Translation-Invariant De-Noising

R.R. Coifman and D.L. Donoho

Yale University and Stanford University

Abstract

De-Noising with the traditional (orthogonal, maximally-decimated) wavelet transform sometimes exhibits visual artifacts; we attribute some of these – for example, Gibbs phenomena in the neighborhood of discontinuities – to the lack of translation invariance of the wavelet basis. One method to suppress such artifacts, termed "cycle spinning" by Coifman, is to "average out" the translation dependence. For a range of shifts, one shifts the data (right or left as the case may be), De-Noises the shifted data, and then unshifts the de-noised data. Doing this for each of a range of shifts, and averaging the several results so obtained, produces a reconstruction subject to far weaker Gibbs phenomena than thresholding based De-Noising using the traditional orthogonal wavelet transform.

Cycle-Spinning over the range of *all* circulant shifts can be accomplished in order $n \log_2(n)$ time; it is equivalent to de-noising using the undecimated or stationary wavelet transform.

Cycle-spinning exhibits benefits outside of wavelet de-noising, for example in cosine packet denoising, where it helps suppress 'clicks'. It also has a counterpart in frequency domain de-noising, where the goal of translation-invariance is replaced by modulation invariance, and the central shift-De-Noise-unshift operation is replaced by modulate-De-Noise-demodulate.

We illustrate these concepts with extensive computational examples; all figures presented here are reproducible using the WaveLab software package.

1 Introduction

In the last few years, there has been considerable interest in the use of wavelet transforms for removing noise from signals and images. One method, applied by the authors and their collaborators, has been the use of transform-based thresholding, working in three steps:

- Transform the noisy data into an orthogonal domain.
- Apply soft or hard thresholding to the resulting coefficients, thereby suppressing those coefficients smaller than a certain amplitude.
- Transform back into the original domain.

This approach has been the most studied and applied when the transform in question is the wavelet transform; then we speak of Wavelet Shrinkage – see [18] and references there. However, it makes sense quite generally, so one can apply thresholding in a fourier, cosine packet, wavelet packet, or in some other orthogonal transform domain [8, 12].

The quantitative theory of this method is now well developed – see again [18, 11, 17] and references therein; one can even show that this approach has various optimality and near-optimality properties in comparison to other methods. Nevertheless, scientists and other users have mentioned to us a number of concerns about artifacts of various kinds that such methods exhibit; our aim in this paper is to discuss such artifacts and to describe a simple set of tools which helps to suppress them. To the extent that simple thresholding in an orthogonal domain may be thought of as first-generation de-noising, we might call the methods developed here "second-generation de-noising"; compare the rejoinder to the discussion in [18].

1.1 Artifacts

The type of artifacts suffered when using transform domain thresholding depend on the kind of transform domain one is working in.

For wavelet de-noising, the artifacts have to do with behavior near singularities. In the neighborhood of discontinuities, wavelet de-noising can exhibit pseudo-Gibbs phenomena, alternating undershoot and overshoot of a specific target level. While these phenomena are much better than in the case of Fourier-based de-noising (in which Gibbs phenomena are global, rather than local, and of large amplitude), it seems reasonable to try to do better still. An important observation about such artifacts: their size is connected intimately with the actual location of the discontinuity. For example, when using Haar wavelets, a discontinuity precisely at location $n/2$ will lead to essentially no pseudo-Gibbs oscillations; a discontinuity near a binary irrational like $n/3$ will lead to significant pseudo-Gibbs oscillations

For Cosine Packet de-noising, which is based on segmentation of the signal using a recursive dyadic grid, de-noised series can exhibit discontinuities at segmentation points. When the signal corresponds to acoustic information, these discontinuities take the form of audible "clicks", which are distracting and have nothing to do with the actual sounds in the original signal.

For Wavelet Packet de-noising, which is based on segmentation of the fourier transform of the signal using a recursive dyadic grid, de-noised series can exhibit mirror symmetries across segmentation points in the frequency domain. When the signal corresponds to acoustic information, these artificial symmetries generate spurious musical notes, which are distracting and have nothing to do with the actual sounds in the original signal.

1.2 Shifts in Time and Frequency

The artifacts we have just mentioned are all connected in some way with the precise alignments between features in the signal and features of basis elements; signals exhibiting similar features but with slightly different alignment in time or frequency might generate fewer of the artifacts we are interested in. We have already mentioned that for the Haar basis, discontinuities at $n/2$ cause no real problems. Similarly, in cosine packet analysis, signals which essentially vanish near segmentation points of a partition will generate few

"click" artifacts, and in wavelet packet analysis, signals with no frequency content near segmentation points in the frequency domain will generate few artifacts in wavelet packet de-noising.

One approach to correct unfortunate mis-alignments between features in the signal and features in a basis: forcibly shift signals so that their features change positions. The hope is that an analysis of the shifted signal will not exhibit the undesirable artifacts, and that this analysis can later be unshifted. To make our meaning clear, we introduce two operators, of time- and frequency- shift respectively. For a signal $(x_t : 0 \le t < n)$, we let S_h denote the circulant shift by h, $(S_h x)_t = x_{(t+h) \bmod n}$, and we let M_ξ denote modulation by ξ: $(M_\xi x)_t = e^{i\xi t} x_t$. Both operators are unitary, and hence invertible: $S_{-h} = (S_h)^{-1}$, $M_{-\xi} = (M_\xi)^{-1}$. In terms of these operators, the idea of shifting to avoid artifacts is just this: given an analysis technique T, calculate, instead of T, the time-shifted version $\tilde{T}(x; S_h) = S_{-h}(T(S_h(x)))$. Or the frequency-shifted version $\tilde{T}(x; M_\xi) = M_{-\xi}(T(M_\xi(x)))$.

The missing ingredient: knowledge of the "correct" choice of the shift parameter h or ξ. One reasonable approach to choosing such a parameter is optimization: develop a quantitative measure of the "well-alignedness" of the signal, choosing a best value of the shift parameter h or ξ. We describe below such an approach, which can be accomplished by a fast and elegant algorithm.

1.3 Averaging Shifts

It may well be that a given signal can be re-aligned to minimize artifacts, but there is no guarantee that this will always be the case. When a signal contains several discontinuities, these may interfere with each other: the best shift for one discontinuity in the signal may also be the worst shift for another discontinuity. Consequently, we have abandoned the hope of locating a single shift, ideal for aligning all features of the signal, and tried instead another approach, which is to apply a *range* of shifts, and *average* over the several results so obtained.

For time shifts this means we consider a range H of shifts and set

$$\bar{T}(x; (S_h)_{h \in H}) = \mathrm{Ave}_{h \in H} S_{-h}(T(S_h(x))) \tag{1}$$

or, in words,

Average[Shift-DeNoise-Unshift]

For frequency shifts this means we consider a range X of modulations and set

$$\bar{T}(x; (M_\xi)_{\xi \in X}) = \mathrm{Ave}_{\xi \in X} M_{-\xi}(T(M_\xi(x))) \tag{2}$$

or, in words,

Average[Modulate-DeNoise-DeModulate] .

One of us - Coifman - in collaboration with L. Woog, M. Goldberg, and N. Saito [2, 22] has been experimenting with ideas of this kind in applied work; he has called this "Cycle Spinning". The other author, after discussions with Coifman, became interested in the topic, developed fast algorithms, and made a series of computational experiments.

1.4 Our Goals

Our purpose in writing this paper is to call the reader's attention to the use of averaging over shifts, its benefits in suppressing artifacts, and to give a number of computational examples. We supply graphical displays of traditional de-noising and a modification using cycle-spinning, as well as numerical tables for quantitative comparison. It turns out that cycle-spinning gives results that are visually better, often dramatically so, and quantitatively better, as nearly halving the mean-squared error in some examples.

We will spend a considerable amount of time on a specific variant: wavelet de-noising averaged over all n circulant shifts. This version of cycle-spinning is, naturally, invariant under circulant shifts, and so translation-invariant - hence the title of the paper. The method can be calculated rapidly - in $n \log(n)$ time, despite appearances.

For the Haar wavelet, we will also show that translation-invariant approaches yield several theoretical advantages. In addition to faster rates of convergence, there is the visually satisfying fact that translation-invariant de-noising is non-oscillatory in expectation.

In this paper, all computational results are reproducible, meaning that the code which generated the figures is available over Internet, following the discipline indicated in [1].

2 Examples

We illustrate the ideas of the introduction by a series of computational examples, based on artificial signals.

2.1 Wavelet Domain De-Noising

Our signal examples are the same as those constructed and analyzed in [18]. Figure 1 shows 4 signals of length $n = 2048$, *Blocks*, *Bumps*, *HeaviSine*, *Doppler*; these were chosen to display various inhomogeneities.

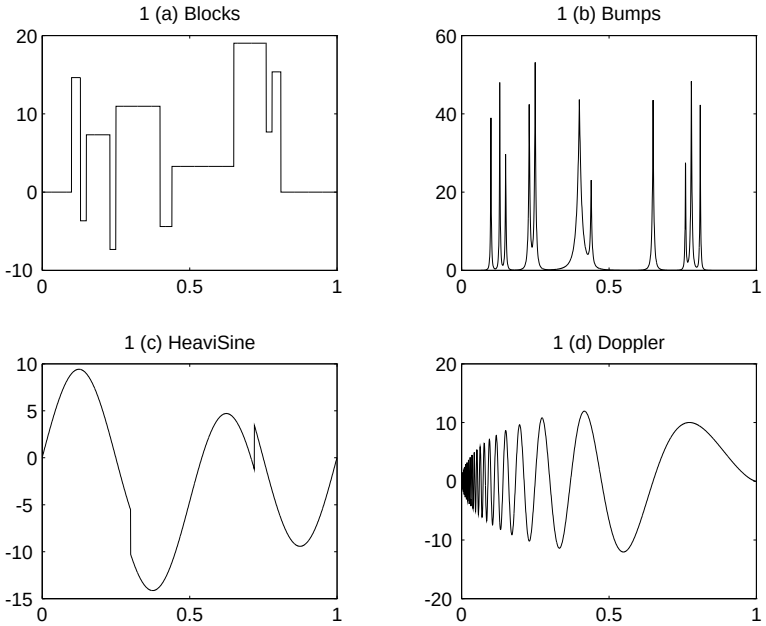


Figure 2 shows noisy versions of the 4 signals, where Gaussian white noise has been added; in each case, the noisy data $y = s + z$, where s is the corresponding noiseless signal and $z \sim_{iid} N(0, 1)$.

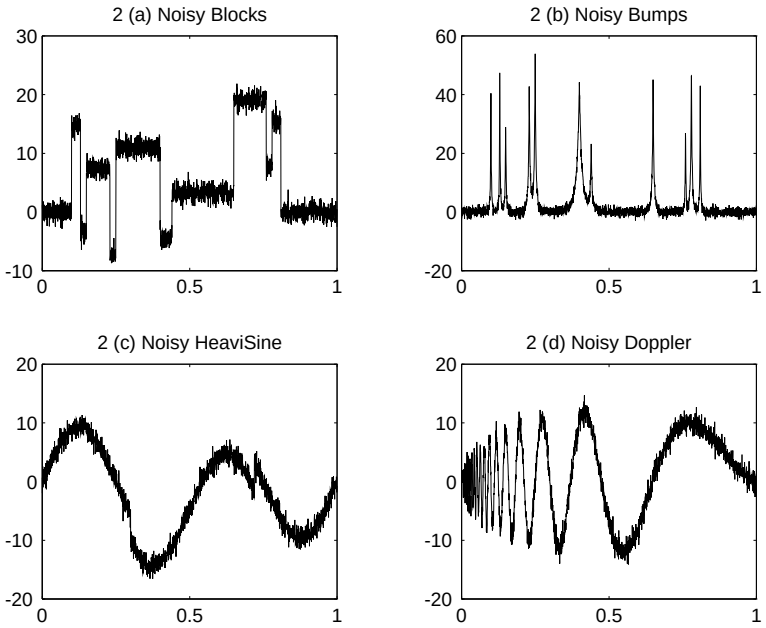
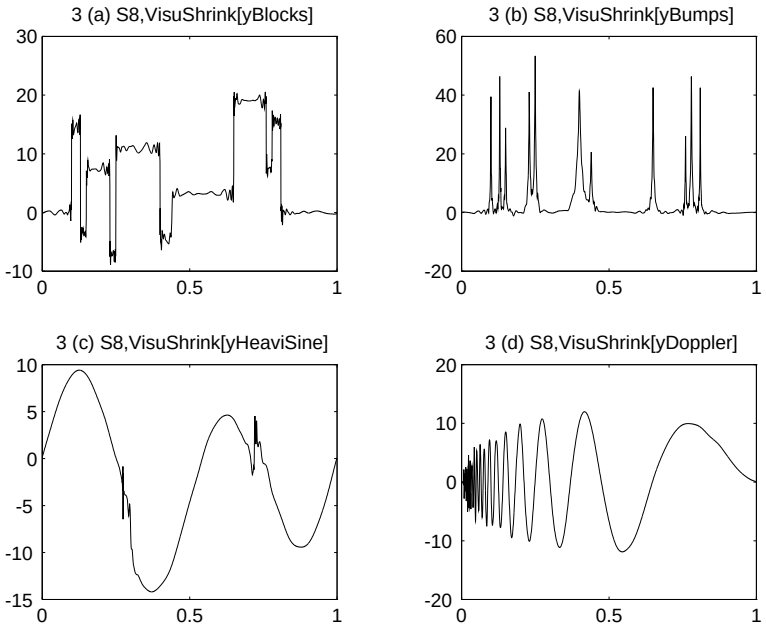


Figure 3 shows simple Wavelet Shrinkage applied to the noisy data. The noisy signal is transformed into the wavelet domain using an orthogonal periodic wavelet transform based on Nearly Symmetric wavelets with 8 vanishing moments. The wavelet coefficients (except at the coarsest level) are subjected to soft thresholding, $\eta_t(w) = \mathrm{sgn}(w)(|w| - t)_+$, with threshold $t = t_n = \sqrt{2\log(n)}\sigma$. The result is then inverse-transformed. Let W_{S8} denote the wavelet transform with Symmlet 8 wavelets. The whole de-noising process amounts to a nonlinear operator $T_{\eta,S8}(y)$, where $T_{\eta,S8} = W_{S8}^{-1} \circ \eta_{t_n} \circ W_{S8}$.



From theory, we know that the resulting figures are substantially noise-free – they look substantially the same under independent realizations – though they are not oscillation-free. The oscillations we see in Figure 3 are especially pronounced in the vicinity of discontinuities and other rapid changes. These are "pseudo-Gibbs" oscillations caused by the fact that the curves in question are partial reconstructions obtained using only terms

from a subset of the wavelet basis. (Indeed, for the De-Noised object, many of the thresholded coefficients are zero, which is the same as saying that the reconstruction uses only a subset of the full set of basis elements.) In contrast to the classical Gibbs-Phenomena associated with Fourier Analysis, the pseudo-Gibbs phenomena are much better behaved – much better localized and much more moderate in oscillation – nevertheless they are visually annoying.

Figure 4 presents results from Cycle-Spinning Wavelet Shrinkage of the 4 noisy signals. Here we apply (1), averaging over the range of shifts $H = H_{16} = \{h : 0 \le h < 16\}$ the denoising operator $T_{\eta,S8}$ based, as above, on soft thresholding in the Symmlet 8 basis. It is evident that the pseudo-Gibbs oscillations are considerably reduced.

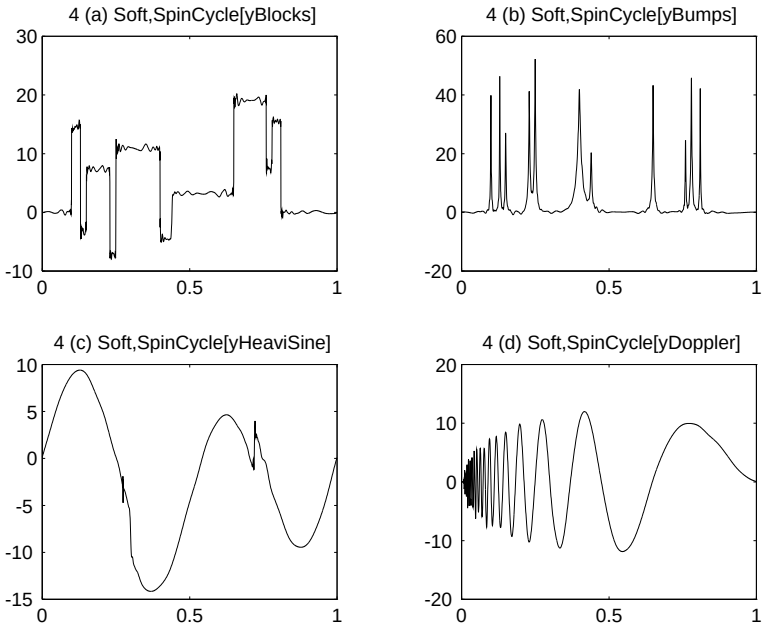
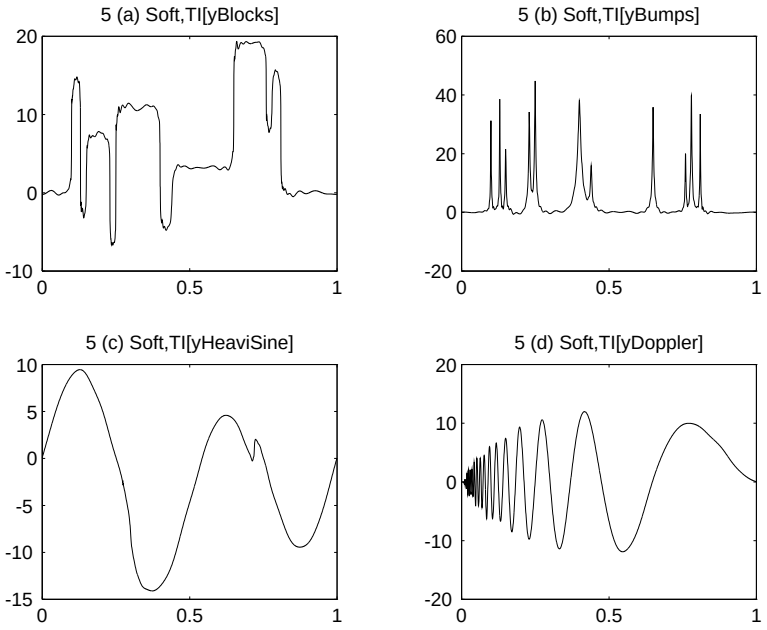


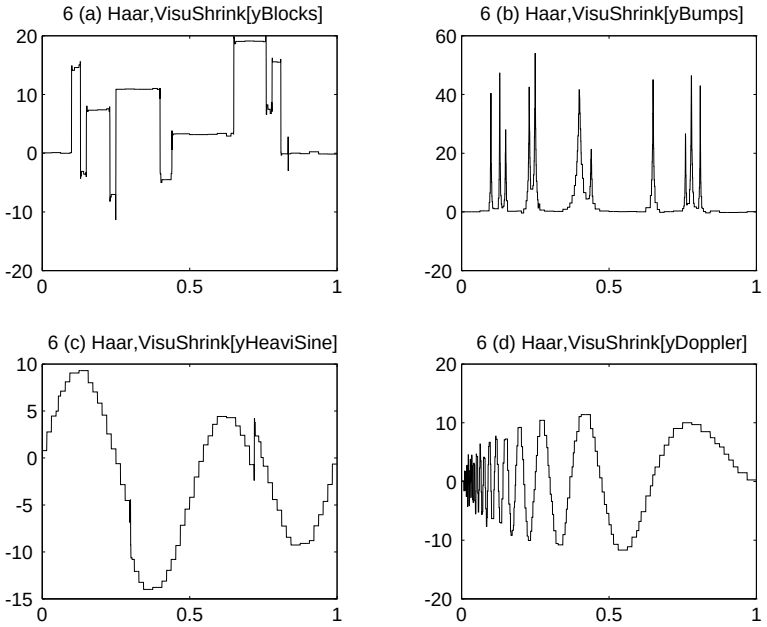
Figure 5 presents results from Fully Translation-Invariant De-Noising on the 4 noisy signals. Here we apply (1), averaging over all n circulant shifts $H = H_n = \{h : 0 \le h < n\}$, using again the De-Noising operator $T_{\eta,S8}$ based, as above, on the Symmlet 8 basis.



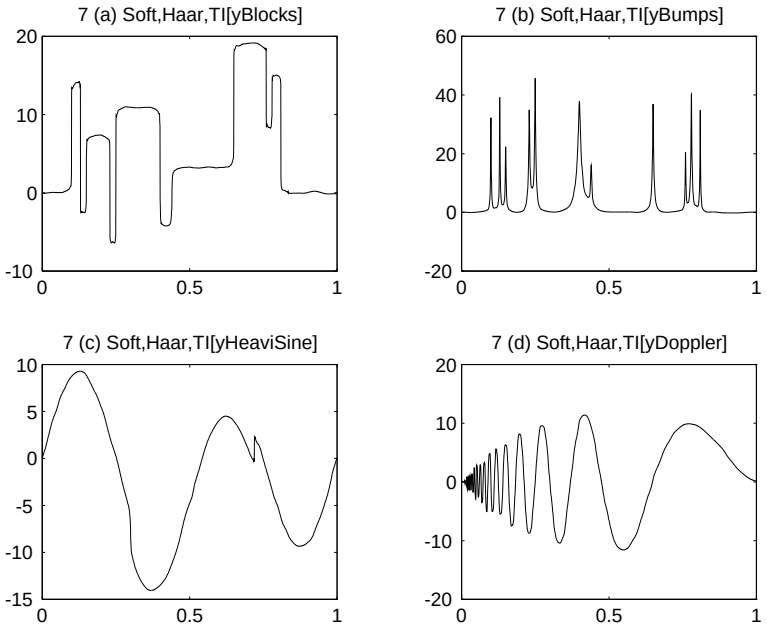
It is evident that the pseudo-Gibbs oscillations are again considerably reduced; even more reduced than in Figure 4. A benefit of the Fully Translation-Invariant approach over the 16-shift approach of Figure 4 is that there are no arbitrary parameters to set – one doesn't have to decide whether to average over 16, or 20, or only 7 shifts.

2.2 Variations on Wavelet De-Noising

Many variations of the above experiments can be conducted.

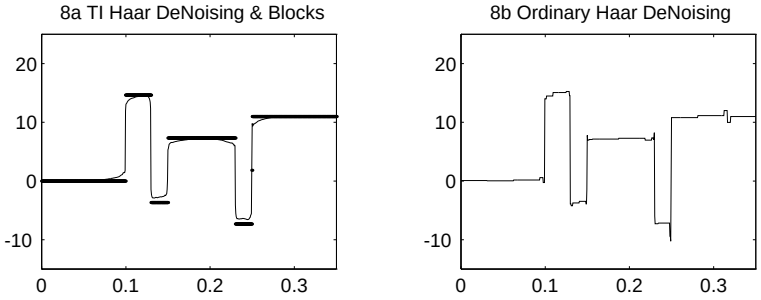


2.2.1 Haar Wavelets. Figure 6 shows the result of standard soft thresholding DeNoising of the four signals in in the Haar basis. Two sorts of artifacts are evident: (a) the stairstep nature of the partial Haar approximations in regions of smooth behavior, and (b) pseudo-Gibbs phenomena in the vicinity of discontinuities.



For comparison, Figure 7 shows Fully Translation-Invariant reconstruction using Haar De-Noising. The reconstruction no longer has a stairstep character. Also, Gibbs phenomena in the neighborhood of discontinuities are suppressed.

To illustrate these points, we give in Figure 8 a closeup of the Fully TI Haar reconstruction of the Blocks object. The noiseless object and the ordinary Haar reconstructions are also indicated. The TI reconstruction is better both in smooth parts and in the vicinity of jumps.



2.2.2 Hard Thresholding. $T_{\nu,S8}$ refers to the operation of applying the nonlinearity $\nu_t(w) = w1_{\{|w|>t\}}$ to all the wavelet coefficients except the coarse scale averages, in a wavelet transform based on Symmlets with 8 vanishing moments. Previous experience with hard thresholding in traditional non-invariant de-noising suggests that it produces somewhat greater oscillations in the vicinity of discontinuities than does soft thresholding. However, one expects that perhaps the translation-invariant approach will damp some of those oscillations.

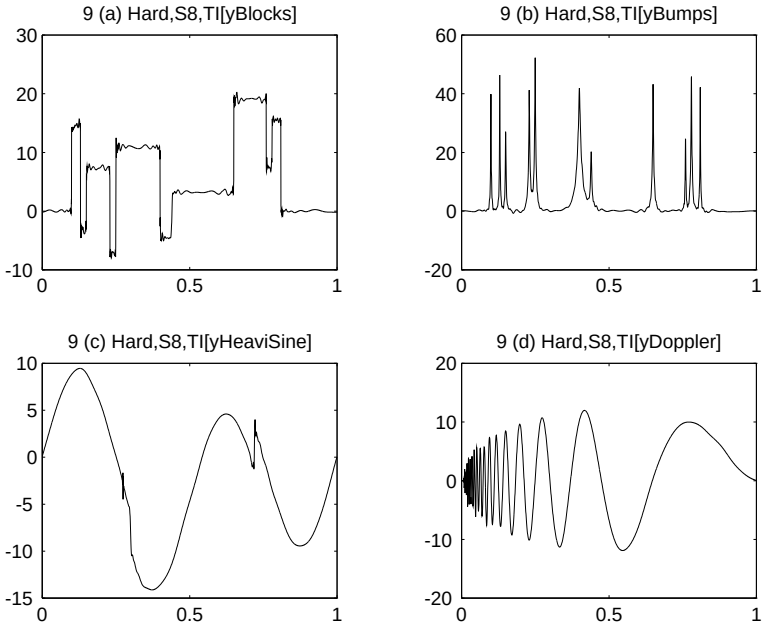
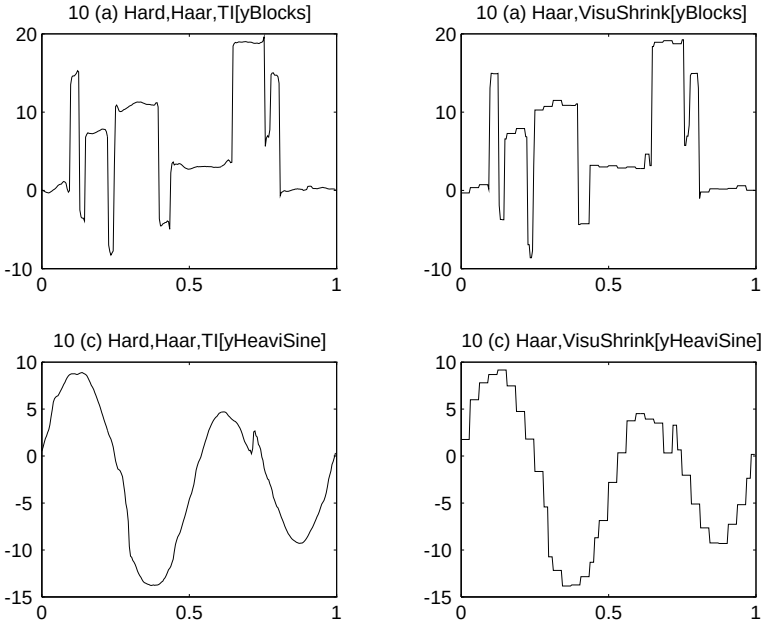


Figure 9 shows this to be the case; it illustrates fully translation-invariant De-Noising based on Hard Thresholding in the Symmlet 8 Domain. Closeups of the reconstruction have a very nice visual appearance.

2.2.3 Lower Sampling Rates. Hard thresholding looks so good when used in a translation-invariant fashion at high sampling rates that one might consider the use of TI methods at much lower rates than would normally be useful with wavelet shrinkage.

Figure 10 shows the result on a signal of length $n = 256$, which is equivalent to 1/8-th the sampling rate used earlier.



For comparison, the traditional de-noising result is indicated.

2.2.4 Lower Thresholds. Of course, wavelet shrinkage has many variants, and the $\sqrt{2\log(n)}\sigma$ variant can be improved for various purposes. The universal $\sqrt{2\log(n)}\sigma$ threshold was designed (see [15, 10]) for the purpose of suppressing noise-induced spikes which spoil the smoothness of reconstructions. However, if one wants only to measure performance by mean-squared error, then lower thresholds are better [15, 16].

Unfortunately, when used in conjunction with translation-invariant de-noising, such lower thresholds result in a very large number of noise spikes, apparently much larger than in the non-invariant case.

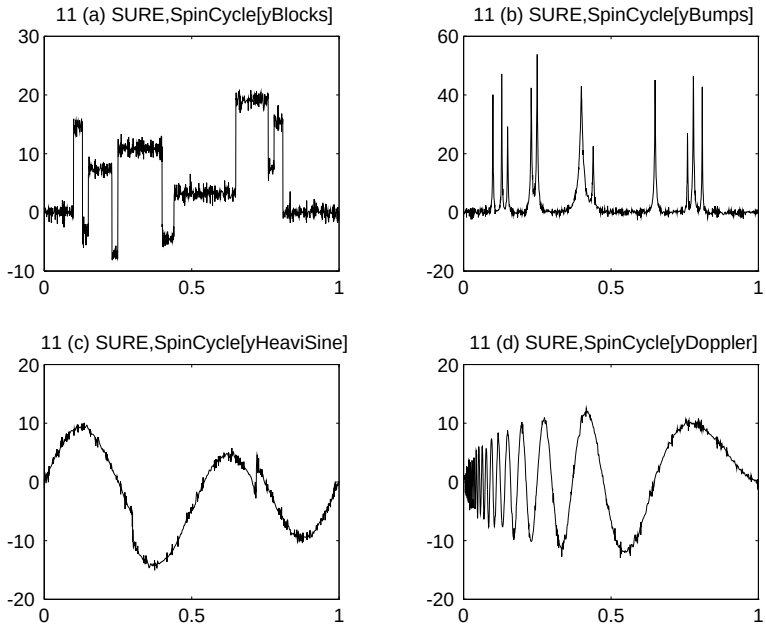


Figure 11 shows the result of Fully-TI de-noising with Symmlet 8 wavelets where the SURE thresholding rules of Donoho and Johnstone [16] has been used. The reconstruction contains many more noise spikes than when a non-invariant scheme is applied.

2.3 Quantitative Measures

How do these methods perform in quantitative terms? We summarize numerical performance:

Table 1. RMSE, S8 Wavelet De-Noising				
	Blocks	Bumps	HeaviSine	Doppler
Traditional	42.6868	45.4356	13.1158	24.2214
Cycle Spin	40.2826	40.3183	12.8068	21.4732
Fully TI	38.2768	39.523	12.9204	20.6080

In general, fully translation-invariant methods achieve better RMSE than either traditional de-noising or cycle-spinning over a range of 16 lags. The quantitative performance benefit is in the range of 10-20%. This relatively modest figure is due to the fact that a large gain is being made but only in a relatively small subset of the time domain.

Table 2. RMSE, Haar Wavelet De-Noising				
	Blocks	Bumps	HeaviSine	Doppler
Soft, Haar	29.7502	55.6212	27.5415	49.2407
Soft, Haar TI	21.8009	39.3873	10.1692	33.2107
Hard, Haar	12.8303	31.8518	19.5161	33.4112
Hard, Haar TI	7.73059	17.947	8.22684	17.6169

In general, hard thresholding behaves quite well, outperforming soft thresholding, with either wavelet. Ordinary Hard Thresholding with Haar wavelets also would have relatively good RMSE, but it would have poor visual performance. Hard Thresholding and Translation Invariance combined give both good visual characteristics and good quantitative characteristics.

2.4 Cosine Packet Domain De-Noising

The general Spin-Cycle approach (1) is of course not limited to use with wavelet transforms. To illustrate this point, we consider its use with transform De-Noising based on adaptive time-frequency bases using cosine packets. For reasons of space, we cannot describe here in detail all the ingredients of adaptive cosine packet bases; the interested reader may consult [7, 24]. The key ideas underlying CP De-Noising are as follows; see also [17]. We have a family of orthogonal bases, each one a segmented discrete cosine basis based on recursive dyadic segmentation of the interval. [17] develops a "SURE" functional for evaluating the suitableness of a given basis for De-Noising; this functional has an additive structure that enables us to use the Coifman-Wickerhauser Best-Basis algorithm. As a result, we can find a best basis ($\equiv$ best partition) in $O(n \log_2(n))$ time. We then can apply either hard or soft thresholding in the chosen basis, using a threshold set, as in [17], at $\sqrt{2 \log_e(n \log_2(n))}\sigma$.

Figure 12 shows part of 2 signals – *MishMash* and *QuadChirp* – and noisy versions of these signals. The signals are chosen to have a time-varying frequency content [17].

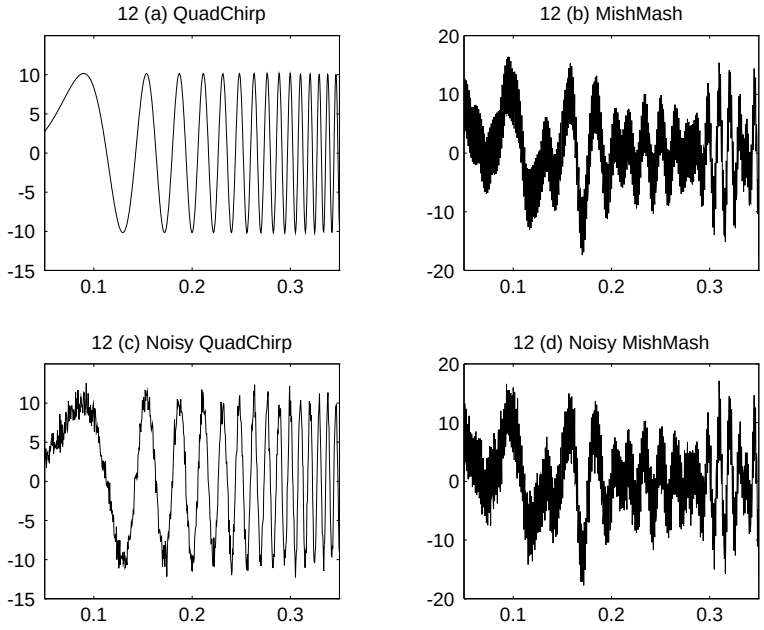
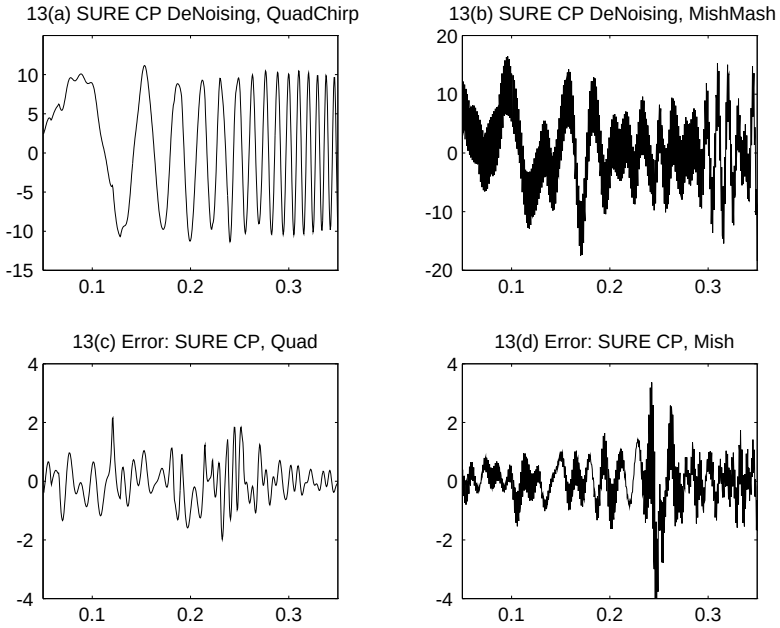


Figure 13 shows Cosine Packet Shrinkage of the two datasets, and the reconstruction errors suffered by this approach. A key point: the largest errors occur near the segmentation points (here, one occurs near the dyadic point 1/4). If the signals are interpreted as acoustic signals, the result will be "click" sounds at the segmentation points.



For comparison, Figure 14 shows Cycle-Spin CP Shrinkage of the 2 signals, where shifts in the range $H_{16} = \{h : 0 \leq h < 16\}$ are used. The reconstruction errors are smaller, particularly in the vicinity of the segmentation point 1/4.

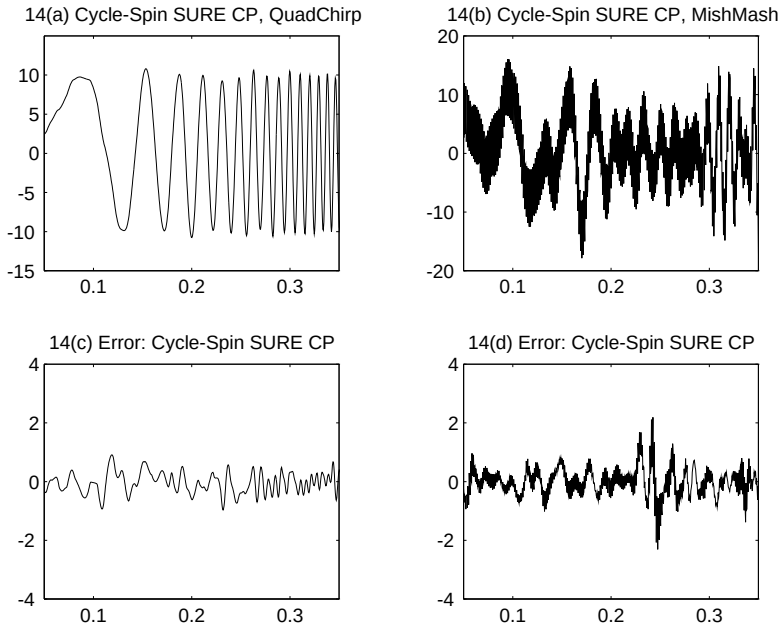


Table 3 shows a comparison of Root Mean Squared Errors for traditional and Cycle-Spin approaches. Evidently, Cycle-Spinning reduces the error by more than a third on the root-mean-square scale.

Table 3. RMSE, Cosine Packet De-Noising		
	QuadChirp	MishMash
Traditional	33.7773	52.0380
Cycle Spin	19.2762	30.9232

2.5 Fourier Domain De-Noising

We now turn to an example of *Modulation-Invariant* De-Noising. Our example will be based in the Fourier domain, so that we may equivalently think of the method as translation-invariant de-noising in the frequency domain.

Figure 15 shows two signals, *RatSine*, *IrratSine*, corresponding to sinusoids with/without exact periodicity for the signal length in question. In addition, noisy versions are provided for the two signals.

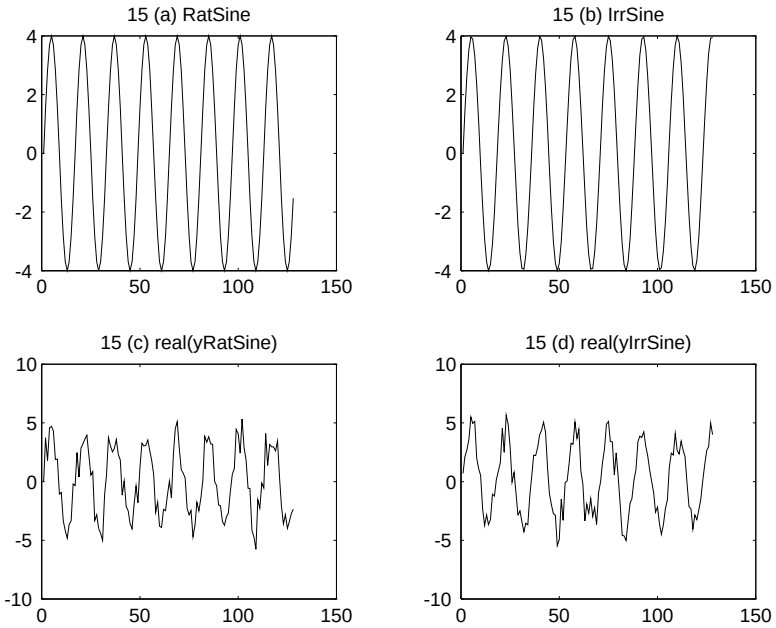
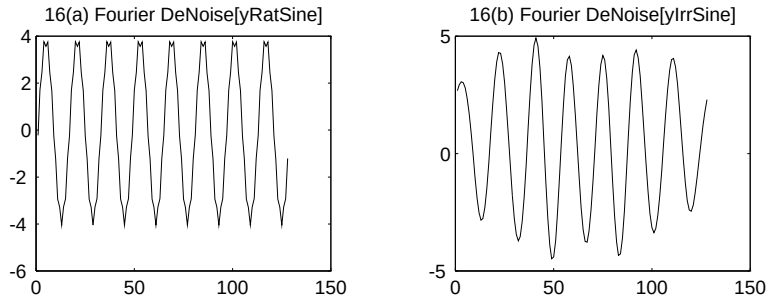


Figure 16 shows the results of Fourier-domain Shrinkage



For comparison, Figure 17 shows the results of Modulation-Based Cycle-Spinning of Fourier Shrinkage for the two noisy signals; with modulates of the form $\xi_h = 2\pi h/(16 \cdot n)$: $0 \leq h < 16$.

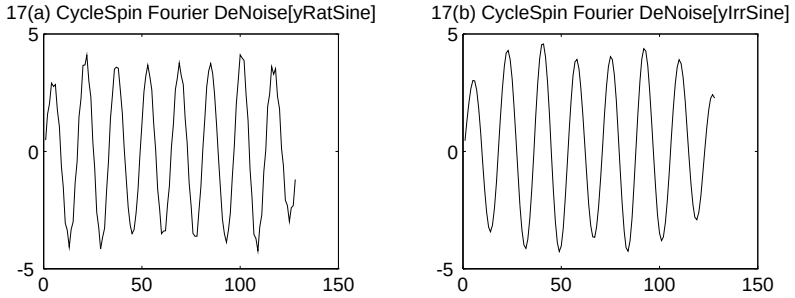


Table 4 shows a comparison of Root Mean Square errors for traditional and Cycle-Spin approaches. Evidently, Cycle-Spinning reduces the error close to 40% in the worst case, while increasing the error in the best case, so that the errors depend much more weakly on the precise value of the frequency underlying the signal.

Table 4. RMSE, Frequency De-Noising		
	RatSine	IrratSine
Traditional	1.10767	8.24615
Cycle Spin	4.469	5.15075

3 Software

We now briefly discuss the tools with which the above experiments where conducted.

3.1 Overview

This paper has been written following the discipline of *Reproducible Research* described in [1], appearing elsewhere in this volume. Every figure and table included here was produced automatically by a software system WaveLab , which is available to the reader over the Internet, via either a World-Wide-Web browser such as Mosaic or Netscape, or via anonymous FTP (URL: `http://playfair.stanford.edu/ wavelab`). WaveLab consists of more than 700 Matlab `.m` files which perform standard wavelet, wavelet packet, cosine packet, and time-frequency analyses. In addition, it contains code which reproduces the figures in published articles generated by several researchers at Stanford.

As a result of adhering to this discipline, the directory `WaveLab/Papers/SpinCycle` contains Matlab code to reproduce the figures of this article. It also contains a directory `WaveLab/Stationary`, which implements the basic translation-invariant wavelet transform tools. These tools implement our fast $O(n \log_2(n))$ translation-invariant denoising algorithm, as well as other tools.

A side effect of adhering to this discipline: there is a special rationale for including figures in this article. Our publication of a figure is not just the publication of a specific computational result; it is the publication of an algorithm, of datasets, and of a series of scripts which invoke the algorithm and datasets. The figures you see in this article are only the visible result of this publication process. The present article serves not only to announce the possibility and benefits of translation-invariant denoising, but also to make available, worldwide and electronically, the specific computational tools which are required.

In particular, as the fully translation-invariant wavelet De-Noising requires more than just programming – there is a mathematical idea as well – the underlying computational tools will be of interest for reasons besides De-Noising.

3.2 Scripts Reproducing Figures

The directory `WaveLab/Papers/SpinCycle` contains scripts for reproducing the figures of this article. Each script is a Matlab `.m` file. It contains, at the top of the file, a series of comment lines (a Matlab "help header") indicating what the file does. For example, the help header of `cspinf05.m` contains the following text

```
% cspinf05: Fully Translation-Invariant S8 Wavelet Shrinkage
%
% Here we test fully translation-invariant denoising, using
% the Translation-Invariant Transform.
% We apply a specific thresholding rule to the
% four noisy signals depicted in Figure 2.
%
% The procedure for DeNoising:
%       1.  Translation-Invariant Transform to Wavelet Domain,
%         Using Nearly-Symmetric Wavelet with 8 vanishing moments.
```

```
%       2.  Apply a soft thresholding nonlinearity, with threshold set to
%           sqrt{2 log(n)}
%       3.  Transform back to the signal domain.
%
%  Remark: run SpinCycleInit and cspinf0[12] before this.
%
```

Note that the first line contains a brief title for the rest of the file; this is called the H1 line in MATLAB parlance and the Matlab command `lookfor` allows the user to search all H1 Lines in the system for text matching a certain keyword. A listing of all the H1 Lines in the directory:

```
% SpinCycleInit: setup all global datastructures for SpinCycle
% PrintAllCSpinFigs: Generate all Encapsulated Postscript for Article

function s = cyclespin(x,k)
function recon = FourierDeNoise(y)
function y = Modulate(x,freq)
function coef = WPDeNoise_TI(basis,y,qmf)

% cspinf01: Four Spatially Inhomogeneous Signals
% cspinf02: Noisy Versions of Four Signals
% cspinf03: Ordinary S8 Wavelet Shrinkage
% cspinf04: Cycle Spinning S8 Wavelet Shrinkage
% cspinf05: Fully Translation-Invariant S8 Wavelet Shrinkage
% cspinf06: Ordinary Haar Wavelet Shrinkage
% cspinf07: Fully TI Haar Wavelet Shrinkage
% cspinf08: Closeup of Fully TI Haar Wavelet Shrinkage
% cspinf09: Fully TI, S8, Hard-Thresholding Shrinkage
% cspinf10: TI, Hard, Haar Shrinkage, Small Sample Size
% cspinf11: Cycle Spinning SURE Threshold Wavelet Shrinkage
% cspinf12: Two examples for time-frequency de-noising
% cspinf13: Ordinary Non-Spinning CP De-Noising
% cspinf14: CycleSpinning CP De-Noising
% cspinf15: Two Sinusoids for Fourier Denoising
% cspinf16: Non-Spinning Fourier De-Noising.
% cspinf17: Cycle-Spinning Fourier De-Noising.
% cspinf18: TI Table of Blocks; Haar Wavelet
% cspinf19: Stat Table of Blocks (= Unscrambling of TI Table)
% cspinf20: Stat Table of Noisy Blocks
% cspinf21: Thresholding Stat Table of Noisy Blocks
```

The reader will recognize the captions of the individual figures seen so far, and a few figures still to appear, as well as a number of computational tools which are invoked by the figure-generating tools.

3.3 Underlying Algorithms

The interested reader can easily inspect the scripts in the SpinCycle directory to see how most of the computations are done; in general, we compute Cycle-Spinning estimates by applying the brute force definitions (1), (2), so no fancy term like "algorithm" needs to be invoked. However, our approach to one special case - wavelet shrinkage averaged across all n-circulant shifts - is clever enough to warrant specific "data structures" and "algorithms".

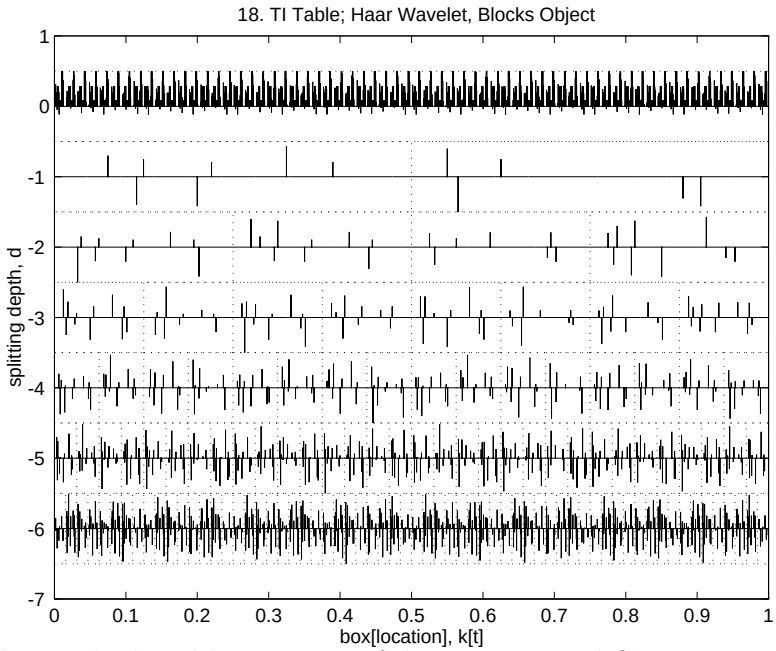
3.3.1 Fully TI De-Noising. A complete statement of the algorithm for $T_{\eta,S8}$ is as follows. We assume that variables `ncol` and `t` have been appropriately initialized, and then execute the code

```
CQF = MakeONFilter('Symmlet',8);                % Symmlet 8 Filter
TITable = FWT_TI(y,3,CQF);                      % Fast TI Transform
for j=2:ncol,
 TITable(:,j) = SoftThresh(TITable(:,j), t);    % Threshold each column
end
yDeNoise = IWT_TI(TITable,CQF);                 % Invert TI Transform
```

The key computations here are done by `FWT_TI` and `IWT_TI` which rapidly go from signal domain to *TI Table* and back. The remainder of the code is routine.

3.3.2 TI Table. We briefly describe the key data structure. Let x be a vector with dyadic length $n = 2^J$, and let $\tau(x)$ be the corresponding TI Table. This has three key properties: (i) for any integer h, the wavelet coefficients of the circulant shift $W[S_h x]$ are contained in the table $\tau(x)$; (ii) the full TI Table for x can be computed in order $n \log_2(n)$ time; and (iii) the extraction of the wavelet transform of any circulant shift, $W[S_h x]$, takes order n time.

The TI Table is an n by D array, where $0 \le D \le \log_2(n)$. The table formally has the same structure as a wavelet packet table or cosine packet table. The d-th column has n entries partitioned into 2^d "boxes", each box having $n/2^d$ entries. The interpretation of the "boxes": they correspond to the 2^d different collections of wavelet coefficients that can occur in the wavelet expansion at level $J - d$ under different shifts of the input vector x.



The TI Table is calculated by a series of decimation and filtering operations, just like the wavelet transform, only an extra-element – a circulant shift – is thrown in. Let G and H stand for the usual downsampling high pass and low pass operations of wavelet theory; let S_h again stand for circulant shift by h; and set $\beta_{J,0} = s$. Then put

$$\alpha_{J-1,0} = GS_0\beta_{J,0}; \quad \alpha_{J-1,1} = GS_1\beta_{J,0}$$

and

$$\beta_{J-1,0} = HS_0\beta_{J,0}; \quad \beta_{J-1,1} = HS_1\beta_{J,0}.$$

Continue recursively:

$$\alpha_{j,2k} = GS_0\beta_{j+1,k}; \quad \alpha_{j,2k+1} = GS_1\beta_{j+1,k},$$

and

$$\beta_{j,2k} = HS_0\beta_{j+1,k}; \quad \beta_{j,2k+1} = HS_1\beta_{j+1,k},$$

and so on. To fill out the TI Table, place, in box k of column d, the vector $\alpha_{J-d,k}$; in an extra column, place all the $\beta_{j,k}$'s computed at the final stage (we use the zero-th column for this).

To state formally the invariance property possessed by TI Tables, suppose we let $\tau(x)$ be the TI Table corresponding to x, and let $\tau(S_h x)$ be the table corresponding to the circulant shift $S_h x$. Then, for each shift $h \in \{0, \ldots, n-1\}$ there is a permutation of matrix entries Π_h so that

$$\Pi_h \tau(x) = \tau(S_h x). \tag{3}$$

The traditional wavelet transform consists of the unshifted data

$$W[x] = (\beta_{j_0,0}, \alpha_{j_0,0}, \alpha_{j_0+1,0}, \ldots, \alpha_{J-1,0}).$$

Since the right side of (3), $\tau(S_h x)$ contains all the coefficients of the wavelet transform of the shift $S_h x$, (3) says that those same wavelet coefficients are already present in the TI Table $\tau(x)$; they just need to be unpacked correctly.

More precisely, the wavelet transform of a circulant shift of x is realized as

$$W[S_h x] = (\beta_{j_0,k_{j_0}}, \alpha_{j_0,k_{j_0}}, \alpha_{j_0+1,k_{j_0+1}}, \ldots, \alpha_{J-1,k_{J-1}})$$

for an appropriate sequence $(k_{j_0}, k_{j_0+1}, \ldots, k_{J-1})$, where each $k_j = 2k_{j+1} + b_j$ and $b_j \in \{0, 1\}$. The bits b_j encode the shift h in a special binary notation.

To extract the wavelet transform of a certain circulant shift from a TI Table, one needs therefore to specify the shift, in terms of a special coding of h. In WaveLab this is specified by setting up a *path* data structure, which is a representation of a path through a rooted complete tree, starting at the root, and ending in a leaf node. Envisioning the root of such a tree as placed at the top, and the leaves at the bottom, the path data structure contains a series of flags saying to go "left" or "right" at each level of descent from the root. This data structure therefore specifies the series of bits b_j needed to extract the required data from the structure.

Elaborating the above discussion with a few computational details will show explicitly that the TI Table can be calculated in order $n \log_2(n)$ time and one can extract from the TI table the n coefficients of any specific shift in order n time.

3.3.3 Inversion. How to go back from a TI Table to the original signal? The idea is to systematically average: start with $j = J - D$ and then for each k in the range $0 \le k < 2^j$, compute (with the help of the usual upsampling operators G^* and H^*)

$$\gamma_k = (S_0 G^* \beta_{j,2k} + S_{-1} G^* \beta_{j,2k+1})/2, \qquad \delta_k = (S_0 H^* \alpha_{j,2k} + S_{-1} H^* \alpha_{j,2k+1})/2,$$

and

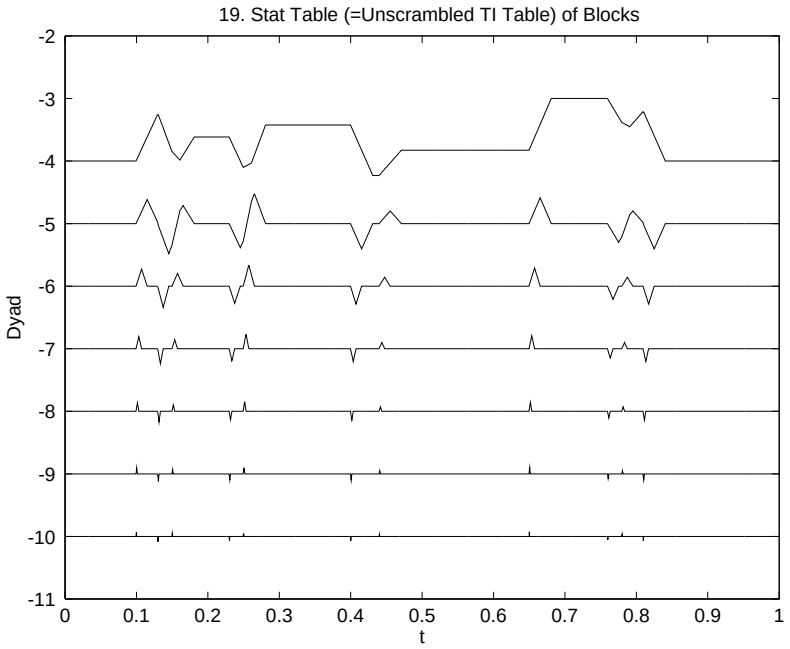
$$\beta_{j+1,k} = \gamma_k + \delta_k.$$

After exhausting all k at one level, set $j = j + 1$ and repeat. After reaching $j = J$, stop. Set $\tilde{s} = \beta_{J,0}$.

When applied to the TI Table generated from signal s, the result is $\tilde{s} = s$. When applied to a thresholded TI Table, the result is an average of all n reconstructions from all n circulant shifts. The fact that each γ_k and δ_k is an average of two possible reconstructions – one from an unshifted series, one from a shifted one – is responsible for this result.

This algorithm takes order n arithmetic at each level, and goes through order $\log_2(n)$ levels, so the whole algorithm takes order $n \log_2(n)$ work.

3.3.4 Stationary Wavelet Transform. The coefficients in the TI Table record information about the signal x in rather a scrambled fashion. Figure 18 shows the TI Table generated under WaveLab for the Blocks signal. Figure 19 shows a special "unscrambled" form of the same table, which we call the Stat Table.



The unscrambled form correlates directly with the underlying scale and spatial structure of the signal. It has an appearance similar to a number of displays which associate $\log_2(n)$ signals of length n each to a signal of length n: (i) the multi-resolution decomposition of Mallat, (ii) what Mallat calls the Undecimated Wavelet Transform, and (iii) what Nason and Silverman call, elsewhere in this volume, the Stationary Wavelet Transform. In fact, while the transform is visually similar to (i) in some ways, it is mathematically different. However, it is identical to both (ii) and (iii) except for some possible details of scaling.

In short the Stat Table $\sigma(x)$ is an n by D array which may be viewed as a collection of $D-1$ discrete time signals $\alpha_j(t)$, $j = j_0, \ldots, J$ and one extra signal $\beta_{j_0}(t)$. By the equivalence with (ii) and (iii) above, we are saying that these signals are of the form

$$\beta_{j_0}(t) = (E_{j_0} \star x)(t)$$

and

$$\alpha_j(t) = (F_j \star x)(t),$$

where E_{j_0} and F_j are discrete time filters of length n, and intrinsic bandwidth $\asymp n/2^j$ samples, and $\star$ denotes circulant convolution. The claim of equivalence is that F_j etc. are equivalent (when the wavelets are appropriately chosen) to the impulse responses of the j-th level of either Mallat's undecimated Wavelet Transform or Nason-Silverman's Stationary wavelet transform.

Owing to the equivalence of the Stationary Wavelet Transform and the unscrambled TI Transform, our algorithm for fully TI de-noising has an equivalent "unscrambled form": it amounts to *thresholding of the stationary wavelet transform.* To visualize this, we simply take the intermediate results of the algorithm described in subsection 3.3.1, and unscramble them.

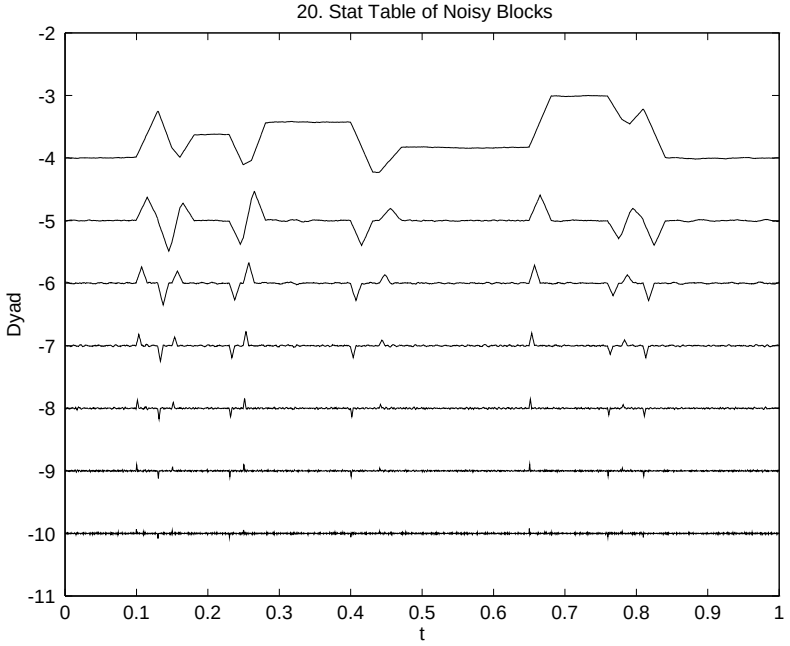
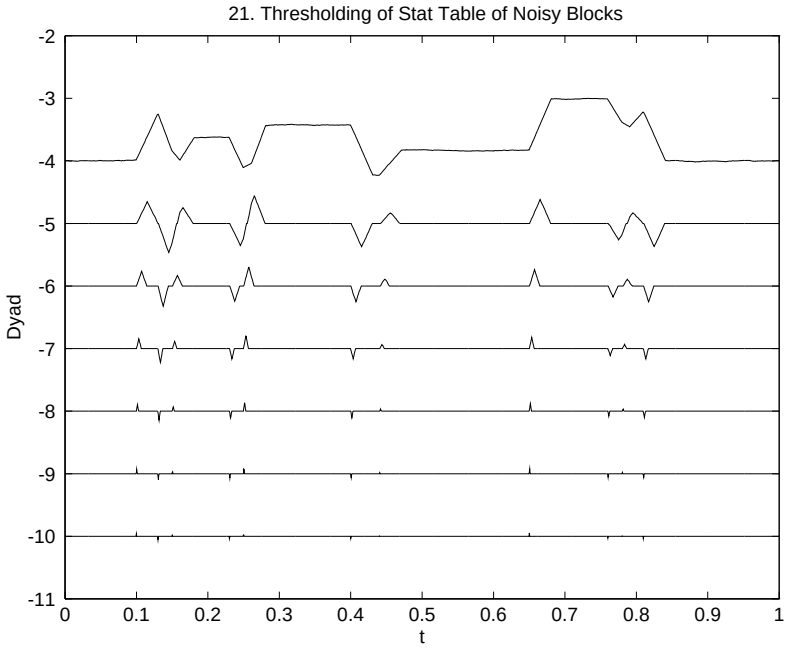


Figure 20 displays the Stat Table of a noisy HeaviSine signal, and Figure 21 displays the thresholded version. The result of inverse transforming this table has been seen already in figure 5.



3.4 Other Translation-Invariant De-Noising Algorithms

The specific algorithm for de-noising we have discussed here is not the only translation-invariant method that can be fashioned from these tools. For completeness, we mention two alternatives here.

3.4.1 Best Shift Algorithm. The transforms of the n different circulant shifts $W[S_h x]$ might be considered n transforms into n different orthogonal bases. Which basis is best?

Suppose that one can measure the quality of a basis by an additive functional

$$\mathcal{E}(h) = \sum_I e((W[S_h x])_I)$$

where $e(w)$ is some fixed function of a scalar w; the so-called Coifman-Wickerhauser entropy $e_{CW}(w) = -w^2 \log(w^2)$ is one example. Then it turns out that a fast algorithm can be developed for finding the optimum h. The algorithm is similar in many ways to the Coifman-Wickerhauser algorithm for best basis in Cosine Packet or Wavelet Packet Libraries [7]. The fast algorithm is implemented in WaveLab in two steps; first, `CalcShiftStat`, which builds a data structure filled with entropy numbers, and second, `BestShift`, which processes the data structure using a dynamic programming algorithm to optimize over all shifts. Both steps together, once the TI table is available, take only order n time; thus one can find and transform into the best shift basis in order $n \log_2(n)$ time.

From the point of view of Mean-Squared error of De-Noising, one could argue, as in [17, 12] that the best basis for De-Noising is one minimizing the expression

$$\mathcal{E}_\lambda(h) = \sum_I \min((v_I^h)^2, \lambda^2 \sigma^2)$$

where $(v_I^h) = W[S_h y]$ is the collection of wavelet coefficients in the h-th basis of the noisy signal, and $\lambda^2 = 2 \log_e(n \log_2(n))$. This is an additive "entropy"-type functional of the wavelet coefficients. The fast algorithm can be used to obtain a best-shift-basis in order $n \log_2(n)$ time. Applying hard or soft thresholding in that basis, and then reconstructing, gives a translation invariant de-noising.

Unfortunately, we believe that it is usually the case that a single best shift is not suitable, as the underlying signal will often contain features interfering with each other, each "wanting" a different shift. We have implemented Best-Shift De-Noising but have not employed it heavily. We have mentioned it here because some readers may find the idea interesting, and because we have included the relevant tools in the WaveLab distribution.

3.4.2 Overcomplete Representation. The operation of transforming an n-long signal x into an n by $\log_2(n)$ TI Table $\tau(x)$ is equivalently calculating the collection of all inner products $\langle x, \phi_I \rangle$ of the signal with a *Dictionary* of $n \log_2(n)$ different waveforms (ϕ_I). These waveforms are the basic orthogonal wavelets in a single orthogonal basis, together with all their circulant shifts. Viewing temporarily $\tau(x)$ as a vector with $n \log_2(n)$ entries, we can symbolize this as $\tau(x) = \Phi^T x$ where Φ is an n by $n \log_2(n)$ matrix.

From this point of view, it is interesting to consider the problem of translation-invariant representation in this dictionary, that is, to find, for a given signal s, a collection of coefficients α satisfying $\sum_I \alpha_I \phi_I = s$. In terms of the matrix Φ this can be written as

$$\Phi \alpha = s$$

As the matrix Φ has fewer rows (n) than columns $(n \log_2(n))$, the problem is underconstrained: there are multiple solutions.

The routine `IWT_TI` referred to above offers a specific solution, based on the method of frames; it essentially calculates the solution $\alpha^\dagger$ of minimum ℓ^2 norm. Equivalently, it applies the generalized inverse $\Phi^\dagger = (\Phi \Phi^T)^{-1} \Phi^T$ to s: $\alpha^\dagger = \Phi^\dagger s$. The result is an average of

all possible reconstructions in all n circulant-shift bases, as revealed by careful examination of the matrix $\Phi^\dagger$.

Another solution has been studied by Chen and Donoho [4], which is to find the element $\alpha^\star$ of minimum ℓ^1 norm. This solution is a nonlinear function of the entries in s and can be obtained by linear programming. The result is a linear combination of at most n waveforms taken from the $n\log_2(n)$ element dictionary; it is therefore more sparse than the method of frames approach used implicitly in this paper. The paper [4] shows how to calculate this sparse solution rapidly; the fast algorithms `FWT_TI` and `IWT_TI` developed for this paper are key ingredients of this task. The solution is translation-invariant since the dictionary is translation-invariant.

[4] also proposes a translation-invariant de-noising method. Assuming noisy data $y_i = f(t_i) + \sigma z_i$ with $z_i \sim_{iid} N(0,1)$. The idea is to solve the penalized problem

$$\min_a \|y - \Phi a\|_2^2 + \lambda \|a\|_1$$

with $\lambda = \sqrt{2\log_e(n\log_2(n))}\sigma$. The solution then furnishes a linear combination of at most n wavelets taken from the dictionary. The constrained quadratic programming algorithm they use is based on iterative application of Φ and Φ^T, that is to say it relies on the same tools `FWT_TI` and `IWT_TI`.

4 Analysis

While the purpose of this paper is *not* a theoretical analysis of the performance benefits of translation-invariance, we think a few brief comments might be helpful for understanding some phenomena which are visually apparent in the earlier figures.

4.1 Improvement in Approximation

With (x_t) a discrete-time signal and $P_j x = \sum_k \langle x, \phi_{j,k}\rangle \phi_{j,k}$ the approximation at scale j by the Haar subspace V_j, and

$$\overline{P}_j = Ave_{h\in H_n}\{S_{-h} P_j S_h\}$$

the translation-invariant approximation, we have two nice properties.

First, $P_j x$ is a piecewise constant function with steps of width $n/2^j$, while $\overline{P}_j x$ has steps of width 1, which can be considerably narrower.

Second, if $x_t = a + bt$ is a straight line, then $P_j x$ is a stairstep sequence approximating the line, while $\overline{P}_j x$ is, except at the end points, a straight line of the same slope as x itself.

To understand these properties, it is convenient to shift viewpoint, and analyze a continuous-time version of "all shift" cycle-spinning. Suppose now that $f(t)$ is a function of t in the interval $[0,1)$, viewed as a circle, and define

$$P_j^* = Ave_{h\in[0,1]}\{S^\circ_{-h}\, P^\circ_j\, S^\circ_h\}$$

where $P^\circ_j = \sum_k \langle x, \phi_{j,k}\rangle \phi_{j,k}$, with $\phi_{j,k}(t) = 2^{j/2} 1_{[k/2^j,(k+1)/2^j)}$ and S°_h refers to the circulant shift on the continuum circle.

Then one has

$$P_j^* = \int_0^1 \Phi_j(t,s) f(s) ds$$

where the kernel Φ_j is in fact convolutional and of bandwidth $\sim 2^{-j}$:

$$\Phi_j(t,s) = 2^j \phi(2^j(t-s))$$

with triangular kernel

$$\phi(u) = (1-|u|)_+.$$

From this representation, three properties are easy to verify.

1. *Continuity.* If $f \in L^\infty[0,1)$, then $(P_j^* f)(t)$ is continuous in t. In contrast, $P_j^\circ f$ is in general discontinuous.

2. *Preserving Lines.* If $f(t) = a+bt$, then $(P_j^* f)(t) = f(t)$ identically on $t \in [2^{-j+1}, 1-2^{-j+1}]$. In contrast $P_j^\circ f$ is in general a stairstep, not a line.

3. *Approximation Order.* If $f \in C^\alpha[0,1)$, then

$$|P_j^* f(t) - f(t)| \le Const. \cdot \|f\|_\alpha \cdot (2^{-j})^\alpha \tag{4}$$

holds for all α in $[0,2)$. For example, set $f_0(t) = \sin(2\pi t)$; then $|P_j^* f_0(t) - f_0(t)| \le C \cdot 2^{-2j}$. In contrast, the type of relation (4) holds for the classical operator P_j° only for $\alpha \in [0,1]$. Moreover, $|P_j^\circ f(t) - f(t)| \asymp 2^{-j}$ even for smooth f like f_0.

In short, the Haar-TI approximation spaces gain in continuity and in approximation order over classical Haar approximation spaces.

4.2 Suppression of Gibbs Phenomena

Figure 8 above showed that Translation-Invariant Haar De-Noising is essentially non-oscillatory. That is, when applied to a discontinuous Heaviside, the Haar-TI de-noising does not overshoot or undershoot the target. In fact, it can be proved that in the presence of normally distributed noise, the expectation of Haar-TI de-noising is non-oscillatory, when soft thresholding is employed.

To show this, suppose we have normally distributed data $W \sim N(\mu, \sigma)$ and define the bias function

$$B(\mu; t, \sigma) = E\eta_t(W) - \mu.$$

We note that B is an odd continuous monotone function of μ.

We use this to analyze components of the reconstruction. To simplify the presentation, we assume a continuous cycle-spinning model as in the previous subsection. Thus, we assume a white noise model and an estimator

$$\hat{f}^*(t) = \int_0^1 \Phi_{j_0}(t,s) y(s) ds + \sum_{j \ge j_0} \int_0^1 \eta(\hat{\alpha}_j(s)) \psi_j(t-s)$$

where η is one of the soft thresholding nonlinearities (η_t), $y(s)ds$ stands for $f(s)ds + W(ds)$ with W a Wiener process, $\psi_j(u) = 2^{j/2}\psi(2^j u)$, and

$$\hat{\alpha}_j(s) = \int_0^1 \psi_j(u-s)y(u)du.$$

We analyze the expectation of this estimator when f is a Heaviside $f(t) = H(t-t_0)$ jumping at a point t_0 well inside the interval $[2^{-j_0+1}, 1-2^{-j_0+1})$. We need the expectation

$$\alpha_j(s) = E\hat{\alpha}_j(s) = \int_0^1 \psi_j(u-s)f(u)du.$$

Using this we can write

$$\hat{f}^*(t) - f(t) = \int \Phi_{j_0}(t,s)y(s)ds - \int \Phi_{j_0}(t,s)f(s)ds + \sum_{j\geq j_0} \int_0^1 (\eta(\hat{\alpha}_j(s)) - \alpha_j(s))\psi_j(t-s),$$

leading to

$$E\hat{f}^*(t) - f(t) = \sum_{j\geq j_0} \beta_j(t)$$

where

$$\beta_j(t) = \int_0^1 B(\alpha_j(u))\psi_j(t-u)du.$$

and B is the bias function corresponding to the given nonlinearity.

The key analytic property we need is the relation:

$$\beta_j(t) \leq 0, \quad t \in [t_0, 1-2^{-j_0+1}), \qquad \beta_j(t) \geq 0, \quad t \in (2^{-j_0+1}, t_0] \tag{5}$$

which follows from elementary manipulations using the Haar wavelet, from the triangular profile of α_j, and from the odd, monotone nature of $B(\mu)$.

The non-oscillatory nature of the expectation follows from (5), because it says that the expectation is greater than the Heaviside on $t < t_0$ and less than the Heaviside on $t > t_0$. There is no alternating overshoot and undershoot.

Note that when the threshold $\sqrt{2\log_e(n)}\sigma$ is used, the bias dominates the variance, and so the properties of the expectation describe the dominant visual features of the estimate. Hence, we have a full explanation for the behavior of Figure 8.

4.3 Improvement in De-Noising performance

One can show, combining ideas of the last two subsections, that the mean-squared-error of TI-Haar Denoising tends to zero at a rate faster than Haar De-Noising in an interesting range of cases. Suppose that we have observations $y_i = f(t_i) + \sigma z_i$, where the z_i are i.i.d. $N(0,1)$. Suppose that the sampling points are equispaced, and that $f \in C^\alpha$ with $\alpha \in (1,2)$. Then ordinary Haar De-Noising will obey $En^{-1}\|\hat{f} - f\|_{\ell^2}^2 \asymp (\frac{\log(n)}{n})^{2/3}$ while TI-Haar De-Noising will obey $En^{-1}\|\hat{f}^* - f\|_{\ell^2}^2 \asymp (\frac{\log(n)}{n})^{2\alpha/2\alpha+1}$. As $\alpha > 1$ this is an improvement at the level of rates.

5 Discussion

5.1 Relation to Other Work

There is nothing new under the sun, not even in a field like wavelets.

Although Coifman had been using SpinCycle ideas before that time, Donoho recalls first hearing of it from Coifman in mid-1993. Stéphane Mallat explained to Donoho in January 1994 that Mallat had been using thresholding on the non-decimating wavelet transform and had found significant visual benefits over comparable thresholding in the decimating transform. As mentioned above, when used with the right wavelets, thresholding the non-decimating transform is mathematically equivalent to Cycle Spinning with all n circulant shifts.

Guy Nason and Bernard Silverman gave talks throughout 1994 on the Stationary Wavelet Transform. As mentioned above, the stationary wavelet transform can be obtained by a permutation of the entries in the TI Table. In their paper in this volume [21], they mention the possibility of applications to De-Noising. This paper may be taken as, in part, an illustration of their comment.

The article of Simoncelli et al. [23] points out that E. Adelson wrote in the early 1980's a patent which, broadly construed, includes the application of threshold-like nonlinearities to "full density laplacian pyramids", which is, broadly speaking the same as thresholding in a non-decimated wavelet transform.

The key algorithmic idea underlying the fast application of SpinCycle ideas in a wavelet setting is already present in Beylkin's work on fast matrix operations [3] which already says that it is possible to compute the wavelet coefficients of all n circulant shifts of a vector in $O(n \log_2(n))$ time.

5.2 Acknowledgements

Coifman's research is supported by ARPA; he would like to acknowledge collaborations with Roland Juglielmi, Naoki Saito, and Leonel Woog.

Donoho would like to thank Iain Johnstone for encouragement and Jon Buckheit and Shaobing Chen for their work on the software. In particular, Shaobing Chen figured out how to correctly "unscramble" TI Tables into Stat Tables, a task which had eluded Donoho for some time.

Computing equipment support for Donoho, and Graduate Support for Jon Buckheit and Shaobing Chen was provided by NSF DMS 92-09130, by the NASA Astrophysics Data Program, and by other sponsors. Part of this work was also supported by an ONR Contract with Statistical Sciences, Inc.

It is a pleasure to acknowledge conversations with Guy Nason, Stéphane Mallat, and Bernard Silverman, which have pointed out the related work described above.

The Organizers of the XV^{eme} Rencontres Franco-Belges deserve special thanks for their efforts in staging a conference on *Wavelets and Statistics*, which we think will stimulate both statisticians and waveleticians. Anestis Antoniadis is also to be congratulated for having inspired a very successful published record of the state-of-the-art.

References

[1]Buckheit, J. and Donoho, D.L. (1995) WaveLab and Reproducible Research. in this Volume.

[2]Berger, J., Coifman, R.R. and Goldberg, M. (1994) Removing Noise from Music using Local Trigonometric Bases and Wavelet Packets. *J. Audio Eng. Soc.* **42**.

[3]Beylkin, G. (1992) On the Representation of Operators in Bases of Compactly Supported Wavelets. *SIAM J. Numer. Anal.*, **29** 1716-1740.

[4]Chen, S. and Donoho, D.L. (1995) Atomic Decomposition by Basis Pursuit. Technical Report, Department of Statistics, Stanford University.

[5]Coifman, R.R. and Meyer, Y. (1991) "Remarques sur l'analyse de Fourier à fenêtre," *Comptes Rendus Acad. Sci. Paris* (A) **312** 259-261.

[6]Coifman, R.R., Meyer, Y. and Wickerhauser, M.V. (1992) "Wavelet analysis and signal processing," pp. 153–178 in *Wavelets and Their Applications*, M. B. Ruskai et al. (eds.), Jones and Bartlett, Boston.

[7]Coifman, R.R. and Wickerhauser, M.V. (1992) "Entropy-based algorithms for best-basis selection." *IEEE Trans. Info. Theory* **38** 713-718.

[8]Coifman, R.R. and Wickerhauser, M.V. (1993) "Wavelets and Adapted Waveform Analysis: A Toolkit for Signal Processing and Numerical Analysis." in *Different Perspectives on Wavelets*, in I. Daubechies, ed. pp. 119-153. Providence, RI: American Mathematical Society.

[9]Daubechies, I. (1992) *Ten Lectures on Wavelets*. Philadelphia: SIAM.

[10]Donoho, D.L. (1992) De-Noising via Soft Thresholding. To appear *IEEE Trans. Info. Thry*, May, 1995.

[11]Donoho, D.L. (1993) Unconditional bases are optimal bases for data compression and for statistical estimation, *Applied and Computational Harmonic Analysis*, **1**, 100-115.

[12]Donoho, D.L. (1993) Nonlinear Wavelet Methods for Recovery of Signals, Images, and Densities from noisy and incomplete data, in *Different Perspectives on Wavelets*, I. Daubechies, ed. pp.173-205. Providence, RI: American Mathematical Society.

[13]Donoho, D.L. (1993) Wavelet Shrinkage and W.V.D. – A Ten-Minute Tour, in *Progress in Wavelet Analysis and Applications.* Y. Meyer and S. Roques, eds. pp. 109-128. , Gif-sur-Yvette (France): Éditions Frontières.

[14]Donoho, D.L. (1994) On Minimum Entropy Segmentation, in *Wavelets: Theory, Algorithms and Applications.* C.K. Chui, L. Montefusco and L. Puccio, Eds. San Diego: Academic Press.

[15]Donoho, D.L. and Johnstone, I.M. (1994) Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, **81**, 425-455.

[16]Donoho, D.L. and Johnstone, I.M. (1992) Adapting to Unknown Smoothness via Wavelet Shrinkage. to appear *JASA*, 1995.

[17]Donoho, D.L. and Johnstone, I.M. (1994) Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes Rendus Acad. Sci. Paris A* **319**, 1317-1322.

[18]Donoho, D.L., I.M. Johnstone, G. Kerkyacharian and D. Picard (1995) Wavelet Shrinkage: Asymptopia. *J. Roy. Statist. Soc.* **B 57** 2, 301-369.

[19]Meyer, Y. (1993) *Wavelets: Algorithms and Applications* Philadelphia: SIAM, 1993.

[20]Mallat, S. and Zhang, S. (1993) Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing,* **41**, 3397-3415.

[21]Nason, Guy, and Silverman, B.W. (1995) The Stationary Wavelet Transform and some Statistical Applications. This Volume.

[22]Saito, N. (1994) *Feature Extraction using Local Discriminant Basis*. Yale Dissertation, December 1994.

[23]Simoncelli, E.P., Freeman. W.T., Adelson, E.H., and Heeger, D.J. (1992) Shiftable multiscale transforms. *IEEE Trans. Info. Theory*, **38**, pp. 587–607.

[24]Wickerhauser, M.V. (1994) *Adapted Wavelet Analysis, from Theory to Software*. AK Peters: Boston.