

### CSCI 576 Assignment 3

Instructor: Parag Havaldar

Assigned on 03/15/2021

Solutions due 03/31/2021 before 10 am, morning

#### THEORY QUESTIONS – (40 points)

##### Question 1: DCT Coding (15 points)

In this question you will try to understand the working of DCT in the context of JPEG. Below is an 8x8 luminance block of pixel values and its corresponding DCT coefficients.

188	180	155	149	179	116	86	96
168	179	168	174	180	111	86	95
150	166	175	189	165	101	88	97
163	165	179	184	135	90	91	96
170	180	178	144	102	87	91	98
175	174	141	104	85	83	88	96
153	134	105	82	83	87	92	96
117	104	86	80	86	90	92	103

- Using the 2D DCT formula, compute the 64 DCT values. Assume that you quantize your DCT coefficients uniformly with  $Q=100$ . What does your table look like after quantization? (2 points)
- In the JPEG pipeline, the quantized DCT values are then further scanned in a zigzag order. Show the resulting zigzag scan when  $Q=100$ , (1 point).
- For this zigzag AC sequence, write down the intermediary notation (2 points)
- Assuming these are luminance values, write down the resulting JPEG bit stream. For the bit stream you may consult standard JPEG VLC and VLI code tables. You will need to refer to the code tables from the ITU-T JPEG standard which also uploaded with your assignment. (2 points)
- What compression ratio do you get for this luminance block ?

##### Question 2: Image Dithering (25 points)

Let's say that we have an original image of size  $m \times n$  with 8 bits per pixel. A certain sub section of this image matrix is represented in a  $12 \times 8$  sub image below and the values are normalized. In the normalized representation shown below, assume that 0 corresponds to white and 9 corresponds to black.

1	2	3	4	5	6	7	8	9	0	1	2
0	1	2	3	4	5	6	7	8	9	0	1
9	0	1	2	3	4	5	6	7	8	9	0
8	9	0	1	2	3	4	5	6	7	8	9
7	8	9	0	1	2	3	4	5	6	7	8
6	7	8	9	0	1	2	3	4	5	6	7
5	6	7	8	9	0	1	2	3	4	5	6
4	5	6	7	8	9	0	1	2	3	4	5

Answer the following questions You may want to code/script a process to generate the final outputs, but only final outputs are expected. Also, we have asked you to plot this 12x8 image block as a gray color image, and its processed outputs as binary black/white images. For reasons of visible clarity (because a 12x8 image block is very small) you may want to show a zoomed or magnified picture.

- Plot the image as an 8-bit gray scale map, that is - create a 12x8 image to show the original gray image block. (2 points)
- If you threshold the above 12x8 image block such that all values below 4.5 were 0 and above 4.5 were 9, how does your output image B/W block look? Plot an image (3 points)
- We can create a better binary image output by using a dithering matrix. Compute the binary output of a dithering operation on the gray level 12x8 image using the dithering matrix D given below. Assume that the sub image block's top left coordinates indexes start as [0,0]. Show a graphical binary image plot of the dithered output. (5 points)

$$D = \begin{bmatrix} 5 & 6 & 8 & 4 \\ 1 & 0 & 3 \\ 5 & 2 & 7 \end{bmatrix}$$

- What if the sub image block's top left coordinate indexes start with [1,1]. Show a graphical binary image plot of the dithered output. (5 points)

## PROGRAMMING DCT vs DWT – (160 points)

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. All input RGB files will be of size 512x512 and in the same rgb format used in the previous assignments. You will read an RGB file and convert the file to an 8x8 block based DCT representation (as used in the JPEG implementation) as well as a DWT representation (as used in the JPEG2000 implementation). Each will contain 262144 frequency coefficients wither DCT or DWT. 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. Your algorithm, whether encoding or decoding, should work on each channel independently. *Display the two outputs side by side.*

Input to your program will be 2 parameters where:

- The first parameter is the name of the input image file. (All images will be given in the rgb format and have a size 512x512 for ease in encoding/decoding)
- The second parameter  $n$  is an integral number that defines the number of coefficients to use for decoding. The value of  $n$  will be specific – (262144, 65536, 16384, 4096, 1024, 256, 64, 16, 4, 1) The interpretation of this parameter for 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 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  $512 \times 512 = 262144$ . Hence the output for each DCT and DWT should be exactly the same as there is no loss and should be the original image.

*MyExe Image.rgb 65536*

Here you are making use of 65536 (one fourth of the total number) of coefficients for decoding. While the number of coefficients are the same for both DCT and DWT decoding, the exact coefficient indexes are different. Refer to the implementation section for this.

*MyExe Image.rgb 16384*

Here you are making use of 16384 ( $1/16^{\text{th}}$  of the total number) of coefficients for decoding. While the number of coefficients are the same for both DCT and DWT decoding, the exact coefficient indexes are different. Refer to the implementation section for this.

### **Implementation**

Your implementation should read the input file and convert (for each channel separately) a DCT representation and a DWT representation.

#### **Encoding:**

For the DCT conversion, break up the image into 8x8 contiguous blocks of 64 pixels each and then perform a DCT for each block for each channel. For the DWT conversion, convert each row (for each channel) into low pass and high pass coefficients followed by the same for each column applied to the output of the row processing. Recurse through the process as explained in class through rows first then the columns next at each recursive iteration, each time operating on the low pass section.

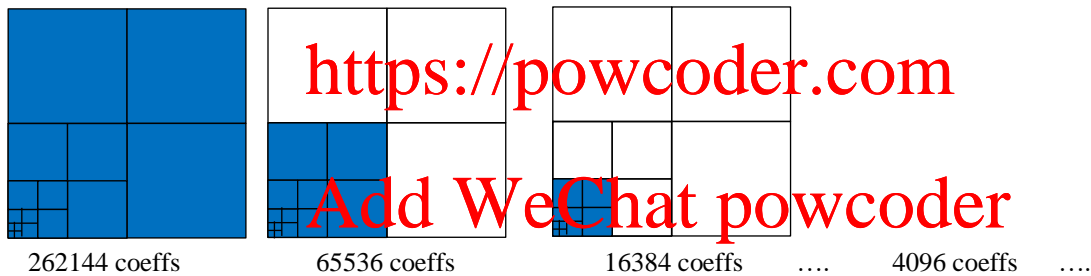
#### **Decoding:**

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 an IDWT. The exact coefficients to zero out are different for both the DCT and DWT cases and explained next.

For a DCT, you want to **select the first  $m$  coefficients in a zig zag order for each 8x8 block** such that  $m = \text{ceil}(n/4096)$  where  $n$  is the number of coefficients given as input. 4096 is the number of 8x8 blocks in a 512x512 image. Thus,  $m$  represents the first few coefficients to use for each 8x8 block during decoding. So for the second test run above ( $n=65536$ ), you will use  $m = \text{ceil}(65536/4096) = 16$ . Each block will be decoded using the first 16 coefficients in zigzag order. The remaining coefficients in each 8x8 block can be set to zero prior to decoding. For the third test run above ( $n=16384$ ), you will use  $m = \text{ceil}(16384/4096) = 4$ . If the ratio of  $n/4096$  becomes fractional, then  $m=1$  because of the ceil function, ie you are ensuring using at least one coefficient of every DCT 8x8 block.

For all your tests, the parameter  $n$  for decoding will correspond to the number of coefficients in the Low Pass outputs for both rows and columns at any level in a DWT. In a DWT, as shown below, there are 262144 coefficients after the hierarchy is processed. You will appropriately use  $n = 262144$ , or 65536, and so on as per the input. For DWT decoding step set all the remaining coefficients to zero and proceed with an IDWT.



### What should you submit?

- Your source code, and your project file or makefile. Please do not include any binaries or data sets. **Zip all the files together and name it as FirstName\_LastName\_Assignment3.zip. This will help us handle submissions better.** 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 extra credit explanations.