Interpolating Scaling Vectors: Application to Signal and Image Denoising

Karsten Koch*
Philipps–Universität Marburg
FB 12 Mathematik und Informatik
Hans-Meerwein Str., Lahnberge
35032 Marburg
Germany

Abstract. Interpolating scaling vectors and multiwavelets are particularly attractive for application purposes, because they provide a natural preprocessing procedure. This paper is concerned with the application of the interpolating orthonormal scaling vectors constructed in [10] to signal and image denoising. The results of several wavelets and multiwavelets for both universal and vector thresholding are compared.

Key words: Interpolating scaling vectors, multiwavelets, signal/image denoising, wavelet thresholding.

AMS subject classification: 42C40, 65D05, 65T60, 94D

1 Introduction

With computers becoming more and more powerful, signal processing has become a subject of increasing importance. In many fields such as astronomy or medical imaging, the obtained data is often corrupted by noise. This can as well be a result of the data acquisition process as a natural phenomenon such as atmospheric distortion.

Since the fundamental work of Donoho and Johnstone [5, 6] wavelet shrinkage and thresholding have become a commonly used tool for signal and image denoising. Wavelet shrinkage or thresholding can be understood as an estimation of the uncorrupted signal in wavelet space. Donoho and Johnstone showed that this estimator is optimal in a certain sense if, besides some other requirements, the noise is uncorrelated.

*This work has been supported by Deutsche Forschungsgemeinschaft, Grants Da 360/4–1, Da 360/4–2.

In recent studies the concept of wavelet shrinkage has been transferred to the multiwavelet setting. Multiwavelets have become very popular because they provide more flexibility than wavelets in the sense that more desirable properties such as smoothness, orthonormality and symmetry or interpolation properties can be achieved simultaneously. In [14] the shrinkage algorithm of Donoho and Johnstone has been directly adapted to the multiwavelet setting. There the main problem is that the discrete multiwavelet transform (DMWT) acts as a multi–input, multi–output (MIMO) filter bank and therefore the data has to be vectorized. This procedure is called preprocessing and in general alters the characteristic of the noise by inducing correlations. In [7] a different thresholding method is proposed in which some but not all of the noise correlations obtained by preprocessing are taken into account. A comparison of these methods can be found in [15].

The aim of this paper is to analyze the performance of the interpolating scaling vectors and the corresponding multiwavelets constructed in [10] in signal and image denoising. For this purpose we compare the results of several wavelets and multiwavelets, including some well–established candidates. The application of interpolating multiwavelets is particularly attractive, because they provide a natural preprocessing procedure which maintains some characteristics of the data and the DMWT.

2 Multiresolution analysis of multiplicity r

In this section we fix some notations needed throughout this paper. An *r-scaling vector* $\Phi := (\phi_0, \ldots, \phi_{r-1})^T$, $r > 0$, is a vector of $L_2(\mathbb{R})$–functions which satisfies a *matrix refinement equation*

$$\Phi(x) = \sum_{k \in \mathbb{Z}} A_k \Phi(2x - k) \tag{1}$$

with the *mask* $\mathbf{A} := (A_k)_{k \in \mathbb{Z}}$ of real $r \times r$ matrices. If Φ is compactly supported, then it gives rise to a *multiresolution analysis of multiplicity* r (cf. [9]), i. e., the spaces

$$V_j := \overline{\operatorname{span} \{\phi_\rho(2^j \cdot -k),\, k \in \mathbb{Z},\, 0 \le \rho < r\}} \tag{2}$$

form a nested sequence $(V_j)_{j \in \mathbb{Z}}$ of shift–invariant closed subspaces of $L_2(\mathbb{R})$ with

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L_2(\mathbb{R}) \quad \text{and} \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}.$$

This multiresolution analysis can be used to construct *r–multiwavelets*, i. e., vectors $\Psi := (\psi_0, \ldots, \psi_{r-1})^T$ of $L_2(\mathbb{R})$–functions for which

$$\left\{ \psi_0\left(2^j \cdot -k\right), \ldots, \psi_{r-1}\left(2^j \cdot -k\right) \;\middle|\; j, k \in \mathbb{Z} \right\}$$

forms a (Riesz) basis of $L_2(\mathbb{R})$. We have to find Ψ such that

$$W_j := \overline{\operatorname{span} \{\psi_\rho(2^j \cdot -k),\, k \in \mathbb{Z},\, 0 \le \rho < r\}}$$

is an algebraic complement W_j of V_j in V_{j+1}. It follows that Ψ can be represented as

$$\Psi(x) = \sum_{k\in\mathbb{Z}} B_k \Phi(2x-k) \tag{3}$$

with real $r \times r$ matrices B_k, see [4, 12] (scalar case) and [8] (vector case) for details.

If the integer translates of a scaling vector Φ are orthonormal, i. e.,

$$\langle \phi_\rho, \phi_\mu(\cdot - k)\rangle = \delta_{0,k}\delta_{\rho,\mu}, \quad 0 \le \rho, \mu < r, \quad k \in \mathbb{Z},$$

where $\langle\cdot,\cdot\rangle$ denotes the usual L_2–inner product, then Φ is called *orthonormal* and Ψ can be constructed in such a manner that

$$\left\{ \psi_0\left(2^j \cdot -k\right), \ldots, \psi_{r-1}\left(2^j \cdot -k\right) \;\middle|\; j,k \in \mathbb{Z} \right\}$$

forms an orthonormal basis of $L_2(\mathbb{R})$.

Another important property of a wavelet or multiwavelet basis is its approximation order m, i. e., for $f \in H^m$ one has Jackson's inequality

$$\inf_{g\in V_j} \{\|f-g\|_{L_2}\} \le C 2^{-jm} \|f\|_{H^m}$$

with V_j as in (2). As in the scalar case, the approximation order of a scaling vector is closely connected to the vanishing moment properties of the corresponding multiwavelet.

For the case $r = 2$ an interpolation property similar to the scalar case can be defined. A continuous 2–scaling vector Φ is called *interpolating* if for $\rho \in \{0,1\}$ Φ satisfies

$$\phi_\rho\left(\frac{n}{2}\right) = \delta_{\rho,n} = \begin{cases} 0, & \text{if } n \in \mathbb{Z}\setminus\{\rho\} \\ 1, & \text{if } n = \rho. \end{cases} \tag{4}$$

Interpolating scaling vectors satisfy a Shannon–like sampling theorem, i. e., for any function $f \in \operatorname{span}_{k,l}\{\phi_0(\cdot-k), \phi_1(\cdot-l)\}$ the equation

$$f(x) = \sum_{k\in\mathbb{Z}} f(k)\phi_0(x-k) + f(k+1/2)\phi_1(x-k) \tag{5}$$

holds.

The simplest examples of (interpolating) 2–scaling vectors stem from interpolating scaling functions. Suppose ϕ is a scaling function with mask $(b_k)_{k\in\mathbb{Z}}$, given by the refinement equation

$$\phi(x) = \sum_{k\in\mathbb{Z}} b_k \phi(2x-k).$$

Then $\Phi(x) := (\phi(2x), \phi(2x+1))^T$ is always a 2–scaling vector with mask $\mathbf{A}$ determined by

$$A_k = \begin{pmatrix} b_{2k} & b_{2k+1} \\ b_{2(k-1)} & b_{2k-1} \end{pmatrix}.$$

Note that this reinterpretation of ϕ preserves interpolation properties.

3 Denoising via thresholding

In this section we give a short introduction to the concept of signal denoising via wavelet and multiwavelet thresholding.

3.1 Scalar case

In general, a *signal* is a scalar sequence $\mathbf{f} := (f_k)_{0 \le k \le M-1}$ taken to be the observations of a function $f(\cdot)$ at (equidistant) discrete time points. Suppose we observe the noisy signal

$$\mathbf{g} = \mathbf{f} + \xi, \tag{6}$$

where $\mathbf{f}$ is the signal of interest and $\xi := (\xi_k)$ is σ^2 variance, zero mean Gaussian white noise, i.e., the ξ_k are independent and identically distributed like $N(0, \sigma^2)$. Applying an orthonormal discrete wavelet transform (DWT) $\mathcal{M}$ to $\mathbf{g}$ leads to

$$\mathcal{M}\mathbf{g} = \mathcal{M}\mathbf{f} + \sigma\mathcal{M}\xi,$$

where $\mathcal{M}\xi$ is also σ^2 variance, zero mean Gaussian white noise. To obtain an estimate $\widetilde{\mathbf{f}}$ of the uncorrupted signal $\mathbf{f}$ Donoho and Johnstone ([5, 6]) proposed the following algorithm:

1. Assume that the signal consists of the coefficients of the scaling function on a level $J > 0$, i.e. $c_{J,k} := f_k$.

2. Apply the DWT to $\mathbf{f}$ down to level 0 to obtain the scaling coefficients $c_{0,k}$, $k = 0, \ldots, N_0 - 1$, and the wavelet coefficients $d_{j,k}$, $j = 0, \ldots, J-1$, $k = 1, \ldots, N_j - 1$.

3. Apply a thresholding procedure η to the multiwavelet coefficients $d_{j,k}$, $j = 0, \ldots, J-1$, $k = 1, \ldots, N_j - 1$, where η is either the hard η_H or soft thresholding rule η_S.

4. Apply the inverse DWT.

The thresholding rules are given by

$$\eta_H(d, \theta) := \begin{cases} d & \text{if } |d| > \theta \\ 0 & \text{else,} \end{cases} \qquad \eta_S(d, \theta) := \begin{cases} \operatorname{sgn}(d)(|d| - \theta) & \text{if } |d| > \theta \\ 0 & \text{else.} \end{cases}$$

It was shown in [5, 6] that the choice of $\theta = \sigma\sqrt{2 \log \widetilde{N}}$, where $\widetilde{N}$ is the number of wavelet coefficients, is asymptotically optimal. The scaling coefficients remain untouched as they carry the smooth part of the signal.

3.2 Vector case

For an orthonormal r–scaling vector Φ and a corresponding multiwavelet Ψ satisfying (1) and (3) the discrete multiwavelet transform (DMWT) is defined analogous to the discrete wavelet transform, cf. [17] for details. Given a starting coefficient sequence $(C_{J,k})_{0\le k\le N-1}$ of real r–dimensional vectors $\left(C_{J,k}^{(0)},\dots,C_{J,k}^{(r-1)}\right)^T$ with $N=2^J$ the corresponding wavelet coefficients are defined recursively by the *analysis equations*

$$\begin{aligned} C_{j-1,k} &= \sqrt{2}\sum_n A_{n-2k}C_{j,n} \\ D_{j-1,k} &= \sqrt{2}\sum_n B_{n-2k}C_{j,n} \end{aligned}$$

for $j=J,\dots,1$. The inverse DMWT is given by the *synthesis equation*

$$C_{j,k}=\sqrt{2}\sum_n\left(A_{k-2n}^T C_{j-1,n}+B_{k-2n}^T D_{j-1,n}\right).$$

In the following we need the matrix notation of the DMWT, i.e., for a starting sequence $(C_{J,k})$ of length N we consider the $N\times N$ matrix $\mathcal{M}_J$ for which

$$\mathcal{M}_J\begin{pmatrix} C_{J,0}\\ \vdots\\ C_{J,N-1}\end{pmatrix}=\begin{pmatrix} C_{0,0}\\ \vdots\\ C_{0,N_0-1}\\ D_{0,0}\\ \vdots\\ D_{0,N_0-1}\\ D_{1,0}\\ \vdots\\ D_{J-1,N_{J-1}-1}\end{pmatrix}$$

with $N_j=2^{J-j}N$ holds. Before applying the DMWT the signal $\mathbf{f}$ has to be preprocessed to obtain the vector valued starting sequence $(C_{J,k})$. In many cases, applying *identity preprocessing*, i.e., simply cutting the signal $\mathbf{f}$ into vectors by $C_{J,k}:=(f_{rk},\dots,f_{r(k+1)-1})^T$, is not recommendable, as the filter banks corresponding to the multiwavelets lack some important approximation properties. This can be bypassed by using *balanced* multiwavelets, see [11] and [13] for details. For several unbalanced multiwavelets various preprocessing methods have been developed (see, e.g., [14, 15, 16, 17]). Two commonly used preprocessing methods are *approximation preprocessing*, where the $(C_{J,k})$ are the coefficients of the scaling vector on level J, so that

$$f(t)=\sum_k C_{J,k}^T\Phi(t-k) \tag{7}$$

holds, and *repeated row preprocessing*, i.e. $C_{J,k}^{(i)}:=c_i f_k$, $i=0,\dots,r-1$, where $c_0=1$ and the c_i are chosen such that the high–pass output of a constant signal is zero.

In general preprocessing is a linear operation on the data, i.e., it can be represented by an $M \times rN$ matrix $\mathcal{P}$ such that

$$\mathcal{P}\begin{pmatrix} f_0 \\ \vdots \\ f_{M-1} \end{pmatrix} = \begin{pmatrix} C_{J,0}^{(0)} \\ \vdots \\ C_{J,0}^{(r-1)} \\ \vdots \\ C_{J,N-1}^{(0)} \\ \vdots \\ C_{J,N-1}^{(r-1)} \end{pmatrix}.$$

This preprocessing yields a drawback in applying the DMWT for several reasons. On one hand, the preprocessing matrix $\mathcal{P}$ has to be constructed individually for each multiwavelet. On the other hand, the intrinsic properties of a given multiwavelet, such as orthonormality or vanishing moments, need not be carried over to the resulting transform. This is due to the fact that preprocessing can impose artificial correlations between the coefficients of the starting sequence $(C_{J,k})$ as is explained in the following. Consider a noisy signal $\mathbf{g}$ as in (6) and let $\mathcal{M}$ denote an orthonormal DMWT. In general $\mathcal{P}$ is not orthonormal and therefore the covariance matrix

$$\operatorname{cov}(\mathcal{M}\mathcal{P}\xi) = \mathcal{M}\mathcal{P}\operatorname{cov}(\xi)\mathcal{P}^T\mathcal{M}^T = \sigma^2\mathcal{M}\mathcal{P}\mathcal{P}^T\mathcal{M}^T$$

is not the identity, i.e., the $(\mathcal{M}\mathcal{P}\xi)_k$ are still normally distributed but not independent. Consequently the Donoho and Johnstone algorithm can not be applied.

Nevertheless Strela et al. ([14, 15]) proposed a very similar approach in which each component of the wavelet coefficients $D_{j,k}$ is thresholded (soft or hard) with a universal threshold parameter

$$\theta := c\sigma\sqrt{2\log \widetilde{N}}. \tag{8}$$

Here $\widetilde{N}$ is r times the number of wavelet coefficients $D_{j,k}$ and $(c\sigma)^2$ is the average of the variances of the $(\mathcal{M}\mathcal{P}\xi)_k$, i.e., the transformed noise $(\mathcal{M}\mathcal{P}\xi)$ is assumed to be Gaussian white noise with variance $(c\sigma)^2$. Although the asymptotics of Donoho and Johnstone do no longer hold, this method seems to work quite well (cf. [15]).

Another approach called *vector thresholding* has been proposed by Downie and Silverman in [7]. They assume that the vector valued wavelet coefficients $D_{j,k}$ are independently distributed, which corresponds to a block diagonal structure of $\operatorname{cov}(\mathcal{M}\mathcal{P}\xi)$. Let $\Xi_{j,k}$ denote the noisy part of the wavelet coefficient such that

$$D_{j,k} = D_{j,k}^* + \Xi_{j,k},$$

where $D_{j,k}^*$ stems from the uncorrupted signal. Each $\Xi_{j,k}$ is an r–dimensional normally distributed random variable. Then with $\theta_{j,k} := D_{j,k}^T \operatorname{cov}(\Xi_{j,k})^{-1} D_{j,k}$ the $D_{j,k}$ are

thresholded via

$$\eta(D_{j,k}) = \begin{cases} D_{j,k} & \text{if } \theta_{j,k} < 2\log \widetilde{N} \\ \mathbf{0} & \text{else,} \end{cases} \tag{9}$$

where $\widetilde{N}$ denotes the number of wavelet coefficients $D_{j,k}$. The covariance matrices $\mathrm{cov}(\Xi_{j,k})$ are the $r \times r$–blocks on the main diagonal of $\mathrm{cov}(\mathcal{M}\mathcal{P}\xi)$ and a simple computation shows that they are independent of the translation parameter k. Therefore, the matrix $\mathcal{M}_J\mathcal{P}\mathcal{P}^T\mathcal{M}_J^T$ has the form

$$\mathcal{M}_J\mathcal{P}\mathcal{P}^T\mathcal{M}_J^T = \begin{pmatrix} B_0^{(0)} & * & * & * & * \\ * & B_1^{(0)} & * & * & * \\ * & * & B_1^{(1)} & * & * \\ * & * & * & \ddots & * \\ * & * & * & * & B_1^{(J-1)} \end{pmatrix} \tag{10}$$

with $N_j \times N_j$ matrices

$$B_e^{(j)} = \begin{pmatrix} V_e^{(j)} & * & * \\ * & \ddots & * \\ * & * & V_e^{(j)} \end{pmatrix}, \quad e \in \{0,1\}, \quad j = 0, \ldots, J-1, \tag{11}$$

of $r \times r$–block matrices. With this notation $\mathrm{cov}(\Xi_{j,k}) = \sigma^2 V_1^{(j)}$ holds. Also for this thresholding procedure an asymptotic optimality can be shown.

Similar to the scalar case, for the processing of two–dimensional signals or images a direct product approach can be used, i.e., after preprocessing the image, the DMWT is first applied to all rows and then to all columns or vice versa. Accordingly, preprocessing can be done by preprocessing first all rows and then all columns of the image. By this procedure we obtain $r \times r$ matrix valued multiwavelet coefficients $D_{j,\mathbf{k},\mathbf{e}}$, $\mathbf{k} = (k_1, k_2)$, which can be divided into three classes — the classes $\mathbf{e} = (1,1)$ obtained by high–pass filtering all rows and columns, $\mathbf{e} = (0,1)$ obtained by high–pass filtering the rows and low–pass filtering the columns and $\mathbf{e} = (1,0)$ obtained by low–pass filtering the rows and high–pass filtering the columns. To illustrate this let X denote an image, $\mathcal{P}$ a suitable preprocessing matrix and $\mathcal{M}$ a one–dimensional DMWT consisting of two steps, then the two–dimensional DMWT of X is given by

$$\mathrm{DMWT}(X) = \mathcal{M}\mathcal{P}X\mathcal{P}^T\mathcal{M}^T = \left(\begin{array}{c|c|c} C_{0,\mathbf{k}} & D_{0,\mathbf{k},(0,1)} & D_{1,\mathbf{k},(0,1)} \\ \hline D_{0,\mathbf{k},(1,0)} & D_{0,\mathbf{k},(1,1)} & \\ \hline D_{1,\mathbf{k},(1,0)} & & D_{1,\mathbf{k},(1,1)} \end{array}\right).$$

This algorithm has already been described in detail in [15]. There, also a generalization of the universal thresholding concept is given. The entries of the wavelet coefficients are thresholded with a thresholding parameter $\theta := c^2\sigma\sqrt{2\log \widetilde{N}}$, where σ^2

is the variance of the noise, $\widetilde{N}$ is r^2 times the number of wavelet coefficients $D_{j,\mathbf{k},\mathbf{e}}$ and for a given multiwavelet c is the constant already obtained in the scalar case (8).

In [1] the vector thresholding approach has been generalized to two-dimensional signals. The matrix valued wavelet coefficients are reorganized in vectors, i.e.,

$$D_{j,\mathbf{k},\mathbf{e}} = \begin{pmatrix} D_{j,\mathbf{k},\mathbf{e}}^{(0,0)} & \dots & D_{j,\mathbf{k},\mathbf{e}}^{(0,r-1)} \\ \vdots & \ddots & \vdots \\ D_{j,\mathbf{k},\mathbf{e}}^{(r-1,0)} & \dots & D_{j,\mathbf{k},\mathbf{e}}^{(r-1,r-1)} \end{pmatrix} \longrightarrow \begin{pmatrix} D_{j,\mathbf{k},\mathbf{e}}^{(0,0)} \\ \vdots \\ D_{j,\mathbf{k},\mathbf{e}}^{(0,r-1)} \\ \vdots \\ D_{j,\mathbf{k},\mathbf{e}}^{(r-1,0)} \\ \vdots \\ D_{j,\mathbf{k},\mathbf{e}}^{(r-1,r-1)} \end{pmatrix},$$

and then thresholded as above. This method is called *matrix thresholding*. The covariance matrices can be obtained similar to the one–dimensional case. Let $\Xi_{j,\mathbf{k},\mathbf{e}}$ denote the noisy part of the wavelet coefficient such that

$$D_{j,\mathbf{k},\mathbf{e}} = D^*_{j,\mathbf{k},\mathbf{e}} + \Xi_{j,\mathbf{k},\mathbf{e}},$$

where $D^*_{j,\mathbf{k},\mathbf{e}}$ stems from the uncorrupted signal. Then $\operatorname{cov}(\Xi_{j,\mathbf{k},\mathbf{e}})$, $\mathbf{e} = (e_1, e_1)$, is an $r^2 \times r^2$ matrix of the form

$$\operatorname{cov}(\Xi_{j,\mathbf{k},\mathbf{e}}) = V_{e_1}^{(j)} \otimes V_{e_2}^{(j)},$$

where $\otimes$ denotes the usual Kronecker product and $V_e^{(j)}$ as in (10) and (11).

Now let us consider a more specific case. Let Φ denote an interpolating orthonormal 2–scaling vector with compact support and assume the signal $\mathbf{f}$ to be sampled on the half integers $f_k = f(k/2)$. As stated in [10], all interpolating orthonormal 2–scaling vectors with compact support are balanced. Therefore, a suitable starting sequence is given by identity preprocessing, i.e., we can choose $C_{J,k} := (f_{2k}, f_{2k+1})^T$. Due to the sampling theorem (5), this also corresponds to approximation preprocessing. Thus, for interpolating orthonormal 2–scaling vectors with compact support identity and approximation preprocessing coincide. Because of $\mathcal{P} = I$, the transform $\mathcal{MP} = \mathcal{M}$ is orthonormal, therefore Gaussian white noise is preserved and we have $\operatorname{cov}(\xi) = \sigma^2 I$. Consequently, for universal thresholding, the algorithm of Donoho and Johnstone can be applied and we have $c = 1$ in (8). Furthermore, because of $V_e^{(j)} \equiv I$, the vector thresholding procedure is significiantly simplified. So, neither the $V_e^{(j)}$ nor the $\theta_{j,k}$ have to be computed and (9) is reduced

$$\eta(D_{j,k}) = \begin{cases} D_{j,k} & \text{if } \|D_{J,k}\|_2 < \sigma\sqrt{2\log \widetilde{N}} \\ \mathbf{0} & \text{else.} \end{cases}$$

For matrix thresholding an analogous result is obtained.

4 Simulations

scaling vector	support	approx. order	Sobolev regularity
Φ_1	[-1,2]	1	1.00
Φ_2	[-2,3]	2	1.50
Φ_3	[-3,4]	3	1.51
CL	[0,2]	2	1.05

Table 1: Sobolev regularity and approximation order of Φ_n

A series of numerical simulations are carried out to evaluate the performance of the interpolating orthonormal scaling vectors Φ_n and the corresponding interpolating and non–interpolating multiwavelets (denoted by 'i', 'ni', respectively) constructed in [10] in signal and image denoising. For comparison we use the Daubechies wavelets D_2 and D_3 and the symmetric orthonormal Chui–Lian multiwavelet (CL) constructed in [3], which has shown the best results in signal and image denoising in [15] . The properties of these scaling vectors are listed in Table 1.

4.1 One–dimensional signals

For the denoising of one–dimensional signals we consider the test functions 'Blocks', 'Bumps', 'HeaviSine' and 'Doppler' defined in [5]. Each test signal $\mathbf{f}$ is obtained by sampling the corresponding test function at $n = 2048$ equidistant points in $[0, 1]$ and for each simulation Gaussian white noise is added. The variance σ^2 of the noise is chosen in such a way that the signal to noise ratio (SNR), defined as $\sqrt{\text{var}(\mathbf{f})/\sigma^2}$, is 2, 4 or 8. For each signal–SNR combination 100 simulations are carried out and the root mean square error (RMSE) of each reconstruction $\tilde{\mathbf{f}}$ is measured, where the RMSE is defined as

$$\text{RMSE}(\tilde{\mathbf{f}}) := \sqrt{\frac{1}{n}\sum_{k=0}^{n-1}(f_i - \tilde{f}_i)^2}.$$

For the CL multiwavelet both approximation and repeated row preprocessing are used (abbreviated with 'a' and 'r') with the same preprocessing matrices as in [15]. It is not recommendable to use repeated row preprocessing for the Φ_n since it introduces artificial correlations, therefore approximation preprocessing is used. For repeated row preprocessed signals $J := 6$ iterations of the DMWT are computed, for approximation preprocessed signals and scalar wavelet transforms $J := 5$ is chosen. In Table 2 the

(multi-)		Blocks			Bumps			HeaviSine			Doppler		
wavelet		2	4	8	2	4	8	2	4	8	2	4	8
Φ_1	i	8.3	6.1	5.5	6.4	6.2	5.2	13.2	11.6	11.0	9.9	8.8	6.5
Φ_1	ni	8.4	6.3	4.4	6.7	5.5	3.5	13.3	11.5	10.7	9.4	8.5	6.6
Φ_2	i	7.8	6.3	5.6	6.5	6.1	5.8	13.4	12.1	11.4	9.9	9.0	7.2
Φ_2	ni	7.6	5.9	6.2	6.7	6.4	6.3	13.4	11.9	10.8	9.6	9.0	7.8
Φ_3	i	7.7	6.5	5.5	6.8	6.3	5.8	13.5	11.8	11.3	10.2	8.7	7.3
Φ_3	ni	7.4	6.5	5.9	6.5	6.5	6.2	13.6	12.0	11.7	9.9	8.4	7.6
CLa		7.5	7.3	7.9	6.8	6.1	6.3	13.2	11.8	11.5	9.4	7.9	6.2
CLr		8.8	7.8	8.0	8.0	7.6	7.1	12.9	12.1	11.4	10.1	8.7	7.8
D_2		7.4	6.8	7.2	6.9	6.3	5.1	12.8	12.3	12.9	9.0	7.0	6.2
D_3		7.5	6.0	6.0	6.5	6.1	6.5	12.6	11.9	11.4	10.0	8.6	7.3

Table 2: Universal thresholding: gain (dB)

processing gains of the universal thresholding methods are shown. According to [15] the processing gain is defined as

$$\text{gain} = 20 \log_{10} \left(\frac{\sigma}{\text{average RMSE}} \right) \text{dB}.$$

Basically, apart from Φ_1 with a non–interpolating wavelet, the denoising results of the Φ_n are very similar to those of the scalar wavelets D_2 and D_3. For strongly perturbed signals (SNR= 2) the Φ_n perform slightly better, for softly perturbed signals (SNR= 8) they perform slightly worse than D_2 and D_3, especially for the 'Blocks' and 'HeaviSine' signals. Analogous to [15] the best denoising results for the 'Blocks' and 'Bumps' signals are obtained by the CL multiwavelet with repeated row preprocessing. This may be due to the symmetry of the CL multiwavelet/scaling vector and due to the repeated row preprocessing inherent redundancy. For the smoother signals 'HeaviSine' and 'Doppler' the results are similar to those of the Φ_n. The CL multiwavelet with approximation preprocessing yields above-average results for the 'Blocks' signal, below-average results for 'Doppler' and average results for the remaining signals. In Table 3 the denoising results of the vector thresholding methods are shown. The processing gains of the Φ_n have risen considerably for all signals but 'HeaviSine'. For the 'Blocks' and the 'Bumps' signal the denoising result are only slightly below those of the CLr multiwavelet obtained by universal thresholding and for the 'Doppler' signal the results of the Φ_n are significiantly better than those of all the other wavelets/multiwavelets. The denoising performance of the CL multiwavelet is also improved by using vector thresholding, especially for the 'Blocks', 'Bumps' and the softly perturbed 'HeaviSine' signal. The reader should note that the performance gap between approximation and repeated row preprocessing is narrowed by vector thresholding.

multi-		Blocks			Bumps			HeaviSine			Doppler		
wavelet		2	4	8	2	4	8	2	4	8	2	4	8
Φ_1	i	8.6	7.2	7.1	7.4	7.3	6.5	12.6	11.5	11.3	10.3	9.3	7.9
Φ_1	ni	8.5	6.8	4.8	7.5	6.3	3.9	12.7	11.6	10.5	10.4	9.5	7.0
Φ_2	i	8.1	7.4	7.0	7.8	7.1	7.3	13.0	11.8	11.4	11.0	9.8	8.6
Φ_2	ni	8.0	7.3	7.6	7.7	7.3	7.5	12.8	11.8	10.9	11.0	9.8	8.9
Φ_3	i	8.2	7.4	7.3	7.6	7.6	6.2	13.1	11.6	11.5	10.7	9.5	8.8
Φ_3	ni	8.1	7.5	7.1	7.6	7.6	7.2	13.1	11.6	11.6	10.7	9.4	9.1
CLa		9.0	8.6	9.7	8.1	8.1	7.2	12.8	12.0	12.6	9.9	8.5	7.1
CLr		8.7	9.3	10.2	8.4	8.0	7.3	13.1	12.3	13.5	10.0	8.8	7.6

Table 3: Vector thresholding: gain (dB)

4.2 Two–dimensional signals

For processing two–dimensional signals or images the direct product approach is used. We study three test images of size 256×256 pixel: the famous 'Lena' image (Fig. 1 a), the 'Camera(man)' image (Fig. 2 a) and a cartoon image (Fig. 3 a).

(multi-)		Lena				Camera				Cartoon			
wavelet		1	2	4	8	1	2	4	8	1	2	4	8
Φ_1	i	8.2	5.5	2.9	0.8	8.1	5.6	3.1	1.0	6.9	4.8	3.4	2.5
Φ_1	ni	8.2	5.5	2.7	0.2	8.1	5.5	2.9	0.5	6.9	4.7	3.1	1.5
Φ_2	i	8.4	5.6	2.9	0.8	8.1	5.7	3.1	1.0	6.9	4.8	3.4	2.5
Φ_2	ni	8.4	5.6	3.0	0.8	8.1	5.6	3.1	1.0	6.8	4.7	3.3	2.5
Φ_3	i	8.3	5.6	2.9	0.8	8.1	5.6	3.1	1.0	6.9	4.8	3.4	2.5
Φ_3	ni	8.3	5.6	3.0	0.9	8.1	5.6	3.0	1.0	6.9	4.8	3.3	2.5
CLa		8.1	5.4	2.8	0.8	8.0	5.3	3.0	1.0	6.6	4.6	3.2	2.7
CLr		8.0	6.3	4.4	2.7	7.8	6.0	4.5	3.0	7.2	6.0	5.3	4.8
D_2		8.1	5.2	2.7	0.7	8.0	5.4	3.0	1.1	6.6	4.4	3.4	3.0
D_3		8.2	5.4	2.8	0.8	8.0	5.5	3.0	1.0	6.7	4.7	3.3	2.5

Table 4: Universal thresholding: gain (dB)

For each image 25 simulations are carried out. In each simulation, a realization of Gaussian white noise corresponding to a signal to noise ratio of $1, 2, 4$ or 8 is added to the image. Then the noisy image is preprocessed and J steps of the DMWT are applied. It turns out that $J := 2$ for approximation and $J := 3$ for repeated row preprocessing yields the best results. Also for the scalar DWT $J := 2$ is chosen. Finally, after applying the universal or matrix thresholding procedure, the inverse DMWT (DWT) and postprocessing are applied.

In Table 4 the processing gains of our simulations for universal thresholding are shown. The gain is averaged over 25 realizations. The multiwavelets corresponding

multi-		Lena				Camera				Cartoon			
wavelet		1	2	4	8	1	2	4	8	1	2	4	8
Φ_1	i	8.6	6.0	3.7	1.9	8.4	6.0	3.8	2.1	7.3	5.5	4.4	3.8
Φ_1	ni	8.6	6.0	3.5	1.9	8.4	5.9	3.6	1.5	7.3	5.4	4.0	2.4
Φ_2	i	8.7	6.1	3.7	1.9	8.6	6.1	3.8	2.1	7.3	5.5	4.4	3.8
Φ_2	ni	8.8	6.1	3.7	1.9	8.5	6.1	3.8	2.1	7.3	5.5	4.4	3.8
Φ_3	i	8.7	6.1	3.7	2.0	8.5	6.0	3.8	2.1	7.3	5.6	4.4	3.7
Φ_3	ni	8.6	6.1	3.7	2.0	8.5	6.0	3.8	2.1	7.3	5.5	4.4	3.7
CLa		8.4	5.8	3.5	1.9	8.3	5.9	3.8	2.2	7.0	5.4	4.8	4.5
CLr		7.9	6.0	4.4	2.8	7.8	6.1	4.5	3.0	6.9	5.9	5.8	5.4

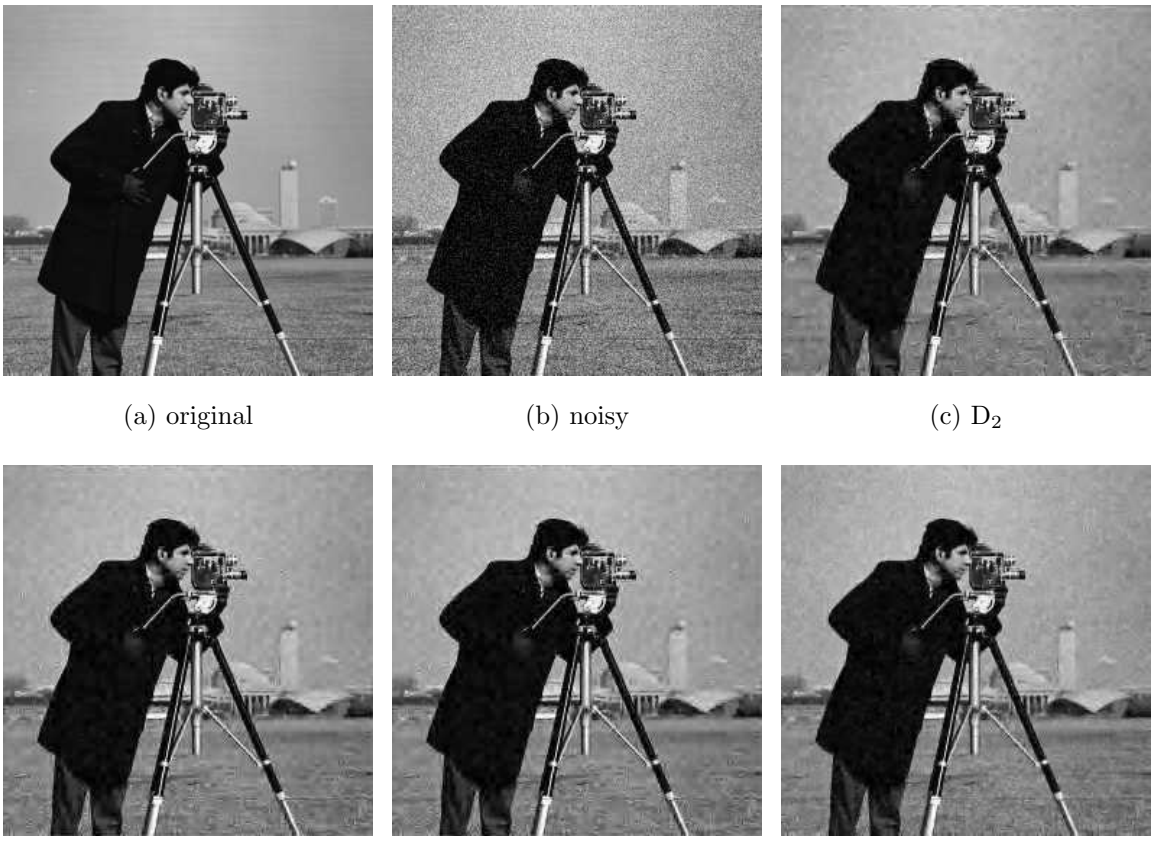
Table 5: Matrix thresholding: gain (dB)

(a) original (b) noisy (c) D_2

(d) Φ_2 'i' (e) Φ_2 'ni' (f) CLr

Figure 1: Lena, SNR= 2

to the scaling vectors Φ_n perform very similar to the scalar wavelets D_2 and D_3 and the Chui–Lian multiwavelet using approximation preprocessing, while they are outperformed by the CL multiwavelet using repeated row preprocessing. Similar to the scalar



(a) original (b) noisy (c) D_2

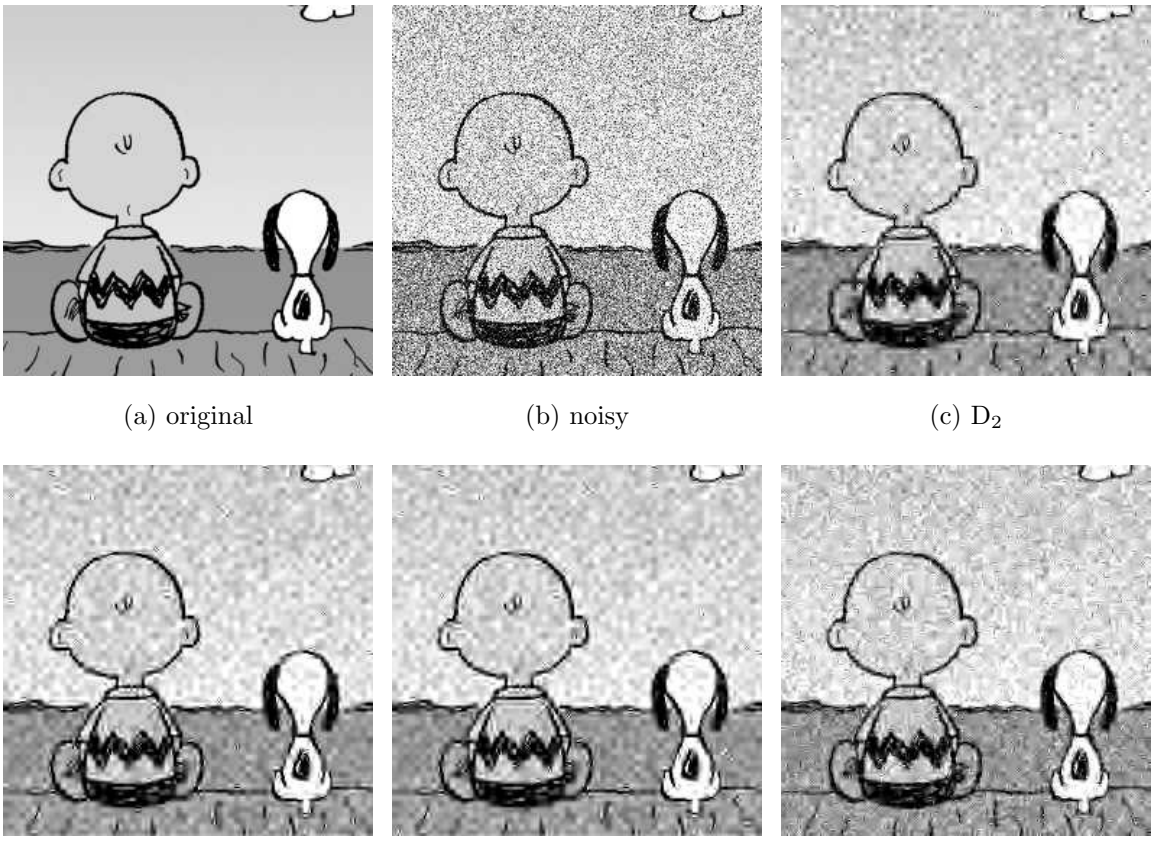
(d) Φ_2 'i' (e) Φ_2 'ni' (f) CLr

Figure 2: Camera, SNR= 4

case, there is no noticable difference between the denoising performances of interpolating and non–interpolating multiwavelets corresponding to the Φ_n.

In Table 5 the processing gains for matrix thresholding, averaged over 25 realizations, are shown. Again, matrix thresholding leads to significiantly improved results – at least for the multiwavelets using approximation preprocessing. The Chui–Lian multiwavelet using repeated row preprocessing gains an improvement only for the softly perturbed (SNR=4 or 8) cartoon image. For strongly perturbed images (SNR= 1) the processing gains of the Φ_n even exceed those of the CLr multiwavelet and for SNR= 2 they come close. In the case of softly perturbed images (SNR= 4 or 8), the Φ_n multiwavelets are still inferior to the CL multiwavelet, although their performance has risen considerably.

In Figure 1, 2 and 3 some typical denoising results for our test images are shown. In each figure (a) denotes the uncorrupted image, (b) the perturbed image and (c)–(f) show the reconstructions obtained by using (c) the scalar D_2 wavelet, the interpolating (d) and non–interpolating (e) multiwavelets corresponding to Φ_2, each using matrix thresholding, and (f) the Chui–Lian multiwavelet using repeated row preprocessing



(a) original (b) noisy (c) D_2

(d) Φ_2 'i' (e) Φ_2 'ni' (f) CLr

Figure 3: Cartoon, SNR= 1

and universal thresholding. In all three figures the results for Φ_2 'i' and 'ni' are very similar, which reflects the similarity already shown in the processing gains.

For the 'Lena' image depicted in Figure 1 a signal to noise ratio of 2 is chosen. The Φ_2 multiwavelets yield reconstructions comparable to that of the CLr multiwavelet. The latter one shows very sharp edges but also some grainy artifacts in the smoother parts of the image, while the former ones give a better recovering of the smooth parts but show a little less sharp edges. The visually worst result is obtained by using the D_2 wavelet which shows some coarse granular artifacts, especially at sharp edges.

In Figure 2 reconstructions of the softly perturbed 'Camera' image are shown. Again, the reconstruction (c) obtained by the D_2 wavelet looks rather blurred and contains strong artifacts at the edges. The visually best result seems to be image (f), in which most details of the original image are recovered, although the smooth background of the image looks somewhat grainy. Though revealing fine details and a smooth background, the reconstructions (d) and (e) show some shadow–like artifacts near the high–contrast edges, which disturbs the overall impression.

Finally, Figure 3 depicts the denoising results for the strongly perturbed comic image. The reconstructions obtained by applying the Φ_2 multiwavelets show a good compromise between recovering sharp edges and smooth parts of the image. Though there are also some shade artifacts, these multiwavelts seem to perform visually best for this image. Image (f) reveals very sharp edges but looks rather grainy and image (c) is very blurred.

5 Conclusion

In this paper we have studied the denoising performance of the interpolating scaling vectors and the corresponding multiwavelets constructed in [10] for one– and two–dimensional data. In general, using universal thresholding, these multiwavelets achieve results similar to those of the orthonormal Daubechies wavelets. By applying vector/matrix thresholding they can cope with the Chui–Lian multiwavelet, which has proven to be very useful in signal and image denoising.

The main advantage of the interpolating scaling vectors/multiwavelets is the simplicity of their application. Due to the interpolation property a straightforward and convenient preprocessing method is given, which preserves at least Gaussian white noise and does not alter the amount of data. Besides this, the vector/matrix thresholding routines are vastly simplified, because the covariance matrices of the multiwavelets coefficients do not have to be computed. These facts allow an implementation of a multiwavelet denoising algorithm which outperforms scalar wavelet algorithms in reconstruction quality while being comparably fast.

A further increase of the denoising performance could be achieved by applying more subtle algorithms like cycle spinning or the approach proposed by Chambolle et al. [2].

Acknowledgement. The author would like to thank Stephan Dahlke and Thorsten Raasch for their helpful comments and accurate review and Bastiaan Zapf for his support concerning the MATLAB routines.

References

[1] S. Bacchelli and S. Papi, *Matrix thresholding for multiwavelet image denoising*, Numerical Algorithms **33** (2003), 41–52.

[2] A. Chambolle, R. DeVore, N.-Y. Lee, and B. Lucier, *Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage*, IEEE Trans. Image Process. **7** (1998), 319–335.

[3] C. K. Chui and J. A. Lian, *A study of orthonormal multiwavelets*, Appl. Numer. Math. **20** (1996), 273–298.

[4] I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math. **41** (1988), 909–996.

[5] D. L. Donoho and I. M. Johnstone, *Ideal spatial adaptation via wavelet shrinkage*, Biometrika **81** (1994), 425–455.

[6] ———, *Adapting to unknown smoothness via wavelet shrinkage*, J. Amer. Statist. Assoc. **90** (1995), 1200–1224.

[7] T. R. Downie and B. W. Silverman, *The discrete multiple wavelet transform and thresholding methods*, Trans. on Signal Processing **46** (1998), no. 9, 2558–2561.

[8] T. N. T. Goodman, S. L. Lee, and W. S. Tang, *Wavelets in wandering subspaces*, Trans. of Amer. Math. Soc. **338** (1993), no. 2, 639–654.

[9] R. Q. Jia and Z. Shen, *Multiresolution and wavelets*, Proc. Edinb. Math. Soc. **37** (1994), 271–300.

[10] K. Koch, *Interpolating scaling vectors*, Bericht 2003–6, Philipps Universität Marburg, 2003.

[11] J. Lebrun and M. Vetterli, *Balanced multiwavelets theory and design*, Trans. on Signal Processing **46** (1998), no. 4, 1119–1125.

[12] S. Mallat, *Multiresolution approximation and wavelet orthonormal bases of* $L_2(\mathbb{R}^d)$, Trans. Am. Math. Soc. **315** (1989), 69–87.

[13] I. W. Selesnick, *Multiwavelet bases with extra approximation properties*, Trans. on Signal Processing **46** (1998), no. 11, 2898–2909.

[14] V. Strela, P. N. Heller, G. Strang, P. Topiwala, and C. Heil, *The application of multiwavelet filter banks to image processing*, IEEE Trans. Image Process. **8** (1999), 548–563.

[15] V. Strela and A. T. Walden, *Signal and image denoising via wavelet thresholding: Orthogonal and biorthogonal, scalar and multiple wavelet transforms*, Nonlinear and Nonstationary Signal Processing (W. F. Fitzgerald, R. L. Smith, A. T. Walden, and P. C. Young, eds.), Cambridge University Press, 2001.

[16] X.-G. Xia, *A new prefilter design for discrete multiwavelet transforms*, Trans. on Signal Processing **46** (1998), no. 6, 1558–1570.

[17] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. Q. Suter, *Design of prefilters for discrete multiwavelet transforms*, Trans. on Signal Processing **44** (1996), no. 4, 25–35.