# ECE 5273
# Test 2

Friday, May 12, 2017
10:30 AM - 12:30 PM

Spring 2017

Dr. Havlicek

Name: SOLUTION

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. (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 | |
|---|---|---|
| ✓ | | (a) **2 pts**. For a linear denoising filter, the window size is usually chosen to balance the tradeoff between noise smoothing and image smoothing. NOTES 5-109 |
| ✓ | | (b) **2 pts**. Laplacian noise is characterized by heavy tails – meaning a higher probability of very large or very small noise samples (outliers) compared to Gaussian noise. NOTES 6-9 |
| | ✓ | (c) **2 pts**. Salt-and-pepper noise is the worst type of additive noise. X NOTES 6-11 |
| | ✓ | (d) **2 pts**. The median filter has a low-pass frequency response. MF is a nonlinear filter → no freq resp |
| | ✓ | (e) **2 pts**. The grayscale morphological CLOSE filter smoothes noise, preserves edges, and eradicates positive-going impulses. X negative NOTES 6-34 |
| ✓ | | (f) **2 pts**. Order statistic filters like the trimmed mean have been shown to be remarkably robust for image denoising. NOTES 6-70 6-72 |
| ✓ | | (g) **2 pts**. The bilateral filter is based on a product of two kernels (weighting functions): one that weights neighbors based on spatial distance and one that weights neighbors based on luminance similarity. NOTES 6-79 |
| | ✓ | (h) **2 pts**. Differential pulse code modulation (DPCM) is used in JPEG for lossy entropy reduction. X lossless NOTES 7-20 |
| | ✓ | (i) **2 pts**. Anisotropic diffusion is especially useful for removing salt-and-pepper noise. NOTES 8-96 |
| ✓ | | (j) **2 pts**. The main idea of the DCT coefficient zig-zag ordering in baseline JPEG is to create long runs of zeros. NOTES 7-77 7-78 |
| ✓ | | (k) **2 pts**. Noise is a huge problem for the gradient-based edge detectors. NOTES 8-58 |
| OH | MY! | (l) **3 pts**. JPEG is an acronym for *Jalapeño Pizza - Extremely Good!* |

2. **25 pts.** The $4 \times 4$ image $\mathbf{I}$ shown on the left below has 8-bit pixels in the range $0 \le I(m,n) \le 255$. This image is transmitted through a communication channel where it is corrupted by IID additive white Laplacian noise with parameter $\sigma = 4$. The received image $\mathbf{J}$ is shown on the right below.

$\mathbf{I} =$

| | | | |
|---|---|---|---|
| 32 | 32 | 32 | 32 |
| 120 | 120 | 32 | 32 |
| 120 | 120 | 120 | 32 |
| 120 | 120 | 120 | 120 |

$\mathbf{J} =$

| | | | | | |
|---|---|---|---|---|---|
| 30 | 30 | 32 | 28 | 35 | 35 |
| 30 | 30 | 32 | 28 | 35 | 35 |
| 125 | 125 | 134 | 39 | 36 | 36 |
| 117 | 117 | 115 | 107 | 27 | 27 |
| 123 | 123 | 120 | 122 | 119 | 119 |
| 123 | 123 | 120 | 122 | 119 | 119 |

(a) **17 pts.** Apply the trimmed mean OS filter $\mathrm{TM}_Q(\mathbf{J}, \mathbf{B})$ with $Q = 2$ and window $\mathbf{B} = \mathrm{SQUARE}(9)$ shown below to denoise the received image. Handle edge effects by replication. Show the denoised image $\mathbf{K}$ in the space provided at the bottom of this page.

$\mathbf{B} =$

| | | |
|---|---|---|
| | | |
| | | |
| | | |

**Note:** Work space is provided on the following page.

(b) **8 pts.** Compute the ISNR for the denoised image.

ISNR is defined on p. 5-115 of the notes

$$\mathrm{ISNR}(K) = 10 \log_{10} \frac{\mathrm{MSE}(J)}{\mathrm{MSE}(K)} = 10 \log_{10} \frac{34.8125}{257.000} = -8.6820 \text{ dB}$$

Show the denoised image here:

$\mathbf{K} =$

| | | | |
|---|---|---|---|
| 50 | 33 | 34 | 35 |
| 101 | 82 | 50 | 34 |
| 122 | 119 | 100 | 67 |
| 121 | 120 | 119 | 117 |

ISNR = $-8.6820$ dB

NOTE: The filter made things worse in this case. That is because, with such a small image, the result is dominated by the edge effects.

$2P+1 = 9 \rightarrow P = 4$

$Q = 2 \;;\; Q = P/2 \checkmark$

$A = [0 \; 0 \; \frac{1}{5} \; \frac{1}{5} \; \frac{1}{5} \; \frac{1}{5} \; \frac{1}{5} \; 0 \; 0]^T$

$\Rightarrow$ The action of the filter is to delete the two smallest samples and the two largest samples from the window set. The remaining five samples are averaged.

| FOR | AVERAGE |
|---|---|
| $J(0,0)$: | $\{\cancel{30}, \cancel{30}, 32, 30, 30, 32, 125, \cancel{125}, \cancel{134}\}$ |
| $J(0,1)$: | $\{30, 32, \cancel{28}, 30, 32, \cancel{28}, \cancel{125}, \cancel{134}, 39\}$ |
| $J(0,2)$: | $\{32, \cancel{28}, 35, 32, \cancel{28}, 35, \cancel{134}, \cancel{39}, 36\}$ |
| $J(0,3)$: | $\{\cancel{28}, 35, 35, \cancel{28}, 35, 35, \cancel{39}, \cancel{36}, 36\}$ |
| $J(1,0)$: | $\{\cancel{30}, \cancel{30}, 32, 125, \cancel{125}, \cancel{134}, 117, 117, 115\}$ |
| $J(1,1)$: | $\{\cancel{30}, 32, \cancel{28}, \cancel{125}, \cancel{134}, 39, 117, 115, 107\}$ |
| $J(1,2)$: | $\{32, \cancel{28}, 35, \cancel{134}, 39, 36, \cancel{115}, 107, \cancel{27}\}$ |
| $J(1,3)$: | $\{28, 35, 35, \cancel{39}, 36, 36, \cancel{107}, \cancel{27}, \cancel{27}\}$ |
| $J(2,0)$: | $\{125, \cancel{125}, \cancel{134}, 117, \cancel{117}, \cancel{115}, 123, 123, 120\}$ |
| $J(2,1)$: | $\{\cancel{125}, \cancel{134}, \cancel{39}, 117, 115, \cancel{107}, 123, 120, 122\}$ |
| $J(2,2)$: | $\{\cancel{134}, 39, \cancel{36}, 115, 107, \cancel{27}, 120, \cancel{122}, 119\}$ |
| $J(2,3)$: | $\{39, 36, 36, 107, \cancel{27}, \cancel{27}, \cancel{122}, \cancel{119}, 119\}$ |
| $J(3,0)$: | $\{117, \cancel{117}, \cancel{115}, \cancel{123}, \cancel{123}, 120, 123, 123, 120\}$ |
| $J(3,1)$: | $\{117, \cancel{115}, \cancel{107}, \cancel{123}, 120, 122, \cancel{123}, 120, 122\}$ |
| $J(3,2)$: | $\{115, \cancel{107}, \cancel{27}, 120, \cancel{122}, 119, 120, \cancel{122}, 119\}$ |
| $J(3,3)$: | $\{107, \cancel{27}, \cancel{27}, \cancel{122}, 119, 119, \cancel{122}, 119, 119\}$ |

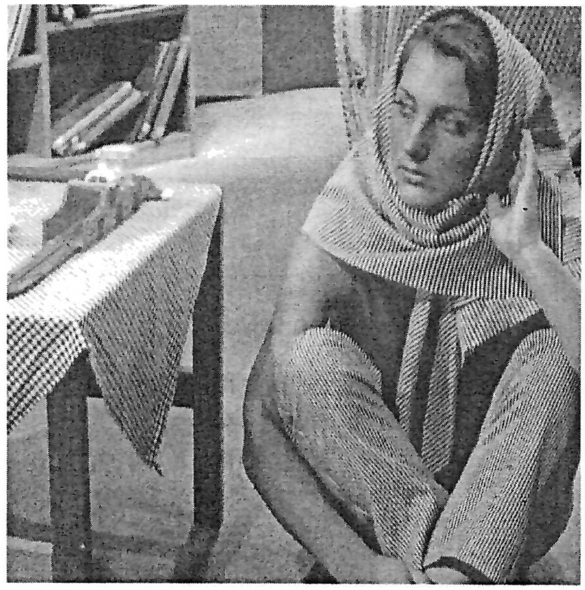
$$J - I = \begin{array}{|c|c|c|c|} \hline -2 & 0 & -4 & 3 \\ \hline 5 & 14 & 7 & 4 \\ \hline -3 & -5 & -13 & -5 \\ \hline 3 & 0 & 2 & -1 \\ \hline \end{array} \qquad (J-I)^2 = \begin{array}{|c|c|c|c|} \hline 4 & 0 & 16 & 9 \\ \hline 25 & 196 & 49 & 16 \\ \hline 9 & 25 & 169 & 25 \\ \hline 9 & 0 & 4 & 1 \\ \hline \end{array}$$

$$MSE(J) = \frac{1}{16} \sum (J-I)^2 = \frac{1}{16} \cdot 557 \approx 34.8125$$

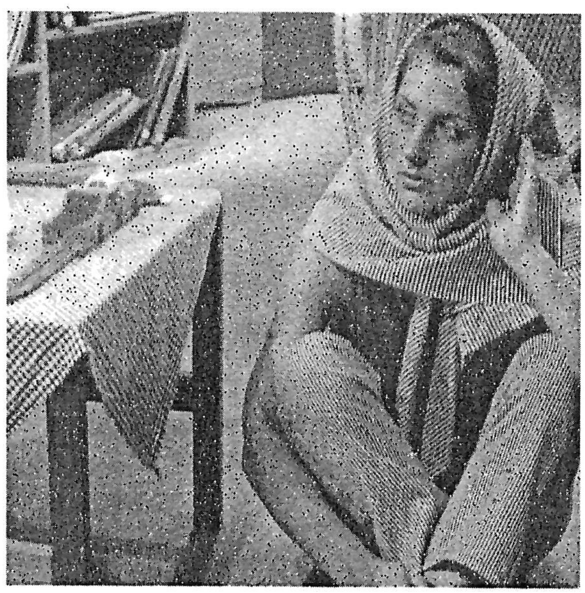
$$K - I = \begin{array}{|c|c|c|c|} \hline 18 & 1 & 2 & 3 \\ \hline -19 & -38 & 18 & 2 \\ \hline 2 & -1 & -20 & 35 \\ \hline 1 & 0 & -1 & -3 \\ \hline \end{array} \qquad (K-I)^2 = \begin{array}{|c|c|c|c|} \hline 324 & 1 & 4 & 9 \\ \hline 361 & 1{,}444 & 324 & 4 \\ \hline 4 & 1 & 400 & 1{,}225 \\ \hline 1 & 0 & 1 & 9 \\ \hline \end{array}$$

$$MSE(K) = \frac{1}{16} \sum (K-I)^2 = \frac{1}{16} \cdot 4{,}112 = 257.0000$$

3. **25 pts**. Consider the original $512 \times 512$ grayscale *Barbara* image $\mathbf{I}$ shown below:



Each pixel has 8 bits. During transmission, the image is corrupted by salt-and-pepper noise and additive white Gaussian noise with standard deviation $\sigma = 12$. The received image $\mathbf{J} = \mathbf{I} + \mathbf{N}$ is shown below:



**Problem 3, cont...**

The following seven filters are applied to $\mathbf{J}$ in an effort to restore the image by denoising:

(a) $\mathbf{K} = \text{AVE}[\mathbf{J}, \mathbf{B}]$, with $\mathbf{B} = \text{SQUARE}(25)$.

(b) $\mathbf{K} = \text{MED}[\mathbf{J}, \mathbf{B}]$, with $\mathbf{B} = \text{SQUARE}(25)$.

(c) $\mathbf{K} = \text{CLOSE}[\mathbf{J}, \mathbf{B}]$, with $\mathbf{B} = \text{SQUARE}(25)$.

(d) $\mathbf{K} = \text{OPEN}[\mathbf{J}, \mathbf{B}]$, with $\mathbf{B} = \text{SQUARE}(25)$.

(e) $\mathbf{K} = \text{CLOSE-OPEN}[\mathbf{J}, \mathbf{B}]$, with $\mathbf{B} = \text{SQUARE}(25)$.

(f) A $128 \times 128$ linear Gaussian low-pass filter with space constant $\sigma = 3$, similar to Example 3 on page 5.61 of the course notes.

(g) A $256 \times 256$ linear Gaussian high-pass filter with space constant $\sigma = 48$, similar to Example 6 on page 5.79 of the course notes.

Filters (a)-(e) are implemented directly in the space domain with edge effects handled by replication. Filters (f) and (g) are implemented by pointwise multiplication of DFT's with appropriate zero padding to obtain linear convolution.

Label the seven filtered images shown below on pages 7-9.

CLOSE-OPEN

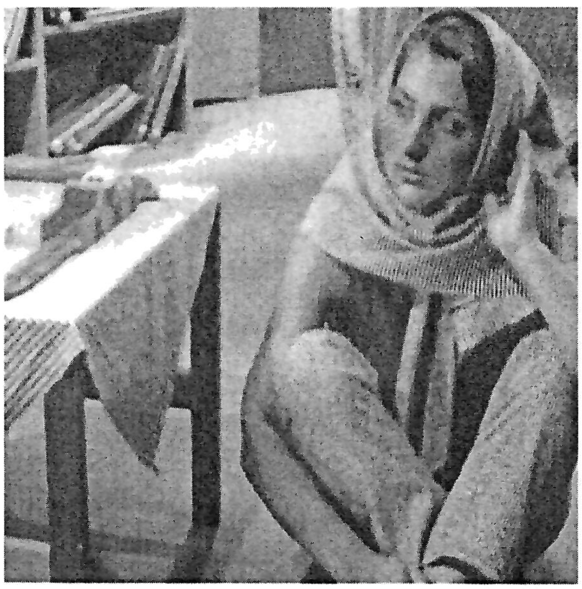
- There is a lot of smoothing, but the edges are sharp.
  → can't be average or Gaussian low-pass

- The image is grossly oversmoothed.
  - results from applying four 5x5 filtering operations; two dilations and two erosions.
    → MUST BE CLOSE-OPEN

Problem 3, cont...

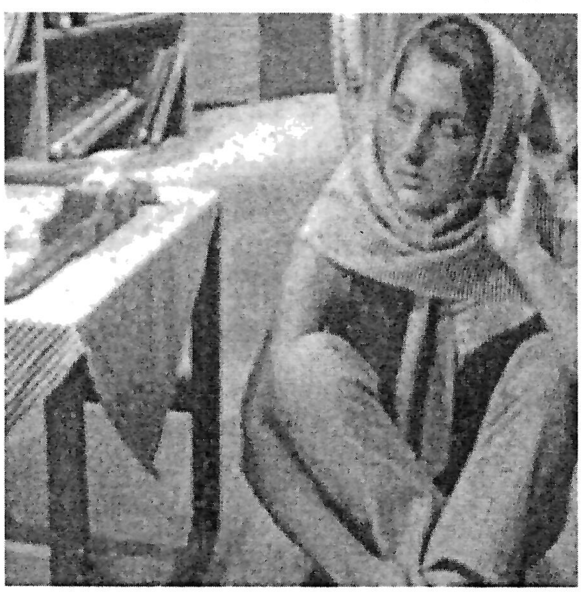
GAUSSIAN LOW-PASS

- clearly a linear low-pass filter, so must be AVE or Gaussian low-pass.
- But the dark bands around the outside edges of the image result from zero padding.
  → Edge effects not handled by replication.
  → Cant be AVE.



MEDIAN

- removes both Gaussian and salt-and-pepper noise
- but edges not blurred
  → Median



AVE

- There is blurring, but it is very slight.
- Gaussian noise removed effectively.
- but some artifacts from salt-and-pepper noise remain.

Problem 3, cont…

OPEN

- Because erosion is done first, "salt" is removed. But artifacts from "pepper" remain.

GAUSSIAN HIGH-PASS

- obviously a hi-pass filter
- noise is amplified instead of attenuated.

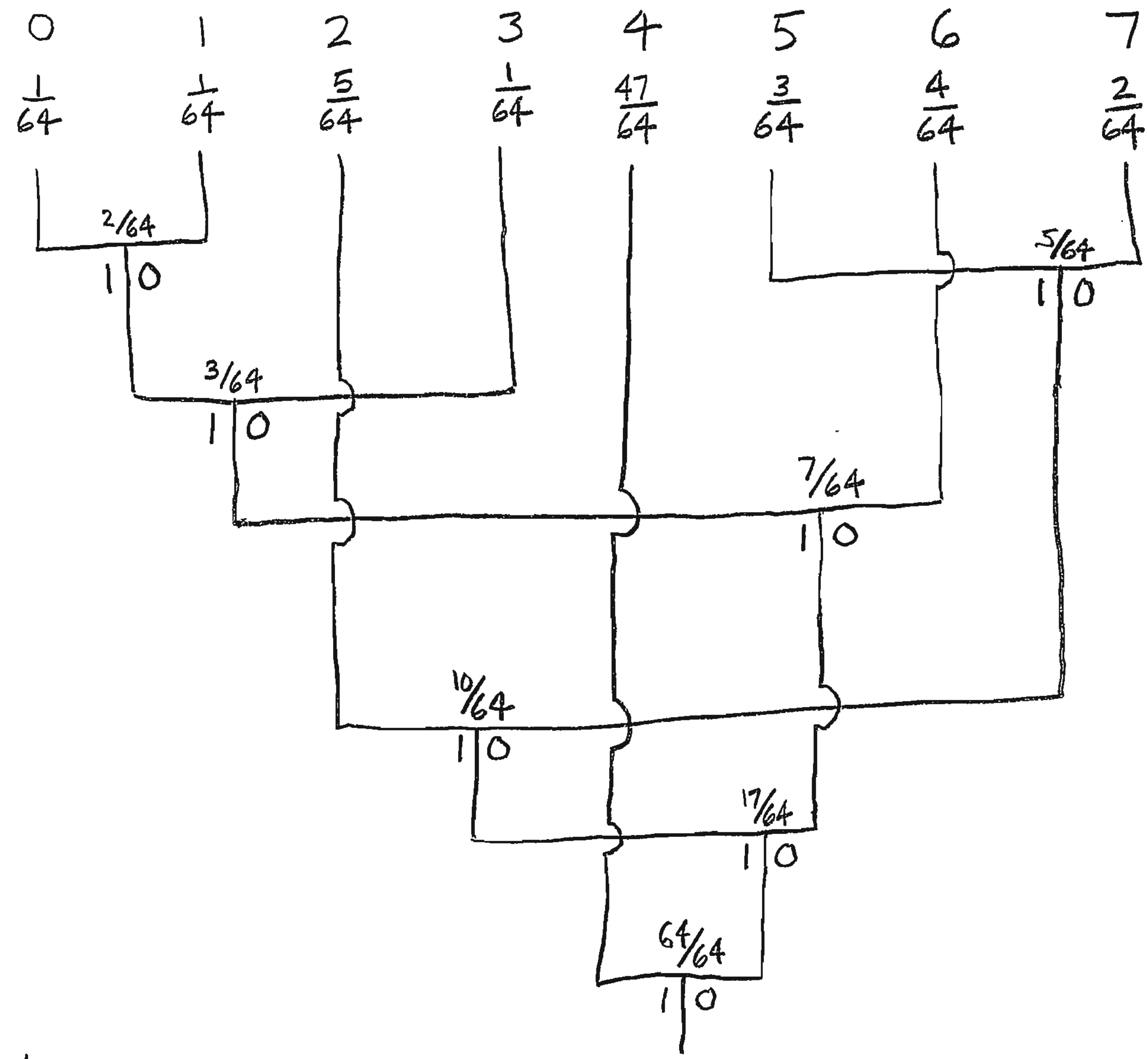
CLOSE

- Dual of the "OPEN".
- Because dilate is done first, "pepper" is removed. But artifacts from "salt" remain.

4. **25 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 |
|---|---|---|---|---|---|---|---|---|
| $p_{\mathbf{I}}(k)$ | $\frac{1}{64}$ | $\frac{1}{64}$ | $\frac{5}{64}$ | $\frac{1}{64}$ | $\frac{47}{64}$ | $\frac{3}{64}$ | $\frac{4}{64}$ | $\frac{2}{64}$ |

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




| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| c(k) | 00111 | 00110 | 011 | 0010 | 1 | 0101 | 000 | 0100 |
| L[c(k)] | 5 | 5 | 3 | 4 | 1 | 4 | 3 | 4 |

Problem 4 cont...

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

Original BPP = 3

$$\text{Coded BPP} = \sum_{k=0}^{7} p_I(k)\, L[c(k)]$$

$$= \underbrace{\tfrac{1}{64}\cdot 5}_{k=0} + \underbrace{\tfrac{1}{64}\cdot 5}_{k=1} + \underbrace{\tfrac{5}{64}\cdot 3}_{k=2} + \underbrace{\tfrac{1}{64}\cdot 4}_{k=3} + \underbrace{\tfrac{47}{64}\cdot 1}_{k=4} + \underbrace{\tfrac{3}{64}\cdot 4}_{k=5}$$

$$+ \underbrace{\tfrac{4}{64}\cdot 3}_{k=6} + \underbrace{\tfrac{2}{64}\cdot 4}_{k=7}$$

$$= \tfrac{1}{64}[5+5+15+4+47+12+12+8] = \tfrac{1}{64}\cdot 108 \approx 1.6875$$

$$CR = \frac{3}{1.6875} = \underline{\underline{1.7778}}$$

(c) **5 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 like $p_I(4) = 47/64$ are <u>not</u> integer powers of 2.