A Wavelet-based Approach to Pitch Detection

Ross Miller

Wavelets are a tool to extract frequency out of a signal. As opposed to the FFT, which decomposes a signal into sine and cosine waves, the wavelet transforms decompose a signal into wavelets, which are brief oscillations that are localized in time. This localization makes it possible for both temporal and frequency information to be preserved in the decomposed signal at the same time.

Properties

There are many different types of wavelet families, each comprised of a mother wavelet and infinitely many daughter wavelets. Each wavelet in these families adheres to a specific set of properties. Among these, all wavelets must have finite energy, and an average value of zero across the entire time domain. Additionally for each wavelet family, the mother wavelet, $\psi(t)$, is defined such that

$$\int_{-\infty}^{\infty} |\psi(t)|^2 \; dt = 1$$

A daughter wavelet is generated from the mother wavelet in its same family by applying a dilation, a, in the x-direction and a translation in the time domain, b. Daughter wavelets are given by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \; \psi(\frac{t-b}{a})$$

Wavelet Transform

In order to make use of wavelets for frequency detection, they need to be convolved with an input signal in order to produce a response.

Each daughter wavelet has a distinct (a,b) pair. When the input signal is convolved with each daughter wavelet, the largest responses will be for those wavelets that most closely match the frequency and offset of the underlying signal.

For a signal x(t), the response when that daughter wavelet is convolved with the input signal is given by

$$X_\psi(a,b) = \int_{-\infty}^{\infty} x(t) * \overline{\psi_{a,b}(t)} \; dt$$

where $*$ is the convolution operator, and each $X_\psi(a,b)$ is called a detail coefficient.

Continuous Wavelet Transform

For the continuous wavelet transform (CWT), the value of $X_\psi(a,b)$ is determined for every value of a and b within the time and frequency domain.

For the CWT, the coefficient is determined at every point in time and at every scale corresponding to a frequency within the target bandwidth.

Once you have found these detail coefficients, they give you all the information you need in order to reconstruct the signal. Each coefficient corresponds to a daughter wavelet with a distinct frequency and offset in time pair. The value of each coefficient tells you what the scaling factor in the y-direction should be. In other words, the detail coefficients are a full decomposition of the original signal into a linear combination of wavelets. The detail coefficients tell you how much of each daughter wavelet is present in the original signal.

Applications of Wavelet Transforms

This insight has important implications in denoising and lossy compression algorithms. Any n-dimensional signal can be decomposed with the discrete wavelet transform (DWT) to produce these detail coefficients. If this signal has little to no random noise (for example, a pure 10Hz sine wave), this transform will result in many coefficients with a value of zero. By using an efficient sparse matrix storing technique, you can avoid storing all of these 0 values, and therefore, you can achieve lossless compression of the input signal.

It is important to note that not all wavelet families can be used to perfectly reconstruct a signal. If the Haar wavelet is used as the mother wavelet, then for a 10Hz sine wave that is transformed and then inverse transformed, some artifacts will be introduced. This can be avoided by choosing a mother wavelet that has the ability to perfectly reconstruct a signal. One wavelet with this property is Daubechies wavelet.

Increased compression ratios can be achieved by not storing in your sparse matrix those coefficients whose absolute values are below some small, but greater than 0, tolerance level. This has the effect of treating these coefficients as zeroes. The number of coefficients to drop is simply based upon the compression ratio. When increased compression is needed, the smallest coefficients should always be dropped first, since they have the least impact, visually, on the reconstructed signal.

DWT is used in the JPEG 2000 standard to compress an image. A JPEG image can be treated as three individual two-dimensional signals, corresponding to the red, blue, and green values at each point in the image. To compress an image by 50%, that has previously been encoded using DWT, you simply need to perform DWT on each of these three input signals, and then set the lowest half of the coefficients from each signal to zero.

Alternatively, you could drop the lowest half of the coefficients across all three of the signals. However, if the image has a lot of large blue and green coefficients, yet has very small red coefficients, dropping the lowest global half of the coefficients could result in an inordinate amount of red coefficients lost. Therefore, the reconstructed signal would appear color-shifted so that it looks more cyan than the original signal (since cyan is the opposite of red). By dealing with each of the color signals independently when dropping coefficients, you could still experience color shifting, but it is much more likely to be insignificant.

As the compression ratio for JPEG 2000 images grows, the algorithm begins to discard larger and larger coefficients. Intuitively, this means that there is much less of a perceptible difference when going from 0 - 1% compression, than there is when going from 90 - 91% compression. Indeed, one of the key advantages of JPEG 2000, aside from progressive downloading, is that it performs remarkably well for low compression ratios.

Denoising

The above technique, used to compress images, can also be used to denoise a signal. The random noise portions of a signal will not produce a large coefficient when convolved with any of the daughter wavelets. However, the noise in the input signal may present itself in the detail coefficients as a small and unexpected response for some daughter wavelets (Figure 4). In other words, none of the daughter wavelets will resonate strongly with any of this noise, due to its randomness. Therefore, the wavelet transforms are resilient to noise. Even in the presence a lot of noise, wavelet transforms can still detect the underlying signal.

Instead of dropping some percentage of coefficients, which is useful when using a specified compression ratio, it is wiser to set some threshold and drop all coefficients below that threshold when doing denoising. This is because if you just drop a percentage of coefficients, the denoising algorithm will perform very differently on a signal with a large signal to noise ratio compared to one with a small signal to noise ratio. In the former case, you risk losing some of the detail coefficients corresponding to an underlying signal even though they are relatively large.

In practice, this threshold should be related to the supremum of the detail coefficients, for a given wavelet scale. This is because the total area underneath each daughter wavelet depends on its dilation factor. Therefore, the supremum of the detail coefficients corresponding to a 5Hz sine wave will be larger than that of a 10Hz sine wave with the same amplitude.

Scaleograms

A scaleogram is a three-dimensional plot of offset in time (b), scale (a), and the absolute value of the detail coefficients, $X\psi(a,b)$, for a wavelet convolved with a signal.

Each mother wavelet must have a total area under the curve equal to one, however there is no restriction on its corresponding frequency. Therefore, the relationship between scale and frequency depends on the choice of wavelet. As the dilation factor of the daughter wavelet increases, so does its wavelength. So, the frequency of a wavelet is inversely related to its scale.

Within the scaleogram, the largest response areas are where the scale and translation make it so the daughter wavelet most closely matches up with the signal. For a pure sinusoidal signal, the coefficients for the target scale vary back and forth between the max response within the scaleogram, and zero (Figure 1). When the dilation factor and translation component of a daughter wavelet most closely match the signal, there will be a maximum response. You will also get a maximum negative response with a daughter wavelet that has been phase shifted half of a wavelength. This wavelength is determined by the characteristic frequency of the respective daughter wavelet. If this daughter wavelet is instead phase shifted half of a wavelength, then the response corresponding to that new daughter wavelet will be zero.

Pitch Detection

I used a very simple algorithm to extract pitch and offset in time from a scaleogram. The scaleogram was constructed by using CWT. Since a proper continuous wavelet transform requires infinite convolution operations, I only sampled the detail coefficients at integer values of scale and for 1000 points across the time domain.

I began by breaking the time domain up into m equally sized intervals. For each interval, I calculated the supremum of the responses, and drew a horizontal line at the corresponding scale, that spans the entire time subinterval. I then used the WaveLab routine scal2freq() to map each scale found to a frequency. I used a hardcoded threshold, so that if the supremum was not larger than that threshold, then the algorithm considers there to be no signal within that interval.

This discretization of the frequency domain introduces some inherent error. This is evidenced in Figure 1, where even though the underlying signal is a 10 Hz sine wave, the pitch detector picks up a frequency of 10.4167 Hz. With more divisions in the frequency domain, the pitch detector would have come closer to the actual solution.

Another type of error introduced is shown in Figure 2. At t = 0.4sec, there is a band of nonzero coefficients starting at the bottom of the plot and moving up and to the right. Despite the nonzero coefficients between 0.4 and 0.6 the frequency within this area is actually zero. This is an illustration of how wavelets are unable to identify instantaneous frequency, but instead have to sample frequency in between some interval. In practice, when convolving wavelets, instead of finding a quadrature from negative infinity to infinity, the right and left ends of the wavelets are cut off where their values are below some trivial values. The resulting wavelet has some length in the x direction Different wavelets have different lengths, and the larger the length of a wavelet, the greater the localization errors that are incurred.

Some wavelets, such as the first Hermitian wavelet, make a single sign change and then converge to 0 on either side of the wavelet. Other wavelets, such as the Morlet wavelet, oscillate around the x-axis and converge to 0 much slower. Figures 2 shows the detail coefficients corresponding with the first Hermitian wavelet, which is simply the normalized first derivative of a particular Gaussian.

Figure 3 shows the same signal with the Morlet wavelet used instead. The Morlet wavelet is constructed by taking a cosine wave and applying a Gaussian over it. In the scaleogram in figure 2, there are distinct bright areas that appear to warp and wrap around each other at the points in time at which the frequency changes. This is most evident at t = 0.2 and 0.4 seconds. This effect is even more pronounced in the scaleogram in figure 3, and it is a direct result of how the Morlet wavelet is less localized than the 1st Hermitian wavelet.

In real applications, DWT would be used instead of CWT. This is because DWT can be done in O(n) time, where n is the number of samples in the input signal. As can be observed in the CWT spectrograms, there is a huge amount of redundancy in the detail coefficients, and so DWT takes advantage of this by being much more selective in the coefficients that it samples. Despite this, these detail coefficients are enough to reconstruct a signal within some known error bound, provided an appropriate wavelet family was used in the transform and inverse transform.

Despite what the name “discrete wavelet transform” may suggest, in order to perfectly reconstruct a signal you would need an unlimited number of detail coefficients. This is because DWT uses a system of dyadic sampling with multiple levels within the scaleogram. Each level corresponds to some subinterval of the bandwidth that will be sampled from. Detail coefficients are only sampled within each of these levels, and within each level, enough samples are used to cover the entire time domain. To get from one level to the next, you have to halve the difference between the current scale’s corresponding frequency and the upper bound of the frequency band that you are sampling from.

The characteristic frequency of a daughter wavelet and its effective width are inversely related. Therefore each new level requires twice the number of detail coefficients than the one before it. To avoid infinite computation, after a certain level number, all of the following levels will be consolidated into one single and final level, called the cork. The cork is used to sample the detail coefficients within the remaining frequency band, once it has been determined that the DWT has been calculated with a sufficient number of levels.

Appendix

Figure 1 – 10Hz sine wave

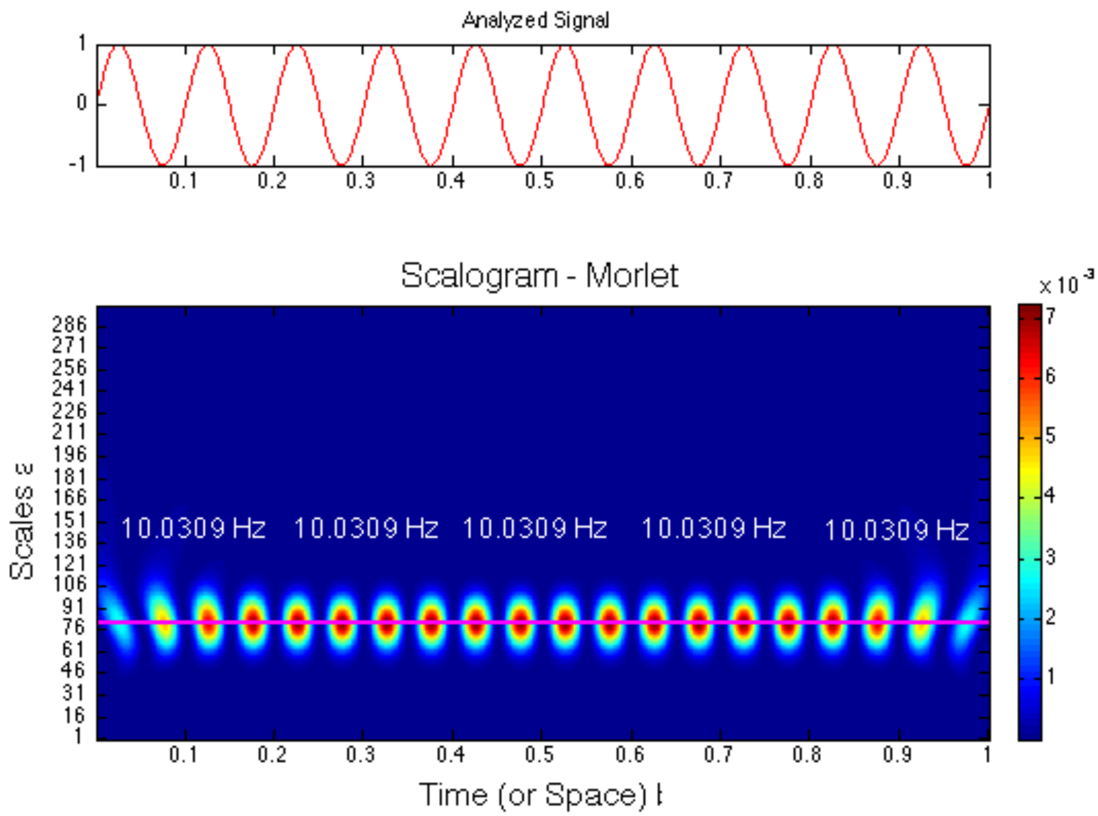


Figure 2 – [10 20 0 40 50] are the frequencies within each of the 5 equally spaced time intervals

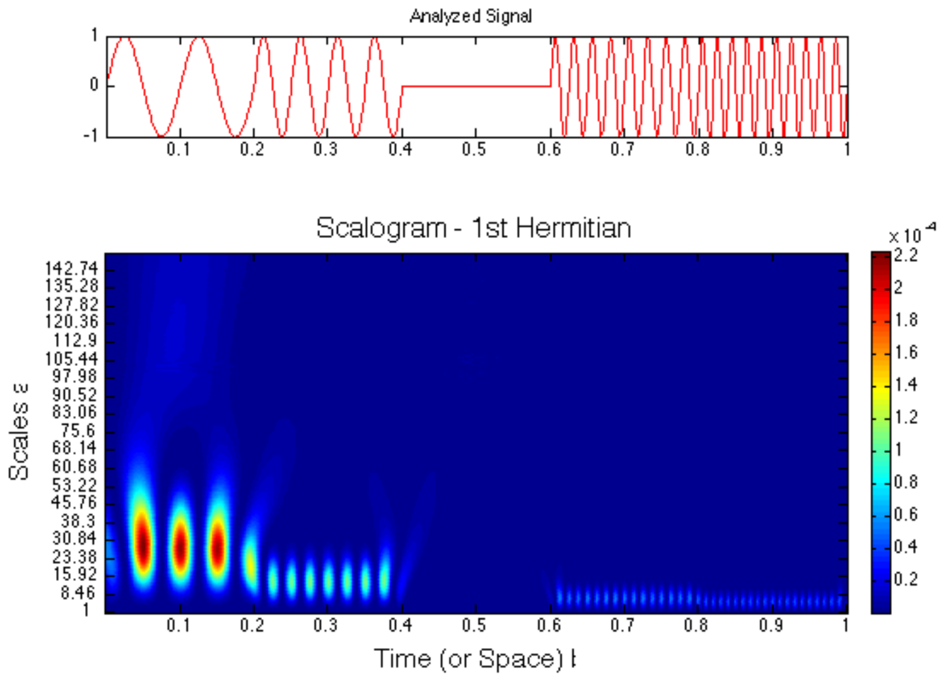


Figure 3 – same signal as in figure 2

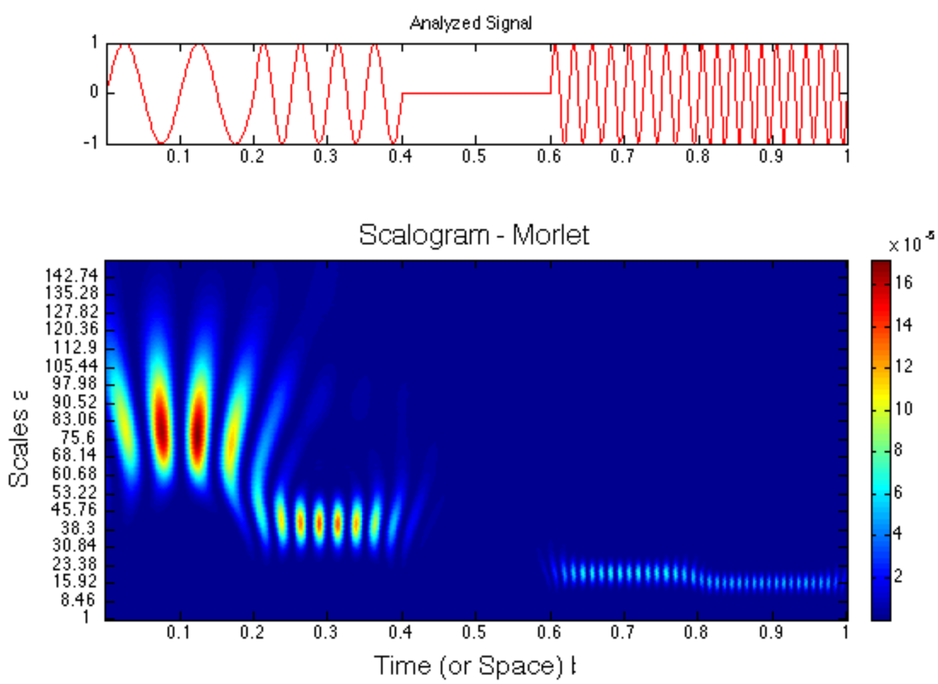
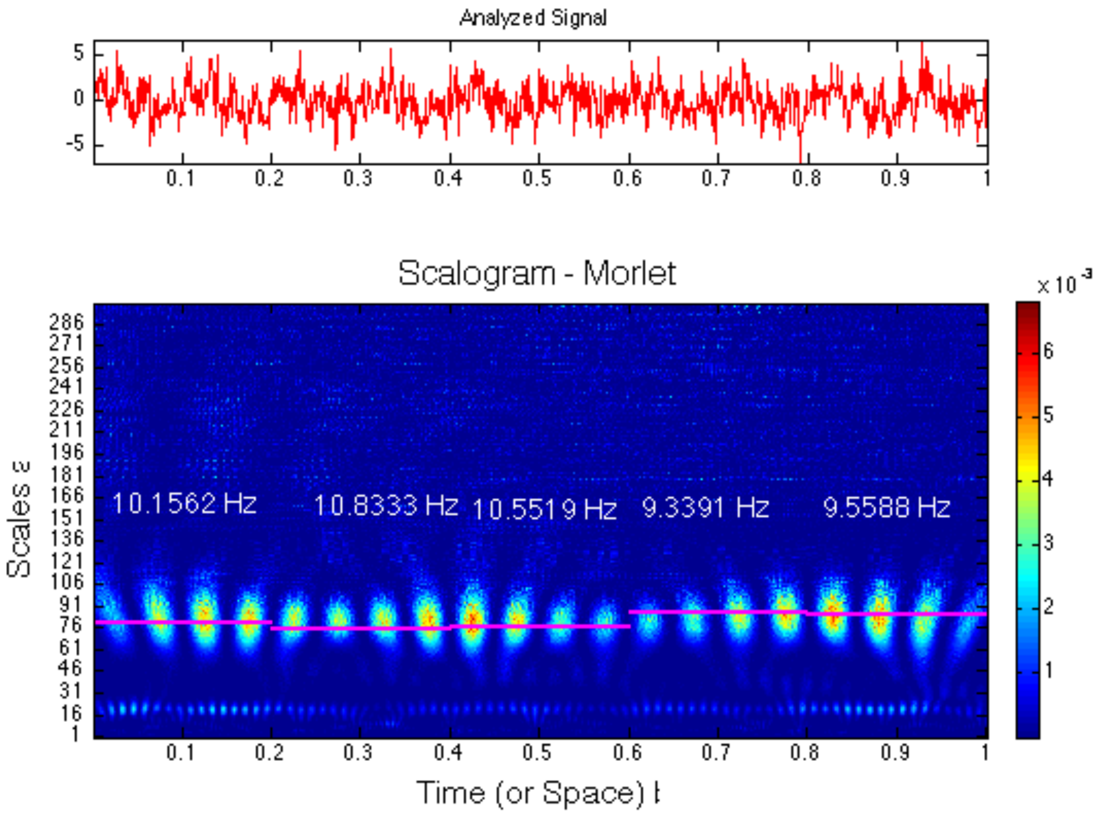


Figure 4 – 10Hz sine wave and 40Hz sine wave on sample amplitude with noise



Appendix

```
%matlab code with the WaveLab toolbox running
clearvars; clf;

xmax = 1;
points = 1000;

%%%%%%%%%%%%%%%
% Figure 1
%%%%%%%%%%%%%%%

cycles = 10;
wname = 'morl';

Fs = 4;
t = 0:xmax/points:xmax;
x  = sin(2*pi*t*cycles);
plot(t,x);
scales = 1:300;

%%%%%%%%%%%%%%%
%Figure 2 and 3 - 10 20 0 40 50
%%%%%%%%%%%%%%%

%wname = 'gaus1'; % figure 2
wname = 'morl'; % figure 3

t = 0:xmax/points:xmax;
Fall = [10 20 0 40 50];
x  = sin(2*pi*t*Fall(1)).*(t<0.2) +...
        sin(2*pi*t*Fall(2)).*(t<0.4).*(t>0.2) +...
        sin(2*pi*t*Fall(3)).*(t<0.6).*(t>0.4) +...
        sin(2*pi*t*Fall(4)).*(t<0.8).*(t>0.6) +...
        sin(2*pi*t*Fall(5)).*(t>0.8);
plot(t,x);
scales = 1:.01:150;

%%%%%%%%%%%%%%%
% Figure 4 - Corrupted by noise
%%%%%%%%%%%%%%%

F1 = 10; F2 = 40;
wname = 'morl';
t = 0:xmax/points:xmax;
x  = sin(2*pi*t*F1) + sin(2*pi*t*F2);
wn = randn(1,length(x));
wn = 1.5*wn/std(wn);
```

```
x = x + wn;
scales = 1:300;

coefs = cwt(x,scales,wname);
t = (1:length(x))./length(x);
clf;
wscalogram_frequency('image',coefs,'scales',scales,'xdata',t,'yd
ata',x);
hold on
TAB_Sca2Frq = scal2frq(scales,wname,1/points);

samples = 5;
treshold = 2;

%%%%%%%%%%%%%%%
% For figures 1 and 4
%%%%%%%%%%%%%%%

firsts = (0:samples-1).*xmax./samples;
lasts = (1:samples).*xmax./samples;

for sampleNum = 1:samples
    firstI = floor(firsts(sampleNum) * (points + 1)) + 1;
    lastI = floor(lasts(sampleNum) * (points + 1)) + 1;
    if lastI > points + 1
        lastI = points+1
    end

    [Ccolumn, Icolumn] = max(abs(coefs(:,firstI:lastI)));
    [Crow, Irow] = max(abs(Ccolumn)); %Crow is total max
    MaxResponse = Icolumn(Irow)
    Freq = TAB_Sca2Frq(MaxResponse)

    if Crow > treshold
        plot([firsts(sampleNum) lasts(sampleNum)],[MaxResponse
MaxResponse],'Color','m','LineWidth',2);
        t = gtext(strcat(num2str(TAB_Sca2Frq(MaxResponse)),'
Hz'));
    else
        t = gtext('No Signal');
    end
    set(t,'FontWeight','bold','FontSize',14,'Color','white')
end

%title('Scalogram - Morlet', 'FontSize', 16);
title('Scalogram - 1st Hermitian', 'FontSize', 16);
set(get(gca,'XLabel'),'FontSize',15)
set(get(gca,'YLabel'),'FontSize',15)
```

Bibliography

Daubechies, Ingrid. *Ten Lectures on Wavelets*. Philadelphia, PA.: Society for Industrial and Applied Mathematics, 1992. Print.

Kaiser, Gerald. "The Fast Haar Transform." *IEEE Potentials* (1998). Web.

Larson, Eric, and Ross Maddox. "Real-Time Time-Domain Pitch Tracking Using Wavelets." Web.

Mallat, Stéphane. *A Wavelet Tour of Signal Processing* . 2nd ed. San Diego: Academic Press, 1999. Print.

Polikar, Robi. "The Engineer's Ultimate Guite to Wavelet Analysis." *Index to Series of Tutorials to Wavelet Transform*. N.p., n.d. Web. 2 May 2011. <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>.

Valens, Clemens. "A Really Friendly Guide To Wavelets." *A Really Friendly Guide To Wavelets*. N.p., n.d. Web. 2 May 2011. <http://polyvalens.pagesperso-orange.fr/clemens/wavelets/wavelets.html>.