National University of Singapore

Department of Mathematics

09/2017 Semester I MA4268 Mathematics in visual data processing **Project 1**

Goal

Discrete Cosine Transfrom is one essential transform for many image processing tasks. The goal of this project is to implement two-dimensional discrete Cosine transform and inverse discrete Cosine transforms for matrices, introduced in the class. Recall that the DCT definition of a 1D signal $\{f[n]\}_{0\leq n<N}$

$$c[n] = \sqrt{\frac{2}{N}}\lambda_n \sum_{k=0}^{N-1} f[k]\cos(\frac{n\pi}{N}(k+\frac{1}{2})), \quad 0 \leq n < N,$$

where

$$\lambda_n = \begin{cases} \frac{1}{\sqrt{2}} & n = 0; \\ 1 & n \neq 0. \end{cases}$$

The inverse transform is defined as

$$f[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \lambda_n c[n]\cos(\frac{n\pi}{N}(k+\frac{1}{2})), \quad 0 \leq k < N.$$

The two-dimensional DCT is a direct extension of the 1D case using tensor product:

$$c[n,m] = \frac{2}{N}\lambda_n\lambda_m \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f[i,j]\cos(\frac{n\pi}{N}(i+\frac{1}{2}))\cos(\frac{m\pi}{N}(j+\frac{1}{2})), \quad \text{for } 0 \leq n,m < N.$$

and the inverse transform is

$$f[i,j] = \frac{2}{N} \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} \lambda_n\lambda_m c[n,m]\cos(\frac{n\pi}{N}(i+\frac{1}{2}))\cos(\frac{m\pi}{N}(j+\frac{1}{2})), \quad \text{for } 0 \leq n,m < N.$$

The two-dimensional DCT can be done via first running 1D DCT on each column (row), followed by running 1D DCT on each row(column), which often is more efficient than directly implementing 2D DCT.

Implementation

In this project, the following two routines are expected.

1. *dct2d.m*

```
DCT2D Computing the 2D discrete cosine transform of a given matrix
    F = dct2d(I) computes the  2D discrete cosine transform of the
    matrix I, returning a cosine transform coefficient matrix F.
```

2. *idct2d.m*

```
IDCT2D Computing the 2D matrix from its discrete cosine coefficients
    I = idct2d(F) computes the  matrix from its discrete cosine
    coefficient matrix F, returning a matrix I.
```

The following assumption is allowed in your implementation: the image size is of $2^N \times 2^N$ where N is a positive integer. You may NOT call or use any code from the following built-in functions in MATLAB:

```
fft, ifft, fft2, ifft2, dct, idct, dct2, idct2
```

Image Data and experiments

Using the sample image of size 256×256 provided by this project, or use you own grey-scale image, to test your code. The running time of your code may be recorded using the following MATLAB functions:

```
tic, toc
```

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 three factors: 1) whether meets the guidelines; 2) Correctness of the output; and (3) Computational efficiency in terms of running time. *Remark: Computational efficiency of Python and MATLAB implementations will be evaluated independently.*

Submissions

You are required to submit the following:

1. *The codes of "dct2d.m" and "idct2d.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 = double(imread('sample.png'));
coeff = dct2d(im_1);
im_2 = uint8(idct2d(coeff));
```

```
subplot(1,2,1); imagesc(coeff);colorbar;
subplot(1,2,2); imshow(im_2);
```

4. *The text file recodes the running time on the enclosed image and any comments you would like to make.*

Please package all your files in a single zip file named by your student ID and upload the zip file into the directory of "student submission" in the workbin of the module in IVLE , e.g., U1234567.zip. The deadline for submission is Friday, 27-Oct-2017. Any submission after the deadline will lead to some penalty on grade.

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