Information Extraction and Data Compression Using Wavelets

Tara Murphy

February 4, 2002

1 Introduction

These notes provide a brief outline of wavelet analysis.

2 The Fourier Transform

In Fourier analysis, $\sin(kx)$ and $\cos(kx)$ form a basis from which all other functions can be built. So a function $f(x)$ can be represented as:

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) \tag{1}$$

where the a_k and b_k are the Fourier coefficients. These coefficients are calculated by:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx \tag{2}$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx \tag{3}$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx \tag{4}$$

The basis functions ($\sin(kx)$ and $\cos(kx)$) are localized in frequency and hence the Fourier transform gives you information about the frequency content of a function. For example, Figure 1 shows a simple sin curve and its corresponding Fourier transform, a single delta function. However, because the basis functions are infinite in the time domain, there are limitations to

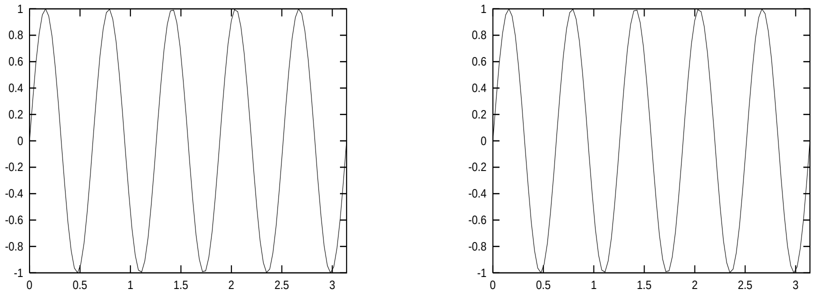

Figure 1:

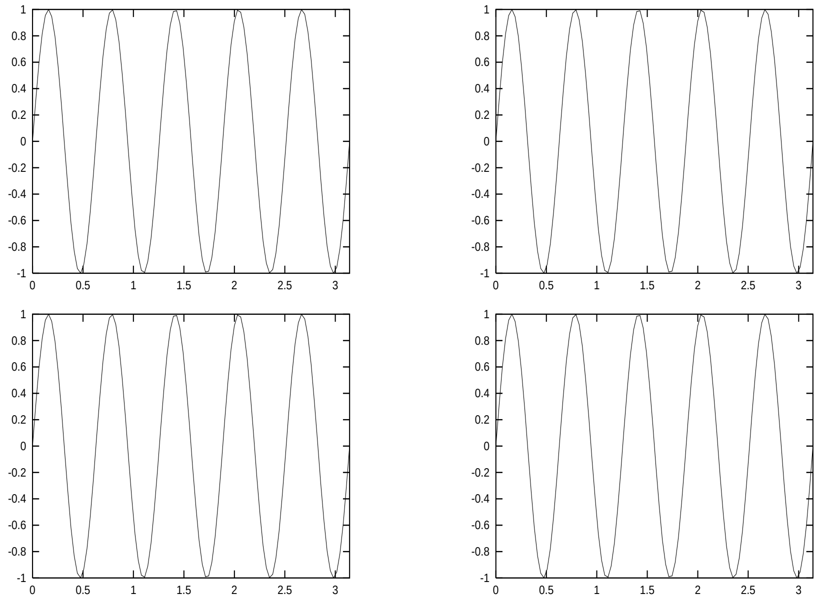

Figure 2:

how much information you can retrieve. For example, Figure 2 shows two different signals in the time domain. The first signal consists of a sin curve initially generated at frequency f_1 and then at some time t later changed to f_2. The Fourier transform shows two delta functions corresponding to each section of the signal. The second signal in the sum of two sin curves, generated similtaneously for the entire time. The Fourier transform of this signal is the same as for the first. This example shows that information about the locality of a function in the time domain is lost with the Fourier transform.

Another problem is that many signals contain discontinuities, which are hard to represent by the $\sin(kx)$ and $\cos(kx)$ functions. For example, Figure 3 shows a step signal and its Fourier transform. To overcome these problems, you have to move from *frequency* analysis to *scale*

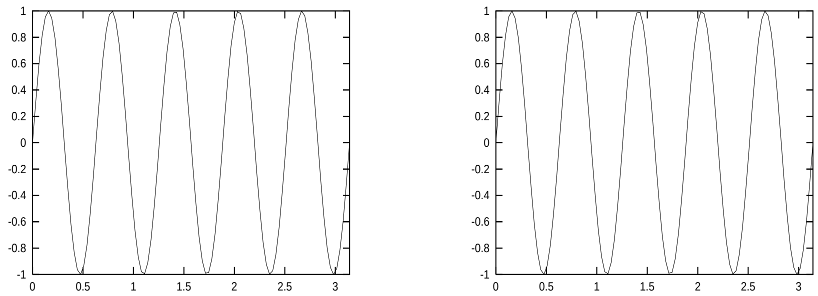

Figure 3:

analysis.

3 The Wavelet Transform

In place of $\sin(kx)$ and $\cos(kx)$, the wavelet transform uses *scale-vaying* basis functions called wavelets, $\psi_{jk}(x)$. So a function $f(x)$ can be represented as:

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{jk}\psi_{jk}(x) \tag{5}$$

where d_{jk} are the wavelet coefficients. Each basis function $\psi_{jk}(x)$ is obtained from one *mother function* $\psi(x)$ by shifting and scaling in the following way:

$$\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k) \tag{6}$$

where j is the scaling parameter and k is the shifting parameter. Figure 4 demonstrates the process of creating a set of wavelets from the mother function. Here, the mother wavelet is the

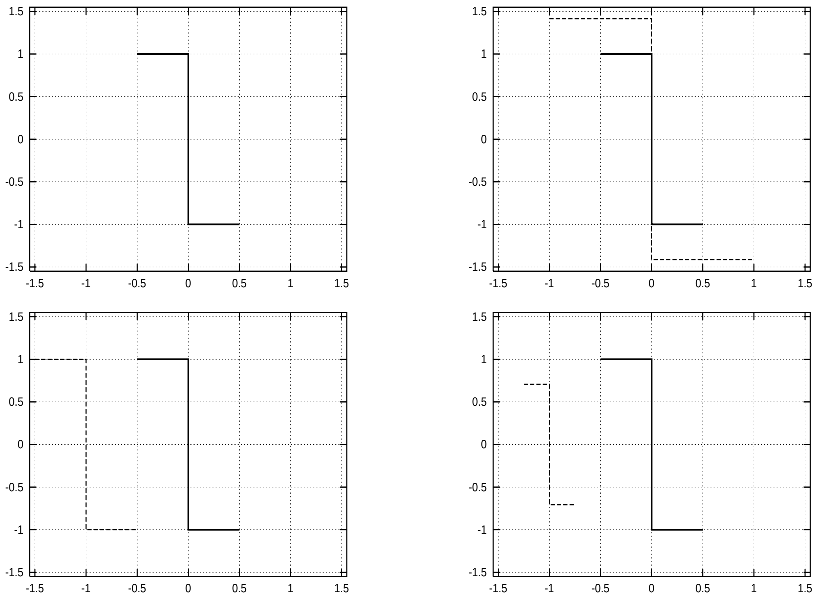

Figure 4: The mother wavelet (top left) can be scaled (top right), shifted (bottom left) or both (bottom right) to create new wavelets

simplest example of an orthogonal wavelet, the Haar function $\psi_H(x)$:

$$\psi_H(x) = \begin{cases} 1 & : \; 0 \leq x < \frac{1}{2} \\ -1 & : \; \frac{1}{2} \leq x < 1 \\ 0 & : \; \text{otherwise} \end{cases} \tag{7}$$

New wavelets can be creating by scaling:

$$\begin{aligned} \psi_{10}(x) &= 2^{1/2}\psi(2^1 x - 0) \\ &= \sqrt{2}\psi(2x) \end{aligned}$$

by shifting:

$$\begin{aligned} \psi_{01}(x) &= 2^0\psi(2^0 x - 1) \\ &= \psi(x - 1) \end{aligned}$$

or by a combination of both:

$$\begin{aligned}\psi_{-11}(x) &= 2^{-1/2}\psi(2^{-1}x-1)\\ &= \frac{1}{\sqrt{2}}\psi(\frac{x}{2}-1)\end{aligned}$$

In this way, you can generate a set of basis functions which include both short wavelets (useful for dealing with noise or discontinuities in the signal) and long wavelets (useful for frequency analysis).

Deriving a particular mother function, $\psi(x)$, involves searching for a function that satisfies certain conditions such as having continuous derivatives and decaying at infinity.

4 The Daubechies Mother Function

In this section we will derive the coefficients for simplest member of the Daubechies family of wavelets D^4.

The simple step function we used in Section 3 has several limitationns. One of them is bad analytic behaviour with the abrupt changes at interval boundries. Another is that it does not have good time-frequency localization.

RESULT

$$h_0 = \frac{1}{4\sqrt{2}}(1+\sqrt{3}) \tag{8}$$

$$h_1 = \frac{1}{4\sqrt{2}}(3+\sqrt{3}) \tag{9}$$

$$h_3 = \frac{1}{4\sqrt{2}}(3-\sqrt{3}) \tag{10}$$

$$h_4 = \frac{1}{4\sqrt{2}}(1-\sqrt{3}) \tag{11}$$

Figure 5 shows the D^4 mother function, and well as some higher order members of the Daubechies family.

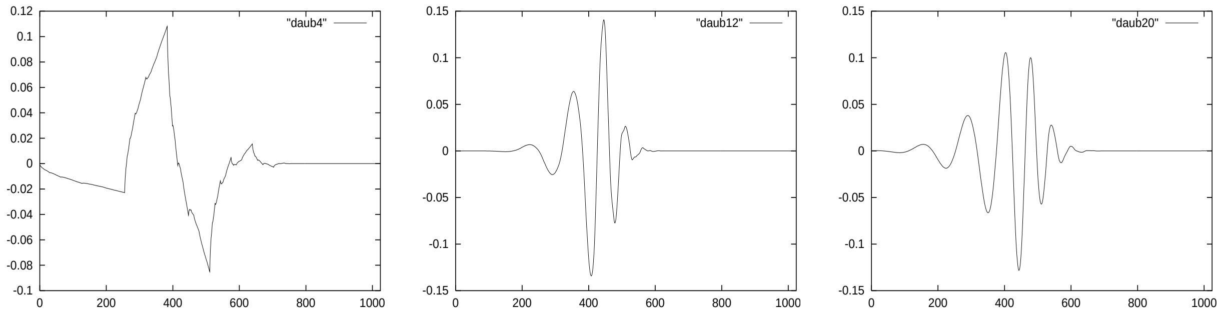

Figure 5: Three examples of wavelet mother functions from the Daubechies family. These were created by inverse transforming a delta function. Left: Daubechies 4. Centre: Daubechies 12. Right: Daubechies 20.

5 The Discrete Wavelet Transform

This section explains the DWT and the pyramid algorithm from NumRec

The *wavelet coefficients* d_{ij} given by Equation **??** are actually linear superpositions of the *fundamental coefficients* that define the wavelets (this will be demonstrated in Equation 14. It is important to make the distinction between these two sets of coefficients and so in this section I will refer to each explicitly. However, in the rest of this thesis, the term *coefficients* will will mean d_{ij} — the result of doing a wavelet transform.

A set of wavelets can be specified by a matrix of its fundamental coefficients. For example, the most compact Daubechies wavelet is of order 4 and so has four fundamental coefficients, c_0, c_1, c_2, c_3 which form a matrix:

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & & & & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & & & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & & & & \\ \vdots & \vdots & & & & & \ddots & & & & \\ & & & & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & & & & & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & & & & & c_3 & -c_2 \end{bmatrix} \tag{12}$$

By using the fact that the above matrix is orthogonal, and by requiring that the sequence $c_3, -c_2, c_1, -c_0$ has a certain number of vanishing moments, the values of the fundamental coefficients can be calculated. For the Daubechies 4 wavelets, these have the values

$$\begin{array}{ll} c_0 = (1+\sqrt{3})/4\sqrt{2} & c_1 = (3+\sqrt{3})/4\sqrt{2} \\ c_2 = (3-\sqrt{3})/4\sqrt{2} & c_3 = (1-\sqrt{3})/4\sqrt{2} \end{array} \tag{13}$$

Values for different wavelets can be found in tables, for example in (**?**). The matrix is actually comprised of two filters (quadrature mirror filters) which perform different operations when applied to the data. The output of the first filter (c_0, c_1, c_2, c_3) is the smooth information about the data. The output of the second filter $(c_3, -c_2, c_1, -c_0)$ is the detail information about the data. So one application of matrix 12, to a data vector of length 8 gives a vector of interleaved smooth (s) and detailed (d) components:

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & \\ & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & & c_3 & -c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \tag{14}$$

However this has only operated on one resolution level — extracting the 2^{N-1} wavelet coefficients that represent the finest structure. To access the next resolution level the matrix must be reapplied to the smoothed data. This is done hierarchically using the pyramid algorithm, separating the detail coefficients at each stage (by rearranging the vector), and reapplying the matrix to the remaining smooth coefficients. In this way, the number of smooth coefficients is reduced by a factor of two each time, until two final smooth coefficients are left, the mother-function coefficients. Hence the resulting vector consists of 2 residual smooth coefficients and $2^N - 2$ wavelet coefficients, d_{ij}, and is the same length as the input data vector. In the case of the above example, only one more application is needed. In this work, the DWT is computed using the pyramid algorithm as implemented in Numerical Recipes and shown in Equation 15:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \begin{matrix} \text{apply 8x8 matrix} \\ \longrightarrow \end{matrix} \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \begin{matrix} \text{rearrange} \\ \longrightarrow \end{matrix} \left[\begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ \hline d_1 \\ d_2 \\ d_3 \\ d_4 \end{array}\right] \begin{matrix} \text{apply 4x4 matrix} \\ \longrightarrow \end{matrix} \left[\begin{array}{c} S_1 \\ D_1 \\ S_2 \\ D_2 \\ \hline d_1 \\ d_2 \\ d_3 \\ d_4 \end{array}\right] \begin{matrix} \text{rearrange} \\ \longrightarrow \end{matrix} \left[\begin{array}{c} S_1 \\ S_2 \\ \hline D_1 \\ D_2 \\ \hline d_1 \\ d_2 \\ d_3 \\ d_4 \end{array}\right] \tag{15}$$

To help visualise this, Figure 6 shows an example of applying the DWT to a sine function and a slightly modified sine function. Comparing the two show how the wavelet transform probes structure on different scales. In both there are high amplitude wavelet coefficients at low wavelet numbers. These are caused by the sine curve itself. In the pure sine curve there is no structure at finer levels than this, and hence all the high wavelet numbers are zero. However, the transform of the modified curve shows some lower amplitude wavelet coefficients representing the fine scale structure — in this case caused by the spike. It should be emphasised that the wavelet coefficients not only detect what frequencies are present, they also record the position of the various features at each resolution scale.

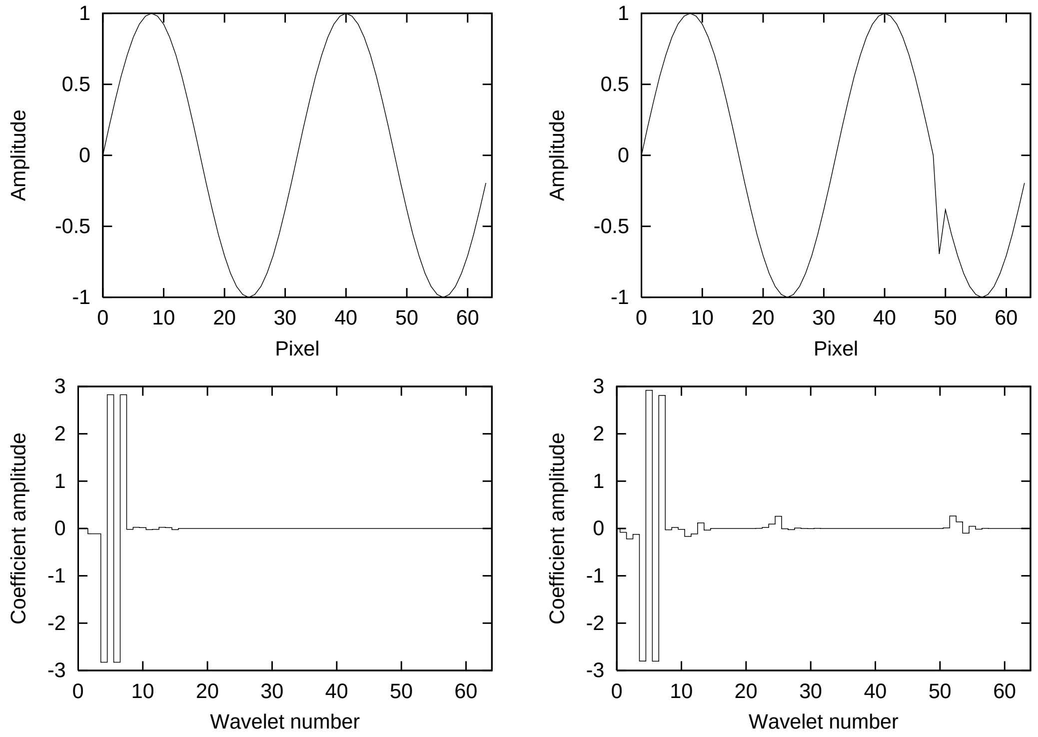

Figure 6: Left: A 64 data point sine curve, and its wavelet transform. Right: The same sine curve, modified by adding a spike, and its wavelet transform.

6 Data Compression

This section explains how wavelets can be used for data compression

7 Information Extraction

This section explains what automatic information extraction in necessary, and how wavelets can be used. Eg. denoising

8 Useful Links and References

This webpage is a good links page for wavelet tutorials, sorfware and applets

`http://www.amara.com/current/wavelet.html`