2007 - 2008
SGN-1656 Signal Processing Laboratory
Wavelets

September 21, 2007

Name :
Student No.:
Group No.:

1 Introduction

1. Explain what you understand from the word "transform". Why are transforms needed in signal and image processing applications? Can you give an example where transform domain analysis is preferred over time or space domain analysis?

2. Compare Fourier Transform, Short Time Fourier Transform and Wavelet Transform in terms of the information they provide about the signal.

   In matlab, generate the following signals with a sampling interval Ts = 1msec and of length 512 points;

   $$x_1(t) = sin(2 * pi * 20 * t) + sin(2 * pi * 35 * t) + sin(2 * pi * 50 * t)$$

   $$x_2(t) = \begin{cases} sin(2 * pi * 20 * t), & 0 \leq t < 170msec \\ sin(2 * pi * 35 * t), & 170msec \leq t < 340msec \\ sin(2 * pi * 50 * t), & 340msec \leq t < 512. \end{cases}$$

   Calculate and plot the Fourier Transform (type "help fft"), Short Time Fourier Transform (type "help spectrogram" or "help specgram") and Continuous Wavelet Transform (type "help cwt") of both signals. Comment on the results. Plot the STFT and CWT results.

3. What do "scaling" and "shifting" correspond to in the wavelet analysis concept? Explain the relations between scaling-shifting and time-frequency.

4. Explain Discrete Wavelet Transform (and its inverse) in terms of filter banks. Use figures.

   Explain the terms "orthogonality" and "biorthogonality" in this context.

5. Check helps of "MakeONFilter", "MirrorFilt", "reverse", "UpSample", "conv", "MakeWavelet". Generate and plot Daubechies 6 wavelet function using "MakeWavelet". Try to obtain the same function in the following way;

   Generate the low pass filter coefficients of Daubechies 6 wavelet. Use the low pass filter coefficients to obtain highpass filter coefficients. Upsample highpass filter coefficients by a factor of 2 and convolve with low pass filter. Upsample the results and convolve with low pass filter successively 4 times. Plot the resulting function after each convolution. Does it approximate the wavelet function?

   Repeat the same process by upsampling lowpass filter coefficients instead of highpass filter coefficients. What does the resulting function approximate now?

6. Include the plots in your report.

2 Applications

1. **Reconstruction from low frequency and high frequency coefficients.** Generate signals 'HeaviSine' and 'TwoChirp' of length 256. (check "MakeSignal"). Compute and plot 3 level DWT (i.e. $L = \log_2(256) - 3$) of both signals using Daubechies 4. (check "FWT_PO"). Suppress the first 32 coefficients to zero and take the inverse DWT. How well are different parts of two signals reconstructed from 224 coefficients? Suppress the last 128 coefficients of both signals to zero and take the inverse DWT. Comment on the results.

2. **Wavelet De-Noising.** Explain Wavelet De-Noising, Hard Thresholding and Soft Thresholding.

   Write a matlab code which

   - Takes a 1-D signal, a threshold value T, threshold type (Hard or Soft thresholding), a wavelet filter and its parameters qmf and par, and a noise variance σ^2 as inputs
   - Adds random generated Gaussian noise with variance σ^2 to the input signal
   - Performs wavelet denoising on the signal-noise mixture (Apply thresholding only on the detail coefficients but not on the coarse approximation coefficients).
   - Outputs the noisy and denoised signals, root mean square error (RMSE) and mean absolute error (MAE)
   - Plots the input signal, noisy and denoised signals on the same figure.

   Use WaveLab and matlab functions "randn","HardThresh","SoftThresh", "MakeONFilter","FWT_PO","IWT_PO",etc.

   Generate three different types of signals, 'Blocks', 'Bumps', and 'HeaviSine', of length $n = 256$. Use the above code to experiment wavelet denoising with Haar and Daubechies 4 filters. Try both hard and soft thresholding with different thresholds $T = T_o = \sqrt{2\log(n)}\sigma,\ T < T_o,\ T > T_o$. Take $\sigma = 0.4$. Comment on the results.

   Try translation invariant wavelet transform (check "FWT_TI") instead of FWT_PO for the same signals with $T = T_o$, and with hard thresholding only. Do the results improve? Why, why not?

   You do not need to include all of the plots in your report. Include only plots of original, noisy, and denoised 'Blocks' signals by soft thresholding, by hard thresholding, and by translation invariant denoising, with $T = T_o$.

3. **Compression.** Give examples of image compression distortions in wavelet image compression. Write a matlab code which

   - Gets the transformation type (orthogonal wavelet FWT2_PO or biorthogonal wavelet transform FWT2_PBS), wavelet filter parameters, filename, and a quantization step size Δ as inputs

- reads an $N \times N$ size image in 8-bit raw binary format into an $N \times N$ matrix of integers from the file specified by filename
- Takes the four scale WT ($L = \log_2 N - 4$) of the input image, quantizes the transform coefficients by a uniform scalar quantizer
- Calculates the entropy of the input image, non-quantized and quantized transform coefficients
- Performs the IWT of non-quantized and quantized wavelet coefficients
- Returns MSE and MAE between the input and IWTransformed image as outputs.

Use the following quantizer

$$out(i,j) = \lfloor \frac{\mid in(i,j) \mid}{\Delta} + 0.5 \rfloor . \Delta . sgn(in(i,j)). \tag{1}$$

Take the input file as "~agotchev/IMAGES/lena512.bmp" which is of size 512×512. Use Symmlet 4, Symmlet 10 (orthogonal) filters with $\Delta = 39$, and biorthogonal spline filter with $par = [3\ 3]$ and $\Delta = 41$, respectively in the code you have written. Tabulate the MSE, MAE, and entropy results, where entropy is defined as

$$H(\mathbf{X}) = -\sum_i p_i \log_2 p_i,$$

$$p_i = \frac{number\ of\ x \in \mathbf{X}\ such\ that\ x = i}{N^2}$$

for a sequence $\mathbf{X}$ of size N^2.

(You can use an entropy calculation code if you have, or write it in any language.)

Zoom into different parts of reconstructed images. Which filter provide better visual reconstruction? Why?

Include all the codes you used in your report.