**1. Answer the following short questions in brief.**

**a) What are 4-, 8-, and m-adjacency?**

Slide 2 Page 55

**b) What is intensity slicing in pseudo-colour images?**

Slide 9 page 38,40-41

**c) What is Adaptive Median Filtering?**

To overcome the drawbacks of the standard median filter, the adaptive median filter is used. noise detection is utilized to determine impulse noise and the degree of noise interference in sub-images. Based on the degree of noise interference, window size could be selected, and the noise found in sub-images is removed by means of improved median filter. Adaptive median filter can suppress noise very effectively while preserving fine details very well and has better filtering performance than the standard median filter. It also shows a great advantage over the standard one in filtering speed.

Slide 6 Page 34 (adaptive filtering), 39-42 (median starts here Pg. 39).

**d) What is Chain Code representation of a digital image?**

A chain code is a lossless compression algorithm for monochrome images. The basic principle of chain codes is to separately encode each connected component, or "blob", in the image.

For each such region, a point on the boundary is selected and its coordinates are transmitted. The encoder then moves along the boundary of the region and, at each step, transmits a symbol representing the direction of this movement.

This continues until the encoder returns to the starting position, at which point the blob has been completely described, and encoding continues with the next blob in the image.

This encoding method is particularly effective for images consisting of a reasonably small number of large connected components.

OR

Slide 10 Page 6

**e) What is unsharp masking?**

Unsharp masking (USM) is an image sharpening technique, often available in digital image processing software. Its name derives from the fact that the technique uses a blurred, or "unsharp", negative image to create a mask of the original image.[1] The unsharp mask is then combined with the original positive image, creating an image that is less blurry than the original. The resulting image, although clearer, may be a less accurate representation of the image's subject. In the context of signal processing, an unsharp mask is generally a linear or nonlinear filter that amplifies the high-frequency components of a signal.

OR

Gonzalez 2nd ed. page 132

**2. a) Discuss the algorithm for histogram equalization.**

Slide 3 Page 23 or <http://www.mee.tcd.ie/~ack/teaching/1e8/histogram_equalisation_slides.pdf>

A **histogram** of a digital image represents intensity distribution by plotting a bar graph with X-axis as pixel intensity value and Y-axis as the frequency of its occurrence.

**Histogram Equalisation** is a technique to **adjust contrast levels and expand the intensity range** in a digital image. Thus, it enhances the image which makes information extraction and further image processing easier.

Following is the algorithm to do histogram equalisation :-

1) Convert the input image into a grayscale image 2) Find frequency of occurrence for each pixel value i.e. histogram of an image (values lie in the range [0, 255] for any grayscale image) 3) Calculate Cumulative frequency of all pixel values 4) Divide the cumulative frequencies by total number of pixels and multiply them by maximum graycount (pixel value) in the image

**b) What is a Mexican Hat filter for edge detection?**

One of the most efficient edge detectors is the Mexican hat filter, also known as the LoG filter (since it involves taking the Laplacian of the Gaussian filtered image). It is a high-pass filter and uses the Gaussian filter for smoothing and noise removal and the Laplacian for edge detection. The weights are distributed in the gliding window. The filter computes a new image which contains great positive values on one side of each steep edge in the original image. On the other side there are great negative values. The location where the computed values cross the horizontal line corresponding to zero values corresponds exactly to the location of the edge in the original image. Detecting the zero crossings makes it possible to produce thin lines representing the edges. The lines can be approximated by polygons or subdivided into digital straight segments thus making the analysis of shapes possible.Gonzalez 2nd Ed. pdf page 593-594.

**c) What are morphological Open & Close operations? Design the morphological operations required for boundary extraction of an object.**

Slide 8 Page 4,24-29. Page 40-41. **AND** Gonzalez Ch 9

**3. a) Briefly discuss about Wiener Filtering for noise removal in frequency domain**

Slide 6, Page 53. Pdf page 273 (2nd ed).

**b) What is first derivative filtering? Derive the convolution masks for a first derivative filter. What is Sobel’s edge detection operator?**

Slide 4 Page 37, 54-56. Page 57. Pdf page 145

**4. a) Briefly discuss Gaussian, Rayleigh and Gamma noise models.**

Slide 6 Page 17,18,19.

**b) Discuss the effects of Harmonic and contra harmonic mean filters for the removal of salt and pepper noise.**

Slide 6 Page 8-9.

**5. Write the algorithm for construction of a Huffman Tree. Use Huffman Coding to encode/decode the text “my name is” followed by your own name in lower case. For example, the text will be “my name is sourav ganguly”, if your name is “Sourav Ganguly”. Create the frequency table, Huffman tree and show the encoding and decoding steps with examples (consider space/blank as a character).**

Use the algorithm below or slide JPEG Compression, pg-25.

**Begin**

**define a node with character, frequency, left and right child of the node for Huffman tree.**

**create a list ‘freq’ to store frequency of each character, initially, all are 0**

**for each character c in the string do**

**increase the frequency for character ch in freq list.**

**done**

**for all type of character ch do**

**if the frequency of ch is non zero then**

**add ch and its frequency as a node of priority queue Q.**

**done**

**while Q is not empty do**

**remove item from Q and assign it to left child of node**

**remove item from Q and assign to the right child of node**

**traverse the node to find the assigned code**

**done**

**End**

[**https://people.ok.ubc.ca/ylucet/DS/Huffman.html**](https://people.ok.ubc.ca/ylucet/DS/Huffman.html) **⇒** Follow the animation to get the make your own huffman tree.

To Generate encoding

**if a left child of node n ≠φ then**

**traverseNode(leftChild(n), code+’0’) //traverse through the left child**

**traverseNode(rightChild(n), code+’1’) //traverse through the right child**

**else**

**display the character and data of the current node.**

Use the above algorithm.