An Introduction to Wavelets

Advanced Linear Algebra
(Linear Algebra II)

Heng-Yu Lin

May 27 2013

Abstract

With the prosperity of the Digital Age, information is nowadays increasingly, if not exclusively, stored and treated digitally. With great advantages promised by digitalization of data, such as the possibility of incredibly massive storage, comes the challenge of classifying, retrieving and analyzing these massive data efficiently. In fact, collecting experimental data leads inevitably to the inclusion of some unnecessary information commonly known as “noise”. Filtering out the noise data in order to reduce the size of data is thus an important task. *Wavelets* are precisely a set of tools specially designed to solve this problem.

Introduction

Simply put, wavelets are a class of functions that are very efficient at discriminating actual data from noise data, hence their application in signal processing as filters. However, unlike other signal processing tools such as Fourier transforms, which only use a linear combination of sines and cosines to approximate a function (which are generally periodic), wavelets transforms use an infinite set of functions of different scales and at different locations to perform the same task. In fact, a *family* of wavelets is composed of an infinite set of functions generated by rescaling and translation of the *scaling function,* also known as the *Father wavelet (phi)* and the complementary function, the *wavelet function,* which is also known as the *Mother wavelet (psi).* The rescaled and translated functions are called *son wavelets* and *daughter wavelets.*

***Haar* wavelets**

In order to have a better understanding of the notions, it is instructive to illustrate the situation with an actual family of wavelets. The *Haar* family, due to its simplicity, is often used as an illustrating example. However, because of its simplicity, it is not very useful when processing actual data. The *Haar* father wavelet is defined by:

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t \leq 1 \\ 0, & \text{if } otherwise \end{cases}$$

whereas the *Haar* mother wavelet is defined by:

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leq t \leq \frac{1}{2} \\ -1, & \text{if } \frac{1}{2} \leq t \leq 1 \\ 0, & \text{if otherwise} \end{cases}$$

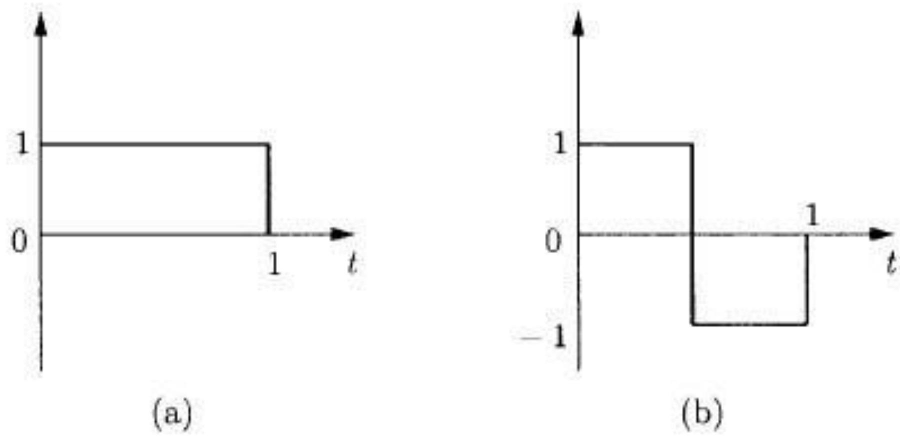

Fig. 1. *Haar* father and mother wavelets.

Fig. 1. illustrates the father and mother wavelet functions of the *Haar* family. In addition, the son wavelets and the daughter wavelets are rescaled and translated according to parameter n and k as follows:

$$\phi_{n,k} = \phi(2^n t - k), k = 0, 1, ..., 2^n - 1$$
$$\psi_{n,k} = \psi(2^n t - k), k = 0, 1, ..., 2^n - 1$$

A Very Large Toolbox

The above definition of daughter wavelets clearly demonstrates the rescaling of the mother wavelet by the parameter n, whereas the translation is dictated by the parameter k. While a change of parameter n allows one to look at the function or signal to be analyzed at different *resolutions* or *scales* (smaller n's for longer, lower frequency wavelets and larger n's for shorter, higher frequency wavelets), a change of parameter k allows one to localize the function at a desired position. Note that *each* daughter and son wavelet is an individual function that is part of the wavelet family. A family of wavelet thus grant one the power to analyze the big picture and examine details at any desired resolution and position *simultaneously* through access of an incredibly large array of functions.

Dilation Equation

In addition, note that the first two sons are simply the father wavelet, but rescaled and translated to be supported in the [0, 1/2] and [1/2, 1] intervals, respectively. Consequently, the father wavelet can be expressed as a linear combination of the first two sons:

$$\phi(t) = c_0\phi(2t) + c_1\phi(2t - 1)$$

In this case, $c_0 = c_1 = 1$.This particular property is a very important relation known as the *dilation equation.* Although this relation seems trivial in the case of *Haar* wavelets, it is an important property that is shared by all wavelet families. In general, the dilation equation, or refinement equation in some of the literature, is as follows:

$$\phi(t) = \sum_k c_k \phi(2t - k)$$

Note that having a finite set of c_k's implies that the scaling function has *compact support*, which means that the scaling function only exists for a finite interval and vanishes to zero outside this interval. For instance, the *Haar* wavelet, which only have two non-zero refinement coefficients, has compact support in the interval [0, 1]. In another case, the *Daubechies 4* wavelets, which has 4 refinement coefficients, are supported in the [0, 3] interval.

Orthogonality and Nested Sequence of Spaces

First, let us define V_n to be the set of functions which are piecewise constant on intervals of length 2^{-n}, starting from zero. For instance, V_2 is the inner product space spanned by the set of functions that are piecewise constant on quarters, and V_1 is the space spanned by the set of functions that are piecewise constant on halves. If we let $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ be an orthonormal basis for V_0**. (**This is a critical condition that most wavelets families must satisfy. It can be easily shown for *Haar* wavelets since the scaling function has compact support in the [0, 1] interval), then the set $\{\phi(2t-k)\}$ is an orthogonal basis for the space V_1. Again, this can be easily shown for the *Haar* wavelets since the son wavelets are in essence the standard orthogonal basis. Thus we obtain the following:

$$\{\phi(t-k)\}_{k\in\mathbb{Z}} = \left\{\phi(2^0 t - k)\right\}_{k\in\mathbb{Z}} \in V_0$$

$$\{\phi(2t-k)\}_{k\in\mathbb{Z}} = \left\{\phi(2^1 t - k)\right\}_{k\in\mathbb{Z}} \in V_1$$

and hence in general:

$$\{\phi(2^n t - k)\}_{k\in\mathbb{Z}} \in V_n.$$

Although this is an informal proof of orthogonality that only applies to *Haar* wavelet, the same spaces V_n thus defined are spanned by the son wavelets of appropriate generation of many other wavelet families. In other words, finding an appropriate basis that spans the spaces V_n is equivalent as defining an entire family of wavelets. In later discussion below, it will be shown how the refinement coefficients defining wavelet families are used to determine spanning basis.

Observe, in addition, that the dilation equation implies that V_0 is a subset of V_1. Thus proceeding inductively, the nested sequence of spaces

$$\{0\} \subseteq \ldots \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq V_2 \subseteq \ldots \subseteq L^2(\mathbb{R})$$

is obtained, where $L^2(\mathbb{R})$ is defined as follows

$$f : [0,1] \to \mathbb{R} \quad s.t. \quad \|f\| = \sqrt{\int_{-\infty}^{\infty} (f(t))^2 dt} < \infty$$

Note that each space V_n is spanned by an orthogonal basis $\{\phi(2^n t - k)\}$. In brief, as the value of n gets larger, the space spanned by the subsequent shorter and finer son wavelets are broader and include the spaces spanned by the previous son wavelets. Conversely, as the value of n gets more negative, the spanning wavelet sons are much longer and larger in size and can only span spaces of smaller dimension. The intersection of all V_n spaces for *Haar* wavelets is then the space spanned by a function that is constant all the time on the real line and also satisfy the condition for $L^2(\mathbb{R})$. Since any non-zero constant function, however close it is to zero, will pick up some distance and the length of the function will inevitably blow up to infinity when the function is extended from negative infinity to positive infinity, the only function satisfying the conditions is the 0 function. Thus we obtain the following:

$$\cap_{n\in\mathbb{Z}} V_n = \{0\}$$

Again, this is a property that can be easily shown for *Haar* wavelets, but it is also shared by all wavelet families.

Constraints Imposed on the Refinement Coefficients

The orthogonality of the function set $\{\phi(2t-k)\}$ will further establish a series of conditions that wavelet families must follow. For instance, since scaling functions satisfy the dilation equation

$$\phi(t) = \sum_k c_k \phi(2t-k) \ ,$$

the value of each c_k can be determined as the projection of $\phi(t)$ onto $\phi(2t-k)$ for the same k
.

$$proj_{\phi(2t-k)}\phi(t) = \frac{\langle \phi(t), \phi(2t-k) \rangle}{\langle \phi(2t-k), \phi(2t-k) \rangle} \phi(2t-k)$$

$$c_k = \frac{\langle \phi(t), \phi(2t-k) \rangle}{\langle \phi(2t-k), \phi(2t-k) \rangle}$$

$$c_k = \frac{\langle \phi(t), \phi(2t-k) \rangle}{\frac{1}{2}}$$

$$c_k = 2 \cdot \langle \phi(t), \phi(2t-k) \rangle$$

In addition, the orthogonality of the $\{\phi(2t-k)\}$ functions further imposes another condition on the refinement coefficients $\{c_k\}$. The following derivation leads to a relation known as *Parseval's formula:*

$$\langle \phi(t), \phi(t) \rangle = \left\langle \sum_k c_k \phi(2t-k), \sum_k c_k \phi(2t-k) \right\rangle$$

$$1 = \sum_k c_k^2 \langle \phi(2t-k), \phi(2t-k) \rangle$$

$$1 = \sum_k c_k^2 \cdot \frac{1}{2}$$

$$\sum_k c_k^2 = 2$$

Moreover, for the wavelet sons to be orthogonal to their translates, the refinement coefficients $\{c_k\}$ must also satisfy the condition derived below:

$$\langle \phi(t), \phi(t-j) \rangle = \left\langle \sum_k c_k \phi(2t-k), \sum_{k+2j} c_{k+2j} \phi(2t-(k+2j)) \right\rangle$$

$$= \sum_{-\infty}^{\infty} c_k c_{k+2j} = \begin{cases} 0 \text{ if } j \neq 0 \\ 2 \text{ if } j = 0 \end{cases}$$

Relation Between Father and Mother Wavelets

It is important to observe that not only does the father wavelet (scaling function) follow a dilation equation, the mother wavelet also satisfy a dilation equation. In general,

$$\psi(t) = \sum_k b_k \phi(2t-k)$$

In the case of *Haar* wavelets, b_0=1, b_1=-1 and the rest of the coefficients are 0. For simplicity of computation, we normalize the $\phi(2t-k)$ functions to have dilation equations with *normalized* refinement coefficients for both the father and the mother wavelets to obtain the following:

$$\phi(t) = \sum_k h_k \sqrt{2}\phi(2t-k)$$
$$\psi(t) = \sum_k g_k \sqrt{2}\phi(2t-k)$$

where $h_k = \frac{c_k}{\sqrt{2}}$ and $g_k = \frac{b_k}{\sqrt{2}}$.

The orthogonality between the father and mother wavelet is a crucial condition to satisfy when creating wavelet families. Therefore, the dot product between father wavelet function and mother wavelet function must be zero. The following computation shows that, with the established orthogonality of the $\phi(2t-k)$ wavelet sons, the condition of orthogonality of the father and mother wavelet simplifies to a condition on the normalized refinement coefficients since we have

$$\langle \phi(t), \psi(t) \rangle = \left\langle \sum_k h_k \sqrt{2}\phi(2t-k), \sum_k g_k \sqrt{2}\phi(2t-m) \right\rangle$$

$$= \sum_k h_k \sum_m g_m \left\langle \sqrt{2}\phi(2t-k), \sqrt{2}\phi(2t-m) \right\rangle$$
$$= \sum_k h_k g_k \left\langle \sqrt{2}\phi(2t-k), \sqrt{2}\phi(2t-k) \right\rangle +$$
$$\sum_{k \neq m} h_k g_m \left\langle \sqrt{2}\phi(2t-k), \sqrt{2}\phi(2t-m) \right\rangle$$

$$= \sum_k 2 h_k g_k \left\langle \phi(2t-k), \phi(2t-k) \right\rangle$$

$$= \sum_k h_k g_k$$

Therefore the orthogonality condition becomes,

$$\sum_k h_k g_k = 0$$

In addition, in order to ensure a *lossless* signal decomposition and recomposition, a condition of orthogonality between the $\psi(t-k)$ functions must also be established. (The lossless signal processing will be further explored later).

$$\langle \psi(t-k), \psi(t-m) \rangle = \left\langle \sum_i g_i \sqrt{2}\phi(2t-2k-i), \sum_j g_j \sqrt{2}\phi(2t-2m-j) \right\rangle$$

$$= \sum_{i,j} 2 g_i g_j \left\langle \phi(2t-(i+2k)), \phi(2t-(j+2m)) \right\rangle$$

$$= \sum_{i+2k=j+2m=q} 2 g_i g_j \left\langle \phi(2t-q), \phi(2t-q) \right\rangle$$

$$= \sum_i g_i g_{i-2(m-k)}$$

Since all terms disappear except for those where the son wavelets are not superposed (due to the orthogonality of the wavelet sons of the same generation), the condition becomes the above equation.

Again, an orthogonality condition imposed upon the mother wavelet and its translates reduces to a condition on its refinement coefficients only:

$$\sum_i g_i g_{i-2(m-k)} = \begin{cases} 0, & \text{if } k \neq m \\ 1, & \text{if } k = m \end{cases}$$

From the above two conditions, it is possible to establish a relation between the father and mother wavelet. In particular, it is possible to establish a relation between their refinement coefficients. Although there are multiple possibilities, the generally adopted relation is described as follows:

$$g_k = (-1)^k h_{1-k}$$

Orthogonal Spaces - Parents Rivalry

Note that the orthogonality and the nested sequence of spaces spanned by the $\{\psi(2^n t - k)\}$ wavelet functions can be shown with the same procedure as for with the father and son wavelet functions. In addition, the orthogonality established and imposed between the father and mother wavelet carries to the sons and daughter wavelets. Consequently, the spaces spanned by the $\{\psi(2^n t - k)\}$ wavelets will all be orthogonal to the spaces spanned by their counterpart siblings of the same generation. For instance, given that $\{\phi(t-k)\}$ is an orthogonal basis that spans the inner product space V_0,$\{\psi(t-k)\}$is an orthogonal basis that spans a perpendicular inner product space $V_0^{\perp}$. Consequently, since V_0 is a subspace of V_1 (as established previously via the nested sequence of spaces), it is possible to express V_1 in terms of V_0 with the knowledge of the ***Orthogonal Decomposition Theorem,*** which states that if W is a finite-dimensional subspace of an inner product space V, then V can be written *uniquely* as $V = W \bigoplus W^{\perp}$.

In fact, the orthogonal decomposition of V_1 is as follows:

$$V_1 = V_0 \bigoplus V_0^{\perp}$$

In general, for each vector space V_n spanned by the son wavelets $\{\phi(2^n t - k)\}$, there is a perpendicular inner product space $V_n^{\perp}$ spanned by daughter wavelets $\{\psi(2^n t - k)\}$. Proceeding inductively, the general case can be established as the following:

$$\begin{aligned} V_n &= V_{n-1} \bigoplus V_{n-1}^{\perp} \\ &= (V_{n-2} \bigoplus V_{n-2}^{\perp}) \bigoplus V_{n-1}^{\perp} \\ &= \ldots \\ &= V_0 \bigoplus V_0^{\perp} \bigoplus V_1^{\perp} \bigoplus V_2^{\perp} \bigoplus \ldots \bigoplus V_{n-1}^{\perp} \end{aligned}$$

The most important piece of information from this analysis of spanning sets and spaces spanned by various wavelet generations is that with a family of wavelets, we have a complete set of orthogonal bases at our disposition. This orthogonality becomes particularly important when we need to decompose and recompose signals and data using wavelets. In fact, it will be shown further in discussion below that father and son wavelets handle the *averaging* of signals, whereas the mother and daughter wavelet handle the *differencing* of signals during the decomposition of the signals. Hence, the orthogonality between all the members of a wavelet family guarantees that during the decomposition of the signal, the data will be neatly cut into orthogonal pieces instead of being intermingled at the endpoints of a truncation. This in turn guarantees a *lossless* recomposition of signal, which means that there isn't any loss of information between the original signal and the processed signal, given that thresholding has not been applied to the decomposed data.

High and Low Pass Filters

Now that the fundamental properties of wavelet functions are settled, the discussion can now move onto the actual processing of signals using the wavelet functions. The sequences of refinement coefficients $\{h_k\}$ and $\{g_k\}$from the dilation equations

$$\phi(t) = \sum_k h_k \sqrt{2}\phi(2t-k) \quad \text{and} \quad \psi(t) = \sum_k g_k \sqrt{2}\phi(2t-k)$$

play very important roles in wavelet transforms. In fact, the essence of wavelet processing lies in a sequence of *weighted averaging* and *weighted differencing* of the signals through filters defined by the refinement coefficients $\{h_k\}$ and $\{g_k\}$. In particular, the k-th entry of the *Low Pass filter,* which performs the weighted averaging, is defined by

$$(Hs)_k = \sum_{j=o}^{2^n-1} h_{j-2k} s_j \quad ,$$

whereas the k-th entry of the *High Pass filter,* which performs the weighted differencing, is defined by

$$(Gs)_k = \sum_{j=o}^{2^n-1} g_{j-2k} s_j \quad .$$

The name Low Pass Filter is given to the weighted averaging operator because it produces a smoother, shorter signal of *lower frequency.* The smoothed out information from the original signal is retained by the High Pass Filter, whose name is given for the *higher frequency* noise data it picks up. Note that these filters are defined for all families of wavelets. However, it is instructive to analyze them with simple wavelet families. For a signal $s = [s_0, s_1, \cdots, s_{2^n-1}]$ of length 2^n, applying with *Haar* wavelets yields, in general:

$$(Hs)_0 = h_0 s_0 + h_1 s_1 = \frac{\sqrt{2}}{2}(s_0 + s_1)$$

$$(Hs)_1 = h_0 s_2 + h_1 s_3 = \frac{\sqrt{2}}{2}(s_2 + s_3)$$
$$\vdots$$
$$(Hs)_{2^{n-1}} = h_0 s_{2^n-2} + h_1 s_{2^n-1} = \frac{\sqrt{2}}{2}(s_{2^n-2} + s_{2^n-1})$$

The concept of "*weighted averaging"* is illustrated here, as we can see that the first term that comes out from the first Low Pass filter is the normalized arithmetic mean of the first two terms of the original signal, and the second term of the filtered signal is the normalized arithmetic mean of the third and fourth term of the original signal. However, the reason that makes the weighted average appear to be a simple arithmetic mean is that for the *Haar* wavelet, $h_0 = h_1 = 1$. In other words, although a weighted average is being taken, the weight that each component takes is the same. This is only true for the *Haar* wavelet. In general, depending on the values of the sequence $\{h_k\}$, each component of the original signal takes a different weight when the average is taken. Weighted averaging is particularly efficient when processing signals with lots of sharp signals since more weight will be allotted around the spike of the signal and less weight is assigned for the endpoints of the spike. Proceeding in such a way will allow preservation of a greater amount of information when taking averages.

A similar situation occurs with the High Pass filters. For the *Haar* wavelet,

$$(Gs)_0 = g_0 s_0 + g_1 s_1 = \frac{\sqrt{2}}{2}(s_0 - s_1)$$
$$(Gs)_1 = g_0 s_2 + g_1 s_3 = \frac{\sqrt{2}}{2}(s_2 - s_3)$$
$$\vdots$$
$$(Gs)_{2^{n-1}} = g_0 s_{2^n-2} + g_1 s_{2^n-1} = \frac{\sqrt{2}}{2}(s_{2^n-2} - s_{2^n-1})$$

Again, processed signal seems to be simple normalized averaged differences of the original signal. However, in general, depending on the sequence $\{g_k\}$, the individual components will take different weights when the differences are taken. Note that the length of a signal will be halved after it passes through the High Pass filter. The same story goes for the Low Pass filter. Further signal processing is then done on the *averaged signal* only; the signal coming out of the High Pass filter is left as is. Newly averaged signals are then iteratively averaged and differences from that average is iteratively taken until the averaged signal is of length 1 (and thus of dimension 1), where no further averaging or differencing can be performed. This sequence of signal processing is known as the Pyramid algorithm and is illustrated by the following diagram

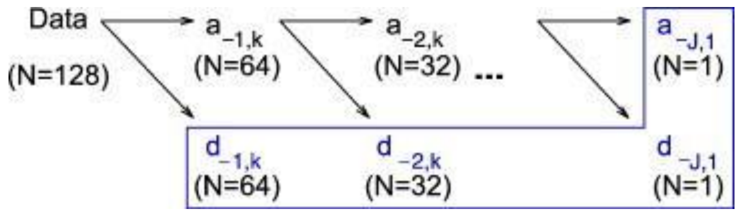

Fig. 2. Pyramid Algorithm Diagram

where $a_{n,k}$ is the weighted average taken at the n-th level, and $d_{n,k}$ is the difference of the previous signal from the average of the same level. N indicates the number of terms (length) of $a_{n,k}$ and $d_{n,k}$ at each level. The concept of analyzing the signal with different resolution is clearly illustrated here, where the later applied filters analyze the signal at larger and larger scales, whereas the earlier applied filters analyze the signal in its minute details. Finally, thoroughly processed signals from each level are then assembled into a string of length 2^n in the following fashion:

$$s^\star = [a_n, d_n, \cdots, d_2, d_1] = \left[H^{2^n} s, GH^{2^n-1}, \cdots, GH^2 s, GHs, Gs\right]$$

In essence, wavelet forward transforms is implemented as the following:

1) Take the weighted average of the signal
2) Take the difference between the original signal and the averaged signal
3) Treat the averaged signal as a new original signal (with halved length) and repeat 1) and 2) until the averaged and differenced signal is of length 1.

The elements of the processed signal are also called *wavelet coefficients.* The first wavelet coefficient is the average of the average....of the average. The remaining coefficients indicate how far away the actual data is from the average taken at each individual level. Therefore, wavelet coefficients are indications of how much detail a particular element of the signal carries. Consequently, eliminating elements with smaller corresponding wavelet coefficients will be of little impact on the integrity of the signal. This process of eliminating smaller wavelet coefficients in order to make more of them 0 is called *thresholding.* Introduction of 0’s leads to significant compression of signal size since strings of 0’s can be easily and efficiently compressed.

Thresholding

There are 3 types of thresholding: hard thresholding, soft thresholding and quantile thresholding.

Hard thresholding substitute all the coefficients whose absolute value is below the selected *tolerance* with zero.

Soft thresholding does the same as hard thresholding, but in addition, all the entries are shifted towards 0 by the same tolerance.

Quantile thresholding ignores the smallest p percent of entries, where p is selected as tolerance.

Dual Operators-Reverse High and Low Pass Filters

After thresholding, wavelet coefficients must be recomposed into a new signal that should be highly similar to the original signal. The k-th entry of the reverse Pyramid algorithms for High Pass filter and Low Pass filter, called *dual operators,* are defined as follow,

$$(H^\star s^\star)_k = \sum_j h_{k-2j} s_j^\star$$
$$(G^\star s^\star)_k = \sum_j g_{k-2j} s_j^\star$$

Again, an example is instructive. Applying the dual operators for the *Haar* wavelets, the last recomposition is as follows:

$$H^\star a_1 = [h_0 a_{1,0}, h_1 a_{1,0}, h_0 a_{1,1}, h_1 a_{1,1}, \cdots, h_0 a_{1,2^{n-1}-1}, h_1 a_{1,2^{n-1}-1}]$$
$$G^\star d_1 = [g_0 d_{1,0}, g_1 d_{1,0}, g_0 d_{1,1}, g_1 d_{1,1}, \cdots, g_0 d_{1,2^{n-1}-1}, g_1 d_{1,2^{n-1}-1}]$$

where $\{a_{1,k}\} = a_{1,0}, a_{1,1}, \cdots, a_{1,2^{n-1}-1}$ are the elements from the *first* averaging and $\{d_{1,k}\} = d_{1,0}, d_{1,1}, \ldots, d_{1,2^{n-1}-1}$ are the elements from the *first* differencing from that average. Note that the dual operators are applied backwards, therefore the a_1 and the d_1 terms are used last. In addition, note that for every k, the $a_{1,k}$ and the $d_{1,k}$ terms appear in two terms as the signal passes through the dual operators. Consequently, a signal's length will be doubled after dual operators are applied. In general, for *Haar* wavelets, applying from $k = n$,which represents the last level of averaging and differencing, therefore first signal to be recomposed, to $k = 1$, which represents the first level of averaging and differencing, therefore the last signal to be recomposed:

$$H^\star a_k = \left[h_0 a_{k,0}, h_1 a_{k,0}, h_0 a_{k,1}, h_1 a_{k,1}, \cdots, h_0 a_{k,2^{n-k}-1}, h_1 a_{k,2^{n-k}-1}\right]$$
$$G^\star d_k = \left[g_0 d_{k,0}, g_1 d_{k,0}, g_0 d_{k,1}, g_1 d_{k,1}, \cdots, g_0 d_{k,2^{n-k}-1}, g_1 d_{k,2^{n-k}-1}\right]$$

Although from the processed signal there is only one term that represents the average whereas the rest of the elements represent differences, as the signal gets iteratively recomposed, the number of terms which represent averages will double at each step. This is shown as the length of a_k and the d_k doubles at each step from $k = n$ to $k = 1$.The dual operators serve as reverse operators of the High and Low Pass filters in the sense that they undo the process of weighted averaging and weighted differencing according to the same weight with which they were originally averaged and differenced. These backward processed averages and differences are then added together to recompose the signal for the averages of the previous level.

Projection on Orthogonal Spaces

Another way to look at the wavelet transforms is that the averages and differences are the projections and the residuals of when decomposing signal into orthogonal spaces. Recall that it is previously established that wavelet daughters occupy orthogonal spaces to their siblings of the same generation:

$$\begin{aligned} V_n &= V_{n-1} \bigoplus V_{n-1}^{\perp} \\ &= (V_{n-2} \bigoplus V_{n-2}^{\perp}) \bigoplus V_{n-1}^{\perp} \\ &= \ldots \\ &= V_0 \bigoplus V_0^{\perp} \bigoplus V_1^{\perp} \bigoplus V_2^{\perp} \bigoplus \ldots \bigoplus V_{n-1}^{\perp} \end{aligned}$$

With this approach, the decomposition of a signal located in the inner product space V_n starts by projecting this signal onto the space V_{n-1} spanned by the son wavelets of the appropriate level. Since V_{n-1} is smaller and is subspace of V_n, according to the Orthogonal Decomposition Theorem, there must be a residual that lies in the orthogonal space $V_{n-1}^{\perp}$. This residual indicates how far away the projection onto V_{n-1} is from the actual signal in V_n. In other words, the projection onto V_{n-1} consists of the first level of weighted averages, whereas the residual in the space $V_{n-1}^{\perp}$ consists of the differences. The projection in V_{n-1} is then further projected onto a even smaller space V_{n-2} and another residual in $V_{n-2}^{\perp}$ is obtained. The process is repeated iteratively until the dimension of both the projection and the residual is 1. It is clearly shown that all the perpendicular spaces $V_k^{\perp}$ spanned by the daughter wavelets house the "differences", whereas the averages are mapped onto the spaces V_k spanned by the son wavelets. This concurs with our previously association of the $\{h_k\}$ sequence with the son wavelets and the averages and the association of the $\{g_k\}$sequence with the daughter wavelets and the differences. The idea of multiresolution analysis is again illustrated here, where the signal is analyzed at multiple resolutions simultaneously.

Other Families of Wavelet

There are currently a great variety of wavelet families readily available for application, such as the ones listed in Fig. 3. However, wavelets are in fact specifically crafted to suit the particular type of data to be processed. The folder *Cranking_the_machine* contains Maple files showing how wavelet functions are crafted according to particular specifications on the refinement coefficients. It is interesting to note that however diverse and different wavelet families appear to be from each other, they are in essence very similar one from another. In fact, most wavelet families are crafted based on a trade-off between *smoothness* and *compact support* of the scaling/wavelet function. Put simply, the smoothness of a scaling function allows better approximation with polynomials, whereas the compact support condition ensures a simpler orthogonality relation. A balanced combination of the two criteria is thus necessary to create a well performing wavelet family. In Fig. 3, *Haar* is the family that demonstrates the compact support characteristic the most evidently. However, it lacks the smoothness characteristic completely. The *Meyer* family, on the other hand, is very smooth, but it does not go all the way to zero at its endpoints.

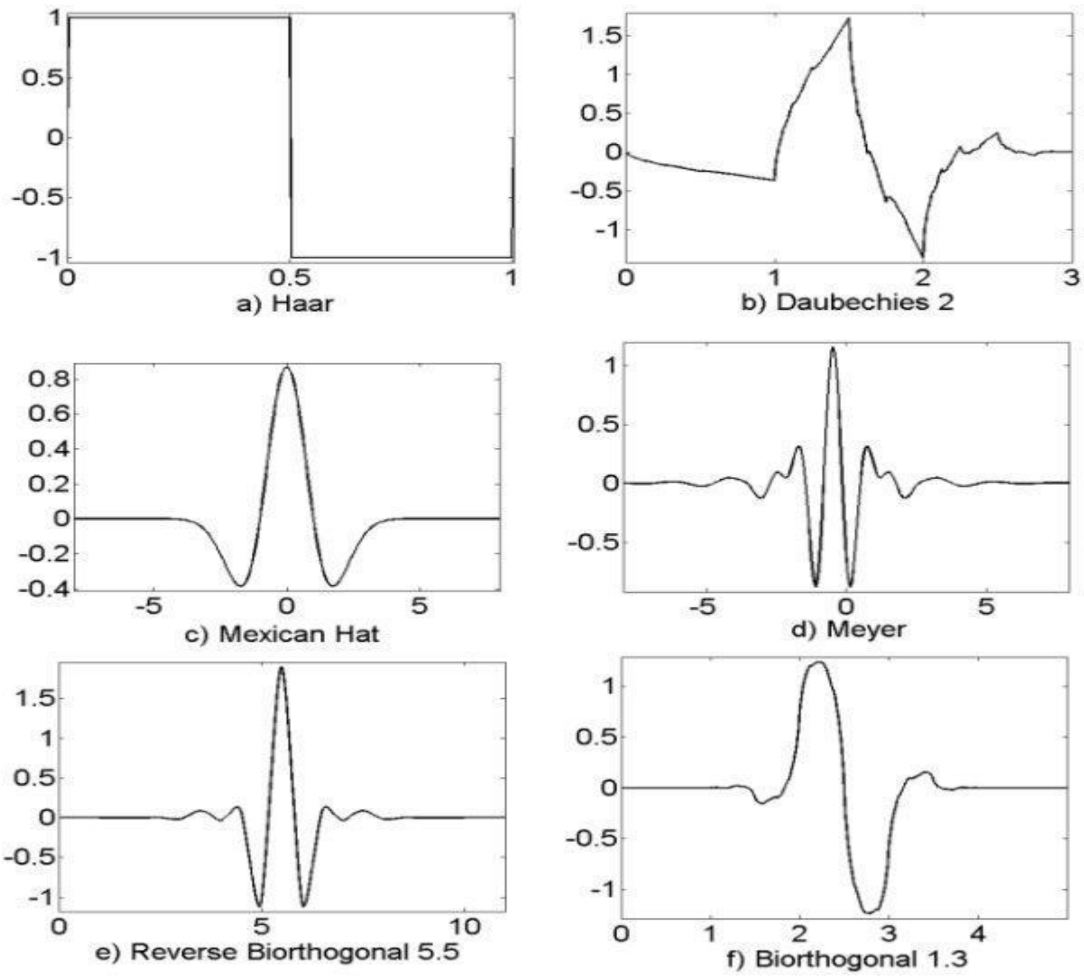

Fig. 3. Wavelet Functions (Mother Wavelets) of Other Wavelet Families

Crafting Wavelet Families

In order to generate families of wavelets with desired properties, such as compact support and orthogonality of the functions, some criteria must be met for their refinement coefficients. Consider the case of 4 refinement coefficients, then the following procedure illustrates the crafting of wavelet families from the set requirements.

Given that $\int_{-\infty}^{\infty} \phi(t)dt \neq 0$, a fact that can be proven with Fourier analysis, and the dilation equation, it can be shown that $\sum_{k=0}^{3} c_k = 2$ for a wavelet function with 4 refinement coefficients.

$$\int_{-\infty}^{\infty} \phi(t)dt = \int_{-\infty}^{\infty} \left[\sum_{k=0}^{3} c_k \phi(2t-k)\right] dt$$

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_{k=0}^{3} c_k \cdot \int_{-\infty}^{\infty} \left[\phi(2t-k)\right] dt$$

$$\int_{-\infty}^{\infty} \phi(t)dt = \sum_{k=0}^{3} c_k \cdot \frac{1}{2} \int_{-\infty}^{\infty} \phi(t)dt$$

$$\sum_{k=0}^{3} c_k = 2$$

From this condition and the previously established two conditions on the refinement coefficient,

$$\sum_{k=-\infty}^{\infty} c_k c_{k+2j} = \begin{cases} 0 \text{ if } j \neq 0 \\ 2 \text{ if } j = 0 \end{cases}$$

a system of 3 equations with 4 unknowns is obtained. Solving the system of equations, the 4 refinement coefficients can be expressed in terms of a parameter θ, as listed below:

$$c_0 = \frac{1}{2}(1 - cos(\theta) + sin(\theta))$$
$$c_1 = \frac{1}{2}(1 + cos(\theta) + sin(\theta))$$
$$c_2 = \frac{1}{2}(1 + cos(\theta) - sin(\theta))$$
$$c_3 = \frac{1}{2}(1 - cos(\theta) - sin(\theta))$$

Therefore, by varying the value of θ, refinement coefficients of various wavelet families are obtained. In particular, the *Haar* wavelet is generated when the value of θ corresponds to Pi/2, and the *Daubechies 4* wavelet is generated when the value of θ corresponds to Pi/3. The first animation in the Maple file *theta_wave* illustrates how the scaling function of a 4 coefficient wavelet family changes as the value of θ varies from 0 to 2Pi. Although the scaling functions adopt radically different shapes, varying from a simple box for *Haar,* to a almost fractal curve around *Daubechies 4,* to a box-shaped curve filled with sharp spikes when the value of θ is at Pi, all of them perform a reasonably decent job when analyzing digital signals that are constant over small intervals. The compatibility of these wavelet families with digital signals is explained by a common characteristic shared amongst all these wavelet families: they all have a compact support. Therefore, these wavelet functions are expected to perform poorly when analyzing smoother analog signals that take shape of polynomials. The Maple files *theta_test, D4_complete,* and *image_Haar* illustrate how a 16 pixel by 16 pixel image is processed by these three different wavelet families. Although they perform very similarly when analyzing simple digital files like the ones generated by PixelImage, the *Daubechies 4* wavelet is considered to be the most elegant and the most versatile of them all because it has the smoothest curve in addition of having a compact support. The above two properties grant the family the ability to approximate polynomials fairly well while keeping the decomposition orthogonal, therefore lossless.

In a similar fashion, the *Daubechies 6* wavelet with 6 refinement coefficients can be determined. In fact, on top of the 3 conditions of orthogonality that the *Daubechies 4* wavelet has to satisfy for orthogonality, the *Daubechies 6* wavelet is designed to be able to process even smoother functions such as constant functions, linear functions and quadratic functions. Therefore, from the following three conditions and the dilation equation for the mother wavelet and the previously established 3 conditions, we obtain a system of 6 equations with 6 unknowns.

$$\int_{-\infty}^{\infty} \psi(t)dt = 0, \int_{-\infty}^{\infty} t\psi(t)dt = 0, \int_{-\infty}^{\infty} t^2\psi(t)dt = 0$$

$$\begin{aligned}
&c_0 + c_1 + c_2 + c_3 + c_4 + c_5 = 2\\
&c_0^2 + c_1^2 + c_2^2 + c_3^2 + c_4^2 + c_5^2 = 2\\
&c_0 \cdot c_2 + c_1 \cdot c_3 + c_2 \cdot c_4 + c_3 \cdot c_5 = 0\\
&-c_0 + c_1 - c_2 + c_3 - c_4 + c_5 = 0\\
&-c_0 + c_2 - 2 \cdot c_3 + 3 \cdot c_4 - 4 \cdot c_5 = 0\\
&-c_0 - c_2 + 4 \cdot c_3 - 9 \cdot c_4 + 16 \cdot c_5 = 0
\end{aligned}$$

The derivation for the last three equations is omitted here. However, it is strongly advised to verify them using proper integration techniques and the dilation equation. Solving the system of equations numerically using Maple 16, the 6 refinement coefficients are obtained. Using these coefficients, it is then possible to again “crank up the machine” to determine the shape of the scaling function. Note that now since the *D6* wavelets are *defined* in such a way that it is better at approximating quadratic functions, the curve of the scaling function is smoother than that of the *D4* wavelet. Please consult the Maple file *D6* in the file *cranking_the_machine.*

Processing with *D6* wavelets essentially follows the same procedures as that of the *D4* wavelets. However, due to the extra 2 refinement coefficients, the *D6* wavelets can only fully process signals of length or of sides that are multiple of 12. For this reason, it is impossible to fully process images generated by PixelImage, which are squares of 16 pixel on each side. An example can be shown for a 12 by 12 matrix or a 24 by 24 matrix, but due to the lack of program to display the image generated, this step is omitted in this article.

Examples of Image Processing with Wavelets

Given a one-dimensional signal of length 2^n, it is possible to describe all the Low and High Pass Filter processing by a series of matrix multiplication. Similarly, the same matrices can perform the processing on a two-dimensional signal (an image) with sides of length 2^n. This is shown in the Maple files *D4_complete* and *Haar_image* attached with the report. In addition, in the file *Haar_wavelet_conversion_process*, it is shown how the wavelet coefficients conversion matrix is obtained with the knowledge of the refinement coefficients. The same concept is then used to derive the wavelet coefficient conversion matrix for the *Daubechies 4* wavelet in the beginning of the file *D4_complete.* However, due to the two extra overlapping refinement coefficients of the *D4* wavelets (3rd and 4th coefficient), an additional shift operator must be considered and applied at each step of conversion. Please refer to the Maple files for further details.

The folder “NewImages” contain images processed by different wavelets, namely *Haar, Daubechies 4* and other 4 coefficients wavelets generated by the Maple file *theta_wave.* Note that all wavelets generated from *theta_wave* are able to process signals, except when theta is equal to Pi, in which case the conversion matrix becomes singular (noninvertible). To have a better idea of compression ratio differences between different wavelet families, the numbers of wavelet coefficients that are set to zero according to the family are listed below (for a hard thresholding of tolerance=10). For *Haar,* 33 wavelet coefficients are set to 0; for *Daubechies 4*, 38; for wavelets generated with the algorithms in *theta_wave* with theta=Pi/4, 36 coefficients are eliminated; and for the wavelets generated with theta=0.99*Pi, 61 wavelet coefficients are set to 0. Observe that all the the processed images have smoother, more homogenized tones in all its shades while preserving the shape of the flower quite accurately.

Applications of Wavelets

Due to their powerful signal processing properties, wavelets are widely used in a great number of domains. For instance, wavelets can be used to denoise plots of stock prices in order to determine and predict future trends of the market; wavelets can be applied to multimedia files (audio or video recordings) to get rid of the background noise; wavelets can be used to compress file size of FBI fingerprints database, etc. In brief, wavelets are a fundamental and powerful tool that can find applications virtually anywhere, especially in this Digital Era.

Conclusion

In summary, wavelets are a special class of functions that are specially crafted to filter noise data efficiently in order to compress the size of files. The efficiency of this processing is mainly attributed to 2 crucial properties of wavelets. The first characteristic is the virtually infinite set of functions at different resolutions and positions that wavelet families offer when analyzing signals. The second property is the orthogonality of these functions, which allows a lossless processing.

In fact, not only do wavelets reduce the size of data considerably for easier storage, searching and retrieval, stripping the data from the random fluctuations that are generally known as noise exposes the “real” trend of time-series data, which is crucial for more accurate extrapolation and prediction.

References

Aboufadel, Edward, and Steven Schlicker. *Discovering Wavelets*. New York: Wiley, 1999. Print.

Graps, Amara. "An Introduction to Wavelets." *Agua Sonic*. N.p., n.d. Web. 16 May 2013. <http://aguasonic.com/Wavelets/Introductions/IEEEwavelet.pdf>.

Kaplan, Ian. "Wavelets and Signal Processing." *Wavelets and Signal Processing*. N.p., n.d. Web. 16 May 2013. <http://www.bearcave.com/misl/misl_tech/wavelets/>.