CBCS SCHEME

USN | | | | | | | | | | |

21EC732

Seventh Semester B.E./B.Tech. Degree Examination, Dec.2024/Jan.2025
Digital Image Processing

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. With neat diagram explain Single image sensor, how it can be used in Sensor Strip and Sensor Array. **(08 Marks)**

b. Explain basic concept of Sampling and Quantization with reference to Digital Image. **(07 Marks)**

c. Calculate the photon energy for visible light for given wavelength range 400 nm to 750 nm. [Plank's constant, $h = 6.63\times10^{-34}$ Js, $C = 3\times10^{8}$ m/s] **(05 Marks)**

OR

2 a. Explain the Brightness Adaption and Discrimination. **(07 Marks)**

b. Explain the Neighbour pixel basic relationship in Digital Images with adjacency connectivity, Regions and Boundaries. **(08 Marks)**

c. Given two pixels P and Q with coordinate positions (–2, –2) and (3, 4) respectively, calculate the distance measure D_e , D_4 , D_8 . **(05 Marks)**

Module-2

3 a. Define 2-D orthogonal and unitary transform. **(06 Marks)**

b. For given orthogonal matrix A and an image u obtain unitary transform.

Given $A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $\quad u = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ **(08 Marks)**

c. Define the properties of unitary transform. **(06 Marks)**

OR

4 a. Define 2-D DFT and its properties. **(06 Marks)**

b. Define cosine transform and its properties. **(06 Marks)**

c. Calculate Haar transform for N = 4

Given Haar function

$$H_a(z) = \frac{1}{\sqrt{N}}\begin{cases} +2^{p/2} , & \frac{q-1}{2^p} \le z < \frac{q-0.5}{2^p} \\ -2^{p/2} , & \frac{q-0.5}{2^p} \le z < \frac{q}{2^p} \\ 0 , & \text{else} \end{cases}$$

$n = \log_2 N$
$p = 0$ to $n - 1$
q range between $1 \le q \le 2^p$
$k = 2^p + q - 1$
$z = 0,\ 1/4,\ 2/4,\ 3/4$ **(08 Marks)**

Module-3

5 a. With necessary graph and equation explain
i) Image Negative
ii) Power law transformation
iii) Intensity level slicing (06 Marks)

b. Compute Histogram equalization for given data:

Table 5(b)

r_k	0	1	2	3	4	5	6	7
n_k	790	1023	850	656	329	245	122	81

for 3 bit image (L = 8) of size 64×64 pixels (MN = 4096) with intensity distribution shown in Table 5(b). Intensity level are integer in range [0, L-1] = [0, 7] (08 Marks)

c. With an example for 2-bit image of size 5×5 define the sample mean, sample variance with equation. (06 Marks)

OR

6 a. Explain with example fundamentals of Spatial Filtering for spatial correlation and convolution for 1-D and 2-D filter. (08 Marks)

b. Using 1st order derivative Image Sharpening (the Gradient) define:
i) Robert's cross gradient operation
ii) Sobel's operators (for 3×3 region) (06 Marks)

c. Define smoothing spatial filters with brief note:
i) Linear Filters
ii) Order Statistic Filter (06 Marks)

Module-4

7 a. With neat block diagram of Homomorphic system, derive Homomorphic filtering approach for Image Enhancement. (08 Marks)

b. Define sharpening of images in frequency domain using
i) Ideal High Pass Filter
ii) Butterworth High Pass Filter
iii) Gaussian High Pass Filter (06 Marks)

c. Give Frequency domain filtering necessary steps followed. (06 Marks)

OR

8 a. Define pseudo color image processing with intensity slicing and intensity to color transformation. (06 Marks)

b. Based on Hardware oriented models classify different color model given color conversion for RGB to HIS and vice versa with relevant equation. (08 Marks)

c. With color fundamentals for primary and secondary colors. (06 Marks)

Module-5

9 a. Write brief note on restoration in presence of only noise using
i) Mean filter ii) Order statistic filter iii) Adaptive filter (08 Marks)

b. Discuss some of the important noise probability density functions. (06 Marks)

c. With help of block diagram give details of Degradation / Restoration process. (06 Marks)

OR

10 a. In digital images discuss about Inverse Filtering. (06 Marks)

b. Explain minimum mean square error (Wiener Filter) in Digital Image Processing. (08 Marks)

c. Discuss periodic noise reduction by frequency domain filtering. (06 Marks)

* * * * *

1.a)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.

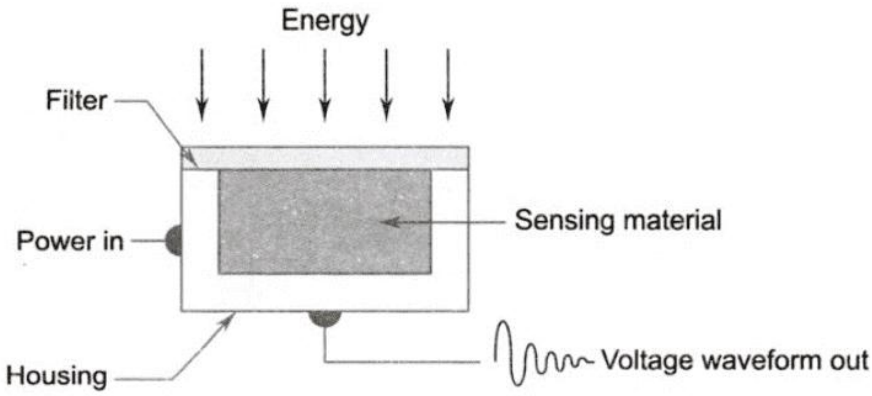


Fig: Single image sensor

F

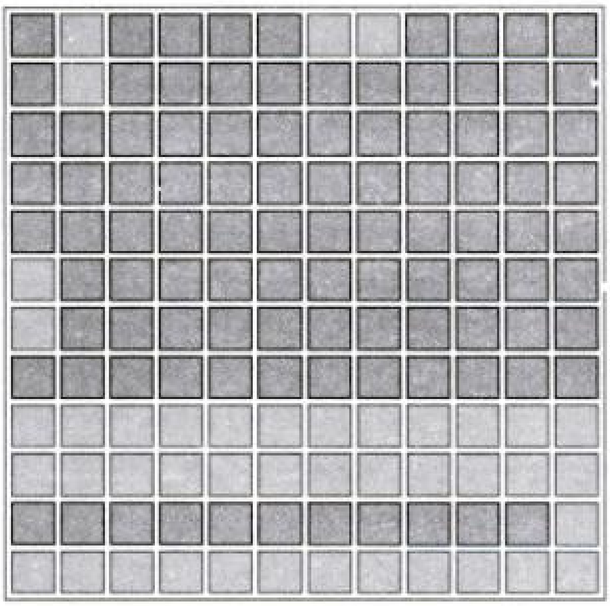


Fig: Array sensor

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.

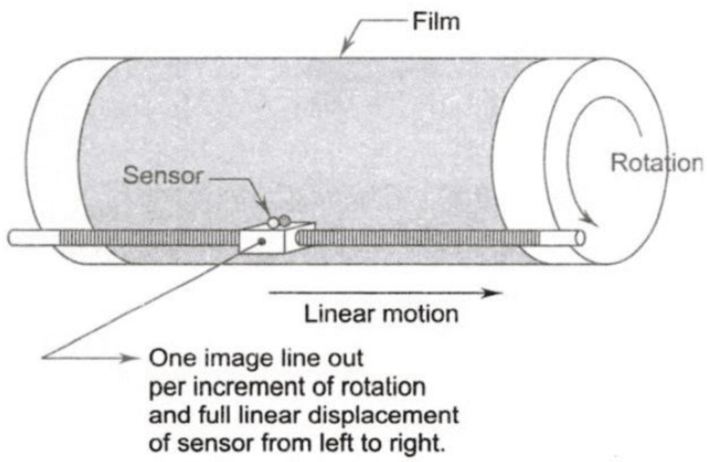


Fig: Combining a single sensor with motion to generate a 2D image

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.

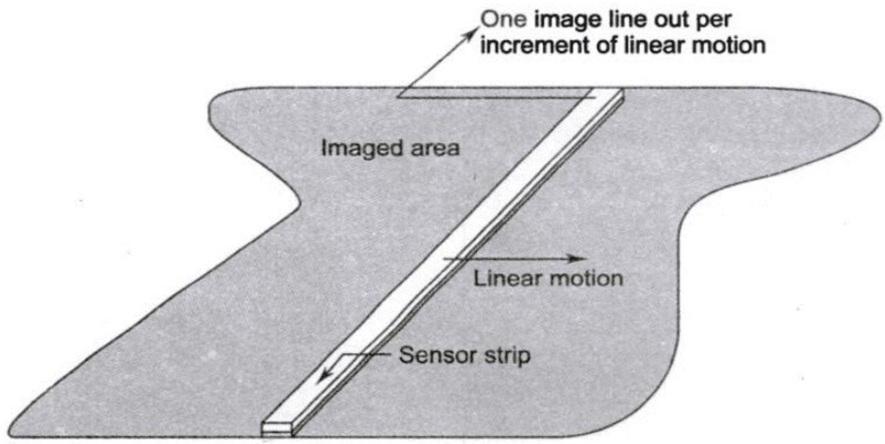


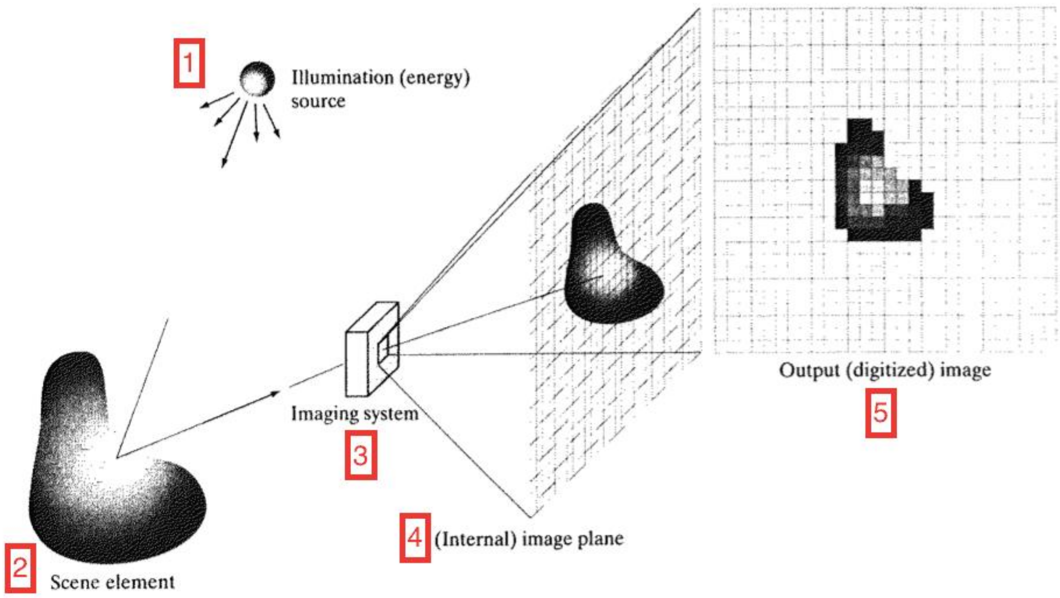
Fig: Linear sensor strip

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.



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.

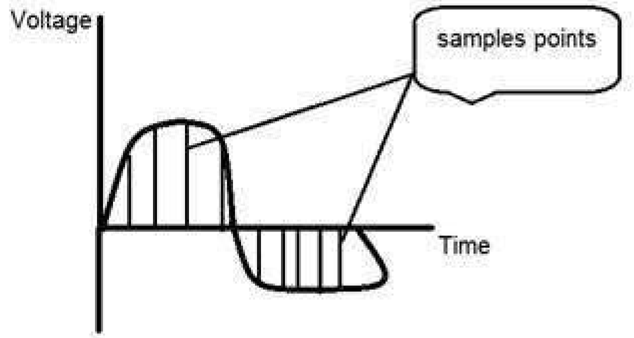
1b).In digital Image Processing, signals captured from the physical world need to be translated into digital form by "Digitization" Process. In order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude. This digitization process involves two main processes called

1. **Sampling:** Digitizing the co-ordinate value is called sampling.
2. **Quantization:** Digitizing the amplitude value is called quantization

Typically, a frame grabber or digitizer is used to sample and quantize the analogue video signal.

Sampling

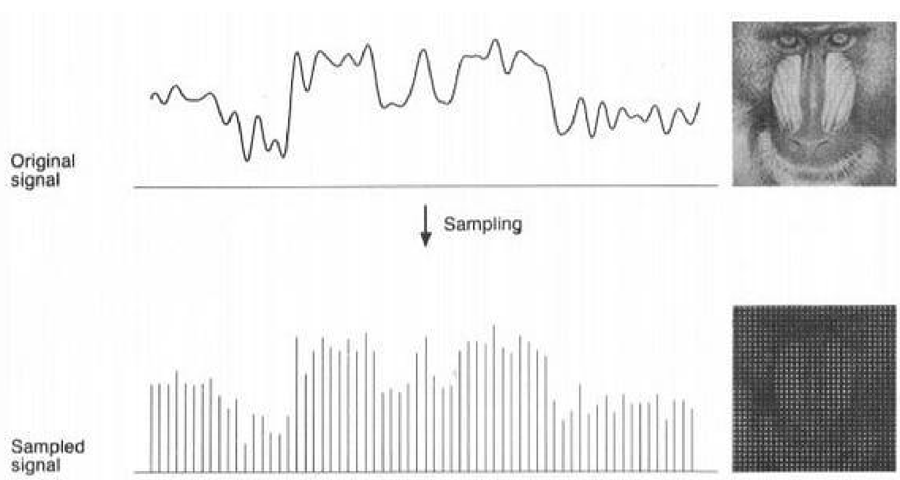
Since an analogue image is continuous not just in its co-ordinates (x axis), but also in its amplitude (y axis), so the part that deals with the digitizing of co-ordinates is known as sampling. In digitizing sampling is done on independent variable. In case of equation y = sin(x), it is done on x variable.



When looking at this image, we can see there are some random variations in the signal caused by noise. In sampling we reduce this noise by taking samples. It is obvious that more samples we take, the quality of the image would be more better, the noise would be more removed and same happens vice versa. However, if you take sampling on the x axis, the signal is not converted to digital format, unless you take sampling of the y-axis too which is known as quantization.

Sampling has a relationship with image pixels. The total number of pixels in an image can be calculated as Pixels = total no of rows * total no of columns. For example, let's say we have total of 36 pixels, that means we have a square image of 6X 6. As we know in sampling, that more samples eventually result in more pixels. So it means that of our continuous signal, we have taken 36 samples on x axis. That refers to 36 pixels of this image. Also the number sample is directly equal to the number of sensors on CCD array.

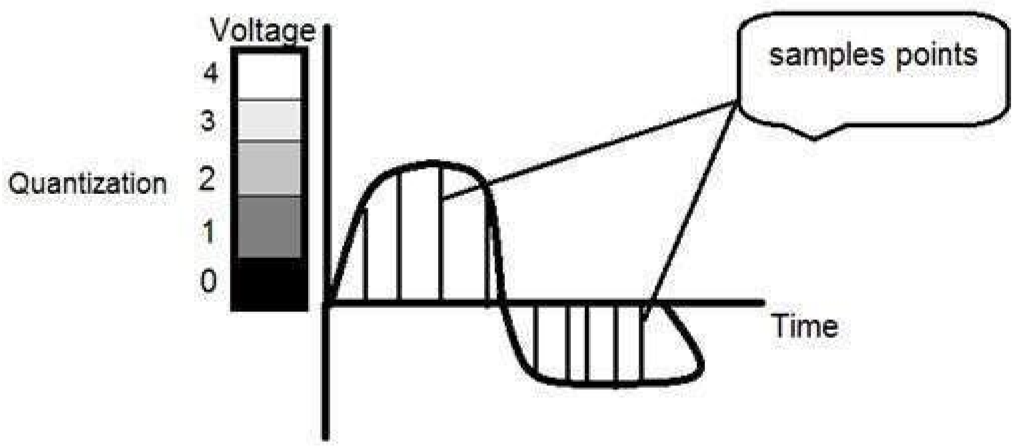
Here is an example for image sampling and how it can be represented using a graph.



Quantization

Quantization is opposite to sampling because it is done on “y axis” while sampling is done on “x axis”. Quantization is a process of transforming a real valued sampled image to one taking only a finite number of distinct values. Under quantization process the amplitude values of the image are digitized. In simple words, when you are quantizing an image, you are actually dividing a signal into quanta(partitions).

Here we assign levels to the values generated by sampling process. In the image showed in sampling explanation, although the samples has been taken, but they were still spanning vertically to a continuous range of gray level values. In the image shown below, these vertically ranging values have been quantized into 5 different levels or partitions. Ranging from 0 black to 4 white. This level could vary according to the type of image you want.



There is a relationship between Quantization with gray level resolution. The above quantized image represents 5 different levels of gray and that means the image formed from this signal, would only have 5 different colors. It would be a black and white image more or less with some colors of gray.

When we want to improve the quality of image, we can increase the levels assign to the sampled image. If we increase this level to 256, it means we have a gray scale image.

Whatever the level which we assign is called as the gray level. Most digital IP devices uses quantization into k equal intervals. If b-bits per pixel are used,

$$\text{No. of quantization levels} = k = 2^b$$

The number of quantization levels should be high enough for human perception of fine shading details in the image. The occurrence of false contours is the main problem in image which has been quantized with insufficient brightness levels. Here is an example for image quantization process.

2.a)Brightness Adaptation and Discrimination

Brightness Adaptation

Brightness adaptation refers to the process by which our visual system adjusts its sensitivity to light based on the level of illumination in the environment. Our eyes have a remarkable ability to adapt to different levels of brightness and adjust our perception of brightness accordingly.

When we are in a low-light environment, our pupils dilate to allow more light into the eye, and our visual system becomes more sensitive to low levels of light. Conversely, in a bright environment, our pupils constrict to limit the amount of light entering the eye, and our visual system becomes less sensitive to light.

This adaptation is critical in image processing because it allows us to see images accurately in a range of lighting conditions. However, it can also create challenges in image processing since the same image can appear differently depending on the lighting conditions in which it is viewed. For example, an image that looks bright and clear in a well-lit room may appear dark and unclear in a dimly lit environment.

Brightness Discrimination

Brightness discrimination refers to the ability of our visual system to distinguish between different levels of brightness in an image. Our visual system can detect differences in brightness levels as small as one percent. This ability is crucial in image processing because it allows us to perceive and distinguish different features and objects in an image.

For example, if we look at a photograph of a person, our visual system can distinguish the brightness levels between the person's hair, skin, and clothing, allowing us to perceive each feature separately.

Brightness discrimination is also essential in image processing applications, such as image segmentation and edge detection, which require the detection of boundaries between different regions of an image. These boundaries are typically defined by changes in brightness levels, and the ability to discriminate these levels accurately is critical in these applications.

2.b)Adjacency between pixels

Let **V** be the set of intensity values used to define adjacency.

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 set V typically contains more elements.

For example, in the adjacency of pixels with a range of possible intensity values 0 to 255, set V could be any subset of these 256 values.

We consider three types of adjacency:

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

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

c) m-adjacency(mixed adjacency): Two pixels p and q with values from V are m-adjacent if

1. q is in N4(p), or
2. 2) q is in ND(p) and the set N4(p)∩N4(q) has no pixels whose values are from V.

Connectivity between pixels

It is an important concept in digital image processing.

It is used for establishing boundaries of objects and components of regions in an image.

Two pixels are said to be connected:

- if they are adjacent in some sense(neighbour pixels,4/8/m-adjacency)
- if their gray levels satisfy a specified criterion of similarity(equal intensity level)

There are three types of connectivity on the basis of adjacency. They are:

a) 4-connectivity: Two or more pixels are said to be 4-connected if they are 4-adjacent with each others.

b) 8-connectivity: Two or more pixels are said to be 8-connected if they are 8-adjacent with each others.

c) m-connectivity: Two or more pixels are said to be m-connected if they are m-adjacent with each others.

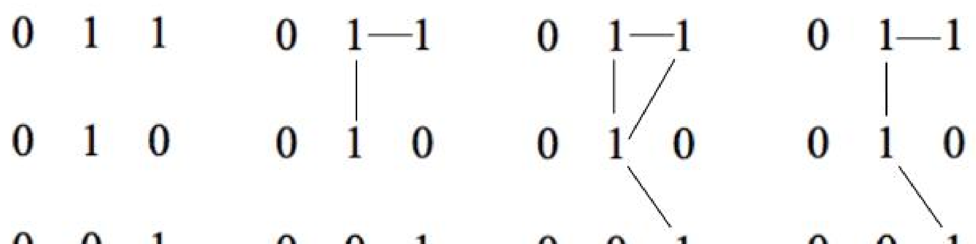


Fig: An arrangement of pixels | **Fig: 4-connectivity of pixels** | **Fig: 8-connectivity of pixels** | **Fig: m-connectivity of pixels**

Region

- A connected set is also called a Region.
- Two regions (let Ri and Rj) are said to be adjacent if their union forms a connected set. Adjacent Regions or joint regions
- Regions that are not adjacent are said to be disjoint regions.
- 4- and 8-adjacency is considered when referring to regions (author)
- Discussing a particular region, type of adjacency must be specified.
- Fig2.25d the two regions are adjacent only if 8-adjacency is considered

Foreground and Background

- Suppose an image contain K disjoint regions Rk , k=1,2,3,…K, none of which touches the image border

- Let Ru denote the union of all the K regions.
- Let (Ru)c denote its compliment.
- We call all the points in Ru the foreground and all the points in (Ru)c the background

Boundary

- **The boundary (border or contour) of a region R is the set of points that are**

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

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} I(x,y,u,v) g(u,v)$$

where, again, $T(u,v,x,y)$ and $I(x,y,u,v)$ are called the **forward and inverse transformation kernels**, respectively.

The forward kernel is said to be **separable** if

$$T(u,v,x,y) = T_1(u,x)T_2(v,y)$$

It is said to be **symmetric** if T_1 is functionally equal to T_2 such that

$$T(u,v,x,y) = T_1(u,x)T_1(v,y)$$

The same comments are valid for the inverse kernel.

If the kernel $T(u,v,x,y)$ of an image transform is separable and symmetric, then the transform $g(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} T(u,v,x,y) f(x,y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} T_1(u,x)T_1(v,y) f(x,y)$ can be written in matrix form as follows

$$\underline{g} = \underline{T}_1 \cdot \underline{f} \cdot \underline{T}_1^{\,T}$$

where $\underline{f}$ is the original image of size $N \times N$, and $\underline{T}_1$ is an $N \times N$ transformation matrix with elements $t_{ij} = T_1(i,j)$. If, in addition, $\underline{T}_1$ is a unitary matrix then the transform is called **separable unitary** and the original image is recovered through the relationship

$$\underline{f} = \underline{T}_1^{*T} \cdot \underline{g} \cdot \underline{T}_1^{*}$$

adjacent to the points in the complement of R.

- **Set of pixels in the region that have at least one background neighbor.**
- **The boundary of the region R is the set of pixels in the region that have one or more neighbors that are not in R.**
- Inner Border: Border of Foreground
- Outer Border: Border of Background

3.a)2D Orthogonal and Unitary transform and its properties

As a one dimensional signal can be represented by an orthonormal set of basis vectors, an image can also be expanded in terms of a discrete set of basis arrays called basis images through a two dimensional (image) transform. For an N X N image f(x,y) the forward and inverse transforms are given below

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

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}I(x,y,u,v)g(u,v)$$

where, again, $T(u,v,x,y)$ and $I(x,y,u,v)$ are called the **forward and inverse transformation kernels**, respectively.
The forward kernel is said to be **separable** if

$$T(u,v,x,y)=T_1(u,x)T_2(v,y)$$

It is said to be **symmetric** if T_1 is functionally equal to T_2 such that

$$T(u,v,x,y)=T_1(u,x)T_1(v,y)$$

The same comments are valid for the inverse kernel.
If the kernel $T(u,v,x,y)$ of an image transform is separable and symmetric, then the transform $g(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}T(u,v,x,y)f(x,y)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}T_1(u,x)T_1(v,y)f(x,y)$ can be written in matrix form as follows

$$\underline{g}=\underline{T}_1\cdot\underline{f}\cdot\underline{T}_1^{T}$$

where $\underline{f}$ is the original image of size $N\times N$, and $\underline{T}_1$ is an $N\times N$ transformation matrix with elements $t_{ij}=T_1(i,j)$. If, in addition, $\underline{T}_1$ is a unitary matrix then the transform is called **separable unitary** and the original image is recovered through the relationship

$$\underline{f}=\underline{T}_1^{*T}\cdot\underline{g}\cdot\underline{T}_1^{*}$$

4.b)A cosine transform is a mathematical operation that represents a signal or function as a sum of cosine functions at different frequencies, essentially capturing the "even" components of a signal; it is particularly useful in signal processing and data compression due to its property of energy compaction, where most of the significant information is concentrated in a few low-frequency cosine coefficients, making it ideal for applications like image and video compression (e.g., JPEG) where redundancy can be removed effectively; key properties of a cosine transform include: orthogonality, energy compaction, separability, and the ability to represent real-valued data using only real numbers.

Key points about cosine transform:

- **Basis functions:**

  Cosine functions with different frequencies act as the basis functions for the transform.

- **Real-valued data:**

  Unlike the Fourier transform, cosine transforms only involve real numbers, making them efficient for processing real-world signals.

- **Energy compaction:**

  Important information about a signal tends to be concentrated in the low-frequency cosine coefficients, allowing for data compression by discarding less significant high-frequency components.

Types of Cosine Transforms:

- **Continuous Cosine Transform (CCT):**

  Applies to continuous functions, using integration to calculate the transform coefficients.

- **Discrete Cosine Transform (DCT):**

  Primarily used in digital signal processing, where the input signal is sampled and represented as a finite sequence, allowing for efficient computation.

Important properties of DCT:

- **Orthogonality:**

  Cosine basis functions are orthogonal to each other, ensuring that the transform coefficients are independent and can be manipulated individually.

- **Separability:**

  For 2D signals, the DCT can be applied separately along rows and columns due to the separable nature of the cosine functions, reducing computational complexity.

- **Inversion property:**

  The original signal can be reconstructed from the DCT coefficients using the inverse transform.

- **Symmetry:**

DCTs often exhibit specific symmetry properties depending on the type of transform (e.g., DCT-II has even symmetry).

Applications of Cosine Transform:

- **Image and video compression:**

  DCT is widely used in image and video compression standards like JPEG and MPEG due to its efficient energy compaction.

- **Signal analysis:**

  Analyzing the frequency components of a signal by examining the DCT coefficients.

- **Speech coding:**

  DCT can be employed in speech coding applications to represent the spectral characteristics of speech signals.

5.a)i)Negative Image:

The negative of an image is achieved by replacing the intensity 'i' in the original image by 'i-1', i.e. the darkest pixels will become the brightest and the brightest pixels will become the darkest. Image negative is produced by subtracting each pixel from the maximum intensity value.
For example in an 8-bit grayscale image, the max intensity value is 255, thus each pixel is subtracted from 255 to produce the output image.
The transformation function used in image negative is :

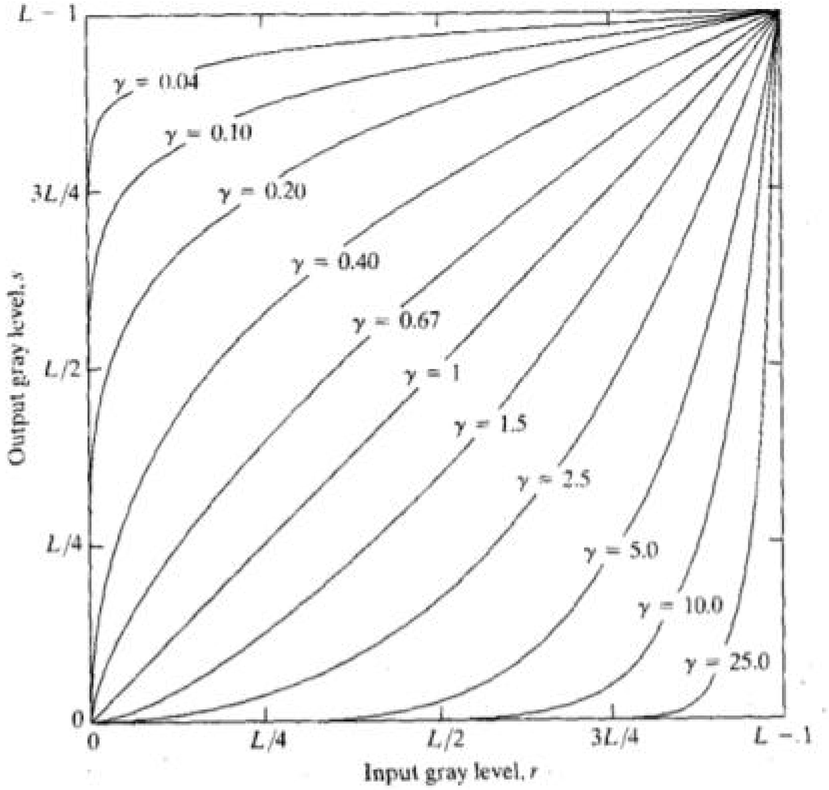
```
s = T(r) = (L – 1) – r


Where L - 1 is the max intensity value,

s is the output pixel value and

r is the input pixel value
```

5.a)ii)Power Law (Gamma) Transformations
"Gamma Correction", most of you might have heard this strange sounding thing. In this blog, we will see what it means and why does it matter to you?
The general form of Power law (Gamma) transformation function is

s = c*rγ

Where, 's' and 'r' are the output and input pixel values, respectively and 'c' and γ are the positive constants. Like log transformation, power law curves with γ <1 map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher input values. Similarly, for γ >1, we get the opposite result which is shown in the figure below

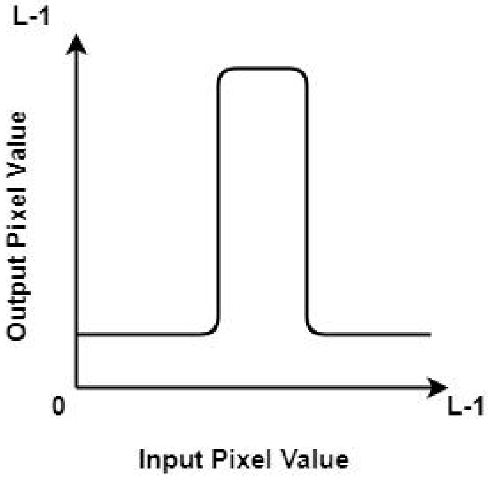
This is also known as gamma correction, gamma encoding or gamma compression. Don’t get confused.
The below curves are generated for r values normalized from 0 to 1. Then multiplied by the scaling constant c corresponding to the bit size used.

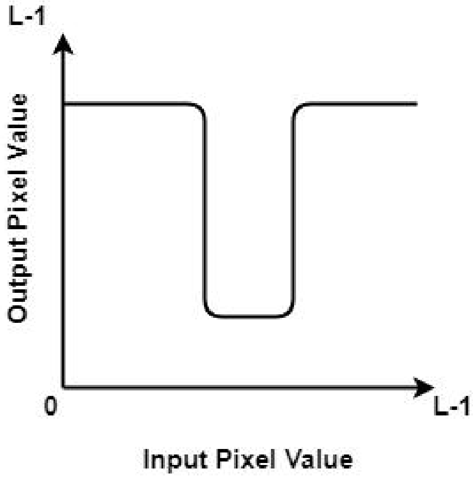


5. a) iii) Intensity-level Slicing

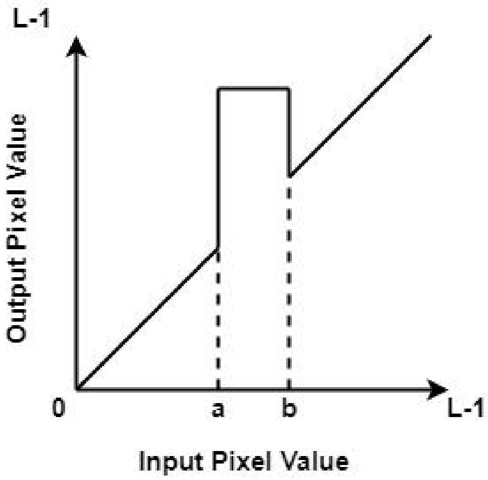
Intensity level slicing means highlighting a specific range of intensities in an image. In other words, we segment certain gray level regions from the rest of the image.
Suppose in an image, your region of interest always take value between say 80 to 150. So, intensity level slicing highlights this range and now instead of looking at the whole image, one can now focus on the highlighted region of interest.
Since, one can think of it as piecewise linear transformation function so this can be implemented in several ways. Here, we will discuss the two basic type of slicing that is more often used.

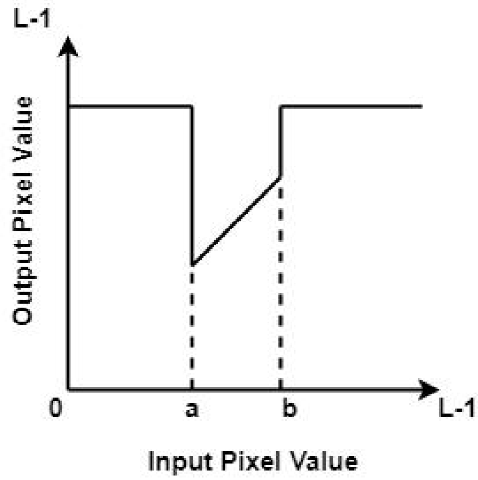
- In the first type, we display the desired range of intensities in white and suppress all other intensities to black or vice versa. This results in a binary image. The transformation function for both the cases is shown below.





- In the second type, we brighten or darken the desired range of intensities(a to b as shown below) and leave other intensities unchanged or vice versa. The transformation function for both the cases, first where the desired range is changed and second where it is unchanged, is shown below.





6.a)In image processing, "spatial filtering" refers to the technique of modifying an image by applying mathematical operations to a small neighborhood of pixels around each pixel, essentially changing the intensity of a pixel based on the values of its neighbors, allowing for image enhancement like smoothing, sharpening, or edge detection; while "spatial correlation" describes the statistical relationship between pixel intensities at different locations within an image, indicating how similar pixel values are across spatial proximity.

Key points about spatial filtering:

Concept:

- A "spatial filter" is essentially a small array (called a kernel or mask) that is applied to each pixel in an image, with each element of the kernel representing a weight applied to the corresponding neighboring pixel value.

Linear vs. Non-linear:

-
- **Linear filters:** Perform a weighted sum of the neighborhood pixel values, where the weights are fixed and independent of the pixel values. Examples include averaging filters (smoothing) and Sobel filters (edge detection).
- **Non-linear filters:** Apply a non-linear operation on the neighborhood pixel values, often using ranking operations like the median filter to remove noise.
-

Applications:

-
- **Smoothing:** Blurring an image by averaging pixel values in a neighborhood, useful for noise reduction.
- **Sharpening:** Highlighting edges by emphasizing intensity differences between neighboring pixels.
- **Edge detection:** Identifying boundaries between objects in an image using filters that respond strongly to sharp intensity changes.

Key points about spatial correlation:

- **Definition:**
- Spatial correlation measures how strongly the intensity of a pixel is related to the intensity of its neighboring pixels.

  High correlation:
- If neighboring pixels tend to have similar intensity values, the spatial correlation is considered high.

  Low correlation:
- If neighboring pixels have very different intensities, the spatial correlation is low.

Relationship between spatial filtering and spatial correlation:

- **Filter design:**
- Understanding the spatial correlation within an image helps in designing appropriate spatial filters. For example, if an image has high spatial correlation, a smoothing filter with a larger kernel might be effective for noise reduction.

  Feature extraction:
- Analyzing spatial correlation can be used to extract features from an image, such as identifying areas with high texture or repeating patterns.

6.b)Robert Operator: This gradient-based operator computes the sum of squares of the differences between diagonally adjacent pixels in an image through discrete differentiation. Then the gradient approximation is made. It uses the following 2 x 2 kernels or masks –

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The **Sobel operator**, sometimes called the **Sobel–Feldman operator** or **Sobel filter**, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges.

The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. If we define **A** as the source image, and $\mathbf{G}_x$ and $\mathbf{G}_y$ are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

spatial Filtering technique is used directly on pixels of an image. Mask is usually considered to be added in size so that it has specific center pixel. This mask is moved on the image such that the center of the mask traverses all image pixels.

6.c)Smoothing Spatial Filter

Smoothing filter is used for blurring and noise reduction in the image. Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.

Types of Smoothing Spatial Filter

1. Linear Filter (Mean Filter)
2. Order Statistics (Non-linear) filter

These are explained as following below.

1. **Mean Filter:** Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask. Below are the types of mean filter:
   - **Averaging filter:** It is used in reduction of the detail in image. All coefficients are equal.
   - **Weighted averaging filter:** In this, pixels are multiplied by different coefficients. Center pixel is multiplied by a higher value than average filter.
2. **Order Statistics Filter:** It is based on the ordering the pixels contained in the image area encompassed by the filter. It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering. Below are the types of order statistics filter:
   - **Minimum filter:** 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window.
   - **Maximum filter:** 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window.
   - **Median filter:** Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

7.a)Homomorphic filtering is sometimes used for image enhancement. It simultaneously normalizes the brightness across an image and increases contrast. Here homomorphic filtering is used to remove multiplicative noise. Illumination and reflectance are not separable, but their approximate locations in the frequency domain may be located. Since illumination and reflectance combine multiplicatively, the components are made additive by taking the logarithm of the image intensity, so that these multiplicative components of the image can be separated linearly in the frequency domain. Illumination variations can be thought of as a multiplicative noise, and can be reduced by filtering in the log domain.

To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decreased, because the high-frequency components are assumed to represent mostly the reflectance in the scene (the amount of light reflected off the object in the scene), whereas the low-frequency components are assumed to represent mostly the illumination in the scene. That is, high-pass filtering is used to suppress low frequencies and amplify high frequencies, in the log-intensity domain.

Operation

Homomorphic filtering can be used for improving the appearance of a grayscale image by simultaneous intensity range compression (illumination) and contrast enhancement (reflection).

$$m(x,y) = i(x,y) \bullet r(x,y)$$

Where,

m = image,

i = illumination,

r = reflectance

We have to transform the equation into frequency domain in order to apply high pass filter. However, it's very difficult to do calculation after applying Fourier transformation to this equation because it's not a product equation anymore. Therefore, we use 'log' to help solve this problem.

$$\ln(m(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$$

Then, applying Fourier transformation

$$F(ln(m(x,y))) = F(ln(i(x,y))) + F(ln(r(x,y)))$$

Or

$$M(u,v) = I(u,v) + R(u,v)$$

Next, applying high-pass filter to the image. To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decrease.

$$N(u,v) = H(u,v) \bullet M(u,v)$$

Where

H = any high-pass filter

N = filtered image in frequency domain

Afterward, returning frequency domain back to the spatial domain by using inverse Fourier transform.

$$n(x, y) = invF(N(u, v))$$

Finally, using the exponential function to eliminate the log we used at the beginning to get the enhanced image

$$newImage(x, y) = exp(n(x, y))$$

7.b)High-Pass Filtering (Sharpening)

A high-pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image – exactly the opposite of the low-pass filter. High-pass filtering works in exactly the same way as low-pass filtering; it just uses a different convolution kernel. In the example below, notice the minus signs for the adjacent pixels. If there is no change in intensity, nothing happens. But if one pixel is brighter than its immediate neighbors, it gets boosted.

0	-1/4	0
-1/4	+2	-1/4
0	-1/4	0

Unfortunately, while low-pass filtering smooths out noise, high-pass filtering does just the opposite: it *amplifies noise*. You can get away with this if the original image is not too noisy; otherwise the noise will overwhelm the image. MaxIm DL includes a very useful "range-restricted filter" option; you can high-pass filter only the brightest parts of the image, where the signal-to-noise ratio is highest.

High-pass filtering can also cause small, faint details to be greatly exaggerated. An over-processed image will look grainy and unnatural, and point sources will have dark donuts around them. So while high-pass filtering can often improve an image by sharpening detail, overdoing it can actually degrade the image quality significantly.

Butterworth High pass filter

In the field of Image Processing, **Butterworth Highpass Filter (BHPF)** is used for image sharpening in the frequency domain. Image Sharpening is a technique to enhance the fine details and highlight the edges in a digital image. It removes low-frequency components from an image and preserves high-frequency components. This Butterworth highpass filter is the reverse operation of the Butterworth lowpass filter. It can be determined using the relation-

$\mathrm{H}_{HP}(u, v) = 1 - H_{LP}(u, v)$ where, $H_{HP}(u, v)$ is the transfer function of the highpass filter and $H_{LP}(u, v)$ is the transfer function of the corresponding lowpass filter. The transfer function of BHPF of order n is defined as-

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

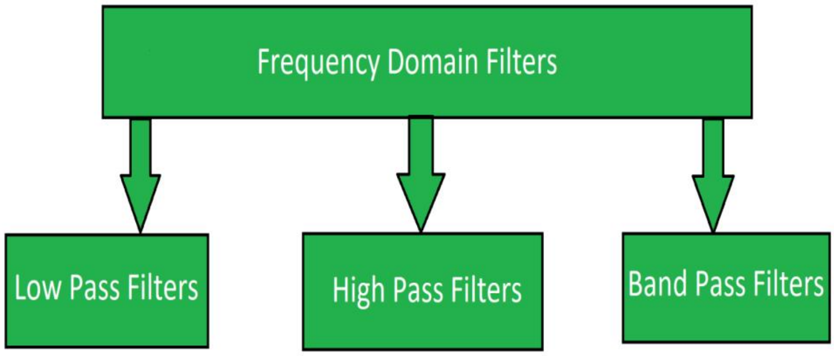
Where,

- D_0 is a positive constant. BHPF passes all the frequencies greater than D_0 value without attenuation and cuts off all the frequencies less than it.
- This D_0 is the transition point between H(u, v) = 1 and H(u, v) = 0, so this is termed as cutoff frequency. But instead of making a sharp cut-off (like, , it introduces a smooth transition from 0 to 1 to reduce ringing artifacts.
- $D(u, v)$ is the Euclidean Distance from any point (u, v) to the origin of the frequency plane, i.e, $D(u, v) = \sqrt{(u^2 + v^2)}$

7.C)Frequency domain filters

Frequency Domain Filters are used for smoothing and sharpening of image by removal of high or low frequency components. Sometimes it is possible of removal of very high and very low frequency. Frequency domain filters are different from spatial domain filters as it basically focuses on the frequency of the images. It is basically done for two basic operation i.e., Smoothing and Sharpening.

These are of 3 types:



Classification of Frequency Domain Filters

1. Low pass filter:
Low pass filter removes the high frequency components that means it keeps low frequency components. It is used for smoothing the image. It is used to smoothen the image by attenuating high frequency components and preserving low frequency components.
Mechanism of low pass filtering in frequency domain is given by:

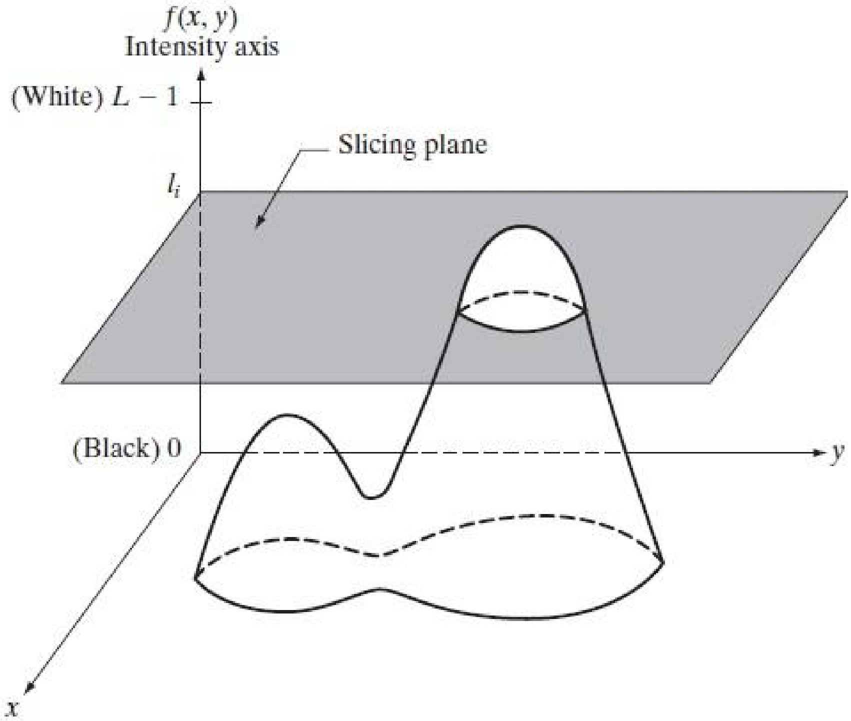
2. High pass filter:
High pass filter removes the low frequency components that means it keeps high frequency components. It is used for sharpening the image. It is used to sharpen the image by attenuating low frequency components and preserving high frequency components.

3. Band pass filter:
Band pass filter removes the very low frequency and very high frequency components that means it keeps the moderate range band of frequencies. Band pass filtering is used to enhance edges while reducing the noise at the same time.

8.a)Intensity Slicing

This is a simple case of pseudocolor image processing. It’s also called **density slicing** or **color coding.**

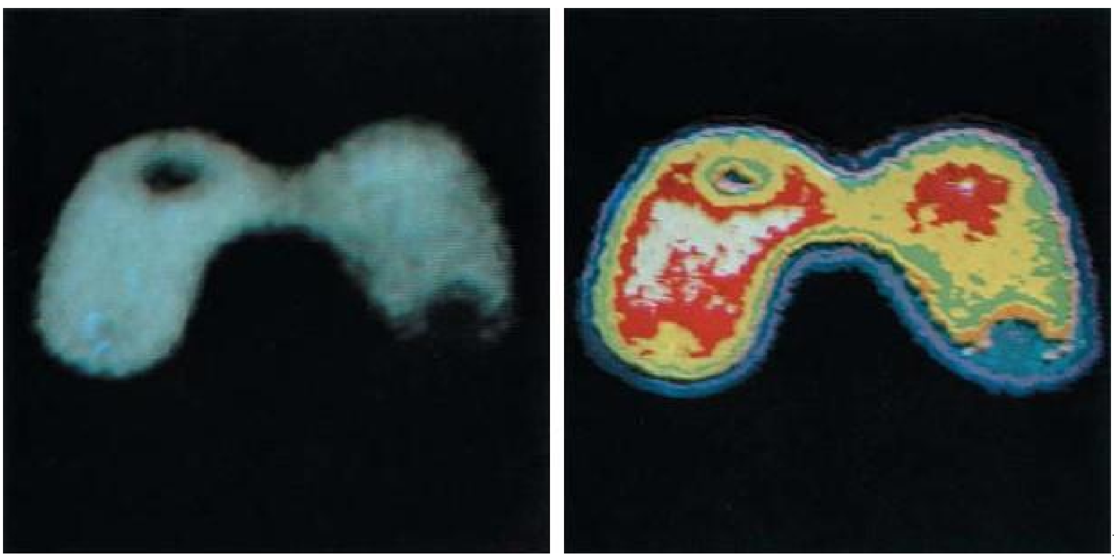
Imagine a grayscale image as a 3D function with the intensity being the third dimension. Placing a plane parallel to the horizontal plane for the pixel position coordinates would *“slice”* the image into two parts. After that we can assign different colors to different levels.



G

Geometric interpretation of the intensity slicing technique.

We are not constrained to use only one plane. Multiple slices result in more flexible representations of the grayscale images. In this case, the grayscale is parted into intervals and each one is assigned a different color.

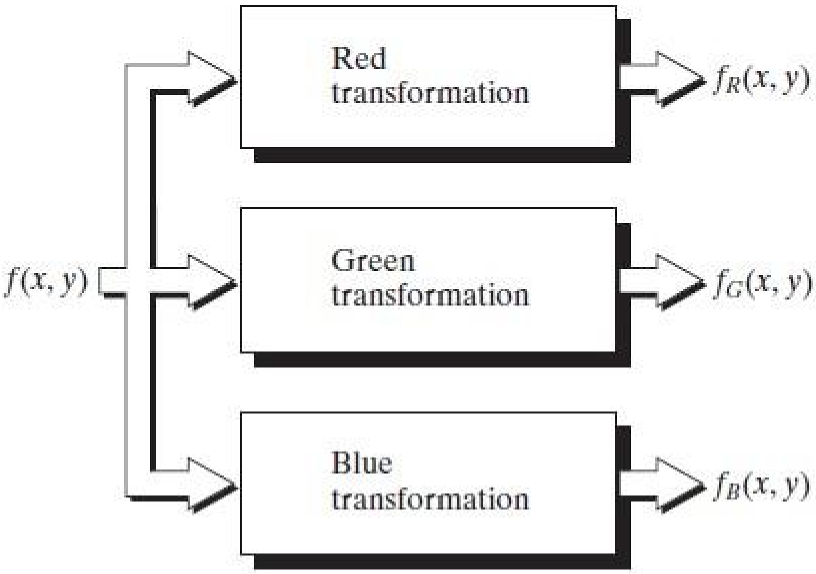


R

result of density slicing into eight colors.

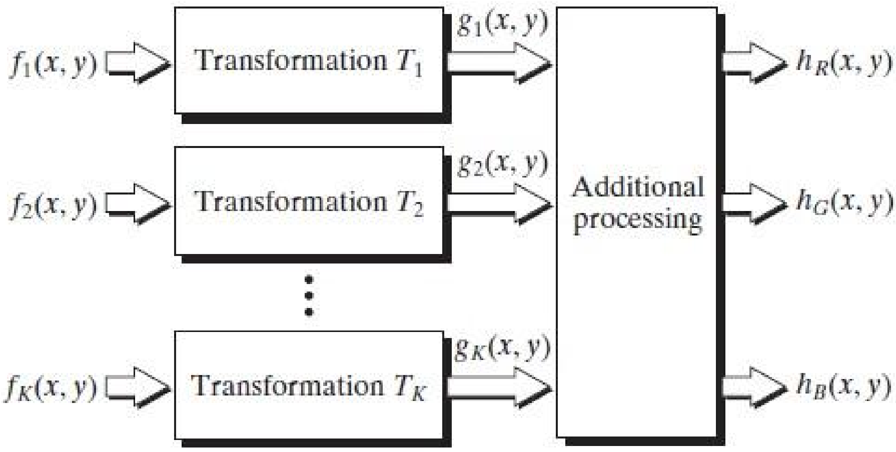
Intensity to Color Transformation

We can generalize the above technique by performing three independent transformations on the intensity of the image, resulting in three images which are the red, green, blue component images used to produce a color image.



Functional block diagram for pseudocolor image processing.

The flexibility can be even more enhanced by using more than one monochrome images, for example, the three components of an RGB and the thermal image.



T

This technique is called **multispectral image processing.**

8.b)RGB TO HIS Conversion

RGB to HSI Conversion:

First, we convert RGB color space image to HSI space beginning with normalizing RGB values:

$$r = \frac{R}{R+G+B}, g = \frac{G}{\quad}$$

$R+G+B$

$$, b = \frac{B}{R+G+B}$$

Each normalized H, S and I components are then obtained by,

$$h = \cos^{-1}\left\{ \frac{0.5\cdot\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}} \right\} \qquad h \in [0,\pi] \text{ for } b \le g$$

$$h = 2\pi - \cos^{-1}\left\{ \frac{0.5\cdot\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}} \right\} \qquad h \in [\pi, 2\pi] \text{ for } b > g$$

$$s = 1 - 3\cdot\min(r, g, b); \qquad s \in [0,1]$$

$$i = (R + G + B)/(3\cdot 255); \qquad i \in [0,1].$$

For convenience, h, s and i values are converted in the ranges of [0,360], [0,100], [0, 255], respectively , by:

$$H = h \cdot 180/\pi; \quad S = s \cdot 100 \text{ and } I = i \cdot 255.$$

8.c)Primary colors are red, yellow, and blue. Secondary colors are made by mixing two primary colors.

Primary colors

- Cannot be made by mixing other colors
- Are the foundation for all other colors

Secondary colors

- Are made by mixing two primary colors
- Are located between primary colors on the color wheel
- Examples include orange, purple, and green

This is the second post on the report of Chapter 6 from the book Digital Image Processing (Rafael C. Gonzalez). You can read about how colors are perceived and common color models in my first post.

There two main categories of color image processing: **pseudocolor** (false colo**r) image processing** and **full-color image processing**. In this post, we will talk about the first one.

True color VS false color

Before talking about false color, what does **true color** mean anyway?

True-color images are images with natural color rendition: a red apple appears red, a blue ocean appears blue, etc. It's difficult to achieve absolute true color in images, but being approximately close to human perception is acceptable.

False color images, on the other hand, sacrifices natural color rendition in order to facilitate the detection of some objects. They are mainly used for satellite and space images.

Pseudocolor images are originally grayscale which are assigned colors based on the intensity values. Typical usage of these images is for thermography where the only available is infrared radiation instead of lights. Another example is elevation map.

9.a)Mean Filter: Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask. Below are the types of mean filter:

- **Averaging filter:** It is used in reduction of the detail in image. All coefficients are equal.

- **Weighted averaging filter:** In this, pixels are multiplied by different coefficients. Center pixel is multiplied by a higher value than average filter.

Order Statistics Filter: It is based on the ordering the pixels contained in the image area encompassed by the filter. It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering. Below are the types of order statistics filter:

- **Minimum filter:** 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window.
- **Maximum filter:** 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window.
- **Median filter:** Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

9.b)Noise Probaibilty Density Functions

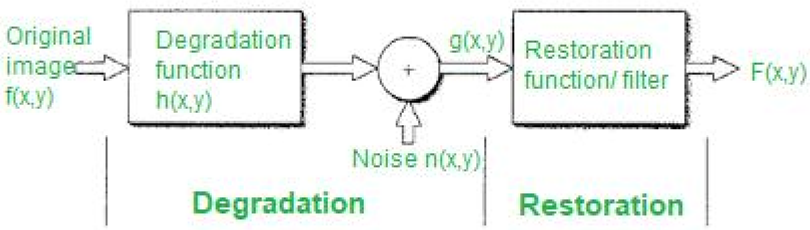
The principal source of noise in digital images arises during image acquisition and transmission. The performance of imaging sensors is affected by a variety of environmental and mechanical factors of the instrument, resulting in the addition of undesirable noise in the image. Images are also corrupted during the transmission process due to non-ideal channel characteristics.

Generally, a mathematical model of image degradation and its restoration is used for processing. The figure below shows the presence of a degradation function *h(x,y)* and an external noise *n(x,y)* component coming into the original image signal *f(x,y)* thereby producing a final degraded image *g(x,y)*. This part composes the degradation model. Mathematically we can write the following :

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

Where * indicates convolution in the spatial domain.

The goal of the restoration function or the restoration filter is to obtain a close replica *F(x,y)* of the original image.



The external noise is probabilistic in nature and there are several noise models used frequently in the field of digital image processing. We have several probability density functions of the noise.

Noise Models

Gaussian Noise:

Because of its mathematical simplicity, the Gaussian noise model is often used in practice and even in situations where they are marginally applicable at best. Here, m is the mean and σ2 is the variance.

Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination or high temperature.

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-m)^2}{2\sigma^2}}$$

Rayleigh Noise

$$p(z) = \frac{2}{b}(z-a)e^{\frac{-(z-a)^2}{b}} for z \geq a, and p(z) = 0 otherwise.$$

Here mean m and variance σ2 are the following:

$$m = a + \sqrt{\pi b/4}$$

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

Rayleigh noise is usually used to characterize noise phenomena in range imaging.

Erlang (or gamma) Noise

$$p(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az} for z \geq 0 and p(z) = 0 otherwise.$$

Here ! indicates factorial. The mean and variance are given below.

$$m = b/a, 2 = b/a2$$

Gamma noise density finds application in laser imaging.

Exponential Noise

$$p(z) = ae^{-az} for z \geq 0 and p(z) = 0 otherwise.$$

Here a > 0. The mean and variance of this noise pdf are:

$$m = 1/a$$

$$2 = 1/\mathrm{a}2$$

This density function is a special case of b = 1.

Exponential noise is also commonly present in cases of laser imaging.

Uniform Noise

$$p(z) = \frac{1}{b-a} if a \leq z \leq b, and p(z) = 0 otherwise.$$

The mean and variance are given below.

$$m = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

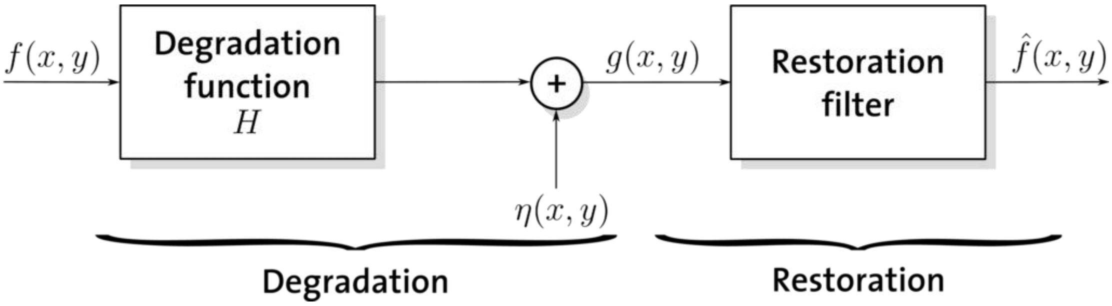
Uniform noise is not practically present but is often used in numerical simulations to analyze systems.

Impulse Noise

$$p(z) = Paforz = a, p(z) = Pbforz = b, p(z) = 0otherwise.$$

If b > a, intensity b will appear as a light dot in the image. Conversely, level a will appear like a black dot in the image. Hence, this presence of white and black dots in the image resembles to salt-and-pepper granules, hence also called salt-and-pepper noise. When either Pa or Pb is zero, it is called unipolar noise. The origin of impulse noise is quick transients such as faulty switching in cameras or other such cases.

9.c)Image restoration is the process of recovering an image that has been degraded by some knowledge of degradation function ***H*** and the additive noise term $\eta(x,y)$. Thus in restoration, degradation is modelled and its inverse process is applied to recover the original image.



Objective of image restoration:

The objective of image restoration is to obtain an estimate of the original image $f(x,y)$. Here, by some knowledge of ***H*** and $\eta(x,y)$, we find the appropriate restoration filters, so that output image $\widehat{f}(x,y)$ is as close as original image $f(x,y)$ as possible since it is practically not possible (or very difficult) to completely (or exactly) restore the original image.

Terminology:

- $g(x,y)$ = degraded image
- $f(x,y)$ = input or original image
- $\widehat{f}(x,y)$ = recovered or restored image
- $\eta(x,y)$ = additive noise term

In spatial domain:

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

where, ⊛ represents convolution

In frequency domain:

After taking fourier transform of the above equation:

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

If the restoration filter applied is $R(u,v)$, then

$$\widehat{F}(u,v) = R(u,v)[G(u,v)]$$

$$\widehat{F}(u,v) = R(u,v)H(u,v)F(u,v) + R(u,v)N(u,v)$$

$$\widehat{F}(u,v) \approx F(u,v) \text{ (for restoration)}$$

as restoration filter $R(u,v)$ is the reverse of degration function $H(u,v)$ and neglecting the noise term. Here, $H(u,v)$ is linear and position invariant.

10.b Wiener Filtering

Performs 2D adaptive Wiener filtering on *M* using a local window *win_w* pixels wide by *win_h* pixels high.

Wiener filtering was one of the first methods developed to reduce additive random noise in images. It works on the assumption that additive noise is a stationary random process, independent of pixel location; the algorithm minimizes the square error between the original and reconstructed images. Wiener filtering is a low-pass filter, but instead of having a single cutoff frequency, it is a space-varying filter designed to use a low cutoff in low-detail regions and a high cutoff to retain detail in regions with edges or other high-variance features. The window size determines the overall frequency cutoff: larger windows correspond to lower cutoff frequencies, and therefore more blurring and noise reduction.

There are several possible implementations for Wiener filtering. The one used in this PTC Mathcad function is the pixel-by-pixel 2D adaptive Wiener filtering proposed by Lee in 1980 (see Two-Dimensional Signal and Image Processing, by Jae

S. Lim, pages 536-40), where a space-varying filter is used, and the additive noise is assumed to be white and zero-mean.

In this algorithm, a pixel *y* in the filtered image is derived from a pixel *x* in the noisy input image by the following transformation:

$$y = \mu_x + \left(x - \mu_x\right) \frac{v_x}{v_x + v_n}$$

where μ_x and v_x are the mean and variance of *x* in a neighborhood around the pixel (the neighborhood size is given by the *win_h* and *win_w* arguments to the function), and v_n is the variance of the additive noise, estimated from the input image. Each pixel in the output is the sum of the local mean from a neighborhood of the input pixel and a local contrast term (x - μ_x) that is scaled so that in high-detail regions, where noise variance (v_n) is much smaller than image variance (v_x), the scaling factor is very close to 1 and the output pixel *y* is very close to the input pixel *x* with little filtering, but in low-detail regions, where the image variance is lower, the output pixel tends to be more like the local mean (that is, it is low-pass filtered).

Boundaries of the image are treated as extending with zero grayscale values, which may make output pixels near the image boundaries invalid (up to the size of the neighborhood window). Also, the neighborhood window should not be larger than the input image.

Periodic Noise By Frequency Domain Filtering: These types of filters are used for this purposeBand Reject Filters: It removes a band of frequencies about the origin of the Fourier transformer. Ideal Band reject Filter: An ideal band reject filter is given by the expression

10.a)

Inverse filtering for image restoration

Inverse filtering is a deterministic and direct method for image restoration.

The images involved must be lexicographically ordered. That means that an image is converted to a column vector by pasting the rows one by one after converting them to columns.

An image of size 256×256 is converted to a column vector of size 65536×1.

The degradation model is written in a matrix form, where the images are vectors and the degradation process is a **huge but sparse** matrix.

$$\mathbf{g} = \mathbf{Hf}$$

- In this problem we know $\mathbf{H}$ and $\mathbf{g}$ and we are looking for a descent $\mathbf{f}$.

- The problem is formulated as follows:
  We are looking to minimize the Euclidian norm of the error, i.e.,
  $$\|\mathbf{n}\|^2 = \|\mathbf{g} - \mathbf{Hf}\|^2$$

- The first derivative of the minimization function must be set to zero.
$$\|\mathbf{g} - \mathbf{Hf}\|^2 = (\mathbf{g} - \mathbf{Hf})^T(\mathbf{g} - \mathbf{Hf}) = \left(\mathbf{g}^T - \mathbf{f}^T\,\mathbf{H}^T\right)(\mathbf{g} - \mathbf{Hf}) =$$
$$\mathbf{g}^T\mathbf{g} - \mathbf{g}^T\,\mathbf{Hf} - \mathbf{f}^T\mathbf{H}^T\,\mathbf{g} + \mathbf{f}^T\mathbf{H}^T\,\mathbf{Hf}$$
$$\frac{\partial\|\mathbf{g}-\mathbf{Hf}\|^2}{\partial\mathbf{f}} = -2\mathbf{H}^T\,\mathbf{g} + 2\mathbf{H}^T\,\mathbf{Hf} = \mathbf{0} \Rightarrow \mathbf{H}^T\,\mathbf{Hf} = \mathbf{H}^T\,\mathbf{g}$$
$$\mathbf{H}^T\,\mathbf{Hf} = \mathbf{H}^T\,\mathbf{g}$$
$$\mathbf{f} = \left(\mathbf{H}^T\,\mathbf{H}\right)^{-1}\mathbf{H}^T\,\mathbf{g}$$
- If $\mathbf{H}$ is a square matrix and its inverse exists then $\mathbf{f} = \mathbf{H}^{-1}\mathbf{g}$

Inverse filtering for image restoration in frequency domain

- We have that
$$\mathbf{H}^T\,\mathbf{Hf} = \mathbf{H}^T\,\mathbf{g}$$

- If we take the DFT of the above relationship in both sides we have:
$$|H(u,v)|^2F(u,v) = H(u,v)^*G(u,v)$$
$$F(u,v) = \frac{H(u,v)^*}{|H(u,v)|^2}G(u,v)$$
$$F(u,v) = \frac{G(u,v)}{H(u,v)}$$

- Note that the most popular types of degradations are low pass filters (out-of-focus blur, motion blur).

10. c)

5.4 Periodic noise reduction by frequency domain filtering

5.4.1 Bandreject Filters

Ideal bandreject filter

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

W **is the width of the band**

Butterworth bandreject filter

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

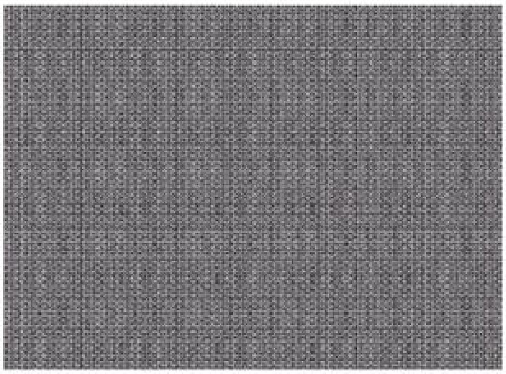
Gaussian bandreject filter

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

5.4.2 Bandpass Filters

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

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



5.4.3 Notch Filters

Ideal notch reject filter

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

$$D_1(u,v) = \left[\, (u - M/2 - u_0)^2 + (v - N/2 - v_0)^2 \,\right]^{1/2}$$
$$D_2(u,v) = \left[\, (u - M/2 + u_0)^2 + (v - N/2 + v_0)^2 \,\right]^{1/2}$$

1.

c) $\lambda_1 = 400 \times 10^{-9}$ m

$\lambda_2 = 750 \times 10^{-9}$ m.

h = Plack's constant $= 6.6 \times 10^{-34}$ J sec

$$E_1 = \frac{hc}{\lambda_1}$$

$$= \frac{6.6 \times 10^{-34} \times 3 \times 10^{8}}{400 \times 10^{-9}} \text{ J}$$

$$= \frac{6.6 \times 10^{-34} \times 3 \times 10^{8}}{400 \times 10^{-9} \times 1.6 \times 10^{-19}} \text{ eV}$$

$$= 3.1 \text{ eV}$$

$$E_2 = \frac{hc}{\lambda_2} = \frac{6.6 \times 10^{-34} \times 3 \times 10^{8}}{750 \times 10^{-9} \times 1.6 \times 10^{-19}}$$

$$= 1.65 \text{ eV}$$

So energy of visible spectrum ranges from 1.65 eV to 3.1 eV.

3.

b) Given:

$$\underline{A} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \underline{u} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

The unitary transform is given as follows:

$$T(\underline{u}) = \underline{A}\,u$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$= \begin{pmatrix} 2.8284 & 4.2426 \\ -1.4142 & -1.4142 \end{pmatrix}$$

4.

c) Using the given defn of Haar matrix, the Haar matrix is given by:

$$H_4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}$$

$$z = \begin{pmatrix} 0 \\ 1/4 \\ 2/4 \\ 3/4 \end{pmatrix}$$

So Haar transform is given as:

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix}\begin{pmatrix} 0 \\ 1/4 \\ 2/4 \\ 3/4 \end{pmatrix}$$

$$= \begin{pmatrix} 3/4 \\ -1/2 \\ -\frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \end{pmatrix}$$

5

b)

r_k	n_k	CDF
0	790	0.19
1	1023	0.44
2	850	0.65
3	656	0.81
4	329	0.89
[illegible]	[illegible]	0.95

Out put of histogram !!

$L = 8$

$\Rightarrow L-1 = 7$

5	243	[illegible]
6	122	0.98
7	81	1

equalization

Round $\left(\frac{y - y_{min}}{y_{max} - y_{min}} (L-1)\right)$

r_k	$CDF(r_k) = y$	
0	0.19	0
1	0.44	2
2	0.65	4
3	0.81	5
4	0.89	6
5	0.95	7
6	0.98	7
7	1	7

sol^n Question 2(c):

$P = (-2, -2) \quad Q = (3, 4)$

$D_e = \sqrt{(3-(-2))^2 + (4-(-2))^2}$

$= \sqrt{5^2 + 6^2}$

$= 7.81$

$D_4 = |3-(-2)| + |4-(-2)|$

$= 5 + 6 = 11$

$D_8 = \max(|3-(-2)|, |4-(-2)|)$

$= 6.$

3.c) Unitary Transformation properties:-

(a) Total energy of original and transformed vectors are equal.

$\| T(x) \|^2 = \| x \|^2$

Mathematically $\| T(\underline{x}) \|_2 = \| \underline{x} \|_2$

(b) Rotation: A unitary transformation is simply a rotation in vector space

(c) <u>Decorrelation</u>:- When input vector elements are highly correlated, the transform coefficients tend to be uncorrelated.

Mathematically, assume $\underline{x} \sim \underline{\Sigma}_x$

$$\Rightarrow \underline{T}\,\underline{x} \sim T^* \Sigma_x T.$$

$$\Rightarrow \underline{T}x \simeq T^* T \underline{\Delta} T^* T$$

Since $T^* T = I$

$$\Rightarrow \underline{T}\,\underline{x} \sim \underline{\Delta}$$ ← uncorrelated

(d) <u>Energy Compaction</u>

Most unitary transforms ... into few of

pack energy ...

$$\underline{\Lambda} = \text{diag}\{\{\lambda_i\}\}.$$

---

4. b) The 1D DCT transform is defined as

$$c(u) = a(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

$$a(u) = \sqrt{1/N}, \quad u = 0$$

$$= \sqrt{2/N}, \quad u = 1, \ldots, N-1$$

In similar manner, the 2D DCT is defined as

$$c(u,v) = a(u)a(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$$\ldots \times \cos\left[\frac{(2x+1)u\pi}{M}\right]$$

$$\times \cos\left[(2y+1)\frac{v\pi}{N}\right]$$

DCT is defined as

The inverse DCT is

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

Properties :-

1. Linearity
2. Excellent Energy Compaction.
3. Spectrum is real.
4. Orthogonality
5. Separability – 2D computations can be separated into 2 1D transforms

---

5.(C)

0	0	0	0	0
0	1	1	1	1
	1	1	1	1
0	0	0	0	0
0	0	0	0	0

Consider the example of this 5×5 grid of a 2-bit image, I, ... $\{0,1\}$

s.t. $I_{ij} \in \{0, 1\}$.

Define $M = 5$, $N = 5$, s.t. $I \in \mathbb{R}^{5 \times 5}$

I:

Sample mean $\mu_I = \frac{1}{25} \sum_{i=1}^{5} \sum_{j=1}^{5} I_{ij}$

$= 8/25 = 0.32$

II:

Sample variance $\sigma_I^2 = \frac{1}{MN} \sum_{i=1}^{5} \sum_{j=1}^{5} (I_{ij} - \mu_I)^2$

$= \frac{1}{25} \times 8 \times 0.68^2$

$= 0.148$

φ
