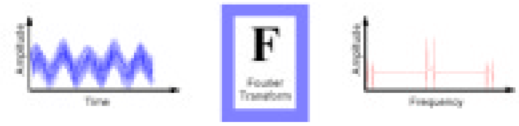
Wavelets: a preview

February 6, 2003

Acknowledgements:
Material compiled from the MATLAB Wavelet Toolbox UG.

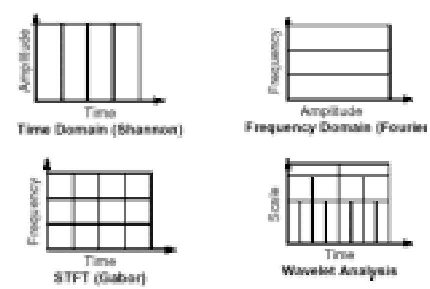
Problem with Fourier…



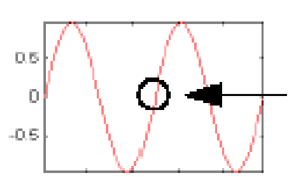
Fourier analysis -- breaks down a signal into constituent sinusoids of different frequencies.
a serious drawback In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell *when* a particular event took place.

Gabor's proposal

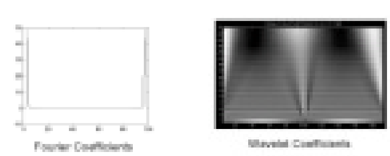
Fourier – Gabor – Wavelet



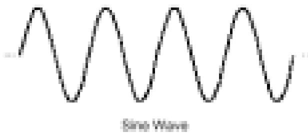
Sinusoid with a small discontinuity



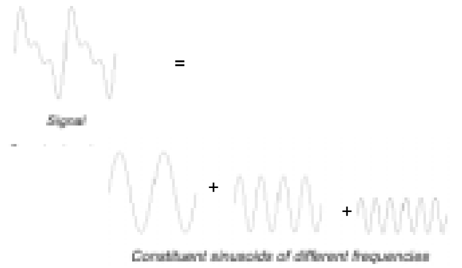
in the transform domain



Localization (or the lack of it)



Fourier decomposition



and the Wavelet decomposition

Fourier transform: $F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$

Similarly, the *continuous wavelet transform* (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function ψ:

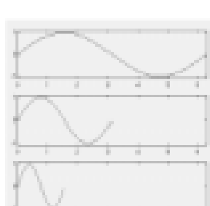
$$c(\text{scale, position}) = \int_{-\infty}^{\infty} f(t)\,\psi(\text{scale, position}, t)\,dt$$

The result of the CWT are many *wavelet coefficients* C, which are a function of scale and position.

Wavelet decomposition –contd.

Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal:

What do we mean by scale?

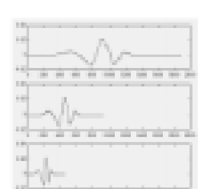


$f(t) = \sin(t) \quad ; \quad a = 1$

$f(t) = \sin(2t) \quad ; \quad a = \frac{1}{2}$

$f(t) = \sin(4t) \quad ; \quad a = \frac{1}{4}$

The scale factor

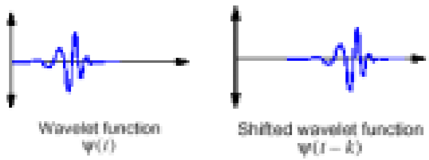


$f(t) = \psi(t) \quad ; \quad a = 1$

$f(t) = \psi(2t) \quad ; \quad a = \frac{1}{2}$

$f(t) = \psi(4t) \quad ; \quad a = \frac{1}{4}$

Shifting

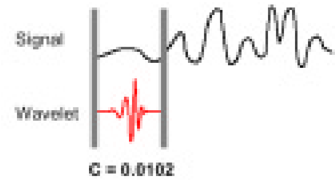


Computing a wavelet transform

It's really a very simple process. In fact, here are the five steps of an easy recipe for creating a CWT:

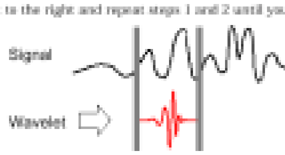
1 Take a wavelet and compare it to a section at the start of the original signal.

2 Calculate a number, C, that represents how closely correlated the wavelet is with this section of the signal. The higher C is, the more the similarity. Note that the results will depend on the shape of the wavelet you choose.

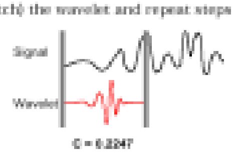


Computing the WT (2)

3 Shift the wavelet to the right and repeat steps 1 and 2 until you've covered the whole signal.

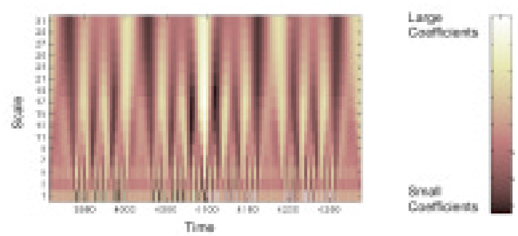


4 Scale (stretch) the wavelet and repeat steps 1 through 3.



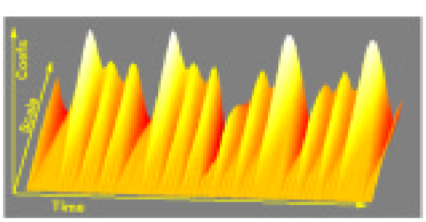
5 Repeat steps 1 through 4 for all scales.

Visualizing the WT coefficients



..in 3-D

These coefficient plots resemble a bumpy surface viewed from above. If you could look at the same surface from the side, you might see something like this:



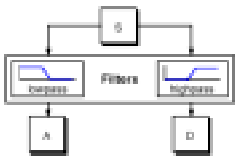
The discrete wavelet transform

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. What if we choose only a subset of scales and positions at which to make our calculations?

It turns out, rather remarkably, that if we choose scales and positions based on powers of two — so-called *dyadic* scales and positions — then our analysis will be much more efficient and just as accurate. We obtain just such an analysis from the *discrete wavelet transform* (DWT).

Approximations and Details

The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components. The filtering process, at its most basic level, looks like this:



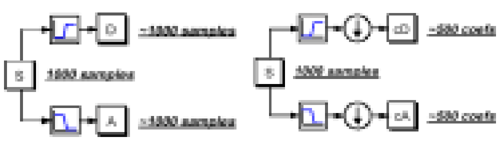
The original signal, S, passes through two complementary filters and emerges as two signals.

Downsampling

Unfortunately, if we actually perform this operation on a real digital signal, we wind up with twice as much data as we started with. Suppose, for instance, that the original signal S consists of 1000 samples of data. Then the approximation and the detail will each have 1000 samples, for a total of 2000.

To correct this problem, we introduce the notion of *downsampling*. This simply means throwing away every second data point. While doing this introduces *aliasing* in the signal components, it turns out we can account for this later on in the process.

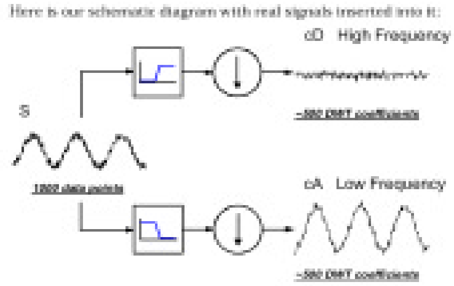
Downsampling (2)



The process on the right, which includes downsampling, produces DWT coefficients.

An example

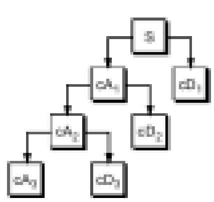
Here is our schematic diagram with real signals inserted into it:



Wavelet Decomposition

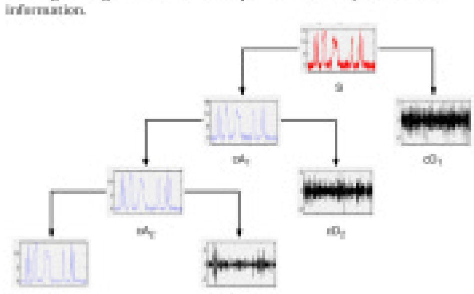
Multiple-Level Decomposition

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower-resolution components. This is called the *wavelet decomposition tree*.

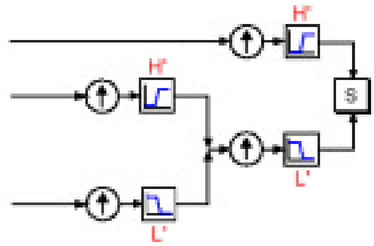


Wavelet decomposition…

Looking at a signal's wavelet decomposition tree can yield valuable information.

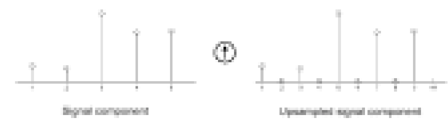


IDWT: reconstruction



Analysis vs Synthesis

Where wavelet analysis involves filtering and downsampling, the wavelet reconstruction process consists of upsampling and filtering. Upsampling is the process of lengthening a signal component by inserting zeros between samples:



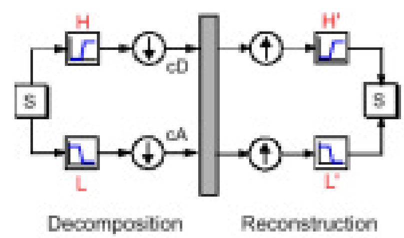
Perfect reconstruction

The filtering part of the reconstruction process also bears some discussion, because it is the choice of filters that is crucial in achieving perfect reconstruction of the original signal.

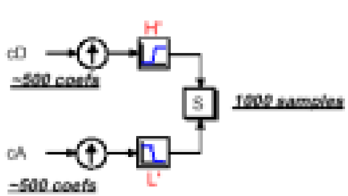
That perfect reconstruction is even possible is noteworthy. Recall that the downsampling of the signal components performed during the decomposition phase introduces a distortion called aliasing. It turns out that by carefully choosing filters for the decomposition and reconstruction phases that are closely related (but not identical), we can "cancel out" the effects of aliasing. This was the breakthrough made possible by the work of Ingrid Daubechies.

A technical discussion of how to design these filters can be found in p. 347 of the book *Wavelets and Filter Banks*, by Strang and Nguyen. The low- and highpass decomposition filters (L and H), together with their associated reconstruction filters (L' and H'), form a system of what are called *quadrature mirror filters*.

Quadrature Mirror Filters

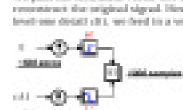


Reconstructing Approximation & Details



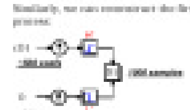
It is also possible to reconstruct the approximations and details themselves from their coefficient vectors. As an example, let's consider how we would reconstruct the first-level approximation A1 from the coefficient vector cA1.

We pass the coefficient vector cA1 through the same process we used to reconstruct the original signal. However, instead of combining it with the level-one detail cD1, we feed in a vector of zeros in place of the details:



The process yields a reconstructed approximation A1, which has the same length as the original signal S and which is a real approximation of it.

Similarly, we can reconstruct the first-level detail D1, using the analogous process:



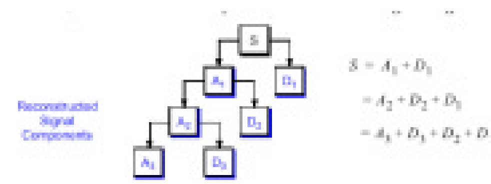
The reconstructed details and approximations are true constituents of the original signal. In fact, we find when we combine them that:

$A_1 + D_1 = S$

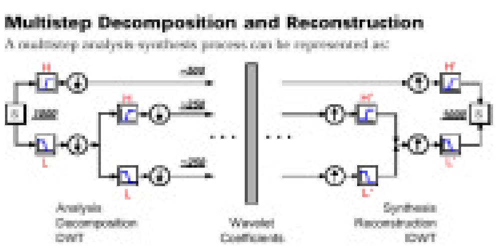
Reconstructing As and Ds..contd..

Note that the coefficient vectors cA1 and cD1 — because they were produced by downsampling, contain aliasing distortion, and are only half the length of the original signal — cannot directly be combined to reproduce the signal. *It is necessary to reconstruct the approximations and details before combining them.*

Reconstructing the signal



Multiscale Analysis



Wavelet Toolbox

- See the wavelet demo
- Wavemenu – compression demo.