Final Exam
12/22/2009, 1PM-4PM

Your Name: _______________________________________________
ID Number: _______________________________________________

Closed book. One sheet of notes (double sided) permitted. Write your solutions in the space provided next to the questions (use both sides if necessary). If you need extra space, you are probably on the wrong track!

<table>
<thead>
<tr>
<th>PROBLEM NO.</th>
<th>FULL POINTS</th>
<th>YOUR POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8 (bonus)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total Points</td>
<td>110</td>
<td></td>
</tr>
</tbody>
</table>
1. (15pt) (a) Find the opening of the binary image, $F$, in Fig. (a) by the structuring
element $H$ in Fig. (b). You can use the grids in Fig. (c) and Fig. (d) to draw the
intermediate and the final results.
(b) Find the gray scale closing of the function sketched in Fig. (e) with the structural
element in Fig. (f). Sketch the intermediate and final result in Figs. (g) and (h).
Comment on the effect of closing and grayscale opening.
2. (15pt) Consider a 2D image \( f(x,y) = \sin(4\pi x - 6\pi y) \).
   
a) (3pt) Determine its Fourier transform \( F(u,v) \) and illustrate the spectrum (i.e., the impulses in the transform) in Fig. (a). Indicate the magnitude of each impulse.
   
b) (3pt) Suppose this signal is sampled uniformly with sampling intervals \( \Delta x = \Delta y = \Delta = 1/5 \). Draw the spectrum of the sampled signal in Fig. (b).
   
c) (4 pt) Suppose the sampled signal is interpolated by an ideal low-pass filter \( h_1(x,y) \) with frequency response
   \[
   H_1(u, v) = \begin{cases} 
   \Delta^2 & \text{if } -f_s/2 < u, v < f_s/2, \\
   0 & \text{otherwise}
   \end{cases}
   \]
   where \( f_s = 1/\Delta \). Draw the spectrum of the reconstructed signal in Fig. (c). Give the spatial representation of the reconstructed signal \( f_{r1}(x,y) \).
   
d) (5pt) Suppose that the sample-and-hold filter (also called nearest neighbor) is used instead. Its filter response can be written as
   \[
   h_2(x, y) = h(x)h(y),
   \]
   where
   \[
   h(x) = \begin{cases} 
   1 & \text{if } |x| < \Delta/2 \\
   0 & \text{otherwise}
   \end{cases}
   \]
   Give the filter response in the frequency domain, \( H_2(u,v) \). If the filter is further band-limited to \(-7 \leq u,v \leq 7\), give the frequency and spatial representation of the reconstructed signal \( f_{r2}(x,y) \).
Fig. (a): Original Signal

Fig. (b): Sampled Signal

Fig. (c): Reconstructed Signal
3. (15 pt) For the following bi-level image, consider the following compression method.
For each new pixel, you predict its value (black or white) using the following method: if its top and left pixels are both white, you predict it to be white. Otherwise, you predict it to be black. You assume the first line and first column are all white and start your processing only in the second line and second column. You create a prediction error image with 0 representing correct prediction, WB representing the event that white is predicted to be black, and BW representing the event that black is predicted to be white. You losslessly code the prediction error image using runlength+huffman coding (as described below).
(a) Generate the predicted image and the prediction error image; you can use the grid provided below.
(b) Create the runlength representation for each line, which starts with the runlength of “0” followed by the non-zero value (WB or BW). The last runlength in a line is indicated by EOL.
(c) Generate a Huffman code for all symbols, including all runlength values and WB and BW.
(d) compute the average bit rate (bits/pixel) using your Huffman code.

4. (15 pt) Answer the following questions
(a) What is the purpose of transform in a transform-coder?
(b) What is spatial scalability? What is quality scalability? How can these features be used in practical applications?
(c) How does the JPEG2000 standard achieve spatial scalability? How does it achieve quality scalability?

5. (10 pt) Suppose you took a picture while the camera is moving with a speed of \((v_x, v_y)\). The camera’s exposure time is \(T\).
(a) How is the captured image \(g(x,y)\) related to the original scene \(f(x,y)\)? If you were to write \(g(x,y) = f(x,y) * h(x,y)\), what is \(h(x,y)\)?
(b) You would like to use the Wiener filter to deblur the image. Suppose you found the Fourier transform of \(h(x,y)\) is \(H(u,v)\). Furthermore, you assume that \(S_n(u,v) / S_f(u,v)\) can be approximated by a constant \(K\). What is the Wiener filter in the Frequency domain? Describe one way to implement the filtering operation.
6. (15 pt) Suppose you captured two images f(u,v) and g(x,y), at two different times, when the underlying scene has some motion. You would like to map image f(u,v) so that it aligns with the image g(x,y) as best as possible. Towards this goal, you found N (N>4) corresponding points between these two images, with (uk,vk) corresponding to (xk,yk), k=1,2,…,N.

(a) Suppose you want to approximate the mapping function between the two images by a bilinear mapping. What is the mapping function you would use to generate the mapped image q(x,y) from f(u,v)? Determine the coefficients of the bilinear mapping from the correspondence between (uk,vk) and (xk,yk).

(b) Write a MATLAB script to create q(x,y) from f(u,v) using the mapping function you determined in step (a). You can make use of the MATLAB function interp2( ) for this purpose.

Hint: syntax of interp2( )
ZI= interp2 (X,Y,Z,XI, YI,)
Z is known at points specified in X, Y
The function interpolates its values in XI, YI from the known values in X,Y, store the result in ZI

7. (15pt) Write a MATLAB script for implementing a simplified transform coder. (pseudo JPEG!) Your program should do the following for each image block of 8x8:
1) shift the image mean value to 0 by subtracting all pixel values by 128;
2) perform 8x8 DCT;
3) quantize the DCT coefficients using a given quantization matrix Qmatrix, scaled by a given factor QP; For simplicity, assume all the transform coefficients have a dynamic range represented by (-M,M), where M is a known value;
4) count the number of non-zero coefficients after quantization, return this number;
5) perform inverse DCT on the quantized coefficients;
6) shift the mean value back by adding 128 to all reconstructed pixel values;
7) perform necessary truncation so that the reconstructed pixel values can be represented by 8 bit unsigned integers, and save the results in the output image;
8) determine the MSE of the reconstructed block and return this value;

Finally, you should sum the numbers of non-zero coefficients for all blocks, and determine the ratio defined as total_number_nonzero_coefficients/total_number_pixels as an approximate compression ratio. Also determine the average MSE of all blocks, and convert it to the average PSNR value. Your program should report the estimated compression ratio and the PSNR, as well as the decompressed image.

For simplicity, assuming the image has only the luminance component represented in 8 bit unsigned integer and it has been read into an unsigned character array inimg[]. Your matlab function should have the following syntax:

[outimg, compression_ratio, PSNR]=DCT_coding(inimg,QP,Qmatrix)

Note that you can use built-in functions in MATLAB for computing 2D DCT and inverse DCT. Also, you are encouraged to use the “blkproc” function of MATLAB to speed up the processing. But this is not required.