**Polytechnic University, Dept. Electrical and Computer Engineering**
**EL6123 --- Video Processing, S08 (Prof. Yao Wang)**
**Final Exam, 5/1/2008, 3:00-5:50**
**Two sheest of notes (double sided) allowed, No peak into neighbors. Cheating will result in failing grade!**

1. (15 pt) Consider coding a 2-D random vector that is uniformly distributed over the region illustrated in Fig. 1(a). Suppose you want to design a codebook with 2 codewords. One possible codebook construction (codeword locations and region partition) is illustrated in Figure 1(b).
   a. Determine the value of y* in the upper triangle in Fig. 1(b) that will minimize the mean square error of the quantizer. Also determine the corresponding minimal mean square error.
   b. Another possible codebook configuration is shown in Fig. 1(c). Is this codebook better or worse than that in Fig. 1(b)? why?

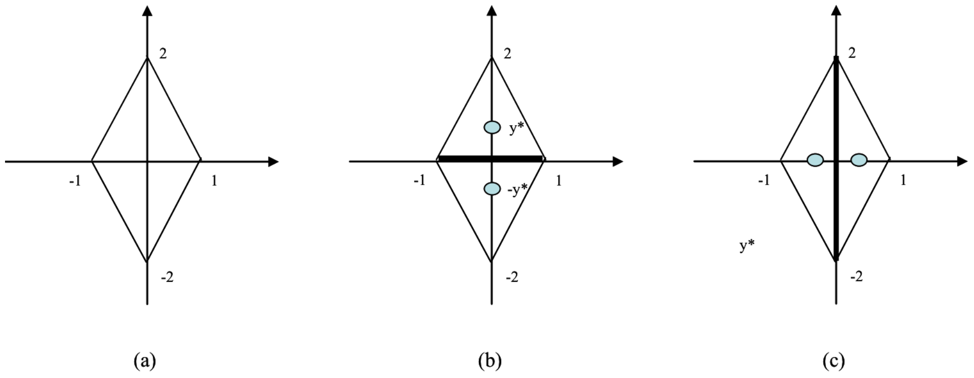

Figure 1

Solution:

a) For given y*, the MSE can be computed as $D(y*) = 4\int_0^1 \int_0^{2(1-x)} ((x-0)^2 + (y-y*)^2)\,dy\,dx$

One can find the optimal y* by minimizing D(y*), which can be done by setting $\frac{\partial D}{\partial y^*} = 0$. Detailed solution skipped. Once the optimal y* is found, you can substitute back into the equation for D(y*) to determine the MSE corresponding to this y*. (My rough calculation tells me y*=(\sqrt(21)-3 )/6. (but I cannot guarantee this is correct).

b) The coded in Fig.(c) is worser, as it has more points in each partition that are farther away from the codeword for that partition, than in Fig. (b).

2. (15 pt) Consider the following predictive coding method (see Fig. 2). A sample in frame n, $F = f_n(x,y)$ is predicted from its two neighboring pixels in the same frame $A = f_n(x-1,y)$ and $B = f_n(x,y-1)$and a pixel in a previous frame $D = f_{n-1}(x,y)$, using the linear predictor: $F = a(A+B)/2 + dD$.
   Suppose all samples have the same variance $\sigma^2$, and the correlations between these samples are $E\{FA\} = E\{FB\} = \rho_s\sigma^2, E\{FD\} = \rho_t\sigma^2$, and all other samples are uncorrelated. Find the optimal values for predictor coefficients *a* and *d* that will minimize the mean square prediction error, and determine the corresponding minimal prediction error. What is the coding gain compared to code each sample directly?
   Hint: you can treat *C=(A+B)/2* as a sample and make use of correlations among *F,C,D.*

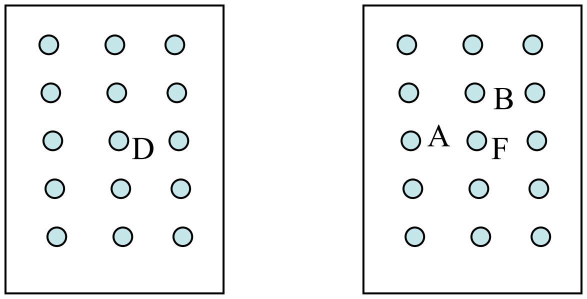

Figure 2

The predictor coefficients should satisfy the following equation:

$$\begin{bmatrix} E(CC) & E(CD) \\ E(DC) & E(DD) \end{bmatrix} \begin{bmatrix} a \\ d \end{bmatrix} = \begin{bmatrix} E(FC) \\ E(FD) \end{bmatrix}$$

E(CC)=E((A+B)*(A+B))/4=(E(A^2)+E(B^2)+2E(AB))/4=σ^2 / 2
E(CD)=E((A+B)D)/2=0
E(FC)=E(F(A+B))/2= \rho_s \sigma^2
E(FD)=\rho_t \sigma^2
The solution is a=2 \rho_s, d=\rho_t.

You can work out the rest.

3. (15 pt) Consider applying transform coding to every three pixels *A, B, C* in an image. The transform is 3-point DCT, whose basis vectors are given below. Suppose every pixel has the same variance $\sigma^2$ and the correlation between samples are $E\{A,B\}=\rho\sigma^2, E\{A,C\}=\rho\sigma^2, E\{B,C\}=0.$ (a) Determine the covariance matrix of the original three pixels. (b) Determine the covariance matrix of the transformed coefficients. (c) Determine the variances of the transformed coefficients. (c) If the total number of bits for the three coefficients is 3*R*, what is the optimal bit allocation to each coefficient? (d) What is the coding gain over direct coding of individual pixels? Express your results in terms of given variables.

The DCT basis vectors are: $\mathbf{u}_0 = \frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\end{bmatrix}, \mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\-1\end{bmatrix}, \mathbf{u}_2 = \frac{1}{\sqrt{6}}\begin{bmatrix}1\\-2\\1\end{bmatrix}.$

Solution: This is straight forward.

4. (15 pt) Consider stereo imaging using parallel camera configuration with a baseline distance of *B* and focal length *F*. Assume a 3D point with coordinate *(X,Y,Z)* is projected into left and right image planes with coordinates $(x_l, y_l)$ and $(x_r, y_r)$. These coordinates are related by

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

a. Suppose the 3D points being imaged fall on a planar surface described by the plane equation $Z = aX + bY + cB$. Show that the horizontal disparity $d = x_l - x_r$ at the image point $(x_r, y_r)$ can be described by an affine function of the form $d(x_r, y_r) = \left(F - (ax_r + by_r)\right)/\left(a/2 + c\right)$

b. Suppose we divide the right image into square blocks, and assume the 3D points corresponding to each block can be approximated as a flat surface so that the disparity function for each block can be modeled using an affine function (as shown in Part a). Propose an algorithm to estimate the affine parameters. You just need to formulate the problem as an optimization problem and describe in words how you may solve the optimization problem. Note that you can try this sub-problem even if you were not able to tackle Part a.

Solution:

$$d = x_l - x_r = \frac{FB}{Z}$$

On the other hand, using the relations given, we have

$$Z = aX + bY + cB = a\left(\frac{x_r Z}{F} + \frac{B}{2}\right) + \frac{b y_r Z}{F} + cB$$

Solvign for Z from above yields

$$Z\left(1 - \frac{a x_r}{F} - \frac{b y_r}{F}\right) = cB + \frac{aB}{2}$$

Or

$$\frac{1}{Z} = \frac{\left(1 - \frac{a x_r}{F} - \frac{b y_r}{F}\right)}{cB + \frac{aB}{2}}$$

Or

$$d(x_r, y_r) = \frac{FB}{Z} = \frac{(F - a x_r - b y_r)}{c + \frac{a}{2}}$$

(b) Here you should solve for a,b,c, for each block by minimizing the matching error defined as

$$E = \sum_{(x_r, y_r) in\ the\ block} (f_r(x_r, y_r) - f_l\,(x_l, y_l))^2$$

$$= \sum_{(x_r, y_r) in\ the\ block} (f_r(x_r, y_r) - f_l\,(x_r + d(x_r, y_r), y_r))^2$$

The optimization problem can be solved, e.g., using a gradient descent method.

5. (20pt) Be brief in your answers to the following questions.
   a. (6 pt) Why do different encoder products (hardware or software) following the same video coding standard have different coding performance? What parts of an encoder does the standard specify? (List one) What components of an encoder design are not subject to the standard and can differentiate among different products? (list 2 components at least)
   b. (8 pt) What are some of the benefits of scalable coding for video streaming applications? (List one at least). What are the three types of scalability? Propose one way to realize temporal scalability with base layer corresponding to a low-frame rate video, and base plus enhancement layer corresponding to a high-frame-rate video. In such a coder, the base layer can be predicted either from the previous frame reconstructed from the base-layer, or the previous frame reconstructed using both base and enhancement layer. What are the pros and cons of each approach?
   c. (6 pt) What causes error propagation in the decoded video when a compressed video bit stream is corrupted due to transmission? (list at least one cause) Describe two encoding tools that can be used to suppress error propagation.

Solution:

a) This is because the video coding standard only specifies the syntax that a bitstream needs to follow, not the method that needs to be used to generate that syntax. For example, the standard specifies that each macroblock should have a certain header, that tells which mode is used to code this block. The meaning of the remaining bits depends on the mode. If it is an intermode, the remaining bits specifies the binary bits corresponding to the quantized DCT coefficients for the inter-prediction error. The standard does not specify how should the encoder determine the mode, nor how should it determine the motion vector. Mode decision and motion estimation are two components that can be optimized by vendors.

b) Scalable coding enables the same video bit stream be accessed by receivers with different bandwidths. There are spatial, temporal, and SNR scalability. One way to generate temporal scalability is by applying a standard video coder to a low-frame rate video (down-sampled form the original, say, every other frames). The enhancement layer corresponds to the remaining frames. Each frame in the enhancement layer (e.g., f(n) can be predicted from either the previous frame (in the base layer) f(n-1), or the previous previous frame (the previous frame in the enhancement layer) f(n-2). For the base layer prediction, if one uses the previous frame f(n-1) (in enhancement layer) to predict f(n) in base layer, the encoder is likely to have a lower prediction error and hence higher coding efficiency, but there will be mismatch at the decoder, if the decoder only receives the base layer. If one uses previous previous frame f(n-2) (in the base layer) to predict f(n), the prediction is likely to be less accurate and hence the coding efficiency will be lower. But this approach does not cause any mismatch error in the receiver if it only receives the base layer.

c) There are multiple reasons to cause error propagation, including temporal prediction, spatial prediction and variable length coding. Some encoding tools that can suppress error propagation includes 1) periodic insertion of intra-blocks (or intra-frames), 2) insertion of synchronization markers; 3) multiple description coding.

d) (20 pt) Write a Matlab code that implements coding of a P-frame in a block-based hybrid video coder. For simplicity, use 8x8 blocks for both prediction and DCT coding, and consider the coding of Y-pixels only. For each block, you may decide to code it in either I-Mode (predict this block from previously coded samples in this frame using one of a few intra-prediction modes) or P-Mode (predict from a best matching block in the previous frame). For either mode, you perform DCT on the prediction error block, quantize the DCT coefficients using a preset QP and quantization matrix QMatrix, code the mode and the quantized DCT coefficients using an entropy coding method.

Denote the frame to be coded by CurrentFrame, the previous frame used for temporal prediction by PrevFrame. Your program should write the resulting bits for successive blocks into a file outfile. Your program should also compute and save the reconstructed frame in QuantizedFrame. Also use Width and Height to denote the width and height of a frame and assume both the width and height are dividable by 8. Furthermore, assume the following functions are available (i.e. can be called by your matlab code). Your program can call these functions as well as other MATLAB functions and functions defined by yourself.

function [mvh, mvv,PredictedMBlock]=MotionEstimation(bx, by, MBlock,PrevFrame):
finds the best matching block for a given 8x8 block (MBlock) in PrevFrame, bx, by are the top left coordinates of this block, [mvh, mvv] are the returned motion vector components, and PredictedMBlock is the best matching macroblock.

function [intra-mode,PredictedMBlock]=IntraPrediction(bx, by, MBlock,CurrentFrame):
intra-mode is the chosen intra-prediction mode that yields the best prediction.

function ImodeEntropyCoding(intra-mode,QuantizedDCTIndexBlock,outfile)
For a block to be coded in I-mode: entropy code information including the side info indicating I-mode, the chosen intra-mode and the quantized DCT indices (QuantizedDCTIndexBlock) and write the resulting bits into a file (outfile).

function PmodeEntropyCoding(mvx,mvy,QuantizedDCTIndexBlock,outfile)
For a block to be coded in P-mode: entropy code information including the side info indicating P-mode, the chosen motion vector (mvx,mvy) and the quantized DCT indices (QuantizedDCTIndexBlock) and write the resulting bits into a file (outfile).

function [QuantizedDCTIndexBlock]=quantizeDCT(DCTBlock,QP,QMatrix)
perform quantization on 8x8 DCT coefficients (in DCTBlock) with quantization parameter QP and QMatrix, return the quantization indices in QuantizedDCTIndexBlock.

function [QuantizedDCTBlock]=dequantizeDCT(QuantizedDCTIndexBlock,QP,QMatrix)
takes the quantized DCT indices of a block (QuantizedDCTIndexBlock) and applies inverse quantization to obtain quantized DCT coefficients (QuantizedDCTBlock)