

Polytechnic University, Dept. Electrical and Computer Engineering
EL612 --- Video Processing, S08 (Prof. Yao Wang)
Midterm Exam, 3/13/2008, 3:00-5:50PM

One sheet of notes (double sided) allowed. Cheating will result in failing grade!

1. (10 pt) Consider the following two raster scan formats: progressive scan using 20 frames/second, 500 lines/frame, and interlaced scan using 40 fields/second, 250 lines/field. For each scan format, determine
- The overall line rate (lines/second)
 - The maximum temporal frequency the system can handle
 - The maximum vertical frequency the system can handle
 - The maximum frequency (cycles per second) in the 1D waveform of the raster signal, assuming the maximum horizontal frequency is similar to maximum vertical frequency, and that the image aspect ratio (width:height) is 2:1.
 - Now suppose we need to digitize the raster scan video into full digital signal. What should be the sampling rate (samples/s) so that the horizontal spacing between samples is equal to vertical spacing, and the samples are aligned vertically?

Note: For parts (b)-(c), assuming a Kell factor=1 for simplicity. For (d)-(e): you only need to consider progressive scan (although the solutions for both are actually the same). Also, for all questions, you can ignore time needed for horizontal and vertical retrace.

2. (10 pt) Consider a video taken while the camera is undergoing a roll (rotation in the Z-axis) and followed by a zoom (change in focal length). Suppose the camera roll by an angle of θ between two frame capture times, the 3-D positions of any object point before and after the roll are related by

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Also, suppose the camera focal length was changed from F to F' . Assume that camera projection can be approximated by the perspective projection, i.e. $x = F \frac{X}{Z}$, $y = F \frac{Y}{Z}$. Derive the 2D motion field induced by this camera motion.

3. (10 pt) Suppose the motion between two adjacent frames f_1 and f_2 can be described well by a global affine mapping, described by:

$$d_x(x, y) = a_0 + a_1x + a_2y,$$

$$d_y(x, y) = b_0 + b_1x + b_2y.$$

One way to estimate the affine parameters a_k, b_k is by minimizing the sum of the squared errors between corresponding pixels in the two frames satisfy, using the first order gradient descent method. Set up your optimization problem using this approach and derive the first order gradient with respect to the bilinear parameters. Write down the gradient descent algorithm as an iterative procedure. You can assume an arbitrary initial solution for the affine parameters.

4. (10 pt) Given a motion field specified by motion vectors \mathbf{d}_k at pixels located at $(x_k, y_k), k = 1, 2, \dots, K$. (assume $K \geq 4$). We would like to approximate the motion field by the following projective mapping,

$$x' = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y}, \quad y' = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y}.$$

how would you derive the motion parameters? Write down the formulae you would use.

5. (10 pt) Consider lossless coding of a binary image (e.g. a scanned image of a black and white page with text and line sketches). Because there are usually consecutive black or white pixels along each line, one popular method is run-length coding. A white run-length refers to the number of consecutive white pixels in a line, and the black run-length refers to the number of consecutive black pixels. Each row in an image can be described by alternating white and black run-lengths. (You always start with a white run-length, which could be zero, and end with a white run-length, which also could be zero). You can pre-design the codewords for all possible white run-lengths and black run-lengths based on their respective probability distributions. Assume the probability of a white run-length= k is $P_W(k), k = 0, 1, \dots, K_{\max}$, and the probability of a black run-length= l is $P_B(l), l = 1, 2, \dots, L_{\max}$. Also assume the total number of white run-lengths after scanning an image is N_W , and the total number of black run-lengths is N_B , and finally the total number of pixels is N .
- Determine the lower bound on the average number of bits needed to code one white run-length, and that for coding one black run-length.
 - Determine the lower bound on the average number of bits per pixel achievable with this coding method.

6. (15 pt) Consider coding a 2-D random vector that is uniformly distributed over a circular region illustrated in Fig. 1(a). Suppose you want to design a codebook with 4 codewords. Figure 1(b),1(c),1(d) illustrate three possible codebooks with their corresponding region partitions.
- (3 pt) Which codebook will minimize the mean square quantization error?
 - (10 pt) Determine the x- and y- coordinate of each codeword in your chosen codebook that will minimize the mean square error. Determine the minimal mean square error. You can leave your solution in terms of some integrals, without getting the explicit solution.
 - (2 pt) Do the other two codebooks satisfy the necessary condition for minimizing the mean square error?

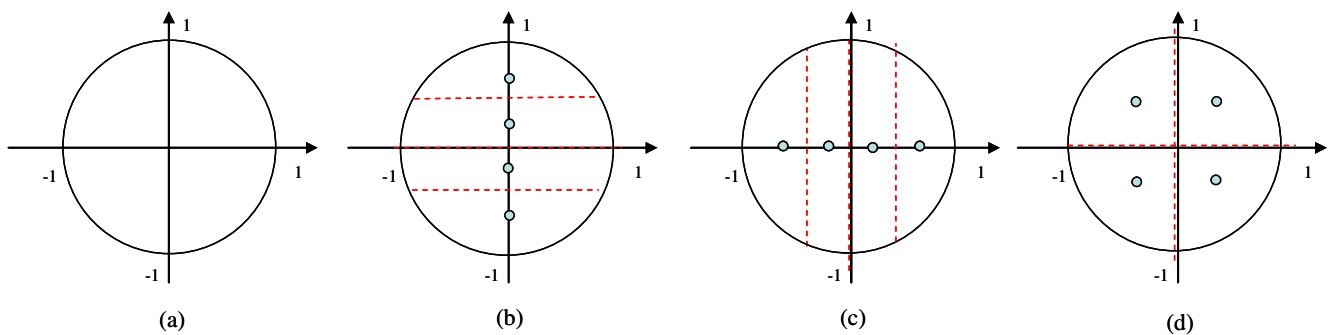


Figure 1

Hint: you should make use of symmetry in deriving your solution.

7. (15 pt) Vector Quantization
- (5 pt) What is the operation count of a nearest neighbor vector quantizer with vector dimension N and bit rate R bits/sample? Consider one addition, one subtraction, one multiplication each as one operation.
 - (10 pt) In order to reduce the complexity, we can code the norm (or gain) of an input vector using a scalar quantizer and the normalized vector (or shape vector) using a vector quantizer. This method is called *gain-shape vector quantizer*. Suppose an input vector is $\mathbf{f} = [f_1, f_2, \dots, f_N]$, its norm (called gain factor) is defined as $G = \left(\sum_n f_n^2 \right)^{1/2}$, the normalized vector is $\tilde{\mathbf{f}} = \frac{1}{G} \mathbf{f}$. If we use R_1 bits to quantize the gain factor G , and R_2 bits/sample to quantize the normalized vector $\tilde{\mathbf{f}}$, where R_1 and R_2 are chosen such that the total bit rate (bits/sample) is the same as in (a) (that is, $R_1 + NR_2 = NR$), what will be the total operation count for this method? What is the saving factor compared to direct vector quantization in part a? (Consider square root as one operation, also assume the scalar quantizer is in general non-uniform, and you need to use the nearest neighbor rule to determine the quantized level.)

8. (20 pt) Write a pseudo-code (in C or matlab style) for performing two level hierarchical block matching algorithm (HBMA). For simplicity, you should use integer-pel search at both levels. Also you should use the same block size of $B \times B$ in both levels. Your program should have the following syntax:

`[mvx,mvy,pimg]=HBMA(img1,img2,R1,R2,B,width,height)`

The input variables are

img1: the anchor image;

img2: the target image;

R1, R2: search range at coarse level (level 1) and the fine level (level 2).

B: block size is $B \times B$;

Width, height: the horizontal and vertical dimension of img1 and img2. For simplicity, assume width and height as well as width/2 and height/2 are all integer multiples of the block size B.

The output variables are:

mvx,mvy: the images storing the horizontal and vertical components of the estimated motion field, respectively;

pimg: the predicted image for the anchor frame using the estimated motion field.

Note that if you need to use some additional image arrays within the program, you should properly allocate its memory. You may also define any function that you can call from your program.

Note that if you cannot complete the entire program, you should try to draw a flowchart of your program to get partial credit.