



## Homework #8

**Due Date: 11:59pm Tuesday 20 May 2014**

In this homework you will write a C++ program that can compress images using the JPEG compression algorithm. Your program will read the image on `stdin` and outputs the resulting image on `stdout`, and get in the required operation on the command line.

Please present a report containing your answers as well as a zip file containing all your code.

1. [4 points] Implement the function `compress_jpeg()` to compress images using the JPEG algorithm. You will need your earlier implementations of the Huffman Coding algorithm and the DCT compression. The compression algorithm should perform the following operations:
  1. Computes the DCT transform for each  $8 \times 8$  block in the input image
  2. Quantize the DCT coefficients using the given quality factor
  3. Perform RLC on the AC coefficients of each  $8 \times 8$  block independently using the zig-zag ordering
  4. Encode the resulting triplets (runlength, size, value) using Huffman coding
  5. Perform DPCM on the DC coefficients of all the  $8 \times 8$  blocks in the image in row-major order

6. Encode resulting pairs (size, value) using Huffman coding

Try the function on the small images `msg1.pgm` and `msg2.pgm` and to make sure it is working correctly.

**Note:** You can use the standard quantization table used in the DCT homework, and so assume it is known to the encoder and the decoder i.e. you don't need to include it in the output compressed file. However, you will need to include the computed Huffman Tree for the entropy coded AC and DC DCT coefficients. Follow the method outlined in the lectures, and consult the book or the Internet for more details.

**Note:** *before* computing the DCT on the input blocks, subtract 128 from each pixel, and add it back *after* computing the inverse DCT.

2. [3 points] Implement the function `decompress_jpeg()` to decompress an image compressed using the function implemented above. The function will perform the inverse of the operations to decompress the image and reconstruct its approximation.

Try the function on the small images `msg1.pgm` and `msg2.pgm` and to make sure it reconstructs a good approximation for them and correctly decodes them their compressed versions.

3. [2 points] Try the JPEG compression/decompression functions on the four images (`sena`, `sensin`,

earth, omaha) using different quality values ( $quality = 0.1, 0.5, 1, 10$ ). Compare the compressed file size to the JPEG compression/decompression in the Linux program `convert` (which is part of the ImageMagick package) with different quality levels (25%, 50%, 75%, 90%). Comment on the image quality and file sizes produced from both algorithms.

## Command Line

You need to modify the main file `hw08.cpp` to include the required functionality. Your program should be named `hw08`, and should be called as follows:

- To compress an image with JPEG and output the representation of the compressed image on the stdout:

```
./hw08 -compress q < input.ppm
```

where  $q$  is the input quality value, the input image is called `input.ppm` and the output is written to `stdout`. For example, to compress `msg1.ppm`, with quality  $q = 1$ , you could run:

```
./hw08 -compress 1 < msg1.ppm
```

- To decompress an image compressed with `-compress` above and output the reconstructed image to the stdout:

```
./hw08 -decompress < input.ppm
```

where the input compressed image is called `input.ppm` and the output is written to `stdout`. For example, to compress/decompress the image `msg1.ppm`, you could run:

```
./hw08 -compress 1 < msg1.ppm | ./hw08 -decompress
```

## **Instructions**

- All code should be implemented in C++ under Linux.
- Please submit your homework in one zip file named as follows: *CMPN206.HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *CMPN206.HW01.Mohamed.Aly.zip*.
- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

## **Grading**

- 9 points: requirements above
- 1 point: submission instructions