

## CS 370 Winter 2018: Assignment 3

**Due March 15, 5 pm.**

Instructor: G. Labahn  
Lectures: MWF 8:30, 11:30

Office: DC3629

e-mail: *glabahn@uwaterloo.ca*  
Office Hours: Tues 11:00-12:00

Instructor: Y. Li  
Lectures: MWF 1:30

Office: DC3623

e-mail: *yuying@uwaterloo.ca*  
Office Hours: Thurs 2:00-3:00

Web Site: cs370 piazza

**Your assignment should be handed in electronically on UW Learn. The submission should include one pdf file containing the assignment answers and the cover sheet and all the m-files required to run the code, with no folder structure (not zipped).**

### Analytic Questions

**1. (10 marks)**

- (a) Calculate by hand the discrete Fourier transform of  $f = (1, 2, 3, 2)$ .
- (b) Calculate by hand the inverse discrete Fourier transform of  $F = (4, -1, 0, -1)$ .

**2. (15 marks)**

Let  $\{F_0, \dots, F_{N-1}\}$  be the DFT of a sequence  $\{f_0, \dots, f_{N-1}\}$  defined by

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n W^{-nk}$$

with  $W = e^{\frac{2\pi i}{N}}$  the  $N^{th}$  root of unity. Show your work and simplify where possible.

- (a) Give a formula for  $F_k$  for all  $k$  when  $f_n = (-1)^n$  for  $n = 0, \dots, N-1$ .
- (b) Suppose  $(f_0, \dots, f_{N-1}) = (-1, \dots, -1, 1, \dots, 1)$  having the first half negative one and remaining half one (with  $N$  even). Show that  $F_{2k} = 0$ ,  $k = 0, 1, \dots, \frac{N}{2} - 1$ .

**3. ( 20 marks)**

Consider the sequence of eight numbers  $f = (1, 0, 2, 0, -1, 0, -2, 0)$ .

- (a) What are the two arrays ( $g$  and  $h$ , each of length 4) that are used in computing the DFT of  $f$  by the FFT method?
- (b) Compute  $G$  and  $H$ , the two DFTs for  $\{g_i\}$  and  $\{h_i\}$ , respectively, using the definition of DFT of 4 values. Simplify if possible.
- (c) Using  $G$  and  $H$  from part b), write out the DFT of the array  $f$ .
- (d) Using the FFT butterfly algorithm, compute by hand the DFT of  $f$ .

4. **(15 marks)** (Edge Sharpening)

A discrete grey scale image is given by a two dimensional array of grey scale values  $f_{ij}$  defined on a square grid  $x_j = jh$   $y_j = jh$  where  $h$  is the grid spacing. An algorithm for edge enhancement is

$$f^* = f - \alpha \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)$$

where  $\alpha$  is a positive, constant parameter. If  $F$  is the two dimensional Discrete Fourier Cosine Transform (DCT) of  $f$  then a continuous approximation to the discrete grey scale values is given by

$$f(x, y) = \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} F_{k,\ell} \cos\left(\frac{2\pi kx}{N}\right) \cos\left(\frac{2\pi \ell y}{N}\right).$$

Apply the edge sharpening algorithm to this continuous approximation.

Edge enhancement is typically viewed as sharpening higher frequencies in an image. If we view the result in terms of the following algorithm

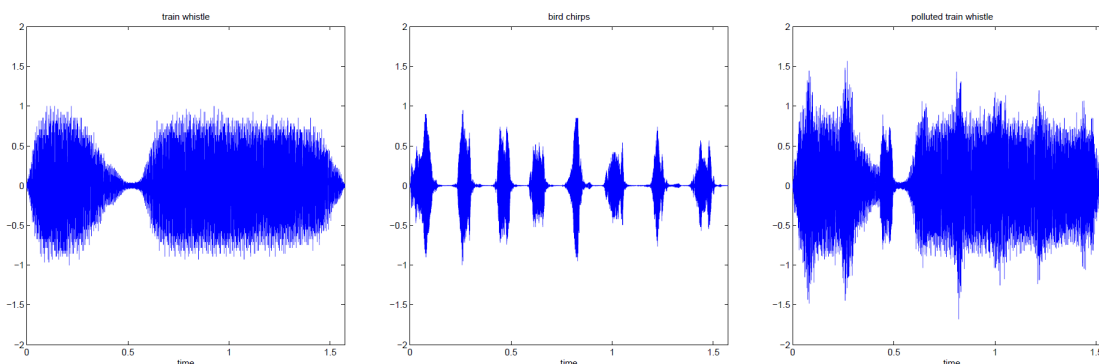
$$\begin{aligned} F &= DCT(f) \\ F^* &= Filter * F \\ f^* &= inverseDCT(F^*) \end{aligned}$$

then interpret the meaning of this filter and explain why it tends to sharpen edges.

### Programming Questions

5. **Signal Processing (20 marks)**

Sound samples of a train whistle are recorded at a rate of 8192 samples per second. The total number of samples is  $N = 12880$ , and thus the length of the signal is 1.57 seconds. However, the recording is polluted by the background noise of bird chirps. The signals of the train whistle, bird chirps, and the polluted train whistle are shown below.



The polluted recording has been obtained simply by superimposing the two signals. As one can see, in the time domain it would be rather difficult to separate the bird chirps from the train whistle in the mixed signal. Your job is to denoise the polluted (noisy) signal by using the discrete Fourier transform. The noisy signal is provided on the website in the file

`train_bird.mat`. Use the `load` command to load the data into your workspace in MATLAB. The file contains the signal, `y`, and the sampling rate, `Fs`. Once the data is loaded you can hear the signal by using the command `soundsc(y,Fs)`.

Write a MATLAB script to do the following:

- (a) Load the data and plot the input signal.
- (b) Compute the DFT of the original signal, and plot the power spectrum. (Hint: use the `fft` and `abs` commands.)
- (c) Implement a low-pass filter to isolate the train whistle, and a corresponding high-pass filter to isolate the bird chirps. (Hint: study the power spectrum carefully, and use trial and error.)
- (d) Plot the denoised signal of the train whistle and its power spectrum.

**Note:** use the `stem` command to plot the power spectrums.

Submit the following:

- A hard copy of your code and the four figures produced by it. Present the plots of the signals and the plots of the spectra side-by-side for comparison. (Hint: use the `subplot(2,2)` command.)
- A brief description of your implementation and the results. In particular: (i) describe what your code does, (ii) identify the frequency threshold you used to filter the signal, and (iii) comment on the similarities and differences between the noisy and denoised signals.

## 6. Image Compression (20 marks)

In this problem, we study the compression of gray-scale images. In Appendix F of the course notes, *Images in Matlab*, you have the information needed to convert such images into a two-dimensional array. Compression is obtained by dropping relatively small Fourier coefficients on  $8 \times 8$  pixel subblocks. By this we mean that if  $\{f_{i,j}\}$  are the original pixel values in a given subblock, and  $\{F_{k,\ell}\}$  are the corresponding DFT, then we drop any  $F_{k,\ell}$  satisfying

$$|F_{k,\ell}| < F_{max} \cdot tol.$$

Here  $F_{max}$  is the maximum of  $\{|F_{k,\ell}|\}$  in each block and  $tol$  is our drop tolerance.

The file `mountain.jpg`, available on the course web site, contains an image to be used in this question.

### a) Compression

Create a MATLAB function, `Compress.m`, that has the following prototype:

$$[Y, drop] = \text{Compress}(X, tol)$$

It takes as inputs the original image, `X`, and the drop tolerance parameter, `tol`, and outputs a compressed image `Y`. It also returns the drop ratio, `drop`, which is defined to be:

$$\text{drop ratio} = \frac{\text{Total number of nonzero Fourier coefficients dropped}}{\text{Total number of pixels in the image}}.$$

If drop ratio = 0, then no nonzero Fourier coefficient is dropped; if drop ratio = 1, then all Fourier coefficients are dropped. In general, it should be between 0 and 1.

Your MATLAB function needs to do the following:

- Compute the 2D Fourier coefficients (**fft2**) for every  $8 \times 8$  subblock.
- For each subblock, set those Fourier coefficients having modulus less than the relative tolerance level to 0.
- Record the number of coefficients dropped.
- Reconstruct the compressed  $8 \times 8$  image array by using the inverse 2D Fourier transform (**ifft2**). **Note:** the reconstructed image array must be set to the real part (**real**) of the inverse transform.
- Return the entire compressed image as **Y** and the drop ratio as **drop**, after all the  $8 \times 8$  subblocks for all the components have been processed.

b) **Compression Levels**

Determine (by trial and error) four values of **tol** resulting in drop ratios of 0.1, 0.45, 0.75 and 0.9. Write a MATLAB script, **MyScript.m**, to do the following:

- Execute **Compress.m** with these sets of **tol** values.
- Display the four compressed images (using **subplot** and **subimage**). Indicate the **tol** value and the resulting drop ratio used for each image.
- Plot the normalized mean square error between the original image and the compressed image vs the drop ratio for the compressed image. If  $Y$  is the processed image, and  $X$  is the original image, the normalized mean square error can be found with the Matlab command:

$$\text{nms\_error} = \sqrt{\text{mean2}((Y-X).^2)/(\text{mean2}(X).^2)};$$

**Submit the following:**

- (a) A copy of the functions **Compress.m** and **MyScript.m**.
- (b) A figure with the 4 plots of the compressed images.
- (c) The error vs drop ratio plot of the compressed images.
- (d) A brief commentary on your results.