National University of Computer and Emerging Sciences, Lahore Campus

Course:	Fundamentals of Computer Vision	Course Code:	CS495
Program:	BS(Computer Science)	Semester:	Fall 2016
Duration:	180 Minutes	Total Marks:	80
Paper Date:	15-Dec-16	Weight	45%
Section:	ALL	Page(s):	8
Exam:	Final Exam	Roll No:	

Instruction/Notes:

Attempt all questions in the space provided to you.
Be Exact & to the point. There are no marks for **stories.**
Please don't attach extra sheet with this paper.

Problem 1: (2+ 8=10 marks)

Describe when it would be suitable to use the Prewitt mask. Sketch the result when one of these masks is applied to the greyscale image below. You only need to compute the values for the central region of the image that is fully covered by the mask. Give answers correct to 2 decimal places.

0	0	0	0	0
0	2	3	4	0
0	1	2	5	0
0	4	2	1	0
0	0	0	0	0

Horizontal

0	0	0	0	0
0	-5	-6	5	0
0	-7	-3	7	0
0	-4	-1	4	0
0	0	0	0	0

Vertical

0	0	0	0	0
0	-3	-8	-7	0
0	-1	2	4	0
0	3	8	7	0
0	0	0	0	0

Problem 2: (7 marks)

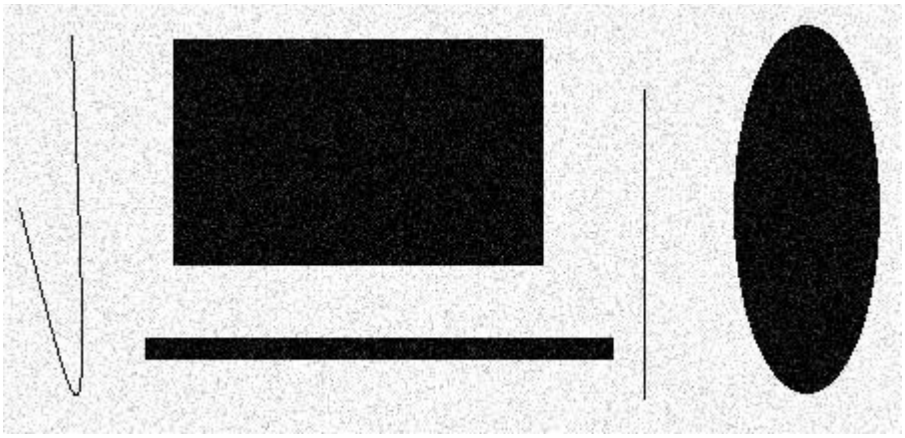
Given the image below before (fig. a) and after (fig. b) a smoothing filter was applied. The size of the filter is a small square (its size is rather small compared to the image size). In your opinion, which one of the following filter types most likely produced the image (fig. b):

1) Mean filter,

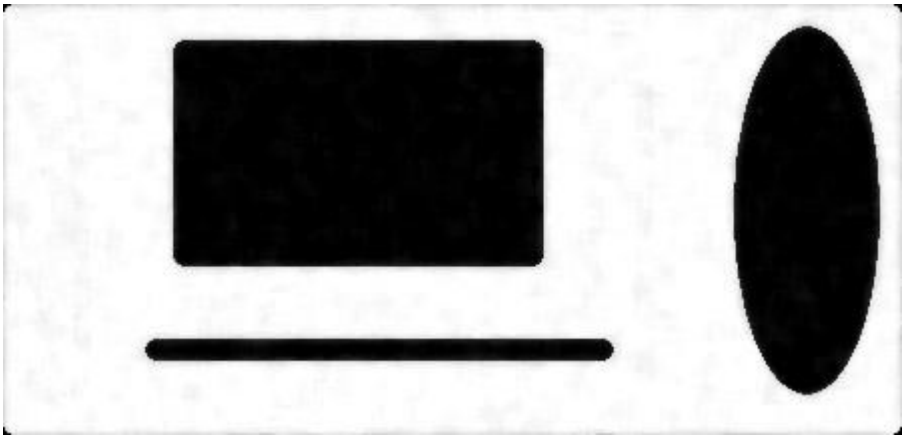
2) Median filter,

3) Gaussian filter. Motivate your answer. No marks without proper justification

Median filter. As thins lines are removed



(a)



(b)

Problem 3: (2+4+2+4+1= 13 marks)

Compare the Canny edge detector and the Laplacian-of-Gaussian (LoG) edge detector for each of the following questions.

(a) Which of these operators is/are rotationally invariant and which is/are non-rotationally invariant?

LoG is isotropic and Canny is non-isotropic.

(b)Canny algorithm depends on two parameters, what are they & how they affect the output?

Canny requires σ and two thresholds for hysteresis.
If σ is high more blur if less then less blur
Too High thresholds → miss data
Too Low thresholds→ more noise

(c) Which of these two operators takes 2nd derivative? Any one (which one) or both?

LoG is 2nd derivative

(d) What is the basis for both these algorithms?

Blur the image prior to edge detection to mitigate the noise effects

(e) Which of these is a Mexican hat operator?

LoG

Problem 4: (20 marks)

Suppose you are given the following image (data12.jpg) of size 100x200 [100 rows and 200 columns]. Line in the image divides the image into two equal halves. What will be the output of the following code? Discuss in detail.

data12.jpg

OutPut:


```
X= imread('data12.jpg');
Spatial= rgb2gray(X);
freqA= fft2(Spatial(:,1:100));
shiftedA= fftshift(freqA);

freqB= fft2(Spatial(:,101:200));
shiftedB= fftshift(freqB);

[X,Y] = meshgrid(-50:49,-50:49);
Z= sqrt(X.^2+Y.^2);
C= Z<25;
D= Z>25;

A= shiftedA.*C;
B= shiftedB.*D;

shifted_back=ifftshift(A);
f(:,1:100)= ifft2(shifted_back);

shifted_back=ifftshift(B);
f(:,101:200)= ifft2(shifted_back);
```

```
cmin = min(min(abs(f)));
cmax = max(max(abs(f)));
figure, imshow(abs(f), [cmin cmax]), colormap gray
```

Problem 5: (15 marks)

Suppose you are given an image I of size 300x300. You have to convolve the image with Gaussian filter to blur it. But unfortunately your convolution method is not working. Only way to do it is by converting it in frequency domain. Write Matlab code to blur the image by Gaussian filter in frequency domain.

You can use fspecial to generate Gaussian filter.

`h = fspecial('gaussian', hsize, sigma)` returns a rotationally symmetric Gaussian lowpass filter of size `hsize` with standard deviation `sigma` (positive). `hsize` can be a vector specifying the number of rows and columns in `h`, or it can be a scalar, in which case `h` is a square matrix. The default value for `hsize` is `[3 3]`; the default value for `sigma` is 0.5.

```
X= imread('elephant.jpg');
size(X)
Spatial= rgb2gray(X);
imshow(Spatial);

freq= fft2(Spatial,800,800);
shifted= fftshift(freq);
figure, imshow(log(1+abs(shifted)),[]);


g = fspecial('gaussian', 800,3);
h1=fft2(g);
figure, imshow(log(1+abs(h1)),[]);
HS=fftshift(h1);
figure, imshow(log(1+abs(HS)),[]);

C=shifted.*HS;



shifted_back=fftshift(C);
f= ifft2(shifted_back);
cmin = min(min(abs(f)));
cmax = max(max(abs(f)));
f = f(400:800,400:800);
figure, imshow(abs(f), [cmin cmax]), colormap gray
```

Problem 6: (15 marks)

Apply isodata (iterative self-organizing data analysis technique) algorithm to determine the threshold of the following image using limit = 7. Show each step

20	200	202	150
10	105	170	250
4	150	185	140
90	2	200	110

Threshold = 116