Final Exam (5/18/2015 1:30-4:30)
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).

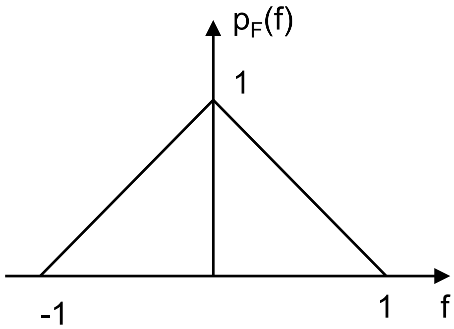
1. (15 pt) Consider a simplified JPEG like image coder, where each 4x4 block is transformed into DCT coefficients and these coefficients are then quantized and coded into binary bits using run-length coding along the zig-zag order. The figure below shows a 4x4 block of quantized DCT coefficient indices. 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 8 different possible values for the magnitude of the non-zero indices (from 1 to 8) so that each can be coded using 3 bits (level k will be coded using binary representation for k-1, i.e. 1 coded as 000, 7 coded as 110). Each non-zero index is coded with 4 bits, with 1 bit for the sign (0 for positive, 1 for negative) and 3 bits for the magnitude. For the runlength, the possible runlength symbols are 0, 1, …, 14, EOB. We will code the run-length values in the range of 0 to 6, the special symbol EOB, and the special symbol indicating the runlength is larger than 6 using a Huffman codebook, and code the remaining possible runlengths (7 to 14) using a fixed length code with 3 bits (with runlength *l* coded using binary representation of *l*-7).
   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)?

8	2	0	-3
0	0	0	0
-1	0	0	0
0	0	1	0

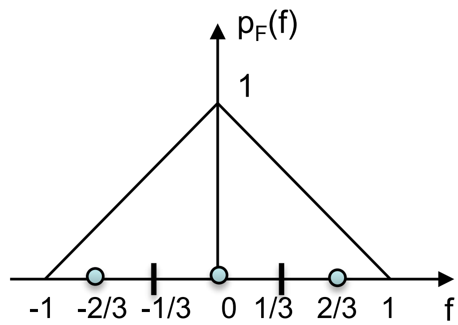
Quantized transformed coefficient indices

symbol	Prob.
0	0.2
1	0.15
2	0.1
3	0.1
4	0.05
5	0.03
6	0.02
EOB	0.15
Flag(>6)	0.2

2. (15 pt) Suppose we need to quantize a random variable F, which has a distribution shown in Fig. (a) below. On Fig. (b), we show a 3-level uniform quantizer, where the circles indicate the reconstruction levels, and the vertical lines indicate the partition boundaries.
   a. Derive the quantization error (in terms of MSE) of the uniform quantizer.
   b. Illustrate on the horizontal axis of Fig. (a) the likely locations of the reconstruction levels and partition boundaries of an optimal quantizer that that minimizes the MSE. Try to use symmetry to reduce the number of unknown variables to optimize.
   c. Derive the actual reconstruction levels and partition boundaries for the optimal quantizer, and the corresponding MSE. Compare it to the uniform quantizer. Note: if you run out of time, you could leave any integrals you write as is, without completing them. Also you should try to make use of the symmetry to reduce the computation as much as you can.



(a)



(b)

3. (15 pt) Suppose you want to code every two adjacent pixels in an image together using a 2-point transform with the following basis vectors:

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

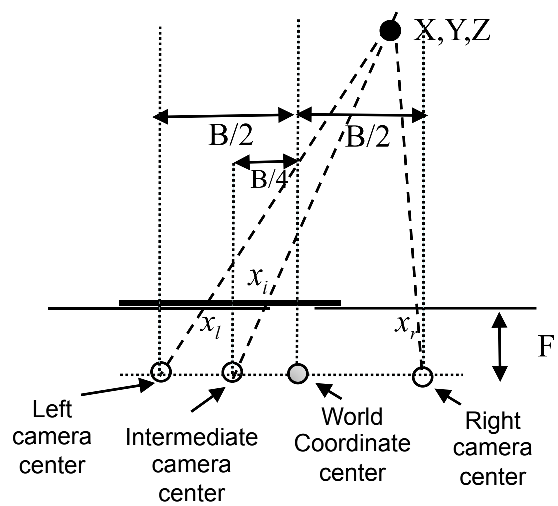
Assume all the pixels have same variance σ_s^2 and every two adjacent pixels have a correlation coefficient of ρ.
   a. What is the covariance matrix that describes the correlation between every two pixels?
   b. What is the covariance matrix of the transformed coefficients?
   c. Suppose you are targeting to use an average bit rate of R bits/pixel, how many bits you would assign to coefficient 1 and coefficient 2? (assume that you can assign fractional or even negative bits).
   d. What would be the average distortion per pixel (in terms of mean square error) using this bit allocation?
   e. What would be the average distortion per pixel if you quantize each pixel directly using R bits/pixel?
   f. From your result, can you say which method (direct coding or transform coding) is better?

   For this problem, assume that the quantization error for a scalar quantizer is related to the signal variance by $\sigma_q^2(R) = \varepsilon^2\sigma^2 2^{-2R}$, where σ^2 is the variance of each sample being quantized and coded, and ε^2 is a constant.

4. (10pt)
   a. Show a block diagram for computing a 1 stage 2D subband decomposition using 1D low pass filter $h_l(n)$ and high pass filter $h_h(n)$ to decompose an image into 4 subimages. You should only use 1D convolution. Your block diagram should use the original image as the input, and output 4 subimages, labeled as LL, LH, HL, and HH.
   b. Show another block diagram on how to compute a 3-level wavelet transform on an image (count the original image as level 1), where you apply the diagram you draw for (a) repeatedly.

5. (10pt)
   a. JPEG2000 achieves spatial scalability by using wavelet transform. Suppose you want to enable 3 spatial layers, show the wavelet decomposition structure (i.e. how is the image decomposed into different subbands) and specify which band(s) each spatial layer includes. If the original image has a size of MxN, what is the spatial resolution of each layer?
   b. H.264/SVC achieves temporal scalability by using hierarchical temporal prediction structure. Suppose you want to code a video into a temporal scalable video stream with 3 temporal layers. Illustrate the temporal prediction structure that you would use. If the original video has a frame rate of 30 Hz, what is the frame rate of your layer 1, layer1+ layer 2 and layer 1+layer2+layer 3?

6. (10pt) Consider a parallel stereo imaging system with baseline distance *B* and focus length *F* (see below). Suppose that for an object point at world coordinate *(X,Y,Z)*, its image position in the left and right view are (x_l, y) and (x_r, y), respectively.
   a. Describe how to estimate the 3D position *X,Y,Z* from x_l, x_r, y.
   b. Suppose we want to generate an intermediate view, whose camera center has a distance of *B/*4 away from the world coordinate origin, as shown below. How would you determine the image coordinate (x_i, y_i) for the same 3D point in this intermediate view? Express x_i, y_i in terms of x_l, x_r, y.



7. (10 pt) Suppose you are given a sequence of video frames taken when the cameraman's hand was not stable, so that successive video frames have some small global shift between each frame. You would like to stabilize this video by finding the shift between each frame and a center frame, and compensate the shift of each frame so that it is aligned with the center frame. (a) Write a MATLAB program that can estimate the global shift between two frames (f1 and f2), with the following syntax: [vx,vy]=EstiamteShift(f1, f2). (b) Write a MATLAB function for stabilizing 3 frames, which calls your EstimateShift( ) function. Your proposed should have a syntax: [g1,g2,g3]=stabilize(f1,f2,f3), where fi are original i-th frame, gi are stabilized i-th frame.

8. (15 pt) (a) Write a MATLAB script for coding a video frame f2 as a B-frame, using f1 and f3 as two reference frames. For each block, it finds the best intra-prediction, uni-directional inter-prediction (f2 from f1), and bi-directional inter-prediction (f2 from f1 and f3), and use the prediction that has the smallest prediction error (in terms of sum of absolute difference). It then form 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]=BframeCoding(f2,f1,f3,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( ) functiona of MATLAB.

- [IntraDirection, 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,vx1,vy1,vx2,vy2); where BestMode is the chosen mode ("0"=intra, "1"=uni-directional inter, "2"=bi-directional inter), IntraDirection is the best intra-prediction direction returned by IntraPred( ), (vx1,vy1) is the motion vector of the current block in f2 with respect to f1, and the (vx2,vy2) is the motion vector of the current block in f2 with respect to f3, both are returned by the MotionEstimation( ) function; 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) Write a main function for coding frames f1, f2, f3, f4, f5 as I-, B-, P-, B-, P-frames. 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.