CompE 565: Multimedia Communication Systems

HW 3: Motion Estimation for Video Compression

Learning Goals: Learn a Motion Estimation technique for reducing the temporal redundancy in video sequences.

Project Description:

Design and implement the full search motion estimation technique for video compression, by using the search window of 32x32 pixels, which is centered on the current macroblock. Use five frames (from frame #6 to frame #10) of the given motion video 'walk_qcif.avi, while ignoring the first five and the remaining frames (note: you should extract these five video frames from the sequence). In QCIF format, the frame size is 176x144 pixels and you should use 4:2:0 format. Please use the IPPPP structure with GOP size of 5 frames, and predict all the P frames from the I frame. You can use the SAD method for motion estimation. You are required to generate the difference frame for each P-frame, and then generate the reconstructed P-frames by performing decoding.

Note: You need not perform DCT and Quantization in this assignment.

Your report should include the following:

- A thorough discussion of your implementation and the simulation results (including the display of error (or residual) frame images, and the video images reconstructed by using motion compensated macroblocks).
- You should also provide a visual representation (i.e., plot) of motion vectors for each frame. **Hint:** Matlab has a function "quiver" that may be used to display motion vectors.
- You should discuss the computational load (number of multiplications, additions, and comparisons) of your motion estimation algorithm.

Motion Estimation

Motion estimation over a macroblock of pixels is a standard approach for estimating motion in a moving image sequence. Three important parameters in motion estimation are: the matching criteria, search technique, and prediction mode.

Use a macroblock size of 16x16 pixels. Recall that for a given displacement **d** = (d_x, d_y), the Sum of Absolute Difference (SAD) for a macroblock $B_k(n)$ is given by:

$$Distortion(B_k(n),d) = \sum_{\forall (x,y) \in B_k(n)} \left| U_k(x, y) - U_r(x + d_x, y + d_y) \right|$$

where, $U_k(x,y)$ represents the pixel intensity in the current frame, and $U_r(x,y)$ represents the pixel intensity in the reference frame (i.e., I frame in this home assignment).

For a macroblock size of 16x16 pixels, SAD requires 2x256 -1 = 511 additions/subtractions.

Exhaustive (or Full) Search Block Matching Algorithm:

In *Exhaustive Search* (ES), every possible displacement within a rectangular search window is attempted. The displacement that produces the minimum distortion is chosen as the motion vector. As shown in Figure 1, if the maximum search range in either direction is w (assuming a square search), $(2w+1)^2$ possible values exist for the motion vector.

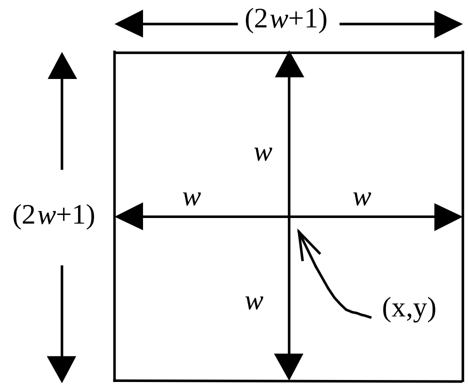

Figure: Motion Search Range

For ES, the distortion measure must therefore be calculated and compared $(2w+1)^2$ times. The resulting motion vector will be the one that minimizes the distortion within the search range. The cost of minimizing the distortion in ES is high computational intensity. For example, for a maximum displacement of $w = 6$, the matching criteria must be evaluated $(2x6+1)^2 = 169$ times. If SAD is chosen as the distortion criteria, each macroblock requires 169x511= 86,359 additions/subtractions and 169 comparisons. For a video frame resolution of 352x288 pixels, the 1,584 macroblock motion vectors need to be calculated (assuming no motion detection).

If we assume a frame rate of 30 frames/second, there are over 4G additions/subtractions and 8M comparisons per second, for a frame size of 352x288. The high computational requirements of ES make it unacceptable for many real time image sequence coding applications.