Final Exam (5/16/2016 10AM-12:30PM)
Closed book, 1 sheet of notes (double sided) allowed.
No peeking into neighbors or unauthorized notes. No calculator or any electronics devices allowed.
Cheating will result in getting an F on the course.
Make sure you write your name and ID on the cover and blue book. Write your answer in the blue book (or on the problem sheet when space are provided on the problem sheet).

1. (15 pt) Consider a simplified JPEG like image coder, where each 4x4 block is transformed into DCT coefficients and these coefficients are then quantized and coded into binary bits using run-length coding along the zig-zag order. The figure below shows a 4x4 block of quantized DCT coefficient indices. To code these indices, we scan them using the zig-zag order. Along this scan order, we code each non-zero index with a fixed length coder, and the zero indices using the runlength of zeros. The last run of zeros will be indicated by a EOB (end of block) symbol. Let us assume that there are at most 4 different possible values for the magnitude of the non-zero indices (from 1 to 4) so that each can be coded using 2 bits. Each non-zero index is coded with 3 bits, with 1 bit for the sign (0 for positive, 1 for negative) and 2 bits for the magnitude. For the runlength, for simplicity, let us assume that the possible runlength symbols are 0, 1, 2, 3, EOB.
   a. Write down the sequence of (non-zero, run-length) pairs that represents this block;
   b. Table 1 provides the probability distribution of possible runlength symbols to be coded using Huffman coding. Design a Huffman code for all possible symbols;
   c. Write down the actual coded bitstream for this block using a combination of your Huffman code and fixed length coding as described. How many bits did you use in total? What is the bit rate (bits/pixel)?

4	0	2	-3
0	0	0	0
-1	0	0	0
0	1	0	0

Quantized transformed coefficient indices

symbol	Prob.
0	0.4
1	0.15
2	0.1
3	0.05
EOB	0.3

2. (15 pt) The following two vectors form an orthonormal basis set for 1D vector of dimension 2.

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

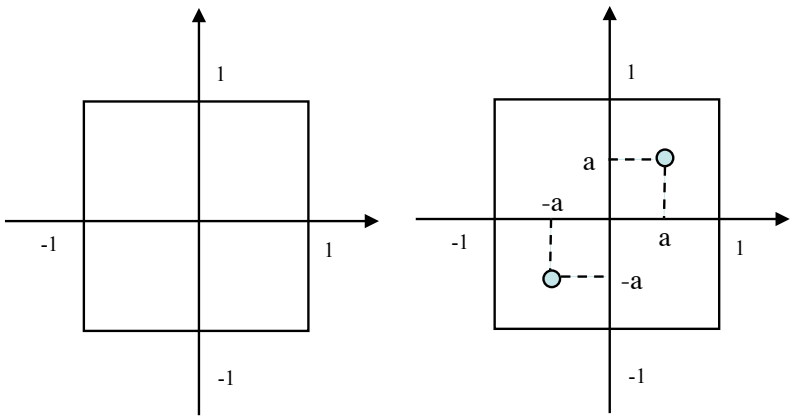
   a. (4pt) Use these 1D bases to construct a 2D orthonormal transform basis set. Write down the 4 basis images.
   b. (4pt) Apply your 2D transform to the following 2x2 image, and find all the transform coefficients.

$$F = \begin{bmatrix} 8 & 8 \\ 4 & 4 \end{bmatrix}$$

   c. (4pt) Quantize the transform coefficients with a uniform quantizer with a stepsize of q=3. For the DC coefficient, you should assume the minimal possible value is 0. That is, values in [0,q) is quantized to q/2. But for AC coefficients, you should assume the quantizer is symmetric, i.e. values in [-q/2, q/2) is quantized to 0. Write down the quantized coefficients.
   d. (3pt) Determine the reconstructed image from the quantized coefficients.

3. (15 pt) Consider coding a sequence of 2-D vectors that is uniformly distributed over the region illustrated in Fig. (a). Suppose you want to design a codebook with 2 codewords. A possible codebook construction is illustrated in Figure (b), where the circles represent the locations of two codewords.
   a. (5 pt) Sketch the partition between the two regions represented by the two codewords that will minimize the MSE. Explain briefly why.
   b. (5pt) Determine the value of a that will minimize the mean square error of the quantizer, together with the partition you gave in part (a).
   c. (5pt) Determine the corresponding minimal mean square error.

   Note: it is OK to just write the integral formula for the above questions.



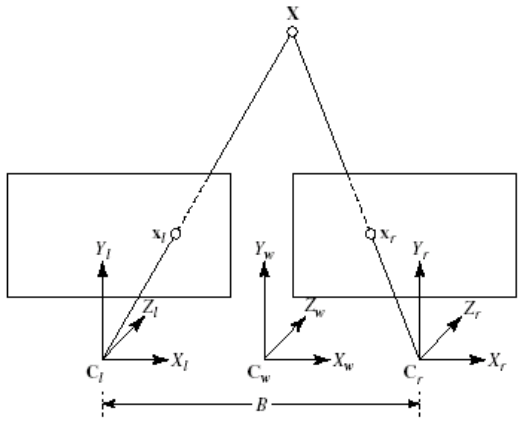
(a) (b)

4. (15 pt) Consider a video coder using bidirectional temporal prediction. A pixel in frame n is predicted from both frame n-1 and frame n+1, using a predictor of the form $f_n(x,y)=af_{n-1}(x',y')+bf_{n+1}(x'',y'')$. Here we assume that through motion estimation, we have identified that pixel (x,y) in frame n corresponds to pixel (x',y') in frame n-1 and pixel (x'',y'') in frame n-2. (In reality, the prediction should be based on decoded values. But for ease of analysis, let us assume the prediction is based on the original values). Assume all the pixels have the same variance σ^2, the corresponding pixels in two adjacent frames (frame n and frame n-1) have a correlation coefficient of ρ and the corresponding pixels in two frames that are two-frame apart (frame n and frame n-2) have a correlation coefficient of ρ^2.
   a) (10pt) Determine the predictor coefficient a,b that will minimize the prediction error variance. Also determine the prediction error variance with this predictor.
   b) (3pt) Assume that the quantization error for coding a variable with variance σ^2 using R bits is $D(R) = \sigma^2 2^{-2R}$. What is the quantization error for coding $f_n(x,y)$ using bidirectional predictive coding with R bits? What is the quantization error for coding $f_n(x,y)$ directly?
   c) (2pt) To reach the same quantization error of D_0, what is the bit rate savings using the predictive coder?

5. (15pt) Consider how to use the Laplacian pyramid representation to create a two-layer image coder. Suppose the original image size is MxN. The base layer image should have a size of M/2 x N/2. The enhancement layer image should have a size of MxN. Assume the image at each layer is coded using a JPEG coder.
   a. (10pt) Show a block diagram for the encoder. You could use a single block to indicate a JPEG encoder, which takes an input image I, and outputs a bit stream for this image, and a decoded image (with quantization error) $\hat{I}$. Your block diagram can also use a block to indicate down-sampling by a factor of 2x2 and a block to indicate up-sampling by a factor of 2x2. The overall encoder should have an original image I as input, and two bits streams as output, one for the base layer, another for the enhancement layer.
   b. (5pt) Show a block diagram for the decoder. You could use a single block to indicate a JPEG decoder, which takes an input bitstream, and output a decoded image $\hat{I}$. The overall decoder should have two bits streams as input (one for the base layer, another for the enhancement layer), and the final reconstructed image as the output.

   Note: If you have trouble drawing the block diagram, you could also write a pseudo code showing the steps of the encoder and decoder processing.

6. (10pt) Consider a stereo imaging system using two parallel cameras with a baseline distance of B, each camera with a focal length of F. The relationships between the image coordinates and 3D world coordinates are shown below.
(a) (5pt) For each pixel (x_l, y) in the left image I_l, how do you estimate its 3D coordinate?
(b) (5pt) Now assume each camera takes a video. Suppose that you have a stereo pair captured at time t, $I_{l,t}$ and $I_{r,t}$, and a stereo pair captured at time t-1, $I_{l,t-1}$ and $I_{r,t-1}$. For each pixel $(x_{l,t}, y_t)$ in the image $I_{l,t}$, how do you estimate its 3D movement between time t and t-1?



(a)

$$X_l = X + \frac{B}{2}, \quad X_r = X - \frac{B}{2}, \quad Y_l = Y_r = Y, \quad Z_l = Z_r = Z;$$

$$x_l = F\frac{X + B/2}{Z}, \quad x_r = F\frac{X - B/2}{Z}, \quad y_l = y_r = y = F\frac{Y}{Z}.$$

7. (15 pt) (a)(10pt)Write a MATLAB script for coding a video frame f2 as a P-frame, using f1 as the reference frame. For each block, it finds the best intra-prediction and inter-prediction (f2 from f1), and use the prediction that has the smallest prediction error (in terms of sum of absolute difference). It then forms the prediction error block, and applies DCT transform on the prediction error (same transform block size as the prediction block), quantize the transform coefficients using a uniform quantizer centered at 0 with a constant quantization stepsize QS. The program should have the following syntax:
[f2q]=PframeCoding(f2,f1,QS,fp)
where f2q is the decoded frame for f2, fp is the pointer to the file storing the encoded bits for this frame.

Assume the following functions are given to you. Write all other operations that are necessary, including deciding which mode to use, performing DCT, quantization, inverse DCT. However, you can call dct2( ) and idct2( ) functions of MATLAB.

[IntraMode, PredBlock]=IntraPred(f, x0,y0,B), where f is the frame being coded, and x0, y0 is the top-left pixel coordinate of the block, and BxB is the block size; IntraMode is the best intra mode found, PredBlock is the best prediction block.
[vx,vy,PredBlock]=MotionEstimation(f1,f2,x0,y0,B), where f1 is the frame being coded, f2 is the reference frame for motion estimation, vx,vy is the motion vector found, PredBlock is the prediction block
[ModeBits]=BinaryEncodingMode(BestMode,IntraDirection,vx,vy); where BestMode is the chosen mode ("0"=intra, "1"=uni-directional inter), IntraDirection is the best intra-prediction direction returned by IntraPred( ), (vx,vy) is the motion vector of the current block in f2 with respect to f1; ModeBits are the binary bits generated for the mode information.
[CoeffBits]=BinaryEncodingCoef(QDCTIndics); where QDCTIndics include the blocks of quantized DCT coefficient indices, CoeffBits are the binary bits generated for QDCTIndics.
AppendBits(Bits, fp): append Bits to the compressed file

(b) (5pt) Write a main function for coding frames f1, f2, f3, f4, f5 as I-, B-, B-, B-, P-frames, using the **hierarchical B structure**, as indicated below. Assume the following functions are available:
[fq]=IframeCoding(f,QS,fp): coding frame f using intra-mode only
[f2q]=PframeCoding(f2,f1,QS,fp): coding frame f2 using either intra-mode or uni-directional prediction from f1
[f2q]=BframeCoding(f2,f1,f3,QS,fp): coding frame f2 using either intra-mode or uni-directional prediction from f1 or bi-directional prediction from f1 and f3.