|  |  |
| --- | --- |
| **Fourth Year Mechatronics Department**  **CSE496-Elective Course (2): Digital Image Processing**  **Mid Term Exam**  **Time Allowed 90 minutes.**  **Spring 2015** | **شعار الكلية**  **Ain Shams University**  **Faculty of Engineering** |
| **Student Name:** | **Student ID:** |

Exam Consists of 5 Questions in 6 Pages. **Students are not allowed to use any ready-made MATLAB functions related to image filtering or histogram operations. Assume any Missing Information. Each Question is Weighted by 20 MARKs.**

1. **Supply the missing words:-**
2. Image processing is used to improve image quality for human preception and computer interpretation.
3. Computer vision defined as a discipline in which the input is an image and the output is a description.
4. The three types of computerized process are 1) low level processing, 2) mid level processing , and 3) high level processing.
5. Digitization means converting continuous sensed data into a digital form.
6. Optical power is 1/f , with f measured in meters. The unit is called the diopeter
7. The human eye brings nearby points into focus on the retina by contracting muscles attached to the lens, thus increasing the power of the lens.
8. If the intensity values of an image range from 0 to 2X10-7, then a logarithmic transformation is recommended.
9. Setting the least significant bit to zero does not affect much the appearance of the image.
10. Sharpening the images is commonly accomplished by performing a spatial differentiation of the image field.
11. Histogram equalization lightens dark images.
12. The median filter is used to remove the salt and pepper noise.
13. Edges are detected by applying a gradient operator, then thresholding.

|  |
| --- |
| **EXTRA SPACE** |
|  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Assume the histogram of the opposite image is modeled/approximated by the Gaussian function where and are the mean and standard deviation of that image, and ***r {Imin, …, Imax}***. Answer the following questions:- | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **8** | **8** |
| **9** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **8** |
| **8** | **9** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **7** | **8** | **9** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **6** | **7** | **8** | **9** | **1** | **2** | **3** | **4** | **5** | **6** |
| **5** | **6** | **7** | **8** | **9** | **1** | **2** | **3** | **4** | **5** |
| **4** | **5** | **6** | **7** | **8** | **9** | **1** | **2** | **3** | **4** |
| **3** | **4** | **5** | **6** | **7** | **8** | **9** | **1** | **2** | **3** |
| **3** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **1** | **2** |
| **3** | **3** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **1** |

* 1. Find and Plot ***pr(r)*** assuming r {1,2,3,4,5,6,7,8,9}. Note that the function must satisfy the PMF conditions.

|  |  |
| --- | --- |
| Solution | |
| =  Normalizing so that sum is 1 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | | 1 | 0.05 | 0.05 | 0.05 | | 2 | 0.08 | 0.09 | 0.14 | | 3 | 0.12 | 0.13 | 0.27 | | 4 | 0.14 | 0.15 | 0.42 | | 5 | 0.16 | 0.17 | 0.59 | | 6 | 0.14 | 0.15 | 0.74 | | 7 | 0.12 | 0.13 | 0.87 | | 8 | 0.08 | 0.09 | 0.96 | | 9 | 0.05 | 0.05 | 0.999 | |

* 1. Adjust the image intensities such that its new histogram matches the PMF/PDF: . You must consider the Gaussian distribution given in (a) as the original image histogram. Find the ***r-z*** map as well as the value of the parameter A.

|  |  |
| --- | --- |
| Solution | |
| Given   * Assuming that is a PDF   To calculate the CDF  For histogram matching | The map is then   |  |  | | --- | --- | | r | z | | 1 | 0.05 | | 2 | 0.15 | | 3 | 0.31 | | 4 | 0.54 | | 5 | 0.89 | | 6 | 1.35 | | 7 | 2.04 | | 8 | 3.22 | | 9 | 6.91 |   Normalizing the map so that the values of z are from 0 to 9   |  |  | | --- | --- | |  |  | | 1 | 1 | | 2 | 1 | | 3 | 1 | | 4 | 2 | | 5 | 2 | | 6 | 3 | | 7 | 3 | | 8 | 5 | | 9 | 9 | |

1. Write a MATLAB function that reads an input image and a mask size nXn. The function carries out a median filter on the given image. Express the time taken by that function in terms of the parameter n. Important: You are not allowed to use the sort() MATLAB function.

|  |
| --- |
| Solution |
| |  |  | | --- | --- | | %Converts 2D matrix to a vector  function [v]=Mat2Vect(im)  [h w]=size(im);  v=zeros(h\*w,1);  c=1;  for x=1:h  for y=1:w  v(c)=im(x,y);  c=c+1;  end  end | %Sorts a vector and returns the median  function [y]=MySort(v)  [n dum]=size(v);  %Implemting Bubble Sorting  for i=1:n-1  for j=i+1:n  if v(i)>v(j)  temp=v(i);  v(i)=v(j);  v(j)=temp;  end  end  end  y=v(ceil(n/2)); | | %Applies median filter on an image  function [y]=MyMedian(im,n)  [h w]=size(im);  a=floor(n/2);  v=zeros(n\*n,1);  y=im;  for i=1+a:h-a  for j=1+a:w-a  ims=im(i-a:i+a,j-a:j+a);  %Converting matrix to a vector  [v]=Mat2Vect(ims);  y(i,j)=MySort(v);  end  end | %A file for testing the filter  clc;  clear all;  close all;  im=imread('lena.jpg');  im=rgb2gray(im);  J = imnoise(im,'salt & pepper',0.02);  figure  imshow(J)  [y]=MyMedian(J,3);  figure  imshow(y) |   For Bubble sorting, there are two for loops. So, the time is proportional to n2 where nXn is the size of the filter. |

1. A) Write a MATLAB function that applies a mask of size nXn on an input image. B) Use the function in (A) to write another function to find the edge map of an image using the Compass operator.

|  |
| --- |
| Solution |
| function output = imfilter2(im, fl)  [m, n ] = size(im);  [k, l ] = size(fl);  output = zeros(m,n);  for x = ceil(k/2):m-floor(k/2)  for y = ceil(l/2):n-floor(l/2)  sum = 0;  for i = 1:k  for j = 1:l  sum = sum +im(x-ceil(k/2)+i, y - ceil(l/2)+j)\*fl(i,j);  end  end  output(x,y) = sum;  end  end  end  function output = compass(im)  [m, n] = size(im);  r1 = imfilter2(im,[1 1 1;  0 0 0;  -1 -1 -1]);  r2 = imfilter2(im,[-1 -1 -1;  0 0 0;  1 1 1]);  r3 = imfilter2(im,[0 1 1;  -1 0 1;  -1 -1 0]);  r4 = imfilter2(im,[0 1 1;  -1 0 1;  -1 -1 0]);  r5 = imfilter2(im,[-1 -1 0;  -1 0 1;  0 1 1]);  r6 = imfilter2(im,[1 1 0;  1 0 -1;  0 -1 -1]);  r7 = imfilter2(im,[1 0 -1;  1 0 -1;  1 0 -1]);  r8 = imfilter2(im,[-1 0 1;  -1 0 1;  -1 0 1]);  output = zeros(m,n);  for x = 1: m  for y = 1:n  output(x,y) = max([r1(x,y), r2(x,y), ...  r3(x,y), r4(x,y), ...  r5(x,y), r6(x,y), ...  r7(x,y), r8(x,y)]);  end  end  end |

1. Using the backward difference approximation, derive a 3D mask for the Un-sharp Masking process. Note that the input of this process is a 3D image.

|  |
| --- |
| Solution |
| * Let the image be * From (1) * For images * For unsharp masking |