ECE 5273
Test 2

Friday, May 8, 2020
10:30 AM - 12:30 PM

Spring 2020 Name: SOLUTION

Dr Havlicek Student Num:________

Directions: This is an open notes test. You may use a clean copy of the course notes as published on the course web site. Other materials are not allowed. You have 120 minutes to complete the test. All work must be your own.

SHOW ALL OF YOUR WORK for maximum partial credit!

GOOD LUCK!

SCORE.

1. (20) ________

2. (20) ________

3. (20) ________

4. (20) ________

5 (20) ________

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

TRUE	FALSE	
	✓	(a) **3 pts**. For large window sizes, the median filter is almost always implemented by pointwise multiplying the DFT's of the appropriately zero padded images. NOT LINEAR!!
✓		(b) **3 pts**. Special zero padding is needed to correctly implement a *zero-phase* 2D LTI filter by pointwise multiplication of DFT's. Notes p. 5.72
✓		(c) **3 pts**. The key innovation of the bilateral filter is that the weights for neighboring pixels are computed using *luminance similarity* in addition to the spatial distance weighting that is normally found in an LTI filter. Notes p. 6.79
✓		(d) **3 pts**. Baseline JPEG achieves image compression by performing lossy quantization of block DCT coefficients followed by lossless DPCM, run-length coding, and entropy coding. Notes p. 7.72
	✓	(e) **3 pts**. The main reason for using normalized cross-correlation in gray scale template matching algorithms is that the normalized cross-correlation is not affected by scaling and rotation. Notes p. 8.42, 8.45
✓		(f) **3 pts**. The Laplacian-of-Gaussian (LoG) and Canny edge detectors are both based on the idea of finding zero crossings in the approximated second derivative of the image. Notes p. 8.67, 8.89
OH MY!		(g) **2 pts**. The acronym "DPCM" was invented in 2020 by the US Centers for Disease Control and Prevention as part of a social distancing campaign; it stands for: "**D**on't **P**ass **C**oronavirus, **M**oron!"

2. **20 pts**. For the window (structuring element) SQUARE(9), give the order statistic (OS) filter weights (coefficients) **A** for

(a) **4 pts.** A median filter:

$$\mathbf{A} = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$$

Notes p. 6.63

(b) **4 pts.** An average filter:

$$\mathbf{A} = [\tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}\ \tfrac{1}{9}]^T$$

Notes p. 6.63

(c) **4 pts.** An OS filter to perform morphological dilation:

$$\mathbf{A} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$$

Same as MAX filter

(d) **4 pts.** An OS filter to perform morphological erosion:

$$\mathbf{A} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

same as min filter

(e) **4 pts.** The trimmed mean OS filter TM_3 with $Q = 3$:

$$\mathbf{A} = [0\ 0\ 0\ \tfrac{1}{3}\ \tfrac{1}{3}\ \tfrac{1}{3}\ 0\ 0\ 0]^T$$

Notes p. 6.71

3. **20 pts.** The noisy 5×5 image $\mathbf{I}$ shown below has 4-bit pixels in the range $0 \leq I(m,n) \leq 15$:

$$\mathbf{I} = \begin{array}{|c|c|c|c|c|}\hline 4 & 8 & 12 & 15 & 13 \\ \hline 11 & 15 & 4 & 9 & 4 \\ \hline 10 & 5 & \mathbf{8} & 2 & 13 \\ \hline 3 & 9 & 11 & 2 & 4 \\ \hline 2 & 4 & 14 & 4 & 15 \\ \hline \end{array}$$

Denoising is to be performed by applying a median filter with the window $\mathbf{B}$ = CROSS(5) shown here:

$$\mathbf{B} = \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

Use Delman's fast bit-serial median filtering algorithm to compute the filtered output value for the middle pixel (8) shown in large boldface type above.

Note: you do **not** have to apply the filter at any other pixels! Just apply it at the center pixel *only* to find the filtered output value for that pixel *only*. More workspace is provided on the following page in case you need it.

The windowed set is $B \circ I\,(2,2) = \{4, 5, 8, 2, 11\}$

Notes pp. 6.56–6.58:

Original Data

$$\begin{matrix} \boxed{\begin{matrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix}} & \begin{matrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{matrix} \end{matrix} \to 0$$

First Change

$$\begin{matrix} \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix} & \boxed{\begin{matrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} \end{matrix} \to 1$$

Second Change

$$\begin{matrix} \begin{matrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{matrix} & \boxed{\begin{matrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix}} & \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{matrix} \end{matrix} \to 0$$

Third Change

$$\begin{matrix} \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix} & \boxed{\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{matrix}} \end{matrix} \to 1$$

$\boxed{\text{MEDIAN} = 0101 = 5}$ ✓

4. **20 pts**. The 256 × 256 image **I** shown at left below has 8-bit pixels in the range $0 \leq I(m,n) \leq 255$. This image is transmitted through a communication channel where it is corrupted by IID additive white Gaussian noise. The received image **J** is at right below.

I

J

The following two filters are applied to perform denoising:

- A 256 × 256 Gaussian low-pass filter with space constant $\sigma = 1.25$, implemented exactly as described on pages 5.64-5.65 of the course notes,
- A bilateral filter with Gaussian weighting functions having $\sigma_G = 1.0$ and $\sigma_H = 0.67$.

Identify and label the two filtered images shown below:

Bilateral Filter
(preserves edges)

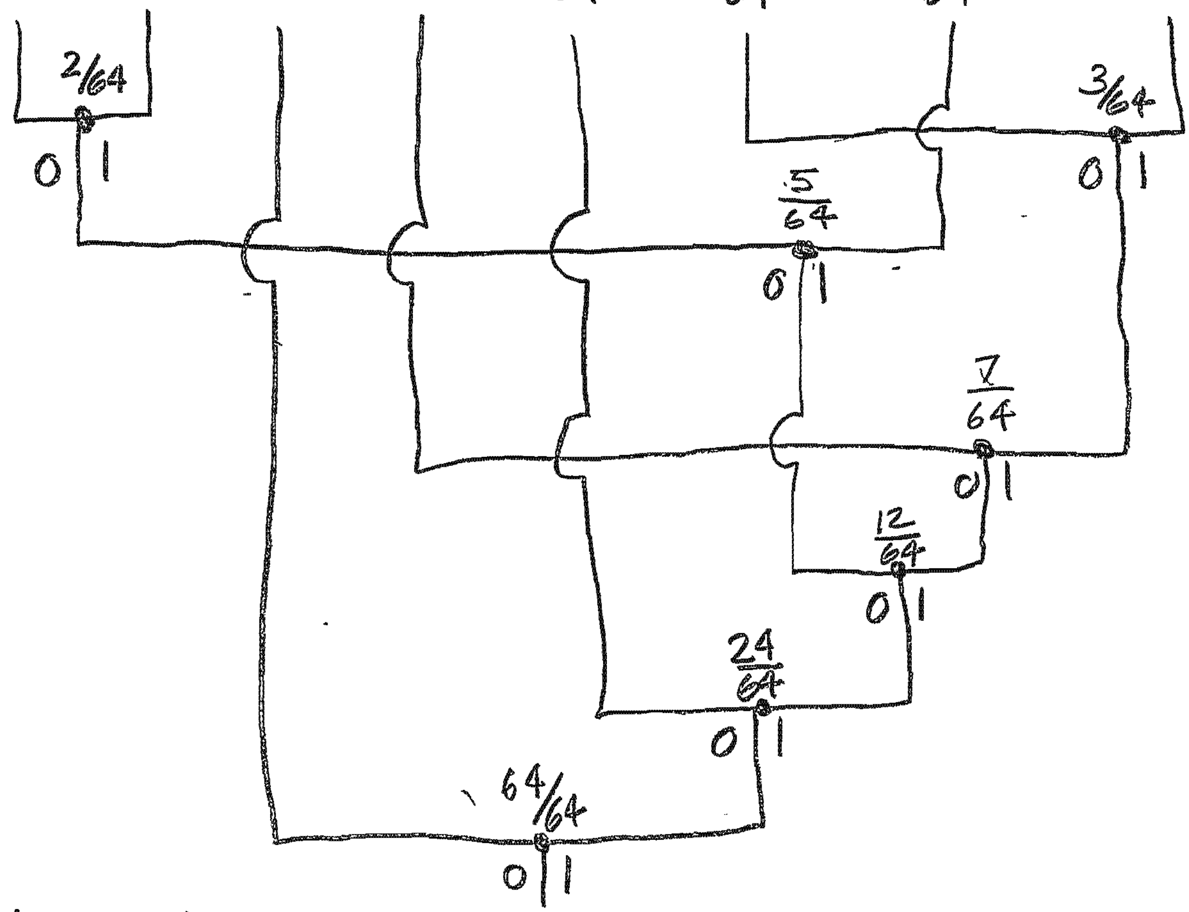
Gaussian Filter
(blurs edges)

5. **20 pts.** Gray scale digital images **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
$p_{\mathbf{I}}(k)$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{5}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{32}$	$\frac{3}{64}$	$\frac{1}{64}$

(a) **12 pts.** Design a Huffman code to encode these images.

K	0	1	2	3	4	5	6	7
$P_I(k)$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{5}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{32}$	$\frac{3}{64}$	$\frac{1}{64}$
	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{40}{64}$	$\frac{4}{64}$	$\frac{12}{64}$	$\frac{2}{64}$	$\frac{3}{64}$	$\frac{1}{64}$



K	0	1	2	3	4	5	6	7
$C(k)$	11000	11001	0	1110	10	11110	1101	11111
$L[C(k)]$	5	5	1	4	2	5	4	5

Problem 5 cont...

(b) **4 pts.** Find the expected BPP (bits per pixel) and CR (compression ratio) for the coded images $C(\mathbf{I})$.

$$BPP(I) = 3$$

$$BPP(\hat{I}) = \underbrace{5\left(\tfrac{1}{64}\right)}_{0} + \underbrace{5\left(\tfrac{1}{64}\right)}_{1} + \underbrace{1\left(\tfrac{40}{64}\right)}_{2} + \underbrace{4\left(\tfrac{4}{64}\right)}_{3} + \underbrace{2\left(\tfrac{12}{64}\right)}_{4}$$
$$+ \underbrace{5\left(\tfrac{2}{64}\right)}_{5} + \underbrace{4\left(\tfrac{3}{64}\right)}_{6} + \underbrace{5\left(\tfrac{1}{64}\right)}_{7}$$

$$= \frac{5+5+40+16+24+10+12+5}{64}$$

$$= \frac{117}{64} = \underline{\underline{1.82813}}$$

$$CR = \frac{BPP(I)}{BPP(\hat{I})} = \frac{3 \cdot 64}{117} = \underline{\underline{1.64103}}$$

(c) **4 pts.** Does your code achieve the theoretical lower bound on BPP for the coded images? Explain why or why not.

<u>NO</u> - Some of the symbol probabilities are <u>not</u> integer powers of 2... like 3/64. //