Which wavelet bases are the best for image denoising ?

Florian Luisier[a], Thierry Blu[a], Brigitte Forster[b] and Michael Unser[a]

[a]Biomedical Imaging Group (BIG), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

[b]Centre for Mathematical Sciences, Munich University of Technology (TUM), Munich, Germany

ABSTRACT

We use a comprehensive set of non-redundant orthogonal wavelet transforms and apply a denoising method called *SUREshrink* in each individual wavelet subband to denoise images corrupted by additive Gaussian white noise. We show that, for various images and a wide range of input noise levels, the orthogonal fractional (α, τ)-B-splines give the best peak signal-to-noise ratio (PSNR), as compared to standard wavelet bases (Daubechies wavelets, symlets and coiflets). Moreover, the selection of the best set (α, τ) can be performed on the MSE estimate (SURE) itself, not on the actual MSE (Oracle).
Finally, the use of complex-valued fractional B-splines leads to even more significant improvements; they also outperform the complex Daubechies wavelets.

Keywords: Wavelet transform, image denoising, wavelet thresholding, wavelets choice

1. INTRODUCTION

In image denoising, the aim is to suppress noise as much as possible while preserving image features. The multiresolution analysis performed by the wavelet transform is a powerful tool used to achieve this goal. Indeed, in the orthonormal wavelet domain, most image information is contained in the largest wavelet coefficients, while the noise is uniformly spread out across all coefficients. Moreover, Gaussian white noise remains white and Gaussian after orthogonal transformation. A good denoising approach thus consists in setting the smallest coefficients to zero and shrinking the remaining ones above a certain threshold. Donoho and Johnstone[1] have proposed a method to find the threshold that minimizes the estimate of the mean squared error. According to this criterion, their *SUREshrink* —also called soft threshold— is optimal.

Many denoising methods have been compared in the literature. However, there are few studies that have systematically investigated the influence of the wavelet base itself for the success of denoising. In this paper, we fill this gap by proposing a comparison between several non-redundant orthogonal wavelet bases according to their performance (**P**eak **S**ignal-to-**N**oise **R**atio) in denoising applications. We evaluate the common Daubechies wavelets and symlets,[2] the coiflets[2] and the orthogonal fractional (α, τ)-B-splines[3] and its complex-valued extension.[4]

In the first part of this paper, we will briefly present the main characteristics of the non-redundant orthogonal wavelet transform, with a particular attention to the construction of the orthogonal fractional (α, τ)-B-splines filterbank. In the second part, we will describe Donoho and Johnstone's *SUREshrink* denoising method. Finally, we will expose our comparison results, which highlight the potential optimality of the orthogonal fractional (α, τ)-B-splines and especially its complex-valued extension. Indeed, we will show that the best orthogonal fractional (α, τ)-B-splines always perform better than the best standard wavelets (Daubechies, symlets and coiflets).

Further author information:
(florian.luisier, thierry.blu, michael.unser)@epfl.ch
forster@ma.tum.de

2. THE NON-REDUNDANT ORTHOGONAL WAVELET TRANSFORM

This section presents a particular type of basis for use in a non-redundant orthogonal wavelet filterbank: the orthogonal fractional (α, τ)-B-splines and its complex-valued extension.

2.1. Principle

Here we only consider non-redundant dyadic orthogonal wavelet transforms. The algorithm consists of an iterated orthogonal filterbank with an analysis and a synthesis part (figure 1). The orthogonality property is achieved by imposing $\tilde{H}(z) = H(\frac{1}{z})$ and $\tilde{G}(z) = G(\frac{1}{z})$ where:

- $\tilde{H}(z)$ is the analysis scaling filter;
- $H(z)$ is the synthesis scaling filter;
- $\tilde{G}(z)$ is the analysis wavelet filter;
- $G(z)$ is the synthesis wavelet filter.

For a perfect reconstruction, the filters must satisfy two conditions:

1. $\tilde{H}(z^{-1})H(z) + \tilde{G}(z^{-1})G(z) = 1$
2. $\tilde{H}(z^{-1})H(-z) + \tilde{G}(z^{-1})G(-z) = 0$

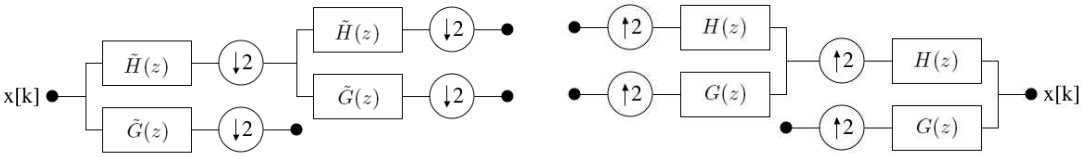

Figure 1. Analysis part (left) and synthesis part (right) of the wavelet transform filterbank.

The key property of an orthogonal wavelet transform is that it transforms Gaussian white noise into Gaussian white noise with the same statistics (mean and variance). Hence, if we call the original image $\mathbf{f}$ and the added Gaussian white noise $\mathbf{n} \sim N(0, \sigma^2)$, the noisy image can be written as $\mathbf{g} = \mathbf{f} + \mathbf{n}$; similarly, we will have in the wavelet domain: $\mathbf{y} = W\mathbf{g} = \mathbf{x} + \mathbf{b}$, with $\mathbf{b} \sim N(0, \sigma^2)$.

2.2. A wide choice of filters

There are many filters which satisfy the perfect reconstruction conditions. For our test (see section 4), we have only retained the most popular ones (Daubechies wavelets, symlets and coiflets)[2] and have chosen to compare them with the orthogonal fractional (α, τ)-B-spline[3] and its complex-valued extension.[4] This section briefly presents these less common wavelet bases.

2.2.1. The fractional (α, τ)-B-splines

The fractional B-spline of real degree α and shift τ is easily expressed in the frequency domain as[3]:

$$\hat{\beta}_\tau^\alpha(\omega) = \Big(\frac{1 - e^{-j\omega}}{j\omega}\Big)^{\frac{\alpha+1}{2}+\tau} \Big(\frac{e^{j\omega} - 1}{j\omega}\Big)^{\frac{\alpha+1}{2}-\tau} \tag{1}$$

These functions satisfy the scaling relation:

$$\hat{\beta}_\tau^\alpha(\omega) = \frac{H_\tau^\alpha(e^{j\frac{\omega}{2}})}{\sqrt{2}} \hat{\beta}_\tau^\alpha(\frac{\omega}{2}) \tag{2}$$

where: $H_\tau^\alpha(e^{j\omega}) = \frac{1}{\sqrt{2}}\Big(2^{-\alpha}(1+e^{j\omega})^{\frac{\alpha+1}{2}-\tau}(1+e^{-j\omega})^{\frac{\alpha+1}{2}+\tau}\Big)$.

In this paper, we only consider orthogonal filterbanks and thus, we orthogonalize the symmetric fractional (α,τ)-B-spline as follows:

$$\hat{\beta}_{\tau,\perp}^\alpha(\omega) = \frac{\hat{\beta}_\tau^\alpha(\omega)}{\sqrt{A_\tau^\alpha(e^{j\omega})}} \tag{3}$$

where

$$A_\tau^\alpha(e^{j\omega}) = \sum_n |\hat{\beta}_\tau^\alpha(\omega+2n\pi)|^2 \tag{4}$$

is the autocorrelation filter of an (α,τ)-B-spline.

Thank to the B-spline scaling relation, one can deduce the orthogonal symmetric fractional (α,τ)-B-spline corresponding synthesis scaling filter:

$$H_{\tau,\perp}^\alpha(z) = H_\tau^\alpha(z)\sqrt{\frac{A_\tau^\alpha(z)}{A_\tau^\alpha(z^2)}} \tag{5}$$

where $z = e^{j\omega}$.

As usual, the frequency response of the generating wavelet synthesis filter is given by:

$$G_{\tau,\perp}^\alpha(z) = -z^{-1}H_{\tau,\perp}^\alpha(-z^{-1}) \tag{6}$$

Despite of the infinite support of the wavelets, the implementation of the orthogonal fractional B-spline wavelet transform can be done exactly using the FFT algorithm,[5] under periodic boundary conditions.

2.2.2. The complex-valued fractional (α,τ)-B-splines

To extend the classical fractional (α,τ)-B-splines to complex-valued functions, we now consider that τ may take complex values. Its imaginary part has the effect of shifting the frequency spectrum of the scaling function, as shown in figure 2. Moreover, this complex extension generates some level of redundancy (real and imaginary part) in the wavelet domain.

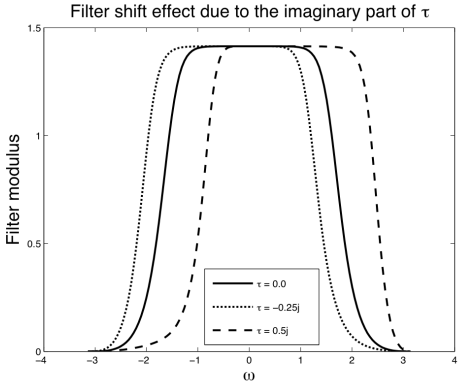

Figure 2. Frequency spectrum shift due to the imaginary part of τ for $\alpha = 2.0$.

In the case of complex-valued B-splines, the orthogonality property is achieved by imposing $\tilde{H}^*(z) = H(\frac{1}{z})$ and $\tilde{G}^*(z) = G(\frac{1}{z})$, where H^* is the filter whose coefficients are complex conjugates of whose of H. Finally, we obtain the filterbank depicted in figure 3 for the complex-valued orthonormal wavelet transform.

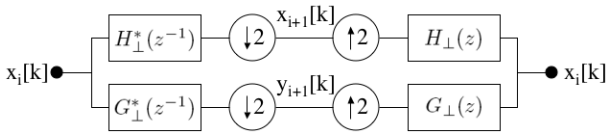

Figure 3. Analysis/synthesis filterbank with complex-valued fractional B-spline filters.

2.2.3. Practical computations

Practically, the infinite sum (4) cannot be computed exactly for most values of α and τ. In order to compute efficiently an accurate estimate of A_τ^α, we use the following acceleration technique, which is obtained in a similar way as for real parameters α, τ (see[5]):

$$A_\tau^\alpha(e^{j2\pi\nu}) \approx U(\nu)e^{4\pi\nu \text{ Im}(\tau)} + V(\nu)e^{-4\pi(1-\nu) \text{ Im}(\tau)} \tag{7}$$

where:

$$\begin{aligned}
0 &\leq \nu = \frac{\omega}{2\pi} \leq 1 \text{ is the normalized frequency} \\
U(\nu) &= \sum_{n=0}^{N-1} |\text{sinc}(\nu+n)|^{2\alpha+1} + \frac{1}{N^{2\alpha}}\left|\frac{\sin(\pi\nu)}{\pi}\right|^{2\alpha+1}\Big[\frac{1}{2\alpha} + \frac{1-2\nu}{2N} \\
&\quad + \frac{1}{12}\frac{(2\alpha+1)(1-6\nu+6\nu^2)}{N^2} - \frac{1}{12}\frac{(2\alpha+1)2\alpha(\nu-1)\nu(2\nu-1)}{N^3}\Big] \\
V(\nu) &= \sum_{n=-N+1}^{-1} |\text{sinc}(\nu+n)|^{2\alpha+1} + \frac{1}{N^{2\alpha}}\left|\frac{\sin(\pi\nu)}{\pi}\right|^{2\alpha+1}\Big[\frac{1}{2\alpha} + \frac{1+2\nu}{2N} \\
&\quad + \frac{1}{12}\frac{(2\alpha+1)(1+6\nu+6\nu^2)}{N^2} + \frac{1}{12}\frac{(2\alpha+1)2\alpha(\nu+1)\nu(2\nu+1)}{N^3}\Big]
\end{aligned}$$

The mathematical analysis of this formula shows indeed that the remainder of the difference between the left-hand side and the right-hand side of (7) is $O\Big(\frac{1}{N^{2\alpha+4}}\Big)$, which ensures an accuracy that is better than 160 dB with $N = 100$ computed terms and for all values of $\alpha > -\frac{1}{2}$.

3. WAVELET THRESHOLDING: *SURESHRINK*

In the following, we consider Donoho's *SUREshrink*[1] as denoising function. This widely used method consists in a non-linear thresholding in the wavelet domain. Indeed, the orthonormal wavelet transform confines most image information in the largest wavelet coefficients, while there is mainly noise in the smaller ones. The idea behind *SUREshrink* is to set to zero all coefficients below a certain threshold value T, while shrinking the remaining ones by this same value; this technique is thus also called *soft thresholding*.

$$\eta(y) = \text{sign}(y)(|y| - T)_+ \tag{8}$$

The soft thresholding function has been shown to be near optimal in the minimax sense.
The threshold value T is then selected so as to minimize the l_2 risk. The mean squared error (MSE) in the image domain is preserved in the wavelet domain, thank to the orthogonality property of the non-redundant wavelet transform. Hence, we can write it as follows:

$$\begin{aligned}
\text{MSE}_{\text{Image domain}} &= \frac{1}{N}\sum_{i=1}^{N}(\hat{f}_i - f_i)^2 \\
&= \frac{1}{N}\sum_{j=1}^{J}\sum_{i=1}^{N_j}(\hat{x}_i^j - x_i^j)^2 \\
&= \text{MSE}_{\text{Wavelet domain}}
\end{aligned} \tag{9}$$

where N is the number of samples; J is the number of channels; N_j is the number of samples in the channel j and x_i^j is the i^{th} sample of the j^{th} channel.
As the non-noisy wavelet coefficients x_i^j are unknown, one needs to estimate the MSE using Stein's unbiased risk estimator (SURE).[6] Its minimization according to our particular estimator $\hat{\mathbf{x}} = \eta(\mathbf{y})$ leads to:

$$\mathrm{SURE}_j(t, \mathbf{y}) = \sigma^2 - \frac{1}{N_j}\Big(2\sigma^2 \cdot \#\{i : |y_i| \leq t\} + \sum_{i=1}^{N_j} \min(|y_i|, t)^2\Big) \tag{10}$$

The resulting threshold is thus:

$$T_j = \mathrm{argmin}\Big(\mathrm{SURE}_j(t, \mathbf{y})\Big) \tag{11}$$

To better adapt to image discontinuities, we will select a new threshold for each wavelet sub-band of successive scales, except the low-pass residual. This method is thus adaptive with respect to the sub-bands.
The SURE principle can also be used to optimize the (α, τ)-parameters of the fractional B-splines. As we can see on table 5, this blind optimization is very accurate (average loss in PSNR ≤ 0.02 dB over eight noise realizations).

4. WAVELET BASES COMPARISON: RESULTS

The aim of our comparison is to point out the differences in term of PSNR due to a particular choice of wavelet bases. The PSNR has been computed using the following formula:

$$\mathrm{PSNR} = 10 \log_{10}\Big(\frac{255^2}{\mathrm{MSE}}\Big) \tag{12}$$

In order to factor out the effect of potential boundary artifacts, we have smoothed the image borders with an Hamming window of size 10 by 10 pixels. The results we show in the following charts have been obtained using the *SUREshrink* denoising method as described in section 3 and the PSNRs have been averaged over eight noise realizations. For the complex-valued (α, τ)-B-splines[4] as well as for the complex extension of the Daubechies,[7] the thresholding has been applied to the modulus of the wavelet coefficients.
Table 1 shows the results obtained with the popular Daubechies[2] (*db*) orthogonal wavelet bases. We can clearly see that the Haar basis (*db1*) is not appropriate for image denoising.
Table 2 shows the results we have reached with the symlets[2] (*sym*) orthogonal wavelet bases. These are a quasi symmetric extension of the Daubechies, which make them more suitable than the classical Daubechies. Globally, the best results are obtained with between four and eight vanishing moments.
Table 3 shows the results of the coiflets[2] (*coif*) orthogonal wavelet bases. These are another extension of the Daubechies wavelets, with vanishing moment conditions both for the wavelets and the scaling functions. They are also more symmetrical than the classical Daubechies (*db*). All these characteristics make them a little more efficient than the symlets. The best results are mainly obtained with four vanishing moments.
Table 4 shows the results we have obtained with the complex extension of the Daubechies[7] (*cxdb*) orthogonal wavelet bases. They are a symmetric version of the classical Daubechies (*db*) and thus perform better than the real-valued Daubechies.
Table 5 displays the best results we have obtained with the fractional B-splines.[3] We can see that the optimization based on the MSE estimate (SURE) gives similar results as the one performed with the actual MSE (oracle), which is an important practical point. The improvements brought by the complex-valued parameters of the fractional B-splines are clearly put in evidence. We must also notice that the best performance of the complex B-splines are always obtained with an imaginary part of the shift τ close to 0.25. Thus, the optimization can only be done for the degree α and the real part of the shift τ with negligible loss in PSNR.
Finally, figure 4 summarizes our results: the complex-valued (α, τ)-B-splines[4] are an efficient wavelet basis for image denoising applications. The gain they induce is on average 0.25 dB which is significant in denoising applications. Among the real-valued orthogonal wavelet bases, the fractional (α, τ)-B-splines[3] give the best results, however they are only slightly better than the best coiflets.[2] They are particularly suitable when a sinc-like basis is required; this is the case when the image spectrum doesn't decay like $|\omega|^{-p}$.

5. CONCLUSION

In this paper, we have proposed a comparison of various wavelet bases following the criterion of their efficiency (PSNR) in image denoising applications. Our results show that the orthogonal fractional (α, τ)-B-splines are an attractive choice of wavelet bases and that their parameters can be optimized by applying the SURE principle. Indeed, a good choice of its real parameters places this wavelet basis at the top of the most popular available wavelet bases such as Daubechies' and its variations. Moreover, the addition of a complex part to these parameters significantly improves the results, which is hardly the case for the complex extension of the Daubechies.

REFERENCES

1. D. L. Donoho and I. M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," *Journal of the American Statistical Association* **90**(432), pp. 1200–1224, 1995.
2. I. Daubechies, "Ten Lectures on Wavelets," *CBMS-NSF Regional Conference Series in Applied Mathematics* **61**, March 1992.
3. T. Blu and M. Unser, "A Complete Family of Scaling Functions: The (α, τ)-Fractional Splines," in *Proceedings of the Twenty-Eighth International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, **VI**, pp. 421–424, (Hong Kong SAR, People's Republic of China), April 6-10, 2003.
4. B. Forster, T. Blu, and M. Unser, "Complex B-Splines and Wavelets," *Second International Conference on Computational Harmonic Analysis, Nineteenth Annual Shanks Lecture (CHA'04)* **61 of Proc.**, May 2004.
5. T. Blu and M. Unser, "The Fractional Spline Wavelet Transform: Definition and Implementation," in *Proceedings of the Twenty-Fifth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00)*, **I**, pp. 512–515, (Istanbul, Turkey), June 5-9, 2000.
6. C. Stein, "Estimation of the Mean of a Multivariate Normal Distribution," *The Annals of Statistics* **9**, pp. 1135–1151, 1981.
7. J.-M. Lina and M. Mayrand, "Complex Daubechies wavelets," *Appl. Comput. Harmonic Anal.* **2**, pp. 219–229, 1995.

DAUBECHIES WAVELETS

Peppers 256×256											
σ	**Input PSNR [db]**	***db1***	***db2***	***db3***	***db4***	***db5***	***db6***	***db7***	***db8***	***db9***	***db10***
5	34.15	36.06	36.47	36.52	**36.61**	36.58	36.54	36.52	36.50	36.45	36.40
10	28.13	31.57	32.24	32.33	**32.45**	32.42	32.40	32.38	32.34	32.30	32.27
15	24.61	29.21	29.98	30.09	**30.23**	**30.23**	30.20	30.19	30.16	30.11	30.09
20	22.11	27.68	28.50	28.61	28.76	**28.77**	28.75	28.75	28.74	28.68	28.65
25	20.17	26.55	27.42	27.53	27.70	27.71	27.70	**27.72**	27.70	27.63	27.61
30	18.59	25.67	26.58	26.69	26.85	**26.89**	26.85	**26.89**	**26.89**	26.82	26.78
50	14.15	23.39	24.38	24.48	24.65	24.63	24.63	24.70	**24.72**	24.64	24.61
House 256×256											
σ	**Input PSNR [db]**	***db1***	***db2***	***db3***	***db4***	***db5***	***db6***	***db7***	***db8***	***db9***	***db10***
5	34.15	36.61	37.10	37.24	**37.26**	37.25	37.24	37.22	37.23	37.21	37.15
10	28.13	32.31	33.07	33.27	33.27	33.25	33.24	33.26	**33.28**	33.23	33.17
15	24.61	30.04	30.94	31.22	31.24	31.20	31.22	31.25	**31.27**	31.19	31.15
20	22.11	28.52	29.55	29.86	29.89	29.85	29.86	29.91	**29.97**	29.84	29.79
25	20.17	27.42	28.52	28.87	28.87	28.82	28.84	28.92	**28.99**	28.83	28.76
30	18.59	26.57	27.70	28.07	28.04	27.98	28.01	28.11	**28.20**	28.01	27.94
50	14.15	24.34	25.50	25.84	25.75	25.62	25.66	25.86	**26.08**	25.75	25.63
MIT 256×256											
σ	**Input PSNR [db]**	***db1***	***db2***	***db3***	***db4***	***db5***	***db6***	***db7***	***db8***	***db9***	***db10***
5	34.15	35.98	36.04	**36.07**	36.03	36.00	35.98	35.94	35.92	35.90	35.87
10	28.13	31.05	31.15	**31.26**	31.23	31.20	31.17	31.13	31.10	31.08	31.06
15	24.61	28.34	28.48	**28.65**	28.64	28.60	28.59	28.54	28.51	28.49	28.46
20	22.11	26.50	26.69	26.91	**26.92**	26.86	26.85	26.82	26.77	26.75	26.75
25	20.17	25.14	25.36	25.63	**25.65**	25.59	25.59	25.55	25.50	25.48	25.49
30	18.59	24.05	24.33	24.63	**24.66**	24.60	24.59	24.57	24.51	24.49	24.50
50	14.15	21.23	21.72	22.05	**22.10**	22.08	22.04	22.05	22.01	21.99	22.03
Bird 256×256											
σ	**Input PSNR [db]**	***db1***	***db2***	***db3***	***db4***	***db5***	***db6***	***db7***	***db8***	***db9***	***db10***
5	34.15	37.39	38.29	38.65	38.76	38.80	38.78	38.78	38.81	**38.82**	38.73
10	28.13	33.14	34.47	34.94	35.11	35.09	35.07	35.09	**35.16**	35.13	35.06
15	24.61	30.95	32.39	32.90	33.06	33.05	33.01	33.05	**33.15**	33.07	33.01
20	22.11	29.53	30.99	31.52	31.65	31.64	31.59	31.61	**31.76**	31.64	31.59
25	20.17	28.48	29.93	30.45	30.58	30.55	30.49	30.51	**30.67**	30.53	30.48
30	18.59	27.65	29.08	29.60	29.71	29.64	29.57	29.61	**29.79**	29.64	29.58
50	14.15	25.43	26.71	27.23	27.31	27.11	27.09	27.21	**27.45**	27.16	27.05

Table 1. Comparison of several Daubechies wavelet bases, coming from Matlab's wavelet toolbox, for various images and noise levels. Non-redundant orthogonal wavelet transform. Number of iterations: 4.

DAUBECHIES SYMLETS

Peppers 256×256										
σ	**Input PSNR [db]**	***sym2***	***sym3***	***sym4***	***sym5***	***sym6***	***sym7***	***sym8***	***sym9***	***sym10***
5	34.15	36.45	36.60	**36.65**	36.63	36.64	**36.65**	36.64	36.59	36.58
10	28.13	32.20	32.45	32.56	32.52	32.56	32.56	**32.57**	32.49	32.50
15	24.61	29.93	30.24	30.36	30.35	30.38	30.39	**30.40**	30.33	30.35
20	22.11	28.44	28.77	28.89	28.92	28.94	28.94	**28.96**	28.89	28.92
25	20.17	27.36	27.70	27.82	27.86	27.88	27.88	**27.92**	27.83	27.86
30	18.59	26.51	26.86	26.98	27.03	27.04	27.04	**27.08**	27.02	27.05
50	14.15	24.31	24.68	24.67	24.82	24.78	24.79	**24.90**	24.76	24.83
House 256×256										
σ	**Input PSNR [db]**	***sym2***	***sym3***	***sym4***	***sym5***	***sym6***	***sym7***	***sym8***	***sym9***	***sym10***
5	34.15	37.10	37.24	37.30	37.31	37.31	**37.32**	37.30	37.28	37.29
10	28.13	33.07	33.27	33.31	33.37	**33.41**	33.40	33.39	33.35	33.40
15	24.61	30.94	31.22	31.25	31.36	**31.42**	31.38	31.39	31.35	31.41
20	22.11	29.55	29.86	29.89	30.02	**30.09**	30.02	30.07	30.03	**30.09**
25	20.17	28.52	28.87	28.87	29.01	**29.09**	29.00	29.06	29.01	**29.09**
30	18.59	27.70	28.07	28.04	28.21	28.27	28.17	28.27	28.20	**28.28**
50	14.15	25.50	25.84	25.70	25.97	25.95	25.86	**26.08**	25.88	26.06
MIT 256×256										
σ	**Input PSNR [db]**	***sym2***	***sym3***	***sym4***	***sym5***	***sym6***	***sym7***	***sym8***	***sym9***	***sym10***
5	34.15	36.04	36.07	**36.09**	36.06	36.06	36.03	36.03	36.02	36.01
10	28.13	31.15	31.26	**31.32**	31.28	31.31	31.24	31.29	31.26	31.27
15	24.61	28.48	28.65	**28.74**	28.67	**28.74**	28.67	28.73	28.66	28.71
20	22.11	26.69	26.91	26.99	26.93	**27.03**	26.96	27.02	26.94	27.00
25	20.17	25.36	25.63	25.71	25.66	**25.75**	25.71	**25.75**	25.67	25.74
30	18.59	24.33	24.63	24.71	24.66	24.75	24.73	**24.77**	24.68	24.75
50	14.15	21.72	22.05	22.10	22.08	22.20	22.22	22.20	22.16	**22.25**
Bird 256×256										
σ	**Input PSNR [db]**	***sym2***	***sym3***	***sym4***	***sym5***	***sym6***	***sym7***	***sym8***	***sym9***	***sym10***
5	34.15	38.29	38.65	38.79	38.81	38.87	**38.95**	38.91	38.84	38.89
10	28.13	34.47	34.94	35.08	35.15	35.25	**35.32**	**35.32**	35.21	35.28
15	24.61	32.39	32.90	33.04	33.11	33.21	33.27	**33.31**	33.17	33.25
20	22.11	30.99	31.52	31.66	31.72	31.77	31.83	**31.92**	31.74	31.82
25	20.17	29.93	30.45	30.58	30.62	30.66	30.70	**30.87**	30.66	30.71
30	18.59	29.08	29.60	29.68	29.75	29.74	29.78	**30.01**	29.73	29.81
50	14.15	26.71	27.23	27.20	27.36	27.18	27.23	**27.65**	27.18	27.39

Table 2. Comparison of several symlets orthogonal bases, coming from Matlab's wavelet toolbox, for various images and noise levels. Non-redundant orthogonal wavelet transform. Number of iterations: 4.

DAUBECHIES COIFLETS

Peppers 256×256						
σ	**Input PSNR [db]**	*coif1*	*coif2*	*coif3*	*coif4*	*coif5*
5	34.15	36.51	**36.66**	36.64	36.63	36.61
10	28.13	32.28	32.57	32.57	**32.58**	32.54
15	24.61	30.00	30.38	30.39	**30.41**	30.38
20	22.11	28.49	28.90	28.93	**28.99**	28.95
25	20.17	27.39	27.84	27.87	**27.93**	27.90
30	18.59	26.54	27.00	27.04	**27.10**	27.06
50	14.15	24.35	24.71	24.79	**24.93**	24.86
House 256×256						
σ	**Input PSNR [db]**	*coif1*	*coif2*	*coif3*	*coif4*	*coif5*
5	34.15	37.10	37.32	37.32	**37.33**	37.29
10	28.13	33.07	33.35	33.41	**33.42**	33.38
15	24.61	30.96	31.30	31.40	**31.42**	31.39
20	22.11	29.57	29.94	30.08	**30.11**	30.07
25	20.17	28.55	28.92	29.08	**29.13**	29.08
30	18.59	27.76	28.10	28.26	**28.34**	28.27
50	14.15	25.55	25.77	25.93	**26.19**	26.03
MIT 256×256						
σ	**Input PSNR [db]**	*coif1*	*coif2*	*coif3*	*coif4*	*coif5*
5	34.15	36.07	**36.10**	36.06	36.05	36.02
10	28.13	31.21	**31.35**	31.32	31.30	31.28
15	24.61	28.54	**28.76**	28.75	28.74	28.72
20	22.11	26.77	**27.03**	**27.03**	**27.03**	27.00
25	20.17	25.45	25.75	25.76	**25.78**	25.75
30	18.59	24.42	24.75	24.75	**24.79**	24.77
50	14.15	21.78	22.14	22.20	22.24	**22.26**
Bird 256×256						
σ	**Input PSNR [db]**	*coif1*	*coif2*	*coif3*	*coif4*	*coif5*
5	34.15	38.32	38.83	38.88	**38.97**	38.89
10	28.13	34.56	35.15	35.24	**35.39**	35.27
15	24.61	32.47	33.11	33.19	**33.39**	33.23
20	22.11	31.07	31.70	31.76	**31.99**	31.81
25	20.17	30.03	30.61	30.66	**30.93**	30.69
30	18.59	29.20	29.72	29.75	**30.06**	29.80
50	14.15	26.87	27.19	27.20	**27.71**	27.36

Table 3. Comparison of several coiflets orthogonal bases, coming from Matlab's wavelet toolbox, for various images and noise levels. Non-redundant orthogonal wavelet transform. Number of iterations: 4.

Peppers 256×256					
σ	**Input PSNR [db]**	***cxdb3***	***cxdb5***	***cxdb7***	***cxdb9***
5	34.15	**36.60**	36.55	36.44	36.37
10	28.13	**32.57**	32.52	32.44	32.33
15	24.61	**30.33**	30.30	30.27	30.23
20	22.11	**28.90**	28.81	28.78	28.71
25	20.17	**27.82**	27.74	27.69	27.65
30	18.59	26.90	**26.91**	26.86	26.78
50	14.15	**24.70**	24.69	24.56	24.50
House 256×256					
σ	**Input PSNR [db]**	***cxdb3***	***cxdb5***	***cxdb7***	***cxdb9***
5	34.15	**37.42**	37.31	37.34	37.19
10	28.13	**33.50**	33.44	33.45	33.34
15	24.61	**31.41**	31.35	31.37	31.25
20	22.11	**30.02**	29.94	29.99	29.86
25	20.17	28.99	**29.02**	28.87	28.81
30	18.59	28.19	**28.24**	28.00	27.93
50	14.15	25.97	**26.08**	25.96	25.75
MIT 256×256					
σ	**Input PSNR [db]**	***cxdb3***	***cxdb5***	***cxdb7***	***cxdb9***
5	34.15	**36.16**	36.10	36.06	36.03
10	28.13	**31.42**	31.28	31.24	31.20
15	24.61	**28.81**	28.78	28.75	28.67
20	22.11	**27.06**	27.00	26.97	26.90
25	20.17	**25.76**	25.68	25.64	25.60
30	18.59	**24.68**	24.66	24.64	24.61
50	14.15	**22.06**	22.02	21.98	21.98
Bird 256×256					
σ	**Input PSNR [db]**	***cxdb3***	***cxdb5***	***cxdb7***	***cxdb9***
5	34.15	38.89	38.92	**39.02**	38.69
10	28.13	35.22	35.20	**35.26**	35.13
15	24.61	33.13	33.12	**33.16**	33.04
20	22.11	**31.77**	**31.77**	31.75	31.59
25	20.17	30.66	**30.69**	30.62	30.50
30	18.59	29.83	**29.91**	29.76	29.59
50	14.15	27.41	**27.64**	27.39	27.32

Table 4. Comparison of several complex versions of Daubechies orthogonal wavelet bases, coming from Matlab's wavelet toolbox, for various images and noise levels. Non-redundant orthogonal wavelet transform. Number of iterations: 4.

Peppers 256×256				
σ	**Input PSNR [db]**	**Best real B-s.**	**A priori best real B-s.**	**Best complex B-s.**
5	34.15	36.68	36.67	**36.90**
10	28.13	32.62	32.61	**32.85**
15	24.61	30.45	30.43	**30.68**
20	22.11	29.01	28.99	**29.21**
25	20.17	27.96	27.94	**28.15**
30	18.59	27.13	27.11	**27.33**
50	14.15	24.95	24.93	**25.12**
House 256×256				
σ	**Input PSNR [db]**	**Best real B-s.**	**A priori best real B-s.**	**Best complex B-s.**
5	34.15	37.35	37.34	**37.65**
10	28.13	33.44	33.42	**33.79**
15	24.61	31.44	31.43	**31.71**
20	22.11	30.14	30.12	**30.36**
25	20.17	29.16	29.14	**29.37**
30	18.59	28.37	28.35	**28.59**
50	14.15	26.21	26.19	**26.43**
MIT 256×256				
σ	**Input PSNR [db]**	**Best real B-s.**	**A priori best real B-s.**	**Best complex B-s.**
5	34.15	36.12	36.12	**36.39**
10	28.13	31.37	31.37	**31.66**
15	24.61	28.80	28.79	**29.10**
20	22.11	27.08	27.07	**27.37**
25	20.17	25.81	25.79	**26.08**
30	18.59	24.82	24.81	**25.05**
50	14.15	22.27	22.26	**22.46**
Bird 256×256				
σ	**Input PSNR [db]**	**Best real B-s.**	**A priori best real B-s.**	**Best complex B-s.**
5	34.15	38.98	38.98	**39.26**
10	28.13	35.42	35.41	**35.61**
15	24.61	33.41	33.41	**33.59**
20	22.11	32.01	32.00	**32.21**
25	20.17	30.95	30.95	**31.17**
30	18.59	30.08	30.08	**30.34**
50	14.15	27.72	27.71	**28.02**

Table 5. Comparison between the best classical orthogonal fractional B-splines, with a posteriori and a priori optimization of its paremeters, and the best complex-valued orthogonal fractional B-splines for various images and noise levels. Non-redundant orthogonal wavelet transform. Iterations: 4.

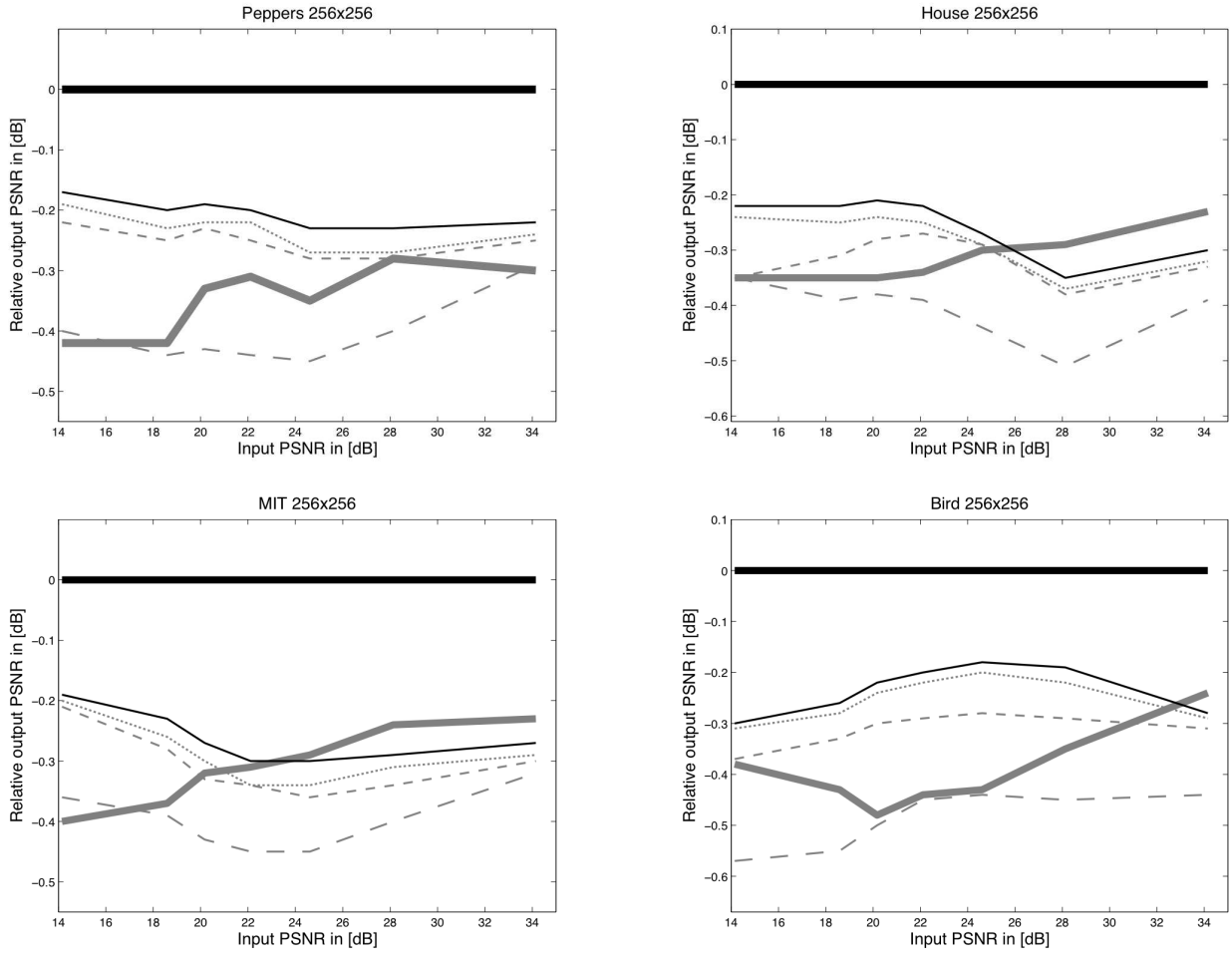

Figure 4. Relative PSNR between the complex-valued orthogonal (α, τ)-B-splines and the best values of other common wavelet bases. The two thickest plain lines are the complex bases: complex-valued orthogonal (α, τ)-B-splines (black) and complex versions of Daubechies (gray). The other lines are the real bases: real-valued orthogonal (α, τ)-B-splines (thin plain black line); coiflets (small-dash gray line); symlets (dash gray line); classical Daubechies (large-dash gray line).