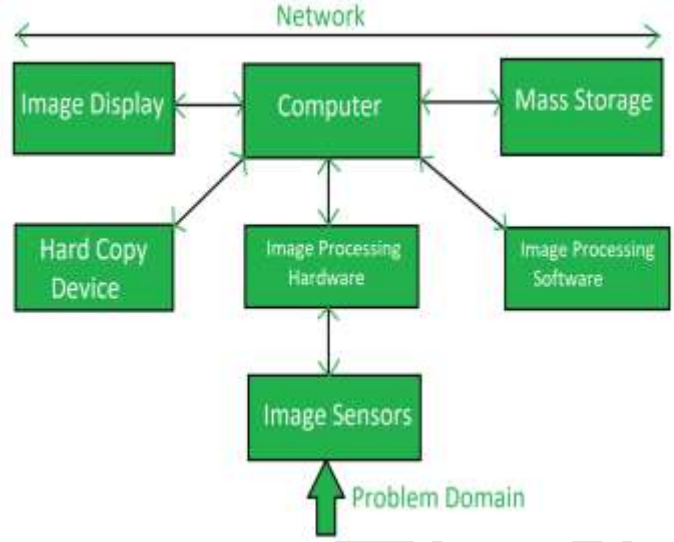
MID-1 Questions
SET-1

1) a) What are the components of an Image Processing System? Explain CO1 L1 6M

Ans Components of Image Processing System

Image Processing System is the combination of the different elements involved in the digital image processing. Digital image processing is the processing of an image by means of a digital computer. Digital image processing uses different computer algorithms to perform image processing on the digital images.
It consists of following components:-



- **Image Sensors:**
  Image sensors senses the intensity, amplitude, co-ordinates and other features of the images and passes the result to the image processing hardware. It includes the problem domain.
- **Image Processing Hardware:**
  Image processing hardware is the dedicated hardware that is used to process the instructions obtained from the image sensors. It passes the result to general purpose computer.
- **Computer:**
  Computer used in the image processing system is the general purpose computer that is used by us in our daily life.
- **Image Processing Software:**
  Image processing software is the software that includes all the mechanisms and algorithms that are used in image processing system.
- **Mass Storage:**
  Mass storage stores the pixels of the images during the processing.
- **Hard Copy Device:**
  Once the image is processed then it is stored in the hard copy device. It can be a pen drive or any external ROM device.
- **Image Display:**
  It includes the monitor or display screen that displays the processed images.
- **Network:**
  Network is the connection of all the above elements of the image processing system.

b) Explain the Adjacency between pixels CO1 L2 6M

Ans

Adjacency and Connectivity

• Let V: a set of intensity values used to define adjacency and connectivity.

• In a binary image, V = {1}, if we are referring to adjacency of pixels with value 1.

• In a gray-scale image, the idea is the same, but V typically contains more elements, for example, V = {180, 181, 182, …, 200}

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

Types of Adjacency

1.4-adjacency: Two pixels p and q with values from V are 4-adjacent if q is in the set N4(p).

2.8-adjacency: Two pixels p and q with values from V are 8-adjacent if q is in the set N8(p).

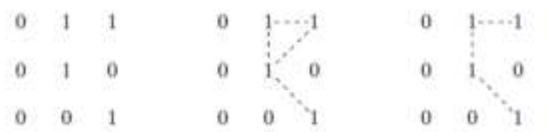
3. m-adjacency =(mixed)

• m-adjacency: Two pixels p and q with values from V are m-adjacent if :

- • q is in N4(p) or
- • q is in ND(p) and the set N4(p) ∩ N4(q) has no pixel whose values are from V (no intersection)

• Mixed adjacency is a modification of 8- adjacency. It is introduced to eliminate the ambiguities that often arise when 8- adjacency is used.

• For example:



a b c

FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

• In this example, we can note that to connect between two pixels (finding a path between two pixels): – In 8-adjacency way, you can find multiple paths between two pixels – While, in m-adjacency, you can find only one path between two pixels

• So, m-adjacency has eliminated the multiple path connection that has been generated by the 8-adjacency.

• Two subsets S1 and S2 are adjacent, if some pixel in S1 is adjacent to some pixel in S2. Adjacent means, either 4-, 8- or m-adjacency.

(Or)

2) a) Explain in detail about image acquisition system **CO1** **L2** **6M**

Ans

Image Acquisition in Digital Image Processing

Digital Image Processing

In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed.

Now the incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to a particular type of energy being detected. The output voltage waveform is the response of the sensor(s) and a digital quantity is obtained from each sensor by digitizing its response.

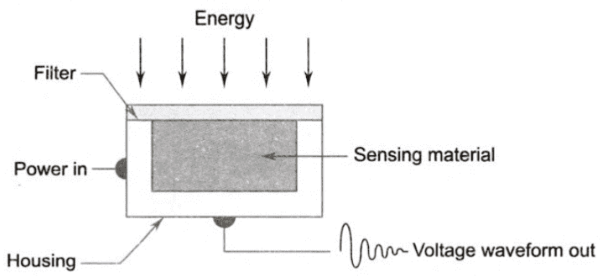


Fig: Single image sensor

Fig: Line sensor

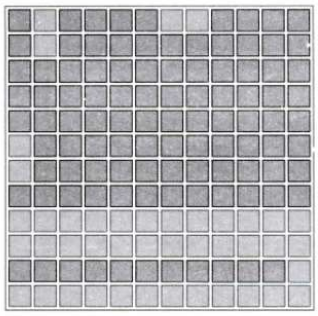


Fig: Array sensor

Image Acquisition using a single sensor:

Example of a single sensor is a photodiode. Now to obtain a two-dimensional image using a single sensor, the motion should be in both x and y directions.

- Rotation provides motion in one direction.
- Linear motion provides motion in the perpendicular direction.

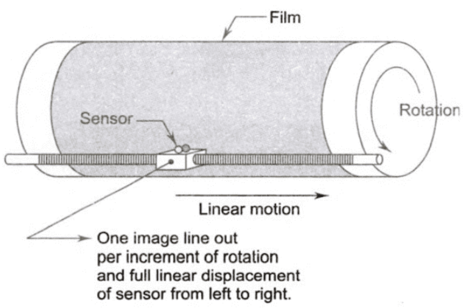


Fig: Combining a single sensor with motion to generate a 2D image

This is an inexpensive method and we can obtain high-resolution images with high precision control. But the downside of this method is that it is slow.

Image Acquisition using a line sensor (sensor strips):

- The sensor strip provides imaging in one direction.

- Motion perpendicular to the strip provides imaging in other direction.

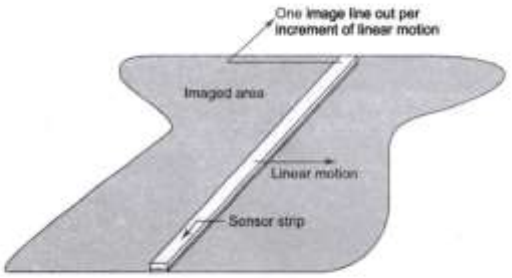


Fig: Linear sensor strip

Image Acquisition using an array sensor:

In this, individual sensors are arranged in the form of a 2-D array. This type of arrangement is found in digital cameras. e.g. CCD array

In this, the response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor. Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or ever hours.

Advantage: Since sensor array is 2D, a complete image can be obtained by focusing the energy pattern onto the surface of the array.

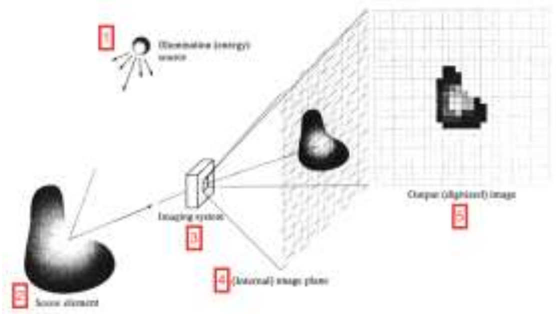


Fig: An example of digital image acquisition using array sensor

The sensor array is coincident with the focal plane, it produces an output proportional to the integral of light received at each sensor.

Digital and analog circuitry sweep these outputs and convert them to a video signal which is then digitized by another section of the imaging system. The output is a digital image.

b) Illustrate how the image is digitized by sampling and quantization process CO1 L2 6M
Ans
Sampling and Quantization in Digital Image Processing

Digital Image Processing

Need of Sampling and Quantization in Digital Image Processing:
Mostly the output of image sensors is in the form of analog signal. Now the problem is that we cannot apply digital image processing and its techniques on analog signals.

This is due to the fact that we cannot store the output of image sensors which are in the form of analog signals because it requires infinite memory to store a signal that can have infinite values. So we have to convert this analog signal into digital signal.

To create a digital image, we need to convert the continuous data into digital form. This conversion from analog to digital involves two processes: sampling and quantization.

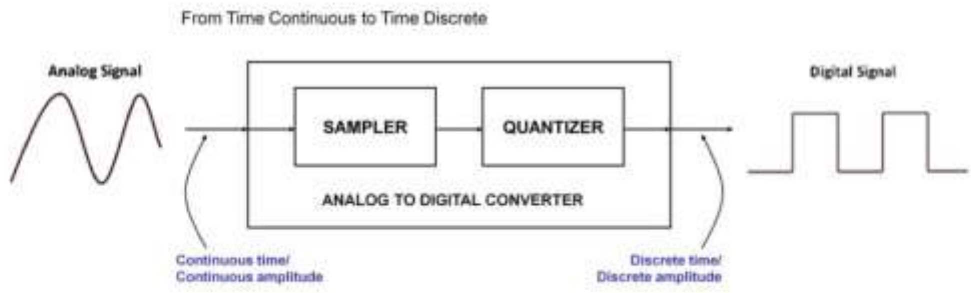
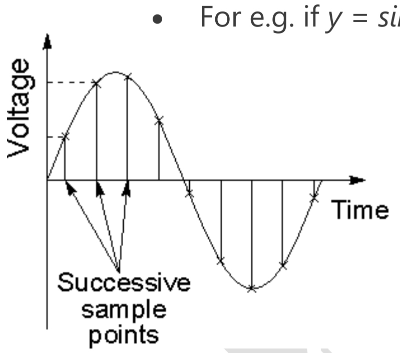


Fig: Analog to Digital Conversion

Sampling -> digitization of coordinate values
Quantization -> digitization of amplitude values

Sampling in Digital Image Processing:

- In this we digitize x-axis in sampling.
- It is done on the independent variable.
- For e.g. if $y = sinx$, it is done on x variable.



- There are some variations in the sampled signal which are random in nature. These variations are due to noise.
- We can reduce this noise by more taking samples. More samples refer to collecting more data i.e. more pixels (in case of an image) which will eventually result in better image quality with less noise present.
- As we know that pixel is the smallest element in an image and for an image represented in the form of a matrix, total no. of pixels is given by:

Total number of pixels = Total number of rows X Total number of columns

- The number of samples taken on the x-axis of a continuous signal refers to the number of pixels of that image.

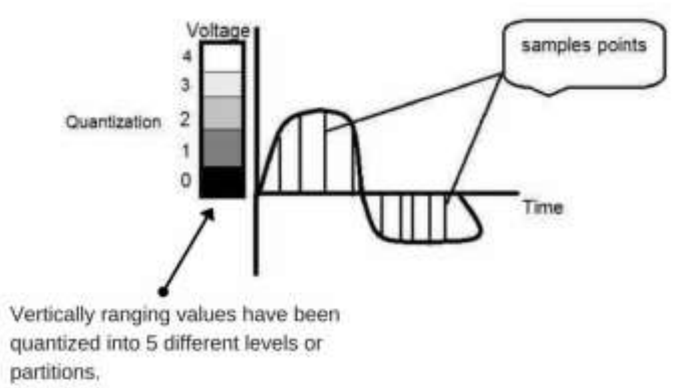
- ***For a CCD array***, if the number of sensors on a CCD array is equal to the number of pixels and number of pixels is equal to the number of samples taken, therefore we can say that number of samples taken is equal to the number of sensors on a CCD array.

No. of sensors on a CCD array = No. of pixels = No. of samples taken

- Oversampling is used for zooming. The difference between sampling and zooming is that sampling is done on signals while zooming is done on the digital image.

Quantization in Digital Image Processing:

- It is opposite of sampling as sampling is done on the x-axis, while quantization is done on the y-axis.
- Digitizing the amplitudes is quantization. In this, we divide the signal amplitude into quanta (partitions).



Relation of Quantization and gray level resolution:

Number of quantas (partitions) = Number of gray levels

- Number of gray levels here means number of different shades of gray.
- To improve image quality, we number of gray levels or gray level resolution up.
- If we increase this level to 256, it is known as the grayscale image.

Where,
L = gray level resolution
k = gray level

Gray level = number of bits per pixel (BPP) = number of levels per pixel

3) a) Derive the basis function of Walsh transform N = 4 **CO2 L3 6M**

Ans

The Walsh transform is an orthogonal transform similar to the Fourier transform but uses the Walsh functions as the basis functions. These functions take values from a discrete set, typically

$\{-1,1\}$and are useful in various digital signal processing applications due to their simplicity and efficiency.

For N=4, the basis functions of the Walsh transform can be derived using the Hadamard matrix. The Walsh functions are closely related to the rows of the Hadamard matrix.

Walsh Functions : The Walsh functions can be derived from the Hadamard matrix by permuting its rows according to the ordering sequence.
The two most common orderings are the Hadamard order and the sequency order.

In Hadamard order, the rows of the Hadamard matrix are used directly as the Walsh functions.

Ex:

for N=4

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

In sequency order, the rows of the Hadamard matrix are permuted so that the number of sign changes per row (sequency) increases.

Let us take row 2 in matrix 1 -1 1 -1 the sign changes from 3 times [1→(-1)→1→(-1)]
So it will be given 3rd row in sequency order matrix

Thus, the sequency ordered Walsh matrix is

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

b) Define DFT. State and prove the Periodicity property **CO2 L2 6M**

Ans

The **Discrete Fourier Transform (DFT)** is a mathematical technique used to transform a discrete sequence of complex numbers into another sequence of complex numbers, which represents the frequency components of the original sequence. The DFT is widely used in signal processing and image analysis.

For a sequence x[n] of length N, the DFT is defined as:

$$X(k) = \sum_{k=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad 0 \le k \le N-1$$

X(k) is periodic with period N i.e., X(k+N) = X(k).

Ex: for N=4

F=[1 1 1 1

1 -j -1 1

1 -1 1 -1

1   j   -1 -j]

PERIODICITY

Statement: If a sequence x(n) is periodic with periodicity of N samples then N-point DFT, X(K) is also periodic with a periodicity of N Samples

Let   x(n)=x(n+N) for all n

x(K)=x(K+N) for all K

PERIODICITY

Proof: By definition of DFT, the(K+N)th coefficient of X(K) is given by,

$$X(K+N)=\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi(k+N)n/}N$$

$$=\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/}N . e^{-j2\pi Nn/}N$$

$$=\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/}N.1 \qquad [e^{-j2\pi n} = 1]$$

X(K+N)=X(K)

(Or)

4) a) Derive the basis function of Walsh Transform CO2 L2 6M

Ans same as previous one

b) Discuss about Discrete Cosine Transform (DCT) with necessary mathematical expressions

CO2 L2 6M

Ans

The Discrete Cosine Transform (DCT) is a widely used transform in signal processing and image compression, particularly in JPEG image compression. The DCT converts a sequence of data into a sum of cosine functions oscillating at different frequencies. It is similar in spirit to the Discrete Fourier Transform (DFT) but uses only real-valued functions (cosines) instead of complex exponentials.

Mathematical Definition

The DCT is defined as follows:

1. **DCT Formula**

For a sequence $x[n]$ of length N, the DCT of $x[n]$ is given by:

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

where:

- k is the index of the DCT coefficient (ranging from 0 to $N-1$).
- $x[n]$ is the input sequence.
- $X[k]$ is the DCT coefficient corresponding to frequency k.
- α_k is a normalization factor defined as:

$$\alpha_k = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k=0 \\ \sqrt{\frac{2}{N}} & \text{for } k>0 \end{cases}$$

The inverse DCT (IDCT) reconstructs the original sequence from its DCT coefficients and is defined as:

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

Types of DCT

There are several variants of the DCT, with the most commonly used ones being:

- **DCT-I**: This variant is less commonly used in practice and has the following formula:

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

- **DCT-II**: This is the most widely used DCT in practice, especially in image and video compression. The formula is:

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

  The IDCT for DCT-II is:

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

- **DCT-III**: This is the inverse of DCT-II and is sometimes used in signal processing:

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

- **DCT-IV**: Less common, used in certain applications:

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

5) Explain Histogram equalization with an example CO3 L2 6M

Ans

Histogram equalization is a technique used in image processing to enhance the contrast of an image. It works by redistributing the image's intensity levels so that the histogram of the output image is approximately uniformly distributed. This technique is particularly useful for improving the visibility of features in images with poor contrast.

Steps in Histogram Equalization

1. **Compute the Histogram**: Determine the frequency of each intensity level in the original image.
2. **Compute the Cumulative Distribution Function (CDF)**: Calculate the cumulative sum of the histogram values, normalized to the range [0, 1].
3. **Normalize the CDF**: Scale the CDF to cover the entire range of pixel values (e.g., [0, 255] for 8-bit images).
4. **Map the Original Intensities**: Replace each pixel intensity in the original image with its corresponding value from the normalized CDF.
5. **Generate the Equalized Image**: Create the new image using the mapped intensities.

Example of Histogram Equalization

Consider an 8-bit grayscale image with the following pixel intensity levels and their corresponding frequencies:

- Intensity Level: 0, 1, 2, 3, 4, 5, 6, 7
- Frequency: 5, 20, 10, 15, 10, 10, 15, 15

1. Compute the Histogram

The histogram is:

Histogram=[5,20,10,15,10,10,15,15]\text{Histogram} = [5, 20, 10, 15, 10, 10, 15, 15]Histogram=[5,20,10,15,10,10,15,15]

2. Compute the Cumulative Distribution Function (CDF)

First, compute the cumulative sum of the histogram:

CDF=[5,25,35,50,60,70,85,100]\text{CDF} = [5, 25, 35, 50, 60, 70, 85, 100]CDF=[5,25,35,50,60,70,85,100]

Normalize the CDF by dividing each value by the total number of pixels (which is 100 in this case):

Normalized CDF=CDFTotal Pixels=[0.05,0.25,0.35,0.50,0.60,0.70,0.85,1.00]\text{Normalized CDF} = \frac{\text{CDF}}{\text{Total Pixels}} = [0.05, 0.25, 0.35, 0.50, 0.60, 0.70, 0.85, 1.00]Normalized CDF=Total PixelsCDF=[0.05,0.25,0.35,0.50,0.60,0.70,0.85,1.00]

3. Normalize the CDF to the Range [0, 255]

Multiply the normalized CDF by the maximum intensity value (255 for 8-bit images):

Equalized CDF=[12.75,63.75,89.25,127.50,153.00,178.50,216.75,255.00]\text{Equalized CDF} = [12.75, 63.75, 89.25, 127.50, 153.00, 178.50, 216.75, 255.00]Equalized CDF=[12.75,63.75,89.25,127.50,153.00,178.50,216.75,255.00]

Round these values to the nearest integer to get:

Equalized CDF=[13,64,89,128,153,179,217,255]\text{Equalized CDF} = [13, 64, 89, 128, 153, 179, 217, 255]Equalized CDF=[13,64,89,128,153,179,217,255]

4. Map the Original Intensities

Replace each pixel intensity in the original image with its corresponding value from the equalized CDF. For example:

- Original Intensity Level 0 maps to 13
- Original Intensity Level 1 maps to 64
- Original Intensity Level 2 maps to 89
- And so on.

5. Generate the Equalized Image

Create the equalized image using the mapped values. If a pixel in the original image had an intensity of 0, it would now have an intensity of 13 in the equalized image. Similarly, an intensity of 1 would be mapped to 64, and so forth.

(orr)

Histogram Equalization:

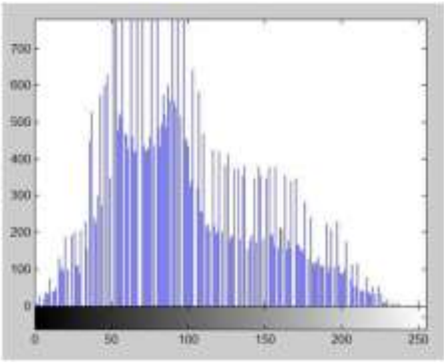
Histogram equalization is used to enhance contrast. It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased.

Lets start histogram equalization by taking this image below as a simple image.

Image

Histogram of this image:

The histogram of this image has been shown below.



Now we will perform histogram equalization to it.

PMF:

First we have to calculate the PMF (probability mass function) of all the pixels in this image. If you donot know how to calculate PMF, please visit our tutorial of PMF calculation.

CDF:

Our next step involves calculation of CDF (cumulative distributive function). Again if you donot know how to calculate CDF , please visit our tutorial of CDF calculation.

Calculate CDF according to gray levels

Lets for instance consider this , that the CDF calculated in the second step looks like this.

Gray Level Value	CDF
0	0.11
1	0.22
2	0.55
3	0.66
4	0.77
5	0.88
6	0.99
7	1

Then in this step you will multiply the CDF value with (Gray levels (minus) 1) .

Considering we have an 3 bpp image. Then number of levels we have are 8. And 1 subtracts 8 is 7. So we multiply CDF by 7. Here what we got after multiplying.

Gray Level Value	CDF	CDF * (Levels-1)
0	0.11	0
1	0.22	1
2	0.55	3
3	0.66	4
4	0.77	5
5	0.88	6
6	0.99	6
7	1	7

Now we have is the last step , in which we have to map the new gray level values into number of pixels.

Lets assume our old gray levels values has these number of pixels.

Gray Level Value	Frequency
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16

Now if we map our new values to , then this is what we got.

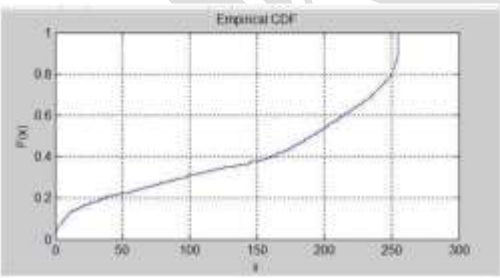
Gray Level Value	New Gray Level Value	Frequency
0	0	2
1	1	4
2	3	6
3	4	8
4	5	10
5	6	12
6	6	14
7	7	16

Now map these new values you are onto histogram , and you are done.

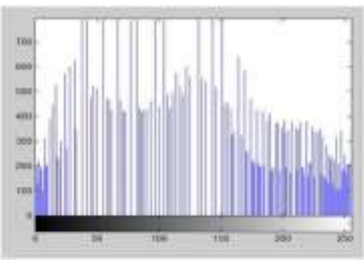
Lets apply this technique to our original image. After applying we got the following image and its following histogram.

Histogram Equalization Image

Cumulative Distributive function of this image



Histogram Equalization histogram

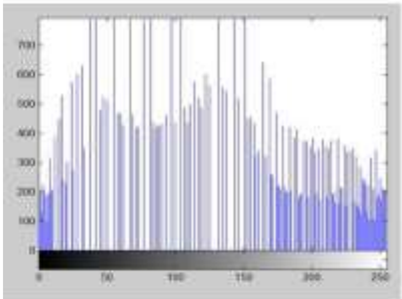


Comparing both the histograms and images

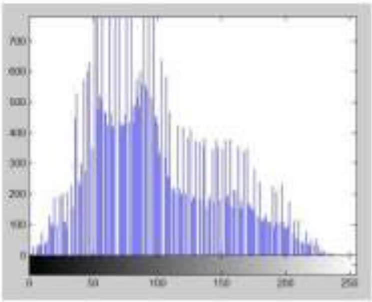
New Image

Old image

New Histogram



Old Histogram



(Or)

6) Discuss about image denosing using spatial mean filters. CO3 L2 6M

Image denoising using spatial mean filters is a fundamental technique in image processing aimed at reducing noise in an image. Noise can be introduced during image acquisition, transmission, or storage, and it often manifests as random variations in pixel intensity. Mean filters are a type of spatial filter used to smooth images by reducing the intensity variations between neighboring pixels.

What is a Spatial Mean Filter?

A spatial mean filter is a simple, linear filter that works by averaging the pixel values in a local neighborhood around each pixel in an image. The basic idea is to replace each pixel value with the mean (average) of the pixel values in its neighborhood. This process helps to smooth out noise by averaging out the variations.

(i) Arithmetic mean filter

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinates in a rectangular subimage window of size m X n, centered at point (x, y).The arithmetic mean filtering process computes the average value of the corrupted image g(x, y) in the area defined by S_{xy}. The value of the restored image *f* at any point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by S_{xy}. In other words

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

This operation can be implemented using a convolution mask in which all coefficients have value 1/mn

(ii) Geometric mean filter

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

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

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

(iii) Harmonic mean filter

The harmonic mean filtering operation is given by the expression

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

The harmonic mean filter works well for salt noise, but fails for pepper noise. It does well also with other types of noise like Gaussian noise.

(iv) Contra harmonic mean filter

The contra harmonic mean filtering operation yields a restored image based on the expression

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

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

SET-2

1) a) What are the various fundamental steps in digital image processing? ExplainCO1 L1 6M

Ans

Fundamental Steps in Digital Image Processing

Digital Image Processing

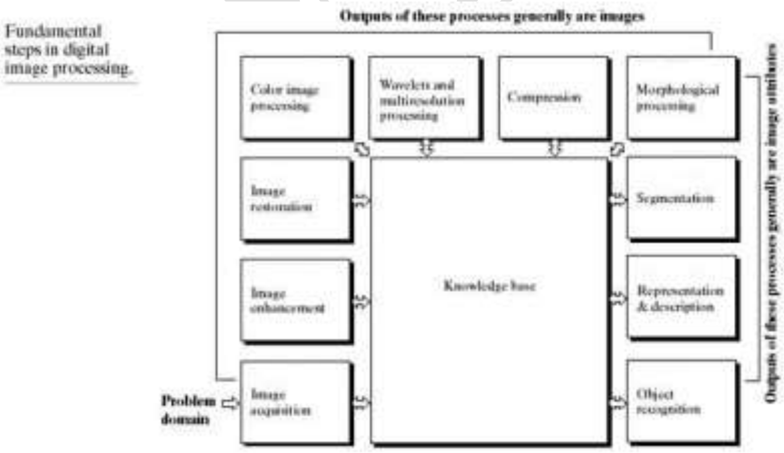


Fig: Fundamental steps in digital image processing

1. Image Acquisition:

In _image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing.

It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed. In image acquisition using pre-processing such as scaling is done.

Read More: Image Acquisition in Digital Image Processing

2. Image Enhancement:

It is the process of adjusting digital images so that the results are more suitable for display or further image analysis. Usually in includes sharpening of images, brightness & contrast adjustment, removal of noise, etc. In image enhancement, we generally try to modify the image, so as to make it more pleasing to the eyes.

It is subjective in nature as for example some people like high saturation images and some people like natural colour. That’s why it is subjective in nature as it differs from person to person.

3. Image Restoration:

It is the process of recovering an image that has been degraded by some knowledge of degraded function H and the additive noise term. Unlike image enhancement, image restoration is completely objective in nature.

Read More: Image Restoration and Image Degradation Model

4. Color Image Processing:

This part handles the _image processing of colored images either as indexed images or RGB images.

5. Wavelets and multiresolution processing:

- Wavelets are small waves of limited duration which are used to calculate wavelet transform which provides time-frequency information.
- Wavelets lead to multiresolution processing in which images are represented in various degrees of resolution.

6. Compression:

Compression deals with the techniques for reducing the storage space required to save an image or the bandwidth required to transmit it.

This is particularly useful for displaying images on the internet as if the size of the image is large, then it uses more bandwidth (data) to display the image from the server and also increases the loading speed of the website.

7. Morphological Processing:

It deals with extracting image components that are useful in representation and description of shape.

It includes basic morphological operations like erosion and dilation. As seen from the block _diagram above that the outputs of morphological processing generally are image attributes.

8. Segmentation:
It is the process of partitioning a digital image into multiple segments. It is generally used to locate objects and boundaries in objects.

9. Representation and Description:

- Representation deals with converting the data into a suitable form for computer processing.
    - Boundary representation: it is used when the focus is on external shape characteristics e.g. corners
    - Regional representation: it is used when the focus in on internal properties e.g. texture
- Description deals with extracting attributes that
    - results in some quantitative information of interest
    - is used for differentiating one class of objects from others

10. Recognition:
It is the process that assigns a label (e.g. car) to an object based on its description.

b) Describe in detail about various image sensors **CO1 L2 6M**

Ans

Image sensors are instruments used for converting optical images into electronic signals. The global image sensors market has grown immensely due to technological advancements in the last decade. This has stimulated diversification in image sensor applications. Image sensors have been placed on board NASA's Interface Region Imaging Spectrograph (IRIS) spacecraft to capture the first images of the Sun.

Types of Image Sensors

There are two types of commonly used image sensors in the market - charged-coupled device (CCD) and complementary metal oxide semiconductor (CMOS) sensors.

CCD Sensors

CCDs are manufactured using an advanced technique that enables transportation of charge across the chip without any distortion, thereby producing very high-quality sensors with reliability and light sensitivity.

The chip output is in the form of analog voltage. The images produced using CCD image sensors possess high quality and low noise. The processing speed is in the range of moderate to fast and the complexity of the sensor is low.

Although CCD sensors consume more power than CMOS sensors, these sensors are widely used in many applications requiring high-quality image data.

CMOS Sensors

CMOS is an integrated circuit technology that has been adapted to capture images. The CMOS sensor components can be easily integrated onto a single chip using conventional manufacturing processes.

The chip output is in the form of digital bits. Although they are complex, CMOS image sensors are said to be easier and cheaper to manufacture than CCD sensors. Each pixel in the CMOS sensor can be read separately. Processing of image is fast while the sensitivity is low. They are, however, comparatively more vulnerable to noise.

The advancement in CMOS image technology is enabling CMOS sensors to move towards higher levels of performance.

Working Principle

Each image sensor is built with an array of photodetectors called pixels that gather photons or single particles of light. The photons in the pixels are then converted to electrons.

The gathered electrons are moved to a signal conditioning and processing circuitry. The electron charge is then output as voltage signal. The voltage signal is then amplified before it is supplied to a camera circuitry.

Digitization enables users to store, archive and retrieve image files electronically in a simple manner.

Applications

Image sensors are widely used in aerospace, defense, automotive, biometrics, and media sectors.

The following is a list of key applications of image sensors:

- Imaging devices such as digital cameras, camera modules, camcorders, smart phones, security cameras, PC cameras, personal digital assistants (PDAs), machine vision, security and surveillance cameras, and videoconferencing
- Optical mouse, document scanning, barcode readers
- Toys and games
- Medical – e.g., dental radiography and pill cameras
- Scientific imaging

(Or)

2) a) Compute the Euclidean Distance (De), City-block Distance (D4) and Chessboard distance (D8) for points p and q, where p and q be (5, 2) and (1, 5) respectively CO1 L2 6M

same as below change the example values

b) Define D4, D8 and Dm distances and explain with an example CO1 L2 6M

Ans

In image processing and computer vision, distances between pixels are important for various operations such as edge detection, image segmentation, and morphological operations. Three common distance metrics are D_4 (Manhattan distance), D_8 (Chessboard distance), and D_m (Euclidean distance). Here's a detailed explanation of each with examples:

D4 Distance (Manhattan Distance)

The D_4 distance, also known as the Manhattan or City Block distance, is the distance between two points measured along axes at right angles. It is calculated as the sum of the absolute differences of their Cartesian coordinates.

Formula:

$$D_4(p, q) = |x_2 - x_1| + |y_2 - y_1|$$

where $p = (x_1, y_1)$ and $q = (x_2, y_2)$.

Example:

Consider two points $p = (2, 3)$ and $q = (5, 1)$.

$$D_4((2,3),(5,1)) = |5 - 2| + |1 - 3| = 3 + 2 = 5$$

D8 Distance (Chessboard Distance)

The D_8 distance, also known as the Chessboard or Chebyshev distance, is the distance between two points considering the greatest difference between their coordinates. It corresponds to the minimum number of moves a king would take to travel between two squares on a chessboard.

Formula:

$$D_8(p, q) = \max(|x_2 - x_1|, |y_2 - y_1|)$$

where $p = (x_1, y_1)$ and $q = (x_2, y_2)$

Example:

Consider two points $p = (2, 3)$ and $q = (5, 1)$.

$$D_8((2,3),(5,1)) = \max(|5-2|, |1-3|) = \max(3,2) = 3$$

Dm Distance (Euclidean Distance)

The D_m distance, commonly known as the Euclidean distance, is the straight-line distance between two points in Euclidean space. It is the most commonly used distance metric in geometry.

Formula:

$$D_m(p,q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where $p = (x_1, y_1)$ and $q = (x_2, y_2)$.

Example:

Consider two points $p = (2, 3)$ and $q = (5, 1)$.

$$D_m((2,3),(5,1)) = \sqrt{(5-2)^2 + (1-3)^2} = \sqrt{3^2 + (-2)^2} = \sqrt{9+4} = \sqrt{13} \approx 3.61$$

3) a) Discuss about Hadamard Transform with necessary mathematical expressions

CO2 L3 6M

Ans

The Hadamard Transform is a linear, orthogonal transformation technique that is widely used in signal processing, image processing, and various other fields. It is particularly noted for its simplicity and computational efficiency. The Hadamard Transform can be considered a generalization of the Fourier Transform, but it uses only addition and subtraction operations, making it very efficient for digital computation.

Definition and Mathematical Expressions

The Hadamard Transform is defined using Hadamard matrices. An n×n Hadamard matrix Hn is a matrix whose entries are either +1 or -1, and rows are mutually orthogonal. This means that the inner product of any two distinct rows (or columns) is zero.

The Hadamard matrix of order n, where n is a power of 2, can be defined recursively as:

Ex: for N=4

[1 1 1 1
1 -1 1 -1
1 1 -1 -1
1 -1 -1 1]

Let's compute the Hadamard Transform for a simple vector (x = [1, 2, 3, 4]) using (H_4): [
H_4 =
[1 1 1 1

1 -1   1 -1
1   1 -1 -1
1 -1 -1 1]

Now, perform the transformation:
X = H_4.x=
[
1 1 1 1
1 -1 1 -1 . [1 2 3 4]
1 1 -1 -1
1 -1 -1 1]
=>
1 + 2 + 3 + 4
1 - 2 + 3 - 4
1 + 2 - 3 - 4
1 - 2 - 3 + 4

= [10 -2 -4 0]

Thus, the Hadamard Transform of \(x \) is (X = [10, -2, -4, 0]).

Properties

1. Orthogonality: The Hadamard matrix is orthogonal, i.e., (H_n^T = H_n^{-1}).
2. Symmetry: The Hadamard matrix is symmetric,(H_n = H_n^T).
3. Fast Computation: The Fast Hadamard Transform (FHT) can be computed in (O(n \log n)) time, similar to the Fast Fourier Transform (FFT).

Applications

1. Signal Processing: Used for noise reduction, data compression, and error detection and correction.
2. Image Processing: Used in image compression techniques such as JPEG and other applications where fast transformation is necessary.
3. Quantum Computing: The Hadamard gate is a fundamental quantum gate that performs the Hadamard transform on qubits.
4. Communications: Used in spread spectrum and code-division multiple access (CDMA) systems.

b) Illustrate the 2D Fourier Transform and its pair. State and prove their properties

CO2 L2 6M

Ans

2D Fourier Transform

The 2D Fourier Transform (FT) is a mathematical transformation used to analyze spatial frequencies in two-dimensional signals or images. Given a 2D function $f(x, y)$, its 2D Fourier Transform is defined as:

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

where:

- $f(x, y)$ is the original spatial domain function.
- $F(u, v)$ is the transformed function in the frequency domain.
- (u, v) are the frequency domain variables.

Properties of the 2D Fourier Transform

1. **Linearity**: The Fourier Transform is a linear operation. If $f(x, y)$ and $g(x, y)$ are two functions and a and b are constants, then:
   $\mathcal{F}\{af(x,y) + bg(x,y)\} = aF(u,v) + bG(u,v)$
2. **Translation**: If $f(x, y)$ is shifted by (x_0, y_0), its Fourier Transform is modulated by a complex exponential:
   $\mathcal{F}\{f(x - x_0, y - y_0)\} = e^{-j2\pi(ux_0+vy_0)}F(u,v)$
3. **Scaling**: If the function is scaled by a and b in the spatial domain, its Fourier Transform is scaled inversely and also scaled in amplitude:
   $\mathcal{F}\{f(ax, by)\} = \frac{1}{|ab|}F\left(\frac{u}{a}, \frac{v}{b}\right)$
4. **Rotation**: Rotating the function in the spatial domain results in a corresponding rotation in the frequency domain.
   If $f(x, y)$ is rotated by an angle θ:
   $\mathcal{F}\{f(x', y')\} = F(u', v')$
   where (x', y') and (u', v') are the rotated coordinates.

Proofs

Linearity

Given $f(x, y)$ and $g(x, y)$ with their respective Fourier Transforms $F(u, v)$ and $G(u, v)$:

$$\mathcal{F}\{af(x,y) + bg(x,y)\}$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(af(x,y) + bg(x,y))e^{-j2\pi(ux+vy)}\,dx\,dy$$
$$= a\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy +$$
$$b\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy$$
$$= aF(u,v) + bG(u,v)$$

Translation

Given a shift by (x_0, y_0):

$$\mathcal{F}\{f(x - x_0, y - y_0)\}$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x - x_0, y - y_0)e^{-j2\pi(ux+vy)}\,dx\,dy$$

$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v) e^{-j2\pi(u(x_0+x)+v(y_0+y))}\, du\, dv$
$= e^{-j2\pi(ux_0+vy_0)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v) e^{-j2\pi(ux+vy)}\, du\, dv$
$= e^{-j2\pi(ux_0+vy_0)} F(u,v)$

Convolution

Given $f(x,y)$ and $g(x,y)$:

$\mathcal{F}\{f(x,y) * g(x,y)\}$
$= \mathcal{F}\left\{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y') g(x-x', y-y')\, dx'\, dy'\right\}$
$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y') g(x-x', y-y')\, dx'\, dy'\right) e^{-j2\pi(ux+vy)}\, dx\, dy$
Using Fubini's theorem to switch the order of integration:
$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y') \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x-x', y-y') e^{-j2\pi(ux+vy)}\, dx\, dy\right) e^{-j2\pi(ux'+vy')}\, dx'\, dy'$
Using the shifting property:
$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y') e^{-j2\pi(ux'+vy')} \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(u',v') e^{-j2\pi(u(u-u')+v(v-v'))}\, du'\, dv'\right) dx'\, dy'$
$= F(u,v) G(u,v)$

Proof

1. **Define the rotated coordinates**:
   Let (x', y') be the rotated coordinates obtained by rotating (x, y) by an angle θ:
   $x' = x\cos\theta + y\sin\theta$
   $y' = -x\sin\theta + y\cos\theta$
2. **Express the rotated function**:
   The rotated function $f(x', y')$ in terms of (x, y) is:
   $f(x', y') = f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$
3. **Fourier Transform of the rotated function**:
   The 2D Fourier Transform of $f(x', y')$ is given by:
   $F(u', v') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') e^{-j2\pi(u'x'+v'y')}\, dx'\, dy'$
4. **Substitute the rotated coordinates into the integral**:
   Change variables from (x', y') back to (x, y):
   $dx'\, dy' = dx\, dy$
   The integral becomes:

The exponent simplifies to:

$-j2\pi(u'x' + v'y') = -j2\pi(u'(x\cos\theta + y\sin\theta) + v'(-x\sin\theta + y\cos\theta))$
$= -j2\pi((u'\cos\theta - v'\sin\theta)x + (u'\sin\theta + v'\cos\theta)y)$

6. **Recognize the new frequency variables**:

   Let:

   $u = u'\cos\theta - v'\sin\theta$
   $v = u'\sin\theta + v'\cos\theta$

   Then the exponent becomes:

   $-j2\pi(ux + vy)$

7. **Express the integral in terms of the new frequencies**:

   The Fourier Transform integral now becomes:

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

(Or)

4) a) Compare the various image transformation techniques CO2 L2 6M

Ans

mage transformation techniques are essential in image processing, enabling tasks such as image enhancement, compression, and feature extraction. Here, we compare several commonly used image transformation techniques: Fourier Transform, Discrete Cosine Transform (DCT), Wavelet Transform, Hadamard Transform, and Radon Transform.

1. Fourier Transform

Description:

- Transforms an image from the spatial domain to the frequency domain.
- Represents the image as a sum of sinusoids with different frequencies.

Mathematical Formulation:

- 2D Discrete Fourier Transform (DFT) for an image $f(x, y)$:

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

Advantages:

- Useful for frequency analysis and filtering.
- Can be efficiently computed using the Fast Fourier Transform (FFT).

Disadvantages:

- Not localized in space, leading to poor performance on non-stationary signals.

Applications:

- Image compression (JPEG).
- Image filtering (low-pass, high-pass filters).

2. Discrete Cosine Transform (DCT)

Description:

- Similar to the Fourier Transform but uses only cosine functions.
- Focuses on representing an image in terms of its frequency components.

Mathematical Formulation:

- 2D DCT for an image f(x,y)f(x, y)f(x,y):

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

where $\alpha(u)$ and $\alpha(v)$ are normalization factors.

Advantages:

- Energy compaction: most of the signal information is concentrated in a few low-frequency components.
- Efficient for image compression.

Disadvantages:

- Poor performance on highly textured or detailed images.

Applications:

- Image compression (JPEG).
- Video compression (MPEG).

3. Wavelet Transform

Description:

- Represents an image in terms of wavelets, which are small wave-like oscillations.
- Provides both time (spatial) and frequency information.

Mathematical Formulation:

- Continuous Wavelet Transform (CWT) for a signal f(t)f(t)f(t):

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) dt$$

where ψ is the mother wavelet, aaa is the scale, and bbb is the translation parameter.

Advantages:

- Multi-resolution analysis.
- Good at representing localized features and edges.

Disadvantages:

- Computationally more complex than DCT and Fourier Transform.

Applications:

- Image compression (JPEG 2000).
- Image denoising and enhancement.

4. Hadamard Transform

Description:

- Uses Hadamard matrices composed of +1 and -1.
- Transforms an image by summing and subtracting pixel values.

Mathematical Formulation:

- For a vector x of length n:

$$X=H_n \cdot x$$

where H_n is the Hadamard matrix of order n.

Advantages:

- Simple and computationally efficient (only additions and subtractions).
- Orthogonal transform.

Disadvantages:

- Less effective at capturing frequency information compared to Fourier and DCT.

Applications:

- Image compression.
- Error correction.

5. Radon Transform

Description:

- Projects an image onto a set of parallel lines at various angles.
- Used for feature extraction and tomographic reconstruction.

Mathematical Formulation:

- For an image f(x,y)f(x, y)f(x,y):

$$R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(\rho - x \cos\theta - y \sin\theta) dx dy$$

where δ is the Dirac delta function.

Advantages:

- Effective for detecting linear features and reconstructing images from projections.

Disadvantages:

- Computationally intensive.
- Requires large amounts of data for accurate reconstruction.

Applications:

- Medical imaging (CT scans).
- Industrial tomography.

Summary

Transform	Advantages	Disadvantages	Applications
Fourier Transform	Frequency analysis, filtering	Not localized in space	Image filtering, JPEG compression
Discrete Cosine Transform (DCT)	Energy compaction, efficient compression	Poor performance on detailed images	JPEG compression, MPEG compression
Wavelet Transform	Multi-resolution analysis, localized features	Computationally complex	JPEG 2000 compression, image denoising
Hadamard Transform	Simple, computationally efficient	Less effective for frequency information	Image compression, error correction
Radon Transform	Effective for linear features, tomographic reconstruction	Computationally intensive	CT scans, industrial tomography

b) Discuss about FFT and state its properties **CO2 L2 6M**

The FFT algorithm efficiently computes the DFT by exploiting the symmetry and periodicity properties of the complex exponential functions used in the DFT. The most common FFT algorithms are the Cooley-Tukey algorithm, which is a divide-and-conquer approach, and its variations.

Properties of FFT

1. **Linearity:**
   - The FFT is a linear transform, meaning that the FFT of a sum of signals is the sum of their FFTs.

   $$\text{FFT}\{ax[n] + by[n]\} = a\text{FFT}\{x[n]\} + b\text{FFT}\{y[n]\}$$

2. **Symmetry (Conjugate Symmetry):**
   - For real-valued input signals, the FFT output exhibits conjugate symmetry. If $x[n]$ is real, then:

   $$X[N-k] = \overline{X[k]}$$

3. **Periodicity:**
   - The DFT and its inverse are periodic with period N:

   $$X[k+N] = X[k] \quad \text{and} \quad x[n+N] = x[n]$$

4. **Time and Frequency Shifting:**
   - Time Shifting: Shifting a signal in the time domain results in a phase shift in the frequency domain.

   $$\text{If } y[n] = x[n-n_0], \text{ then } Y[k] = X[k]e^{-j\frac{2\pi}{N}kn_0}$$

   - Frequency Shifting: Multiplying a signal by a complex exponential in the time domain results in a shift in the frequency domain.

   $$\text{If } y[n] = x[n]e^{j\frac{2\pi}{N}k_0 n}, \text{ then } Y[k] = X[(k-k_0) \mod N]$$

5. **Convolution:**
   - Convolution in the time domain corresponds to multiplication in the frequency domain and vice versa.

   $$y[n] = x[n] * h[n] \quad \text{corresponds to} \quad Y[k] = X[k]H[k]$$

6. **Parseval's Theorem:**
   - The total energy of a signal is preserved in both the time and frequency domains.

   $$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2$$

7. **Duality:**
   - The DFT of the DFT is proportional to the original sequence, possibly reversed and scaled.

   $$\text{DFT}\{\text{DFT}\{x[n]\}\} \propto x[-n]$$

5) Discuss how the derivative filters are used in Digital Image Enhancement? CO3L2 6M

Ans

(Or)

6) Discuss about image de-nosing using spatial mean filters. CO3 L2 6M

Repeated in set -1