15

Wavelet Analysis: Time Domain Properties

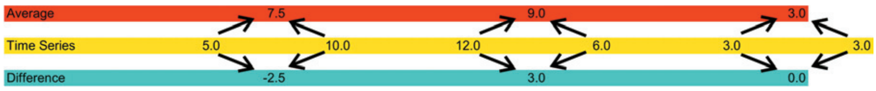
15.1 INTRODUCTION

Although the mathematics for wavelet analysis have existed for about a century, most of their applications in signal processing, feature detection, and data compression have been developed over the past few decades. Wavelet analysis is very useful for analyzing physiological systems because, as opposed to most classical signal analysis approaches, it provides the means to detect and analyze nonstationarity in signals. The simplest wavelet is the Haar wavelet, first described in the early 1900s by Alfred Haar. A few other famous, more recent contributors to the field of wavelet analysis are Morlet, Mallat, and Daubechies. A great practical and simple introduction into wavelets is given by Walker (1999), and a thorough overview can be found in Mallat (1998).

This chapter introduces the techniques of wavelet analysis using the simplest example, the ***Haar*** wavelet. In the follow-up sections we will extend this to the application of the ***Daubechies*** wavelet. First we will explore the procedures used to obtain the wavelet transform and then we will discuss some of the mathematical details.

15.2 WAVELET TRANSFORM

The underlying principle of wavelet analysis is most easily explained by considering a sampled time series (5.0, 10.0, 12.0, 6.0, 3.0, 3.0, . . .), which we want to examine for trends and fluctuations over subsequent pairs:



In this example, the average (red) of subsequent pairs is $[x(n-1)+x(n)]/2$, and the difference (blue) is $[x(n-1)-x(n)]/2$. In this process, no information was lost because the original time series (yellow) can be reconstructed from a combination of the average and difference vectors: the first value (5.0) is the sum of average and difference (7.5 – 2.5 = 5.0), the second point of the time series is the difference (7.5 – (–2.5) = 10.0), and so on. Because the time series contains the same information as the average and difference signals, we may represent it either in its original, in raw form as [5.0, 10.0, 12.0, 6.0, 3.0, 3.0, . . .], or in a transformed fashion as a combination of the average and difference forms [7.5, 9.0, 3.0, . . .] [–2.5, 3.0, 0.0, . . .]. As we will see, aside from a factor of $\sqrt{2}$, application of the Haar wavelet transform is almost identical to calculating the average and difference as shown here.

15.2.1 Haar Wavelet and Scaling Signals

The level-1 Haar wavelet and the associated scaling signal are shown in Figure 15.1. In this section, we start with a level-1 wavelet, leaving an introduction to the meaning of different level transforms and higher-level wavelets for Section 15.2.4. Both signals in Figure 15.1 are square waves with amplitudes of $\frac{1}{\sqrt{2}}$, the wavelet is biphasic and the scaling signal is non-negative. Let's consider the transform of an input signal of *N* samples. The first step is to define the level-1 Haar wavelet (*W*) and scaling signal (*S*) as vectors of length *N*:

$$W_1^1 = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right] \tag{15.1}$$

and

$$S_1^1 = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \ldots, 0\right] \tag{15.2}$$

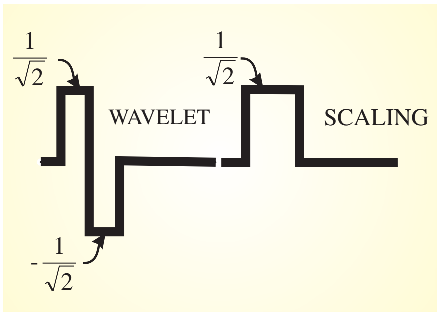


Figure 15.1 The level-1 Haar wavelet and scaling signal.

In W and S, the superscripts indicate that the signals are level-1 and subscripts indicate that both signals start at the first position in the vectors.

As in the preceding numerical example, we will use the scaling and wavelet signals to determine the ***trend*** (= weighted average) and the ***fluctuation*** (= weighted difference) in a time series. To demonstrate, we start with a function G that is sampled N times at regular time intervals:

$$G = [g_1, g_2, g_3, \ldots, g_N] \tag{15.3}$$

The trend of the first 2 points can be obtained from

$$t_1 = \frac{g_1 + g_2}{\sqrt{2}} = G.S_1^1 \tag{15.4}$$

The second part of Equation (15.4) shows that t_1 is the ***scalar product*** of vectors G and S_1^1. Similarly, the fluctuation between the first 2 points is

$$f_1 = \frac{g_1 - g_2}{\sqrt{2}} = G.W_1^1 \tag{15.5}$$

Again, Equation (15.4), the second part of Equation (15.5), shows that f_1 is the scalar product of vectors G and W_1^1.

Notes:
1. The reason for the weighting factor (i.e., division by the $\sqrt{2}$ instead of simply dividing by 2) is to preserve the energy content across the transformed variables. This is further discussed in Section 15.2.3.
2. Obtaining the trend as the sum of g_1 and g_2 weighted by $1/\sqrt{2}$ effectively means that the average of the 2 data points is multiplied by $\sqrt{2}$ $\left(\text{i.e., } \frac{g_1 + g_2}{2}\sqrt{2} = \frac{g_1 + g_2}{\sqrt{2}}\right)$. The same relationship exists between the fluctuation and the difference.

Continuing with this procedure, we shift wavelet and scaling signals by two positions:

$$W_2^1 = \left[0, 0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, \ldots, 0\right] \tag{15.6}$$

and

$$S_2^1 = \left[0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \ldots, 0\right] \tag{15.7}$$

Note that the subscripts for *S* and *W* are now 2, reflecting the shift of the signals to the second pair of data points. We now repeat the process of calculating the trend and fluctuation. Using the scalar product notation, the weighted average of points 3 and 4 can be obtained from

$$t_2 = G.S_2^1 \tag{15.8}$$

The weighted difference between points 3 and 4:

$$f_2 = G.W_2^1 \tag{15.9}$$

We continue to shift the wavelet and scaling signals in steps of two until the end of signal *G*. Because the length of *G* is *N*, we will obtain *N*/2 trend values and *N*/2 fluctuation values — that is, for all $m = 1, 2, 3, \ldots, N/2$ we obtain the following expression for the trend values:

$$t_m = \frac{g_{2m-1} + g_{2m}}{\sqrt{2}} = G.S_m^1 \tag{15.10}$$

Similarly, the weighted difference between subsequent pairs of points is

$$f_m = \frac{g_{2m-1} - g_{2m}}{\sqrt{2}} = G.W_m^1 \tag{15.11}$$

We now group all the weighted averages and differences into two vectors:

$$a^1 = [t_1, t_2, \ldots, t_{N/2}] \tag{15.12}$$

$$\text{and} \quad d^1 = [f_1, f_2, \ldots, f_{N/2}] \tag{15.13}$$

The superscripts of *a* and *d* indicate that we have used level-1 vectors. Again, the subscripts of the elements *t* and *f* indicate the position of the wavelet and scale signals within the original *N* data points.

15.2.2 Level-1 Haar Transform and the Inverse Haar Transform

The level-1 Haar transform can be defined from the preceding as a procedure to determine the trend and fluctuation components of a signal. In the previous example, the level-1 Haar transform of *G* is a^1 and d^1:

$$G \overset{H_1}{\longmapsto} (a^1 | d^1) \tag{15.14}$$

In the example time series [5.0, 10.0, 12.0, 6.0, 3.0, 3.0] we used earlier in Section 15.2, the level-1 Haar transform produces the two vectors: [7.5 2, 9.0 2, 3.0 2, −2.5 2, 3.0 2, 0.0], a result very similar to the averages and differences calculated in Section 15.2. A graphical example of a 1-level Haar transform is shown in Figure 15.2. The input signal (red) consists of a set of oscillations. The results of the level-1 Haar transform (black) consists of two main parts: the trend (1–512) and the fluctuation (513–1024).

The first half of the transform result — produced with the scaling signal *S* — contains high amplitudes, while the second part of the

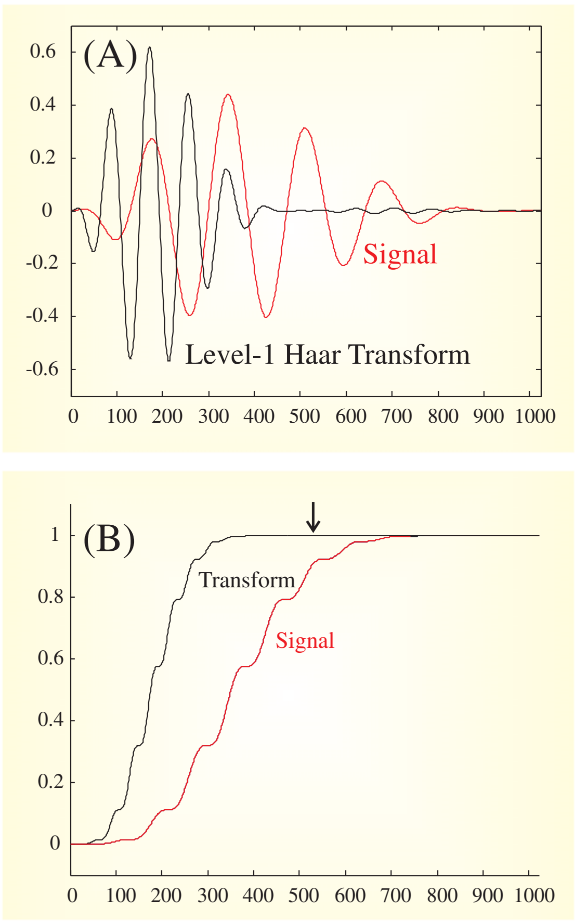


Figure 15.2 Example of a level-1 Haar transform. (A) The transform (black) of an input wave (red). (B) The cumulative energy shows that for the transformed wave the trend signal contains most of the energy — that is, at point 512 (arrow) the ratio is ~1.

transform — produced with the wavelet W — produces a low amplitude signal. This seems a trivial observation, but it is a critical aspect of the analysis (for reasons that will soon become clear).

Use this MATLAB routine to produce Figure 15.2:

```
% pr15_1.m
% A level-1 Haar Wavelet Analysis

clear;

N=1024;                                      % # of points

for n=1:N;m=(n-1)/N;g(n)=20*m^2*(1-m)        % input  signal
   ^4*cos(12*pi*m);end;

for m=1:N/2;
   a(m)=(g(2*m-1)+g(2*m))/sqrt(2);           % Use direct formulas for
d(m)=(g(2*m-1)-g(2*m))/sqrt(2);              %  t and f
end;

H1=[a d];                                    % The level-1 Haar
                                             %  transform

% plot results
figure
plot(g,'r');
hold
plot(H1,'k');
axis([0 1024 -0.7 0.7]);
xlabel ('Time (Sample#)')
ylabel ('Amplitude')
title(' Original Signal (red) and 1-Level Haar Transform (black)')
```

The ***inverse Haar transform*** starts from the a^1 and d^1 transformed vectors and allows us to recreate the original function G again. In the particular case of the Haar transform, the inverse procedure can be expressed in the form of a summation if we define A^1 and D^1 as

$$A^1 = [t_1, t_1, t_2, t_2, \ldots\ldots\ldots\ldots, t_{N/2}, t_{n/2}]/\sqrt{2} \tag{15.15}$$

and $$D^1 = [f_1, -f_1, f_2, -f_2, \ldots\ldots\ldots\ldots\ldots, f_{N/2}, -f_{n/2}]/\sqrt{2} \tag{15.16}$$

Please note that the doubling of t_n and f_n are not typos and that each second f_n of the pair is associated with a minus sign. Also note that all terms in Equations (15.15) and (15.16) are divided by $\sqrt{2}$. This corrects for the fact that we multiplied the average and difference with $\sqrt{2}$ (see Equations (15.4) and (15.5) with the associated note). The inverse Haar transform is therefore simply the sum of both vectors:

$$G = A^1 + D^1 \tag{15.17}$$

Equations (15.15) and (15.16) can be put in a (more formal) vector form:

$$\begin{aligned} A^1 &= t_1 S_1^1 + t_2 S_2^2 + \ldots + t_{N/2} S_{N/2}^1 \\ &= (G.S_1^1)S_1^1 + (G.S_2^1)S_2^1 + \ldots + (G.S_{N/2}^1)S_{N/2}^1 \end{aligned} \tag{15.18}$$

$$\begin{aligned} D^1 &= f_1 W_1^1 + f_2 W_2^1 + \ldots + f_{N/2} W_{N/2}^1 \\ &= (G.W_1^1)W_1^1 + (G.W_2^1)W_2^1 + \ldots + (G.W_{N/2}^1)W_{N/2}^1 \end{aligned} \tag{15.19}$$

Note that each term in the above equations (15.18) and (15.19) is a vector. In equation (15.18), $(G.S_m^1)$ is a scalar product representing t_m (Equation (15.10)); the factor $(G.S_m^1)$ multiplied with vector S_m^1 produces a vector $[0, 0, \ldots, t_m, t_m, \ldots, 0]/\sqrt{2}$. The sum of these vectors for all m generates the expression for A^1 in equation (15.15). The same procedure can be followed for the wavelet to obtain equation (15.19). The advantage of this notation is that it is easily extended from level 1 to a higher level transform and can also be applied to inversion of other transforms besides the Haar transform.

15.2.3 Energy of the Level-1 Transform

The Haar transform looks fairly simple (a weighted average and weighted difference). The only apparent nuisance in this simple transform is the $\sqrt{2}$ factor that appears in the wavelet definition, the transform, and the inverse transform. There is a reason for this $\sqrt{2}$ correction, namely the conservation of energy across domains. As with the Fourier transform, we would like to keep the energy content of the signal the same across the signal transformations and inverse transformations (Parseval's theorem, Appendix 7.1). For the level-1 Haar transform, the necessary correction factor is $\sqrt{2}$, though this normalization factor is necessarily different for higher levels of the Haar transform (to be discussed in Section 15.2.4) or different types of wavelet transforms.

Here, we define the energy of a sample as the square of the sampled value. This means that the energy of the first two samples of G is $g_1^2 + g_2^2$, and the first elements derived in the transform from these samples are t_1 and f_1. Thus, to preserve energy in this representation, we want to satisfy the following relationship:

$$g_1^2 + g_2^2 = t_1^2 + f_1^2 \tag{15.20}$$

We want this relationship to be true for all pairs and their associated trend and fluctuation values. We can use Equations (15.10) and (15.11) to show that the relationship in Equation (15.20) is indeed correct for all m:

$$t_m^2 = \frac{g_{2m-1}^2 + g_{2m}^2 + 2g_{2m-1}g_{2m}}{2}$$

$$f_m^2 = \frac{g_{2m-1}^2 + g_{2m}^2 - 2g_{2m-1}g_{2m}}{2}$$

$$\overline{\phantom{t_m^2 + f_m^2 = g_{2m-1}^2 + g_{2m}^2}}\;+$$

$$t_m^2 + f_m^2 = g_{2m-1}^2 + g_{2m}^2$$

As shown in Section 15.2.2 (equation (15.17)), the inverse wavelet transform procedure exactly recreates the original function G, and here we showed that the transform and its inverse also preserve the energy content.

Interestingly, if we look into the distribution of energy in the original and transformed signals in Figure 15.2, most energy is transferred to the trend part of the transform and very little shows up in the fluctuation signal (e.g., $f_m \approx 0$). This distribution becomes clear if we look at the cumulative energy distribution of the squared signals. In MATLAB you may use the command cumsum to calculate the cumulative sum of the elements in a vector. ***After running pr15_1.m***, type the following commands to evaluate the distribution of the energy:

```
figure;hold;
plot(cumsum(g.^2)/max(cumsum(g.^2)))
plot(cumsum(H1.^2)/max(cumsum(g.^2)),’r’)
```

Align the obtained plot of the cumulative energy with the one obtained with the script (as in Fig. 15.2). Because both cumulative plots are normalized to the maximum power of the input signal max(cumsum(g.^2)), you can see that at around sample 512 (i.e., the transition point from the trend vector to the fluctuation values, the arrow in Fig. 15.2B) close to 100% of the energy is already captured (you can use the [x y]=ginput command to read x and y values from the figure).

15.2.4 Multiresolution Analysis (MRA)

The procedure we described for the level-1 Haar transform can be repeated multiple times. By recursively applying the transform to the trend signal (the weighted average), we obtain higher-level transforms (Fig. 15.3). Note that we leave the fluctuation (= weighted difference) intact and continue to split the weighted average only!

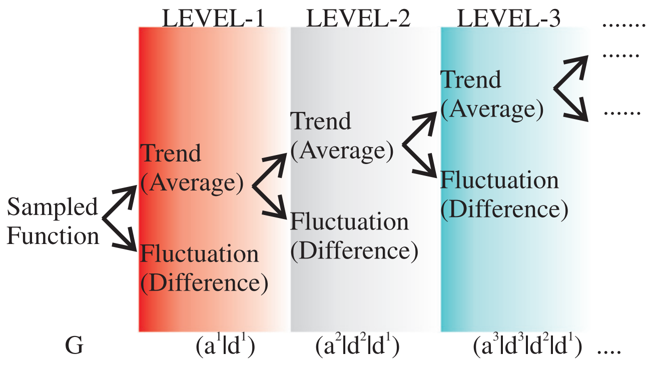


Figure 15.3 Higher-level Haar transforms of a sampled function G.

Note: This is only one possible approach. The so-called wavelet packet transform transforms *both* the trends and fluctuations. The wavelet packet transform (using the Haar scaling and wavelets) is called the Walsh transform.

Using the transform in Equation (15.14) repeatedly, we get

$$\begin{tikzcd} G \arrow[r, mapsto, "H_1"] & (a_1 \quad d_1) \arrow[d, "H_1"] \\ & (a_2 \quad d_2) \end{tikzcd}$$

Combining these results after using the level-1 transform twice creates the level-2 transform:

$$G \overset{H_2}{\mapsto} (a_2\ d_2\ d_1)$$

Generally, at the level-n transform we get

$$G \overset{H_n}{\mapsto} (a_n\ d_n\ d_{n-1} \ldots d_1)$$

If we do not spend too much time thinking about possible optimization techniques to accomplish this procedure in fewer steps, we can simply use the algorithm from the level-1 Haar program multiple times to obtain the repeated action, the result of which is depicted in Figure 15.4. This procedure is known as multiresolution analysis (MRA).

The following listing is a snippet of a MATLAB sample program performing MRA:

```
% pr15_2.m
% multihaar
% Multi Resolution Analysis MRA Haar Wavelet Analysis
% by a repeated level-1 transform

clear;

N=1024;                                        % # of points

for n=1:N;m=(n-1)/N;g(n)=20*m^2*(1-m)^4*cos(12*pi*m);end;
                                               % input signal

for m=1:N/2;
   a1(m)=(g(2*m-1)+g(2*m))/sqrt(2);            % Use direct formulas for t
                                               % and f (See equations 15.4
                                               % and 15.5)
   d1(m)=(g(2*m-1)-g(2*m))/sqrt(2);
end;

H1=[a1 d1];                                    % The 1-level Haar transform

for m=1:N/4;
   a2(m)=(a1(2*m-1)+a1(2*m))/sqrt(2);          % Use direct formulas for t
   d2(m)=(a1(2*m-1)-a1(2*m))/sqrt(2);          % and f
end;

H2=[a2 d2 d1];                                 % The 2-level Haar transform

for m=1:N/8;
   a3(m)=(a2(2*m-1)+a2(2*m))/sqrt(2);          % Use direct formulas for t
   d3(m)=(a2(2*m-1)-a2(2*m))/sqrt(2);          % and f
end;

H3=[a3 d3 d2 d1];                              % The 3-level Haar transform

% ETC ETC complete the analysis up to a10 and d10 to get Fig. 15.4
```

The trend result of the program is shown in Figure 15.4. The output of the MRA program listed here will also display graphs of the fluctuation

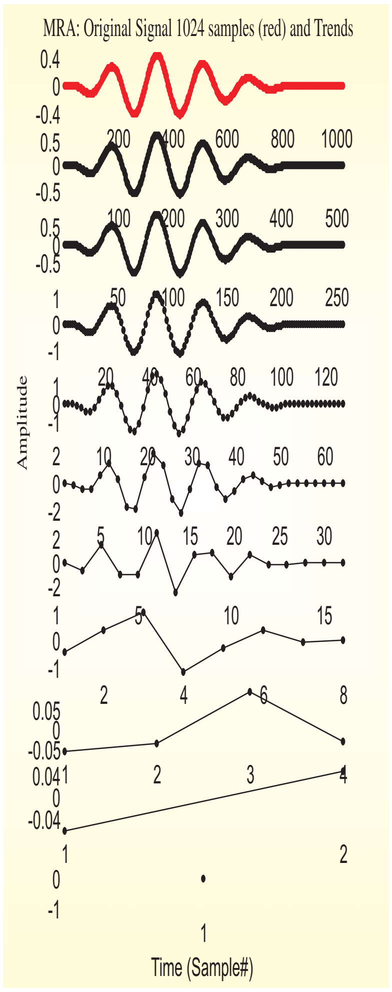


Figure 15.4 Multiresolution analysis (MRA) showing the trends (averages) of subsequent levels of the Haar transform using the procedure depicted in Figure 15.3. The fluctuation signals are not shown here but can be obtained from MATLAB script pr15_2.m.

signals. The input signal is the same as the one in Figure 15.2A, and the first trend signal corresponds with the left half of the transformed signal in Figure 15.2A.

The idea in multiresolution analysis is to repeatedly use the level-1 transforms, effectively leading to higher-level scaling and wavelet signals. For the sake of computational efficiency, however, instead of applying the level-1 transform repeatedly, one can also formulate higher-level transforms directly. To generate these higher-level functions, we start with a general form of the wavelet, which for the Haar can be represented in continuous time as a biphasic square wave Fig. 15.1:

$$W_H(t) = \begin{cases} 1 & \textit{if} \quad 0 \le t < \frac{1}{2} \\ -1 & \textit{if} \quad \frac{1}{2} \le t < 1 \\ 0 & \textit{otherwise} \end{cases} \tag{15.21}$$

The function W_H is the so-called mother wavelet. As we observed in the examples in Section 15.2.2, we obtain the wavelet transform of an input time series by ***translating*** the wavelet operation over the input. Similarly, to study the correlation at different scales, the mother wavelet is stretched (***dilated***). By using wavelets of ***different scales, we can produce different levels of the wavelet transform***.

The dilation k and translation n of the mother wavelet can be expressed by

$$W(t)_n^k = \frac{1}{\sqrt{2^k}} W_H\left(\frac{t - 2^k n}{2^k}\right) \tag{15.22}$$

In the discrete time version, the ***support*** (set of time indices where the wavelet is nonzero) for the k-level wavelet is 2^k. For instance, the level-2 Haar wavelet is

$$W_1^2 = \left[\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, 0, 0, \ldots, 0\right] \tag{15.23}$$

Compare the preceding wavelet with the level-1 Haar wavelet W_1^1 in Equation (15.1). Note that the nonzero values change by a factor of $\frac{1}{\sqrt{2^k}}$, and the support increased from 2 to 4 points.

15.2.5 Application of Wavelets in Signal Compression and Denoising

Considering Figure 15.4 where the features of the original waveform are preserved in fewer samples, it is not difficult to imagine that the energy compaction property of the Haar transform could be used for data compression purposes. The original signal in Figure 15.4 (red) contains 1024 samples; each subsequent trend signal reduces the number of samples by a factor of 2. Of course, compression will at some point only be accomplished at the cost of detail in the signal (energy that is stored in the fluctuation parts of the transform). However, in the example in Figure 15.4 it can be seen that the first steps, reducing 1024 to 256 points, seem to preserve the overall signal features rather well. Further compression beyond 256 points leads to severe loss of signal properties.

Alternatively, wavelet transforms may help in signal processing tasks such as the removal of a noise. For example, if a signal is contaminated with high-frequency noise, the fluctuation part of the transform will mainly represent the noise component. Removal of the fluctuation signal, followed by an inverse transform may be an efficient approach to "clean up" time series and pictures.

15.3 OTHER WAVELET FUNCTIONS

Currently a large set of different wavelets and wavelet analysis packages are available in signal processing. Depending on their purpose (signal compression, detection of transient phenomena in the time domain, quantifying instantaneous frequency components, etc.), they include real (e.g., Haar wavelet) and complex (Morlet wavelet), even symmetric (e.g., Mexican Hat wavelet) and odd symmetric (e.g., Haar wavelet) forms, and so on. This rich set of types of varied wavelet and associated scaling signals are possible because they only have to satisfy a few fairly reasonable conditions (Appendix 15.1). It would be beyond the scope of this introduction to discuss different types of wavelets, but we want to consider at least one other well-known type, the Daubechies wavelet, in the following section. The purpose of introducing the Daub4 scaling signal and wavelet is to illustrate how different types of signals are optimized for different signal processing tasks.

15.3.1 Daubechies Wavelet

The Daub4 scaling signal and wavelet have a support of 4 points. The four values $Ds4(i)$ for the scaling signal for Daub4 are

$$\begin{aligned} Ds4(1) &= \frac{1+\sqrt{3}}{4\sqrt{2}} \quad Ds4(2) = \frac{3+\sqrt{3}}{4\sqrt{2}} \\ Ds4(3) &= \frac{3-\sqrt{3}}{4\sqrt{2}} \quad Ds4(4) = \frac{1-\sqrt{3}}{4\sqrt{2}} \end{aligned} \tag{15.24}$$

These values create the following scaling signals:

$$\begin{aligned} DS4_1^1 &= [Ds4(1), Ds4(2), Ds4(3), Ds4(4), 0, 0, 0, \ldots, 0] \\ DS4_2^1 &= [0, 0, Ds4(1), Ds4(2), Ds4(3), Ds4(4), 0, \ldots, 0] \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ DS4_{(n/2)-1}^1 &= [0, 0, 0, \ldots, 0, Ds4(1), Ds4(2), Ds4(3), Ds4(4)] \\ DS4_{n/2}^1 &= [Ds4(3), Ds4(4), 0, 0, 0, \ldots, 0, Ds4(1), Ds4(2)] \end{aligned} \tag{15.25}$$

Note that the level-1 scaling signal translates in steps of two as does the Haar scaling signal. The difference is that in the last step ($DS4^1_{n/2}$ in Equation (15.25)), the coefficients ***wrap around*** to the beginning of the vector.

The associated Daub4 wavelet is defined by

$$
\begin{aligned}
Dw4(1) &= \frac{1-\sqrt{3}}{4\sqrt{2}} \quad Dw4(2) = \frac{\sqrt{3}-3}{4\sqrt{2}} \\
Dw4(3) &= \frac{3+\sqrt{3}}{4\sqrt{2}} \quad Dw4(4) = \frac{-1-\sqrt{3}}{4\sqrt{2}}
\end{aligned}
\tag{15.26}
$$

and the level-1 translations are

$$
\begin{aligned}
DW4^1_1 &= [Dw4(1), Dw4(2), Dw4(3), Dw4(4), 0, 0, 0, \ldots, 0] \\
DW4^1_2 &= [0, 0, Dw4(1), Dw4(2), Dw4(3), Dw4(4), 0, \ldots, 0] \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
DW4^1_{(n/2)-1} &= [0, 0, 0, \ldots, 0, Dw4(1), Dw4(2), Dw4(3), Dw4(4)] \\
DW4^1_{n/2} &= [Dw4(3), Dw4(4), 0, 0, 0, \ldots, 0, Dw4(1), Dw4(2)]
\end{aligned}
\tag{15.27}
$$

Note that the last one, $DW4^1_{n/2}$, also wraps around.

The example in Figure 15.5 shows the results of the level-1 Haar and Daubechies transforms on two types of signal. This example shows that the Haar and Daub4 wavelets have different talents with respect to successfully compressing signals. The Daub4 transform compresses the oscillatory waveform in the upper panel of Figure 15.5 rather well (i.e., there is not much energy left in the fluctuation signal). The square wave, however, is more efficiently compressed by the Haar transform.

Run daubechies1 in MATLAB and compare the output of this program with daubechies2, haar1, and haar2 scripts:

```
% daubechies1
% Level-1 Daubechies Wavelet Analysis

clear;

% Define the Daub4 scaling (alpha) and wavelet (beta) coeff
alpha(1)=(1+sqrt(3))/(4*sqrt(2));
alpha(2)=(3+sqrt(3))/(4*sqrt(2));
alpha(3)=(3-sqrt(3))/(4*sqrt(2));
alpha(4)=(1-sqrt(3))/(4*sqrt(2));
```

```
beta(1)=alpha(4);
beta(2)=-alpha(3);
beta(3)=alpha(2);
beta(4)=-alpha(1);

% # of points
N=1024;

% input signal
for n=1:N;m=(n-1)/N;
   g(n)=20*m^2*(1-m)^4*cos(12*pi*m);
end;

% Ignore the wrap around at the end!!
for m=1:N/2-2;
   % Use direct formulas for t and f
   a(m)=(g(2*m-1)*alpha(1)+g(2*m)*alpha(2)+g(2*m+1)*alpha(3)+g(2*m+2)
                                                            *alpha(4));
   d(m)=(g(2*m-1)*beta(1)+g(2*m)*beta(2)+g(2*m+1)*beta(3)+g(2*m+2)*
beta(4));
end;

% The level-1 Daub4 transform
D1=[a d];

figure

plot(g,'r');
hold
plot(D1,'k');
axis([0 1024 -0.7 0.7]);
xlabel ('Time (Sample#)')
ylabel ('Amplitude')
title(' Original Signal (red) and Level-1 Daubechies Transform (black)')
```

Both the Haar and Daubechies wavelets can be applied at different levels and used to compress signals. The more closely the wavelet matches the input curve, the closer the difference signal is to zero and the better the quality of the compression in the average signal. Better quality is judged by the level of energy of the original signal that is preserved by the average or the (loss of) energy present in the fluctuation (e.g., Fig. 15.2B). Progressively higher levels of Daub wavelets are designed so that they fit higher-order polynomials. As a general rule, one can apply a

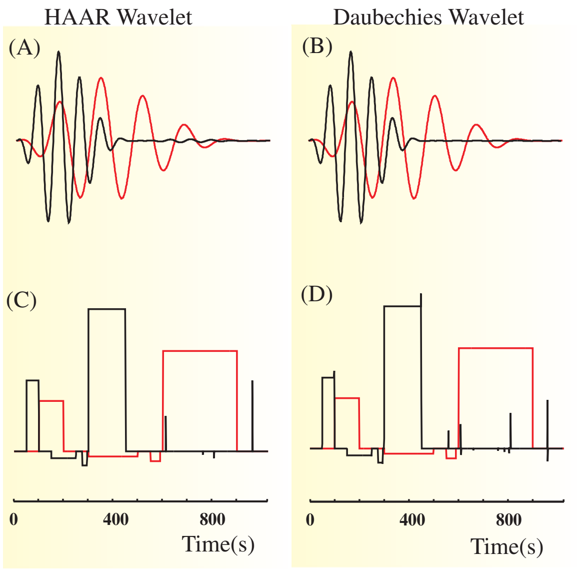


Figure 15.5 Level-1 wavelet transforms of an oscillatory signal and a signal with transients. Both waves were analyzed using the Haar and Daubechies (Daub4) wavelets. The red signal is the original, input, and the black traces represent the first average and difference signals (the same arrangement as in Fig. 15.2). Note that compression of the oscillatory wave in (A) and (B) is done most efficiently by the Daubechies wavelet (i.e., the d^1 signal is almost 0 in the latter case). For the wave shown in (C) and (D), the Haar wavelet compresses better. The plots can be obtained with MATLAB scripts haar1.m, haar2.m, daubechies1.m, and daubechies2.m.

Daub$N$ wavelet transform to polynomials of the order $< N/2$. For instance, if the input signal over the support of the wavelet is largely linear, one uses a Daub4 wavelet; for a quadratic signal, one applies the Daub6 wavelet; and so forth. In other words, if the input function over the support of a j-level Daub$N$ wavelet is a polynomial of the order $< N/2$, then the difference signal (fluctuation) ≈ 0. For obvious reasons, this feature plays an important role when compressing data sets.

15.4 TWO-DIMENSIONAL APPLICATION

Just as you apply a one-dimensional wavelet transform on a vector, you can also apply the same procedure to a two-dimensional matrix. Such applications are used in image compression and analysis. In this case, the average/trend image can be considered a compressed version of the original data, while the difference/fluctuation image shows how successful compression was. Alternatively, similar to the filter procedure shown in Figure 13.4 (Chapter 13), one can use the difference images as edge detectors. This property can also be used to enhance edges in images by multiplication of the difference signal of a transformed image with a factor >1, followed by an inverse transform.

An example of a wavelet transform of an image is shown in Figure 15.6. When transforming an image with the Haar wavelet, we follow the same

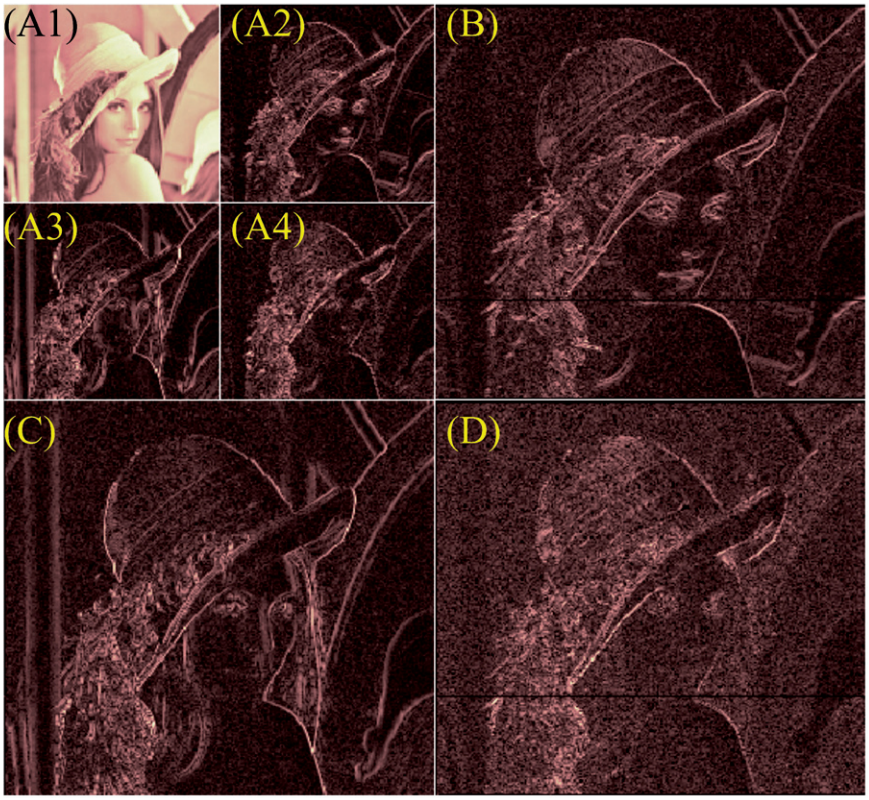


Figure 15.6 Two-dimensional Haar wavelet transform of an image. The top-left panel (A1) shows the trend (compressed) image, and the fluctuations (edges) are shown in the remaining panels. For instance, the top-right panel (B) shows the level-1 fluctuations in the columns (vertical lines) and therefore includes enhanced horizontal edges; the bottom-left panel (C) is the result of level-1 fluctuations along the rows (horizontal lines) and predominantly depicts the vertical edges. The bottom-right panel (D) is a combination of both vertical and horizontal procedures and therefore mainly depicts diagonal edges. The panels (B), (C), and (D) represent the level-1 Haar transform fluctuation, while panels (A2), (A2), and (A3) represent the equivalent fluctuations for the level-2 transform. Accordingly, (A1) depicts the trend result of the level-2 transform.

procedure as we used in Equation (15.14) for both the horizontal (rows) and vertical (columns) directions. This generates four new pictures: the average (a_H) and fluctuation (f_H) from the horizontal pass through the data and the average (a_V) and fluctuation (f_V) from the vertical pass. The fluctuation in the rows has a tendency to detect vertically oriented transitions (edges), and the fluctuations in the columns detect the horizontally oriented edges. To complete the two-dimenstional procedure, we can perform the vertical transform on f_H and the horizontal transform on f_V; the result of either procedure produces the same image (a result where the transform is applied on both the columns and rows), thereby emphasizing the diagonal fluctuation (f_D). We have now five images: a_H, a_V, f_H, f_V, and f_D. A sixth image a_1 is obtained from applying the vertical average procedure on a_H or the horizontal procedure on a_V. The results of the transform of image I can be arranged into four panels:

$$I \mapsto \left(\frac{a_1 \,|\, f_V}{f_H \,|\, f_D} \right) \tag{15.28}$$

Just as with a one-dimensional time series, the procedure we followed to transform the original image I can be repeated on a_1 to obtain the level-2 transform. Repeating the same transform recursively leads to multiresolution analysis applied to images. In this case, the upper-left quadrant occupied by a_1 is split again into four subpanels containing a_2 and the associated fluctuation signals. A concrete example of a level-2 Haar transform of Lena's image wavelet on the image of Lena is shown in Figure 15.6.

APPENDIX 15.1

A wavelet basis function W such as the one in Equation (15.21) must satisfy a set of conditions. Two of these conditions relate to the time domain: the wavelet must have zero average $\int_{-\infty}^{\infty} W\,dt = 0$ and finite energy $\int_{-\infty}^{\infty} |W|^2\,dt < \infty$. Usually one prefers an energy value for both scaling and wavelet signals normalized to 1 (i.e., $\int_{-\infty}^{\infty} |W|^2\,dt = 1$). For example, the level-1 or level-2 Haar wavelets clearly satisfy these conditions. The averages are

$$\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} = 0 \quad \text{and} \quad \frac{1}{2} + \frac{1}{2} - \frac{1}{2} - \frac{1}{2} = 0, \text{ respectively}$$

The sum of squares are

$$\left(\frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2 = 1 \quad \text{and} \quad \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = 1, \text{ respectively}$$

Such a normalized value allows one also to interpret the energy function as a probability density function (PDF) for the process represented by the wavelet.

A third condition that is often included relates to the Fourier transform $W(\omega)$ of the wavelet. This condition requires that the following norm must be finite:

$$\int_{-\infty}^{\infty} \frac{|W(\omega)|^2}{|\omega|} d\omega < \infty$$