CS 576 Fall 2014– Assignment 2
Instructor: Parag Havaldar

Assigned on Monday 10/06/14
Solutions due on Friday 10/24/14 (before 12:00 noon)
Late Policy: None, unless prior arrangement has been made

This assignment will help you gain an understanding of issues that relate to image compression, by comparing and contrasting the frequency space representations using the Discrete Cosine Transform and the Discrete Wavelet Transform. You will read an RGB file and convert the file to a 8x8 block based DCT representation (as used in the JPEG implementation) and a DWT representation. Depending on the second parameter *n* you will decode both the representations using only *n* coefficients and display them side to side to compare your results. Remember all input files will have the same format as explained on the class website. They will be of size 512x512 (intentionally square so that decoding can be done easily) and your encoder/decoder process will work on each channel independently.

Input to your program will be 2 parameters where:

- The first parameter is the name of the input image file. (file format description provided and is similar to the first assignment format of RGB)
- The second parameter is an integral number that defines the number of coefficients to use for decoding. The interpretation of this parameter of decoding is different for both the DCT and DWT cases so as to use the same number of coefficients. Please see the implementation section for an explanation

Typical invocations to your to your program would look like

MyExe Image.rgb 262144
Here you are making use of all the coefficients to decode because the total number of coefficients for each channel are going to be 512*512= 262144. Hence the output for each DCT and DWT should be exactly the same as the image with no loss.

MyExe Image.rgb 131072
Here you are making use of 131072 (half of the total number) of coefficients for decoding. The exact coefficients you use will vary depending on DCT or DWT. Refer to the implementation section for this.

MyExe Image.rgb 16384
Here you are making use of 16384 (one-sixteenth of the total number) of coefficients for decoding. The exact coefficients you use will vary depending on DCT or DWT. Refer to the implementation section for this.

Implementation (75 points)

Your implementation should read the input file and convert (for each channel) a DCT representation and a DWT representation. Based on the input parameter of the number of coefficients to use, you need to appropriately decode by zeroing out the unrequested coefficients (just setting the coefficients to zero) and then perform an IDCT or a IDWT. The exact coefficients to zero out are different for both cases and explained next.

For a DCT, you want to **select the first m coefficients in a zig order for each 8x8 block** such that *m* = round(*n*/4096) where *n* is the number of coefficients given as input. So for the second test run above (*n*=131072), you will use *m* = 32 = round(131072/4096). Each block will be decoded using the first 32 coefficients in zigzag order. The remaining can be set to zero prior to decoding. For the third test run above (n=16384), you will use *m* = 4 = round (16384/4096).

For a DWT, you want to **select the first n coefficients in a zig order for the image** where *n* is the number of coefficients given as input. DWT encodes the entire image so, here you will use the first n coefficients in zig zag order.

Progressive Analysis of DCT vs DWT (25 points)

Here you will create an animation which will take incremental steps of decoding in order to study the output quality of your DCT vs DWT implementation. For this invocation we will put *n* = -1, because the incremental number of coefficients are predefined as shown below.

MyExe Image.rgb -1
This suggests the program has predefined incremental settings on the number of coefficients to use and is going to loop through them incrementally.

We have 4096 8x8 blocks in the image. So let us start by using one coefficient of each block and then incrementing it on each iteration. Total number of progressive iterations would be 64

For the DCT decoding, you will use

- first iteration - the DC coefficient for each block (total 4096 coefficients)
- second iteration – the DC, AC_1 coefficient of each block (total 8192 coefficients)
- third iteration – the DC, AC_1, AC_2 coefficient of each block (total 12288 coefficients)

…
…

- sixty forth iteration – the DC, AC_1, AC_2 …. AC_{64} coefficient of each block (total 512*512= 262144 coefficients)

For the DWT decoding, you will use

- first iteration - the first 4096 coefficients in zigzag order.
- second iteration – the first 8192 coefficients in zigzag order.
- third iteration – the first 12288 coefficients in zigzag order

….
….

- sixty forth iteration – all the total 512*512= 262144 coefficients.

What should you submit ?

- Your source code, and your project file or makefile. Please confirm submission procedure from the TAs. Please do not submit any binaries or data sets. We will compile your program and execute our tests accordingly.
- Along with the program, also submit an electronic document (word, pdf, pagemaker etc) for the written part and any other analysis/extra credit explanations.

The following is a step by step navigation on D2L for submitting the homework
1. Login to courses.uscden.net
2. Select CSCI 576 course from the right hand side tab on the home page
3. Click on My Tools and select Dropbox
4. On this page you will see a folder called Assignment 2, click on it and make the submission by uploading your files and add any comments for the grader