The Sea of Wavelets

Bernard J.T. Jones[1]

Kapteyn Institute, University of Groningen, 9700 AV Groningen, NL
jones@astro.rug.nl

Summary. Wavelet analysis has become a major tool in many aspects of data handling, whether it be statistical analysis, noise removal or image reconstruction. Wavelet analysis has worked its way into fields as diverse as economics, medicine, geophysics, music and cosmology.

Wavelets formalise the process of examining the characteristics of data on many scales through the use of special orthogonal function sets. At each scale we have a shrunken, less detailed, version of the original data. This comes with enough information on the residual details that were smoothed during the shrinking process to reconstruct the original data if required. Transformations using wavelets are reversible in the sense that the original data can be reconstructed from the transformed data.

The idea of looking at data on different scales provides an important approach to data analysis: this is what *Multi-Resolution analysis* is about. Not all Multi-Resolution analysis is done using wavelets, but using wavelets adds some important features to such analysis.

One of the main difficulties facing the would-be user of wavelet techniques is to extract what is needed from the vast and varied literature on the subject. In particular there seems to be a gulf between users and uses of discrete and continuous transforms which this overview hopes to clarify. My task here is to provide an easily accessible introduction to this vast and complex subject.

1 Wavelets Everywhere

1.1 The Origin of the Idea

Wavelet transforms were recognised long ago. Notably, the Haar Transform dates from 1910 [27] and the Gabor's localised Fourier Transform [24] dates from the 1946. These had found specific engineering applications, but had not, at the time, been recognised as being specific examples of a far larger class of transforms. On the mathematical side we should note that Littlewood and Paley [28] provided an important description of a scaling decomposition as early as 1931. Later, the Calderón-Zygmund theory of singular integral

Data Analysis in Cosmology, Lecture Notes in Physics, vol. 665. Edited by V. J. Martnez, E. Saar, E. Martnez-Gonzlez, and M.-J. Pons-Bordera. Berlin: Springer, 2009., p.523-563

operators was to play a major role in establishing the mathematical fundamentals of wavelet bases.

The recent explosion in the development and application of wavelets, as we now know them, started some twenty years ago. The foundations were laid by Grossmann and Morlet [26], who worked for an oil company analyzing seismic data, and by Mallat [31]. Grossmann and Morlet gave us the formal definition of the continuous wavelet transform. The crucial relationship with the theory of filters was established somewhat later: it is now more than ten years since the seminal papers of Meyer [34] and Daubechies [16] and since Daubechies published her famous lecture notes *Ten Lectures on Wavelets* [2].

An important key development of the mid 1990's, was the so-called "Lifting" mechanism for generating wavelets [41] [42]. Lifting could be used to define wavelets and to improve their properties. All finite filters related to wavelets can be obtained via the lifting technique [18]. The lifting technique provides a starting point for defining wavelet transforms on irregular point sets or on a sphere.

Related ideas are the pyramidal and quad-tree decomposition of image data in which an image is described by examining it at various resolutions [12][38]. The down-sampling required to traverse the pyramid could be achieved by any of a variety of methods: averaging groups of pixels, taking the median of a group of pixels or by some nonlinear transform that preserved features. Down-sampling using wavelets is an important case.

1.2 What can wavelets do that is interesting?

Why bother with wavelets? Do they provide us with anything new? It is almost trendy to include the word "wavelet" in the title of a paper when all that has been done is to smooth or filter the data with a function or filter that has been dubbed a wavelet because it satisfies certain mathematical conditions. We say "smooth or filter" because the world of wavelets is divided into two apparently distinct parts: continuous wavelet transforms and data filters defined at discrete points. As we shall see, these are not really distinct: the situation has parallels in the Heisenberg or Schrodinger representations of states in quantum mechanics.

Functions or filters that are wavelets provide us with a *hierarchical* set of functions or filters that have important orthogonality properties. Mostly, but not always, these functions and filters are defined on a finite interval (compact support), so they act locally.

The word "hierarchical" is significant: we are simultaneously looking at the data on a range of scales. In the world of continuous wavelets all scales are looked at, whereas in the world of wavelet filters the scales differ by factors of two (frequency doubling). The word "local" is also important: it means that the function or filter acts hierarchically on a limited subset of the data, thereby making the data analysis local (as opposed to the Fourier transform which transforms the entire data set at one go). The local and hierarchical properties mean that we are effectively zooming into our data using a microscope. At low magnification we see the overall trends, and as we zoom in we see more and more detail without being distracted by the surroundings.

Wavelets remove trends and highlight residuals in a hierarchical way. The residuals thus found can be subjected to scale dependent statistical analysis with a view to separating out noise component and leaving behind de-noised data. This has an obvious role in image processing. The advantage of wavelets in this respect is that, in general, they are of finite extent and so can analyse local trends in a way that Fourier analysis cannot. This is one reason wavelets have made an impact in the analysis of non-stationary time series.

Wavelets also provide a way of representing a function or data in terms of a hierarchy of orthogonal functions. This is important in music, for example, in trying to separate out the sound from different instruments, or to identify the notes that make up a chord. Wavelets provide a hierarchical way of representing and compacting data. The wavelet representation is often very efficient and allows for high performance lossy or lossless compression. The efficiency of the wavelet representation leads to high performance algorithms for computation.

An interesting example is provided by the astronomically familiar problem of calculating potential interactions among particles. It is necessary to calculate sums of the form

$$V_j = \sum_{i=1}^{N} \frac{m_i m_j}{|x_i - x_j|} \tag{1}$$

At first glance this involves calculating and summing N^2 terms. Over the past decades, techniques have been developed to reduce this to on the order of $N \log N$ computations by using hierarchical and multipole expansions of the potential distribution [13] [25]. The work of Rokhlin and collaborators on this problem [11] [10] has reduced the calculation to $O(N)$ by use of wavelets. Note, however, that $O(N)$ computation can be achieved without wavelets by a combination of treecodes and fast multipole moment evaluations (Dehnen [19] [20]). For a different wavelet based approach to this problem see the recent papers by Romeo et al. [37]. There is no stopping progress!

Wavelets are indeed interesting and provide powerful new tools. However, there is almost an excess of goodies: the class of wavelets is enormous and the user is confronted by the question of which members of this class best suit the problem at hand. People often give up on this question and use the simplest continuous wavelet, the Mexican Hat, or the simplest discrete point filter, the Haar wavelet without asking why they made that choice or seeking a better alternative. I hope this article will provide a relatively simple look into the rich fruits of the sea of wavelets.

1.3 What will we learn from all this?

Apart form the hope that this will provide, for some, a relatively straighforward introduction to a rapidly developing subject that is full of possibilities, there are some points, not usually stated, that I wish to emphasise.

Firstly, if you are interested only in data analysis, but not in data reconstruction, multi-resolution analysis is a very powerful tool. You do not need wavelets for this: there are many very powerful alternatives. Wavelets can be used to construct the multi-resolution data pyramids and they may offer some, as yet largely unexplored, possibilities that will provide a different view of data.

Next, one has to distinguish clearly whether continuous wavelet analysis or discrete wavelet filters are the appropriate tool for a particular task. Having decided whether to go continuous or discrete, the burning question is which wavelet to use: there is, as we shall see, a whole ocean of wavelets. Different jobs require different tools. And remember that even continuous wavelets require discretization in order that the integrals can be evaluated.

Finally, the Lifting technique for creating designer wavelets may provide a powerful tool for attacking special problems. We have seen the importance of this with regard to defining wavelets on a sphere, or on non-regularly distributed point sets.

1.4 Decisions, decisions

The development of wavelet theory has taken place on many fronts and can be compared with the 19th. century development of the theory of orthogonal polynomial sets that are solutions of ordinary differential equations. There are so many sets of orthogonal polynomials that the question frequently asked is “which should I use to study this data or solve this equation?”. Since the polynomials are intimately linked with eigenvalue problems of differential equations the choice is most often motivated by the nature of the problem: problems on a sphere are invariably solved using the Legendre Polynomials. However, when it comes to using orthogonal polynomial sets to analyse data the situation rapidly becomes confused. Fitting a curve through a set of points is often done using a simple polynomial of the form $a+bx+cx^2+\cdots+x^N$, but there are far better choices: Tchebycheff polynomials give guarantees about the nature of the residuals or any of a vast variety of spline polynomials give guarantees about the differentiability of the fitted curve.

The fact that wavelets are not, in general, associated with particular differential equations means that they are divorced from a direct relationship with physical problem. The generation and application of wavelets is thus led by interest in specific problems in particular disciplines.

1.5 Different Approaches

Contemporary wavelet analysis is the result of the convergence of developments in several quite distinct areas: function theory, engineering filters for signal processing, image processing and so on. As a consequence the literature is quite diverse and often quite difficult to grasp. Engineering papers taking the filter theory approach frequently use z-transforms, mathematical papers use function theory and so on. One of the goals af this article is to present and inter-relate material from these approaches. Thus we shall cover some elementary material on filters and data representation by orthogonal function sets, and touch on the z-transforms that dominate the engineering literature.

Astronomical use of wavelets seems to be largely confined to continuous wavelet transforms based mainly on a few specific wavelet functions. The so-called Mexican Hat wavelet holds a predominant position in this respect, perhaps because it is among the simplest of wavelets even if it is possibly not the best choice.

One of the main issues is to grasp the relationship between the continuous and discrete wavelet transforms, and to know which transform to use and when. Discussions of this are frequently shrouded in function theory (for good reasons), which can obfuscate what is actually going on.

Then there is the realization that the class of potentially useful wavelets is enormous. Why then should we fix on any particular choice? Indeed, in recent years an important general procedure for generating what might be termed "designer wavelets" has become of prime importance. The so-called "Lifting" process (section 7) provides a mechanism which can be used to design wavelets having specific properties, and moreover to generalize the concept of wavelet analysis to wavelets on non-planar surfaces (e.g: spheres) and on non-uniformly distributed point sets.

1.6 Learning about wavelets

There is now a significant number of fine texts on wavelets written from a wide variety of points of view. The World Wide Web provides an even greater repository of information, among which are many fine review articles. This article has been put together using a great number of these sources: I have taken what appear to me to be the best bits from among all I have read and put them into a (hopefully) coherent article. If in places you get a feeling of deja-vu, you will have read an article that I also enjoyed and found particularly useful.

For pedagogical reasons I will start, as have done other reviewers of this subject, with the Recipe Approach: "this is what a simple so-and-so wavelet transform is" and "this is how to apply it". We shall then see this as part of a larger and deeper concept which we shall develop into a discussion of continuous and discrete wavelet transforms. I will present a list of the more

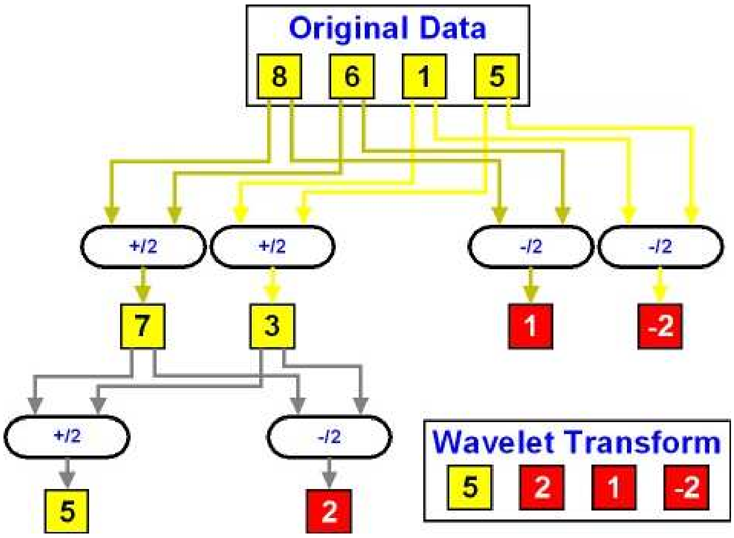

Fig. 1. Trivial wavelet transform of four data points

famous wavelet transforms and then apply them to a couple of simple problems so that we can see both continuous and discrete wavelets in action. Then finally we shall get into some of the wider issues such as the different kinds of wavelet analysis that are available and their broader applications.

2 First examples

2.1 The simplest: the Haar Transform

We shall consider a one-dimensional data set: values of a function on a set of uniformly distributed points on a line. To keep things really simple we shall work with just four points: $S_0 : \{8, 6, 1, 5\}$. Figure 1 shows the data transformations.

At the first stage we produce a new data set comprising the pairs $S_1 : \{7, 3\}$ and $D_1 : \{1, -2\}$. S_1 is obtained by adding neighbouring pairs of data values, D_1 is obtained by differencing neighbouring pairs of data values. At the second stage we do to S_1 exactly what we did to S_0 and transform the pair $S_1 : \{7, 3\}$ into $S_2 : \{5\}$ and $D_2 : \{2\}$ by summing and differencing neighbouring data values in S_1.

This process results in the so-called "multi-resolution analysis":

$$S_0 : \{8, 6, 1, 5\} \rightarrow \{S_2, D_2, D_1\} = \{5, 2, 1, -2\} \tag{2}$$

The process is reversible, we can go from $\{S_2, D_2, D_1\}$ back to $S_0 : \{8, 6, 1, 5\}$ just by adding or subtracting neighbouring pairs of numbers and using appropriate normalizations. The sequence S_i is a set of views of the data at decreasing resolution (we averaged neighboring pairs to generate the next member of the sequence). The sequence D_i provides, at each level, a view of the detail in the data that is not contained in the corresponding S_i. The S_i are sometimes called the "Smooth" coefficients and the D_i are sometimes called the "Detail" coefficients. The names are not unique: one of the confusing things is the variety of names that are used to describe the same thing. Table 2.1 lists some of the names used.

S	D
Low-pass	High-pass
Sum	Difference
Smooth	Detail
Synthesis	Analysis
Trend	Fluctuation

Table 1. Synonyms for transformed data components

This shows "what" a wavelet transform is (albeit a rather trivial one) but not "why" we might have bothered. But where is the wavelet? What did we gain?

Part of the answer to this is that the wavelet representation of the data is in many ways more efficient than the original data array. The sequence of D_i allow new methods for filtering and compressing data. In this trivial example, the transform was executed by simply taking sums or differences of neighbouring elements of the data array. More sophisticated manipulation will yield different and possibly more useful transformed data sets.

2.2 A more formal look

Let us recapitulate on and formalize the above procedure so that we can generalize it. The first step is to take an ordered data set $\{U\}$:

$$\{U : u_0, u_1, u_2, u_3, \dots\} = \{a, b, c, d, \dots\} \tag{3}$$

Now create a new data set $\{S\}$ from $\{U\}$ by averaging neighbouring values:

$$\{S : s_0, s_1, \dots\} = \{\tfrac{1}{2}(a+b), \tfrac{1}{2}(c+d), \dots\} \tag{4}$$

$\{S\}$ has half as many elements as $\{U\}$ and looks rather similar: it is a downsized version of $\{U\}$, having half the resolution. Some of the information in $\{U\}$ has been lost in the averaging process that created $\{U\}$; the missing information is contained in $\{D\}$.

Now construct the sequence $\{D\}$ of differences between neighbours:

$$\{D : d_0, d_1, \dots\} = \{\tfrac{1}{2}(a-b), \tfrac{1}{2}(c-d), \dots\} \tag{5}$$

Note that $\{D\}$ has the same number of elements as does $\{S\}$.

We can use these differences to reconstruct the original data $\{U\}$ from the averaged sequence $\{S\}$ by means of the (trivial) formulae:

$$\begin{aligned} a &= s_0 + d_0 = \tfrac{1}{2}(a+b) + \tfrac{1}{2}(a-b), \\ b &= s_0 - d_0 = \tfrac{1}{2}(a+b) - \tfrac{1}{2}(a-b), \\ c &= s_1 + d_1 = \tfrac{1}{2}(c+d) + \tfrac{1}{2}(c-d), \\ d &= s_1 - d_1 = \tfrac{1}{2}(c+d) - \tfrac{1}{2}(c-d), \\ & \quad etc. \end{aligned} \tag{6}$$

This is the simplest example of a wavelet transform: the Haar transform (we have suppresssed normalization factors of $\sqrt{2}$ for the sake of clarity).

This, like other wavelet transforms, is a linear mapping of the data values $\{U\}$ into another set of data values $\{S, D\}$. The importance of this process is that the new set of values may possess some advantageous properties that were not apparent in the original data set. The quest is to select a wavelet transform that is best suited to the particular kind of data analysis that is required for $\{U\}$.

2.3 Alternative strategies

In the preceding we averaged pairs of neighbouring numbers. We could have chosen some more complex form of averaging involving four or more numbers. We could also have stored the data in a slightly different way. The following example is an important alternative strategy:

$$\begin{aligned} \{S' : s'_0, s'_1, \dots\} &= \{a, c, \dots\} \\ \{D' : d'_0, d'_1, \dots\} &= \{b-a, d-c, \dots\} \end{aligned} \tag{7}$$

Compare equation (4) and (5). The reconstruction from this sequence is trivial.

What's the difference? In equation (7) the datum b is only used once (when calculating the difference d'_0), and this calculation can be done in-place since b is not used again. In the first method (equations (4) and (5)), a and b are both used twice. Thus memory and cpu are used more efficiently and that may be important when transforming large amounts of data.

Yet another strategy is as follows. We use values $a, c, e, \dots$ at the even numbered data points to make a prediction of what the values $b, d, \dots$ at the odd numbered points would be on the basis of linear interpolation, and then calculate and store a correction for this rather naive prediction:

$$
\begin{aligned}
\mu_0 &= \tfrac{1}{2}(a+c), \\
\delta_1 &= b - \mu_0, \\
\mu_1 &= \tfrac{1}{2}(c+e), \\
\delta_2 &= d - \mu_1, \\
\{S' : s'_0, s'_1, \dots\} &= \{a, c, e, \dots\} \\
\{D' : d'_0, d'_1, \dots\} &= \{\delta_0, \delta_1, \dots\}
\end{aligned}
\tag{8}
$$

Here μ_0 is the naïve prediction, based on linear interpolation, of what the value at the intermediate point should be. δ_0 corrects this prediction so that $b = \mu_0 + \delta_0$. We might note at this point that the mean of the coefficients in the original signal $\{a, b, c, \dots\}$ is not equal to the mean of the downsized coefficients $\{a, c, e, \dots\}$: the "energy" of the signal has not been preserved. This is a general property of this kind of interpolating wavelet. We shall show later how to apply a further correction to the values $s'_0, s'_0, \dots$ so that the desirable property of energy conservation can be realized.

This seemingly trivial point lies at the basis of an important technique in wavelet analysis: the so-called Lifting Operation, by which we can refine a given wavelet transform in such a way that there is the possibility of satisfying additional constraints. The above (trivial) analysis can usefully be viewed in a slightly different way. The $\{S' : s'_i\}$ can be interpreted as *predictors* of the values of the data. The $\{D' : \delta_i\}$ are the *corrector* numbers required for the correction of the coarser prediction. The final step will be to update the prediction so that additional constraints (such as energy conservation) are satisfied.

The lifting process is certainly one of the most significant development in wavelet theory. It allows us, for example, how to define wavelets on a sphere, how to deal with irregularly distributed data points and how to define wavelet bases for solving partial differential equations.

function	name(s)	equation
$\phi(x)$	Scaling Function	(15)
	Father Wavelet	
$\phi_{n,k}(x)$	Rescaled translate of $\phi(x)$	(16)
h_k	Refinement coefficients	(15)
$\psi(x)$	Mother Wavelet	(26)
$\psi_{n,k}(x)$	Rescaled translate of $\psi(x)$	(27)
g_k	Wavelet numbers	(32)

Table 2. Some of the wavelet jargon

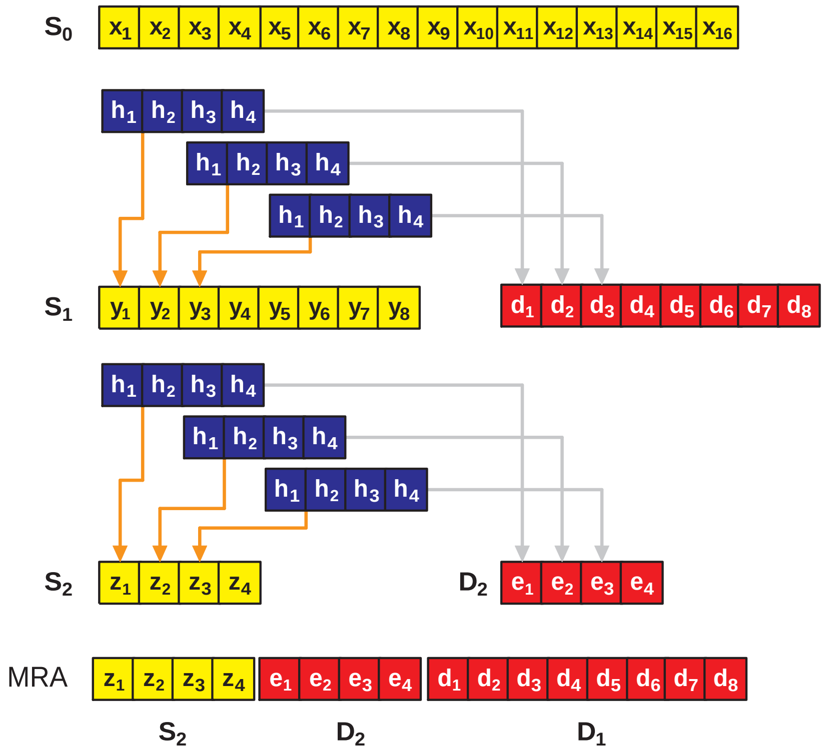

Fig. 2. Multi-resolution analysis

3 Some formal definitions

The langauge of wavelets is replete with definitions and notational complexities. We have “Mother” and “Father” wavelets, scaling functions, and so on. Table 2.3 summarizes some of these terms and give pointers to where in this article they are defined.

3.1 Multi-resolution analysis - heuristics

Before starting on this subject, look at figure 2, which is simply a more general version of figure 1. In this figure, 16 points are depicted as being transformed, and two levels of the transform are being shown. Look at the end result, the line labelled “MRA”. The first four points S_2 are the level 2 shrunken version of the level 1 data S_1. The next four are the detail data D_2 for level 2. This goes with S_2 and is what as left over when S_1 was shrunk to S_2. The last eight data points of MRA are the detail data D_1 that went with S_1 when S_0 was transformed. We could write this level 2 transformed data set as

$$MRA_L(k) = \{z_{L,1},..,z_{L,4},e_{L,1},..,e_{L,4},d_{L-1,1},..,d_{L-1,8},..\}, \quad L = 2. \quad (9)$$

Figure 2 depicts this transformed data array.

What is important is that the shrunk data sets $S_0, S_1, S_2, \ldots$ are transformed, "downsized", for want ofa better term, versions of the original data, using rescaled versions of the same scaling functions which we can denote by $\phi_L(x)$. Likewise the detail data sets $D_1, D_2, \ldots$ are transformed versions of the original data, using rescaled versions of the same detail, or analysis, functions which we can denote by $\psi_L(x)$. It would be ideal if the components S_L and D_L were orthogonal. This would be achieved if the functions $\phi_L(x)$ and $\psi_L(x)$ were orthogonal.

Focus on the $L^{\text{th.}}$ level and let the corresponding scaling and detail basis functions be $\phi_L(x)$ and $\psi_L(x)$. We might hazard a guess then that the original data $f(x)$ can be written in terms of what we have at the $L^{\text{th.}}$ level as a sum

$$\underbrace{f(x)}_{\text{data}} = \underbrace{\sum_k s_L \phi_{L,k}(x)}_{\text{downsized}} + \underbrace{\sum_k}_{\text{levels}} \underbrace{\sum_{j\leq L} \overbrace{d_{j,k}}^{\text{wavelet coefficients}} \psi_{j,k}(x)}_{\text{details}} \tag{10}$$

The detail data involves all levels $0, \ldots, L$, so basis functions $\psi_k(x)$ at all levels are involved. Only one shrunk set of data is present in $MRA_L(x)$, that is S_L and that is described only by the scaling functions $\phi_L(x)$ appropriate to that level.

If the number of data points is a power of 2, going to as many levels as possible reduces the first sum to a single term, a constant. This is the full wavelet transform of the original data. It should be noted in passing that equation (10) may not be the best way of computing the transform since the double sum may be inefficient to calculate as it stands. A large part of wavelet technology is devoted to making these transforms more efficient: as in Fourier analysis there is a fast Fourier transform, in wavelet analysis there is a fast wavelet transform.

At this point we need to know what are suitable sets of scaling functions $\phi_L(x)$ and wavelet functions $\psi_L(x)$.

3.2 The Scaling Function

Formally, the *scaling function* $\phi(x)$ is a, possibly complex, square integrable function of a real argument x that is normalised so that

$$\int \phi(x)dx = 1 \tag{11}$$

$$\int |\phi(x)|^2 dx = 1 \tag{12}$$

It has the additional property that its translates

$$\phi_k = \{\phi(x-k)\} \tag{13}$$

form an orthonormal set:

$$< \phi_k, \phi_m >= \int_{-\infty}^{\infty} \phi(x-k)\phi(x-m)dx = \delta_{k,m} \tag{14}$$

We use $< \phi_k, \phi_m >$ to denote the scalar product.

Orthogonality makes the functions useful since they can be used to represent any function on the interval where they live. In particular the function $\phi(x)$ itself has representation of the form

$$\phi(x) = \sqrt{2} \sum_{k=0}^{k=L-1} h_k \phi(2x-k) \tag{15}$$

for some L and some set of L coefficients h_k. The value of L is related to the number of vanishing moments of the function $\phi(x)$.

The h_k are referred to as the *refinement coefficients*. The $\sqrt{2}$ factor is conventional in theoretical work, though in computational work it may be dropped in order to speed up calculations by removing non-integer factors. This equation, called the *dilation equation*, expresses the original function $\phi(x)$ in terms of the same function scaled by a factor two. The scaling function is occasionally referred to as the *Father Wavelet*.

Functions satisfying this kind of equation are called m-refinable functions.

For notatinal convenience we will occasionally use the symbol $\phi_{n,k}(x)$ to denote the rescaled and shifted version of $\phi(x)$:

$$\phi_{n,k}(x) = \phi(2^n x - k) \tag{16}$$

and if the value of k is not pertinent we may shorten this to $\phi_n(x)$.

From (15) we can express the refinement coefficients h_k in terms of the scaling function using orthogonality of the $\phi(2x-k)$:

$$h_k = \sqrt{2} \int \phi(x)\phi(2x-k)dx, \quad k = 0, ..., L-1 \tag{17}$$

There are L of these coefficients. If $\phi(x)$ were a complex valued function, we would use its complex conjugate $\bar{\phi}(x)$ in the integrand.

The condition (11) taken with the dilation equation (15) imposes a condition on the sum of the h_k:

$$\sum_{k}^{L-1} h_k = \sqrt{2} \tag{18}$$

Multiplying equation (15) by $\phi(x+p)$, integrating and using equation (17) yields the very important orthogonailty relationship for the refinement coefficients h_k:

$$\sum_{k}^{L-1} h_k h_{k+2p} = \delta_{0,p} \tag{19}$$

When $p = 0$ this shows that the sum of the squares of the h_k is unity: hence the reason for the $\sqrt{2}$ normalization in the dilation equation (15). This relationship will play an important role later on: we shall see how it works on data sets.

The refinement coefficients h_k provide the all-important link between the wavelet scaling function $\phi(x)$ and the discrete version of the wavelet transform. Indeed, it is often the case that we know the h_k but have no analytic expression for the function $\phi(x)$. We can, in effect, do wavelet analysis without ever knowing what the wavelets are.

3.3 The Mother Wavelet

We can now construct sets of orthogonal functions $\psi_k(x)$ that are the translations $\psi_k(x) = \psi(x-k)$ of some function $\psi(x)$, and which are orthogonal to the translates $\phi_k(x) = \phi(x-k)$ of the scaling function $\phi(x)$:

$$< \phi_k(x), \psi_n(x) >= \int \phi(x-k)\psi(x-n)dx0. \tag{20}$$

We shall insist on a few conditions. Firstly that the function $\psi(x)$ have mean zero:

$$\int_{-\infty}^{\infty} \psi(x)dx = 0. \tag{21}$$

In other words, $\psi(x)$ must be oscillatory with mean value zero. It is the oscillatory nature of these functions that led to the name "wavelet". Furthermore $\psi(x)$ should be square integrable:

$$\int |\psi(x)|^2 dx = 1 \tag{22}$$

(this assures us that the function is not everywhere zero!) and its translates

$$\{\psi(x-k)\} \tag{23}$$

should form an orthonormal basis. This latter condition means that the function $\psi(mx)$ can be expressed as a linear sum of the $\psi(x-k)$. Both these last conditions exclude the Fourier transform basis $e^{i\omega x}$ from the class of wavelets.

This definition is rather general and it is usual to add further conditions that make the functions useful in terms of data analysis. One important such condition is the "*admissability condition*" which imposes a constraint on the Fourier transform of $\psi(x)$: the Fourier transform of $\psi(x)$

$$\Psi(k) = \int_{-\infty}^{\infty} \psi(x) e^{-2\pi i k x} dx \tag{24}$$

should satisfy Calderón's admissibility condition

$$0 < \int_0^{\infty} \frac{|\Psi(k)|^2}{k} dk < \infty \tag{25}$$

This ensures that the wavelet transform of a function should be reversible: we can reconstruct the function from its transform.

Since the $\phi(x)$ are an orthonormal set, there is a set of coefficients g_k such that

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \phi(2x - k) \tag{26}$$

for the same L as in the dilation equation (15). The set $\{g_k\}$ is referred to as the *scaling filter*. By analogy with (16) we can introduce the notation

$$\psi_{n,k}(x) = \psi(2^n x - k) \tag{27}$$

and if the value of k is not pertinent we may shorten this to $\psi_n(x)$.

Using the orthogonality of $\phi(x)$ we find that

$$g_k = \sqrt{2} \int \phi(x) \psi(2x - k) dx, \quad k = 0, ..., L-1 \tag{28}$$

As with the h_k, we may know the g_k without explicitly knowing the function $\psi(x)$.

Substituting equations (15) and (26) into (20) yields

$$< \phi(x), \psi(x) >= \sum_{k=0}^{L-1} h_k g_k = 0 \tag{29}$$

Likewise, we can use the orthogonality of the translates $\psi_k(x) = \psi(x - k)$ to obtain another condition on the g_k:

$$< \psi(x - k), \psi(x - m) >= \sum_i g_i g_{i-2(k-m)} = \delta_{k,m} \tag{30}$$

Equation (21) telling us that the mean of the mother wavelet function is zero implies a condition on the g_k:

$$\sum_i g_i = 0 \tag{31}$$

This follows from (26) and (11).

Equations (29) and (30) have an important solution, referred to as the *quadrature mirror relationship*:

$$g_k = (-1)^k h_{L-k-1} \tag{32}$$

This is discussed in section 10.2 (see equation (129)). So we see that the refinement coefficients h_k completely define the transforms. The scaling function $\phi(x)$ and the mother wavelet $\psi(x)$ allow continuous wavelet transforms, while the refinement coefficients h_k allow discrete wavelet transforms.

3.4 The Continuous Wavelet Transform

The wavelet transform of a function $f(x)$ is just the convolution of the function with the wavelet. Given a function $f(x)$ and a wavelet $\psi(x)$ we may be able to construct, for any a and b, the localised scaling transforms $F(a,b)$ of $f(x)$ by evaluating the integrals

$$F(a,b) = \frac{1}{\sqrt{a}} \int_0^\infty f(x)\psi\left(\frac{x-b}{a}\right) dx \tag{33}$$

We would like this transform to be invertible so that $f(x)$ can be reconstructed from $F(a,b)$ by using the same function $\psi(x)$ as follows:

$$f(x) = \frac{1}{C} \int_{-\infty}^\infty \int_0^\infty F(a,b) \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \frac{dadb}{a^2} \tag{34}$$

for some finite normalization constant C. The requirement that C be finite is equivalent to the constraint (25).

The continuous wavelet transform maps a function of one variable into a function of two variables, scale and location: it contains a lot of redundant information. We can see a as providing a microscope that allows us to zoom in on what is happening in the vicinity of a point b. It is the statement "... in the vicinity of a point b" that is important since this is quite unlike the Fourier Transform which merely describes an average behaviour on all scales.

If at some point b, we see the value of $F(a,b)$ rise as we make a smaller, we would surmise that we were looking at some kind of spike or even a singularity in the function $f(x)$. This might be construed as evidence of noise. On the other hand, if we saw scaling behaviour for ever larger values of a we might surmise that there was structure in the signal at this point.

What we are saying is that the structure of the data is reflected in the behaviour of its singularities, and that the Continuous Wavelet Transfrom is a good tool for getting at this. The fact that the continuous wavelet transform leads to a description of local singularities in the signal structure further suggests that the Continuous Wavelet Transform is somehow related to the Multifractal description of the data (see section 11.10).

3.5 Vanishing moments

The m^{th} moment of the wavelet function $\psi(x)$ is defined to be

$$\mathcal{M}_m = \int_{-\infty}^{\infty} x^m \psi(x) dx \tag{35}$$

Daubechies ([17]) argued that good filters should have vanishing moments, i.e. $\mathcal{M}_m = 0$ for $m = 0, \ldots, r-1$ for some positive integer r:

$$\int_{-\infty}^{\infty} x^m \psi(x) dx = 0, \quad \forall m : 0 \leq m < r \tag{36}$$

Under these circumstances $\psi(x)$ is said to be orthogonal to all polynomials of order up to degree r. Let me emphasise that (36) is an option.

If $\psi(x)$ and $r-1$ of its derivatives are continuous everywhere, then it has r vanishing moments (the converse is not true). Continuity of $\psi(x)$ and its derivatives is one ingredient in minimizing artefacts in signal analysis that might be due to $\psi(x)$ itself rather than the data. It is always a good idea to have the first moment (the mean) vanish since that conserves the energy in the signal. We have seen (section 2.3) that simple interpolating filters do not have this property: we shall see how to correct this by using the lifting process.

However, there is no free lunch: the rate of convergence of the wavelet approximation as a function of levels used is lessened as the number of vanishing moments increases. It is then pertinent to ask what is the best wavelet to use from the point of view of vanishing moments and speed of convergence. The answer, provided by Daubechies, is the wavelet system known as "Coiflets"(section 5.3).

Imposing the zero-moment condition of equation (36) leads to additional constraints on the values of the h_k:

$$\sum_{k=0}^{r-1} h_k k^m = 0, \quad 0 < m < r. \tag{37}$$

Imposing this condition improves the match between the original data values and the smoothed, transformed, values by fitting a higher order polynomial to sets of filtered points. This leads to a family of wavelets called Coiflets.

3.6 Representing data

We now turn to the all-important issue of how to repesent data in terms of wavelets, and how to invert the transform so as to recover the data. This is technically quite complex, but it is central to the issue of generating and using wavelet transforms and their inverse.

Consider a function $f(x)$. It can be viewed at two neighbouring resolutions and represented in the following two ways:

$$f(x) = \sum_k s_j[k]\phi(2^j x - k) \tag{38}$$

$$= \sum_k s_{j-1}[k]\phi(2^{j-1}x - k) + \sum_k d_{j-1}[k]\psi(2^{j-1}x - k) \tag{39}$$

The first of these is the view from what might be called "level j" and the second is the view from "level $j-1$". At level j we are seeing the function in all its detail: it is described by the coefficient set $s_j[k]$. At level $j-1$ we have split it into a smooth part described by the coefficient set $s_{j-1}[k]$ and a detail part described by the coefficient set $d_{j-1}[k]$.

Multiply these equations by $\phi(2^{j-1}x - k)$ and integrate, using the orthogonality relations (14) and the definitions of the refinement coefficients (17):

$$s_{j-1}[m] = \sum_k s_j[k]h_{k-2m} \tag{40}$$

Likewise, integrating with $\psi(2^{j-1}x - k)$ and using (28) gives

$$d_{j-1}[m] = \sum_k s_j[k]g_{k-2m} \tag{41}$$

We can rewrite the indices on the right-hand sides to give:

$$s_{j-1}[m] = \sum_n h_n s_j[n + 2m] \tag{42}$$

$$d_{j-1}[m] = \sum_n g_n s_j[n + 2m] \tag{43}$$

So $s_{j-1}[m]$ is derived from $s_j[2m]$ and its neighbours, with weightings given by the h_n. (Note that some authors replace n by $-n$ and redefine h_n and g_n accordingly so that these equations look more like a convolution).

The fact that the $s_j[2m]$ and its neighbours are used introduces a subtelty into the inversion of these equations, we shall come to that below.

With these equations, the coefficient sets $s_j[k]$ and $d_j[k]$ can be calculated recursively. It is the essence of the fast wavelet transform. The recursion is depicted in Figure 2. The starting point for the recursion is the original data $f(x)$ sampled at discrete points:

$$s_0[k] = f(x)\delta(x - k) \tag{44}$$

Since data is usually presented that way, this does not pose a problem. However, if the analytic form of the function $f(x)$ is given, this choice for the initial samples $s_0[k]$ would not be unique: it would depend on the binning. Likewise, if $f(x)$ were defined on some irregular point set, the values of the

$s_0[k]$ would depend on the interpolation used to create a regularly spaced data set.

The recursive sums (42) and (43) are generally far easier to compute than integrals like

$$s_j[k] \simeq \int f(x)\phi(2^j x - k)dx \tag{45}$$

$$d_j[k] \simeq \int f(x)\psi(2^j x - k)dx \tag{46}$$

(normalization factors have been omitted). Moreover, recall that we might not even know what the function $\phi(x)$ is, though we may know the coefficient set $\{h_k\}$!

3.7 Orthogonality revisited - reverse transforms

It is worth recapitulating on the various relationships that have been used in performing wavelet transforms and their inverses. We introduced a scaling function $\phi(x)$ which was associated with a set of refinement coefficients $\{h_k\}$ (the low-pass filter). The corresponding wavelet $\psi(x)$ was associated with a set of filter coefficients $\{g_k\}$ (the high-pass filter) related to the $\{h_k\}$ via the quadrature mirror relationship (32).

We have seen how to go from data to a representation in terms of wavelet coefficients (equations (42) and (43)). We have not said how to go backwards from these coefficients and recover $f(x)$ from the $s_j[k]$ and $d_j[k]$. Equation (39) does this, but it is expressed in terms of the functions $\phi(x)$ and $\psi(x)$, which we might not know explicitly. We would rather have an expression explicitly involving filters like the $\{h_k\}$ and $\{g_k\}$.

The analogue of equation (39) that suggests itself is

$$s_j[k] = \sum_n \bar{h}_{k-2n} s_{j-1}[n] + \sum_n \bar{g}_{k-2n} d_{j-1}[n] \tag{47}$$

for some coefficient sets $\{\bar{h}_k\}$ and $\{\bar{g}_k\}$ as yet to be determined. This can be done by imposing orthogonality conditions on the coefficient sets $\{h_k\}$ and $\{g_k\}$, $\{\bar{h}_k\}$ and $\{\bar{g}_k\}$, and between the barred and unbarred coefficient sets. Such coefficient sets are referred to as *bi-orthogonal*.

An important example of such coefficient sets is the 5-3 wavelet that we shall come to in a different context later:

$$\{h_k\} = \tfrac{\sqrt{2}}{8}\{-1, 2, 6, 2, -1\} \quad \{g_k\} = \tfrac{\sqrt{2}}{4}\{-1, -2, -1\} \tag{48}$$

$$\{\bar{h}_k\} = \tfrac{\sqrt{2}}{4}\{1, 2, 1\} \quad \{\bar{g}_k\} = \tfrac{\sqrt{2}}{8}\{-1, -2, 6, -2, -1\} \tag{49}$$

The way in which the coefficients for the inverse transform are used is not at first sight obvious since each of the forward transformations reduces the size of the data by a factor two. We see this in equation (47) where the summation on the right-hand side involves $\bar{h}_{2n}$ coupling with the data values $s[n]$. We can show this if we write out the inverse transformation for the coefficient set (48) for even and odd k:

$$s_j[2n] = s_{j-1}[n] - \tfrac{1}{4}\{d_{j-1}[n-1] + d_{j-1}[n]\} \tag{50}$$

$$\begin{aligned} s_j[2n+1] = \tfrac{1}{2}\{s_{j-1}[n] + s_{j-1}[n+1]\} + \\ \tfrac{1}{8}\{-d_{j-1}[n-1] + 6d_{j-1}[n] - d_{j-1}[n+1]\} \end{aligned} \tag{51}$$

We see how the various terms from (47) pick out different components of the $\{\bar{h}_k\}$ and $\{\bar{g}_k\}$ in (48) and (49) for even or odd k.

We can impose a further, more restrictive, condition on the coefficients:

$$\bar{h}_k = g_{-k} \tag{52}$$

$$\bar{g}_k = h_{-k} \tag{53}$$

Wavelet coefficient sets satisfying these conditions are referred to as *orthogonal*. They must have an even number of coefficients. This symmetry means that the forward and backward transforms involve the same coefficients.

4 Famous scaling functions and wavelets

4.1 Haar

The simplest wavelet is the Haar Wavelet whose scaling function is

$$\phi_{Haar}(x) = \begin{cases} 1 & \text{if } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

It can be readuily verified that it satisfies the dilation equation

$$\phi_{Haar}(x) = \phi_{Haar}(2x) + \phi_{Haar}(2x-1) \tag{55}$$

There are just two terms (i.e: $L = 2$ in equation (15)) and the coefficients h_i are

$$\begin{aligned} h_0 &= 1/\sqrt{2} \\ h_1 &= 1/\sqrt{2} \end{aligned} \tag{56}$$

The Haar scaling function has compact support but is discontinuous. This is the wavelet used in the simple example of section 2.1.

The Haar Mother wavelet is

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \le x < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{57}$$

This can be derived from equations (15) and (32).

4.2 Hat, or Tent, Scaling function

The scaling function is very simple

$$\phi_{Tent}(x) = \begin{cases} x & 0 \le x < 1 \\ 2 - x & 1 \le x < 2 \\ 0 & \text{otherwise} \end{cases} \tag{58}$$

and is a member of an important family of wavelets ("CDF(2,2)"). The dilation equation can be derived by expressing the tent function in terms of the Haar function:

$$\phi_{Tent}(x) = \int_0^x [\phi_{Haar}(u) - \phi_{Haar}(u-1)]\, du \tag{59}$$

and using the Haar dilation equation (55) to replace u in the integrand with $2u$. It is then easily found that

$$\phi_{Tent}(x) = \tfrac{1}{2}\left[\phi_{Tent}(2x) + 2\phi_{Tent}(2x-1) + \phi_{Tent}(2x-2)\right] \tag{60}$$

The coefficient set for this hat function follows from equation (15)

$$\tfrac{1}{\sqrt{2}}\{\tfrac{1}{2}, 1, \tfrac{1}{2}\} \tag{61}$$

This has compact support and is symmetric. It is the coefficient set $\{\bar{h}_k\}$ of equation (49).

4.3 Quadratic Battle-Lemarié wavelet

These wavelets are based on B-splines. The scaling function is bell-shaped with compact support:

$$\phi_{BM2}(x) = \begin{cases} \tfrac{1}{2}x^2 & 0 \le x < 1, \\ -x^2 + 3x - \tfrac{3}{2} & 1 \le x < 2, \\ \tfrac{1}{2}(x-3)^2 & 2 \le x < 3, \\ 0 & \text{otherwise.} \end{cases} \tag{62}$$

and satisfies the dilation equation:

$$\phi_{BM2}(x) = \\ \tfrac{1}{4}\phi_{BM2}(2x) + \tfrac{3}{4}\phi_{BM2}(2x-1) + \tfrac{3}{4}\phi_{BM2}(2x-2) + \tfrac{1}{4}\phi_{BM2}(2x-3) \tag{63}$$

This equation can be derived by expressing $\phi_{BM2}(x)$ in terms of $\phi_{Tent}(x)$:

$$\phi_{BM2}(x) = \int_0^x [\phi_{Tent}(u) - \phi_{Tent}(u-1)]\, du \tag{64}$$

and using scaling relationships for $\phi_{Tent}(u)$ to express the integrand in terms of $\phi_{Tent}(2u)$. A simple change of integration variable $2u \to v$ then yields (63).

The cubic Battle-Lemarié wavelet can be similarly derived and has scaling function

$$\phi_{BM3}(x) = \frac{1}{8}\phi_{BM3}(2x)+\frac{1}{2}\phi_{BM3}(2x-1)+\frac{3}{4}\phi_{BM3}(2x-2)+\frac{1}{2}\phi_{BM3}(2x-3)+\frac{1}{8}\phi_{BM3}(2x-4) \tag{65}$$

4.4 Shannon scaling function

This scaling function does not have compact support, but it is smooth: all derivatives exist and are continuous:

$$\phi(x) = \begin{cases} \dfrac{\sin(\pi t)}{\pi t} & t \neq 0, \\ 1 & t = 0. \end{cases} \tag{66}$$

The corresponding wavelet function $\psi(x)$, sometimes called a *sinclet*, can be shown to be

$$\psi(x) = \begin{cases} \dfrac{\sin 2\pi t - \sin(\pi t)}{\pi t} & t \neq 0, \\ 1 & t = 0. \end{cases} \tag{67}$$

and the refinement coefficients are

$$\begin{aligned} h_0 &= g_0 = \frac{\sqrt{2}}{2} \\ h_{2n} &= g_{2n} = 0 \\ h_{2n+1} &= -g_{2n+1} = (-1)^n \frac{\sqrt{2}}{(2n+1)\pi} \end{aligned} \tag{68}$$

Since the wavelet does not have finite support there is an infinite number of these coefficients.

A signal $s(t)$ is referred to as being "band-limited" if its Fourier transform has compact support. In practical terms this means that the Fourier transform of the signal is zero for frequencies ω higher than some cut-off frequency: $|\omega| > \sigma$. A band-limited signal can be properly sampled at the *Nyquist interval* $T = \pi/\sigma$ by giving its values at the points nT $(-\infty < n < \infty)$. This is the essence of the Shannon sampling theorem.

Such a band-limited signal has a representation of the form:

$$s(t+\tau) = \sum_{n=-\infty}^{\infty} s(t+nT)\frac{\sin\sigma(\tau - nT)}{\sigma(\tau - nT)}, \quad T = \frac{\pi}{\sigma} \tag{69}$$

Putting $T = 2^{-(m+1)}$, this can be re-expressed in terms of the Shannon scaling function (66) as

$$s(t+\tau) = \sum_{n=-\infty}^{\infty} s\left(t + \frac{n}{2^{m+1}}\right) \phi_{Shannon}(2^{m+1}\tau - n) \tag{70}$$

for all t and τ. In practical terms we can never know the entire past and future of the signal, as is implicit in this representation. See Papoulis [4] section 11-5 for a thorough discussion of this.

4.5 Mexican Hat, or Marr, wavelet

The Mexican Hat wavelet, so-called because of its appearance, is perhaps the most popularly used wavelet among the set of continuous wavelet transforms (though there is no particular scientific reason why it should be so popular). Marr [32] recognized the physical significance of this for the human visual system. There is no simple form for the scaling function, but the mother wavelet is

$$\psi(x) = \frac{2}{\pi^{1/4}\sqrt{3}\sigma}(1 - \frac{x^2}{\sigma^2})e^{-x^2/2\sigma^2} \tag{71}$$

This function is just the second derivative of a Gaussian. Note that this function does not have compact support.

The Mexican Hat function can usefully be approximated using

$$\begin{aligned} G(x;\sigma) &= \frac{1}{\sqrt{2\pi}\sigma}e^{-x^2/2\sigma^2} \\ H(x;\sigma) = G''(x;\sigma) &\approx \alpha G(\alpha x;\sigma) - G(x;\sigma), \quad \alpha \simeq 1.6 \end{aligned} \tag{72}$$

This shows that the wavelet is approximately the difference between Gaussians of different widths.

4.6 Morlet Wavelet

The Morlet wavelet is a complex exponential whose amplitude is modulated by a Gaussian:

$$\psi(x) = \pi^{-1/4}e^{-i\omega x}e^{-x^2} \tag{73}$$

Since this function does not integrate to zero it is not admissable. Nonetheless, for frequencies $\omega > 10$ or so, the integral is so small that approximate reconstruction is satisfactory.

There are two scales in the Morlet wavelet: the width of the wave packet and the frequency. This can be made more explicit by rewriting (73) as

$$\psi(x) = \sigma^{-1/2}e^{-i\omega x\sigma}e^{-(x/\sigma)^2} \tag{74}$$

(up to a proportionality constant). The width parameter is then σ and the frequency parameter is ω. In this form the wavelet is sometimes referred to as the Gabor Wavelet since it follows the spirit of the Gabor transformation (123).

A modification of the Morlet wavelet that is admissable is

$$\psi(x) = Ce^{-i\omega x}e^{-x^2}(1 - \sqrt{2}e^{-\omega^2/4}) \tag{75}$$

where the normalization constant C depends on ω.

5 Famous wavelet filters

5.1 Daubechies 4-point

After the Haar wavelet, the most famous wavelet is arguably the first member of a huge class of wavelets discovered by Daubechies. For $L = 4$, conditions (18) and (19) are simply

$$\begin{aligned} h_1 + h_2 + h_3 + h_4 &= \sqrt{2} \\ h_1^2 + h_2^2 + h_3^2 + h_4^2 &= 1 \\ h_1h_3 + h_2h_4 &= 0 \end{aligned} \tag{76}$$

This is only three equations for the four h_k, so we can impose a further condition.

$$0g_1 + 1g_2 + 2g_3 + 3g_4 = 0 \tag{77}$$

which expresses a zero-first-moment relationship among the wavelet coefficients g_k. Using $g_k = (-1)^k h_{L-k-1}$ (see equation(32)) we get

$$0h_4 - 1h_3 + 2h_2 - 3h_1 = 0 \tag{78}$$

Solving these four equations for h_k, we find that this four-point transform has coefficients

$$h_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{79}$$

It is perhaps surprising that these equations have a closed form expression.

5.2 Daubechies 6-point

For $L = 6$, conditions (18) and (19) are simply

$$\begin{aligned} h_1 + h_2 + h_3 + h_4 + h_5 + h_6 &= \sqrt{2} \\ h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 &= 1 \\ h_1h_3 + h_2h_4 + h_3h_5 + h_4h_6 &= 0 \\ g_1 + g_2 + g_3 + g_4 + g_5 + g_6 &= 0 \end{aligned} \tag{80}$$

We can use $g_k = (-1)^k h_{L-k-1}$ to replace the g_k with h_k's in the last of these. This is only four equations for the six h_k, so we can impose two further conditions on the moments of the g_k:

$$\begin{aligned} 0g_1 + 1g_2 + 2g_3 + 3g_4 + 4g_5 + 5g_6 &= 0 \\ 0^2 g_1 + 1^2 g_2 + 2^2 g_3 + 3^2 g_4 + 4^2 g_5 + 5^2 g_6 &= 0 \end{aligned} \tag{81}$$

and again we can replace g_k's with h_k's.

Amazingly, these too have a solution that is expressible in terms of surds. It can be found in several places (including Numerical Recipes [5]), but for completeness here it is:

$$\begin{aligned} h_1 &= (1 + \sqrt{10} + \sqrt{5 + 2\sqrt{10}})/14\sqrt{2} & h_2 &= (5 + \sqrt{10} + 3\sqrt{5 + 2\sqrt{10}})/14\sqrt{2} \\ h_3 &= (10 - 2\sqrt{10} + 2\sqrt{5 + 2\sqrt{10}})/14\sqrt{2} & h_4 &= (10 - 2\sqrt{10} - 2\sqrt{5 + 2\sqrt{10}})/14\sqrt{2} \\ h_5 &= (5 + \sqrt{10} - 3\sqrt{5 + 2\sqrt{10}})/14\sqrt{2} & h_6 &= (1 + \sqrt{10} - \sqrt{5 + 2\sqrt{10}})/14\sqrt{2} \end{aligned} \tag{82}$$

5.3 Coiflet 6-point

The Coiflets are wavelets designed so that all moments vanish, so there are additional constraints. As before it is necessary to satisfy the relationships

$$\begin{aligned} h_1 + h_2 + h_3 + h_4 + h_5 + h_6 &= \sqrt{2} \\ h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 &= 1 \\ g_1 + g_2 + g_3 + g_4 + g_5 + g_6 &= 0 \end{aligned} \tag{83}$$

(compare equations (80)). We can use $g_k = (-1)^k h_{L-k-1}$ to replace the g_k with h_k's in the last of these. The additional constraints are:

$$\begin{aligned} 0g_1 + 1g_2 + 2g_3 + 3g_4 + 4g_5 + 5g_6 &= 0 \\ -2h_1 - 1h_2 + 0h_3 + 1h_4 + 2h_5 + 3h_6 &= 0 \\ (-2)^2 h_1 + (-1)^2 h_2 + 0^2 h_3 + 1^2 h_4 + 2^2 h_5 + 3^2 h_6 &= 0 \end{aligned} \tag{84}$$

(compare equations (81)). Solving these leads to the 6-point Coiflet coefficients

$$h_1 = \frac{1-\sqrt{7}}{16\sqrt{2}}, \quad h_2 = \frac{5+\sqrt{7}}{16\sqrt{2}}, \quad h_3 = \frac{14+2\sqrt{7}}{16\sqrt{2}},$$
$$h_4 = \frac{14-2\sqrt{7}}{16\sqrt{2}}, \quad h_5 = \frac{1-\sqrt{7}}{16\sqrt{2}}, \quad h_6 = \frac{-3+\sqrt{7}}{16\sqrt{2}} \tag{85}$$

Again, it is remarkable that these are expressable in closed form.

5.4 Deslaurier-Dubuc Wavelet

Deslauriers and Dubuc provided an important method for interpolating functions defined at integer points by a process of successive refinement at binary rational points [22] [23]. The interpolating function is continuous at the rational interpolating points. Given values of the function at integer points, $f(x-nh)$, we can interpolate the value of $f(x)$ from its neighbours using the simple interpolation

$$f(x) = \tfrac{9}{16}[f(x-h)+f(x+h)] - \tfrac{1}{16}[f(x-3h)+f(x+3h)] \tag{86}$$

This is just Lagrange four-point interpolation to the mid-point.

Successive refinement at binary rational points gives rise to a series of wavelet transforms. This four-point interpolation leads to the transformation

$$\begin{aligned} d_{j-1}[k] &= x_j[2k+1] \\ &\quad + \tfrac{1}{16}(-x_j[2k-2] + 9x_j[2k] + 9x_j[2k+2] - x_j[2k+4]) \\ s_{j-1}[k] &= x_j[2k] \\ &\quad - \tfrac{1}{32}(-d_{j-1}[k-2] + 9d_{j-1}[k-1] + 9d_{j-1}[k] - d_{j-1}[k+1]) \end{aligned} \tag{87}$$

This equation for the wavelet transform is not expressed directly in terms of the coefficients h_k as were the previous examples. Rather, it has been written in the form that would be used in computation of the transform. The calculation of $s_{j-1}[k]$ involves the previously calculated $d_{j-1}[k]$ instead of the function values at the data points. This, as we shall see, is because (87) has been derived from (86) via the Lifting process.

This is an interpolating wavelet (by construction!). The manner of its definition at binary rational points means that this is somewhat different from usual wavelets in that it is not related to a square-integrable function. However, that does not make it any less useful.

The important generalization of equation (86) introduced by Deslauriers and Dubuc [23] was to write

$$f(x) = \sum_{k \in Z} F(\tfrac{k}{2}) f(x+kh) \tag{88}$$

where the weighting function $F(x)$ satisfies the relationship

$$F(\tfrac{x}{2}) = \sum_{k \in Z} F(\tfrac{k}{2}) F(x-k) \tag{89}$$

With this the interpolated version of the original function $f(x)$ can be written as the convolution

$$f(x) = \sum_{k \in Z} f(x) F(x-k) \quad x \in R \tag{90}$$

For $F(x)$ Deslauriers and Dubuc selected the family $\mathcal{L}^L_{2m-1}(x)$ of Lagrange polynomials of degree $L-1$ with $M = 2L$ zeros at $-L+1, -L+3, \dots, L-1$. The general Lagrange polynomials of degree N having zeros on a set of points $\{x_k\}$ can be written

$$\mathcal{L}^N_i(x) = \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_N)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_N)}, \quad i = 0,\dots,N \tag{91}$$

where the factor $(x-x_i)$ is missing from the numerator. So in the Deslauriers and Dubuc case we have

$$F(x): \quad \mathcal{L}^L_{2m-1}(x) = \prod_{l=-M+1, l\neq m}^{M} \frac{x-(2l-1)}{(2m-1)-(2l-1)}, \quad L = 2M \tag{92}$$

and then

$$f(x) = \sum_{k=-M+1}^{M} \mathcal{L}^L_{2k-1}(0) f(x+(2k-1)h), \quad L = 2M \tag{93}$$

which, considering what went before, is remarkably simple. The case $L = 4$ is commonly used: this is the transform given in equation (87).

6 The Matrix View

it is useful to express the wavelet transform as a matrix operating on a data vector. This can provide some useful insights.

6.1 The 4-point orthogonal transform

We can illustrate the use of these coefficients as a simple matrix multiplying a column vector of $2N$ data points. We have the following relationships among the four refinement coefficients h_k:

$$\begin{aligned} h_1 + h_2 + h_3 + h_4 &= \sqrt{2} \\ h_1^2 + h_2^2 + h_3^2 + h_4^2 &= 1 \\ h_1 h_3 + h_2 h_4 &= 0 \end{aligned} \tag{94}$$

The first of these comes from the normalization, while the last two come from equation (19). Note that we have three equations constraining the four coefficients h_i: the four-point wavelets are a one-parameter set. This gives rise to the notion of "tunable" wavelets. The corresponding wavelet numbers (equation (32)) are

$$g_1 = h_4, \quad g_2 = -h_3, \quad g_3 = -h_2, \quad g_4 = h_1 \tag{95}$$

For $2N$ data points the transformation matrix is:

$$W^{(2N)} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0\ldots0 & 0 & 0 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0\ldots0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 & 0\ldots0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0\ldots0 & 0 & 0 \\ \multicolumn{9}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ h_3 & h_4 & 0 & 0 & 0 & 0 & 0\ldots0 & h_1 & h_2 \\ g_3 & g_4 & 0 & 0 & 0 & 0 & 0\ldots0 & g_1 & g_2 \end{bmatrix} \tag{96}$$

There are as many columns and rows as there are data points. Note the wrap-around for the last pair of data points. This is a generally undesirable feature unless the data is periodic. There are a number of available strategies for avoiding this problem.

It is easy to verify from equations (94) and (95) that

$$W^T W = I \tag{97}$$

The matrix W is orthogonal: its transpose is its inverse and so doing a wavelet transform of a wavelet transform brings us back to where we started.

If we denote the data by $U^{(2N)}$ and its transform by $V^{(2N)}$ we have

$$V_i^{(2N)} = \sum_{i=1}^{N} W_{ij}^{(2N)} U_j^{(2N)}, \quad i = 1, \ldots, 2N \tag{98}$$

Because of the orthogonality of W, the inverse transform recovering the original data is just the wavelet transform of V.

We note that the entries of the data vector V are an alternating mixture of sum and difference values: all odd numbered entries are created using the h_k coefficients, while all the even numbered entries are generated using the g_k coefficients. It is customary to split the odd and even, (i.e. the sum and difference) parts into two separate vectors, one consisting entirely of entries created by the h_k and the other of entries created by the g_k.

In practise this is achieved by defining two $2N \times N$ matrices $\mathbf{G}$ and $\mathbf{H}$:

$$\mathbf{H}^{(2N\times N)} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0\ldots0 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 & 0\ldots0 & 0 & 0 \\ \multicolumn{9}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ h_3 & h_4 & 0 & 0 & 0 & 0 & 0\ldots0 & h_1 & h_2 \end{bmatrix} \tag{99}$$

$$\mathbf{G}^{(2N\times N))} = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & 0 & 0 & 0\ldots0 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 & 0\ldots0 & 0 & 0 \\ \multicolumn{9}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ g_3 & g_4 & 0 & 0 & 0 & 0 & 0\ldots0 & g_1 & g_2 \end{bmatrix} \tag{100}$$

Note that, by virtue of (95) the rows of $\mathbf{G}$ are orthogonal to one another and by virtue of (94) the rows of $\mathbf{H}$ are orthogonal to one another. The rows

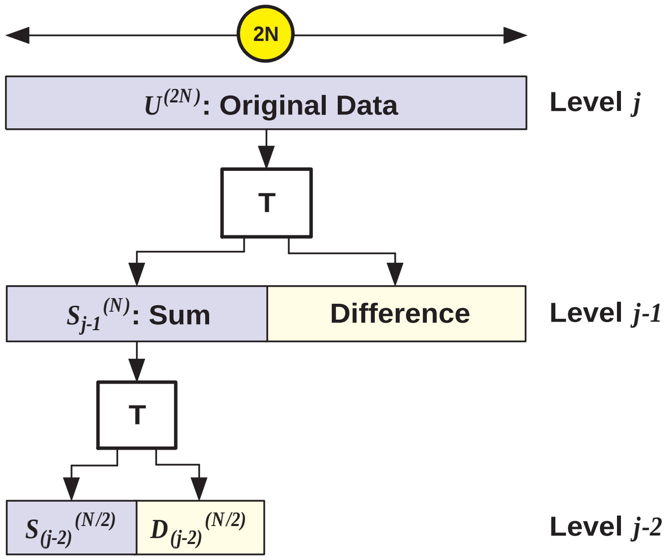

Fig. 3. Discrete Wavelet Transform produces a multi-resolution view of the data by splitting data in a hierarchy of sums and differences

of $\mathbf{G}$ are also orthogonal to the rows of $\mathbf{H}$. We have orthogonality everywhere and such wavelet transforms are referred to a "bi-orthogonal".

We can usefully block the two matrices $\mathbf{G}$ and $\mathbf{H}$ together to form a version of W whose rows have been rearranged:

$$\mathbf{W}^{(2N\times 2N)} = \begin{bmatrix} \mathbf{H} \\ \mathbf{G} \end{bmatrix} \tag{101}$$

The wavelet transform of the $2N$ component data vector $U^{(2N)}$ can then be written

$$\begin{bmatrix} S^{(N)} \\ D^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{G} \end{bmatrix} \begin{bmatrix} U^{(2N)} \end{bmatrix} \tag{102}$$

This differs from (98) only in the organisation of the transformed data coefficients. We have split these coefficients into two sets: a sum $S^{(N)}$ and a difference $D^{(N)}$. The sum is a reduced version of the original data and the differences are what is needed to reconstruct the original data from this sum.

6.2 The multi-resolution view

Having noted that S is a data array of size N, half the size of the original data array, we can proceed to do an N-point wavelet transform of S, producing a yet smaller (lower resolution) representation of the original data:

$$\begin{bmatrix} S_{j-2}^{(N/2)} \\ D_{j-2}^{(N/2)} \end{bmatrix} = \begin{bmatrix} \mathbf{H}^{N\times N/2} \\ \mathbf{G}^{N\times N/2} \end{bmatrix} \begin{bmatrix} S_{j-1}^{(N)} \end{bmatrix} \tag{103}$$

We have explicitly written in the dimensions of the various arrays to emphasise that this is not the same transform that was executed at the first pass.

The end result of this process is a hierarchy of copies of the original data, together with a hierarchy of the difference data that is required to restore one level from the previous (smaller) levels. Each level has half the resolution of its predecessor.

6.3 The 5-3 wavelet again

We have met the 5-3 wavelet in section 3.7 and will meet it yet again in section 7.2 from the point of view of the Lifting process. It is straightforward to write down the matrix versions of this wavelet transform from equation (48) for the forward transform and equations (50) and (51) for the backward transform.

The forward transform is

$$\begin{bmatrix} \dots \\ S[n] \\ D[n] \\ S[n+1] \\ D[n+1] \\ \dots \end{bmatrix} = \begin{bmatrix} \dots\dots\dots\dots\dots\dots\dots\dots \\ -\frac{1}{8} \; \frac{1}{4} \; \boldsymbol{\frac{3}{4}} \; \frac{1}{4} \; -\frac{1}{8} \; 0 \; 0 \; 0 \dots \\ 0 \; 0 \; -\frac{1}{2} \; 1 \; -\frac{1}{2} \; 0 \; 0 \; 0 \dots \\ 0 \; 0 \; -\frac{1}{8} \; \frac{1}{4} \; \frac{3}{4} \; \frac{1}{4} \; -\frac{1}{8} \; 0 \dots \\ 0 \; 0 \; 0 \; 0 \; -\frac{1}{2} \; 1 \; -\frac{1}{2} \; 0 \dots \\ \dots\dots\dots\dots\dots\dots\dots\dots \end{bmatrix} \begin{bmatrix} \dots \\ S[2n-2] \\ S[2n-1] \\ \mathbf{S[2n]} \\ S[2n+1] \\ S[2n+2] \\ \dots \end{bmatrix} \tag{104}$$

Notice the alignment of the elements in the rows and the alignment of the rows and columns. The $\boldsymbol{\frac{3}{4}}$ has been emphasised to show that its horizontal position corresponds with the $\mathbf{S[2n]}$ in the data column.

The backward transform is

$$\begin{bmatrix} \dots \\ S[2n] \\ S[2n+1] \\ S[2n+2] \\ S[2n+3] \\ \dots \end{bmatrix} = \begin{bmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 \; -\frac{1}{4} \; \mathbf{1} \; -\frac{1}{4} \; 0 \; 0 \; 0 \; 0 \dots 0 \; 0 \\ 0 \; -\frac{1}{8} \; \frac{1}{2} \; \frac{3}{4} \; \frac{1}{2} \; -\frac{1}{8} \; 0 \; 0 \dots 0 \; 0 \\ 0 \; 0 \; 0 \; -\frac{1}{4} \; 1 \; -\frac{1}{4} \; 0 \; 0 \dots 0 \; 0 \\ 0 \; 0 \; 0 \; -\frac{1}{8} \; \frac{1}{2} \; \frac{3}{4} \; \frac{1}{2} \; -\frac{1}{8} \dots 0 \; 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \end{bmatrix} \begin{bmatrix} \dots \\ S[n-1] \\ D[n-1] \\ \mathbf{S[n]} \\ D[n] \\ S[n+1] \\ D[n+1] \\ \dots \end{bmatrix} \tag{105}$$

Again, the alignment is important: the **1** in the first row of the matrix must line up with the $\mathbf{S[n]}$ of the data. In that way the $\frac{3}{4}$ of the following row multiplies the $D[n]$.

Notice that, because of the interleaving of the smooth and detail values, this matrix organization is not computationally efficient if the transform is to be done to several levels. The rows and columns of the transform matrices have to be re-organized to handle that.

7 Lifting

We alluded to the so-called "lifting" process in our preamble on wavelet transforms (Section 2.3, equation (8)). There we saw that we could take the values at odd numbered points as the smoothed (downsized) sample values, and use the average of these values as a prediction of what the intermediate values would be. The detail values would then be the difference between the prediction and the actual value.

This is the first part of a three-step process that involves splitting the data into odd and even points, making a prediction on the basis of the even points about the values at the odd points, and then updating the prediction.

In this section we shall change the notation slightly so that, for example, $s[k]$ might refer to the data value at the point k, as opposed to using the functional form $s(x)$.

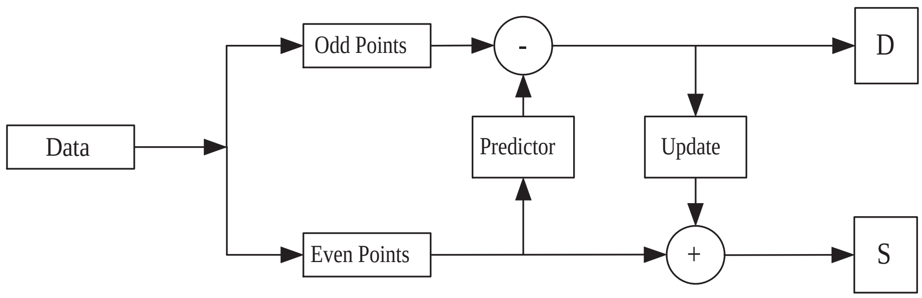

Fig. 4. Lifting strategy

7.1 Split, Predict and Update

If the data points are denoted by $s_j[k]$ at level j, the data $s_{j-1}[k]$ at the next level are:

$$s'_{j-1}[k] = s_j[2k]. \tag{106}$$

The prediction for the point $2k+1$ is based on $s_j[2k]$ and $s_j[2k+2]$:

$$\mu_j[2k+1] = (s_j[2k] + s_j[2k+2])/2. \tag{107}$$

so that the detail coefficient at level $j-1$ is

$$d_{j-1}[k] = s_j[2k+1] - \mu_j[2k+1] = s_j[2k+1] - \tfrac{1}{2}(s_j[2k] + s_j[2k+2]). \quad (108)$$

This recalls what was done in section 2.3. However, we note that with this prescription

$$\sum_k s'_{j-1}[k] \neq \tfrac{1}{2} \sum_k s_j[2k]. \quad (109)$$

The average value of the signal (the energy) is not preserved.

It would have been nicer if the average value of the signal had been preserved. We can fix this by adjusting the $s_{j-1}[k]$ in (106) using the values of the $d_{j-1}[k]$ we have just calculated:

$$s_{j-1}[k] = s_j[2k] - \tfrac{1}{4}(d_{j-1}[k-1] + d_{j-1}[k]). \quad (110)$$

It is easy to verify that now

$$\sum_k s_{j-1}[k] = \tfrac{1}{2} \sum_k s_j[2k]. \quad (111)$$

This adjustment or updating step is an important part of the process.

Although we have used a specific predictor and update, the three-stage process described is quite generic. The data values are split into values at the odd and even points. The even points are used to make a prediction of the value at an intermediate odd point. That is compared with the actual value and the comparison is used to update the the even point data values. Figure 4 summarizes the three important steps.

Inverting the transform is simply a matter of running the diagram backwards. Exactly how this is done will be made clearer in the next section.

7.2 The 5-3 wavelet again

The above process is described by the equations

$$\begin{aligned} d_{j-1}[k] &= s_j[2k+1] - \tfrac{1}{2}(s_j[2k] + s_j[2k+2]). \\ s_{j-1}[k] &= s_j[2k] + \tfrac{1}{4}(d_{j-1}[k-1] + d_{j-1}[k]). \end{aligned} \quad (112)$$

and is described in figure 5. The backwards transform, going from the set of values $s_{j-1}[k]$ and $d_{j-1}[k]$ to recover $s_j[k]$ is obtained by reversing the arrows and changing the signs of the coefficients:

$$\begin{aligned} s_j[2k] &= s_{j-1}[k] - \tfrac{1}{4}(d_{j-1}[k-1] + d_{j-1}[k]) \\ s_j[2k+1] &= d_{j-1}[k] + \tfrac{1}{2}(s_j[2k] + s_j[2k+2]) \end{aligned} \quad (113)$$

This is formally referred to as "CDF(2,2)".

If we unravel the forward transform recursion relation (112), and change the notation slightly for convenience, we can show that

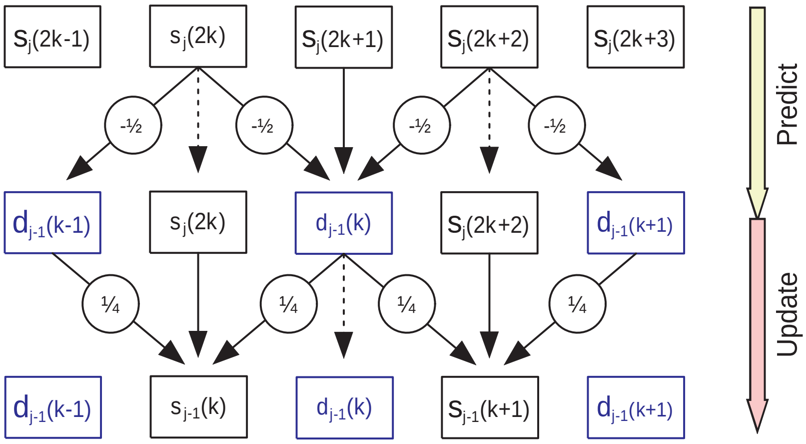

Fig. 5. Wavelet construction by Lifting. The inverse transform is derived by reversing the arrows and changing the signs of the multipliers.

$$\begin{aligned} S[1] &= -\tfrac{1}{8}s[0] + \tfrac{1}{4}s[1] + \tfrac{3}{4}s[2] + \tfrac{1}{4}s[3] - \tfrac{1}{8}s[4] \\ D[1] &= -\tfrac{1}{2}s[2] + s[3] - \tfrac{1}{2}s[4] \end{aligned} \tag{114}$$

In other words the wavelet transform is described by filter coefficients

$$\begin{aligned} \{h\} &= \{-\tfrac{1}{8}, \tfrac{1}{4}, \tfrac{3}{4}, \tfrac{1}{4}, -\tfrac{1}{8}\}, \\ \{g\} &= \{-\tfrac{1}{2}, 1, -\tfrac{1}{2}\} \end{aligned} \tag{115}$$

Up to normalization factors, this is the 5-3 coefficient set encountered in section 3.7. The lifting version (113) of the inverse transform is somewhat easier to appreciate than the direct version of equations (50) and (51).

7.3 9-7 wavelet

This is the wavelet used in the JPEG2000 update of the JPEG image storage format. It is defined through its lifting steps as follows:

$$\begin{aligned} d^{(1)}_{j-1}[k] &= s_j[2k+1] + [\alpha(s_j[2k] + s_j[2k+2]) + \tfrac{1}{2}] \\ s^{(1)}_{j-1}[k] &= s_j[2k] + [\beta(d^{(1)}_{j-1}[k] + d^{(1)}_{j-1}[k-1]) + \tfrac{1}{2}] \\ d_{j-1}[k] &= d^{(1)}_{j-1}[k] + [\gamma(s^{(1)}_{j-1}[k] + s^{(1)}_{j-1}[k+1]) + \tfrac{1}{2}] \\ s_{j-l}[k] &= s^{(1)}_{j-1}[k] + [\delta(d_{j-l}[k] + d_{j-1}[k-1]) + \tfrac{1}{2}] \end{aligned} \tag{116}$$

where the coefficients $\alpha, \beta, \gamma, \delta$ are

$$\begin{aligned} \alpha &= -1.586134342, \quad \beta = -0.05298011854 \\ \gamma &= 0.8829110762, \quad \delta = 0.4435068522 \end{aligned} \tag{117}$$

The intermediate values $d_l^{(1)}, s_l^{(1)}$ are calculated first and then updated. There are 9 coefficients in the smoothing part of the wavelet and 7 in the analysis part, and both parts have four vanishing moments.

The explicit values of the coefficients are given in table 7.3.

k	low-pass	hi-pass
0	+0.6029490182363579	+1.115087052456994
±1	+0.2668641184428723	−0.5912717631142470
±2	−0.07822326652898785	−0.05754352622849957
±3	−0.01686411844287495	+0.09127176311424948
±4	+0.026748741080976	

Table 3. Coefficients for the Daubechies 9-7 wavelet

7.4 Wavelets on irregularly distributed point sets

In one dimension, the cheap way out is to ignore the fact that the data points are irregularly distributed and just calculate as if they were on a regular mesh. This can work, but it is difficult to deal properly and consistently with large gaps in the point distribution.

It is better to use a systematic interpolation method, such as Lagrangian interpolation or spline curves, to fit the data as given, though we are still left wondering whether this actually solves the large gap problem.

The lifting procedure would then suggest that we fit such a curve to all even numbered points, predict the values at the odd points and calculate and store the residuals. If the residuals are small, data compression and reconstruction will be possible. An alternative is to use points for prediction where the sampling frequency is highest rather than every other point regardless of sampling density.

In two or more dimensions this is not so easy since there is no natural extension of the concept of "odd" and "even" points. However, we do have a concept of "neighbouring" points via constructs like the Delaunay tesselation and this provides the possibility of "predicting" values at a point from the Delaunay neighbours or some related set (such as the "natural neighbours"). The question of which predictor to use is more complex.

8 Transforms in 2-D: Image Data

So far it has been assumed that the data is a one-dimensional set of values (like a time series or the value of a function on an axis). Wavelet transforms of two-dimensional data is straightforward: just transform all the rows or columns of the data as if they were independent one-dimensional data sets. However, to make the transformation process useful is not quite as straightforward and there are several possible approaches.

8.1 Schemes

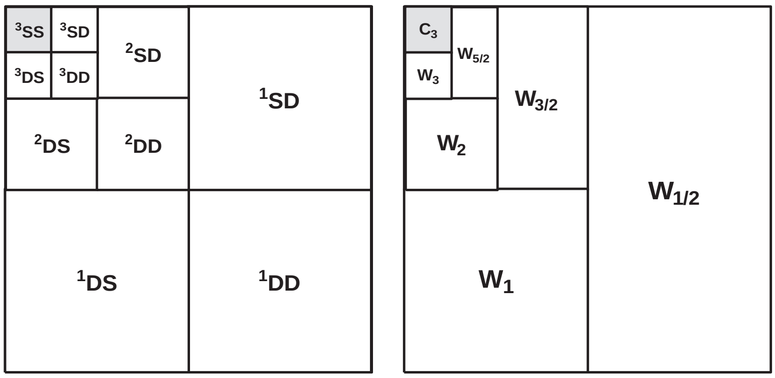

Fig. 6. Left: Mallat partition. Right: Feauveau partition.

Image data is a 2-dimensional array of pixels, usually rectangular though occasionally triangular sampling is used. The general idea behind performing a wavelet transform on rectangular sampled image data is to treat all rows and columns of the image as independent data strings. However, that leaves several available strategies for handling the rows and columns.

We start by transforming all the rows, keeping the smoothed data of the rows on the left and the detail data on the right. Then there is a choice. The first choice is to treat the columns as though they were data arrays and regardless of whether they contain smoothed or detailed row data. This produces four regions which we label **SS**, **SD**, **DS** and **DD** depending on what kind of data is there(see left panel of figure 6).

The **DD** part of the array is horizontal detail data that itself has been transformed to give a vertical detail of the horizontal detail (!). The **SS** part of the array is just a half size image. We can repeat the process dividing the **SS** part into 4 components as before. This is referred to as Mallat's scheme.

Alternatively, having done the first horizontal row transform, we could transform only the columns of horizontal smoothed data, leaving the horizontal details alone. This again leaves a half-size shrunken image in the top left of the array, and this can in turn be subdivided. This is referred to as Feauveau's scheme.

Although the Feauveau scheme requires only half the computational effort of the Mallat scheme, the latter is most commonly used since it treats the horizontal and vertical directions equally. From here on we shall stick to discussing the Mallat partition.

8.2 Interpretations

The four sections labelled **SS**, **SD**, **DS** and **DD** each reveal a different aspect of the image data. The **SS** part is simply a shrunken version of the image using the chosen scaling function. The **DD** part is detailed data that arises from computing details in both horizontal and vertical directions. The **SD** and **DS** parts reveal respectively the vertical and horizontal structures within the image.

The detail part of the wavelet transform is effectively a derivative of the data. Hence noise in the original data is strongly manifest in the details. If one can recognise in the detail components what is noise and what is image, then cleaning the image becomes possible. The methods that are available for noise handling are numerous and even look vaguely familiar: there is a wavelet analogue of the Weiner Filter, for example.

The main difficulty is to annihilate the noise without doing damage to the data that represents features. Most of that damage is damage to edges that delineate fine image structure: some noise filters may simply smooth the image, thereby wiping out faint structures. Context sensitive noise filtering involving feature recognition through nonlinear filters can be very effective.

It is often the case when compressing image data that the **DD** can be set to zero without suffering too much visible damage to the reconstructed image.

8.3 An example

As a specific example we take an image of builings in Valencia designed by the architect Calatrava [2]. Figure 7 shows the original while the left panel of figure 8 shows two levels of the wavelet transform. The image of the wavelet transform has been enhanced so as to reveal fine detail in the wavelet transform. One is struck by the large amount of noisy (uncorrelated) data in the **SD**, **DS** and **DD** components. If the noise component is filtered out the figure on the right results. We see a lot of black spaces: it is the encoding of these spaces that lead to strong compression.

Fig. 7. Calatrava architecture in Valencia

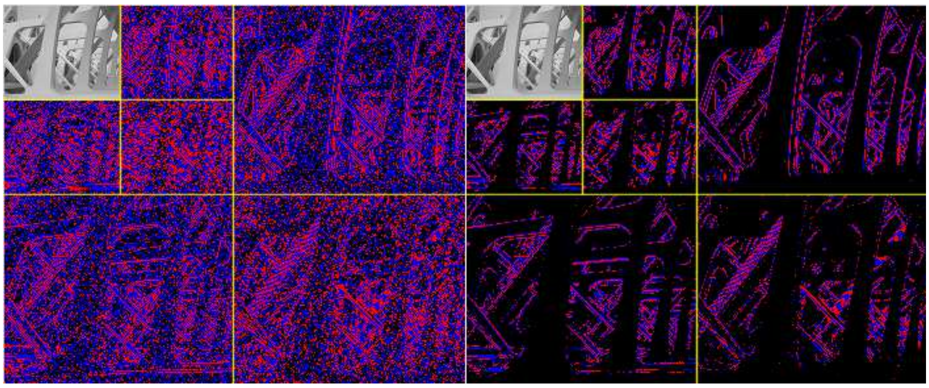

Fig. 8. Left: Wavelet transform of the Calatrava architecture image. Right: Censored version of wavelet transform of Calatrava image. The gamma has been changed for ease of viewing.

9 A specific example

The following example of the use of the continuous wavelet transform comes from James Walker's magnificent "Primer on Wavelets and their Scientific Applications" [8]. It illustrates what difference the choice of wavelet makes to the analysis. I have somewhat amplified that discussion by considering the effect of noise on the data.

The example starts with a one dimensional signal consisting of three wave packets. The wave packets are themselves built from two sinusoidal oscillations having different frequencies, localized by appying Gaussian window (see figure 9). Two of the packets are single frequency and the third is a mixture

[2]I am grateful to Rien van de Weygaert for this fine picture. Moreover, it makes a change not to use the image of Lena

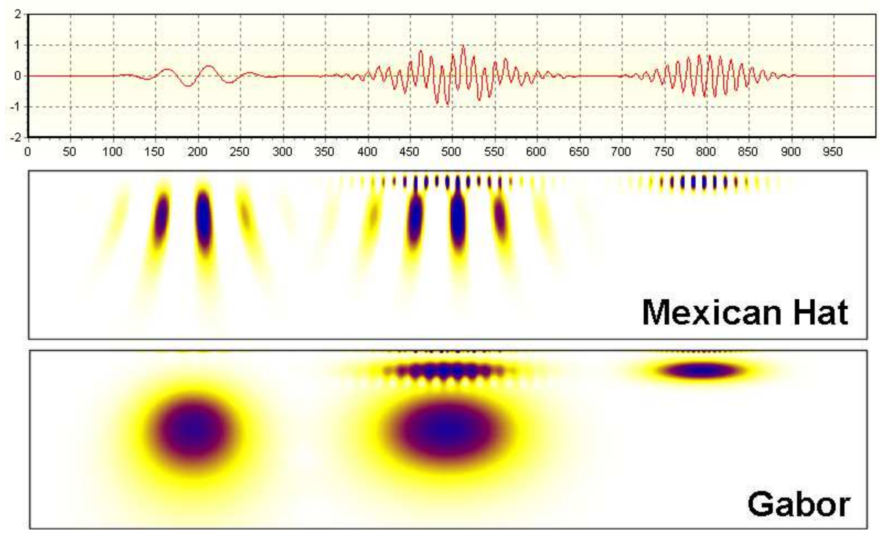

Fig. 9. Top panel: the original data consisting of three wave packets, the central one being a mixture of the other two. Middle panel: Mexican Hat scalogram. Bottom Panel: the Gabor scalogram.

of both. The question is whether we can use wavelet analysis to recognise how the mixture is built.

The figures show the data and scalograms generated by applying a Mexican Hat wavelet transforms and complex Gabor wavelet transforms. Both these transforms have a width parameter w: the wavelet transforms are computed for a series of values of w and stacked to form the scalogram with small values of w at the top and larger ones at the bottom.

The data consists of regularly spaced samples of the function

$$\begin{aligned} s(x) = {} & \sin(40\pi x) e^{-100\pi(x-0.2)^2} \\ & + [\sin(40\pi x) + 2\cos(160\pi x)] e^{-50\pi(x-0.5)^2} \\ & + 2\sin(160\pi x) e^{-100\pi(x-0.8)^2} \end{aligned} \tag{118}$$

The analysing wavelets are the Mexican Hat wavelet and the complex Gabor wavelet. The Mexican Hat used here is

$$\psi(x) = \frac{2\pi}{w^{1/2}} \left[1 - 2\pi \left(\frac{x}{w} \right)^2 \right] e^{-\pi(x/w)^2} \tag{119}$$

where w is a parameter that is varied to generate the scalogram. The data is transformed for a series of values of w.

The Gabor wavelet has real and imaginary parts that can be written as:

$$\begin{aligned} \psi_R(x) &= w^{-1/2} e^{-\pi(x/w)^2} \cos(2\pi\nu x/w) \\ \psi_I(x) &= w^{-1/2} e^{-\pi(x/w)^2} \sin(2\pi\nu x/w) \end{aligned} \tag{120}$$

The Gabor wavelet has two parameters w and ν. The width paramter w is just like the width paramter w in the Mexican Hat wavelet and is used to generate the scalogram. The frequency ν is an important extra parameter that can be used to tune the analysis. The figures shown here use a value of ν that is optimised for the analysis of the noise-free data and the same value of ν is used for the noisy data. In fact the noisy data could be shown to better advantage by using different values of ν, but we shall not go into that here.

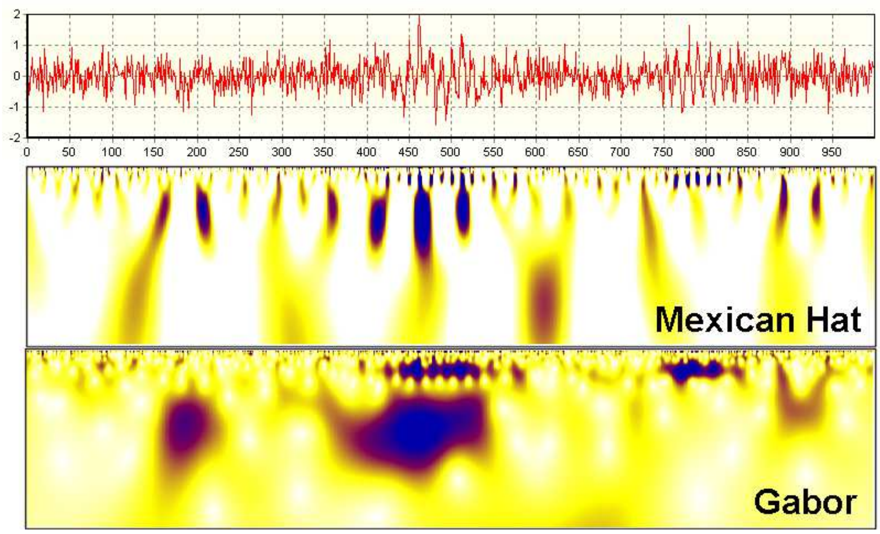

Fig. 10. Top panel: the noise-added data consisting of three wave packets, the central one being a mixture of the other two. Middle panel: Mexican Hat scalogram. Bottom Panel: the Gabor scalogram.

A cursory glance at the clean data analysis in figure 9 shows the difference between the Mexican Hat and Gabor wavelet analyses. While both analyses are capable of revealing the components, even when they are mixed, the differentiation is far clearer in the case of the Gabor analysis where we have used the extra parameter ν to advantage to resolve the mixture into two wuite separate islands. What is important is that the choice of wavelet effects how easily and unambiguously the result can be interpreted.

This is even more so when looking at the same data when has been polluted by additive noise (figure 10) having amplitude comparable with the amplitude of the original signal (can you see the signal in the noise, even though you know what you are looking for?). The noise that has been added here has a slightly red spectrum in order to make the signal discrimination more difficult.

The resulting scalograms have been distorted by the noise, but the situation is nonetheless still clear in the case of the Gabor wavelet analysis. The message from this simple example is that you should carefully formulate the questions that are to be asked of the data and choose the best wavelet to provide the answers. There is no lack of available transforms! The question that remains unanswered is how to best choose that wavelet and at the moment the only answer is “trial and error”.

10 Other data transforms

10.1 Windowed Fourier transform

The familiar Fourier transform of a function $f(x)$ is

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx \tag{121}$$

which has inverse

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} dx \tag{122}$$

(There are restrictions on $f(x)$ in order that these integrals exist). This looks like a special case of the continuous wavelet transform, but such appearances are misleading. $G(\omega)$ describes the frequency content of the function $f(x)$ over the entire domain of $f(x)$: the frequency information, while accurate, is not localised.

Localization of frequency information can be achieved by using a windowed version of the Fourier Trasform (also known as the Gabor Transform):

$$F(\omega) = \int_{-\infty}^{\infty} f(x) W(x) e^{-i\omega x} dx \tag{123}$$

for some window function $W(x)$. The problem with this form of the transform is that $W(x)$ does not itself depend on the frequency ω.

10.2 Fourier transform

Wavelets made their way into the second edition of *Numerical Recipes* [5]. That short review opens up a perspective into the world of filters by defining a function $H(\omega)$ from the filter coefficients $\{h_i\}$:

$$H(\omega) = \sum_j h_j e^{ij\omega} \tag{124}$$

$H(\omega)$ is periodic with period 2π. From the orthogonality relations (14) and the dilation equation (15) it can be shown that $H(\omega)$ satisfies the equations

$$\frac{1}{2}|H(0)|^2 = 1 \tag{125}$$

$$\frac{1}{2}\left[|H(\omega)|^2 + |H(\omega+\pi)|^2\right] = 1 \tag{126}$$

Any further conditions on the h_j impose further conditions on $H(\omega)$.

What is interesting is that it is simple to concoct functions satisfying all these conditions and so by inverting equation (124) come up with a set of N wavelet coefficients:

$$h_j = \frac{1}{N}\sum_{k=0}^{N-1} H\left(\frac{2\pi k}{N}\right) e^{-2\pi ijk/N} \tag{127}$$

However, not all $H(\omega)$ will produce wavelets having compact support.

Equation (26) expressing the mother wavelet $\psi(x)$ in terms of the scaling functions $\phi(x)$ gave rise to the set of scaling filter coefficients g_k that are related to the h_k via the quadrature mirror relationship $g_k = (-1)^k h_{L-k-1}$ (see equation (32)). If we write

$$G(\omega) = \sum_j g_j e^{ij\omega} \tag{128}$$

then it can be shown that

$$G(\omega) = e^{-i\omega} H^*(\omega+\pi) \tag{129}$$

$H(\omega)$ and $G(\omega)$ are called "quadrature mirror filters" [40]. The derivation of any of these equations in nontrivial.

So given $H(\omega)$ (say from (127)) we can determine both the scaling function $\phi(x)$ and the mother wavelet $\psi(x)$, and hence the complete multi-resolution analysis. Owing to the recursive nature of the wavelet bases, all computations should be performed using the quadrature mirror filters $H(\omega)$ and $G(\omega)$.

We can define the correlation functions of the $\phi(x)$ and $\psi(x)$ as

$$\Phi(x) = \int_{-\infty}^{\infty} \phi(t)\phi(t-x)dt \tag{130}$$

$$\Psi(x) = \int_{-\infty}^{\infty} \psi(t)\psi(t-x)dt \tag{131}$$

It can be shown, using the properties of Fourier transforms, that

$$\Psi(x) = 2\Phi(2x) - \Phi(x) \tag{132}$$

This should be compared with the approximation of a Mexican Hat wavelet by the difference of two Gaussians in equation (72).

10.3 The z-transform

In the engineering literature we encounter an important variant of the Fourier ransform called the z-transform. The Fourier series, or frequency domain, representation of a one dimensional data set $\{x(i)\}$ is given by

$$X(\omega) = \sum_{k \in \mathbf{Z}} x(k) e^{ik\omega} = X(e^{i\omega}), \quad \omega \in \mathbf{R} \tag{133}$$

(when this sum exists). The last equality has been written so as to emphasise that the dependence is in fact on the variable $e^{i\omega}$ and that X is periodic with period 2π when viewed as a function of ω . With this we can substitute

$$z = e^{i\omega} \tag{134}$$

and so rewrite the Fourier series as

$$X(z) = \sum_{k \in \mathbf{Z}} x(k) z^k \tag{135}$$

The difference now is that we can analytically extend the z-dependence of $X(z)$ away from the circle on which $X(\omega)$ was defined. There is of course a singularity at $z = 0$. In the particular case when the original data $x(k)$ is defined on a finite interval, $X(z)$ is simply a polynomial and can be analytically extended over the entire complex plane, with the exception of the origin.

The z-transform is linear, we have an entire battery of tools from complex variable theory that can be used, and it retains many nice properties of the Fourier representation fom which we derived it. An example of the last is the convolution theorem. If

$$w(n) = \sum_{k \in \mathbf{Z}} x(k) y(n-k) \tag{136}$$

then

$$W(z) = X(z)Y(z) \tag{137}$$

A further result is that if, from the sequence $\mathbf{x} = \{x(i)\}$, we define a second sequence $\mathbf{x}_{left} = \{x(i+1)\}$, then

$$X_{left}(z) = zX(z) \tag{138}$$

which is trival to verify. There is a similar result for the right-shifted sequence $\mathbf{x}_{right} = \{x(i-1)\}$:

$$X_{right}(z) = z^{-1}X(z) \tag{139}$$

Two of the most important operations concern the “2-down-sampled” and “2-up-sampled” sequences derived from the sequence $\{x(i)\}$. By 2-down-sampling we mean the sequence $\{x_{2\downarrow}(i)\}$ created from $\{x(i)\}$ by deleting all odd-indexed entries from the sequence and re-indexing:

$$x_{2\downarrow}(i) = x(2i), \quad i \in \mathbf{Z} \tag{140}$$

By 2-up-sampling we mean the sequence $\{x_{2\uparrow}(i)\}$ created from $\{x(i)\}$ by adding a new zero valued term in between every pair of terms $x(m)$ and $x(m+1)$, and then re-indexing the sequence:

$$x_{2\uparrow}(i) = \begin{cases} 0 & \text{if } i \in \mathbf{Z} \text{ is odd} \\ x([i/2]) & \text{if } i \in \mathbf{Z} \text{ is even} \end{cases} \tag{141}$$

These two sequences have z-transforms

$$X_{2\downarrow}(z) = \frac{1}{2}\left(X(z^{1/2}) + X(-z^{1/2})\right) \tag{142}$$

$$X_{2\uparrow}(z) = X(z^2) \tag{143}$$

The shift and up and down sampling operators are the basic tools for the design of very complex wavelet filters. This is an entire subject in its own right.

The z-transform is not used to compute the transform of a dataset, rather it is used as a theoretical tool to construct and analyse wavelet filters. It provides, for example, a tool for classifying wavelets through the properties of the zeroes of the various wavelet generators.

11 Further issues - in brief

11.1 Wavelet packet transforms

There is no reason why the Difference data D resulting from a wavelet transform $\{S, D\}$ of a dataset $\{X\}$ should not itself be transformed: doing so might, for example, give an improved compression of the data or provide for better noise handling. The skill here is to decide how many levels of the D-data transform are needed and to choose an appropriate wavelet at each level.

This procedure can be particularly effective when dealing with two dimensional image data. We have noted that the **SD** and **DS** parts of the transform show predominately vertical and horizontal image structure. There is some advantage to be gained in doing a vertical transform on the **SD** component and a horizontal transfrom on the **DS** part. Again the choice of wavelet and the number of levels to transform is a key issue. In practise it is possible to test several strategies and choose the most effective.

11.2 Period doubling

Period-doubling is implicit in everything we have done so far (specifically because the scaling functions satisfy the dilation equation (15)). This does not normally pose any problems in using wavelets for analysis. For some problems, particularly problems where the time dependence of a variable is the subject of investigation, it may be useful to have more levels. Chiu et al [15] have sought to generalize the basic scaling equations to

$$\phi_k^{j,n}(x) = (2^j \kappa_n)^{\frac{1}{2}} \phi(2^j \kappa_n x - k) \tag{144}$$

$$\psi_k^{j,n}(x) = (2^j \kappa_n)^{\frac{1}{2}} \psi(2^j \kappa_n x - k) \tag{145}$$

$$\kappa_n = \frac{2^N}{2^N + n} \tag{146}$$

in order to provide $2^N - 1$ additional levels of analysis.

An obvious alternative is to use the continuous wavelet transform if the scaling function is known explicitly.

11.3 Shift Invariance

The usual wavelet transforms are not invariant under a linear shift in the data. Thus the coefficients of the wavelet transform of the shifted data are not he same as the coefficients of the wavelet transform of the original data: there may be a significant redistribution of energy among the wavelet components. This may or may not be a problem, it depends what you are doing with your wavelet transform. Matching images, such as stereo pairs, benefits form using shift-invariant wavelets.

Wavelets having shift invariance can be designed by going to complex wavelets, but one of the costs of this is that some redundancy is introduced and there is a considerable computational overhead. The subject is reviewed by Kingsbury [29] (this paper uses the z-transform).

11.4 Horizontal-Vertical bias

Two-dimensional wavelet transforms are often regarded as a horizontal transform followed by a vertical transform: this inevitably produces a bias towards vertically or horizontally aligned structures. We can see this in the two dimension wavelet map where the **DD** component that describes diagonal structure is always of lower order than the **SD** and **DS** components (one more differentiation is implied in **DD**).

11.5 Locality

Wavelets are localized (compact support) and so, in one dimension, can deal with discontinuities more effectively than Fourier representations. One reason wavelets are thought to be useful in economics (time series analysis) because they can deal with sudden change.

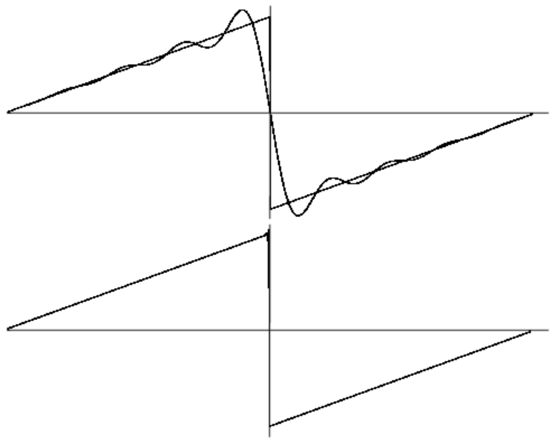

Fig. 11. Top: sawtooth approximated by 17 Fourier coefficients. Bottom: sawtooth approximated by 17 wavelet coefficients

This is nicely illustrated in Figure 11 where we compare the Fourier and wavelet representations of a simple sawtooth function using 17 terms in the respective expansions. The Fourier representation shows a strong "Gibbs" effect that is entirely absent in the wavelet representation.

However, 2-d wavelets suffer problems with handling linear features and as a consequence produce artefacts. One solution that has been proposed that deals with this is the generalization to "Ridglets".

11.6 Orthogonality

The wavelets we have been dealing with have various orthogonality properties. The orthogonality is only necessary if we are to reconstruct the original data from the transformed data. If we have no interest in the reconstruction, there is no need to insist on orthogonality.

11.7 Continuous versus discrete transforms

We see two types of wavelet transform: the Continuous Wave transform (CWT) as exemplified by the mexican hat wavelet, and the Discrete Wavelet Transform (DWT) as exemplified by Dauechies' wavelets. In order to use the Continuous Wavelet transform on a data set whose values are given on a discrete set of points it is inevitably necessary to discretise the Wavelet function in order to do the integrals, so in a practical sense the continuous wavelet transform reduces to a discrete transform. However, there is more to it than that.

11.8 Discretizing the continuous wavelet transform

If we consider the wavelet transform

$$F(a,b) = \frac{1}{\sqrt{a}} \int_0^\infty f(x)\psi\left(\frac{x-b}{a}\right) dx \tag{147}$$

and choose

$$a = 2^{-j}, \quad b = 2^{-j}k \tag{148}$$

for some k, we get what looks like a discrete wavelet transform with scales parametrized by the level number j. This certainly allows us to construct a hierarchical representation of the data $f(x)$. However, this transform is not in general invertible.

11.9 Hybrid Approaches

There is some advantage to be gained by working on image data in the Fourier domain and doing the Wavelet operations there before transforming back to the real data space.

11.10 Multifractals

It was remarked that the wavelet transform of $f(x)$, $F(a,b)$, is dominated by cusps that represent the scaling behaviour of $f(x)$ in local neighbourhoods. Given a wavelet that has vanishing moment of all orders up to some N, this cuspy behaviour has asymptotic behaviour

$$F(a,b) \sim |a|^{h(x_0)} \tag{149}$$

for some function $h(x)$ describing the scaling of the singularity behaviour of $f(x)$. If we look in real space, as opposed to Wavelet space, the local behaviour of the function $f(x)$ in the vicintiy of spikes will have the form

$$|f(x) - f(x_0)| \sim |x - x_0|^{\alpha_0} \tag{150}$$

α_0 is called the Hölder or singularity exponent. In other disciplines, notably in the area of finance modelling, $H = 1 - \alpha_0/2$ is referred to as the Hurst exponent, after H.E. Hurst who studied the structure of the river Nile. The Hurst exponent is calculated by comparing the range of the data with its standard deviation: the so-called “R/S” statistic. This exponent α_0 will be an erratic function of position x_0, and so we may prefer to write α_0 explicitly as a function of position, $\alpha_0(x)$, in order to emphasise this.

12 Further Reading

I have used a great number of references from papers, the world wide web and books in compiling this review, too many to mention individually. Here I list some of the main sources which will perhaps aid others wishing to plunge further into this wavelet sea.

12.1 General Introduction

James Walker's *Primer on Wavelets and their Scientific Applications* [8] was one of my starting points. It is full of instructive examples, one of which I have borrowed here (Section 9). *Discovering Wavelets* by Aboufadel and Schlicker [1] is perhaps more formal, but it makes for an excellent course book since everything is worked out, it has problems and answers and there are sample routines written in *Maple*.

There are excellent review articles available on the World Wide Web by Dremin, Ivanov and Nechitailo [21], and by Bultheel which are available from

`http://arxiv.org/abs/hep-ph/0101182`

`http://www.cs.kuleuven.ac.be/ ade/WWW/WAVE/contents.html`.

respectively.

12.2 Applications

The monograph *Ripples in Mathematics: the Discrete Wavelet Transform* [3] by Jensen and Cour-Harbo focusses on the discrete wavelet transform as might be applied to signal processing. The focus is on the Lifting process and filters. Many practical problems are addressed, such as what to do at the boundaries of finite stretches of data. A magnificent book from which I have learned, and am still learning, a lot. Examples are in *MATLAB* format.

The proceedings of the Royal Society discussion meeting *Wavelets: the Key to Intermittent Information?* edited by B.W. Silverman and J.C. Vassilicos [6] is a fine collection of articles spanning a large range of applications of wavelets. There are papers on the analysis of economic and financial data, intermittency in fluid motion, statistics, and other diverse topics. A fascinating insight into what wavelets can do for us, showing how work across different fields stimulates advances in the subject.

12.3 Astronomical

The only kid on the block is the monograph *Image Processing and Data Analysis: the Multiscale Approach* by Starck, Murtagh and Bijaoui who were perhaps the pioneers in exploiting wavelet and multiscale analysis into astronomy. The book is perhaps more about multiscale image analysis than about wavelets as such, but it is also a rich source of information on advanced imaging techniques in general.

There are many papers using wavelets to analyse various all-sky microwave background data sets (COBE, WMAP and Planck). To cite but a few: Barreiro et al. [9], Cayon et al. [14], Starck et al. [39], Rocha et al. [36]. The paper by Wiaux et al. [44] is particularly interesting in that they generate a wavelet on a sphere via inverse stereographic projection. McEwen et al. [33] use Morlet wavelets on the sphere.

The paper of Maisinger et al. [30] looks at using Maximum Entropy Reconstruction techniques in wavelet space to denoise and deconvolve microwave background maps and images in general.

12.4 Web Sites

The World Wide Web is a vast, almost inpenetrable, source of information: the amount of wavelet related content is truly awesome. However, there are a few sites that might be described as primary resources.

`http://www.wavelet.org`

`http://www.amara.com/current/wavelet.html`

`http://www.ondelette.com/indexen.html`

`http://www-stat.stanford.edu/~wavelab/`

The first of these issues an e-zine with valuable up to date information on preprints, books and meetings. The second is a fine general resource: almost anything can be found from here. The third is a messageboard (in English) with some useful resources such as programs. Wavelab is a collection of MatLab functions that do all kinds of things with wavelets.

Acknowledgements

It was a great pleasure participating in this meeting and I have Vicent Martinez and Enn Saar to thank for that. Together with Sylvestre Paredes they were responsible for my getting into this subject and related areas such as the multifractal analysis of structure. Rien van de Weygaert has been an invaluable source of discussion about such things. He also provided me with a library of fine pictures of the Valencia architecture of Calatrava.

References

1. E. Aboufadel and S. Schlicker: *Discovering Wavelets* John Wiley (1999)
2. I. Daubechies: *Ten Lectures on Wavelets*, Philadelphia, SIAM.
3. A. Jensen and A. la Cour-Harbo: *Ripples in Mathematics: The Discrete Wavelet Transform*. Springer (2001)
4. A. Papoulis: *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 3rd. edition (1991)
5. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery: *Numerical Recipes: the Art of Scientific Computing* Cambridge Univerity Press (2002)

6. B.W. Silverman and J.C. Vassilicos: *Wavelets: the Key to Intermittent Information?* Oxford University Press (1999)
7. J.-L. Starck, F. Murtagh and A. Bijaoui: *Image Processing and Data Analysis: the Multiscale Approach* Cambridge University Press (1998)
8. J.S. Walker: *Wavelets and their Scientific Applications* Chapman and Hall (1999).
9. R.B. Barreiro, E. Martinez-Gonzalez and J.L. Sanz: Mon. Not. Roy. Astr. Soc. **322**, 411 (2001)
10. G. Beylkin, R. R. Coifman and V. Rokhlin: Comm. on Pure and Applied Math., **44** 141, (1991)
11. . Carrier, L. Greengard and V. Rokhlin: SIAM Journal of Scientifc and Statistical Computing, **9** 4, (1988)
12. P.J. Burt and A.E. Adelson: IEEE Transaction on Communications **31** (1983)
13. J. Barnes and P. Hut: Nature **324**, 446 (1986)
14. L. Cayon, J.L. Sanz, E. Martinez-Gonzalez, A.J. Banday, F. Argueso, J.E. Gallegos, K.M. Gorski and G. Hinshaw: Mon. Not. Roy. Astr. Soc. **326**, 1243 (2001)
15. C.K. Chiu, J.C. Goswami and A.K. Chan: Numerische Mathematik **70**, 283 (1995)
16. I. Daubechies: Communications on Pure and Applied Mathematics **41**, 909 (1988)
17. I. Daubechies: SIAM Journal on Mathematical Analysis, **24**, 499 (1993)
18. I. Daubechies and W. Sweldens: J. Fourier Anal. Appl. **4**, 245 (1998)
19. W. Dehnen: J. Comp. Physics, **179**, 27 (2002)
20. W. Dehnen: Mon. Not. Roy. Astron. Soc. **324**, 273 (2001)
21. I.M. Dremin, O.V. Ivanov amd V.A. Nechitailo: Phys. Usp., **44**, 447 (2001)
22. S. Dubuc: J. Math. Anal. and Appl. **114**, 185 (1986)
23. G. Deslauriers and S. Dubuc: Constructive Approx. **5**, 49 (1989)
24. D. Gabor: Journal of the IEE: **93**, 429 (1946)
25. L. Greengard and V. Rohklin: Acta Numerica **6**, 229 (1997)
26. A. Grossmann and J. Morlet: SIAM Journal of Mathematical Analysis **15**, 723 (1984)
27. A. Haar: Mathematische Annalen **69**, 331 (1910)
28. J. Littlewood and R. Paley: J. London Mathematical Soc. **6**, 20 (1931)
29. N.G. Kingsbury: p165 *Wavelets: the Key to Intermittent Information?* ed B.W. Silverman and J.C. Vassilicos. Oxford University Press (1999)
30. K. Maisinger, M.P. Hobson and A.N. Lasenby: Mon. Not. Roy. Astr. Soc. **347** 339 (2004)
31. S. Mallat: IEEE Transactions on Pattern Analysis and Machine intelligence **11**, 674 (1989)
32. S. Marr: *Vision* W.H. Freeman (1982)
33. J.D. McEwen, M.P. Hobson, A.N. Lasenby and D.J. Mortlock: Mon. Not. Roy. Astr. Soc. **359** 1583 (2005)
34. Y. Meyer, S. Jaffard, O. Rioul: Pour la Science 28 (1987)
35. E. PierPaoli, S. Anthoine, K. Huffenberger and I. Daubechies Mon. Not. R. Astron. Soc.: **359**, 261 (2005)
36. G. Rocha, L. Cayon, R. Bowen, A. Canavezes, J. Silk, A.J. Banday, K.M. Gorski: Mon.Not.Roy.Astron.Soc. **351**, 769 (2004)
37. A.B. Romeo, C. Horellou, J. Bergh: Mon.Not.Roy.Astron.Soc. **354**, 1280 (2004)

38. H. Samet: ACM COmputing Surveys **16**, 187 (1984)
39. J.L. Starck, N. Aghanim, O. Forni: Astron.Astrophys. **416**, 9 (2004)
40. M.J. Smith and T.P. Barnwell: IEEE Transactions on ASSP **34**, 434 (1986)
41. W. Sweldens: Appl. Comput. Harmon. Anal. **3**, 186, (1996)
42. W. Sweldens: SIAM J. Math. Anal. **29**, 511 (1997)
43. W. Sweldens and P. Schroder: *Wavelets in Computer Graphics*, p15. ACM SIGGRAPH course notes, (1996)
44. Y. Wiaux, L. Jacques and P. Vandergheynst: Astrophys.J. **632**, 15 (2005)