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 multi-dimensional space. Each horizontal slice of the upside down pyramid is a diamond shaped region and it is called an \( \ell_1 \)-ball. The upside down pyramid is called the epigraph set of the \( \ell_1 \)-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 \( w_0 \) be a vector in \( \mathbb{R}^K \). The orthogonal projection \( w_{po} \) of \( w_0 \) onto the hyperplane \( h = a^T w = \sum_{n=1}^{K} a[n] w[n] \) is given by

\[
w_{po}[n] = w_0[n] + \frac{h - \sum_{n=1}^{K} a[n] w_0[n]}{\|a\|_2} a[n], \quad n = 1, 2, \ldots, K,
\]

where \( w_0[n] \), \( w_{po}[n] \), and \( a[n] \) are the \( n \)th entries of the vectors \( w_0 \), \( w_{po} \), and \( a \), respectively, and \( \|a\|_2 \) is the Euclidean length (norm) of the vector \( a \).

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 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 \( v \) be a discrete-time signal and \( x \) be its noisy version, i.e., \( x[n] = v[n] + \xi[n], n = 1, 2, \ldots, N \), where \( \xi \) is some additive, independent and identically distributed (i.i.d.), zero-mean, white Gaussian noise with variance \( \sigma^2 \). An \( L \)-level discrete wavelet transform of \( x \) is computed and the lowband signal \( x_L \) and wavelet signals \( w_1, w_2, \ldots, w_L \) are obtained. After this step, wavelet signals are soft-thresholded as shown in Figure 1. The soft-threshold value, \( \theta \), 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 \( x \), and the constant \( \gamma \) is defined in [3]. In (2), the noise variance \( \sigma^2 \) has to be known or properly estimated from the observations, \( x \), 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 \( \gamma \) and \( \sigma \) in Donoho’s method.

It is possible to define a soft-threshold \( \theta_i \) for each wavelet signal \( w_i \). Here we present how to estimate a soft-threshold value \( \theta_i \) for each wavelet signal \( w_i \) using a deterministic approach based on linear algebra and orthogonal projections. In this approach, there is no need to estimate the variance \( \sigma^2 \). Thresholds are automatically determined by orthogonal projections onto an upside-down pyramid shaped region, which is the epigraph set of the \( \ell_1 \)-norm cost function.

**WAVELET SIGNALS DenoISING WIth PROJECTIONS ONTO \ell_1-BALLS**

Let us first study the projection of wavelet signals \( w_1, w_2, \ldots, w_L \) onto \( \ell_1 \)-balls, which
we will use to describe the projection onto the epigraph set of $\ell_1$-norm cost function. We will use the term vector and signal in an interchangeable manner from now on. An $\ell_1$-ball $C_i$, with size $d_i$, is defined as follows:

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

where $w[n]$ is the $n$th component of the vector $w$ and $d_i$ is the size of the $\ell_1$-ball. In other words, an $\ell_1$-ball is a diamond shaped region bounded by a collection of hyperplanes as depicted in Figure 2. The orthogonal projection of a wavelet vector $w_i$ onto an $\ell_1$-ball is mathematically defined as follows:

$$w_{ps} = \arg\min \| w_i - w \| \quad \text{such that } \| w_i \| = \sum_n |w_i[n]| \leq d_i \quad (4)$$

where $w_i$ is the $i$th wavelet signal, $\| \|_1$ is the Euclidean norm, and $\| \|_1$ is the $\ell_1$-norm. The orthogonal projection operation onto an $\ell_1$-ball is graphically shown in Figure 2. When $\| w_i \| \leq d_i$ is satisfied, the wavelet signal is inside the ball, the projection has no effect and $w_{ps} = w_i$. In general, it can be shown that the orthogonal projection operation soft-thresholds each wavelet coefficient $w[n]$ as follows:

$$w_{ps}[n] = \text{sign}(w[n]) \max(|w[n] - \theta|, 0) \quad (5)$$

where $\text{sign}(w[n])$ is the sign of $w[n]$, and $\theta$ is a soft-thresholding constant whose value is determined according to the size of the $\ell_1$-ball, $d_i$ [11]. Algorithm 1 is an example of a method to solve the minimization problem (4) and thereby provide the constant $\theta_i$ for a given $d_i$ value [11].

Projection of a wavelet signal onto an $\ell_1$-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 $\theta_i$. As a result, wavelet coefficients, which are probably due to noise, are removed by the projection operation. Projection operation onto an $\ell_1$-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 $x_L$ is not processed because $x_L$ 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 $\ell_1$-ball, $d_i$. We estimate the size of the $\ell_1$-ball, $d_i$, by projecting $w_i$ onto the epigraph set of the $\ell_1$-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 $\ell_1$-balls or diamond-shaped regions with different sizes ranging from 0 to $d_{\text{max},i} = \sum_n |w_i[n]|$, whose value is the $\ell_1$-norm of $w_i$. When we orthogonally project $w_i$ onto the upside down pyramid, we not only estimate the size of the $\ell_1$-ball, but also soft-threshold the wavelet signal $w_i$ as discussed in the following section.

**ESTIMATION OF DENOISING THRESHOLDS**

The epigraph set of the $\ell_1$-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 $\ell_1$-ball. The smallest value of the $\ell_1$-ball is 0, which is at the bottom of the pyramid. The largest value of the $\ell_1$-ball in the upside-down pyramid is $d_{\text{max},i} = \| w_i \|_1$, which is determined by the boundary of the $\ell_1$-ball touching the wavelet signal $w_i$, i.e., the wavelet signal $w_i$ is on one of the boundary hyperplanes of the $\ell_1$-ball.

Orthogonal projection of $w_i$ onto an $\ell_1$-ball with $d = 0$ produces an all-zero result. Projection of $w_i$ onto an $\ell_1$-ball with size $d_{\text{max},i}$ does not change $w_i$ because $w_i$ is on the boundary of the $\ell_1$-ball. Therefore, for meaningful results, the size of the $\ell_1$-ball, $d_i = z_{ps}$, must satisfy the inequality $0 < z_{ps} < d_{\text{max},i}$ for denoising. This condition can be expressed as follows:

$$\| w_i \|_1 = \sum_{k=1}^K |w_i[k]| \leq z_{ps} \quad (6)$$

where $K$ is the length of the wavelet vector $w = [w[1], w[2], ..., w[K]]^T \in \mathbb{R}^K$. The condition (6) corresponds to the epigraph set $C$ of the $\ell_1$-norm cost function in $\mathbb{R}^{n+1}$, which is graphically illustrated.

---

**Algorithm 1: Order ($K \log(K)$) algorithm implementing projection onto the $\ell_1$-ball with size $d_i$.**

1): Inputs:

A vector $w_i = [w_i[1], ..., w_i[K]]$ and a scalar $d_i > 0$

2): Initialize:

Sort $|w_i[n]|$ for $n = 1, ..., K$ and obtain the rank ordered sequence $\mu_1 \geq \mu_2 \geq ... \geq \mu_K$. The soft-threshold value, $\theta_i$, is given by

$$\theta_i = \frac{1}{\rho} \left( \sum_{n=1}^K |w_i[n]| - d_i \right) \quad \text{such that } \rho = \max(j \in \{1, 2, ..., K\})$$

$$\mu_j - \frac{1}{\rho} \left( \sum_{i=1}^j |w_i[n]| - d_i \right) > 0$$

3): Output:

$$w_{ps}[n] = \text{sign}(w_i[n]) \max(|w_i[n] - \theta_i|, 0), \quad n = 1, 2, ..., K$$

---
The baseline mathematical sign is the gray-shaded region. 

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

$$
\sum_{i=1}^{K} \text{sign} (w_i[n]) \cdot w_i[n] - z_{pi} = 0,
$$

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

$$
w_{pi}[n] = w_o[n] - \frac{\sum_{i=1}^{K} |w_i[n]|}{K+1} \cdot \text{sign} (w_i[n])
$$

where $K+1 = \| \text{sign} (w_i[1]), \ldots, \text{sign} (w_i[k]), -1 \|$. and the last component $z_{pi}$ of $w_{pi}$ is given by

$$
z_{pi} = \frac{\sum_{i=1}^{K} \text{sign} (w_i[n]) \cdot w_i[n]}{K+1} = \frac{\sum_{i=1}^{K} |w_i[n]|}{K+1}.
$$

As mentioned earlier above, this orthogonal projection operation also determines the size of the $\ell_1$-ball, $d = z_{pi}$, which can be verified using geometry.

In general, the projection vector $w_{pi}$ may or may not be the projection of $w_f$ onto the epigraph set $C$. In Figures 2 and 3, it is. The $\ell_1$-ball in Figure 2 can be interpreted as the projection of 3-D $\ell_1$-ball onto 2-D plane (view from the top). The issue comes from the fact that projecting onto the $\ell_1$-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 $\ell_1$-ball, nor to the epigraph set of the $\ell_1$-ball, because $w_{po}$ is not on the $\ell_1$-ball. Such cases can easily be spotted by checking the signs of the entries of the projection vectors. If the signs of the entries $w_{pi}[n]$ of projection
vector \( w_p \) are the same as \( w_i[n] \) for all \( n \), then the \( w_p \) is on the epigraph set \( C \). otherwise \( w_p \) is not on the \( \ell_1 \)-ball. If \( w_p \) is not on the \( \ell_1 \)-ball we can still project \( w_i \) onto the \( \ell_1 \)-ball using Algorithm 1 or Duchi et al.’s \( \ell_1 \)-ball projection algorithm [11] using the value of \( d_i = z_p \) determined in (11).

In summary, we have the following two steps: 1) project \( w_i = [w_i^T, 0]^T \) onto the boundary hyperplane of the epigraph set \( C \) and determine \( d_i \) using (11); and 2) if \( \text{sign}(w_i[n]) = \text{sign}(w_p[n]) \) for all \( n \), \( w_p \) is the projection vector; otherwise, use \( d_i = z_p \) in Algorithm 1 to determine the final projection vector. Since there are \( i = 1, 2, ..., L \) wavelet signals, each wavelet signal \( w_i \) should be projected onto possibly distinct \( \ell_1 \)-balls with sizes \( d_i \). Notice that \( d_i \) is not the value of the soft-threshold, it is the size of the \( \ell_1 \)-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 \( w_p \) and \( w_i \). 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 \( w_i \) to obtain \( w_{po} \), which then projected back to the quadrant of \( w_o \) to obtain the denoised version \( w_{po} \). This process can be iterated a couple of times to approach the orthogonal projection vector \( w_{po} \) as shown in Figure 2.

Stronger denoising of the input vector is simply a matter of selecting a \( z_p \) value smaller than \( z_p \) 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.**

![IT IS ALSO POSSIBLE TO USE A PYRAMIDAL STRUCTURE FOR SIGNAL DECOMPOSITION INSTEAD OF THE WAVELET TRANSFORM.](image)

**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 \( \omega_0 \) 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 \([0, (\pi/2^L)]\) frequency band of the signal \( x \). Therefore, \((\pi/2^L)\) must be comparable to \( \omega_0 \) so that the actual signal components are not soft-thresholded. Only wavelet signals \( w_1, ..., w_{L-1}, w_L \), whose Fourier transforms approximately occupy the bands \([\pi/2^L, \pi], ..., [(\pi/2^L-1), (\pi/2^{L-1})], ..., [(\pi/2^2), (\pi/2^1)]\) 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 \( \omega_0 \) 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 \( \omega_0 = (\pi/46) \) as shown in Figure 5. Therefore, \( L = 5 \) \((\pi/2^2) > \omega_0 \) 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 \( x_0 \) is subtracted.

![IT IS ALSO POSSIBLE TO USE A PYRAMIDAL STRUCTURE FOR SIGNAL DECOMPOSITION INSTEAD OF THE WAVELET TRANSFORM.](image)

**[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.
from the noisy signal $x$ to obtain the high-pass signal $x_{hp}$ as shown in [12]. The highpass signal $x_{hp}$ is projected onto the epigraph of $\ell_1$-norm cost function and the denoised signal $x_{ld}$ is obtained. Projection onto the epigraph set of $\ell_1$-ball (PES-$\ell_1$) 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 $x_{ld}$ and $x_{hp}$.

In Figure 6, the signal is restored using PES-$\ell_1$ with a pyramid structure, PES-$\ell_1$ 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-$\ell_1$ with pyramidal and PES-$\ell_1$ with wavelet method produce better denoising results than the other soft-thresholding methods. The SNR is calculated using the formula: 

$$SNR = 20 \times \log_{10}\left(\frac{\|x_{orig}\|}{\|x_{orig} - x_{ld}\|}\right).$$

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 $\ell_1$-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 $\ell_1$-ball using (10) and (11) is determined, the orthogonal projection onto an $\ell_1$-ball operation is an order ($K$) operation. Equations (10) and (11) only involve multiplications by $\pm 1$.

**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 assume 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**


