

The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
COMP4421 (Fall 2016)

Assignment 3

Total = 100 marks

Due: 11:55pm, Dec 6, 2016

Assignments must be submitted via CANVAS

Late Policy: 10% reduction; only one day late is allowed, i.e. 11:55pm, Dec 7

Overview

This assignment consists of two sections: programming section and written section. Both programming and written parts should be submitted via the CANVAS system. If you would like to finish the written assignment with hand writing, you may scan and upload it.

In the programming section, you need to finish an incomplete function which is a part of JPEG lossy-compression using discrete cosine transform (DCT). A set of M-files can be obtained from CANVAS and the code skeleton has been implemented. You need to complete the missing implementations in the programming section.

In the written section, you need to answer one question about image compression.

Programming assignment specifics (60%)

You need to complete the implementation in the *compression_jpeg.m* file, which is a part of JPEG lossy encoder/decoder using DCT. The steps of this compressor can be summarized as follows:

- (1) Subtract the image intensity by 128.
- (2) Partition the input image into 8x8 blocks. For example, if the input image has a resolution of 128x128, there are 16*16=256 blocks after partitioning.
- (3) Apply DCT to each of the 8x8 block (use command "DCT2" in Matlab)
- (4) Perform quantization with Quantization Table and convert all the quantized coefficients into integer.
- (5) Following zig-zag order, convert all 8x8 blocks into vectors with 64 elements.

The procedure for the decompression of the image is the simple inverse of the above steps:

- (1) Convert the vectors back to 8x8 blocks.
- (2) Multiply the blocks with quantization table.
- (3) Apply Inverse DCT to each block.
- (4) Group all 8x8 blocks back to form an image.
- (5) Add 128 to each pixel of the image.

You need to implement step (3) & (4) in compression and step (2) & (3) in decompression.

Sample runs of the programming assignment

sample_run.fig is the sample output. You are supposed to obtain similar output on the screen when you run the following command in the MATLAB environment:

```
>> compression_jpeg
```

Written assignment specifics (40%)

Lossless compression (Huffman coding).

Given a 1-D image profile as follows (intensity values range from 0 to 15):

| | | | | | | | | | | | | | | | | |
|-----------|---|---|----|----|---|---|---|---|---|----|----|----|----|----|----|----|
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Intensity | 1 | 1 | 15 | 12 | 2 | 3 | 9 | 7 | 7 | 1 | 9 | 12 | 1 | 1 | 12 | 12 |

Perform Huffman coding for the image and calculate the compression ratios.

Show all your steps clearly.

For Huffman coding, show your steps in terms of the tables as shown in the Huffman Encoding part of the lecture notes (Image Compression). You should assign label '1' to the intensity with higher frequency in the Huffman code assignment table.

In the calculations of compression ratios, ignore the storage overhead due to the lookup tables.