Noname manuscript No.
(will be inserted by the editor)

A wavelet filter comparison on multiple datasets for signal compression and denoising

Alessandro Gnutti · Fabrizio Guerrini ·
Nicola Adami · Pierangelo Migliorati ·
Riccardo Leonardi

Received: date / Accepted: date

Abstract In this paper, we explicitly analyze the performance effects of several orthogonal and bi-orthogonal wavelet families. For each family, we explore the impact of the filter order (length) and the decomposition depth in the multiresolution representation. In particular, two contexts of use are examined: compression and denoising. In both cases, the experiments are carried out on a large dataset of different signal kinds, including various image sets and 1D signals (audio, electrocardiogram and seismic). Results for all the considered wavelets are shown on each dataset. Collectively, the study suggests that a meticulous choice of wavelet parameters significantly alters the performance of the above mentioned tasks. To the best of authors' knowledge, this work represents the most complete analysis and comparison between wavelet filters. Therefore, it represents a valuable benchmark for future works.

Keywords Sub-band coding · Discrete Wavelet Transform · Wavelet filter comparison · multiresolution analysis.

1 Introduction

Wavelets are a mathematical framework largely used for many signal processing applications [1,2]. Differently from classical transforms, like the Discrete Fourier Transform (DFT), the Discrete Walsh-Hadamard Transform (DWHT), the Discrete Cosine Transform (DCT), and so on, the Discrete Wavelet Transform (DWT) can be specifically optimized to the considered target objective by adjusting several design properties such as coding gain, smoothness, stop-band attenuation of the wavelet filters, etc.

One of the most outstanding applications is transform-based image compression. Typically, in image coding the data is transformed to remove redundancy.

Alessandro Gnutti · Fabrizio Guerrini · Nicola Adami · Pierangelo Migliorati · Riccardo Leonardi
Department of Information Engineering, University of Brescia – CNIT, Via Branze 38, 25123, Brescia, Italy.
E-mail: firstname.lastname@unibs.it

Then, the transformed coefficients are quantized, which is the only lossy operation. Finally, the quantized values are entropy coded. The wavelet transform has positively contributed to the compression field because of its remarkable energy compaction, in addition to a significant correspondence with the human visual system. It is today one of the most well established compression technique [3]. This is specifically due to the tight connection with the concept of multiresolution analysis, *i.e.*, sub-band decomposition through filter banks.

Basically, the wavelet basis functions have short support for high frequencies and long support for low frequencies. So, very few bits are sufficient to describe extended smooth areas of an image, while potential extra bits may be added to represent high frequency details. Furthermore, working with variable-length basis functions also allows to avoid blocking artifacts. The latter are instead typically present with linear block transforms which use fixed-length eigenvectors, *e.g.*, at medium-low bit rates in images compressed using the well-known JPEG (Joint Photographic Experts Group) standard.

Of course, wavelets can be used not just for 2D signals, namely images. For example, the separable processing can be extended from 2D to 3D volumes for video compression. In particular, a 3D separable filter bank is used to transform the video sequences, adopting specific approaches such as Motion-Compensated Temporal Filtering (MCTF) [4]. Moreover, wavelets are also well suited for speech and audio compression. Indeed, human hearing is associated with critical bands, and such non-uniform frequency intervals can be well estimated by tree-structured filter banks. Still concerning the 1D domain, compression of electrocardiogram (ECG) waveforms can take advantage of a wavelet representation too. As a matter of fact, wavelets are able to minimize the distortion of ECG data, while keeping all the significant signal features needed to detect potential heart arrhythmia or disorders.

Another signal processing field where wavelets play an important role is denoising. The considered signals usually exhibit low frequency energy concentration. Under this assumption, thresholding, *i.e.*, the zeroing of coefficients with energy lower than a given threshold, generally returns a lowpass version of the original signal. Also, in practical scenarios the noise power is much smaller than the signal power, and the noise source has approximately constant power spectral density. This means that thresholding the wavelet coefficients at the last level of the multiresolution decomposition of a noisy signal basically removes the high frequency content associated to noise. Clearly, this process will also delete part of the signal power. However, for denoising, simple sub-band thresholding allows to achieve good Signal Noise to Ratio (SNR) performance, thanks to the excellent approximation properties of the wavelet decomposition.

1.1 Contributions

In this paper, we examine a large set of wavelet functions in order to provide a performance comparison in both compression and denoising frameworks. The evaluation will be carried out by exploring how some of the main wavelet properties impact the efficiency of the investigated system, namely:

- orthogonality *vs.* bi-orthogonality;

- filter order (strictly associated to the filter length) that, in turn, determines other characteristics, such as regularity, degree of smoothness, and band attenuation;
- number of levels in the sub-band decomposition.

For both tasks, the analysis will be performed on multiple databases, including standard and high definition images (2D signals), as well as audio, ECG and seismic signals (1D signals). Differently from other similar works, that we discuss in what follows, this paper expands its test set beyond specific cases, and compare a large group of the best well known wavelet families. Observing the results that will be shown, we will try to infer some rules of thumb for the wavelet family best selection on the particularly considered frameworks. While doing so, this paper also suggests to always adopt feasible procedures, like the one presented here, to determine the most suitable DWT in any processing context. To this aim, we also provide the original code, and the experimental datasets, to let the interested reader further his/her wavelet-based research topic.

The rest of the paper is organized as follows. Section 2 presents the state of the art related to some of the aforementioned signal processing tasks, in which wavelets are largely used. Then, in Section 3, the main theoretical aspects are illustrated. The description of the experimental evaluation is provided in Section 4, with a focus on dataset selection, considered wavelet properties, and quality evaluation metrics. Section 5 reports and discusses the experimental results. Finally, some conclusions are drawn in Section 6.

2 Related work

Wavelet-based methods have significant advantages in many signal processing fields, applicable to various types of data. The first important works in the literature on the practical use of wavelets date to the early 90s. These very influential papers [5,6,7] concern image compression. Shortly after, JPEG 2000 defined an image coding system based on the wavelet transform [8], that superseded the preceding DCT-based JPEG standard. In JPEG 2000 (or J2K), two types of wavelet filters are used, namely Cohen-Daubechies-Feauveau (CDF) 5/3 for lossless compression, and CDF 9/7 for lossy compression. However, no evidence was given to prove the optimality under some criterion for such filters. Then, other wavelet image compression methodologies have been introduced as well, such as the popular zerotree-based algorithms [9,10], techniques using morphological operators [11], and so on.

Still focusing on visual data, wavelet-based algorithms have been later combined with more modern techniques, *e.g.*, compressive sensing [12,13], or applied to recently introduced typologies of visual data. Examples of the latter include hyperspectral [14] and volumetric medical [15] images. 3D data have been considered as well, *e.g.*, in [16] the authors analyze the ciclostationary oscillations in a 3D wavelet coding framework. Such oscillations model the reconstruction error, which is due to the wavelet sub-bands quantization, with respect to the filter properties. Wavelets have been also proposed for scalable video coding [17]. There are many more recent applications that have exploited the potentialities of the wavelet analysis. Examples include, but are not limited to denoising [18,19,20], watermarking [21,22], resolution enhancement [23,24], and image fusion [25,26, 27].

The wavelet transform is largely used to deal with other types of signals too. For example, in [28] the authors propose an adaptive audio watermarking algorithm based on the singular value decomposition in the wavelet domain. Wavelets have been employed for electrocardiogram (ECG) signals too. In [29], the authors use wavelets for ECG signals denoising. In [30], the ability of the DWT to provide good time and frequency resolutions is exploited to decipher the hidden complexities in the ECG, allowing to classify them. Other wavelet-based methods have been applied on seismic data. For instance, in [31] wavelets-based denoising techniques are used to enhance the first-arrival picking on seismic traces. In [32], the wavelet transform is proposed to obtain a sparse representation of the seismic data for compression purposes.

Typically, all of the just mentioned signal processing applications do not take into account, much less operate, a meticulous analysis to choose which particular wavelet function could be the best performing (in some sense). Actually, we shall show how an accurate selection of wavelet characteristics may actually improve any particular task results. On the other hand, a handful of works have been already proposed in order to identify optimal wavelets in limited settings. For example, some of these papers [33,34,35] report a performance comparison of various wavelet filters for image compression. However, the analysis has always been limited to a restricted set of standard resolution images, thus reducing the universality of the results. Then, another work [36] have proposed a procedure to choose the wavelet filter more suited for ECG signals denoising, with the specific goal to ensure accurate signal peaks. Furthermore, the analysis in [37] has searched for the optimal wavelet parameters in order to construct functional brain networks. The objective was to enhance the classification accuracy for psychiatric disease and neurological disorders.

All things considered, the knowledge of an appropriate setting of wavelet parameters is fundamental in many wavelet-based signal processing tasks. An all-encompassing analysis aimed to identify optimal settings in all possible contexts of use is a very ambitious task. Nevertheless, this paper provides a first attempt in this direction.

3 Background

The wavelet transform may operate both in the continuous and in the discrete time domain [1], respectively referred to as Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). Its core principle is that it represents a signal as a sum of wavelet functions at different scales and locations. There are two stages: decomposition (or analysis), and reconstruction (or synthesis). Basically, the decomposition stage requires two waveforms: one to represent the high frequencies (a wavelet function $\psi(t)$) and, when the scaling is not iterated infinitely, one to represent the low frequencies (a scaling function $\phi(t)$). These two waveforms are translated and scaled on the time axis. Low scales are able to describe fast variations of the signal, while high scale are used to represent the smooth parts. Instead, translations allow to characterize the content of the signal at different locations. In the end, the result of the decomposition stage, *i.e.*, the

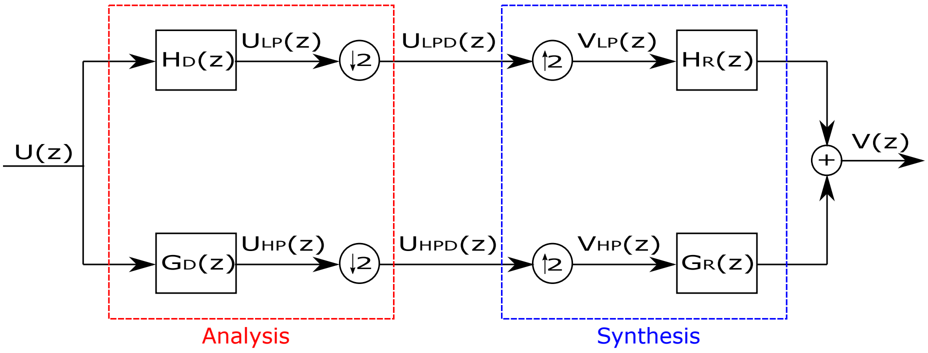


Fig. 1: A 1D, two channels filter bank.

wavelet transform, is a set of coefficients that quantify the similarity between the wavelets, for each position and scale, with the input signal.

Focusing on discrete-time signals, one of the main results for discrete wavelet analysis was the connection with Perfect Reconstruction (PR) filter banks [38]. A filter bank is a structure of filters that allows to separate an input signal into sub-bands. The term “Perfect Reconstruction” refers to the fact that, under specific constraints, the sub-bands can be recombined to reconstruct, without any distortion, the original signal.

The architecture of a 1D, two channels, PR filter bank is shown in Fig. 1 (in the $\mathscr{Z}$-transform domain). First, the input signal $U(z)$ is decomposed into two signals by the low-pass filter $H_D(z)$ and the high-pass filter $G_D(z)$, leading respectively to the signals $U_{LP}(z)$ and $U_{HP}(z)$. Then, both of the filtered versions of $U(z)$ are decimated by a factor of 2, generating $U_{LPD}(z)$ and $U_{HPD}(z)$. The set of these operations represents the analysis phase of the sub-band decomposition process. To reconstruct the original signal, the filtered and down-sampled signals $U_{LPD}(z)$ and $U_{HPD}(z)$ are first numerically interpolated by a factor of 2, producing the signals $V_{LP}(z)$ and $V_{HP}(z)$. Then, the synthesis stage is completed by passing the up-sampled signals through the reconstruction low-pass and high-pass filters, $H_R(z)$ and $G_R(z)$, respectively. The recovered signal $V(z)$ is obtained by summing the two outputs.

Although down-sampling preserves the original sampling rate, it introduces aliasing, since $H_D(z)$ and $G_D(z)$ in general are not ideal, brick-wall filters. Furthermore, there is also amplitude and phase distortion associated with the analysis filters. Nevertheless, by properly designing the filters, a PR filter bank with an l-step delay can be achieved, so that $V(z) = U(z)z^{-l}$. In particular, the two conditions that have to be satisfied are the following:

$$\underbrace{H_R(z)H_D(-z) + G_R(z)G_D(-z) = 0}_{Alias\ cancellation} \tag{1}$$

$$\underbrace{H_R(z)H_D(z) + G_R(z)G_D(z) = 2z^{-l}}_{No\ distortion} \tag{2}$$

By iterating the analysis step, the signal can be decomposed into different sub-bands, leading to a pyramid structure, or dyadic tree. To elaborate, for the

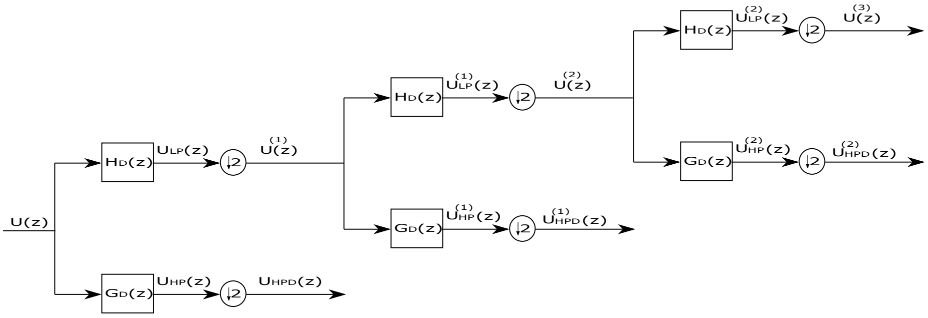


Fig. 2: The analysis stage of a three-levels, 1D filter bank.

second level, the low-pass and sub-sampled signal output by the first level is filtered again using the same low-pass and high-pass filters, *i.e.*, $H_D(z)$ and $G_D(z)$. Then, decimation follows too. This procedure can be repeated for a number of times (a three-levels decomposition is depicted in Fig. 2). The output coefficients tree is known as the Discrete Wavelet Transform, or DWT. The DWT is particularly useful since it allows to achieve a sparse distribution of the signal energy into a small number of wavelet coefficients.

DWT for 2D signals, *e.g.*, images, can be derived from the 1D DWT, adopting a separable framework. The 2D separable DWT is usually implemented as a 1D row transform followed by a 1D column transform. Thus, a wavelet-based multiresolution analysis, as used in image compression, generates a hierarchic pyramidal structure. The number of decomposition levels, *i.e.*, the number of employed 2D filter stages, determines the depth of the sub-band partition. As an example, a three-levels, 2D decomposition is shown in Fig. 3. At the end of each decomposition level, four signals are generated. These signals are obtained by concatenating the low-pass (L) and high-pass (H) filtering in the row direction with the low-pass and high-pass filtering in the column direction. Referring to Fig. 3, after the first decomposition level the four obtained sub-bands are usually named LL, LH, HL and HH. Then, the multiresolution process continues by iterating the decomposition process on the LL sub-band alone, this way generating four more filtered signals, namely LLLL, LLLH, LLHL, and LLHH. Finally, LLLL is decomposed again into four additional sub-bands, producing LLLLLL, LLLLLH, LLLLHL, and LLLLHH. Of course, the process could continue until the desired decomposition depth is reached.

According with the constraints expressed in Eqs. (1) and (2), the wavelet decomposition can be designed with different peculiarities. In particular, the choices related to the wavelet filters, including their length, and the multiresolution depth, lead to DWTs with defined characteristics, namely: approximation order, regularity, smoothness and magnitude response. Based on the application goal and the examined signal nature, some of these properties may play a more important role than others. Generally speaking, it is clear that the employed DWT properties strongly influence the performance of a given wavelet-based task. A more detailed discussion on these and other properties, *e.g.*, orthogonality, and how they can be achieved is deferred until Section 4.4.

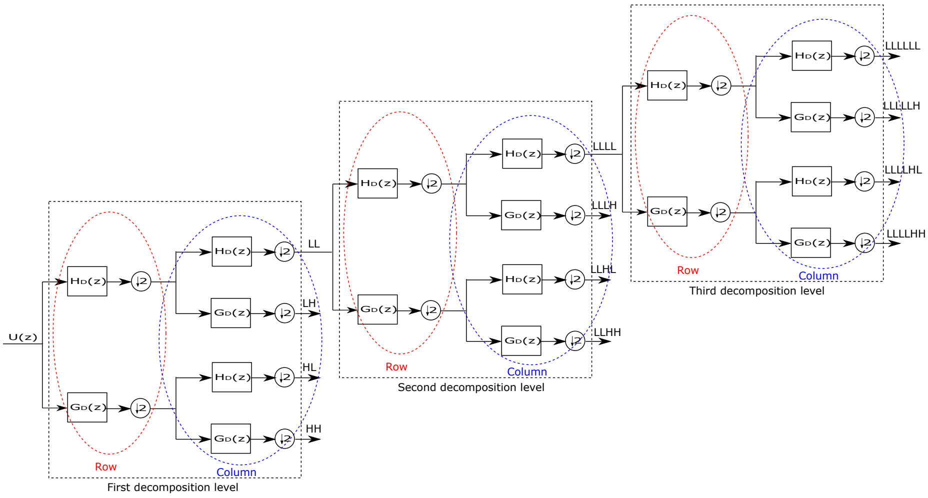

U(z)
HD(z)
GD(z)
↓2
LL
LH
HL
HH
Row
Column
First decomposition level
LLLL
LLLH
LLHL
LLHH
Second decomposition level
LLLLLL
LLLLLH
LLLLHL
LLLLHH
Third decomposition level

Fig. 3: The analysis stage of a three-levels, 2D filter bank.

4 Description of the experiments

In this section, we first present the datasets used in our study, describing the various categories taken into account. Then, the procedure to evaluate the performance for each wavelet setting is illustrated. The results are classified depending on the considered application, namely compression and denoising. Next, the wavelet filters and the corresponding properties considered in this paper are presented. At the end of this section, preliminary results will also show the impact of the number of levels reached in the wavelet decomposition. The experiments have been implemented in MATLAB. The code and the datasets used in the experiments are publicly available: they can be found at [39].

4.1 Dataset selection

It is clear that it is not possible to identify a universal optimal set of wavelet parameters independently by the input data content. Nonetheless, in this work the experiments will cover a large number of variegated datasets, in the hope of deriving a useful set of rules, or at least some hints, from the subsequent experiments.

For 2D signals, *i.e.*, images, the considered datasets have been taken from [40] and [41]. In the former, the images are in a standard definition (SD) quality. They are of various sizes (256×256 pixels, 512×512 pixels, or 1024×1024 pixels), and all of them are gray-scale and 8 bpp (bits per pixel). They are further divided in three categories based on the content type: *miscellaneous* (41 images), *aerial* (30 images) and *textures* (64 images). The second dataset instead contains high definition (HD) quality images coming from a wide variety of sources. They include both standard range (8 bpp) and high dynamic range (16 bpp) images. In total, this dataset comprises 30 images of sizes ranging from 2000×3000 to 7216×5412 pixels.

For the non-visual data, three types of 1D signals have been collected:

- several audio tracks, consisting of various musical genres and vocal sound effects, sampled at CD quality (44.1 kHz, 16 bits per sample);
- a selection of ECG signals taken from the PhysioNet database [42];
- a number of seismic data extracted from the IRIS database [43].

4.2 First application: compression

Here, we report the procedure used to evaluate the compression efficiency of the 2D and 1D DWT. In particular, we describe the coding technique and the quality assessments employed to characterize the performance.

In order to test the compression ability, when a wavelet decomposition is performed on a $r{\times}c$ image I with a precision of b bits per pixel, the Embedded Zerotree Wavelet (EZW) technique is used to encode the wavelet coefficients [9]. Even if more recent coding/compression schemes may be employed for the same goal, *e.g.*, Set Partitioning in Hierarchical Trees (SPIHT), Set Partitioned Embedded Block (SPECK), and Embedded Block Coding with Optimized Truncation (EBCOT), the simplicity of EZW makes it more suitable to appreciate the approximation abilities of the different types of wavelets. Furthermore, it has been shown that by properly modifying the EZW algorithm, it can reach similar, or even superior, performance with respect to the other techniques [44].

To compute the rate-distortion (R-D) curves, as the target rate passed to the encoder is varied, the quality of the uncompressed image $\hat{I}$ at a given rate is evaluated by computing two different metrics. The first is the Peak Signal-to-Noise Ratio (PSNR), which is the most common objective measure, given by:

$$PSNR = 10\log_{10}\left(\frac{D^2}{MSE}\right) \tag{3}$$

where $D{=}2^b{-}1$ is the dynamic range of the pixel values, and b represents the bit depth. The Mean Square Error (MSE) for $r \times c$ images is computed by:

$$MSE = \frac{1}{rc}\sum_{i=0}^{r-1}\sum_{j=0}^{c-1}[I(i,j) - \hat{I}(i,j)]^2 \tag{4}$$

The second metric is the Structural SIMilarity (SSIM) index, used for predicting the perceived quality of digital images and video sequences. The SSIM index is calculated taking overlapping, same-size image windows on matching locations in the reference and the uncompressed image. The quality assessment between two windows, say x and y, is then computed by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5}$$

where μ_x and σ_x are the average and standard deviation of x, μ_y and σ_y are the average and standard deviation of y, and σ_{xy} is the covariance of x and y. The constants c_1 and c_2 are equal to $(k_1D)^2$ and $(k_2D)^2$, respectively, where $k_1{=}0.01$ and $k_2 = 0.03$. Finally, the SSIM metric is taken as the average of $SSIM(x, y)$ across all window locations.

As mentioned, these metrics are associated to the bit rate of the uncompressed image to build the R-D curves. In addition, the Bjøntegaard's metric [45] (*BD* rate) will be used to compare them. It works by computing the average gain in PSNR, or the average per-cent saving in bit rate.

On the other hand, for 1D signals, after decomposing a signal s into S using the DWT, the wavelet expansion coefficients are discarded using a global positive threshold, generating the truncated transform $\tilde{S}$ (we also briefly consider other thresholding strategies as well in the next section). The compression score C_s is given by the percentage of discarded coefficients (namely, that are put to 0). Instead, the recovered energy E_r (that is, the ℓ_2-norm of the reconstructed signal compared to that of the original one, given in percentage) is given by:

$$E_r = 100\frac{||\tilde{S}||}{||S||} \overset{(*)}{=} 100\frac{||\hat{s}||}{||s||} \tag{6}$$

The $(*)$ indicates the hypothesis of orthogonal filters.In that case, E_r can be also evaluated using $\hat{s}$, which is the uncompressed version of s, namely, the signal reconstructed from $\tilde{S}$.

Note that, for every considered dataset, each performance curve represents the average between the outputs associated to all of the elements of the dataset. Standard deviations and/or variance ranges are also provided, where necessary, to attest the statistical significance of the results.

4.3 Second application: denoising

To test the denoising ability, a white, zero-mean, Gaussian noise signal w, with variance σ_w (and thus power $P_w = \sigma_w^2$), is added to the reference signal x with power P_x, generating the noisy signal $y = x + w$. The denoised version of y, say $\hat{x}$, is generated by an empirical Bayesian method with a posterior median threshold rule applied to the wavelet coefficients. The similarity between x and $\hat{x}$ is measured by calculating the MSE between the two signals. Different Signal-to-Noise Ratio ($SNR = P_x/P_w$) values are considered by modifying P_w, the power of the noise added to the signal. Note that a number of noise realizations equals to the cardinality of the considered dataset is generated for each investigated noise level. So, in the end several realizations of noise are taken into account in the overall process, this way giving a statistical soundness to the results.

The same procedure is independently applied both for 1D and 2D signals. As for the compression framework, for every considered dataset, the performance curve represents the average between the outputs associated to all of the elements of the dataset, and other statistical moments like the variance are provided where needed.

4.4 Wavelet properties

In this section, we discuss the wavelet properties that have been investigated in this work, that is orthonormality, filter order and decomposition depth. While doing so, we discuss why we have chosen the wavelet families considered in this work. We will also show with some experiments how the depth of the decomposition can strongly affect the filter performance. Therefore, we will report the optimal number of decomposition levels for each signal type, based on the considered application.

4.4.1 Orthonormality

Orthonormal filters bring to orthonormal wavelet functions, leading to an energy preserving transform. For an orthogonal PR filter bank [46], the synthesis filters are time-reversed versions of the analysis filters, that is to say $H_R(z)=H_D(-z)$ and $G_R(z)=G_D(-z)$. In addition, the high-pass filter is the delayed alternating flip of the low-pass filter, *i.e.*, $G_D(z)=-z^{-L}H_D(-z^{-1})$, where L is the filter length. This means that the entire filter bank is completely defined by just one filter, namely the low-pass analysis filter $H_D(z)$.

One of the advantages of the orthonormality property is that the MSE originated by the quantization of the DWT coefficients is the same as the MSE generated in the reconstructed signal. In other words, orthogonal wavelet filters preserve the energy in the analysis stage. This is a crucial property in different fields of signal processing, *e.g.*, in data compression, where the well-known process "transformation-quantization" is performed. In this case, the energy preservation property allows to straightforwardly design the quantizer in the transform domain. Furthermore, orthogonality allows fast implementation algorithms. For all these reasons, in this paper we consider the Daubechies Wavelets (*db*) and Coiflet Wavelets (*coif*), two classical orthonormal wavelet filters. They are largely used for data compression, *e.g.*, the *coif* wavelet has been used for a fingerprint image compression analysis, as described in [47]. They are also used in data denoising, *e.g.*, the *db* wavelet has been tested for speech denoising, as reported in [48].

Conversely, bi-orthogonal wavelets (*bior*) do not constitute an orthogonal transform. However, differently from the orthonormal case, the bi-orthogonal wavelet functions can be symmetric. So, the associated filters have the important linear phase property, which may be convenient for some application contexts. In the case of a bi-orthogonal PR filter bank, the PR conditions are given by $G_D(z)=H_R(-z)$ and $G_R(z)=H_D(-z)$. Then, in this case we need to design two filters, the analysis and the synthesis low-pass filters, while still satisfying the perfect reconstruction conditions. Generally, bi-orthogonal wavelets play a fundamental role especially in image processing, thanks to their linear phase property. Indeed, they are used both in image compression and denoising, where using bi-orthogonal wavelets typically does not introduce visual distortions in the image.

Bi-orthogonal wavelet transforms can be also designed by using lifting scheme techniques [49]. Lifting schemes enable to construct transforms with predetermined properties, and provide a means for flexible adaptation of the transforms to any problem under consideration. For example, in [50] the authors have proposed a new family of bi-orthogonal wavelet transforms, by inspecting polynomial splines derived in a lifting manner. To obtain different transforms, various combinations of both Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters have been explored for the prediction and update steps used in the lifting methodology. The experiments have been tested for image compression, and the results show comparable performance with respect to the other state-of-the-art wavelets. More recently, in [51, 52] the authors have extended the previous cited work by considering a larger set of configurations, focusing on the search for the best order of the filters. Since their performance is quite interesting for compression, we include some of these new bi-orthogonal wavelets in our overall analysis. However, to keep a fair comparison with respect to the other wavelets, we will just consider filters with a finite impulse response: such filters can be derived from [50].

4.4.2 Filter order and filter length

Each wavelet family can be specified through an integer N that establishes the filter order. Low filter orders lead to a compact support in the time domain, allowing a better time localization and, consequently, a good preservation of content details. On the contrary, high filter orders generate wide functions in the time domain, that are able to achieve a high degree of smoothness, hence a good frequency localization. Moreover, it has to be noted that increasing N causes a deterioration in the implementation efficiency, since it adds complexity in the DWT computation, as shown by the computational efficiency that we discuss in the next section.

The filter length L is determined by the filter order, however, the relation between N and L varies depending on the wavelet family. For example, for the Daubechies Wavelets (*db*) the filter length is $L = 2N$, whereas for the Coiflet Wavelets (*coif*) it is $L = 6N$.

Differently from orthogonal wavelets, bi-orthogonal filters admit distinct orders for decomposition (N_D) and reconstruction (N_R), say, $biorN_D.N_R$. It means that the length of the analysis and synthesis filters can be different too. Generally, for bi-orthogonal wavelets, the filter length is approximately equal to $2N_D + 2$ for decomposition filters and $2N_R + 2$ for reconstruction filters. However, the exact value of L has to be specifically computed for each filter type.

In the following, the filter orders examined in our experiments are listed for each wavelet family. Note that testing a larger number of orders allows to include more wavelets with different characteristics, such as the level of smoothness, so that the different nature of the data may be better represented. To distinguish traditional bi-orthogonal wavelets from bi-orthogonal wavelets implemented in a lifting mode, we add the subscript l to the latter. Therefore, they are:

1. *db*: 1, 2, 3, 4, 5, and 10;
2. *coif*: 1, 2, 3, 4, and 5;
3. *bior*: 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5, 4.4, 5.5, and 6.8.
4. $bior_l$: 4.4, 6.6, and 4.6.

4.4.3 Decomposition level

One additional crucial property of the wavelet transform is the depth of the decomposition, say J, that indicates how many times the input signal is processed through the analysis stage, *i.e.*, filtered and decimated. Thus, this parameter determines the resolution of the lowest level in the wavelet domain. Typically, increasing the decomposition depth allows to distribute the signal energy into a smaller number of transform coefficients, therefore inducing the desired property of sparsity.

However, an unlimited number of levels is not recommended, basically for two reasons. First, there exists an upper bound, that is automatically associated to both the signal size and the filter length. They determine the maximum decomposition level, say J_{max}, beyond which the PR condition stops to be valid. For orthogonal and bi-orthogonal wavelets, $J_{max} = \lfloor \log_2 \frac{M}{L-1} \rfloor$, where M is the 1D signal length (or the smallest dimension, in case of multidimensional signals), and L is the filter length, as mentioned above. Second, increasing too much the number of filtering stages may lead to a saturation in the subsequent coding efficiency.

In what follows, we investigate how the choice of J can as a matter of fact impact the filter performance. In Fig. 4, the R-D curves related to the examined wavelet families are reported for SD *miscellaneous* images[1]. To assure coherence in the results, data are grouped and compared depending on their size, so that J_{max} remains fixed for a given wavelet family and for all the images. Each graph shows a comparison of the performance associated to a specific wavelet filter by varying the number of decompositions J. In the majority of cases, the performance curves show that $J = J_{max}$ leads to the best performance, although there are some exceptions. For example, *db3* exhibits better results for $J = J_{max} - 1$, similarly to *db10* and *coif3*, specifically for high rates. Furthermore, at high rates *bior1.3* and *bior3.1* perform better when $J = J_{max} - 2$.

Proceeding with the investigation, Fig. 5 shows the R-D curves related to the HD standard precision image set[2]. $J = J_{max}$ still usually provides the best results, however there are some exceptions in this case as well. For example, we can observe that for *bior3.1* the decomposition levels $J = 3, 4, 5$ appear to be slightly more performing than $J = J_{max} = 8$. Also, *bior3.7* and *db10* give better performance when the multiresolution analysis stops at $J_{max} - 1$ or $J_{max} - 2$.

To summarize, the optimal decomposition levels number for each wavelet filter is reported in Table 1. It shows the performance with respect to all the above mentioned image datasets, namely *miscellaneous*, *aerial* and *textures* SD images, and both standard and high dynamic range for HD images. When two values are reported for a given wavelet filter, the top one refers to low rates, whereas the bottom one refers to high rates. The threshold between low and high rates is defined as 2 bpp for SD images and 0.5 bpp for HD standard dynamic range images, respectively, chosen empirically where the performance curves intersect. The decomposition levels given in Table 1 will be used for the wavelet filter comparison test, described in the next section. Note that for *Aerials* and *Textures* images, J_{max} is always the best decomposition level, independently of the wavelet filter.

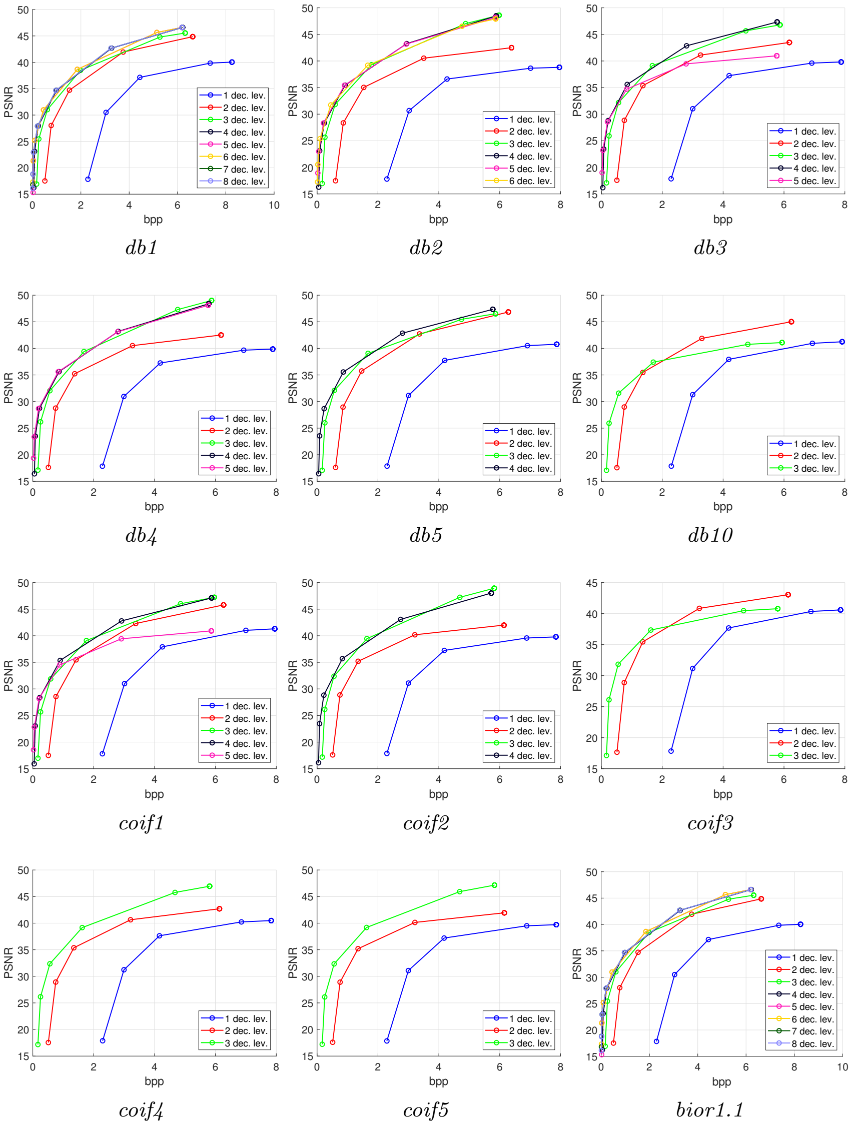
The same study has been conducted on 1D signals (audio, ECG, and seismic data) as well. The corresponding R-D curves can be generated through the code in [39]. In these cases, $J = J_{max}$ always provides the optimal decomposition level. This result is not surprising, since a global positive threshold to zero the wavelet coefficients is used. In fact, the finer is the resolution, the more sparse are the wavelet coefficients. For similar reasons, for denoising the maximum decomposition level will also provide the best performance, independently from the wavelet family. Therefore, for 1D signal compression and 1D/2D signal denoising, the decomposition levels number will be set to $J = J_{max}$, for all the wavelet filters. We defer further discussion on these outcomes until Section 5.4.

Note that we have chosen a global positive threshold since in our experiments it has shown better performance than involving level-dependent thresholds. Furthermore, in the latter case, selecting optimal level-dependent thresholds can be

[1] For conciseness sake, the curves associated to the SD *aerials* and *textures* are not shown here. However, they can be reproduced from [39]. We report that they are consistent with the ones provided in this paper.

[2] Again, refer to [39] for the HD high precision image set performance.

arduous, since the procedure typically needs to set a lot of parameters. However, for the sake of completeness, we mention here that should a different threshold be adopted for each level of the wavelet decomposition, the optimal decomposition levels number could assume different values with respect to J_{max}.



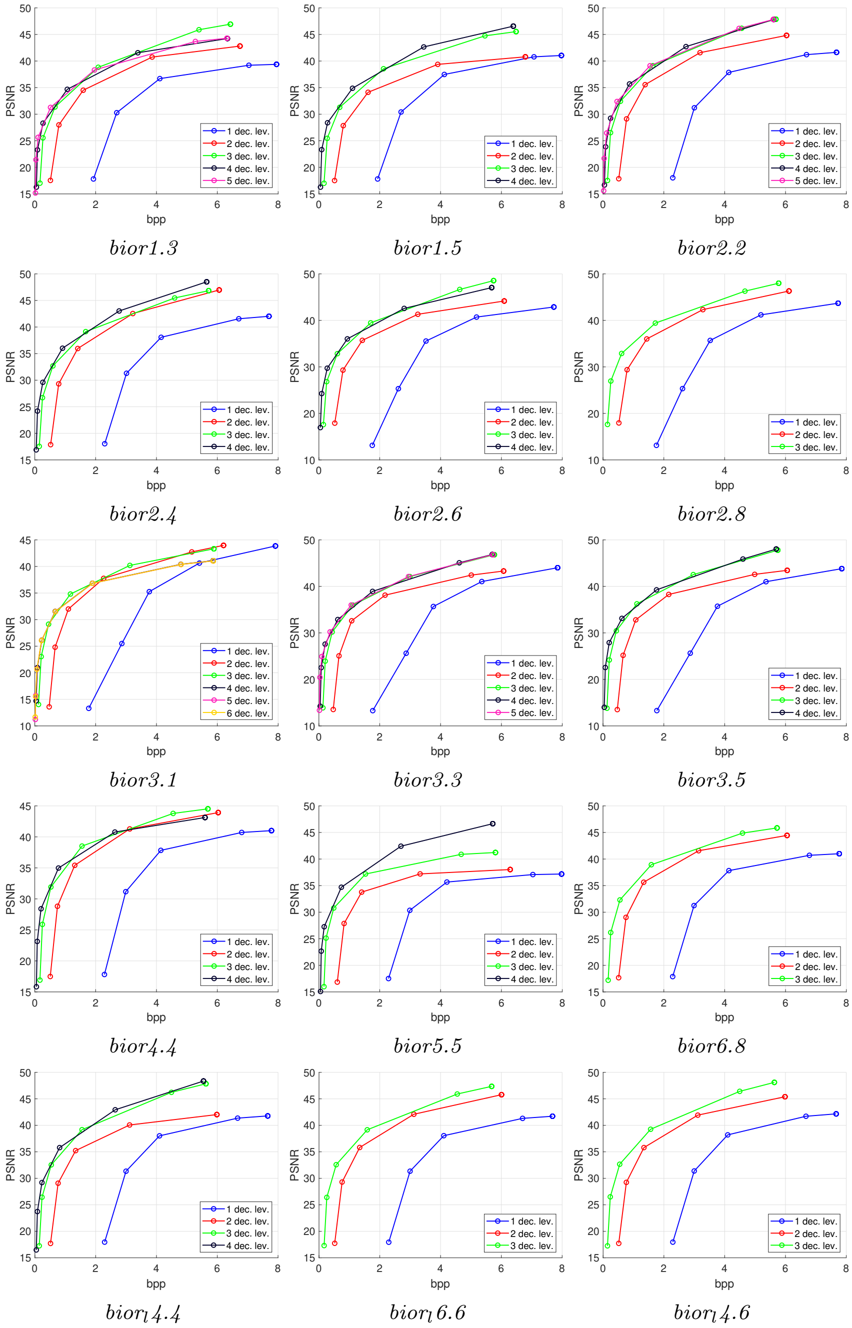
db1 *db2* *db3*

db4 *db5* *db10*

coif1 *coif2* *coif3*

coif4 *coif5* *bior1.1*

Fig. 4: Image compression performance curves for different types of filter wavelets, for different values of J. These curves refer to SD 256 × 256 *Miscellaneous* images.



bior1.3 *bior1.5* *bior2.2*

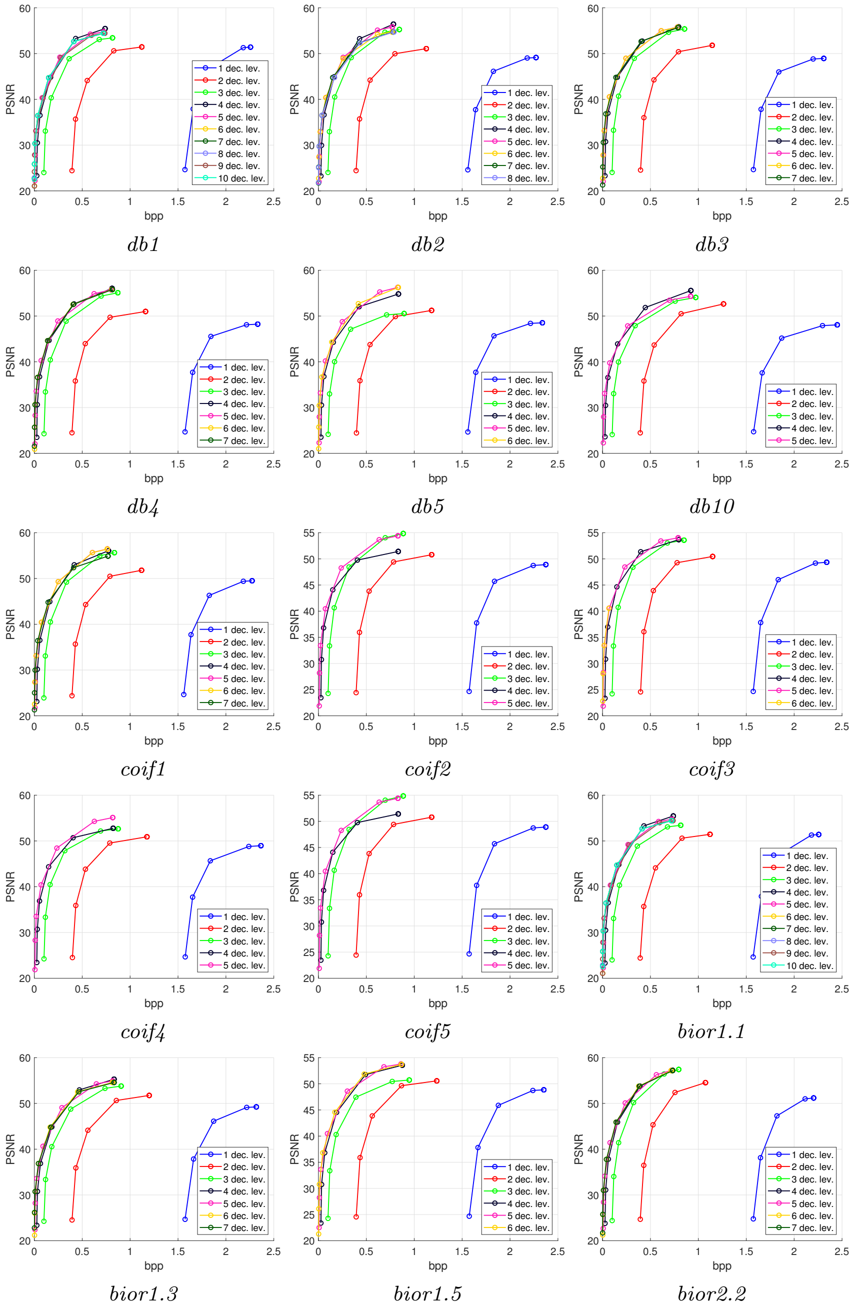
bior2.4 *bior2.6* *bior2.8*

bior3.1 *bior3.3* *bior3.5*

bior4.4 *bior5.5* *bior6.8*

$bior_l 4.4$ $bior_l 6.6$ $bior_l 4.6$

Fig. 4: (cont'd) Image compression performance curves for different types of filter wavelets, for different values of J. These curves refer to SD 256×256 *Miscellaneous* images.



db1 *db2* *db3*

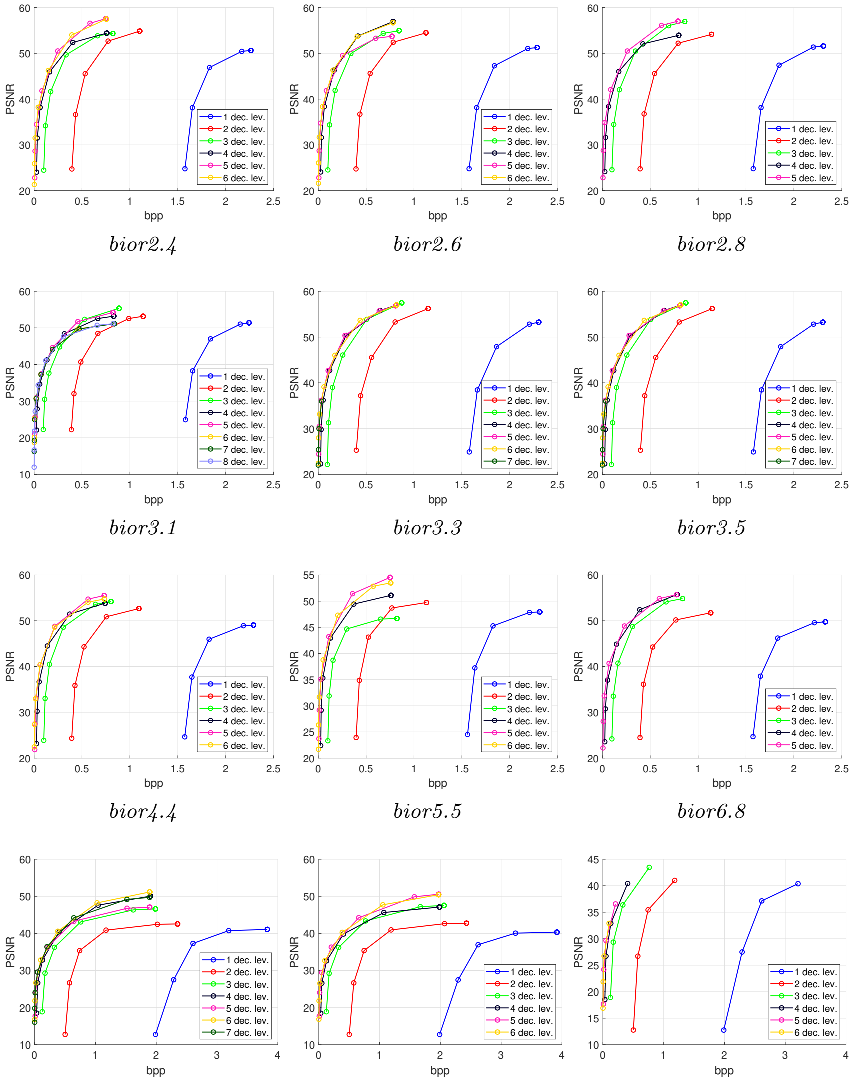
db4 *db5* *db10*

coif1 *coif2* *coif3*

coif4 *coif5* *bior1.1*

bior1.3 *bior1.5* *bior2.2*

Fig. 5: Image compression performance for various filter wavelets, and for different J values. These curves refer to HD standard precision 2048×2048 images.



bior2.4 *bior2.6* *bior2.8*

bior3.1 *bior3.3* *bior3.5*

bior4.4 *bior5.5* *bior6.8*

$bior_l$4.4 *$bior_l$6.6* *$bior_l$4.6*

Fig. 5: (cont'd) Image compression performance for various filter wavelets, and for different J values. These curves refer to HD standard precision 2048×2048 images.

As an example of such behavior, we discuss the case of denoising when the level-dependent thresholds are computed using the Birgé-Massart strategy with sparsity parameter set to 3. In this case, the optimal decomposition level always varies between 2 and 3, and often J_{max} actually leads to much poorer performance. In Fig. 6a, the curves associated to the high dynamic range and HD images dataset are shown as a demonstrative example. As usual, different local thresholds techniques can be tested using the code placed in [39].

Table 1: Optimal decomposition levels for image compression. The values are expressed taking the maximum level J_{max} as the reference for each wavelet filter.

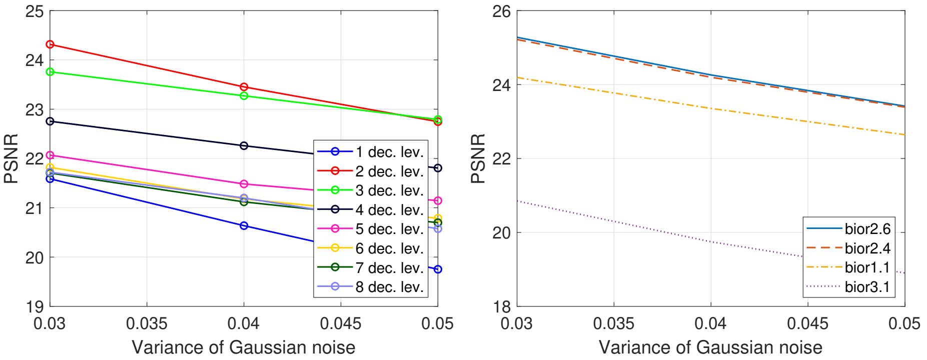
Wavelet filter	**Filter order**	**Dataset**				
		SD images			**HD images**	
		Miscellaneous	*Aerials*	*Textures*	Standard precision	High precision
Daubechies	*db1*	J_{max}	J_{max}	J_{max}	J_{max} $J_{max} - 6$	$J_{max} - 2$
	db2	$J_{max} - 1$			J_{max} $J_{max} - 4$	$J_{max} - 3$
	db3	$J_{max} - 1$			J_{max}	$J_{max} - 1$
	db4	J_{max}			J_{max}	J_{max}
	db5	J_{max}			J_{max}	J_{max}-2
	db10	J_{max} $J_{max} - 1$			J_{max} $J_{max} - 1$	J_{max}
Coiflets	*coif1*	$J_{max} - 1$			$J_{max} - 1$	J_{max}
	coif2	J_{max}			J_{max}	J_{max}-1
	coif3	J_{max} $J_{max} - 1$			$J_{max} - 1$	J_{max}-2
	coif4	J_{max}			J_{max}	J_{max}-2
	coif5	J_{max}			J_{max}	J_{max}-1
Biorthogonal	*bior1.1*	J_{max}			$J_{max} - 6$	J_{max}-4
	bior1.3	J_{max} $J_{max} - 2$			$J_{max} - 2$	$J_{max} - 1$
	bior1.5	J_{max}			J_{max}	J_{max}-1
	bior2.2	$J_{max} - 1$			J_{max}	J_{max}-3
	bior2.4	J_{max}			J_{max}	J_{max}
	bior2.6	J_{max}			J_{max}	J_{max}-2
	bior2.8	J_{max}			J_{max}	J_{max}-1
	bior3.1	J_{max} $J_{max} - 3$			J_{max} $J_{max} - 5$	J_{max}
	bior3.3	J_{max}			J_{max}	J_{max}-2
	bior3.5	J_{max}			J_{max}-1	J_{max}
	bior3.7	J_{max}			J_{max}-1	J_{max}-1
	bior3.9	J_{max}			J_{max} $J_{max} - 1$	J_{max}-3
	bior4.4	J_{max}			$J_{max} - 1$	J_{max}
	bior5.5	J_{max}			$J_{max} - 1$	J_{max}-1
	bior6.8	J_{max}			J_{max}	J_{max}-1
	$bior_l4.4$	J_{max}			J_{max}	J_{max}-2
	$bior_l6.6$	J_{max}			J_{max}-1	J_{max}
	$bior_l4.6$	J_{max}			J_{max}	J_{max}-2

5 Results

In this section, the experimental outcomes of this study are reported. The results are partitioned into three subsections, that is:

1. image compression;
2. image denoising;
3. 1D signal compression and denoising.

The discussion on the results then follows in Section 5.4.



(a) Image denoising performance for various J, using level-dependent thresholds, for the *bior1.1* wavelet. The optimum levels number is 2, which is no longer close to J_{max}.

(b) Denoising performance for $J = 2$ with level-dependent thresholds, for a subset of wavelet families (the two best and worst performing ones).

Fig. 6: An example of employing a local threshold strategy in the analysis. On the left, the impact on the number of levels. On the right, the denoising performance using $J = 2$, which is worse than those obtained with the global threshold (Fig. 8). The curves are associated to the HD high precision 2048×2048 image dataset.

5.1 Image compression

The left column of Fig. 7 presents the R-D curves describing the performance of the wavelet filters in the compression framework. They are presented in the following order: *miscellaneous*, *aerials* and *textures* SD images, followed by standard and high dynamic range, HD images. For each dataset, for conciseness four curves are shown, associated to the two best wavelet families (blue and red lines, in order of performance), and the two worst ones (yellow and violet, in order of performance). Note that each point of the curves represents the average between the PSNR outputs associated to all of the elements of the dataset for a given rate.

For example, for the *miscellaneous* dataset the wavelet *db4* turns out to be the optimal filter, immediately followed by *bior2.2*, whereas *db10* and *bior3.1* lead to the worst performance. It can be observed that the wavelet *bior3.1* represents the worst choice for all the image sets, except for HD high dynamic range images where it precedes *db10*. For this last dataset, the performance gap between the various wavelets is narrow in any case.

To further extend the statistical analysis, in Table 2 the standard deviations are also reported, for some bpp values. The average values are of course the same as those shown in Fig. 7 for the corresponding cases. However, for the sake of readability, just the values corresponding to the “winner” wavelet for each dataset are actually shown. For each rate, both the PSNR (top) and the SSIM (bottom) results are presented.

In Table 3 the Bjøntegaard’s metric is used to measure to what degree the best wavelet outperforms the other filters, in terms of rate saving. Each column is referred to a specific image dataset. A negative percentage means the bit rate decrease related to the best wavelet with respect to the second, the second-to-last and the last wavelet, respectively, for the same PSNR. The reported values show that an accurate choice of the wavelet filter allows to save more than 50% of the bit rate, as happens for *miscellaneous*, *aerials* and HD images. For the sake of completeness, Table 4 reports the associated average gain in PSNR as well. In this case, positive numbers indicate an average increase of PSNR for the same bit rate.

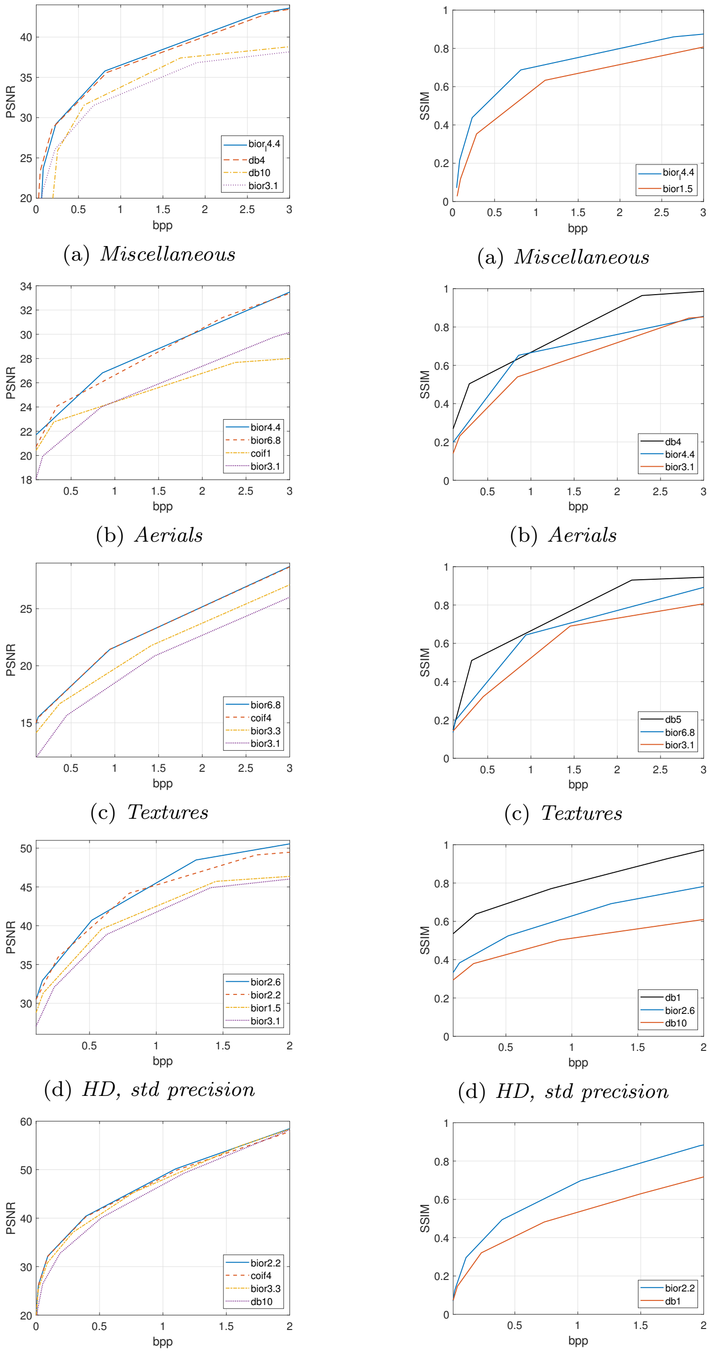
In addition to an objective quality measure, that is, employing metrics such as the PSNR, for digital images a perceptual assessment can be also useful to appreciate the performance of a compression algorithm. In this study, the SSIM index is therefore computed for predicting the perceived quality of the reconstructed image. To this purpose, the right column of Fig. 7 illustrates the SSIM curves, varying the bit rate. For each image dataset, three curves are shown. The first curve indicates the wavelet filter leading to the highest SSIM values (black line). The second one represents the wavelet that resulted the best one in terms of R-D performance (blue line). Finally, the third one points out the filter returning the lowest performance in R-D measured in terms of SSIM (red line). As before, Table 2 reports the standard deviation values.

Examining these results, in general the performance associated to a perceptual quality measure seems in contradiction with respect to that related to an objective quality measure. For example, by keeping *aerials* as the reference image set, the experiments show that *db4* outperforms *bior4.4* by more than 0.1 on average in the normalized SSIM scale. However, the latter is instead the best wavelet in terms of the objective PSNR metric. Interestingly, SD *miscellaneous* and HD, high dynamic range images are not subject to this consideration. The two datasets behave consistently across metrics: indeed, the filters *bior**4.4* and *bior2.2*, respectively, provide the best performing filters both in terms of PSNR and SSIM.

Finally, we provide here a quick survey on the computational time complexity. These times are obtained for running the Matlab code at [39] on a standard desktop computer (Matlab v2019a, Intel Core i7 @3.07GHz, 12GB RAM). For both SD and HD images, *bior3.1* is the fastest filter, while *coif4* is the slowest one. The times when employing the former range from 0.19 ms to 2.61 ms, while the latter ones range from 0.29 ms to 3.08 ms. The just reported maximum and minimum times are obtained when a bit rate equal to 0.2 bpp and 2 bpp are fixed, respectively. Note that these times take into account the whole experimental process, that is, compression and subsequent decompression.

5.2 Image denoising

To recap, in order to test the denoising ability of different wavelet filters, a zero-mean Gaussian white noise with varying variance σ_w is added to the considered image. Then, a denoised version is obtained using a global positive threshold to discard the less significant wavelet coefficients. Such threshold is calculated as the universal threshold of Donoho and Johnstone [53], suitably scaled by a robust estimate of the variance. The values considered for σ_w range from 0.01 to 0.05. Note that such values refer to images whose pixel values go from 0 to 1. Therefore, depending on the image dynamic, the appropriate normalization needs to be applied.



(a) *Miscellaneous* (a) *Miscellaneous*

(b) *Aerials* (b) *Aerials*

(c) *Textures* (c) *Textures*

(d) *HD, std precision* (d) *HD, std precision*

(e) *HD, high precision* (e) *HD, high precision*

R-D curves. R-SSIM curves.

Fig. 7: DWT comparison test for image compression.

Table 2: Average and standard deviation corresponding to the best performing wavelet (reported in the last rows) associated to each dataset for image compression, in terms of PSNR (top value in each box) and SSIM (bottom value).

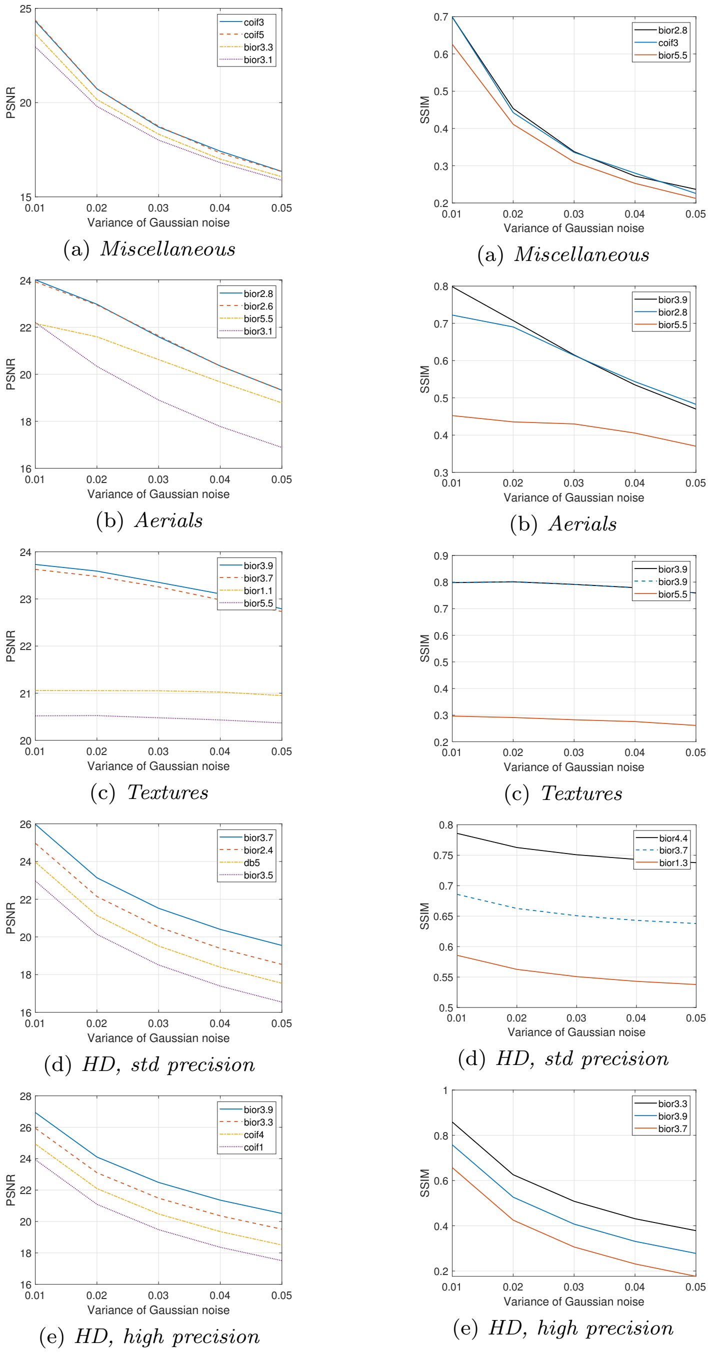
		Dataset				
		SD images			**HD images**	
		Miscellaneous	*Aerials*	*Textures*	*Standard precision*	*High precision*
bpp	0.25	30±1.01	23±0.99	16.9±1.7	36±1.26	37±1.14
		0.45±0.03	0.49±0.03	0.33.9±0.07	0.62±0.04	0.4±0.06
	0.5	32±0.90	24.2±0.82	18±1.34	40,1±1.14	41.8±0.93
		0.58±0.05	0.56±0.03	0.55±0.08	0.69±0.05	0.52±0.05
	1	36±1.18	27.2±0.97	22±1.39	45.2±0.72	49±1.12
		0.71±0.06	0.66±0.03	0.65±0.07	0.8±0.05	0.68±0.05
	1.5	38±1.13	28.8±0.97	23.2±1.58	49±0.91	54±0.87
		0.76±0.04	0.79±0.04	0.79±0.07	0.89±0.05	0.79±0.04
	2	40±1.15	30.3±1.21	25.1±1.26	50.1±1.12	58.1±1.14
		0.81±0.03	0.91±0.04	0.9±0.06	0.97±0.02	0.87±0.05
PSNR	**Wavelet**	$bior_l4.4$	*bior4.4*	*bior6.8*	*bior2.6*	*bior2.2*
SSIM		$bior_l4.4$	*db4*	*db5*	*db1*	*bior2.2*

Table 3: *BD* metric (delta rate) associated to Fig. 7, to compare the R-D curves.

Wavelet rank	**Dataset**				
	SD images			**HD images**	
	Miscellaneous	*Aerials*	*Textures*	Standard precision	High precision
1st	$bior_l4.4$	*bior4.4*	*bior6.8*	*bior2.6*	*bior5.5*
2nd	*db4*	*bior6.8*	*coif4*	*bior2.2*	*bior2.6*
	-5.96%	-15.38%	-0.38%	-4.71%	-3.35%
...	...	...	...	...	...
Second-to-last	*db10*	*coif1*	*bior3.3*	*bior1.5*	*bior3.1*
	-53.49%	-39.28%	-16.79	-31.21%	-33.38%
Last	*bior3.1*	*bior3.1*	*bior3.1*	*bior3.1*	*db1*
	-57.95%	-55.65%	-40.74	-51.42%	-52.21%

Table 4: *BD* metric (delta PSNR) associated to Fig. 7, to compare the R-D curves.

Wavelet rank	**Dataset**				
	SD images			**HD images**	
	Miscellaneous	*Aerials*	*Textures*	Standard precision	High precision
1st	$bior_l4.4$	*bior4.4*	*bior6.8*	*bior2.6*	*bior5.5*
2nd	*db4*	*bior6.8*	*coif4*	*bior2.2*	*bior2.6*
	0.32	0.35	0.02	0.1	0.21
...	...	...	...	...	...
Second-to-last	*db10*	*coif1*	*bior3.3*	*bior1.5*	*bior3.1*
	3.68	2.71	1.43	1.92	2.02
Last	*bior3.1*	*bior3.1*	*bior3.1*	*bior3.1*	*db1*
	3.97	3.1	2.95	3.6	3.4



(a) *Miscellaneous*

(b) *Aerials*

(c) *Textures*

(d) *HD, std precision*

(e) *HD, high precision*

σ_w-PSNR curves.

(a) *Miscellaneous*

(b) *Aerials*

(c) *Textures*

(d) *HD, std precision*

(e) *HD, high precision*

σ_w-SSIM curves.

Fig. 8: Comparison test for image denoising.

Table 5: Average and standard deviation corresponding to the best performing wavelet (reported in the last rows) associated to each dataset for image denoising, in terms of PSNR (top value in each box) and SSIM (bottom value).

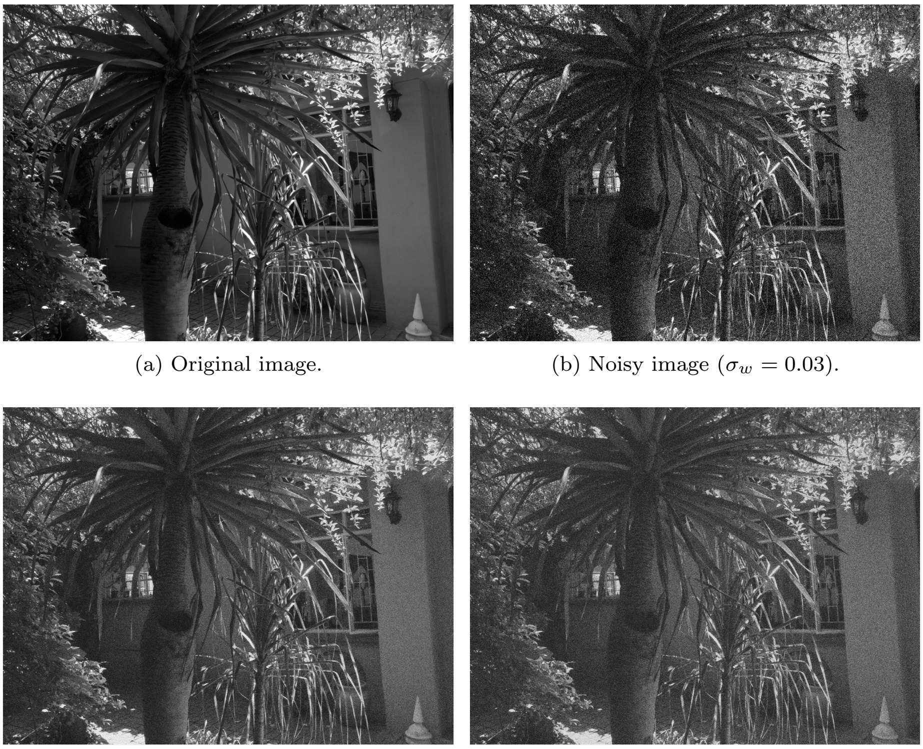
		Dataset				
		SD images			**HD images**	
		Miscellaneous	*Aerials*	*Textures*	*Standard precision*	*High precision*
σ_w	0.01	24.1±0.98	24.1±1.19	23.8±1.41	26±0.9	27±0.97
		0.71±0.06	0.81±0.07	0.8.9±0.05	0.78±0.07	0.85±0.04
	0.02	21.2±1.18	23.1±0.91	23.6±1.24	23.1±1.1	24.1±0.85
		0.46±0.11	0.71±0.09	0.8±0.07	0.76±0.08	0.62±0.09
	0.03	18.5±0.91	21.6±1.05	23.3±1.34	21.6±0.92	22.5±1.03
		0.34±0.07	0.62±0.11	0.79±0.08	0.75±0.1	0.52±0.12
	0.04	17.2±1.12	20.3±0.77	23.1±1.08	20.4±0.96	21.5±0.97
		0.27±0.09	0.53±0.08	0.78±0.07	0.74±0.1	0.43±0.14
	0.05	16.5±0.82	19.5±0.72	22.8±1.36	19.6±1.02	20.4±1.11
		0.24±0.05	0.47±0.11	0.76±0.07	0.73±0.1	0.39±0.14
PSNR	**Wavelet**	*coif3*	*bior2.8*	*bior3.9*	*bior3.7*	*bior3.9*
SSIM		*bior2.8*	*bior3.9*	*bior3.9*	*bior4.4*	*bior3.3*

In Fig. 8, the outcomes of the denoising experiments are shown. Similarly to the compression case described earlier, the results are depicted in two columns. In the left column, the curves describing the PSNR for different σ_w are reported, limiting the plots to the two best and worst performing filters. In the right column, the SSIM value are indicated for the best and worst performing wavelets, in addition to the best one in terms of PSNR.

Similarly to what we have done for the compression case, in order to provide additional statistical significance to the result, Table 5 further reports the standard deviation values associated to the best performing wavelet, for each dataset and for both metrics. The values are reported for various σ_w. Note that the average values can be also extracted from Fig. 8.

As we already mentioned, the global positive threshold strategy has been selected since it has achieved higher performance than involving a level-dependent threshold. However, as an example, Fig. 6b illustrates the PSNR performance curves when a local threshold is applied on the high dynamic range and HD images datasets. In this case, the wavelet decomposition stops at the second level for each family. In fact, recall from Section 4.4.3 and Fig. 6a that this represents the optimal decomposition level in this case.

To provide another way to appreciate to which extent the filter can impact the efficiency of wavelet denoising, a visual comparison is illustrated in Fig. 9. In particular, Fig. 9a shows a reference image acquired from the HD standard dynamic range image dataset. A 2D Gaussian noise with $\sigma_w = 0.03$ is added to the original image, generating the noisy version depicted in Fig. 9b. Finally, Figs. 9c and 9d depict the denoised images generated by using the *bior3.7* and *bior3.5* wavelets. As indicated by the left column of Fig. 8, they are the best and worst performing filters, respectively. Notably, both the PSNR and SSIM metrics strongly indicate a superior performance when *bior3.7* is employed in the



(a) Original image.

(b) Noisy image ($\sigma_w = 0.03$).

(c) Input image after denoising, using *bior3.7*. PSNR = 24.54 - SSIM = 0.68.

(d) Input image after denoising, using *bior3.5*. PSNR = 22.98 - SSIM = 0.56.

Fig. 9: Visual comparison between the denoised images. The most and less performing wavelets are employed, that is, *bior3.7* and *bior3.5*, respectively.

denoising procedure in place of *bior3.5*. The different perceptual quality of the two denoised images clearly exemplifies how the correct choice of wavelet parameters can be crucial, and thus how a properly wavelet filter setting allows to exploit the characteristics of the image.

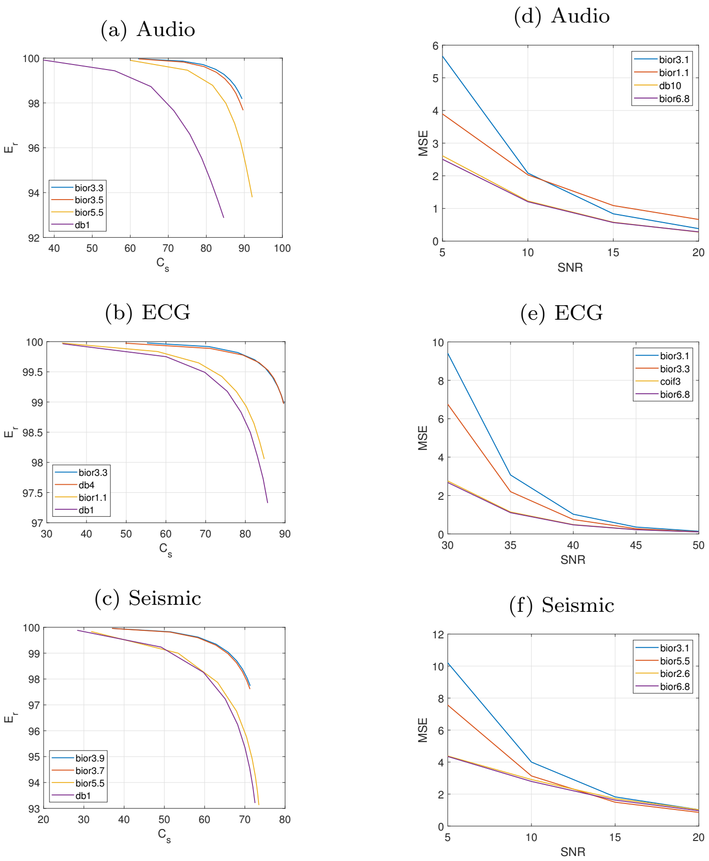
As we did for image compression, we include here a brief analysis on the time efficiency. For both SD and HD images, *db3* results to be the fastest wavelet, while *coif3* is the slowest one. On average, the former takes 51.1 μs, while the latter takes about 89.3 μs, of course independently by the noise variance.

5.3 1D signals compression and denosing

In Fig. 10 the results obtained for 1D signals are reported. In particular, the left column shows the compression curves for audio, ECG and seismic signals. They depict the relation between the percentage of wavelet coefficients that have been discarded, *i.e.*, the compression score C_s, and the recovered energy E_r. It can be noted that the wavelet *bior3.3* outperforms the others filters for audio and ECG

Table 6: Standard deviation range associated to the curves of the best performing wavelets reported in Fig. 10.

Application	**Dataset (1D signals)**		
	Audio	ECG	Seismic
Compression	*bior3.3*	*bior3.3*	*bior3.9*
	0.1 - 0.9	0.1 - 0.4	0.1 - 0.8
Denoising	*bior3.1*	*bior3.1*	*bior3.1*
	0.3 - 1.5	0.2 - 1.6	0.5 - 1.8


(a) Audio

(b) ECG

(c) Seismic

(d) Audio

(e) ECG

(f) Seismic

Compression.

Denoising.

Fig. 10: DWT comparison test for 1D signals compression and denoising.

signals, whereas *bior3.9* achieves the best result for seismic data. Interestingly, the wavelet *db1* completely fails in all of the datasets.

Next, in the experiments aimed to test the denoising ability of the wavelet filters, the 1D white Gaussian noise is added to the original signal. As usual, we vary the SNR, the power ratio between the original signal and the noise. The considered values for the SNR range from 5dB to 20dB for audio and seismic signals, and from 30dB to 50dB for ECG signals. This choice gives the opportunity to visualize comparable values of MSE for every dataset: in fact, ECG signals are usually much more sensitive to noise. Therefore, in the right column of Fig. 10 the denoising curves are depicted. Interestingly, both the “winner” wavelet, namely *bior3.1*, and the less performing one, namely *bior6.8*, are shared for all three types of signal. This evidence strengthens both the adeptness and the inefficiency of such wavelet types for processing generic 1D signals in tasks.

In Table 6, the standard deviation ranges corresponding to the best performing wavelet curves (given above each value), for both applications and for each dataset, are reported. Since they are always quite narrow, for conciseness we have chosen to just indicate the minimum and maximum standard deviation found across every working point of the corresponding curves.

5.4 Discussion

The understanding of why a certain filter is optimal in a specific task for a given input data is of course a challenging problem. However, this work throws some light on which families of filters are adapted (or not adapted) based on the signal type and the considered task, by suggesting correct and adverse parameter setting for wavelet-based applications.

Generally speaking, the results show that an accurate selection of the wavelet parameters notably affects the performance for the above considered tasks. Tables 7 and 8 review the conclusions that we have drawn from our experiments. They report the better and worst wavelet selection for each dataset, based on objective and subjective quality measures.

The results in the compression framework seem to confirm that the particular image type and content have a strong influence on the most performing filter. That is because a suitable selection of wavelet filters allows to exploit the particular image set characteristics, and so it is possible to obtain a sparser representation in the transform domain. Furthermore, we can also confirm that the PSNR-based performance evaluation can be misleading when perceptual quality is sought after. This in yet another indicator that the best wavelet function is highly dependent on the application type. However, usually bi-orthogonal wavelets provides the safest bet in this context, since they usually perform very well at all bit rates.

For image denoising also, from the presented results it is possible to observe how the characteristics and the peculiarities of the examined images play a key role in the selection the optimal wavelet filter. Mainly, bi-orthogonal wavelets emerge as a more than valid choice in this context as well. For example, *bior3.9* results to be the best wavelet for *textures* and HD, high precision images in terms of the objective metric. Furthermore, adopting this filter returns the best performance also for *aerials* and *textures* images when considering SSIM as the subjective measure. Interestingly enough, the experiments carried out on *miscellaneous* images report a different and unique behavior. Indeed, the two most performing wavelet filters in terms of PSNR are *coif3* and *coif5*, with very similar performance (the two curves

Table 7: Final recap indicating the best and worst wavelet filters for each dataset and task in terms of quadratic error.

Dataset		Compression		Denoising	
		First	Last	First	Last
SD images	*Miscellaneous*	$bior_{l}4.4$	*bior3.1*	*coif3*	*bior3.1*
	Aerials	*bior4.4*	*bior3.1*	*bior2.8*	*bior3.1*
	Textures	*bior6.8*	*bior3.1*	*bior3.9*	*bior5.5*
HD images	Standard precision	*bior2.6*	*bior3.1*	*bior3.7*	*bior3.5*
	High precision	*bior2.2*	*db10*	*bior3.9*	*coif1*
1D signals	Audio	*bior3.3*	*db1*	*bior3.1*	*bior6.8*
	ECG	*bior3.3*	*db1*	*bior3.1*	*bior6.8*
	Seismic	*bior3.9*	*db1*	*bior3.1*	*bior6.8*

Table 8: Final recap indicating the best and worst wavelet filters for each dataset and task in terms of the perceptual metric.

Dataset		Compression		Denoising	
		First	Last	First	Last
SD images	*Miscellaneous*	$bior_{l}4.4$	*bior1.5*	*bior2.8*	*bior5.5*
	Aerials	*db4*	*bior3.1*	*bior3.9*	*bior5.5*
	Textures	*db5*	*bior3.1*	*bior3.9*	*bior5.5*
HD images	Standard precision	*db1*	*db10*	*bior4.4*	*bior1.3*
	High precision	*bior2.2*	*db1*	*bior3.3*	*bior3.7*

basically overlap). In addition, *coif3* shows high performance from a SSIM point of view too, even if it is slightly outperformed by the *bior2.8* filter. Therefore, though the landscape is more varied in the denoising framework, bi-orthogonal wavelets still represent a valid choice.

Finally, we move on to 1D signals. Here, the conclusions that can be drawn appear to be a bit less data-dependent and more general. As a matter of fact, for data compression *bior3.3* returns the highest performance for both audio and ECG signals (note that for ECG data, also *db4* is an efficient solution), while *bior3.9* outperforms the other wavelet filters when applied to seismic data. On the contrary, *db1* produces the worst results for all the three considered datasets: this fact is not surprising since the waveform of that filter does not fit with the dynamic nature of those signals. As a last note, for denoising application the experimental results lead to a comprehensive conclusion: indeed, *bior3.1* is evidently the best wavelet filter independently by the considered 1D signal datasets.

6 Conclusions

In this paper, a number of wavelet filters have been compared in order to test their efficiency in two important signal processing applications, *i.e.*, compression and denoising. The investigated wavelets have either orthogonal or bi-orthogonal properties. For each of them, it has been experimentally shown how much the choice of the filter influences the performance in the task. In particular, the filter order (or equivalently the filter length) and the decomposition depth are the most crucial. The experiments have been conducted on a large variety of datasets, including standard definition images (aerials, textures, and miscellaneous), high definition images (standard and high dynamic range), and 1D signals (ECG, seismic, and audio).

It is expected that the results, and the accompanying discussion, provided here can be significantly extended to other similar application domains in which wavelet-based representations are adopted. In addition, by adopting the same analysis carried out in this paper, it is possible to choose the best wavelet family for any task at hand. Indeed, this work proves that just choosing the wrong DWT may significantly harm the performance.

Furthermore, the kind of study presented in this paper can be also effectively used as a support for different analysis purposes. For example, one possible target could be to derive a deeper knowledge of the characteristics of the signals considered in this work. In fact, by analyzing the reasons why a particular relation $(signal, application) \leftrightarrows (wavelet)$ works better than another one, which still is a challenging issue, it could be possible to reveal significant and valuable data and filter properties to exploit in different signal processing tasks. We hope that the code at [39] can help to foster research on these and similar research fields.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
2. M. Vetterli, J. Kovačevic, Wavelets and Subband Coding, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.
3. S. Mallat, A wavelet tour of signal processing, Elsevier, 1999.
4. J. Ohm, Three-dimensional subband coding with motion compensation, IEEE Transactions on Image Processing 3 (5) (1994) 559–571.
5. A. S. Lewis, G. Knowles, Image compression using the 2-d wavelet transform, IEEE Transactions on Image Processing 1 (2) (1992) 244–250.
6. M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Transactions on Image Processing 1 (2) (1992) 205–220.
7. R. A. DeVore, B. Jawerth, B. J. Lucier, Image compression through wavelet transform coding, IEEE Transactions on Information Theory 38 (2) (1992) 719–746.
8. B. E. Usevitch, A tutorial on modern lossy wavelet image compression: foundations of jpeg 2000, IEEE Signal Processing Magazine 18 (5) (2001) 22–35.
9. J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing 41 (12) (1993) 3445–3462.

10. A. Said, W. A. Pearlman, et al., A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on circuits and systems for video technology 6 (3) (1996) 243–250.
11. F. Lazzaroni, R. Leonardi, A. Signoroni, High-performance embedded morphological wavelet coding, IEEE Signal Processing Letters 10 (10) (2003) 293–295.
12. M. A. Qureshi, M. Deriche, A new wavelet based efficient image compression algorithm using compressive sensing, Multimedia Tools and Applications 75 (12) (2016) 6737–6754.
13. C. Deng, W. Lin, B. Lee, C. T. Lau, Robust image coding based upon compressive sensing, IEEE Transactions on Multimedia 14 (2) (2012) 278–290.
14. A. Karami, M. Yazdi, G. Mercier, Compression of hyperspectral images using discerete wavelet transform and tucker decomposition, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 5 (2) (2012) 444–450.
15. T. Bruylants, A. Munteanu, P. Schelkens, Wavelet based volumetric medical image compression, Signal Processing: Image Communication 31 (2015) 112 – 133.
16. R. Leonardi, A. Signoroni, Cyclostationary error analysis and filter properties in a 3d wavelet coding framework, Signal Processing Image Communication (2006) 653–675.
17. N. Adami, A. Signoroni, R. Leonardi, State-of-the-art and trends in scalable video compression with wavelet-based approaches, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1238–1255.
18. F. Luisier, C. Vonesch, T. Blu, M. Unser, Fast interscale wavelet denoising of poisson-corrupted images, Signal Processing 90 (2) (2010) 415 – 427.
19. F. Luisier, T. Blu, M. Unser, Image denoising in mixed poisson–gaussian noise, IEEE Transactions on Image Processing 20 (3) (2011) 696–708.
20. S. Parrilli, M. Poderico, C. V. Angelino, L. Verdoliva, A nonlocal sar image denoising algorithm based on llmmse wavelet shrinkage, IEEE Transactions on Geoscience and Remote Sensing 50 (2) (2012) 606–616.
21. C. Lai, C. Tsai, Digital image watermarking using discrete wavelet transform and singular value decomposition, IEEE Transactions on Instrumentation and Measurement 59 (11) (2010) 3060–3063.
22. F. Guerrini, M. Okuda, N. Adami, R. Leonardi, High dynamic range image watermarking robust against tone-mapping operators, IEEE Transactions on Information Forensics and Security 6 (2) (2011) 283–295.
23. H. Demirel, G. Anbarjafari, Image resolution enhancement by using discrete and stationary wavelet decomposition, IEEE Transactions on Image Processing 20 (5) (2011) 1458–1460.
24. H. Demirel, G. Anbarjafari, Discrete wavelet transform-based satellite image resolution enhancement, IEEE Transactions on Geoscience and Remote Sensing 49 (6) (2011) 1997–2004.
25. R. Singh, A. Khare, Fusion of multimodal medical images using daubechies complex wavelet transform – a multiresolution approach, Information Fusion 19 (2014) 49 – 60, special Issue on Information Fusion in Medical Image Computing and Systems.
26. Y. Yang, D. S. Park, S. Huang, N. Rao, Medical image fusion via an effective wavelet-based approach, EURASIP J. Adv. Signal Process 2010 (2010) 44:1–44:13.
27. S. Li, B. Yang, J. Hu, Performance comparison of different multi-resolution transforms for image fusion, Information Fusion 12 (2) (2011) 74 – 84.
28. V. B. K, I. Sengupta, A. Das, An adaptive audio watermarking based on the singular value decomposition in the wavelet domain, Digital Signal Processing 20 (6) (2010) 1547 – 1558.
29. M. A. Kabir, C. Shahnaz, Denoising of ecg signals based on noise reduction algorithms in emd and wavelet domains, Biomedical Signal Processing and Control 7 (5) (2012) 481 – 489.
30. R. J. Martis, U. R. Acharya, L. C. Min, Ecg beat classification using pca, lda, ica and discrete wavelet transform, Biomedical Signal Processing and Control 8 (5) (2013) 437 – 448.
31. S. Gaci, The use of wavelet-based denoising techniques to enhance the first-arrival picking on seismic traces, IEEE Transactions on Geoscience and Remote Sensing 52 (8) (2014) 4558–4563.
32. J. Ma, G. Plonka, H. Chauris, A new sparse representation of seismic data using adaptive easy-path wavelet transform, IEEE Geoscience and Remote Sensing Letters 7 (3) (2010) 540–544.
33. J. D. Villasenor, B. Belzer, J. Liao, Wavelet filter evaluation for image compression, IEEE Transactions on Image Processing 4 (8) (1995) 1053–1060.

34. M. Grgic, M. Ravnjak, B. Zovko-Cihlar, Filter comparison in wavelet transform of still images, in: ISIE '99. Proceedings of the IEEE International Symposium on Industrial Electronics (Cat. No.99TH8465), Vol. 1, 1999, pp. 105–110 vol.1.
35. S. Grgic, M. Grgic, B. Zovko-Cihlar, Performance analysis of image compression using wavelets, IEEE Transactions on Industrial Electronics 48 (3) (2001) 682–695.
36. B. N. Singh, A. K. Tiwari, Optimal selection of wavelet basis function applied to ecg signal denoising, Digital Signal Processing 16 (3) (2006) 275 – 287.
37. Z. Zhang, Q. K. Telesford, C. Giusti, K. O. Lim, D. S. Bassett, Choosing wavelet methods, filters, and lengths for functional brain network construction, PLOS ONE 11 (2016) 1–24.
38. G. Strang, T. Nguyen, Wavelets and filter banks, rev. ed Edition, Wellesley, MA : Wellesley-Cambridge Press, 1997.
39. Alessandro Gnutti, Github repository, `https://github.com/AlessandroGnutti/A-wavelet-filter-comparison-on-multiple-datasets-for-signal-comp-and-den`, [Online; accessed 24-September-2019] (2019).
40. The usc-sipi image database, `http://sipi.usc.edu/database/`, [Online; accessed 24-September-2019] (2019).
41. Image compression benchmark, `http://imagecompression.info/`, [Online; accessed 24-September-2019] (2019).
42. A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) 215–220.
43. Incorporated research institutions for seismology data, `https://www.iris.edu/hq/resource/bb_processing_matlab`, [Online; accessed 24-September-2019] (2019).
44. R. Boujelbene, Y. Jemaa, M. Zribi, A comparative study of recent improvements in wavelet-based image coding schemes, Multimedia Tools and Applications 78 (2019) 1649–1683.
45. G. Bjontegaard, Calculation of average psnr differences between rd-curves, ITU-T VCEG-M33 (2001).
46. M. Vetterli, C. Herley, Wavelets and filter banks: Theory and design, Trans. Sig. Proc. 40 (9) (Sep. 1992).
47. R. Islam, F. Bulbul, S. S. Shanta, Performance analysis of coiflet-type wavelets for a fingerprint image compression by using wavelet and wavelet packet transform, International Journal of Computer Science and Engineering Survey 3 (2012) 79–87.
48. Ing Yann Soon, Soo Ngee Koh, Chai Kiat Yeo, Wavelet for speech denoising, in: TENCON '97 Brisbane - Australia. Proceedings of IEEE TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications (Cat. No.97CH36162), Vol. 2, 1997, pp. 479–482 vol.2.
49. W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, Applied and Computational Harmonic Analysis 3 (2) (1996) 186 – 200.
50. A. Z. Averbuch, V. A. Zheludev, A new family of spline-based biorthogonal wavelet transforms and their application to image compression, IEEE Transactions on Image Processing 13 (7) (2004) 993–1007.
51. R. Boujelbene, Y. Ben Jemaa, M. Zribi, Toward an optimal b-spline wavelet transform for image compression, in: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016, pp. 1–8.
52. R. Boujelbene, Y. B. Jemaa, M. Zribi, An efficient codec for image compression based on spline wavelet transform and improved spiht algorithm, in: 2017 International Conference on High Performance Computing Simulation (HPCS), 2017, pp. 819–825.
53. D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (3) (1994) 425–455.