

Wavelet Transforms and Signal Processing

Jesse Connell

December 2014

1 Overview

An explanation of wavelets and their construction, use, and meaning can be challenging, simply because the field is so broad. As Strang and Nguyen explain, perhaps a little blithely, “a wavelet is a small wave” [SN97, p. 26]. This simple phrase does capture the essential requirements for a wavelet: a nonzero function with zero mean, and with localization in time and frequency. When used as vectors spanning a function space, wavelets allow functions to be projected onto them in order to learn from or work with the functions in more useful ways. In particular, their localization in *time* separates wavelet methods from classic Fourier techniques. Other than that, very few of the rules hold strict in all cases: Characteristics like orthogonality and compact support may or may not be present, and the dyadic tiling in the time/scale plane of the “classic case” is not necessarily the only structure possible.

In this limited space we will restrict ourselves to a subset of all these possibilities. In particular we will focus on examples of 1D functions of time, like audio or bioelectric signals, though the topic easily expands to signals in higher dimensions and other variables, like images across two spatial dimensions or video. We focus on the simplest “dyadic” tiling in the time/frequency plane, corresponding to a logarithmic or “octave” decomposition [SN97]. Numerical examples given here were created with MATLAB R2014b, using built-in tools and the Wavelet Toolbox. (The encyclopedic depth of MATLAB’s tools makes them invaluable for exploring the topic, but open-source tools like Scientific and Numeric Python and community projects like PyWavelets [Was12] are entering the space as well. It will be interesting to see their progression in the near future.)

In terms of introductory texts, the breadth of interpretations and perspectives on the topic is striking. In *Wavelets and Filter Banks*, Strang and Nguyen present wavelets hand-in-hand with filter bank design, with matrix operations taking a lead role [SN97]. They point out Stéphane Mallat’s discovery of this filter bank connection [SN97, p. 33], but Mallat’s own *Wavelet Tour of Signal Processing* draws largely from a multiresolution perspective [Mal99]. Goswami and Chan’s *Fundamentals of Wavelets* blend Hilbert space and multiresolution perspectives with matrix representations for discrete time cases [GC99], and Burrus, Gopinanth, and Guo’s *Introduction to Wavelets and Wavelet Transforms* alternates filter bank and Hilbert space viewpoints [BGG98]. All of these perspectives yield insight on wavelets, and we will attempt to draw on some of that insight here.

Our goal here is to build up enough of the fundamental aspects of wavelets and wavelet transforms so that the idea can be applied to an example case. We first cover the basic motivation in going beyond Fourier analysis, and explore the multiple viewpoints on the topic. We include a brief comparison of some of the simplest wavelet systems to emphasize the inherent trade off in time and frequency with any analysis method. We end with a basic attempt at using the wavelet transform in a practical signal processing application: denoising an acquired biological signal. We also touch on an example in the literature of applying wavelets for the compression, as opposed to denoising, of these signals.

2 Fundamentals

2.1 Motivation

Mallat refers to the “indisputable hegemony of the Fourier transform” with its reliance on a time-invariant operator, despite our interest in the transient parts of a signal [Mal99, p. 1]. The ability to pick transients out of a time-varying signal has a multitude of examples, but to give one: consider a signal with sharp peaks at specific times combined with constant tones across all times. By eye or ear, we can see or hear the “clicks” in the signal and separate them from the constant tones. The usefulness of wavelets can be phrased as the ability to match that intuition with mathematical rigor.

We are used to the idea of presenting a logarithmic frequency axis for Fourier transforms, but the transform itself produces a linear range of evenly-spaced frequencies. This does *not* match our own intuition about what constitutes a logical spacing of frequencies! An “even spacing” of frequencies when moving from one note to a higher octave and to the next higher octave is a *doubling* of frequencies with each increase. As the equally-spaced lines on the staff of written music reveal, what intuitively appears to be an equal frequency range compared to the one below is actually a range *twice as large*. This implies that our own “frequency resolution” decreases logarithmically as frequency increases. In his article on Wavelets for American Scientist, Strang explains the doubling that occurs in each increase of scale in a wavelet representation in terms of a set of instruments contributing to a symphony.

In her *Ten Lectures on Wavelets*, Ingrid Daubechies provides a visual explanation of the challenges of achieving good localization in both time and frequency [Dau92, Fig. 1.3], which Goswami and Chan refer to for their own example [GC99, Fig. 4.5]. Following suit, this concise example is worth repeating here.

Figure 1 shows a time-varying signal composed of three tones and three impulses, given by

$$f(t) = \sin(2\pi v_1 t) + \sin(2\pi v_2 t) + \sin(2\pi v_3 t) + \alpha(-2\delta(0) + \delta(t_1) - \delta(t_2))$$

Where $\alpha = 3$, $v_1 = 500\text{Hz}$, $v_2 = 1\text{kHz}$, $v_3 = 2\text{kHz}$, $t_1 = -2\text{ms}$, $t_2 = 2\text{ms}$, $\delta(t)$ being the Dirac delta. Following the original example, our function was sampled at 8 kHz and $\delta(t)$ was approximated with

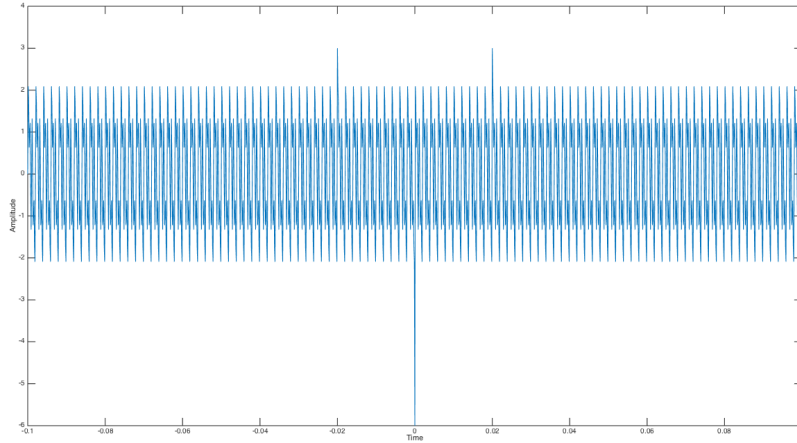
$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n = 1 \end{cases}$$

(The frequencies and amplitudes are chosen so that $f(t)$ has zero-mean and the sine wave frequencies are evenly spaced on a logarithmic scale.)

Figure 2 shows a set of short-time Fourier transform plots of f , using hamming windows with 50% overlap. In each plot the window width is halved. For very long windows, frequency resolution is dominant, and the transform approaches a traditional FFT. The impulses are nowhere to be seen. Since time resolution is so poor, they only appear in the form of an even contribution at all frequencies, as implied by their own Fourier transform. As window length is cut in half at each step, the pure sine waves become doubly smeared out in frequency (and the maximum value as shown in the color bar correspondingly drops by 50%.) But now the impulses become visible; we can simultaneously see their even spread across frequencies on the vertical axis and their occurrence in time on the horizontal. As we continue to improve our time resolution with smaller windows, we improve our localization of the impulses, but at the expense of frequency resolution. Finally the separation between the three constant tones is lost, just as we have finally achieved good localization of the impulses in time. In the analysis of a signal, why can’t we separately consider both the tones we hear over longer times, and the events we detect over shorter ones? We can, and in essence that analysis *is* the wavelet transform.

Figure 3 shows the same signal considered earlier, processed with a continuous wavelet transform, using a Morlet wavelet. The idea of giving these different times and scales of detail their own time-varying functions, and considering the subspaces these functions span relates to the multiresolution analysis perspective. Our analysis at one level of detail can contribute to our analysis at the following level, with this iterative process implemented in filter bank design. Each of these views contributes to a cohesive and powerful method for signal processing.

Figure 1: Signal composed of three constant tones and three clicks



2.2 Hilbert Space and Multiresolution Viewpoint

Even if the practical aspects of implementing a wavelet system computationally occurs in discrete, finite dimensions, an infinite-dimensional representation can provide key insights into the ideas. Viewing a set of wavelet functions as vectors in a Hilbert space allows us to use the properties of orthogonality between vectors, the norm of a vector, and a change of basis vectors, applying them to continuous signals in time or space.

In a classic Fourier transform, each component vector in the basis is a different complex exponential with the relation between the components given by the frequency ω : $f(\omega) = e^{i\omega t}$. Since the inner product of any one of these functions with any other is zero, they are orthogonal to one another:

$$\langle f(\omega_1), f(\omega_2) \rangle = \int_{-\infty}^{\infty} f^*(\omega_1) \cdot f(\omega_2) dt = 0, \omega_1 \neq \omega_2$$

The norm or “length” of a component is not unity:

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int_{-\infty}^{\infty} e^{-j\omega t} e^{j\omega t} d\omega} = \infty$$

But putting all these together, we do have a sort of orthonormality in this infinite-dimensional frequency domain:

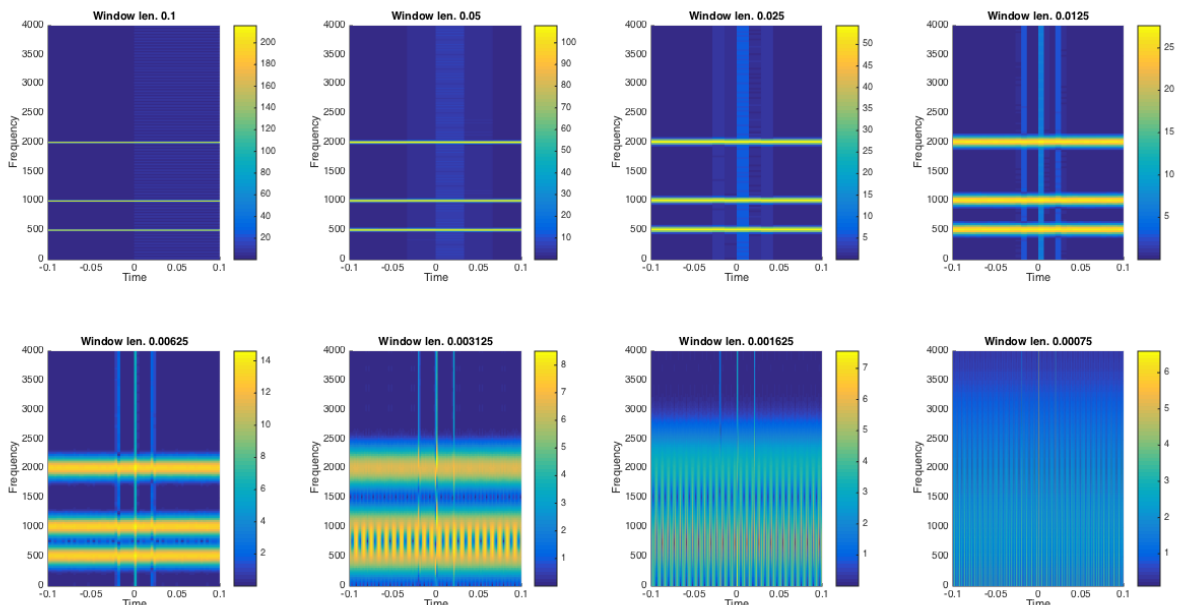
$$\langle f(\omega_1), f(\omega_2) \rangle = \int_{-\infty}^{\infty} f^*(\omega_1) \cdot f(\omega_2) dt = \delta(\omega_1 - \omega_2)$$

By projecting a signal onto this new basis, we have effectively rotated it onto a new set of axes, but have kept the length of our “vector” the same [SN97, p. xi]. As we can imagine geometrically in two dimensions (jumping ahead, Section 2.6 draws on this for examples), we can project onto a basis, scale the basis vectors using our projections, and sum the vectors for a round trip back to the original representation.

2.3 Alfréd Haar’s Creation

In 1909 Alfréd Haar published his PhD thesis, *Zur Theorie der orthogonalen Funktionensysteme*. Although terms like “wavelets” would make no appearance until decades later, Haar’s paper introduce what was later recognized to be a fundamental wavelet system. In Chapter 3, Haar names “the complete orthogonal function

Figure 2: STFT with half-overlapping Hamming windows of varying widths



system χ , the most simple representative of that class of orthogonal systems” [Haa09]. Roughly seventy years later, Daubechies names entire families of wavelet functions of which Haar’s wavelet can be seen as a starting point [Dau92, p. 16].

In Haar’s system, a particular family of piecewise linear functions can be used as a basis for representing any given function of finite energy. He used the pattern $\chi_n^{(k)}(s)$ with n and k representing what are now referred to as scale and time. $\chi_0(s)$ is simply 1 on the interval $[0, 1]$, and $\chi_1(s)$ is 1 from $[0, \frac{1}{2})$ and -1 from $(\frac{1}{2}, 1]$. In a more modern phrasing, $\chi_0(s)$ is $\phi(t)$, the scaling function, and $\chi_n^{(k)}(s)$ for $n > 0$ is $\psi_{j,k}(t)$, the wavelet function at each scale and time. As scale increases by integers, the wavelet functions at that scale become shorter in time by a factor of 2, and amplitude higher by a factor of $\sqrt{2}$ (preserving norm). The k parameter delays these shortened functions in time. Taken together, an expression relates the wavelet functions at higher scale to the lowest-scaled version, or *mother wavelet*:

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (1)$$

Haar’s insight was that, as this set of functions are orthogonal to one another and have unit norm, we can project a function onto these scaled and shifted χ functions and obtain a complete description of the function in this new domain. Here instead of t (or s), we have j, k (or n, k). A synthesis expression for a time-domain function could be written as

$$f(t) = \sum_{j,k} a_{j,k} \psi_{j,k}(t)$$

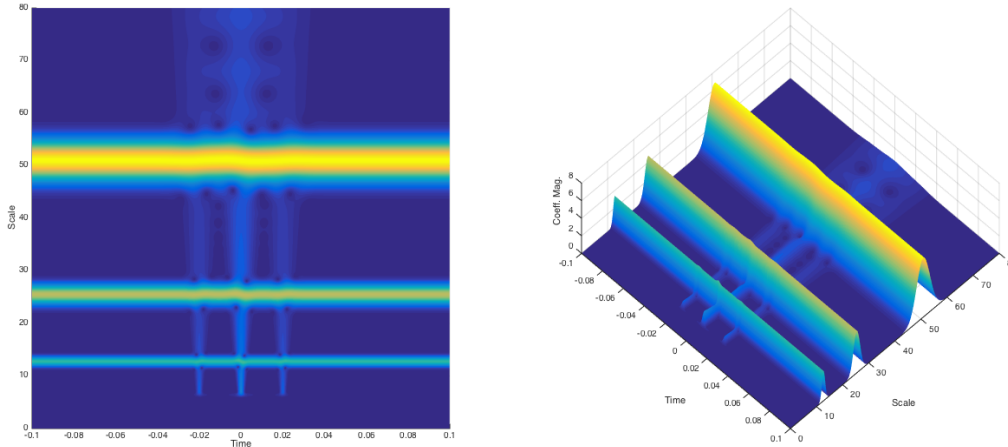
And thanks to the orthonormality of the basis functions, we can determine the scaling coefficients $a_{j,k}$ with inner products, or projections, onto those basis functions:

$$a_{j,k} = \langle \psi_{j,k}(t), f(t) \rangle$$

And finally, those Hilbert space projections have a simple integral form:

$$\langle \psi_{j,k}, f(t) \rangle = \int_{-\infty}^{\infty} \psi_{j,k}^*(t) f(t) dx$$

Figure 3: Magnitude of a continuous wavelet transform of $f(t)$ with Morlet wavelet



(In this classic orthonormal wavelet basis, the same basic idea applies as in the Fourier example earlier, but with a different change of variables instead of switching between pure time and pure frequency.)

Burrus, Gopinath, and Guo in particular use this to provide a foundation for the Hilbert space view of the transform [BGG98]. (This is fitting, as Hilbert was Haar’s thesis advisor [Soc]!) Starting from Haar’s simple scaling function, they introduce the vector spaces these scaled functions span. A set of these time-shifted box functions at any given scale clearly also span the same spaces at each lower scale. (Any function that could be built as the sum of a set of shifted boxes of one width could just as easily be built up using half-width boxes, for example.) This means that the scaling functions are *not* orthogonal, but rather, span nested spaces like Matryoshka dolls [BGG98, Fig. 2.1]; a signal that falls entirely within the space spanned by scaling functions at one scale could also be projected solely onto the scaling functions at any higher scale.

Figure 4 shows an electrocardiogram recording (obtained from the “PhysioBank ATM” database of physiological signals [Phy]). The original signal over ten seconds is shown on the bottom plot. The left four plots are projections onto the corresponding Haar scaling functions taken at higher and higher scales. The scaling factors are multiplied by each version of the scaling function and superimposed, to give an approximation of the signal. The right four plots show the signal projected onto the corresponding Haar wavelet functions at each of the same scales (and superimposed on the corresponding vectors, as with the left plots). On the left, increasing scale adds detail to the projection given at each lower scale, but also includes those same portions of the signal all over again. These functions allow us to see increasingly fine detail, but don’t perform any strict separation of the signal detail between scales. There is no method to make use of this as a signal processing technique with a “round trip” to an alternate basis and back to our original again.

Strang points out that in a discrete time-domain signal, we already have an implicit choice of basis [Str92]; this is simply the set of Dirac delta functions positioned at each possible time: $\delta[t - t_0]$, $t \in \mathbb{Z}$. Any function in the time domain can be shown as a summation of scaled versions of each of these impulses. What about an infinite-dimensional (continuous time) case? Imagining taking scale to ∞ in Burrus’ example and the example plots here on the left side, we know we should see our original signal represented again across time. Taking the limit of the scaling function shows this as well:

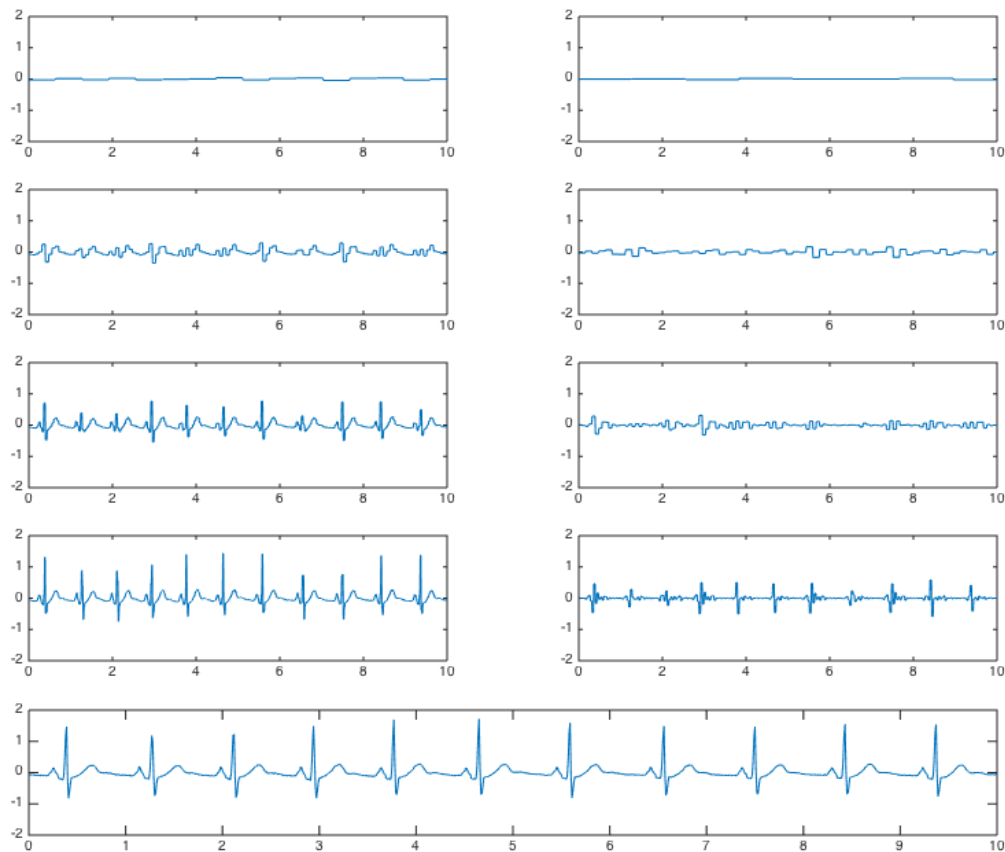
$$\lim_{j \rightarrow \infty} \phi_{j,k}(t) = \lim_{j \rightarrow \infty} 2^{j/2} (u(\frac{t}{2^j}) - u(\frac{t-1}{2^j})) = \delta(t - k)$$

(where $u(t)$ is the unit step function.) Then projecting our signal onto this basis simply picks out every

value across time, giving our original unique representation of the signal. The scaling function sets the stage for a change of basis, but when looking across scales we haven't really "gone" anywhere yet.

If the space of signals that could be described by scaling functions at one level of scale is incrementally larger than that at the next lower scale, we can gain more insight by looking at the *differences* between one space and the next: this is the space spanned by the corresponding wavelet functions at that scale! In this way the the projections onto the wavelets at each scale tell us *new* information, entirely separate from the scales below, because of the orthogonality of the wavelets between scales. This is what the four right plots in Figure 4 show– at each scale we have a new layer of information, separate from what came before.

Figure 4: An ECG recording projected with increasing scale onto Haar's scaling functions on the left, and Haar's wavelet functions on the right, with original signal plotted at bottom. (In practice the scaling function example was implemented by summing projections into the corresponding wavelet spaces up to each level; The wavelet function example shows each individual level of scale separately.)



2.4 Filter Bank Viewpoint

In the practical case of finite, discrete time, matrix notation and the corresponding iterative filter banks are both an effective explanation and a practical method of implementation. At first we might simply see

a finite list of values, for a vector in N-dimensional space: $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$. Our wavelet vectors, just like the functions before, are shifted and scaled versions of the mother wavelet: $\psi_{J,K} = 2^{J/2}\psi[2^J n - K]$ where $\psi[n]$, the mother wavelet, is of the lowest-scale in the space. Using the Haar wavelet, as before, this is $\psi[n] = [1/2 \ 1/2 \ -1/2 \ -1/2]^T$. Moving up to scale $J = 1$, we have two translations of a squeezed wavelet: $\psi_{1,0} = [\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}} \ 0 \ 0]^T$ and $\psi_{1,1} = [0 \ 0 \ \frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}}]^T$. Clearly two increments of scale are all we can sensibly have in this limited space, but the last missing piece is the scaling function itself at the “very bottom” in scale: $\phi = [1/2 \ 1/2 \ 1/2 \ 1/2]^T$

Following the geometric interpretation from before, we *could* project our signal vector onto each of these separately to get our coefficients. For the scaling term, we have:

$$\langle x[n], \phi[n] \rangle = \mathbf{x} \cdot \phi = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix} = \frac{1}{2} x_1 + \frac{1}{2} x_2 + \frac{1}{2} x_3 + \frac{1}{2} x_4$$

And likewise for the wavelet projections:

$$\langle x[n], \psi_0[n] \rangle = \mathbf{x} \cdot \psi_0 = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{bmatrix} = \frac{1}{2} x_1 + \frac{1}{2} x_2 - \frac{1}{2} x_3 - \frac{1}{2} x_4$$

$$\langle x[n], \psi_{1,0}[n] \rangle = \mathbf{x} \cdot \psi_{1,0} = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} x_1 - \frac{1}{\sqrt{2}} x_2$$

$$\langle x[n], \psi_{1,1}[n] \rangle = \mathbf{x} \cdot \psi_{1,1} = [x_1 \ x_2 \ x_3 \ x_4] \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} x_3 - \frac{1}{\sqrt{2}} x_4$$

With these four dot products, we have the complete set of projections onto our alternate basis. *But this isn't the right way.* We have repeated calculations across the projections. A logical first step in seeing this more clearly is to structure the individual vectors of the basis into a matrix, and the set of projections as the operation of this analysis matrix on our signal vector:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

This isn't substantially different, but it lets us start to make observations about the transform as a whole. Since each vector is orthogonal to each other vector and each has unit norm, the entire matrix is *orthogonal*, and with that property we get other useful information: the inverse matrix is just the transpose. If we start with our vector of coefficients and want to convert back into a time-domain signal, we simply use this inverse synthesis matrix to reverse the transform.

If we break the previous matrix multiplication into two steps, we can avoid repeating the same calculations when finding the coefficients:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Strang and Nguyen present this in a reduced notation [SN97, p. 31], emphasizing the sparsity of these individual matrices:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & & & & & \\ \frac{1}{2} & -\frac{1}{2} & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & & & & & \\ & \frac{1}{2} & \frac{1}{2} & & & & & & \\ & & -\frac{1}{2} & -\frac{1}{2} & & & & & \\ & & & \frac{1}{2} & \frac{1}{2} & & & & \\ & & & & -\frac{1}{2} & -\frac{1}{2} & & & \\ & & & & & \frac{1}{2} & \frac{1}{2} & & \\ & & & & & & -\frac{1}{2} & -\frac{1}{2} & \\ & & & & & & & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Using these sparse matrices allows the lower scale coefficients to be computed from the next higher scale, moving from maximum scale to zero. The highest-scale wavelet coefficients are left as-is after the first iteration, corresponding to the identity sub-matrix in the second iteration. The only real computation in the second step is in using the previous scaling terms to find the “lower detail” wavelet and scaling coefficients at the next lower scale. We can see the iteration even more clearly in an 8-dimensional case, while follow Strang and Nguyen’s notation of $r = \frac{1}{\sqrt{2}}$ (and noting that the higher exponents below yield *smaller* components across each row, keeping each norm at unity) to give the full analysis matrix:

$$\mathbf{A} = \begin{bmatrix} r^3 & r^3 & r^3 & r^3 & r^3 & r^3 & r^3 & r^3 \\ r^3 & r^3 & r^3 & r^3 & -r^3 & -r^3 & -r^3 & -r^3 \\ r^2 & r^2 & -r^2 & -r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r^2 & r^2 & -r^2 & -r^2 \\ r & -r & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r & -r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r & -r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r & -r \end{bmatrix}$$

And split into the iterative form:

$$\mathbf{A} = \begin{bmatrix} r & r & & & & & & & \\ r & -r & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} r & r & & & & & & & \\ & r & r & & & & & & \\ r & -r & & & & & & & \\ & & r & -r & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} r & r & & & & & & & \\ & r & r & & & & & & \\ & & r & r & & & & & \\ & & & r & r & & & & \\ r & -r & & & & r & r & & \\ & & r & -r & & & & & \\ & & & r & -r & & & & \\ & & & & r & -r & & & \\ & & & & & r & -r & & \end{bmatrix}$$

Using $8 = 2^3$ dimensions shows the logarithmic nature of the process; for eight dimensions, we generalize the input signal three times, scaling values down by a factor of $\sqrt{2}$ to maintain unit norm for each more general projection than the one before. Strang and Nguyen use this pattern to show two key points: this iterative multiplication of sparse matrices *is itself* the Fast Wavelet Transform, the wavelet equivalent of the Fast Fourier Transform; and, iteratively applying these sums and differences against recursively generalized or “abbreviated” forms of our signal is naturally the same as using an iterative filter bank with a logarithmic tree structure.

With each piece of a filter bank we have an impulse response with its own coefficients. Strang and Nguyen explain that the analysis matrix here corresponds to the iteration of averaging and differencing filters with two coefficients, with the output of the averaging process passed on for more averages and differences. The components are related to one another through the high/low frequency splitting (averaging and differencing) and downsampling, but what other coefficients could we use, corresponding to what other filters and what other wavelets? They move on to explain how more intricate filters can be matched with more advanced wavelet functions; for now we stay with our simple case, but touch on its limitations below.

2.5 Duality of Haar and Sinc Wavelets

Switching our domain from pure time to pure frequency shows the limitation in applying Haar’s system as a signal processing technique. To do this we take the Fourier transform of the most basic Haar wavelet, and

consider how the function changes in the frequency domain as we vary scale. To start with, our mother Haar wavelet can be written as

$$\psi(t) = [u(t) - 2u(t - \frac{1}{2}) + u(t + 1)]$$

Using the standard trick for time derivatives and Fourier transforms ($\frac{df}{dt}$ transforms to $i\omega F(\omega)$ [Mal99, Eq. 2.21]), we can switch to a derivative form of the wavelet, take the transform, and switch back. First, for the derivative and its transform:

$$\frac{d\psi}{dt} = \delta(t) - 2\delta(t - \frac{1}{2}) + \delta(t - 1)$$

$$\begin{aligned} \mathcal{F}\left(\frac{d\psi}{dt}\right) &= \int_{-\infty}^{\infty} \frac{d\psi}{dt} e^{-i\omega t} dt \\ &= \int_{-\infty}^{\infty} [\delta(t) - 2\delta(t - \frac{1}{2}) + \delta(t - 1)] e^{-i\omega t} dt \\ &= e^{-i\omega 0} - 2e^{-i\omega \frac{1}{2}} + e^{-i\omega 1} \\ &= 1 - 2e^{-i\omega \frac{1}{2}} + e^{-i\omega} \end{aligned}$$

Then, for the transform of ψ :

$$\begin{aligned} \Psi = \mathcal{F}(\psi) &= \frac{1}{i\omega} \mathcal{F}\left(\frac{d\psi}{dt}\right) \\ &= \frac{1}{i\omega} - \frac{2}{i\omega} e^{-i\omega \frac{1}{2}} + \frac{1}{i\omega} e^{-i\omega} \end{aligned}$$

Finally, if we take into account time scaling [Mal99, Eq. 2.20], we have a general form for a Haar wavelet in the frequency domain:

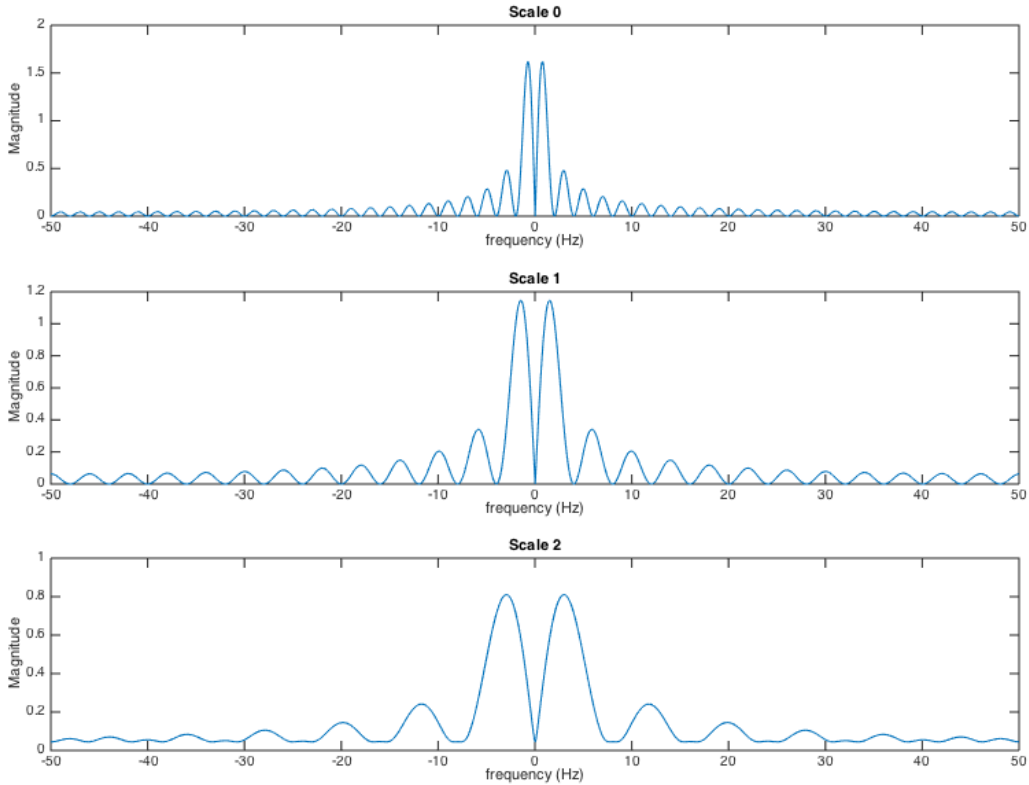
$$\begin{aligned} \Psi(2^j t) = \mathcal{F}(\psi(2^j t)) &= 2^{-j} \Psi(2^j \omega) = 2^{-j} \left[\frac{1}{i2^j \omega} - \frac{2}{i2^j \omega} e^{-i2^j \omega \frac{1}{2}} + \frac{1}{i2^j \omega} e^{-i2^j \omega} \right] \\ &= \frac{i2^{-2j}}{\omega} \left[2e^{-i2^j \omega \frac{1}{2}} - e^{-i2^j \omega} - 1 \right] \end{aligned}$$

So, we have a function that drops off in magnitude with increasing frequency as $|\frac{1}{\omega}|$ [Dau92, p. 10], but higher scale stretches the frequency axis, effectively slowing the decay. Figure 5 shows plots of the magnitude against frequency for increasing scale. The large pair of peaks marks the Haar wavelet's localization in frequency, but it starts poor and gets worse with the stretching of the axis. This is the downside of the trivially-simple compact support we have in the time domain for the Haar wavelets: the sharp angles of these functions have considerable spread in frequency. Wavelets provide more flexibility in working around the uncertainty principle, but we can't escape it; compact support in one domain precludes it in the other.

But, what if we start off with a simple, compactly-supported function in the frequency domain instead? Strang and Nguyen pass through the topic only briefly, on the basis that actually implementing a filter with such a frequency response would necessarily have infinite impulse response [SN97, p. 51], i.e., could *not* be compactly supported in time. But as a logical partner to Haar in the frequency domain, it is worth considering. Figure 6 shows the ideal case of a frequency-domain function that, used as a filter on an input signal, would partition the signal into octaves with each increase in scale in the time domain. Unlike in the Haar case, this corresponds to a perfect alignment between *scale* and *frequency*.

But now that we have this "perfect alignment" in frequency, we no longer have it in time: as referred to earlier, the time-domain function for such a wavelet has infinite support. Even though most of a projection's value is localized in time, it still draws on the value of an input signal at *all times*. We have traded time-domain simplicity for frequency-domain simplicity, gaining one by losing the other. This emphasizes the limitations of taking a simplistic view of the "time-frequency tiling" idea. (Mallat makes this idea concrete with a proof by contradiction, attempting to define an expression for a function with compact support in

Figure 5: Magnitude of Fourier transform of Haar wavelet



both domains and obtaining a conflicting result for the time domain function [Mal99, Theorem 2.6].) One last observation here: Considering when we would have a balance between time and frequency yields the *Gabor wavelet*, with very similar expressions in both and frequency [Mal99, Eq. 4.62]:

$$g(t) = \frac{1}{(\sigma^2\pi)^{1/4}} \exp\left(\frac{-t^2}{2\sigma^2}\right)$$

$$G(\omega) = (4\sigma^2\pi)^{1/4} \exp\left(\frac{-\sigma^2\omega^2}{2}\right)$$

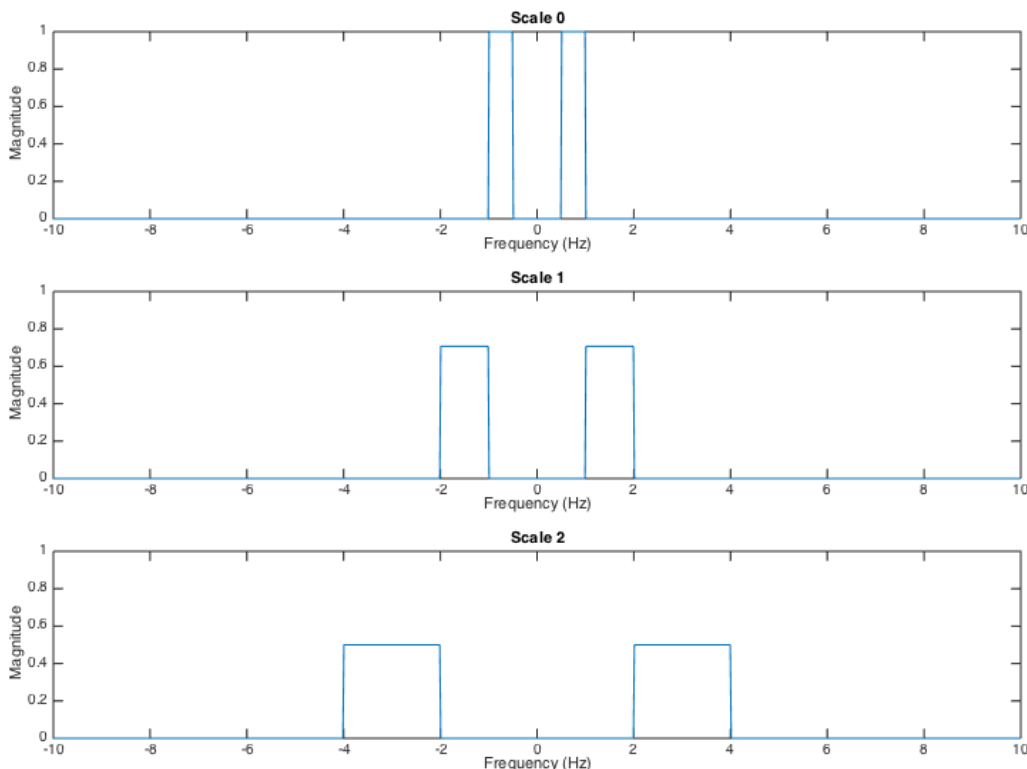
2.6 A Note about Frames

Mallat introduces a frame as a family (more general than a “basis”) of Hilbert space vectors that is *not* orthonormal, but which still maintain restrictions on the partitioning of a signal’s energy when projected onto the vectors [Mal99, p. 126]:

$$A\|f\|^2 \leq \sum_{n \in \Gamma} |\langle f, \phi_n \rangle|^2 \leq B\|f\|^2$$

In this case the norm of the vector in the new space isn’t guaranteed, and we can only say that it falls within a *range* between A and B. If A and B are the same, the energy is always scaled by the same factor, and the frame is “tight.”

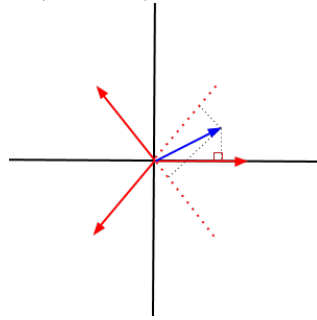
Figure 6: Magnitude of Fourier transform of Sinc wavelet



A two-dimensional analog to this would be a set of more than two vectors that lie on different lines (i.e, are not orthogonal but are not linearly dependent) as shown in Figure 7. Projecting a vector in our 2D space onto these *does* give a unique representation, but we can't simply add the resulting projections together to get our original vector back. One issue is that the resulting vector is scaled by some factor that falls between A and B. (Strang and Nguyen provide a clear explanation for how we can get back to our original vector: we need a second family of vectors, $\tilde{\phi}_n$, for which $\langle \tilde{\phi}_{n,i}, \phi_{n,j} \rangle = \delta(i - j)$.) The scaling factor given by the lower bound A is in a sense a measure of the redundancy of the frame in the space it spans [Dau92, p. 57]. In two dimensional space, we can imagine projecting a vector onto three nonorthogonal unit vectors pointing in different directions, and observe the "redundancy factor" by looking at both the original norm of the vector and the sum of the squares of the projections. In Daubechies' two-dimensional example, we have three evenly-spaced vectors, 120° apart, which gives a tight frame with a redundancy of $3/2$. Figure 8 shows a few other examples in 2D on the left panels. The right panels show the effect of projections onto these frames using a "test vector" of unit length with angle ranging from 0 to 2π . The "bad" choice of uneven spacing between vectors in the frame (in the bottom pair of plots) shows a varying effect on this test vector that depends on angle, corresponding to the A and B in the Hilbert space representations.

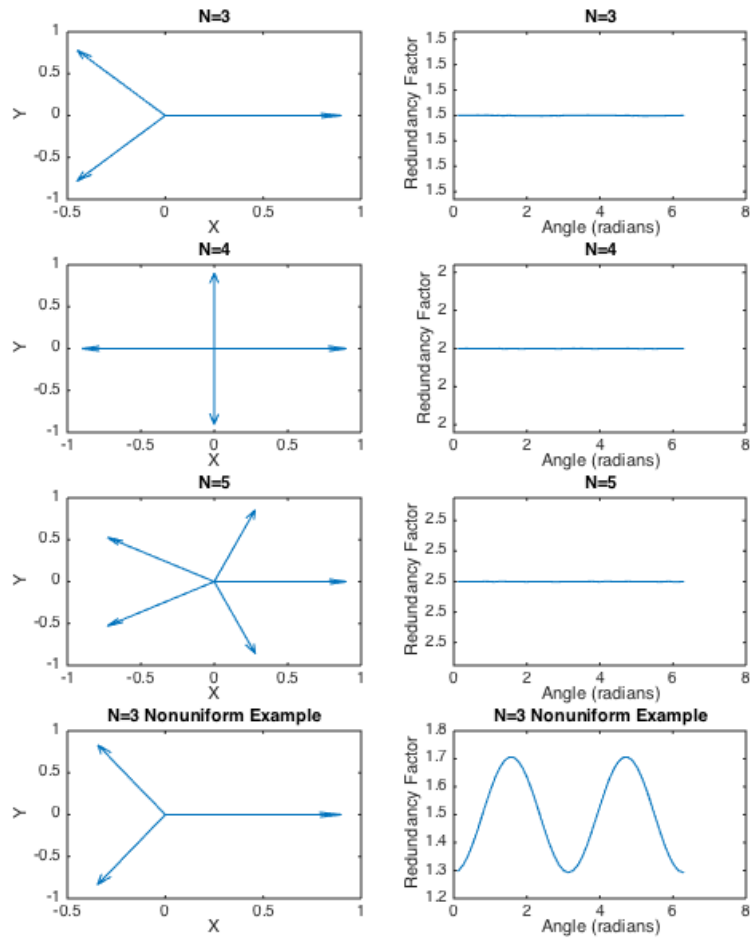
But what is the value in introducing this extra complication? With $A, B > 1$, we have redundancy in our representation, since individual components could be substituted with scaled sums of one another. Intuitively this seems undesirable, since we have a clean, two-way transform with an orthonormal basis. Mallat points out that redundancy can be beneficial if the *projections onto the frame components themselves* are noisy in some way. A thought of where we might see this issue arise is when implementing the transform

Figure 7: Projection of an example vector (blue line) onto a frame (three red lines) in two dimensional space



numerically. We must be concerned about the stability of the methods, taking into account the effect of small perturbations [Dau92, p. 55]. Without frames, performing and reversing the transform numerically could leave us far from where we started, simply from round-off error. *Frames allow us to make the method numerically stable and robust* [Dau92, p. 56, p. 98].

Figure 8: Effect of different frames on the norm of vectors rotated around the unit circle. The first three frames have evenly-spaced vectors in angle, while the last has an uneven distribution of three vectors around the unit circle $(0, \frac{5\pi}{8}, \frac{11\pi}{8})$.



3 Applications

3.1 A Basic Attempt at ECG Denoising

Figure 9: Example ECG waveform on top, with 60 Hz noise (in the form of a pure sine wave) added to the signal on the bottom.

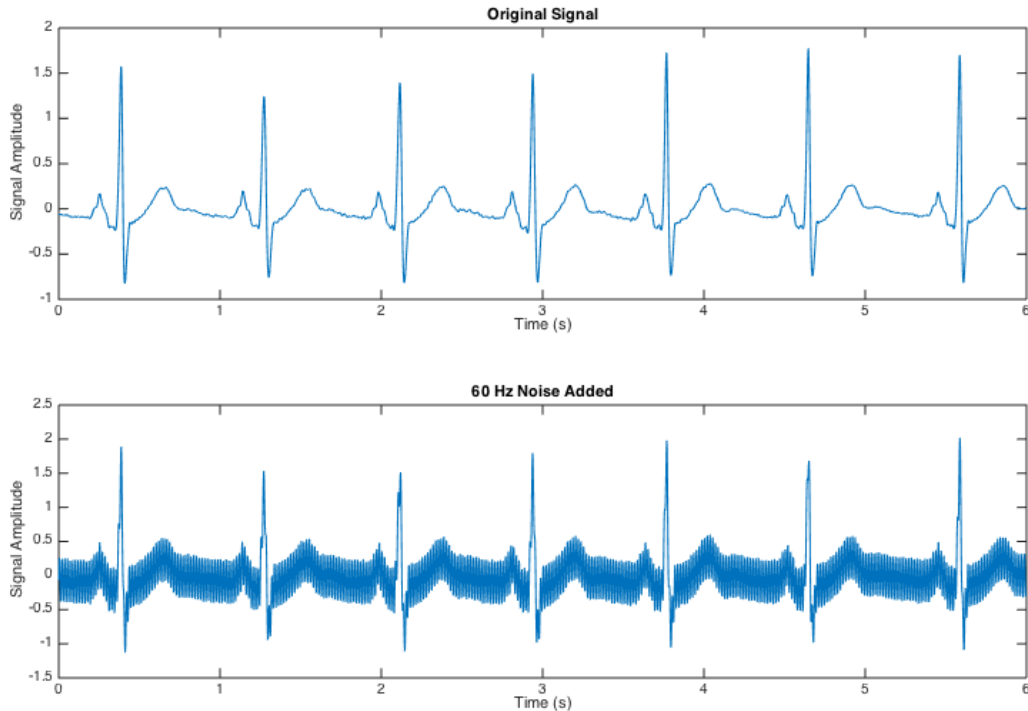
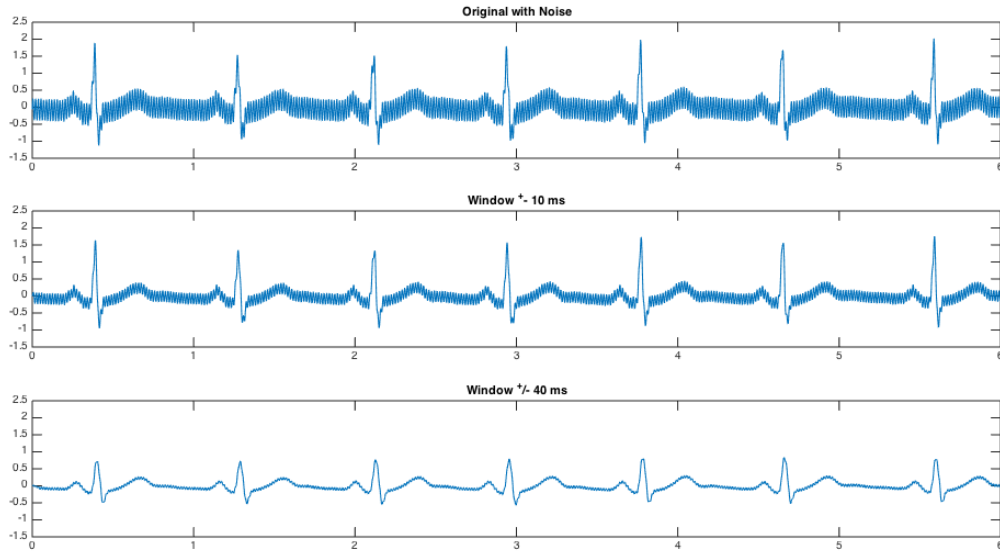


Figure 9 shows another view of the same electrocardiogram waveform shown on page 6. The upper plot shows six seconds of the original recording. In the bottom plot, a 60 Hz sine wave has been added to the signal, as a crude simulation of a recording with power line noise overlaid – mimicking a commonplace filtering task in acquiring these sorts of signals.

In the simplest case, we could use a traditional lowpass filter (implemented in hardware with simple passive components, or software as a moving average) and smooth out the added noise. The cutoff frequency would be midway between the bulk of the signal content and noise. A lower cutoff gives a smoother signal, but taken too far, the higher-frequency components of the signal are filtered too. In practice the large sharp peak (the QRS complex) suffers most obviously, since its sudden change in time correspond to energy at higher frequencies. Figure 10 shows this problem visually: as we increase window size from one plot to the next for a greater effect in a moving average filter, the main peak in each cycle becomes increasingly filtered along with the noise.

More complex linear filtering techniques can improve on this trade-off with more abrupt separation in the frequency domain, but the change in amplitude at location of the QRS peaks raises an interesting point: can we separate signal from noise by taking into account *amplitude of frequency components at specific times*, when both our noise and components of our signal could be in similar frequency ranges? Yes! Burrus, Gopinath, and Guo describe exactly this strategy under the heading of “Nonlinear Filtering or Denoising

Figure 10: A set of software-based moving averages on the noisy version of the signal.



with the DWT” [BGG98, Sec. 10.3]. A straightforward technique is *hard thresholding* of coefficients in the wavelet domain; as presented in [BGG98], for wavelet coefficients Y , this is:

$$\hat{X} = T_h(Y, t) = \begin{cases} Y, & |Y| \geq t \\ 0, & |Y| < t \end{cases}$$

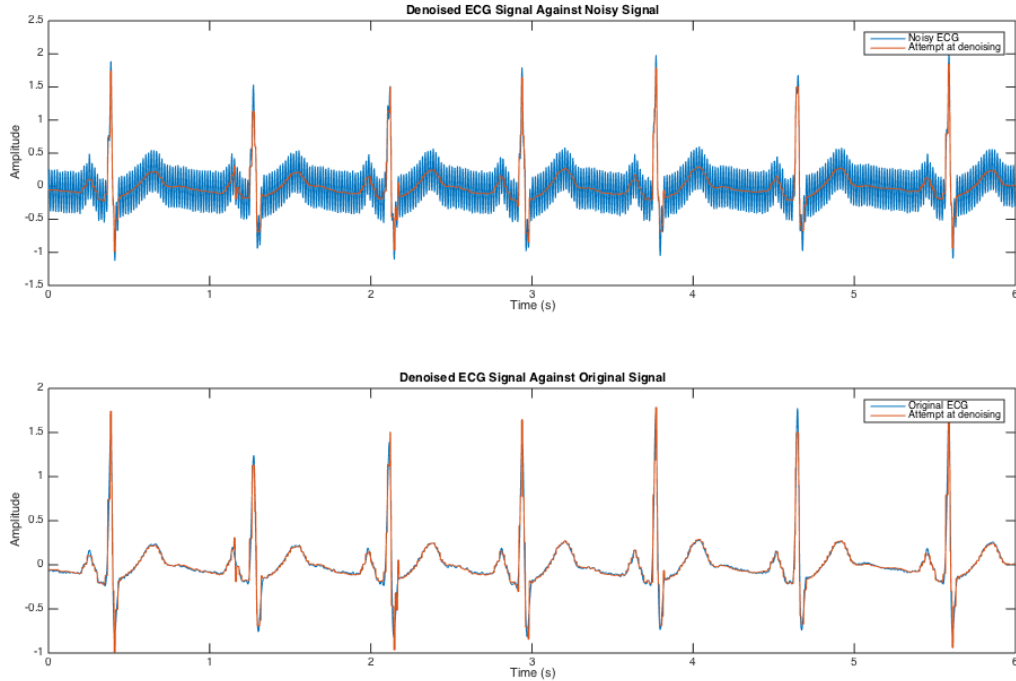
Where taking the inverse transform to produce $x(t)$ creates the denoised time domain signal. To implement a hard thresholding scheme for this simple trial, we took a signal sampled at 500 Hz and truncated it as far as possible while still giving an even power-of-two split for 2^{14} samples (about 30 seconds). A Haar wavelet decomposition using MATLAB’s `wavedec` function was then taken up to scale 14 for “full coverage” (as scale 14 yields pointwise differences on the original signal). A threshold was applied to the top three scales: coefficients above the threshold were left as-is and coefficients below were removed. The actual threshold value was taken fairly arbitrarily for this example to be 0.7, or about half of peak amplitude. The altered coefficients were fed back into MATLAB’s `waverec` function to synthesize the denoised signal back in the time domain.

Figure 11 shows the final result: in this admittedly simplistic example, we did reach our goal! The noise is removed, with minimal effect on the original signal. (An occasional jagged edge in one QRS peak or another reveals that this is *not* an LTI operation, in terms of how frequency components are treated; the thresholding implicitly makes high frequencies that occur at the times of the large peaks different from those between peaks.) The complexity of implementing the forward and reverse transforms comes with a clear benefit inside the new domain: a concise expression for the change we want to apply that matches well with intuition.

3.2 ECG Compression

Strang and Nguyen provide a brief application example in the form of *compressing* electrocardiogram recordings [SN97, p. 385], drawing from Nguyen’s research into the topic [DNT95]. To allow evaluation of the quality of the compressed signal, they use the percent RMS difference between original and compressed signals:

Figure 11: The final result of attempted denoising of ECG signal with hard threshold in wavelet domain.



$$PRD = \sqrt{\frac{\sum x_{or}(n) - x_{re}(n)^2}{\sum x_{or}(n)^2}} \times 100\%$$

(They point out that preservation of diagnostic details in the signal is the mark of successful compression, rather than low PRD per-se. But, the PRD does allow a point of objective comparison between different algorithms.)

Showing a comparison of compression techniques with varying effective bit rates, their compression technique shows an order of magnitude improvement in PRD compared to AZTEC, another commonly-used ECG compression system. This is easily confirmed in the example plots showing the same 2.5 second recording compressed with several methods; both non-wavelet methods suffer from distortion in the form of sudden jumps and blockiness in the signals. The main strength in the application of the wavelet method is the flexibility of the basis function length; varying length along with scale allows for a concise representation without the characteristic “blocking” of other methods [SN97, p. 368]. Referring back to an earlier example we see the same concept where other methods produce blocking or ringing in compressed images due to discontinuities in the basis functions.

As with their example using images, the core of the compression technique for ECG signals is in allocating bits in accordance with the relative partitioning of signal energy into the different components; in the image case, Strang and Nguyen argue that, “clearly, subimages with lower energy levels should have fewer bits.” Rather than simply zeroing out coefficients that are considered negligible, this shows a more nuanced approach, where the resolution of the coefficients themselves is varied depending on their significance. (This has a comparison with how a data acquisition system should provide dynamic range; a given number of bits should be matched to signal amplitude to maximize resolution.) Coefficients that are of the lowest significance can be allocated zero bits for their storage, while coefficients that are larger can be stored

with increasing quality. The actual algorithm for assigning bits might consider any number of criteria (the authors mention distortion, energy, and entropy), but the fundamental idea is the same. The final step in the overall compression method is to group zero coefficients together so that an entropy coding method can work effectively. But, in terms of the wavelet-based approach, the central idea is in the quantization of the coefficients.

4 Concluding Remarks and Further Steps

Reaching back to Haar’s work around 1910, and culminating in a flurry of activity in the 1980’s and 1990’s, wavelet techniques have matured into a recognized area of applied mathematics and signal processing. In more recent years we can see this in the extensive support for wavelets in both commercial software, as with MATLAB’s Wavelet Toolbox [Mat], and open-source and community implementations, such as PyWavelets [Was12]. PyWavelets’ author has also created a web-based “dictionary” of common wavelet families [Was08].

The range covered here has admittedly been limited. We’ve focused on exploring the absolutely simplest structures that could be called wavelets, trying to develop intuition into the main ideas and the most basic applications. In practical usage so much more has been developing since Haar’s work a century ago: Ingrid Daubechies’ work on more advanced compactly-supported wavelets, the recognition of the connection to filter bank design by Stéphane Mallat, and additional wavelet structures like those proposed by Yves Meyer, to name a few. As explored here very briefly, wavelet-based signal processing has particular strengths in biomedical signal processing [BGG98, p. 218]; an obvious next step is to investigate the usage of the more advanced and modern techniques in this particular area.

References

- [BGG98] Sidney Burrus, Ramesh Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelet Transformations: A Primer*. Prentice Hall, 1998.
- [Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industry and Applied Mathematics, 1992.
- [DNT95] Adrianus Djohan, Truong Nguyen, and Willis Tompkins. Ecg compression using discrete symmetric wavelet transform. In *IEEE Engineering in Medicine and Biology, Theme 1: Cardiovascular System*, pages 167–168, 1995.
- [GC99] Jaideva Goswami and Andrew Chan. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. John Wiley and Sons, 1999.
- [Haa09] Alfréd Haar. *Zur Theorie der orthogonalen Funktionensysteme [On the Theory of Orthogonal Function Systems]*. PhD thesis, University of Göttingen, 1909. (Translation by Georg Zimmermann).
- [Mal99] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [Mat] The Mathworks. Wavelet toolbox. <http://www.mathworks.com/products/wavelet/>. Accessed: 2014-12-10.
- [Phy] PhysioNet. Physiobank atm. <http://www.physionet.org/cgi-bin/atm/ATM>. Accessed: 2014-11-29.
- [SN97] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, 1997.
- [Soc] American Mathematical Society. The mathematics genealogy project – alfred haar. <http://www.genealogy.ams.org/id.php?id=7375>. Accessed: 2014-12-14.
- [Str92] Gilbert Strang. Wavelets. *American Scientist*, 82:250–255, 1992.
- [Was08] Filip Wasilewski. Wavelet properties browser. <http://wavelets.pybytes.com/>, 2008. Accessed: 2014-12-10.
- [Was12] Filip Wasilewski. Pywavelets - discrete wavelet transform in python. <http://wavelets.pybytes.com/>, 2012. Accessed: 2014-12-10.