University of California, San Diego
CSE 126 Multimedia Systems
Spring 2003

Inst: Prof P. V. Rangan
Midterm Exam (Form A)
Solution

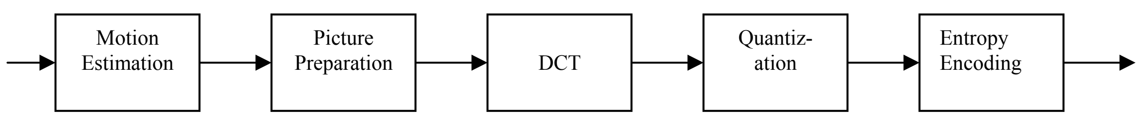
Assume the following input (before encoding) frame sequence (note that it is different from the one given in class) into an MPEG Encoder: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, …, and so on Assume the resolution of each frame is 320 by 240. The frames are encoded as follows:

frame 1, 13, … : I frames
frame 2, 7, 12, … : P frames
frame 3, 4, 5, 6, 8, 9, 10, 11, … : B frames

After each I frame, similar sequence of P and B frames repeat.

(1) Write the block diagram of video encoding in MPEG and describe in one sentence each of the blocks/stages in the diagram. Also state the inputs and outputs at each of the stages. Assume that the input to the video encoding is a sequence of raw frames of video. Describe briefly the motion compensation stages (i.e., the process of finding motion vectors), and then the JPEG stage, out of which should come the run-length encoded, quantized DCT Coefficients. Ignore the audio part, packetization, etc.

The video encoding in MPEG consists of 5 stages as described below:



1.1) Motion Estimation - breaks up a sequence of frames into smaller subsequences (GOPs), and encodes differences within a group by exploiting a relationship (motion) between successive frames.

Input: a raw sequence of frames
Output: a sequence of GOPs, each consisting of a subsequence of I/B/P frames

1.2) Picture Preparation – splits each frame into three components (Y,U,V). Then, elements in U and Y planes are subsampled by a factor of 2 and 8x8 blocks of values (pixel in Y component and meta-region in U and V components) are sent to the remaining stages.

Input: same as output from the previous stage
Output: block of size 8x8 values

1.3) DCT – transform 8x8 blocks from spatial domain into frequency domain

Input: same as output from the previous stage
Output: 8x8 blocks, whose elements are composed of 1 DC coefficient and 63 AC coefficients.

1.4) Quantization – quantize the results of the DCT to finite values so that they can be represented by a sequence of bits.

Input: same as the output from the previous stage
Output: 8x8 blocks of quantized values

1.5) Entropy Encoding – Parse the 8x8 blocks in a zig-zag order and compress the resulting sequence with run-length encoding.

Input: same as the output from the previous stage
Output: a sequence of encoded values

(2) Write the sequence "S" of frames AFTER they are encoded: i.e., the sequence in which they will ultimately be decoded. Call the first and thirteenth frames after being encoded as I to be: I1 and I13; The second , seventh, and twelfth frames after being encoded as P frames to be: P2, P7, and P12. The third, fourth, fifth, sixth, eighth, ninth, tenth, eleventh frames after being encoded as B frames to be: B3, B4, B5, B6, B8, B9, B10, B11. The sequence "S" that you write is just a sequence of all of these frames, exactly in the sequence in which they will be decoded.

I_1, P_2, P_7, B_3, B_4, B_5, B_6, P_{12}, B_8, B_9, B_{10}, B_{11}, I_{13}, …

(3) Does the above sequence "S" consist of independent or dependent GOPs?

Independent. No frame in one group is predicted from any other frame in the different group.

(4) What is the maximum number of motion vectors that might result while encoding all of the frames from I1 through P12?

Since there is no motion vector in I frame, we need to check only P and B frames. We first determine how many macro-blocks are in one frame. The size of a macro-block is 256 pixels and the total number of pixels in one frame is 320x240; therefore, the total number of macro-blocks in one frame is (320x240)/256. We know that one macro-block in P frame has one corresponding motion vector whereas one macro-block in B frame can have two corresponding macro-blocks. Hence, the maximum number of motion vectors in the frame sequence from I_1 to P_{12}, which consists of 3 P frames and 8 B frames, is ((320x240)/256) x (3 + 8x2)

(5) In the above sequence, which are the forward and backward reference frames for frame B6?

Forward reference frame – P_2
Backward reference frame – P_7

(6) Assume that to encode B6, the encoder searches both of its forward and backward reference frames entirely for best matches for each of B6's macroblocks.
Supposing that:

- computing the mean square error (MSE) between a 16x16 region in a reference frame and a macroblock in B6 takes 10 cpu cycles.
- averaging two 16x16 regions takes 5 cpu cycles.

Ignore the time taken by operations other than the above two. In the worst case, totally how many cycles are needed for the process of finding the motion vectors for all of the macroblocks in B6?

Macroblocks can be encoded via intra-coding, forward prediction, backward prediction or bi-directional prediction. To find the best reference blocks (if there is any) to predict a macroblock, all of these encoding possibilities must be entirely checked. We consider each of them separately to determine how many cycles are required to find the motion vector for each macroblock.

6.1 Intra-coded – there is no MSE and 16x16 region averaging computation in this encoding, so it takes 0 cycle.

6.2 Forward prediction – it is required to find the MSE between a macroblock and all possible 16x16 blocks in the reference frames (P_2). One MSE computation takes 10 cycles and there are totally (320-15)x(240-15) 16x16-blocks in the reference frame, so it takes 10 x (320-15)x(240-15) cycles

6.3 Backward prediction – same number as forward prediction

6.4 Bidirectional prediction – in this category, a macroblock is predicted from two blocks: one from its forward reference frame and another from its backward reference frame. We need to check all possible combinations of these two blocks, which can be counted as $((320-15)\mathrm{x}(240-15))^2$. Each check needs 15 cpu cycles (5 cycles for 16x16 block averaging and 10 cycles for MSE computation. Hence, it requires totally $15\mathrm{x}\ ((320-15)\mathrm{x}(240-15))^2$ cycles.

In conclusion, the process of finding the motion vector for all macroblocks in B frames takes the following cpu cycles:

(# macroblock) x (# cycles per macroblock) = $((320\mathrm{x}240)/256)\ \mathrm{x}\ ((2\mathrm{x}10\mathrm{x}(320-15)\mathrm{x}(240-15) + (15\mathrm{x}((320-15)\mathrm{x}(240-15))^2)))$)

Consider one such macroblock in B6, after its motion vectors are computed. The macroblock is now passed through JPEG-like encoding. For this Macroblock, compute the following:

(7) How many blocks of Y, U, and V will this macroblock result in?

We know that a macro-block is a 16x16 block. Since Y block is a 8x8 block and U/V block is a 16x16 block (after subsampled by a factor of 2), there are 4 Y, 1 U and 1 V blocks in one macro-block

(8) How many DCT Coefficients (give the maximum number possible) will this macroblock result in?

One 8x8 block contains64 DCT coefficients; therefore, one macro-block consists of (4+1+1)x64 DCT coefficients.

(9) How many DC Coefficients will this macroblock result in?

One block has one 1 DC coefficient; therefore, one macro-block contains only (4+1+1) DC coefficients.

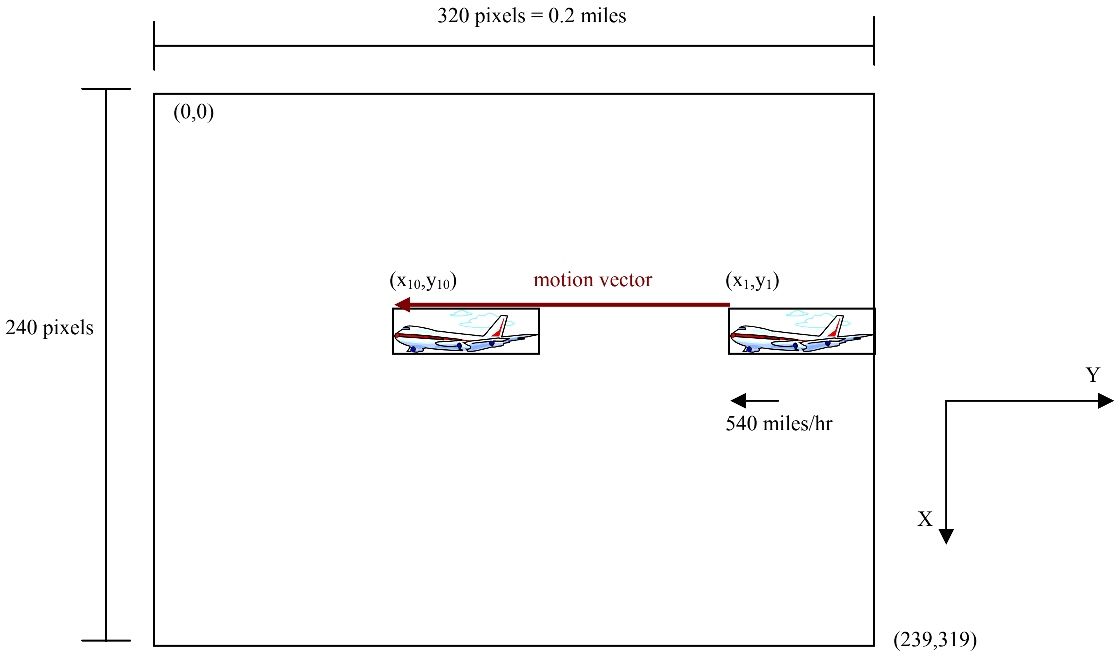
(10) If the DC coefficient for the very first block of Y-component, belonging to the very first macroblock of frame B6 is lost, what exactly do we lose among all of the blocks of Y, U, and V for this frame?

The DC coefficients are encoded using DPCM which is subject to error propagation. If the DC coefficients of the very first block in the frame is lost, all DC coefficients of y-component blocks will be lost. Nevertheless, because y-,u-,v-component blocks are encoded separately, the loss of y-component block doesn't affect u- and v-component blocks.

(11) Consider a video of resolution 320x240 and 15 frames per second. Assume that you are trying to encode it using MPEG-1. Supposing it is that of an airplane flying in the sky, and in the first frame, the airplane is on the right most part of the frame, and the size of the airplane exactly fits one macroblock. The airplane moves right to left horizontally at 540 miles per hour, and the frame is showing 1/5 mile wide track. Assume camera does not move. Let one GOP be 15 frames with every tenth frame being a P frame (i.e., first frame is I, next 8 frames are B, 10^{th} frame is P, next five frames are B, next is I, and it repeats).

What will be the motion vector for the airplane-macroblock in the first P frame? For purposes of this question, assume that motion vector for any macroblock MB is defined as the two dimensional vector, specified by its start and destination. The start is the top left corner of its reference macro-region (call it as MR), and destination is MB's top left corner. So you have to give the coordinates of the start and the destination of this motion vector. Assume the coordinate system has its origin at the top left corner of a frame, horizontal rightwards is Y direction, and vertical downwards is X direction, and each pixel counts as one unit of length along either direction. Carefully check your answer. Do not overlook any factor.

In this question, we need to find the motion vector of the airplane-macroblock in the tenth frame which is predicted from the airplane-reference-block in the first frame.



Let (x_1, y_1), (x_{10},y_{10}) be coordinates of the top-left corner of the airplane-blocks in the first frame and the tenth frame respectively. The motion vector can be written as $((x_{10}-x_1)$, $(y_{10}-y_1))$. Since the airplane doesn't change its height, the difference between x_{10} and x_1 is 0. As a result, we only need to determine the difference between y_{10} and y_1, which is, in another word, the negative value of the distance that the airplane moves horizontally.

Since the refresh rate of the video is 15 frames/sec, the tenth frame is the snapshot of the airplane 9/15 second behind that of the first frame. We can compute the distance that the airplane move as follows:

distance = time x speed
= (9/15 sec) x (540 miles/hr)
= (9/15 sec) x (540/3600 miles/sec)
= (9/100 miles)
= (9/100 miles) x (320/(1/5) pixels/miles)
= 144 pixels.

Because the airplane move from left to right by 144 pixels from the first frame to the tenth frame, the motion vector of the airplance-macroblock is (0,-144).