Applied Mathematics and Computation 500 (2025) 129440

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Full Length Article

Sparse image representation through multiple multiresolution analysis

Mariantonia Cotronei [a],*, Dörte Rüweler [b], Tomas Sauer [b]

[a] DIIES, Università Mediterranea di Reggio Calabria, via Graziella loc. Feo di Vito, 89122 Reggio Calabria, Italy

[b] Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung & FORWISS, Universität Passau, 94032 Passau, Germany

ARTICLE INFO

Keywords:
Wavelets
Filterbanks
Multiresolution analysis
Image representation
Sparsification
Compression

ABSTRACT

We present a strategy for image data sparsification based on a multiple multiresolution representation obtained through a structured tree of filterbanks, where both the filters and decimation matrices may vary with the decomposition level. As an extension of standard wavelet and wavelet-like approaches, our method also captures directional anisotropic information of the image while maintaining a controlled implementation complexity due to its filterbank structure and to the possibility of expressing the employed 2-D filters in an almost separable aspect. The focus of this work is on the transformation stage of image compression, emphasizing the sparsification of the transformed data. The proposed algorithm exploits the redundancy of the transformed image by applying an efficient sparse selection strategy, retaining a minimal yet representative subset of coefficients while preserving most of the energy of the data.

1. Introduction

It is well known that wavelet techniques provide efficient representations of images, since they often are able to convey a large amount of information by a small number of coefficients. This inherent *sparsity* property makes such techniques the fundamental part of an image compression pipeline. More precisely, the approximate sparsity of a transformation, i.e., a large number of negligible coefficients it produces, is combined with thresholding and quantization to obtain the final compression rates. In this context, the fast decay of wavelet coefficients is the main reason why tensor product wavelets became a de facto standard in the creation of the JPEG2000 standard.

For an efficient discrete implementation, a bidimensional wavelet transform is performed through the application of a 2-D filterbank consisting of an *analysis* and a *synthesis* part, both described and executed in terms of convolutions and down-/upsampling operations. In most classical approaches, the filterbank (possibly changing at each level, as in [1]) is *separable* in the sense that it is given as tensor product of two 1-D filterbanks which are applied to the horizontal and vertical directions. While being conceptually simple and easy to implement, this approach fails in capturing directional features in the image data other than just those aligned with the coordinate axes.

Due to this observation, several approaches for the construction of nonseparable filterbanks have been proposed, for example based on a quincunx subsampling strategy [5,15]. The design of such filters is computationally involved, and only in a few cases it produces anisotropic basis functions that can be adapted to capture, for example, discontinuities along curves.

---

* Corresponding author.
E-mail addresses: mariantonia.cotronei@unirc.it (M. Cotronei), doerte.rueweler@uni-passau.de (D. Rüweler), tomas.sauer@uni-passau.de (T. Sauer).

https://doi.org/10.1016/j.amc.2025.129440
Received 18 May 2024; Received in revised form 31 January 2025; Accepted 24 March 2025

The design of bases that derive efficient sparse multiscale representations of anisotropic image features, in particular directional edges, has been investigated by many researchers, producing a quite impressive number of "-lets" as variations of the standard wavelet bases; just to mention a few: ridgelets and curvelets [4,2,3,12], bandlets [20,21], contourlets [9], directionlets [25], grouplets [22], wedgelets [10], shearlets and wavelets with composite dilations [14,16–18,11], etc.

In many situations, such constructions lead to very sparse and efficient *feature adapted* representations. Indeed, the quality measured in terms of fast decay of the N-term approximation error, i.e. the error between the original image and its approximation with N retained transform coefficient, becomes theoretically optimal for cartoon-like images with smooth boundaries. There are, however, drawbacks. The first is that usually an extra directional parameter has to be introduced that explicitly describes the direction of directional features and adds to the computational complexity of the transform. More important, the implementation of such strategies is typically based on the discretization of a procedure from the continuous domain with the well-known drawbacks in terms of efficiency: such approaches require oversampling and do not possess a simple and convenient (critically sampled) filterbank structure.

In this paper we present a strategy for directionally adaptive image sparsification based on a generalization of the discrete wavelet transform, namely a *multiple wavelet transform*. In this framework, the analysis and synthesis steps are not implemented in terms of a *single* perfect reconstruction filterbank, but as a composition of different (usually non-separable) filter operators and scaling matrices, as a variation of the approach presented in [17]. Such a combination allows for a directional adaptation of the process to the image data and a proper detection of their singularities along curves.

Unlike the standard multiresolution analysis (MRA) decomposition, the *multiple multiresolution analysis* decomposition produces a tree of representations of the input data, representing the alternating applications of two or more different filterbanks, where each branch of the tree corresponds to a complete decomposition of the image. This clearly increases the computational effort, but such a redundancy can be exploited for sparsification reasons, since there will be only few nodes in the associated MMRA tree containing the most relevant information, all the rest being neglectable. In other words, the amount of retained, relevant information from the redundant representation is the same as from a standard wavelet decomposition. Intuitively, these relevant nodes are associated to the *directional* features of the image. Based on this concept we propose an algorithm for strategically selecting the nodes containing most of the energy of the data, for a fixed compression ratio.

The experimental tests show that in comparison with standard wavelet techniques, a better quality in the reconstruction can be obtained from the same amount of coefficients. These are the only methods for an immediate comparison as they act directly on the image only by means of filterbanks. For the other directional methods mentioned above, the concrete way how the continuous approach is discretized significantly affects the performance of the method, making a fair comparison difficult, if not impossible.

Due to the increased *computational cost* of computing several multiresolution analyses, it is obvious that such kind of procedure cannot be used for "on the fly" or real-time data compression. Rather, this method is a good candidate, for instance, for persistent compressions, that is, in situations where large quantities of digital images must be stored, say, in permanent repositories.

The focus of this work is on the transformation stage of the compression and aims to *sparsify* the image data by producing a large number of negligible coefficients. In this respect, the proposed method provides a foundation for subsequent steps; quantization and coding, although being essential for completing the compression process, are outside the scope of this paper.

The organization of the paper is as follows. In Section 2, after fixing the notation, while recalling basic facts on filterbanks and classical MRA, we formally define the MMRA. In Section 3 we illustrate the proposed sparsification algorithm and give an interpretation of the strategy used. In Section 4 we show some of the simulation results obtained after an extensive experimentation, where we let the parameters involved in the procedure varying in different ways. Finally, in Section 5 we conclude and mention directions for future researches.

2. The multiple multiresolution analysis framework

2.1. Images and filterbanks

A digital image c is an element of $\ell_0(\mathbb{Z}^2)$, the set of all finitely supported real sequences defined on the integer lattice $\mathbb{Z}^2$. We use the notation $c = \left(c(\alpha) \,:\, \alpha \in \mathbb{Z}^2\right)$ for such sequences.

An integer sublattice is a subset of $\mathbb{Z}^2$ consisting of points obtained as linear combinations of two linearly independent integer vectors with integer coefficients, where it is common to combine them as columns into a generator matrix M. Recall that the choice of M is not unique for a particular lattice. It is a known fact, cf. [13,19], that, given M, the lattice $\mathbb{Z}^2$ is partitioned into the $d = |\det(M)|$ *cosets* modulo M, in the sense that

$$\mathbb{Z}^2 = \bigcup_{k=0}^{d-1} (\eta_k + M\mathbb{Z}^2),$$

where we have denoted by η_k, $k = 0, \dots, d-1$, the *coset representers*, chosen as the elements of $M[0,1)^2 \cap \mathbb{Z}^2$ in the so-called *fundamental parallelepiped*; this set is isomorphic to $\mathbb{Z}^2/M\mathbb{Z}^2$. Note that in the univariate case this boils down to the well-known $\mathbb{Z}_k = \mathbb{Z}/k\mathbb{Z} = \{0, \dots, k-1\}$.

In what follows, we make use of *expansive* generator matrices, i.e., matrices that have eigenvalues greater than 1 in modulus. In our context, such matrices M are also called *scaling* or *dilation* matrices. Indeed, this requirement on the eigenvalues is necessary in imaging applications since it implies that the sets $M^{-n}\mathbb{Z}^2$ converge to $\mathbb{R}^2$, so that in the limit the discrete model becomes a continuous one.

Sampling over suitable sub-lattices is the key operation in the discrete implementation of wavelet algorithms. In particular, if c is an image and M is an expansive sampling matrix, the *downsampling* ($\downarrow_M$) and *upsampling* ($\uparrow_M$) operators corresponding to M are respectively defined by:

$$\downarrow_M c = c(M\cdot)$$

and

$$\uparrow_M c(\alpha) = \begin{cases} c(M^{-1}\alpha), & \text{if } \alpha \in M\mathbb{Z}^2, \\ 0, & \text{otherwise.} \end{cases}$$

We recall that a finite bivariate *linear time invariant filter* is an operator $F : \ell(\mathbb{Z}^2) \to \ell(\mathbb{Z}^2)$ that acts as a convolution

$$Fc = f * c = \sum_{\alpha\in\mathbb{Z}^2} f(\cdot - \alpha)c(\alpha),$$

where the sequence $f \in \ell_0(\mathbb{Z}^2)$ is known as the *impulse response* of the filter. In what follows we will often identify the filter with its impulse response sequence.

Given a scaling matrix M with $d = |\det(M)|$, a d-channel critically sampled *filterbank* consists of d *analysis* and *synthesis* filters. If we denote such filters with F_j, G_j, respectively, $j = 0, \dots, d-1$, then we can describe the analysis part of the filterbank as the operator:

$$F : c \mapsto [c_j^1 = \downarrow_M (F_j * c) \; : \; j = 0, \dots, d-1]$$

and the synthesis part as

$$G : \left[c_j^1 \; : \; j = 0, \dots, d-1\right] \mapsto \sum_{j=0}^{d} G_j * (\uparrow_M c_j).$$

The analysis step splits a given image into d subband components, typically one low-pass (scaling) and $d-1$ high-pass (wavelet) ones while the *synthesis part* recombines those subbands into the same image as the original one provided that it is of perfect reconstruction type, that is $GF = I$.

The iteration of the above analysis process on the low-pass part allows to get the usual *multiresolution decomposition* of an image c.

As mentioned before, our aim is to construct a framework where anisotropic features of the image content can be captured. To this aim we concentrate our attention on *anisotropic* scaling matrices, whose eigenvalues are different in modulus. An example of anisotropic matrix is the parabolic scaling $\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$, which, in combination with shear matrices, playing the role of rotations, is the key ingredient of the discrete *shearlet transform*, cf. [17,18,16].

One of the advantages of using diagonal, either isotropic or anisotropic, scaling matrices in the context of filterbanks lies in the fact that the sampling operations just correspond to 1-D sampling along each coordinate axis and thus can be efficiently realized in a separable way. For implementing the sampling operations in the case of general scaling matrices, is it helpful to take advantage of the *Smith factorization* of the dilation matrix M, as $M = PDQ$ where P, Q are unimodular, i.e. $|\det P| = |\det Q| = 1$, and $D \geq 0$ is a diagonal matrix. Such a decomposition turns out to be useful also in the construction of 2D filterbank systems in a separable-like way. As shown in [6,7] the starting point of such a construction are *univariate* filters with dilation factors corresponding to the diagonal elements of D, which are then suitably combined and sampled using the matrices P, Q. The resulting system of multivariate filterbanks and wavelets then inherits the properties of the starting one (orthogonality, interpolatory, vanishing moments).

As a consequence of [6,7], we will assume that all our filters in the wavelet decomposition are *orthogonal* ones and that the data $c = \left(c(\alpha) : \alpha \in \mathbb{Z}^2\right)$ are the coefficients of the orthogonal projections

$$c(\alpha) = \int_{\mathbb{R}^2} f(x)\,\varphi(x-\alpha)\,dx, \qquad \alpha \in \mathbb{Z}^2, \tag{1}$$

with respect to an orthonormal scaling function φ, *refinable* with respect to M, i.e.,

$$\varphi = \sum_{\alpha\in\mathbb{Z}^2} a(\alpha)\,\varphi(M\cdot - \alpha), \tag{2}$$

$$\int_{\mathbb{R}^2} \varphi(x)\,\varphi(x-\alpha)\,dx = \delta_{\alpha,0}. \tag{3}$$

The sequence a in the refinement equation (2) is well known to be the low pass reconstruction filter of the associated filter bank, in fact, the wavelet analysis can be built completely from the low pass part of the filter bank, cf. [23]. Throughout this paper, we will assume that our filter bank is associated to an orthogonal wavelet system.

2.2. Multiple multiresolution analysis

The main motivation of our work is to make use of the redundant representation of image data provided by a *multiple multiresolution analysis* (MMRA) approach.

To recall the definition of MMRA from [23,7,8], let $c = (c(\alpha) : \alpha \in \mathbb{Z}^2)$ be a given (input) real 2D sequence.

In a standard MRA, based on the d-critically sampled filterbank pair (F, G), and on a single scaling matrix M, the sequence at the very first level is decomposed as

$$c_j^1 = \downarrow_M (F_j * c), \qquad j \in \mathbb{Z}_d,$$

and the process is then iterated up to a final level L according to the following scheme:

$$c_j^\ell = \downarrow_M (F_j * c_0^{\ell-1}), \qquad j \in \mathbb{Z}_d,\ \ell = 1, \dots, L,$$

where $c^0 = c$. Here, the implicit understanding is that F_0 is a *low-pass filter*, i.e., $F_0 * 1 = 1$, and then the coefficients $c_0^{(\ell)} = (c_0^{(\ell)}(\alpha) : \alpha \in \mathbb{Z}^2)$ are the scaling coefficients, while $c_j^{(\ell)}$, $j = 1, \dots, d-1$, are the (generalized) wavelet coefficients, at least as long as the filterbank has the perfect reconstruction property.

Suppose now that we are given a finite set of filterbanks, consisting of m analysis-synthesis operator pairs (F^k, G^k), $k \in \mathbb{Z}_m$, each associated with a scaling matrix M_k that may and will depend on k. Note that it is the central aspect of the concept to vary not only the filters but also the scaling matrices, but we still assume that all matrices have the same downsampling factor, i.e.,

$$|\det M_k| = d, \qquad k \in \mathbb{Z}_m. \tag{4}$$

Then the first level of a MMRA produces the $m \times d$ sequences:

$$c_j^{(k)} = \downarrow_{M_k} (F_j^k * c_0), \qquad k \in \mathbb{Z}_m,\ j \in \mathbb{Z}_d$$

Observe that the upper index now has a different meaning, indicated by writing it as a 1-tuple over $\mathbb{Z}_m$; this takes into account the fact that the number of produced sequences is $m \times d$ and not just d as before.

At the generic level $\ell + 1$, $1 \le \ell \le L - 1$, of the iterated scheme, we further decompose the *low pass part* of the preceding step as

$$c_j^{(\mu,k)} = \downarrow_{M_k} (F_j^k * c_0^{(\mu)}), \qquad \mu \in \mathbb{Z}_m^\ell,\ k \in \mathbb{Z}_m,\ j \in \mathbb{Z}_d.$$

The upper index $(\mu, k) \in \mathbb{Z}_m^{\ell+1}$ stands for the concatenation of the ℓ-tuple μ with $k \in \mathbb{Z}_m$.

The length of the upper index μ now represents the decomposition level. At the final level L the scheme computes $m^L \times d$ sequences with an obvious redundancy among them. The decomposition process can be better depicted as a tree with m^L branches, where the nodes on each branch μ are associated to the decomposition coefficients $c^\mu = \left(c_j^\nu \in \ell(\mathbb{Z}) : j \in \mathbb{Z}_d \right)$ and

$$c^\nu = \left(c_j^\nu \in \ell(\mathbb{Z}) : j \in \mathbb{Z}_d \setminus \{0\} \right), \qquad \nu \prec \mu,$$

where $\nu \prec \mu$ means that ν is an initial segment of μ, that is, $\mu = (\nu, \dots)$. Together, these coefficients form a wavelet-like decomposition along the branch μ and will be collected in the vector

$$c(\mu) = \left(\left(c_j^\nu : j \in \mathbb{Z}_d \setminus \{0\}, \nu \preceq \mu \right), c_0^\mu \right) \tag{5}$$

of the wavelet coefficients and the low-pass part c_0^μ. If the decomposition level and therefore the downsampling is large enough, c_0^μ will essentially consist of a single pixel and will become irrelevant, otherwise it has to be considered in the algorithm.

Each final branch can be represented by an integer number ranging from 0 to $m^L - 1$ and can be identified with the L-tuple μ, related to the path followed in the decomposition, by means of the m-adic expansion

$$\gamma = \gamma(\mu) = \sum_{k=0}^{L-1} \mu_k m^k, \qquad \mu \in \mathbb{Z}_m^L.$$

Intuitively, the similarity between two decompositions μ, μ' is related to $|\mu - \mu'|$. For a quantitative result see [24].

The MMRA decomposition tree is illustrated in Fig. 1 for the simple case of two filterbanks associated to two scaling matrices M_0, M_1 and 3 levels of decomposition. Each path is identified with a number γ ranging from 0 to 7. For example the path $\gamma = 5$ (identified with the triplet $(1, 0, 1)$) produces the decomposition coefficients

$$c(1,0,1) = ((c_j^{(1)}, c_j^{(1,0)}, c_j^{(1,0,1)} : j \in \mathbb{Z}_d \setminus \{0\}), c_0^{(1,0,1)}).$$

The reconstruction of the original data is obtained through the synthesis schemes implemented on each branch of the MMRA tree, which read as:

$$c_0^\mu = \sum_{j \in \mathbb{Z}_d} G_j^k * \uparrow_{M_k} c_j^{(\mu,k)}, \qquad k \in \mathbb{Z}_m,\ \mu \in \mathbb{Z}_m^\ell,$$

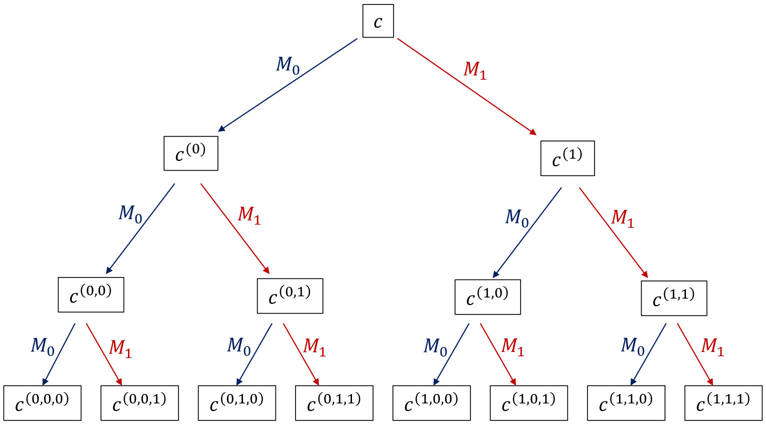


Fig. 1. MMRA decomposition tree up to 3 levels of decomposition.

for $\ell = L-1, \dots, 1$, so that c can be simultaneously recovered as:

$$c = \sum_{j=0}^{d-1} G_j^k * \uparrow_{M_k} c_j^{(k)}, \qquad k \in \mathbb{Z}_m$$

3. The algorithm

The algorithm introduced in this section makes use of the directional filterbank constructed in [7], associated to the unimodularly equivalent matrices

$$M_0 = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}, \qquad M_1 = \begin{pmatrix} 3 & -1 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} M_0 \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}. \tag{6}$$

In [7], these matrices have shown to provide *directional resolution*, that is, the matrix product

$$M_\gamma = M_{\gamma_\ell} \cdots M_{\gamma_1}, \qquad \gamma \in \mathbb{Z}_m^\ell, \tag{7}$$

related to a path of length ℓ has the property that $3^n M_\mu^{-1} e_2$ approximates any direction of the form $\binom{p}{1}$, $p \in \mathbb{R}$, to arbitrary precision for sufficiently large ℓ and $\mu = \mu(p)$ that depends on p, cf. [7, Theorem 6.3]. Note, however, that the dependence between p and $\mu(p)$ is not as straightforward as in the case of shearlets. Intuitively this observation indicates that, by a proper choice of μ, the *anisotropic* wavelet decomposition provides large values at directional vertices, which is illustrated in Fig. 2: different branches of the tree highlight vertices according to different directions.

This means that for an efficient sparsification of the image we need to consider decomposition along several branches of the tree that correspond to a variation of directional features.

The compression, or more precisely *sparsification*, algorithm that we are going to introduce exploits the redundancy of the MMRA decomposition tree by selecting the most relevant coefficients from *each* branch or at least several branches of the tree and combining them in a proper way. Although the computational effort required for multiple decompositions clearly exceeds that of a single wavelet transform, the directional adaptation is expected to provide a better reconstruction of the original image from the same number of nonzero coefficients.

Let $s \in (0,1)$ denote the *compression ratio*, i.e., the percentage of nonzero coefficients permitted, where the associated *sparsity* is defined as $1-s$. The number of coefficients to be retained in the branch is $s\,N$, where N is the number of coefficients in the branch which roughly equals the number of pixels in the image, depending on the precise implementation of the fast wavelet transform.

Let $\Gamma \subset \mathbb{Z}_m^L$ denote an arrangement of branches in a decomposition tree, numbered as $\Gamma = \left(\gamma^1, \dots, \gamma^n\right)$ where $\gamma^j \neq \gamma^k$, $1 \le j < k \le n$, which restricts n to $n \le m^L$. We call Γ the list of *active paths*.

Remark 1. The number n is a trade-off between directional resolution and computational efficiency. It is clear that in general $n = m^L$ is infeasible, so a good selection of the n branches may become relevant and could be done a priori or adaptively. In the section to follow, we collect some experimental information about this choice.

To the set Γ of active branches, we associate a vector $\omega \in \mathbb{R}_+^n$ of weights, i.e.,

$$\omega_j > 0, \qquad \sum_{j=1}^n \omega_j = 1;$$

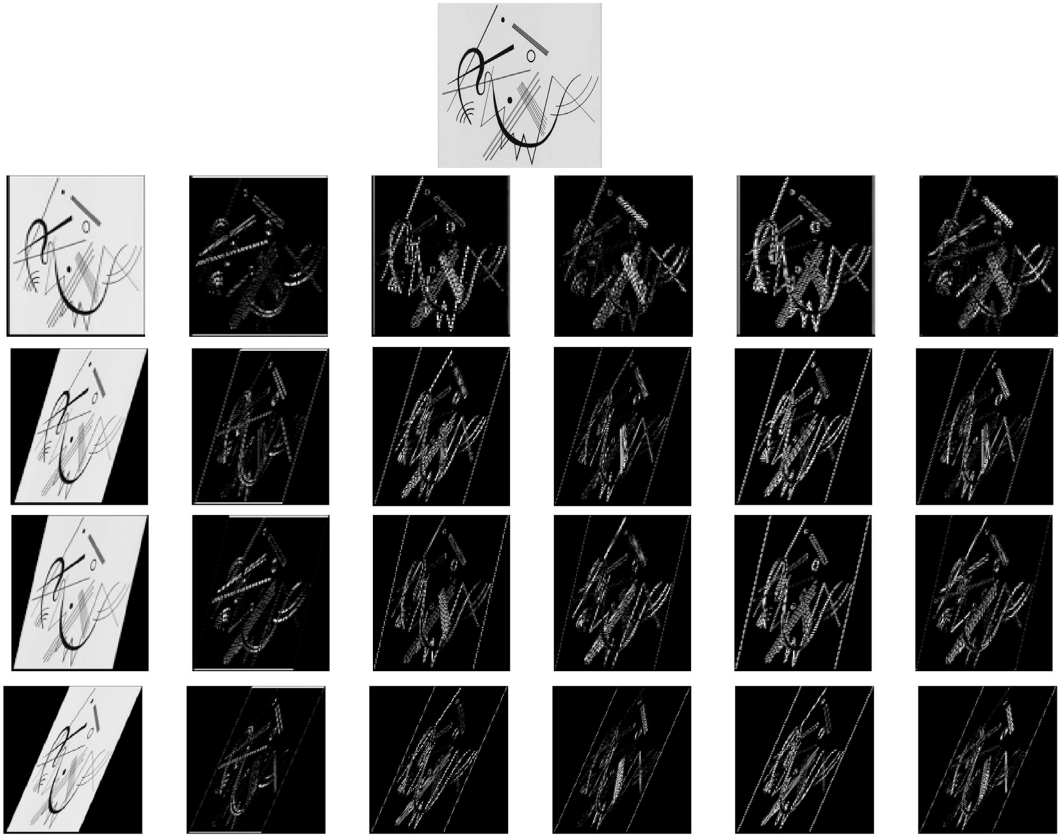


Fig. 2. Coefficients at level 2 obtained from the MMRA decomposition with filterbanks associated to the scaling matrices (6). From the top to the bottom: input image c_0, $c^{(0,0)}$, $c^{(0,1)}$, $c^{(1,0)}$, $c^{(1,1)}$.

Now we can describe the algorithm. Initializing $I_0 := I$ as the original image, we iterate on $k = 1, \dots, n$ and first compute, according to (5), a decomposition $c(\gamma^k)$ of I_{k-1} along the path γ^k. Among the *normalized* wavelet coefficients of $c(\gamma^k)$, we pick at most $\omega_k s N$ coefficients $\hat{c}(\gamma^k)$ that remain nonzero, all other ones are set to zero. The thresholded coefficients $\hat{c}(\gamma^k)$ reconstruct to an image $\hat{I}_{k-1}$, based on which we set $I_k = I_{k-1} - \hat{I}_{k-1}$.

Remark 2. The computation of the wavelet decomposition as well as the reconstruction along a single track by means of the filterbank-FWT is $O(N)$, where N is the number of pixel, hence the computation of the coefficients $\hat{c}(\gamma^k)$ has a computational effort of $O(nN)$. Moreover,

$$\left\| \left(\hat{c}(\gamma^1), \dots, \hat{c}(\gamma^n) \right) \right\|_0 = \sum_{k=1}^{n} \left\| \hat{c}(\gamma^k) \right\|_0 \le \sum_{k=1}^{n} s\omega_k N \le sN$$

by construction which yields an overall compression ratio of $\frac{s}{n}$ for the representation with nN coefficients.

Remark 3. Even if the decomposition uses redundant bases, it is not redundant since each additional branch is a decomposition of the difference between the original and all the preceding images.

Since we use an orthonormal wavelet basis, we have that $\| I_{k-1} \|_2 = \| \tilde{I}_{k-1} \|_2 + \| I_k \|_2$ or

$$\| I_k \|_2 = \| I_{k-1} \|_2 - \| \hat{I}_{k-1} \|_2 = \left\| c(\gamma^j) - \hat{c}(\gamma^j) \right\|_2 . \tag{8}$$

Note that the difference on the right hand side is precisely the 2-norm of the coefficients that are thresholded to zero. Iterating (8), we then get

$$\| I_k \|_2 = \| I \|_2 - \sum_{j=0}^{k-1} \| \hat{I}_j \|_2 = \| I \|_2 - \sum_{j=0}^{k-1} \| \hat{c}(\gamma^j) \|_2 \tag{9}$$

Remark 4. The difference to a classical wavelet decomposition consists in the fact that we have sN nonzero coefficients among the nN coefficients of the multiple wavelet decomposition and not among the N elements of a single wavelet decomposition. The empiric entropy of this decomposition is

$$E = E^* - \frac{nN - sN}{nN} \log \frac{nN - sN}{nN} = E^* - \left(1 - \frac{s}{n}\right) \log \left(1 - \frac{s}{n}\right)$$

where E^* denotes the empiric entropy of the nonzero coefficients, and therefore even corresponds to the entropy of a compression ratio of $\frac{s}{n}$ which is smaller than that with a compression ratio of s. Therefore, entropy encoders can be expected to reproduce this compression ratio and the result is comparable to that of compression the standard wavelet decomposition with a compression ratio of s.

The analytic appeal of orthogonal wavelets lies in the fact that the discrete coefficients can be interpreted as coefficients of an orthogonal decomposition, i.e.,

$$c(\gamma)(\alpha) = \int_{\mathbb{R}^2} f(t)\, \varphi_\gamma(t - \alpha)\, dt$$

for some scaling function on finest level that, however, depends on the full path γ in a multiple multiresolution, cf. [23]. Then the coefficients represent the projection of the function on the finest level L of the multiresolution of the branch γ

$$\sum_{\alpha \in \mathbb{Z}^2} c(\alpha)\, \varphi_\gamma(\cdot - \alpha).$$

If we now interpret $c(\gamma^{k-1})$ in this way, then the new coefficients $c(\gamma^{k-1}) - \hat{c}(\gamma^{k-1})$ still represent I_k with respect to γ^{k-1} while for a decomposition with respect to γ^k the basis function φ_{γ^k} would be needed based on the coefficients

$$c(\gamma^k)(\beta) = \sum_{\alpha \in \mathbb{Z}^2} \left(c(\gamma^{k-1}) - \hat{c}(\gamma^{k-1}) \right)(\alpha) \int_{\mathbb{R}^2} \varphi_{\gamma^{k-1}}(x - \alpha) \varphi_{\gamma^k}(x - \beta)\, dx$$

which corresponds to $c(\gamma^k) = \Phi_{\gamma^{k-1},\gamma^k} \left(c(\gamma^{k-1}) - \hat{c}(\gamma^{k-1}) \right)$ with the Gramian

$$\Phi_{\gamma^{k-1},\gamma^k} := \left[\int_{\mathbb{R}^2} \varphi_{\gamma^{k-1}}(x - \alpha) \varphi_{\gamma^k}(x - \beta)\, dx \,:\, \alpha, \beta \in \mathbb{Z}^s \right]$$

of the refinable functions. Due to the local support of the functions, this is a banded matrix. In our practical applications we suppress this multiplication since from a filterbank point of view the decomposition is still correct and efficient and we do not rely on analysis properties of the underlying functions. In analytic applications like the solution of PDEs by wavelet Galerkin methods, the use of the Gramian would be necessary, however.

For reference, we illustrate the algorithm in the following pseudocode

Algorithm 1 Sparsification procedure.

```
Require: $I$, $L$, $s$, $\omega$, $\Gamma$
$I_0 \leftarrow I$
$\tilde{I}_0 \leftarrow 0$
for $k = 1$ to $n = \#\Gamma$ do
    compute the decomposition $c(\gamma^k)$ of $I_{k-1}$
    compute the threshold value $\tau_k$ such that at most $\omega_k sN$ coefficients in $c(\gamma^k)$ are $\geq \tau_k$ in modulus
    threshold $c(\gamma^k)$ with respect to $\tau_k$ giving $\hat{c}(\gamma^k)$
    reconstruct $\tilde{I}_k$ from $\hat{c}(\gamma^k)$
    $I_k \leftarrow I_{k-1} - \tilde{I}_k$
end for
return $\hat{c}(\Gamma) = \left(\hat{c}(\gamma^1), \ldots, \hat{c}(\gamma^n)\right)$
```

For the sake of completeness, we present a flowchart of the entire compression pipeline (including quantization and coding) in Fig. 3.

For the experimental case studies to follow, note that there remains flexibility in the algorithm with respect to the strategy how to distribute the energy of the signal over the branches used in the decomposition. There are various ways to do that. First of all, the selection of n and the order of the branches can be chosen in a lot of ways, and, second, the distribution of energy, controlled by ω allows for flexibility. We will compare two approaches for the latter, namely,

1. uniformly distributed:

$$\omega = \left(\frac{1}{n}, \ldots, \frac{1}{n} \right) \tag{10}$$

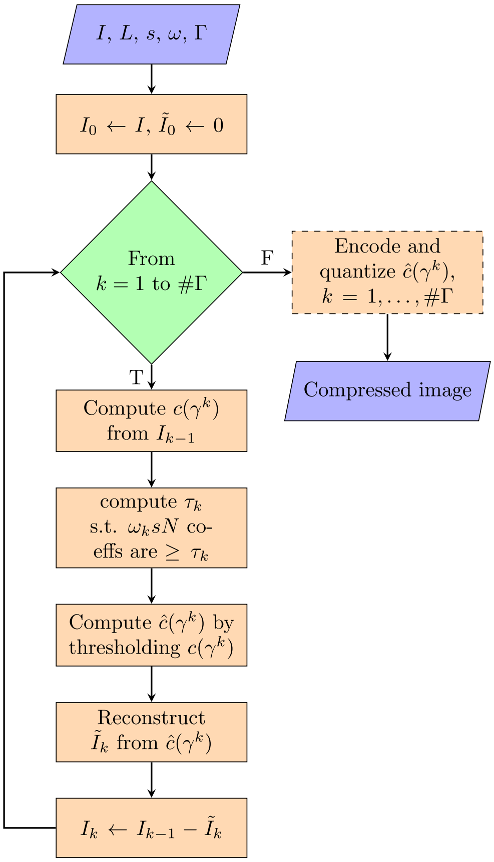


Fig. 3. Compression pipeline.

2. geometrically distributed:

$$\omega = \lambda \left(1, 1-\lambda, \dots, (1-\lambda)^{n-1}\right), \quad \lambda < 1. \tag{11}$$

The second strategy uses a higher “budget” of coefficients along the first paths and collects fewer coefficients in the remaining ones. Note that in general we cannot expect that the number of nonzero coefficients in $c(\gamma^k)$ equals exactly $\omega_k sN$ since the latter is even unlikely to be an integer. Effectively, we are thus working with the weights

$$\frac{\lfloor \omega_k sN \rfloor}{sN} \leq \omega_k, \qquad k = 1, \dots, n.$$

The reconstruction procedure can be described as follows. From $\hat{c}(\Gamma) = (\hat{c}(\gamma^1), \dots, \hat{c}(\gamma^n))$ we compute the reconstructions of each $\hat{c}(\gamma^k)$, $k = 1, \dots, n$, by means of the synthesis part of a filterbank with respect to the path γ^k. From the resulting vector $\left(G(\hat{c}(\gamma^1)), \dots, G(\hat{c}(\gamma^n))\right)$ of difference images we obtain the reconstructed image I as

$$I = \sum_{k=1}^{n} G(\hat{c}(\gamma^k)). \tag{12}$$

Remark 5. Note that, since the underlying wavelet filterbanks are orthogonal, the reconstruction procedure is *exact* if there no thresholding took place in the compression algorithm. Moreover, due to (8) and (9), the mean squared error and hence the PSNR of the sparsification depend directly on the 2-norm of the coefficients that have been thresholded to zero.

4. Experimental results

The strategy described in the previous section has been tested over a large grayscale image dataset. For our experiments, we use orthogonal filterbanks associated to the determinant-6 anisotropic matrices $M_0 = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$ and $M_1 = \begin{pmatrix} 3 & -1 \\ 0 & 2 \end{pmatrix}$, whose explicit expression is provided in [7]. The distinctive features of these filterbanks, besides their anisotropic properties, which enable direction detection

Algorithm 2 Reconstruction procedure.

```
Require: $\hat{c}(\Gamma) = \left(\hat{c}(\gamma^1), \dots, \hat{c}(\gamma^n)\right)$
$I \leftarrow 0$
for $k = 1$ to $n$ do
    compute the reconstruction $\hat{I}_k$ from $\hat{c}(\gamma^k)$
    $I \leftarrow I + \hat{I}_k$
end for
return $I$
```

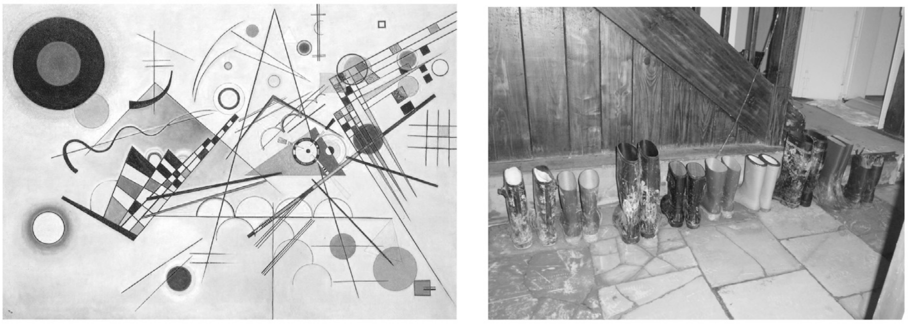


Fig. 4. Test images: Vassily Kandinsky - Composition 8, 1923 (left); “boots” image (right).

include a separable-like structure, facilitating an *efficient implementation* of the MMRA decomposition and reconstruction as well as the small determinant of the scaling matrices. Specifically, since the determinant of the scaling matrices is related to the number of decomposition subbands, filters associated with matrices of low determinant are preferable to ensure a *low computational cost*.

We present some results mostly obtained by applying our method to a famous Kandinsky painting (Fig. 4, left), which is well-suited to our context due to its numerous dominant directions, which are expected to be effectively captured by the anisotropic and directional detection capabilities of our approach. Additionally, we tested the well-known “Lena” image, still a commonly used as a benchmark in image compression. Lastly, for a specific test comparing our results with those obtained using Daubechies filters, we included a natural scene image with dominant directions, referred to as the “Boots” image (Fig. 4, right).

All tests have been performed using MATLAB. Specifically, the tests on the Lena image were conducted on a computer equipped with 64 GB of RAM and an Intel Core i7-14700 processor (28 cores). The tests on the Kandinsky and “Boots” images were carried out on a system with 32 GB of RAM and an Intel Core i7-4750 CPU @ 3.60 GHz (8 cores).

We begin by presenting results obtained on the 1090×555 image of the Kandinsky picture. Since $N = 604950$, the number of decomposition levels is fixed to $L = 8$, which results in $2^8 = 256$ different branches, and the total number of possible arrangements Γ is $255!$.

In all experiments, we choose $\mu^1 = (0, \dots, 0)$, so that the path indexed by $\gamma^1 = 0$ is the first one explored in the decomposition tree. This is due to the observation that for all other tested arrangements with $\mu^1 \neq (0, \dots, 0)$, hence $\gamma^1 \neq 0$, the quality of the reconstruction is always worse. The quality of the reconstruction is measured in terms of *PSNR (Peak Signal-to-Noise Ratio)* and *SSIM (Structural Similarity Index Measure)*.

4.1. Testing canonical ordering with varying weight vectors and compression factors

In our first experiment, we set $\Gamma = (0, 1, \dots, 255)$, referring to this configuration as *canonical ordering*, denoted by Γ_{can}. We fixed the compression ratio s, i.e., the ratio of coefficients to be retained, to 10%, 5%, and 1%, and used the two different options for the weight vector ω proposed in the previous section: the geometrical distribution (11), with several choices for the parameter λ, and the uniform distribution (10) with $n = 256$.

In Table 1 we can see the influence of the different values for the parameter λ and of the compression ratios on the number of active paths explored by the algorithm. The PSNR and SSIM values are also reported.

In Fig. 5, the results are shown from a graphical perspective. In particular, we depict the behavior of the PSNR in terms of the number of active paths. Fig. 6 presents a zoom-in version of Fig. 5 without the uniform choice. The plots in the figures confirm the intuition that the higher the value of λ, the higher is the impact of the first used branch on the result: more coefficients are retained from the first path. It is worthwhile to mention here that the exponential decay of the factors $(1 - \lambda)^n$ reduces the number of paths that are considered in the algorithm and therefore also the computing time as λ grows; however, for $\lambda \to 1$ also means that effectively we just have a single anisotropic wavelet decomposition without directional adaption.

4.2. Shuffling the ordering: testing two weight vectors with a 10% compression ratio

In the second set of experiments we illustrate how the reconstruction results are affected by the branch arrangement Γ. Over a total of $255!$ possibilities, we have used a naive way of “shuffling”, only fixing a portion of Γ. More specifically, we have considered

Table 1

Kandinsky image. Number of paths (denoted by n), PSNR values, and SSIM values obtained for different values of the compression ratio and for different λs. Note that the first row is related to $\omega = (1/256, \ldots, 1/256)$, while the rest of the rows refer to geometrically distributed ones. The ordering of the path exploration corresponds to the canonical one Γ_{can}.

	1%			5%			10%		
λ	n	PSNR	SSIM	n	PSNR	SSIM	n	PSNR	SSIM
1/256	256	20.339	0.159	256	25.894	0.305	256	29.966	0.411
0.166	40	20.335	0.154	49	26.466	0.326	53	31.138	0.434
0.20	34	20.332	0.153	41	26.465	0.325	44	31.095	0.433
0.25	27	20.328	0.154	33	26.465	0.325	35	31.088	0.432
0.33	20	20.282	0.154	24	26.374	0.322	26	31.091	0.431
0.50	13	20.153	0.150	15	26.320	0.321	16	31.094	0.431
0.60	10	20.143	0.150	12	26.306	0.321	12	31.094	0.431
0.80	6	20.134	0.149	7	26.265	0.320	7	31.092	0.431
0.90	4	20.131	0.149	5	26.263	0.320	5	31.090	0.431
0.95	3	20.130	0.149	4	26.263	0.320	4	31.089	0.431
1.00	1	20.129	0.148	1	26.262	0.320	1	31.084	0.431

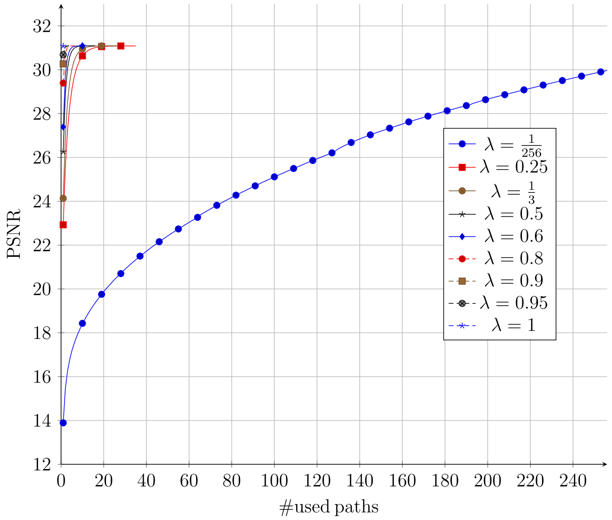


Fig. 5. Kandinsky image. PSNR as a function of the number of active paths for different values of λ where again $\lambda = 1/256$ denotes the uniform choice. The compression ratio is fixed to $s = 10\%$ and ordering corresponds to the canonical one.

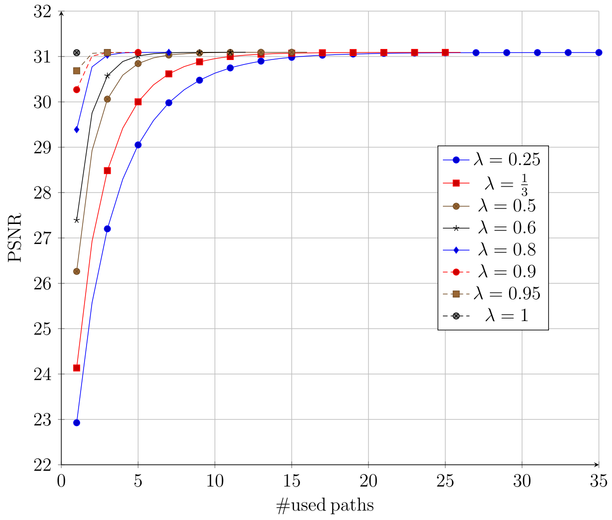
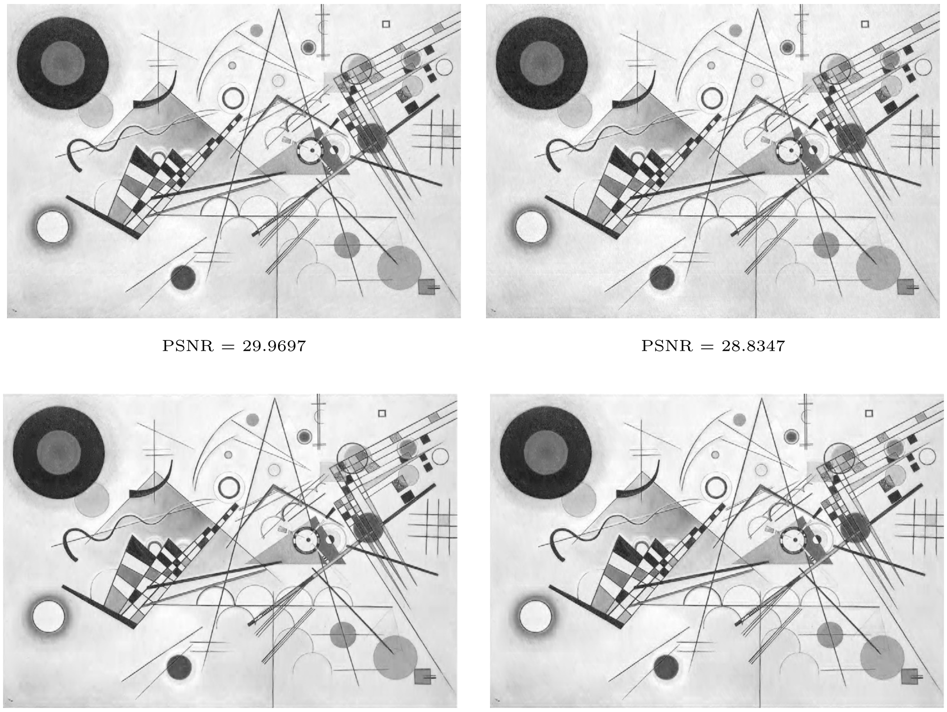


Fig. 6. Kandinsky image. As in Fig. 5, but without the uniform choice.

Table 2
Kandinsky image. Comparison between the uniform and the exponential choice in terms of PSNR values (minimum, maximum, mean and variance) obtained after 4048 test with the uniform and 6050 test with the exponential choice for ω.

ω	min	max	mean	variance
$(\frac{1}{256}, \dots, \frac{1}{256})$	28.8347	29.9697	29.2281	0.0198
$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$	30.1090	31.2427	30.9298	0.0815



PSNR = 29.9697　PSNR = 28.8347

PSNR = 31.2427　PSNR = 30.1090

Fig. 7. The best (left column) and worst (right column) reconstructions in terms of PSNR for $\omega = (\frac{1}{256}, \dots, \frac{1}{256})$ (first row) and $\omega = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$ (second row). $s = 10\%$.

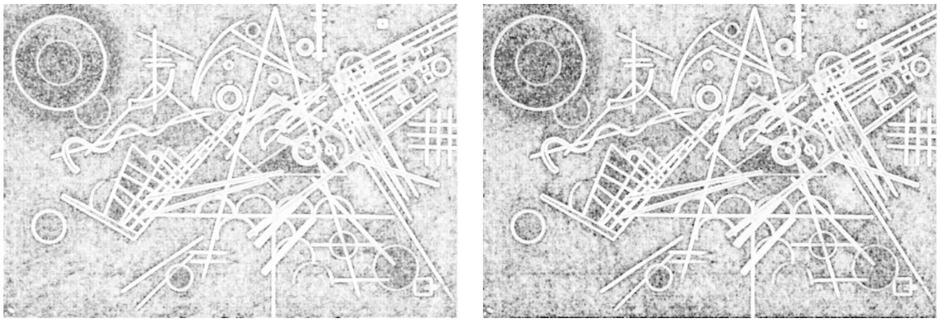
orderings of the form $\Gamma = (0, \Gamma_{\text{var}}, \gamma^{17}, \dots, \gamma^{256})$, varying $\Gamma_{\text{var}} = (\gamma^2, \dots, \gamma^{16})$, $\gamma^k \in \{1, \dots, 255\}$, $1 \le k \le 16$, and let the remaining part be $(\gamma^{17}, \dots, \gamma^{256}) = \Gamma_{\text{can}} \setminus (0, \Gamma_{\text{var}})$. This lead to a total of 4048 different arrangements of branches with the uniform choice and 6050 tests with the exponential choice for ω and the parameter $\lambda = 1/2$. The compression ratio is fixed to $s = 10\%$ in these experiments.

In Table 2 we report the results in terms of the maximum and the minimum values of PSNR, together with the average value and the variance. It turns out that the choice of a non-uniform λ achieves better results in average, with less computation time since only 16 branches are needed instead of 256. However, the variance among the results is higher which indicated sensitivity to the choice of important directions that are processed first. In the uniform distribution case with $n = 256$ we needed 10-15 minutes and in the geometric distribution case with $\lambda = 1/2$ the program ran 30 seconds to 1 minute. The actual reconstructions with respect to the best and worst PSNR are shown in Fig. 7. The same reconstructed images are provided as SSIM maps in Fig. 8.

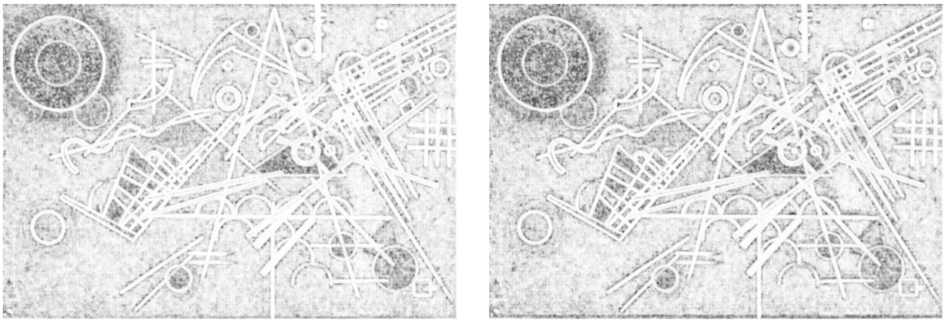
4.3. A shuffling strategy for path ordering

We next consider a “shuffling” strategy for ordering the paths, by alternating between the first and last 128 branches of the decomposition tree, that is, by selecting $\Gamma = (0, 128, 1, 129, 2, 130, 3, 131, 4, \dots, 127, 255)$, with which we reach a PSNR value of 30.6626, which is even larger than the maximum value for each arrangement of branches used in the previous experiment with the uniform choice for ω. We mention here that the same holds true for the Lena image to be considered later, achieving a PSNR value of 37.2123. Thus, this choice of Γ is preferable when using the whole decomposition tree.

Using the shuffling strategy for Γ, we also tested different choices for ω and different values for the compression ratio, since the latter also affects the number of active paths in a geometrical distribution case of ω. In Fig. 9 one can see what happens for each value of the compression ratio. At small compression ratios, the shuffling strategy with equally distributed values in ω over all possible branches gives the best option.



PSNR $= 29.9697$, SSIM $= 0.8335$ PSNR $= 28.8347$, SSIM $= 0.7765$



PSNR $= 31.2427$, SSIM $= 0.8562$ PSNR $= 30.1090$, SSIM $= 0.8350$

Fig. 8. SSIM maps for Fig. 7. White means that reconstruction and the original are identical, whereas black means they are totally different.

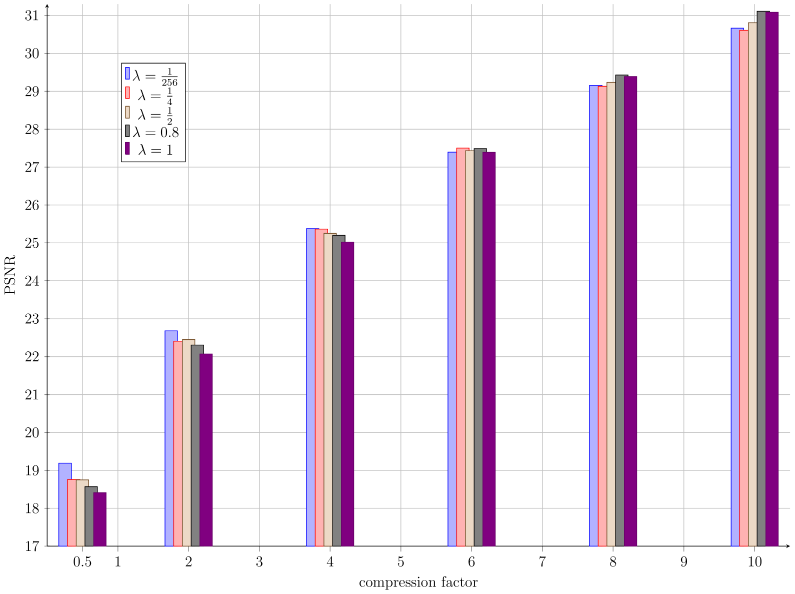


Fig. 9. Kandinsky image. PSNRs for different values of the compression ratio and λ, when the ordering $\Gamma = (0, 128, 1, 129, 2, 130, 3, 131, 4, \ldots, 127, 255)$ is adopted.

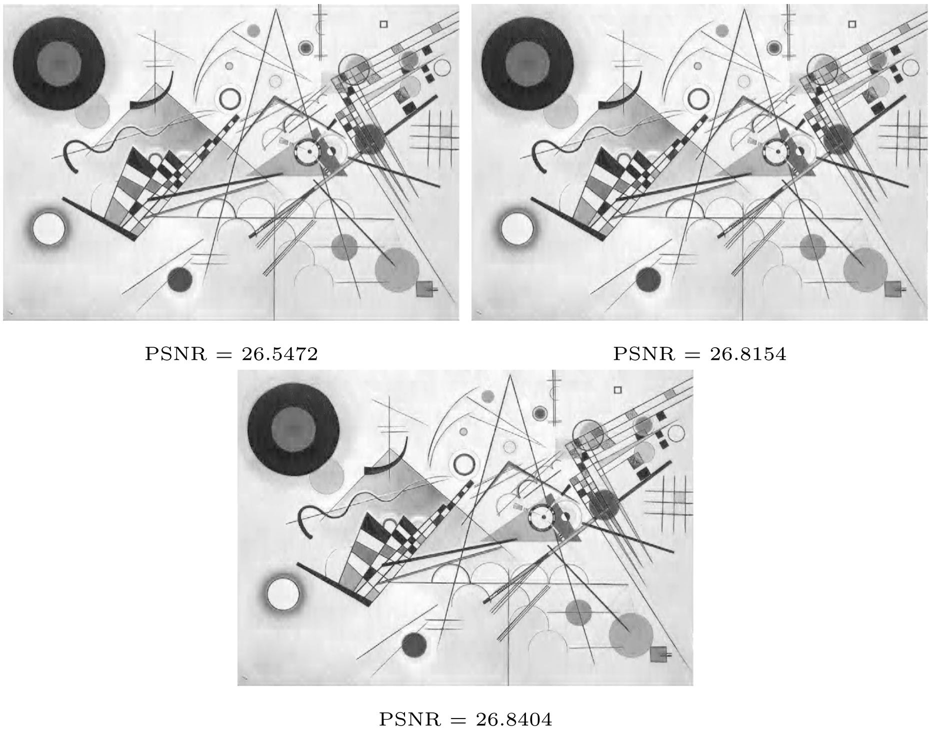
4.4. *Different weights with a fixed number of paths*

In the next experiment, we fix the compression ratio to $s = 5\%$ and choose $\omega = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots)$. Since this results in using 15 branches, we make a comparison with the equally distributed vector $\omega = (\frac{1}{15}, \frac{1}{15}, \frac{1}{15}, \ldots)$. Fixing $\gamma^1 = 0$, we have $\binom{255}{14}$ possibilities, but we consider only 3306 different arrangements.

The statistical results for the PSNR values are reported in Table 3. The first observation is that the uniform choice for ω, where each branch has the same weight, allows to get the higher maximal PSNR. Nevertheless, the mean value is lower then in the exponential

Table 3
Kandinsky image. Comparison between the exponential choice for $\lambda = 1/2$ and the uniform choice for $n = 15$ and $n = 16$ in terms of PSNRvalues (minimum, maximum, mean and variance) obtained after the 3306 tests performed as described in 4.4. The compression ratio here is $s = 5\%$.

ω	min	max	mean	variance
$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$	25.8787	26.5472	26.3184	0.018
$(\frac{1}{15}, \frac{1}{15}, \frac{1}{15}, \dots)$	25.1783	26.8154	26.0388	0.0573
$(\frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \dots)$	25.1882	26.8404	26.0552	0.0547


PSNR = 26.5472 PSNR = 26.8154

PSNR = 26.8404

Fig. 10. Clockwise from top left: $\Gamma_{\lambda_{\text{expo}}} = (1, 8, 148, 13, 109, 121, 7, 18, 237, 138, 95, 94, 40, 91, 87)$ $\Gamma_{\omega_{\text{equidis}}} = (1, 2, 3, 4, 252, 6, 7, 8, 228, 10, 227, 20, 13, 14, 190)$ $\Gamma_{\tilde{\omega}_{\text{equidis}}} = (1, 2, 3, 4, 252, 6, 7, 8, 228, 10, 227, 20, 13, 14, 190, 16)$, for $s = 5\%$, which are the best arrangement of branches in this case.

case, where we also have a lower variance, since only the first few branches have a significant impact than the remaining ones in the arrangement.

Comparing the choices $\omega = (\frac{1}{15}, \frac{1}{15}, \frac{1}{15}, \dots)$ and $\omega = (\frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \dots)$, on the other hand, we see that in the second case, by using one more branch, we improve the mean PSNR and reduce the variance. Hence, as expected, the quality of results increases with the number of paths.

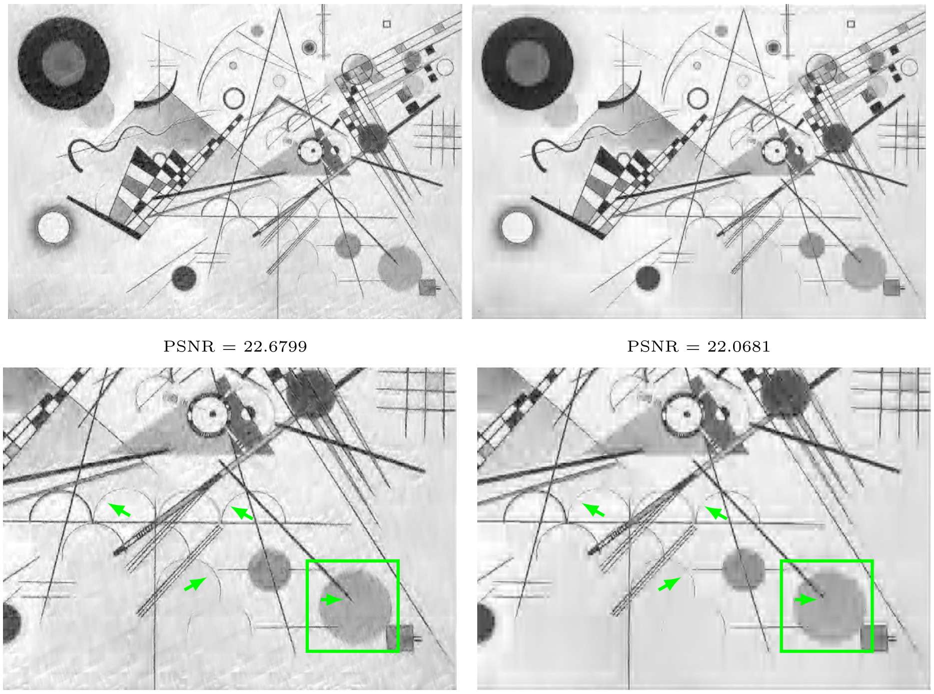
The arrangements of the paths allowing to reach the maximum PSNR for each choice of ω can be explicitly given, namely as:

$$\Gamma_{(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)} = (1, 8, 148, 13, 109, 121, 7, 18, 237, 138, 95, 94, 40, 91, 87)$$
$$\Gamma_{(\frac{1}{15}, \frac{1}{15}, \frac{1}{15}, \dots)} = (1, 2, 3, 4, 252, 6, 7, 8, 228, 10, 227, 20, 13, 14, 190)$$
$$\Gamma_{(\frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \dots)} = (1, 2, 3, 4, 252, 6, 7, 8, 228, 10, 227, 20, 13, 14, 190, 16).$$

They give rise to the reconstructions illustrated in Fig. 10.

4.5. Compression ratio 2%

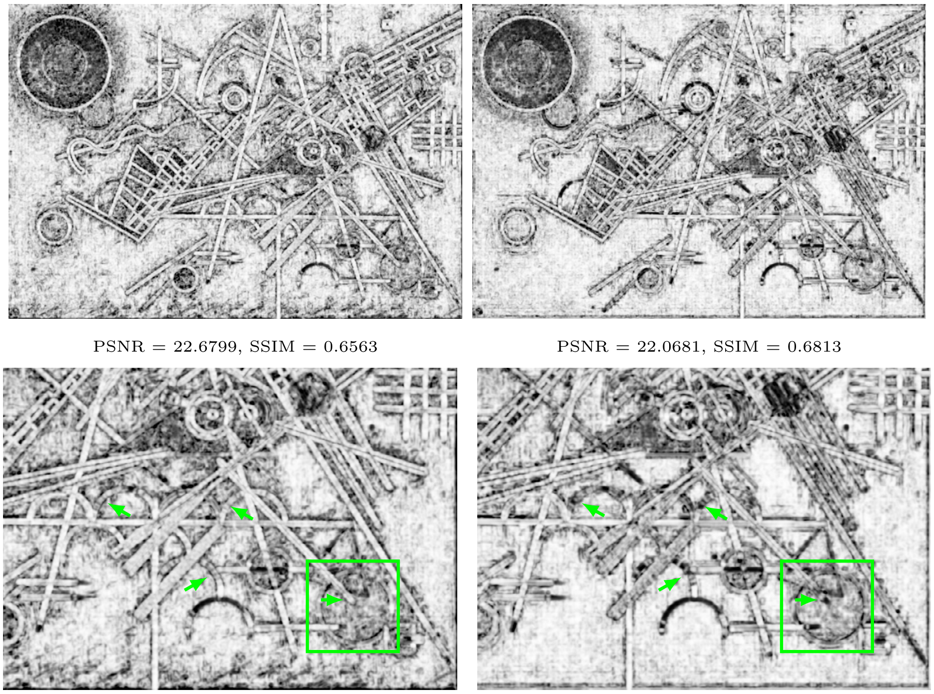
We continue with an example of reconstruction, after a more aggressive sparsification with a compression ratio of 2% and $\omega = (\frac{1}{256}, \frac{1}{256}, \dots, \frac{1}{256})$ compared to first branch only, i.e. $\omega = (1, 0, \dots, 0)$. To check the quality of the reconstructed images, we show both the entire images and a zoom into details in Fig. 11 and Fig. 12. From visual inspection, it is evident that the use of several paths gives better results in terms of reconstruction of directional features. This has consistently been observed also in several other



PSNR = 22.6799

PSNR = 22.0681

Fig. 11. Reconstructions obtained for $\omega = (\frac{1}{256}, \frac{1}{256}, \ldots, \frac{1}{256})$ (left) and $\omega = (1, 0, \ldots, 0)$ (right) for $s = 2\%$. In the second row we see a detail view. Note that in the left image the edges of the half circle are better reconstructed than on the right hand side (green arrows). The circle in the lower right corner (green box) on the left side is also rounder or smoother than the one on the right side. Also the tip of the line is better reconstructed (see arrow in the green box).



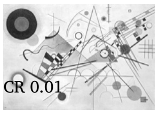
PSNR = 22.6799, SSIM = 0.6563

PSNR = 22.0681, SSIM = 0.6813

Fig. 12. Fig. 11 represented as SSIM images.

experiments, demonstrating that working on multiple branches rather than on a single one, as in a standard wavelet decomposition, is advantageous.

Table 4
Comparison between MMRA and a standard wavelet sparsification with Daubechies filters, on two different images (artificial and natural scenes).

			MMRA	Daub
	CR 0.05	PSNR	26.841	26.503
		SSIM	0.33641	0.32523
CR 0.01	PSNR	20.443	20.225	
		SSIM	0.17187	0.15638

			MMRA	Daub
	CR 0.05	PSNR	34.561	33.981
		SSIM	0.58934	0.56207
CR 0.01	PSNR	28.045	27.922	
		SSIM	0.29534	0.26637

Table 5
Lena image. Number of paths, PSNR values, SSIM values, and computation time (in seconds) obtained for different compression ratios and for different λs. Again, the first row is related to the choice of uniformly distributed weights $\omega = (1/256, \dots, 1/256)$, while the rest of the rows refer to geometrically distributed ones. The ordering of the path exploration corresponds to the canonical one Γ_{can}.

	1%				5%				10%			
λ	n	PSNR	SSIM	time	n	PSNR	SSIM	time	n	PSNR	SSIM	time
1/256	256	26.841	0.295	559	256	33.359	0.514	547	256	36.926	0.622	569
0.166	34	27.263	0.314	73	43	34.137	0.540	89	46	37.766	0.636	99
0.20	29	27.244	3.314	62	36	34.132	0.540	75	39	37.757	0.636	85
0.25	23	27.206	0.313	50	29	34.077	0.538	73	31	37.747	0.635	67
0.33	17	27.168	0.312	37	21	33.974	0.535	46	23	37.746	0.636	50
0.50	11	27.073	0.309	24	13	33.913	0.533	28	14	37.731	0.636	31
0.60	9	26.964	0.305	20	10	33.897	0.532	22	11	37.723	0.635	25
0.80	5	26.864	0.300	12	6	33.861	0.531	13	7	37.699	0.635	16
0.90	4	26.837	0.299	9	5	33.842	0.530	11	5	37.687	0.635	12
0.95	3	26.823	0.299	7	4	33.834	0.530	9	4	37.681	0.634	10
1.00	1	26.799	0.299	3	1	33.828	0.530	3	1	37.674	0.634	3

4.6. A comparison with Daubechies wavelets

We complete our analysis by comparing our approach to the standard wavelet compression method, employing a 2-D separable version of the Daubechies wavelets with 2 vanishing moments, at significantly low compression ratios. The number of vanishing moments was chosen to match those of the wavelet used in the MMRA. We evaluated this comparison across two distinct images: the Kandinsky artwork and the natural scene "boots" displaying various directional anisotropic patterns. We computed the quality of the reconstruction both in terms of PSNR and SSIM. We considered the best results obtained by our algorithm after 4000 random arrangements of the visited branches and varying the weight vector between an uniformly distributed and a geometrically distributed ones. The results, summarized in Table 4, demonstrate superior performance achieved by our approach.

4.7. Some additional tests on the Lena image

To conclude, we performed the same experiments as above to the famous "Lena" image that still is considered by many a benchmark in image processing. Since this image has fewer dominant directional features, the effects of the MMRA can be expected to be less significant. Some results, obtained by varying the CR and the λ parameter, are shown in Figs. 5, 14, 16. In Table 5 we list some results for the canonical ordering of the paths and variations of λ as described in section 4.1. Due to the characteristics of the image Lena, mainly its lack of predominant directions, we get, as expected, different results than with the other images, since it provides fewer edges and therefore structure where we can recognize the influence of the different number of used paths.

Also when we compare the uniform distributed and the geometrically distributed ω in terms of the PSNR values, see Table 6, we see that we reach an overall higher PSNR value than with the Kandinsky image. With the uniform distributed ω we have a higher variance than with the Kandinsky image (0.0198 (Kandinsky) versus 0.0325 (Lena)). But with the geometrically distributed ω we have a lower variance than with the Kandinsky image (0.0815 (Kandinsky) versus 0.0411 (Lena)). Since the SSIM values are similar, these variations are not visible to the eye. In accordance with the IEEE policy, we do not show the image itself, but illustrate the

Table 6

Lena image. Comparison between the uniform and the exponential choice for ω with $n = 256$ and $\lambda = \frac{1}{2}$ respectively, in terms of PSNR values (minimum, maximum, mean and variance) obtained after 4048 tests performed as described in the section 4.2. The compression ratio here is $s = 10\%$. The time is given in minutes.

ω	min	max	mean	variance	min time	max time	avg. time
$(\frac{1}{256}, \dots, \frac{1}{256})$	35.369	36.383	35.898	0.032579	8.95	11.51	9.51
$(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$	37.004	37.862	37.679	0.041136	0.49	1.02	0.53

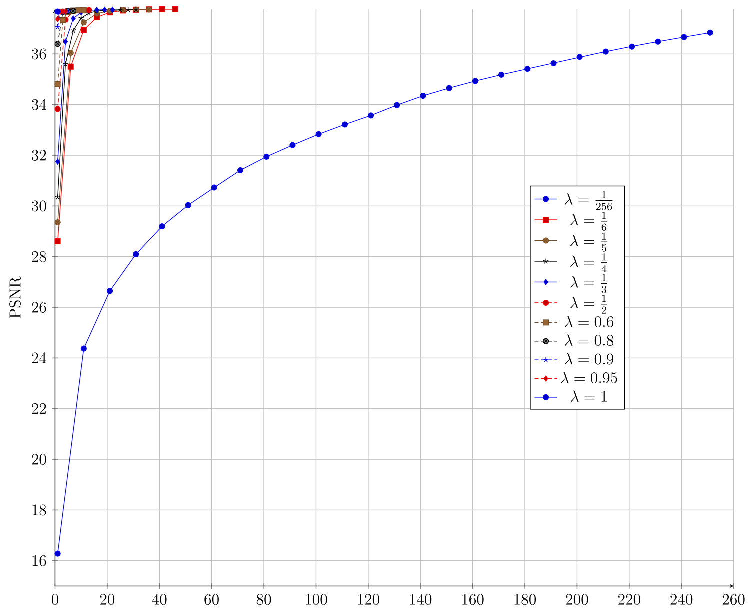


Fig. 13. Lena image. PSNR as a function of the number of active paths as in Fig. 5.

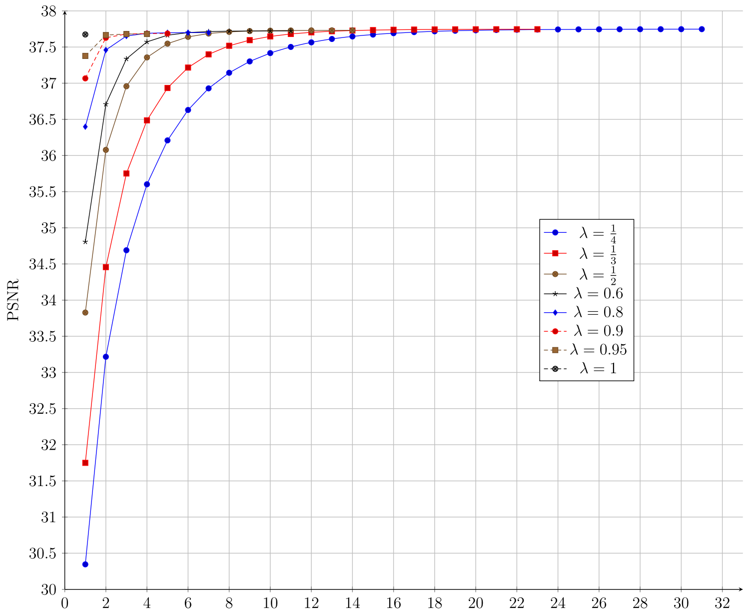
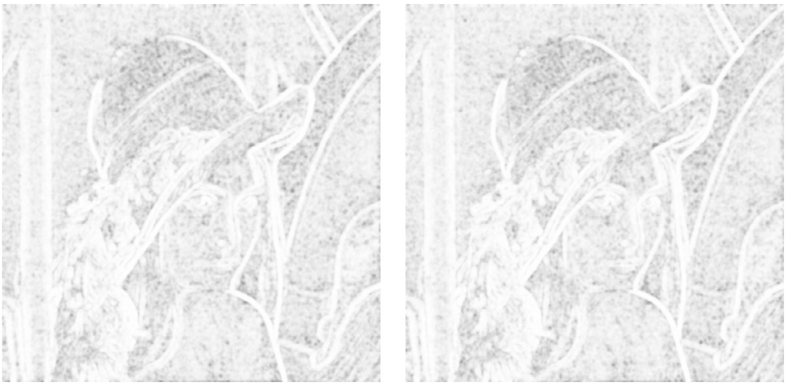
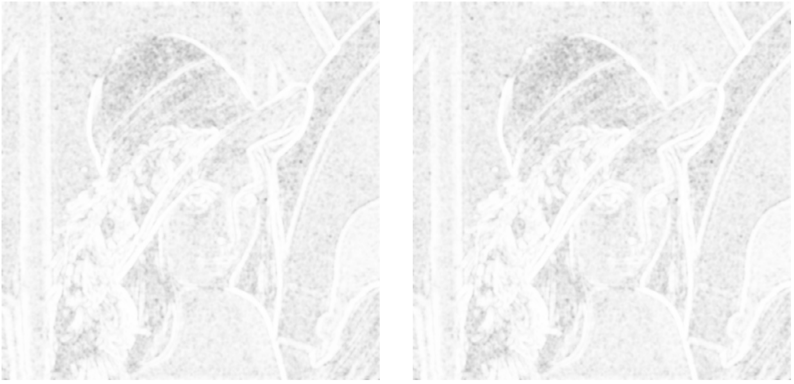


Fig. 14. Lena image. Same of Fig. 13, with a zoom-in effect (the result for the uniform choice $\lambda = 1/256$ is not plotted).



PSNR = 36.383, SSIM = 0.9066 PSNR = 35.369, SSIM = 0.9076



PSNR = 37.862, SSIM = 0.9319 PSNR = 37.004, SSIM = 0.9319

Fig. 15. SSIM maps as in Fig. 8 for the Lena image with $\omega = (\frac{1}{256}, \dots, \frac{1}{256})$ (first row) and $\omega = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$ (second row), and $s = 10\%$.

situation only through the structural similarity index map Fig. 15. We chose the reconstructions with respect to the best and worst PSNR, and they are very similar with respect to the SSIM.

5. Conclusion

We have realized and tested an algorithm suitable for image sparsification based on a multiple multiresolution representation of given image data that can be realized efficiently by means of filterbanks. Our method contributes to the transformation stage of image compression, emphasizing the sparsification of the transformed data. The algorithm leverages the redundancy inherent in the MMRA decomposition by employing an efficient sparse selection strategy, specifically retaining a minimal yet representative subset of coefficients that preserve most of the energy of the data. This approach allows for better image quality at the same compression rate or achieves a better compression rate at the same quality. This advantage goes beyond mere PSNR comparison and our method outperforms standard wavelet techniques especially in structural terms, see Table 4. Moreover, visual inspection shows that our approach particularly improves the preservations of edges as predicted by the theory.

The extra computational effort required by multiple multiresolution makes the algorithm not suitable for “on-the-fly” compression, but rather for efficiently storing data, especially when data has to be persisted. The question which branches of the tree should be included for the compressed data becomes relevant as trying all branches is neither tractable nor feasible. To that end, we offer heuristics on how to select the branches; exhaustive search shows that these heuristics do not give an optimal selection but that the results are close to the optimum in terms of quality. The selection procedure can be tuned to specific images or specific classes of images, but to that end more experiments as well as a deeper theoretical understanding of the function of the branches in an MMRA are still needed for an automatic assessment of the optimal parameters. We plan to address these issues in future research.

Acknowledgement

Mariantonia Cotronei is member of RITA (Research Italian network on Approximation), of UMI-TAA research group, and of Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica which partially supported this research with GNCS 2024 project CUP n.E53C23001670001. This work was also partially supported by the EU under the Italian National Recovery and Resilience Plan (PNRR), TECH4YOU project, ECS-00000009, CUP C33C22000290006.

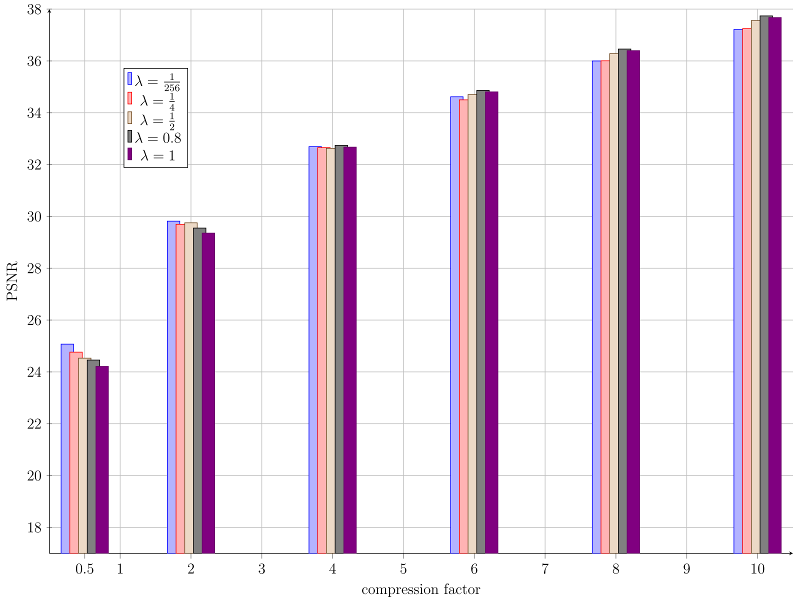


Fig. 16. Lena image. PSNRs for different values of the compression ratio and λ, when the ordering $\Gamma = (0, 128, 1, 129, 2, 130, 3, 131, 4, \ldots, 127, 255)$ is adopted.

We would like to thank the anonymous referees for their valuable comments and suggestions that have helped improve the quality of this paper.

Data availability

Data will be made available on request.

References

[1] V. Bruni, M. Cotronei, F. Pitolli, A family of level-dependent biorthogonal wavelet filters for image processing, J. Comput. Appl. Math. 367 (2020) 112467.
[2] E. Candès, D. Donoho, Curvelets: a surprisingly effective nonadaptive representation of objects with edges, in: L.S.A. Cohen, C. Rabut (Eds.), Curve and Surface Fitting: Saint-Malo 1999, Vanderbilt Univ. Press, 2003.
[3] E.J. Candès, D.L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, Commun. Pure Appl. Math. 57 (2003) 219–266.
[4] E.J. Candès, D.L. Donoho, Ridgelets: a key to higher-dimensional intermittency?, Philos. Trans. A Math. Phys. Eng. Sci. 357 (1760) (1999) 2495–2509.
[5] M. Charina, C. Conti, C. Cotronei, T. Sauer, Bivariate two-band wavelets demystified, Linear Algebra Appl. 608 (2021) 13–36.
[6] M. Cotronei, D. Ghisi, M. Rossini, T. Sauer, An anisotropic directional subdivision and multiresolution scheme, Adv. Comput. Math. 41 (2015) 709–726.
[7] M. Cotronei, M. Rossini, T. Sauer, E. Volontè, Filters for anisotropic wavelet decompositions, J. Comput. Appl. Math. 349 (2019) 316–330.
[8] M. Cotronei, D. Rüweler, T. Sauer, Multiple filterbanks for image processing: implementation issues, Math. Comput. Simul. 176 (2020) 147–159.
[9] M.N. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Trans. Image Process. 14 (12) (2005) 2091–2106.
[10] D.L. Donoho, Wedgelets: nearly-minimax estimation of edges, Ann. Stat. 27 (1999) 859–897.
[11] G.R. Easley, D. Labate, Critically sampled wavelets with composite dilations, IEEE Trans. Image Process. 21 (2) (2012) 550–561.
[12] J. Fadili, J.-L. Starck, Curvelets and ridgelets, in: R.A. Meyers (Ed.), Computational Complexity: Theory, Techniques, and Applications, Springer, New York, 2012, pp. 754–773.
[13] K. Gröchenig, W. Madych, Multiresolution analysis, Haar bases and self-similar tilings of $\mathbb{R}^n$, IEEE Trans. Inf. Theory 38 (2) (1992) 556–568.
[14] K. Guo, D. Labate, W.-Q. Lim, G. Weiss, E. Wilson, Wavelets with composite dilations and their MRA properties, Appl. Comput. Harmon. Anal. 20 (2) (2006) 202–236.
[15] J. Kovačević, M. Vetterli, Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbb{R}^n$, IEEE Trans. Inf. Theory 38 (3) (1992) 533–555.
[16] G. Kutyniok, D. Labate (Eds.), Shearlets: Multiscale Analysis for Multivariate Data, Birkhäuser, Boston, 2011.
[17] G. Kutyniok, T. Sauer, Adaptive directional subdivision schemes and shearlet multiresolution analysis, SIAM J. Math. Anal. 41 (4) (2009) 1436–1471.
[18] G. Kutyniok, M. Shahram, X. Zhuang, Shearlab: a rational design of a digital parabolic scaling algorithm, SIAM J. Imaging Sci. 5 (4) (2011) 1291–1332.
[19] V. Latour, J. Müller, W. Nickel, Stationary subdivison for general scaling matrices, Math. Z. 227 (4) (1998) 645–661.
[20] E. Le Pennec, S. Mallat, Bandelet image approximation and compression, Multiscale Model. Simul. 4 (3) (2005) 992–1039.
[21] S. Mallat, G. Peyre, A review of bandlet methods for geometrical image representation, Numer. Algorithms 44 (2007) 205–234.
[22] S. Mallat, Geometrical grouplets, Appl. Comput. Harmon. Anal. 26 (2) (2009) 161–180.
[23] T. Sauer, Shearlet multiresolution and multiple refinement, in: G. Kutyniok, D. Labate (Eds.), Shearlets: Multiscale Analysis for Multivariate Data, Birkhäuser, Boston, 2011.
[24] T. Sauer, Multiple subdivision, in: J.-D. Boissonnat, et al. (Eds.), Curves and Surfaces 2011, in: Lecture Notes in Computer Science, vol. 6920, Springer, 2011, pp. 612–628.
[25] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, P.L. Dragotti, Directionlets: anisotropic multidirectional representation with separable filtering, IEEE Trans. Image Process. 15 (7) (2006) 1916–1933.