CS 370 Fall 2008: Assignment 3

Instructor: Professor Keith Geddes

Lectures: MWF 3:30-4:20 MC 2017

Web Site: UW-ACE

Due: Thu Oct 30, 2008, 5:00 pm, in the Assignment Boxes, 3rd Floor MC

Note: Although you do not have to complete this assignment before the Midterm Test, some portions of this material will be covered on the test. Specifically, you should complete Task 1 and Task 2 as part of your preparations for the Midterm Test.

Task 1: Discrete Fourier Transform (DFT)

(a) Let $f = [1\ 2\ 3\ 4]$. Compute the Fourier coefficients of f by hand using the definition of the discrete Fourier Transform (DFT).

(b) Let f_n, $n = 0, \ldots, N-1$ be real periodic input data with $N = 2^m$, and $f_{N+n} = f_n$. Let F_k, $k = 0, \ldots, N-1$ be the DFT of f_n. Show that if the f_n values satisfy the symmetry $f_n = f_{N-n}$ then the Fourier coefficients F_k are all purely real numbers.

(c) Let f_n, $n = 0, \ldots, N-1$ be given data values (real or complex) and let F_k, $k = 0, \ldots, N-1$ be the DFT of f_n. Show that

$$\sum_{k=0}^{N-1} F_k \overline{F_k} = \frac{1}{N} \sum_{n=0}^{N-1} f_n \overline{f_n}\,.$$

Task 2: Fast Fourier Transform (FFT)

Let $f = [1\ 2\ 3\ 4\ 1\ 2\ 3\ 4]$. In this question, you will compute the Fourier coefficients of f by hand using an FFT algorithm. More precisely, you will illustrate one level of the recursion inherent in the particular FFT algorithm outlined in section 7.7 of the Course Notes.

Note that this FFT algorithm (from Section 7.7) uses a slightly different divide and conquer method than the other FFT algorithm discussed in class which used a "polynomial evaluation" point of view.

(a) What are the two vectors g and h, each of length 4, that are used in computing the DFT of f by this FFT algorithm?

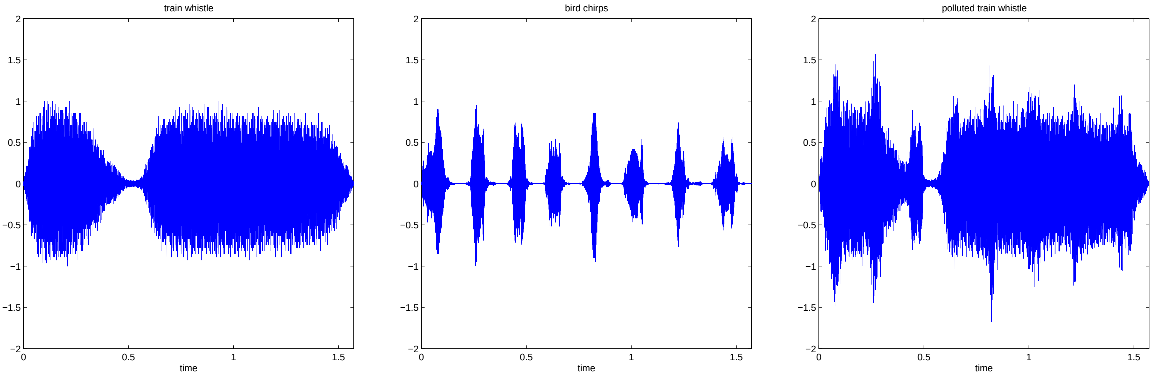
(b) Compute G and H, the two DFTs for $\{g_i\}$ and $\{h_i\}$, respectively, using the definition of the DFT of 4 values. Simplify if possible.

(c) Let F denote the DFT of the array f. Using G and H from part (b), calculate F as it would be done by the FFT algorithm.

Show your work.

Task 3: Signal Processing

Sound samples of a train whistle are recorded at a rate of 8192 samples per second. The total number of samples is $N = 12880$. 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 in the following Figure.



By listening to the sound signals, it is clear that the train whistle is composed of two consecutive tones, and the bird chirps eight distinct tones. The polluted recording is obtained by superimposing the two signals. As one can see, in the time domain, it is rather difficult to remove the bird chirps signal from the polluted recording as it is mixed with the train whistle.

In this part of the assignment, you are asked to denoise the polluted signal by using the discrete Fourier transform. The noisy signal is stored in the file `train_bird.wav` on UW-ACE in the Assignment 3 directory.

Specifically, write a MATLAB script, `denoised.m`, to do the following:

- read in and plot the noisy signal; use the command
  `[f freq Nbits] = wavread('train_bird.wav');`
- compute the discrete Fourier transform `F` of the noisy signal, and plot the power spectrum (Hint: use the `abs` command)
- create a denoised signal by
  - identifying the range of coefficients of `F` associated with the bird chirps by trial and error (Hint: study the power spectrum carefully and listen to the sound of your proposed denoised signal; it is also useful to listen to the separated signal of the bird chirps)
  - setting these coefficients to zero
- plot the denoised signal and its power spectrum.

It is very helpful to hear the noisy and denoised signals, and the separated bird chirps. You can do so in MATLAB using the `sound` command.

What to submit for Task 3

Put the following together and label the figures:

1. A brief (one or two sentences) comment on the similarities and differences between the power spectrum of the noisy and denoised signals.
2. A brief comment on the the similarities and differences between the noisy and denoised signals.
3. A statement about which Fourier coefficients you have set to zero.
4. The program listing of `denoised.m` and the 4 figures produced by it. Present the plots of the signals and the plots of the spectra side-by-side for comparison (use the `subplot(2,2)` command).

Task 4: Image Compression

This image compression method is based on the idea of representing the image using only a subset of its Fourier coefficients, those with the largest modulus. From the Assignment 3 directory on UW-ACE, copy the file `mc_dc.jpg` to your assignment directory. Create a Matlab function, `compress`, that has the following prototype:

```
[g, drop] = compress(f, tol)
```

where `f` is the original image, `tol` is the drop tolerance parameter, `g` is the output compressed image, and `drop` is the drop ratio defined to be

$$\texttt{drop} = \frac{\texttt{total number of Fourier coefficients dropped}}{\texttt{total number of pixels in the image}} .$$

If the drop ratio is 0, then no Fourier coefficients were dropped. If the drop ratio is 0.5, then half of the Fourier coefficients were dropped. The drop ratio should be between 0 and 1.

Your `compress` function should break the original image into 16 × 16 subblocks, and do the following for each:

- Compute the 2D DFT (using `fft2`).
- Set to zero the Fourier coefficients whose modulus is smaller than `tol`.
- Record the number of coefficients dropped.
- Reconstruct the compressed 16 × 16 image using the inverse 2D DFT (`ifft2`).
  Note: the reconstructed image is the **real part** of the result.

After all the 16 × 16 subblocks have been processed, return the entire compressed image as `g` and the drop ratio as `drop`.

After creating and debugging `compress.m`, write a Matlab script called `compression_demo.m` that loads the image `mc_dc.jpg` and

- sets four `tol` values that result in drop ratios of approximately 0.5, 0.75, 0.95 and 0.99,
- executes `compress` with each of these `tol` values, and
- displays the four compressed images using `subplot(2,2,...)`.

What to submit for Task 4

Put the following together and label each page:

1. a listing of your `compress.m` function;
2. a listing of the Matlab script `compression_demo.m`;
3. a printout of the four images, each labeled with the appropriate drop ratio.