A. Enis Cetin and
Mohammad Tofighi

Projection-Based Wavelet Denoising

In this lecture note, we describe a wavelet domain denoising method consisting of making orthogonal projections of wavelet (subbands) signals of the noisy signal onto an upside down pyramid-shaped region in a multidimensional space. Each horizontal slice of the upside down pyramid is a diamond shaped region and it is called an ℓ_1-ball. The upside down pyramid is called the epigraph set of the ℓ_1-norm cost function. We show that the method leads to soft-thresholding as in standard wavelet denoising methods. Orthogonal projection operations automatically determine the soft-threshold values of the wavelet signals.

PREREQUISITES

Prerequisites for understanding the material of this article are linear algebra, discrete-time signal processing, and wavelets. Orthogonal projection of a vector onto a hyperplane is the key mathematical operation used in this lecture note. Let $\mathbf{w}_o$ be a vector in $\mathbb{R}^K$. The orthogonal projection $\mathbf{w}_{po}$ of $\mathbf{w}_o$ onto the hyperplane $h = \mathbf{a}^T\mathbf{w} = \sum_{n=1}^{K} \mathbf{a}[n]\mathbf{w}[n]$ is given by

$$\mathbf{w}_{po}[n] = \mathbf{w}_o[n] + \frac{h - \sum_{n=1}^{K} \mathbf{a}[n]\mathbf{w}_o[n]}{\|\mathbf{a}\|_2^2}\mathbf{a}[n] \quad n = 1,2,\ldots,K, \tag{1}$$

where $\mathbf{w}_o[n]$, $\mathbf{w}_{po}[n]$, and $\mathbf{a}[n]$ are the nth entries of the vectors $\mathbf{w}_o$, $\mathbf{w}_{po}$, and $\mathbf{a}$, respectively, and $\|\mathbf{a}\|_2$ is the Euclidean length (norm) of the vector $\mathbf{a}$.

In this lecture note, orthogonal projections onto an upside down-shaped pyramid are computed. Each face of the upside down pyramid is a wedge-shaped subset of a hyperplane. Therefore, we can make an orthogonal projection onto an upside down pyramid by performing an orthogonal projection onto a face of the pyramid.

Orthogonal projection onto a hyperplane is also routinely used in the well-known normalized least mean squares (NLMS) adaptive filtering algorithm and many online learning algorithms [1].

PROBLEM STATEMENT

Denoising refers to the process of reducing noise in a given signal, image, and video. In standard wavelet denoising, a signal corrupted by additive noise is transformed to the wavelet domain and the resulting wavelet signals are soft- or hard-thresholded. After this step, the denoised signal is reconstructed from the thresholded wavelet signals [2], [3]. Thresholding wavelet coefficients intuitively makes sense because wavelet signals obtained from an orthogonal or biorthogonal wavelet filter bank exhibit large amplitude coefficients only around edges or jumps of the original signal. The assumption is that other small amplitude wavelet coefficients should be due to noise. Signals that can be represented with a small number of coefficients are called sparse signals and it turns out that most natural signals are sparse in some transfer domain [4], [5]. A wide range of wavelet denoising methods that take advantage of the sparse nature of practical signals in wavelet domain are developed using this baseline denoising idea by Donoho and Johnstone; see, e.g., [2]–[4] and [6]–[9].

Consider the following basic denoising framework. Let $\mathbf{v}$ be a discrete-time signal and $\mathbf{x}$ be its noisy version, i.e., $\mathbf{x}[n] = \mathbf{v}[n] + \boldsymbol{\xi}[n], n = 1,2,\ldots,N$, where $\boldsymbol{\xi}$ is some additive, independent and identically distributed (i.i.d.), zero-mean, white Gaussian noise with variance σ^2. An L-level discrete wavelet transform of $\mathbf{x}$ is computed and the lowband signal $\mathbf{x}_L$ and wavelet signals $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$ are obtained. After this step, wavelet signals are soft-thresholded as shown in Figure 1. The soft-threshold value, θ, can be selected in many ways using statistical methods [3], [4], [6], [10]. Donoho proposed the following threshold for all wavelet signals:

$$\theta = \gamma.\sigma.\sqrt{2\log(N)/N}, \tag{2}$$

where N is the number of samples of the signal $\mathbf{x}$, and the constant γ is defined in [3]. In (2), the noise variance σ^2 has to be known or properly estimated from the observations, $\mathbf{x}$, which may be difficult to achieve in practice. In [3], a single threshold is used for all wavelet signals. We refer the reader to [3], [4], [6], and [10] for many ways of estimating the parameters γ and σ in Donoho's method.

It is possible to define a soft-threshold θ_i for each wavelet signal $\mathbf{w}_i$. Here we present how to estimate a soft-threshold value θ_i for each wavelet signal $\mathbf{w}_i$ using a deterministic approach based on linear algebra and orthogonal projections. In this approach, there is no need to estimate the variance σ^2. Thresholds are automatically determined by orthogonal projections onto an upside-down pyramid shaped region, which is the epigraph set of the ℓ_1-norm cost function.

WAVELET SIGNALS DENOISING WITH PROJECTIONS ONTO ℓ_1-BALLS

Let us first study the projection of wavelet signals $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$ onto ℓ_1-balls, which

Digital Object Identifier 10.1109/MSP.2015.2440051
Date of publication: 13 August 2015

we will use to describe the projection onto the epigraph set of ℓ_1-norm cost function. We will use the term vector and signal in an interchangeable manner from now on. An ℓ_1-ball C_i, with size d_i is defined as follows:

$$C_i = \{\mathbf{w} \in \mathbb{R}^N : \sum_n |\mathbf{w}[n]| \le d_i\}, \quad (3)$$

where $\mathbf{w}[n]$ is the nth component of the vector $\mathbf{w}$ and d_i is the size of the ℓ_1-ball. In other words, an ℓ_1-ball is the set of vectors characterized by the fact that the sum of the magnitude of its components is lower than some specified value. Geometrically, such an ℓ_1-ball is a diamond shaped region bounded by a collection of hyperplanes as depicted in Figure 2. The orthogonal projection of a wavelet vector $\mathbf{w}_i$ onto an ℓ_1-ball is mathematically defined as follows:

$$\mathbf{w}_{pi} = \arg\min \|\mathbf{w}_i - \mathbf{w}\|_2^2$$
$$\text{such that } \|\mathbf{w}_i\|_1 = \sum_n |\mathbf{w}_i[n]| \le d_i, \quad (4)$$

where $\mathbf{w}_i$ is the ith wavelet signal, $\|\,.\,\|_2$ is the Euclidean norm, and $\|\,.\,\|_1$ is the ℓ_1-norm. The orthogonal projection operation onto an ℓ_1-ball is graphically shown in Figure 2. When $\|\mathbf{w}_i\|_1 \le d_i$ is satisfied, the wavelet signal is inside the ball, the projection has no effect and $\mathbf{w}_{pl} = \mathbf{w}_l$. In general, it can be shown that the orthogonal projection operation soft-thresholds each wavelet coefficient $\mathbf{w}_i[n]$ as follows:

$$\mathbf{w}_{pi}[n] = \text{sign}(\mathbf{w}_i[n]) \max\{(|\mathbf{w}_i[n]| - \theta_i), 0\}, \quad (5)$$

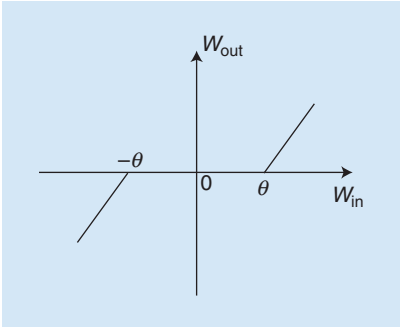
where $\text{sign}(\mathbf{w}_i[n])$ is the sign of $\mathbf{w}_i[n]$, and θ_i is a soft-thresholding constant whose value is determined according to the size of the ℓ_1-ball, d_i [11]. Algorithm 1 is an example of a method to solve the minimization problem (4) and thereby provide the constant θ_i for a given d_i value [11].

Projection of a wavelet signal onto an ℓ_1-ball reduces the amplitude of the wavelet coefficients of the input vector and eliminates the small valued wavelet coefficients, which are lower than the threshold θ_i. As a result, wavelet coefficients, which are probably due to noise, are removed by the projection operation. Projection operation onto an ℓ_1-ball retains the edges and sharp variation regions of the original signal because wavelet signals have large amplitude valued coefficients corresponding to edges [2] and they are not significantly affected by soft-thresholding. In standard wavelet denoising methods, the low-band signal $\mathbf{x}_L$ is not processed because $\mathbf{x}_L$ is a low resolution version of the original signal containing large amplitude coefficients almost for all n for most practical signals and images.

The next step is the estimation of the size of the ℓ_1-ball, d_i. We estimate the size of the ℓ_1-ball, d_i, by projecting $\mathbf{w}_i$ onto the epigraph set of the ℓ_1-norm cost function, which is an upside-down pyramid in $\mathbb{R}^{N+1}$ as shown in Figure 3. An upside-down pyramid is constructed by a family of ℓ_1-balls or diamond-shaped regions with different sizes ranging from 0 to $d_{\max,i} = \sum_n |\mathbf{w}_i[n]|$, whose value is the ℓ_1-norm of $\mathbf{w}_i$. When we orthogonally project $\mathbf{w}_i$ onto the upside down pyramid, we not only estimate the size of the ℓ_1-ball, but also soft-threshold the wavelet signal $\mathbf{w}_i$ as discussed in the following section.

ESTIMATION OF DENOISING THRESHOLDS

The epigraph set of the ℓ_1-norm cost function is an upside-down pyramid shaped region as shown in Figure 3. Each horizontal slice of the upside down pyramid is an ℓ_1-ball. The smallest value of the ℓ_1-ball is 0, which is at the bottom of the pyramid. The largest value of the ℓ_1-ball in the upside-down pyramid is $d_{\max,i} = \|\mathbf{w}_i\|_1$, which is determined by the boundary of the ℓ_1-ball touching the wavelet signal $\mathbf{w}_i$, i.e., the wavelet signal $\mathbf{w}_i$ is on one of the boundary hyperplanes of the ℓ_1-ball.



[FIG1] Soft-thresholding: $\mathbf{w}_{out}[n] = \text{sign}(\mathbf{w}_{in}[n]) \max\{(|\mathbf{w}_{in}[n]| - \theta), 0\}$.

Orthogonal projection of $\mathbf{w}_i$ onto an ℓ_1-ball with $d = 0$ produces an all-zero result. Projection of $\mathbf{w}_i$ onto an ℓ_1-ball with size $d_{\max,i}$, does not change $\mathbf{w}_i$ because $\mathbf{w}_i$ is on the boundary of the ℓ_1-ball. Therefore, for meaningful results, the size of the ℓ_1-ball, $d_i = z_{pi}$, must satisfy the inequality $0 < z_{pi} < d_{\max,i}$, for denoising. This condition can be expressed as follows:

$$\|\mathbf{w}_i\|_1 = \sum_{k=1}^{K} |\mathbf{w}_i[k]| \le z_{pi}, \quad (6)$$

where K is the length of the wavelet vector $\mathbf{w} = [\mathbf{w}[1], \mathbf{w}[2], \ldots, \mathbf{w}[K]]^T \in \mathbb{R}^K$. The condition (6) corresponds to the epigraph set C of the ℓ_1-norm cost function in $\mathbb{R}^{K+1}$, which is graphically illustrated

---

Algorithm 1: Order $(K\log(K))$ algorithm implementing projection onto the ℓ_1-ball with size d_i.

---

1): **Inputs:**

A vector $\mathbf{w}_i = [\mathbf{w}_i[1], \ldots, \mathbf{w}_i[K]]$ and a scalar $d_i > 0$

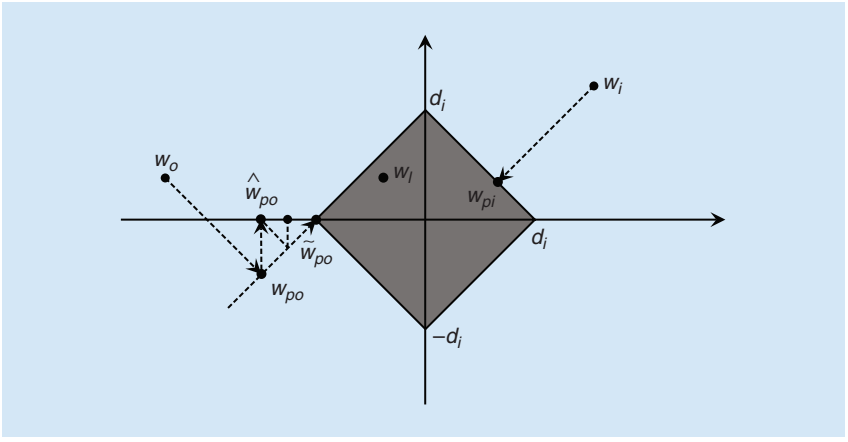
2): **Initialize:**

Sort $|\mathbf{w}_i[n]|$ for $n = 1, \ldots, K$ and obtain the rank ordered sequence $\mu_1 \ge \mu_2 \ge, \ldots, \ge \mu_K$. The soft-threshold value, θ_i, is given by

$$\theta_i = \frac{1}{\rho}\left(\sum_{n=1}^{\rho} \mu_n - d_i\right) \text{ such that } \rho = \max\{j \in \{1,2,\ldots,K\} : \mu_j - \frac{1}{j}\left(\sum_{r=1}^{j} \mu_r - d_i\right) > 0\}$$

3): **Output:**

$$\mathbf{w}_{pi}[n] = \text{sign}(\mathbf{w}_i[n]) \max\{|\mathbf{w}_i[n]| - \theta_i, 0\}, \; n = 1,2,\ldots,K$$



[FIG2] A graphical illustration of s projection onto an ℓ_1-ball with size d_i: Vectors w_{pi} and $\tilde{\mathrm{w}}_{po}$ are orthogonal projections of w_i and w_o onto an ℓ_1-ball with size d_i, respectively. The vector w_l is inside the ball, $\|\mathrm{w}_l\|_1 \le d_i$, and projection has no effect: $\mathrm{w}_{pl} = \mathrm{w}_l$.

in Figure 3 for $\mathrm{w}_i \in \mathbb{R}^2$ [8], [9]. The epigraph set C is defined in $\mathbb{R}^{N+1}$ as follows:

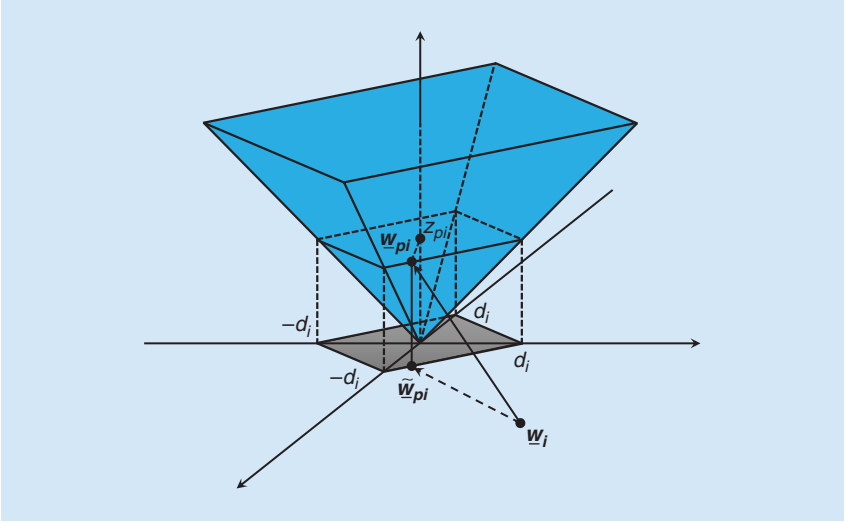
$$C = \{\underline{\mathrm{w}}_i = [\mathrm{w}_i^T, z_{pi}]^T \in \mathbb{R}^{K+1} : \|\mathrm{w}_i\|_1 \le z_{pi},\ z_{pi} \le d_{\max,i}\}, \quad (7)$$

which represents a family of ℓ_1-balls for $0 < z_{pi} \le d_{\max,i}$ in $\mathbb{R}^{K+1}$. In (7) there are $K+1$ variables: $\mathrm{w}_i[1], \ldots, \mathrm{w}_i[K]$, and z_{pi}. Since the space is now $K+1$ dimensional, we increase the size of wavelet signals by one:

$$\underline{\mathrm{w}}_i = [\mathrm{w}_i^T, 0]^T = [\mathrm{w}_i[1], \mathrm{w}_i[2], \ldots, \mathrm{w}_i[K], 0]^T, \quad (8)$$

where $\underline{\mathrm{w}}_i \in \mathbb{R}^{K+1}$. The signal $\underline{\mathrm{w}}_i$ is the $K+1$ dimensional version of vector $\mathrm{w}_i \in \mathbb{R}^K$. From now on, we underline vectors in $\mathbb{R}^{K+1}$ to distinguish them from K-dimensional vectors.

The extended wavelet vector $\underline{\mathrm{w}}_i$ can be projected onto the epigraph set C to determine the vector $\underline{\mathrm{w}}_{pi} = [\mathrm{w}_{pi}[1], \ldots, \mathrm{w}_{pi}[K], z_{pi}]^T$ as graphically illustrated in Figure 3. This projection is unique and is the closest vector on the epigraph set to $\underline{\mathrm{w}}_i = [\mathrm{w}_i^T, 0]^T$. The baseline mathematical operation is an orthogonal projection onto a hyperplane which is the face (boundary) of the epigraph set C in the quadrant of the $\underline{\mathrm{w}}_i$. The orthogonal projection $\underline{\mathrm{w}}_{pi}$ of $\underline{\mathrm{w}}_i$ is a denoised version of $\underline{\mathrm{w}}_i$ because it is equivalent to the orthogonal projection of w_i onto the ℓ_1-ball with size z_{pi} in $\mathbb{R}^K$, as graphically illustrated in Figure 3.



[FIG3] Projection of $\mathrm{w}_i[n]$ onto the epigraph set of ℓ_1-norm cost function: $C = \{\mathrm{w}: \sum_{n=0}^{K-1} |\mathrm{w}[k]| \le z_{pi}\}$, gray-shaded region.

Orthogonal projection onto the epigraph set C can be computed in two steps. In the first step, $[\mathrm{w}_i^T, 0]^T$ is projected onto the boundary hyperplane of the epigraph set which is defined as:

$$\sum_{n=1}^{K} \mathrm{sign}(\mathrm{w}_i[n]) . \mathrm{w}_i[n] - z_{pi} = 0, \quad (9)$$

where the coefficients of the above hyperplane are determined according to the signs of $\mathrm{w}_i[n]$. This hyperplane determines the boundary of the epigraph set C facing the vector $\underline{\mathrm{w}}_i$ as shown in Figure 3. The projection vector $\underline{\mathrm{w}}_{pi}$ onto the hyperplane (9) in $\mathbb{R}^{K+1}$ is determined using (1):

$$\mathrm{w}_{pi}[n] = \mathrm{w}_i[n] - \frac{\sum_{n=1}^{K} |\mathrm{w}_i[n]|}{K+1} \mathrm{sign}(\mathrm{w}_i[n]) \quad n = 1, 2, \ldots, K, \quad (10)$$

where $K+1 = \|[\mathrm{sign}(\mathrm{w}_i[1]), \ldots, \mathrm{sign}(\mathrm{w}_i[k]), -1]\|^2$ and the last component z_{pi} of $\underline{\mathrm{w}}_{pi}$ is given by

$$z_{pi} = \frac{\sum_{n=1}^{K} \mathrm{sign}(\mathrm{w}_i[n]) \mathrm{w}_i[n]}{K+1} = \frac{\sum_{n=1}^{K} |\mathrm{w}_i[n]|}{K+1}. \quad (11)$$

As mentioned earlier above, this orthogonal projection operation also determines the size of the ℓ_1-ball, $d_i = z_{pi}$, which can be verified using geometry.

In general, the projection vector $\underline{\mathrm{w}}_{pi}$ may or may not be the projection of $\underline{\mathrm{w}}_i$ onto the epigraph set C. In Figures 2 and 3, it is. The ℓ_1-ball in Figure 2 can be interpreted as the projection of 3-D ℓ_1-ball onto 2-D plane (view from the top). The issue comes from the fact that projecting onto the ℓ_1-ball has been simplified to projecting onto a single hyperplane, which may not yield the desired result in some specific geometrical configurations. For instance, in Figure 2, the vector w_{po} is neither the orthogonal projection of w_o onto the ℓ_1-ball, nor to the epigraph set of the ℓ_1-ball, because w_{po} is not on the ℓ_1-ball. Such cases can easily be spotted by checking the signs of the entries of the projection vectors. If the signs of the entries $\mathrm{w}_{pi}[n]$ of projection

vector w_{pi} are the same as $\mathrm{w}_i[n]$ for all n, then the w_{pi} is on the epigraph set C, otherwise w_{pi} is not on the ℓ_1-ball. If w_{pi} is not on the ℓ_1-ball we can still project w_i onto the ℓ_1-ball using Algorithm 1 or Duchi et al's ℓ_1-ball projection algorithm [11] using the value of $d_i = z_{pi}$ determined in (11).

In summary, we have the following two steps: 1) project $\underline{\mathrm{w}}_i = [\mathrm{w}_i^T, 0]^T$ onto the boundary hyperplane of the epigraph set C and determine d_i using (11); and 2) if $\mathrm{sign}(\mathrm{w}_i[n]) = \mathrm{sign}(\mathrm{w}_{pi}[n])$ for all n, w_{pi} is the projection vector; otherwise, use $d_i = z_{pi}$ in Algorithm 1 to determine the final projection vector. Since there are $i = 1, 2, \ldots, L$ wavelet signals, each wavelet signal w_i should be projected onto possibly distinct ℓ_1-balls with sizes d_i. Notice that d_i is not the value of the soft-threshold, it is the size of the ℓ_1-ball. The value of the soft-threshold is determined using Algorithm 1.

In practice, we may further simplify step 2 in denoising applications. Our goal is to zero-out insignificant wavelet coefficients. Therefore, we compare signs of entries of w_{po} and w_o. We can zero-out those entries whose signs change after the orthogonal projection. Step 2 is then becomes as is shown in (12) below.

This operation is also graphically illustrated in Figure 2. The vector w_o is projected onto the boundary hyperplane facing w_o to obtain w_{po}, which then projected back to the quadrant of w_o to obtain the denoised version $\widehat{\mathrm{w}}_{po}$. This process can be iterated a couple of times to approach the orthogonal projection vector $\widetilde{\mathrm{w}}_{po}$ as shown in Figure 2.

Stronger denoising of the input vector is simply a matter of selecting a z_p value smaller than z_{pi} in (11). A z_p value closer to zero leads to a higher threshold and forces more wavelet coefficients to be zero after the projection operation.

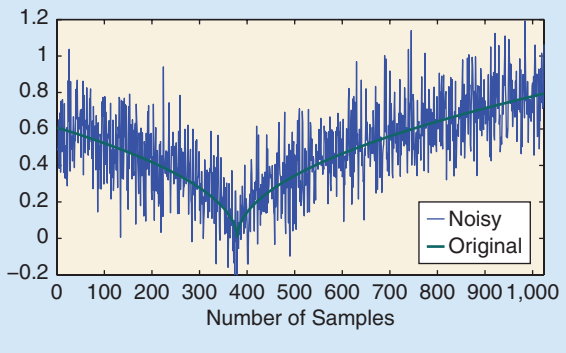
IT IS ALSO POSSIBLE TO USE A PYRAMIDAL STRUCTURE FOR SIGNAL DECOMPOSITION INSTEAD OF THE WAVELET TRANSFORM.

HOW TO DETERMINE THE NUMBER OF WAVELET DECOMPOSITION LEVELS

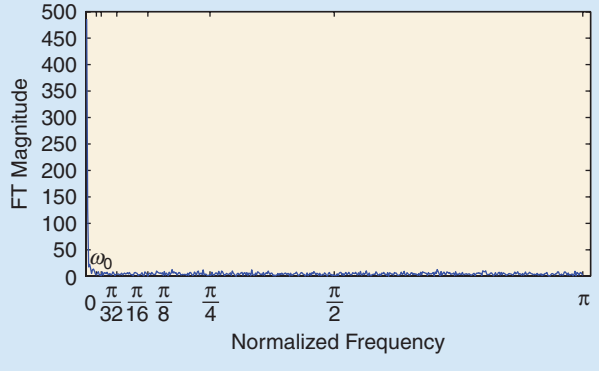
There are many ways to estimate the number of wavelet decomposition levels [6]. It is also possible to use the Fourier transform of the noisy signal to approximately estimate the bandwidth of the signal. Once the bandwidth ω_0 of the original signal is approximately determined, it can be used to estimate the number of wavelet transform levels and the bandwidth of the low-band signal x_L. In an L-level wavelet decomposition, the low-band signal x_L approximately comes from the $[0, (\pi/2^L)]$ frequency band of the signal x. Therefore, $(\pi/2^L)$ must be comparable to ω_0 so that the actual signal components are not soft-thresholded. Only wavelet signals $\mathrm{w}_1, \ldots, \mathrm{w}_{L-1}, \mathrm{w}_L$, whose Fourier transforms approximately occupy the bands $[(\pi/2), \pi], \ldots, [(\pi/2^{L-1}), (\pi/2^{L-2})], [(\pi/2^L), (\pi/2^{L-1})]$, respectively, should be soft-thresholded in denoising. For example, consider the `cusp` signal defined in MATLAB. It is possible to estimate an approximate frequency value ω_0 for this signal. The `cusp` signal is corrupted by additive zero-mean white Gaussian noise with $\sigma = 20\%$ of the maximum amplitude of the original signal as shown in Figure 4. The magnitude of the Fourier transform of the `cusp` signal is shown in Figure 5. For this signal, an $L = 5$ level wavelet decomposition is suitable because the magnitude of the Fourier transform approaches the noise floor level at high frequencies after $\omega_0 \approx (\pi/46)$ as shown in Figure 5. Therefore, $L = 5$ $((\pi/2^5) > \omega_0)$ is selected as the number of wavelet decomposition levels.

It is also possible to use a pyramidal structure for signal decomposition instead of the wavelet transform. In this case, the noisy signal is filtered with a lowpass filter with a cut-off frequency of $(\pi/8)$ and the output x_{lp} is subtracted

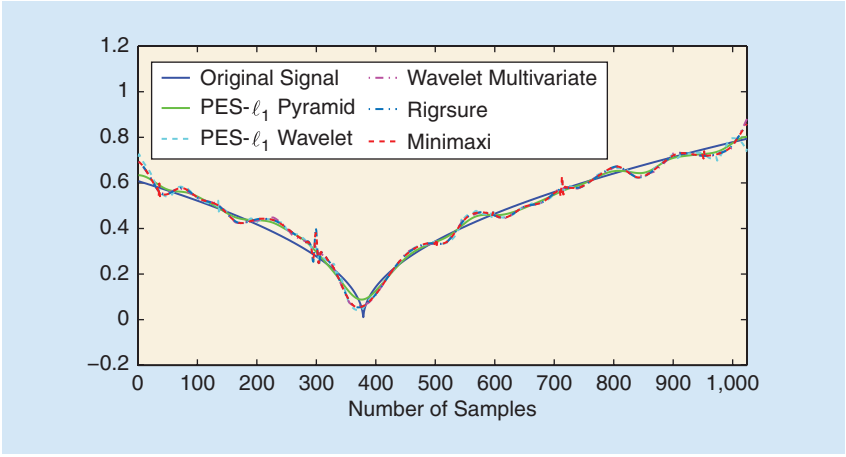
$$\widehat{\mathrm{w}}_{po}[n] = \begin{cases} \mathrm{w}_{po}[n], & \text{if } \mathrm{sign}(\mathrm{w}_{po}[n]) = \mathrm{sign}(\mathrm{w}_o[n]) \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$



[FIG4] The `cusp` signal and its corrupted version with Gaussian noise with $\sigma = 20\%$ of maximum amplitude of the original signal.



[FIG5] The discrete-time Fourier transform magnitude of `cusp` signal corrupted by noise. The wavelet decomposition level L is selected as five satisfying $(\pi/2^5) > \omega_0$, which is the approximate bandwidth of the signal.



[FIG6] Original `cusp` signal (blue), denoised signal (green) using PES-ℓ_1-ball with pyramid; SNR = 28.26 dB, denoised signal (cyan) using PES-ℓ_1-ball with wavelet; SNR = 25.30 dB, denoised signal (magenta) using MATLAB wavelet multivariate method; SNR = 25.08 dB [6], denoised signal (petroleum blue) using wavelet denoising `rigrsure` algorithm [3]; SNR = 23.28 dB, and denoised signal (red) using wavelet denoising `minimaxi` algorithm [7]; SNR = 24.52 dB.

from the noisy signal x to obtain the high-pass signal $\mathbf{x}_{\text{hp}}$ as shown in [12]. The highpass signal $\mathbf{x}_{\text{hp}}$ is projected onto the epigraph of ℓ_1-norm cost function and the denoised signal $\mathbf{x}_{\text{hd}}$ is obtained. Projection onto the epigraph set of ℓ_1-ball (PES-ℓ_1) removes the noise by soft-thresholding. The pyramidal signal decomposition approach is similar to the undecimated wavelet transform. The denoised signal is reconstructed by adding $\mathbf{x}_{\text{hd}}$ and $\mathbf{x}_{\text{lp}}$.

In Figure 6, the signal is restored using PES-ℓ_1 with a pyramid structure, PES-ℓ_1 with wavelet, MATLAB's wavelet multivariate denoising algorithm [6], MATLAB's soft-thresholding denoising algorithms (`minimaxi` and `rigrsure` thresholds), and wavelet thresholding denoising method. The denoised signals have signal-to-noise (SNR) values equal to 28.26, 25.30, 25.08, 23.28, and 24.52 dB, respectively. In average, PES-ℓ_1 with pyramid and PES-ℓ_1 with wavelet method produce better denoising results than the other soft-thresholding methods. The SNR is calculated using the formula: $\text{SNR} = 20 \times \log_{10}(\|\mathbf{x}_{\text{orig}}\|/\|\mathbf{x}_{\text{orig}} - \mathbf{x}_{\text{rec}}\|)$. Extensive simulation results and the denoising software are available on the Internet [12].

CONCLUSIONS

PROS

Orthogonal projection-based denoising is computationally efficient because projection onto a boundary hyperplane of an ℓ_1-ball or the epigraph set can be implemented by performing only one division and $K+1$ additions and/or subtractions, and sign computations. Once the size of the ℓ_1-ball using (10) and (11) is determined, the orthogonal projection onto an ℓ_1-ball operation is an order (K) operation. Equations (10) and (11) only involve multiplications by ± 1.

CONS

It is not possible to incorporate any prior knowledge about the noise probability density function or any other statistical information to the orthogonal projection based denoising method. However, it produces good denoising results under additive white Gaussian noise. Most of the denoising methods available in MATLAB also assumes that the noise is additive, white Gaussian.

ACKNOWLEDGMENT

This work is funded by the Scientific and Technological Research Council of Turkey (TUBITAK) under project 113E069.

AUTHORS

A. Enis Cetin (cetin@bilkent.edu.tr) is a professor in the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey. His main research interests are multimedia signal processing and its applications. He is a Fellow of the IEEE.

Mohammad Tofighi (tofighi@ee.bilkent.edu.tr) is an M.Sc. student in the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey. His research interests include signal and image processing, inverse problems in signal processing, computer vision, pattern recognition, and machine learning. He is a Student Member of the IEEE.

REFERENCES

[1] K. Slavakis, S.-J. Kim, G. Mateos, and G. Giannakis, "Stochastic approximation vis-a-vis online learning for big data analytics," *IEEE Signal Processing Mag.*, vol. 31, no. 6, pp. 124–129, Nov. 2014.

[2] S. Mallat and W.-L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 617–643, Mar. 1992.

[3] D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[4] S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1532–1546, Sept. 2000.

[5] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Mag.*, vol. 24, no. 4, pp. 118–121, July 2007.

[6] M. Aminghafari, N. Cheze, and J.-M. Poggi, "Multivariate denoising using wavelets and principal component analysis," *Computat. Stat. Data Anal.*, vol. 50, no. 9, pp. 2381–2398, 2006.

[7] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Am. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.

[8] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "Epigraphical projection and proximal tools for solving constrained convex optimization problems," *Signal, Image, Video Process.*, 2014, pp. 1–13.

[9] M. Tofighi, K. Kose, and A. E. Cetin, "Denoising using projections onto the epigraph set of convex cost functions," in *Proc. 2014 IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2014, pp. 2709–2713.

[10] J. Fowler, "The redundant discrete wavelet transform and additive noise," *IEEE Signal Processing Lett.*, vol. 12, no. 9, pp. 629–632, Sept. 2005.

[11] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Machine Learning* (*ICML'08*). New York: ACM, 2008, pp. 272–279.

[12] (2014). PES-L1 Denoising Software. [Online]. Available: http://signal.ee.bilkent.edu.tr/1DDenoisingSoftware.html