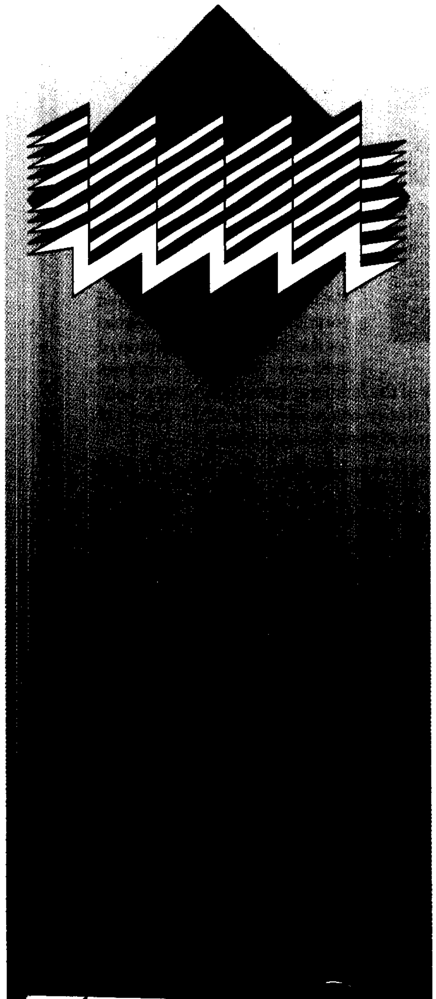

An Introduction to Wavelets

Amara Graps

THE FUNDAMENTAL IDEA BEHIND WAVELETS IS TO analyze according to scale. Indeed, some researchers feel that using wavelets means adopting a whole new mind-set or perspective in processing data.

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800s, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions. However, in wavelet analysis, the scale that we use to look at data plays a special role. Wavelet algorithms process data at different scales or resolutions. If we look at a signal (or a function) through a large "window," we would notice gross features. Similarly, if we look at a signal through a small "window," we would notice small features. The result in wavelet analysis is to see both the forest and the trees, so to speak.

This makes wavelets interesting and useful. For many decades scientists have wanted more appropriate functions than the sines and cosines, which are the basis of Fourier analysis, to approximate choppy signals.[1] By their definition, these functions are nonlocal (and stretch out to infinity). They therefore do a very poor job in approximating sharp spikes. But with wavelet analysis, we can use approximating functions that are contained neatly in finite domains. Wavelets are well-suited for approximating data with sharp discontinuities.

The wavelet analysis procedure is to adopt a wavelet prototype function, called an *analyzing wavelet* or *mother wavelet*. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion (using coefficients in a linear combi-

Wavelets were developed independently by mathematicians, quantum physicists, electrical engineers, and geologists, but collaborations among these fields during the last decade have led to new and varied applications. What are wavelets, and why might they be useful to you?

An earlier version of this article will appear in the *Official Scientific & Engineering Applications of the Macintosh 1995 Conference CD-ROM*, MacSciTech, Worcester, Mass., 1995.

1070-9924/95/$4.00 © 1995 IEEE

nation of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. And if you further choose the wavelets best adapted to your data, or truncate the coefficients below a threshold, your data are sparsely represented. This sparse coding makes wavelets an excellent tool in the field of data compression.

Other applied fields that are using wavelets include astronomy, acoustics, nuclear engineering, sub-band coding, signal and image processing, neurophysiology, music, magnetic resonance imaging, speech discrimination, optics, fractals, turbulence, earthquake prediction, radar, human vision, and pure mathematics applications such as solving partial differential equations.

Historical perspective

In the history of mathematics, wavelet analysis has many different origins.[2] Much of the work was performed in the 1930s and, at the time, the separate efforts did not appear to be parts of a coherent theory.

Pre-1930

Before 1930, the main branch of mathematics leading to wavelets began with Joseph Fourier (1807) and his theories of frequency analysis, now often referred to as Fourier synthesis. He asserted that any 2π-periodic function $f(x)$ is the sum

$$a_0 + \sum_{k=1}^{\infty} \left(a_k \cos kx + b_k \sin kx \right)$$

of its Fourier series, where the coefficients a_0, a_k, and b_k are calculated by

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

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

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

Fourier's assertion played an essential role in the evolution of the ideas mathematicians had about functions. He opened up the door to a new functional universe.

After 1807, by exploring the meaning of functions, Fourier series convergence, and orthogonal systems, mathematicians gradually were led from their previous notion of *frequency analysis* to the notion of *scale analysis*. That is, analyzing $f(x)$ by creating mathematical structures that vary in scale. How? Construct a function, shift it by some amount, and change its scale. Apply that structure in approximating a signal. Now repeat the procedure. Take that basic structure, shift it, and scale it again. Apply it to the same signal to get a new approximation. And so on. It turns out that this sort of scale analysis is less sensitive to noise because it measures the average fluctuations of the signal at different scales.

The first mention of wavelets appeared in an appendix to the thesis of A. Haar (1909). One property of the Haar wavelet is that it has *compact support*, which means that it vanishes outside of a finite interval. Unfortunately, Haar wavelets are not continuously differentiable, which somewhat limits their applications.

The 1930s

In the 1930s, several groups working independently researched the representation of functions using *scale-varying basis functions*. Understanding the concepts of basis functions and scale-varying basis functions is key to understanding wavelets; the sidebar on the next page provides a short detour lesson for those interested.

By using a scale-varying basis function called the Haar basis function (more on this later) Paul Levy, a 1930s physicist, investigated Brownian motion, a type of random signal.[2] He found the Haar basis function superior to Fourier basis functions for studying small complicated details in the Brownian motion.

Another 1930s research effort by Littlewood, Paley, and Stein involved computing the energy of a function $f(x)$:

$$energy = \frac{1}{2} \int_0^{2\pi} \left| f(x) \right|^2 dx$$

The computation produced different results if the energy was concentrated around a few points or distributed over a larger interval. This result disturbed the scientists because it indicated that energy might not be conserved. The researchers discovered a function that can vary in scale *and* can conserve energy when computing the functional energy. Their work provided David Marr with an effective algorithm for numerical image processing using wavelets in the early 1980s.

1960–1980

Between 1960 and 1980, the mathematicians

Guido Weiss and Ronald R. Coifman studied the simplest elements of a function space, called *atoms*, with the goal of finding the atoms for a common function and finding the assembly rules that allow the reconstruction of all the elements of the function space using these atoms. In 1980, Grossman and Morlet, a physicist and an engineer, broadly defined wavelets in the context of quantum physics. These two researchers provided a way of thinking about wavelets based on physical intuition.

Post-1980

In 1985, Stephane Mallat gave wavelets an additional jump-start through his work in digital signal processing. He discovered some relationships between quadrature mirror filters, pyramidal algorithms, and orthonormal wavelet bases (more on these later). Inspired in part by these results, Y. Meyer constructed the first nontrivial wavelets. Unlike Haar wavelets, Meyer wavelets are continuously differentiable; however, they do not have compact support. A couple of years later, Ingrid Daubechies used Mallat's work to construct a set of wavelet orthonormal basis functions that are perhaps the most elegant, and have become the cornerstone of wavelet applications today.

Fourier analysis

Fourier's representation of functions as a superposition of sines and cosines has become ubiquitous for both the analytic and numerical solution of differential equations and for the analysis and treatment of communication signals. Fourier analysis and wavelet analysis have some very strong links.

Fourier transforms

The Fourier transform's utility lies in its ability to analyze a signal in the time domain for its frequency content. The transform works by first translating a function in the time domain into a function in the frequency domain. The signal can then be analyzed for its frequency content because the Fourier coefficients of the transformed function represent the contribution of each sine and cosine function at each frequency. An inverse Fourier transform does just what you'd expect, transform data from the frequency domain into the time domain.

Discrete Fourier transforms

The discrete Fourier transform (DFT) estimates the Fourier transform of a function from a finite number of its sampled points. The sampled points are supposed to be typical of what the signal looks like at all other times.

The DFT has symmetry properties almost exactly the same as the continuous Fourier transform. In addition, the formula for the inverse discrete Fourier transform is easily calculated using the one for the discrete Fourier transform because the two formulas are almost identical.

Detour: What Are Basis Functions?

It is simpler to explain a basis function if we move out of the realm of analog (functions) and into the realm of digital (vectors).[1]

Every two-dimensional vector (x, y) is a combination of the vector $(1, 0)$ and $(0, 1)$. These two vectors are the basis vectors for (x, y). Why? Notice that x multiplied by $(1, 0)$ is the vector $(x, 0)$, and y multiplied by $(0, 1)$ is the vector $(0, y)$. The sum is (x, y).

The best basis vectors have the valuable extra property that the vectors are perpendicular, or orthogonal to each other. For the basis vector $(1, 0)$ and $(0, 1)$, this criterion is satisfied.

Now let's go back to the analog world, and see how to relate these concepts to basis functions. Instead of the vector (x, y) we have a function $f(x)$. Imagine that $f(x)$ is a musical tone, say the note A in a particular octave. We can construct A by adding sines and cosines using combinations of amplitudes and frequencies. The sines and cosines are the basis functions in this example, and the elements of Fourier synthesis. For the sines and cosines chosen, we can set the additional requirement that they be orthogonal. How? By choosing the appropriate combination of sine and cosine function terms whose inner product adds up to zero. The particular set of functions that are orthogonal and that construct $f(x)$ are our orthogonal basis functions for this problem.

What are scale-varying basis functions?

A basis function varies in scale by chopping up the same function or data space using different scale sizes. For example, imagine we have a signal over the domain from 0 to 1. We can divide the signal with two step functions that range from 0 to 1/2 and 1/2 to 1. Then we can divide the original signal again using four step functions from 0 to 1/4, 1/4 to 1/2, 1/2 to 3/4, and 3/4 to 1. And so on. Each set of representations codes the original signal with a particular resolution or scale.

Reference

1. G. Strang, "Wavelets," *American Scientist*, Vol. 82, 1992, pp. 250–255.

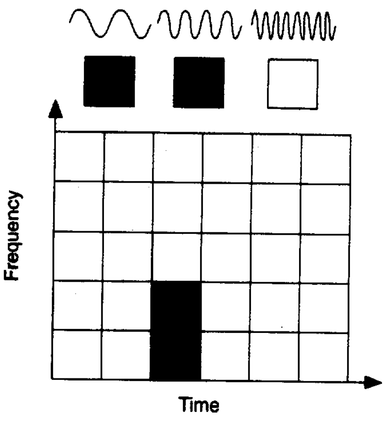

Figure 1. Fourier basis functions, time–frequency tiles, and coverage of the time–frequency plane.

Windowed Fourier transforms

If $f(t)$ is a nonperiodic signal, the summation of the periodic functions (sine and cosine) does not accurately represent the signal. You could artificially extend the signal to make it periodic but it would require additional continuity at the endpoints. The windowed Fourier transform (WFT) is one solution to the problem of better representing the nonperiodic signal. The WFT can be used to give information about signals simultaneously in the time and frequency domains.

With the WFT, the input signal $f(t)$ is chopped up into sections, and each section is analyzed for its frequency content separately. If the signal has sharp transitions, we window the input data so that the sections converge to zero at the endpoints.[3] This windowing is accomplished via a weight function that places less emphasis near the interval's endpoints than in the middle. The effect of the window is to localize the signal in time.

Fast Fourier transforms

To approximate a function by samples, and to approximate the Fourier integral by the discrete Fourier transform, requires applying a matrix whose order is the number of sample points n. Since multiplying an $n \times n$ matrix by a vector costs on the order of n^2 arithmetic operations, the problem gets quickly worse as the number of sample points increases. However, if the samples are uniformly spaced, then the Fourier matrix can be factored into a product of just a few sparse matrices, and the resulting factors can be applied to a vector in a total of order $n \log n$ arithmetic operations. This is the so-called fast Fourier transform.[4]

Wavelet transforms versus Fourier transforms

The fast Fourier transform (FFT) and the discrete wavelet transform (DWT) are both linear operations that generate a data structure containing $\log_2 n$ segments of various lengths, usually filling and transforming it into a different data vector of length 2^n.

The mathematical properties of the matrices involved in the transforms are similar as well. The inverse transform matrix for both the FFT and the DWT is the transpose of the original. As a result, both transforms can be viewed as a rotation in function space to a different domain. For the FFT, this new domain contains basis functions that are sines and cosines. For the wavelet transform, this new domain contains more complicated basis functions called wavelets, mother wavelets, or analyzing wavelets.

Both transforms have another similarity. The basis functions are localized in frequency, making mathematical tools such as power spectra (how much power is contained in a frequency interval) and scalegrams (to be defined below) useful at picking out frequencies and calculating power distributions.

Dissimilarities

The most interesting dissimilarity between these two kinds of transforms is that individual wavelet functions are *localized in space*. Fourier sine and cosine functions are not. This localization feature, along with wavelets' localization of frequency, makes many functions and operators using wavelets "sparse" when transformed into the wavelet domain. This sparseness, in turn, makes wavelets useful for a number of applications such as data compression, feature detection in images, and noise removal from time series.

One way to see the time-frequency resolution differences between the two transforms is to look at the basis function coverage of the time–frequency plane.[5] Figure 1 shows a windowed Fourier transform, where the window is simply a square wave. The square wave window truncates the sine or cosine function to fit a window of a particular width. Because a single window is used for all frequencies in the WFT, the resolution of the analysis is the same at all locations in the time–frequency plane.

An advantage of wavelet transforms is that the windows vary. In order to isolate signal discontinuities, one would like to have some very short basis functions. At the same time, in order to obtain detailed frequency analysis, one would like to have some very long basis functions. A way to achieve this is to have short high-frequency basis functions and long low-frequency ones. This happy medium is exactly what you get with wavelet transforms. Figure 2 shows the coverage in the time-frequency plane with one wavelet function, the Daubechies wavelet.

One thing to remember is that wavelet transforms do not have a single set of basis functions like the Fourier transform, which utilizes just the sine and cosine functions. Instead, wavelet transforms have an infinite set of possible basis functions. Thus wavelet analysis provides immediate access to information that can be obscured by other time-frequency methods such as Fourier analysis.

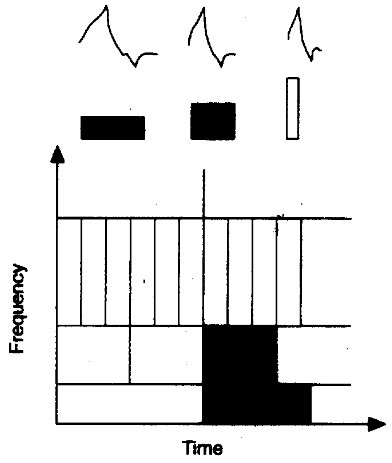

Figure 2. Daubechies wavelet basis functions, time–frequency tiles, and coverage of the time–frequency plane.

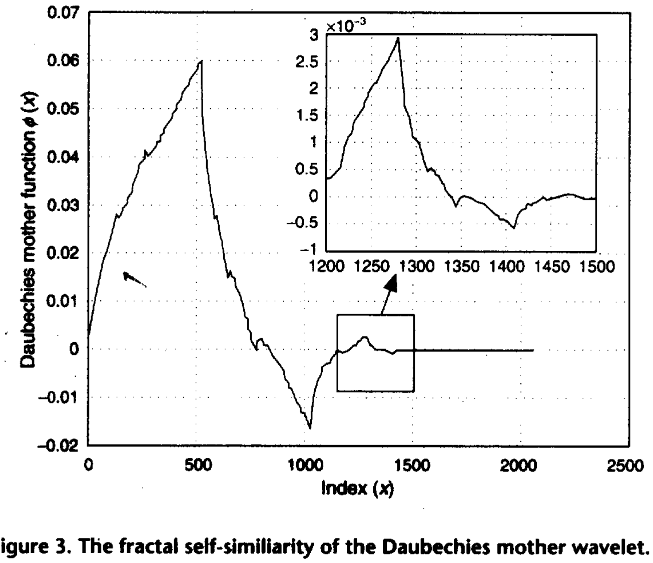

Figure 3. The fractal self-simillarity of the Daubechies mother wavelet. This figure was generated using the Wavelab command >wave=MakeWavelet(2, -4, 'Daubechies', 4, 'Mother', 2048). The inset figure was created by zooming into the region $x = 1{,}200$ to $1{,}500$.

What do some wavelets look like?

Wavelet transforms comprise an infinite set. The different wavelet families make different trade-offs between how compactly the basis functions are localized in space and how smooth they are.

Some of the wavelet bases have fractal structure. The Daubechies wavelet family is one example (see Figure 3).

Within each family of wavelets (such as the Daubechies family) are wavelet subclasses distinguished by the number of coefficients and by the level of iteration. Wavelets are classified within a family most often by the *number of vanishing moments*. This is an extra set of mathematical relationships for the coefficients that must be satisfied, and is directly related to the number of coefficients.[1] For example, within the Coiflet wavelet family are Coiflets with two vanishing moments, and Coiflets with three vanishing moments. Figure 4 illustrates several different wavelet families.

Wavelet analysis

Now we begin our tour of wavelet theory, when we analyze our signal in time for its frequency content. Unlike Fourier analysis, in which we analyze signals using sines and cosines, now we use wavelet functions.

The discrete Wavelet transform

Dilations and translations of the mother function, or analyzing wavelet $F(x)$, define an orthogonal basis, our wavelet basis function:

$$\Phi_{(s,l)}(x) = 2^{-s/2}\Phi\left(2^{-s}x - l\right)$$

The variables s and l are integers that scale and dilate the mother function Φ to generate a family of discrete wavelets, such as a Daubechies wavelet family. The scale index s indicates the wavelet's width, and the location index l gives its position. Notice that the mother functions are rescaled, or "dilated" by powers of two, and translated by integers. What makes wavelet bases especially interesting is the self-similarity caused by the scales and dilations. Once we know about the mother functions, we know everything about the basis.

To span our data domain at different resolutions, the analyzing wavelet Φ is used in a scaling equation:

$$W(x) = \sum_{k=-1}^{N-2}(-1)^k c_{k+1}\Phi(2x + k)$$

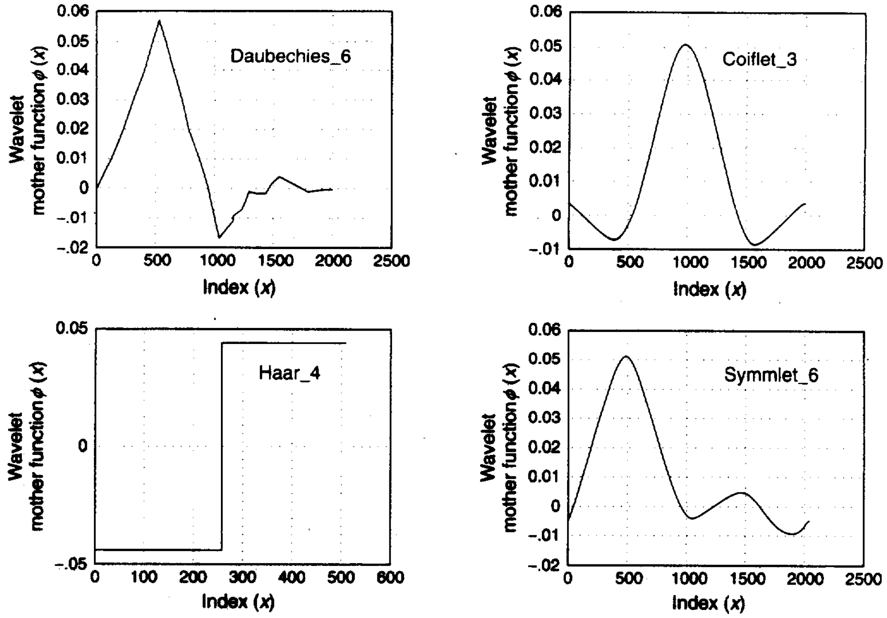

Figure 4. Several different families of wavelets. The nmmber next to the wavelet name represents the number of vanishing moments (a stringent mathematical definition related to the number of wavelet coefficients) for the subclass of wavelet. These figures were generated using Wavelab.

where $W(x)$ is the scaling function for the mother function Φ, and c_k are the *wavelet coefficients*. The wavelet coefficients must satisfy linear and quadratic constraints of the form

$$\sum_{k=0}^{N-1} c_k = 2 \qquad \sum_{k=0}^{N-1} c_k c_l = 2\delta_{l,0}$$

where δ is the delta function and l is the location index.

One of the most useful features of wavelets is the ease with which a scientist can choose the defining coefficients for a given wavelet system to be adapted for a given problem. In Daubechies' original paper,[6] she developed specific families of wavelet systems that were very good for representing polynomial behavior. The Haar wavelet is even simpler, and is often used for educational purposes.

It is helpful to think of the coefficients $\{c_0, \ldots, c_n\}$ as a filter. The filter or coefficients are placed in a transformation matrix, which is applied to a raw data vector. The coefficients are ordered using two dominant patterns, one that works as a smoothing filter (like a moving average), and one pattern that works to bring out the data's "detail" information. These two orderings of the coefficients are called a *quadrature mirror filter pair* in signal-processing parlance. A more detailed description of the transformation matrix can be found elsewhere.[4]

To complete our discussion of the DWT, let's look at how the wavelet coefficient matrix is applied to the data vector. The matrix is applied in a hierarchical algorithm, sometimes called a *pyramidal algorithm*. The wavelet coefficients are arranged so that odd rows contain an ordering of wavelet coefficients that act as the smoothing filter, and the even rows contain an ordering of wavelet coefficients with different signs that act to bring out the data's detail. The matrix is first applied to the original, full-length vector. Then the vector is smoothed and halved, and the matrix is applied again. Then the smoothed, halved vector is smoothed, and halved again, and the matrix applied once more. This process continues until a trivial number of "smooth-smooth-smooth..." data remain. That is, each matrix application brings out a higher resolution of the data while at the same time smoothing the remaining data. The output of the DWT consists of the remaining "smooth (etc.)" components, and all of the accumulated "detail" components.

The fast wavelet transform

The DWT matrix is not sparse in general, so we face the same complexity issues that we previously faced for the discrete Fourier transform.[7] We solve it as we did for the FFT, by factoring the DWT into a product of a few sparse matrices using self-similarity properties. The result is an algorithm that requires only order n operations to transform to an n-sample vector. This is the "fast" DWT of Mallat and Daubechies.

Wavelet packets

The wavelet transform is actually a subset of a far more versatile transform, the wavelet packet transform.[8]

Wavelet packets are particular linear combinations of wavelets.[7] They form bases which retain many of the orthogonality, smoothness, and localization properties of their parent wavelets. The coefficients in the linear combinations are computed by a recursive algorithm making each newly computed sequence of wavelet packet coefficients the root of its own analysis tree.

Adapted waveforms

Because we have a choice among an infinite set of basis functions, we may wish to find the best basis function for a given representation of a signal.[7] A *basis of adapted waveform* is the best basis function for a given signal representation. The chosen basis carries substantial information about the signal, and if the basis description is efficient (that is, very few terms in the expansion are needed to represent the signal), then that signal information has been compressed.

According to Wickerhauser,[7] some desirable properties for adapted wavelet bases are

(1) speedy computation of inner products with the other basis functions;
(2) speedy superposition of the basis functions;
(3) good spatial localization, so researchers can identify the position of a signal that is contributing a large component;
(4) good frequency localization, so researchers can identify signal oscillations; and
(5) independence, so that not too many basis elements match the same portion of the signal.

For adapted waveform analysis, researchers seek a basis in which the coefficients, when rearranged in decreasing order, decrease as rapidly as possible. To measure rates of decrease they use tools from classical harmonic analysis, including calculation of *information cost functions*. This is defined as the expense of storing the chosen representation. Examples of such functions include the number above a threshold, concentration, entropy, logarithm of energy, Gauss-Markov calculations, and the theoretical dimension of a sequence.

Wavelet applications

The following applications show just a small sample of what researchers can do with wavelets.

Computer and human vision

In the early 1980s, David Marr began work at MIT's Artificial Intelligence Laboratory on artificial vision for robots. He is an expert on the human visual system, and his goal was to learn why the first attempts to construct a robot capable of understanding its surroundings were unsuccessful.[2]

Marr believed that it was important to establish scientific foundations for vision, and that while doing so, one must limit the scope of investigation by excluding everything that depends on training, culture, and so on, and focus on the mechanical or involuntary aspects of vision. This low-level vision is the part that enables us to recreate the three-dimensional organization of the physical world around us from the excitations that stimulate the retina. Marr asked these questions:

- How is it possible to define the contours of objects from variations in their light intensity?
- How is it possible to sense depth?
- How is movement sensed?

He then developed working algorithmic solutions to answer each of these questions.

Marr's theory was that image processing in the human visual system has a complicated hierarchical structure that involves several layers of processing. At each processing level, the retinal system provides a visual representation that scales progressively in a geometrical manner. His arguments hinged on the detection of intensity changes. He theorized that intensity changes occur at different scales in an image, so that their optimal detection requires the use of operators of different sizes. He also theorized that sudden intensity changes produce a peak or trough in the first derivative of the image. These two hypotheses require that a vision filter have two characteristics: it should be a differential operator, and it should be capable of being tuned to act at any desired scale. Marr's operator is referred to today as a Marr wavelet.

FBI fingerprint compression

Between 1924 and today, the US Federal Bureau of Investigation has collected about 30 million sets of fingerprints.[7] The archive consists mainly of inked impressions on paper cards. Facsimile scans of the impressions are distributed among law enforcement agencies, but the digitization quality is often low. Because a number of jurisdictions are experimenting with digital storage of the prints, incompatibilities between data formats have recently become a problem. This problem led to a demand in the criminal-justice community for a digitization and compression standard.

In 1993, the FBI's Criminal Justice Information Services Division developed standards for fingerprint digitization and compression in cooperation with the National Institute of Standards and Technology, Los Alamos National Laboratory, commercial vendors, and criminal-justice communities.[9]

Let's put the problem of storing the data of digital fingerprints in perspective. Fingerprint images are digitized at a resolution of 500 pixels per inch with 256 levels of gray-scale information per pixel. A single fingerprint is about 700,000 pixels and needs about 0.6 Mbytes to store. A pair of hands, then, requires about 6 Mbytes of storage. So digitizing the FBI's current archive would result in about 200 terabytes of data. (Notice also that at today's prices of about $900 per Gbyte for hard-disk storage, the cost of storing these uncompressed images would be about $200 million.) Obviously, data compression is important to bring these numbers down.

The data compression standard WSQ (Wavelet/Scalar Quantization) implements a hand-tuned custom wavelet basis developed after extensive testing on a collection of fingerprints. The best compression ratio achieved with these wavelets is 26:1 (see Figure 5).

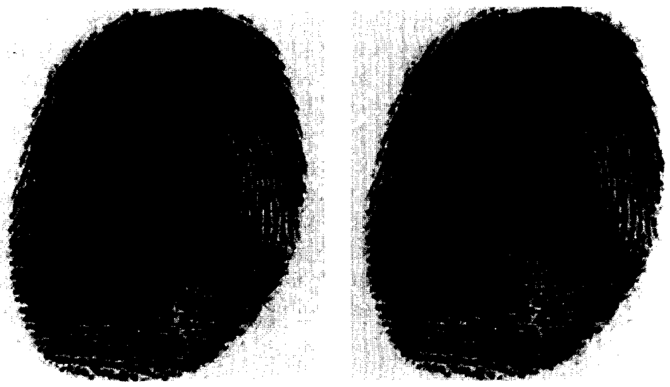

Figure 5. An FBI-digitized left thumb fingerprint. The image on the left is the original; the one on the right is reconstructed from a 26:1 compression. These images can be retrieved by anonymous FTP at ftp.c3.lanl.gov (128.165.21.64) in the directory pub/WSQ/print_data. (Courtesy Chris Brislawn, Los Alamos National Laboratory)

Denoising noisy data

In diverse fields, from planetary science to molecular spectroscopy, scientists are faced with the problem of recovering a true signal from incomplete, indirect, or noisy data. Can wavelets help solve this problem? The answer is certainly yes, through a technique, called *wavelet shrinkage and thresholding*, that David Donoho of Stanford University has worked on for a number of years.[10]

The technique works in the following way. When you decompose a data set using wavelets, you use filters that act as *averaging* filters, and others that produce *details*.[11] Some of the resulting wavelet coefficients correspond to details in the data set. If the details are small, they might be omitted without substantially affecting the main features of the data set. The idea of *thresholding*, then, is to set to zero all coefficients that are less than a particular threshold. These coefficients are used in an inverse wavelet transformation to reconstruct the data set. Figure 6 shows a pair of "before" and "after" illustrations of a nuclear magnetic resonance signal that has been transformed, thresholded, and inverse-transformed. The technique is a significant step

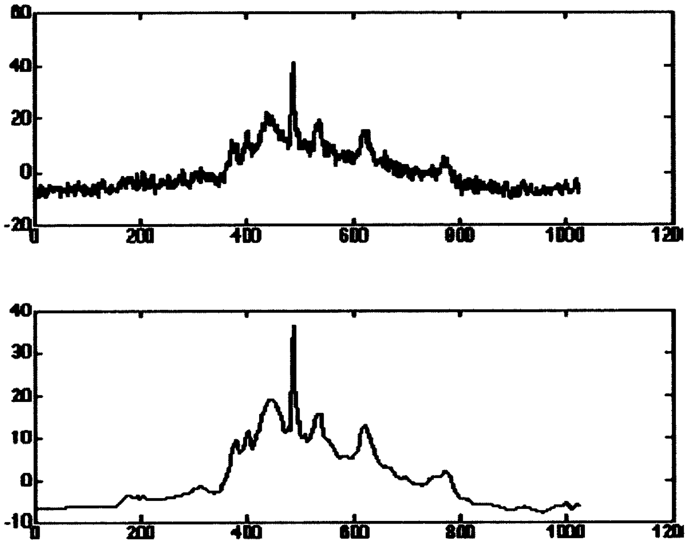

Figure 6. "Before" and "after" illustrations of a nuclear magnetic resonance signal. The original signal is at the top, the denoised signal at the bottom. (Images courtesy David Donoho, Stanford University; NMR data courtesy Adrian Maudsley, VA Medical Center, San Francisco)

forward in handling noisy data because the denoising is carried out without smoothing out the sharp structures. The result is a cleaned-up signal that still shows important details.

Figure 7 displays an image created by Donoho of Ingrid Daubechies (an active researcher in wavelet analysis and the inventor of smooth orthonormal wavelets of compact support), and then several close-up images of her eye: an original, an image with noise added, and finally a denoised image. To denoise the image Donoho

(1) transformed the image to the wavelet domain using Coiflets with three vanishing moments,
(2) applied a threshold at two standard deviations, and
(3) inverse-transformed the image to the signal domain.

Detecting self-similar behavior in a time series

Wavelet analysis is proving to be a powerful tool for characterizing behavior, especially self-similar behavior, over a wide range of time scales.

In 1993, Scargle and colleagues at NASA Ames Research Center and elsewhere investigated the quasiperiodic oscillations (QPOs) and very low frequency noise (VLFN) from an astronomical X-ray accretion source, Sco X-1, as possibly being caused by the same physical phenomenon.[12] Sco X-1 is part of a close binary star system in which one member is a late main sequence star and the other member (Sco X-1) is a compact star generating bright X rays. The causes for QPOs in X-ray sources have been actively investigated in the past, but other aperiodic phenomena such as VLFNs have not been similarly linked in the models. Their Sco X-1 data set was an interesting 5-20 keV Exosat satellite time series consisting of a wide range of time scales, from 2 ms to almost 10 hours.

Galactic X-ray sources are often caused by the accretion of gas from one star to another in a binary star system. The accreted object is usually a compact, massive star such as a white dwarf, neutron star, or black hole. Gas from the less massive star flows to the other star via an accretion disk around the compact star (that is, a disk of matter around the compact star flowing inward). The variable luminosities are caused by irregularities in the gas flow. The details of the gas flow are not well known.

The researchers noticed that the luminosity of Sco X-1 varied in a self-similar manner, that is, the statistical character of the luminosities examined at different time resolutions remained the same. Since one of the great strengths of wavelets is that they can process information effectively at different scales, Scargle used a wavelet tool called a scalegram to investigate the time series.

Scargle defines a *scalegram* of a time series as the average of the squares of the wavelet coefficients at a given scale. Plotted as a function of scale, it depicts much of the same information as does the Fourier power spectrum plotted as a function of frequency. Implementing the scalegram involves summing the product of the data with a wavelet function, whereas implementing the Fourier power spectrum involves summing the data with a sine or cosine function. The formulation of the scalegram makes it a more convenient tool than the Fourier transform because certain relationships between the different time scales become easier to see and correct, such as seeing and correcting for photon noise.

The scalegram for the time series clearly showed the QPOs and VLFNs, and the investigators were able to calculate a power law to the frequencies. Subsequent simulations suggested that Sco X-1's luminosity fluctuations may be due to a chaotic accretion flow.

Musical tones

Victor Wickerhauser has suggested that wavelet packets could be useful in sound synthesis.[13] His idea is that a single wavelet packet generator could replace a large number of oscillators. Through experimentation, a musician could determine combinations of wave packets that produce especially interesting sounds.

Wickerhauser feels that sound synthesis is a natural use of wavelets. Say you want to approximate the sound of a musical instrument. A sample of the notes produced by the instrument could be decomposed into wavelet packet coefficients. Reproducing the note would then require reloading those coefficients into a wavelet packet generator and playing back the result. Transient characteristics such as attack and decay—roughly, the intensity variations of how the sound starts and ends—could be controlled separately (for example, with envelope generators), or by using longer wave packets and encoding those properties as well into each note. Any of these processes could be controlled in real time, for example, by a keyboard.

Notice that the musical instrument could just as well be a human voice, and the notes words or phonemes.

A wavelet-packet-based music synthesizer could store many complex sounds efficiently because

- ♦ wavelet packet coefficients, like wavelet coefficients, are mostly very small for digital samples of smooth signals; and
- ♦ discarding coefficients below a predetermined cutoff introduces only small errors when we are compressing the data for smooth signals.

Similarly, a wavelet-packet-based speech synthesizer could be used to reconstruct highly compressed speech signals. Figure 8 illustrates a wavelet musical tone, or *toneburst.*

Sources of information on wavelets

The amount of wavelets-related software is growing. Many sources are on the Internet. If you are looking for articles and preprints, browse through some of the Internet sites listed next. You may find papers in subdirectories named /reports or /papers.

Stanford University

Wavelab is a Matlab wavelets library available from Stanford statistics professors David Donoho and Iain Johnstone, Stanford graduate students Jonathan Buckheit and Shaobing Chen, and Jeffrey Scargle at NASA Ames Research Center. The software consists of roughly 600 scripts, M-files, Mex-files, data sets, self-running demonstrations, and on-line documentation, and can be retrieved by anonymous FTP at playfair.stanford.edu in /pub/wavelab. I am currently writing IDL versions of many of these procedures.

I used Wavelab to produce some of the figures in this article. For example, to produce the four wavelets in Figure 4, I typed the following commands in Wavelab:

```
>wave = MakeWavelet(2,-4,'Daubechies',6,
  'Mother', 2048);
>wave = MakeWavelet(2,-4,'Coiflet',3,
  'Mother', 2048);
>wave = MakeWavelet(0,0,'Haar',4,
  'Mother', 512);
>wave = MakeWavelet(2,-4,'Symmlet',6,
  'Mother', 2048)
```

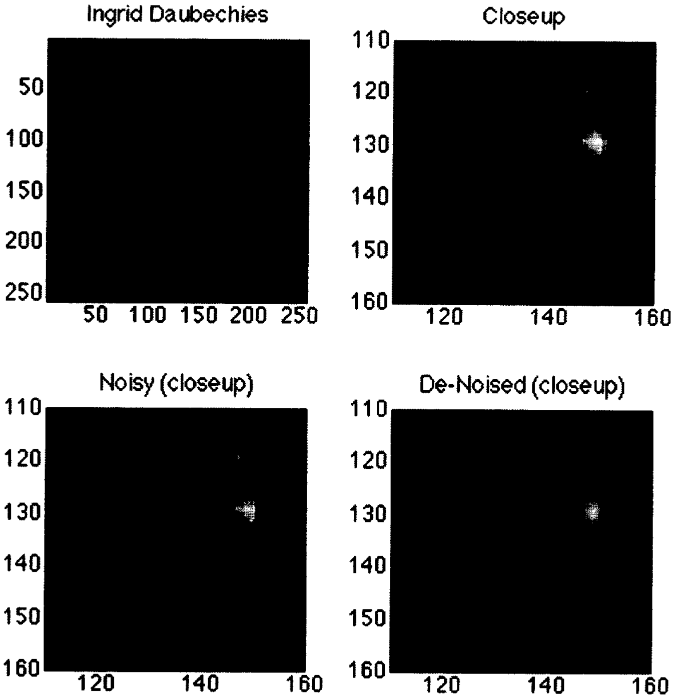

Figure 7. Denoising an image of Ingrid Daubechies' left eye. The top-left image is the original. At top right is a close-up image of her left eye. At bottom left is a close-up image with noise added. At bottom right is a close-up image, denoised. The photograph of Daubechies was taken at the 1993 AMS winter meetings with a Canon XapShot video still-frame camera. (Courtesy David Donoho)

WavBox is another Matlab wavelet toolbox from Stanford. Information on WavBox is available by anonymous FTP from the directory /pub/taswell at the site simplicity.stanford.edu.

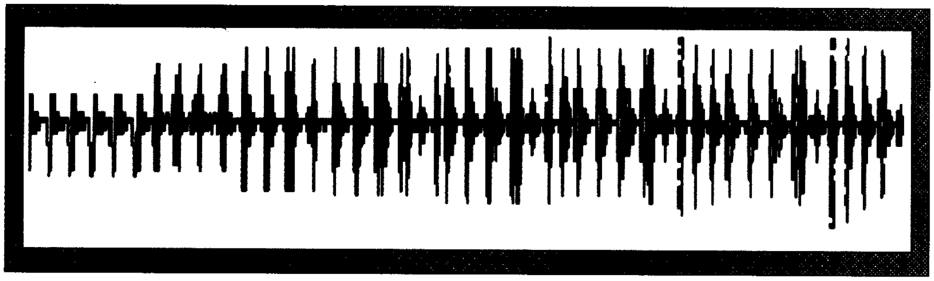

Figure 8. Wavelets for music: a graphical representation of a Wickerhauser toneburst. This screenshot of the toneburst was taken while it was playing in the Macintosh commercial sound program Kaboom! Factory, and then converted into a Macintosh System 7 SFIL formatted file. (Toneburst courtesy Victor Wickerhauser)

IRISA

WaveLib is a C library of wavelet functions developed by IRISA, Campus de Beaulieu, Rennes, France. It generates wavelets and filters, performs wavelet transforms on 1D and 2D signals, calculates entropy, performs thresholding, and so on. In addition, this package contains an interface with Matlab which displays a decomposition of a signal in the time-frequency plane as well as other useful graphical wavelet displays. WaveLib is described in a technical report available by anonymous FTP from /techreports/1994/PI-964.ps.Z at the site ftp.irisa.fr.

Rice University

The Computational Mathematics Laboratory has made available wavelet software that can be retrieved by anonymous FTP at cml.rice.edu (128.42.62.23), in /pub/software.

Yale University

The Mathematics Department has made available wavelet software which can be retrieved by anonymous FTP at pascal.math.yale.edu (128.36.23.1), in /pub/wavelets.

University of Missouri

Some wavelets educational software can be found by anonymous FTP at pandemonium.physics.missouri.edu in the directory /pub/wavelets.

Books with code

The book by Wickerhauser has C code.[7] The book by Crandall has C and Mathematica code.[1] The tutorial by Vidakovic has Mathematica code.[11] The second-edition book by Press et al. has a brief section on wavelets with Fortran or C code.[4]

Some WWW home pages

A number of Internet sites have World Wide Web home pages displaying wavelet-related topics. The following is just a sample.

- http://www.c3.lanl.gov/~brislawn/main.html (Chris Brislawn's fingerprint WSQ compression information)
- http://www.mat.sbg.ac.at/~uhl/wav.ht ml (Department of Mathematics, Salzburg University)
- http://playfair.stanford.edu/~wavelab/ (Wavelab Matlab software)
- http://www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html (Univ. of British Columbia Computer Science Department)
- http://www.mathsoft.com/wavelets.html (Matlab wavelet resources)
- http://www.math.scarolina.edu:/~wavelet/ (back issues of the Wavelet Digest)
- http://www.best.com/~agraps/current/wavelet.html (my wavelet page)

Subscribing to the *Wavelet Digest*

By subscribing to the *Wavelet Digest* you'll hear the latest announcements of available software, find out about errors in wavelet texts, find out about wavelet conferences, learn answers to questions that you may have thought about, as well as ask questions of the experts that read it.

To subscribe, send e-mail to wavelet@math.scarolina.edu with "subscribe" as the subject. To unsubscribe, e-mail with "unsubscribe" followed by your e-mail address as subject. To change your address, unsubscribe and resubscribe.

Preprints, references, and back issues can be obtained from their information servers. You can FTP at ftp.math.scarolina.edu (/pub/wavelet) or contact the gopher server gopher.math.scarolina.edu. The Web address is listed above.

Most of basic wavelet theory has been done. The mathematics have been worked out in excruciating detail, and wavelet theory is now in the refinement stage. This involves generalizing and extending wavelets, such as in extending wavelet packet techniques.

The future of wavelets lies in the as-yet uncharted territory of applications. Wavelet techniques have not been thoroughly worked out in such applications as practical data analysis, where for example discretely sampled time-series data might need to be analyzed. Such applications offer exciting avenues for exploration. ◆

References

1. R. Crandall, *Projects in Scientific Computation*, Springer-Verlag, New York, 1994, pp. 197–198, 211–212.
2. Y. Meyer, *Wavelets: Algorithms and Applications*, Soc. for Industrial and Applied Mathematics, Philadelphia, 1993, pp. 13–31, 101–105.
3. G. Kaiser, *A Friendly Guide to Wavelets*, Birkhauser, Boston, 1994, pp. 44–45.
4. W. Press et al., *Numerical Recipes in Fortran*, Cambridge Univ. Press, New York, 1992, pp. 498–499, 584–602.
5. M. Vetterli and C. Herley, "Wavelets and Filter Banks: Theory and Design," *IEEE Trans. Signal Processing*, Vol.

40, 1992, pp. 2,207–2,232.

6. I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. on Pure and Applied Math.*, Vol. 41, 1988, pp. 906–966.
7. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A.K. Peters, Boston, 1994, pp. 213–214, 237, 273–274, 387.
8. M.A. Cody, "The Wavelet Packet Transform," *Dr. Dobb's J.*, Vol. 19, Apr. 1994, pp. 44–46, 50–54.
9. J. Bradley, C. Brislawn, and T. Hopper, "The FBI Wavelet/Scalar Quantization Standard for Gray-Scale Fingerprint Image Compression," Tech. Report LA-UR-93-1659, Los Alamos Nat'l Lab, Los Alamos, N.M., 1993.
10. D.L. Donoho, "Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data," *Different Perspectives on Wavelets, Proc. of Symposia in Applied Mathematics*, Vol. 47, I. Daubechies, ed., Amer. Math. Soc., Providence, R.I., 1993, pp. 173–205.
11. B. Vidakovic and P. Muller, "Wavelets for Kids," 1994, unpublished. Available by FTP at ftp.isds.duke.edu in directory /pub/brani/papers/wav4kids[A-B].ps.Z.
12. J. Scargle et al., "The Quasi-Periodic Oscillations and Very Low Frequency Noise of Scorpius X-1 as Transient Chaos: A Dripping Handrail?" *Astrophysical J.*, Vol. 411, 1993, pp. L91–L94.
13. M.V. Wickerhauser, "Acoustic Signal Compression with Wave Packets," 1989. Available by anonymous FTP at pascal.math.yale.edu, filename acoustic.tex.

Amara Graps is a computational physicist and consultant working on numerical analysis, scientific research, technical writing, and WWW site projects for companies as well as government laboratories (NASA Ames) and universities (Stanford). Her work experience, primarily in astronomy, astrophysics, and planetary science research, was gained from her current cooperative agreement at NASA Ames, where she has been associated for nine years, and previous jobs at the University of Colorado and at the Jet Propulsion Laboratory. She earned her BS in physics in 1984 from the University of California, Irvine, and her MS in physics (with computational physics option) in 1991 from San Jose State University. She is currently writing the IDL version of a freeware Matlab wavelet algorithm package called Wavelab, a joint Stanford/NASA Ames project.

Graps can be reached at 22724 Majestic Oak Way, Cupertino, CA 95014; e-mail, agraps@netcom.com; or URL http://www.best.com/~agraps/agraps.html.