# THE WHAT, HOW, AND WHY OF WAVELET SHRINKAGE DENOISING

*Wavelet shrinkage denoising provides a new way to reduce noise in signals. The author demonstrates 1D and 2D examples, tests the performance of various ideal and practical Fourier- and wavelet-based denoising procedures, and makes recommendations for practitioners.*

Applied scientists and engineers who work with data obtained from the real world know that signals do not exist without noise. Under ideal conditions, this noise might decrease to such negligible levels that for all practical purposes, denoising is not necessary. Unfortunately, we usually must remove the noise corrupting a signal to recover that signal and proceed with further data analysis. However, should this noise removal take place in the original signal (time-space) domain or in a transform domain? If the latter, should we use the Fourier transform for the time-frequency domain or the wavelet transform for the time-scale domain?

Enthusiastic supporters have described the development of wavelet transforms as revolutionizing modern signal and image processing over the past two decades. Conservative observers, however, describe this new field as contributing additional useful tools to a growing toolbox of transforms.[1] A particular wavelet method called *wavelet shrinkage denoising* has caused its zealous advocates to claim that it "offers all that we might desire of a technique, from optimality to generality."[2] Inquiring skeptics, however, might be loath to accept these claims based on asymptotic theory without persuasive evidence from real-world experiments. Fortunately, a burgeoning literature is now addressing these concerns, leading to a more realistic appraisal of wavelet shrinkage denoising's utility. Let's examine this wavelet method through a couple of examples and experiments.

1521-9615/00/$10.00 © 2000 IEEE

CARL TASWELL
*Computational Toolsmiths*

## A simple explanation and a 1D example

Wavelet shrinkage denoising should not be confused with smoothing (despite the use by some authors of the term *smoothing* as a synonym for the term *denoising*). Whereas smoothing removes high frequencies and retains low ones, denoising attempts to remove whatever noise is present and retain whatever signal is present regardless of the signal's frequency content (for example, when we denoise noise-corrupted music, we want to preserve both the treble and the bass). Wavelet shrinkage denoising does involve shrinking (nonlinear soft thresholding) in the wavelet transform domain, and consists of three steps: a linear forward wavelet transform, a nonlinear shrinkage denoising, and a linear inverse wavelet transform. The nonlinear shrinking of coefficients in the transform domain distinguishes this procedure from entirely linear denoising methods. Furthermore, wavelet shrinkage denoising is considered a nonparametric method. Thus, it is distinct from parametric methods[3] in which we must estimate parameters for a particular model that must be assumed a priori. (For example, the most commonly cited parametric method uses least squares to estimate the parameters $a$ and $b$ in the model $y = ax + b$.)

Figure 1 displays a practical 1D example demon-

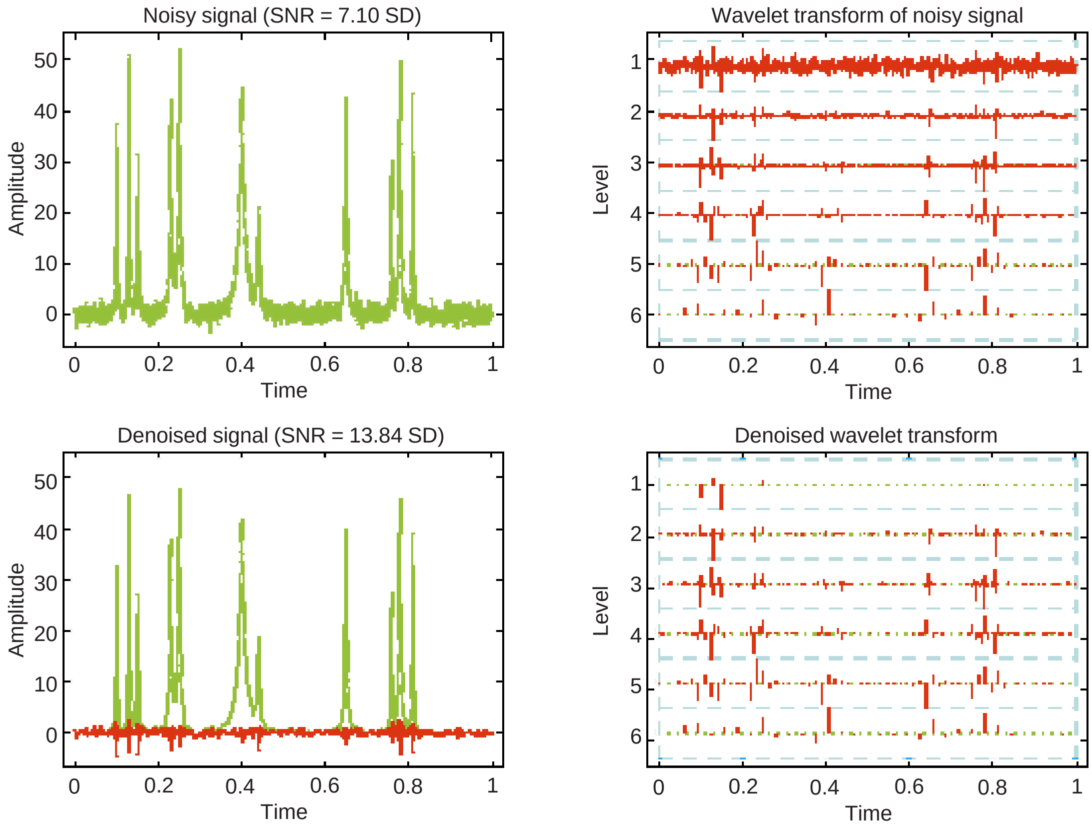


**Figure 1. Wavelet shrinkage denoising with 'SUR,' DROLA (16; 8), $n$ = 2,048, $L$ = 6.**

strating the three steps of wavelet shrinkage denoising with plots of a known test signal with added noise, the wavelet transform, the denoised wavelet transform, and the denoised signal estimate. In the latter, the green curve is the estimate and the red curve is the difference between this estimate and the original true signal without noise. WavBox software (version 4.5b3)[4] generated all results and figures reported here, using filters from the systematized collection of Daubechies wavelets,[5] in particular, the Daubechies Real Orthogonal Least Asymmetric (DROLA) filters. (See the "Code for the 1D example" sidebar for further details.)

### A more precise definition

Assume that the observed data

$$X(t) = S(t) + N(t)$$

contains the true signal $S(t)$ with additive noise $N(t)$ as functions in time $t$ to be sampled. Let $\mathcal{W}(\cdot)$ and $\mathcal{W}^{-1}(\cdot)$ denote the forward and inverse wavelet transform operators. Let $\mathcal{D}(\cdot, \lambda)$ denote the denoising operator with soft threshold $\lambda$. We intend to wavelet shrinkage denoise $X(t)$ to recover $\hat{S}(t)$ as an estimate of $S(t)$. Then the three steps

$$Y = \mathcal{W}(X)$$
$$Z = \mathcal{D}(Y, \lambda)$$
$$\hat{S} = \mathcal{W}^{-1}(Z)$$

summarize the procedure. Of course, this summary of principles does not reveal the details involving implementation of the operators $\mathcal{W}$ or $\mathcal{D}$, or selection of the threshold $\lambda$.

Let's focus on $\lambda$ and $\mathcal{D}$. Given threshold $\lambda$ for data $U$ (in any arbitrary domain, signal, transform, or otherwise), the rule

$$\mathcal{D}(U, \lambda) \equiv \text{sgn}(U)\max(0, |U| - \lambda)$$

defines nonlinear soft thresholding. The operator $\mathcal{D}$ nulls all values of $U$ for which $|U| = \lambda$ and shrinks toward the origin by an amount $\lambda$ all values of $U$ for which $|U| > \lambda$. It is the latter aspect that has led to $\mathcal{D}$ being called the shrinkage operator in addition to the soft thresholding operator.

### Variations on a theme

How do we determine $\lambda$? Let's say that the data has sample size $n$ if it is sampled at $n$ points $t_i$ such that $X_i \equiv X(t_i)$. Then for an orthogonal $W$, there will also be $n$ transform coefficients $Y_j$. If we prefer to use a threshold (such as the minimax threshold or the universal threshold)[6] that depends only on $n$, then $\lambda$ can be predetermined and we can use

## Code for the 1D example

In the WavBox software library, wavelet shrinkage denoising has been implemented in the `wsdenois` function. The following Matlab code generated Figure 1, in which all functions except `sprintf` are WavBox functions.

```
signam = 'Spires'; n = 2048; % initialize
     various settings
styp = 'STD'; spar = 7; zmf = 0; dtyp =
     'SUR'; hts = 0; vai = 0;
setwb('MROTYP','dwt1','FILCLA','orth','FILTY
     P','drola','ANAPAR',8);
setwb('CONTYP,''cps,''PHATYP,''peak,''SELSIZ',
     [n,1],'SAMFRE,'n,'DESLEV,'6);
getwb('USESEL'); % verify WavBox settings
S = scaleval(stsignal(signam),
     styp,spar,zmf); % scaled test signal
X = addnoise(S); % add Gaussian noise to
     signal
amp = axislims(X); % amplitude limits for
     signal plots
[R,Z,Y] = wsdenois(X,dtyp); % recovered
     signal in R
tit = sprintf('Wavelet Shrinkage Denoising:
     %s, %s, %s, %s, L = %g, n = %g',...
     signam,dtyp,getwb('FILNAM',0),getw
     ('CONTYP',0),getwb('MAXLEV',0),n);
 hax = multplot([2,2],loc,nam,tag,tit);
      % create multiple plot axis handles
tit = sprintf('Noisy Signal (SNR = %.2f
     SD)',esterror(S,X,styp));
plotsee(X,[],tit,[],amp,[],[],hax(1,1));
     % plot signal estimate error
tit = sprintf('Denoised Signal (SNR = %.2f
     SD)',esterror(S,R,styp));
plotsee(R,S,tit,[],amp,[],[],hax(2,1));
tit = 'Wavelet Transform of Noisy Signal';
plotdwt(Y,hts,vai,[],[],tit,hax(1,2)); % plot dis-
crete wavelet transform
tit = 'Denoised Wavelet Transform';
plotdwt(Z,hts,vai,[],[],tit,hax(2,2));
```

WavBox's software library has an extensive set of utilities including the `setwb` and `getwb` functions for automatically configuring and testing the wavelet transform parameters.[1] The above Matlab code excerpt produces four subplots in the figure window and returns the following output from the `getwb` function in the command window:

```
SignalInputDimension = 1
SignalInputSelectedSize = 2048 x 1
MappingClass = DSWT
MappingType = DWT
MappingSize = 2048 x 1
MultiResolutOutputClass = DWB
```

the three-step denoising procedure already described. However, if we prefer to use a data-adaptive threshold $\lambda = d(U)$ (such as the threshold selected by Stein's unbiased risk estimator (SURE))[7] that depends not just on $n$ but on $U$ (which again represents the data in any generic domain), then we must use a four-step procedure

$$Y = \mathcal{W}(X)$$
$$\lambda = d(Y)$$
$$Z = \mathcal{D}(Y, \lambda)$$
$$\hat{S} = \mathcal{W}^{-1}(Z)$$

for wavelet shrinkage denoising. Now we are distinguishing between the operator $d(\cdot)$, which selects the threshold, and the operator $\mathcal{D}(\cdot, \cdot)$, which performs the thresholding.

We won't review implementation of $\mathcal{W}$ here. Recall, however, that its analysis and synthesis wavelet filter banks, single-level convolutions and boundary treatment, and the total number $L$ of iterated multiresolution levels[8] must specify a wavelet transform. Thus, we can generate many different kinds of wavelet shrinkage denoising procedures by combining different choices for $\mathcal{W}(\cdot)$ and $d(\cdot)$. If we let $\mathcal{D}$ denote either the soft thresholding operator $\mathcal{D}_s$ or the hard thresholding operator $\mathcal{D}_h$,[6] then by combining choices for $\mathcal{W}(\cdot)$, $\mathcal{D}(\cdot,\cdot)$, and $d(\cdot)$ we can generate even more different kinds of wavelet-based denoising.

David Donoho and other researchers principally developed denoising by thresholding in the wavelet domain.[6,7,9] They introduced *RiskShrink* with the minimax threshold, *VisuShrink* with the universal threshold, and discussed both hard and soft thresholds in a general context that included ideal denoising in both the wavelet and Fourier domains.[6] They introduced *SureShrink* with the SURE threshold, *WaveJS* with the James-Stein threshold, and *LPJS* also with the James-Stein threshold but in the Fourier domain instead of the wavelet domain.[7] Here, for consistency of mnemonics, I rename *LPJS* to *FourJS*, analogous to *WaveJS*. Also, I label these various denoising procedures respectively *RIS*, *VIS*, *IWD*, *IFD*, *SUR*, *WJS*, and *FJS* for use here as abbreviations in the text and figure legends.

What distinguishes all these variations? Clearly, we can classify them by transform domain, Fourier or wavelet, as well as by intent of use. An ideal procedure requires a priori knowledge of the noise, whereas a practical procedure does not—ideal procedures are only used for purposes of comparison in theoretical analysis and simulation experiments. Moreover, we can classify the procedures according to whether

```
MultiResolutOutputType = DWT1
MultiResolutOutputSize = 2048 x 1
ConvolutionClass = CSFB
ConvolutionType = CPS
PhaseShiftType = PEAK
ExtensionType = C
MaximumLevel = 6
ScaleLengths
 2048 1
 1024 1
 512  1
 256  1
 128  1
 64   1
 32   1
FilterBankName = DROLA(16;8)
FilterBankDelay = 15
FilterBankError = 5.55112e-016
BiorthogonalityError = 5.55112e-016
OrthogonalityError = 7.77156e-016
SingleLevelConvolError = 6.90015e-016
MultiLevelMappingError = 1.26636e-015
```

Simply demonstrating a call to `wsdenois` does not reveal much about its internal workings. The following code excerpt shows the relevant calls that operate inside `wsdenois` in the case when the threshold depends only on *n* and no rescaling is performed prior to thresholding.

```
t = estthrsh(n,ten); % ten is threshold estimator
      name
Y = dwt(X); Z = Y;
for l = levels,
  for b = blocks,
  [i,j] = tabilc(l,b); % table of indices to
      levels blocks cells
  Z(i,j) = thrshold(Z(i,j),t,trn); % trn is
      threshold rule name
  end,
end,
R = idwt(Z); % recovered estimate of S in X
   = S + N
```

Of course, `dwt` and `idwt` correspond to $\mathcal{W}(\cdot)$ and $\mathcal{W}^{-1}(\cdot)$, while `estthrsh` and `thrshold` correspond to $d(\cdot)$ and $\mathcal{D}(\cdot,\cdot)$, respectively. Although the utilities `setwb` and `getwb` are unique to the WavBox software library, the important principles of wavelet shrinkage denoising demonstrated here with both math and code can be implemented in any programming language with calls to the corresponding functions in the appropriate libraries available for that language.

**References**


1. C. Taswell, "WavBox 4: A Software Toolbox for Wavelet Transforms and Adaptive Wavelet Packet Decompositions," *Proc. Villard de Lans Conf.*, Springer-Verlag, New York, 1995, pp. 361–375.


they use a single threshold globally for all relevant parts of the transform or multiple thresholds locally for different parts of it (Fourier frequency bands or wavelet multiresolution levels). For example, VIS is a practical, wavelet-domain, global threshold procedure in which we use $\lambda = \sqrt{2 \log n}$ for all levels $l = 1, \ldots, L$ from fine to coarse. As another example, SUR is also a practical wavelet procedure, but it uses a local threshold $\lambda_l$ estimated adaptively for each level *l*.

## A Monte Carlo experiment and a 2D example

I performed the first Monte Carlo experiment comparing any of these denoising procedures in work that was later published in a paper authored by David Donoho and Iain M. Johnstone.[6] Other authors have since published additional experiments.[2] Most of this work has examined four test signals that Donoho and Johnstone originally called *Doppler*, *HeaviSine*, *Blocks*, and *Bumps*.[6] Here, I've renamed the latter *Spires*. I could have also called it *Peaks*, but not Bumps, which seems inappropriate because bumps are usually rounded and not pointed. These four test signals with spatial inhomogeneity—along with two test signals with fractal regularity called Weierstrass and van der Waerden—are displayed as the standardized test signals in Figure 2, with additive white noise (SNR = 10 dB) in Figure 3, and as the denoised signal estimates in Figure 4 for one trial of SUR at $n = 1{,}024$.

Figure 5 shows results from multiple trials of all seven labeled denoising procedures over a range of values of *n* in a new Monte Carlo experiment with plots of SNR in dB versus $\log_2 n$. At each value of *n*, *L* is set to the maximum possible for that *n*. Another experiment held *L* constant as *n* increased. Both IWD and IFD are ideal procedures requiring a priori knowledge of the noise. All others are practical procedures in which the noise must be estimated and the transform coefficients scaled prior to thresholding. Restricting attention to the practical procedures, SUR, WJS, and FJS appear to perform well, but it is impossible to declare any of the procedures as the best under all test cases and sample sizes. However, we can declare VIS as the worst for all *n* and all of the six test signals investigated. If a Fourier-based method can perform as well as or better than a wavelet-based method, then these results would seem to counter the claims of optimality and generality for wavelet shrinkage denoising mentioned earlier.

Nevertheless, the theoretical claims of opti-

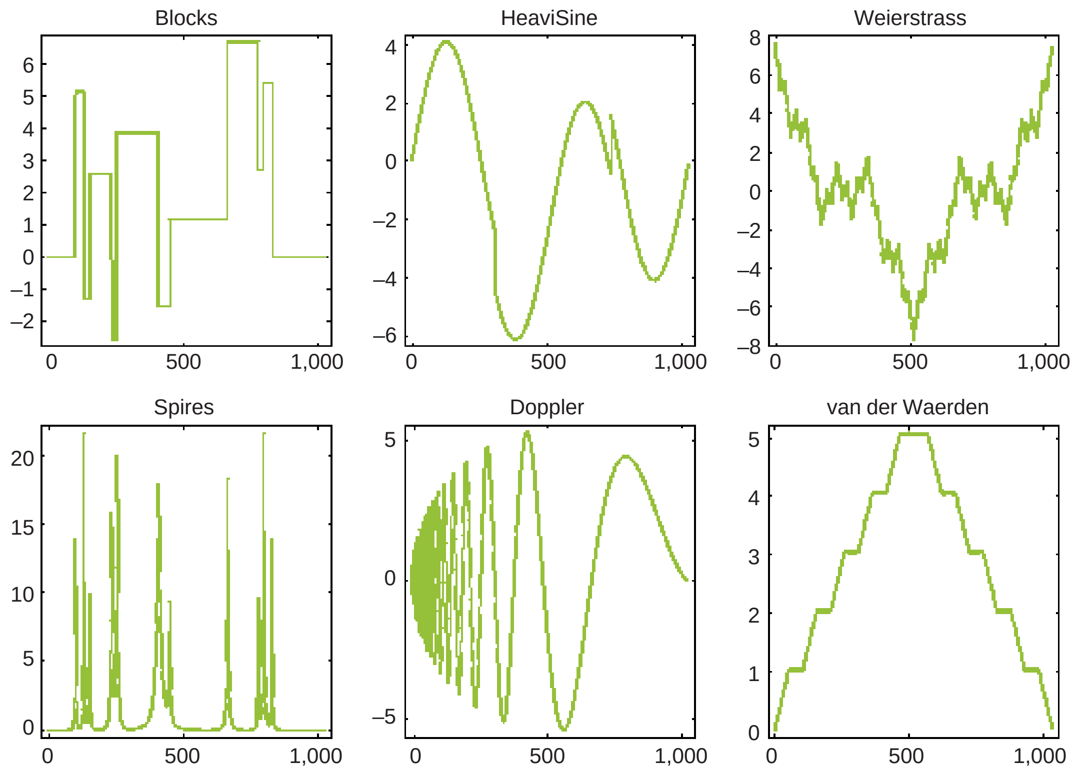


**Figure 2. Standardized test signals with $n$ = 1,024.**

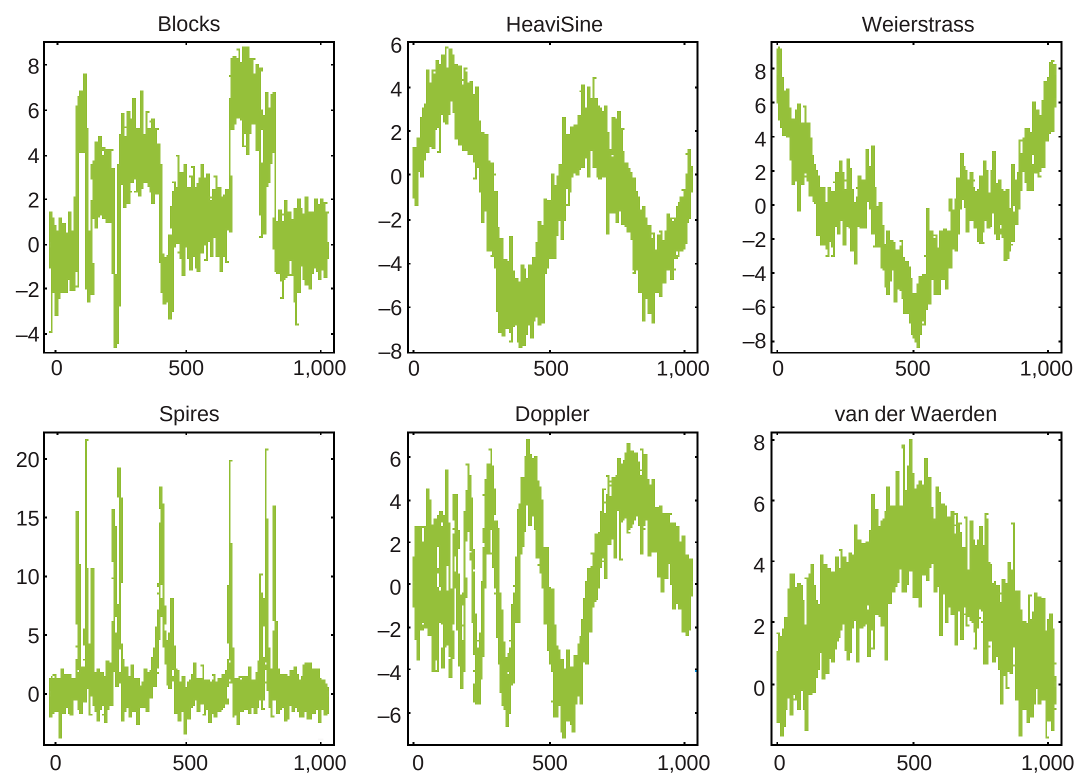


**Figure 3. Noisy test signals with $n$ = 1,024, SNR = 10.**

mality and generality pertain to a wide range of local and global measures of error, not just the one displayed in Figure 5, which is SNR measured in decibels. In fact, we can obtain varying results with different experimental conditions (signal classes, noise levels, sample sizes, or wavelet transform parameters) and error measures including the $l^1$, $l^2$, and $l^\infty$ norms as well as the

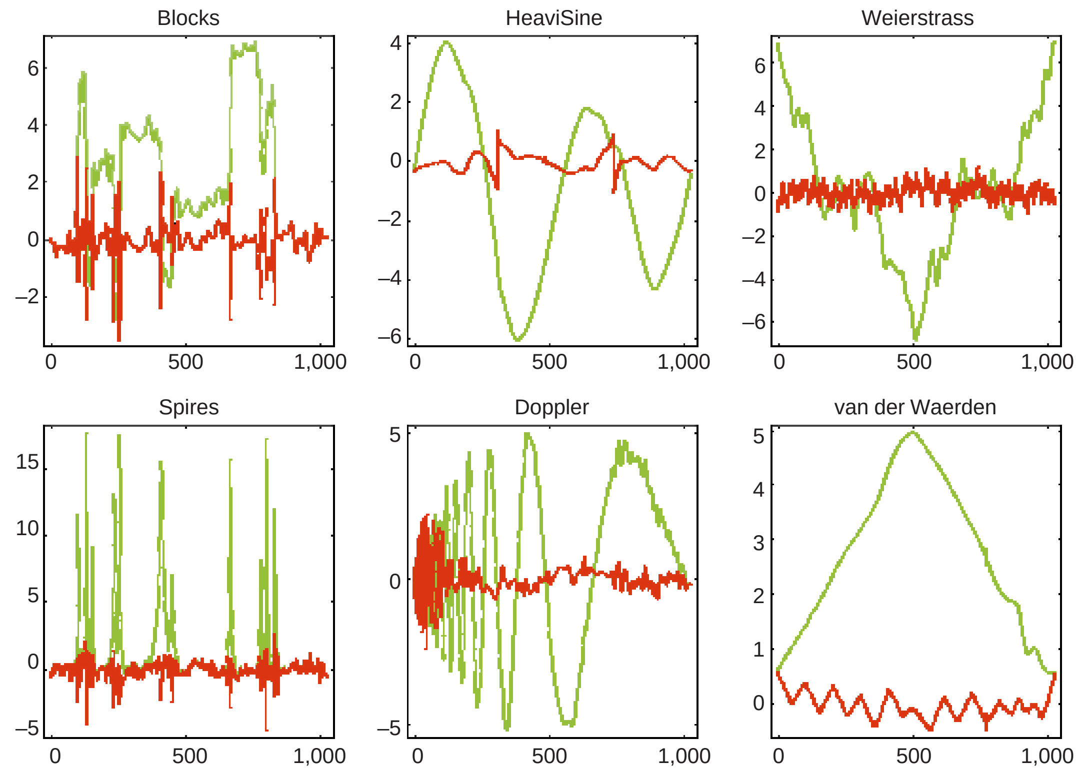


**Figure 4. Wavelet shrinkage denoising with 'SUR,' DROLA (10; 5), $n$ = 2,048, $L$ = 5.**

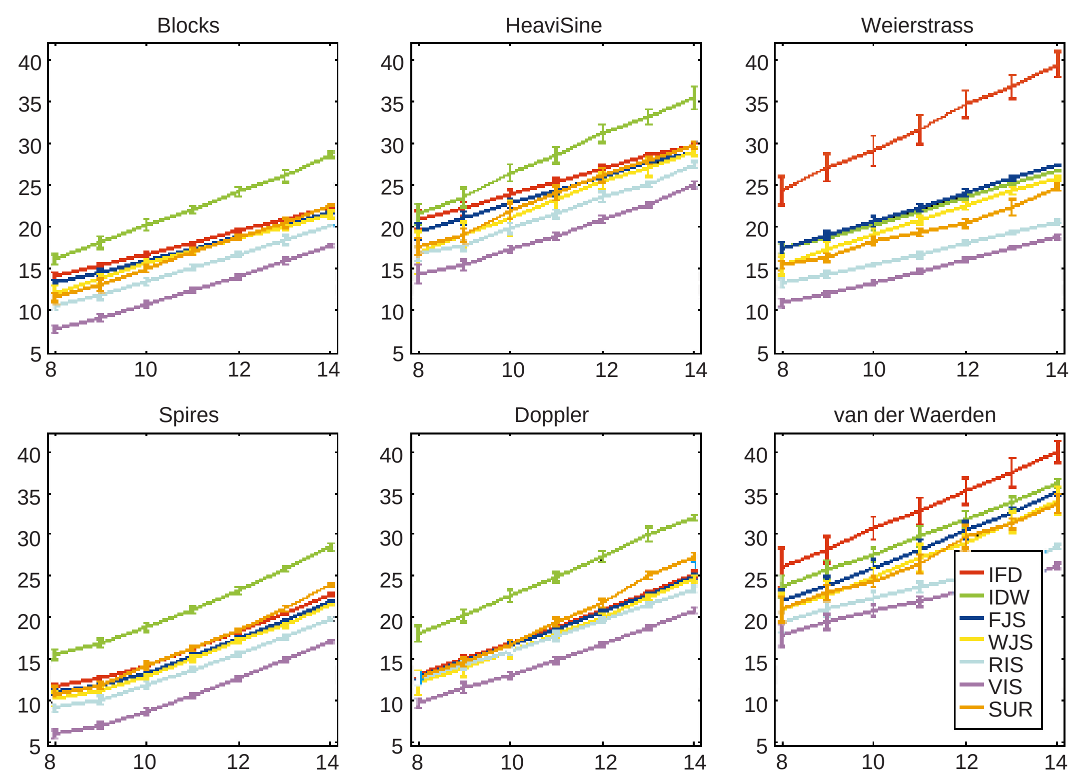


**Figure 5. Monte Carlo experiment comparing various denoising methods. Each curve plots SNR versus $L$.**

SNR (measured in standard deviations and in decibels). Which measure of error is most relevant? What about other figures of merit? For example, what if the error is not measured numerically, but rather is judged visually by the human eye and mind? In this case, Donoho and other researchers[2,6] have claimed that VIS performs best. To test this claim in a directly relevant manner, let's corrupt a photographic image with noise, and then denoise it with the VIS, RIS, and SUR procedures. Figure 6 displays the results obtained with SUR, which performed the best as judged solely by an aesthetic visual comparison with the original.

## An old debate between statistical theory and experiment

Ideally, the interplay between theory and experiment should provide the most productive progress in science and engineering. Too often, however, a rift has developed between theoreticians and experimentalists. Especially in statistics, theoreticians prove theorems based on asymptotic principles unrealistically requiring infinitely large sample sizes, whereas experimentalists perform experiments based on either real or synthesized data requiring only finitely small sample sizes. When do the large-sample theorems apply to small-sample experiments? Ultimately, the debate must be resolved by a choice of philosophy of approach and interpretation of common sense.

With regard to wavelet shrinkage denoising, the theoretical justifications and arguments in its favor remain highly compelling. The procedure does not require any assumptions about the nature of the signal, permits discontinuities and spatial variation in the signal, and exploits the spatially adaptive multiresolution features essential to the wavelet transform. Furthermore, the procedure exploits the fact that the wavelet transform maps white noise in the signal domain to white noise in the transform domain. Thus, although signal energy becomes more concentrated into fewer coefficients in the transform domain, noise energy does not. It is this important principle that enables the separation of signal from noise.

Wavelet shrinkage denoising has been theoretically proven to be nearly optimal from the following perspectives: spatial adaptation, estimation when local smoothness is unknown, and estimation when global smoothness is unknown. In effect, no alternative procedure can perform better without knowing a priori the signal's smoothness class. But is it really necessary or appropriate to use a procedure that is in this sense theoretically optimal and general under most measures of local and global error for data about which there is no a priori knowledge?

The answer is that it probably isn't necessary for most practitioners who know something about their data and concern themselves often with only one critical outcome measure rather than many. For example, if we feed the denoised signal's features into a neural-network pattern recognizer, then the rate of successful classification should determine the ultimate measure by which to compare various denoising procedures.

If we adopt the commonsense approach to practical problem solving, the practitioner should exploit any and all a priori information available for his or her particular problem, and use an appropriate denoising procedure as determined by the most relevant outcome measure. Determining the most appropriate procedure necessarily involves experiments to compare the performance of a wavelet shrinkage denoising method (com-

Original   Noisy (SNR = 10 dB)   Denoised (SNR = 18 dB)

**Figure 6. Wavelet shrinkage denoising with 'SUR,' DROLA (10; 5), $n = 464 \times 320$, $L = 4$.**

prising the most effective combination of wavelet transform parameters and denoising rules and thresholds for the range of sample sizes and noise levels expected) with any other methods under consideration. In addition, we must consider issues of computational complexity. Algorithm complexity might be measured according to CPU computing time and floating-point operations, or the number and kind of algorithm steps and their impact on firmware or hardware requirements.

It is unlikely that one particular wavelet shrinkage denoising procedure will be suitable, no less optimal, for all practical problems. However, it is likely that there will be many practical problems, for which after appropriate experimentation, wavelet-based denoising with either hard or soft thresholding proves to be the most effective procedure. Using wavelet-based denoising of the log-periodogram to estimate the power spectrum might prove to be one such important application with great promise for further development.[10]

## References


1. S. Shearman, "Wavelet Software Targets a Broad Range of Users," *Personal Eng. & Instrumentation News*, Vol. 15, No. 3, Mar. 1998, pp. 25–30.
2. D.L. Donoho et al., "Wavelet Shrinkage: Asymptopia? With Discussion," *J. Royal Statistical Soc., Series B*, Vol. 57, No. 2, 1995, pp. 301–369.
3. S. Shearman, "Parametric Curve-Fitting Software Satisfies Many Needs," *Personal Eng. & Instrumentation News*, Vol. 15, No. 9, Sept. 1998, pp. 24–31.
4. C. Taswell, *WavBox Software Library and Reference Manual*, Computational Toolsmiths, Stanford, Calif., 1998.
5. C. Taswell, *Constraint-Selected and Search-Optimized Families of Daubechies Wavelet Filters Computable by Spectral Factorization*, Tech. Report CT-1999-05, Toolsmiths, Stanford, Calif., 1999; www.toolsmiths.com.
6. D.L. Donoho and I.M. Johnstone, "Ideal Spatial Adaptation via Wavelet Shrinkage," *Biometrika*, Vol. 81, Sept. 1994, pp. 425–455.
7. D.L. Donoho and I.M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage, *Biometrika*, Vol. 81, Sept. 1994, pp. 425–455.
8. C. Taswell, "Specifications and Standards for Reproducibility of Wavelet Transforms," *Proc. Int'l Conf. Signal Processing Applications and Technology*, Miller Freeman, San Francisco, 1996, pp. 1923–1927.
9. D.L. Donoho, "Denoising via Soft Thresholding," *IEEE Trans. Information Theory*, Vol. 41, May 1995, pp. 613–627.
10. P. Moulin, "Wavelet Thresholding Techniques for Power Spectrum Estimation," *IEEE Trans. Signal Processing*, Vol. 42, Nov. 1994, pp. 3126–3136.



**Carl Taswell** works at Computational Toolsmiths in Stanford, California. His research and technical interests focus on wavelet transforms with data compression and pattern-recognition applications in biomedical signal and image processing. He has authored several software packages including LDACPA software, ELIDA software, the WavBox software library, and the FirWav filter library. He received a BA in biochemistry from Harvard University, and an MS in mathematics and an MD in medicine from New York University. Additionally, he received an MS and a PhD in scientific computing and computational mathematics from Stanford University. Contact him at ctaswell@toolsmiths.com; www.toolsmiths.com.