ECE 5273
Test 2

Wednesday, April 23/30, 2008
4:30 PM - 5:45 PM

Spring 2008 Name: SOLUTION

Dr. Havlicek Student Num: ____________

Directions: This is an open notes, open book test. You have 75 minutes to complete the test. All work must be your own.

SHOW ALL OF YOUR WORK for maximum partial credit!

GOOD LUCK!

SCORE:

1. (25) ________
2. (25) ________
3. (25) ________
4. (25) ________

TOTAL (100):

____

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this test.

Name: ____________ Date: ____________

1. **25 pts**. True or False. Mark *True* only if the statement is **always** true.

TRUE	FALSE	
	X	(a) **3 pts**. To implement the linear convolution of two $N \times N$ images using pointwise multiplication of FFT's, it is generally necessary to periodically extend the images to a size of $2N \times 2N$ before the FFT's are calculated.
	X	(b) **3 pts**. The *Difference of Gaussians* filter, or DoG filter, is an example of a high-pass filter.
	X	(c) **3 pts**. Homomorphic filters are most useful for reducing additive white noise.
X		(d) **3 pts**. For a white noise digital image, the autocorrelation function is given by the Kronecker delta $\delta[m,n]$.
	X	(e) **2 pts**. The psuedo-Weiner filter is a popular choice for solving the classical image restoration problem because it produces a solution that is always at least as good as the best regularization algorithm.
	X	(f) **2 pts**. The median filter is a low-pass filter that is especially effective for reducing additive Gaussian white noise.
	X	(g) **2 pts**. The gray scale morphological DILATE filter is identical to the gray scale order statistic MIN filter.
X		(h) **2 pts**. In anisotropic diffusion, it is usually important to apply a smoothing filter in order to make the diffusion coefficients less sensitive to high frequency noise.
	X	(i) **2 pts**. The main advantages of the gradient based edge detectors is low computational complexity and low sensitivity to noise.
	X	(j) **3 pts**. Huffman coding, quantization, and run-length coding were selected for the JPEG image compression standard because the combination of these three techniques guarantees that the compression ratio reaches the theoretically optimal value for a lossless code.

2. **25 pts.**

(a) **15 pts.** Discuss the Laplacian-of-Gaussian (LoG or $\nabla^2 G$) edge detector.

- Discuss the conceptual steps that the detector performs.
- Explain the purpose of each step and the rationale behind it.
- Explain how the LoG edge detector is usually implemented in practice. Be sure to explain why it is typically unnecessary to implement the spatial derivatives of the input image explicitly (**hint:** see part (b) below).

A conceptual block diagram for the LoG is shown on page 7.32 of the notes.

- The first step is to apply a low-pass Gaussian filter G. This filter attenuates noise & fine scale undesired edges such as those due to texture, so that only the desired larger scale edges will be detected.
- The second step is to apply the Laplacian (2nd derivative) operator to the Gaussian filtered image. Edges are associated with a change in concavity, which corresponds to zero crossings in the Laplacian.
- The third step is to detect (e.g. Mark) the zero crossings in the Laplacian of the Gaussian filtered image. If significant noise is present, it may be necessary to apply a threshold to the gradient magnitude of the LoG filtered image to reject spurious edges.

⇒ In practice, we implement the LoG by applying the Laplacian to the Gaussian filter and then discretizing. Then the DFT of the image can be multiplied with the DFT of the $\nabla^2 G$ filter function, making it unnecessary to differentiate the image.

Problem 2 cont...

(b) **10 pts.** For the 1-D continuous-time case, where convolution is given by

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\, d\tau = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)\, d\tau,$$

prove that differentiation commutes with convolution, *e.g.*, prove that

$$\frac{d}{dt}[x(t) * h(t)] = x'(t) * h(t) = x(t) * h'(t).$$

Chain rule: $\frac{d}{dt}x(u) = \frac{d}{du}x(u)\frac{du}{dt}$

$\Rightarrow$ with $u = t-\tau$,

$$\frac{d}{dt}x(t-\tau) = \left[\frac{d}{d(t-\tau)}x(t-\tau)\right]\left[\frac{d}{dt}(t-\tau)\right] = x'(t-\tau)\cdot 1$$
$$= x'(t-\tau)$$

Likewise, $\frac{d}{dt}h(t-\tau) = h'(t-\tau)$.

Now, $\frac{d}{dt}\left[x(t)*h(t)\right] = \frac{d}{dt}\int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,d\tau$

$$= \int_{-\infty}^{\infty} x(\tau)\left[\frac{d}{dt}h(t-\tau)\right]d\tau = \int_{-\infty}^{\infty} x(\tau)h'(t-\tau)\,d\tau.$$
$$= x(t) * h'(t)$$

Similarly, $\frac{d}{dt}\left[x(t)*h(t)\right] = \frac{d}{dt}\int_{-\infty}^{\infty} x(t-\tau)h(\tau)\,d\tau$

$$= \int_{-\infty}^{\infty}\left[\frac{d}{dt}x(t-\tau)\right]h(\tau)\,d\tau$$
$$= \int_{-\infty}^{\infty} x'(t-\tau)h(\tau)\,d\tau = x'(t) * h(t)$$

All together:

$$\frac{d}{dt}\left[x(t)*h(t)\right] = x'(t)*h(t) = x(t)*h'(t),$$

e.g. Differentiation commutes with convolution.

3. **25 pts.** Consider the 11×11 gray scale image $\mathbb{I}$ shown below, where each pixel takes a value in the range $0 \le I(i,j) \le 255$:

$\mathbb{I}_1 =$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	45	45	0	0	0	0	0
0	0	0	0	45	45	0	0	0	0	0
0	0	0	45	45	45	45	45	0	0	0
0	0	0	45	45	45	45	45	0	0	0
0	0	0	45	45	45	45	45	0	0	0
0	0	0	45	45	45	45	45	0	0	0
0	0	0	45	45	45	45	45	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

This image is a template showing the shape of a part that will be fabricated in a factory. In the figure above, the outline of the part is shown as a heavy black line; this line is shown only to help you interpret the image. The heavy line is **not** actually part of the digital image data.

At the factory, the image will be thresholded to determine the shape of the part that is to be manufactured. Therefore, what is important about the image $\mathbb{I}_1$ is that the background has small pixel values and the "object" (the part) has large pixel values. The actual pixel values themselves are **not** important as long as the object shape can easily be separated from the background.

The image $\mathbb{I}_1$ is transmitted from the design lab to the factory over an unreliable wireless communication channel. During transmission, the image is corrupted with additive noise. The image $\mathbb{I}_2$ that is actually received at the factory is shown in the top figure on the next page.

Notice that many pixels in $\mathbb{I}_2$ have been changed slightly. That is of no concern. However, what is a problem is that some pixels have been dramatically changed with the result that the shape obtained by thresholding $\mathbb{I}_2$ is no longer correct.

Use a gray scale mophological filter to correct the shape of the part in the received image $\mathbb{I}_2$. Explain and justify your choice of filter. Show the corrected image. Handle edge effects by replication. For the structuring element (window), use $\mathbb{B} = \text{SQUARE}(9)$. This structuring element is shown below on the next page. There are also several blank 11×11 images on the following page that you may use for work space.

Problem 3 cont…

$\mathbb{I}_2 =$

0	0	1	0	0	2	0	0	0	0	1
0	0	0	2	0	2	0	2	0	1	0
0	0	2	1	45	45	0	0	2	0	0
1	0	0	0	46	45	1	0	0	2	3
1	1	0	46	45	45	44	45	0	0	2
1	0	2	45	45	44	45	46	0	0	3
2	2	3	46	45	3	45	45	0	3	2
2	2	2	45	44	2	1	46	0	0	4
1	2	1	44	45	46	1	45	0	1	0
0	2	0	0	1	1	0	2	0	0	1
0	1	1	0	0	0	1	1	0	0	0

SQUARE(9) =

Explain your answer here:

The pixels that need to be corrected are the 3, 2, 1, and 1 that now occur within the original shape of the part.

Since these are all "negative-going" errors, use **CLOSE**. This is better than median, which would fix most (but not all) of the "bad" pixels, but would also cut off the corners of the object.

$$\mathbb{J} = \text{CLOSE}[\mathbb{I}, \mathbb{B}] = \text{ERODE}[\text{DILATE}[\mathbb{I}, \mathbb{B}], \mathbb{B}]$$
$$= \text{MIN}[\text{MAX}[\mathbb{I}, \mathbb{B}], \mathbb{B}]$$

0	0	0	0	1	0	0	2	0	0	0	0	1	1	1
0	0	0	0	1	0	0	2	0	0	0	0	1	1	1
0	0	0	0	1	0	0	2	0	0	0	0	1	1	1
0	0	0	0	0	2	0	2	0	2	0	1	0	0	0
0	0	0	0	2	1	45	45	0	0	2	0	0	0	0
1	1	1	0	0	0	46	45	1	0	0	2	3	3	3
1	1	1	1	0	46	45	45	44	45	0	0	2	2	2
1	1	1	0	2	45	45	44	45	46	0	0	3	3	3
2	2	2	2	3	46	45	3	45	45	0	3	2	2	2
2	2	2	2	2	45	44	2	1	46	0	0	4	4	4
1	1	1	2	1	44	45	46	1	45	0	1	0	0	0
0	0	0	2	0	0	1	1	0	2	0	0	1	1	1
0	0	0	1	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1	0	0	0	0	0

$\mathbb{I}_2$ extended by replication

$\mathbb{J}_1 = \mathrm{MAX}[\mathbb{I}_2, \mathbb{B}]$

0	0	1	1	1	2	2	2	0	0	1	1	1
0	0	1	2	2	2	2	2	2	2	1	1	1
0	0	2	2	45	45	45	45	2	2	2	1	1
1	1	2	2	46	46	46	45	2	2	3	3	3
1	1	2	46	46	46	46	45	45	45	3	3	3
1	1	2	46	46	46	46	46	46	46	3	3	3
2	2	3	46	46	46	45	46	46	46	3	3	3
2	2	3	46	46	46	45	46	46	46	4	4	4
2	2	3	46	46	46	46	46	46	46	4	4	4
2	2	2	45	45	46	46	46	46	46	4	4	4
1	2	2	44	45	46	46	46	45	45	1	1	1
0	2	2	2	1	1	1	2	2	2	1	1	1
0	1	1	1	1	0	1	1	1	1	0	0	0

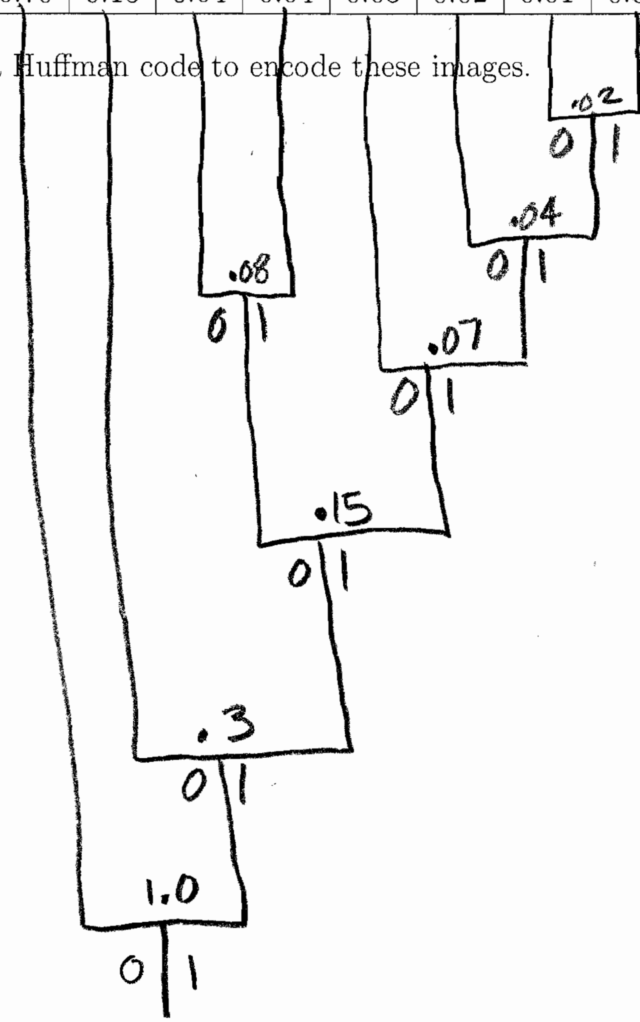
$\mathbb{J} = \mathrm{MIN}[\mathbb{J}_1, \mathbb{B}]$

0	0	1	1	1	2	0	0	0	0	1
0	0	1	2	2	2	2	2	1	1	1
0	0	2	2	45	45	2	2	2	1	1
1	1	2	2	46	45	2	2	2	2	3
1	1	2	46	45	45	45	45	3	3	3
1	1	2	46	45	45	45	46	3	3	3
2	2	3	46	45	45	45	46	3	3	3
2	2	2	45	45	45	45	46	4	4	4
1	2	2	44	45	46	45	45	1	1	1
0	2	1	1	1	1	1	2	1	1	1
0	1	1	0	0	0	1	1	0	0	0

4. **25 pts.** Gray scale digital images $\mathbb{I}$ with 3 bits per pixel and gray levels in the range $\{0, 1, \ldots, 7\}$ are modeled as coming from an information source with the following source symbol probabilities (normalized histogram):

k	0	1	2	3	4	5	6	7
$p_{\mathbb{I}}(k)$	0.70	0.15	0.04	0.04	0.03	0.02	0.01	0.01

(a) **15 pts.** Design a Huffman code to encode these images.



K	0	1	2	3	4	5	6	7
$\hat{K}$	0	10	1100	1101	1110	11110	111110	111111
$L(\hat{K})$	1	2	4	4	4	5	6	6

Problem 4 cont. . .

(b) **5 pts**. Find the expected BPP (bits per pixel) and CR (compression ratio) for the coded images $C(\mathbb{I})$.

Original BPP = 3

$$\text{New BPP} = (.7)(1) + (.15)2 + (.04)4 + (.04)4 + (.03)4 + (.02)5 + (.01)6 + (.01)6$$

$$= \underline{\underline{1.66}}$$

$$CR = \frac{3}{1.66} = \underline{\underline{1.80723:1}}$$

(c) **5 pts**. Does your code reach the theoretical bound on maximum entropy reduction? Explain why or why not.

No... to realize the theoretical lower band all the symbol probabilities must be powers of 2, which is not the case here.

e.g. 0.7 is not an integer power of 2.