Experiments with Adapted Wavelet De-Noising for Medical Signals and Images*

Ronald R. Coifman, Ph.D.† Mladen Victor Wickerhauser, Ph.D.‡

November 16, 1995

Abstract

We first describe some new libraries of waveforms, including wavelets, wavelet packets, and local sines and cosines, which are well-adapted to representing biological and biomedical signals. By expanding a signal in a library of waveforms which are well-localized in both time and frequency, we can separate coherent structures from incoherent noise. We experiment with one implementation of adapted wavelet de-noising, and compute the signal-to-noise ratio improvement obtained for certain simple signals.

1 Time and frequency analysis

In [12] we describe tools for adapting methods of analysis to various tasks occurring in harmonic and numerical analysis and signal processing. The main point is that by choosing a new coordinate system, in which space and frequency are simultaneously localized, one can more easily separate the signal into coherent structures and incoherent noise. This decomposition is intimately related to efficiency in representation (i.e., compression) and to pattern extraction, or structural understanding.

*Research supported by NSF, AFOSR, and the Southwestern Bell Telephone Company

†Department of Mathematics, Yale University

‡Department of Mathematics, Washington University in St.Louis

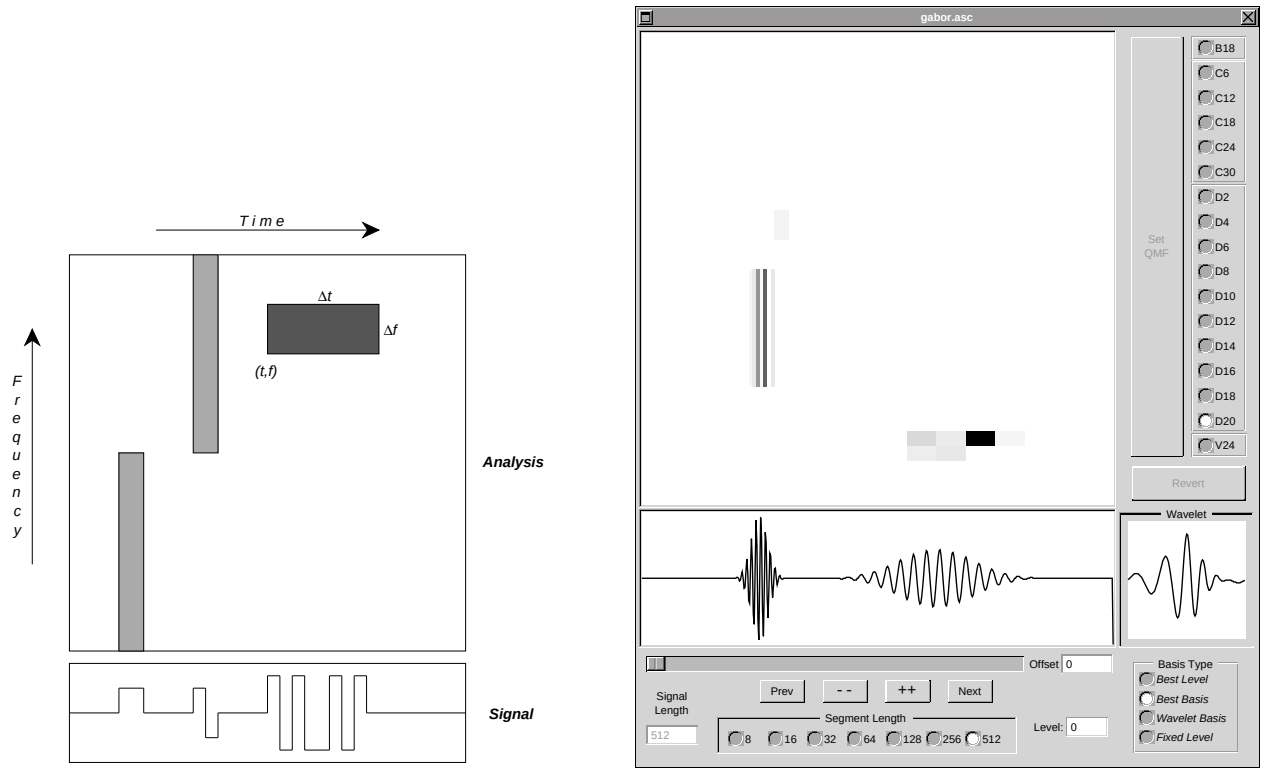

Figure 1: Idealized and actual analysis in the time-frequency plane.

The separation is done by choosing an appropriate decomposition of the signal into components in the *time-frequency plane*—an abstract 2-dimensional signal representation in which time and frequency are the horizontal and vertical axes, respectively. A waveform is represented by a rectangle in this plane, as seen in the left half of Figure 1. Let us call such a rectangle an *information cell*. The position in time and the main frequency can be read from the coordinates of the center of the rectangle. The uncertainty in time and the uncertainty in position are given by the width and height of the rectangle, respectively. Heisenberg's inequality, or the *uncertainty principle*, implies that the area of such a rectangle can never be less than 1. The amplitude of a waveform can be encoded by darkening the rectangle in proportion to its waveform's energy.

We will only use time-frequency atoms in our analysis; these are waveforms which are so well localized in both time and frequency that the areas of their information cells must be close to 1. They may be represented by information cells of equal area in the time-frequency plane. A basis of such atoms corresponds to a cover of the time-frequency plane by rectangles, and we will depict an orthonormal basis by using disjoint rectangles of exactly equal (unit) area. It is well known that only the *Gaussian function* $g(t) = e^{-t^2}$ and its variations have the minimal information cell area. The other time-frequency atoms are not too far off, though, and we will avoid the many restrictions of the Gaussian by relaxing the minimal area condition. The only price we will have to pay is that a single atom might in practice require a few of the approximate atoms. The right half of Figure 1 shows the output of such an approximate analysis on two Gaussian time-frequency plane atoms.

An orthogonal adapted wavelet analysis [12] produces a numerical recipe for the decomposition; the covering partitions are chosen to achieve maximum efficiency with respect to an *information cost function*. Not only does this approach shed light on classical analysis methods, it also suggests new methods of operating on the signals we have analyzed. These include nonstandard forms for linear operators [1, 11], which better display the interactions among different signal parts, and discrete time-frequency plane approximations for the evaluation of complicated nonlinear operators [13] and large scale computation.

2 Example libraries of waveforms

The appropriate analysis method is chosen from a catalogue of tools for application to a particular problem. In an orthogonal adapted waveform analysis, the user begins with a collection of standard libraries of waveforms—called *wavelets*, *wavelet packets*, and *windowed trigonometric waveforms*—which can be combined to fit specific classes of signals. All these functions are time-frequency atoms. Examples of such waveforms are displayed in Figure 2.

These libraries are used because they come equipped with fast numerical algorithms.

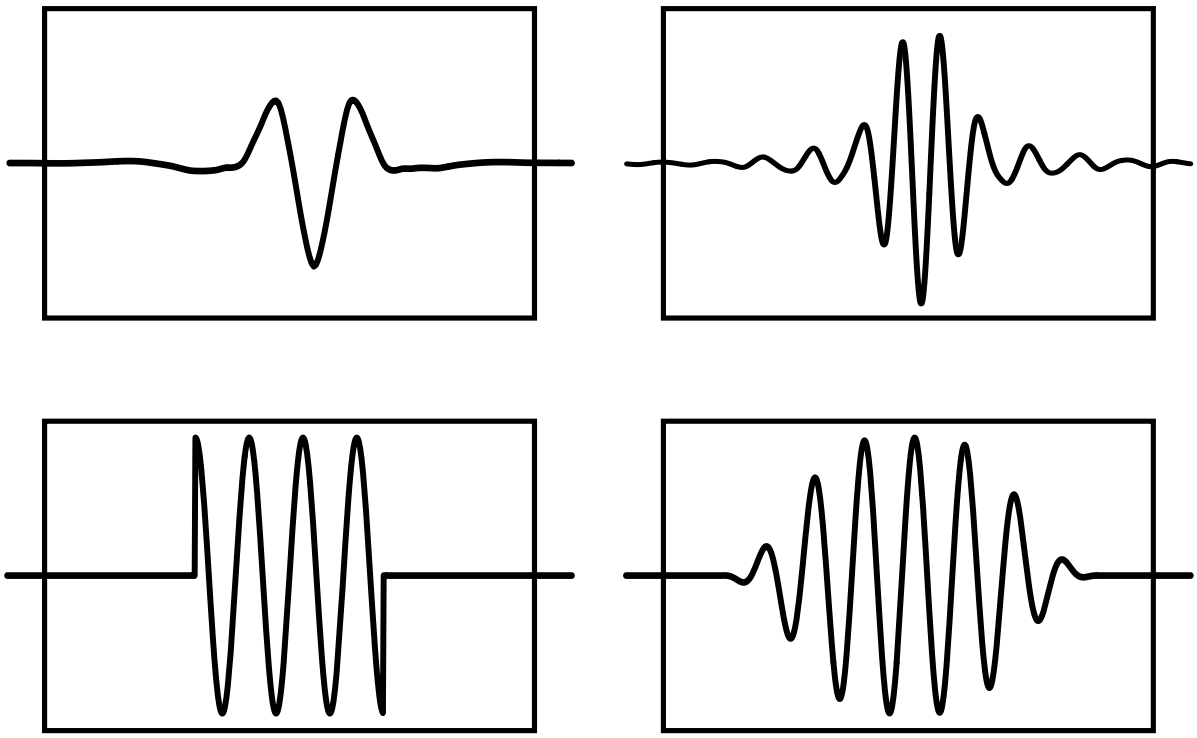

Figure 2: Example waveforms: wavelet, wavelet packet, block cosine and smoothly windowed cosine functions.

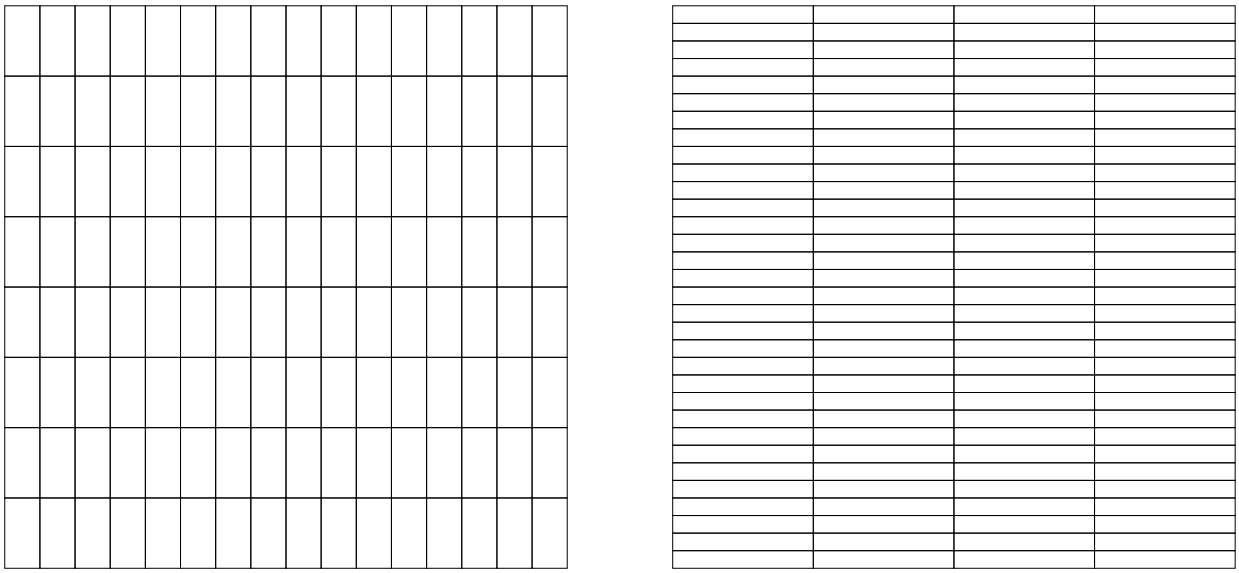

Figure 3: Narrow and wide-windowed Fourier tilings of the time-frequency plane.

They allow us to perform data compression, de-noising, and diagnostic feature extraction in real time. The process of analysis is usually done by comparing acquired segments of data with stored waveforms. The numerical comparison algorithm itself is fast and perfectly conditioned, always being a factored sparse orthogonal transformation. Then the most efficient orthonormal basis for compression of the signal is selected and used to extract and manipulate relevant features.

Consider first the short-time or windowed Fourier transform. Here the basis functions are exponentials, or maybe sines and cosines, which are enveloped so they oscillate only for a short time before going back to zero. They yield a tiling of the time-frequency plane by congruent rectangles, whose dimensions depend upon the window size. Two choices of window size are shown in Figure 3. It is still necessary to choose a window size appropriate to the analysis. Our measure of quality will be the amount of white space, or negligible waveforms, in the time frequency analysis of a signal. Lots of white space means that most of the components have negligible energy, so that that the signal energy is concentrated into

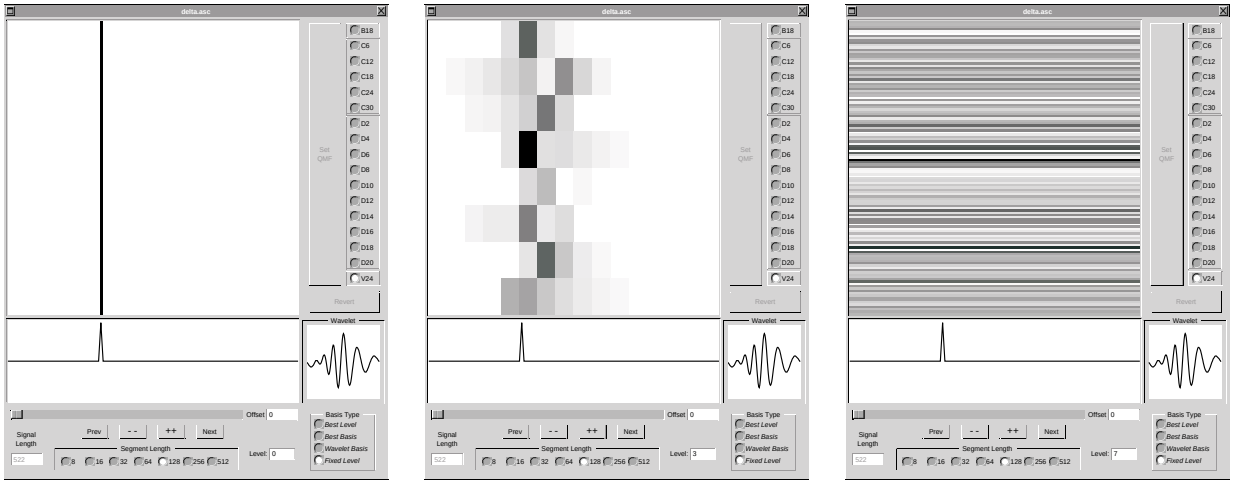

Figure 4: Time-frequency analysis of a sharp impulse at increasing window sizes.

just a few waveforms.

Very short windows are most efficient for sharp impulses, while long windows correspond to information cells which spread energy all over the time-frequency plane, as seen in Figure 4. Conversely, long windows are more efficient than short ones for nearly continuous tones, as depicted in Figure 5. Hence it can be useful to examine the signal in many window sizes at once and then to choose the *best basis* in the sense of efficient representation.

Remark. The Fourier transform is a rotation by 90 degrees in the time-frequency plane. This is most evident in Figure 6, which shows the idealized information cells corresponding to grid-point samples (i.e., *Dirac mass* "waveforms") or sampled pure sine waves as time-frequency atoms. It is also possible to apply the *Hermite semigroup transform* or *angular Fourier transform* to obtain information cells which make arbitrary angles with the time and frequency axes.

Wavelet analysis [6] corresponds to windowing frequency space in "octave" windows. Since the information cells have equal area, they cover the time-frequency plane in the

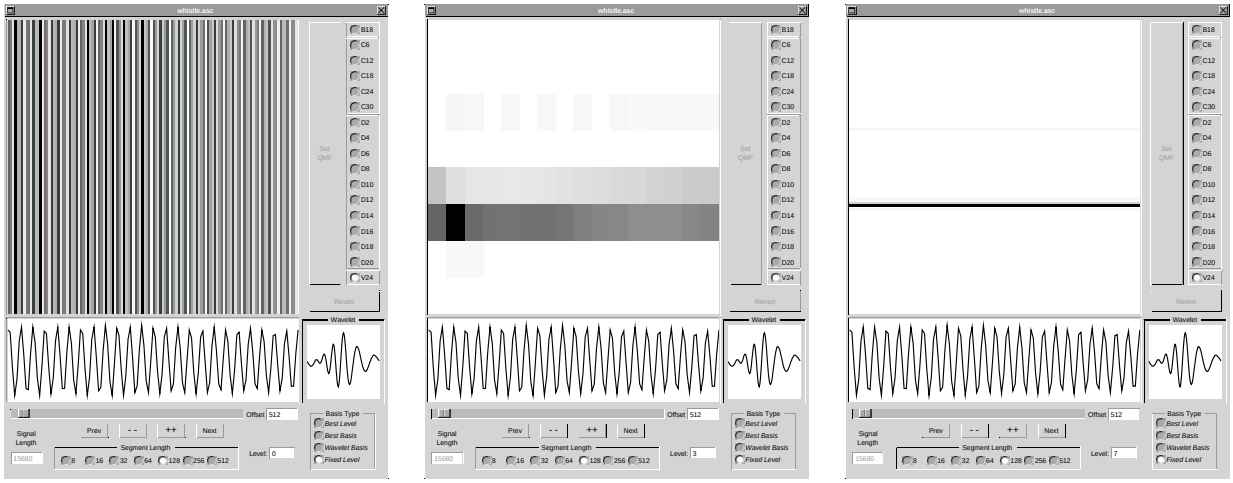

Figure 5: Time-frequency analysis of a nearly pure tone at increasing window sizes.

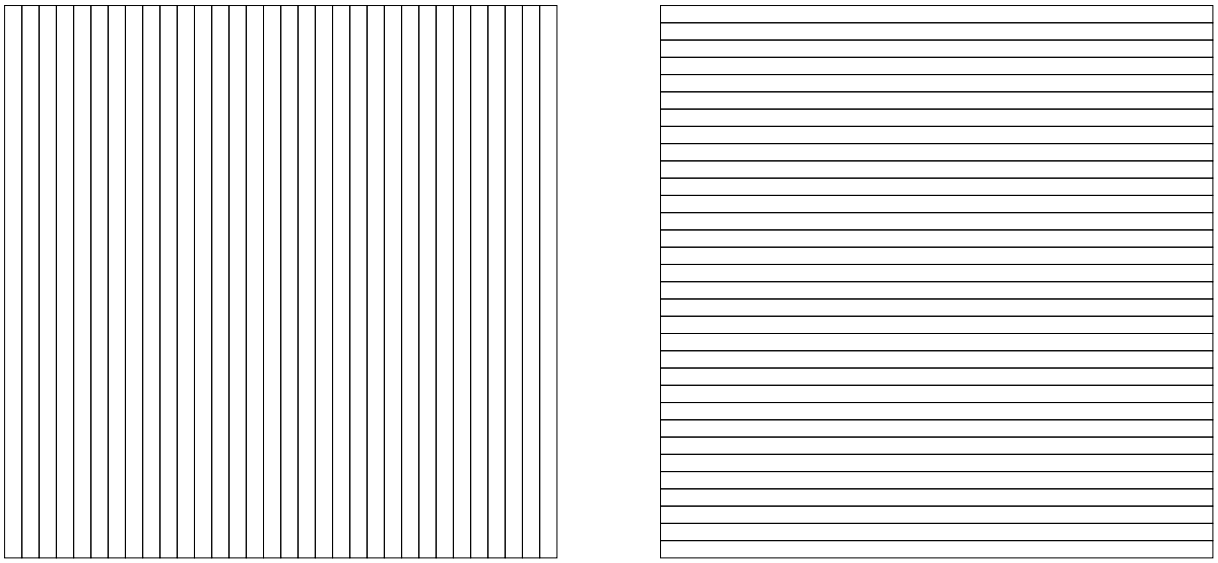

Figure 6: Dirac and Fourier decompositions of the time-frequency plane.

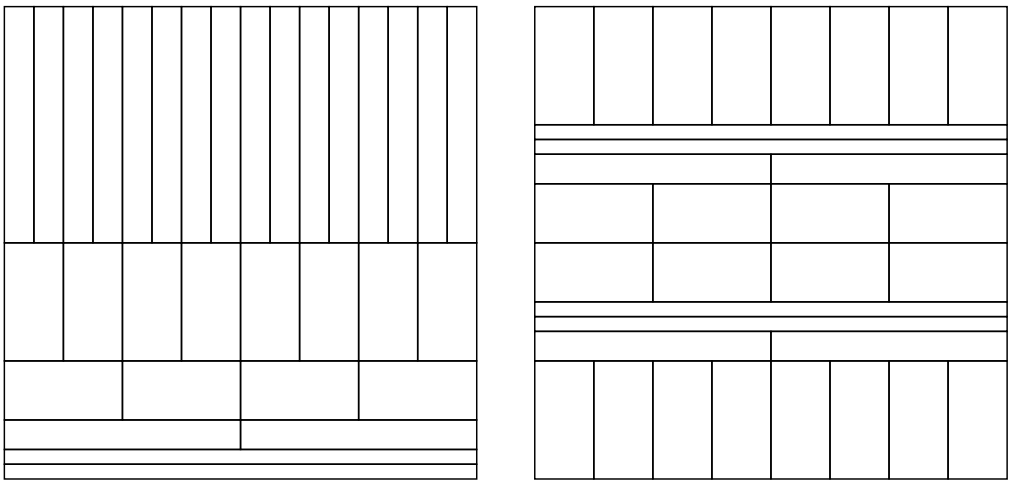

Figure 7: Dyadic wavelet and wavelet packet tiling of the time-frequency plane.

manner depicted in the left half of Figure 7. A natural extension therefore is provided by allowing all dyadic windows in frequency space and adapted window choice. This sort of analysis is equivalent to wavelet packet analysis; one example is shown in the right half of Figure 7.

In the wavelet library, there are both *wavelets*, traditionally denoted $\psi = \psi(t)$, and *scaling functions*, traditionally denoted $\phi = \phi(t)$. The pair of functions satisfy the *two-scale equations*, which in one normalization take the following form:

$$\phi(t) = \sqrt{2} \sum_{k \in \mathbf{Z}} h(k)\phi(2t-k); \tag{1}$$

$$\psi(t) = \sqrt{2} \sum_{k \in \mathbf{Z}} g(k)\phi(2t-k); \qquad g(k) = (-1)^k h(1-k). \tag{2}$$

Here $h = \{h(k) : k \in \mathbf{Z}\}$ is a sequence of coefficients which defines a related sequence of coefficients $g = \{g(k) : k \in \mathbf{Z}\}$. All properties of the wavelet library are determined by h.

The expansion of a signal $u = u(t)$ in the wavelet library starts with the sequence of

inner products of the signal with integer translates of the scaling functions:

$$x(k) \stackrel{\text{def}}{=} \int_{\mathbf{R}} u(t)\phi(t-k)\,dt, \qquad k \in \mathbf{Z}. \tag{3}$$

This amounts to a projection of the original signal, which is an almost arbitrary function of a continuous variable $t \in \mathbf{R}$, onto the much more restricted subspace of functions superposed from integer translates of the nice bump ϕ. Whether this projected image faithfully retains the important properties of the original depends on h and on the other assumptions placed on u. From the new sequence $x = \{x(k) : k \in \mathbf{Z}\}$, the inner products of u with dilated and translated wavelets may be computed with the following pair of *filtering* operations:

$$Hx(n) \stackrel{\text{def}}{=} \sum_{k\in\mathbf{Z}} h(2n-k)x(k); \quad Gx(n) \stackrel{\text{def}}{=} \sum_{k\in\mathbf{Z}} g(2n-k)x(k); \qquad n \in \mathbf{Z}. \tag{4}$$

This works because wavelets and scaling functions are related by the two-scale equations. For example,

$$\int_{\mathbf{R}} u(t)2^{-j/2}\psi(2^{-j}t-k)\,dt = GH^{j-1}x(k), \tag{5}$$

for any integer $j \geq 0$ and any integer k. All but the last of the inner products are obtained from x by applying a certain number of H filters followed by a single G filter. The procedure is depicted in Figure 8, where the wavelet transform consists of the numbers to be found in the shaded boxes. It is called the *discrete wavelet transform* since it can only recover the discrete approximation x of the original signal u.

The *wavelet packet* library is constructed by recursion of the wavelet algorithm, using more G filters. This library, introduced in [3] and described in detail in [12], contains the wavelet basis, Walsh functions, and smooth versions of Walsh functions called wavelet packets. The basic functions may be denoted by $w_n = w_n(t)$, where $n \geq 0$ is a nominal frequency index. They satisfy a generalization of the two-scale equations:

$$w_{2n}(t) = \sqrt{2}\sum_{k\in\mathbf{Z}} h(k)w_n(2t-k); \tag{6}$$

$$w_{2n+1}(t) = \sqrt{2}\sum_{k\in\mathbf{Z}} g(k)w_n(2t-k); \qquad n = 0, 1, 2, \ldots \tag{7}$$

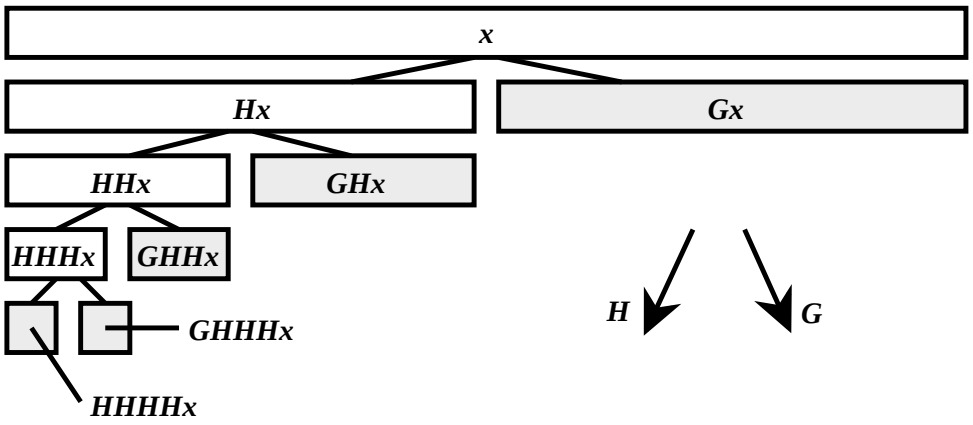

Figure 8: Discrete wavelet transform.

The initial function w_0 is just the scaling function ϕ defined by the first two-scale equation. Likewise, $w_1 = \psi$, and it is easily seen how w_n is developed for all nonnegative n. As with wavelets, the properties of all the functions $\{w_n, n \geq 0\}$ are determined by the sequence h.

The expansion of a signal $u = u(t)$ in the wavelet packet library begins, like for wavelets, with the sequence x of inner products with scaling functions defined by Equation 3. Recursive application of the H and G operators defined in Equation 4 then produces inner products with the other wavelet packets. For example,

$$\int_{\mathbf{R}} u(t) 2^{-3/2} w_5(2^{-3}t - k)\, dt = GHGx(k), \tag{8}$$

for $k \in \mathbf{Z}$. Here the nominal frequency is 5 and the scale index is 3. The complete procedure for all wavelet packets is depicted in Figure 9. The inner products in Equation 8 will be found in the shaded box labeled $GHGx$, which is number 5 from the left in level 3 from the top since the indexing begins at 0.

The numbers in the shaded boxes in Figure 9 constitute one particular *basis subset*, that is, a minimal set from which the sequence x can be exactly recovered. There are other basis subsets available, for example the one shaded in Figure 10. That is recognizable as

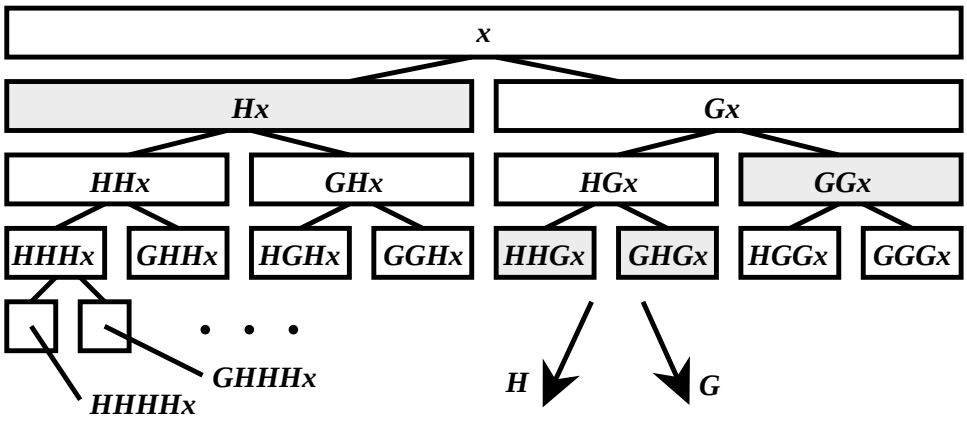

Figure 9: Discrete wavelet packet expansion and an example basis.

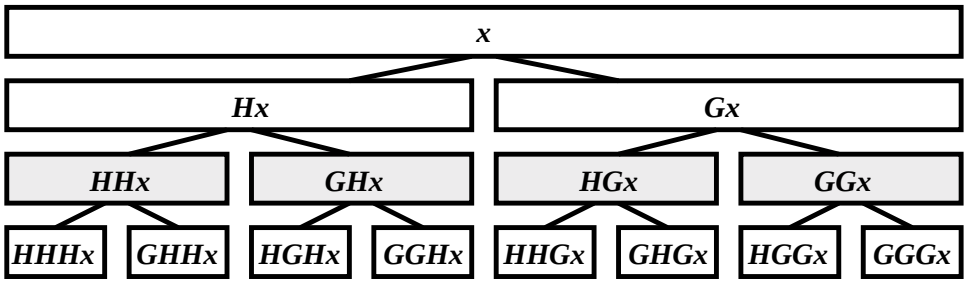

Figure 10: Complete level basis of wavelet packets.

a *complete subband decomposition* of the original signal. To get many more examples, the *graph basis theorem* ([12], p.244) can be used. It shows that the leaves of any binary subtree of the wavelet packet tree form a basis subset.

Wavelet packet analysis algorithms permit us to perform an adapted Fourier windowing directly in the time domain by successive filtering of a function into different frequency bands. The window size selection algorithm, in this context, gives an adapted subband coding algorithm. It should be mentioned that all of these algorithms have higher dimensional generalizations; in particular, they can be used to analyze still images and movies in the

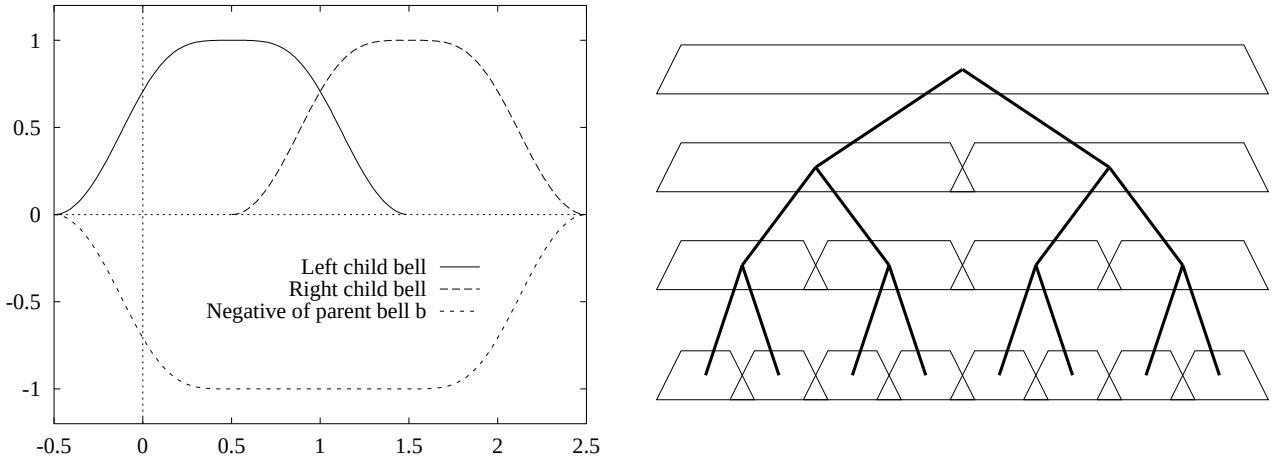

Figure 11: Big window from two small windows; the procedure iterated

same way that we are analyzing acoustic signals.

To illustrate a complete analysis in a library, we start with a description of an algorithm to compute the windowed Fourier sine expansion of a function on an interval from the Fourier expansion of its restrictions to the left half and the right half. This procedure is depicted in the left half of Figure 11. The main idea is that all the signals under the big window or "bell," which is shown taking negative values, are combinations of signals under the two smaller windows. We see that in order to compute the Fourier expansion on the large interval, we can start with adjacent pairs of small intervals, combine coefficients to obtain the expansion on their union, and continue until we reach the largest interval at the top level. This scheme is depicted in the right half of Figure 11. Along the way we have obtained all dyadic windowed Fourier transforms as intermediate computations. We notice that every disjoint collection of intervals and their orthogonal bases provides us with an orthogonal basis for the union.

3 Choosing the "best basis"

A natural question that arises in connection with the windowed Fourier transform is how to place the windows. The window selection has a big effect on the number of large coefficients in the expansion. So let us now turn to the question of optimizing the windows to obtain an efficient representation of a function.

The *best basis* algorithm introduced in [4] fits a time-frequency plane cover to the signal so as to best concentrate the shading into the fewest information cells in the time-frequency plane. This method can use rectangular information cells of all aspect ratios. The *best level* algorithm, described in [12], fits a cover of congruent rectangles to the signal, so as to best concentrate the shading. The congruence restriction on the rectangles is sometimes useful to avoid undesirable artifacts in partial reconstructions.

We can proceed as follows in our adapted windowed Fourier transform example: we start with the adjacent small intervals and determine the expansion coefficients on each one separately. We then compute the expansion coefficients on the union. Now we can choose that expansion for which the number of coefficients needed to capture 99% of the energy is smallest. Or, we can choose that expansion whose "cost" is smallest: information cost, coding cost, error cost, ...

We compare the cost of the chosen expansions on two adjacent unions of pairs to the expansion on their union and again pick the best. We continue until we reach an optimal distribution of time windows.

This algorithm segments a signal into portions that are individually easy to describe. It can be combined with an appropriate recognition criterion to segment continuous speech into voiced and unvoiced segments, as done in [5] and depicted in Figure 12, prior to classification into phonemes.

The adapted wavelet packet decomposition uses the same choice algorithm, only it produces an adapted segmentation of the frequency axis rather than the time axis.

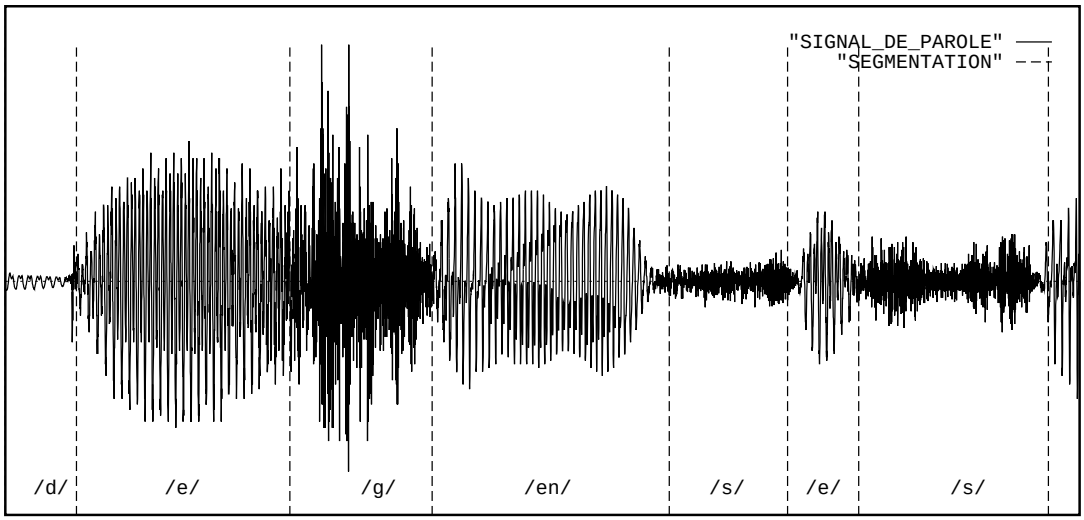

Figure 12: Automatic segmentation of a phrase into voiced and unvoiced portions.

4 Compression

The *compressibility* of a sampled signal is the ratio of the total area of the time-frequency plane (N, for a signal sampled at N points) divided by the total area of the dark information cells (each of area 1). We may automatically analyze signals by expanding them in the best basis, then drawing the corresponding time-frequency plane representation. The negligible components need not be drawn, as it is not relevant which particular basis is chosen for a subspace containing negligible energy.

As done for the Gaussians in Figure 1, signals can be automatically analyzed in their best wavelet packet bases by a computer program "WPLab" [10]. The user selects a transform by picking an analyzing quadrature filter from a list of 17 at the right. The "mother wavelet" [6] determined by that filter is displayed in the small square window at the lower right, to indicate roughly what the time-frequency atoms look like. The signal is plotted in the

rectangular window at bottom, and the time-frequency plane representation is drawn in the large main square window. Examples of canonical signals for this analysis are *chirps* (oscillatory signals with increasing modulation), *spikes* (sharp transients), *whistles* (almost periodic functions), and combinations of all three such as human speech; examples of best basis analyses for these may be seen in [12]

5 Adapted waveform "de-noising"

It is also possible to expand a signal in several libraries of waveforms and then to choose the library which best represents it. Or, if no library does particularly well, we can peel off layers of a signal by taking one or a few waveforms out at a time, then re-analyzing the remainders. These are examples of *meta-algorithms* which are used at a high level to choose an appropriate analysis for the given signal.

As an application combining these ideas we now describe an algorithm for *de-noising* or, more precisely, *coherent structure extraction*. This is a difficult and ill-defined problem, not least because what is "noise" is not always well-defined. We chose instead to view an N-sample signal as being noisy or incoherent relative to a basis of waveforms if it does not correlate well with the waveforms of the basis, i.e., if its entropy is of the same order of magnitude as

$$\log(N) - \epsilon \tag{9}$$

with small ϵ. From this notion, we are led to the following iterative algorithm based on the several previously-defined libraries of orthonormal bases. We start with a signal f of length N, find the best basis in each library and select from among them the "best" best basis, the one minimizing the cost of representing f. We put the coefficients of f with respect to this basis into decreasing order of amplitude.

The rate at which the coefficients decrease controls the *theoretical dimension* N_0, which is a number between 1 and N describing how many of the coefficients are significant. We

can define N_0 in several ways; the simplest is to count the coefficients with amplitudes above some threshold. Another is to exponentiate the entropy of the coefficient sequence, which matches the criterion in Equation 9.

Theoretical dimension is a kind of information cost. We will say that the signal is *incoherent* if its theoretical dimension is greater than a preset "bankruptcy" threshold $\beta > 0$. The threshold β is chosen to determine if unacceptably bad compression was obtained even with the best choice of waveforms. This condition terminates the iteration when further decompositions gain nothing.

If the signal is not incoherent, then we can pick a fraction $\delta > 0$ and decompose f into $c_1 + r_1$, where c_1 is reconstructed from the δN_0 big coefficients, while r_1 is the remainder reconstructed from the small ones. We proceed by using $r_1, r_2, \ldots$ as the signal and iterating the decomposition. The procedure is depicted in Figure 13. We can stop after a fixed number of decompositions, or else we can iterate until we are left with a remainder whose theoretical dimension exceeds β. We then superpose the coherent parts to get the coherent part of the signal. What remains qualifies as *noise* to us, because it cannot be well-represented by any sequence of our adapted waveforms. Thus the adapted waveform de-noising algorithm peels a particular signal into layers; we take as many of the top layers as we want, assured that the bottom layers are not cost-effective to represent.

The two parameters, β and δ, can be adjusted to match an *a priori* estimate of the signal-to-noise ratio, or can be adjusted by feedback to get the cleanest-looking signal if no noise model is known.

Adapted waveform de-noising is a fast approximate version of the *matching pursuit* procedure described by Mallat [8]. There the waveforms are Gaussians, and just one component is extracted at each iteration. That procedure always produces the best decomposition, at the cost of many more iterations plus more work for each iteration. Mallat's stopping criterion is to test the amplitude ratio of successive extracted amplitudes; this is a method of recognizing remainders which have the statistics of random noise.

As an example, we start with a mechanical rumble masked by the noise of aquatic life, recorded through an underwater microphone. The calculations were performed by the program "denoise" [7, 2], using $\delta = 0.5$ and manually limiting the number of iterations to 4. Figure 14 shows the original signal paired with its de-noised version. Note that very little smoothing of the signal has taken place. Figures 15, 16, 17 and 18 respectively show the coherent parts and the remainders of the first 4 iterations. Notice how the total energy in each successive coherent part decreases, while the remainders continue to have roughly the same energy as the original.

Figures 19, 20, 21 and 22 respectively show the successive reconstructions from the coherent parts paired with a plot of the best-basis coefficient amplitudes of the remainders, rearranged into decreasing order. A visual estimate of the theoretical dimension from these plots gives evidence after the fact that little is gained after 4 iterations.

6 Experiments with SNR improvement

6.1 Procedure

The software used to perform these experiments is a demonstration package called "denoise" which is maintained and distributed by FMA&H Corporation. It was modified by the authors from the version available by anonymous FTP [2]. It consists of scripts and executables for one and two-dimensional adapted wavelet denoising, together with some utility functions for signal formatting, adding noise, and computing signal to noise ratio (SNR) in dB.

We began with a 512-point sampled sine function of amplitude 64, biased to the range $[64, 196]$, with 7 cycles in the 512-point period. This is plotted in Figure 23. We then added pseudorandom numbers to each sample, approximating independent uniformly distributed random noise with the right variance to produce the files `sine+0db.asc`, `sine+2db.asc`, etc. The filename indicates the nominal signal-to-noise ratio in decibels. We then ran `denoise -i4 -m9 -t0.2 sine+2db.asc`, etc., to produce the superposition of the four coherent parts

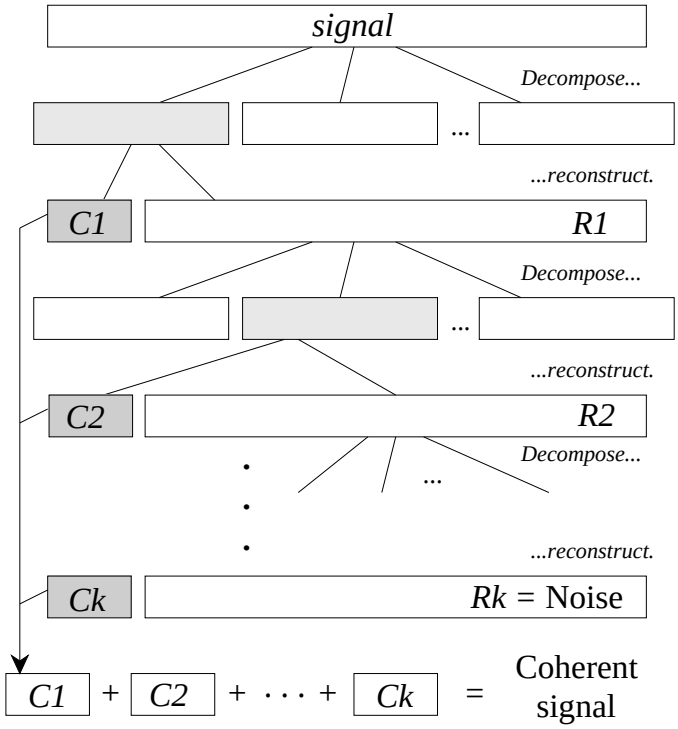

Figure 13: Schematic of adapted waveform de-noising.

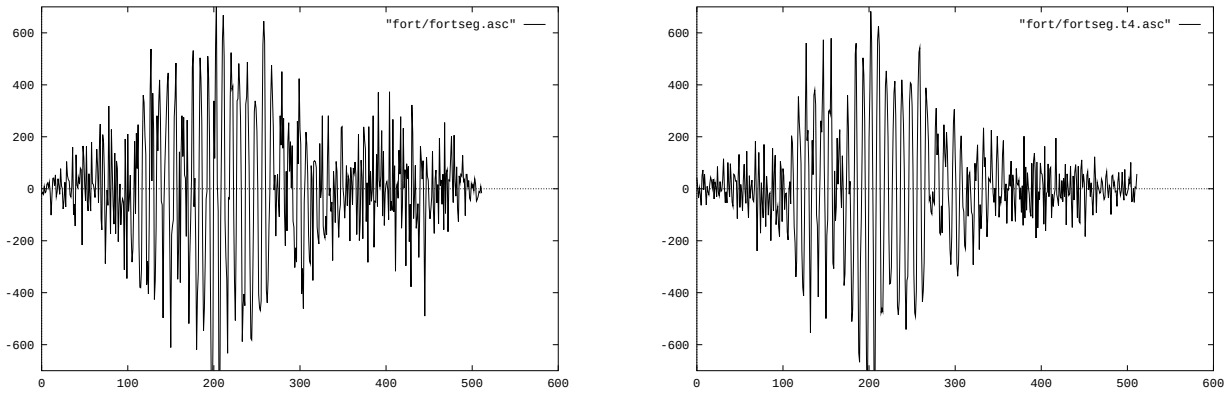

Figure 14: Original and de-noised signals.

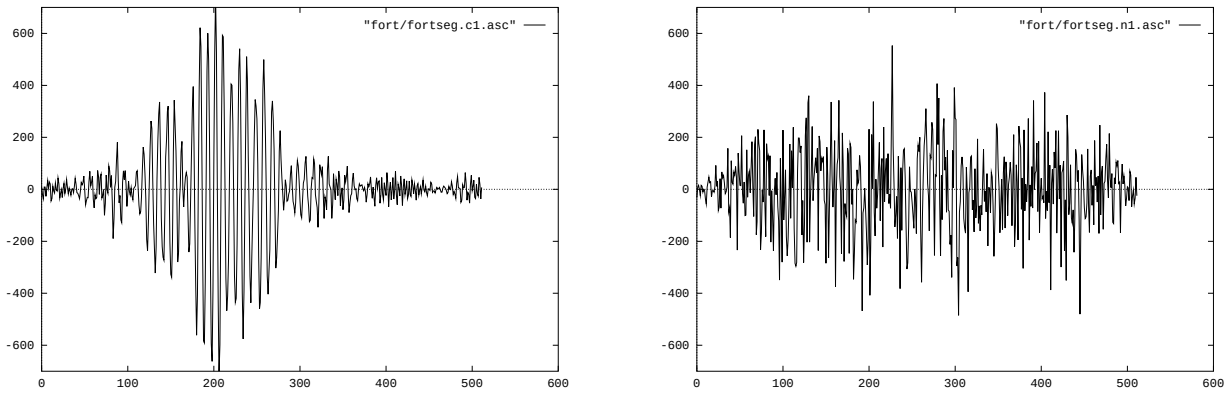

Figure 15: First coherent part and first remainder part.

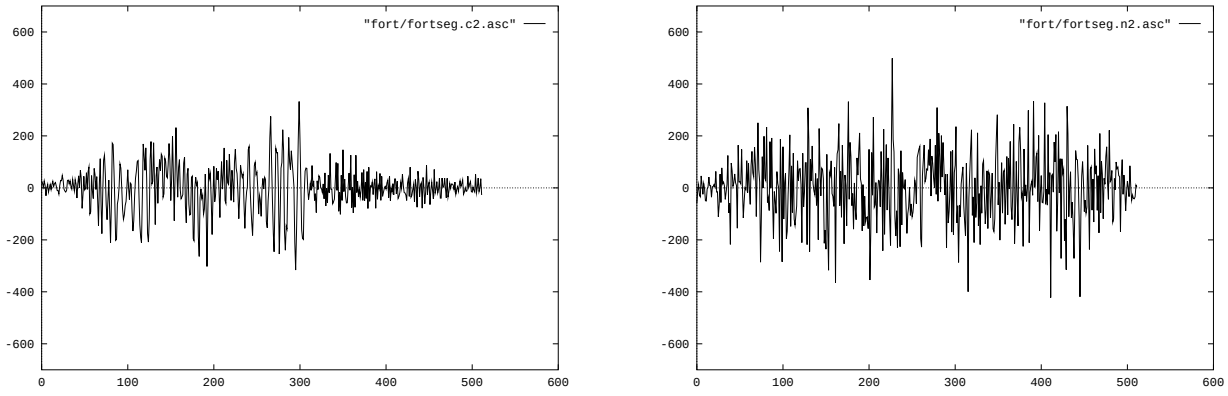

Figure 16: Second coherent part and second remainder part.

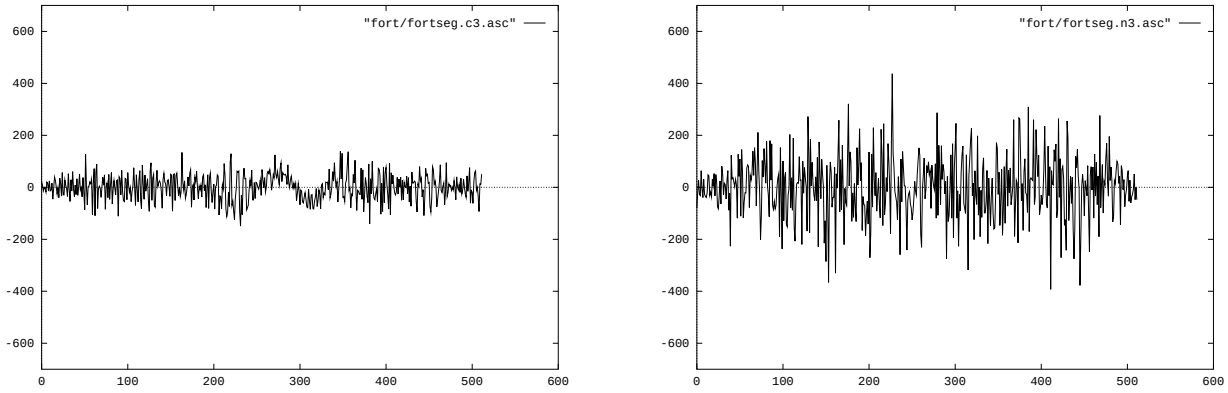

Figure 17: Third coherent part and third remainder part.

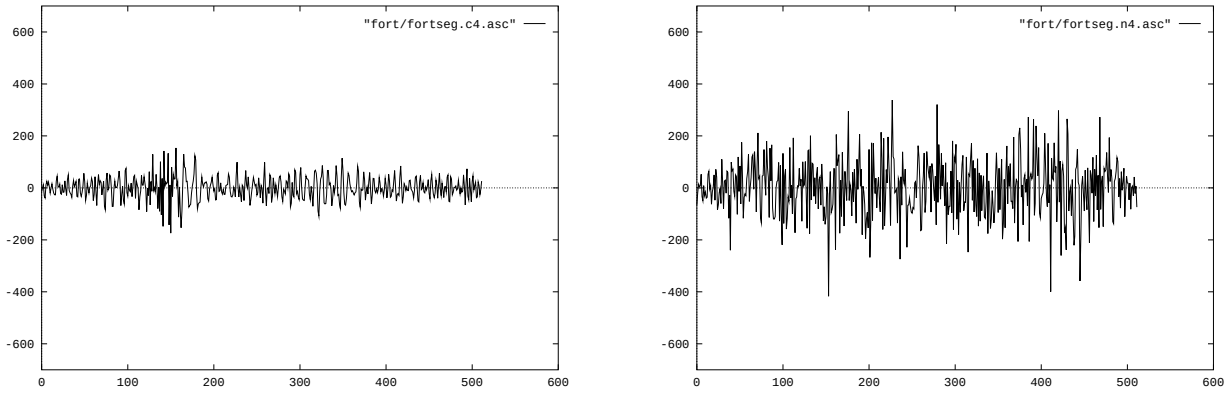

Figure 18: Fourth coherent part and fourth noise part.

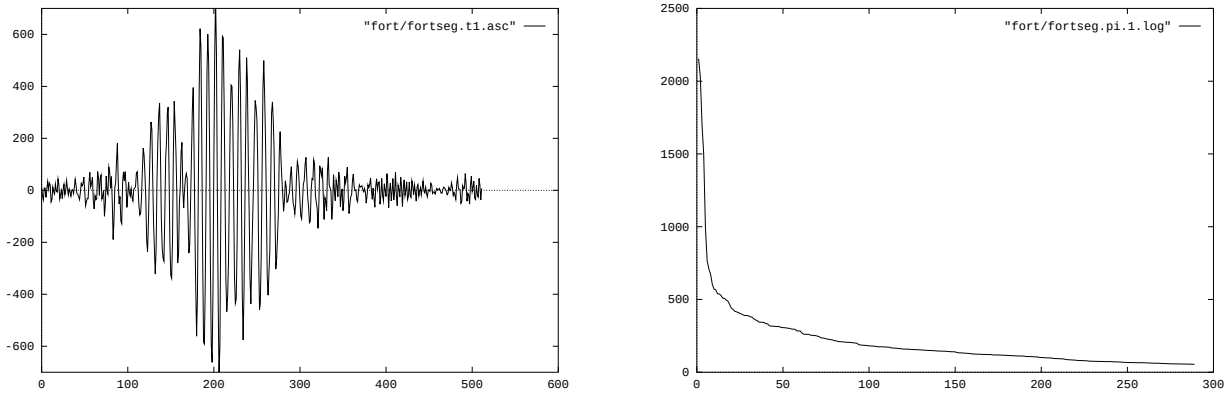

Figure 19: First reconstruction and its sorted remainder coefficients.

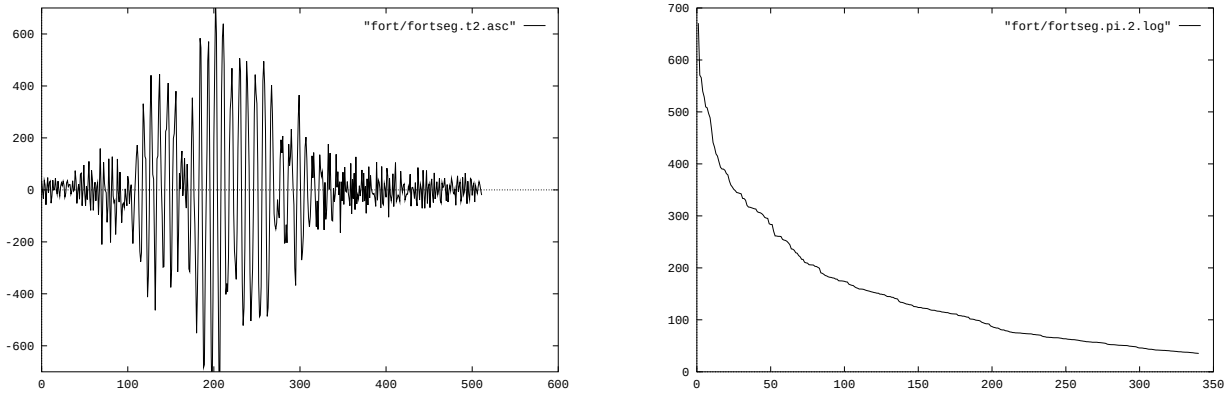

Figure 20: Second reconstruction and its sorted remainder coefficients.

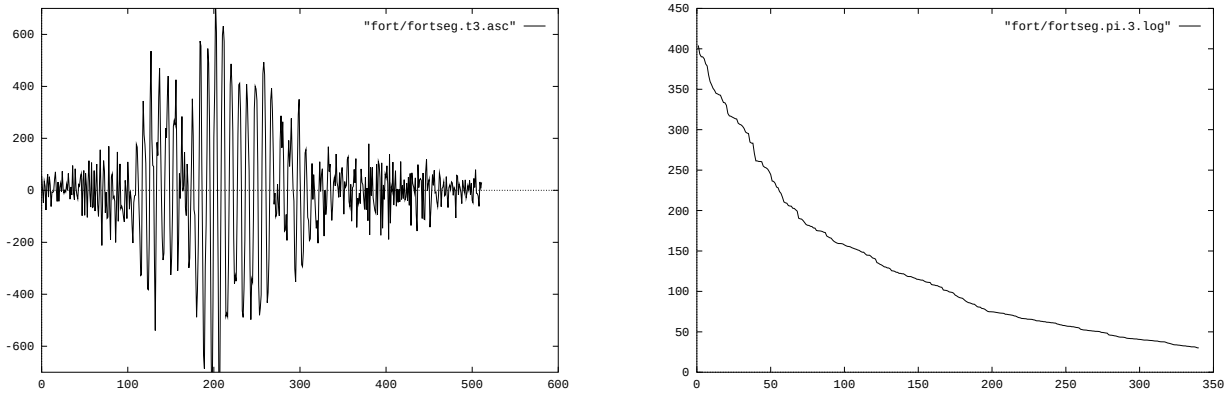

Figure 21: Third reconstruction and its sorted remainder coefficients.

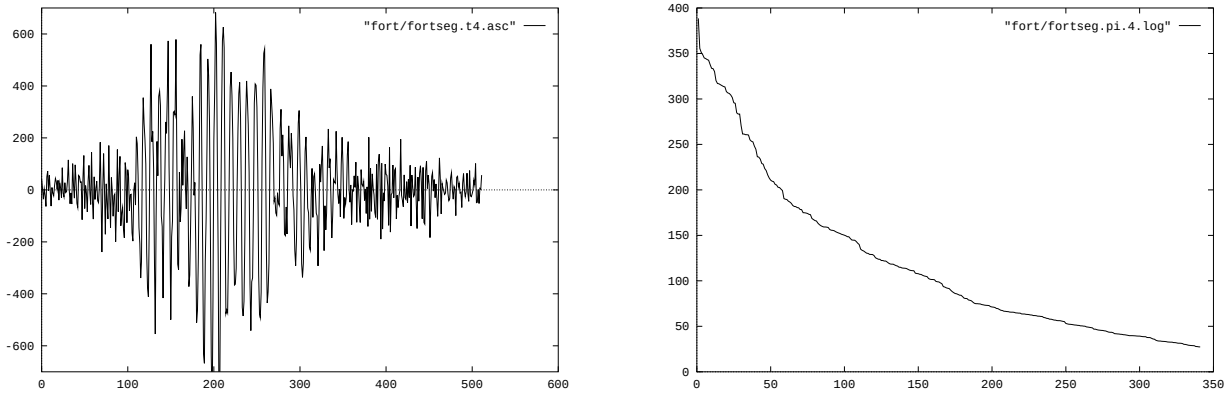

Figure 22: Fourth reconstruction and its sorted remainder coefficients.

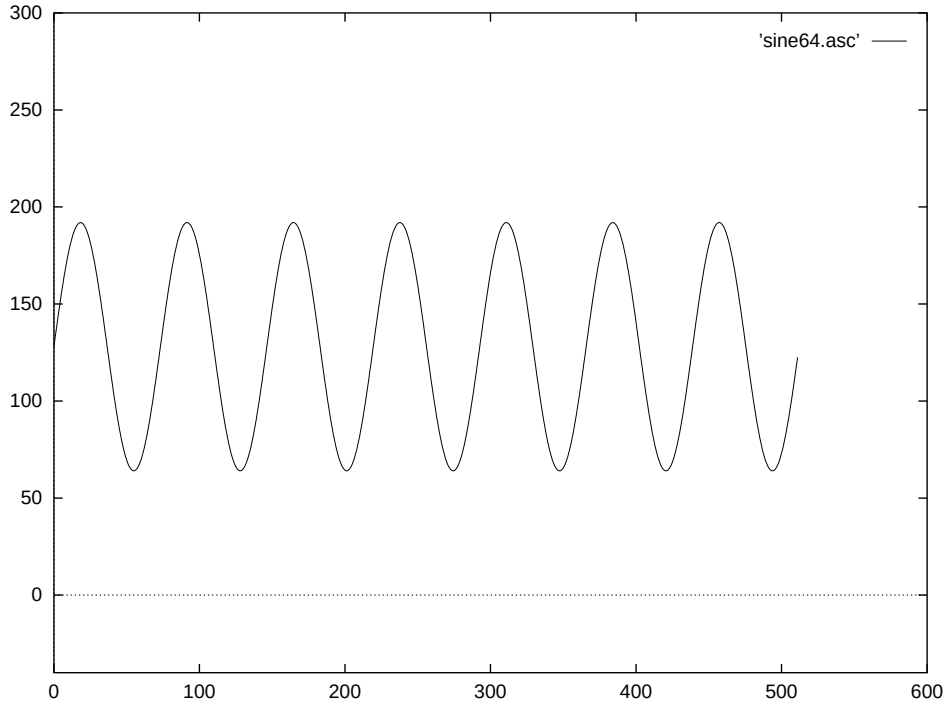

Figure 23: Original sampled sine function

`sine+2db.t4.asc`. This is the denoised signal.

Since the original signal was known, we could compare the actual signal-to-noise ratios of the noised and de-noised sines using `snr sine64.asc sine+2db.asc`, etc., and `snr sine64.asc sine+2db.t4.asc`.

6.2 Results

Actual signal to noise ratio and the gain obtained by denoising are listed in Table 1. The graphs of the functions that produced these results are listed below.

We used 4 iterations and a high threshold, chosen to make the denoised graphs look best. We tried 4 combinations of "`-t`" and " -i" parameter values. Further experiments would yield better results for a target SNR.

SNR(dB):	Nominal	Actual	De-noised	Gain	Noised and De-Noised Filenames
	-4	-3.5	6.5	10.0	sine-4db.asc, sine-4db.t4.asc
	-2	-1.8	7.0	8.8	sine-2db.asc, sine-2db.t4.asc
	+0	0.1	12.2	12.1	sine+0db.asc, sine+0db.t4.asc
	+2	2.1	10.4	8.3	sine+2db.asc, sine+2db.t4.asc
	+4	4.1	11.1	7.0	sine+4db.asc, sine+4db.t4.asc
	+8	8.2	14.2	6.0	sine+8db.asc, sine+8db.t4.asc
	+16	16.4	17.4	1.0	sine+16db.asc, sine+16db.t4.asc
	+24	25.0	28.1	3.1	sine+24db.asc, sine+24db.t4.asc

Table 1: Signal to noise ratio improvement from applying "denoise" to a noisy sine function.

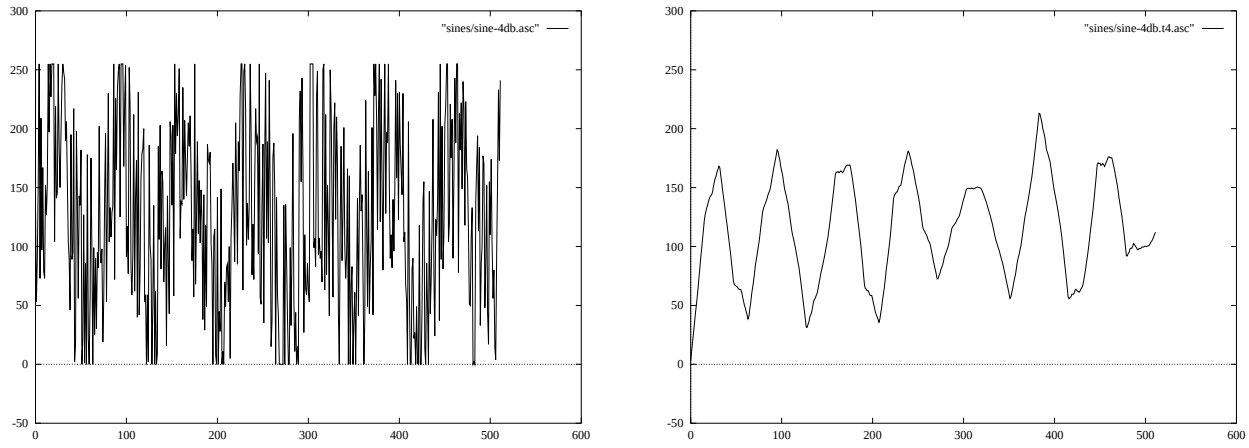

Figure 24: Four-iteration reconstruction of sine with -4 dB SNR

The noised and de-noised SNRs are computed from the middle half of the signal (samples 128 to 384) in order to avoid edge effects. If edges are included, the gain is lower.

7 Conclusion

We have analyzed one-dimensional sampled continuous waveforms by decomposing them into building blocks or atoms. The main desirable feature is that the atoms be well-localized both in the time domain and in the frequency domain, in the manner of windowed sines and cosines

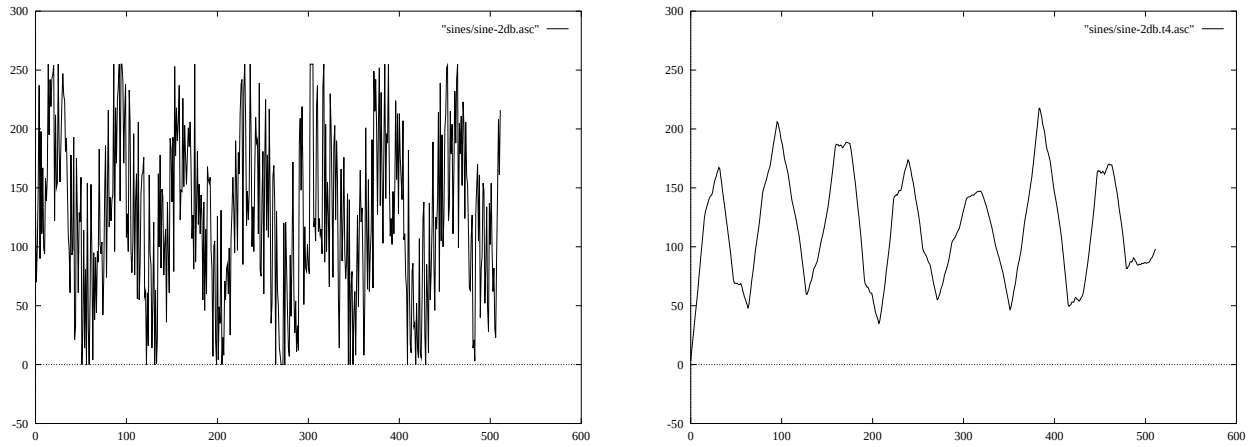

Figure 25: Four-iteration reconstruction of sine with -2 dB SNR

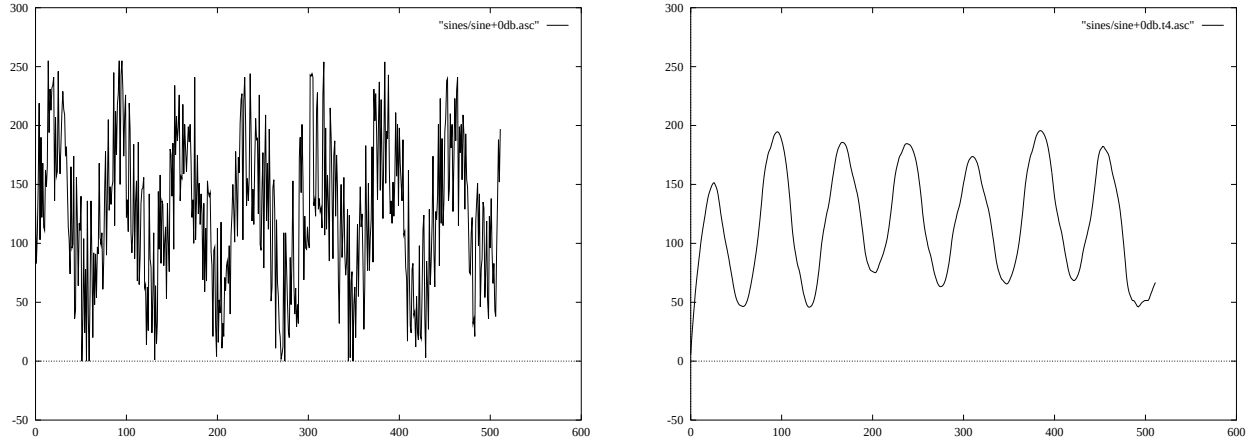

Figure 26: Four-iteration reconstruction of sine with 0 dB SNR

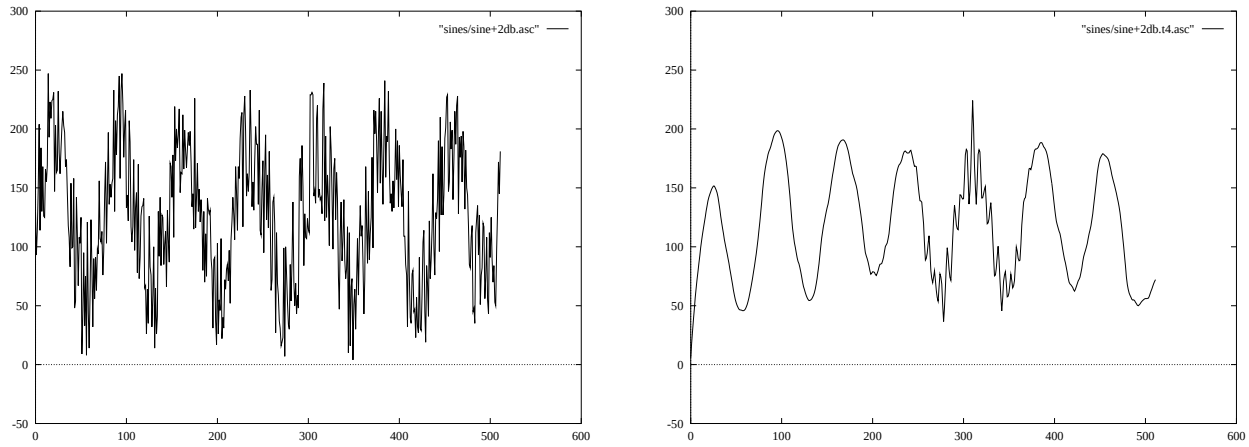

Figure 27: Four-iteration reconstruction of sine with +2 dB SNR

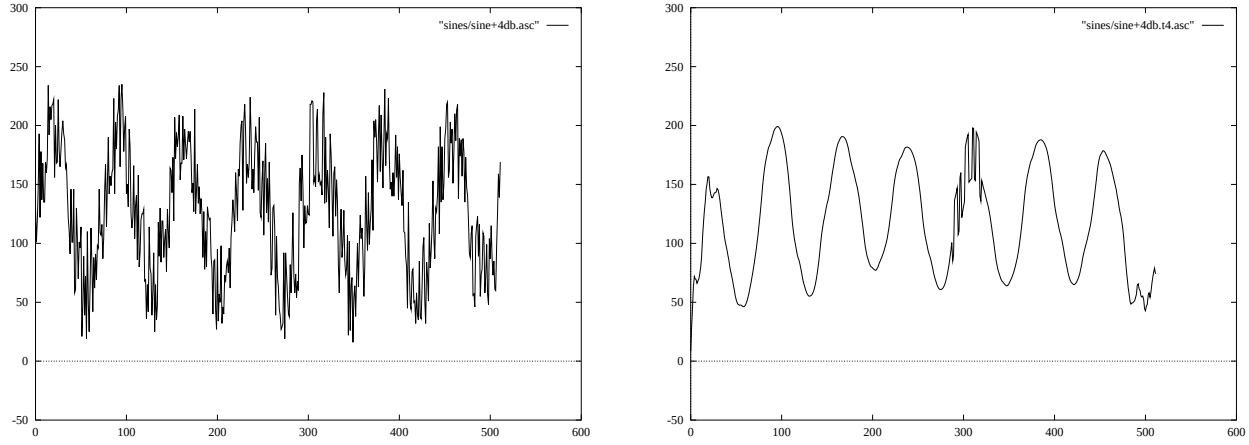

Figure 28: Four-iteration reconstruction of sine with +4 dB SNR

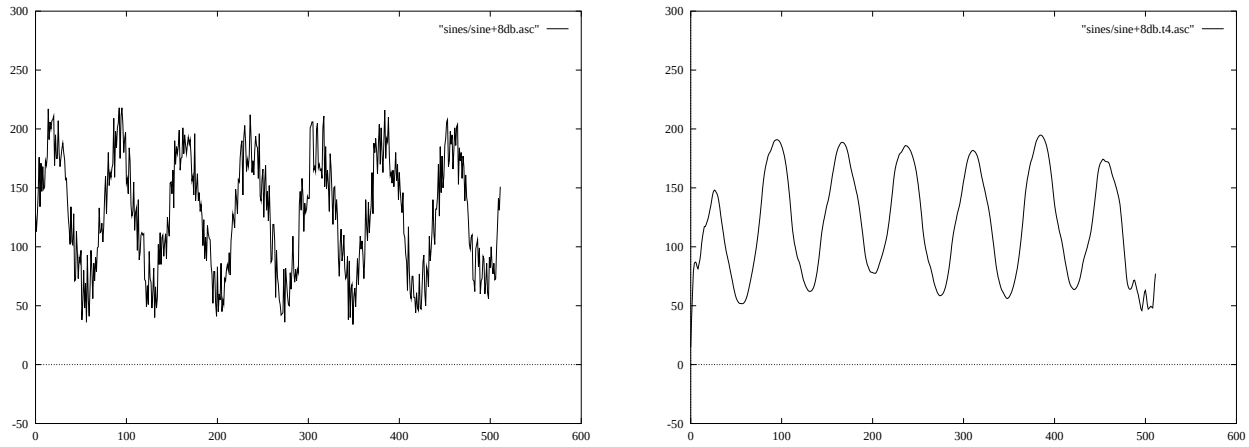

Figure 29: Four-iteration reconstruction of sine with +8 dB SNR

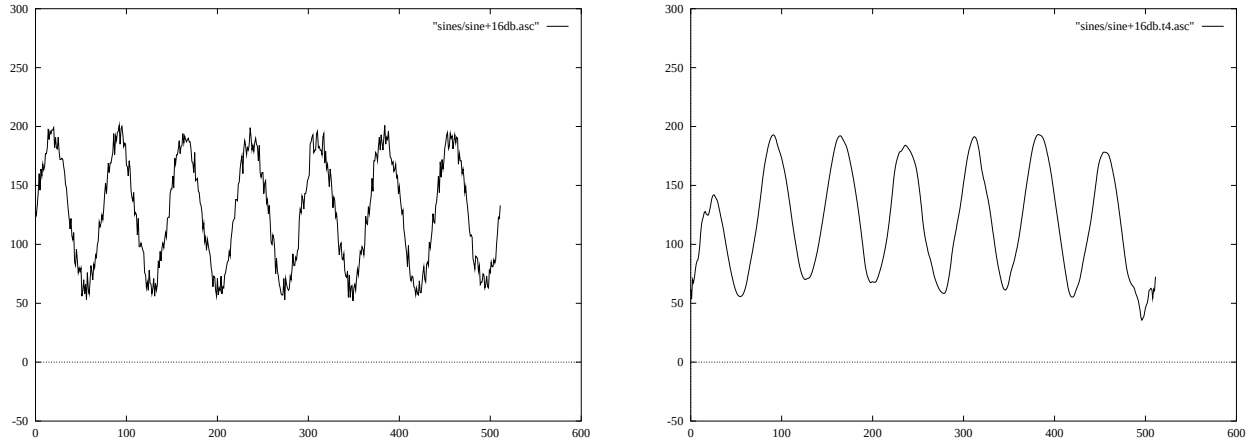

Figure 30: Four-iteration reconstruction of sine with +16 dB SNR

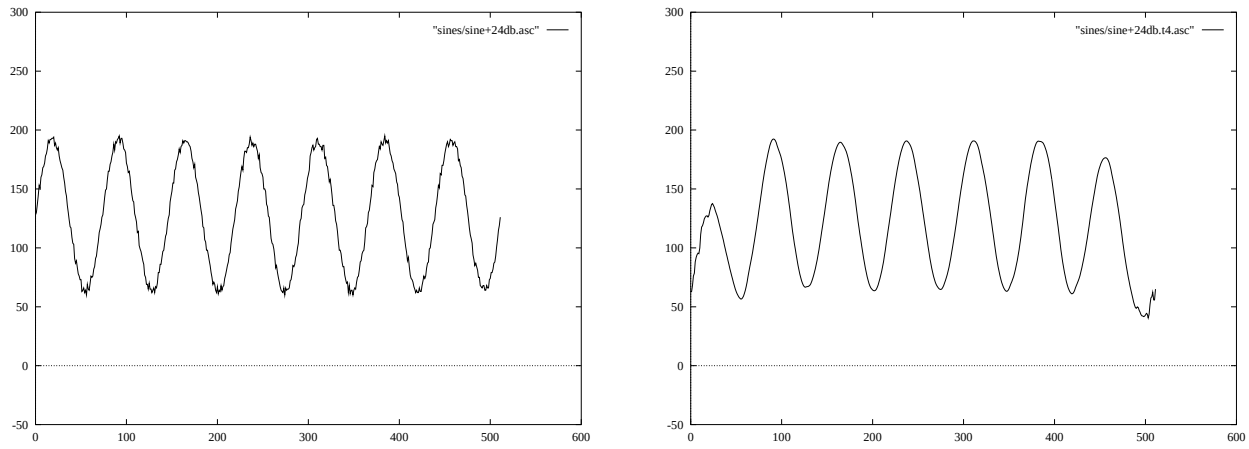

Figure 31: Four-iteration reconstruction of sine with +24 dB SNR

or Gabor functions. Wavelets and related functions like wavelet packets and local cosines can also be used in such decompositions. They have the added advantage that the resulting expansions are orthogonal or energy-preserving, allowing us to compare and adapt expansions to signals in order to minimize the cost of representation. Such adapted decompositions perform compression and analysis simultaneously. We have designed an idealized graphical presentation of the time-frequency information obtained by such a best-adapted waveform analysis, and from such presentations we can recognize and extract transient features such as parts of speech. Finally, we have shown that the negligibly small components in the analysis may be treated as noise and discarded, and we have designed an iterative algorithm for extracting coherent signals from such noise. Adapted wavelet analyses are practical for realistic signal sizes because the underlying algorithms have low computational complexity.

A Instructions and sample output for the program "denoise"

A.1 Summary of the algorithm

Denoise takes as input a one-dimensional signal such as a soundfile, and:

1. breaks it into windows of desired length (a power of two).
2. in each window, tries a wavelet packet transform with a number of filters, keeps the one which has the lowest entropy for its best basis, and sorts the coefficients by decreasing amplitude.
3. in each window, eliminates the coefficients in the wavelet packet that have an amplitude less than a certain energy threshold.
4. in each window, evaluates a cost function of the coefficients repeatedly until the cost is higher than a certain cost threshold. The coefficients not considered are discarded
5. reconstructs a new signal from the remaining coefficients

The parameters that can be modified are:

1. the cost function which is used in step 4 above
2. the energy threshold which is used in step 3 above
3. the threshold for the cost function
4. the number of times this process is iterated

A.2 Manual page

DENOISE(1) UNIX Programmer's Manual DENOISE(1)

NAME

denoise - denoise a one-dimensional ASCII signal file

SYNOPSIS

denoise [-wleitmf] filename

DESCRIPTION

Denoise takes as input a one-dimensional signal such as a soundfile, and:

- breaks it into windows of desired length (a power of two).
- in each window, tries a wavelet packet transform with a number of filters, keeps the one which has the lowest entropy for its best basis, and sorts the coefficients by decreasing amplitude.
- in each window, eliminates the coefficients in the wavelet packet that have an amplitude less than a certain energy threshold.
- Either:
  - in each window, evaluates a cost function of the coefficients repeatedly until the cost is higher than a certain cost threshold. The coefficients

not considered are discarded.
- does Wiener-filtering, by extrapolating the noise distribution (exponential least squares fit).
- does Wiener-extraction (determining the number of coefficients to retain from the exponential fit of the noise coefficients).

- reconstructs a new signal from the remaining coefficients

The parameters that can be modified are:

- the energy threshold which is used in step 3 above.
- the threshold for the cost function.
- the number of times this process is iterated.

OPTIONS

-l Produce a log of all wavelet packet coefficients in the selected basis (before E-thresholding).

-w Use Wiener-filtering. The -t option is meaningless when used with Wiener-filtering. Note that an E-threshold must be specified otherwise one will be provided.

-x Use Wiener-extraction. The -t option is meaningless when used with Wiener-extraction. Note that an E-threshold must be specified otherwise one will be

provided.

-s Do smooth cuts. The rejected coefficients are modulated by a bell function to make the algorithm less brutal.

-e threshold
select the energy-threshold to use. Possible values for threshold are of the form number , in which case the threshold is assumed to be an absolute value, or number% , where the threshold will be relative to the signal window energy.

-i number
set the number of iterations for the algorithm. Default is 1.

-t threshold
set the threshold for the cost function. Default is 1.

-m number
set the maximum number of levels in the wavelet packet expansion. Default is 9. The window width is 2 to the power of the maximum level.

-f parse the arguments quietly. This is useful for error-checking in shell-scripts.

```
SEE ALSO
     winfold(1), gnuplot(1), denoise2d(1)

BUGS
     Only ASCII files can be used as input.
```

A.3 Output from `denoise -i4 -m9 -t0.2 sine+8db.asc`

```
Copyright (C)1989-94 Wickerhauser Consulting,
R.R. Coifman, F. Majid, Y. Meyer, M.V. Wickerhauser.
Patents applied for.  All rights reserved.
 Version 2.0, 11 September 1994.

denoise1d: sound filtering with periodized wavelet packets.
by Fazal Majid, 1992.

*** Window 1 ***
:    Iteration 1
:    Best-basis search:
:    using Daubechies  8 -- entropy = 1.368005
:    Reconstruction phase:
:    retained 512 coefficients after E-thresholding
:    retained 7 coefficients after cost-thresholding
:    noise = 2.3 % of original energy
:
:    Iteration 2
:    Best-basis search:
:    using Coifman 12 -- entropy = 7.143196
```

```
:    Reconstruction phase:
:    retained 512 coefficients after E-thresholding
:    retained 1 coefficients after cost-thresholding
:    noise = 2.2 % of original energy
:
:    Iteration 3
:    Best-basis search:
:    using Coifman 12 -- entropy = 7.218138
:    Reconstruction phase:
:    retained 512 coefficients after E-thresholding
:    retained 1 coefficients after cost-thresholding
:    noise = 2.1 % of original energy
:
:    Iteration 4
:    Best-basis search:
:    using Coifman 12 -- entropy = 7.253392
:    Reconstruction phase:
:    retained 512 coefficients after E-thresholding
:    retained 1 coefficients after cost-thresholding
:    noise = 2.0 % of original energy
:
```

References

[1] Gregory Beylkin, Ronald R. Coifman, and Vladimir Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, XLIV:141–183, 1991.

[2] Ronald R. Coifman, Fazal Majid, and Mladen Victor Wickerhauser. Denoise. Available from Fast Mathematical Algorithms and Hardware Corporation, 1020 Sherman Avenue, Hamden, CT 06514 USA, 1992.

[3] Ronald R. Coifman, Yves Meyer, Stephen R. Quake, and Mladen Victor Wickerhauser. Signal processing and compression with wavelet packets. In Meyer and Roques [9], pages 77–93.

[4] Ronald R. Coifman and Mladen Victor Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 32:712–718, March 1992.

[5] Christophe D'Alessandro, Xiang Fang, Eva Wesfreid, and Mladen Victor Wickerhauser. Speech signal segmentation via Malvar wavelets. In Meyer and Roques [9], pages 305–308.

[6] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM Press, Philadelphia, 1992.

[7] Fazal Majid. Applications des paquets d'ondelettes au débruitage du signal. Preprint, Department of Mathematics, Yale University, 28 July 1992. Rapport d'Option, Ecole Polytechnique.

[8] Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[9] Yves Meyer and Sylvie Roques, editors. *Progress in Wavelet Analysis and Applications*, Proceedings of the International Conference "Wavelets and Applications," Toulouse, France, 8–13 June 1992, Gif-sur-Yvette, France, 1993. Observatoire Midi-Pyrénées de l'Université Paul Sabatier, Editions Frontieres.

[10] David Rochberg and Mladen Victor Wickerhauser. WPLab version 3.03 (for NeXT computers). Available by anonymous ftp from [14], 1992.

[11] Mladen Victor Wickerhauser. Computation with adapted time-frequency atoms. In Meyer and Roques [9], pages 175–184.

[12] Mladen Victor Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. AK Peters, Ltd., Wellesley, Massachusetts, 9 May 1994. With optional diskette.

[13] Mladen Victor Wickerhauser. Large-rank approximate principal component analysis with wavelets for signal feature discrimination and the inversion of complicated maps. *Journal of Chemical Information and Computer Science*, 34(5):1036–1046, September/October 1994. Proceedings of Math-Chem-Comp 1993, Rovinj, Croatia.

[14] wuarchive. Washington University in St. Louis. InterNet Anonymous File Transfer (ftp) Site wuarchive.wustl.edu [128.252.135.4], 1991.