

# Hysteresis thresholding: a graph-based wavelet block denoising algorithm

## Abstract

This communication aims to combine several previously proposed wavelet denoising algorithms into a novel heuristic block method. The proposed “hysteresis” thresholding uses two thresholds simultaneously in order to combine detection and minimal alteration of informative features of the processed signal. This approach exploits the graph structure of the wavelet decomposition to detect clusters of significant wavelet coefficients. The new algorithm is compared with classical denoising methods on simulated benchmark signals.

**wavelets, block denoising, graph structure, transitive closure**

## I. INTRODUCTION

One of the most important applications of wavelets is the signal denoising. The fundamental hypothesis is that wavelets are correlated with the informative signal and uncorrelated with the noise, which globally means that large absolute value coefficients correspond to signal and small coefficients to noise. Therefore, noise cancelling (as well as compression) can be performed by thresholding: small coefficients will be discarded and the few remaining large magnitude coefficients will be used to reconstruct the informative signal. Several approaches were proposed since the pioneering works of Donoho *et al.* [1, 2] (see Antoniadis *et al.* for a comprehensive review [3, 4]). In this communication, we propose to combine previously introduced thresholds in a novel block-denoising approach that takes into account the local wavelet graph structure.

This paper is organized as follows. In the next section, we recall the most critical issues in wavelet denoising and some of the proposed solutions, emphasizing the methods at the origin of our work. Section 3 introduces the graph-based approach and the new “hysteresis” thresholding. The fourth section presents and discusses the simulation results, comparing them with classic denoising algorithms, and it is followed by a short conclusion.

## II. WAVELET DENOISING

We consider the model  $z = x + n$ , where  $z$  is the noisy discrete-time signal (length  $N$ ),  $x$  is the noise-free unknown version of  $z$  and  $n$  the noise. Synthetically, the discrete orthogonal wavelet transform (DWT) of  $z$  writes:

$$z = \sum_{p,j} w_z^{j,p} \psi^{j,p} + \sum_p w_z^{M,p} \phi^{M,p}, \quad (1)$$

where  $j = [1..M]$  is the scale,  $p = [1..2^{-M}N]$  the position,  $\psi$  the wavelet,  $\phi$  the scaling function and  $M$  is the analysis depth [5]. The wavelet coefficients vectors of  $z$ ,  $x$  and  $n$  are related by:

$$w_z = w_x + w_n \quad (2)$$

The denoised signal's coefficients vector  $\hat{w}_x$  is estimated by modifying  $w_z = [w_z^{M,p} w_z^{j,p}]$  (generally, the approximation coefficients  $w_z^{M,p}$  remain unchanged). Two main families of approaches are proposed in the literature: term-by-term and block approaches. In the former, the coefficients of  $\hat{w}_x$  estimate write as:

$$\hat{w}_x = g(w_z)w_z,$$

with  $g(w_z)$  a shrinkage function applied on the measured signal coefficients.

In the second case (block thresholding), the shrinkage function depends not only on the currently evaluated coefficient, but also on a group or block  $B$  neighboring coefficients  $w_{z,b}$  (the  $b$  subscript indicates the adjacent coefficients from the considered block  $B$ ). Moreover, the shrinkage can be applied for the whole considered block  $w_{z,b}$  or for a given subset  $w_{z,b1} \in w_{z,b}$  (in practical implementations [13], the size of block  $b$  is 3 and  $w_{z,b1}$  is the middle element):

$$\hat{w}_{x,b1} = g(w_{z,b})w_{z,b1}.$$

Several algorithms from these two families are briefly presented next. They are integrated in the novel hysteresis algorithm and/or used for comparison.

### A. Term-by-term shrinkage

As the noise-free coefficient vector is presumably sparse, the simplest solution is provided by the well-known hard shrinkage (thresholding):

$$g(w_z) = \max(0, \text{sign}(|w_z| - T)), \quad (3)$$

with  $T$  the threshold value. The most widely used value for  $T$  is the universal threshold proposed  $T_U$  by Donoho and Johnstone in their algorithm *VisuShrink* (designated as *Visu* in the sequel) [1]. The goal of *Visu* is the *complete asymptotic elimination of the normal Gaussian noise* and it can be shown (using extreme values statistics) that this is achieved by setting  $T_U = \sqrt{2 \log N}$ . The main drawback of the method (very appealing because of its simplicity and of its visually attractive results) is its focus on the noise: eliminating *all* the noise often leads to a less precise reconstruction of the signal of interest.

A different approach is proposed by the *Sure* algorithm [2], who aims to *estimate as precisely as possible the “clean” signal* by minimizing an estimate of the mean squared error (*MSE*) between the denoised signal and the original one, known as the *Stein Unbiased Risk Estimator*. This risk is minimized by exhaustively searching the optimal threshold among the coefficients  $w_z$ . The algorithm was developed under specific conditions on the function  $g$  (weak differentiability). In particular, the most widely used is the soft thresholding function:

$$g(w_z) = \max \left( 0, \frac{|w_z| - T}{|w_z|} \right) \quad (4)$$

The obtained threshold  $T_S$  (or thresholds, as the method is usually implemented by scale) are lower than the  $T_U$  and the obtained signal has a noisier appearance.

The iterative method we proposed in [6, 7] follows a different approach, inspired by [8–10]: the threshold  $T_M$  is obtained by a fast fixed-point parameter-free algorithm based on a generalized Gaussian (GG) modelling of the wavelet coefficients. The resulting “minimal denoising” (called hereafter *MinD*) ensures robust detection of high value coefficients (outliers) and thus leads to maximum information extraction from the measured signal.

As the universal threshold proposed by *Visu*, the threshold computed by *MinD* does not impose specific characteristics on the function  $g(\cdot)$ . Therefore, both methods can be applied either with hard (3) or soft (4) shrinkage.

### B. Block shrinkage

A quite natural hypothesis (most of the times verified in applications) is that large wavelet coefficients appear in clusters, or blocks. The approaches introduced by Hall, Kerkyacharian and Picard [11] and Cai *et al.* [12–14] aim to take into account local features of the signal by adapting the thresholds by blocks of neighboring coefficients. The most well-known methods are *BlockJS* (*BJS*) and *NeighBlock*

(*NB*) (with its simpler version *NeighCoef*) [12, 13]. The two algorithms propose a similar philosophy as *Sure*: estimate as precisely as possible a signal corrupted by noise by minimizing the James-Stein estimate of the *MSE* risk. This minimization leads to a particular form of the shrinkage function  $g(\cdot)$ , which applies to all coefficients  $\mathbf{w}_{z,b}$  in a given block of size  $L$ :

$$g(\mathbf{w}_{z,b}) = \max\left(0, \frac{\|\mathbf{w}_{z,b}\|^2 - T_L}{\|\mathbf{w}_{z,b}\|^2}\right), \quad (5)$$

with  $\|\mathbf{x}_b\|^2$  the energy of the block and  $T_L$  a block-size depending threshold. For *BJS*,  $T_L = \lambda L \sigma^2$ , with  $\lambda = 4.505\dots$ ,  $L = \log N$  ( $N$  being the signal length) and  $\sigma$  the estimated noise power (by scale or global, for colored and white noise respectively). The difference between *BJS* and *NB* consists in the use of the function  $g(\cdot)$ : while in *BJS* all coefficients in the block are shrunk by the same amount ( $g(\mathbf{w}_{z,b})$ ), in *NB* this is done only for the  $L/2$  coefficients located in the center of the considered block. In this case, the blocks will not be disjoint but overlapped by  $L/4$  (see [13] for more details). The neighborhood proposed in these approaches is constructed scale-by-scale, without inter-scale dependencies. Extensions taking into account this last item were proposed recently by [15, 16]<sup>1</sup>.

### III. GRAPH-BASED HYSTERESIS APPROACH

As mentioned previously, most of the wavelet denoising algorithms treat the coefficients either individually or in blocks grouped by scale. These approaches don't take into account more general neighborhoods, spanned over scales as well as in time. This point is easily captured by the graph structure of the DWT.

#### A. Graph structures

Currently, the most complete investigation of the graph structure of the DWT (implicitly addressing both inter- and intra-scale relations between coefficients) is done by Crouse *et al.* [20]. The authors (who attach a Hidden Markov Model (HMM) to the graph to compute the thresholds) propose three different architectures and thus types of neighborhoods: the Independent Mixture IM (*i.e.*, the neighboring coefficients are not considered in the procedure), the Hidden Markov Chain HMC (*i.e.*, considering dependencies by scale) and the Hidden Markov Tree HMT (binary trees resulting from the wavelet decomposition). A graphical representation is given figure 1.

<sup>1</sup>Bayesian block denoising methods (for example [17–19]) are not treated here.

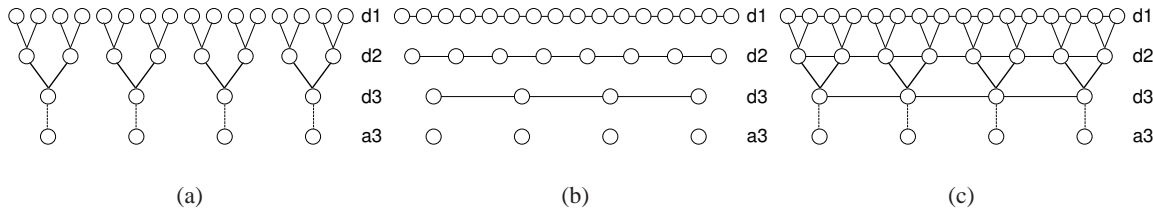


Fig. 1. Graph types: (a) tree; (b) scale; (c) complete.

The graph representation emphasizes two important properties of real signals, *i.e.*, **persistence** and **clustering**. Following [20], we define by persistence the fact that “large/small values of wavelet coefficients tend to propagate across scales” and by clustering the fact that “if a particular wavelet coefficients is large/small, then adjacent coefficients are very likely to also be large/small”. More precisely, the wavelet binary tree at a given position (time interval) permits to explore persistence (figure 1(a)), while the chains of wavelet coefficients situated on the same scale (frequency band) illustrate clustering (figure 1(b)). If both persistence and clustering are considered, than a complete forest-like graph must be taken into account (figure 1(c)).

A more formal representation of graphs is given by the adjacency matrix  $\mathbf{A}$ . For a graph having  $N$  nodes (*i.e.*,  $N$  wavelet coefficients),  $\mathbf{A}$  is a square  $N \times N$  boolean matrix with elements  $A(i, j) = 1$  if there is an edge between nodes  $i$  and  $j$  and 0 elsewhere. The obtained matrix is symmetric, and the non-zero elements on line (column)  $i$  correspond to the coefficients connected to element  $w_z(i)$  of the wavelet coefficients vector  $\mathbf{w}_z$ . By convention, we consider every node connected to itself (the diagonal of  $\mathbf{A}$  is filled with 1).

### B. Pruned graphs

The three term-by-term denoising methods presented in the previous section have different rationales: the universal thresholding *Visu* uses a high threshold aiming to *select only* informative coefficients and *eliminate all* noise, while *Sure* and *MinD* use a low threshold to *eliminate only* noise and therefore to *select all* the signal. A heuristic combination can be then proposed: use a *Visu* computed high threshold  $T_h = T_U$  to select *blocks* of significant wavelet coefficients, and a *Sure* or *MinD* low threshold  $T_l = T_S(T_M)$  to fix the limits of the selected blocks. In the graph structure, the neighborhood of a “very large” coefficient  $|w_z| > T_h$  selected by *Visu* will be formed by the connected “large enough” coefficients selected by *Sure* or *MinD* ( $|w_z| > T_l$ ). These coefficients are called further-on *max-selected* and *min-*

*selected* respectively. The resulting blocks, without *a priori* predefined shapes, will naturally integrate both persistence and clustering properties of real signals.

More formally, the issue of choosing the connected min-selected coefficients is equivalent to a three-step pruning of the wavelet graph, and we give further-on a possible procedure (see also figure 2 for an illustrative example):

- 1) For the considered graph adjacency matrix  $A$  (corresponding to graph 2(a)), keep only the 1's corresponding to min-selected elements (marked nodes) by forcing other lines and columns to 0:

$$A(i, \cdot) = A(\cdot, i) = 0, \quad \forall |w_z(i)| < T_l \quad (6)$$

Note the newly obtained adjacency matrix  $A_1$ . The resulting graph (figure 2(b)) is a collection of disjoint blocks (clusters) of min-selected coefficients. For a given cluster, we note  $B$  the set of the indices  $b$  of the coefficients belonging to it,  $|w_z(b)| \geq T_l$ .

- 2) Apply a transitive closure procedure on the pruned graph. Basically, a transitive closure leads to a modified graph that connects by a direct edge all nodes connected through multiple nodes/edges: if there is a path between two nodes, than a new edge between them is added to the graph. This approach leads to an iterative computation of a new adjacency matrix, known as Warshall's algorithm: at each step  $l$ ,

$$A_{1,l}(\alpha, \beta) = A_{1,l-1}(\alpha, \beta) \vee \left( A_{1,l-1}(\alpha, \gamma) \wedge A_{1,l-1}(\gamma, \beta) \right) \quad (7)$$

At convergence, all the nodes in a cluster  $B$  will be connected by an edge (figure 2(c)). Consequently, if the nodes  $b_i$  and  $b_j$  belong to the cluster  $B$ , the elements of the obtained adjacency matrix  $A_2$  are  $A_2(b_i, b_j) = A_2(b_j, b_i) = 1, \quad \forall b_i, b_j \in B$ . An implementation issue must be noted here: a complete iterative transitive closure (7) may lead to a very slow and memory consuming algorithm, especially for long and not very sparse signals (large  $N$ ). This problem depends on the chosen graph structure: tree-type graphs (figure 1(a)) have a limited number of possible connections depending on the wavelet decomposition depth, while scale-type (figure 1(b)) or complete graphs (figure 1(c)) are only limited by the length of the signal. Therefore, the maximum number of iterations should be limited by the user. We propose here to limit them to the maximum number of levels of the wavelet decomposition. Consequently, the number of edges between a max- and a min-selected node (and thus the size of the resulting blocks) is bounded by the depth of the wavelet tree;

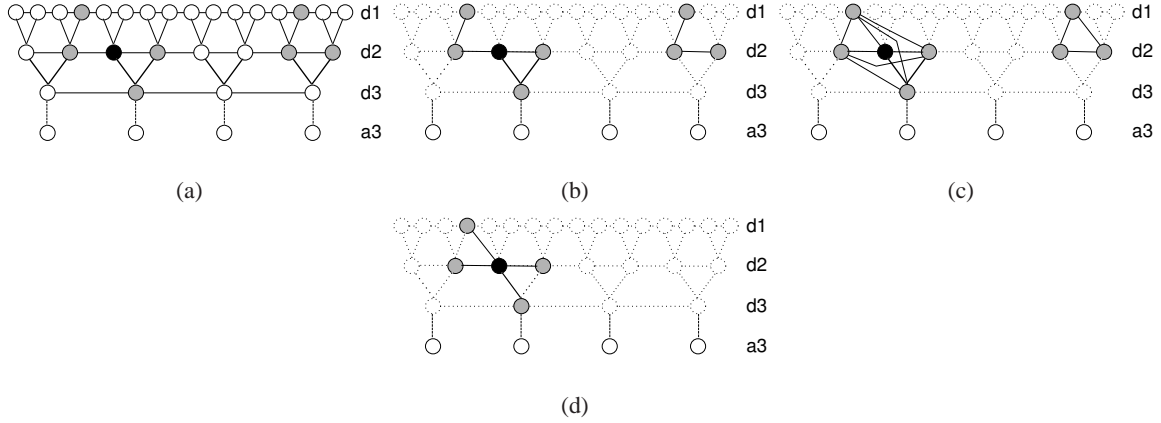


Fig. 2. Pruning algorithm: (a) Original graph: max-selected and min-selected coefficients are figured in black and gray respectively; (b) First pruning: discard the small values; (c) Transitive closure; (d) Second pruning and final result. The approximation scale is not affected.

- 3) Construct a final adjacency matrix  $\mathbf{A}_3$  by selecting the blocks containing at least one max-selected coefficient, *i.e.*, only the lines (columns)  $i$  of  $\mathbf{A}_2$  corresponding to the coefficients  $|w_z(i)| \geq T_h$  (figure 2(d)):

$$A_2(i, \cdot) = A_2(\cdot, i) = 0, \quad \forall |w_z(i)| < T_h. \quad (8)$$

The non-null lines (columns) of this  $\mathbf{A}_3$  correspond to the wavelet coefficients kept for the reconstruction.

The final denoising algorithm will have the following structure:

- 1) Wavelet transform the measured signal.
- 2) Choose the graph type (scale, tree or complete).
- 3) Apply *Sure* or *MinD* (global or scale by scale) to mark min-selected coefficients and the corresponding nodes of the graph.
- 4) Apply *Visu* to mark max-selected coefficients and nodes.
- 5) Prune according to the given procedure the chosen graph.
- 6) Reconstruct by inverse wavelet transform.

#### IV. RESULTS AND DISCUSSION

Several versions of the hysteresis approach were implemented:

- considering either *Sure* (noted *HyVS*) or *MinD* (noted *HyVM*) algorithms for the low threshold (*Visu* for the high threshold);
- considering the three types of graph-structures (by scale, subscript  $s$ , by binary tree, subscript  $t$  and the complete graph, subscript  $c$ );
- adapted to white or colored noise (white: the noise power is estimated on the first details level for *Visu* and *Sure*, fixed point iterations are performed on all coefficients for *MinD*; colored: scale by scale noise estimation for *Visu* and *Sure*, by-scale fixed-point descent for *MinD*);
- considering soft (4) or hard (3) thresholding.

The algorithms were compared (mean squared error *MSE*) to classical term-by-term *Visu*, *Sure* and *MinD*, as well as to the block methods *BJS* and *NB* (adapted from the scripts of the free toolbox of Antoniadis *et al.* [21]).

Concerning the thresholding strategy, it must be mentioned here that *Sure*, *BJS* and *NB* were specifically developed for soft-type shrinkage functions  $g(\cdot)$ , while *Visu* and *MinD* only compute threshold values and the user can choose the shrinkage strategy. The results we present here in details concern only the soft thresholding, *i.e.*, the low threshold value (given by *Sure* or by *MinD*) was subtracted from the selected coefficients. Hard thresholding particularities will be briefly exposed at the end of the section. The approximation scale was kept unchanged and was not considered in the graph construction (for sparse signals, the results might be improved if the approximation is thresholded also).

The tested signals are the usual *Bumps*, *Blocks*, *HeaviSine* and *Doppler* [1], with lengths from 512 to 16384. Noise was added, with signal to noise ratios (SNR) of 3, 5 and 7. Three types of noise were used: Gaussian white noise, low-pass filtered Gaussian noise ( $4^{th}$  order Butterworth filter, normalized cut-off frequency at 0.25) and high-pass filtered Gaussian noise ( $4^{th}$  order Butterworth filter, normalized cut-off frequency at 0.25).

Twenty simulations were performed for each combination of signal type, length and SNR. For the white noise case, the most significant numerical results are presented in table I. These results are averaged by signal type, signal length and signal to noise ratio, in order to illustrate independently the influence of these parameters on the algorithms' performances.

Afig:graphsss it can be seen table I, the signal type has a certain but limited influence on the performances: for the first 2 signals (*Blocks* and *Bumps*), the novel hysteresis approach based on universal and SURE thresholds and applied on a complete-type graph (*HyVS<sub>c</sub>*) performs slightly better than classic



		<i>Visu</i>	<i>Sure</i>	<i>MinD</i>	<i>BJS</i>	<i>NB</i>	<i>HyVS<sub>s</sub></i>	<i>HyVS<sub>t</sub></i>	<i>HyVS<sub>c</sub></i>	<i>HyVM<sub>c</sub></i>
Signal	<i>Blocks</i>	0.4786	0.1999	0.4435	0.3051	0.3313	0.2304	0.2030	<b>0.1852</b>	0.4465
	<i>Bumps</i>	0.6512	0.1772	0.6400	0.2460	0.2904	0.1790	0.1772	<b>0.1615</b>	0.6399
	<i>Heavisine</i>	0.0437	0.0450	0.0421	0.0433	0.0443	0.0408	0.0399	<b>0.0396</b>	0.0410
	<i>Doppler</i>	0.1596	0.0797	0.1479	<b>0.0618</b>	0.0752	0.0665	0.0731	0.0663	0.1473
Length	512	0.8813	0.2721	0.9266	0.4062	0.4564	0.2920	0.2730	<b>0.2527</b>	0.9275
	1024	0.5192	0.1866	0.4973	0.2304	0.2630	0.1914	0.1797	<b>0.1657</b>	0.4980
	2048	0.2988	0.1209	0.2516	0.1414	0.1750	0.1223	0.1173	<b>0.1057</b>	0.2534
	4096	0.1580	0.0783	0.1226	0.0975	0.1015	0.0755	0.0750	<b>0.0674</b>	0.1230
	8192	0.0894	0.0543	0.0682	0.0641	0.0689	0.0538	0.0550	<b>0.0494</b>	0.0680
	16386	0.0530	0.0405	0.0440	0.0447	0.0471	0.0400	0.0399	<b>0.0380</b>	0.0423
SNR	3	0.5311	0.2238	0.4829	0.2942	0.3269	0.2335	0.2216	<b>0.2031</b>	0.4835
	5	0.2857	0.0974	0.2825	0.1277	0.1461	0.0997	0.0953	<b>0.0874</b>	0.2828
	7	0.1831	0.0551	0.1897	0.0702	0.0829	0.0544	0.0531	<b>0.0490</b>	0.1898

TABLE I

MSE FOR WHITE NOISE CASE COMPARISON. RESULTS ARE PRESENTED BY SIGNAL TYPE, BY SIGNAL LENGTH AND BY SNR. THE ALGORITHMS USED A UNIQUE NOISE ESTIMATION (HIGHEST LEVEL FOR *Visu*, *Sure*, *BJS* AND *NB*, RESPECTIVELY A GLOBAL APPROACH FOR *MinD*). THE RESULTS ARE DISPLAYED FOR THE BEST VERSIONS OF THE HYSTERESIS ALGORITHM (*HyVS* FOR *Sure* LOW THRESHOLD, *HyVM* FOR *MinD* LOW THRESHOLD, SUBSCRIPTS *t*, *s*, *c* STANDING FOR TREE, SCALE AND COMPLETE GRAPH TYPE RESPECTIVELY).

*SureShrink*. The other two graph types (scale *HyVS<sub>s</sub>* and tree *HyVS<sub>t</sub>*) follow closely, with a rather important gap between them and the block-approaches *BJS* and *NB*. On *Heavisine*, all hysteresis type approaches perform better than the other algorithms (*HyVM<sub>c</sub>* included), while *Sure* shows the worst performances (*MinD* being better than the other term-by-term thresholding methods, as well as than the tested block approaches). On the contrary, on the frequency rich *Doppler*, *BJS* shows the best performance, followed by the three *HyVS* algorithms. The ratios of improvement vary: *HyVS<sub>c</sub>* outperforms *Sure* and *BJS* by mean squares errors smaller by 8 to 40% for *Bumps*, *Blocks* and *HeaviSine* while *BJS* is better by about 8% for the *Doppler* signal.

The signal length  $N$  plays an important role in the algorithms performances. All of them improve when  $N$  increases, the amelioration being more marked for asymptotic procedures as *VisuShrink* and for probability density estimation methods like *MinD*. Globally, *HyVS* algorithm (for the three graph types) shows very good performances regardless of the signal length, compared to the others, with the complete graph approach *HyVS<sub>c</sub>* being constantly the most efficient. *SureShrink* is the next in line, followed by the block-thresholding methods *BJS* and *NB*. For the longest tested signals ( $N = 16386$ ), minimal denoising based methods *HyVM<sub>c</sub>* and *MinD* outperform these last two algorithms (*BJS* and

*NB*). On the other hand, the computational burden and thus the execution time increases differently with  $N$ . If for 2048 samples signals,  $HyVS_c$  and  $HyVM_c$  are slightly faster than  $BJS$  (but slower than  $Sure$ ), for 16386 length signals the difference becomes important: while  $BJS$  is only 1.5 times slower, both hysteresis methods slow their execution about 40 times. This augmentation is due to the fact that complete  $N \times N$  graph adjacency matrices (although sparse) are stored and manipulated in our implementation, but more optimal programming solutions can be found by taking into account the proposed bound for the block size (depth of the wavelet decomposition).

As for the influence of the noise power, all algorithms are affected in the same way and their order remains generally unchanged:  $HyVS_c$  shows constantly the best performances, while the other algorithms from the same family (tree based and scale based graphs) and  $SureShrink$  come next.

For the low-frequency and high-frequency noise cases, we tested versions of the described algorithms adapted to colored noise (scale by scale estimation of  $\sigma$  for  $VisuShrink$  and  $SureShrink$  based methods, respectively scale by scale probability density estimation and thresholding for  $MinD$  based methods). The relative performances of the algorithms are quite similar most of the time, therefore the complete numerical results are not given here. Still, some particular points must be outlined:

- the performances of minimal denoising based algorithms ( $MinD$  and  $HyVM$ ) degrade sharply when the signal length  $N$  is low, as the estimation of the generalized Gaussian (GG) parameters becomes problematic. Therefore, their use is not recommended, unless the signal length is important and/or the coefficient distribution is close to a GG;
- for low-frequency noise, scale adapted denoising tend to do more harm than good: the noise power might be over-estimated on the low frequency scales and the resulting thresholds might be too high, degrading the signal. This is especially true when the SNR is low, when the algorithms using a unique noise estimation perform better than the scale-adapted ones.

Finally, we should note that the comparative results obtained for hard-thresholding confirm the better performances of the hysteresis algorithms. As expected, the main challengers among term by term thresholding are  $VisuShrink$  and  $MinD$ ,  $SureShrink$  being designed for soft-thresholding. Among block denoising approaches, both  $BlockJS$  and  $NeighBlock$  perform better than term-by-term methods, but they are outperformed by hysteresis implementations, either  $HyVS$  or  $HyVM$ . Between these two families,  $Sure$  based  $HyVS$  gives better results than minimal denoising based  $HyVM$ .

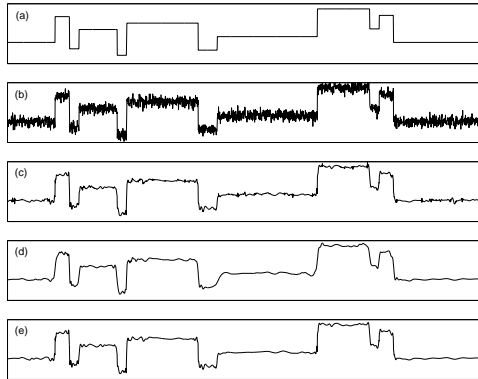


Fig. 3. Denoising example. White noise was added to a 2048 points *Blocks* signal (SNR=3): (a) original signal; (b) noisy signal; (c) *SureShrink* denoising (MSE=0.1514); (d) *BlockJS* denoising (MSE=0.2354); (e) *HyVSc* denoising (MSE=0.1382).

Denoising examples for the best algorithm of each family (term-by-term, classical block-thresholding and hysteresis thresholding) are given figure 3.

## V. CONCLUSION AND FUTURE RESEARCH

In this paper, we present a new heuristic method for wavelet denoising. Our approach, which can be considered a block-denoising type method, is based on the graph structure of the orthogonal wavelet decomposition. We propose to combine previously developed algorithms to modify the graph structure of the wavelet decomposition. This two thresholds approach, called “hysteresis denoising”, aims to take into account the property of real signals to have wavelet coefficients grouped in clusters, spanned both over time and over scales. Therefore, a high valued threshold is implemented to select only informative signal coefficients (*i.e.*, to eliminate asymptotically all the noise), while a low threshold is used to define neighborhoods of the selected coefficients (clusters, or blocks) on which the signal of interest is accurately estimated. The newly proposed method is compared successfully with other wavelet denoising algorithms, which it outperforms most of the time.

By its nature, the hysteresis thresholding is adapted to the detection and the denoising of isolated transients in long time series (*i.e.*, very sparse signals). The validation of this hypothesis, along with some new improvements on the minimal denoising method (itself adapted to transient detection) and with more extensive comparisons (for example with the recently proposed *SureBlock* shrinkage [14], which shows rather similar results) are currently under study and will make the subject of a future work.

## VI. APPENDIX

The proposed hysteresis approach is based on the following thresholds:

- high threshold (*Visu* universal threshold):

$$T_H = T_U = \sigma \sqrt{2 \log N},$$

with  $N$  the signal length and  $\sigma$  the noise' (robust) standard deviation;

- low threshold for *HyVS* (*Sure* threshold):

$$T_L = T_S = \arg \min_{T \in \{w_z(k)\}} \left( N\sigma^2 + T^2 P + \sum_{\substack{k \\ |w_z(k)| < T}} [w_z(k)^2] - 2(N - P)\sigma^2 \right)$$

with  $P$  the number of wavelet coefficients having absolute values  $\geq T$ ;

- low threshold for *HyVM* (*MinD* threshold)  $T_L = T_M$  obtained at convergence of the following fixed-point iteration:

$$T_{i+1} = F_{am} \sqrt{\frac{1}{N} \sum_k [w_z(k) \cdot \max(0, \text{sign}(|w_z(k)| - T_i))]^2}, \quad \text{with}$$

$$F_{am} = \sqrt{\frac{3\Gamma(\frac{1}{u})}{u}} (ue)^{\frac{1}{u}}$$

For the  $F_{am}$  value, the parameter  $u$  is the estimated shape parameter of the generalized gaussian distribution of the wavelet coefficients:

$$p_{\sigma,u}(w) = \alpha e^{-|\beta w|^u} \quad \text{with}$$

$$\beta = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/u)}{\Gamma(1/u)}}, \quad \alpha = \frac{\beta u}{2\Gamma(1/u)}, \quad \Gamma(u) = \int_0^\infty e^{-x} x^{u-1} dx,$$

All thresholds can be computed globally (regardless of the scale) or scale by scale, to deal with colored noise / signal (see cited references [1, 5, 7] for details).

## REFERENCES

- [1] D. Donoho, I. Johnstone, Ideal spatial adaptation via wavelet shrinkage, *Biometrika* 81 (1994) 425–455.
- [2] D. Donoho, I. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, *Journal of the American Statistical Association* 90 (1995) 1200–1224.

- [3] A. Antoniadis, J. Bigot, T. Sapatinas, Wavelet estimators in nonparametric regression: a comparative simulation study, *Journal of Statistical Software* 6 (6) (2001) 1–83.
- [4] A. Antoniadis, Wavelet methods in statistics: Some recent developments and their applications, *Statistics Surveys* 1 (2007) 16–55.
- [5] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1999.
- [6] R. Ranta, C. Heinrich, V. Louis-Dorr, D. Wolf, Interpretation and improvement of an iterative wavelet-based denoising method, *IEEE Signal Process. Lett.* 10 (8) (2003) 239–241.
- [7] R. Ranta, V. Louis-Dorr, C. Heinrich, D. Wolf, Iterative wavelet-based denoising methods and robust outlier detection, *IEEE Signal Process. Lett.* 12 (8) (2005) 557–560.
- [8] J.-L. Starck, F. Murtagh, Image restoration using wavelet transform, *Astronomy and Astrophysics* 288 (1994) 342–348.
- [9] R. Coifman, M. Wickerhauser, Experiments with adapted wavelet de-noising for medical signals and images, in: M. Akay (Ed.), *Time-Frequency and Wavelets in Biomedical Engineering*, IEEE Press, 1998, pp. 323–346.
- [10] L. Hadjileontiadis, L. Liatsos, C. Mavrogiannis, T. Rokkas, S. Panas, Enhancement of bowel sounds by wavelet-based filtering, *IEEE Trans. Biomed. Eng.* 47 (7) (2000) 876–886.
- [11] P. Hall, G. Kerkycharian, D. Picard, Block threshold rules for curve estimation using kernel and wavelet methods, *The Annals of Statistics* 26 (3) (1998) 922–942.
- [12] T. Cai, Adaptive wavelet estimation: a block thresholding and oracle inequality approach, *The Annals of Statistics* 27 (3) (1999) 898–924.
- [13] T. Cai, B. Silverman, Incorporating information on neighbouring coefficients into wavelet estimation, *Sankhyā: The Indian Journal of Statistics. Special issue on Wavelets* 63 (2) (2001) 127–148.
- [14] T. Cai, H. Zhou, A data-driven block thresholding approach to wavelet estimation, *The Annals of Statistics* to appear.  
URL <http://www-stat.wharton.upenn.edu/~tcai/Papers.html>
- [15] L. Zhen, H. Zhengjia, Z. Yanyang, W. Yanxue, Customized wavelet denoising using intra- and inter-scale dependency for bearing fault detection, *Journal of Sound and Vibration* 313 (1–2) (2008) 342–359.
- [16] F. Luisier, T. Blu, M. Unser, A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding, *IEEE Trans. Image Process.* 16 (3) (2007) 593–606.

- [17] J. Liu, P. Moulin, Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients, *IEEE Trans. Image Process.* 10 (10) (2001) 1647–1658.
- [18] L. Sendur, I. Selesnick, Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency, *IEEE Trans. Signal Process.* 50 (11) (2002) 2744–2756.
- [19] A. Pizurica, W. Philips, Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising, *IEEE Trans. Image Process.* 15 (3) (2006) 654–665.
- [20] M. Crouse, R. Nowak, R. Baraniuk, Wavelet-based statistical signal processing using hidden markov models, *IEEE Trans. Signal Process.* 46 (4) (1998) 886–902.
- [21] A. Antoniadis, J. Bigot, T. Sapatinas, Wavelet denoising. Procedures in Matlab.  
URL <http://www-lmc.imag.fr/SMS/software/GaussianWaveDen/index.html>