[After reading notes & watching video lectures Learn these MQ]

INTRODUCTION TO IMAGE PROCESSING 1

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***Define sampling theorem.*** *[R.T.U. 2011]*

Ans. Sampling Theorem

A band limited image f(x, y) sampled uniformly on a rectangular grid with spacing Δx, Δy can be recovered without error from the sample values $f(m\Delta x, n\Delta y)$ provided the sampling rate is greater than the Nyquist rate, that is

$$\frac{1}{\Delta x} = \xi_{xs} > 2\xi_{x0},$$

$$\frac{1}{\Delta y} = \xi_{ys} > 2\xi_{y0}$$

Moreover, the reconstructed image is given by the interpolation formula

$$f(x,y) = \sum\sum_{m,n=-\infty}^{\infty} f'(m\Delta x, n\Delta y)\left(\frac{\sin(x\xi_{xs} - m)\pi}{(x\xi_{xs} - m)\pi}\right)\left(\frac{\sin(y\xi_{ys} - n)\pi}{(y\xi_{ys} - n)\pi}\right)$$

Q.2 ***Define Image sampling.***

Ans. Digitization of spatial coordinates (x, y) is called Image Sampling. To be suitable for computer processing, an image function f(x, y) must be digitized both spatially and in magnitude.

Q.3 ***List the steps involved in digital image processing.***

Ans. The steps involved in digital image processing are :

(i) Image Acquisition
(ii) Preprocessing
(iii) Segmentation
(iv) Representation and description
(v) Recognition and interpretation.

Q.4 ***How cones and rods are distributed in retina?***

Ans. In each eye, cones are in the range 6-7 million, and rods are in the range 75-150 million.

Q.5 ***Define Image.***

Ans. An image may be defined as a two dimensional function f(x, y) where x & y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, discrete quantities we call the image as Digital Image.

PART-B

Q.6 ***Explain different types of mathematical tools which are used in digital image processing.*** **[R.T.U. 2019]**

Ans. Mathematical Tools used in Digital Image Processing

1. Array Versus Matrix Operations – Array operations in images are carried out on a pixel – by – pixel basis. For example, raising an image to a power means that each

Q.8 *Explain the physiology of human eye with the help of a neat figure.* **[R.T.U. 2018]**

Ans. Physiology of Human Eye : The complex structure of the human eye creates an interesting relationship with light.

The human eye comprises several functional elements

- The *cornea* is the 'window' of the eye and helps to focus light.
- The *pupil* (or iris diaphragm) determines how much light enters the eye - it constricts on exposure to bright light.
- The *lens* provides a focus adjustment capability.
- The *retina* is a light sensing region at the back of the eye.

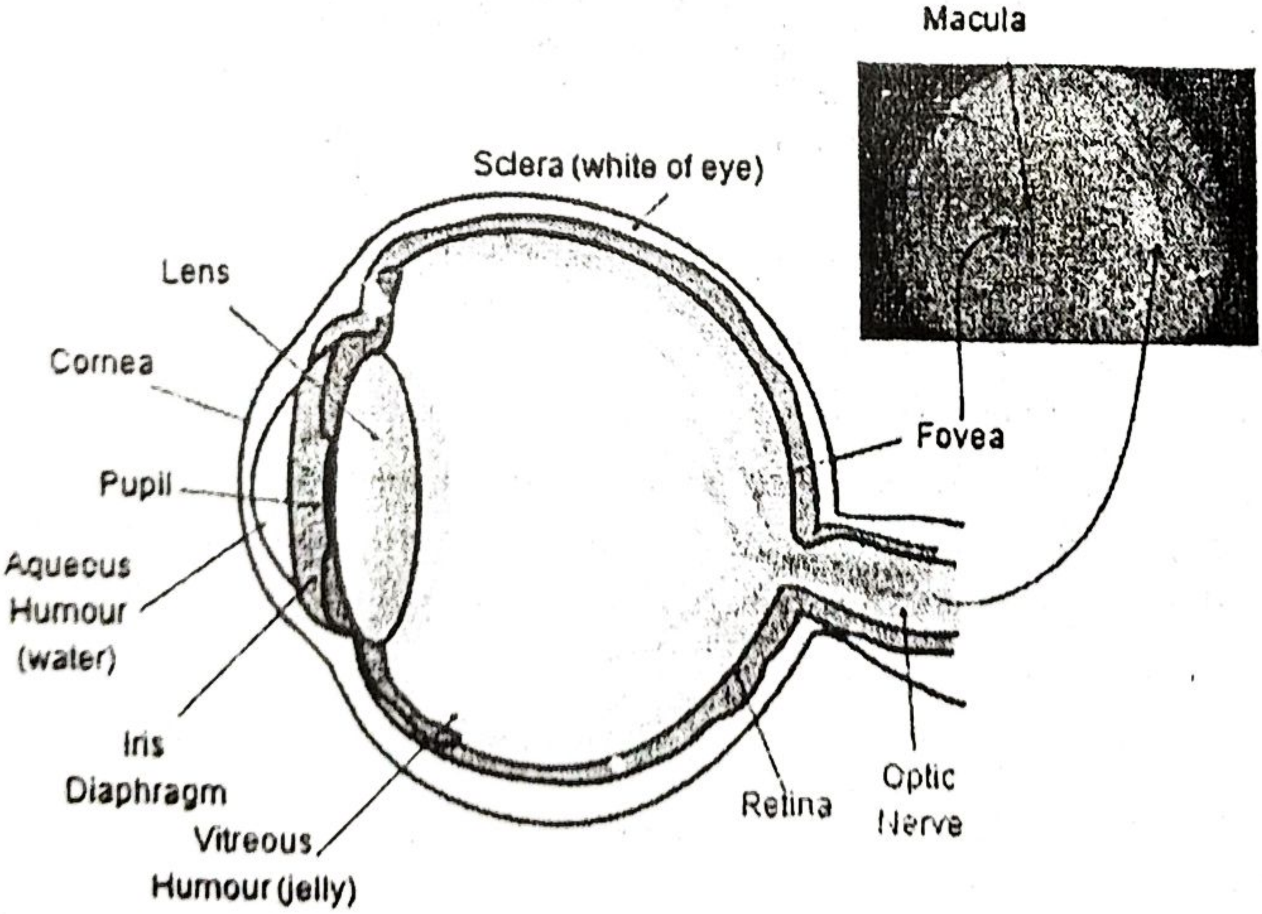


Fig. The Human Eye (Side Schematic)

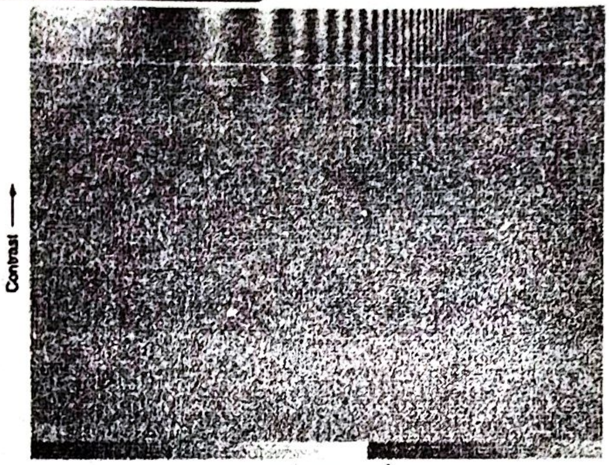


Fig. 3 : MTF measurements of the human visual system by modulated sine-wave grating.

Q.10 *Explain color vision model with example.* [R.T.U. 2017]

OR

Write short note on Color Vision Model. [R.T.U. 2018]

Ans. Color Vision Model : Color vision model (also known as color model or color space on color system) is the specification of colors in some standard and accepted way. It is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.

The classification of color models can be done by

(i) Hardware –oriented models.

(ii) Color description-oriented models.

The hardware-oriented models most commonly used are the RGB (Red, Green, Blue) model for color monitors and a broad class of color video cameras; the CMY (Cyan, Magenta, Yellow) and CMYK (Cyan, Magenta, Yellow and Black) models for color printing. On the other hand, most widely used description oriented models are HSI (Hue, Saturation and Intensity) model, which describes and interpret different properties of colors. The HSI model also helps decoupling the color and gray scale information of an image.

RGB model is the most widely used color model in hardware application. In this model, each color in an image appear in its primary color components-Red, Green, Blue. The model follows cartesian coordinate system.

The above shown figure is the subspace of RGB model. The primary values are positioned at the three axial corners of the cube, and the secondary color components are at the other three corners of the cube. Black is at the origin and white is at the farthest corner from black; and the line joining black and white corner is known as the gray-scale. The different colors in this models are pointed on or inside this cube, and the value of a particular point is determined by the vector distance from the origin (Black). The general assumption is that all the color values are normalized, so the cube shown in the above figure is a unit cube, which signifies, all the color values will all in the range of [0,1].

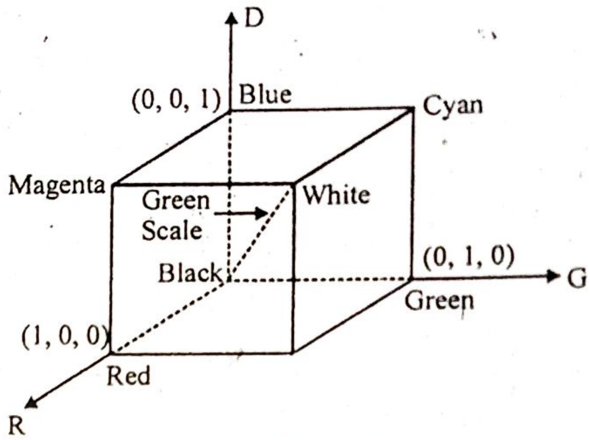


Fig. : RGB Model

In practical application each pixel of a color image has a triplet value of R, G and B, which in combination, produces the composite color image. So for example, if each of the color component (Red, Green and Blue component) image is an 8-bit image, then the depth of an RGB is 24bit. This is known as full color image depth.

Q.11 *Explain in brief about Photometry and Mach band effect.* [R.T.U. 2018]

Ans. Photometry : Photometry is the science of the measurement of light, in terms of its *perceived* brightness to the human eye.

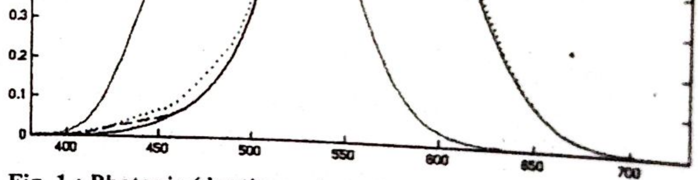


Fig. 1 : Photopic (daytime-adapted, black curve) and scotopic

It is distinct from radiometry, which is the science of measurement of radiant energy (including light) in terms

Q.16 *What are the applications of image processing? Explain components of image processing.*

[R.T.U. 2019]

OR

Explain the applications of digital image processing. [R.T.U. 2017]

Ans. Applications of Digital Image Processing : Some of the major fields in which digital image processing is widely used are mentioned below:

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

Image Sharpening and Restoration : Image sharpening and restoration refers here to process images that have been captured from the modern camera to make them a better image or to manipulate those images in way to achieve desired result. It refers to do what photoshop usually does.

This includes zooming, blurring, sharpening, gray scale to color conversion, detecting edges and vice versa, image retrieval and image recognition.

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

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

UV Imaging : In the field of remote sensing , the area of the earth is scanned by a satellite or from a very high ground and then it is analyzed to obtain information about it. One particular application of digital image processing in the field of remote sensing is to detect infrastructure damages caused by an earthquake.

As it takes longer time to grasp damage, even if serious damages are focused on. Since the area effected by the earthquake is sometimes so wide , that it not possible to examine it with human eye in order to estimate damages. Even if it is , then it is very hectic and time consuming procedure. So a solution to this is found in digital image processing. An image of the effected area is captured from the above ground and then it is analyzed to detect the various types of damage done by the earthquake.

The key steps include in the analysis are :

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

Transmission and Encoding : The very first image that has been transmitted over the wire was from London to New York via a submarine cable.

The picture that was sent took three hours to reach from one place to another.

Now just imagine, that today we are able to see live video feed, or live cctv footage from one continent to another with just a delay of seconds. It means that a lot of work has been done in this field too. This field does not only focus on transmission, but also on encoding. Many different formats have been developed for high or low bandwidth to encode photos and then stream it over the internet or etc.

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

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

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

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

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

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

Components of an Image Processing System : This section briefly outlines the capabilities of modern image processing systems. A general purpose image acquisition and processing system typically consists of four essential components:

1. An image acquisition system. In the simplest case, this could be a CCD camera, a flatbed scanner or a video recorder.
2. A device known as a frame grabber to convert the electrical signal (normally an analog video signal) of the image acquisition system into a digital image that can be stored.
3. A personal computer or a workstation that provides the processing power.
4. Image processing software that provides the tools to manipulate and analyze the images.

Q.17 *Differentiate image quantization and scalar quantization.* [R.T.U. 2019]

OR

What is image quantization? Explain the scalar and image quantization in detail. [R.T.U. 2016, 2011]

Ans. Image Quantization : The step subsequent to sampling in image digitization is quantization.

A quantizer maps a continuous variable u into a discrete variable u', which takes values from a finite set $\{r_1, ..., r_L\}$ of numbers. This mapping is generally a staircase function (Fig.) and the quantization rule is as follows: Define $\{t_k, k = 1,..., L + 1\}$ as a set of increasing transition or decision levels with t_1 and t_{L+1} as the minimum and maximum values, respectively, of u. If u lies in interval (t_k, t_{k+1}), then it is mapped to r_k, the k^{th} reconstruction level.

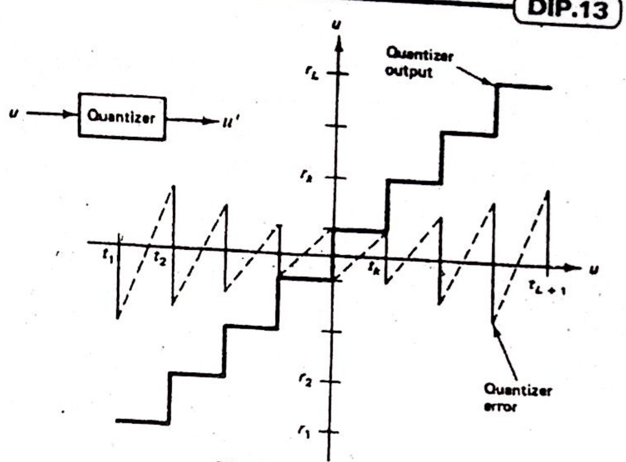


Fig. 1 : A quantizer

Zero memory quantizers are useful in image coding techniques such as pulse code modulation (PCM), differential PCM, transform coding, and so on. Note that the quantizer mapping is irreversible; that is, for a given quantizer output, the input value cannot be determined uniquely. Hence, a quantizer introduces distortion, which any reasonable design method must attempt to minimize. There are several quantizer designs available that offer various trade-offs between simplicity and performance.

Scalar Quantization:

In scalar quantization, the quantization output is the result of division of the input data by a quantization parameter, with rounding towards the nearest integer. If X is an input sample and Q is a quantization parameter, the quantized output is

$$X_Q = \text{Round}\left(\frac{X}{Q}\right)$$

There are different types of scalar quantization techniques-uniform quantization, non-uniform quantization, and adaptive quantization.

Uniform Quantization : Consider X_{max} is the maximum value from an input source and the input values are uniformly distributed in the range $[-X_{max}, +X_{max}]$. We can design an N-level uniform scalar quantizer by dividing the interval $[-X_{max}, +X_{max}]$ into N equally sized sub-intervals. The length Δ of each sub-interval is called the step size of the uniform quantizer which is given by

$$\Delta = \frac{2X_{max}}{N}$$

Characteristics of an 8-level uniform scalar quantizer, is shown in Fig. 2. Horizontal axis represents the input and the vertical axis represents the corresponding value after quantization and inverse quantization. Hence any value in between (2Δ, 3Δ), say as an example, will be approximated to 2.5Δ.

Uniform scalar quantization is very simple and straightforward for implementation. It is designed based on assumption that the input source is uniformly distributed. But often probability of distribution of the source symbols is not uniform in nature and the uniform scalar quantization results in poor reconstructed quality. As a result, there is a necessity to design non-uniform quantizers for these types of sources.

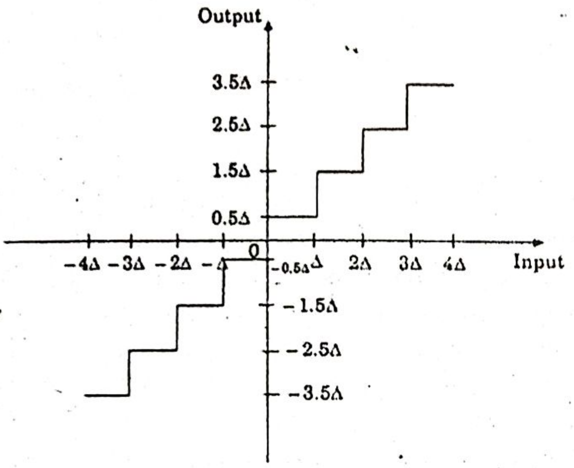


Fig. 2 : Uniform scalar quantizer with step size Δ

Non-uniform Quantization : In order to minimize the average distortion in the reconstructed image because of the quantization, we can lightly quantize the transform coefficients or prediction error values in the region of high importance and heavily quantize the corresponding coefficients in a less important region in the image. One way to achieve this is to use non-uniform quantization such that the quantization steps are smaller for the samples those have more concentration in the curve of probability distribution of the samples. For example, distribution of the prediction error values is more concentrated at the origin of the curve compared to prediction error values further from the origin. As a result, we can use smaller quantization steps for the prediction error values around the origin compared to the other prediction error values. If the interval $[-X_{max} + X_{max}]$ in the quantizer is non-uniformly divided so that the relation between input and output from the quantizer can match to any desired linear function, then it is called the non-uniform quantizer. The characteristic of a general non-uniform quantizer is shown in Fig. 3, as an example, where the quantization step points $(..., X_1, X_2, ..., X_{N-1}, X_N)$ and the output levels $(Y_1, Y_2, ..., Y_N)$ of an N-level quantizer are fixed and they can be selected to minimize some function of the quantization error when the probability distribution of the input is known.

Adaptive quantization. When the statistics of the source symbols considerably changes in the process, the fixed and predefined uniform or non-uniform quantizers fails to yield good results. In this situation, the quantizers need to be adaptive with the changes of the source statistics. There are two classes of adaptive quantizers forward adaptive quantizers and backward adaptive quantizers. In a forward adaptive quantizer, the quantizer extracts the quantization step size from the input. In this approach the source data is divided into blocks. Each block is independently analyzed and the quantization step size is determined based on its statistical distribution. In a backward adaptive quantizer, the quantization step size is determined based on the previously reconstructed output signals from the quantizer. Both the forward and backward adaptive quantizers have their own advantages and disadvantages.

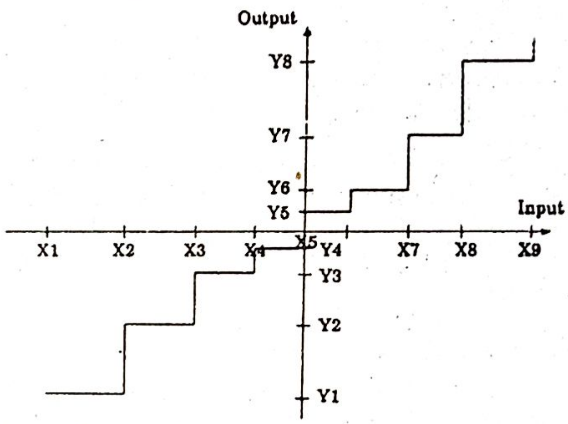


Fig. 3 : Characteristic of a non-uniform scalar quantizer

Q.18 ***What is Digital Image Processing? Give fundamental steps in DIP. Explain each block.*** **[R.T.U. 2010]**

OR

Define the image. Explain the steps of digital image processing with suitable diagram. **[R.T.U. 2017]**

OR

Give fundamental steps in Digital Image Processing. Explain each block. **[R.T.U. 2013]**

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

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

There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum; low, mid and high-level processes. Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. Mid-level processing on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours and the identity of individual objects). Finally, higher-level processing involves "making sense" of an ensemble of recognized objects, as in image analysis and at the far end of the continuum, performing the cognitive functions normally associated with vision.

Fundamental Steps in Digital Image Processing

(i) The acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.

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

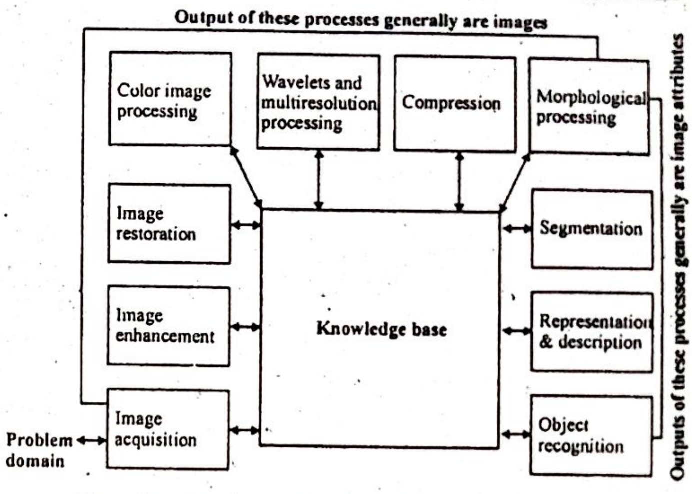


Fig. : Fundamental steps in digital image processing

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

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

(v) Wavelets are the foundation for representing images in various degrees of resolution.

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

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

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

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

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

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter-related 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 applications. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules. This distinction is made in fig. by the use of double-headed arrows between the processing modules and the knowledge base, as opposed to single-headed arrows linking the processing modules.

Both sides of this equation are equivalent ways of expressing a digital image quantitatively. The right side is a matrix of real numbers. Each element of this matrix is called an *image element*, *picture element*, *pixel*, or *pel*. The terms *image* and *pixel* are used throughout the book to denote a digital image and its elements.

In some discussions it is advantageous to use a more traditional matrix notation to denote a digital image and its elements :

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad ....(2)$$

Clearly, $a_{ij} = f(x = i, y = j) = f(i, j)$, so Eqs. (1) and (2) are identical matrices. We can even represent an image as a vector, v. For example, a column vector of size MN × 1 is formed by letting the first M elements of v be the first column of A, the next M elements be the second column, and so on. Alternatively, we can use the rows instead of the columns of A to form such a vector. Either representation is valid, as long as we are consistent.

Image Sampling and Quantization

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

Basic Concepts in Sampling and Quantization

The basic idea behind sampling and quantization is illustrated in fig.2(a) shows a continuous image, $f(x, y)$, that we want to convert to digital form. An image may be continuous with respect to the x-and y-co-ordinates and also in amplitude. To convert it to digital form, we have to sample the functions in both co-ordinates and in amplitude. Digitizing the co-ordinate values is called *sampling*. Digitizing the amplitude values is called *quantization*.

The one-dimensional function shown in fig. 2(b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in fig. 2(a). The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in fig. 2(c). The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of fig. 2(c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in fig. 2(d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

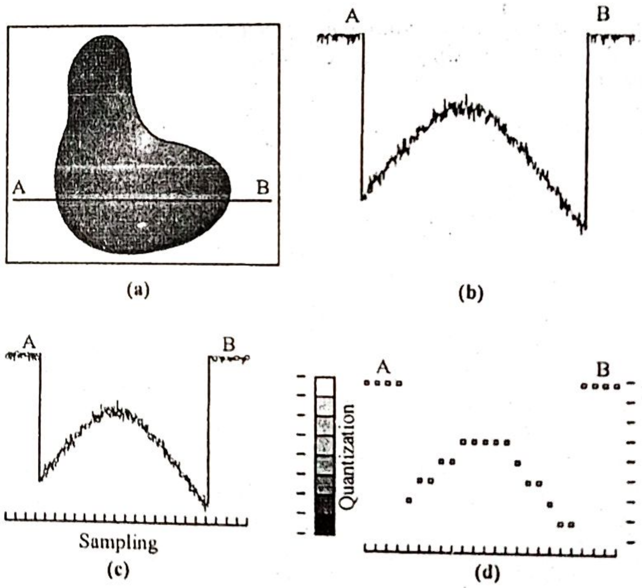


Fig. 2 : Generating a digital image (a) Continuous image, (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization, (c) Sampling and quantization, (d) Digital scan line

Sampling in the manner just described assumes that we have a continuous image in both co-ordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle, there is almost no limits are established by imperfections in the optics used to focus on the sensor an illumination spot that is inconsistent with the fine resolution achievable with mechanical displacements.

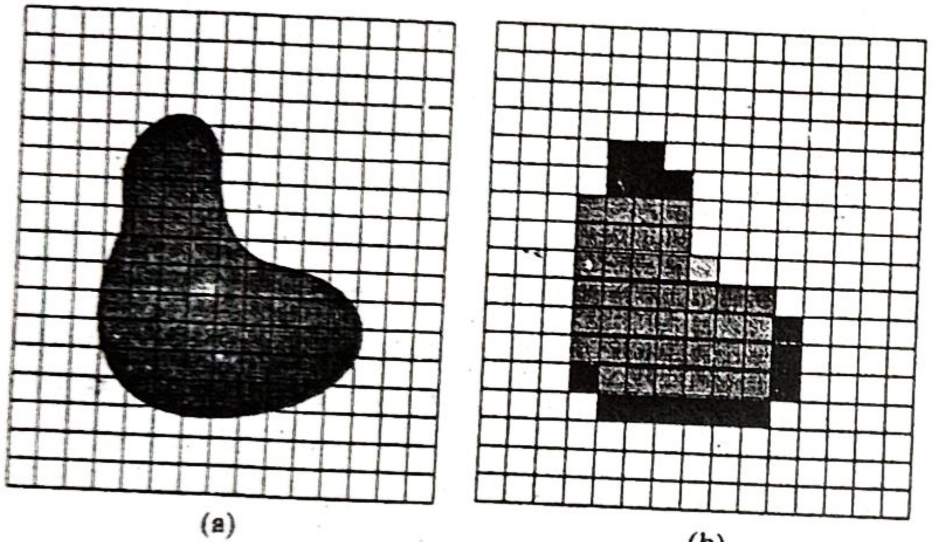


Fig. 3 : (a) Continuous image projected onto a sensor array, (b) Result of image sampling and quantization

When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor output completes the process of generating a digital image.

When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Quantization of the sensor outputs is as before. Figure 3 illustrates this concept. Fig. 3(a) shows a continuous image projected onto the plane of an array sensor. Fig. 3(b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization.

❑❑❑

TRANSFORMATION AND FILTERING 2

ARS QUESTIONS

Forward Transform : The sequence of x(n) is given by $x(n) = \{x_0, x_1, x_2 ... x_{N-1}\}$.

$$X(k) = \sum_{n=0}^{N-1} x(n)\ e^{-j2*pi*nk/N}; k = 0,1,2,....N-1$$

Reverse Transforms

$$X(n) = \sum_{k=0}^{N-1} x(k)\ e^{-j2*pi*nk/N}; n = 0,1,2,....N-1$$

Q.5 *Define Histogram.*

Ans. The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $h(r_k) = n_k$, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k.

PART-B

Q.6 *Describe the basic principles of image enhancement by*
(a) Spatial domain methods
(b) Frequency domain methods

[R.T.U. 2019, 2016]

Ans.(a) Image Enhancement by Spatial Domain Methods : 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

$$g(x, y) = T[f(x, y)] \quad ...(1)$$

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). In addition, T can operate on a set of input images, such as performing the pixel-by-pixel sum of K images for noise reduction.

The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular subimage area centered at (x, y), as Fig. 1 shows.

The center of the subimage is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood. Although other neighborhood shapes, such as approximations to a circle, sometimes are used, square and rectangular arrays are by far the most predominant because of their ease of implementation.

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

$$s = T(r) \quad ...(2)$$

where, for simplicity in notation, r and s are variables denoting, respectively, the gray level of f(x, y) and g(x, y) at any point (x, y). For example, if T(r) has the form shown in Fig. 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.

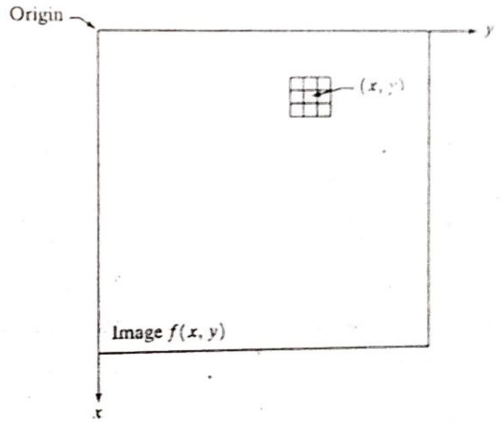


Fig. 1 : neighborhood about a point (x, y) in an image.

The opposite effect takes place for values of r above m. In the limiting case shown in Fig. 2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function. Some fairly simple, yet powerful, processing approaches can be formulated with gray-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as point processing.

Larger neighborhoods allow considerably more flexibility. The general approach is to use a function of the values of f in a predefined neighborhood of (x, y) to determine the value of g at (x, y). 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. 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.

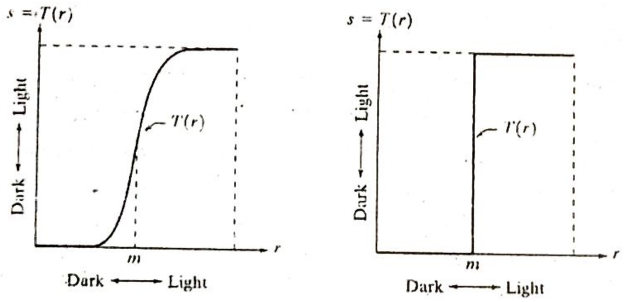


Fig. 2 : Graylevel transformation functions for contrast enhancement.

(b) Image Enhancement by Frequency Domain Methods : The convolution theorem is the basis for the frequency domain approaches. Let G(x, y) be an image formed by the convolution of the image f(x, y) and a linear position invariant operator H(x, y) and is given by,

$$G(x, y) = H(x,y) * f(x,y) \quad ...(1)$$

where * represents the convolution operation. Then, from the convolution theorem, the equation (1) can be written in the frequency domain as,

$$G(u, v) = H(u, v)\,F(u, v) \quad ...(2)$$

In the linear system theory, the transform H (u, v) is called the Transfer function. The various image enhancement problems can be expressed in the form of equation (1). In a typical image enhancement application, the image f(x, y) is given and the objective after the computation of F(u, v) is to select H(u, v) so that the desired image can be given by the equation

$$g(x, y) = F^{-1}[H(u, v)\,F(u, v)] \quad ...(3)$$

This equation shows some highlighted feature of the original image f(x, y). For example, the edges in f(x, y) can be highlighted using a function H(u, v) that emphasize the high frequency component of F(u, v).

Figure illustrates the various steps involved in the enhancement approach based on frequency domain.

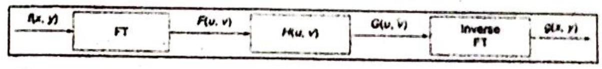


Fig. : Enhancement steps in frequency domain approaches

Q.7 ***How processing of quantized variables is done?***
[R.T.U. 2018]

Ans. Processing Quantized Variables : Numbers within a digital computer that represent image variables, such as luminance or tristimulus values, normally are input as the integer codes corresponding to the quantization reconstruction levels of the variables. If the quantization is linear, the *j*th integer value is given by

$$j = \left[(J-1)\frac{f-a_L}{a_U-a_L}\right]_N$$

where *J* is the maximum integer value, *f* is the unquantized pixel value over a lower-to-upper range of a_L to a_U, and $[\cdot]_N$ denotes the nearest integer value of the argument. The corresponding reconstruction value is

$$r_j = \frac{a_U-a_L}{J}j+\frac{a_U-a_L}{2J}+a_L$$

Hence, r_j is linearly proportional to *j*. If the computer processing operation is itself linear, the integer code *j* can be numerically processed rather than the real number r_j. However, if nonlinear processing is to be performed, for example, taking the logarithm of a pixel, it is necessary to process r_j as a real variable rather than the integer *j* because the operation is scale dependent. If the quantization is nonlinear, all processing must be performed in the real variable domain.

In a digital computer, there are two major forms of numeric representation: real and integer. Real numbers are stored in floating-point form, and typically have a large dynamic range with fine precision. Integer numbers can be strictly positive or bipolar (negative or positive). The two's complement number system is commonly used in computers and digital processing hardware for representing bipolar integers. The general format is as follows:

$$S.M_1,M_2,...,M_B\text{-}1$$

where *S* is a sign bit (0 for positive, 1 for negative), followed, conceptually, by a binary point, M_b denotes a magnitude bit, and *B* is the number of bits in the computer word. Table lists the two's complement correspondence between integer, fractional and decimal numbers for a 4-bit word. In this representation, all pixels

Its Fourier transform is

$$F[s'(t)] = 1 - \frac{a}{a + j2\pi f} = \frac{j2\pi f}{a + j2\pi f}$$

$$= j2\pi f\, S(f)$$

PART-C

Q.16 ***Derive the transformation required for histogram equalization.***

OR

Describe histogram equalization. Obtain histogram equalization for the following image segment of size 5×5. Write the interface on image segment before and after equalization.

[R.T.U. 2019]

OR

What do you understand by Histograms processing. Explain its specifications.

[R.T.U. 2017]

OR

What is histogram linearization?

OR

What is histogram equalization?

Ans. When continuous intensity values (in variable r) are considered, a transformation to the continuous variable s is of the form

$$S = T(r), \qquad 0 \le r \le L-1$$

Assumptions to be made :

(a) T(r) is strictly monotonically increasing function.

(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

These assumptions have to be made so that the mappings from s back to r are one-to-one.

The intensity levels may be considered to be random in the range [0, L − 1] and the pdf is $P_r(r)$. When r is transformed to s, the resulting pdf is

$$P_s(s) = P_r(r)\left|\frac{dr}{ds}\right|$$

Let us consider the transformation :

$$s = T(r) = (L-1)\int_0^r P_r(w)dw \qquad ...(1)$$

where w is a dummy variable (for integration).

Now,

$$\frac{ds}{dr} = \frac{d}{dr}T(r) = (L-1)\frac{d}{dr}\left[\int_0^r P_r(w)dw\right]$$

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

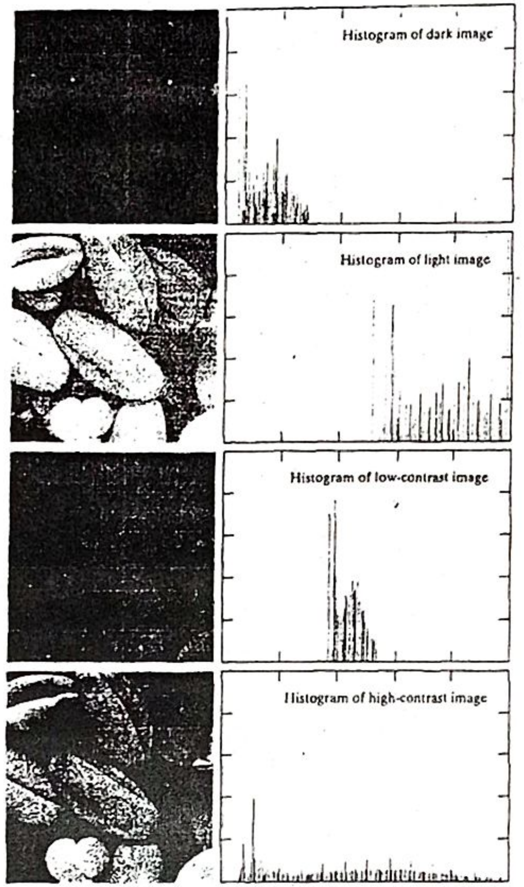


Fig. 1 : Four basic types : dark, light, low-contrast, high-contrast, and their corresponding histograms

$$\Rightarrow \qquad P_s(s) = P_r(r)\left|\frac{dr}{ds}\right|$$

$$= P_r(r)\left|\frac{1}{(L-1)P_r(r)}\right|$$

$$= \frac{1}{L-1}, \qquad 0 \le s \le L-1$$

Thus, $P_s(s)$ is a uniform pdf. If the intensities in an image are uniformly distributed, the image contrast will be high (good).

For discrete values, the probability that $r = r_k$ (a particular value of intensity) is approximately given by

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

where n_k = number of occurrences of the pixel of intensity k, MN is the total number of pixels in the image of size M × N. The plot of $p_r(r_k)$ vs. r_k is referred to as a histogram.

In the discrete form, transformation Eq. (1) is:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j)$$

or

$$s_k = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j,$$

$$k = 0, 1, \ldots, L-1$$

This transformation equation $T(r_k) = s_k$ is called a histogram equalizations/histogram linearization transformation. This transformation results in an image whose intensities are (nearly) uniformly distributed in the entire range [0, L − 1].

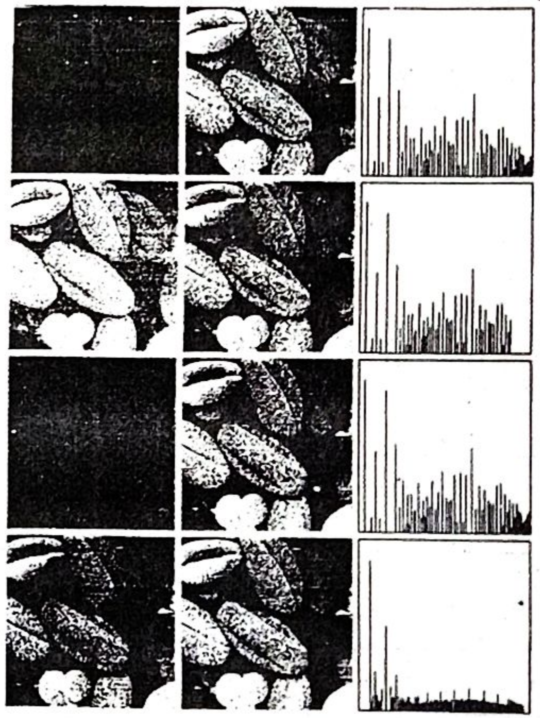


Fig. 2 : Left Column : image from fig. 1 centre column : corresponding histogram equalized images, Right column : histograms of the images in the centre column

Histogram Equalization Example

- Consider a color model with only 10 shades of gray 0 – 9.
- Consider a simple image with only 25 pixels.

1	1	2	2	3
1	5	9	1	3
0	4	4	9	9
0	1	2	7	6
9	8	0	1	2

Histogram Equalization Example

Step 1 : Count the number of pixels with each intensity.

Intensity	Count
0	3
1	6
2	4
3	2
4	2
5	1
6	1
7	1
8	1
9	4

1	1	2	2	3
1	5	9	1	3
0	4	4	9	9
0	1	2	7	6
9	8	0	1	2

Step 2 : Normalize the counts to fractions or percentages.

Intensity	Count	Fraction
0	3	3/25
1	6	6/25
2	4	4/25
3	2	2/25
4	2	2/25
5	1	1/25
6	1	1/25
7	1	1/25
8	1	1/25
9	4	4/25

Step 3 : Compute the cumulative distribution function CDF.

- Probability a pixel's intensity is less than or equal to the given intensity.
- Just a running total of the fractions / percentages from step 2.

Intensity	Count	Fraction	Cumulative Distribution
0	3	3/25	3/25
1	6	6/25	9/25(3 + 6)
2	4	4/25	13/25(3+6+4)
3	2	2/25	15/25
4	2	2/25	17/25
5	1	1/25	18/25
6	1	1/25	19/25
7	1	1/25	20/25
8	1	1/25	21/25
9	4	4/25	25/25

Step 4 : Scale Cumulative distribution to intensity range.

Intensity	Count	Fraction	CDF	Scaled Intensity
0	3	3/25	3/25	0(10*3/25 = 1 – 1 = 0)
1	6	6/25	9/25	3
2	4	4/25	13/25	4
3	2	2/25	15/25	5
4	2	2/25	17/25	6
5	1	1/25	18/25	6
6	1	1/25	19/25	7
7	1	1/25	20/25	7
8	1	1/25	21/25	7
9	4	4/25	25/25	9

Step 5 : The scaled intensities become a lookup table to apply to original image.

Intensity of Original	Intensity of Result
0	0
1	3
2	4
3	5
4	6

Intensity of Original	Intensity of Result
5	6
6	7
7	7
8	7
9	9

Step 6 : Apply lookup table.

Original	0	1	2	3	4	5	6	7	8	9
Result	0	3	4	5	6	6	7	7	7	9

1	1	2	2	3
1	5	9	1	3
0	4	4	9	9
0	1	2	7	6
9	8	0	1	2

Original

3	3	4	4	5
3	6	9	3	5
0	6	6	9	9
0	3	4	7	7
9	7	0	3	4

Result

0s stay 0, 1s become 3, 2s become 4 and so forth.

(a) Recall Actual Image

(b) Resulting Image

Fig.

Q.17 ***What do you mean by Fourier transforms? Explain its properties in detail.*** [R.T.U. 2016]

OR

List out the properties of 2D Fourier transform. Explain spatial filtering. **[R.T.U. 2019]**

OR

Explain the various properties of two dimensional Fourier transform. **[R.T.U. 2018]**

OR

Write short note on Fourier transform. **[R.T.U. 2018]**

OR

Explain the properties of Fourier transform in detail. **[R.T.U. 2017]**

Ans. Fourier Transforms : The discrete two-dimensional Fourier transform of an image array is defined in series form as·

$$f(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)\exp\left\{\frac{-2\pi i}{N}(uj+vk)\right\} \quad ...(1)$$

where $i = \sqrt{-1}$ and the discrete inverse transform is given by

$$F(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(u,v)\exp\left\{\frac{2\pi i}{N}(uj+vk)\right\} \quad ...(2)$$

The indices (u, v) are called the spatial frequencies of the transformation in analogy with the continuous Fourier transform. Because the transform kernels are separable and symmetric, the two dimensional transforms can be computed as sequential row and column one-dimensional transforms. The basis functions of the transform are complex exponentials that may be decomposed into sine and cosine components. The resulting Fourier transform pairs then become

$$A(j,k;u,v) = \exp\left\{\frac{-2\pi i}{N}(uj+vk)\right\}$$

$$= \cos\left\{\frac{2\pi}{N}(uj+vk)\right\} - i\sin\left\{\frac{2\pi}{N}(uj+vk)\right\} \quad ...(3)$$

$$B(j,k;u,v) = \exp\left\{\frac{2\pi i}{N}(uj+vk)\right\}$$

$$= \cos\left\{\frac{2\pi}{N}(uj+vk)\right\} + i\sin\left\{\frac{2\pi}{N}(uj+vk)\right\} \quad ...(4)$$

The spectral component at the origin of the Fourier domain

$$f(0,0) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k) \quad ...(5)$$

is equal to *N* times the spatial average of the image plane. Making the substitutions $u = u + mN$, $v = v + nN$ in eq.1 and 2, where *m* and *n* are constants, results in

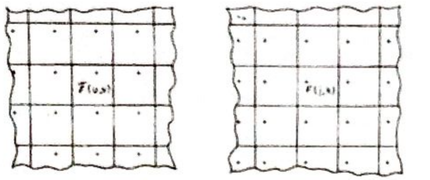
$f(u + m N, v + n N)$

$$= \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)\exp\left\{\frac{-2\pi i}{N}(uj+vk)\right\}$$
$$\exp\{-2\pi i\,(mj+nk)\} \quad ...(6)$$

For all integer values of m and n, the second exponential term of eq. 7 assumes a value of unity and the transform domain is found to be periodic. Thus, as shown in figure (a)

$$f(u+mN,\ v+nN) = f(u,\ v) \quad ...(7)$$

for m, n = 0 ± 1, ± 2, ...



(a) Fourier domain (b) Spatial domain

Fig. : Periodic image and Fourier transform arrays

The two-dimensional Fourier transform of an image is essentially a Fourier series representation of a two-dimensional field. For the Fourier series representation to be valid, the field must be periodic. Thus, as shown in figure (b), the original image must be considered to be periodic horizontally and vertically. The right side of the image therefore abuts the left side, and the top and bottom of the image are adjacent. Spatial frequencies along the coordinate axes of the transform plane arise from these transitions.

Properties of Fourier transform:

The Fourier transform has following properties

$$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0) \quad ...(1)$$

and

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}, \quad ...(2)$$

where the double arrow is used to designate a Fourier transform pair. When $u_0 = M/2$ and $v_0 = N/2$, it follows that

$$e^{-j2\pi(u_0x/M+v_0y/N)} = e^{j\pi(x+y)}$$
$$= (-1)^{x+y}$$

In this case, eq.(1) becomes

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2) \quad ...(3)$$

and similarly,

$$f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{(u+v)} \quad ...(4)$$

We see that eq. (3) is used for centering the transform. These results are based on the variables u and v having values in the range [0, M – 1] and [0, N – 1], respectively. In a computer implementation these variables will run from u = 1 to M and v = 1 to N, in which case the actual center of the transform will be at u = (M/2) + 1 and v = (N/2) + 1.

1. Distributivity and Scaling

From the definition of the Fourier transform it follows that

$$\Im[f_1(x,y)+f_2(x,y)] = \Im[f_1(x,y)] + \Im[f_2(x,y)] \quad ...(5)$$

and, in general that

$$\Im[f_1(x,y)\cdot f_2(x,y)] \neq \Im[f_1(x,y)]\cdot\Im[f_2(x,y)] \quad ...(6)$$

In other words, the Fourier transform is distributive over addition, but not over multiplication. Identical comments apply to the inverse Fourier transform. Similarly for two scalars a and b,

$$af(x,y) \Leftrightarrow aF(u,v) \quad ...(7)$$

and $$f(ax, by) \Leftrightarrow \frac{1}{|ab|}F(u/a, v/b). \quad ...(8)$$

2. Rotation

If we introduce the polar co-ordinates

$x = r\cos\theta,\ y = r\cos\theta$

$u = \omega\cos\varphi,\ v = \omega\cos\varphi$

then $f(x, y)$ and F(u, v) become $f(r, \theta)$ and F(ω, φ), respectively. Direct substitution into the definition of the Fourier transform yields

$$f(r, \theta+\theta_0) \Leftrightarrow F(\omega, \varphi+\theta_0) \quad ...(9)$$

This expression indicates that rotating $f(x, y)$ by an angle θ_0 rotates F(u, v) by the same angle. Similarly, rotating F(u, v) rotates $f(x, y)$ by the same angle.

3. Periodicity and Conjugate Symmetry

The discrete Fourier transform has following periodicity properties :

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

The inverse transform is also periodic

$$f(x,y) = f(x+M, y) = f(x, y+N) = F(x+M, y+N)$$

and for conjugate property

$$F(u,v) = F^*(-u,-v)$$

from which allow that spectrum also is symmetric about origin.

$$|F(u,v)| = |F(-u,-v)|$$

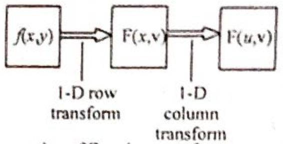
4. Separability

The discrete Fourier transform can be expressed in separable form

$$F(u,v) = \frac{1}{M}\sum_{x=0}^{M-1} e^{-(j2\pi ux/M)}\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi vy/N}$$
$$= \frac{1}{M}\sum_{x=0}^{M-1} F(x,v)e^{-j2\pi ux/M}$$

where $f(x, v) = \frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi vy/N}$

for each value of x and v = 0, 1, 2,..., N – 1. This equation is complete 1 - D Fourier transform, which is shown in fig.



Other properties of Fourier transform are as follows:

Time/Frequency Duality

The duality property is one that is not shared by the Laplace transform. While slightly confusing perhaps at first, it essentially doubles the size of our F.T. table. The duality property follows from the similarity of the forward and inverse F.T. It states that if

$f(t) \Leftrightarrow F(\omega)$

Then,

$F(t) \Leftrightarrow 2\pi(-\omega)$

Where, the function on the left is the function of time and the function on the right is the function of frequency.

Time-Shift Property

If,

$f(t) \Leftrightarrow F(\omega)$

then,

$f(t-t_0) \Leftrightarrow F(\omega)e^{-j\omega t_0}$

In other words, a shift in time corresponds to a change in phase in the F.T.

Frequency-shift Property

This innocuous-looking property forms a basis for every radio and TV transmitter in the world! It simply states that if

$f(t) \Leftrightarrow F(\omega)$

Then,

$f(t)e^{j\omega_0 t} \Leftrightarrow F(\omega-\omega_0)$

This property is the dual of the time-shift property.

Frequency-shift Property

This innocuous-looking property forms a basis for every radio and TV transmitter in the world! It simply states that if,

$f(t) \Leftrightarrow F(\omega)$

Then,

$f(t)e^{j\omega_0 t} \Leftrightarrow F(\omega-\omega_0)$

Note that this is a dual of the time-shift property.

Time Differentiation

If,

$f(t) \Leftrightarrow F(\omega)$

Then,

$\frac{df}{dt} \Leftrightarrow j\omega F(\omega)$

Time Integration

We saw above that,

$$\int_{-\infty} f(x)dx \Leftrightarrow \frac{F(\omega)}{j\omega} + \pi F(0)\delta(\omega)$$

If f is zero mean (i.e. F(0) = 0) then

$$\int_{-\infty} f(x)dx \Leftrightarrow \frac{F(\omega)}{j\omega}$$

Spatial Filtering : Spatial domain refers to the plane containing the pixels of an image. Spatial filtering corresponds to the operations, by working in a neighborhood of every pixel in an image.

So, the filtering has two components

(i) Neighborhood,

(ii) Predefined operation.

The neighborhood consists of original pixels of image and predefined operation consist of filter coefficients defined by some set of rule to obtain desired results. Here is an example of spatial filtering using 3 × 3 neighborhood. At any point (x,y) in the image, the response of the filter g(x,y) is sum of the products of filter coefficients and the neighborhood overlaid by the filter.

$$g(x,y) = w(-1,-1).f(x-1, y-1) + w(-1,0).f(x-1,y) + ... + w(0,0).f(x,y) + ... + w(1,1).f(x+1, y+1)$$

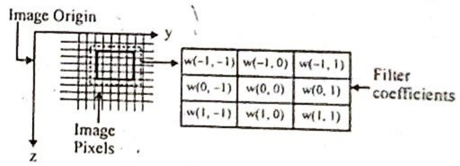


Fig. 1

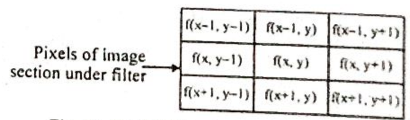


Fig. 2 : Spatial Filtering using 3×3 Filter

example of the finite-area convolution operator for a 2 ×2 ($N = 2$) input data array, a 4 ×4 ($M = 4$) output data array and a 3 ×3 ($L = 3$) impulse response array. The integer pairs (i, j) at each element of D represent the element (i, j) of $H\,i\,j(\,,\,)$. The basic structure of D can be seen more clearly in the larger matrix depicted in Figure. In this example, $M = 16$, $N = 8$, $L = 9$, and the impulse response has a symmetrical Gaussian shape. Note that D is a 256 x 64 matrix in this example.

Following the same technique as that leading to equation, the matrix form of the superposition operator may be written as

$$Q = \sum_{m=1}^{M}\sum_{n=1}^{N} D_{m,n} F v_n u_m^T$$

If the impulse response is spatially invariant and is of separable form such that

$$H = h_C h_R^T$$

where h_R and h_C are column vectors representing row and column impulse responses, respectively, then

$$D = D_C \otimes D_R$$

The matrices D_R and D_C are $M \times N$ matrices of the form

$$D_R = \begin{bmatrix} h_R(1) & 0 & \cdots & 0 \\ h_R(2) & h_R(1) & & \vdots \\ h_R(3) & h_R(2) & \cdots & 0 \\ \vdots & & & h_R(1) \\ h_R(L) & & & \vdots \\ 0 & & & \vdots \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & h_R(L) \end{bmatrix}$$

The two-dimensional convolution operation may then be computed by sequential row and column one-dimensional convolutions. Thus,

$$Q = D_C F D_R^T.$$

In vector form, the general finite-area superposition or convolution operator requires N^2L^2 operations if the zero-valued multiplications of D are avoided.

Q.8 *Explain the various noise models in details.* [R.T.U. 2016]

Ans. Noise Models:

Different noises and their PDF are given below

(a) Gaussian Noise: PDF of gaussian (white noise)is given by

$$P(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(z-\mu)^2/2\sigma^2} \qquad \ldots (1)$$

Where, z = gray levels

μ = mean value of z

σ = standard deviation

σ^2 = variance of z.

Gaussian Noise is generated due to

- Electronic circuit noise
- In Sensor noise due to poor illumination
- Sensor noise due to high temperature.

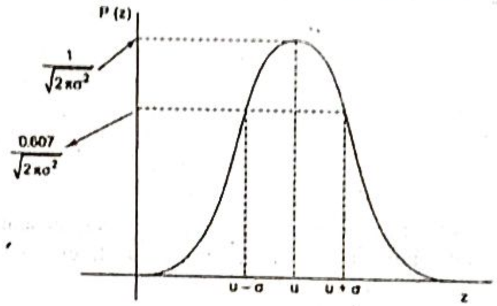


Fig. 1 : Gaussian Noise

(b) Rayleigh Noise: The PDF of Rayleigh noise is given by

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

Here the value of mean

$$\mu = a + \sqrt{\frac{\pi b}{4}} \qquad \ldots (3)$$

and variance,

$$\sigma^2 = \frac{b(4-\pi)}{a} \qquad \ldots (4)$$

Rayleigh noise is generated due to Range imaging

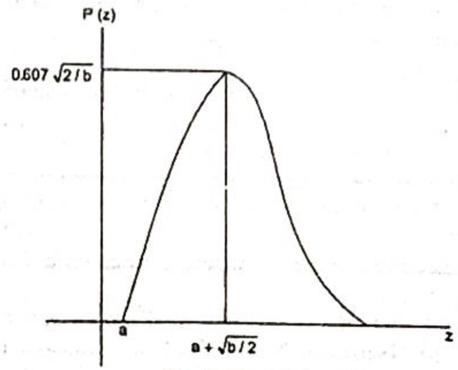


Fig. 2 : Rayleigh noise.

(c) Exponential Noise: The PDF is given by

$$P(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \qquad \ldots (5)$$

mean given by

$$\mu = \frac{1}{a} \qquad \ldots (6)$$

Variance given by

$$\sigma^2 = \frac{1}{a^2} \qquad \ldots (7)$$

Its application is laser imaging.

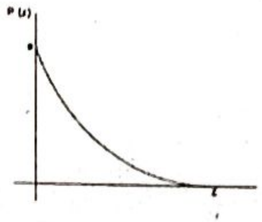


Fig. 3 : Exponential noise

(d) Uniform Noise: The PDF of uniform noise is given by

$$P(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \qquad \ldots (8)$$

Mean value is given by

$$\mu = \frac{a+b}{2} \qquad \ldots (9)$$

Variance given by $\sigma^2 = \frac{(b-a)^2}{12}$ ...(10)

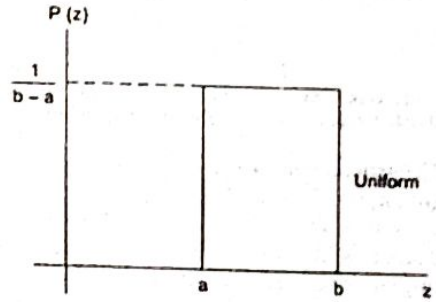


Fig. 4 : Uniform noise

(e) Impulse (Salt and pepper noise): The PDF of impulse noise is given by

$$P(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \qquad \ldots (11)$$

Q.9 *Explain basic concept of image enhancement.* **[R.T.U. 2011]**

Ans. Image Enhancement : In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo coloring, noise filtering, sharpening and magnifying. Image enhancement is useful in feature extraction, image analysis and visual information display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent.

Image enhancement techniques, such as contrast stretching, map each gray level into another gray level by a predetermined transformation. An example is the histogram equalization method, where the input gray levels are mapped so that the output gray level distribution is uniform. This has been found to be a powerful method of enhancement of low contrast images (see Fig.). Other enhancement techniques perform local neighborhood operations as in convolution, transform operations is in the discrete Fourier transform, and other operations as in pseudo coloring where a gray level image is mapped into a color image by assigning different colors to different features.

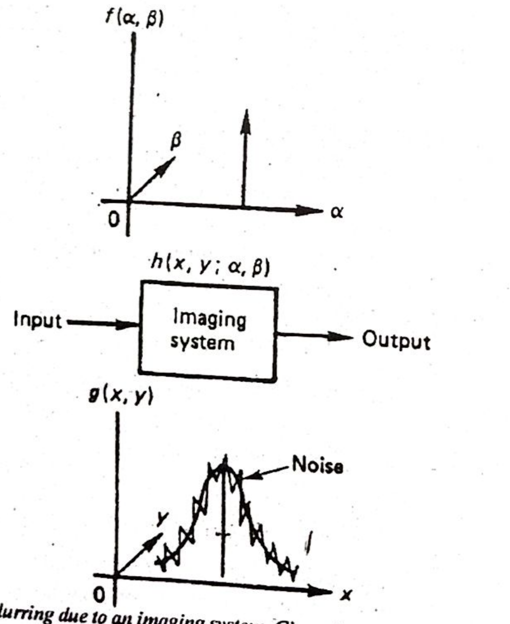


Fig. : Blurring due to an imaging system. Given the noisy and blurred image the image restoration problem is to find an estimate of the input image f(t, y)

PART-C

Q.10 *Write short note on :*
(i) Homomorphic filter
(ii) Inverse filter
(iii) Wiener filter **[R.T.U. 2019]**

OR

Design homomorphic filtering. Explain homomorphic filtering model. How do we get back the modified image? **[R.T.U. 2016]**

OR

Design Homo morphic filtering. How do we get back the modified image? **[R.T.U. 2017]**

Ans.(i) Homomorphic Filter : The digital images we process are created from optical images. Optical images consist of two primary components, the lighting component and the reflectance component. The lighting component results from the lighting conditions present when the image is captured, and can change as the lighting conditions change. The reflectance component results from the way the objects in the image reflect light and are determined by the intrinsic properties of the object itself, which (normally) do not change. In many applications it is useful to enhance the reflectance component, while reducing the contribution from the lighting component. Homomorphic filtering is a frequency domain filtering process that compresses the brightness (from the lighting conditions), while enhancing the contrast (from the reflectance). The image model for homomorphic filters is as follows:

$$I(r,c) = L(r,c)R(r,c)$$

where L(r, c) represents the contribution of the lighting conditions and R(r, c) represents the contribution of the reflectance properties of the objects.

The homomorphic filtering process assumes that L(r,c) consists of primarily slow spatial changes (low spatial frequencies), and is responsible for the overall range of the brightness in the image. The assumptions for R(r,c) are that it consists primarily of high spatial frequency information, which is especially true at object boundaries, and it is responsible for the local contrast (the spread of the brightness range within a small spatial area). These simplifying assumptions are valid for many types of real images.

The homomorphic filtering process consists of five steps: (1) a natural log transform (base e), (2) the Fourier transform, (3) Filtering, (4) the inverse Fourier transform, and (5) the inverse log function-the exponential. This process is illustrated in a block diagram in Figure 1. The first step allows us to decouple the L(r,c) and R(r,c) components, since the logarithm function changes a product into a sum. Step 2 puts the image into the frequency domain, so that we can perform the filtering in Step 3. Next, Steps 4 and 5 do the inverse transforms from Steps 1 and 2, to get our image data back into the spatial domain. The only factor left to be considered is the filter function, H(u, v).

The typical filter for the homomorphic filtering process. Here we can specify three parameters-the high frequency gain, the low frequency gain, and the cutoff frequency. Typically the high frequency gain is greater than 1, and the low frequency gain is less than 1. This provides us with the desired effect of boosting the R(r,c) components, while reducing the L(r,c) components. The selection of the cutoff frequency is highly application-specific, and needs to be chosen so that no important information is lost. In practice the values for all three parameters are often determined empirically.

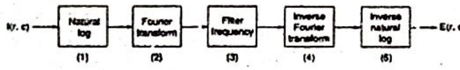


Fig. 1 The homomorphic filtering process

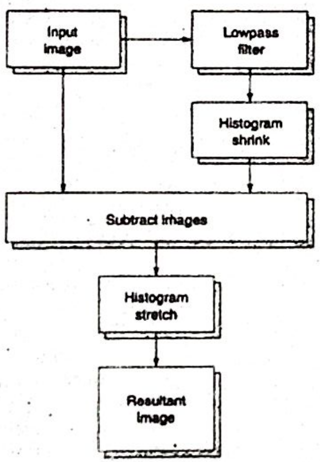


Fig. 2 : Unsharp Maskign enhancement algorithm flowchart. Unsharp masking subtracts a blurred (lowpassed) version of the original, which is similar to addring an edge-enchanced (highpassed) version

Result from application of homomorphic filtering to a poor image. In this case, the homomorphic filter returns an image of low contrast, so the contrast is enhanced by a histogram stretch procedure. A comparison is made between the homomorphic filter followed by a histogram stretch and simply stretching the original image's histogram. We see that the homomorphic filter provides an image with greater visual detail, where it is much easier to see the pilot in the helicopter.

Figure 2 Unsharp masking enhancement algorithm flowchart. Unsharp masking subtracts a blurred (lowpassed) version of the original, which is similar to adding an edge-enhanced (highpassed) version.

A blurred negative onto the corresponding positive film to produce a sharper result. The process is similar to adding a detail enhanced (highpass) version of the image to the original. To improve image contrast we have included histogram modification as part of our unsharp masking enhancement algorithm. A flowchart for this process is shown in Figure 2. Here we see that the original image is lowpass filtered, followed by a histogram shrink to the lowpass filtered image. The resultant image from these two operations is then subtracted from the original image, and the result of this operation undergoes a histogram stretch to restore the image contrast. This process works because subtracting a slowly changing edge (the lowpass filtered image) from faster changing edges (in the original), has the visual effect of causing overshoot and undershoot at the edges, which has the effect of emphasizing the edges. By the scaling the lowpassed image with a histogram shrink we can control the amount of edge emphasis desired. In Result of application of the unsharp masking algorithm with different ranges for the histogram shrink process. Here we see that as the range for the histogram shrink is increased, the resulting image has a greater edge emphasis.

(ii) Inverse Filter : The simplest approach to restoration is direct inverse filtering, where we compute 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 the degradation function :

$$\hat{F}(u, v), = \frac{G(u, v)}{H(u, v)} \quad ...(1)$$

If H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \quad ...(2)$$

where h(x, y) is the spatial representation of the degradation

function and, the symbol "★" indicates convolution. We know that convolution in the spatial domain is analogous to multiplication in the frequency domain, so we may write the model in Eq. (2) in an equivalent frequency domain representation :

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad ...(3)$$

where the terms in capital letters are the Fourier transforms of the corresponding terms in Eq. (2).

Substituting the right side of Eq. (3) for G(u, v) in Eq. (1) yields

$$\hat{F}(u, v), = F(u, v) + \frac{N(u, v)}{H(u, v)} \quad ...(4)$$

This is an interesting expression. It tells us that even if we know the degradation function we cannot recover the under graded image [the inverse Fourier transform of F(u, v)] exactly because N(u, v) is not known. If the degradation function has zero or very small values, then the ratio N(u, v)/H(u, v) could easily dominate the estimate $\hat{F}(u, v)$. This, in fact, is frequently the case, as will be demonstrated shortly.

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

(iii) Minimum Mean Square Error (Wiener) Filtering: This method is founded on considering images and noise as random processes and the objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized. This error measure is given by

$$e^2 = E\{(f - \hat{f})^2\} \quad ...(i)$$

Where E{.} is the expected value of the argument. It is assumed that the noise and the image are uncorrelated that one or the other has zero mean and that the gray levels in the estimate are a linear function of the levels in the degraded image. Based on these conditions, the minimum of the error function in Eq.(i) is given in the frequency domain by the expression.

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

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

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

Where we used the fact that the product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squared. This result is known as the Wiener filter. The filter, which consists of the terms inside the brackets, also is commonly referred to as the minimum mean square error filter or the least square filter. Note from the first line in Eq.(ii) that the Wiener filter does not have the same problem as the inverse filter with zeros in the degradation function. Unless both H(u,v) and $S_\eta(u,v)$ are zero for the same value(s) of u and v.

The terms in Eq. (ii) are as follows:

H (u,v) = degradation function

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

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

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

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

As before, H (u,v) is the transform of the degradation function and G(u,v) is the transform of the degraded image. The restored image in the spatial domain is given by the inverse Fourier transform of the frequency – domain estimate $\hat{F}(u,v)$. Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter.

When we are dealing with spectrally while noise, the spectrum $|N(u,v)|^2$ is constant. Which simplifies things considerably. However, the power spectrum of the undegraded image seldom is known. An approach used frequently when these quantities are not known or cannot be estimated is to approximate Eq.(ii) by the expression

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

Q.11 ***Explain the expression for observed image when the degradation are linear, position invariant.*** **[R.T.U. 2019]**

Ans.

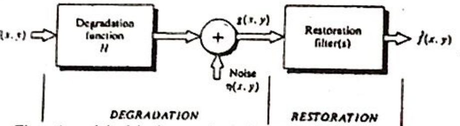


Fig. : A model of the image degradation/restoration process

Linear, Position – invariant Degradations

The input – output relationship in Figure before the restoration can be expressed as

$$g(x,y) = H[f(x,y)] + \eta(x,y) \quad ...(i)$$

First, we assume that $\eta(x,y) = 0$ so that $g(x,y) = H[f(x,y)]$. H is linear if

$$H[af_1(x,y) + bf_2(x,y)] = aH[f_1(x,y)] + bH[f_2(x,y)] \quad ...(ii)$$

Where a and b are scalars and $f_1(x,y)$ and $f_2(x,y)$ are any two input images. If $a = b = 1$, then Eq. (ii) becomes

$$H[f_1(x,y) + f_2(x,y)] = H[f_1(x,y)] + H[f_2(x,y)] \quad ...(iii)$$

If $f_2(x,y) = 0$, becomes

$$H[af_1(x,y)] = aH[f_1(x,y)] \quad ...(iv)$$

Which is called the property of homogeneity. It says that the response to a constant multiple of any input is equal to the response to that input multiplied by the same constant.

An operator having the input – output relationship

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

is said to be position (space) invariant if

$$H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta) \quad ...(v)$$

For any f(x,y) and any α and β Eq.(v) indicates that the response at any point in the image depends only on the value of the input at that point, not on its position.

With a slight change in notation in the definition of the impulse in

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\delta(t-t_0, z-z_0)\,dt\,dz = f(t_0, z_0)$$

f(x,y) can be expressed as

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta \quad ...(vi)$$

Assuming $\eta(x,y) = 0$, then substituting Eq.(vi) into Eq.(i) we have

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

$$= H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta\right] \quad ...(vii)$$

If H is a linear operator, then

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H[f(\alpha,\beta)\delta(x-\alpha, y-\beta)\,d\alpha\,d\beta] \quad ...(viii)$$

Since $f(\alpha,\beta)$ is independent of x and y, using the homogeneity property, it follows that

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)H[\delta(x-\alpha, y-\beta)]\,d\alpha\,d\beta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h[\delta(x,\alpha,y,\beta)]\,d\alpha\,d\beta \quad ...(ix)$$

Where the term

$$h(x,\alpha,y,\beta) = H[\delta(x-\alpha, y-\beta)] \quad ...(x)$$

is called the impulse response of H.

In other words, if $\eta(x,y) = 0$, then $h(x,\alpha,y,\beta)$ is the response of H to an impulse at (x,y).

Equation

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)\,d\alpha\,d\beta \quad ...(xi)$$

is called the superposition (or Fredholm) integral of the first kind and is a fundamental result at the core of linear system theory.

If H is position invariant, from

$$H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta),$$

We have

$$H[\delta(x-\alpha, y-\beta)] = h(x-\alpha, y-\beta), \quad ...(xii)$$

And Eq. (xi) reduces to

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha, y-\beta)\,d\alpha\,d\beta \quad ...(xiii)$$

The expression Eq. (xiii) is the case of convolution integral

$$f(t)*h(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)\,d\tau$$

Being extended to 2-D.

Equation (xiii) tells us that knowing the impulse of a linear system allows us to compute its response, g, to any input f. The result is simply the convolution of the impulse response and the input function.

In the presence of additive noise, Eq. (xi) becomes

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)\,d\alpha\,d\beta + \eta(x,y) \quad ...(xiv)$$

If H is position invariant, it becomes

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha, y-\beta)\,d\alpha\,d\beta + \eta(x,y) \quad ...(xv)$$

Assuming that the values of the random noise $\eta(x,y)$ are independent of position, we have

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

Based on the convolution theorem, we can express Eq. (xvi) in the frequency domain as

$$G(u,v) = H(u,v)F(u,v) + N(u,v). \quad ...(xvii)$$

In summary, a linear, spatially invariant degradation system with additive noise can be modeled in the spatial domain as the convolution of the degradation function with an image, followed by the additive of noise (as expressed in Eq. (xvi)).

Q.12 ***What is image restoration model? Explain paint and spatial image restoration models.*** [R.T.U. 2019]

OR

Explain the general image restoration models. [R.T.U. 2016]

OR

Explain image degradation and restoration process. [R.T.U. 2017]

OR

Explain the Image Degradation Model and Restoration Process. [R.T.U. 2013]

OR

Explain Image Degradation Model. Explain various types of Noise and their PDF usually occurs in Image. [R.T.U. 2010]

Ans. Model of the Image Degradation/Restoration Process : As fig. shows, the degradation process is modeled as a degradation function that, together with an additive noise term, operates on an input image $f(x, y)$ to produce a degraded image $g(x, y)$, some knowledge about the degradation function H and some knowledge about the additive noise term $\eta(x, y)$, the objective of restoration is to obtain an estimate $\hat{f}(x, y)$ of the original image.

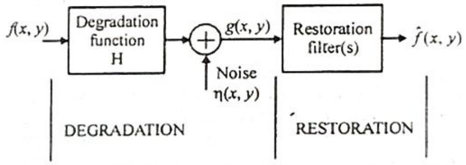


Fig. : A model of the image degradation/restoration process

H is a linear, position-invariant process, then the degraded image is given in the spatial domain by

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \quad ...(1)$$

where $h(x, y)$ is the spatial representation of the degradation function and the symbol "*" indicates spatial convolution. Convolution in the spatial domain is equal to multiplication in the frequency domain, so we may write the model in Equation (1) in an equivalent frequency domain representation :

$$G(u, v) = H(u, v)\,F(u, v) + N(u, v) \quad ...(2)$$

where the terms in capital letters are the Fourier transforms of the corresponding terms in equation (1).

Paint and Spatial Image Restoration Model Process : Fig. shows the model of image restoration process, which includes the noise adding the noise with the original image and restoring the original image from the noisy one by using restoration filter. Noise modeling is performed for impulse noise, speckle noise and Gaussian noise. Impulse noise usually occurs separately from the noise normally introduced during the acquisition. It alters pixels randomly making their values much different from the true values and very often, totally different from those of neighboring pixels as well. Impulse noise appears in the image as a sprinkle of dark and light spots. Noise generators such as digital computing devices, RF circuits, and ignition systems, may radiate noise signals and affect television signals. The television images may easily be corrupted by impulse noise through wireless channels.

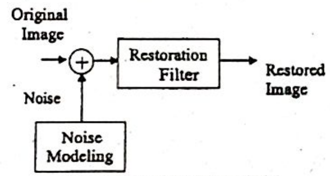


Fig. : Image Restoration Model

Q.13 ***Define the process of restoration. Explain any four important noise probability density functions.*** [R.T.U. 2019]

OR

Explain various types of Noise & their PDF usually occurs in Image. [R.T.U. 2013]

Ans. The Process of Restoration : *Refer to Q.12.*

Noise Probability Density Functions : The spatial noise descriptor is the statistical behavior of the gray-level values in the noise component of the model. These may be considered random variables, characterized by a probability density function (PDF). The following are among the most common PDFs found in image processing applications.

1. Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called normal) noise models are used frequently in practice.

In fact, this tractability is so convenient that it often results. Gaussian models being used in situations in which they are marginally applicable at best.

The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \quad ...(1)$$

where z represents gray level, μ is the mean of average value of z and σ is its standard deviation.

IMAGE COMPRESSION 4

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 *What is Huffman Coding?*

Ans. Huffman compression reduces the average code length used to represent the symbols of an alphabet. Symbols of the source alphabet, which occur frequently, are assigned with short length codes. The general strategy is to allow the code length to vary from character to character and to ensure that the frequently occuring characters have shorter codes.

Q.2 *What is Arithmetic Coding?*

Ans. Arithmetic compression is an alternative to Huffman compression; it enables characters to be represented as fractional bit lengths. Arithmetic coding works by representing a number by an interval of real numbers greater or equal to zero, but less than one. As a message becomes longer, the interval needed to represent it becomes smaller and smaller, and the number of bits needed to specify it increases.

Q.3 *What are the basic steps in JPEG?*

Ans. The major steps in JPEG coding involves :

- DCT (Discrete Cosine Transformation)
- Quantization
- Zigzag Scan
- DPCM on DC components
- RLE on AC components
- Entropy Coding

Q.4 *What are two main types of Data compression?*

Ans. Lossless compression can recover the exact original data after compression. It is used mainly for compressing database records, spreadsheets or word processing files, where exact replication of the original is essential.

Lossy compression will result in a certain loss of accuracy in exchange for a substantial increase in compression. Lossy compression is more effective when used to compress graphic images and digitised voice where losses outside visual or aural perception can be tolerated.

Q.5 *Define coding redundancy.*

Ans. If the gray level of an image is coded in way that uses more code words than necessary to represent each gray level, then the resulting image is said to contain coding redundancy.

PART-B

Q.6 *What are the basic steps in JPEG compression? Explain.* **[R.T.U. 2019]**

OR

Describe JPEG in detail. **[R.T.U. 2016]**

OR

Write short note on JPEG compression. **[R.T.U. 2017]**

Ans. JPEG compression : JPEG refers to a wide variety of possible image compression approaches that have been collected into a single standard. In this section we attempt to describe JPEG in a somewhat general but comprehensible way. A very complete description of the JPEG standard has been presented in Pennabaker and Mitchell (1993). The JPEG standard has both lossless and lossy components. In addition, the entropy coding employed by JPEG can be either Huffman coding or binary arithmetic coding. Figures 1 and 2 present very general image compression models that help describe the JPEG standard. In Figure 1 the compression process is broken into two basic functions: modeling the image data and entropy coding the description provided by a particular model. As the figure indicates, the modeling and entropy coding are separate. Hence whether Huffman or arithmetic entropy codes are used is irrelevant to the modeling. Any standard application-specific or image-specific coding tables can be used for entropy coding. The reverse process is illustrated in Figure 2. The modes of operation for JPEG are depicted in Figure 3. Two basic functional modes exist: nonhierarchical and hierarchical. Within the nonhierarchical modes are the sequential lossless and the lossy DCT-based sequential and progressive modes. The sequential modes progress through an image segment in a strict left-to-right, top-to-bottom pass. The progressive modes allow several refinements through an image segment, with increasing quality after each refinement. The hierarchical mode allows combinations of nonhierarchical modes, progressive coding with increasing resolution, coding of difference images, and multiple frames per image (the nonhierarchical modes allow only a single frame per image).

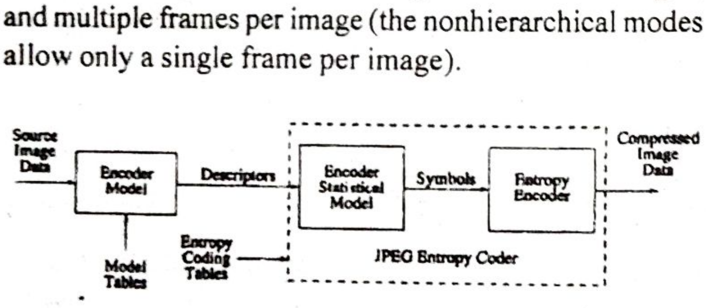


Fig. 1 : General JPEG encoder models.

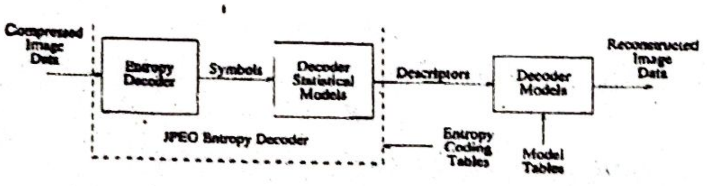


Fig.2 : General JPEG decoder models.

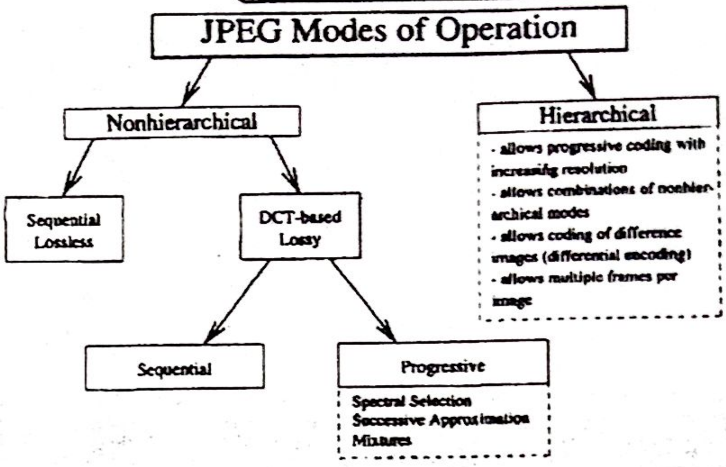


Fig. 3 : JPEG modes of operation.

Q.7 ***Explain in detail various techniques for Contrast manipulation.*** **[R.T.U. 2018]**

Ans. Contrast Manipulation in Digital Images : The term contrast refers to the amount of color or grayscale differentiation that exists between various image features in both analog and digital images. Images having a higher contrast level generally display a greater degree of color or grayscale variation than those of lower contrast.

The contrast of a digital image is related to the RGB intensity (grayscale) values of the optical image and the accuracy of the digitizing device used to capture the optical image. The RGB intensity refers to the amount of red, green, and blue light energy actually reflected by, or transmitted through, the physical specimen at a given point. Although a number of factors play a role in specimen contrast, insufficient or non-uniform illumination and/or an incorrectly adjusted microscope can result in underexposure or blurring of specimen details, which is often a major cause of low contrast in digital images. The RGB intensity range utilized in construction of a digital image is known as the dynamic range of the image and is a function of both the intensity range of the optical image and the accuracy of the camera used to capture the image. If the dynamic range of the digital image is severely limited by the camera's digitizer, then too few intensity levels will be available in each color channel to represent the subtle differences of intensity that may occur in the optical image. As a result, image contrast will suffer.

Deficiencies in digital image contrast can often be corrected by utilizing an intensity transformation operation, which is an algorithm designed to transform each input brightness value to a corresponding output brightness value via a transfer function. The purpose of the transfer function is simply to define a set of rules for assigning input pixel

Q.13 Describe Lossy compression techniques.

[R.T.U. 2017]

Ans. Lossy Compression Technique : Lossy compression simply means that some data from the original image file is lost. In general a lossy compression is implemented using spatial domain encoding and transform domain encoding methods. Spatial domain techniques generally make use of the prediction function in which value of the current pixel is determined by the knowledge of previously coded pixel. Delta modulation and pulse code modulation are example of predictive coding.

In transform domain technique, image transforms are used to decorrelate the pixels. Thus the image information is packed into a small numbers of coefficients. The coefficients in the transform domain are then quantized to reduce the number of allocated bits. The error or loss in information is due to quantization step. The resulting quantized coefficients are of different probabilities and an entropy coding scheme can further reduce the number of required bits. Transform coding is commonly adopted method for lossy image compression as it provides greater data compression as compared to predictive methods.

The transforms used to decorrelate the image pixels are Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Walsh Hadamard Transform (WHT), Karhunen-Loeve Transform (KLT) and Discrete Wavelet Transform (DWT). Among these transforms, DET and DWT are the most popular techniques. Joint photographic experts group (JPEG) is a compression algorithm based on DCT transforms. The main drawback of DCT is that blocking artifacts are visible at lower bitrates. DWT on

PART-C

Q.15 *Draw a image compression model and describe the work of each block.* **[R.T.U. 2019]**

OR

Explain Image Compression model. **[R.T.U. 2010]**

OR

How these Redundancy is Removed? Explain with the help of Image Compression Model. **[R.T.U. 2013]**

Ans. Image Compression Models

As figure 1 shows, a compression system consists of two distinct structural blocks: an encoder and a decoder. An input image $f(x, y)$ is fed into the encoder, which creates a set of symbols form the input data. After transmission over the channel, the encoded representation is fed to the decoder where a reconstructed output image $\hat{f}(x,y)$ is generated. In general, $\hat{f}(x,y)$ may or may not be an exact replica of $f(x, y)$. If it is, the system is error free or information preserving; if not, some level of distortion is present in the reconstructed image.

Both the encoder and decoder shown in figure 1 consist of two relatively independent functions or subblocks. The encoder is made up of a source encoder, which removes input redundancies and a channel encoder, which increases the noise immunity of the source encoder's output. As would be expected, the decoder includes a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free (not prone to error), the channel encoder and decoder are omitted and the general encoder and decoder become the source encoder and decoder, respectively.

Source Encoder and Decoder

The source encoder is responsible for reducing or eliminating any coding, interpixel or psychovisual redundancies in the input image. The specific application and associated fidelity requirements dictate the best encoding approach to use in any given situation. Normally, the approach can be modeled by a series of three independent operations. As fig.2(a) shows, each operation is designed to reduce the redundancies. Fig.2(b) depicts the corresponding source decoder.

In the first stage of the source encoding process, the mapper transforms the input data in the (usually nonvisual) format designed to reduce interpixel

redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image.

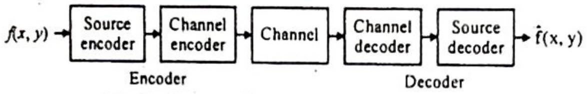


Fig.1 : A general compression system model

Run-length coding is an example of a mapping that directly results in data compression in this initial stage of the overall source encoding process. The representation of an image by a set of transform coefficients is an example of the opposite case. Here, the mapper transforms the image into an array of coefficients, making its interpixel redundancies more accessible for compression in later stage of the encoding process.

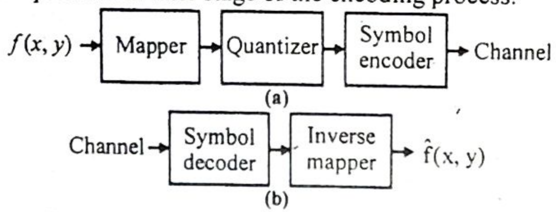


Fig. 2 : (a) Source encoder & (b) Source decoder model

The second stage or quantizer block in figure 2(a), reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. This stage reduces the psychovisual redundancies of the input image. This operation is irreversible. Thus it must be omitted when error-free compression is desired.

In the third and final stage of the source encoding process, the symbol coder creates a fixed or variable-length code to represent the quantizer output and maps the output in accordance with the code. The term symbol coder distinguishes this coding operation from the overall source encoding process. In most cases, a variable-length code is used to represent the mapped and quantized data set. It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy. The operation, of course, is reversible. Upon completion of the symbol coding step, the input image has been processed to remove each of the three redundancies.

Fig.2(a) shows the source encoding process as three successive operations, but all three operations are not necessarily included in every compression system. For example, the quantizer must be omitted when error-free compression is desiréd. In addition, some compression techniques normally are modeled by merging blocks that are physically separate in fig.2(a).

The source decoder shown in fig.2(b) contains only two components : a symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model shown in fig.2(b).

Q.16 *Write short notes on :*
(a) Interpixel redundancy
(b) Coding redundancy
(c) Psychovisual redundancy **[R.T.U. 2017]**

OR

Write short note on:
(a) Inter pixel redundancy
(b) Coding redundancy **[R.T.U. 2019]**

Ans.(a) Interpixel Redundancy : Consider the images shown in fig.(a) and (b). As fig.(c) and (d) show, these images have virtually identical histograms. Note also that histograms are trimodal, indicating the presence of three dominant rages of gray-level values. Because the gray levels in these images are not equally probable, variable-length coding can be used to reduce the coding redundancy that would result from a straight or natural binary encoding of their pixels. The coding process, however, would not alter the level of correlation between the pixels within the images. In other words, the codes used to represent the gray levels of each image have nothing to do with the correlation between pixels. These correlations result from the structural or geometric relationships between the objects in the image.

Fig.(e) and (f) show the respective autocorrelation coefficients computed along one line of each image. These coefficients were computed using a normalized equation in which

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)} \qquad \ldots(1)$$

where

$$A(\Delta n) = \frac{1}{N-\Delta n} \sum_{y=0}^{N-1-\Delta n} f(x,y) f(x, y+\Delta n) \qquad \ldots(2)$$

The scaling factor in equation (2) accounts for the varying number of sum terms that arise for each integer value of Δn. Of course, Δn must be strictly less than N, the number of pixels on a line. The variable x is the co-ordinate of the line used in the computation. Note the dramatic difference between the shape of the function shown in fig.(e) and (f). Their shapes can be qualitatively related to the structure in the images in fig.(a) and (b). This relationship is particularly noticeable in fig.(f), where the high correlation between pixels separated by 45 and 90 samples can be directly related to the spacing between the vertically oriented matches of fig.(b).

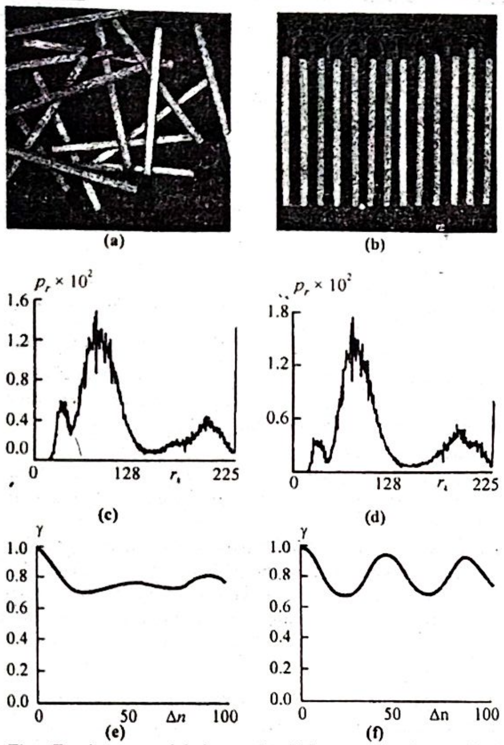


Fig. : Two images and their gray-level histograms and normalized autocorrelation coefficients along one line

In addition, the adjacent pixels of both images are highly correlated. When Δn is 1, γ is 0.9922 and 0.9928 for the images of fig.(a) and (b), respectively. These values are typical of most properly sampled television images.

These illustrations reflect another important form of data redundancy- one directly related to the interpixel correlations within an image. Because the value of any given pixel can be reasonably predicted from the value of its neighbors, the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names, including spatial redundancy, geometric redundancy and interframe redundancy, have been coined to refer to these interpixel dependencies. We use the term interpixel redundancy to encompass them all.

In order to reduce the interpixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "nonvisual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove interpixel redundancy) are referred to as mappings. They are called reversible mappings if the original image elements can be reconstructed from the transformed data set.

Ans.(b) Coding Redundancy : Let us assume, that a discrete random variable r_k in the interval [0, 1] represents the gray levels of an image and that each r_k occurs with probability $p_r(r_k)$.

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

Where L is the number of gray levels, n_k is the number of times that the k^{th} gray level appears in the image and n is the total number of pixels in the image. If the number of bits required to represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)\, p_r(r_k) \qquad ...(2)$$

That is, the average length of the code words assigned to the various gray-level values is found by summing the product of the bits used to represent each gray level and the probability that the gray level occurs. Thus the total number of bits required to code an M × N image is MNL_{avg}.

Representing the gray levels of an image with a natural m-bit binary code reduces the right-hand side of equation (2) to m bits. That is L_{avg} = m when m is substituted for $\ell(r_k)$. Then the constant m may be taken outside the summation, leaving only the sum of the $p_r(r_k)$ for $0 \le k \le L-1$, which, of course, equals 1.

Ans.(c) Psychovisual Redundancy : We know that the brightness of a region, as perceived by the eye, depends on factors other than simply the light reflected by the region. For example, intensity variations (Mach bands) can be perceived in an area of constant intensity. Such phenomena result from the fact that the eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psychovisually redundant. It can be eliminated without significantly impairing the quality of image perception.

That psychovisual redundancies exist should not come as a surprise, because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image. In general, an observer searches for distinguishing features such as edges or textural regional and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process.

Psychovisual redundancy is fundamentally different from the redundancies. Unlike coding and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisually redundant data results in a loss quantitative information, it is commonly referred to as quantization. This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. As it is irreversible operation (visual information is lost), quantization results in lossy data compression.

Q.17 *Explain Lossy and Lossless coding techniques.*
[R.T.U. 2019]

Ans. Lossless Image Coding Techniques : Image compression is the application of data compression on digital images. The objective is to reduce redundancy of the image data to be able to store or transmit data in an efficient form. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photographs in applications where minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless. Typical methods for lossless image compression are as follows:

(i) Run-length Encoding : Run-length encoding (RLE) is used as a default method in PCX and as one of possible method in BMP, TGA and TIFF. RLE is a very simple form of data compression in which runs of data are stored as a single data value and its count, rather than as the original run. This is most useful for data that contain many such runs, for example, relatively simple graphic images such as icons, line drawings and animations. It is nor recommended for use with files that do not have many runs as it could potentially double the file size.

(ii) Differential Pulse-code Modulation and Predictive Coding : Differential pulse-code modulation (DPCM) was invented by C. Chapin Cutler at Bell Labs in 1950 and his patent includes both methods. DPCM is a signal encoder that uses the baseline of PCM but adds some functionality based on the prediction of the samples of the signal. The input can be an analog signal or a digital signal. If the input is a continuous-time analog signal, it needs to be sampled first so that a discrete time signal is the input to the DPCM encoder. There are two options. The first one is to take the values of two consecutive samples; the difference between the first one and the next is calculated and the difference is further entropy coded. The other one is, instead of taking a difference relative to a previous input sample, the difference relative to the output of a local model of the decoder process is taken, and in this option, the difference can be quantized, which allows a good way to incorporate controlled loss in the encoding.

(iii) Entropy Encoding : In information theory, entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. One of the main types of entropy coding creates and assigns a unique prefix code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol by the corresponding variable-length prefix codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes. According to Shannon's source coding theorem, the optimal code length for a symbol is $\log_b P$, where b is the number of symbols used to make output codes and P is the probability of the input symbol. Two most commonly used entropy encoding techniques are Huffman coding and arithmetic coding. If the approximate entropy characteristics of a data stream are known in advance, a simpler static code may be useful.

(iv) Adaptive Dictionary Algorithms : Adaptive dictionary algorithms are used in GIF and TIFF and the typical one is the LZW algorithm. It is a universal lossless data compression algorithm created by Lempel, Ziv and Welch. It was published by Welch in 1984 as an improved implementation of the LZ78 algorithm published by Lempel and Ziv in 1978 The algorithm is designed to be fast to implement but is not usually optimal because it performs only limited analysis of the data.

(v) Deflation : Deflation is used in portable network graphics (PNG), MNG and TIFF. It is a lossless data compression algorithm that uses a combination of the LZ77 algorithm and Huffman coding. It was originally defined by Phil Katz for version 2 of his PKZIP archiving tool and was later specified in RFc1951. Deflation is widely thought to be free of any subsisting patents and at a time before the patent on LZW (which is used in the GIF file format) expired, this has led to its use in gzip compressed files and

PNG image files, in addition to the ZIP format for which Katz originally designed it.

Lossy Image Coding Techniques : Typical methods for lossy image compression are as follows :

(i) Color Space Reduction : The main idea of color space reduction is to reduce the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette. This method can be combined with dithering to avoid posterization.

(ii) Chroma Subsampling : Chroma subsampling takes advantage of the fact that the eye perceives spatial changes in brightness more sharply than those in color, by averaging or dropping some of the chrominance information in the image. It is used in many video encoding schemes, both analog and digital, and also in JPEG encoding. Because the human visual system is less sensitive to the position and motion of color than luminance, bandwidth can be optimized by storing more luminance detail than color detail. At normal viewing distances, there is no perceptible loss incurred by sampling the color detail at a lower rate.

(iii) Transform Coding : Transform coding is the most commonly used method. Transform coding is a type of data compression for "natural" data like audio signals or photographic images. The transformation is typically lossy, resulting in a lower quality copy of the original input. A Fourier-related transform such as DCT or the wavelet transform is applied, followed by quantization and entropy coding. In transform coding, knowledge of the application is used to choose information to be discarded, thereby lowering its bandwidth. The remaining information can then be compressed via a variety of methods. When the output is decoded, the result may not be identical to the original input, but it is expected to be close enough for the purpose of the application. The JPEG format is an example of transform coding, one that examines small blocks of the image and "average out" the color using a DCT to form an image with far fewer colors in total.

(iv) Fractal Compression : Fractal compression is a lossy image compression method using fractals to achieve high compression ratios. The method is best suited for photographs of natural scenes such as trees, mountains, ferns and clouds. The fractal compression technique relies on the fact that in certain images parts of the image resemble other parts of the same image. Fractal algorithms convert these parts, or more precisely, geometric shapes into mathematical data called "fractal codes" which are used to re-create the encoded image. Fractal compression differs from pixel-based compression schemes such as JPEG, GIF and MPEG since no pixels are saved. Once an image has been converted into fractal code, its relationship to a specific resolution has been lost and it becomes resolution independent. The image can be re-created to fill any screen size without the introduction of image artifacts or loss of sharpness that occurs in pixel-based compression schemes. With fractal compression, encoding is extremely computationally expensive because of the search used to find the self-similarities. However, decoding is quite fast. At common compression ratios, up to about 50:1,- fractal compression provides similar results to DCT-based algorithms such as JPEG. At high compression ratios, fractal compression may offer superior quality.

The following four subsections focus on two famous block-based lossy image compression schemes and two famous image coding standards.

Q.18 *Write short notes on :*
(i) Optical system models
(ii) Discrete image restoration models

[R.T.U. 2018]

Ans.(i) Optical System Model : Optical systems such as telescopes and microscopes in conjunction with cameras are used to produce images in light-sensitive media such as film or electronic sensors. All the light that is emitted by a particular point on an object and passes through the optical system falls on the corresponding point on the light-sensitive medium. Film records the light intensity at a particular point chemically, while electronic sensors record the intensity digitally. An image captured on film cannot easily be manipulated, but images recorded digitally can easily be post-processed.

Digital raster images are stored as a grid (rows and columns) of numbers. Each number represents the brightness of a given color at one pixel of the image. A pixel is the smallest individual element in an image. For a gray-scale image a raster map is a two-dimensional array of small integers. The largest possible integer in this array determines the gray-level resolution.

Inspect the gray-scale image and the corresponding two-dimensional array!

According to Fourier's theorem any reasonably continuous function defined over some distance L can be synthesized by a sum of harmonic (sine and cosine) functions whose wavelengths are integral submultiples of L, (such as L/2, L/3, ...). Let f(x) be such a function. Then we may write

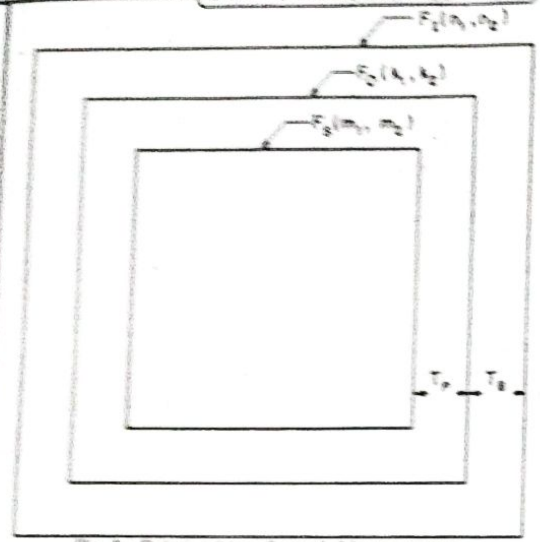


Fig. 2 : Relationships of sampled image arrays.

where the blur matrix B_p contains samples of $P(x, y)$ and B_B contains samples of $B(x, y)$. The nonlinear operation of point-by-point nonlinear transformation defined as. That is

$$f_P(i) = O_P\{f_B(i)\}.$$

single equation for the observed physical image samples in terms of points on the ideal image:

$$f_S = B_P O_P\{B_B f_I\} + B_P n.$$

First, if the point nonlinearity is absent,

$$f_S = Bf_I + n_B$$

where $B = B_P B_B$ and $n_B = B_P n$. This is the classical discrete model consisting of a set of linear equations with measurement uncertainty. Another case that will be defined for later discussion occurs when the spatial blur of the physical image digitizer is negligible. In this case,

$$f_S = O_P\{Bf_I\} + n$$

where $B = B_B$.

Q.19 *Find the Huffman coding for given seven letters along with probability of occurrence*

Letters	*Probability*
B	*0.110*
A	*0.154*
G	*0.011*
D	*0.063*
E	*0.059*
C	*0.072*
F	*0.015*

[R.T.U. 2016]

OR

Explain Huffman coding with example. [R.T.U. 2017]

Ans. Steps for the Huffman Coding :

1. Arrange the letters according to descending order of probability of occurrence.

   A = 0.154
   B=0.110
   C = 0.072
   D = 0.063
   E = 0.059
   F= 0.015
   G = 0.011.

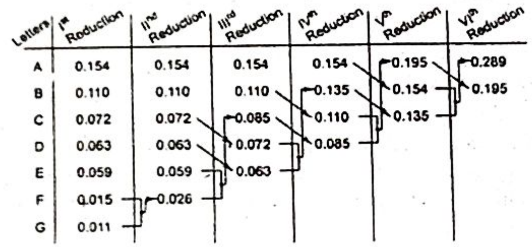
2. Now add lowest two and get new probability (combined of F and G) and place that new probability at appropriate place according to descending order.
3. We left at last only two probabilities because we need binary coding that has only two symbols '1' and '0'.
4. Now start coding: (a) For last two probabilities assign '0' and '1'. (b) Now follow the arrows. The probability of previous step always is related to a probability of next step. Just put the same code for two related (shown by arrow) for both probabilities. (c) Again for last two; put '0' and '1' at prefix. (d) Proceed till last (1"reduction) stage just like that.
5. We will see that we will get more binary digits for less probable data and less binary digits for high probable data.

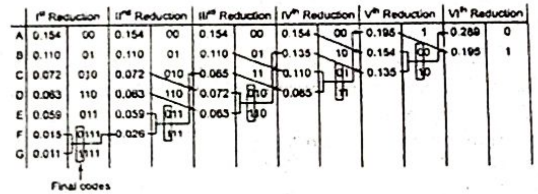
The advantages is that the letter/symbol occurring again and again will now take less bits for representation (and symbol /letter occurring less very take more binary bits) so automatically total required bits will be decreased.

A = 0.154	0.154
B = 0.110	0.110
C = 0.072	0.072
D = 0.063	0.063
E = 0.059	0.059
F = 0.015	0.026
G = 0.011	

We have added lowest two probability (0.015 + 0.011) = 0.026 and placed that in descending order.

(3) Repeat step (2) till only two probabilities are left.





Q.20 *What is Redundancy in Digital Image? Describe various types of Redundancy in Images.* [R.T.U. 2010]

OR

What is Redundancy? Describe various types of Redundancy in digital Image. [R.T.U. 2013]

Ans. Redundancy

Data redundancy is a central issue in digital image compression. It is not an abstract concept but a mathematically quantifiable entity. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (the one characterized by n_1) can be defined as

$$R_D = 1 - \frac{1}{C_R} \quad \text{...(1)}$$

where C_R, commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2} \quad \text{...(2)}$$

For the case $n_2 = n_1$, $C_R = 1$ and $R_D = 0$, indicating that (relative to the second data set) the first representation of the information contains no redundant data. When $n_2 << n_1$, $C_R \rightarrow \infty$ and $R_D \rightarrow 1$, implying significant compression and highly redundant data. Finally, when $n_2 >> n_1$, $C_R \rightarrow 0$ and $R_D \rightarrow -\infty$, indicating that the second data set contains much more data than the original representation. This, of course, is the normally undesirable case of data expansion. In general, C_R and R_D lie in the open intervals $(0, \infty)$ and $(-\infty, 1)$, respectively. A practical compression ratio, such as 10 (or 10:1), means that the

Q.6 ***Explain the technique of thresholding for segmentation.*** **[R.T.U. 2019]**

OR

Explain about thresholding. **[R.T.U. 2017]**

Ans. Thresholding : Thresholding techniques produce segments having pixels with similar intensities. Thresholding is a useful technique for establishing boundaries in images that contain solid objects resting on a contrasting background. There exist a large number of gray-level based segmentation methods using either global or local image information. The thresholding technique requires that an object has homogenous intensity and a background with a different intensity level. Such an image can be segmented into two regions by simple thresholding.

Global Thresholding :

Global thresholding is the simplest and most widely used of all possible segmentation method. In globa thresholding, a threshold value of 0 is chosen and the following condition is imposed:

$$f(m,n) = \begin{cases} 1 & \text{if } f(m,n) \geq 0 \\ 0 & \text{else} \end{cases}$$

Above equation is a complete description of a binarisation algorithm; it contains no indication on how to select the value of the threshold parameter θ. The value of θ has to be selected in an optimal manner. Global thresholding will suffer when pixels from different segments overlap in their use of intensities. If this is due to noise, a technique such as the minimum-error method can estimate the underlying cluster parameters and choose the thresholds to minimise the classification error. If the overlap is due to variation in illumination across the image, variable thresholding could be used. This can be visualised as a form of local segmentation.

Adaptive Thresholding

Global thresholding, or fixed thresholding, works well if the objects of interest have a reasonably uniform interior gray leveland rest on a background of unequal but a relatively uniform gray level. In many cases, the background gray level is not constant, and object contrast varies within an image. In such cases, a threshold that works well in one area might not work well in other areas of the image. In these cases, it is convenient to use a threshold gray level that is a slowly varying function of position in the image.

Histogram-Based Threshold Selection

An image containing an object on a contrasting background has a biomodal gray-level histogram. the two peaks correspond to the relatively large number of points inside and outside the object. The dip between the peaks corresponds to the relatively few points around the edge of the object. This dip is commonly used to establish the threshold gray level. The histogram is the derivative of the area function for an object whose boundary is defined by thresholding:

$$H(D) = -\frac{d}{dD} A(D)$$

where D represents the gray level,A(D) represents the area of the object obtained by thresholding at gray level D, and H(D) represents the histogram.

Increasing the threshold from D to D + ΔD cause only a slight decrease in area if D is near the dip of the histogram. Therefore, placing the threshold at the dip in the histogram minimises the sensitivity of the area measurement to small errors in threshold selection.

If the image containing the object is noisy and not large, the histogram itself will be noisy. Unless the dip is uncommonly sharp, the noise can make its location obscure, or at least unreliable from one image to the next. This effect can be overcome to some extent by smoothing the histogram using either a convolution filter or the curve-fitting procedure.

Limitation of Thresholding Techniques

Thresholding is often used as an initial step in a sequence of image-processing operations. The main limitation of thresholding techniques is that in its simplest form, only two classes are generated and it cannot be applied to multi-channel images. A thresholding technique does not take into account the spatial characteristics of an image. This causes it to be sensitive to noise and intensity inhomogeneities.

Q.7 ***Describe how hough transform used for boundary shape detection.*** [R.T.U. 2019]

OR

Explain hough transforms. [R.T.U. 2017]

Ans. Hough Transform : It is a techniques to isolate the curves of a given shape/shapes in a given image. Classical Hough Transform can locate regular curves like straight lines, circles, parabolas, ellipses, etc.

The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as y = mx + b and can be graphically plotted for each pair of image points (x,y). In the Hough transform, a main idea is to consider the characteristics of the straight line not as image points (x_1,y_1), (x_2,y_2), etc., but instead, in terms of its parameters, i.e., the slope parameter m and the intercept parameter b. Based on that fact, the straight line y = mx + b can be represented as a point (b, m) in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameters m and b. For computational reasons, it is therefore better to use a different pair of parameters, denoted r and θ (theta), for the lines in the Hough transform. These are the Polar Coordinates.

The parameter r represents the distance between the line and the origin, while θ is the angel of the vector from the origin to this closest point (see Coordinates). Using this parameterization, the equation of the line can be written as

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

which can be rearranged to r = x cosθ + y sinθ

It is therefore possible to associate with each line of the image a pair (r, θ) which is unique if θ ∈ [0,π) and r ∈ R, or if θ ∈ [0, 2π) and sometimes r referred to as Hough space for the set of straight lines in two dimensions.

This representation makes the Hough transform conceptually very close to the two-dimensional Radon transform.

For an arbitrary point on the image plane with coordinates, e.g., (x_0, y_0), the lines that go through are

$$r(\theta) = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta,$$

where r (the distance between the line and the origin) is determined by θ,

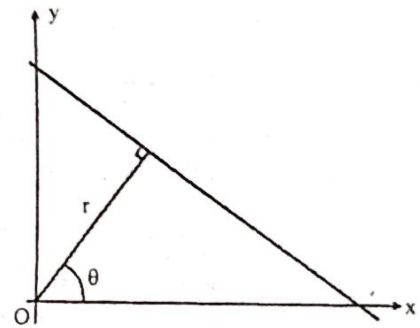


Fig.

This corresponds to a sinusoidal curve in the (r,θ) plane, which is unique to that point. If the curves corresponding to two points are superimposed, the location (in the Hough space) where they cross corresponds to a line (in the original image space) that passes through both points. More generally, a set of points that form a straight line will produce sinusoids which cross at the parameters for that line. Thus, the problem of detecting collinear points can be converted to the problem of finding concurrent curves.

Q.8 ***Explain the concept of hit or miss transformation.*** [R.T.U. 2018]

Ans. Hit or Miss Transformations : The two basic morphological operations, dilation and erosion, plus many variants can be defined and implemented by a hit-or-miss transformation. The concept is quite simple. Conceptually, a small odd-sized mask, typically 3 ×3 , is scanned over a binary image. If the binary-valued pattern of the mask matches the state of the pixels under the mask (hit), an output pixel in spatial correspondence to the center pixel of the mask is set to some desired binary state. For a pattern mismatch (miss), the output pixel is set to the opposite binary state. For example, to perform simple binary noise cleaning, if the isolated 3 ×3 pixel pattern

0	0	0
0	1	0
0	0	0

3. Single response : There should be only a single response to a true edge. The distance between peaks of the gradient when only noise is present, denoted as x_m, is set to some fraction k of the operator width factor W. Thus

$$x_m = kW$$

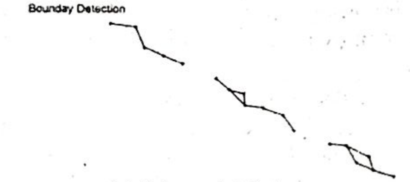
Canny has combined these three criteria by maximizing the product of SNR and LOC subject to the constraint. Because of the complexity of the formulation, no analytic solution has been found, but a variational approach has been developed.

Q.11 ***Write a short note on EDGE linking.*** **[R.T.U. 2016]**

OR

Write a short note on heuristic edge linking methods.

Ans. The edge segmentation technique developed by Roberts is typical of the philosophy of many heuristic edge-linking methods. In Robert's method, edge gradients are examined in 4 x 4 pixels blocks.



(a) Edge point linkages

(b) Elimination of multiple linkages and bridging

Fig.1 : Roberts edge linking

The pixel whose magnitude gradient is largest is declared a tentative edge point if its magnitude is greater than a threshold value. Then north, east, south and west-oriented lines of length 5 are fitted to the gradient data about the tentative edge point. If the ratio of the best fit to the worst fit, measured in terms of the fit correlation is greater than a second threshold, the tentative edge point is declared valid and it is assigned the direction of the best fit. Next, straight lines are fitted between pairs of edge points if they are in adjacent 4 × 4 blocks and if the line direction is within ±23 degrees of the edge direction of either edge point. Those points failing to meet the linking criteria are discarded. A typical boundary at this stage, shown in figure 1(a) will contain gaps and multiply connected edge points. Small triangles are eliminated by deleting the longest side. Small rectangles are replaced by their longest diagonal, as indicated in figure 1(b). Short spur lines are also deleted. At this stage, short gaps are bridged by straight-line connection. This form of edge linking can be used with a wide variety of edge detectors.

Robinson has suggested a simple but effective edge-linking algorithm in which edge points from an edge detector providing eight edge compass directions are examined in 3 × 3 blocks as indicated in figure 2. The edge point in the center of the block is declared a valid edge if it possesses directional neighbors in the proper orientation. Extensions to larger windows should be beneficial but the number of potential valid edge connections will grow rapidly with window size.

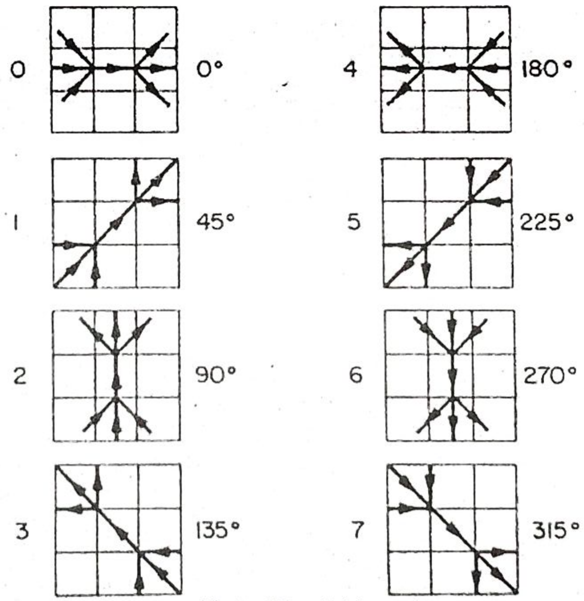
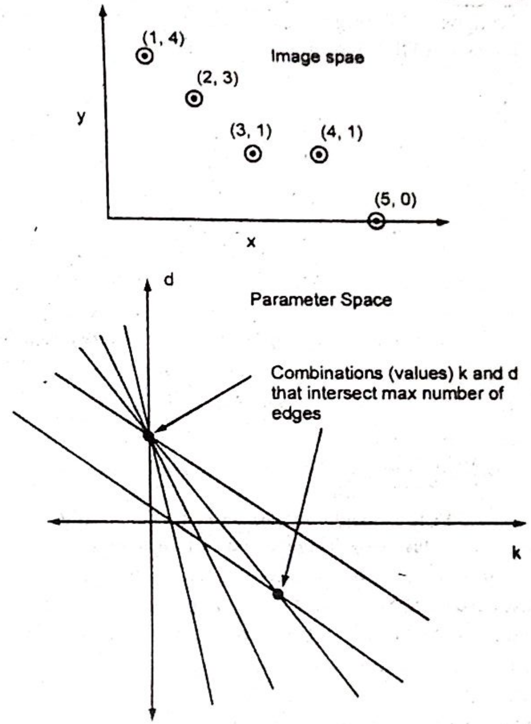


Fig.2 : Edge linking rules

Q.12 ***Given 5 points, use Hough transform to draw a line joining these points (1, 4), (2, 3) (3, 1), (4, 1), (5, 0).*** **[R.T.U. 2016]**

Ans. Each of the given point corresponds to the following line that can be plotted

(1, 4) → d = –k + 4
(2, 3) → d = –2k + 3
(3, 1) → d = –3k + 1
(4, 1) → d = –4k + 1
(5, 0) → d = –5k



After finding optimal values of k and d, we can draw them as lines in image space.

Q.13 ***What is clustering segmentation ? How it is different from region segmentation.*** **[R.T.U. 2012]**

Ans. Consider a vector $x = [x_1, x_2, \ldots, x_N]^T$ of measurements at each pixel coordinate (j, k) in an image. The measurements could be point multispectral values, point color components or derived color components or they could be neighborhood feature measurements such as the moving window mean, standard deviation and mode. If the measurement set is to be effective for image segmentation, data collected at various pixels within a segment of common attribute should be similar. That is, the data should be tightly clustered in an N-dimensional measurement space. If this condition holds, the segmenter design task becomes one of subdividing the N-dimensional measurement space into mutually exclusive compartments, each of which envelops typical data clusters for each image segment. Figure 1 illustrates the concept for two features. In the segmentation process, if a measurement vector for a pixel falls within a measurement space compartment, the pixel is assigned the segment name or label of that compartment. Figure 2 is a flowchart that describes a simplified version of

PART-C

Q.15 *Describe the segmentation process in digital image processing. Explain the fundamental of edge based segmentation.* **[R.T.U. 2019]**

Ans. Segmentation : It is the process of dividing an image into its constituent parts. The goal of segmentation is to partition the object of interest according to the requirement of specific application.

Image Segmentation : In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like Marching cubes.

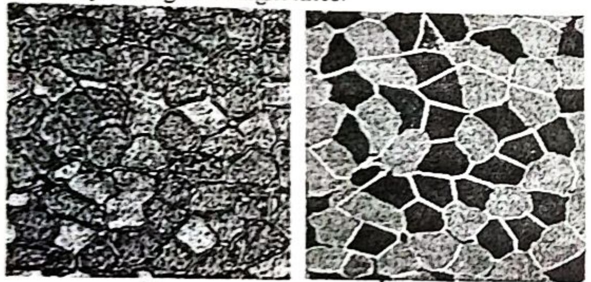
Some of the practical applications of image segmentation are:

- Content-based image retrieval
- Machine vision
- Medical imaging
- Locate tumors and other pathologies
- Measure tissue volumes
- Diagnosis, study of anatomical structure
- Surgery planning
- Virtual surgery simulation
- Intra-surgery navigation
- Object detection
- Pedestrian detection
- Face detection
- Brake light detection
- Locate objects in satellite images (roads, forests, crops, etc.)

- Recognition Tasks
- Face recognition
- Fingerprint recognition
- Iris recognition
- Traffic control systems
- Video surveillance

Several general-purpose algorithms and techniques have been developed for image segmentation. To be useful, these techniques must typically be combined with a domain's specific knowledge in order to effectively solve the domain's segmentation problems.

Edge-based Segmentation : The results of threshold-based segmentation are usually less than perfect. Often, a scientist will have to make changes to the results of automatic segmentation. One simple way of doing this is by using a computer mouse to control a screen cursor and draw boundary lines between regions. Fig.1(a) shows the boundaries obtained by thresholding the muscle fibres image with the contrast stretched so that values between 0 and 5 are displayed as shades of grey ranging from white to black and values exceeding 5 are all displayed as black. This display can be used as an aid to determine where extra boundaries need to be inserted to fully segment all muscle fibres. Fig.1(b) shows the result after manually adding 71 straight lines.

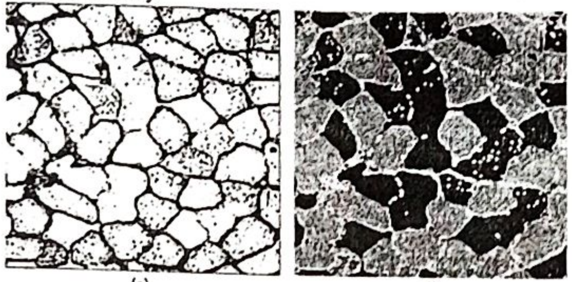


(a) (b)

Fig. 1 : (a) Boundaries obtained by thresholding the muscle fibres image, superimposed on output for Prewitt's filter, with values between 0 and 5 displayed in progressively darker shades of grey and values in excess of 5 displayed as black. (b) Manual segmentation of the image by addition of extra lines to boundaries obtained by thresholding, superimposed in white on the original image.

Algorithms are available for semi-automatically drawing edges, whereby the scientist's rough lines are smoothed and perturbed to maximize some criterion of match with the image. Alternatively, edge finding can be made fully automatic, although not necessarily fully successful. Fig.2(a) shows the result of applying Prewitt's edge filter to the muscle fibre image. In the display, the filter output has been thresholded at a value of 5; all pixels exceeding 5 are labelled as edge pixels and displayed as black. Connected chains of edge pixels divide the image

into regions. Segmentation can be achieved by allocating to a single category all non-edge pixels which are not separated by an edge. Rosenfeld and Pfaltz (1966) gave an efficient algorithm for doing this for 4 and 8-connected regions, termed a connected components algorithm. We will describe the algorithm in words and then mathematically.



(a) (b)

Fig. 2 : (a) Thresholded output from Prewitt's edge filter applied to muscle fibres image: values greater than 5 are displayed as black, those less than or equal to 5 as white. (b) Boundaries produced from connected regions in (a), superimposed in white on the original image.

The algorithm operates on a raster scan, in which each pixel is visited in turn, starting at the top-left corner of the image and scanning along each row, finishing at the bottom-right corner. For each non-edge pixel, (i,j) the following conditions are checked. If its already-visited neighbours – (i – 1, j) and (i, j – 1) in the 4-connnected case, also (i – 1, j + 1) and (i – 1, j + 1) in the 8-connected case are all edge pixels, then a new category is created and (i,j) is allocated to it. Alternatively, if all its non-edge neighbours are in a single category, then (i,j) is also placed in that category. The final possibility is that neighbours belong to two or more categories, in which case (i,j) is allocated to one of them and a note is kept that these categories are connected and therefore should from then on be considered as a single category. More formally, for the simpler case of 4-connected regions:

- Initialize the count of the number of categories by setting K = 0.
- Consider each pixel (i,j) in turn in a raster scan, proceeding row by row (i=1,.....n), and for each value of i taking j=1,......n.
- One of four possibilities apply to pixel (i,j):

1. If (i,j) is an edge pixel then nothing needs to be done.
2. If both previously-visited neighbours, (i – 1,j) and (i, j – 1), are edge pixels, then a new category has to be created for (i,j):

$$K \rightarrow K+1,\ h_k = K,\ g_{ij} = K,$$

Where the entries in $h_1,\ldots\ldots,h_k$ are used to keep track of which categories are equivalent, and g_{ij} records the category label for pixel (i,j).

3. If just one of the two neighbours is an edge pixel, then (i,j) is assigned the same label as the other one :

$$g_{ij} = \begin{cases} g_{i-1,j} & \text{If } (i, j-1) \text{ is the edge pixel,} \\ g_{i,j-1} & \text{otherwise.} \end{cases}$$

4. The final possibility is that neither neighbour is an edge pixel, in which case (i,j) is given the same label as one of them :

$$g_{ij} = g_{i-1,j}$$

and if the neighbours have labels which have not been marked as equivalent, i.e. $h_{g_{i-1,j}} \neq h_{g_{i,j-1}}$, then this needs to be done (because they are connected at pixel (i,j)). The equivalence is recorded by changing the entries in $h_1,\ldots,h_k$, as follows:

- Set $l_1 = \min(h_{g_{i-1,j}}, h_{g_{i,j-1}})$
  and $l_2 = \max(h_{g_{i-1,j}}, h_{g_{i,j-1}})$.
- For each value of k from 1 to K,
  if $h_k = l_2$ then $h_k = l_1$

Finally, after all the pixels have been considered, the array of labels is revised, taking into account which categories have been marked for amalgamation:

After application of the labelling algorithm, superfluous edge pixels – that is, those which do not separate classes – can be removed: any edge-pixel which has neighbours only of one category is assigned to that category.

Fig. 2(b) shows the result of applying the labelling algorithm with edges as shown in Fig. 2(a), and removing superfluous edge pixels. The white boundaries have been superimposed on the original image. Similarly, small segments (say less than 500 pixels in size) which do not touch the borders of the image can be removed. The segmentation has done better than simple thresholding, but has failed to separate all fibres because of gaps in output from Prewitt's edge filter. Martelli (1976) among others, has proposed algorithms for bridging these gaps.

Another edge-detection filter the Laplacian-of-Gaussian. This filter can also be used to segment images, using the zero-crossings of the output to specify positions of boundaries. One advantage over Prewitt's filter is that the zero-crossings always form closed boundaries. Fig. 3(a) shows output from the Laplacian-of-Gaussian filter ($\sigma^2 = 27$), applied to the muscle fibres image. Boundaries corresponding to weak edges can be suppressed by applying a threshold to the average gradient strength around a boundary. Fig. 3(b) shows zero-crossing of the Laplacian-of-Gaussian filter with average gradients exceeding unity, superimposed on the original image. In this application the result can be seen to be little better than simple thresholding.

ive not
$h_{gi, j-1}$,
ey are
corded
vs:

dered,
which

ithm,
o not
which
o that

elling
oving
been
small
o not
The
ding,
ps in
nong
gaps.
n-of-
ages,
tions
s that
. Fig.
filter
laries
ed by
round
f the
lients
ge. In
better

Fig. 3 : (a) Output form Laplacian-of-Gaussian filter ($\sigma^2 = 27$) applied to muscle fibres image (b) Zero-crossings from (a) with average image gradients in excess of 1.0, in white superimposed on the image.

Q.16 *Explain the region growing method for segmentation in image processing.* **[R.T.U. 2019]**

OR

Explain region based segmentation with suitable example. **[R.T.U. 2017, 2016]**

OR

Explain in brief various region segmentation methods. **[R.T.U. 2018]**

OR

What are Region Segmentation Methods. Explain with example. **[R.T.U. 2013]**

OR

Explain the region growing approach for image segmentation.

Ans. Region Growing : Region growing is one of the conceptually simplest approaches to image segmentation; neighboring pixels of similar amplitude are grouped together to form a segmented region. However, in practice, constraints, some of which are reasonably complex, must be placed on the growth pattern to achieve acceptable results.

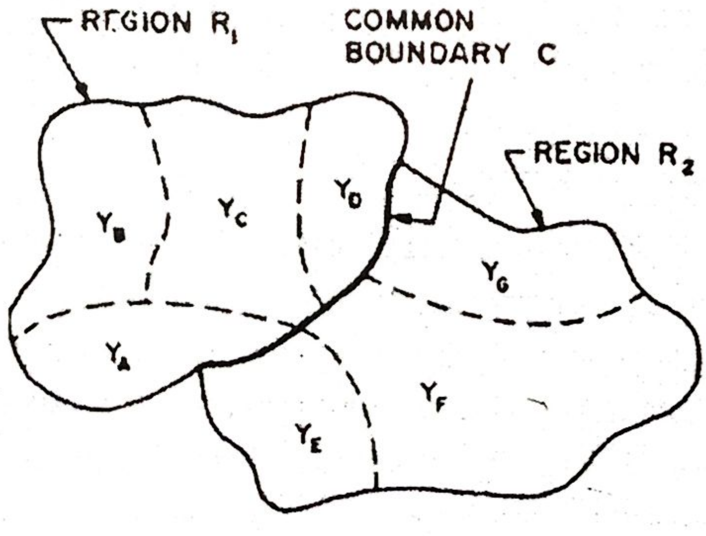


Fig. : Region-growing geometry

Brice and Fenema have developed a region-growing method based on a set of simple growth rules. In the first stage of the process, pairs of quantized pixels are combined together in groups called atomic regions if they are of the same amplitude and are four-connected. Two heuristic rules are next invoked to dissolve weak boundaries between atomic boundaries. Referring to Figure. let R_1 and R_2 be two adjacent regions with perimeters P_1 and P_2 respectively, which have previously been merged.

After the initial stages of region growing, a region may contain previously merged subregions of different amplitude values. Also, let C denote the length of the common boundary and let D represent the length of that portion of C for which the amplitude difference Y across the boundary is smaller than a significance factor ε_1. The regions R_1 and R_2 are then merged if

$$\frac{D}{MIN\{P_1, P_2\}} > \varepsilon_2 \quad ...(1)$$

where ε_2 is a constant typically set at $\varepsilon_2 = \frac{1}{2}$. This heuristic prevents merger of adjacent regions of the same approximate size, but permits smaller regions to be absorbed into larger regions. The second rule merges weak common boundaries remaining after application of the first rule. Adjacent regions are merged if

$$\frac{D}{C} > \varepsilon_3 \quad ...(2)$$

where ε_3 is a constant set at about $\varepsilon_3 = \frac{3}{4}$. Application of only the second rule tends to overmerge regions.

The Brice and Fenema region growing method provides reasonably accurate segmentation of simple scenes with few objects and little texture but does not perform well on more complex scenes. Yakimovsky has attempted to improve the region-growing concept by establishing merging constraints based on estimated Bayesian probability densities of feature measurements of each region.

1. Split and Merge

Split and merge image segmentation techniques are based on a quad tree data representation whereby a square image segment is broken (split) into four quadrants if the original image segment is nonuniform in attribute. If four neighboring squares are found to be uniform, they are replaced (merge) by a single square composed of the four adjacent squares.

In principle, the split and merge process could start at the full image level and initiate split operations. This approach tends to be computationally intensive. Conversely. beginning at the individual pixel level and making initial merges has the drawback that region uniformity measures are limited at the single pixel level. Initializing the split and merge process at an intermediate level enables the use of more powerful uniformity tests without excessive computation.

The simplest uniformity measure is to compute the difference between the largest and smallest pixels of a segment. Fukada has proposed the segment variance as a uniformity measure. Chen and Pavlidis suggest more complex statistical measures of uniformity. The basic split and merge process tends to produce rather blocky segments because of the rule that square blocks are either split or merged. Horowitz and Pavlidis have proposed a modification of the basic process whereby adjacent pairs of regions are merged if they are sufficiently uniform.

2. Watershed

Topographic and hydrology concepts have proved useful in the development of region segmentation methods. In this context, a monochrome image is considered to be an altitude surface in which high-amplitude pixels correspond to ridge points, and low-amplitude pixels correspond to valley points. If a drop of water were to fall on any point of the altitude surface, it would move to a lower altitude until it reached a local altitude minimum.

The accumulation of water in the vicinity of a local minimum is called a catchment basin. All points that drain into a common catchment basin are part of the same watershed. A valley is a region that is surrounded by a ridge. A ridge is the loci of maximum gradient of the altitude surface. There are two basic algorithmic approaches to the computation of the watershed of an image: rainfall and flooding.

In the rainfall approach, local minima are found throughout the image, Each local minima is given a unique tag. Adjacent local minima are combined with a unique tag. Next, a conceptual water drop is placed at each untagged pixel. The drop moves to its lower-amplitude neighbor until it reaches a tagged pixel, at which time it assumes the tag value. Figure illustrates a section of a digital image encompassing a watershed in which the local minimum pixel is black and the dashed line indicates the path of a water drop to the local minimum.

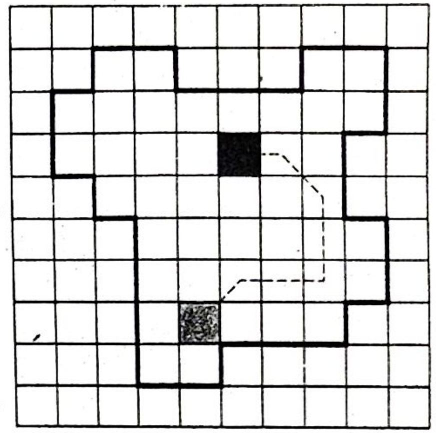


Fig. : Rainfall watershed.

In the flooding approach, conceptual single pixel holes are pierced at each local minima, and the amplitude surface is lowered into a large body of water. The water enters the holes and proceeds to till each catchment basin. If a basin is about to overflow. a conceptual dam is built on its surrounding ridge line to a height equal to the highest-altitude ridge point.

Q.17 ***Discuss gradient operators. Write 3 × 3 region two dimensional sobel mask and express their partial derivative equations.*** **[R.T.U. 2019]**

OR

Explain first order derivative edge detection and second order derivative edge detection. **[R.T.U. 2013]**

OR

Explain edge detection in detail. **[R.T.U. 2017]**

Ans. Gradient Operators : The gradient of an image f(x,y) at the location (x,y) is given by the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad ...(i)$$

The gradient vector points in the direction of the maximum rate of change of f at (x,y). In the edge detection we employ the magnitude of the gradient vector and it is denoted as

$$\nabla f = \text{mag}(\nabla f) = \left[G_x^2 + G_y^2\right]^{\frac{1}{2}} \quad ...(ii)$$

magn
given
impor
axis.
obtain
pixel i
derivat
equiva
operat
And
shown
two 3
P_1
P_4
P_7
Fi
center
used to
also be
in fig.
of G_x a