Final Scheme of Valuation/Answer Key

(Scheme of evaluation (marks in brackets) and answers of problems/key)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, JUNE 2023(2019 Scheme)

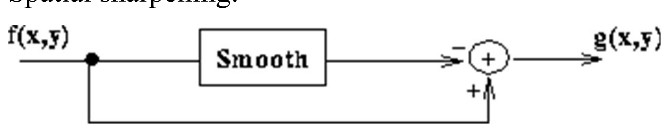
Course Code: CST438

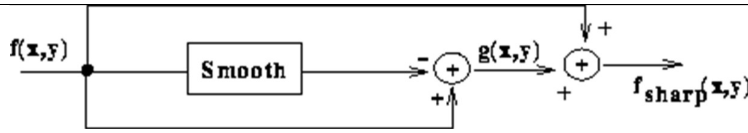
Course Name: IMAGE PROCESSING TECHNIQUE

Max. Marks: 100		**Duration: 3 Hours**

PART A

		Answer all questions, each carries 3 marks.	Marks
NOTE:		***The answers given are only a guide to the evaluators, to be read and marks allotted based on the student answers. There may be different approaches to a same answer. The notations and variables used may differ in case of equations.***	
1		(i) Gray Level: refers to a scalar measure of intensity that ranges from Black (0) to White (255) and in-between having different shades of gray. {Due to processing storage and hardware consideration, the number gray levels typically is an integer power of 2. L=2k Then, the number, b, of bites required to store a digital image is B=M *N* k When M=N, the equation become b=N^2 *k When an image can have 2k gray levels, it is referred to as "k- bit". An image with 256 possible gray levels is called an "8- bit image" (256=28).} (ii) Hue – Color attribute that describes a pure color Saturation – Gives a measure of the degree to which a pure color is diluted by white light.	(1) (1) (1)
2		Brief Points given below: Photopic Vision: Bright Light Vision, Vision due to Cone receptors. Scotopic Vision: Dim Light Vision, Vision due Rod receptors	 (1.5) (1.5)

3		Properties of 2D Fourier Transform. **(If an attempt has been made – may allot 2 marks)** **(with atleast 3 common properties given – may allot 3 marks)** Separability, Periodicity, Conjugate Property, Translation, Rotation, Distributivity and Scaling, Convolution and Correlation, Laplacia.	(3)
4		**[Equation – 1 M ; Steps – 1 M, Final Answer – 1 M]** **[If the student has Attempted – may allot 2 M]** DFT: $F[k] = \sum_{x=0}^{N-1} f(x)e^{-j2\pi kx/N}$ Where, k = 0, 1, 2, 3 and N = 4 f(x) = (1 0 0 1} F[0] = 2 F[1] = 1+j F[2] = 0 F[3] = 1 –j => F[k] = { 2, 1+j, 0,1-j}	(1) (2)
5		Steps for filtering in frequency domain: {step 1 & 6 : 1 Mark, other steps together: 2 Marks} 1. Multiply the input image by $(-1)^{x+y}$ to center the transform to u = M/2 and v = N/2. 2. Compute F(u,v) , the DFT of the image 3. Multiply F(u,v) by the selected filter function H(u,v) 4. Compute the inverse DFT of the result in 3. 5. Obtain the real part of the result in 4. 6. Multiply the result in 5 by $(-1)^{x+y}$ to cancel the multiplication in step 1.	 (1) (2)
6		**{There can be different approaches to answer the same. If found correct full marks to be allotted}** The unsharp filter is a simple sharpening operator . The unsharp filtering technique is commonly used in the photographic and printing industries for crispening edges. Spatial sharpening:  The complete unsharp filtering operation:	 (1) (1)

		 $f_{sharp}(x,y) = f(x,y) + k * g(x,y)$ {where k is a scaling constant. Reasonable values for k vary between 0.2 and 0.7, with the larger values providing increasing amounts of sharpening.}	(1)
7		**[Note: If a brief on Mean Filters has been written – may allot 2 M (out of 3)]** Ans: 1. Arithmetic Mean filter 2. Harmonic Mean filter **3.** Geometric Mean filter	(1) (1) (1)
8		[**Definition – 1 M ; Approaches listing – 2 M]** Image segmentation is the division of an image into regions or categories, which correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories There are three general approaches to segmentation, (Any two to be written) (i) Thresholding (ii) Edge-based methods (iii) Region-based methods.	(1) (2)
9		**{ if ONLY definition of both given: 2 Marks}** **{if difference stated clearly : 3 Marks}** Dilation and Erosion are basic morphological processing operations that produce contrasting results when applied to either gray-scale or binary images. • **Dilation:** Dilation is the reverse process with regions growing out from their boundaries. Dilation is A XOR B. • **Erosion:** Erosion involves the removal of pixels ate the edges of the region. Erosion is just the dual of Dilation. Both dilation and erosion are produced by the interaction of s set called a structuring element(SE). **Difference between Dilation and Erosion: (Any three points to be written)**	(3)

Dilation	Erosion
It increases the size of the objects.	It decreases the size of the objects.
It fills the holes and broken areas.	It removes the small anomalies.
It connects the areas that are separated by space smaller than structuring element.	It reduces the brightness of the bright objects.
It increases the brightness of the objects.	It removes the objects smaller than the structuring element.
Distributive, duality, translation and decomposition properties are followed.	It also follows the different properties like duality etc.
It is XOR of A and B.	It is dual of dilation.
It is used prior in Closing operation.	It is used later in Closing operation.
It is used later in Opening operation.	It is used prior in Opening operation.

10

{Explanation – 1 M; Figure – 1 M; Uses -1M}

Chain code:

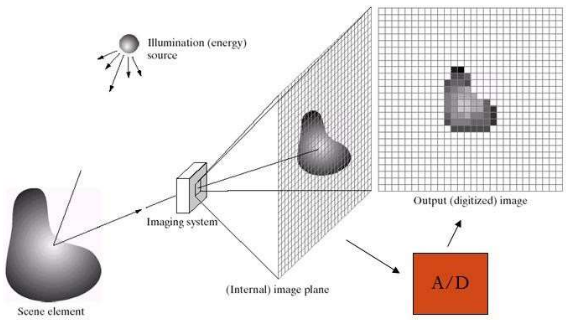
The co-ordinates of any continuous boundary of an object can be represented as a string of numbers where each number represents a particular direction in which the next point on the connected line is present. One point is taken as the reference/starting point and on plotting the points generated from the chain, the original figure can be re-drawn. (1)

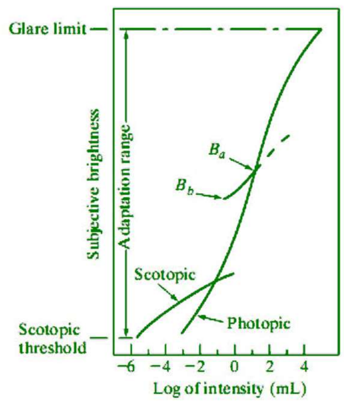
The chain code can be either a 4-connected neighbourhood or a 8-connected neighbourhood.

Have two major classes: absolute chain code and relative chain code.

			(1)
		Use of Chain code: Chain code is a lossless compression technique used for representing an object in images. or Chain code is a method of describing the shape of the boundary of an object in an image.	(1)
		PART B	
		Answer any one full question from each module, each carries 14 marks.	
		Module I	
11	a)	**[Interpolation Explanation – 2 M** **Types of interpolation Techniques: (ANY TWO to be written) – 6 :-** **(3 marks for each technique; For Explanation : 2 Marks , Equation + Plot : 1 Mark)]** **Interpolation** is the process of using known data to estimate values at unknown locations. This works in two directions and tries to achieve the best approximation of a pixel's intensity based on the values of surrounding pixels. As it's an approximation method image will always lose some quality when interpolated. Image interpolation occurs especially when an image is resized or distorted(remapped) from a one-pixel grid to another.	(2)
		1. Nearest Neighbor Interpolation This is the most basic form of interpolation. The nearest Neighbor algorithm only considers one pixel, the closest one to the interpolated point. {This requires the least processing time of all the interpolation algorithms. And has the effect of simply making each pixel bigger.} 2. Bilinear Interpolation Considers the closest 2x2 neighborhood of known pixel values (total 4 pixels) surrounding the unknown pixel and then takes the weighted average of these values to assign the unknown pixel.	(3+3)

		{This will create smoother-looking images than the nearest neighbor and needs more processing time.} 3. Bicubic Interpolation Bicubic Interpolation Considers the closest 4x4 neighborhood of known pixel values (total of 16 pixels) surrounding the unknown pixels. Since the known pixels are at various distances from the unknown pixel, closer pixels will give higher weighting. {This produces noticeably sharper images than the nearest neighbor and bilinear interpolations.}	
	b)	**{Explanation – 4 M; Figure – 2 M}** A Simple Image Model: An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be nonzero and finite; that is o<f(x,y) <co.	(1)
		The function f(x,y) may be characterized by two components- The amount of the source illumination incident on the scene being viewed. (a) The amount of the source illumination reflected back by the objects in the scene These are called illumination and reflectance components and are denoted by i(x,y) an r (x,y) respectively.	(1)
		The functions combine as a product to form f(x,y). We call the intensity of a monochrome image at any coordinates (x,y) the gray level (l) of the image at that point l= f (x, y.) $L_{min} \leq l \leq L_{max}$; L_{min} is to be positive and L_{max} must be finite $$L_{min} = i_{min}\ r_{min}$$ $$L_{max} = i_{max}\ r_{max}$$	(1)
		The interval [L_{min}, L_{max}] is called gray scale. Common practice is to shift this interval numerically to the interval [0, L-l] where l=0 is	(1)

		considered black and l= L-1 is considered white on the gray scale. All intermediate values are shades of gray or gray varying from black to white.  a b c d e FIGURE An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.	(2)
		OR	
12	a)	**{Explanation – 4 M; Plot – 2M; Significance of Webber Ratio – 2M}** **{If an attempt has been made and the answer is found conceptually correct – 5 M (Out of 8) }** Brightness Adaptation: the ability to distinguish differences in brightness. This appears to vary according to the wavelength of light. Brightness Discrimination: an apparent decrease in the intensity of a stimulus after exposure to a high level of incident illumination. Plot:	(2) (2) (2)

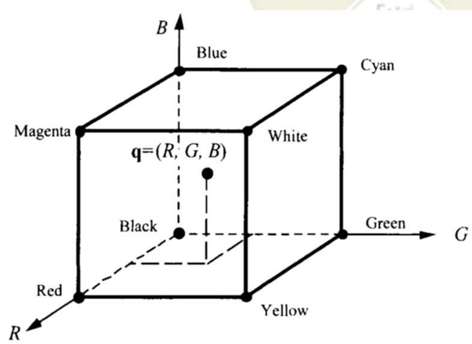


(2)

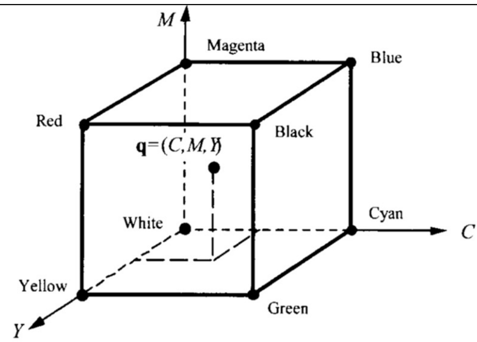
Significance of Webber Ratio: A small value of Weber ratio means Good brightness discrimination. A large value of Weber ratio means Poor brightness discrimination.

b) Color Models: **{ANY TWO models needs to be explained; 3 Marks for each Model with 2 M for explanation and 1 M for the diagrammatic representation of the model)** (3+3)

1. RGB Color Model - The most commonly employed color space is the RGB color space, which is based on the additive mixture of three primary colors R, G ,and B.



2. CMY Color Model - In the printout of a color image with a color printer, the CMy(K) color space, with the subtractive primary colors cyan, magenta, and yellow as well as possibly an additional black (Karbon), is used. A subtractive color space forms the basis for the printing process.

		 **3.** YIQ Color Model – Color coordinate system with the coordinates Y, I , and Q was defined for transmission purposes. To transmit a color signal efficiently, the RNGNBN signal was more conveniently coded from a linear transformation. The luminance signal is coded in the Y-component. The additional portions I (in-phase) and Q (quadrature) contain the entire chromaticity information that is also denoted as chrominance signal in television technology. I and Q are transmitted by a much shorter waveband since the Y signal contains by far the largest part of the information. The Y signal contains no color information so that the YZQ system remains compatible with the black-white system. By using only the Y signal in a black-and-white television, gray-level images can be displayed, which would not be possible by a direct transmission of the RNGNBN signal. **4.** HSI Color Model In the *HSI* color space *hue, saturation,* and *intensily* are used as coordinate axes. This color space is well suited for the processing of color images and for visually defining interpretable local characteristics.	
		Module II	
13	a)	**[Note: The equation given in the scheme is for guidance, Marks to be awarded if the student writes the correct equation, variables and symbols may differ.** **[DCT – Explanation – 3 M; Equation – 1M]** The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub- bands) of differing importance (with respect to the	

		image's visual quality). It transforms a signal or image from the spatial domain to the frequency domain.	(4)
		The general equation for a 2D (*N* by *M* image) DCT is defined by the following equation: $$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i).\Lambda(j).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$ and the corresponding *inverse* 2D DCT transform is simple $F^{-1}(u,v)$, i.e.:where $$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$	
		[For the basic operations of DCT - If attempted can be given 3 M (Out of 4).] A Brief of some basic operation of the DCT for reference: • The input image is N by M; • f(i,j) is the intensity of the pixel in row i and column j; • F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix. • For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. • Compression is achieved since the lower right values represent higher frequencies,and are often small - small enough to be neglected with little visible distortion. • For DCT input is an 8 by 8 array of integers. This array contains each pixel's grayscale level; The input image is divided into blocks.	(4)

b) The Fourier transform can be represented as F = Wf where f is the input and W is the DFT matrix. (1)

Taking N = 4, we form a DFT matrix

W = [1 1 1 1 ; 1 –j -1 j ; 1 -1 1 -1 ; 1 j -1 –j] (1)

To chec whether W is unitary or not, the following relationship is checked

W. W*' = 1 (1)

W =

1	1	1	1
1	-j	-1	j
1	-1	1	-1
1	j	-1	-j

W*' = (1)

1	1	1	1
1	j	-1	-j
1	-1	1	-1
1	-j	-1	j

WW*' = (1)

4	0	0	0
0	4	0	0
0	0	4	0
0	0	0	4

- ⇨ WW*' = 4 [I] , hence unitary transform.
- ⇨ 4 indicates that the DFT matrix is not normalized. This is compensated by using 1/N in the forward or inverse transform. (1)

Or

⇨ ¼

4	0	0	0
0	4	0	0
0	0	4	0
0	0	0	4

=>

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

		OR	
14	a)	**[Process of Computation – 2M; DCT Kernel and transpose – 3M; Steps – 2M; Final Answer – 1 M]** 2D DCT of f(x,y) = $\begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$ F = Kernel * f(x,y) * Kernel ^{T} DCT Kernel ➔ $\begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{pmatrix}$ Kernel ^{T} ➔ $\begin{pmatrix} 0.5 & 0.6532 & 0.5 & 0.2706 \\ 0.5 & 0.2706 & -0.5 & -0.6533 \\ 0.5 & -0.2706 & -0.5 & 0.6533 \\ 0.5 & -0.6532 & 0.5 & -0.2706 \end{pmatrix}$ **F**= $\begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{pmatrix} \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.6532 & 0.5 & 0.2706 \\ 0.5 & 0.2706 & -0.5 & -0.6533 \\ 0.5 & -0.2706 & -0.5 & 0.6533 \\ 0.5 & -0.6532 & 0.5 & -0.2706 \end{pmatrix}$ Intermediate steps……. = $\begin{pmatrix} 6 & 0.3025 & -1 & 0.9235 \\ 0 & -0.1463 & -0.3825 & -0.3532 \\ 0 & 0 & 0 & 0 \\ 0 & -0.3533 & -0.923 & -0.8535 \end{pmatrix}$	(8)
	b)	**[Explanation – 4M ; Equation – 1M; Matrix form – 1 M]** **[If the student has given the proper explanation without equation but the matrix form has been given – may allot the full 6M accordingly]** **[The answer given below is for reference only, variations may be there]** Hadamard Transform : The Hadamard transform works similarly to the Fourier transform: it takes a vector and maps it to its frequency components, which are the Walsh functions. Instead of sine waves in the Fourier transform, the Walsh functions are discrete "square waves". This makes it easier to compute.	(6)

The 2-D Hadamard transform:

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

=>

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

Inverse 2-D Hadamard transform:

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

=>

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

MATRIX FORM

The size of a Hadamard matrix is a power of two, 2^n x 2^n. We can build a Hadamard matrix for a given n recursively.

$$H_1=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

And then build block matrices in the same pattern to get

$$H_n=\begin{pmatrix}H_{n-1} & H_{n-1}\\ H_{n-1} & -H_{n-1}\end{pmatrix}$$

Module III

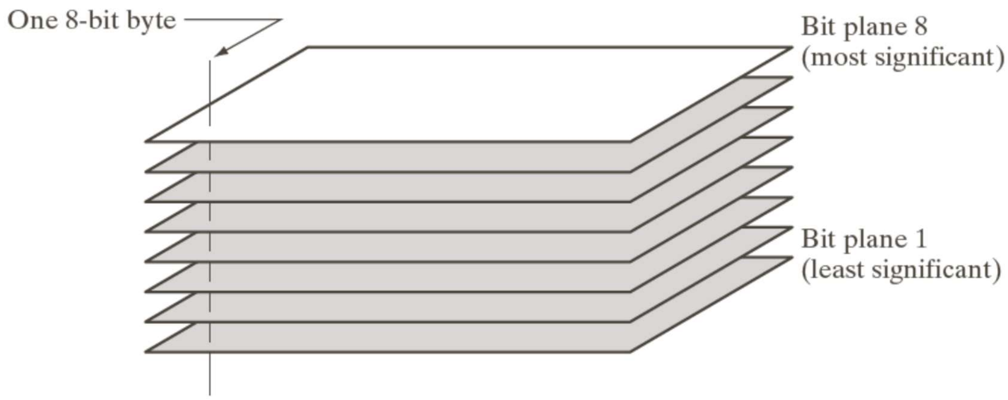
15 a) Enhancing an image using a standard Spatial averaging filter of size 3x3.

$$I=\begin{pmatrix}1 & 2 & 1 & 2\\ 1 & 2 & 12 & 2\\ 3 & 2 & 1 & 2\\ 3 & 3 & 2 & 1\end{pmatrix}$$

{Filter Kernal / Mask – 2 M; Steps – 2 M; Final Answer- 1M}

		{Effects of border/ side pixels – 2M} Standard Average Filter Mask: $1/9\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}$ $\begin{pmatrix}2&1&2\\2&12&2\\2&1&2\end{pmatrix}x1/9\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}$ =1/9 [2(1) +1(1)+2(1)+2(1)+12(1)+2(1)+2(1)+1(1)+2(1) = 2.88 => 3 The pixel with value 12 is replaced by value 3 in the processed image. This process is repeated for all pixels in the input image I. Final answer to be given. [**NOTE: 2 Marks may be given if attempted**.] The effects of border/ side pixels during the process of filtering: At the edges of an image we are missing pixels to form a neighbourhood How can they be overcome? There are a few approaches to dealing with missing edge pixels: – Omit missing pixels • Only works with some filters • Can add extra code and slow down processing – Pad the image • Typically with either all white or all black pixels – Replicate border pixels – Truncate the image – Allow pixels *wrap around* the image Can cause some strange image artefacts	
	b)	**{Explanation – 2M ; Figure – 1M; Problem Solving – 3 M}** Bit Plane Slicing Bit Plane Slicing (BPS) is a method of expressing an image in which each pixel is represented by one or more bits of the byte. To incorporate hidden data in any slice of eight slices, the BPS approach requires a bit slicing algorithm. Each pixel is represented by 8 bits in general.	(2)

Here the contributions made by each bit is highlighted.



(1)

Demonstration:

$$B = \begin{pmatrix} 4 & 3 & 3 & 6 \\ 3 & 3 & 6 & 4 \\ 3 & 6 & 4 & 4 \\ 6 & 4 & 4 & 4 \end{pmatrix}$$

(1)

Each pixel is represented using 4 bits.

$$\Rightarrow \begin{pmatrix} 0100 & 0011 & 0011 & 0110 \\ 0011 & 0011 & 0110 & 0100 \\ 0011 & 0110 & 0100 & 0100 \\ 0110 & 0100 & 0100 & 0100 \end{pmatrix}$$

(2)

⇨ The corresponding Four Bit Planes:

$$\Rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

OR

16 a) **{Each filter – 3 M with Explanation – 2M and Figure / Plot – 1M}**

Smoothing Frequency Domain Filters:

1. IDEAL LOW-PASS FILTER

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

$H(u,v) = 1$, if $D(u,v) \le D_0$ 0, if $D(u,v) > D_0$

(2)

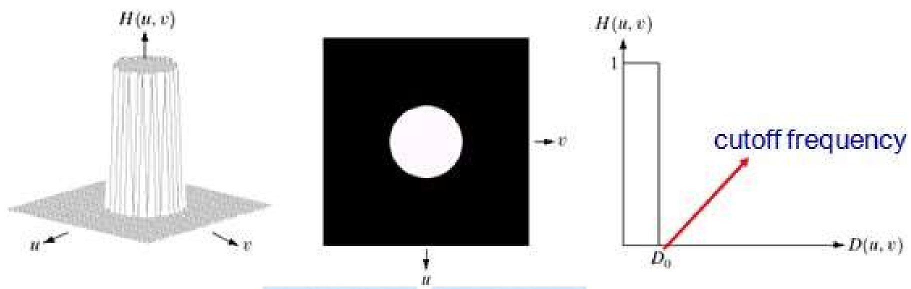
Where D0 is a positive constant and D(u,v) is the distance between a point (u,v) in thefrequency domain and the center of the frequency rectangle; that is

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

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

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

VISUALIZATION: IDEAL LOW PASS FILTER:



(1)

2. BUTTERWORTH LOW-PASS FILTER

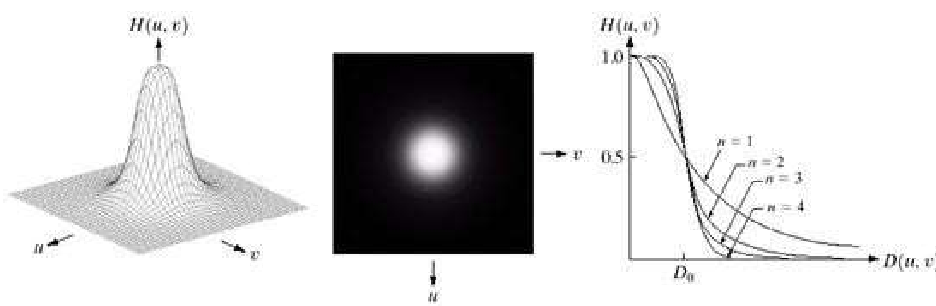
Transfer function of a Butterworth lowpass filter (BLPF) of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as (2)

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

-

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

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



(1)

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

3. GAUSSIAN LOWPASS FILTERS

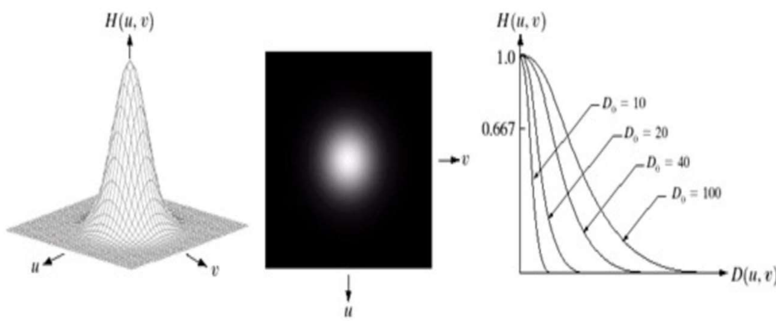
The form of these filters in two dimensions is given by (2)

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

Where D_0 is the cutoff frequency. When $D(u,v) = D_0$, the GLPF is down to 0.607 of its maximum value. This means that a spatial Gaussian filter, obtained by computing the IDFT of above equation., will have no ringing.

- This transfer function is smooth , like Butterworth filter.
- Gaussian in frequency domain remains a Gaussian in spatial domain.

(1)



b) **[NOTE: If attempted – may be given 3 M (Out of 5)**

Spatial convolution:

I = [1 , 2 , 4 , 1] h= [1 2 1]

{Explanation – 3 M

Solving of Example – 2 M}

Process:

The filter is flipped and passed across the sequence. Since the given filter is symmetric the mask and its flipped version remains the same. (1)

$$g(x) = h * f(x) = \sum_{i=-n}^{n} h(i) f(x-i)$$

g(x): output ; h: filter ; * means convolution, (2)

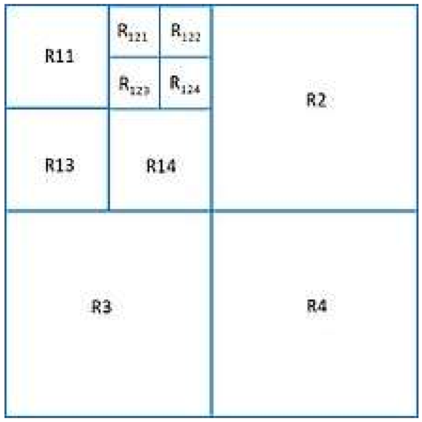
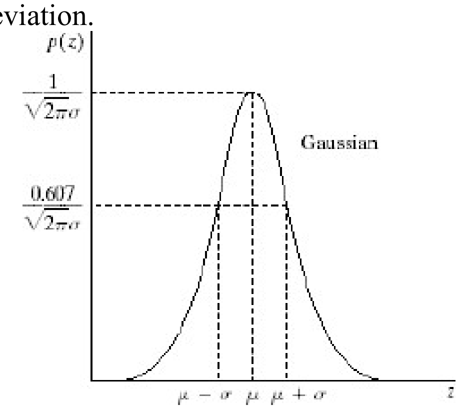
f(x): input, n = |_ width of filter / 2 _|

|_ _|: rounds down, for example: |_ 1.7_| = 1

For example: Filter (h): [1 2 1]

		width = 3 => n=1 ; h(-1) = 1 ; h(0) = 2 ; h(1) = 1; Given : Mask h = [1 2 1] I = [1 2 4 1] For n = 1 i=-1 => f(x-(-1))= f(x+1)=4 i=0 => f(x-0)= f(x)=2 i=1 => f(x-1)= f(x-1)=1 ⇨ 1x4+2x2+1x1 = 4 +4+1=9 ⇨ Hence, in the processed image f(x) is replaced by 9. ⇨ After Normalizing it is : 9/3 = 3. ⇨ Avoiding edge pixels : ⇨ [__ , 9 , 11, __] ➔ [__, 3, 3, __].	(2)
		Module IV	
17	a)	[**NOTE: N_4(p) – is generally used to refer to 4 neighbourhood representation. What is needed is the students have to design the MIN and MAX filter masks by giving 0 to the diagonal pixels.** • **If an attempt is made – May allot 5 (Out of 8)** • **If an attempt is made and partially the answer/ approach is correct may allot 8 (Out of 8 marks)]** **{SPLIT UP: MIN filter Mask – 1 M;** **MAX filter mask – 1M;** **Problem Solving with MIN filter – 3 M (2 M for steps and 1 M for final answer.** **Problem Solving with MAX filter – 3 M (2 M for steps and 1 M for final answer }** N_4(p), 3x3 MIN Filter mask: $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	(1) (1)

		$N_4(p)$, 3x3 MAX Filter mask: $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ **I** = $\begin{pmatrix} 10 & 11 & 11 & 12 \\ 13 & 13 & 24 & 13 \\ 13 & 3 & 12 & 13 \\ 3 & 10 & 12 & 10 \end{pmatrix}$ Process: I* MIN filter ➔ $\begin{pmatrix} 10 & 11 & 11 \\ 13 & 13 & 24 \\ 13 & 3 & 12 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ ➔ MIN (_11,_13, 13,24,_,3,_) ➔ 3 Centre pixel value 13 in the image is replaced by value 3 in processed image. Final Output: $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 11 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$; This is Omitting pixels at the edges.(any method can be used , accordingly the final output will vary) Similarly, Considering the MAX Value the final Output of processing with MAX filter: Process … to be given…… $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 24 & 24 & 0 \\ 0 & 13 & 24 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$; This is Omitting pixels at the edges.(any method can be used , accordingly the final output will vary)	(2) (1) (2) (1)
	b)	**Split and merge segmentation** : technique used to segment a image into its homogeneous parts.. The image is successively split into quadrants based on a **homogeneity criterion** and similar regions are merged to create the segmented result. The technique incorporates a **quadtree** data structure - there is a parent-child node relationship. The total region is a parent, and each of the four splits is a child. Algorithm: • Define the criterion to be used for homogeneity • Split the image into equal size regions	(2) (2)

		• Calculate homogeneity for each region • If the region is homogeneous, then merge it with neighbors • The process is repeated until all regions pass the homogeneity test 	(2)
		OR	
18	a)	Noise Models: ANY THREE Models: 3 Marks for each model with 1 M for equation and 1 M for Plot. 1 M for the name of the noise model. If only Names of all models listed – then 3 M. **Gaussian Noise:** These noise models are used frequently in practices because of its tractability in both spatial and frequency domain. The PDF of Gaussian random variable is $$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$ Where z represents the gray level, μ= mean of average value of z, σ= standard deviation. 	(3+3+3)

Rayleigh Noise:

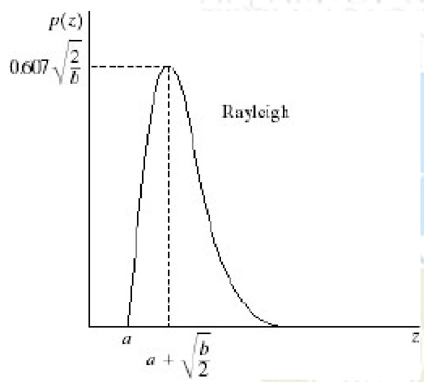
Unlike Gaussian distribution, the Rayleigh distribution is no

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

symmetric. It is given bythe formula.

The mean and variance of this density is

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



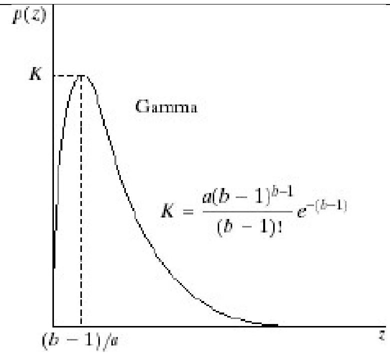
Gamma Noise:

The PDF of Erlang noise is given by

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

The mean and variance of this density are given by

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



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

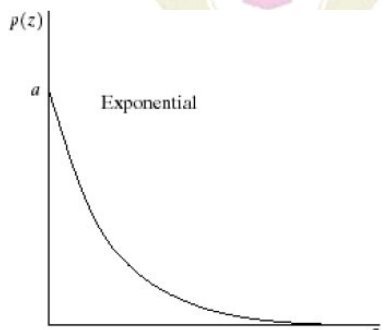
Exponential Noise:

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

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

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

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



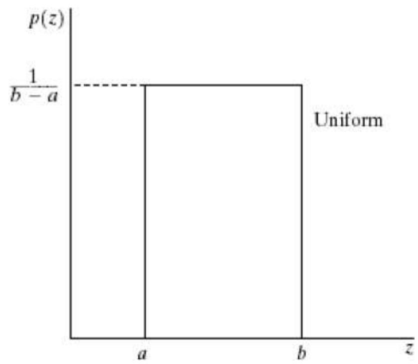
Uniform Noise:

The PDF of uniform noise is given by

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

The mean and variance of this noise is

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

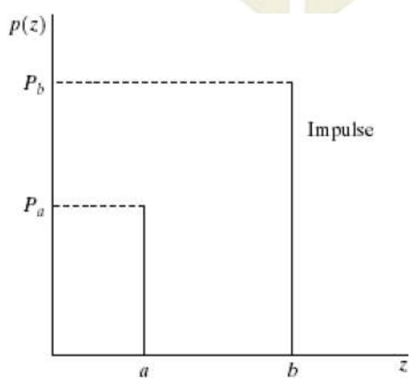


Impulse (salt & pepper) Noise:

In this case, the noise is signal dependent, and is multiplied to the image.The PDF of bipolar (impulse) noise is given by

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

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



b) Steps in Otsu's Algorithm: {**marks to be given based on correctness of the algorithm and steps included.**} (5)

1. Compute the normalized histogram. Denote the components of the histogram by *pi*, for *i*=1 to *L*-1
2. Compute the cumulative sums, *P1(k)*, for *k*=1 to *L*-1 using P1(k)

3. Compute the cumulative means – average intensity up to level *k*, *m(k)* for *k*=1 to *L*-1 using *m(k)*
4. Compute the global mean intensity mG using mG
5. Compute the between class variance
6. Obtain the Otsu threshold *k** as the value of *k* for which the between-class variance is maximum . If the maximum is not unique, obtain *k** by averaging the values of *k* corresponding to the various maxima detected.
7. Obtain the separability measure at *k*=*k**

Module V

19 a) i) **Importance of Hit-Miss function – 2 M.**

locates particular configuration of pixels. (2)

i.e., extracts each pixel located in a neighborhood exactly matching the template defined by the structuring element.

[**NOTE: For the problem part**

- **4 M (Out of 6) Marks may be given if attempted**
- **6 M (Out of 6) Marks may be given accordingly if the process is explained correctly]**

{Explanation given is only for reference }

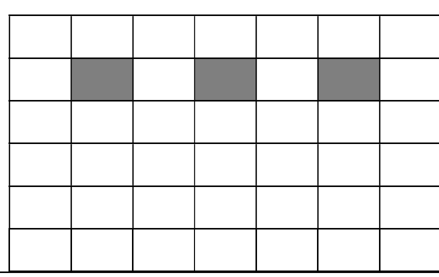
Explanation: In mathematical morphology, hit-or-miss transform is an operation that detects a given configuration (or pattern) in a binary image, using the morphological erosion operator and a pair of disjoint structuring elements. The result of the hit-or-miss transform is the set of positions where the first structuring element fits in the foreground of the input image, and the second structuring element misses it completely.

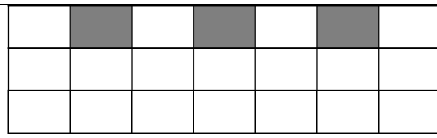
Let B_1 and B_2 be two structuring elements satisfying $B1 \cap B2 = \emptyset$

The hit-or-miss transform of a given image A by B is given by:

$A \otimes B = (A \ominus B1) \cap (Ac \ominus B2)$ (6)

ii) Output Image:



			
	b)	**Opening** and **Closing** are dual operations used in Digital Image Processing for restoring an eroded image. Opening is generally used to restore or recover the original image to the maximum possible extent. Closing is generally used to smoother the contour of the distorted image and fuse back the narrow breaks and long thin gulfs. Closing is also used for getting rid of the small holes of the obtained image. The combination of **Opening** and **Closing** is generally used to clean up artifacts in the segmented image before using the image for digital analysis. Opening Process: Erosion followed by Dilation $A \circ B = (A \ominus B) \oplus B$ (definition – 1 M ; equation – 1M) Closing Process: Dilation followed by Erosion. $A \bullet B = (A \oplus B) \ominus B$ (definition – 1 M ; equation – 1M)	(1) (1) (2) (2)
		OR	
20	a)	Regional Descriptors: **{ If only names listed atleast 4 – 6 M (out of 8) } else** **{If an overall explanation has been given – 4 M (out of 8)}** **{ If the student has given atleast 4 descriptors with definition then – 8 Marks)** Perimeter , Area , Shape factor , Range, Median, Mean, Variance, Standard Deviation, Coef. Of Variation, Skewness, Kurtosis; Topological descriptors and Texture descriptors.	(2 *4 = 8)

	b)	Minimum Perimeter Polygon: **{ Explanation - 4 M and Algorithm – 2 M}** The goal is to represent the shape in a given boundary using the fewest possible number of sequences.	(1)
		Method: Enclose boundary in a grid. Allow boundary to shrink. The vertices of the polygon are all inner or outer corners of the grid. Traverse the 4-connected boundary of the circumscribed shape. Concave vertices on this boundary have "mirrors" on the outer boundary. The boundary is described by inner convex and outer concave vertices.	(2)
		Definitions: Form a list whose rows are the coordinates of each vertex and whether that vertex is W or B. The concave verttices must be mirrored, the vertices must be in sequential order, and the first uppermost, leftmost vertex V_O is a W vertex. There is a white crawler (W_C) and a black crawler (B_C). The Wc crawls along the convex W vertices, and the B_C crawls along the mirrored concave B vertices.	(1)
		MPP Algorithm: 1. Set $W_C=B_C=V_O$ 2. (a) V_K is on the positive side of the line (V_L,W_C) [$sgn(V_L,W_C,V_K)>0$ (b) V_K is on the negative side of the line (V_L,W_C) or is collinear with it [$sgn(V_L,W_C,V_K)<=0$; V_K is on the positive side of the line (V_L,B_C) or is collinear with it [$sgn(V_L,B_C,V_K)>=0$ (c) V_K is on the negative side of the line (V_L,B_C) [$sgn(V_L,B_C,V_K)<0$ If condition (a) holds the next MPP vertex is W_C and $V_L=W_C$; set $W_C=B_C=V_L$ and continue with the next vertex. If condition (b) holds V_K becomes a candidate MPP vertex. Set $W_C=V_K$ if V_K is convex otherwise set $B_C=V_K$. Continue with next vertex. If condition (c) holds the next vertex is B_C and $V_L=B_C$.	(2)

		Re-initialize the algorithm by setting $W_C=B_C=V_L$ and continue with the next vertex after $V_{L.}$ 3. Continue until the first vertex is reached again.	
****			