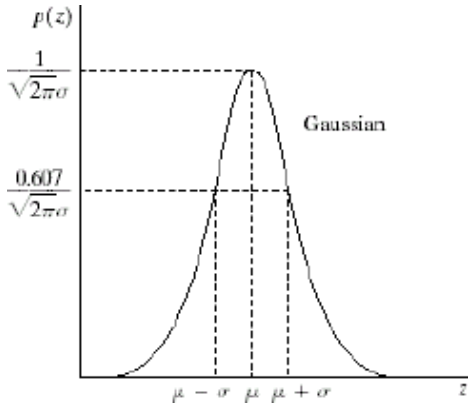
The principal source of noise in digital images arises during image acquisition and /or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted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

Gaussian Noise:

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

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

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



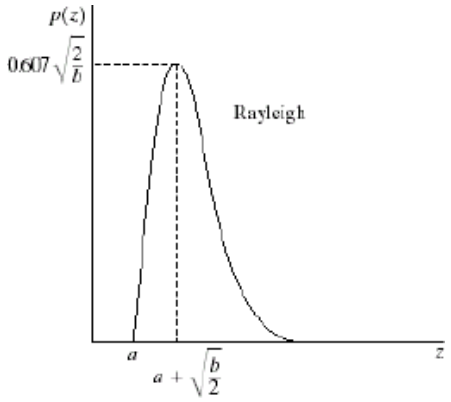
Rayleigh Noise:

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

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

The mean and variance of this density is

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



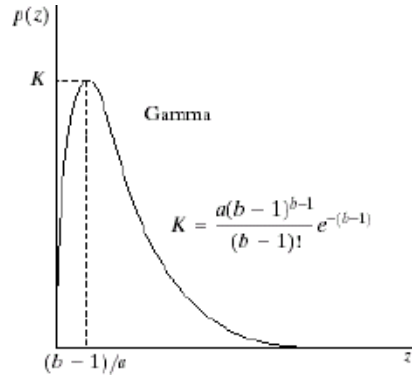
(iii) Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

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



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

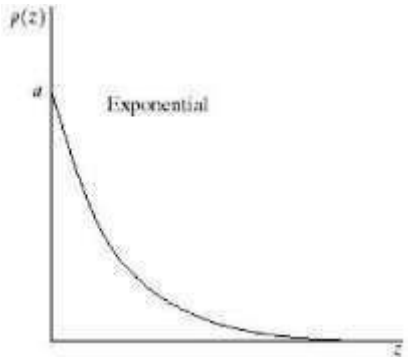
(iv) Exponential Noise:

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

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

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

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



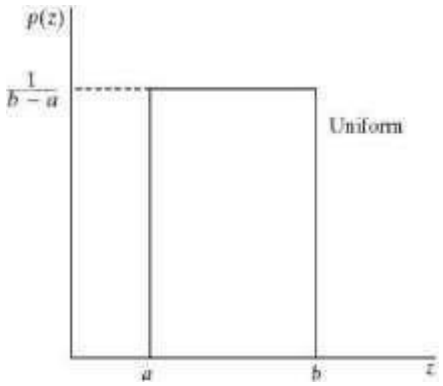
(v) Uniform Noise:

The PDF of uniform noise is given by

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

The mean and variance of this noise is

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



(vi) Impulse (salt & pepper) Noise:

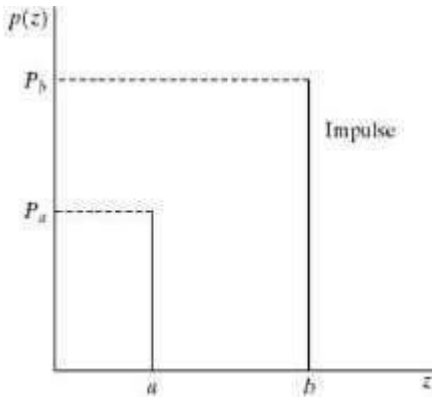
In this case, the noise is signal dependent, and is multiplied to the image.

The PDF of bipolar (impulse) noise is given by

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

$$b > a$$

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



Restoration in the presence of Noise only- Spatial filtering:

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

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

or

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

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

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

i) Mean Filter:

ii) (a)Arithmetic Mean filter:

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

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

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

(b)Geometric Mean filter:

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

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

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

(c) Harmonic Mean filter:

The harmonic mean filtering operation is given by the expression

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

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

(d) Order statistics filter:

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

(e) Median filter:

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

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

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

(e) Max and Min filter:

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

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

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

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

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

(f) Midpoint filter:

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

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

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

Periodic Noise by Frequency domain filtering:

These types of filters are used for this purpose-

Band Reject Filters:

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

Ideal Band reject Filter:

An ideal band reject filter is given by the expression

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

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

W- the width of the band

Do- the radial center of the frequency rectangle.

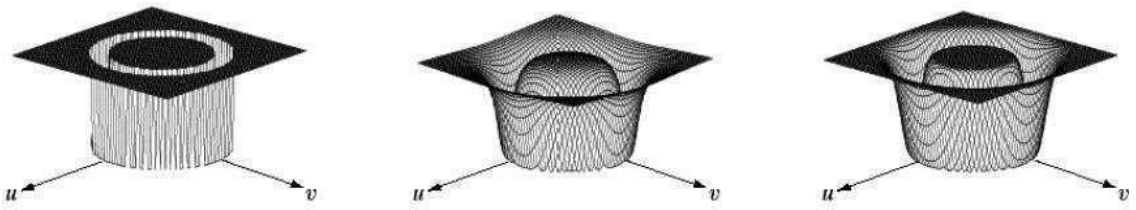
Butterworth Band reject Filter:

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

Gaussian Band reject Filter:

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

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



a b c

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

Band pass Filter:

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

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

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

Notch Filters:

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

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

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

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

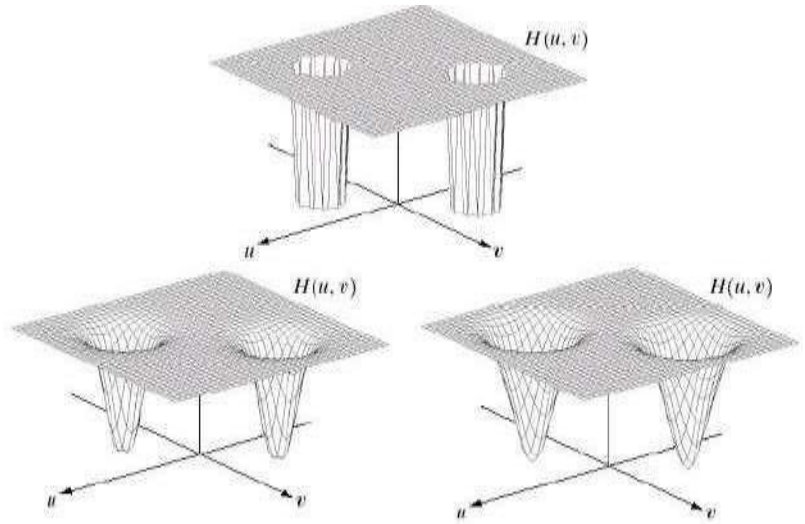
$$D_2(u,v) = \sqrt{(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2}$$

Ideal, butterworth, Gaussian notch filters

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

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

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



a
b c

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

Inverse Filtering:

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

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

We know that

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

Therefore

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

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

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

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

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

Minimum mean Square Error (Wiener) filtering:

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

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

The error measure is given by

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

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

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

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

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

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

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

Where H(u,v)= degradation function

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

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

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

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

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

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

Where K is a specified constant.

Constrained least squares filtering:

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

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

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

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

The minimum of a criterion function C defined as

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

Subject to the constraint

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

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

The frequency domain solution to this optimization problem is given by

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

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

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

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

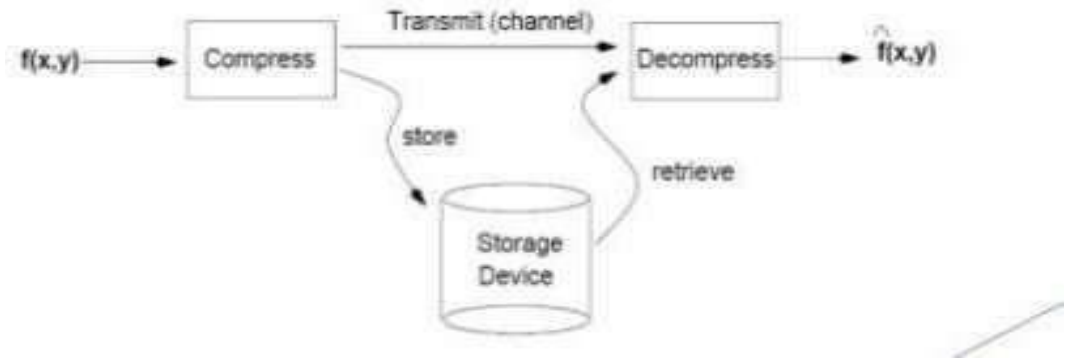
UNIT-III

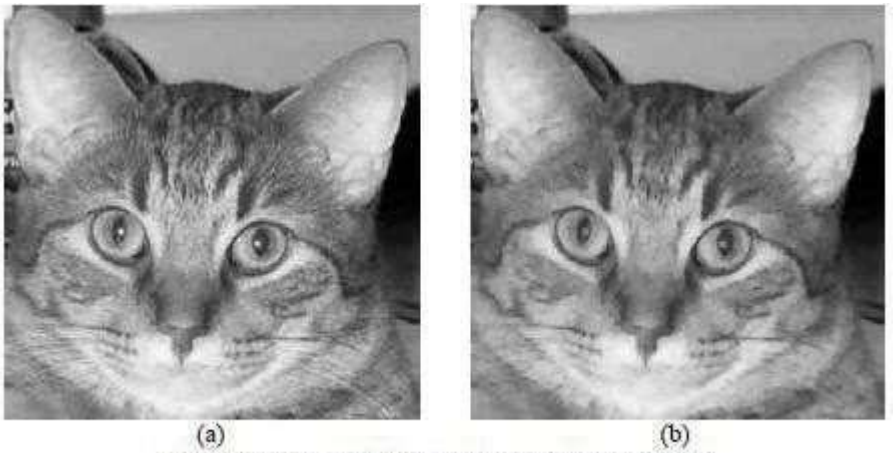
IMAGE COMPRESSION

Definition: Image compression deals with reducing the amount of data required to represent a digital image by removing of redundant data.

Images can be represented in digital format in many ways. Encoding the contents of a 2-D image in a raw bitmap (raster) format is usually not economical and may result in very large files. Since raw image representations usually require a large amount of storage space (and proportionally long transmission times in the case of file uploads/ downloads), most image file formats employ some type of compression. The need to save storage space and shorten transmission time, as well as the human visual system tolerance to a modest amount of loss, have been the driving factors behind image compression techniques.

Goal of image compression: The goal of image compression is to reduce the amount of data required to represent a digital image.





(a) (b)

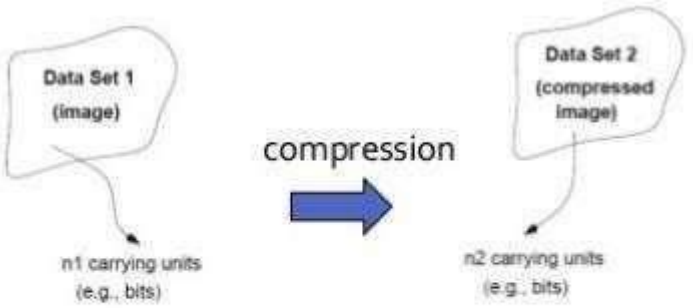
Figure 12.8 (a) Original gray image. (b) Opened image.

Data $\neq$ Information:

- Data and information are not synonymous terms!
- Data is the means by which information is conveyed.
- Data compression aims to reduce the amount of data required to represent a given quality of information while preserving as much information as possible.
- The same amount of information can be represented by various amount of data.

Ex1: You have an extra class after completion of 3.50 p.m
Ex2: Extra class have been scheduled after 7th hour for you.
Ex3: After 3.50 p.m you should attended extra class.

Definition of compression ratio:



Compression ratic $C_R = \frac{n_1}{n_2}$

Definitions of Data Redundancy:

➢ Relative data redundancy: $R_D = 1 - \frac{1}{C_R}$

Example:

If $C_R = \frac{10}{1}$, then $R_D = 1 - \frac{1}{10} = 0.9$
(90% of the data in dataset 1 is redundant)

if $n_2 = n_1$, then $C_R = 1$, $R_D = 0$
if $n_2 \ll n_1$, then $C_R \to \infty$, $R_D \to 1$

Coding redundancy:

- Code: a list of symbols (letters, numbers, bits etc.,)
- Code word: a sequence of symbol used to represent a piece of information or an event (e.g., gray levels).
- Code word length: number of symbols in each code word.

Example: (binary code, symbols: 0,1, length: 3)

0: 000	4: 100
1: 001	5: 101
2: 010	6: 110
3: 011	7: 111

N x M image
r_k: k-th gray level
$P(r_k)$: probability of r_k
$l(r_k)$: # of bits for r_k

Expected value:

$$E(X) = \sum_x xP(X = x)$$

Average # of bits: $L_{avg} = E(l(r_k)) = \sum_{k=0}^{L-1} l(r_k)P(r_k)$

Total # of bits: NML_{avg}

Coding Redundancy (con'd)

Case 1: $l(r_k)$ = constant length

Example:

r_k	$p(r_k)$	Code 1	$l(r_k)$
$r_0 = 0$	0.19	000	3
$r_1 = 1/7$	0.25	001	3
$r_2 = 2/7$	0.21	010	3
$r_3 = 3/7$	0.16	011	3
$r_4 = 4/7$	0.08	100	3
$r_5 = 5/7$	0.06	101	3
$r_6 = 6/7$	0.03	110	3
$r_7 = 1$	0.02	111	3

Assume an image with $L = 8$

Assume $l(r_k) = 3$, $L_{avg} = \sum_{k=0}^{7} 3P(r_k) = 3\sum_{k=0}^{7} P(r_k) = 3$ bits

Total number of bits: $3NM$

Case 2: $l(r_k)$ = variable length

r_k	$p(r_k)$	Code 1	$l(r_k)$	Code 2	$l(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

Table 6.1 Variable-Length Coding Example

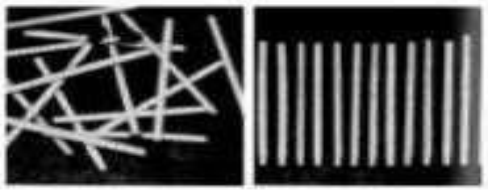
$$L_{avg} = \sum_{k=0}^{7} l(r_k)P(r_k) = 2.7 \text{ bits}$$

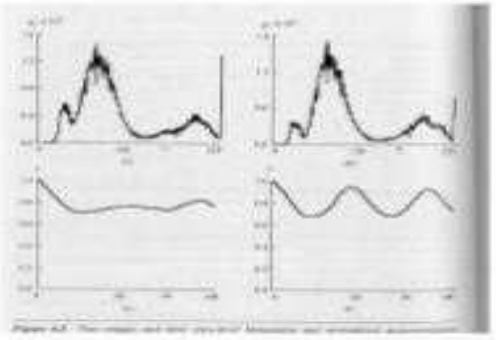
$$C_R = \frac{3}{2.7} = 1.11 \text{ (about 10\%)}$$

$$R_D = 1 - \frac{1}{1.11} = 0.099$$

Interpixel redundancy

- Interpixel redundancy implies that pixel values are correlated (i.e., a pixel value can be reasonably predicted by its neighbors).





$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f(x)g(x+a)da$$

autocorrelation: f(x)=g(x)

Interpixel redundancy (cont'd)

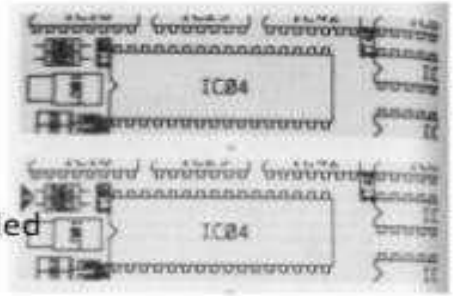
To reduce interpixel redundancy, the data must be transformed in another format (i.e., using a transformation)

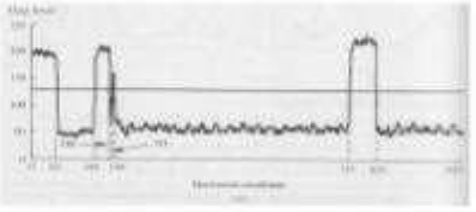
- e.g., thresholding, DFT, DWT, etc.

Example:

original

thresholded





Run-length encoding:

(1,63) (0,87) (1,37) (0,5) (1,4) (0, 556) (1,42) (0,258)

Using 11 bits/pair:

88 bits are required (compared to 1024 ?)

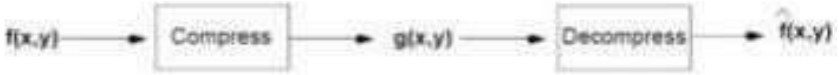
Psychovisual redundancy

The human eye does not respond with equal sensitivity to all visual information.

It is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.

Idea: discard data that is perceptually insignificant!

Fidelity Criteria



$$\hat{f}(x,y) = f(x,y) + e(x,y)$$

- How close $f(x,y)$ $\hat{f}(x,y)$?

- Criteria
  - Subjective: based on human observers
  - Objective: mathematically defined criteria

Subjective Fidelity Criteria

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Objective Fidelity Criteria

- Root mean square error (RMS)

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

- Mean-square

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\hat{f}(x,y))^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\hat{f}(x,y) - f(x,y))^2}$$

COMPRESSION METHODS OF IMAGES:

Compression methods can be ***lossy***,

when a tolerable degree of deterioration in the visual quality of the resulting image is acceptable,

or ***lossless***,

when the image is encoded in its full quality. The overall results of the compression process, both in terms of storage savings – usually expressed numerically in terms of compression ratio (CR) or bits per pixel (bpp) – as well as resulting quality loss (for the case of lossy techniques) may vary depending on the technique, format, options (such as the quality setting for JPEG), and the image contents.

As a general guideline, ***lossy compression*** should be used for general purpose photographic images.

whereas ***lossless compression*** should be preferred when dealing with line art, technical drawings, cartoons, etc. or images in which no loss of detail may be tolerable (most notably, space images and medical images).

Fundamentals of visual data compression

The general problem of image compression is to reduce the amount of data required to represent a digital image or video and the underlying basis of the reduction process is the removal of redundant data. Mathematically, visual data compression typically involves

transforming (encoding) a 2-D pixel array into a statistically uncorrelated data set. This transformation is applied prior to storage or transmission. At some later time, the compressed image is decompressed to reconstruct the original image information (preserving or lossless techniques) or an approximation of it (lossy techniques).

Redundancy

Data compression is the process of reducing the amount of data required to represent a given quantity of information. Different amounts of data might be used to communicate the same amount of information. If the same information can be represented using different amounts of data, it is reasonable to believe that the representation that requires more data contains what is technically called ***data redundancy.***

Image compression and coding techniques explore three types of redundancies: ***coding*** redundancy, ***interpixel*** (spatial) redundancy, and *psychovisual* redundancy. The way each of them is explored is briefly described below.

- **Coding redundancy:** consists in using variable-length codewords selected as to match the statistics of the original source, in this case, the image itself or a processed version of its pixel values. This type of coding is always reversible and usually implemented using look-up tables (LUTs). Examples of image coding schemes that explore coding redundancy are the Huffman codes and the arithmetic coding technique.
- **Interpixel redundancy:** this type of redundancy – sometimes called spatial redundancy, interframe redundancy, or geometric redundancy – exploits the fact that an image very often contains strongly correlated pixels, in other words, large regions whose pixel values are the same or almost the same. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighboring pixels. In order to do so, the original 2-D array of pixels is usually mapped into a different format, e.g., an array of differences between adjacent pixels. If the original image pixels can be reconstructed from the transformed data set the mapping is said to be reversible. Examples of compression techniques that explore the interpixel redundancy include: Constant Area Coding (CAC), (1-D or 2-D) Run-Length Encoding (RLE) techniques, and many predictive coding algorithms such as Differential Pulse Code Modulation (DPCM).

- **Psycho visual redundancy: many experiments on the psychophysical aspects of human** vision have proven that the human eye does not respond with equal sensitivity to all incoming visual information; some pieces of information are more important than others. The knowledge of which particular types of information are more or less relevant to the final human user have led to image and video compression techniques that aim at eliminating or reducing any amount of data that is psycho visually redundant. The end result of applying these techniques is a compressed image file, whose size and quality are smaller than the original information, but whose resulting quality is still acceptable for the application at hand.

The loss of quality that ensues as a byproduct of such techniques is frequently called *quantization,* as to indicate that a wider range of input values is normally mapped into a narrower range of output values thorough an irreversible process. In order to establish the nature and extent of information loss, different fidelity criteria (some objective such as root

mean square (RMS) error, some subjective, such as pair wise comparison of two images encoded with different quality settings) can be used. Most of the image coding algorithms in use today exploit this type of redundancy, such as the Discrete Cosine Transform (DCT)-based algorithm at the heart of the JPEG encoding standard.

IMAGE COMPRESSION AND CODING MODELS

Figure 1 shows a general image compression model. It consists of a source encoder, a channel encoder, the storage or transmission media (also referred to as *channel*), a channel decoder, and a source decoder. The source encoder reduces or eliminates any redundancies in the input image, which usually leads to bit savings. Source encoding techniques are the primary focus of this discussion. The channel encoder increase noise immunity of source encoder's output, usually adding extra bits to achieve its goals. If the channel is noise-free, the channel encoder and decoder may be omitted. At the receiver's side, the channel and source decoder perform the opposite functions and ultimately recover (an approximation of) the original image.

Figure 2 shows the source encoder in further detail. Its main components are:

- **Mapper:** transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation is generally reversible and may or may not directly reduce the amount of data required to represent the image.

- **Quantizer:** reduces the accuracy of the mappers output in accordance with some pre-established fidelity criterion. Reduces the psychovisual redundancies of the input image. This operation is not reversible and must be omitted if lossless compression is desired.

- **Symbol (entropy) encoder:** creates a fixed- or variable-length code to represent the quantizes output and maps the output in accordance with the code. In most cases, a variable-length code is used. This operation is reversible.

Error-free compression

Error-free compression techniques usually rely on entropy-based encoding algorithms. The concept of entropy is mathematically described in equation (1):

where:

a_j is a symbol produced by the information source

$P(a_j)$ is the probability of that symbol

J is the total number of different symbols

$H(z)$ is the entropy of the source.

The concept of entropy provides an upper bound on how much compression can be achieved, given the probability distribution of the source. In other words, it establishes a theoretical limit on the amount of lossless compression that can be achieved using entropy encoding techniques alone.

Variable Length Coding (VLC)

Most entropy-based encoding techniques rely on assigning variable-length codewords to each symbol, whereas the most likely symbols are assigned shorter codewords. In the case of image coding, the symbols may be raw pixel values or the numerical values obtained at the output of the mapper stage (e.g., differences between consecutive pixels, run-lengths, etc.). The most popular entropy-based encoding technique is the Huffman code. It provides the least amount of information units (bits) per source symbol. It is described in more detail in a separate short article.

Run-length encoding (RLE)

RLE is one of the simplest data compression techniques. It consists of replacing a sequence (run) of identical symbols by a pair containing the symbol and the run length. It is used as the primary compression technique in the 1-D CCITT Group 3 fax standard and in conjunction with other techniques in the JPEG image compression standard (described in a separate short article).

Differential coding

Differential coding techniques explore the interpixel redundancy in digital images. The basic idea consists of applying a simple difference operator to neighboring pixels to calculate a difference image, whose values are likely to follow within a much narrower range than the original gray-level range. As a consequence of this narrower distribution – and consequently reduced entropy – Huffman coding or other VLC schemes will produce shorter codewords for the difference image.

Predictive coding

Predictive coding techniques constitute another example of exploration of interpixel redundancy, in which the basic idea is to encode only the new information in each pixel. This new information is usually defined as the difference between the actual and the predicted value of that pixel.

Figure 3 shows the main blocks of a lossless predictive encoder. The key component is the predictor, whose function is to generate an estimated (predicted) value for each pixel from the input image based on previous pixel values. The predictor's output is rounded to the nearest integer and compared with the actual pixel value: the difference between the two –

called *prediction error* – is then encoded by a VLC encoder. Since prediction errors are likely to be smaller than the original pixel values, the VLC encoder will likely generate shorter codewords.

There are several local, global, and adaptive prediction algorithms in the literature. In most cases, the predicted pixel value is a linear combination of previous pixels.

Dictionary-based coding

Dictionary-based coding techniques are based on the idea of incrementally building a dictionary (table) while receiving the data. Unlike VLC techniques, dictionary-based techniques use fixed-length codewords to represent variable-length strings of symbols that commonly occur together. Consequently, there is no need to calculate, store, or transmit the probability distribution of the source, which makes these algorithms extremely convenient and popular. The best-known variant of dictionary-based coding algorithms is the LZW (Lempel-Ziv-Welch) encoding scheme, used in popular multimedia file formats such as GIF, TIFF, and PDF.

Lossy compression

Lossy compression techniques deliberately introduce a certain amount of distortion to the encoded image, exploring the psychovisual redundancies of the original image. These techniques must find an appropriate balance between the amount of error (loss) and the resulting bit savings.

Quantization

The quantization stage is at the core of any lossy image encoding algorithm. Quantization, in at the encoder side, means partitioning of the input data range into a smaller set of values. There are two main types of quantizers: scalar quantizers and vector quantizers. A scalar quantizer partitions the domain of input values into a smaller number of intervals. If the output intervals are equally spaced, which is the simplest way to do it, the process is called *uniform scalar quantization;* otherwise, for reasons usually related to minimization of total distortion, it is called *nonuniform scalar quantization.* One of the most popular nonuniform quantizers is the Lloyd-Max quantizer. Vector quantization (VQ) techniques extend the basic principles of scalar quantization to multiple dimensions. Because of its fast lookup capabilities at the decoder side, VQ-based coding schemes are particularly attractive to multimedia applications.

Transform coding

The techniques discussed so far work directly on the pixel values and are usually called *spatial domain techniques.* Transform coding techniques use a reversible, linear mathematical transform to map the pixel values onto a set of coefficients, which are then quantized and encoded. The key factor behind the success of transform-based coding schemes many of the resulting coefficients for most natural images have small magnitudes and can be quantized (or discarded altogether) without causing significant distortion in the decoded image. Different mathematical transforms, such as Fourier (DFT), Walsh-Hadamard (WHT), and Karhunen-Loeve (KLT), have been considered for the task. For compression purposes, the higher the capability of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) has become the

most widely used transform coding technique.

Wavelet coding

Wavelet coding techniques are also based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. The main difference between wavelet coding and DCT-based coding (Figure 4) is the omission of the first stage. Because wavelet transforms are capable of representing an input signal with multiple levels of resolution, and yet maintain the useful compaction properties of the DCT, the subdivision of the input image into smaller sub images is no longer necessary. Wavelet coding has been at the core of the latest image compression standards, most notably JPEG 2000, which is discussed in a separate short article.

Image compression standards

Work on international standards for image compression started in the late 1970s with the CCITT (currently ITU-T) need to standardize binary image compression algorithms for Group 3 facsimile communications. Since then, many other committees and standards have been formed to produce *de jure* standards (such as JPEG), while several commercially successful initiatives have effectively become *de facto* standards (such as GIF). Image compression standards bring about many benefits, such as: (1) easier exchange of image files between different devices and applications; (2) reuse of existing hardware and software for a wider array of products; (3) existence of benchmarks and reference data sets for new and alternative developments.

Binary image compression standards

Work on binary image compression standards was initially motivated by CCITT Group 3 and 4 facsimile standards. The Group 3 standard uses a non-adaptive, 1-D RLE technique in which the last K-1 lines of each group of K lines (for K = 2 or 4) are optionally coded in a 2-D manner, using the *Modified Relative Element Address Designate* (MREAD) algorithm. The Group 4 standard uses only the MREAD coding algorithm. Both classes of algorithms are non-adaptive and were optimized for a set of eight test images, containing a mix of representative documents, which sometimes resulted in data expansion when applied to different types of documents (e.g., half-tone images).. The Joint Bilevel Image Group (JBIG)– a joint committee of the ITU-T and ISO – has addressed these limitations and proposed two new standards

(JBIG and JBIG2) which can be used to compress binary and gray-scale images of up to 6 gray-coded bits/pixel.

Continuous tone still image compression standards

For photograph quality images (both grayscale and color), different standards have been proposed, mostly based on lossy compression techniques. The most popular standard in this category, by far, is the JPEG standard, a lossy, DCT-based coding algorithm. Despite its great popularity and adoption, ranging from digital cameras to the World Wide Web, certain limitations of the original JPEG algorithm have motivated the recent development of two alternative standards, JPEG 2000 and JPEG-LS (lossless). JPEG, JPEG 2000, and JPEG-LS are described in

separate short articles.

Encode each pixel ignoring their inter-pixel dependencies. Among methods are:

1. ***Entropy Coding:*** Every block of an image is entropy encoded based upon the Pk's within a block. This produces variable length code for each block depending on spatial activities within the blocks.
2. ***Run-Length Encoding:*** Scan the image horizontally or vertically and while scanning assign a group of pixel with the same intensity into a pair (gi , li) where gi is the intensity and li is the length of the "run". This method can also be used for detecting edges and boundaries of an object. It is mostly used for images with a small number of gray levels and is not effective for highly textured images.

Example 1: Consider the following 8×8 image.

4	4	4	4	4	4	4	0
4	5	5	5	5	5	4	0
4	5	6	6	6	5	4	0
4	5	6	7	6	5	4	0
4	5	6	6	6	5	4	0
4	5	5	5	5	5	4	0
4	4	4	4	4	4	4	0
4	4	4	4	4	4	4	0

The run-length codes using vertical (continuous top-down) scanning mode are:

(4,9)	(5,5)	(4,3)	(5,1)	(6,3)
(5,1)	(4,3)	(5,1)	(6,1)	(7,1)
(6,1)	(5,1)	(4,3)	(5,1)	(6,3)
(5,1)	(4,3)	(5,5)	(4,10)	(0,8)

i.e. total of 20 pairs = 40 numbers. The horizontal scanning would lead to 34 pairs = 68 numbers, which is more than the actual number of pixels (i.e. 64).

Example 2: Let the transition probabilities for run-length encoding of a binary image (0:black and 1:white) be $p_0 = P(0/1)$ and $p1 = P(1/0)$. Assuming all runs are independent, find (a) average run lengths, (b) entropies of white and black runs, and (c) compression ratio.

Solution:

A run of length $l \geq 1$ can be represented by a Geometric random variable (Grv) X_i with PMF $P(X_i = l) = p_i (1-p_i)^{t-1}$ with i = 0,1 which corresponds to happening of 1^{st} occurrences of 0 or 1 after l independent trials. (Note that (1-P(0/1)) = P(1/1) and (1-P(1/0)) = P(0/0)) and Thus, for the average we have

$$\mu_{X_i} = \sum_{l=1}^{\infty} lP(X_i = l) = \sum_{l=1}^{\infty} lp_i(1-p_i)^{l-1}$$

which using series $\sum_{n=1}^{\infty} na^{n-1} = \frac{1}{(1-a)^2}$ reduces to $\mu_{X_i} = \frac{1}{p_i}$. The entropy is given by

$$\begin{aligned} H_{X_i} &= -\sum_{l=1}^{\infty} P(X_i = l) log_2 P(X_i = l) \\ &= -p_i \sum_{l=1}^{\infty} (1-p_i)^{l-1} [log_2 p_i + (l-1) log_2 (1-p_i)] \end{aligned}$$

Using the same series formula, we get

$$H_{X_i} = -\frac{1}{p_i}[p_i log_2 p_i + (1-p_i) log_2 (1-p_i)]$$

. The achievable compression ratio is

$$C = \frac{H_{X_0} + H_{X_1}}{\mu_{X_0} + \mu_{X_1}} = \frac{H_{X_0} P_0}{\mu_{X_0}} + \frac{H_{X_1} P_1}{\mu_{X_1}}$$

where $P_i = \frac{p_i}{p_0 + p_i}$ are the *a priori* probabilities of black and white pixels.

Huffman Encoding Algorithm*:* It consists of the following steps.

1. Arrange symbols with probability Pk‟s in a decreasing order and consider them as “leaf nodes” of a tree.
2. Merge two nodes with smallest prob to form a new node whose prob is the sum of the two merged nodes. Go to Step 1 and repeat until only two nodes are left (“root nodes”). 3 Arbitrarily assign 1‟s and 0‟s to each pair of branches merging into a node. 4 Read sequentially from root node to the leaf nodes to form the associated code for each symbol.

Example 3: For the same image in the previous example, which requires 3 bits/pixel using standard PCM we can arrange the table on the next page.

Gray levels	# occurrences	P_k	C_k	β_k	$P_k\beta_k$	$-P_k \log_2 P_k$
0	8	0.125	0000	4	0.5	0.375
1	0	0	-	0	-	-
2	0	0	-	0	-	-
3	0	0	-	0	-	-
4	31	0.484	1	1	0.484	0.507
5	16	0.25	01	2	0.5	0.5
6	8	0.125	001	3	0.375	0.375
7	1	0.016	0001	4	0.64	0.095
	64	1			R	H

Codewords C_ks are obtained by constructing the binary tree as in Fig. 5.

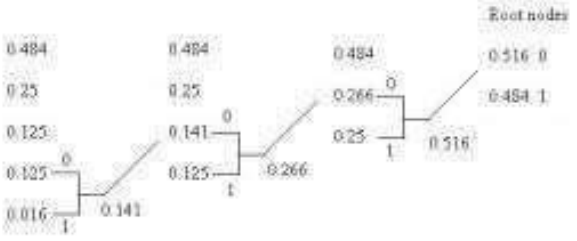


Fig. Tree structure for Huffman Encoding

Note that in this case, we have

$$R = \sum_{k=1}^{8} \beta_k P_k = 1.923 \text{ bits/pixel}$$

$$H = -\sum_{k=1}^{8} P_k \log_2 P_k = 1.852 \text{ bits/pixel}$$

Thus,

$$1.852 \leq R = 1.923 \leq H + \frac{1}{L} = 1.977$$

i.e., an average of 2 bits/pixel (instead of 3 bits/pixel using PCM) can be used to code the image. However, the drawback of the standard Huffman encoding method is that the codes have variable lengths.

PREDICTIVE ENCODING:

Idea: Remove mutual redundancy among successive pixels in a region of support (ROS) or neighborhood and encode only the new information. This mehtod is based upon linear prediction. Let us start with 1-D linear predictors. An N^{th} order linear prediction of x(n) based on N previous samples is generated using a 1-D autoregressive (AR) model.

$$\hat{x}(n) = a_1 x(n-1) + a_2 x(n-2) + \cdots + a_N x(n-N)$$

a_i s are model coefficients determined based on some sample signals. Now instead of encoding x(n) the prediction error.

$$e(n) = x(n) - \hat{x}(n)$$

Is encoded as it requires substantially small number of bits. Then, at the receiver we reconstruct x(n) using the previous encoded values x(n-k) and the encoded error signal, i.e.,

$$x(n) = \hat{x}(n) + e(n)$$

This method is also referred to as differential PCM (DPCM).

Minimum Variance Prediction

The predictor

$$\hat{x}(n) = \sum_{i=1}^{N} a_i x(n-i)$$

is the best N^{th} order linear mean-squared predictor of $x(n)$, which minimizes the MSE

$$\epsilon = E\left[\left(x(n) - \hat{x}(n)\right)^2\right]$$

This minimization wrt a_k's results in the following "orthogonal property"

$$\frac{\partial \epsilon}{\partial a_k} = -2E\left[\left(x(n) - \hat{x}(n)\right) x(n-k)\right] = 0, \qquad 1 \le k \le N$$

which leads to the normal equation

$$r_{xx}(k) - \sum_{i=1}^{N} a_i r_{xx}(k-i) = \sigma_e^2 \delta(k), \qquad 0 \le k \le N$$

where $r_{xx}(k)$ is the autocorrelation of the data $x(n)$ and σ_e^2 is the variance of the driving process $e(n)$.

To understand the need for comapct image representation, consider the amount of data required to represent a **2** hour standard Definition(SD) using 720 x 480 x 24 bit pixel arrays.

A video is a sequence of video frames where each frame is full color still image. Because video player must display the frames sequentially at rates near 30 fps. Standard definition data must be accessed 30fps x (720 x 480) ppf x 3bpp = 31,104,000 bps.

fps: frames per second, ***ppf***: pixels per frame, ***bpp***: bytes per pixel, ***bps***: bytes per second.

Thus a 2 hour movie consists of : = 31,104,000 bps x (60^2) sph x 2hrs

where sph is second per hour = 2.24 x 10^{11} bytes = 224 GB of data.

TWENTY SEVEN 8.5 GB dual layer DVD‟s are needed to store it.

To put 2 hours movie on a single DVD, each frame must be compressed by a factor of around 26.3.

The compression must be even higher for HD, where image resolution reach 1920 x 1080 x 24 bits per image.

Webpage images & High-resolution digital camera photos also are compressed to save storage space & reduce transmission time.

Residential Internet connection delivers data at speeds ranging from 56kbps (conventional phone line) to more than 12 mbps (broadband).

Time required to transmit a small 128 x 128 x 24 bit full color image over this range of speed is from 7.0 to 0.03 sec.

Compression can reduce the transmission time by a factor of around 2 to 10 or more.

Similarly, number of uncompressed full color images that an 8 Megapixel digital camere can store on a 1GB Memory card can be increased.

Data compression: It refers to the process of reducting the amount of data required to represent a given quantity of information.

Data Vs Information:

Data and information are not the same thing; data are the means by which information is conveyed.

Because various amount of data can be used to represent the same amount of information, representations that contain irrelevant or repeated information are said to contain redundant.

In today‟s multimedia wireless communication, major issue is bandwidth needed to satisfy real time transmission of image data. Compression is one of the good solutions to address this issue. Transform based compression algorithms are widely used in the field of compression, because of their de-correlation and other properties, useful in compression. In this paper, comparative study of compression methods is done based on their types. This paper addresses the issue of importance of transform in image compression and selecting particular transform for image compression. A comparative study of performance of a variety of different image transforms is done base on compression ratio, entropy and time factor.

THE FLOW OF IMAGE COMPRESSION CODING:

What is the so-called image compression coding? Image compression coding is to store the image into bit-stream as compact as possible and to display the decoded image in the monitor as exact as possible. Now consider an encoder and a decoder as shown in Fig. 1.3. When the encoder receives the original image file, the image file will be converted into a series of binary data, which is called the bit-stream. The decoder then receives the encoded bit-stream and decodes it to form the decoded image. If the total data quantity of the bit-stream is less than the total data quantity of the original image, then this is called image compression. The full compression flow is as shown in Fig. 1.3.

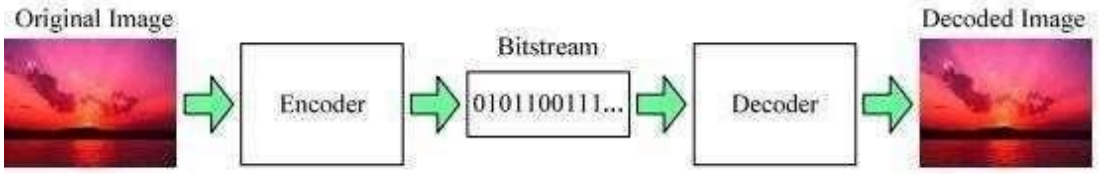


Fig. 1.3 The basic flow of image compression coding

The compression ratio is defined as follows:

$$Cr = \frac{n1}{n2}, \qquad (1.2)$$

where $n1$ is the data rate of original image and $n2$ is that of the encoded bit-stream.

In order to evaluate the performance of the image compression coding, it is necessary to define a measurement that can estimate the difference between the original image and the decoded image. Two common used measurements are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR), which are defined in (1.3) and (1.4), respectively. f(x,y) is the pixel value of the original image, and f'(x,y)is the pixel value of the decoded image. Most image compression systems are designed to minimize the MSE and maximize the PSNR.

$$MSE = \sqrt{\frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}[f(x,y)-f'(x,y)]^2}{WH}} \qquad (1.3)$$

$$PSNR = 20\log_{10}\frac{255}{MSE} \qquad (1.4)$$

The general encoding architecture of image compression system is shown is Fig. 1.4. The fundamental theory and concept of each functional block will be introduced in the following sections.

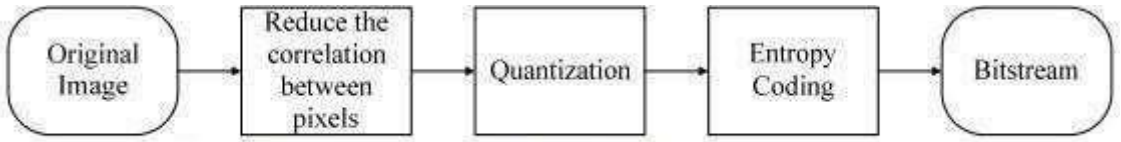


Fig. 1.4 The general encoding flow of image compression

Reduce the Correlation between Pixels

The reason for compression is that the correlation between one pixel and its neighbor pixels is very high, or we can say that the values of one pixel and its adjacent pixels are very similar. Once the correlation between the pixels is reduced, we can take advantage of the statistical characteristics and the variable length coding theory to reduce the storage quantity. This is the most important part of the image compression algorithm; there are a lot of relevant processing methods being proposed. The best-known methods are as follows:

- Predictive Coding: Predictive Coding such as DPCM (Differential Pulse Code Modulation) is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element.
- Orthogonal Transform: Karhunen-Loeve Transform (KLT) and Discrete Cosine Transform (DCT) are the two most well-known orthogonal transforms. The DCT-based image compression standard such as JPEG is a lossy coding method that will result in some loss of details and unrecoverable distortion.
- Sub band Coding: Sub band Coding such as Discrete Wavelet Transform (DWT) is also a lossy coding method. The objective of sub band coding is to divide the spectrum of one image into the lowpass and the highpass components. JPEG 2000 is a 2-dimension DWT based image compression standard.

QUANTIZATION

The objective of quantization is to reduce the precision and to achieve higher

compression ratio. For instance, the original image uses 8 bits to store one element for every pixel; if we use less bits such as 6 bits to save the information of the image, then the storage quantity will be reduced, and the image can be compressed. The shortcoming of quantization is that it is a lossy operation, which will result into loss of precision and unrecoverable distortion. The image compression standards such as JPEG and JPEG 2000 have their own quantization methods, and the details of relevant theory will be introduced in the chapter 2.

ENTROPY CODING

The main objective of entropy coding is to achieve less average length of the image. Entropy coding assigns code words to the corresponding symbols according to the probability of the symbols. In general, the entropy encoders are used to compress the data by replacing symbols represented by equal-length codes with the code words whose length is inverse proportional to corresponding probability. The entropy encoder of JPEG and JPEG 2000 will also be introduced in the chapter 2.

AN OVERVIEW OF IMAGE COMPRESSION STANDARD:

In this chapter, we will introduce the fundamental theory of two well-known image compression standards –JPEG and JPEG 2000.

JPEG – JOINT PICTURE EXPERT GROUP

Fig. 2.1 and 2.2 shows the Encoder and Decoder model of JPEG. We will introduce the operation and fundamental theory of each block in the following sections.

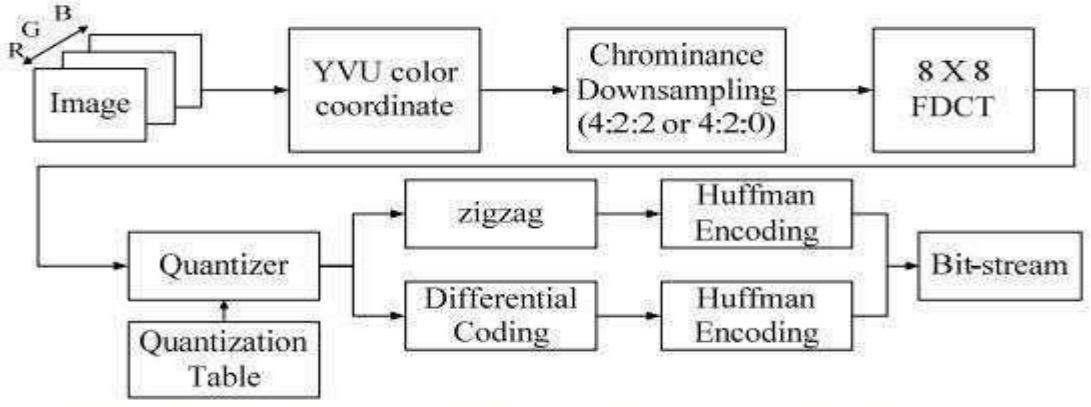


Fig. 2.1 The Encoder model of JPEG compression standard

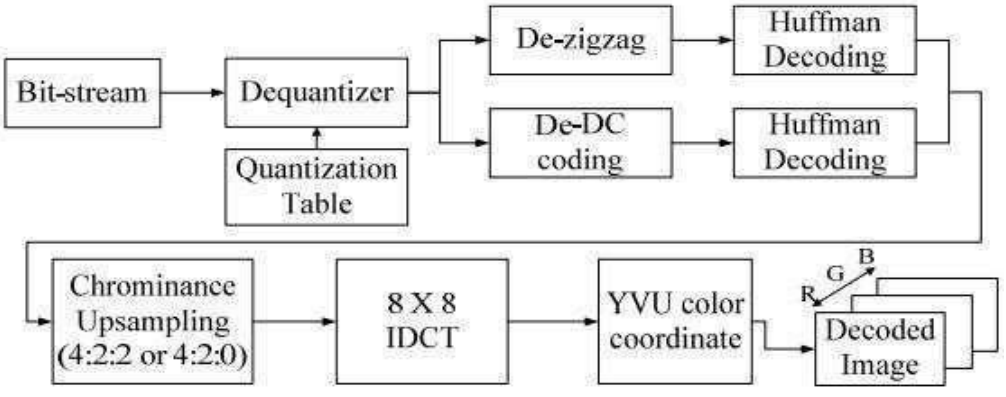


Fig. 2.2 The Decoder model of JPEG compression standard

DISCRETE COSINE TRANSFORM

The next step after color coordinate conversion is to divide the three color components of the image into many 8×8 blocks. The mathematical definition of the Forward DCT and the Inverse DCT are as follows:

Forward DCT

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

$$\text{for } u=0,\ldots,N-1 \text{ and } v=0,\ldots,N-1 \qquad \textbf{(2.1)}$$

$$\text{where } N=8 \text{ and } C(k)=\begin{cases}1/\sqrt{2} & \text{for } k=0\\ 1 & \text{otherwise}\end{cases}$$

Inverse DCT

$$f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \qquad \textbf{(2.2)}$$

$$\text{for } x=0,\ldots,N-1 \text{ and } y=0,\ldots,N-1 \text{ where } N=8$$

The f(x,y) is the value of each pixel in the selected 8×8 block, and the F(u,v) is the DCT coefficient after transformation. The transformation of the 8×8 block is also a 8×8 block composed of F(u,v).

The DCT is closely related to the DFT. Both of them taking a set of points from the spatial domain and transform them into an equivalent representation in the frequency domain. However, why DCT is more appropriate for image compression than DFT? The two main reasons are:

1. The DCT can concentrate the energy of the transformed signal in low frequency, whereas the DFT can not. According to Parseval‟s theorem, theenergy is the same in the spatial domain and in the frequency domain. Because the human eyes are less sensitive to the low frequency component, we can focus on the low frequency component and reduce the contribution of the high frequency component after taking DCT.
2. 2. For image compression, the DCT can reduce the blocking effect than the DFT.

After transformation, the element in the upper most left corresponding to zero frequency in both directions is the “DC coefficient” and the rest are called “AC coefficients.”

Quantization in JPEG:

Quantization is the step where we actually throw away data. The DCT is a lossless procedure. The data can be precisely recovered through the IDCT (this isn‟t entirely true because in reality no physical implementation can compute with perfect accuracy). During

Quantization every coefficients in the 8×8 DCT matrix is divided by a corresponding quantization value. The quantized coefficient is defined in (2.3), and the reverse the process can be achieved by the (2.4).

$$F(u,v)_{Quantization} = round\left(\frac{F(u,v)}{Q(u,v)}\right) \quad (2.3)$$

$$F(u,v)_{deQ} = F(u,v)_{Quantization} \times Q(u,v) \quad (2.4)$$

The goal of quantization is to reduce most of the less important high frequency DCT coefficients to zero, the more zeros we generate the better the image will compress. The matrix Q generally has lower numbers in the upper left direction and large numbers in the lower right direction. Though the high-frequency components are removed, the IDCT still can obtain an approximate matrix which is close to the original 8×8 block matrix. The JPEG committee has recommended certain Q matrix that work well and the performance is close to the optimal condition, the Q matrix for luminance and chrominance components is defined in (2.5) and (2.6)

$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad (2.5)$$

$$Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix} \quad (2.6)$$

ZIGZAG SCAN:

After quantization, the DC coefficient is treated separately from the 63 AC coefficients. The DC coefficient is a measure of the average value of the original 64 image samples. Because there is usually strong correlation between the DC coefficients of adjacent 8×8 blocks, the quantized DC coefficient is encoded as the difference from the DC term of the previous block. This special treatment is worthwhile, as DC coefficients frequently contain a significant fraction of the total image energy. The other 63 entries are the AC

components. They are treated separately from the DC coefficients in the entropy coding process.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

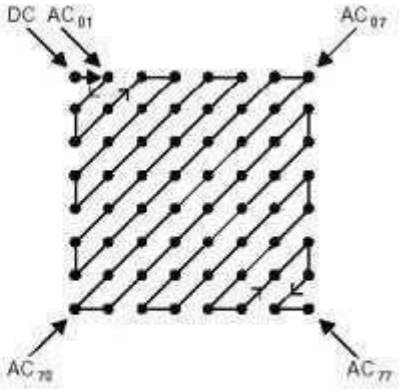


Fig. 2.3 The zigzag scan order

Entropy Coding in JPEG

Differential Coding:

The mathematical representation of the differential coding is:

$$Diff_i = DC_i - DC_{i-1} \qquad (2.7)$$

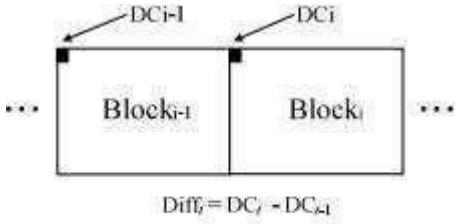


Fig. 2.4 Differential Coding

We set DC0 = 0. DC of the current block DCi will be equal to DCi-1 + Diffi . Therefore, in the JPEG file, the first coefficient is actually the difference of DCs. Then the difference is encoded with Huffman coding algorithm together with the encoding of AC coefficients.

Different types of redundancies in digital image

(i) Redundancy can be broadly classified into Statistical redundancy and Psycho visual redundancy.

(ii) Statistical redundancy can be classified into inter-pixel redundancy and coding redundancy.

(iii) Inter-pixel can be further classified into spatial redundancy and temporal redundancy.

(iv) Spatial redundancy or correlation between neighboring pixel values.

(v) Spectral redundancy or correlation between different color planes or spectral bands.

(vi) Temporal redundancy or correlation between adjacent frames in a sequence of images in video applications.

(vii) Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible.

(viii) In digital image compression, three basic data redundancies can be identified and exploited: Coding redundancy, Inter-pixel redundancy and Psychovisual redundancy.

- **Coding Redundancy:**
    - Coding redundancy is associated with the representation of information.
    - The information is represented in the form of codes.
    - If the gray levels of an image are coded in a way that uses more code symbols than absolutely necessary to represent each gray level then the resulting image is said to contain coding redundancy.
- **Inter-pixel Spatial Redundancy:**
    - Interpixel redundancy is due to the correlation between the neighboring pixels in an image.
    - That means neighboring pixels are not statistically independent. The gray levels are not equally probable.
    - The value of any given pixel can be predicated from the value of its neighbors that is they are highly correlated.
    - The information carried by individual pixel is relatively small. To reduce the interpixel redundancy the difference between adjacent pixels can be used to represent an image.
- **Inter-pixel Temporal Redundancy:**
    - Interpixel temporal redundancy is the statistical correlation between pixels from successive frames in video sequence.
    - Temporal redundancy is also called interframe redundancy. Temporal redundancy can be exploited using motion compensated predictive coding.
    - Removing a large amount of redundancy leads to efficient video compression.
- **Psychovisual Redundancy:**
    - The Psychovisual redundancies exist because human perception does not

involve quantitative analysis of every pixel or luminance value in the image.

- It‟s elimination is real visual information is possible only because the information itself is not essential for normal visual processing.

UNIT IV
BASIC STEPS OF VIDEO PROCESSING

Analog Video

We used to live in a world of analog images and video, where we dealt withphoto graphic film, analog TV sets, videocassette recorders (VCRs), and camcorders.

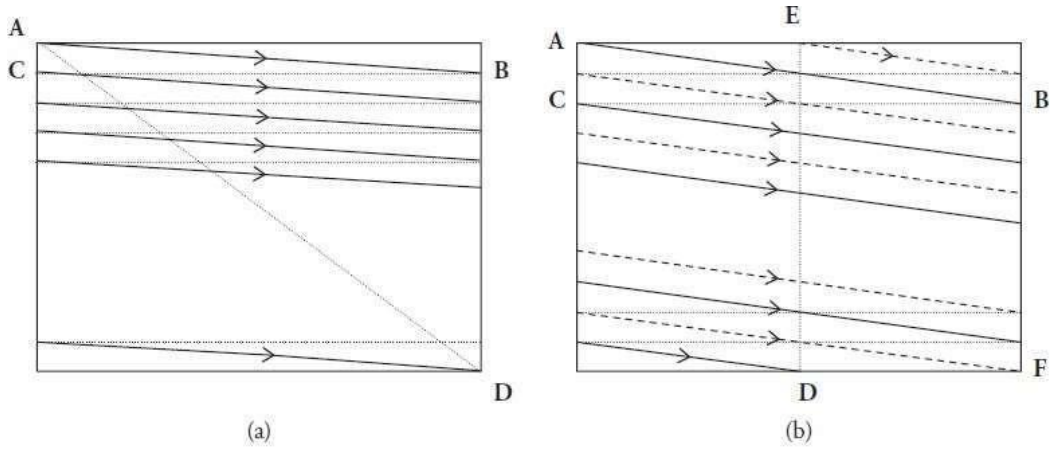
For video distribution, we relied on analog TV broadcasts and analog cable TV, which transmitted predetermined programming at a fixed rate. Analog video, due to its nature, provided a very limited amount of interactivity, e.g., only channel selection on the TV and fast-forward search and slow- motion replay on the VCR. Additionally, we had to live with the NTSC/PAL/SECAM analog signal formats with their well-known artifacts and very low still-frame image quality. In order to display NTSC signals on computer monitors or European TV sets, we needed expensive transcoders. In order to display a smaller version of the NTSC picture in a corner of the monitor, we first had to digitize the whole picture and then digitally reduce its size. Searching a video archive for particular footage required tedious visual scanning of a whole bunch of videotapes. Motion pictures were recorded on photographic film, which is a high-resolution analog medium, or on laser discs as analog signals using optical technology. Manipulation of analog video is not an easy task, since it requires digitization of the analog signal into digital form first.

Today almost all video capture, processing, transmission, storage, and search are in digital form. In this section, we describe the nature of the analog-video signal because an understanding of history of video and the limitations of analog video formats is important. For example, interlaced scanning originates from the history of analog video. We note that video digitized from analog sources is limited by the resolution and the artifacts of the respective analog signal.

Progressive vs. Interlaced Scanning

The analog-video signal refers to a one-dimensional (1D) signal $s(t)$ of time that is obtained by sampling $sc(x1, x2, t)$ in the vertical $x2$ and temporal coordinates. This conversion of 3D spatiotemporal signal into a 1D temporal signal by periodic vertical-temporal sampling is called scanning. The signal $s(t)$, then, captures the time-varying image intensity $sc(x1, x2, t)$ only along the scan lines. It also contains the timing information and blanking signals needed to align pictures.

The most commonly used scanning methods are progressive scanning and interlaced scanning. Progressive scan traces a complete picture, called a frame, at every D*t* sec. The spot flies back from B to C, called the horizontal retrace, and from D to A, called the vertical



Scanning raster: (a) progressive scan; (b) interlaced scan.

retrace, as shown in Figure, theuses progressive scanning with Δt=1/72 sec for monitors. On the other hand, the TV industry uses 2:1 interlaced scan where the odd-numbered and even numbered lines, called the odd field and the even field, respectively, are traced in turn.

Analog-Video Signal Formats

Some important parameters of the video signal are the vertical resolution, aspect ratio, and frame/field rate. The vertical resolution is related to the number of scan lines per frame. The aspect ratio is the ratio of the width to the height of a frame. The human eye does not perceive flicker if the refresh rate of the display is more than 50 Hz. However, for analog TV systems, such a high frame rate, while preserving the vertical resolution, requires a large transmission bandwidth. Thus, it was determined that analog TV systems should use interlaced scanning, which trades vertical resolution to reduced flickering within a fixed bandwidth. An example analog-video signal $s(t)$. Blanking pulses (black) are inserted during the retrace intervals to blank out retrace lines on the monitor. Sync pulses are added on top of the blanking pulses to synchronize the receiver's horizontal and vertical sweep circuits. The sync pulses ensure thatthe picture starts at the top-left corner of the receiving monitor. The timing of the sync pulses is, of course, different for progressive and interlaced video. Several analog-video signal standards, which are obsolete today, have different image parameters (e.g., spatial and temporal resolution) and differ in the way they handle color. These can be grouped as: i)component analog video; ii) composite video; and iii) S-video (Y/C video).

Component analog video refers to individual red (R), green (G), and blue (B) video signals

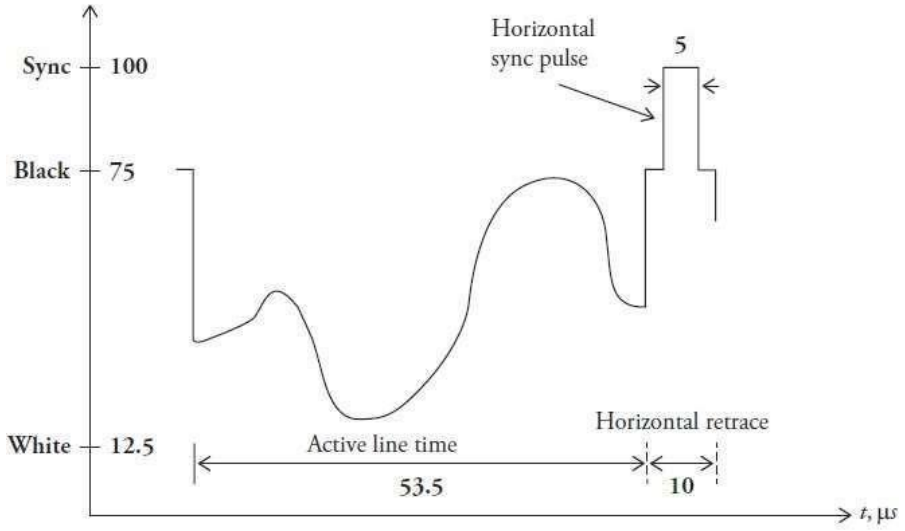


Figure 2.6 Analog-video signal for one full line.

Composite-video format encodes the chrominance components on top of the luminance signal for distribution as a single signal that has the same bandwidth as the luminance signal. Different composite video formats,

e.g., NTSC (National Television Systems Committee), PAL (Phase Alternation Line), and SECAM (System Electronique Color Avec Memoire), have been used in different regions of the world. The composite signal usually results in errors in color rendition, known as hue and saturation errors, because of inaccuracies in the separation of the color signals. S-video is a compromise between the composite video and component video, where we represent the video with two component signals,

a luminance and a composite chrominance signal. The chrominance signals have been based on (I,Q) or (U,V) representation for NTSC, PAL, or SECAM systems. S-video was used in consumer-quality videocassette recorders and analog camcorders to obtain image quality better than that of composite video. Cameras specifically designed for analog television pickup from motion picture film were called telecine cameras. They employed frame-rate conversion from 24 frames/sec to 60 fields/sec.

Analog-to-Digital Conversion

The analog-to-digital (A/D) conversion process consists of pre-filtering (for antialiasing), sampling, and quantization of component (R, G, B) signal or composite signal. The ITU (International Telecommunications Union) and SMPTE (Society of Motion Picture and Television Engineers) have standardized sampling parameters for both component and composite video to enable easy exchange of digital video across different platforms. For A/D conversion of *component signals*, the horizontal sampling rate of 13.5 MHz for the luma component and 6.75 MHz for two chromatic components were chosen, because they satisfy the following requirements:

1. Minimum sampling frequency (Nyquist rate) should be 4.2 3 2 5 8.4 MHz for 525/30

NTSC luma and 5 3 2 5 10 MHz for 625/50 PAL luma signals.

2. Sampling rate should be an integral multiple of the line rate, so samples in successive lines are correctly aligned (on top of each other).

3. For sampling *component signals*, there should be a single rate for 525/30 and 625/50 systems; i.e, the sampling rate should be an integral multiple of line rates (lines/sec) of both 29.97 3 525 5 15,734and 25 3 625 5 15,625.

For sampling the *composite signal*, the sampling frequency must be an integral multiple of the sub- carrier frequency to simplify composite signal to RGB decoding of sampled signal. It is possible to operate at 3 or 4 times the subcarrier frequency, although most systems choose to employ 4 3 3.58 5
14.32 MHz for NTSC and 4 3 4.43 5 17.72 MHz for PAL signals, respectively.

Digital Video

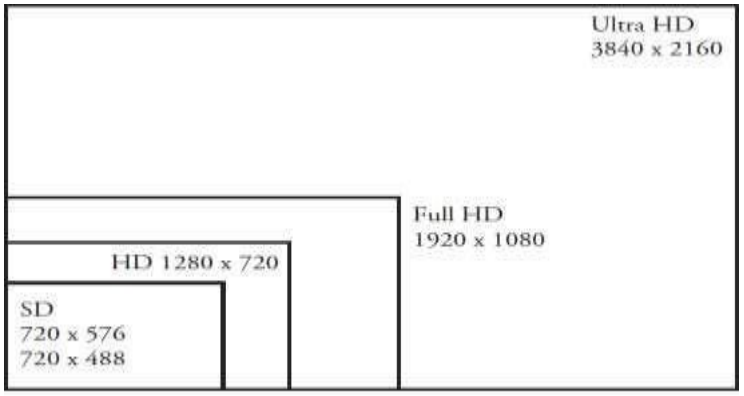
We have experienced a digital media revolution in the last couple of decades. TV and cinema have gone all-digital and high-definition, and most movies and some TV broadcasts are now in 3D format. High-definition digital video has landed on laptops, tablets, and cellular phones with high-quality media streaming over the Internet. Apart from the more robust form of the digital signal, the main advantage of digital representation and transmission is that they make it easier to provide a diverse range of services over the same network. Digital video brings broadcasting, cinema, computers, and communications industries together in a truly revolutionary manner, where telephone, cable TV, and Internet service providers have become fierce competitors. A single device can serve as a personal computer, a high-definition TV, and a videophone. We can now capture live video on a mobile device, apply digital processing on a laptop or tablet, and/or print still frames at a local printer. Other applications of digital video include medical imaging, surveillance for military and law enforcement, and intelligent highway systems.

Spatial Resolution and Frame Rate

Digital-video systems use component color representation. Digital color cameras provide individual RGB component outputs. Component color video avoids the artifacts that result from analog composite encoding. In digital video, there is no need for blanking or sync pulses, since it is clear where a new line starts given the number of pixels per line. The horizontal and vertical resolution of digital video is related to the pixel sampling density, i.e., the number of pixels per unit distance. The number of pixels per line and the number of lines per frame is used to classify video as standard, high, or ultra-high definition,. In low-resolution digital video, pixelation (aliasing) artifact arises due to lack of sufficient spatial

resolution. It manifests itself as jagged edges resulting from individual pixels becoming visible. The visibility of pixelation artifacts varies with the size of the display and the viewing distance. This is quite different from analog video where the lack of spatial- resolution results in blurring of image in the respective direction. The frame/field rate is typically 50/60 Hz, although some displays use frame interpolation to display at 100/120, 200 or even 400 Hz.

The notation 50i (or 60i) indicates interlaced video with 50 (60) fields/sec, which corresponds to 25 (30) pictures/sec obtained by weaving the two fields together. On the other hand, 50p (60p) denotes50 (60) full progressive frames/sec.The arrangement of pixels and lines in a contiguous region of the memory is called a bitmap. There are five key parameters of a bitmap: the starting address in the memory, the number of pixels per line, the pitch value, the number of lines and the number of bits per pixel. The pitch value specifies the distance in memory from the start of one line to the next. The most common use of pitch different the number of pixels per line is to set pitch to the next highest power of 2, which may help certain applications run faster. Also, when dealing with interlaced inputs, setting the pitch to double the number of pixels per line facilitates writing lines from each field alternately in memory. This will form a "weaved frame" in a contiguous region of the memory.



Digital-video spatial-resolution formats.

Color, Dynamic Range, and Bit-Depth

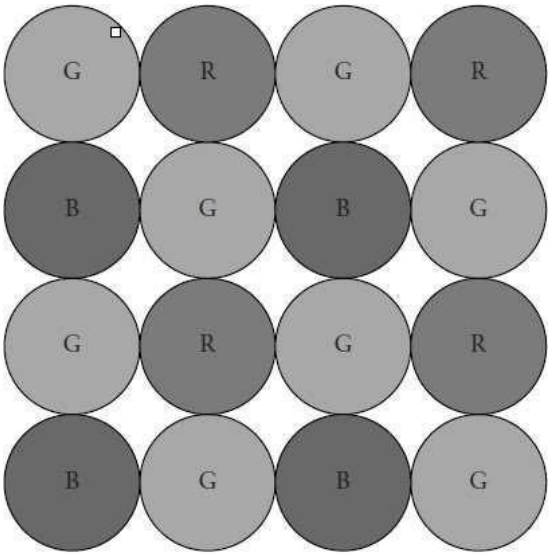
This section addresses color representation, dynamic range, and bit-depth in digital images/video.

Color Capture and Display

Color cameras can be the three-sensor type or single-sensor type. Three-sensor cameras capture R, G, and B components using different CCD panels, using an optical beam splitter; however, they may suffer from synchronicity problems and high cost, while single-sensor cameras often have to compromise spatial resolution. This isbecause a color filter array is used so that each CCD element captures one of R, G, or B pixels in some periodic pattern. A commonly used color filter pattern is the Bayer array, shown in Figure 2.8, where two out of every four pixels are green, one is red, and one is blue, since green signal contributes the most

to the luminance channel.

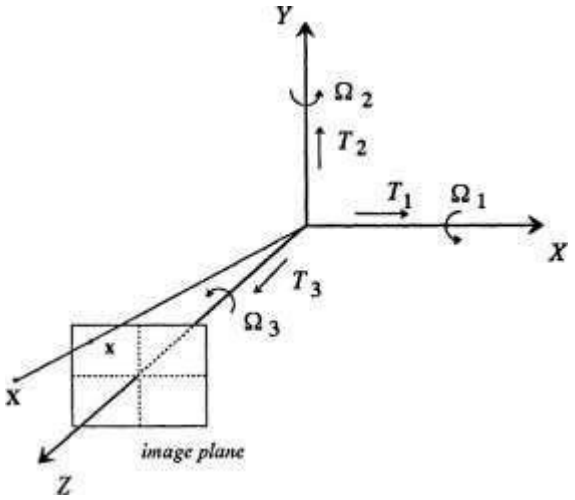
The missing pixel values in each color channel are computed by linear or adaptive



Bayer color-filter array pattern.

Time-Varying Image Formation

Definition of the quantity to be measured, the *2-d motion field.* It then discusses the forms of spatiotemporal intensity structure that result from a relatively simple scene model incorporating smooth surfaces with both diffuse and specular components of reflection. The model serves to emphasize the common occurrences of local geometric deformation, smooth contrast variations, and multiple image velocities in a local image neighborhood. This is an important departure from other approaches, most of which are based on a model of image translation and the assumption of smooth and unique 2-d velocity fields.



2-D Motion Field

motion analysis is to detect and measure the variations in the projected image locations of physical surface points as the camera and/or objects in the scene move. When an object moves with respect to the camera, a point on the surface of the object can be thought to trace out a 3-d particle path as a function of time Its derivative yields the instantaneous 3-d velocity of the point. The projection of the point and its path onto the image plane yields a 2-d

Geometric Image Formation

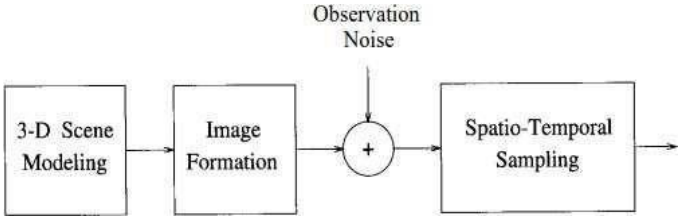
Imaging systems capture 2-D projections of a time-varying 3-D scene. This projection can be represented by a mapping from a 4-D space to a 3-D space,

$$\begin{aligned} f: \quad & R^4 \rightarrow R^3 \\ & (X_1, X_2, X_3, t) \rightarrow (x_1, x_2, t) \end{aligned}$$

where (Xi,X2,X3), the 3-D world coordinates, (X1, X2), the 2-D image plane coordinates, and t, time, are continuous variables, Here, we consider two types of projection, perspective (central) and orthographic (parallel), which are described in the following.

Perspective Projection

Perspective projection reflects image formation using an ideal pinhole camera according to the principles of geometrical optics. Thus, all the rays from the object pass through the

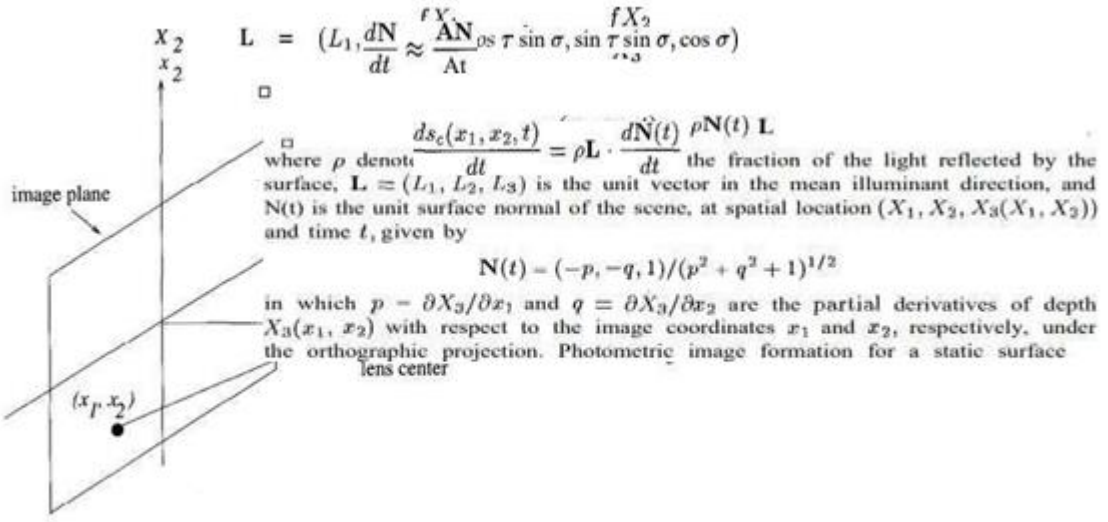


center of projection, which corresponds to the center of the lens. For this reason, it is also known as “central projection.” Perspective projection is illustrated in Figure 2.4 when the center of projection is between the object and the image plane, and the image plane coincides with the (Xi, X2) plane of the world coordinate system. . The algebraic relations that describe the perspective transformation for the configuration shown in Figure 2.4 can be obtained based on similar triangles formed by drawing perpendicular lines from the object point (Xi, X2, Xs) and the image point (X1, x2,0) to the X3 axis, respectively. This leads to

$$\frac{x_1}{f} = -\frac{X_1}{X_3 - f} \quad \text{and} \quad \frac{x_2}{f} = -\frac{X_2}{X_3 - f}$$

$$x_1 = \frac{fX_1}{f - X_3} \quad \text{and} \quad x_2 = \frac{fX_2}{f - X_3}$$

where f denotes the distance from the center of projection to the image plane. If we move the center of projection to coincide with the origin of the world coordinates, a simple change of variables yields the following equivalent expressions:



$$\mathbf{L} = (L_1, \frac{d\mathbf{N}}{dt} \approx \frac{\mathbf{\Delta N}}{\mathrm{At}} \cos\tau \sin\sigma, \sin\tau \sin\sigma, \cos\sigma)$$

$$\frac{ds_c(x_1, x_2, t)}{dt} = \rho \mathbf{L} \cdot \frac{d\mathbf{N}(t)}{dt} \quad \rho \mathbf{N}(t)\ \mathbf{L}$$

where ρ denote the fraction of the light reflected by the surface, $\mathbf{L} = (L_1, L_2, L_3)$ is the unit vector in the mean illuminant direction, and N(t) is the unit surface normal of the scene, at spatial location $(X_1, X_2, X_3(X_1, X_2))$ and time t, given by

$$\mathbf{N}(t) = (-p, -q, 1)/(p^2 + q^2 + 1)^{1/2}$$

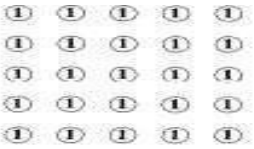
in which $p = \partial X_3/\partial x_1$ and $q = \partial X_3/\partial x_2$ are the partial derivatives of depth $X_3(x_1, x_2)$ with respect to the image coordinates x_1 and x_2, respectively, under the orthographic projection. Photometric image formation for a static surface

Perspective projection model.

SAMPLING THE VIDEO SIGNAL

Digital video can be obtained by sampling analog video in the horizontal direction along the scan lines, or by applying an inherently 3-D sampling structure to sample the time-varying image, as in the case of some solid-state sensors. Examples of the most popular 3-D sampling structures

In these figures, each circle indicates a pixel location, and the number inside the circle indicates the time of sampling. The first three sampling structures are lattices, whereas the sampling structure is not a lattice, but a union of two cosets of a lattice. The vector c in shows the displacement of one coset with respect to the other. Other 3-D sampling structures can be found. The theory of sampling on lattices and other special M- D structures. It will be seen that the most suitable sampling structure for a time varying image depends on its spatio-temporal frequency content. The sampling structures shown here are field- or frame-instantaneous; that is, a complete field or frame is acquired at one time instant. An alternative strategy is time-sequential sampling, where individual samples are taken oneat a time according to a prescribed ordering which is repeated after one complete frame

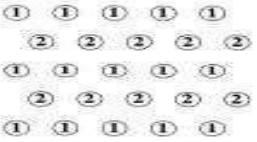


$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & 0 & 0 \\ 0 & \Delta x_2 & 0 \\ 0 & 0 & \Delta t \end{bmatrix}$$

Orthogonal sampling lattice [Dub 85] (©1985 IEEE).

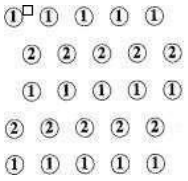
$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & 0 & 0 \\ 0 & 2\Delta x_2 & \Delta x_2 \\ 0 & 0 & \Delta t/2 \end{bmatrix}$$

Vertically aligned 2:1 line-interlaced lattice [Dub 85] (©1985 I



$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & 0 & \Delta x_1/2 \\ 0 & 2\Delta x_2 & \Delta x_2 \\ 0 & 0 & \Delta t/2 \end{bmatrix}$$

Field-quincunx sampling lattice [Dub 85] (©1985 IEEE).

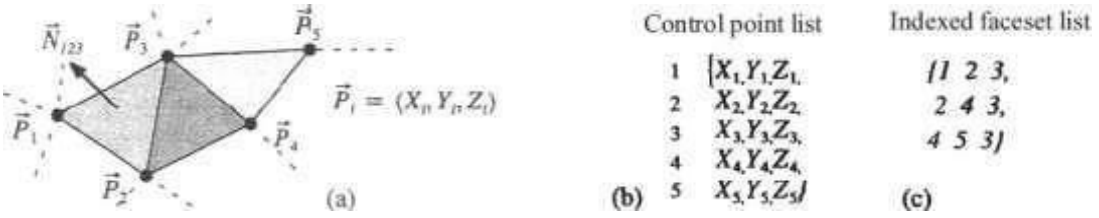


$$\mathbf{V} = \begin{bmatrix} \Delta x_1 & \Delta x_1/2 & 0 \\ 0 & 2\Delta x_2 & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ \Delta x_2 \\ \Delta t/2 \end{bmatrix}$$

Unit V
2-D Motion Estimation

Motion Model

An object can be rigid or flexible. The motion of a rigid object can be described in terms of a translation vector T = (Tx; Ty; Tz) T and a rotation matrix [R]. Let us first describe how a point of an object movesin3Dspace using the motion parameters T and [R]. The translation vector T describes a displacement of a point from X to X0 by Tx; Ty; Tz in the directions of the coordinate



Wireframe for object shape description:

(a) mesh of triangles to represent the object surface

(b) representation of the mesh of triangles as a list of control points and a list of triangles that references the list of control points axes X; Y;Z, respectively

$$X^1 = X + T$$

If an object is translated, holds for all object points. If an object rotates around the origin of the3Dspace, we describe the motion of its points with the rotation matrix [R]:

$$[R] = [Rz] \cdot [Ry] \cdot [Rx]$$

This rotation matrix rotates a point around the axes X; Y and Z in this order. It is computed

$$[\mathbf{R}_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix},$$

$$[\mathbf{R}_y] = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix},$$

$$[\mathbf{R}_z] = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

from the rotation matrices that rotate a point just around one axis. These individual rotation matrices are

where θ_x, θ_y, and θ_z are the rotation angles w.r.t each axis. Finally, we have $[\mathbf{R}]$ as a function of the rotation angles θ_x, θ_y and θ_z

$$[\mathbf{R}] = \begin{bmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z - \cos\theta_x \sin\theta_z & \cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z \\ \cos\theta_y \sin\theta_z & \sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix}.$$

The rotation matrix [R] is an orthonormal matrix, i.e. it satisfy

$$[R]^T = [R]^{-1}$$

$$\det[R] = 1$$

$$\det[R] = -1$$

Now, we can express the motion of a point X on the object surface from X to X^1 as

$$X^1 = [R] \cdot X + T$$

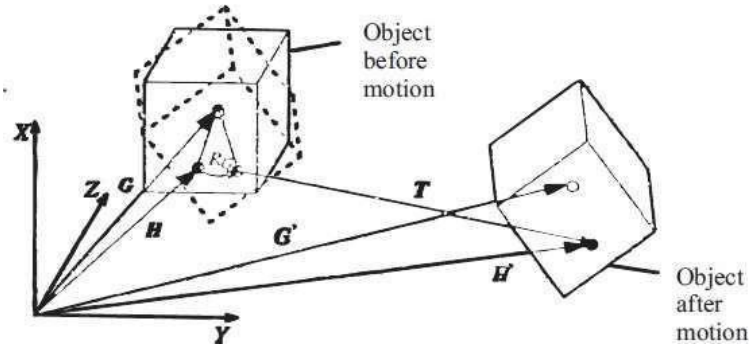
For many motion estimation algorithms, the non-linear rotation matrix according has to be linearized with respect to the rotation angles. Assuming rotation angles are small so that

$$[\mathbf{R}] \approx [\mathbf{R}'] = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}.$$

rotates the point X on the object surface around the center of the world coordinate system. In the case that the object is far away from the center of the coordinate system and the object is just rotating around its own center, a large translation vector is required in this motion representation. This can be avoided by defining a local coordinate system for each object and by defining rotation and translation of the object with respect to its own center C = (Cx; Cy; Cz)T, which would also be the center of the local coordinate system

$$\mathbf{X}' = [\mathbf{R}] \cdot (\mathbf{X} - \mathbf{C}) + \mathbf{C} + \mathbf{T}.$$

Assuming C = 0, the rotation matrices are identical whereas the translation vectors differ for identical physical motion.



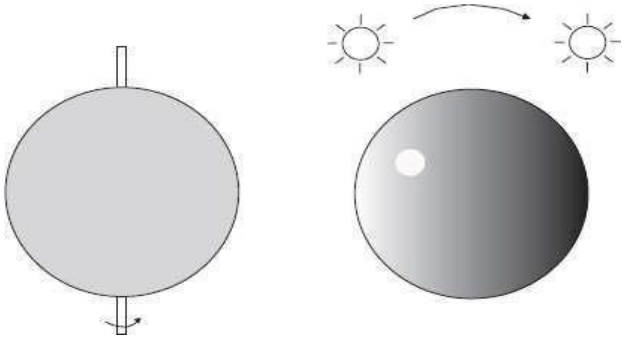
TWO-DIMENSIONAL MOTION ESTIMATION

In the second example, the sphere is stationary, but is illuminated by a point light source thatis rotating around the sphere. The motion of the light source causes the movement of the refection light spot on the sphere, which in turn can make the eye believe the sphere is rotating. The observed or apparent 2D motion is referred to as optical flow in computer vision literature.

Optical Flow Equation and Ambiguity in Motion Estimation

Consider a video sequence whose luminance variation is represented by (x; y; t):1 Suppose an imaged point (x; y) at time t is moved to (x+dx ; y+dy) at time t+dt.

Under the constant intensity assumption**.**



The optical flow is not always the same as the true motion field.

In (a), a sphere is rotating under a constant ambient illumination, but the observed image does not change.

In (b), a point light source is rotating around a stationary sphere, causing the highlight point on the sphere to rotate

Using Taylor's expansion, when d_x, d_y, d_t are small, we have

$$\psi(x+d_x, y+d_y, t+d_t) = \psi(x,y,t) + \frac{\partial \psi}{\partial x}d_x + \frac{\partial \psi}{\partial y}d_y + \frac{\partial \psi}{\partial t}d_t.$$

images of the same object point at different times have the same luminance value

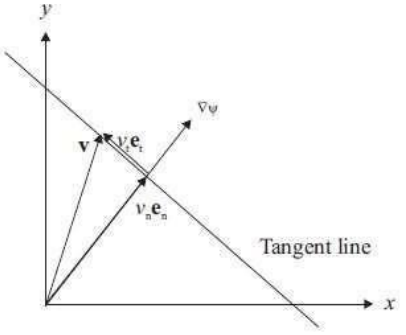
Ψ(x + dx; y + dy; t + dt) = Ψ (x; y; t)

$$\frac{\partial \psi}{\partial x}d_x + \frac{\partial \psi}{\partial y}d_y + \frac{\partial \psi}{\partial t}d_t = 0.$$

The above equation is written in terms of the motion vector (dx; dy). Dividing both sides by dt yields

$$\frac{\partial \psi}{\partial x}v_x + \frac{\partial \psi}{\partial y}v_y + \frac{\partial \psi}{\partial t} = 0 \quad \text{or} \quad \nabla\psi^T \mathbf{v} + \frac{\partial \psi}{\partial t} = 0.$$

where (v_x, v_y) represents the velocity vector, $\nabla\psi = \left[\frac{\partial \psi}{\partial x}, \frac{\partial \psi}{\partial y}\right]^T$ is the spatial gradient vector of $\psi(x,y,t)$. In arriving at the above equation, we have assumed that d_t is small, so that $v_x = d_x/d_t, v_y = d_y/d_t$. The above equation is commonly known as the *optical flow equation*.[2] The conditions for this relation to hold is the same as that for the constant intensity assumption, and have been discussed previously in



Decomposition of motion $\mathbf{v}$ into normal $(v_n\mathbf{e}_n)$ and tangent $v_t\mathbf{e}_t$ components.
Given $\nabla\psi$ and $\frac{\partial \psi}{\partial t}$, any MV on the tangent line satisfies the optical flow equation.

the flow vector v at any point x can be decomposed into two orthogonal components as

v =v n e n+ vt et

where en is the direction vector of the image gradient r , to be called the normal direction, and

$$v_n \|\nabla\psi\| + \frac{\partial \psi}{\partial t} = 0,$$

e_t is orthogonal to e_n, to be called the tangent direction. The optical flow equation can be written as

where $\|\nabla T\|$ is the magnitude of the gradient vector.

At any pixel x, one cannot determine the motion vector v based on $\|\nabla T\|$ and $\frac{\partial \psi}{\partial t}$ alone. There is only one equation for two unknowns (v_x and v_y, or v_n and v_t). In fact, the underdetermined component is v_t: To solve both unknowns, one needs to impose additional constraints. The most common constraint is that the flow vectors should vary smoothly spatially, so that one can make use of the intensity variation over a small neighborhood surrounding x to estimate the motion at x

Given $\|\nabla T\|$ and $\frac{\partial \psi}{\partial t}$ the projection of the motion vector along the normal direction is fixed, with $v_n = \frac{\partial \psi}{\partial t} / \|\nabla T\|$; whereas the projection onto the tangent direction, v_t, is undetermined. Any value of v_t would satisfy the optical flow equation. this means that any point on the tangent line will satisfy the optical flow equation. This ambiguity in estimating the motion vector is known as the aperture problem.

Pixel-Based Motion Estimation

In pixel-based motion estimation, one tries to estimate a motion vector for every pixel. Obviously, this problem is ill-defined. If one uses the constant intensity assumption, for every pixel in the anchor frame, there are many pixels in the tracked frame that have exactly the same image intensity. If one uses the optical flow equation instead, the problem is again indeterminate, because there is only one equation for two unknowns. To circumvent this problem, there are in general four approaches. First, one can use the regularization technique to enforce some smoothness constraints on the motion

Regularization Using Motion Smoothness Constraint

Horn and Schunck proposed to estimate the motion vectors by minimizing the following objective function, which is a combination of the flow-based criterion and a motion smoothness criterion:

$$E\left(\mathbf{v}(\mathbf{x})\right) = \sum_{\mathbf{x}\in\Lambda} \left(\frac{\partial\psi}{\partial x}v_x + \frac{\partial\psi}{\partial y}v_y + \frac{\partial\psi}{\partial t}\right)^2 + w_s\left(\|\nabla v_x\|^2 + \|\nabla v_y\|^2\right).$$

In their original algorithm, the spatial gradient of v_x and v_y are approximated by $\nabla v_x = [v_x(x,y) - v_x(x-1,y), v_x(x,y) - v_x(x,y-1)]^T$, $\nabla v_y = [v_y(x,y) - v_y(x-1,y), v_y(x,y) - v_y(x,y-1)]^T$. The minimization of the above error function is accomplished by a gradient-based method known as Gauss-Siedel method.

Using a Multipoint Neighborhood

the motion vectors of all the pixels in a neighborhood $B(x_n)$ surrounding it are the same, being dn. To determine dn, one can either minimize the prediction error over $B(x_n)$, or solve the optical flow equation using a least squares method. Here we present the first approach. To estimate the motion vector dn for x_n, we minimize the DFD error over $B(x_n)$:

$$E_n(\mathbf{d}_n) = \frac{1}{2}\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left(\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})\right)^2,$$

where w(x) are the weights assigned to pixel x. Usually, the weight decreases as the distance from x to x_n increases. The gradient with respect to d_n is

$$E_n(\mathbf{d}_n) = \frac{1}{2}\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left(\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})\right)^2,$$

where $w(\mathbf{x})$ are the weights assigned to pixel $\mathbf{x}$. Usually, the weight decreases as the distance from $\mathbf{x}$ to $\mathbf{x}_n$ increases.

The gradient with respect to $\mathbf{d}_n$ is

$$\mathbf{g}_n = \frac{\partial E_n}{\partial \mathbf{d}_n} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x}) e(\mathbf{x},\mathbf{d}_n)\left.\frac{\partial\psi_2}{\partial\mathbf{x}}\right|_{\mathbf{x}+\mathbf{d}_n},$$

where $e(\mathbf{x},\mathbf{d}_n) = \psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x})$ is the DFD at $\mathbf{x}$ with the estimate $\mathbf{d}_n$. Let $\mathbf{d}_n^l$ represent the estimate at the l-th iteration, the first order gradient descent method would yield the following update algorithm

$$\mathbf{d}_n^{l+1} = \mathbf{d}_n^l - \alpha\mathbf{g}_n(\mathbf{d}_n^l).$$

From Eq. (6.3.3), the update at each iteration depends on the sum of the image gradients at various pixels scaled by the weighted DFD values at those pixels.

One can also derive an iterative algorithm using the Newton-Ralphson method. From Eq. (6.3.3), the Hessian matrix is

$$\mathbf{H}_n = \frac{\partial^2 E_n}{\partial \mathbf{d}_n^2} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left.\frac{\partial\psi_2}{\partial\mathbf{x}}\frac{\partial\psi_2}{\partial\mathbf{x}}^T\right|_{\mathbf{x}+\mathbf{d}_n} + w(\mathbf{x})e(\mathbf{x},\mathbf{d}_n)\left.\frac{\partial^2\psi_2}{\partial\mathbf{x}^2}\right|_{\mathbf{x}+\mathbf{d}_n}$$

$$\approx \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\left.\frac{\partial\psi_2}{\partial\mathbf{x}}\frac{\partial\psi_2}{\partial\mathbf{x}}^T\right|_{\mathbf{x}+\mathbf{d}_n}.$$

The Newton-Ralphson update algorithm is then (See Appendix B):

$$\mathbf{d}_n^{l+1} = \mathbf{d}_n^l - \alpha\mathbf{H}(\mathbf{d}_n^l)^{-1}\mathbf{g}_n(\mathbf{d}_n^l).$$

This algorithm converges faster than the first order gradient descent method, but it requires more computation in each iteration. Instead of using gradient-based update algorithms, one can also use exhaustive search to nd the dn that yields the minimal error within a denied search range.

This will lead to the exhaustive block matching algorithm (EBMA)

Region-Based Motion Estimation

By region-based motion estimation, we mean to segment the underlying image frame into multiple regions and estimate the motion parameters of each region. The segmentation should be such that a single parametric motion model can represent well the motion in each region. Obviously, region segmentation is dependent on the motion model used for characterizing each region. The simplest approach is to require each region undergo the same translational motion

Motion-Based Region Segmentation

As described already, motion-based segmentation refers to the partitioning of amotion eld into multiple regions so that the motion within each region can be described by a single set of motion parameters. Here we present two approaches for accomplishing this task. The first approach uses a clustering technique to identify similar motion vectors. The second approach uses a layered approach to estimate the underlying region and associated motions sequentially, starting from the region with the most dominant motion.

Clustering

Consider the case when the motion model for each region is a pure translation. Then the segmentation task is to group all spatially connected pixels with similar motion vectors into one region. This can be easily accomplished by an automated clustering method, such as the K-means or the ISODATA method

This is an iterative process: Starting from an initial partitioning, the mean motion vector, known as the centroid, of each region is calculated. Then each pixel is reclassed into the region whose centroid is closest to the motion vector of this pixel. This results in a new partition and the above two steps can be repeated until the partition does not change any more. In this process, the spatial connectivity is not considered. Therefore, the resulting regions may contain pixels that are not spatially connected. Postprocessing steps may be applied at the end of the iterations to improve the spatial connectivity of the resulting regions. For example, a single region may be split into several sub-regions so that each region is a spatially connected subset, isolated pixels may be merged into its surrounding region, and regionally region boundaries can be smoothed using morphological operators. When the motion model for each region is not a simple translation, motion-based clustering is not as straight forward. This is because one cannot use the similarity between motion vectors as the criterion for performing clustering. One way is to and a set of motion parameters for each

pixel by fitting the motion vectors in its neighborhood into a specied model. Then one can employ the clustering method described previously, by replacing the raw motion vector with the motion parameter vector. If the original motion field is given in the block-based representation using a higher order model, then one can cluster blocks with similar motion parameters into the same region. Similarly, with the mesh-based motion representation, one can derive a set of motion parameters for each element based on its nodal displacements, and then clustering the elements with similar parameters into the same region

Layered Approach Very often, the motion field in a scene can be decomposed into layers, with the first layer representing the most dominant motion, the second layer

the less dominant one, and so on. Here, the dominance of a motion is determined by the area of the region undergoing the corresponding motion. The most dominant motion is often a rejection of the camera motion, which acts the entire imaged domain. For example, in a video clip of a tennis play, the background will be the first layer, which usually undergoes a coherent global camera motion; the player the second layer (which usually contains several sub-object level motions corresponding

to the movements of different parts of the body), the racket the third, and the ball the fourth layer. To extract the motion parameters in different layers, one can use the robust estimator method recursively. First, we try to model the motion field of the entire frame by a single setof parameters, and continuously eliminate outlier pixels from the remaining inlier group, until all the pixels within the inlier group can be modeled well. This will yield the first dominant region (corresponding to the inlier region) and its associated motion.

The same approach can then be applied to the remaining pixels (the outlier region) to identify the next dominant region and its motion. This process continues until no more outlier pixels exist.

As before, postprocessing may be invoked at the end of the iterations to improve the spatial connectivity of the resulting regions. This is the sequential approach described. For the above scheme to work well, the inlier region must be larger than the outlier region at any iteration. This means that the largest region must be greater than the combined area of all other regions, the second largest region must be greater than the combined area of the remaining regions, and so on. This condition is satisfied in most video scenes, which often contain a stationary background that covers a large portion of the underlying image, and different moving objects with varying sizes.

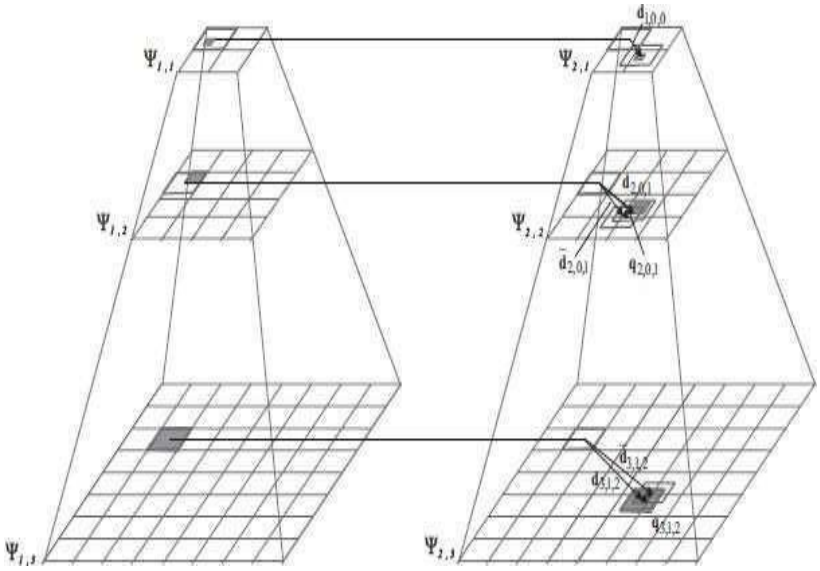
Multi-Resolution Motion Estimation

various motion estimation approaches can be reduced to solving an error minimization problem.Two major diffulties associated with obtaining the correct solution are:

i) The minimization function usually has many local minima and it is not easy to

reach theglobal minimum, unless it is close to the chosen initial solution; and

ii) The amount of computation involved in the minimization process is very high. Both problems can be combated by the multiresolution approach, which searches the solution for an optimization problem in successively



The multi-resolution approach for motion estimation in a general setting, which is applicable to anymotion.

Relation between Image Intensity and Motion

Almost all motion estimation algorithms are based on the constant intensity assumption the optical derived based on this assumption. This enables us to estimate motion by identifying pixels with similar intensity, subject to some motion models. Note that this assumption is valid only when the illumination source is ambient and temporally invariant, and that the object surface is diffusely.

When the motion direction is orthogonal to image intensity gradient, or if the image gradient is zero, motion does not induce changes in image intensity. This is the inherent limit of intensity-based motion estimation methods.

Key Components in Motion Estimation

Motion representation: This depends on the partition used to divide a frame (pixel-based, block-based, mesh-based, region-based, global), the motion model used for each region of the partition (block, mesh-element, object-region, or entire frame), and the constraint between motions in adjacent regions. Motion estimation criterion: We presented three criteria for estimating the motion parameters over each region:

i) minimizing DFD (when the motion is small, this is equivalent to the method based on the optical-equation)

ii) making the resulting motion field as smooth as possible across regions, while minimizing

DFD;

iii) maximizing the a posterior probability of the motion field given the observed frames.

iii) essentially requires

i) and ii). Instead of minimizing DFD, one can also detect peaks in the PCF, when the motion in a region is a pure translation.

Optimization methods:

For chosen representation and criterion, the motion estimation problem is usually converted to an optimization (minimization or maximization) problem, which can be accomplished by using exhaustive search or gradient based search. To speech up the search and avoid being trapped in local minima, a multi-resolution procedure can be used.

Application of Motion Estimation in Video Coding

Motion estimation is a key element in any video coding system., an effective video coding method is to use block-wise temporal prediction, by which a block in a frame to be coded is predicted from its corresponding block in a previously coded frame, then the prediction error is coded. To minimize the bit rate for coding the prediction error, the appropriate criterion for motion estimation is to minimize the prediction error. The fact that the estimated motion field does not necessarily resemble the actual motion eld is not problematic in such applications. Therefore, block matching algorithms (EBMA and its fast variant including HBMA) offer simple and effective solutions.

using the MV estimated for each block directly for the prediction of that block, one can use a weighted average of the predicted values based on the MV's estimated for its neighboring blocks. This is known as overlapping block motion compensation, Note that in the above video coding method, the motion vectors also need to be coded, in addition to the prediction error. Therefore, minimizing the prediction error alone is not the best criterion to use. Since a smoother motion field requires fewer bits to code, imposing smoothness in the estimated motion field, if done properly, can help improve the overall coding efficiency. More advanced motion estimation algorithms therefore operate by minimizing the total bit rate used for coding the MVs and the prediction errors

To overcome the blocking artifacts produced by block-based motion estimation methods, high-order block-based (DBMA), mesh-based or a combination of block based, mesh-based, and/or DBMA can be applied. However, these more complicated schemes usually do not lead to Signiant gain in the coding efficiency.

In more advanced video coding schemes global motion estimation is usually applied to the entire frame, prior to block-based motion estimation, to compensate the effect of camera motion.

Moreover, an entire frame is usually segmented into several regions or objects, and the motion parameters for each region or object are estimated using the global motion estimation method. By rest searching the solution in a coarse resolution, one can usually obtain a solution that is close to the true motion. In addition, by limiting the search in each resolution to a small neighborhood of the solution obtained in the previous resolution, the total number of searches can be reduced, compared to that required by directly searching in the nest resolution over a large.