HW SET #4 (DUE BEFORE CLASS ON FEB 17, WED)

Problem 1 (100 points) Write a program for motion estimation. Use the same three test sequences as in HW SET #1. If you use a web browser to download these files, make sure you check the file size after downloading them. Each file should be $352\times288\times1.5\times10=1{,}520{,}640$ bytes. Or you can access them directly from `/afs/ece/class/ece796/public_html/videos/*`. Please see the `readme.txt` file to familiarize yourself with the file format. Each video sequence is composed of 10 pictures. If you use PC and wish to view these sequences in action, you can find a `YUVplayer.exe` in the same directory (Remember to select CIF, not QCIF). Each picture is composed of the Y component at the full resolution, and the U and V components at the quarter resolution. Use only the Y component for motion estimation. Each macroblock is of size 16×16. Use the *MAD* (mean absolute distortion) for block matching. That is, given the reference picture $F(x,y)$ (the previous picture), for each 16×16 array of pels in the source picture $G(x,y)$, find the motion vector (dx,dy) such that

$$MAD=\frac{1}{256}\sum_{i=0}^{15}\sum_{j=0}^{15}\left|G(x+i,y+j)-F(x+dx+i,y+dy+j)\right|$$

is minimized. By convention, (x,y) refers to the upper left corner of the macroblock, indices i and j refer to pel values to the right and down, and displacements dx and dy are positive when pointing to the right and down. Use only integer values for dx and dy. The range for both dx and dy is $[-15,15]$. Use exhaustive search for all possible (dx,dy) in this range. If there is a tie, choose the motion vector that has smaller magnitude (i.e., smaller $\sqrt{dx^2+dy^2}$). Store the motion vectors (dx,dy) in a text (ASCII) file in the following format:

```
0  0    (-2,9)
0  1    (-1,7)
0  2    (-1,5)
0  3    (0,1)
…
```

in the order of left to right for each row of macroblocks, and then top to bottom in each picture, and then continuously for all the nine pictures (The 0-th picture is used only as a reference picture). In other words, the first column of the file indicates the row number of the macroblock, the second column indicates the sequence number of the macroblock in this row, and the third column indicates the corresponding motion vector (dx,dy). Note that motion vectors are not allow to point outside the picture, so please constrain the motion vectors of macroblocks on the picture boundary.

We will test your program with some test sequences. Please deposit your code into your directory in `/afs/ece/class/ece796/handin/[your userID]/hw4`. Put all your source files (`*.h`, `*.c`, etc.) and the executable (please name it `motion_est`) there. Please also include a

(OVER→)

`readme.txt` file if the way to re-compile your code to generate the executable may not obvious for the TA. The TA will test your code by typing:

```
motion_est inputfilename outputfilename
```

where `inputfilename` is the name of a test sequence in the yuv format. Our TA will then check the content of `outputfilename` to verify your results. Make sure that `outputfilename` should be a text (ASCII) file.

In addition, please compute the histogram of the motion vectors you obtain for each of the three test sequences. Each histogram should be an array of 31×31 elements, with each element indicating the frequency of a particular motion vector (dx, dy). Plot each histogram as a 2D image. Make sure you normalize the intensity of these images properly (or, if you use MATLAB for plotting these images, make sure you define the color map properly).

You will find that using exhaustive search for motion estimation is very computationally intensive. Please feel free to be creative to produce the most efficient code for the exhaustive search.