ELEC 431
Digital Signal Processing
Homework 14

Due Friday, February 18, 2003

This problem investigates the Haar wavelet transform and its application to image compression and denoising.

1. Develop a Matlab function for computing the J-level, $J = 1, 2, \ldots, \log_2 N$, Haar wavelet transform of an $N \times N$ image (assume N is a power of 2). Your function can repeatedly call the function `haar.m` that I wrote and discussed in class, or you can start from scratch. Your function should take two arguments: an input image and the number of levels J you wish to compute. Your function should output an array of $N \times N$ wavelet coefficients (in the arrangement discussed in class).

2. Develop a Matlab function for computing the inverse J-level, $J = 1, 2, \ldots, \log_2 N$, Haar wavelet transform of an $N \times N$ array of Haar wavelet coefficients. Your function could repeatedly call the function `ihaar.m` which reverses the process of `haar.m`, for example. Your function should take two arguments: an input array of Haar wavelet coefficients and the number of levels J in that array. Your function should output an image reconstructed from the input coefficients.

3. Test your forward and inverse Haar transform code by applying it to the `cameraguy` image. You can test it with other (grayscale) images if you want, too.

4. Image Compression: Compare the compression performance of the Haar wavelet transform with the Fourier transform using the test image(s). Compute the transforms, set all coefficients to zero except for the largest (in magnitude) 25%, 10%, 5%, and 1% and reconstruct an approximation to the original image by applying the corresponding inverse transform. This simulates the process of compressing by factors of 1/4, 1/10, 1/20, and 1/100. Display the resulting images and comment on the relative quality of wavelet vs. Fourier results.

5. Image Denoising: Compare the denoising performance of the Haar wavelet transform with the Fourier transform (`fft2`) using the test image(s). Add a small amount of Gaussian white noise (with variance σ^2) to each image, compute the transforms, set all coefficients to zero except those whose magnitude is larger than 3σ (you can also try factors other than 3), and reconstruct an estimate of the original image by applying the corresponding inverse transform. Note that we have defined the Haar wavelet transform to be orthonormal, so the noise variance in the wavelet coefficients is σ^2 (the same as in each pixel). However, Matlab's `fft2` is not normalized, and you should verify the proper normalization level (I think you should use a threshold of $3N\sigma$ to account for the lack of normalization). Display results and comment on the difference between the wavelet and Fourier methods.

6. Denoising vs. Wiener Filtering: Construct a `fft2` based Wiener filter and compare its performance to the Haar wavelet and `fft2` denoising schemes investigated above Display results and comment on the difference between the denoising and Wiener methods.