Digital Image Processing: 525 U0920 / ESOE 5096

Computer Assignment 4
Wavelet Transforms and their Applications

Due: the week after next, 9:00pm

*Hand in online via **NTU COOL***

Total score: 180

In this assignment, we will use MATLAB's Wavelet Toolbox along with provided custom functions to perform a series of experiments, which hopefully will make you better understand the properties of wavelets and their vast applications. The provided set of routines allows wavelet-based image processing using the Image Processing Toolbox without the need of the Wavelet Toolbox, which you will be familiar with after the completion of this assignment and use in case no Wavelet Toolbox is available in the future.

What to turn in:

Matlab source codes that perform each individual experiment and the associated results and comments if any. Please consolidate them into one single file.

1. Execution time comparison (20%)

The Wavelet Toolbox has a function called **wavedec2** that can be used to perform the multilevel 2-D wavelet decomposition. Alternatively, the provided custom function **wavefast** can do the same job. The goal is to compare the execution times of **wavedec2** and **wavefast** using function **timeit**. For example, we can obtain transform and computation time using

```
w1 = @() wavedec2(img, n, wname);
t1 = timeit(w1);
```

Similarly, we can obtain the transform and computation time for **wavefast**, which are denoted as **w2** and **t2**, respectively. Load the image of *hurricane-katrina.tif* using

```
img = imread('hurricane-katrina.tif');
```

With the same **wname** of **'db4'** (4th order Daubechies) compute the following:

(a) The ratios of **t2** to **t1** for **n = 1 to 10** and plot it. (10%)

(b) Plot the maximum absolute differences between **w1** and **w2** for **n = 1 to 10**. (10%)

2. Transform coefficient display (40%)

As experimented in Problem 1, the custom function **wavefast** provides attractive results with neglectable error and compatible execution time comparing to **wavedec2**. Another handy custom function is **wavedisplay**, which displays wavelet decomposition coefficients. First, load the image *zoneplate.tif* and compute the wavelet decomposition coefficients using the Haar wavelets, i.e.,

[cm, sm] = wavefast(img, 2, 'haar');

You can display the coefficient matrices using

wavedisplay(cm, sm, scale);

where **scale** is a parameter for magnifying the coefficients. Please display the coefficients using different settings of **scale**:

(a) Display with the default setting, i.e., without the **scale**. (10%)

(b) Magnify default by the scale factor, say, **scale = 8**. (10%)

(c) Magnify absolute values by abs(scale) with **scale = -8**. (10%)

What differences do you observe from these three different displays? (10%)

3. Wavelet directionality and edge detection (30%)

Consider the 500×500 *test_pattern.tif* image and compute the 4th order Symlets wavelet transform using

[cm, sm] = wavefast(img, 1, 'sym4');

(a) Display the wavelet coefficients with **wavedisplay(cm, sm, -6);** Observe the horizontal, vertical, and diagonal directionality of the single-scale wavelet transform. (10%)

(b) The custom function **wavecut** allows you to zero coefficients in a wavelet decomposition structure using **[nc, ym] = wavecut('a', cm, sm);** Display the zeroed coefficients with **wavedisplay(nc, sm, -6);** What do you observe in comparison to (a)? (10%)

(c) The custom function **waveback** computes inverse FWTs for multi-level decomposition by using **waveback(nc, sm, 'sym4');** After taking the absolute value of this output matrix and converting the type with **mat2gray**, display the image. What do you observe? (10%)

4. Wavelet-based image smoothing (50%)

Wavelets, like their Fourier counterparts, are effective instruments for image smoothing. Consider again the *hurricane-katrina.tif* image and take its Cohen-Daubechies-Feauveau biorthogonal wavelet transform using

[cm, sm] = wavefast(img, 4, 'bior6.8');
where a four-scale decomposition is performed. To perform the smoothing process, we make use of the custom function **wavezero**, which zeroes wavelet transform detail coefficients with
[nc, g8] = wavezero(cm, sm, level, 'bior6.8');
where **level** indicates the level number of detail coefficients to be zeroed. The output **g8** is the inverse transform image ready for display.

Show me (a) the smoothed image **g8** and (b) the difference image between **img** and **g8** using different **level** values of 1, 2, 3, and 4, respectively. (20%)
Repeat the process but adding Gaussian noise to **img** with a zero mean and 0.15 variance value before the transform. (20%) What setting of **level** the filtered image has the best visual quality similar to the original image? Report its PSNR. (10%)

5. Progressive image reconstruction (40%)

Consider the transmission and reconstruction of the four-scale Antonini-Barlaud-Mathieu-Daubechies wavelet transform in the *barbara.tif* image with
[cm, sm] = wavefast(img, 4, 'jpeg9.7');
First, show the transform coefficients using (5%)
wavedisplay(cm, sm, 8);
Each image in the structure is stored as a multiscale wavelet decomposition, which is well suited to progressive reconstruction applications. The custom function **wavecopy** allows you to perform reconstruction by copying fetches coefficients of a wavelet decomposition structure using
yi = wavecopy('a', cm, sm);
If you show this image with **imshow(mat2gray(yi));**, you will see a low-resolution version corresponding to the top left image in the wavelet coefficient figure. A higher-resolution approximation can be constructed by taking one-scale inverse FWT back with
[cm, sm] = waveback(cm, sm, 'jpeg9.7', 1);
Now, you can display this image using again
yi = wavecopy('a', cm, sm);
imshow(mat2gray(yi));
You can repeat this process to show all reconstructed image until the final image that matches the original image resolution. (a) Display the five reconstructed images with different resolutions. (25%) (b) Show the difference image between the final reconstruction and the original image. Are these two images identical? (10%)