National University of Singapore

Department of Mathematics

10/2017 Semester I MA4268 Mathematics in visual data processing **Project 2**

Scenario

Image compression is one key module in imaging system which aims at reducing the storage of images without loss of important information. Let I denote the image for compression. The goal of image compression to find a integer coefficient matrix C such that it requires less storage than I, while the image reconstructed from C is close to I. One strategy to achieve compression is to find such a matrix C whose most entries are zero.

Let r denote compression ratio which represents the ratio of the numbers of non-zero entries of C and the number of the pixels of the image I. Taking $r = 10$ and image size 256×256 for example. The compression ratio is then $10 : 1$ which means the number of non-zero entries of C is no large than $(0.1 * 256^2) \approx 6553$.

Procedure and implementation

In this project, we will implement some key parts of one popular compression technique, namely JPEG. The procedure of image compression is described as follows.

1. Partition an image $I \in \mathbb{R}^{M\times N}$ into non-overlapping image patches of size 8×8, denoted by $\{P_{m,n}\} \subset \mathbb{R}^{8\times 8}$. The total number of such patches will be $\frac{MN}{64}$.

2. For each image patch $P_{m,n}$, decompose it by 2D discrete Cosine transform and let $C_{m,n} \in \mathbb{R}^{8\times 8}$ denote the corresponding coefficient matrix.

3. Constructing the integer coefficient matrix $C \in \mathbb{R}^{M\times N}$ by concatenating all 8×8 matrices $C_{m,n} \in \mathbb{R}^{8\times 8}$ and converting all entries to 32-bit signed rintegers:

4. Setting $(1-\frac{1}{r})MN$ (rounding it to integer) entries of C with smallest magnitude to zero.

The procedure of image de-compression is described as follows.

1. Partition the coefficient matrix $C \in \mathbb{R}^{M\times N}$ into non-overlapping patches of size 8×8, denoted by $D_{m,n} \subset \mathbb{R}^{8\times 8}$. The total number of such patches will be $\frac{MN}{64}$.

2. For each coefficient patch $D_{m,n}$, reconstruct an image patch of size 8×8 by running 2D inverse discrete Cosine transform on it, and let $Q_{m,n} \in \mathbb{R}^{8\times 8}$ denote the corresponding image patch.

3. Constructing the image $J \in \mathbb{R}^{M\times N}$ by concatenating all 8×8 image patches $Q_{m,n} \in \mathbb{R}^{8\times 8}$, and converting all entries to unsigned 8-bit integers.

In this project, the following routine is expected.

1. *imcomp.m*

```
IMCOMP Compressing the input image with input compression ratio.
C = imcomp(I,r) computes the coefficient matrix of  an image I
with input compression ratio r, returning a  coefficient matrix C
with most entries being zero.
```

2. *imdecomp.m*

```
IMDECOMP Computing the compressed image from an integer matrix C.
J = imdecomp(C) computes an image  from the input integer matrix C,
returning an image J.
```

Remark I: The format of the entries of the matrix I and J should be 8-bit unsigned integer (*uint8* in Matlab), and the format of the entries of the matrix C should be 32-bit signed integer (*int32* in Matlab). You might find more information on these two formate by type: *help uint8* and *help int32* in Matlab.

Remark II: Before doing calculations on data in the format of *uint8* and *int32*, always first convert them to doubel-precision floating numbers, which can be done by the built-in Matlab function *double*.

Data for testing

Use the grey-scale image provided by Project 1, or use your own with image size that can be divided by 8.

Grading Policy

This is an individual project. You may discuss the ideas with your classmates, but the sharing of code is strictly prohibited. You are not allowed to use any code from online sources or from other classmates' projects. The grade of the submitted project is based on the following two factors: 1) whether meets the guidelines; 2) the correctness of the results.

Submissions

You are required to submit the following:

1. *The codes of "imcomp.m" and "imdecomp.m"*

2. *The test grey-scale image*

3. *The matlab script file "demo.m" for running the above routines on the enclosed image. The sample code could be as follows.*

```
im_1 = imread('sample.png');
r = 10;
c = imcomp(im_1,r);
im_2 = imdecomp(c);
imwrite(im_2,'sample_new.png')
subplot(1,2,1); imshow(im_1)
subplot(1,2,2); imshow(im_2);
```

Please package all your files in a single zip file named by your student ID and upload the zip file into IVLE, e.g., U1234567.zip. The deadline for submission is Wed., 29-Nov-2017. Any submission after the deadline will lead to some penalty.

If you have any question, please contact me by email: matjh@nus.edu.sg for help.