Signal Processing Applications of Wavelets

Arthur Asuncion
Information and Computer Science
University of California, Irvine
aasuncio@uci.edu

ABSTRACT:

Wavelets are powerful mechanisms for analyzing and processing digital signals. The wavelet transform translates the time-amplitude representation of a signal to a time-frequency representation that is encapsulated as a set of wavelet coefficients. These wavelet coefficients can be manipulated in a frequency-dependent manner to achieve various digital signal processing effects. The inverse wavelet transform can then convert the manipulated wavelet coefficients back to the normal time-amplitude representation in order to yield a modified signal.

After an overview of Fourier and wavelet transforms, the Haar wavelet and the Daubechies wavelet are described in this paper. Several signal processing and musical applications of wavelets, including denoising, wavelet filtering, and data compression, are investigated. A Java implementation of a wavelet-based effects processor is also presented.

KEYWORDS:

Wavelet Transform – Haar Wavelet – Daubechies Wavelet – Digital Effects

AN INTRODUCTION TO WAVELETS

From digital signal processing to computer vision, wavelets have been widely utilized to analyze and transform discrete data. The concept of wavelets is rooted in many disciplines, including mathematics, physics, and engineering [1]. The 1980s witnessed a new wave of wavelet discoveries, like multiresolution analysis and orthonormal compactly supported wavelets. These advances have revolutionized the field and have led to many novel applications of wavelets.

A wavelet, which literally means *little wave*, is an oscillating zero-average function that is well localized in a small period of time. A wavelet function, known as a *mother* wavelet, gives rise to a family of wavelets that are *translated* (shifted) and *dilated* (stretched or compressed) versions of the original mother wavelet [2].

Wavelets have great utility in the area of digital signal processing. A digital signal can be represented as a summation of wavelets that are fundamentally identical except for the translation and dilation factors (or coefficients). Hence, a signal can be represented entirely by wavelet coefficients. These coefficients provide important frequency and temporal information which can be used to analyze a signal. Furthermore, the signal can be processed in the wavelet coefficient domain before being transformed back to the normal time-amplitude representation. Thus, wavelets facilitate a unique framework for digital signal processing.

FOURIER TRANSFORM

Wavelet literature typically includes overviews of Fourier analysis and the Fourier transform [1][3], since the Fourier transform is a conceptual precursor to the wavelet transform. Two centuries ago, Joseph Fourier showed that any periodic signal can be decomposed into a summation of sinusoids.

The Fourier transform converts a signal into a frequency spectrum derived from the frequencies of the sinusoids. The Fourier transform is presented below:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} \, dt$$

The exponential factor represents the sinusoidal component (via Euler's Relation), f represents a particular frequency, and $x(t)$ represents the input signal as a function of time [4]. Essentially, this integral is an inner product which correlates the input signal with the sinusoidal component. A fast discrete method that computes the integral above is known as the Fast Fourier Transform.

Due to the Heisenberg uncertainty principle, a fundamental tradeoff exists between frequency resolution and time resolution [4]. In the Fourier transform, a longer input time signal increases the accuracy of frequency information at the cost of losing temporal information. While the Fourier transform is an excellent tool for spectral analysis, the Fourier transform is not capable of splitting the time-frequency tradeoff into fine levels of granularity based on frequency.

WAVELET TRANSFORM

The wavelet transform is a fine-grained approach that seeks to achieve an optimal balance between frequency resolution and time resolution. At higher frequencies, the transform gains temporal information in exchange for a loss in frequency information, while at lower frequencies, the transform gains frequency information in exchange for a loss in temporal information. This fine-grained approach in handling the tradeoff is useful for digital signal and music applications, since transients normally occur at high frequencies (thus needing a higher time resolution), and lower frequencies usually require a higher frequency resolution.

Like the Fourier transform, the wavelet transform can be represented as an integral:

$$Wx(u,s) = \int_{-\infty}^{\infty} x(t)\Psi_{u,s}(t)dt$$

In the integral above, the input signal x(t) is correlated with the wavelet with translation parameter *u* and dilation parameter *s* [3]. This transform converts a signal into coefficients that represent *both* time and frequency information, with more time resolution at high frequencies, and more frequency resolution at low frequencies. The dilation of the wavelet enables the fine-grained tradeoff to occur. As with the Fourier transform, there are fast discrete ways of computing the wavelet transform.

One fast way of computing a wavelet transform is with a cascade of filters [3]. The input signal is fed into two filters, *H* and *G*. The filters produce two sets of coefficients which are both down-sampled by a factor of 2. As shown in Figure 1, this procedure is recursively applied to the set of coefficient that comes out of the *H* filter. One assumption of the wavelet transform is that the number of samples in the input signal is a power of 2. If the number of samples is not a power of 2, the signal can be zero-padded to achieve this criterion.

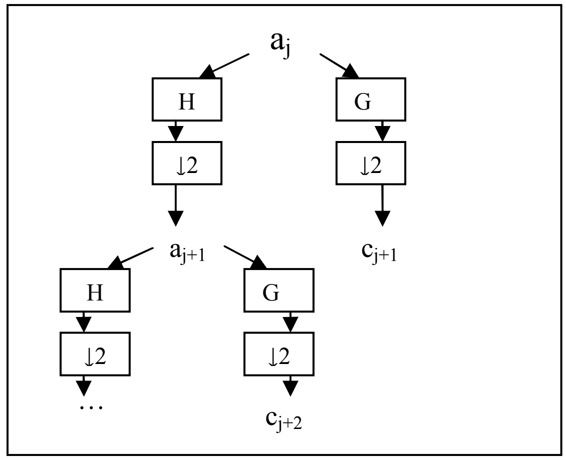

Figure 1. Fast Wavelet Transform Using Filters

HAAR WAVELET

The Haar wavelet, which Alfred Haar discovered in 1910, is both powerful and pedagogically simple. The basic Haar wavelet is a piecewise constant function that is defined as follows [5]:

$$\Psi_{[0,1[}(r) = \begin{cases} 1, & 0 \le r < \frac{1}{2} \\ -1, & \frac{1}{2} \le r < 1 \\ 0, & otherwise \end{cases}$$

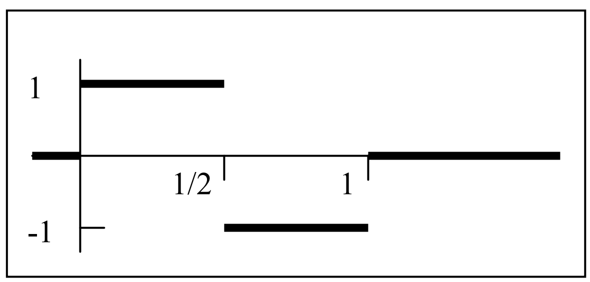

Figure 2. The Standard Haar Wavelet

The Haar wavelet transform recursively replaces adjacent pairs of *steps* in the signal with a wider *step* and a wavelet [5]. A *step* φ is a function that is 1 in a continuous region and zero everywhere else.

Consider a simple signal *f* of two samples: {7, 1}. The Haar wavelet transform calculates the average value coefficient, (7 + 1)/2, and the change coefficient (7-1)/2. The average value is the coefficient for the wider step, while the change value is the coefficient for the standard Haar wavelet [5]. The transform is presented below, and a graphical depiction is shown in Figure 3.

$$f = \frac{(7+1)}{2}\varphi_{[0,1[} + \frac{(7-1)}{2}\Psi_{[0,1[}$$

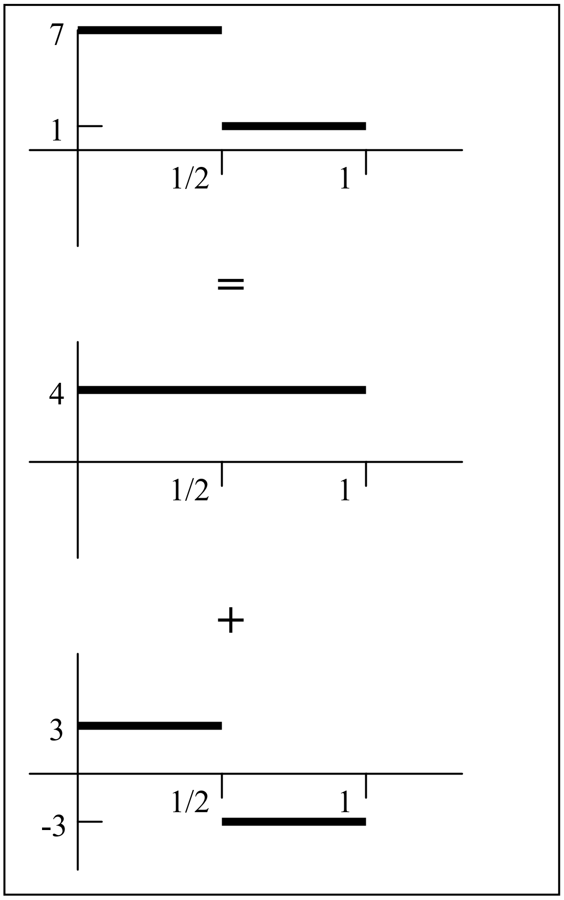

Figure 3. Haar Wavelet Transform on Signal with 2 Samples

Consider another signal *f* that has 8 values: {3, -1, 4, 8, 0, -2, 7, 1}. The Haar wavelet transform on this signal follows the procedure shown in Figure 1. The wavelet transform needs to undergo log(8)=3 sweeps, with the recursion being applied to the average value coefficients. Figure 4 details the derivation of the wavelet transform of signal *f*.

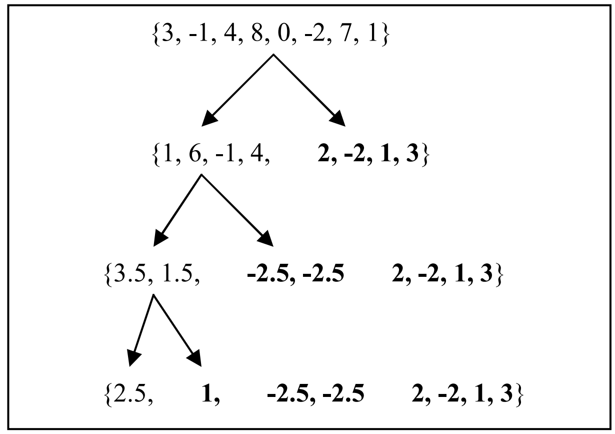

Figure 4. Derivation of Haar wavelet transform coefficients, ordered from low to high frequencies

One difference between the Fourier transform and the wavelet transform is that the frequency channels (or bins) of the Fourier transform are equally spaced, while the frequency channels of the wavelet transform are logarithmically spaced [4].

In Figure 4, the frequency of the rightmost channel, which has coefficients of 2, -2, 1, 3, is twice as great as the frequency of the middle channel, which has coefficients of -2.5, -2.5. Likewise, the frequency of the middle channel is twice the frequency of the lowest channel, which has a coefficient of 1. The left-most result of the Haar wavelet transform, 2.5, is the average value of the whole signal.

DAUBECHIES WAVELET

Named after Ingrid Daubechies, the Daubechies wavelet is more complicated than the Haar wavelet. Daubechies wavelets are continuous; thus, they are more computationally expensive to use than the Haar wavelet, which is discrete [5].

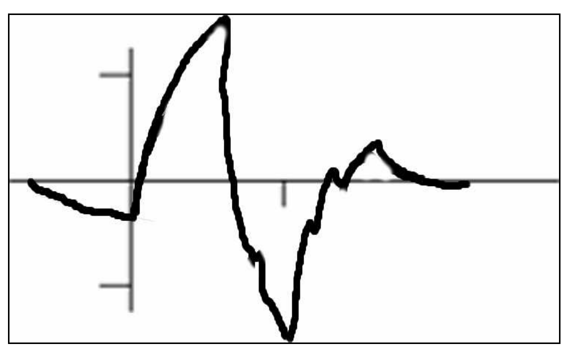

Figure 5: Daubechies Wavelet

The Daubechies 4 filter can be used to perform the Daubechies wavelet transform. The inverse filter can then reconstruct the signal out of the wavelet coefficients. The following 'lifted' filter equations encapsulate the Daubechies wavelet transform [6]:

1. $a[n] = S[2n] + \sqrt{3}S[2n+1]$
2. $c[n] = S[2n] + \frac{\sqrt{3}}{4}a[n] - \frac{\sqrt{3}-2}{4}a[n-1]$
3. $a[n] = a[n] - c[n+1]$
4. $a[n] = \frac{\sqrt{3}-1}{\sqrt{2}}a[n]$
5. $c[n] = \frac{\sqrt{3}+1}{\sqrt{2}}c[n]$

These equations are performed in sequential order for each iteration of the wavelet transform. *S* stores the signal array, *a* stores the low-pass coefficients, and *c* stores the high-pass coefficients. Succeeding iterations of the Daubechies wavelet transform would then apply these filter equations to the low pass coefficients. This transform follows the same general process that was depicted in Figure 1.

The implementation of the inverse Daubechies wavelet transform is straightforward. The inverse algorithm can be achieved by reversing the order of the equations above and reversing the operations within those equations. Thus, the signal *S* can be reconstructed from *a* and *c* coefficients.

APPLICATIONS OF WAVELETS

This section outlines several major applications of wavelets in signal processing. A Java program called AWE (Art's Wavelet Effects) has been implemented by the author in order to test the utility of wavelet transforms in signal processing.

ANALYSIS & VISUALIZATION

While excellent methods for spectral analysis already exist (like Fourier analysis), wavelets can also be a powerful tools for analyzing signals. The wavelet transform facilitates multiresolution analysis, since the wavelet transform can be recursively applied to the signal to achieve any level of accuracy. In essence, the wavelet transform allows for multiple perspectives on the signal, from a coarse-grained overview of the signal, to a detailed accurate view of the signal.

The visualization of wavelet coefficients is an integral aspect of analyzing a signal. There exist different ways to visually represent wavelet coefficients. A time-frequency plane can be used to depict the distribution of energy among time and frequency [6]. This distribution of energy is determined by wavelet coefficients.

AWE can visualize the wavelet transform coefficients in two different ways. Like a spectrum plot, the first visualization simply graphs the wavelet transform coefficients in order from low to high frequencies, with logarithmically spacing. One consequence of the logarithmic spacing is that the highest frequency occupies the entire right half of the plot. The second visualization is similar in concept to the sonogram [4] and the time-frequency plane. The wavelet coefficients are mapped onto a triangular shape. The apex of the triangle represents the coefficient for the lowest frequency, while the bottom of the triangle represents the coefficients for the highest frequency. Each row in the triangle represents a different frequency channel. This triangular representation of the wavelet coefficients inversely corresponds to the wavelet domain of influence in time, which is also a triangle [4].

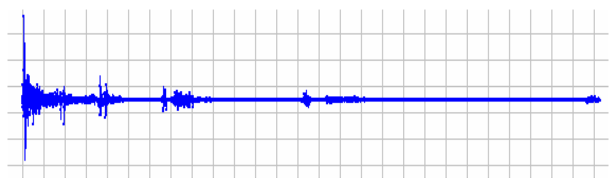

Figure 6: First visualization in AWE

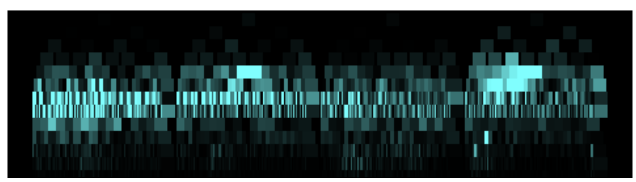

Figure 7: Second visualization in AWE

DENOISING

Noise removal is a commonly cited application of wavelets [2][5][6]; however, denoising appears to be used more in image processing than in sound processing. Denoising is essentially accomplished by amplifying or reducing certain frequency channels in order to mitigate noise.

If the signal is known to be within a certain range of frequencies, then all of the frequency channels outside of this range can be turned off in order to reduce noise. If random noise is in the signal, the removal of small variations within the signal can help to denoise the signal [6]. Small signal variations can be reduced by setting the coefficients of the highest frequency channel to zero.

AWE facilitates the denoising of signals. Frequency channels can be both strengthened and weakened using AWE's filter interface.

WAVELET FILTERING

Using the same approach as denoising, the wavelet coefficients can be manipulated to achieve a simple form of equalization. Channels can be amplified to highlight certain frequencies or reduced to suppress certain frequencies [4]. Thus, filtering can be done in the wavelet domain. AWE allows for the manipulation of ten frequency channels.

COMPRESSION

Another widely cited application of wavelets is the data compression of signals. Wavelets are being used in new JPEG and MPEG standards to compress data.

Compression can be accomplished through several ways. One way is to pass the wavelet coefficients through a threshold function. If a wavelet coefficient is above a specified threshold, then this coefficient is important, since it provides some measurable contribution to the signal. This coefficient would be kept. However, if the coefficient is below the specified threshold, the compression scheme would turn the coefficient to zero.

Data compression can also be accomplished removing the high frequencies from the signal. Since the frequencies are logarithmically spaced, the higher frequencies need much more memory space than the lower frequencies. Assuming that frequencies are spaced by powers of two, the removal of only the top two frequency channels reduces the storage space needed to ¼ of the original space.

AWE provides a simulation of wavelet data compression. Since the Daubechies wavelet is continuous and the Haar wavelet is discrete, data compression is more sonically pleasing with the Daubechies wavelet than with the Haar wavelet.

REAL-TIME MUSICAL EFFECTS

Other interesting sound effects can be generated by manipulating the wavelet coefficients. Modulation effects can be produced by multiplying the coefficients by a cosine wave. AWE provides a simple demonstration of this modulation effect. Furthermore, cross-synthesis among two sets of wavelet coefficients can be achieved. AWE also provides a demonstration of cross-synthesis. In AWE, the wavelet coefficients of one signal act as thresholds for the wavelet coefficients of the other signal.

Many other applications of wavelets, like pitch shifting and the comb wavelet transform, are described in the literature [4].

COMPUTATIONAL PERFORMANCE ISSUES

While the wavelet transforms may not be as computationally efficient as the Fast Fourier Transform, the transforms can still be accomplished near real-time. AWE implements both the Haar and Daubechies transforms along with various signal effects, and the sound still proceeds smoothly. The visualization of the wavelet coefficients is the most computationally expensive aspect in AWE. Thus, the program provides a mechanism to stop the visualization in order to ensure the efficient processing of sound.

CONCLUSION

Wavelet-based signal processing can be achieved by manipulating wavelet coefficients. Denoising, filtering, and data compression are all possible in the wavelet domain. Different wavelets can be used in the forward and inverse wavelet transforms, and the particular wavelet that is used influences the sonic characteristics of the effects.

ACKNOWLEDGEMENTS

DSP discussions with Professor Dobrian were helpful in writing this paper. Professor Meyer in EECS also provided some useful insight into the Haar wavelet.

REFERENCES

[1] Debnath, Lokenath. **Wavelet Transforms & Their Applications**. Birkhauser, 2002. P. 12.
[2] Teolis, Anthony. **Computational Signal Processing with Wavelets**. Birkhauser, 1998. P. 60-61, 173-174.
[3] Mallat, Stephane. **A Wavelet Tour of Signal Processing**. 2nd ed. Academic Press, 1999. P. 5, 257.
[4] Roads, Curtis. **The Computer Music Tutorial**. MIT Press, 1996. P. 446, 581-589, 1087.
[5] Nievergelt, Yves. **Wavelets Made Easy**. Birkhauser, 1999. P. 3-60.
[6] Jensen, A, and Cour-Harbo, A. **Ripples in Mathematics: The Discrete Wavelet Transform**. Springer, 2001. P. 102, 155-158.