Smoothing irregularly sampled signals using wavelets and cross validation

Maarten Jansen
Adhemar Bultheel

Report TW 289, April 1999

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Smoothing irregularly sampled signals using wavelets and cross validation

Maarten Jansen
Adhemar Bultheel

Report TW 289, April 1999

Department of Computer Science, K.U.Leuven

Abstract

Coefficient thresholding is a popular method in wavelet based noise reduction. A wavelet decomposition is typically a sparse representation of noise-free signals: the essential information is captured by a limited number of large, important coefficients, while the main part of coefficients is close to zero. Replacing these small coefficients by zero is a straightforward way to reduce noise variance without affecting the noise-free signal too much. Recently, algorithms have been developed for wavelet decompositions of non-equidistant samples, using the so called lifting scheme and second generation wavelets. We investigate how to apply these algorithms to reduce noise in signals on a non-equidistant grid. The paper also illustrates that the method of generalized cross validation in these settings still succeeds in finding good thresholds.

Unlike other methods, we do not 'precondition' the input, but use the lifting scheme to deal with the irregularity of the grid. This approach seems more natural and shows promising results. However, instability problems arise from the actual scheme. This article describes the method and explains in what cases problems may occur.

Keywords : Noise reduction, second generation wavelets, lifting, thresholding, cross validation.
AMS(MOS) Classification : 41A30, 93E14, 60G35, 62G07, 65D10, 62J02

Smoothing irregularly sampled signals using wavelets and cross validation

Maarten Jansen and Adhemar Bultheel

Department of Computer Science
K.U.Leuven
Celestijnenlaan 200A, B3001 Heverlee - Belgium
`phone: ++32 16 32 7080 fax: ++32 16 32 7996`
`e-mail: maarten.jansen@cs.kuleuven.ac.be`

Technical Report 289, April 1999

Abstract

Coefficient thresholding is a popular method in wavelet based noise reduction. A wavelet decomposition is typically a sparse representation of noise-free signals: the essential information is captured by a limited number of large, important coefficients, while the main part of coefficients is close to zero. Replacing these small coefficients by zero is a straightforward way to reduce noise variance without affecting the noise-free signal too much. Recently, algorithms have been developed for wavelet decompositions of non-equidistant samples, using the so called lifting scheme and second generation wavelets. We investigate how to apply these algorithms to reduce noise in signals on a non-equidistant grid. The paper also illustrates that the method of generalized cross validation in these settings still succeeds in finding good thresholds.

Unlike other methods, we do not 'precondition' the input, but use the lifting scheme to deal with the irregularity of the grid. This approach seems more natural and shows promising results. However, instability problems arise from the actual scheme. This article describes the method and explains in what cases problems may occur.

keywords:
Noise reduction, second generation wavelets, lifting, thresholding, cross validation.

1 Introduction

Classical wavelet theory and algorithms [2, 10] are based on Fourier analysis, as a tool for filter design. A wavelet decomposition writes a function as a linear combination of dilations and translations of one function $\psi(x)$. This procedure algorithm requires equidistant samples. Recently, a new algorithm for wavelet transforms has been developed, called the *lifting scheme* [11]. Not only this algorithm is faster, but it also allows an extension of wavelet theory to the case of data on an irregular grid [12].

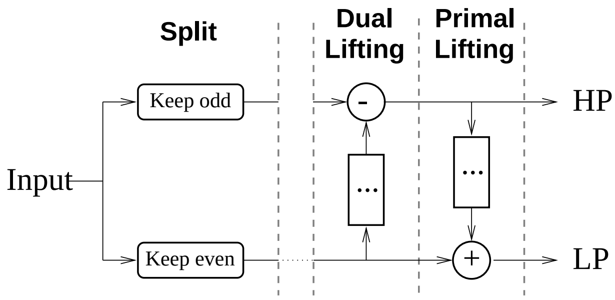

Figure 1: Decomposition of a filter bank into lifting steps. The first type of lifting is called *dual lifting* or a *prediction step.* The other type is *primal lifting* or *update.*

A wavelet threshold procedure [4] starts with a discrete wavelet transform of the input. In a second step, coefficients beneath a certain threshold are replaced by zero. Inverse transform yields the result. The main issue in this procedure is the selection of the threshold. This parameter should be chosen so that the eventual result is as close as possible to the unknown noise-free signal. Here we use the method of generalized cross validation [14, 15, 9], which does not need an estimate of the amount of noise.

This paper investigates how to adapt a wavelet threshold procedure for coefficients on a non-equidistant grid. Unlike other methods [7, 1], we do not 'precondition' the input, but use the lifting scheme to deal with the irregularity of the grid. This approach seems more natural and shows promising results. However, instability problems arise from the actual scheme.

2 Lifting and non-equidistant data

A classical wavelet decomposition algorithm has the structure of a repeated filter bank algorithm. In the first instance, the lifting scheme decomposes a filter bank operation in a number of consecutive lifting steps [3]. This series starts by splitting the input vector into points with odd and points with even index (Figure 1).

There exist two kinds of lifting operations. The first, called dual lifting, subtracts a filtered version of the "even" input from the "odd" input. The second, called primal lifting, adds a filtered version of the dual lifting output to the so far untouched "even" input. One way to interpret the dual lifting step, is the following: we assume that the "even" input and the "odd" input are highly correlated. This is certainly true in the first lifting step, where the "even" and the "odd" input are directly originating from one input signal. We now try to *predict* the odd samples by a prediction operator (a filter) on the even ones. By subtracting this prediction from the odd samples, we reduce correlation. These differences are high-pass coefficients. The second lifting step is an update step. It can be interpreted as a way to preserve the average (and higher moments) in the low-pass coefficients.

Although a general decomposition into lifting steps may consist of several stages with primal and dual lifting, this study restricts itself to two simple

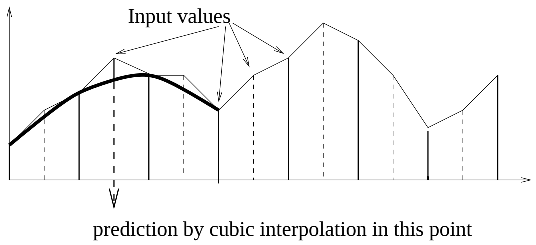

Figure 2: A cubic interpolation as a prediction operator. The thin, piecewise linear line links the input data. The bold line is an example of a cubic polynomial, interpolating 4 successive points with even index.

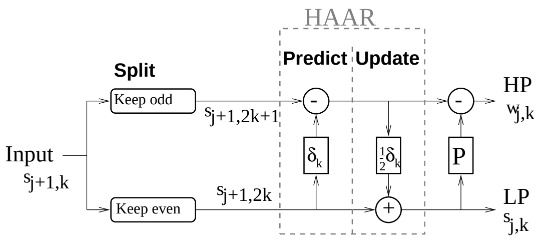

Figure 3: Lifting scheme based on average interpolation. This is a Haar transform, followed by one more prediction step. A second update step (not shown on figure) is also possible.

cases. The first scheme is based on interpolating subdivision: it consists of one prediction step followed by one update step. The prediction formula is an interpolating polynomial, see Fig. 2 The second scheme is average interpolation: it is a Haar prediction and update, followed by one more prediction step, see Fig. 3 This second prediction step ensures that if the input are *averages* of polynomials on the intervals between the data points, all detail coefficients are zero. An extra update may follow to enhance the number of vanishing moments of the dual wavelets.

This lifting philosophy is by no means limited to equidistant samples. The idea of interpolation, for instance, can be extended to an irregular grid, as shows Fig. 4 in the case of linear interpolation.

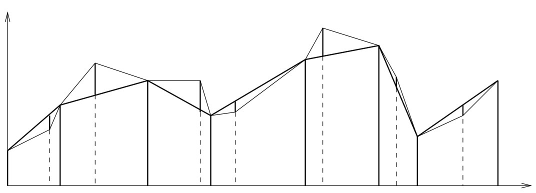

Figure 4: Linear prediction operator on an irregular grid.

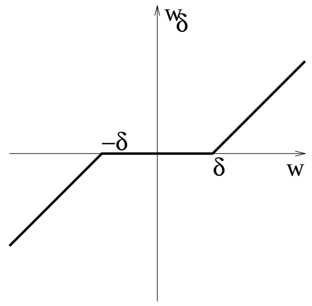

Figure 5: Soft-thresholding: Coefficients below a threshold value λ are replaced by 0, whereas others are shrunk.

3 Thresholding second generation wavelets

3.1 Thresholding, bias and variance

Suppose we have the following model of discrete data, corrupted by noise:

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\eta}.$$

The first step of a threshold algorithm is a forward wavelet transform of the noisy data vector $\boldsymbol{y}$:

$$\boldsymbol{w} = \tilde{W}\boldsymbol{y},$$

where $\tilde{W}$ is the forward wavelet transformation matrix. Coefficients below a threshold value λ are replaced by 0, whereas others are shrunk, as illustrated in Fig. 5: this is soft-thresholding. We denote by $\boldsymbol{w_\lambda}$ the vector of thresholded coefficients. An alternative for this procedure would be hard-thresholding: a hard-threshold leaves coefficients with magnitude above the threshold untouched. While at first sight hard-thresholding may seem a more natural approach, soft-thresholding is a more continuous operation, and it is mathematically more tractable. Moreover, hard-thresholding often leads to a result with more spikes and artifacts. The noise reduction algorithm concludes with an inverse transform:

$$\boldsymbol{y_\lambda} = W\boldsymbol{w_\lambda},$$

where $W = \tilde{W}^{-1}$ is the inverse wavelet transformation matrix. Thresholding reduces noise at the price of introducing bias. The optimal threshold finds the best compromise, e.g. in ℓ_2-sense: it minimizes the mean squared error with respect to the unknown data:

$$\lambda_{\mathrm{MSE}} = \arg\min_{\lambda} \|\boldsymbol{w_\lambda} - \boldsymbol{v}\|^2,$$

Intuitively, it is clear that the optimal threshold is (approximately) proportional to the amount of noise on the wavelet coefficients. This is confirmed by the asymptotic behavior of this threshold. Under mild conditions it behaves like:

$$\lambda_{\mathrm{MSE}} \sim \sqrt{2\log N}\,\sigma.$$

(Note that this is also the value of the “universal” threshold [5].)

If we know that these data are defined on an irregular grid:

$$y_i = f(x_i) + \eta_i, i = 1, \ldots, N,$$

where the x_i are not equistant, then we may apply a second generation transform. Since this transform takes into account the lattice of the data, the noise standard deviation is different for each coefficient, even if the noise on the input data had a constant standard deviation. This lack of stationarity makes thresholding difficult: if the amount of noise is different for each coefficient, it is hard to remove it decently by only one threshold.

However, if we know the covariance structure of the input noise, we can compute the variance fluctuation:

$$D = \tilde{W} C \tilde{W}^T,$$

where $C = \mathrm{E}\boldsymbol{\eta}\boldsymbol{\eta}^T$ is the correlation matrix of the input, and D is the correlation matrix in the wavelet domain. If C is a banded matrix, D can be computed in a linear amount of time. In practical cases, the exact values of C are often unknown, but the structure of C may be known, i.e. C may be known up to a constant. The case of white noise, for instance, corresponds to: $C = cI$, with I the unit matrix.

The normalised coefficients

$$\tilde{w_i} = w_i / \sqrt{D_{ii}}$$

do have a constant variance and thresholding these coefficients makes more sense than thresholding the original ones.

3.2 Threshold selection

The matrix D may not contain the exact variances, but only the structure of the covariance matrix. This is the case if we know the structure of the correlation of the input noise. In many practical situations, for instance, it is reasonable to assume that the noise is white and stationary without specifying the exact noise level.

To find a good threshold without using an estimate for this noise level, we apply the method of generalized cross validation (GCV) [14, 15, 9]. GCV is a function of the threshold value which mimics the MSE-function. Unlike this MSE, GCV only uses known variables and there is no noise deviation involved:

$$GCV(\lambda) = \frac{\frac{1}{N}\|\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{w}}_{\boldsymbol{\lambda}}\|^2}{[\frac{N_0}{N}]^2}.$$

In this equation $\tilde{\boldsymbol{w}}_{\boldsymbol{\lambda}}$ is the vector of thresholded normalised coefficients and N_0/N is the fraction of coefficients replaced by zero by this particular threshold value.

The minimizer of this function is an asymptotically optimal threshold in ℓ_2-sense, i.e. for N sufficiently large, the minimizer of $GCV(\delta)$ also minimizes the mean square error of the result, as compared to the unknown, noise-free wavelet coefficients [9]. This property still holds if the coefficient noise is correlated [8].

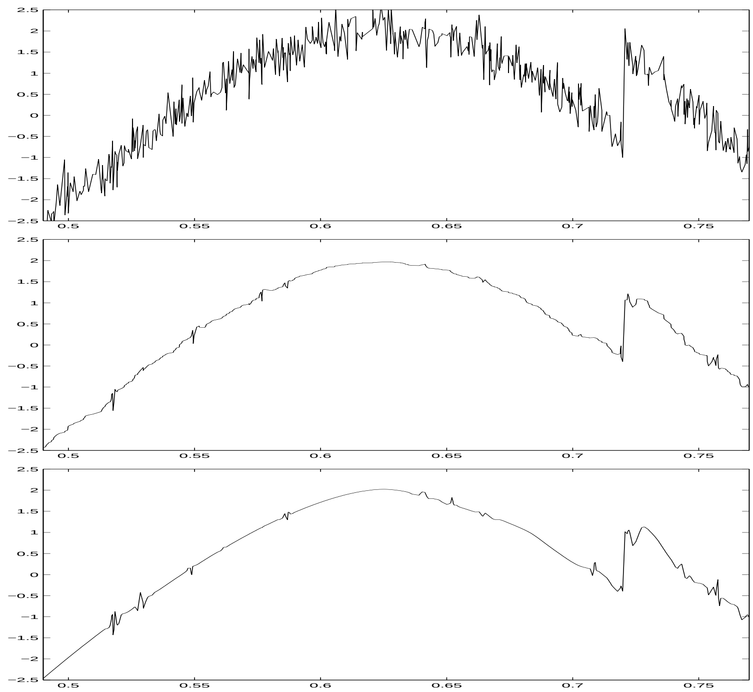

Figure 6: Example 1: Top: noisy "heavy sine" function on a "not too" irregular grid. The grid was obtained as an ordered set of randomly chosen points on the interval $[0, 1]$. Middle: result of a threshold algorithm on a classical wavelet transform. We run the lifting scheme but tell the algorithm that the grid is regular. The result is noisy, because the regular grid transform does not correspond to the real grid. Bottom: result of the same algorithm based on the actual grid. In both cases, we use GCV to estimate the MSE-threshold.

3.3 Two examples

We illustrate the effect of using second generation wavelets with two examples. In the first example, the grid was obtained by selecting $N = 2048$ points x_k at random between 0 and 1. These points were ordered and used as sampling points for the "heavy sine" function [6]:

$$\begin{aligned} f(x) &= 4\sin(4\pi x) - \mathrm{sign}(x - 0.3) - \mathrm{sign}(0.72 - x) \\ y_k &= f(x_k) + \eta_k, k = 1, \ldots, N \end{aligned}$$

[6]. The figures show a detail of 600 points. The algorithm used a lifting-scheme, based on cubic interpolation for prediction and a two-taps update filter. We can neglect the grid structure, i.e. we run the algorithm with an equidistant grid, this means that we are smoothing the data (k, y_k) instead of x_k, y_k. The result is noisy, because the regular grid transform does not correspond to the real grid. The spikes in the result are inherent for this simple threshold algorithm. More sophisticated algorithms should be able to remove them. However, the curve in between these spikes is much smoother if we use second generation wavelets.

A second example is a damped sine ($f(x) = \mathrm{e}^{-x}\sin 4\pi x$) on an extremely irregular grid. This grid was constructed as follows: we choose approximately

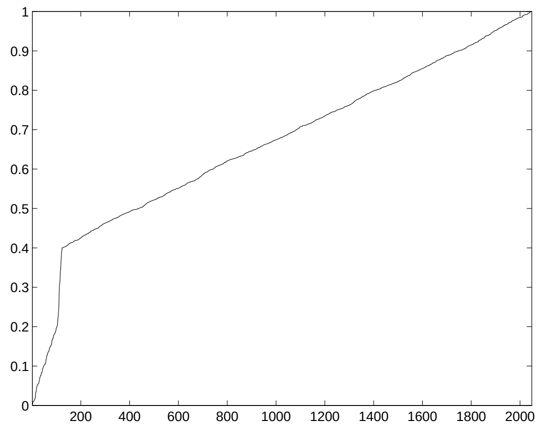

Figure 7: The grid of Example 2: this very irregular grid was constructed as follows: we choose approximately 100 samples at random between 0 and 0.2, about 10 samples between 0.2 and 0.4 and about 1940 samples between 0.4 and 1.

100 samples at random between 0 and 0.2, about 10 samples between 0.2 and 0.4 and about 1940 samples between 0.4 and 1. Figure 7 plots the grid point versus the point number. If we add white and homoscedastic (second order stationary) noise to this function, we get the upper plot of Figure 8. The left part of this plot looks less noisy, but this is because data points in the right tail are much closer to each other. As for the previous example, second generation wavelets give a generally smoother result, but in this case, this scheme introduces a tremendous bias, not only in the region with few data points, but also at places where data are given close to each other. One could argue that this example is somehow artificial. Moreover, the phenomena seem to appear mostly at coarse scale, and it is a common practice to leave coefficients at coarse scales untouched. However, if we run the same algorithm pretending the grid to be regular, the result is quite fair, apart from the grid irregularities, of course. We now investigate where this bias comes from and what we can do to make the second generation algorithm perform at least as well as the "first" generation wavelets.

4 The bias

4.1 The problem

The bias comes from the fact that the second generation wavelet transform may be far from orthogonal. This appears in several effects, which sometimes enhance each other.

1. A small coefficient may have a wide *impact*, especially when it is related to a region with only a few samples. Thresholding it causes an important effect in the original domain.

2. Basis functions sometimes have a large overlap, especially in the neighborhood of boundaries. Large individual coefficients may then compensate

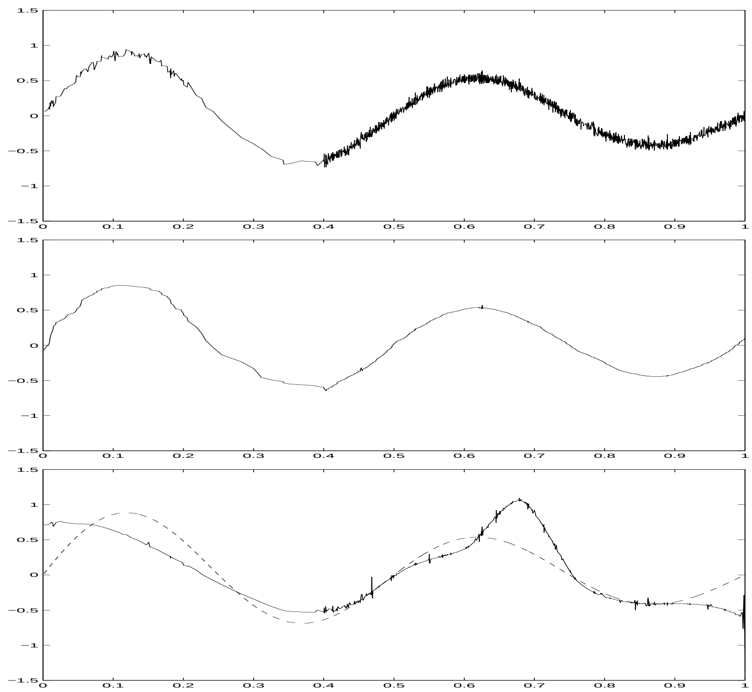

Figure 8: Example 2: Top: noisy signal ($f(x) = \mathrm{e}^{-x} \sin 4\pi x$) on the grid of Figure 7. Middle: result of a threshold algorithm on a classical wavelet transform. The lack of smoothness in this result reflects the irregularity of the grid. Using second generation wavelets leads to a much smoother result, but for this example, this scheme causes an unacceptable bias.

each other, resulting in a signal with a relatively small energy. Thresholding these coefficients destroys the balance between the large coefficients and causes artifacts: hidden components suddenly become visible.

3. A transform has a bad condition number if it is sensible to errors on the input. As a matter of fact, thresholding can be considered as an artificial error on the input, and typically, the threshold is much larger than machine-precision! If the transform is not stable, there is no guarantee that the output is close to the input, even if it is so in wavelet-domain.

4. The threshold is proportional to the standard deviation of a coefficient. Unlike in the stable case, coefficients with large variance may correspond to basis functions with large energy. Or, equivalently, dividing wavelet coefficients by their standard deviation may cause important coefficients to become relatively small.

The bad condition of such a wavelet transform plays a role in other applications too, of course. From the statistical point of view, we are specifically interested in the interaction between variance normalization and bias, as described in our last item of this enumeration. In short, a bad conditioned transform makes it difficult, if not impossible, to predict the effect of a threshold on a coefficient.

4.2 Where does the bad condition come from?

At this moment we have no exact theoretical explanation of this bad condition. Possibly, the update step plays an important role in this phenomenon: it turns out that an update filter with two taps $A_{j,k}$ and $B_{j,k+1}$ defines a wavelet function at scale j and place k as a combination of scaling function at two scales [13]:

$$\psi_{jk} = \varphi_{j+1,2k+1} + A_{jk}\varphi_{j,k} + B_{j,k+1}\varphi_{j,k+1}.$$

If the update filter coefficients are large, ψ_{jk} is close to the subspace spanned by the scaling functions $\varphi_{j,k}$ at the same scale.

It is clear that the lifting theory as such neglects the notion of scale: if a sequence of dense samples is followed by a large gap, the transform operates on phenomena at different scales in one single step. In one way or another, the transform should be re-organized so that it deals with phenomena at one scale in each step. However, this reordering of downsampling the coefficients seems not trivial at all.

We remark that at least the Haar transform remains orthogonal on an irregular grid. For the CDF 2,2-transform, which corresponds to linear interpolation prediction and a simple update, the problems remain marginal. In both cases, there is almost no mixture of scales possible: one (for Haar) and even two (CDF 2,2) prediction points never show a structure with two different scales. In a cubic interpolation scheme, however, the four interpolation points may reflect phenomena at two different scales, for instance, if three points are close to each other and the fourth is at a long distance from this cluster.

Figure 9 illustrates that a small error in one of the interpolation points may cause a serious error in the points where this interpolating polynomial is used as a prediction. The figure shows the errors caused by a unit error in one of the interpolation points: this function is the *difference* between the correct

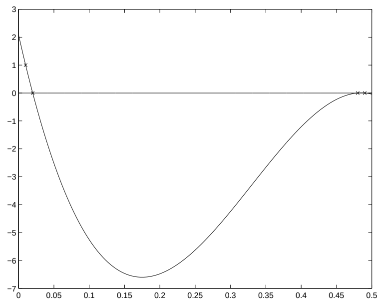

Figure 9: Effect on the interpolating polynomial of an error in the first interpolating point. This function is the difference between the correct interpolating polynomial (not shown here) and the polynomial that comes out if the error in the first interpolating point (0.01) equals one. This illustrates the problem that these error function may become large if the interpolating points are far from equidistant.

interpolating polynomial (not shown here) and the polynomial that comes out if the error in the first interpolating point (0.01 in the example) equals one. This difference or error function itself is a Lagrange interpolating polynomial.

5 How to deal with the bias?

Essentially, there are two possible ways to overcome the problem of the bad condition. The first is trying to modify the transform so that it becomes more stable. Since at this moment we do not completely understand the origin of the instability, and because we believe that reorganizing the algorithm would be a rather hard job, we prefer an alternative solution. We examine which coefficients are dangerous to threshold, and how to find an appropriate value for these coefficients.

5.1 Computing the impact of a threshold

In the first instance we try to save the coefficients that correspond to large energy basis functions from thresholding. We examine for each coefficient the influence of a threshold proportional to its noise level $\lambda = k\sigma_i$. We have that

$$\sigma_i = \sqrt{(\tilde{W} C \tilde{W}^T)_{ii}}.$$

We assume that the input noise is uncorrelated (white) and homoscedastic: $C = \sigma^2 I$. Each coefficient w_i corresponds to a basis function. The 2-norm of this function can be computed as:

$$E_i = \int_{-\infty}^{\infty} \psi_i^2(x)\, dx = \sqrt{(W^T S_J W)_{ii}} = \sqrt{\left[(\tilde{W} S_J^{-1} \tilde{W}^T)^{-1}\right]_{ii}},$$

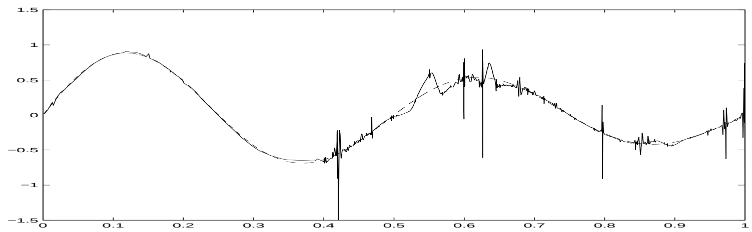

Figure 10: Result if we preserve coefficients with a large impact from thresholding. The most serious bias has gone, but the result has lost smoothness and it is difficult to define a threshold between coefficients with large and small impact.

where S_J is a diagonal matrix containing the squared norms of the scaling function at the initial, fine resolution:

$$S_{J,kk} = \int_{-\infty}^{\infty} \varphi_{J,k}^2(x)\,dx \approx \int_{(x_{k-1}+x_k)/2}^{(x_k+x_{k+1})/2} 1^2\,dx.$$

This norm E_i is a measure for the effect of a "unit-threshold". The total effect $\Delta_{\boldsymbol{y}}$ of thresholding is given by the following expression of impact:

$$\Delta_{\boldsymbol{y}} = k\sqrt{(\tilde{W}C\tilde{W}^T)_{ii}\left[(\tilde{W}S_J^{-1}\tilde{W}^T)^{-1}\right]_{ii}}.$$

For orthonormal transforms on a regular grid and with uncorrelated, homoscedastic noise, this effect would be independent of W: it only depends on the threshold value $\Delta_{\boldsymbol{y}} = k\sigma = \lambda$. Figure 10 shows the result if we preserve coefficients with a large impact from thresholding. The most serious bias has gone, but the result has lost smoothness and it is difficult to define a threshold between coefficients with large and small impact.

5.2 Correlation between coefficients and hidden components

The computation in the previous section only takes into account the 2-norm of separate basis functions. The inner product of two functions, which is responsible for inter-coefficient correlations, does not appear in the algorithm. The peaks in the result are the consequence of this approach, as illustrates the following example. Figure 11 shows an experiment where one particular second-generation wavelet coefficient of the noisy signal was replaced by zero. Inverse transform reveals a tremendous effect. The coefficient had a rather large magnitude, and apparently also a wide impact, but comparison of the results in Figure 11 and Figure 8 indicates that the same coefficient was classified as not important by the threshold algorithm. This is because not only its magnitude was large, but so was its variance. If we remove the same threshold from the noise-free wavelet coefficients, we get the reconstruction in Figure 12. The difference with the original function is hardly visible. The threshold algorithm was right to remove it. A simple example in $\mathbb{R}^3$ makes clear what happens. Suppose we have the basis vectors $\{(-1/2, \sqrt{3}/2, 0), (-1/2, -\sqrt{3}/2, 0), (1, 0, \varepsilon)\}$. If ε is small, this basis has an extremely bad condition. Suppose the noise is $(0, 0, \varepsilon)$

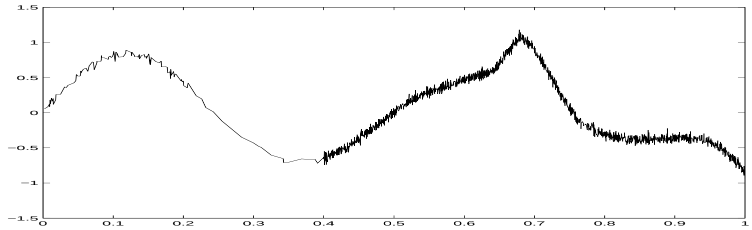

Figure 11: Reconstruction after removing one coefficient from the noisy transform. The effect is enormous, but the coefficient was rather big.

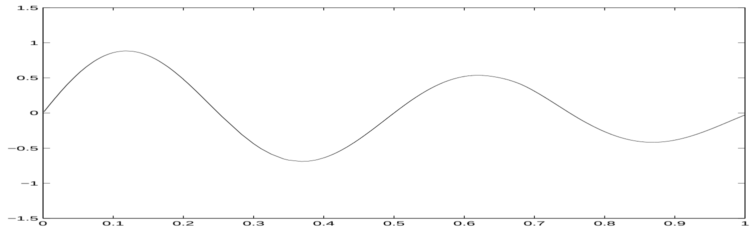

Figure 12: Reconstruction after removing the same coefficient as in Figure 11 from the noise-free transform. The effect is quasi nihil.

in the canonical basis, then its coordinates in this oblique basis are $(1, 1, 1)$. If one or two of these coordinates are thresholded, "hidden components" become clear. This bad condition can only be detected with a global analysis: none of the basis vectors is close to another one. In the example of Figure 11, the noise made small coefficients big, because it did not fit well into the oblique basis. Removing some of these coefficients uncovers these hidden components. The result of Figure 10 does not contain the same bias as in Figure 11. This means that the computation of the impact of the coefficients saved the coefficient of Figure 11 from being thresholded. This is not what we want: not only it does not correspond to what the noise-free coefficient says (this is what Donoho and Johnstone call the "Oracle"), but also, if we keep this large, purely noisy coefficient, we have to keep all the others that compensate for its effect. These are hard to find, and if we find them, we end up with a result without any noise-*reduction* at this place. We would like to remove all of these large noise coefficients and therefore we want a reliable estimation of the noise-free signal: this estimation does not have to be smooth, but it should learn us which of the big coefficients are really important and which are due to noise. Unlike the classical (bi-)orthogonal transform, the second-generation transform no longer guarantees that coefficients with a large magnitude are important.

Another unpleasant consequence is the fact that scaling coefficients which are not further transformed may carry a lot of noise too. Most algorithms do not threshold scaling coefficients, and this may uncover, once more, hidden noise components. A reliable estimation of the noise-free signal could give us an idea of the effect of the noise on the low-resolution scaling coefficients.

5.3 Starting from a first-generation solution

We know that if the transform neglects the grid structure, the result reflects the irregularity of the grid. The result is non-smooth, which means that it has no sparse representation in a second-generation basis. Apart from that, the result is fairly reliable, in the sense that bias is restricted by the Riesz-constants of the transform: if we are thresholding in the wavelet domain, we know what we are doing in the original signal domain. Let $\boldsymbol{w}^{(1)}$ be the second generation wavelet coefficients of this first generation solution $\boldsymbol{y}^{(1)}$. Our objective is to find a sparsely represented signal close to $\boldsymbol{y}^{(1)}$. To this end, we use the thresholded coefficients $\boldsymbol{w}_{\boldsymbol{\lambda}}$ of the second-generation transform of the noise.

If a coefficient w_i corresponds to a wavelet that lies on an interval where the second-generation solution $\boldsymbol{y}_{\boldsymbol{\lambda}}$ shows no bias, we can choose as output:

$$\hat{w}_i = w_{\lambda i}$$

To do so, we have to define in which data points $\boldsymbol{y}_{\boldsymbol{\lambda}}$ is biased and we have to mark the coefficients that correspond to these points. We say that $y_{\lambda i}$ is biased if

$$|y_{\lambda i} - y_i^{(1)}| > \hat{\sigma},$$

where:

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i^{(1)})^2}$$

is an estimate of the noise variance (we suppose that the noise is stationary). This definition is subject to the remaining noise and the irregular grid effects in $y_i^{(1)}$. Because we expect that bias has typically a range of more than one data point, we first filter out isolated points that were classified as biased, before the actual marking of the corresponding wavelet coefficients.

For all these marked coefficients w_i we compute the value of

$$(w_{\lambda i} - w_i^{(1)})^2 \int_{-\infty}^{\infty} \psi_i^2(x)\, dx = (w_{\lambda i} - w_i^{(1)})^2 (W^T S_J W)_{ii},$$

which quantifies the effect on the output if we replace $w_i^{(1)}$ by $w_{\lambda i}$. If we compute the sum of these effects over all marked coefficients, we see that a few of them are responsible for the major part of the bias. These coefficients, together with the untouched scaling coefficients, keep their value $w_i^{(1)}$. All others undergo the same procedure as the unmarked coefficients.

This procedure eliminates all large coefficients that do not interfere with biased reconstruction points. This is how the algorithm gets rid of most hidden noise components.

Instead of marking wavelet coefficients that correspond to intervals with bias, we can also compute for *all* coefficients the value of:

$$B_i = (w_{\lambda i} - w_i^{(1)})^2 \int_{-\infty}^{\infty} \psi_i^2(x)\chi_{\text{bias}}(x)\, dx.$$

$\chi_{\text{bias}}(x)$ is an indicator function which is one on all intervals with bias. The above value measures the participation of w_i in the bias. If M is a diagonal matrix with $M_{kk} = 1$ if the corresponding data point x_k has been marked as biased and $M_{kk} = 0$ otherwise, B_i can be computed as:

$$B_i = (w_{\lambda i} - w_i^{(1)})^2 (W^T (S_J M) W)_{ii}.$$

Marking the coefficients with the highest values gives results very close to the first selecting procedure.

5.4 The proposed algorithm

The objective of the algorithm is to combine the smooth reconstruction of a second-generation procedure with the reliable estimation of the classical transform. We call $\tilde{W}$ and W the forward and inverse second generation transform, as before, and $\tilde{U}$ and U are the transform matrices if we do not take into account the grid structure. The algorithm goes as follows:

1. Compute $\boldsymbol{w} = \tilde{W}\boldsymbol{y}$ and $\boldsymbol{u} = \tilde{U}\boldsymbol{y}$.

2. Compute the structure of the covariance matrix of the wavelet coefficients:

$$D = \tilde{W} C \tilde{W}^T,$$

   and similarly for $\tilde{U}$. C contains the covariance matrix of the input (up to constant; we do not use an estimate of the noise variance). We assume that the noise is homoscedastic and uncorrelated: $C = I$. In that case, the computation of D has linear complexity.

3. Normalize the coefficients with these variances and select for both sets of wavelet coefficients a threshold λ and μ, e.g. by minimizing $\mathrm{GCV}_{\boldsymbol{w}}(\lambda)$ and $\mathrm{GCV}_{\boldsymbol{u}}(\mu)$. And apply a soft-threshold to get the thresholded vectors $\boldsymbol{w_\lambda}$ and $\boldsymbol{u_\mu}$.

4. Compute $\boldsymbol{y^{(1)}} = U\boldsymbol{u_\mu}$ and $\boldsymbol{w^{(1)}} = \tilde{W}\boldsymbol{y^{(1)}}$. Use the not further transformed scaling coefficients in $\boldsymbol{w^{(1)}}$ as an estimate for the corresponding noise-free scaling coefficients. Replace the noisy scaling coefficients in $\boldsymbol{w_\lambda}$ by these values and compute $\boldsymbol{y_\lambda} = W\boldsymbol{w_\lambda}$.

5. Estimate the noise standard deviation by:

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i^{(1)})^2}.$$

   Among all data points $i = 1, \ldots, N$ we mark those for which $|y_{\lambda i} - y_i^{(1)}| > \hat{\sigma}$, $y_{\lambda i}$ as biased. We filter out isolated labels, since we consider bias as a more-than-one-point phenomenon. We mark all coefficients corresponding to basis functions on intervals with bias.

6. Compute the 2-norm of each basis function, these can be found in the diagonal of the matrix $W^T S_J W$. The computation of this diagonal is of linear complexity.

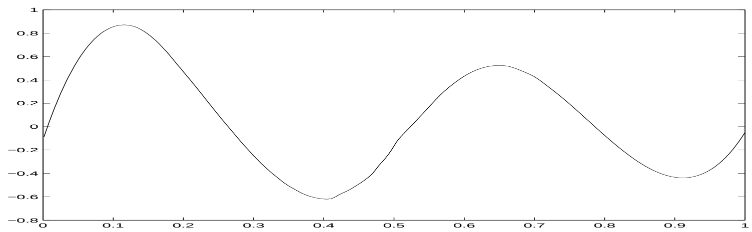

Figure 13: Result of the proposed algorithm. It is smooth and close to the noise-free signal.

7. Among all marked coefficients $w_{\lambda i}$, unmark those for which

$$(w_{\lambda i} - w_i^{(1)})^2 (W^T S_J W)_{ii}$$

is too small. Make sure that all scaling coefficients are marked.

8. For all coefficients $i = 1, \ldots, N$ select the appropriate value:

   (a) If a coefficient is marked, let: $\hat{w}_i = w_i^{(1)}$,

   (b) for the others, select: $\hat{w}_i = w_{\lambda i}$.

9. The output is:

$$\hat{\boldsymbol{y}} = W\hat{\boldsymbol{w}}.$$

This algorithm requires 3 forward and 3 inverse transforms, but the order of complexity is still linear. The computation of $W^T S_J W$ and $\tilde{W} C \tilde{W}^T$ are the most time consuming steps.

5.5 Results and discussion

Figure 13 contains a plot of the result of the proposed algorithm. It is smooth and close to the noise-free signal. Figure 14 focuses on a detail and illustrates the importance of the grid: if neglect the grid structure, the result is non-smooth. The wavelet transform used a cubic interpolation as prediction filter, followed by a two taps update-filter, designed to create the dual wavelets with two vanishing moments. The input signal had $N = 2048$ data points, and we leave 8 scaling coefficients untransformed, and so untouched by the threshold. For the reconstruction, only 18 from the 2048 wavelet coefficients, including the 8 scaling coefficients, were taken from $\boldsymbol{w}^{(1)}$, all the others were based on the thresholded second generation coefficients.

6 Conclusion

This paper discussed the application of second generation wavelets to reduce noise in irregularly spaced data. Bad conditioned transforms may cause problems, especially where the grid is very irregular and in the neighborhood of the boundaries. We have proposed an algorithm that starts from a reliable, but non-smooth solution based on a transform that neglects the grid structure. The algorithm then carefully selects coefficients that can be replaced by the

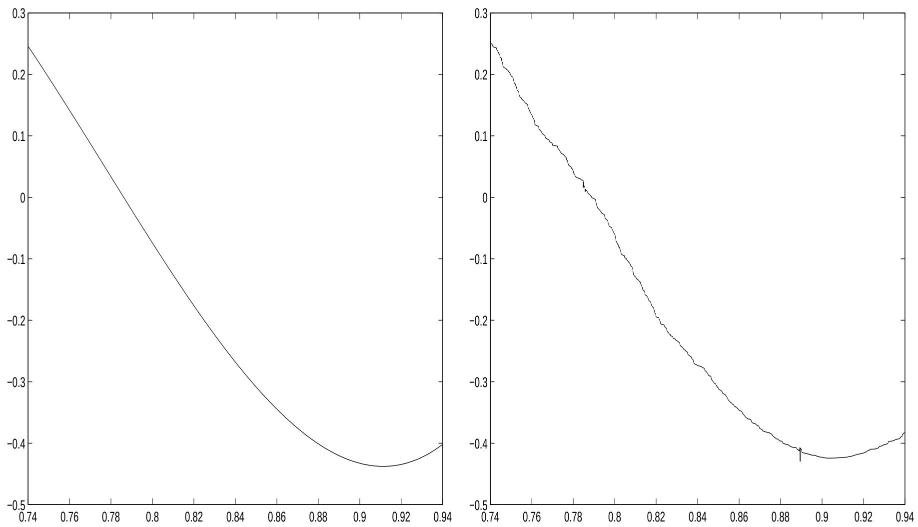

Figure 14: Left: detail of Figure 13. Right: the reconstruction of classical procedure (no grid structure) on the same interval. This reconstruction carries the irregularity of the grid.

more sparse representation obtained by directly thresholding second generation wavelet coefficients.

Acknowledgements

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors. The first author is financed by a grant from the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). The work of the second author was performed as part of the project "Orthogonal Systems and their applications" of the FWO (Fund for Scientific Research - Flanders, Belgium) under grant # G0278.97.

References

[1] T. Cai and L.D. Brown. Wavelet shrinkage for nonequispaced samples. *Annals of Statistics*, To appear, 1998.

[2] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[3] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[4] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[5] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[6] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.*, 90:1200–1224, 1995.

[7] P. Hall and B. A. Turlach. Interpolation methods for nonlinear wavelet regression with irregularly spaced design. *Annals of Statistics*, 25(5):1912 – 1925, 1997.

[8] M. Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for images with correlated noise. *IEEE Transactions on Image Processing*, page to appear, 1998.

[9] M. Jansen, M. Malfait, and A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56(1):33–44, January 1997.

[10] G Strang and T. Nguyen. *Wavelets and filter banks.* Wellesley-Cambridge Press, Box 812060, Wellesley MA 02181, fax 617-253-4358, 1996.

[11] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[12] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2), 1997.

[13] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes. ACM, 1996.

[14] G. Wahba. *Spline Models for Observational Data*, chapter 4, pages 45–65. CBMS-NSF Regional Conf. Series in Appl. Math. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[15] N. Weyrich and G. T. Warhola. De-noising using wavelets and cross validation. In S.P. Singh, editor, *Approximation Theory, Wavelets and Applications*, volume 454 of *NATO ASI Series C*, pages 523–532, 1995.