9. Wavelets

Wavelet analysis developed in the largely mathematical literature in the 1980's and began to be used commonly in geophysics in the 1990's. Wavelets can be used in signal analysis, image processing and data compression. They are useful for sorting out scale information, while still maintaining some degree of time or space locality. Wavelets are used to compress and store fingerprint information by the FBI. Because the structure functions are obtained by scaling and translating one or two "mother functions", time-scale wavelets are particularly appropriate for analyzing fields that are fractal. Wavelets can be appropriate for analyzing non-stationary time series, whereas Fourier analysis generally is not. They can be applied to time series as a sort of fusion between filtering and Fourier analysis. Wavelets can be used to compress the information in two-dimensional images from satellites or ground based remote sensing techniques such as radars. Wavelets are useful because as you remove the highest frequencies, local information is retained and the image looks like a low resolution version of the full pictures. With Fourier analysis, or other global functional fits, the image may lose all resemblance to the picture, after a few harmonics are removed. This is because wavelets are a hierarchy of local fits, and retain some time localization information, and Fourier or polynomial fits are global fits, usually.

In general, you can think of wavelets as a compromise between looking at digital data at the sampled times, in which case you maximize the information about how things are located in time, and looking at data through a Fourier analysis in frequency space, in which you maximize your information about how things are localized in frequency and give up all information about how things are located in time. In wavelet analysis we retain some frequency localization and some time localization, so it is a compromise.

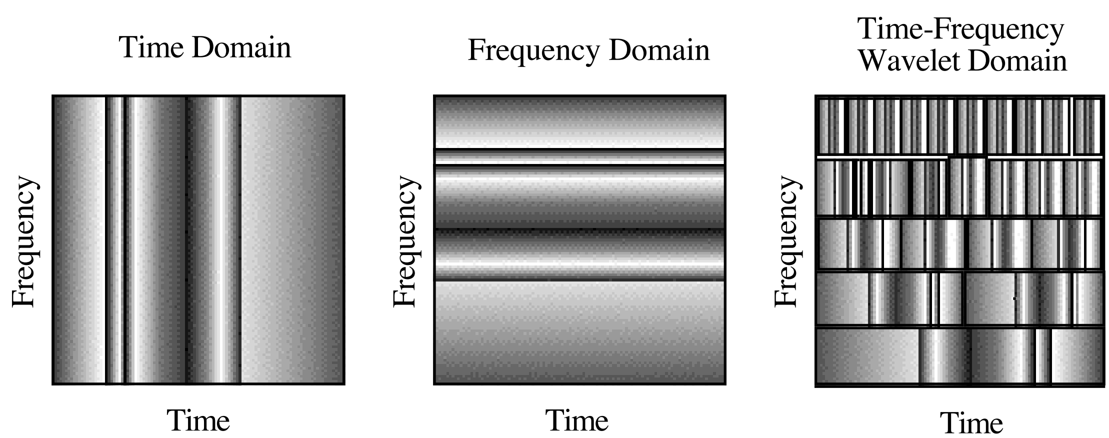


Figure. 1. In the time domain we have full time resolution, but no frequency localization or separation. In the Fourier domain we have full frequency resolution but no time separation. In the wavelet domain we have some time localization and some frequency localization.

9.1 Wavelet Types

According to Meyer(1993), two fundamental types of wavelets can be considered, the Grossmann-Morlet time-scale wavelets and the Gabor-Malvar time-frequency wavelets. The more commonly used type in geophysics is probably the time-scale wavelet. These wavelets form bases in which a signal can be decomposed into a wide range of scales, in what is called a "multiresolution analysis". From this comes the obvious application in image compression, as one can call up additional detail as required until the exact image at the original resolution is reconstructed. The intervening coarse resolution images will look like the full resolution one, just fuzzier. This is not true in general of Fourier analysis, where throwing out the last few harmonics can cause the picture to change dramatically.

Time-scale wavelets are defined in reference to a "mother function" $\psi(t)$ of some real variable *t*. The mother function is required to have several characteristics: it must oscillate, and it must be localized in the sense that it decreases rapidly to zero as $|t|$ tends to infinity. It is also very helpfult to require that the mother function have a certain number of zero moments, according to:

$$0 = \int_{-\infty}^{\infty} \psi(t)\, dt = \ldots = \int_{-\infty}^{\infty} t^{m-1} \psi(t)\, dt \tag{9.1}$$

The mother function can be used to generate a whole family of wavelets by translating and scaling the mother wavelet.

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a > 0, \;\; b \in \Re. \tag{9.2}$$

Here b is the translation parameter and a is the scaling parameter. Provided that $\psi(t)$ is real-valued, this collection of wavelets can be used as an orthonormal basis. The coefficients of this expansion can be obtained through the usual projection.

$$\Psi_{(a,b)} = \int_{-\infty}^{\infty} f(t)\, \psi_{(a,b)}(t)\, dt \tag{9.3}$$

These coefficients measure the variations of the field $f(t)$ about the point *b*, with the scale given by *a*. Wavelet analysis of this type can be performed on discrete data using quadrature mirror filters and pyramid algorithms. It is also possible sometimes to compute the transform using a Fourier transform technique.

Time-frequency wavelets are constructed with the idea that you take a wave, $\cos(\omega t + \varphi)$, divide it into segments, and keep only one (Gabor 1946). This leaves a "wavelet" with three parameters: a starting time, an ending time, and a frequency. Recent innovations have provided more practical algorithms for the time-frequency wavelet that are useful with discrete data. You might imagine that such a representation would be very useful in music and speech coding.

The trick in using wavelets is to find a set of them that provides a description that is optimal in some sense to the problem at hand. If wavelet analysis in general, or the particular set chosen, is not well-suited to the problem at hand, they can be no help or, worse, lead to deeper confusion. For the non-expert like us, who just wants to get a useful representation, one is probably restricted to choosing from among a library of established wavelet bases, and most probably from among those for which software is already written. This library is growing, as are the techniques for deteriming whether an appropriate representation has been chosen. Matlab has a wavelet toolbox, which includes Haar, Daubechies, Biorthogonal, Coiflets, Symlets, Morlet, Mexican Hat and Meyer wavelets.

We focus here in these notes on discrete wavelets and the discrete wavelet transform (DWT) and their applications. Wavelets are basis sets for expansion which, unlike Fourier series, have not only a characteristic frequency or scale, but also a location. They can be orthogonal, biorthogonal, or nonorthogonal. So we imagine first that we have some sort of linear series expansion of a signal $x(t)$.

$$x(t) = \sum_i \alpha_i \varphi_i \tag{9.4}$$

Normally we would wish that φ_i form a complete orthogonal set on the space in which x is defined, so that any x can be expressed in terms of this basis set. When a Fourier Series expansion is performed the resulting coefficients α_i can be used to describe the distribution of the variance in frequency space by computing the power spectrum, so that a scale separation is performed, but the information about the behavior of particular scales as a function of time is lost. One can get around this partially by computing a series of short term Fourier transforms (STFT) on series of length T, which might be shorter than the total length of record, but long enough to discriminate the frequency of interest from others. These short records could be partially overlapping, so that the scale analysis could be plotted two-dimensionally in frequency-time coordinates, so that the temporal behavior of the variance in the frequencies of interest could be studied.

9.2 The Haar Wavelet

Haar(1910) and others were seeking functional expansions that would converge to explain other functions that were not the sine and consine series of Fourier(1807). He sought an orthonormal system $h_n(t)$ of functions on the interval [0,1] such that for any function $f(t)$, the series,

$$f(t) = \sum \langle f, h_n \rangle h_n(t) \tag{9.5}$$

would converge uniformly. The angle brackets indicate a suitably defined inner product on the interval [0,1]. Haar began with the initial function,

$$h(t) = \begin{cases} 1.0 & [0, 1/2] \\ -1.0 & [1/2, 1] \\ 0.0 & elsewhere \end{cases} \tag{9.6}$$

Building on this basic function Haar defines his sequence of expansion functions according to,

$$n = 2^j + k \qquad j \geq 0, \quad 0 \leq k \leq 2^j \tag{9.7}$$

$$h_n(t) = 2^{j/2}\, h\left(2^j t - k\right) \tag{9.8}$$

each of these functions is supported (has nonzero values) on the dyadic interval,

$$I_n = \left[k\, 2^{-j}, (k+1) 2^{-j}\right] \tag{9.9}$$

which is included in the inverval [0,1] if $0 \leq k \leq 2^j$. To complete the set, one must add the function $h_0(t) = 1$ on the inverval [0,1]. The series $h_n(t)$ then forms an orthonormal basis on [0,1]. By looking carefully at (9.7)-(9.9) one can see that the series is the basic step function repeated on intervals that decrease in scale and increase in number by the factor of two at each level, where *j* is the level index and *k* is the number of functions of a given scale necessary to span the interval [0,1].

Let's consider the Haar expansion of a time series to illustrate the concept of discrete wavelet analysis in a very simple form. The discrete Haar wavelet is a two point sum and difference representation. In discrete work, it is handier to start with the smallest scale and work upward to the bigger ones. For the discrete Haar wavelet to converge, the total number of data points in the time series must be a power of two. The basis functions are given by,

$$\varphi_{2k}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k, 2k+1 \\ 0 & otherwise \end{cases} \qquad \varphi_{2k+1}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k, \\ -\frac{1}{\sqrt{2}} & n = 2k+1 \\ 0 & otherwise \end{cases} \tag{9.10}$$

Where *n* is the time series index. Even and odd (2k and 2k+1) indexed functions are, respectively, sum and differences of two adjacent time points, with the factor of one on square root of two thrown in to make the basis set orthornormal. Successive even and odd functions are just translations by an even number of time steps of the other even and odd functions. The individual functions are thus very localized to two adjacent time points.

Since the Haar functions are orthogonal, we can derive their coefficients using the relation,

$$\alpha_i = \langle \varphi_i, x(t) \rangle \tag{9.11}$$

where the angle brackets indicate a suitably defined inner product.

It may be easier to see how this is all working by considering how (9.11) looks when expressed in matrix notation, and using the abbreviation $a = \frac{1}{\sqrt{2}}$.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ .. \end{bmatrix} = \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_1(2) \\ y_2(2) \\ y_1(4) \\ y_2(4) \\ \ldots \end{bmatrix} = \begin{bmatrix} a & a & & & & & \\ a & -a & & & & & \\ & & a & a & & & \\ & & a & -a & & & \\ & & & & a & a & \\ & & & & a & -a & \\ & & & & & & .. \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ \ldots \end{bmatrix} \tag{9.12}$$

We can think of y_1 and y_2 as the time series of the coefficients of the even and odd Haar wavelets, respectively. These have only half the time resolution of the original series. You can think of y_1 as a low-frequency representation of x(t) and y_2 as the high frequency details. Often in wavelet analysis literature, the smooth function (a,a) would be called the scaling function φ, and the wavy one (a,-a) would be called the wavelet ψ. The projection into the coefficient space of the two Haar functions is equivalent to filtering followed by "down sampling", by taking only every other point of the filtered time series. The Haar transform is an example of a two-channel filter bank. It sorts the original series into two filtered data sets. The Haar filter functions are members of a special class of filter function pairs called a quadrature mirror filter pair. After the filtering is done the sum of the energies (or variances) in the two filtered time series is equal to the variance in the original time sereis.

$$|y_1|^2 + |y_2|^2 = |x|^2 \tag{9.13}$$

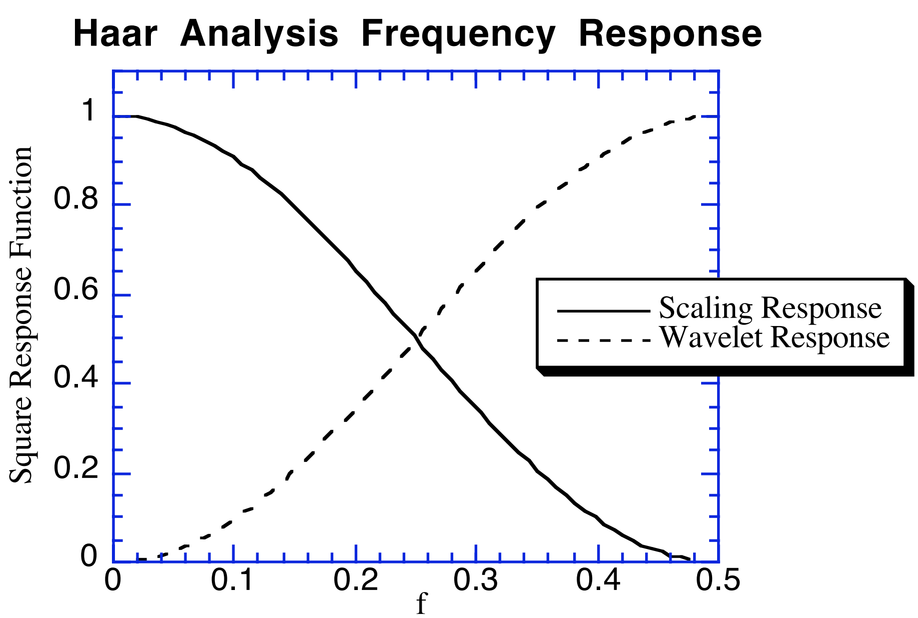
Since we are thinking of a wavelet transform as a filtering operation, now is a good time to think about the scaling achieved by this filtering process. Remember, from the previous chapter on filtering of time series, how we determine the frequency response of the filter from its coefficients. Since this is a non-recursive filter as it stands, we know from the time-shifting theorem that the Fourier transform of the data F(f), will be modified by being multiplied by the transfer function of the filter, which is given by,

$$H(f) = \sum_{n=0}^{M} a_n e^{-i2\pi nf} \qquad (9.14)$$

The squared response function shows how the filter process would affect the power spectrum. As an exercise, one may show that the squared response function for the scaling (a,a) and wavelet (a,-a) filtering operations are, respectively, where $a = 1/\sqrt{2}$, then

$$|H(f)|^2_{scaling} = \cos^2(\pi f) \quad and \quad |H(f)|^2_{wavelet} = \sin^2(\pi f) \qquad (9.17)$$

From these formulas one can see that the squared response functions are complements of eachother, so that the variance that is rejected by one is the variance that is passed by the other. This is the required characteristic of quadrature mirror filters, and will result in the preservation of power as the expansion in these wavelets continues.



The Haar wavelet representation has the advantage of very good time localization, but the frequency resolution is minimal. Also, it is not smooth. It is not a very attractive wavelet basis. You could get much better frequency resolution by using sinc functions as the basis set, but to get very fine frequency resolution you would end up with very poor time resolution. A compromise is needed.

Pyramid Scheme:

Applying the Haar transform reduces the original N data point time series $x(t)$ into two time series of length N/2, which are y_1 and y_2 , respectively. One of these contains the smoothed information and the other contains the detail information. The smoothed one could be transformed again with the Haar wavelets again, producing two time series of length N/4, with smoothed and detail information, and so on, keeping the details and doing an additional transform of the smoothed time series each time. If the original time series was some power of 2, $N=2^n$, then this process, called a pyramid algorithm, would terminate when the last two time series were the coefficients of the time mean and the difference between the mean of the first half of the time series and the last half of the time series. The number of coefficients at the end would total N, and would contain all of the information in the original time series, organized according to scale and location, as defined by the Haar wavelet family. The original mother functions of (1,1) and (1,-1) on an interval of two time points are stretched, or dilated in factors of 2 to create a sequence of daughter wavelets with increasingly large scale.

Let's suppose we started with a time series of 8 data, and performed successive Haar transforms on this time series. The diagram below is intended to give some idea of how the original data would be transformed into a representation in Haar functions using the pyramid scheme. The notation is a little primitive. The first subscript indicates whether it is the first-smoothed, or second-detailed Haar function coefficient. The second subscript indicates the total span of the wavelet-the number of time points it stretches over. The original set span two data points, but the span doubles every time the transform is applied to the smoothed transformation from the previous level of the pyramid. The number in parenthesis indicates the approximate time point at the center of the wavelet in question. This is the time we would plot the coefficient at, if we wanted to see how this particular scale was evolving in time.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \Rightarrow \begin{bmatrix} y_{22}(1.5) \\ y_{22}(3.5) \\ y_{22}(5.5) \\ y_{22}(7.5) \end{bmatrix}; \begin{bmatrix} y_{24}(2.5) \\ y_{24}(6.5) \end{bmatrix}; \begin{bmatrix} y_{18}(4.5) \\ y_{28}(4.5) \end{bmatrix} \tag{9.16}$$

At the end of the scheme we have the coefficients of the Haar function that is the same at all 8 points, y_{18}, and the coefficient of the Haar function that is positive for the first 4 times and negative for the last 4 times y_{28}, which is the last bit of detail. The time at which these are valid is right in the center of the time series. Each level represents a particular scale, but in the case of the Haar wavelet, the scale separation is crude. We can reconstruct the original time series from the Haar coefficients if we want. This discussion of the Haar wavelet set introduces the concept of multiresolution. The wavelet basis is capable of localizing signals in both time and frequency simultaneously. Of course there is an uncertainty principle at work, because if we want to isolate frequencies very exactly, then we must give up time localization (sinc wavelet), and if we want to localize very finely in time, then we must give up on precise frequency localization (Haar wavelet).

In seeking other possible basis function sets on which we would like to expand we consider the following desirable characteristics:

(1) Good localization in both time and frequency (these conflict so we must compromise)
(2) Simplicity, and ease of construction and characterization
(3) Invariance under certail elementary operations such as translation
(4) Smoothness, continuity and differentiability
(5) Good moment properties, zero moments up to some order.

9.3 Daubechies Wavelet Filter Coefficients:

From the example of the Haar wavelet, we can see that a wavelet transform is equivalent to a filtering process with two filters that are quadrature mirror filters and divide the time series into a wavelet part, which represents the detail, and another smoothed part. Daubechies(1988) discovered an important and useful class of such filter coefficients. The simplest set has only 4 coefficients (DAUB4), and will serve as a useful illustration. Consider the following transformation acting on a data vector to its right.

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & & & & & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & & & & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & \bullet & \bullet & & & \\ & & & & & \bullet & \bullet & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & & & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & & & & & & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & & & & & & c_3 & -c_2 \end{bmatrix} \tag{9.17}$$

The action of this matrix is to perform two convolutions with different, but related, filters, (c_0, c_1, c_2, c_3) $=H$ and $(c_3, -c_2, c_1, -c_0)$ $=G$, each resulting time series of filtered data points is then decimated by half, so that only half as many data points remain, then both filtered time series, thus decimated, are interleaved. We can think of H as the smoothing filter and G as the wavelet filter. They produce the smooth and detail information, respectively. The filter G is chosen to make the filtered response to a sufficiently smooth input as small as possible, and this is done by making the moments of G zero. When p moments are zero, we say that G satisfies an approximation condition of order p.

If we require an approximation condition of order p=2, then the coefficients for the DAUB4 wavelet must satisfy,

$$c_3 - c_2 + c_1 - c_0 = 0 \tag{9.18}$$

$$0c_3 - 1c_2 + 2c_1 - 3c_0 = 0 \tag{9.19}$$

For the transformation of the data vector to be useful, one must be able to reconstruct the original data from its smooth and detail components. This can be assured by requiring that the matrix (9.17) is orthogonal, so that its inverse is just its transpose. In discrete space, this is the equivalent of the orthogonality condition for continuous functions. The orthogonality condition places two additional constraints on the coefficients, which can be derived by multiplying (9.17) by its transpose and requiring that the product be the unit matrix.

$${c_3}^2 + {c_2}^2 + {c_1}^2 + {c_0}^2 = 1 \tag{9.20}$$

$$c_3 c_1 + c_2 c_0 = 0 \tag{9.21}$$

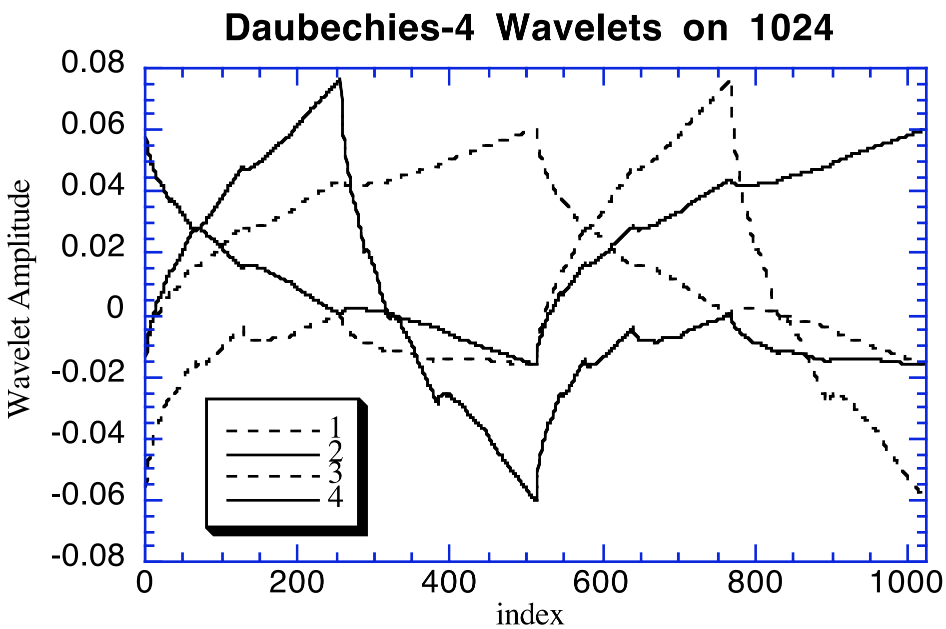
These four equations for the coefficients have a unique solution up to a left-right reversal. DAUB4 is only the simplest of a family of wavelet sets with the number of coefficients increasing by two each time (4, 6, 8, 12, . . . 20, . . .). Each time we add two more coefficients we add an additional orthogonality constraint and raise the number of zero moments, or the approximation condition order, by one. Daubechies(1988) has tabulated the coefficients for lots of these, and they can be inserted into computer programs provided by Press, et al.(1992).

The discrete wavelet transform proceeds by the pyramid algorithm. A coefficient matrix like (9.17) is applied hierarchically. After the first transform of a data vector of length N, the detail information is stored in the last $N/2$ elements of the transformed vector, and another transform of the $N/2$ smooth components is performed to provide a detail vector and a smooth vector each of length $N/4$. Then the detail at this level is stored and another transformation of the $N/4$ smooth vector is performed. This continues until only one smooth coefficient and one detail coefficient remain, at which point N coefficients of the transformed coefficient vector have been obtained. We can illustrate this process with an initial vector of length N=8.

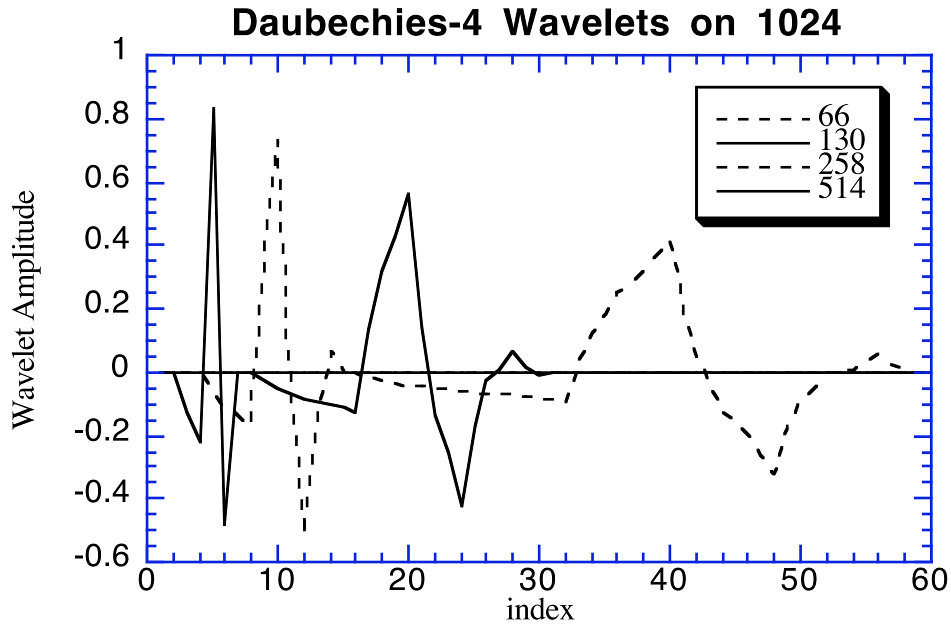
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \xrightarrow{transform} \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \xrightarrow{permute} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \xrightarrow{transform} \begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \xrightarrow{permute} \begin{bmatrix} S_1 \\ S_2 \\ D_1 \\ D_2 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \qquad (9.22)$$

If the original data were a higher power of two, there would be more stages in the pyramid transformation, but the ending point is always two detail coefficients and two smoothed coefficients for the final level. The *d*'s are called "wavelet coeffients". The final *S* coefficients could be called "mother-function coefficients", or mother and father coefficients, but are often also called wavelet coefficients. Since each stage of the process is an orthogonal linear operation, the sum of all these transformations is also an orthogonal operation. To invert the procedure and change the coefficients back to the original data vector, one simply reverses the process, using the transpose of the transformation matrix at each level of the pyramid.

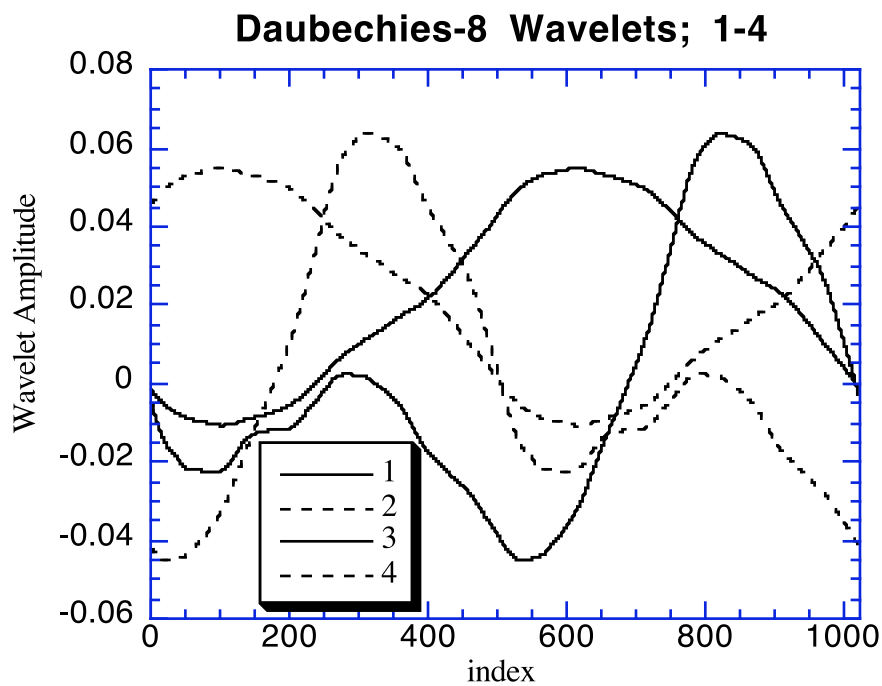
Although the pyramid scheme only requires the coeffients of the fundamental quadrature mirror filter, the structure of the wavelets can be reconstructed by placing a one in the element of the coefficient vector for the wavelet structure you want, place zeros in all other locations, and then do the inverse transform to produce the physical space representation of the wavelet structure. One can easily see by taking the transpose of (9.17) and operating on vectors with ones in various elements, that the wavelet structure at the first level of wavelet detail is just the wavelet filter coefficients themselves. Higher up the pyramid structure the wavelets take on more details that are not obvious from the coeffients alone. For example the following diagram shows the DAUB4 wavelet structures from a transformation of length 1024 corresponding to coefficients 1,2,3 and 4. These are the father, mother and first two wavelets- the largest scale wavelets, corresponding to the lowest coefficients for DAUB4 on 1024. The DAUB4 wavelet has kinks where the first derivitive does not exist, but it exists "almost" everywhere. The mother and father have the same scale but different shapes, with the father being the smoother one and the mother the basic wavelet. The 3 and 4 wavelets are the first born. They have the same structure, but are shifted in location so as to be orthogonal. All subsequent children have this characteristic, but decrease in scale by a factor of 2 and increase in number by a factor of 2.

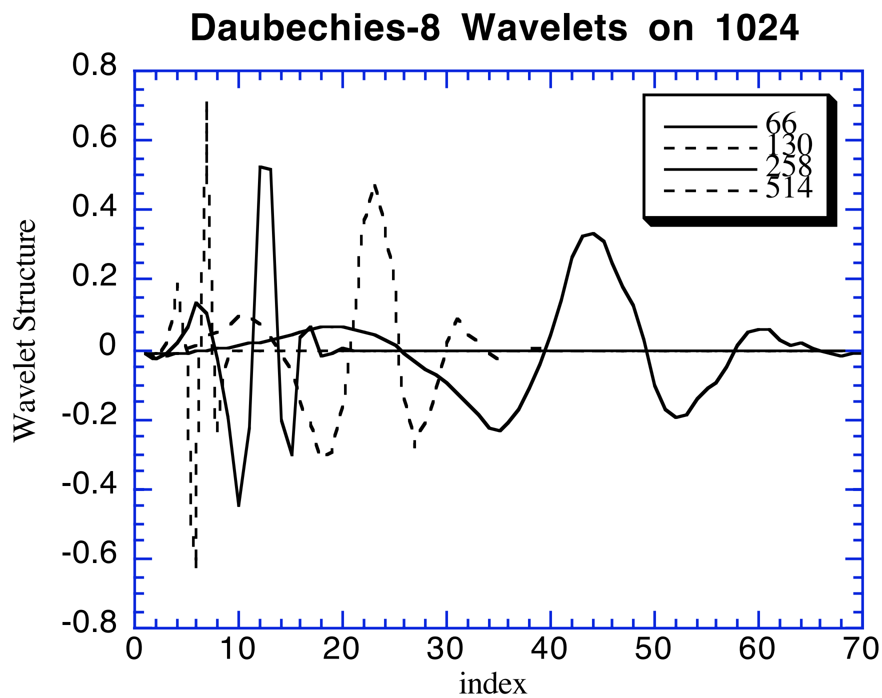


Let's look at the grandchildren. The wavelet for coefficient 514 is of the smallest scale and is localized near the beginning of the time series. The structure is just the filter coefficients shifted in time into the beginning of the data a little. Lower coefficients correspond to wavelets with progressively doubled scale, and their structures take on a little more detail at this order of approximation(DAUB4). We show only the left part of the 1024 vector space, since this is where these wavelets have amplitude. We show here the wavelets for coefficients 514, 258, 130 and 66. These are all located near the beginning of the time series, but each represent scales that differ by factors of 2. To obtain the next wavelet in each level, you would keep the same structure but shift it to the right by 2, 4, 8, and 16 time units, respectively.

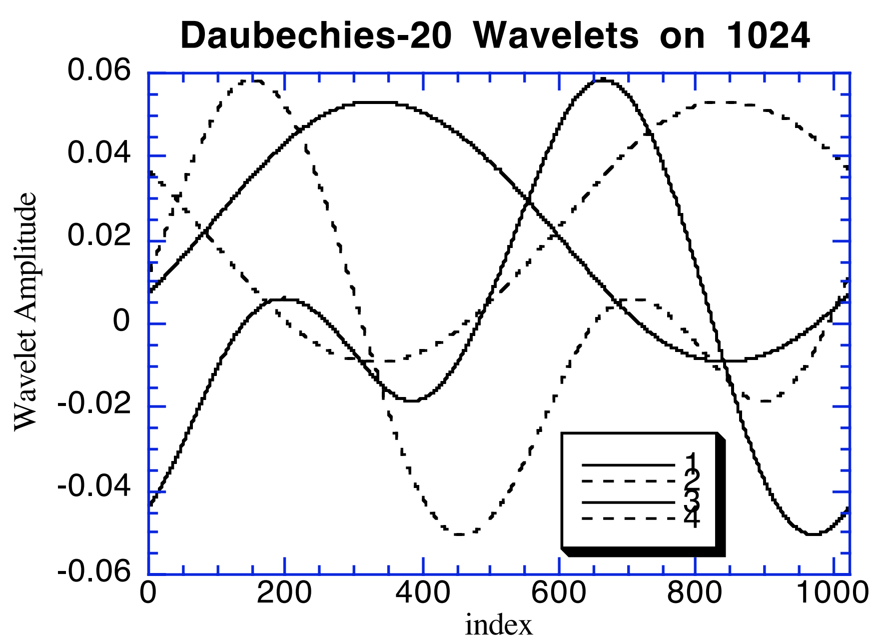


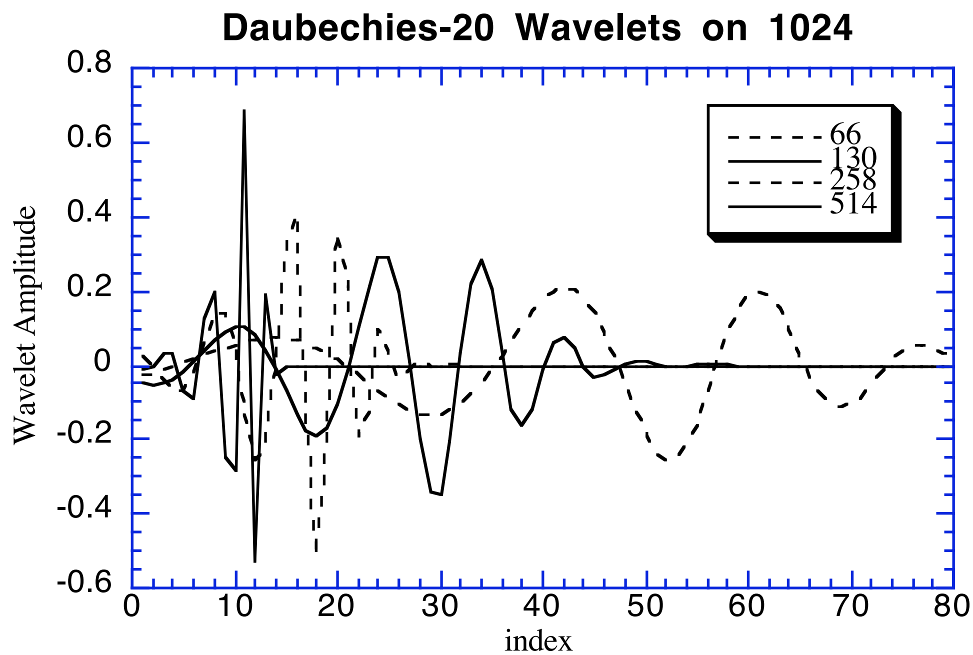
Higher order wavelets, such as DAUB8, shown below have higher order continuous derivatives. They are not quite as local as a lower order Daubechies wavelet set, since the wavelet of smallest scale is supported over a larger number of data points.





The DAUB-20 wavelet produces even more smoothness, and less localization.





9.4 Wavelet Types and Properties

TBD

9.5 The Inverse Problem in Music: Would Wavelets really help?

Suppose you are an ethnomusicologist and you have recorded the tunes and harmonies of a primitive, but musical tribe in the central Amazon Basin. You want to convert the recording into a score based on the western system of music. This is the inverse problem in music. You have the voiced music, but you want it converted into musical notation. The forward problem would be if you had sheet music and you wanted to create the sound. This is a good problem in digital signal processing and time series analysis.

In some of the references for wavelets music is used as an example of a kind of mixed time-frequency multiresolution problem for wavelets. However, most of the dyadic wavelet bases resolve frequences that differ by factors of two. That is a whole octave, and so is too coarse frequency resolution to be useful for music scoring. As we shall see, to get the required frequency resolution to resolve the individual notes within an octave, one does better to just use Fourier Analysis.

The Well-Tempered Clavier:

The western musical scale is divided up into octaves, the frequencies of the succeeding octaves differ by factors of 2. Each of these octaves is divided into 12 semitones, whose frequencies have the ratio ~1.05946 =α, so that $\alpha^{12} = 2$, or $\alpha = \sqrt[12]{2}$. So all we need to do is pick the frequency of some reference note and we can construct the frequencies of the entire chromatic scale of music. Two tunings are used. The classic is the 'Concert A'; the A above middle C is tuned to 440 Hz. Computer musicians prefer to tune middle C to 256 Hz. If you do an analysis based on powers of two, more of the notes are generated or picked out precisely by the analysis ($256 = 2^8$). These two tunings are not compatible, since they differ in most places by close to half a step. If you have defined your reference note and frequency, then you can compute the frequencies of all the other notes in the system from the following relationship.

$$f_{m+n} = 2^{\left(\log_2 f_m + n/12\right)}$$

where f_m is the reference frequency, and n is the number of half steps from the reference frequency to the note of which you want the frequency f_{m+n}. Below are the four octaves about middle C for the Concert A tuning.

Table: The frequencies of the four octaves about middle C for the Concert A tuning. In each octave an index of half steps with middle C defined as zero is given, along with the frequency in Hertz (cycles per second) and the corresponding note name.

C Below C			Below C			Middle C			Above C		
-24	65.406	C	-12	130.813	C	0	261.626	C	12	523.251	C
-23	69.296	Db	-11	138.591	Db	1	277.183	Db	13	554.365	Db
-22	73.416	D	-10	146.832	D	2	293.665	D	14	587.330	D
-21	77.782	Eb	-9	155.563	Eb	3	311.127	Eb	15	622.254	Eb
-20	82.407	E	-8	164.814	E	4	329.628	E	16	659.255	E
-19	87.307	F	-7	174.614	F	5	349.228	F	17	698.456	F
-18	92.499	Gb	-6	184.997	Gb	6	369.994	Gb	18	739.989	Gb
-17	97.999	G	-5	195.998	G	7	391.995	G	19	783.991	G
-16	103.826	Ab	-4	207.652	Ab	8	415.305	Ab	20	830.609	Ab
-15	110.000	A	-3	220.000	A	9	440.000	A	21	880.000	A
-14	116.541	Bb	-2	233.082	Bb	10	466.164	Bb	22	932.328	Bb
-13	123.471	B	-1	246.942	B	11	493.883	B	23	987.767	B
-12	130.813	C	0	261.626	C	12	523.251	C	24	1046.502	C

Notice that the frequency spacing is proportional to frequency itself. If we wanted to distinguish these notes using wavelet or harmonic analysis we would want to be able to distinguish half tones in the lowest octave. The difference between C and Db in the lowest octave is 69.296 - 65.406 = 3.89 Hz. To distinguish these frequencies we need to sample a long enough time so that the wavelet structures we project onto the data get significantly out of phase on this time interval. Then one wavelet will project well onto the harmonic of interest, but the next one won't. If two frequencies differ by Δf , then they become different in phase by one cycle in a time such that $\Delta f\, T = 1 cycle$, or:

$$\Delta f = \frac{1}{T} \quad or \quad T = \frac{1}{\Delta f}$$

This is the same as the formula for the bandwidth of a Fourier spectral analysis. If we give a Fourier analysis a time interval of T to work on, it can distinguish frequencies separated by 1/T, the bandwidth of the spectral analysis. Since we need good frequency resolution, and we have FFT software readily available, it is attractive to use a moving block FFT spectral analysis to detect the notes, and not mess with wavelets. To exactly separate that C from that Db, you would need to do a Fourier analysis in blocks of 1/(3.89 Hz), or about a quarter of a second. Since many notes are not held for a full quarter second, you may want to do the analysis more often than once every quarter second, so you would have overlapping periods of record and compute a new spectrum every eighth, sixteenth, or thirtysecond of a second. To get better time resolution in the higher frequencies, where you don't need so much frequency resolution, you could use shorter blocks of say an eighth of a second to do the spectral analysis to find the notes.

Sound is recorded digitally on Cds with a sampling rate of 44,100 Hz to resolve the 20,000 Hz signals that some people can hear. The human voice is very well reproduced with about 8,000 samples per second. So if it is a voice recording, you may as well reduce the sampling rate to about 8 kHz. When this is done a quarter second is about 2048 samples, which works really well for FFT analysis. Someone reasonably familiar with Matlab can write a program to do this type of frequency analysis in a few hours. Below is an example of a frequency analysis of a male gospel singer. This was done with an FFT of length 2048 8kHz samples, done and plotted every 256 samples. Thus about a 1/4 second sample is taken every 1/32 of a second. A Hanning window was used. The Hanning window effectively shortens the sampling interval while smoothing the spectrum a little.

The analysis is able to capture the frequency of the notes sung, which can then be read off the table above. The temporal resolution of the frequency analysis is good enough to capture the trilling of the held notes around 6 and 8.5 seconds. You see a hint of the resonance higher harmonic between seconds 7 and 9. The high frequencies around 7.5 seconds are talking or clapping in the background. Without the resonances and the clapping, it would be a simple matter to write software to convert this frequency analysis into MIDI events, which could then be introduced to a standard musical notation package to produce sheet music from original recordings. Thence the musical inverse problem would be solved, so long as the music is based on the western system with the A-tuning.

The next figure shows the time frequency analysis at a later time when the other members of the quartet are also singing. I have chosen a section where little resonance is present. At other times we see more than 4 peaks, even though only 4 people are singing. This is because of the harmonics generated by the individual singers. A professional quality voice is characterized by the richness and pleasing quality of the harmonics that are generated. Good singers can control the amount of the higher frequency resonances that they produce and generate interesting variations that way. Luciano Pavarotti is gifted with a voice with lots of harmonic richness and color.

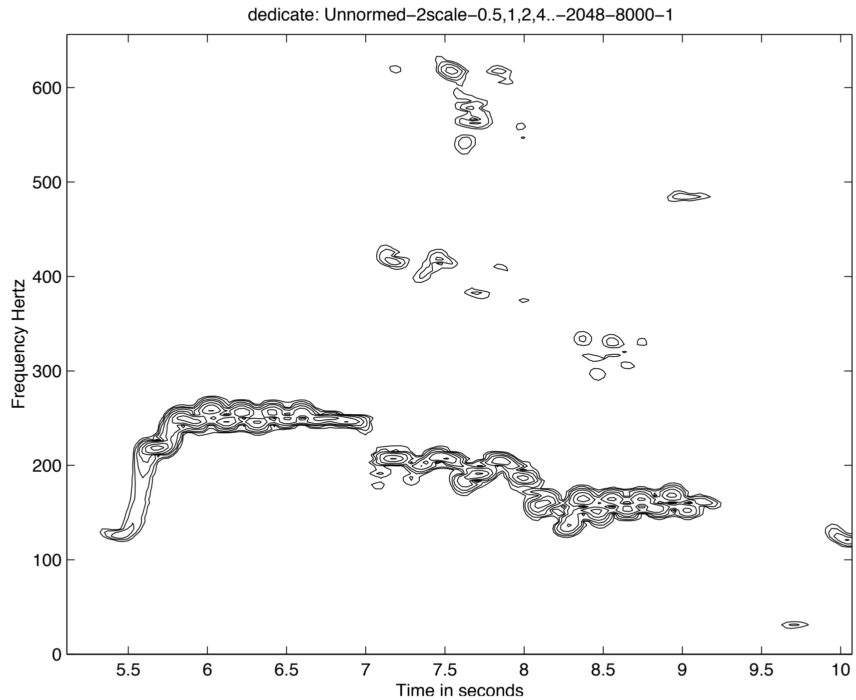


Figure: Frequency analysis of male vocalist. Contours are spaced in powers of two.

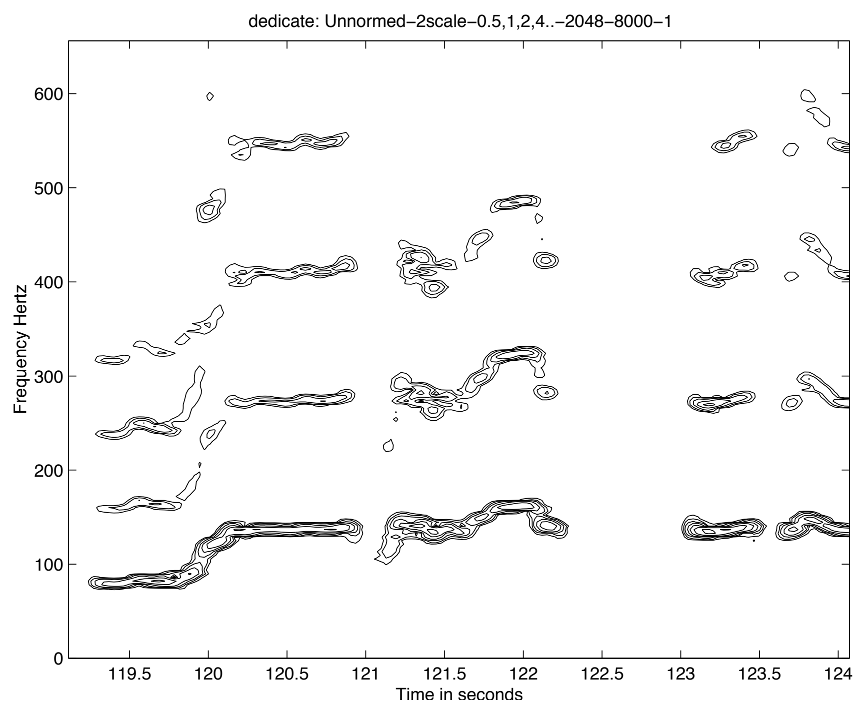


Figure: Frequency analysis of gospel quartet singing harmony.

References:

Chiu, C. K., 1992: *An Introduction to Wavelets.* Academic Press, Harcourt Brace Jovanovich, 266.

Daubechies, I., 1988: Wavelets and quadrature filters(?). *Comm. Pure Appl. Math.*, **41**, 909-996.

Daubechies, I., 1992: *Ten Lectures on Wavelets,.* SIAM, 357.

Farge, M., 1992: Wavelet transforms and their applications to turbulence. *Ann. Rev. Fluid Mech.*, **24**, 395-457.

Fournier, A., 1996: Wavelet analysis of observed geopotential and wind: Blocking and local energy coupling across scales. *Proceedings of the SPIE The International Society for Optical Engineering*, **2825**, 570-81.

Gamage, N. and W. Blumen, 1993: Comparative analysis of low-level cold fronts: Wavelet, Fourier and empirical orthogonal function decompositions. *Mon. Wea. Rev.*, **121**, 2867-2878.

Gollmer, S., Harshvardhan, R. F. Cahalan and J. B. Snider, 1995: Windowed and wavelet analysis of marine stratocumulus cloud inhomogeneity. *J. Atmos. Sci.*, **52**, 3013-3030.

Gu, D. and S. G. H. Philander, 1995: Secular changes of annual and interannual variability in the tropics during the last century. *J. Climate*, **8**, 864-876.

Hernandez, E. and G. L. Weiss, 1996: *A First Course on Wavelets.* CRC Press Inc., 489.

Hubbard, B. B., 1996: *The World According to Wavelets: The Story of a Mathematical Technique in the Making*. A.K. Peters, Ltd., 286.

Hudgins, L. and H. Jianping, 1996: Bivariate wavelet analysis of Asia monsoon and ENSO. *Adv. Atmos. Sci.*, **13**, 299-312.

Huffman, J. C., 1994: Wavelets and image compression. *SMPTE Journal*, **103**, 723-7.

Kaspersen, J. H. and L. Hudgins, 1996: Wavelet quadrature methods for detecting coherent structures in fluid turbulence. *Proceedings of the SPIE The International Society for Optical Engineering*, **2825**, 540-50.

Kumar, P. and E. Foufoula-Georgiou, 1993: A new look at rainfall fluctuations and scaling properties of spatial rainfall using orthogonal wavelets. *J. Appl. Meteor.*, **32**, 209-222.

Kumar, P., 1995: A wavelet based methodology for scale-space anisotropic analysis. *Published by: American Geophys. Union. Geophysical Research Letters*, **22**, 2777-80.

Lau, K.-M. and H. Weng, 1995: Climate signal detection using wavelets transform: How to make a time series sing. *Bull. Amer. Meteor. Soc.*, **76**, 2391-2402.

Lin, Z. S., W. L. Bian and W. H. You, 1996: The wavelets and hierarchies of the climate system. *Meteorology and Atmospheric Physics*, **61**, 19- 26.

Meyers, S. D., B. G. Kelly and J. J. O'Brien, 1993: An introduction to wavelet analysis in oceanography and meteorology: With applications to the dispersion of Yanai waves. *Mon. Wea. Rev.*, **121**, 2858-2866.

Meyers, Y., 1993: *Wavelets: Algorithms and Applications.* SIAM, Philadelphia, 133.

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, 1992: *Numerical Recipes.* Second Edition, Cambridge U. Press, Cambridge, UK, 963.

Prokoph, A. and F. Barthelmes, 1996: Detection of nonstationarities in geological time series: wavelet transform of chaotic and cyclic sequences. *Computers & Geosciences*, **22**, 1097-108.

Strang, G. and T. Nguyen, 1996: *Wavelets Filter Banks.* Wellesley-Cambridge Press, 672.

Strang, G., 1989: Wavelets and dilation equations. *SIAM Review*, **31**, 614-627.

Teolis, A., 1996: *Computational Signal Processing with Wavelets.* Birkhauser, Boston,

Vetterli, M. and J. Kovacevic, 1995: *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, N.J., 488.

Wang, B. and W. Y., 1996: Temporal structure of the Southern Oscillation as revealed by waveform and wavelet analysis. *J. Climate*, **9**, 1586-1598.

Weng, H.-Y. and K. M. Lau, 1994: Wavelets, period doubling, and time-frequency localization with application to organization of convection over the tropical western Pacific. *J. Atmos. Sci.*, **51**, 2523-2541.

Wojtaszczyk, P., 1997: *A Mathematical Introduction to Wavelets.* Cambridge University Press, 280.

Wu, Y. and B. Tao, 1996: Detection of chaos based on wavelet transform. *Proceedings of the SPIE The International Society for Optical Engineering*, **2825**, 130-8.