Indian Institute of Information Technology Allahabad
End Sem Question Paper

Course Name: Image and Video Processing **Course Instructor/ Co-ordinator:** Prof. Anupam/Prof. Vrijendra Singh/Dr. Navjot Singh
Course Code: PC-IT-IVP303 **Program Name(s):** B.Tech. (IT) – 5th sem
Exam Date: 27/11/2024 **MM:** 40

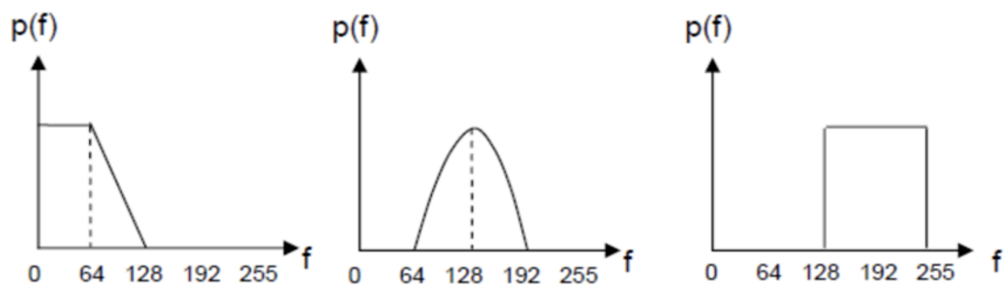
Duration: 3 hrs. **Maximum Marks: 40**

Instructions:

- *All questions are compulsory. All the subparts of a question are to be attempted together.*
- *A basic calculator is allowed.*

Q1. Answer the following. **[8]**

a. The histograms of the three images are illustrated below. For each image, sketch a transformation function in the figure below that will help to equalize the histogram.



Solution:

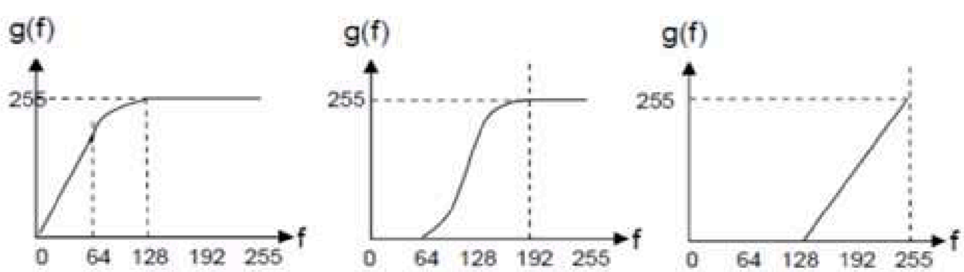


Figure 1A Figure 1B Figure 1C

b. For the image below, find a transformation function that will approximately equalize its histogram, draw the transformed image and give the histogram of the processed image. Assume that the processed images can only take values from 0 to 7.

0	0	1	1	2
0	1	1	2	4

1	1	2	4	5
1	3	4	5	6
3	3	5	6	7

Solution:

b)

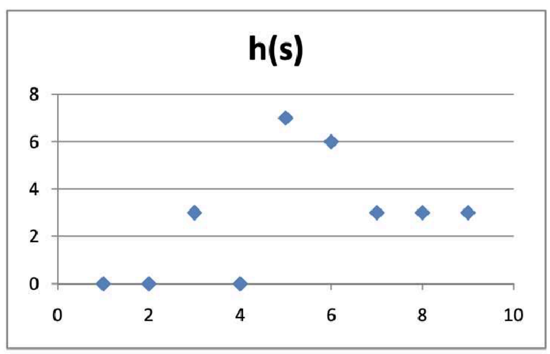
Histogram equalization:

r	h(r)	H(r)	s =(int(7*H(r)/25))	s	h(s)
0	3	3	1	0	0
1	7	10	3	1	3
2	3	13	4	2	0
3	3	16	4	3	7
4	3	19	5	4	6
5	3	22	6	5	3
6	2	24	7	6	3
7	1	25	7	7	3

Resultant image:

1	1	3	3	4
1	3	3	4	5
3	3	4	5	6
3	4	5	6	7
4	4	6	7	7

Histogram of Resultant Image:



Q2. Apply Hit-or-Miss transform over the binary image of size 5×5 with the following pixel values (1 for foreground and 0 for background) using the structuring element containing the object (value 1) and background (value 0).
[6]

0	0	0	0	0
0	1	1	1	0

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

Image

0	1	0
1	1	1
0	1	0

Structuring Element

Solution:

Q3. An image of size 4×4 contains the following grayscale intensity values: **[6]**

10	12	11	10
25	27	26	24
45	47	46	44
80	82	81	79

Apply K-means clustering with $K = 2$ clusters to segment the image based on intensity values. Assume the initial cluster centroids are $C_1 = 15$ and $C_2 = 50$.

a. Perform one iteration of K-means clustering.
b. Determine the updated cluster centroids after one iteration.
c. Show the segmented matrix of the image after one iteration.

Solution:

Q4. For the 0-7 gray level image given below: **[6]**

0	5	3	3	2

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

a. Resample the image by interpolating to obtain 9×9 image.

Solution:

a)

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

b. Threshold the image with thresholding rules specified below to convert it to a binary image. *Thresholding rule is defined as cumulative histogram of the biggest valued pixel divided by number of gray levels.*

Solution:

b) Threshold value is 1 / 8 = 0.125

Histograms	*Probability density function*	*After Thresholding*
H(0) = 3	K(0) = 0.12	T(0) = 0
H(1) = 4	K(1) = 0.16	T(1) = 1
H(2) = 2	K(2) = 0.08	T(2) = 0
H(3) = 3	K(3) = 0.12	T(3) = 0
H(4) = 4	K(4) = 0.16	T(4) = 1
H(5) = 2	K(5) = 0.08	T(5) = 0
H(6) = 3	K(6) = 0.12	T(6) = 0
H(7) = 4	K(7) = 0.16	T(7) = 1

Original image

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

Thresholded image

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

c. Rotate the binary image 180º by applying the bellowed rotational transformation matrix defined as:

cos θ	-sin θ	
	sin θ	cos θ

Solution:

c) Rotational Transformation Matrix is:

-1	0	0
	0	-1

1 valued pizel coordinates:
(1,1), (1,2), (1,3), (1,4),
(2,1), (2,2),
(3,1), (3,2), (3,3), (3,4),
(4,1), (4,2)

Multiply transformation matrix with all 1 valued coordinate of vector.
(1,1), (1,2), (1,3), (1,4) => (-1,-1), (-1, -2), (-1,-3), (-1,-4)
(2,1), (2,2) => (-2,-1), (-2, -2)
(3,1), (3,2), (3,3), (3,4) => (-3,-1), (-3, -2), (-3,-3), (-3,-4)
(4,1), (4,2) => (-4, -1), (-4, -2)

Shift all values by 4 =>
(-1,-1), (-1, -2), (-1,-3), (-1,-4) => (3,3), (3,2), (3,1), (3,0)
(-2,-1), (-2, -2) => (2,3), (2,2)
(-3,-1), (-3, -2), (-3,-3), (-3,-4) => (1,3), (1,2), (1,1), (1,0)
(-4,-1), (-4, -2) => (0,3), (0,2)

Finally, locate the 1 valued pixels to specified coordinates.

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

Q5. Write short notes on any three of the following: **[6]**

a. Weiner filter for image restoration

Minimum Mean Square Error Filtering (Wiener Filtering)

This approach incorporates both the degradation function and statistical characteristics of noise into the restoration process.

Image and noise are random processes

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

The objective is to find an estimation for *f* such that e^2 is minimized

1. $\hat{F}(u,v)$ = Fourier transform of the estimate of the undegraded image.
2. $G(u,v)$ = Fourier transform of the degraded image.
3. $H(u,v)$ = degradation transfer function (Fourier transform of the spatial degradation).
4. $H^*(u,v)$ = complex conjugate of $H(u,v)$.
5. $|H(u,v)|^2 = H^*(u,v)H(u,v)$.
6. $S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise [see Eq. (4-89)]†
7. $S_f(u,v) = |F(u,v)|^2$ = power spectrum of the undegraded image.

$$\begin{aligned}
\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v) \\
&= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v) \\
&= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)
\end{aligned}$$

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

$S_f(u,v) = |F(u,v)|^2 = \text{power spectrum of the undegraded image}$

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

Constant

Unknown

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

b. Canny edge detection method

Solution:

The process of Canny edge detection algorithm can be broken down to five different steps:

- Apply Gaussian filter to smooth the image in order to remove the noise

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2} \right); 1 \le i, j \le (2k+1)$$

- Find the intensity gradients of the image

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$

- Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
- Apply double threshold to determine potential edges
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

c. Hough transform

Solution:

Elegant method for direct object recognition

- Edges need not be connected
- Complete object need not be visible
- Key Idea: Edges VOTE for the possible model

Image and Parameter Spaces

Equation of Line: $y = mx + c$

Find: (m, c)

Consider point: (x_i, y_i)

$y_i = mx_i + c \quad or \quad c = -x_i m + y_i$

Parameter space also called Hough Space

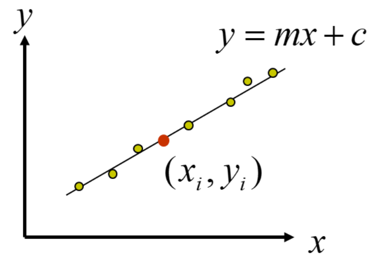
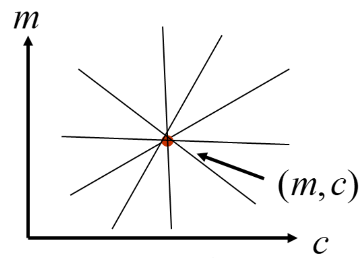


Image Space



Parameter Space

Line Detection by Hough Transform

Algorithm:

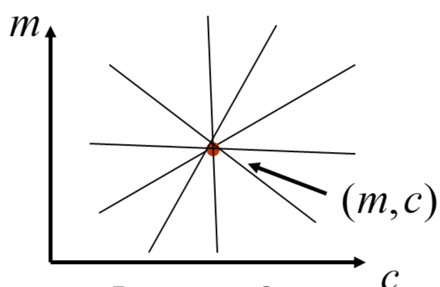
- Quantize Parameter Space (m, c)
- Create Accumulator Array $A(m, c)$
- Set $A(m, c) = 0 \quad \forall m, c$
- For each image edge (x_i, y_i) increment:

$$A(m, c) = A(m, c) + 1$$

- If (m, c) lies on the line:

$$c = -x_i m + y_i$$

- Find local maxima in $A(m, c)$



Parameter Space

$A(m, c)$

	1						1		
		1				1			
			1		1				
				2					
			1		1				
		1				1			
	1						1		

d. Morphological algorithm for Convex Hull

Solution:

A set A is said to be convex if the straight line segment joining any two points in A lies entirely within A. The convex hull H of an arbitrary set S is the smallest convex set containing S.

Let B^i, $i = 1,\ 2,\ 3,\ 4$, represent the four structuring elements.

The procedure consists of implementing the equation:

$$X_k^i = (X_{k-1} * B^i) \cup A$$
$$i = 1,2,3,4 \ \text{ and } \ k = 1,2,3,\ldots$$

with $X_0^i = A$.

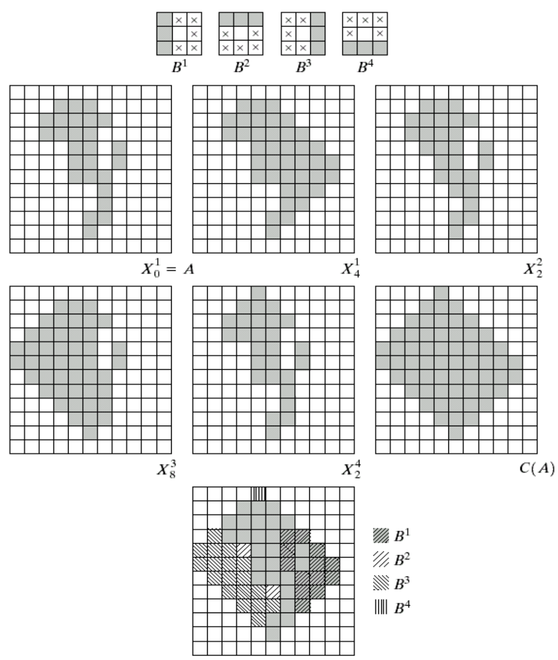
When the procedure converges, or $X_k^i = X_{k-1}^i$, let $D^i = X_k^i$,

the convex hull of A is

$$C(A) = \bigcup_{i=1}^{4} D^i$$

a
b c d
e f g
h

FIGURE 9.19 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.



Q6. Huffman encoding is used to compress a grayscale image of size 256×256, which has 5 intensity levels with the following probabilities of occurrence: **[8]**

Intensity Level	0	1	2	3	4
Probability	0.4	0.2	0.2	0.1	0.1

a. Construct the Huffman tree and generate the Huffman codes for each intensity level.
b. Calculate the average code length (in bits) for the Huffman-encoded image.
c. Determine the size of the Huffman-compressed image in bytes.

d. Compare the Huffman-compressed image size with the original uncompressed image size (where each intensity level is stored using 3 bits per pixel) and compute the compression ratio.

Solution:

Original Image Size (in bits):

$$\text{Original Size} = \text{Number of Pixels} \times \text{Bits per Pixel}$$

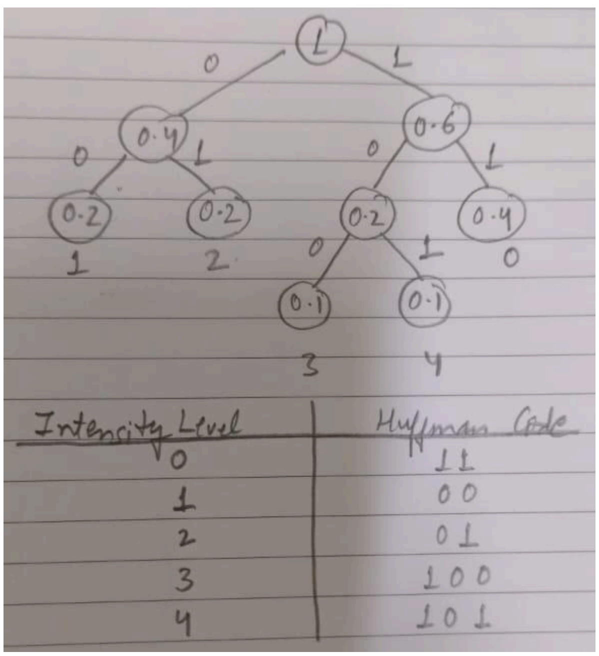
Huffman Compressed Size (in bits):

$$\text{Compressed Size} = \text{Number of Pixels} \times \text{Average Code Length}$$

Compression Ratio:

$$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}}$$

Note: Huffman tree and its code may vary depending upon what is put of left and right side. But the length of the code for every intensity level must be like: intensity-length (0-2, 1-2, 2-2, 3-3, 4-3)



1. **Huffman Codes**: The Huffman tree generates the following codes for each intensity level:

Intensity Level	Probability	Huffman Code
0	0.4	11
1	0.2	00
2	0.2	01
3	0.1	100
4	0.1	101

2. **Average Code Length**:

$$\text{Average Code Length} = (0.4 \times 2) + (0.2 \times 2) + (0.2 \times 2) + (0.1 \times 3) + (0.1 \times 3) = 2.2 \, \text{bits}$$

3. **Image Sizes**:

   - **Original Uncompressed Size**:

$$\text{Original Size (bits)} = 256 \times 256 \times 3 = 196,608 \, \text{bits} \, (24,576 \, \text{bytes})$$

   - **Huffman Compressed Size**:

$$\text{Compressed Size (bits)} = 256 \times 256 \times 2.2 = 144,179.2 \, \text{bits} \, (18,022.4 \, \text{bytes})$$

4. **Compression Ratio**:

$$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}} = \frac{196,608}{144,179.2} \approx 1.36$$