

Sparse Image Representation by Directionlets

Vladan Velisavljević

Deutsche Telekom Laboratories, Berlin, Germany

Martin Vetterli

LCAV, Ecole Polytechnique Fédérale de Lausanne, Switzerland

Baltasar Beferull-Lozano

GSIC, Universidad de Valencia, Spain

Pier Luigi Dragotti

Imperial College, London, UK

June 15, 2009

Abstract

In spite of the success of the standard wavelet transform (WT) in image processing in recent years, the efficiency and sparsity of its representation is limited by the spatial symmetry and separability of its basis functions built in the horizontal and vertical directions. One-dimensional (1-D) discontinuities in images (edges or contours) that are very important elements in visual perception, intersect too many wavelet basis functions and lead to a non-sparse representation. To capture efficiently these elongated structures characterized by geometrical regularity along different directions (not only the horizontal and vertical), a more complex *multi-directional* (M-DIR) and *asymmetric* transform is required. We present a lattice-based *perfect reconstruction* and *critically sampled* asymmetric M-DIR WT. The transform retains the *separable* filtering and subsampling and the simplicity of computations and filter design from the standard two-dimensional (2-D) WT, unlike what is the case for some other directional transform constructions (e.g. curvelets, contourlets or edgelets). The corresponding asymmetric basis functions, called *directionlets*, have directional vanishing moments (DVM) along *any* two directions with rational slopes, which allows for a sparser representation of elongated and oriented features. As a consequence of the improved sparsity, directionlets provide an efficient tool for non-linear approximation (NLA) of images, significantly outperforming the standard 2-D WT. Furthermore, directionlets combined with wavelet-based image compression methods lead to a gain in the performance in terms of both the mean-square error and visual quality, especially at the low bit-rate compression, while having the same complexity. Finally, a shift-invariant non-subsampled version of directionlets is successfully implemented in image interpolation, where critical sampling is not a key requirement.

Key words: Directional transforms, Directional vanishing moments, Image coding, Image interpolation, Image orientation analysis, Multiresolution analysis, Separable transforms, Sparse representation, Wavelet transforms

1. Introduction

The task of providing efficient representations of images is a central problem in many image processing areas, such as denoising, compression and feature extraction. An efficient transform-based representation requires *sparsity*, that is, a large amount of information has to be compressed and expressed by a parsimonious set of transform coefficients.

Among many different transforms, the wavelet transform (WT) has become very successful in the last decade because it provides a good multi-resolution representation of one-dimensional (1-D) piecewise smooth signals, as explained by Donoho et al. (1998) and Vetterli and Kovačević (1995). The multi-resolution representation is achieved by iterative subband decompositions using two-band 1-D filter banks.

The application of wavelets to image processing requires the design of two-dimensional (2-D) wavelet bases. The most common approach is to construct such bases using 2-D separable filter banks, which consist of the direct product of the 1-D filter banks in the horizontal and vertical directions. Filtering with high-pass (HP) filters in the wavelet filter banks with enough vanishing moments (or zeros in the frequency domain at $\omega = 0$) along these two directions leads to a sparse representation of smooth signals. This method is conceptually simple and has very low complexity while all the 1-D wavelet theory carries over. These are the main reasons why it has been adopted in the image compression standard JPEG-2000 (Skodras et al., 2001).

However, despite this success, the standard separable 2-D WT fails to provide a sparse representation in the presence of 1-D discontinuities in images, like edges or contours. These discontinuities, being highly elongated objects, are characterized by a geometric coherence that is not properly captured by the standard 2-D WT. Namely, many wavelet basis functions intersect a discontinuity and this leads to many large magnitude coefficients, as illustrated in Figure 1(a).

The reason for the inefficiency of the standard 2-D WT resides in the *spatial symmetry* of its construction, that is, filtering and subsampling operations are applied equally along both the horizontal and vertical directions at each scale (see Figure 2(a)). As a result, the corresponding filters, obtained as direct products of 1-D filters, are symmetric at all scales, as shown in Figure 2(c),(d) for two different 1-D filter banks.

To improve the representation of elongated objects, the basis functions are required to be asymmetric and to “match” such objects, as schematically illustrated in Figure 1(b).¹ However, ensuring an efficient matching between these *asymmetric basis functions* and objects in images is a non-trivial task. Asymmetric basis functions have already been considered and exploited in adaptive and non-adaptive constructions. For instance, bandelets (LePennec and Mallat, 2000, 2003) provide an adaptive scheme for image transform, whereas edgelets, wedgelets (Donoho, 1999, Romberg et al., 2002, 2003, Wakin et al., 2006), curvelets (Candès and Donoho, 1999, 2002) and contourlets (Do and Vetterli, 2005) are examples of fixed, non-adaptive transforms. These methods build dictionaries of asymmetric basis functions that provide a sparse representation of edges in images. Furthermore, to achieve a good non-linear approximation (NLA) behavior, it has been shown by Candès and Donoho (1999) that the key feature is the parabolic scaling relation between the length and width of basis functions. However, the implementation of these transforms usually requires *oversampling* having *higher complexity* when compared to the standard WT, and require *non-separable processing* (convolution) and *non-separable filter design*. Furthermore, in some of these constructions (e.g. curvelets, Candès and Donoho, 1999) the design of the associated filters is performed in the *continuous domain* and this makes it difficult to use them directly on discrete images and achieve perfect reconstruction.

Another method that exploits directionality and resides on content-based adaptation of transform directions has been reported by Taubman and Zakhor (1994). The authors segment image and resample and transform the segments separately so that the dominant directions are aligned with the horizontal or vertical direction. Similarly, Wang et al. (2005) apply the WT along curves such that the energy in the HP subband is minimized.

Several recently proposed directionally adaptive approaches use the lifting scheme (Daubechies and Sweldens, 1998) in image compression algorithms. This scheme is exploited by Gerek and Çetin (2006), where transform directions are adapted pixel-wise throughout images. A similar adaptation is used by Ding et al. (2004) and Chang and Girod (2007), but with more (9 and 11, respectively) different directions. In addition, the method proposed by Ding et al. (2004) uses the pixel values at fractional coordinates obtained by interpolation. Lifting is also implemented by Wang et al. (2005) and Mehrseresht and

¹Notice that we used the terms *isotropic* and *anisotropic* basis functions in our previous work (Velisavljević et al., 2006, 2007), but we think the terms *symmetric* and *asymmetric* are more appropriate to use.

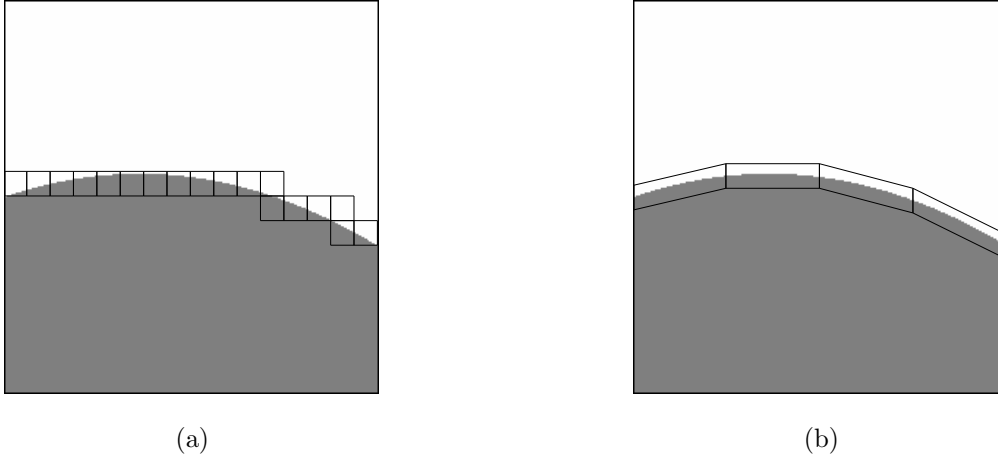


Figure 1: A simple image with one discontinuity along a smooth curve is represented by the two types of basis functions: symmetric and asymmetric. The support of these basis functions is shown schematically as black rectangles. (a) Symmetric basis functions generate a large number of significant coefficients around the discontinuity. (b) Asymmetric basis functions trace the discontinuity line and produce just a few significant coefficients.

Taubman (2006), where a wavelet packet decomposition is applied. However, even though these methods are computationally efficient and provide good compression results, they show a weaker performance when combined with zerotree-based compression algorithms.

Some other notable approaches use non-separable 2-D filter banks and subsampling (e.g. quincunx), as proposed by Kovačević (1991), Kovačević and Vetterli (1992) and Cohen and Daubechies (1993), but these methods are computationally complex and require a challenging design of the associated 2-D filter banks. Furthermore, several general multi-dimensional multi-band filter design methods have been proposed by Bamberger and Smith (1992), Chen and Vaidyanathan (1993), Tay and Kingsbury (1993) and Phoong et al. (1995) resulting in filters with separable polyphase components.

Notice that the standard WT uses only horizontal and vertical directions and the HP filters in this transform have vanishing moments only along these directions. Since characterization of features in synthetic and natural images involves many more than these two standard directions, *multi-directionality* and *directional vanishing moments* (DVM) play an important role in pursuing sparse representations.

Several other approaches also analyze geometrical structures in images, like polynomial modeling with quadtree segmentation (Shukla et al., 2005), footprints (Dragotti and Vetterli, 2001, 2003), multiscale transform (Cohen and Matei, 2001), etc. Apart from exploiting geometrical coherence, multi-directional (M-DIR) processing has also been applied to image denoising and classification. Examples of such transforms are the steerable pyramids (Simoncelli et al. 1992), the cortex transform (Watson, 1987), complex wavelets (Kingsbury, 2001), the directional wavelet analysis (Zuidwijk, 2000), directional filter banks (Bamberger and Smith, 1992, Phoong et al., 1995, Rosiles and Smith, 2003), brushlets (Meyer and Coifman, 1997), and the associative representation of visual information (Granlund and Knutsson, 1990). Some other methods involve directionally adaptive processing in order to preserve edges in images (Muresan and Parks, 2000, Li and Orchard, 2001, Hirakawa and Parks, 2005), whereas the method proposed by Cunha et al. (2006) imposes DVM in either critically sampled or oversampled filter banks. However, all of them fail to provide a *perfect reconstruction* and *critical* and *separable sampling* while keeping filter design completely in the *discrete domain* and with filters having DVM along *arbitrary directions*.

Our goal is to construct an asymmetric perfect reconstruction and critically sampled transform with HP filters having DVM along different directions, while retaining the simplicity, separability and filter design from the standard 2-D WT. The transform construction is based on partitioning of the discrete space using integer lattices, where the 1-D filtering is performed along lines across the lattice (as also explained by Velisavljević et al., 2006). The corresponding basis functions are called *directionlets*. We show that directionlets have good NLA and compression properties as compared to the performance achieved by some other overcomplete transform constructions (e.g. bandelets, wedgelets, curvelets or contourlets). At the same time, directionlets are superior to the standard separable 2-D WT having the same complexity. We also demonstrate a successful implementation of non-subsampled directionlets

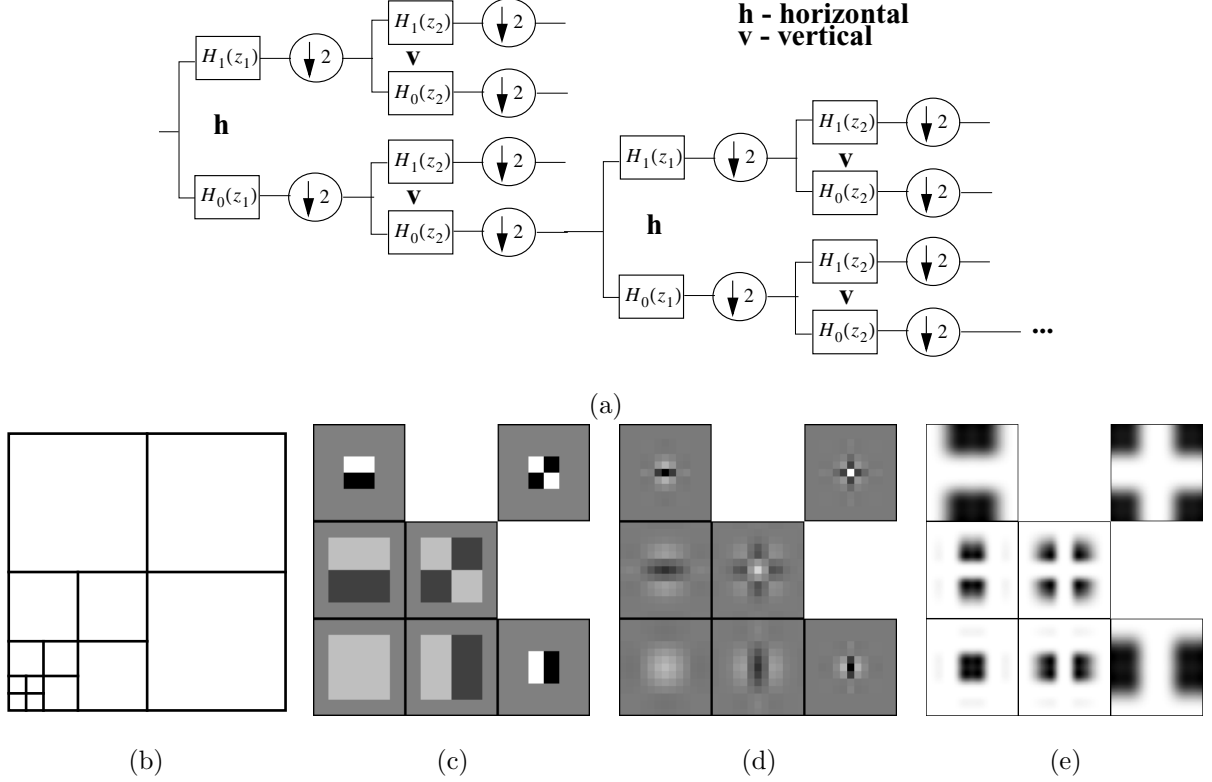


Figure 2: The standard 2-D WT is symmetric. (a) The filtering and subsampling operations are applied equally in both directions at each scale of the transform. (b) The corresponding decomposition in frequency. The basis functions obtained in this way are symmetric at each scale as shown in (c) for Haar and in (d) for biorthogonal "9-7" 1-D scaling and wavelet functions (given by Antonini et al., 1992). (e) The corresponding Fourier transforms of the basis functions obtained from the "9-7" 1-D filters.

in image interpolation, where shift-invariance of the overcomplete transform plays an important role in capturing directional features, while critical sampling is not required.

The outline of the paper is as follows. In Section 2, we present asymmetric constructions of the WT and explain the corresponding properties. Then, in Section 3, we show the inefficiency of the M-DIR transforms built on digital lines in order to motivate the need for an integer lattice-based construction. We also explain the construction of the skewed asymmetric lattice-based transforms. These transforms are applied in NLA and compression of images and the analysis of the results is exposed in Section 4 and 5, respectively. An overcomplete version of directionlets is implemented in image interpolation and the corresponding methods and performance are demonstrated in Section 6. Finally, we conclude in Section 7.

2. Asymmetric Wavelet Transforms

As explained in Section 1, the standard WT produces symmetric basis functions, which fail to provide a sparse representation of edges in images. However, a modified method that we explain next improves this representation and, at the same time, retains the 1-D filtering and subsampling operations. In the sequel of this section, we give two examples of asymmetric transform construction that still inherit the simplicity of processing and filter design from the standard WT. Furthermore, these two asymmetric transforms are critically sampled and lead to perfect reconstruction.

2.1. Fully Separable Decomposition

Inspired by the *geometrical period* of the Dutch painter Piet Mondrian (1872 - 1944), who established *neoplasticism* and *De Stijl* in Europe in the beginning of the 20th century (Milner, 2002), we define a

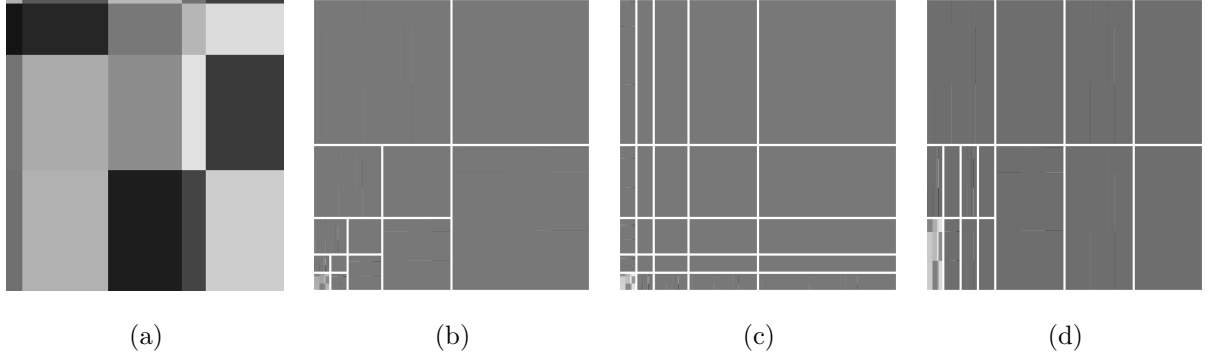


Figure 3: (a) An image from the class $\text{Mondrian}(k_1, k_2)$. The image is transformed by the three transforms: (b) standard WT, (c) FSWT, (d) AWT(2,1) with 1-D wavelet filters having enough vanishing moments.

simple class of piecewise polynomial images denoted as $\text{Mondrian}(k_1, k_2)$.

Definition 1. The class $\text{Mondrian}(k_1, k_2)$ contains piecewise polynomial images of the size $M \times M$ with k_1 horizontal and k_2 vertical discontinuities.

An example of the image from the class $\text{Mondrian}(k_1, k_2)$ is shown in Figure 3(a). This class is not efficiently represented by the standard WT. The discontinuities lead to too many nonzero coefficients, as shown in Figure 3(b) and expressed in the proposition below.

Proposition 1. Given an $M \times M$ image from the class $\text{Mondrian}(k_1, k_2)$, the number of nonzero transform coefficients in the band-pass subbands produced by the standard 2-D WT with the 1-D wavelets having enough vanishing moments is given by

$$N = O[(k_1 + k_2)M]. \quad (1)$$

Proof. Recall that a polynomial of the n th order is annihilated by a wavelet that has at least $n + 1$ vanishing moments. Thus, the three band-pass subbands at the j th level of the standard WT, where $1 \leq j \leq \log_2 M$, contain $N_1 = O[k_1 M / 2^j + k_2]$, $N_2 = O[k_1 + k_2 M / 2^j]$, and $N_3 = O[k_1 + k_2]$ nonzero coefficients, respectively. The total number of nonzero coefficients across scales is given by the sum $\sum_{j=1}^{\log_2 M} (N_1 + N_2 + N_3) = O[(k_1 + k_2)M]$. \square

To improve compactness of the representation of the class $\text{Mondrian}(k_1, k_2)$, we define the *fully separable WT* (FSWT). In this transform, a full 1-D WT is applied in the horizontal direction (each row of image) and then, on each output, a full 1-D WT is applied in the vertical direction (each column). The decomposition scheme is shown in Figure 4(a). Notice that such a decomposition has already been proposed by Westerink (1989), Nowak and Baraniuk (1999) and Rosiene and Nguyen (1999), where it is referred to as tensor wavelet basis.

The FSWT provides asymmetric basis functions that are better adapted to the elongated objects, such as the discontinuities in the class $\text{Mondrian}(k_1, k_2)$, as illustrated in Figure 4(c). The representation efficiency is strongly improved, as can be seen in Figure 3(c) from the resulting sparsity and the corresponding order is given in Proposition 2.

Proposition 2. Given an $M \times M$ image from the class $\text{Mondrian}(k_1, k_2)$, the number of nonzero transform coefficients in the band-pass subbands produced by the FSWT with the 1-D wavelets having enough vanishing moments is given by

$$O[(k_1 + k_2)(\log_2 M)^2]. \quad (2)$$

Proof. Each band-pass subband is indexed by (j_1, j_2) , where $1 \leq j_1, j_2 \leq \log_2 M$ determine the number of the horizontal and vertical transforms, respectively. The subband (j_1, j_2) contains $O[k_1 + k_2]$ nonzero transform coefficients and, therefore, the total number of nonzero coefficients is given by the sum $\sum_{j_1=1}^{\log_2 M} \sum_{j_2=1}^{\log_2 M} O[k_1 + k_2] = O[(k_1 + k_2)(\log_2 M)^2]$. \square

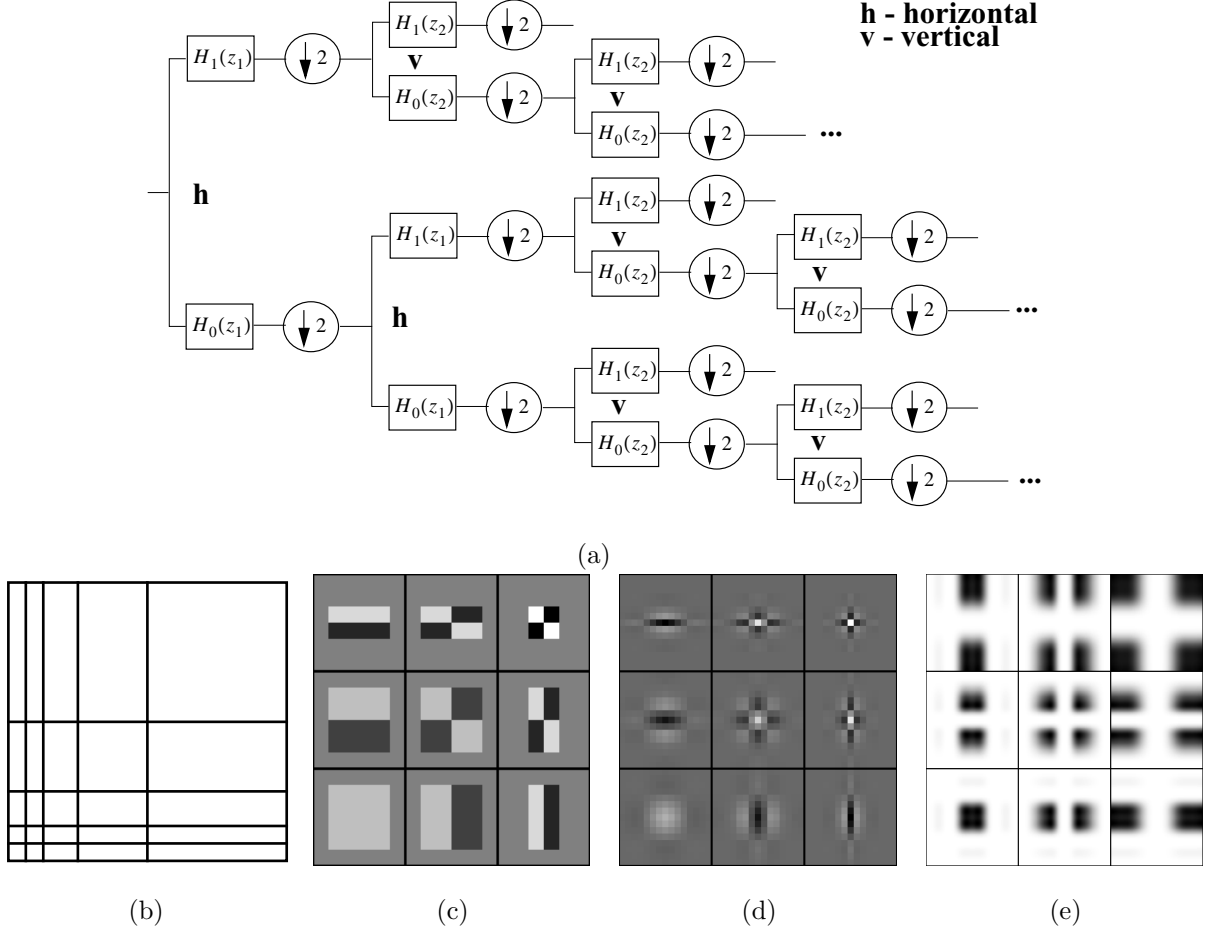


Figure 4: The FSWT is asymmetric, as the number of 1-D transforms is not equal in the two directions. (a) An example of the transform scheme. Only 2 steps in each direction are shown. (b) The decomposition in frequency that corresponds to the construction in (a) with 4 steps in each direction. The asymmetric basis functions obtained from the (c) Haar and (d) biorthogonal "9-7" 1-D scaling and wavelet functions. (e) The corresponding Fourier transform of the basis functions obtained from the "9-7" 1-D filters.

The performance of the FSWT on the class $\text{Mondrian}(k_1, k_2)$, given by Eq. (2), is substantially better than the result of the standard WT, given by Eq. (1); namely, there is an exponential improvement in terms of M . The improvement is a consequence of asymmetry of the basis functions that is matched to the elongation of the objects within the class. However, the FSWT performs well only when it is applied on Mondrian-like images, while natural images contain features that are not well represented by straight horizontal and vertical lines.

Notice that if a transformed image contains a curve (or any discontinuity that is not a straight line), then the FSWT fails, as the number of nonzero coefficients grows exponentially across scales. Intuitively, the failure happens because the FSWT enforces a higher asymmetry (or elongation of the basis functions) than the one that is required in order to provide a compact representation of objects in natural images. To overcome this problem, we introduce an asymmetric transform, which performs better on a larger class of images.

2.2. Asymmetric Wavelet Decomposition

In the *asymmetric WT* (AWT) the number of transforms applied along the horizontal and vertical directions is unequal, that is, there are n_1 horizontal and n_2 vertical transforms per scale, where n_1 is not necessarily equal to n_2 . Then, the iteration is continued in the low-pass (LP), like in the standard WT. We denote such an asymmetric transform as $\text{AWT}(n_1, n_2)$. The *asymmetry factor* $\rho = n_1/n_2$ determines elongation of the basis functions of the $\text{AWT}(n_1, n_2)$. An example of the construction and basis functions

Table 1: Orders of approximation by the standard WT, FSWT and AWT applied on the class **Mondrian**(k_1, k_2).

Standard WT	FSWT	AWT
$(k_1 + k_2)M$	$(k_1 + k_2)(\log_2 M)^2$	$(k_1 a + k_2/a)M$

is shown in Figure 5, where the AWT(2,1) is used.

Notice that both the standard WT and the FSWT can be expressed in terms of the AWT. The standard WT is simply given by AWT(1,1). However, the representation of the FSWT is more complex and it is given as a concatenation of two AWTs. The first transform is AWT($n_{1max}, 0$) that produces $n_{1max} + 1$ subbands and it is followed by the AWT($0, n_{2max}$) applied on each subband. The arguments n_{1max} and n_{2max} determine the maximal number of transforms in the two directions and they depend on the size of the image.

Even though the AWT is not the most appropriate representation for the particular case of Mondrian-like images, it improves approximation of more general classes of images, as shown in Section 4. Figure 3(d) shows the result of the AWT(2,1) of an image from the class **Mondrian**(k_1, k_2). The order of the number of nonzero coefficients is given by the following proposition.

Proposition 3. *Given an $M \times M$ image from the class **Mondrian**(k_1, k_2), the number of nonzero transform coefficients in band-pass subbands produced by the AWT(n_1, n_2) with 1-D wavelets having enough vanishing moments is given by*

$$O \left[\left(ak_1 + \frac{1}{a}k_2 \right) M \right], \text{ where } a = \frac{2^{n_2} - 1}{2^{n_1} - 1}. \quad (3)$$

Proof. The number of nonzero coefficients at the j th level of the AWT(n_1, n_2) is given by

$$\begin{aligned} n(j) = & O \left[k_1 (2^{n_2} - 1) \cdot \frac{M}{2^{n_1 j}} + k_1 (2^{n_1} - 1) \cdot 2^{n_2} \right. \\ & \left. + k_2 (2^{n_1} - 1) \cdot \frac{M}{2^{n_2 j}} + k_2 (2^{n_2} - 1) \cdot 2^{n_1} \right]. \end{aligned}$$

The total number of nonzero coefficients across scales is, therefore,

$$N = \sum_{j=1}^{\frac{\log_2 M}{\max(n_1, n_2)}} n(j) = O \left[\left(ak_1 + \frac{1}{a}k_2 \right) M \right].$$

□

Notice that the result in Proposition 3 is a generalization of the result in Proposition 1. Table 1 summarizes the orders of nonzero coefficients in band-pass subbands produced by the three transforms applied on the class **Mondrian**(k_1, k_2).

The transforms explained in this section are applied in the horizontal and vertical directions only. More general transforms can be obtained by imposing vanishing moments along different directions. These transforms provide an efficient representation of more general classes of images, involving more than only the two standard directions, as shown in the next section.

3. Directional Wavelet Transforms

Several transform constructions that lead to asymmetric basis functions have been presented in Section 2. However, all the constructions, including the standard WT, use only horizontal and vertical directions. Notice also that the HP filters in these transforms have vanishing moments only along these two directions. Here, we present a lattice-based transform, which exploits multi-directionality and retains the simplicity of computations and filter design from the standard WT.

In the sequel, we explain the problem of approximation of directions in the discrete space \mathbb{Z}^2 and we introduce the concept of directional interaction. Then, we propose a new lattice-based method that allows for a generalization of the transform constructions from Section 2 to include separable 1-D filtering and subsampling across multiple directions, not only horizontal and vertical. Finally, we give the polyphase analysis of the lattice-based transforms.

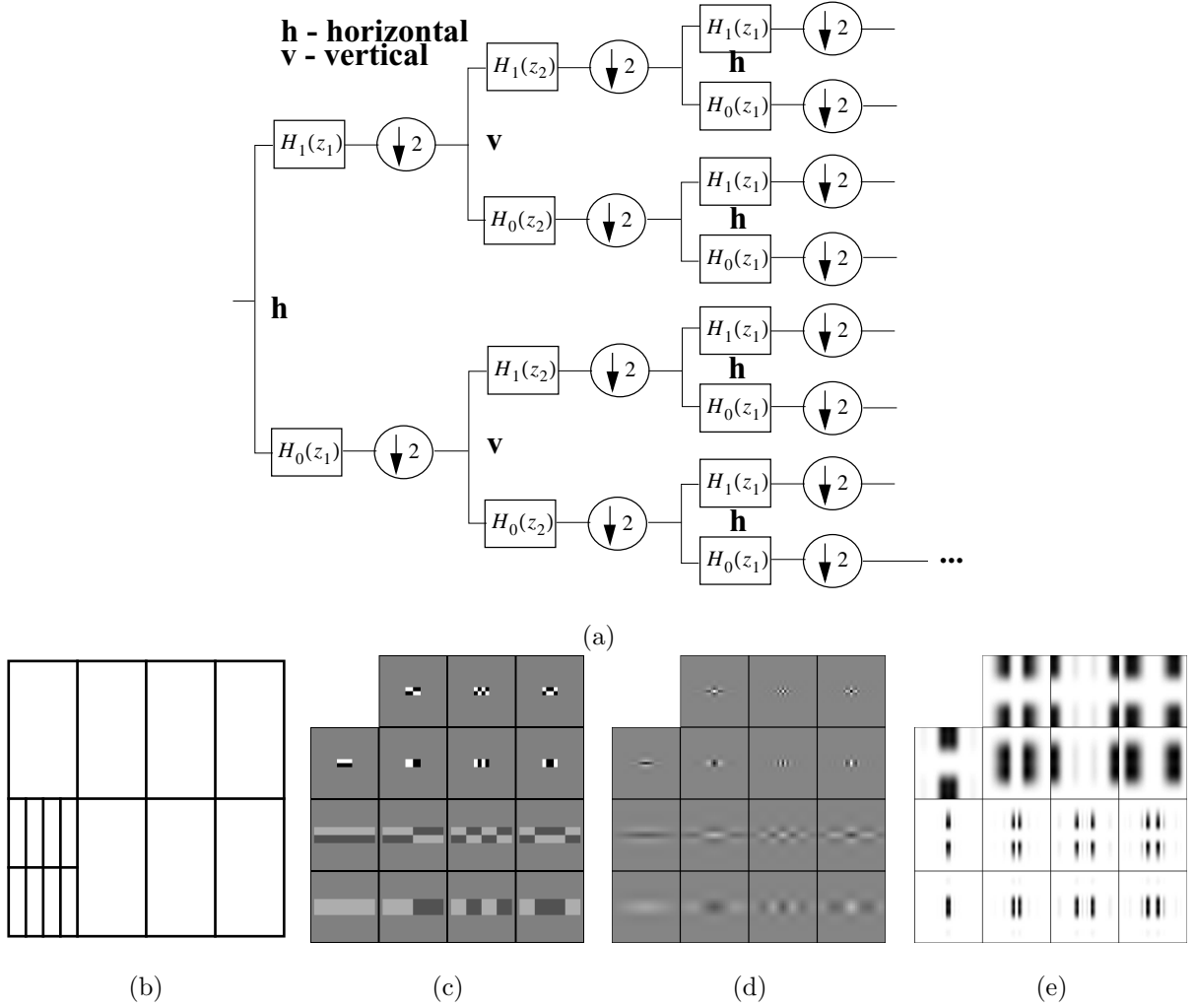


Figure 5: The AWT allows for asymmetric iteration of the filtering and subsampling applied on the LP, similarly as in the standard WT. Although this transform does not improve approximation of the class $\text{Mondrian}(k_1, k_2)$, it provides an efficient approximation tool for more general classes of images (Section 4). (a) The filtering scheme for the AWT(2,1), where one step of iteration is shown. (b) The decomposition in frequency. The basis functions obtained from the (c) Haar and (d) biorthogonal "9-7" 1-D scaling and wavelet functions. The basis functions are elongated by the factor $\rho = 2$. (e) The corresponding Fourier transform of the basis functions obtained from the "9-7" 1-D filters.

3.1. Discretization of Directions

To apply a discrete transform in the discrete space \mathbb{Z}^2 in a certain direction, we need to define the pixels that approximate the chosen direction. This problem has been considered in computer graphics in the 1960's by Bresenham (1965) as well as by Foley et al. (1990) and Chan and Yip (1996).

Recall that the set of points $(x, y) \in \mathbb{R}^2$ represents a continuous line with the slope r and intercept b if the following equality is satisfied:

$$y = rx + b. \quad (4)$$

The discrete approximation of Eq. (4) is called *digital line* $L(r, n)$. To preserve critical sampling in the transform, given a slope r , every pixel belongs to one and only one digital line $L(r, n)$. In that case, we say that, given a slope r , the set of digital lines $\{L(r, n) : n \in \mathbb{Z}\}$, *partitions the discrete space* \mathbb{Z}^2 .

The definitions of digital lines proposed by Bresenham (1965), Foley et al. (1990) and Chan and Yip (1996) are similar and here we give the definition that is a variation of the one given by Bresenham (1965). We show also below that such digital lines partition the discrete space \mathbb{Z}^2 .

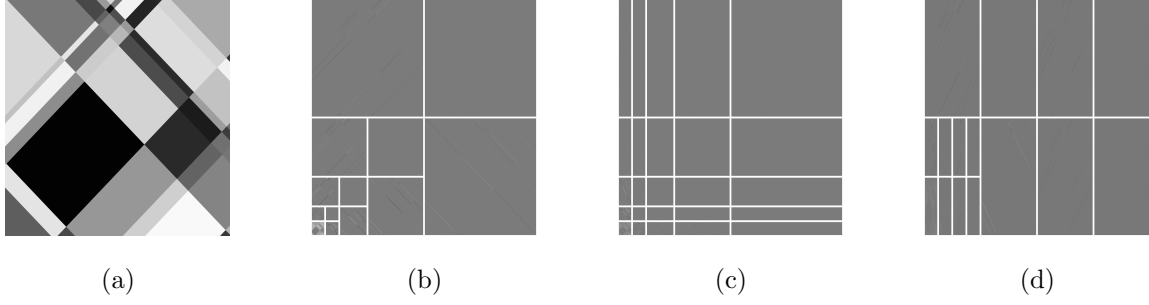


Figure 6: (a) An example of an image from the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, for $\mathbf{M} = [\mathbf{v}_1, \mathbf{v}_2]^T$, where $\mathbf{v}_1 = [1, 1]$ and $\mathbf{v}_2 = [-1, 1]$. The image is transformed using (b) S-WT, (c) S-FSWT, and (d) S-AWT($\mathbf{M}_\Lambda, 2, 1$) (directionlets), where all the transforms are built on the lattice Λ determined by the generator matrix $\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$.

Definition 2. Given a rational slope r , the digital line $L(r, n)$, where $n \in \mathbb{Z}$, is defined as the set of pixels (i, j) such that

$$\begin{aligned} j &= \lceil ri \rceil + n, \quad i \in \mathbb{Z}, |r| \leq 1, \text{ or} \\ i &= \lceil j/r \rceil + n, \quad j \in \mathbb{Z}, |r| > 1. \end{aligned} \quad (5)$$

Proposition 4. Given a rational slope r , the set of digital lines $\{L(r, n) : n \in \mathbb{Z}\}$ partitions the discrete space \mathbb{Z}^2 .

Proof. We give the proof only for the case $|r| \leq 1$. Similar arguments can be used for the other case.

For each pixel $(i, j) \in \mathbb{Z}^2$, we can find the intercept $n = j - \lceil ri \rceil$ such that the pixel belongs to the digital line $L(r, n)$. Furthermore, from Eq. (5) it follows that this intercept is unique. Therefore, the parallel digital lines $L(r, n)$, for a fixed rational r and $n \in \mathbb{Z}$, partitions the discrete space \mathbb{Z}^2 . \square

In the sequel, we show why the concept of digital lines do not provide an efficient framework when transforms are applied in different directions and critical sampling is enforced.

3.2. Directional Interaction

To explain the problem of *directional interaction*, let us first generalize the class **Mondrian** allowing for more directions. The class $\mathbf{S}\text{-Mondrian}$ consists of the skewed Mondrian-like images along two directions with the rational slopes $r_1 = b_1/a_1$ and $r_2 = b_2/a_2$, where a_1, a_2, b_1 , and b_2 are integers. To simplify notation, the two slopes are jointly denoted by the matrix

$$\mathbf{M}(r_1, r_2) = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}.$$

Definition 3. The class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ contains piecewise polynomial images of the size $M \times M$ with k_1 and k_2 discontinuities along the digital lines $L(r_1, n)$ and $L(r_2, n)$, respectively, where $n \in \mathbb{Z}$, $r_1 = b_1/a_1$, $r_2 = b_2/a_2$, and $a_1, a_2, b_1, b_2 \in \mathbb{Z}$.

Notice that the class $\mathbf{Mondrian}(k_1, k_2)$ is a special case of the larger class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ when $\mathbf{M}(r_1, r_2) = \mathbf{I}_2$. An example of an image from the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ is shown in Figure 6(a). Notice also that only the lines with rational slopes are used in the class $\mathbf{S}\text{-Mondrian}$. However, in spite of this constraint, a wealth of directions is still available, as we will explain in Section 3.3.

To provide a sparse representation of the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ and following the ideas from Section 2, we apply a 1-D WT along the digital lines $L(r_1, n)$, for $n \in \mathbb{Z}$. The transform produces two types of nonzero coefficients, that is, the coefficients corresponding to the discontinuities with the slopes r_1 and r_2 .

Since the HP filter imposes vanishing moments along the digital lines with the slope r_1 , the coefficients along this direction are annihilated in the HP subband, whereas the coefficients along the second direction with the slope r_2 are retained in both subbands. However, after subsampling, unlike in the case of the standard directions, the coefficients along the second direction are not aligned, that is, they cannot be

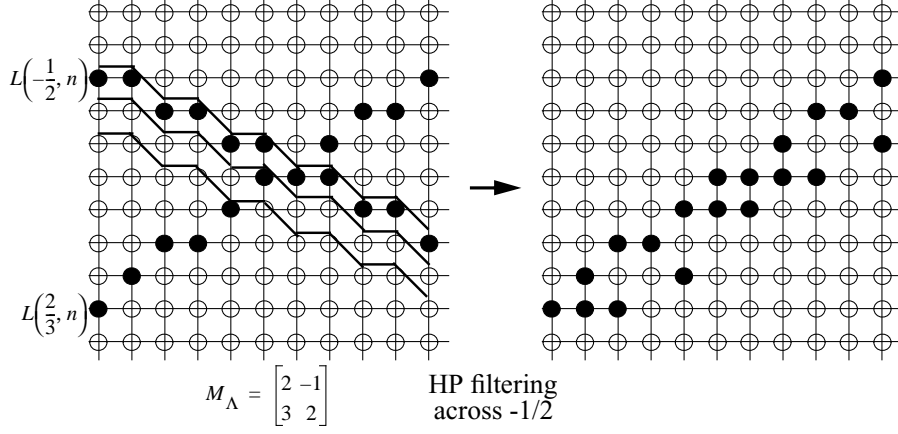


Figure 7: A 1-D WT is applied on an image from the class $\mathbf{S}\text{-Mondrian}(M(-1/2, 2/3), 1, 1)$ along the digital lines $L(-1/2, n)$. The HP filtering annihilates the digital line with the slope $-1/2$. However, the nonzero coefficients produced by the other line with the slope $2/3$ are not aligned in the digital lines $L(2/3, n)$. This is called *directional interaction*. Although the transform along digital lines is efficient when applied in oversampled schemes, it fails to provide a systematic subsampling method when critical sampling is enforced.

clustered in the digital lines with the slope r_2 . Therefore, the following 1-D WT applied along the digital lines with the slope r_2 does not annihilate the coefficients along the second direction and, hence, it yields a non-sparse representation. We call this phenomenon *directional interaction*. The proof is trivial and is omitted here. An example is shown in Figure 7.

Notice also that the concept of digital lines does not provide a systematic rule for subsampling in the case of iteration of the filtering and subsampling along the directions with the slopes r_1 and r_2 when critical sampling is enforced. To overcome the directional interaction and to propose an organized iterated subsampling method we use the concept of integer lattices.

3.3. Lattice-based Filtering and Subsampling

Instead of applying a transform along digital lines, we propose a method based on integer lattices (Conway and Sloane, 1998). We also prove that the lattice-based transforms can avoid directional interaction and are capable of providing the same order of approximation for the class $\mathbf{S}\text{-Mondrian}$ as the FSWT achieves for the class $\mathbf{Mondrian}$.

A full-rank integer lattice Λ consists of the points obtained as linear combinations of two linearly independent vectors, where both the components of the vectors and the coefficients are integers. Any integer lattice Λ is a sublattice of the cubic integer lattice \mathbb{Z}^2 , that is, $\Lambda \subset \mathbb{Z}^2$. The lattice Λ can be represented by a non-unique generator matrix

$$\mathbf{M}_\Lambda = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}, \text{ where } a_1, a_2, b_1, b_2 \in \mathbb{Z}. \quad (6)$$

Recall that the cubic lattice \mathbb{Z}^2 can be partitioned into $|\det(\mathbf{M}_\Lambda)|$ cosets of the lattice Λ (as shown by Conway and Sloane, 1998), where each coset is determined by the shift vector \mathbf{s}_k , for $k = 0, 1, \dots, |\det(\mathbf{M}_\Lambda)| - 1$. Therefore, the lattice Λ with the corresponding generator matrix \mathbf{M}_Λ given by Eq. (6), partitions each digital line $L(r_1, n)$, for $r_1 = b_1/a_1$, into *co-lines*. Notice that a co-line is simply the intersection between a coset and a digital line. Similarly, the digital line $L(r_2, n)$, for $r_2 = b_2/a_2$, is also partitioned into the corresponding co-lines (Figure 8).

We denote as $CL_{\mathbf{s}_k}(r_1, n)$ the co-line obtained as the intersection between the k th coset of the lattice Λ and the digital line $L(r_1, n)$. Notice that the co-line $CL_{\mathbf{s}_k}(r_1, n)$ consists of the pixels $\{c_1 \mathbf{d}_1 + c_2 \mathbf{d}_2 + \mathbf{s}_k : c_1 \in \mathbb{Z}, \}$, given a fixed $c_2 \in \mathbb{Z}$, where $n = \lceil c_2(b_2 - r_1 a_2) + s_{k,2} - r_1 s_{k,1} \rceil$ and $\mathbf{s}_k = [s_{k,1}, s_{k,2}]$.

Now we apply the 1-D WT (including the 1-D both filtering and subsampling operations) along the co-lines $\{CL_{\mathbf{s}_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(\mathbf{M}_\Lambda)| - 1\}$ (see also Velisavljević et al., 2003). Notice that both filtering and subsampling are applied in each of the cosets separately. Furthermore, each filtering

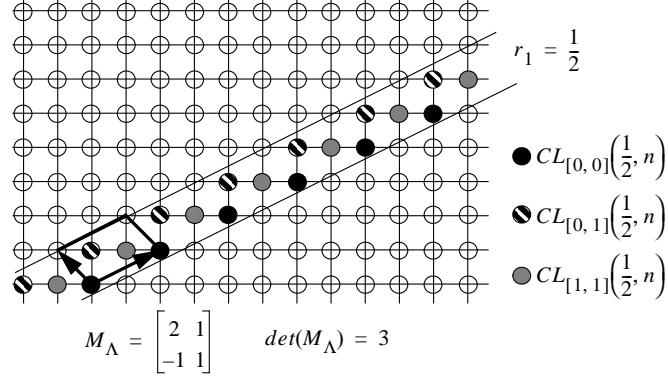


Figure 8: The intersections between the 3 cosets of the lattice Λ given by the generator matrix \mathbf{M}_Λ and the digital lines $L(r_1 = 1/2, n)$, for $n \in \mathbb{Z}$, are the co-lines $CL_{[0,0]}(1/2, n)$, $CL_{[0,1]}(1/2, n)$, and $CL_{[1,1]}(1/2, n)$.

operation is purely 1-D. After subsampling, the retained points belong to the sublattice Λ' of the lattice Λ ($\Lambda' \subset \Lambda$) with the corresponding generator matrix given by (see Figure 9(a))

$$\mathbf{M}_{\Lambda'} = \mathbf{D}_s \cdot \mathbf{M}_\Lambda = \begin{pmatrix} 2\mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}.$$

Here, \mathbf{D}_s is the horizontal subsampling operator, that is,

$$\mathbf{D}_s = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

We call the direction along the first vector \mathbf{d}_1 (with the slope $r_1 = b_1/a_1$), the *transform direction*. Similarly, the direction along the second vector \mathbf{d}_2 we call the *alignment direction*.

Therefore, since the filtering and subsampling are applied to each coset separately, the pixels retained after the subsampling are clustered in co-lines along the alignment direction. This property is crucial to avoid directional interaction (see Figure 9(b)).

Proposition 5. *Given a 1-D WT applied along the set of co-lines $\{CL_{\mathbf{s}_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(\mathbf{M}_\Lambda)| - 1\}$ to an image from the class $\mathcal{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, the transform coefficients in band-pass subbands that correspond to the discontinuities with the slope r_2 are aligned, that is, they can be clustered in the co-lines $CL_{\mathbf{s}_k}(r_2, n)$, $n \in \mathbb{Z}$.*

Proof. Recall that the co-line $CL_{\mathbf{s}_k}(r_1, n)$ consists of the pixels $\{(i, j)\}$, such that $i = c_1 a_1 + c_2 a_2 + s_{k,1}$ and $j = c_1 b_1 + c_2 b_2 + s_{k,2}$, for each $c_1 \in \mathbb{Z}$ and a fixed $c_2 \in \mathbb{Z}$. After the subsampling, the retained pixels belong to the lattice Λ' and, thus, the corresponding co-lines consist of the pixels $\{(i, j)\}$, such that $i = c_1 \cdot 2a_1 + c_2 a_2 + s_{k,1}$ and $j = c_1 \cdot 2b_1 + c_2 b_2 + s_{k,2}$, for each $c_1 \in \mathbb{Z}$ and a fixed $c_2 \in \mathbb{Z}$.

Notice that the co-lines $CL_{\mathbf{s}_k}(r_2, n)$ with the other slope r_2 that correspond to the lattice Λ' consist of the same pixels. Therefore all the retained pixels are aligned in the direction with the slope r_2 . \square

Combining lattices with the different constructions given in Section 2, we build *skewed wavelet transforms*.

3.4. Skewed Wavelet Transforms

The transforms defined in Section 2 (the standard WT, FSWT, and AWT) are inefficient when applied to the class $\mathcal{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, unless $\mathbf{M}(r_1, r_2)$ is the identity matrix. Since the directions of the transforms and discontinuities in images are not matched, the transforms fail to provide a compact representation. The following proposition gives the orders of approximation that can be achieved by the three transforms with the standard directions.

Proposition 6. *Given an $M \times M$ image from the class $\mathcal{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, where $\mathbf{M}(r_1, r_2)$ is not the identity matrix, the standard WT, FSWT, and AWT with 1-D wavelets, having enough vanishing moments, provide $O[(k_1 + k_2)M]$ nonzero transform coefficients in band-pass subbands.*

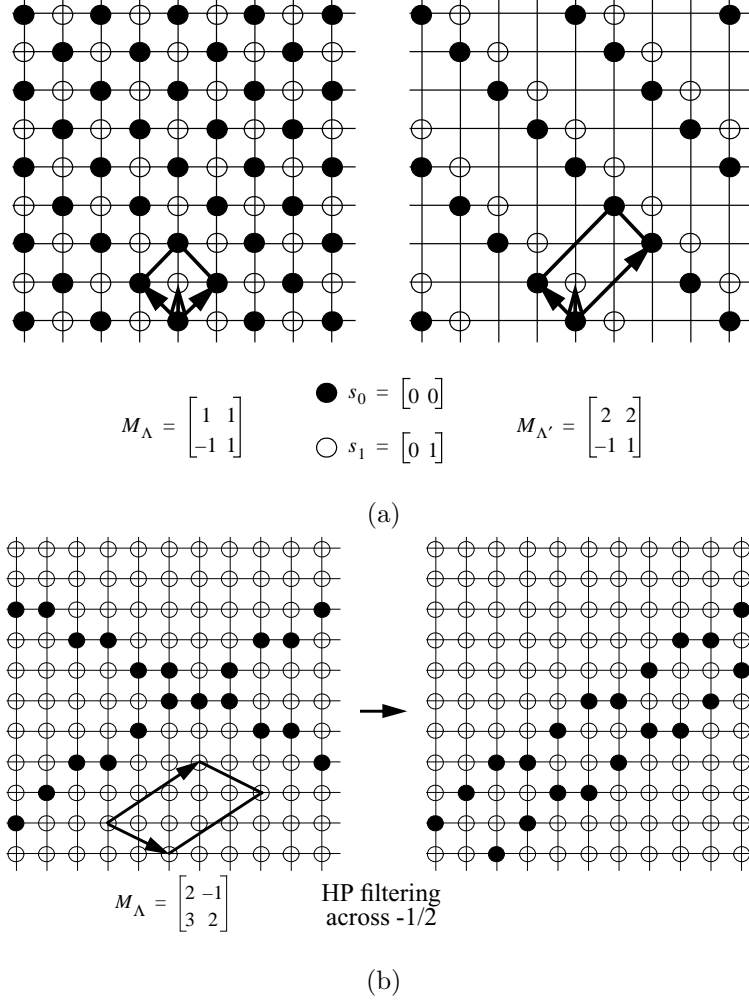


Figure 9: (a) The lattice Λ is determined by the generator matrix \mathbf{M}_{Λ} . 1-D Filtering is applied along the co-lines $\{CL_{\mathbf{s}_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(\mathbf{M}_{\Lambda})| - 1\}$, where the slope r_1 corresponds to the vector $[1, 1]$, that is, along 45° . The pixels retained after the subsampling belong to the lattice $\Lambda' \subset \Lambda$ determined by the generator matrix $\mathbf{M}_{\Lambda'}$. Notice that filtering and subsampling are applied separately in two cosets, determined by the shift vectors \mathbf{s}_0 and \mathbf{s}_1 . (b) The nonzero pixels obtained after one step of the lattice-based filtering operation applied on the same example as in Figure 7 are clustered in the digital lines with the slope $2/3$.

Proof. The subbands produced by the FSWT are indexed by (j_1, j_2) , where $1 \leq j_1, j_2 \leq \log_2 M$. Each subband contains $O[k_1 M/2^{j_1} + k_2 M/2^{j_2}]$ nonzero coefficients. The total number is given by

$$N = \sum_{j_1=1}^{\log_2 M} \sum_{j_2=1}^{\log_2 M} O \left[k_1 \frac{M}{2^{j_1}} + k_2 \frac{M}{2^{j_2}} \right] = O[(k_1 + k_2)M].$$

Notice that the standard WT, as a special case of the AWT, has the same behavior. Thus, we give the proof only for the AWT. The $\text{AWT}(n_1, n_2)$ produces $2^{n_1+n_2} - 1$ band-pass and HP subbands at each scale j . Each of these subbands contain $n(j) = O[(2^{n_1+n_2} - 1)M(2^{-n_1 j} + 2^{-n_2 j})]$ nonzero coefficients. Therefore, the total number of nonzero coefficients is given by

$$\sum_{j=1}^{\frac{\log_2 M}{\max(n_1, n_2)}} n(j) = O[(k_1 + k_2)M].$$

□

Table 2: Orders of approximation by the S-WT, S-FSWT and S-AWT (directionlets) built on the lattice Λ determined by \mathbf{M}_Λ applied on the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$.

	$\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$	$\mathbf{M}_\Lambda \neq \mathbf{M}(r_1, r_2)$
S-WT	$(k_1 + k_2)M$	$(k_1 + k_2)M$
S-FSWT	$(k_1 + k_2)(\log_2 M)^2$	$(k_1 + k_2)M$
S-AWT	$(k_1 a + k_2/a)M$	$(k_1 + k_2)M$

Using integer lattices, we define three new transforms, which are *skewed versions* of the standard WT, FSWT, and AWT. Given a lattice Λ , the skewed transforms are applied along co-lines in the transform and alignment directions of the lattice Λ , retaining the same frequency decompositions as the corresponding transforms along the standard directions explained in Section 2. Thus, following the notation introduced in Section 2.2, we denote as $\mathbf{S}\text{-AWT}(\mathbf{M}_\Lambda, n_1, n_2)$ the skewed asymmetric transform built on the lattice Λ that has n_1 and n_2 transforms in one iteration step along the transform and alignment directions, respectively. We call the basis functions of the S-AWT *directionlets* since they are asymmetric and have a specific direction. Similarly, we denote the skewed standard WT as S-WT and the skewed FSWT as S-FSWT. The corresponding basis functions are shown in Figure 10 for the directions along the vectors $\mathbf{d}_1 = [1, 1]$ and $\mathbf{d}_2 = [-1, 1]$. Notice that the skewed transforms are applied in all cosets of the lattice Λ separately.

The basis functions of the skewed transforms have DVMs in *any* two directions with rational slopes. Recall that the L th order DVM along the direction with a rational slope $r_1 = b_1/a_1$ is equivalent to requiring the z -transform of a basis function to have a factor $(1 - z_1^{-a_1} z_2^{-b_1})^L$ (Do and Vetterli, 2005, Viscito and Allebach, 1991). The following proposition gives the number and directions of the DVM in directionlets.

Proposition 7. *Assume that the directionlets of the $\mathbf{S}\text{-AWT}(\mathbf{M}_\Lambda, n_1, n_2)$ are obtained using a 1-D wavelet with L vanishing moments. Then, at each scale of the iteration, there are:*

- (a) $2^{n_1} - 1$ directionlets with the L th order DVM along the transform direction of the lattice Λ ,
- (b) $2^{n_2} - 1$ directionlets with the L th order DVM along the alignment direction of the lattice Λ , and
- (c) $(2^{n_1} - 1)(2^{n_2} - 1)$ directionlets with the L th order DVM along both directions.

Proof. Recall first from Viscito and Allebach (1991) that 1-D filtering using the filter $H(z)$ along the transform direction of the lattice Λ is equivalent to filtering in the 2-D discrete space using $H(z_1^{a_1} z_2^{b_1})$. Similarly, filtering along the alignment direction of the lattice Λ is equivalent to filtering in the 2-D discrete space using $H(z_1^{a_2} z_2^{b_2})$. Since the 1-D HP filter has L vanishing moments, its z -transform has a factor $(1 - z^{-1})^L$. Therefore, the HP filtering along the transform and alignment directions uses the equivalent filters with the factors $(1 - z_1^{-a_1} z_2^{-b_1})^L$ and $(1 - z_1^{-a_2} z_2^{-b_2})^L$, respectively, in the z -transforms.

Filtering using the 1-D two-band filter bank along two directions in the construction of the S-AWT (see Figure 5(a)) yields (a) $2^{n_1} - 1$ subbands with HP filtering along only the transform direction, (b) $2^{n_2} - 1$ subbands with HP filtering along only the alignment direction, and (c) $(2^{n_1} - 1)(2^{n_2} - 1)$ subbands with HP filtering along both directions. Thus, the statement of the proposition follows directly. \square

Efficiency of the representation of the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ by the three skewed transforms depends on matching between the directions of discontinuities and the directions used in these transforms. If these directions are matched, then the orders of nonzero coefficients in band-pass subbands are equal to the orders calculated in Section 2 (see Table 1). Otherwise, they are given by Proposition 6. The following proposition formalizes this statement. The proof is omitted since it uses the same arguments as in Propositions 1 to 3.

Proposition 8. *Given an $M \times M$ image from the class $\mathbf{S}\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, the S-WT, S-FSWT and $\mathbf{S}\text{-AWT}(\mathbf{M}_\Lambda, n_1, n_2)$ with 1-D wavelets having enough vanishing moments built on the lattice Λ determined by the generator matrix $\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$ give $O[(k_1 + k_2)M]$, $O[(k_1 + k_2)(\log_2 M)^2]$ and $O[(k_1 a + k_2/a)M]$ nonzero coefficients in band-pass subbands, respectively. Here, $a = (2^{n_2} - 1)/(2^{n_1} - 1)$.*

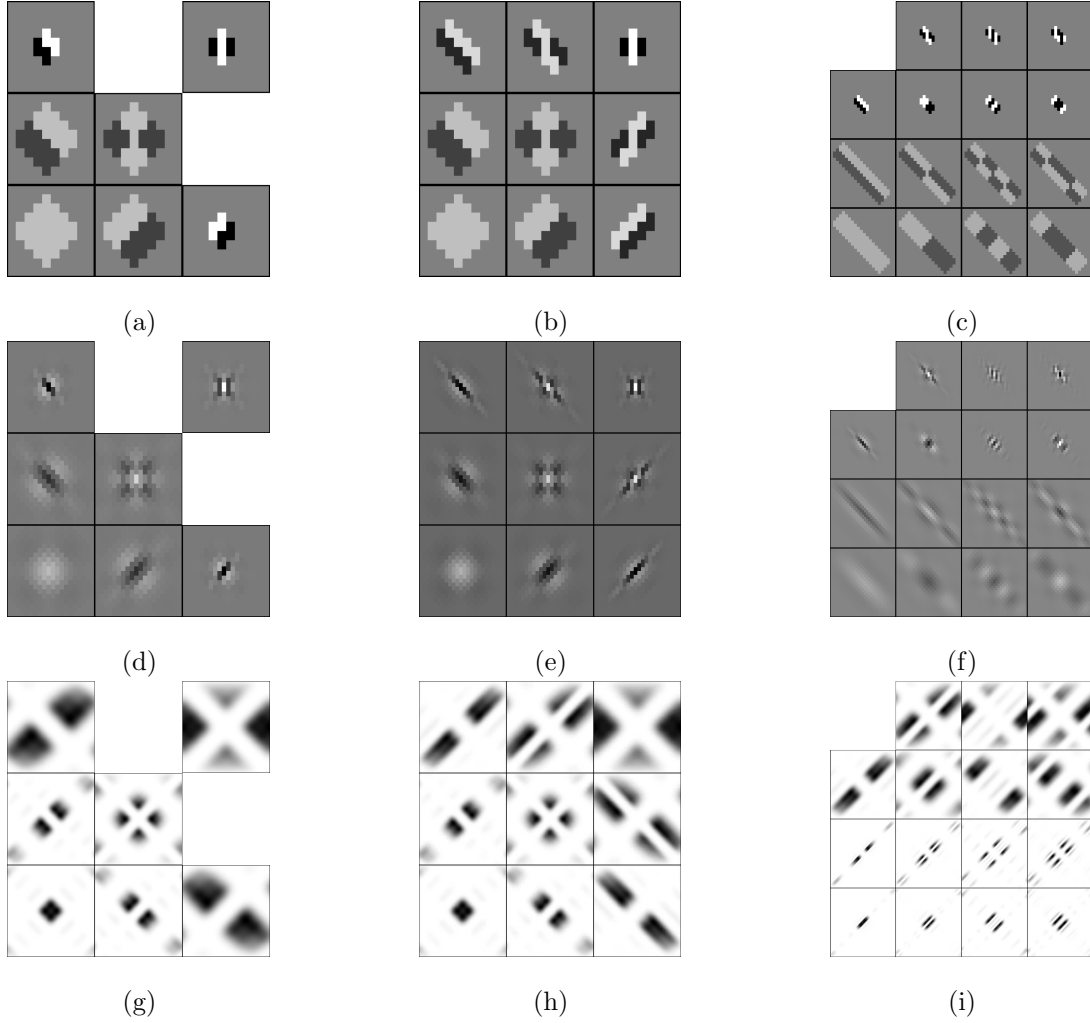


Figure 10: The basis functions obtained by the skewed transforms using the Haar 1-D scaling and wavelet functions: (a) S-WT, (b) S-FSWT, (c) S-AWT($\mathbf{M}_\Lambda, 2, 1$) (directionlets). The same, but with the biorthogonal "9-7" 1-D scaling and wavelet functions: (d) S-WT, (e) S-FSWT, (f) S-AWT($\mathbf{M}_\Lambda, 2, 1$) (directionlets). In all cases $\mathbf{M}_\Lambda = [\mathbf{d}_1, \mathbf{d}_2]^T$, where $\mathbf{d}_1 = [1, 1]$, and $\mathbf{d}_2 = [-1, 1]$. The DVMs are imposed along the vectors \mathbf{d}_1 and \mathbf{d}_2 , that is, along 45° and -45° . The corresponding Fourier transforms: (g) S-WT, (h) S-FSWT, (i) S-AWT($\mathbf{M}_\Lambda, 2, 1$) (directionlets).

The transforms of the image shown in Figure 6(a) are given in Figure 6(b)-(d). The applied transforms are S-WT, S-FSWT, and S-AWT($\mathbf{M}_\Lambda, 2, 1$), where $\mathbf{M}(r_1, r_2) = \mathbf{M}_\Lambda$. Table 2 summarizes the orders of nonzero coefficients in band-pass subbands in the case of both matched and mismatched directions.

Notice that the lattice-based method allows for a more general construction of M-DIR transforms using more than two directions in an arbitrary order. Such M-DIR transforms and their properties are beyond the scope of this paper. More details are given by Velisavljević et al. (2003) and Velisavljević (2005).

3.5. Polyphase Representation

Filtering and subsampling across lattices, as explained in Section 3.3, can be efficiently represented in the polyphase domain. Recall first that a two-band 1-D filter bank $(H_0(z), H_1(z))$ followed by a subsampler by the factor 2 can be given in terms of the polyphase components as (Vetterli and Kovačević, 1995)

$$\begin{aligned} H_0(z) &= H_{00}(z^2) + zH_{01}(z^2) \\ H_1(z) &= H_{10}(z^2) + zH_{11}(z^2). \end{aligned}$$

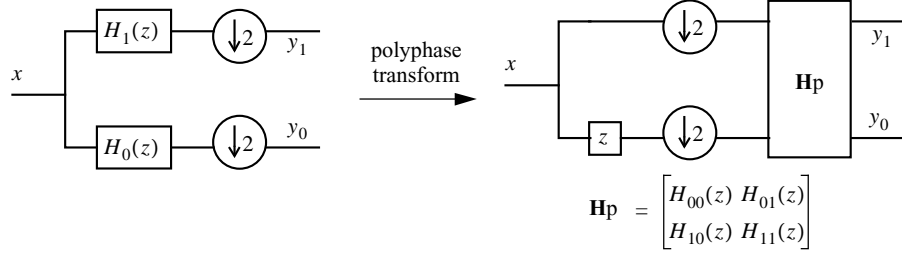


Figure 11: A 1-D filter bank $(H_0(z), H_1(z))$ with the subsampling factor 2 is represented in the polyphase domain with the corresponding polyphase components $H_{00}(z)$, $H_{01}(z)$, $H_{10}(z)$, and $H_{11}(z)$.

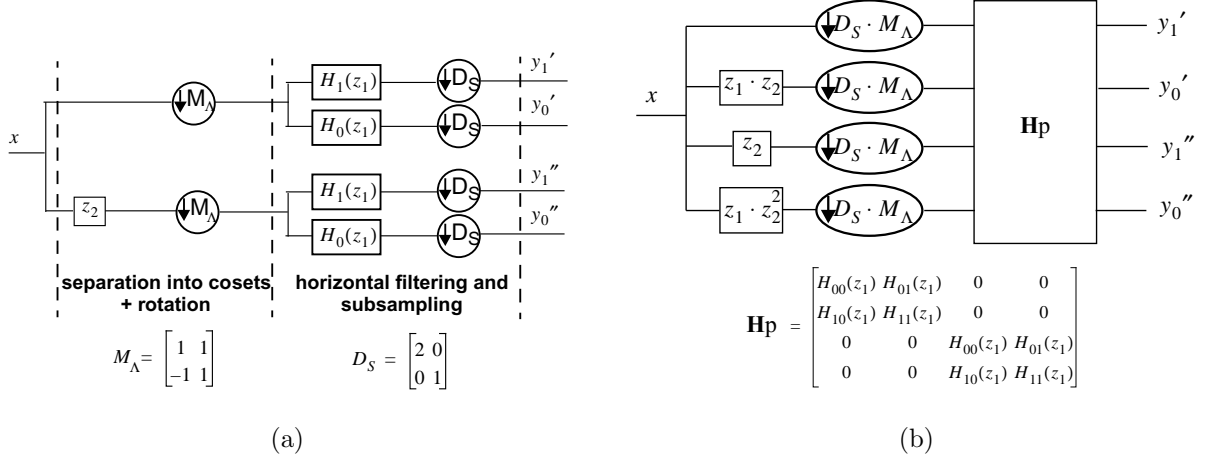


Figure 12: (a) The 2-D two-band filter bank applied in the example shown in Figure 9(a). Filtering and subsampling are applied in 2 cosets separately. (b) Equivalent polyphase representation contains 4 components. The polyphase transform \mathbf{H}_p is block-diagonal.

Here, H_{00} , H_{01} , H_{10} , and H_{11} are the polyphase components of the filters $H_0(z)$ and $H_1(z)$ that correspond to even and odd samples of the impulse response, respectively. Such a polyphase representation is shown in Figure 11.

Similarly, we can find the equivalent polyphase components of a 2-D filter bank $(H_0(\mathbf{z}), H_1(\mathbf{z}))$, where $\mathbf{z} = (z_1, z_2)$, applied in the lattice-based method, as explained in Section 3.3. Recall that the filters $H_0(\mathbf{z})$ and $H_1(\mathbf{z})$ used in this method are purely 1-D filters, that is, $H_0(\mathbf{z}) = H_0(z_1)$ and $H_1(\mathbf{z}) = H_1(z_1)$. To illustrate this polyphase decomposition, we consider the particular example with the lattice Λ determined by the generator matrix

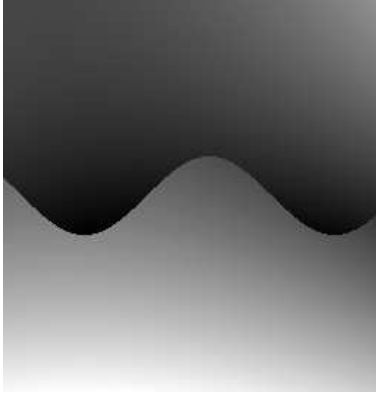
$$\mathbf{M}_\Lambda = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix},$$

as shown in Figure 9(a). Recall that the lattice-based filtering and subsampling are applied in each coset of the lattice Λ separately. Thus, the equivalent scheme has two sections, which are (a) separation into two cosets and (b) 1-D filtering and subsampling in the transform direction (Figure 12(a)). Notice that filtering in the transform direction is performed as horizontal filtering preceded by rotation by the generator matrix \mathbf{M}_Λ .

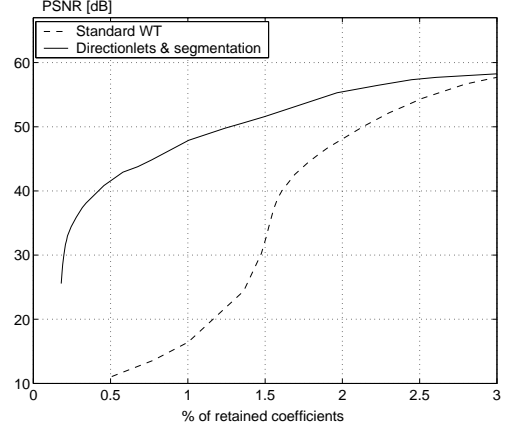
Since the total subsampling rate is $|\det(\mathbf{D}_s \cdot \mathbf{M}_\Lambda)| = 4$, the polyphase representation of such a filter bank consists of 4 polyphase components. The equivalent polyphase representation is shown in Figure 12(b), where the polyphase transform \mathbf{H}_p is *block-diagonal*, that is,

$$\mathbf{H}_p = \begin{pmatrix} H_{00}(z_1) & H_{01}(z_1) & 0 & 0 \\ H_{10}(z_1) & H_{11}(z_1) & 0 & 0 \\ 0 & 0 & H_{00}(z_1) & H_{01}(z_1) \\ 0 & 0 & H_{10}(z_1) & H_{11}(z_1) \end{pmatrix}.$$

Notice that the block-diagonal polyphase transform with two identical blocks is a consequence of the



(a)



(b)

Figure 13: An example of NLA of an image from the class C^2/C^2 . (a) An image from the class C^2/C^2 is approximated using the standard WT and the S-AWT($\Lambda, 2, 1$) with spatial segmentation. (b) The MSE expressed in terms of PSNR is significantly reduced in the case of the S-AWT($\Lambda, 2, 1$).

separable transforms applied across cosets. This property allows for a simple filter design and computational efficiency in the polyphase domain. Such separability in the polyphase domain has also been used in other 2-D filter bank designs by Bamberger and Smith (1992) and Phoong et al. (1995).

4. Non-linear Image Approximation

The main task of approximation is to represent a signal by a portion of the transform coefficients, while the rest of them is set to zero. The transform can be critically sampled (bases) or oversampled (frames). The approximation with N retained transform coefficients is also called *N-term approximation*. We distinguish between linear approximation (LA) and non-linear approximation (NLA). In the first, the indexes of the retained coefficients are fixed, whereas in the latter, they are adapted to the content of the signal.

Owing to truncation of the coefficients, the approximating signal does not match exactly the original one. The quality of the approximation is commonly measured in terms of *mean-square error* (MSE), that is, for a signal \mathbf{x} and its N -term approximation $\hat{\mathbf{x}}_N$, the MSE is given by $\|\mathbf{x} - \hat{\mathbf{x}}_N\|^2$. Notice that, given a signal \mathbf{x} and its transform $\mathbf{y} = \mathbf{F} \cdot \mathbf{x}$, where \mathbf{F} is a tight frame or an orthogonal basis, we have the following inequality

$$\|\mathbf{x} - \hat{\mathbf{x}}_N\|^2 \leq \frac{1}{A} \|\mathbf{y} - \hat{\mathbf{y}}_N\|^2, \quad (7)$$

where $\hat{\mathbf{y}}_N$ corresponds to the truncated version of \mathbf{y} with N retained coefficients, the N -term approximation $\hat{\mathbf{x}}_N$ is given by $\hat{\mathbf{x}}_N = A^{-1} \mathbf{F}^T \cdot \hat{\mathbf{y}}_N$, and A is the frame bound of \mathbf{F} . Equality in Eq. (7) holds if the transform \mathbf{F} is an orthogonal basis (for more details see Velisavljević et al., 2006).

In the orthogonal case, the optimal strategy to minimize the MSE is to retain the largest-magnitude transform coefficients (as shown by Mallat, 1997). Notice that the MSE decays as the number N of retained coefficients (approximants) grows.

Compression using orthogonal transforms is an extension of NLA that consists of (a) approximation, (b) indexing the retained coefficients, and (c) quantization of the coefficients. Some algorithms merge quantization and NLA into a single operation producing an embedded bitstream, like SPIHT proposed by Said and Pearlman (1996) or zerotrees explained by Shapiro (1993). Thus, the MSE (in this case also called *distortion*) is affected by the two factors: (a) truncation error due to NLA and (b) quantization error.

The asymptotic *rate of decay* of the MSE, as N tends to infinity, is a fundamental approximation property of the transform and this value allows us to compare approximation performance of different transforms. The higher the rate of decay, the more efficient the transform is. Similarly, the rate of decay

in compression is defined as the asymptotic behavior of the distortion D , as the bitrate R tends to infinity (this is frequently called *R-D behavior*).

Mallat (1997) and DeVore (1998) showed that, for a 2-D piecewise C^2 smooth signal $f(x_1, x_2)$ with a 1-D C^2 smooth discontinuity curve (which we call C^2/C^2 signal), the lower bound of the MSE is given by $O[N^{-2}]$. Here, C^2 smoothness of both 1-D and 2-D functions means that the functions are twice continuously differentiable. An example of a C^2/C^2 image is shown in Figure 13(a).

Notice that the standard WT is far from optimal having the rate of decay equal to $O[N^{-1}]$ (Mallat, 1997, Donoho et al., 1998). Some other adaptive or non-adaptive methods have been shown to improve substantially the approximation power. Curvelets ((Candès and Donoho, 1999, 2002)) and contourlets (Do and Vetterli, 2005) can achieve the rate $O[N^{-2}(\log N)^3]$, which is nearly optimal. Furthermore, bandelets (LePennec and Mallat, 2000, 2003) and wedgelets (Donoho, 1999, Romberg et al., 2002, 2003, Wakin et al., 2006) have been shown to perform indeed optimally. However, notice that none of these methods is based on critically sampled filter banks, which are very convenient for compression. Furthermore, a complex non-separable processing is sometimes required.

As we showed in Sections 2 and 3, asymmetry and multi-directionality improve the approximation power of the WT while keeping separability, simplicity, and critical sampling. However, the S-FSWT cannot yield a high rate of decay since it fails to provide a sparse representation of C^2/C^2 images. In turn, the S-AWT is capable of producing a compact representation, but it is still sensitive to the choice of the transform and alignment directions.

Synthetic (including also C^2/C^2) and natural images have geometrical features that vary over the space. Directionality, thus, can be considered as a local characteristic, defined in a small neighborhood. This implies the necessity for *spatial segmentation* as a way of partitioning an image into smaller segments with one or a few dominant directions per segment.

The S-AWT is applied on a segmented image, where the transform and alignment directions are chosen independently in each segment. The transform outperforms the standard WT in both approximation and compression rate of decay of the MSE (i.e. distortion). The following theorem gives the rate of decay for C^2/C^2 images.

Theorem 1. *Given a 2-D C^2/C^2 function $f(x_1, x_2)$ and $\alpha = (\sqrt{17} - 1)/2 \approx 1.562$,*

(a) *The N -term approximation by the S-AWT using spatial segmentation achieves*

$$MSE = \|f - \hat{f}_N\|^2 = O[N^{-\alpha}].$$

In that case the optimal asymmetry factor is $\rho^ = \alpha$.*

(b) *Compression by the S-AWT, using spatial segmentation and using R bits for encoding, can achieve the distortion D given by*

$$D = O[R^{-\alpha}].$$

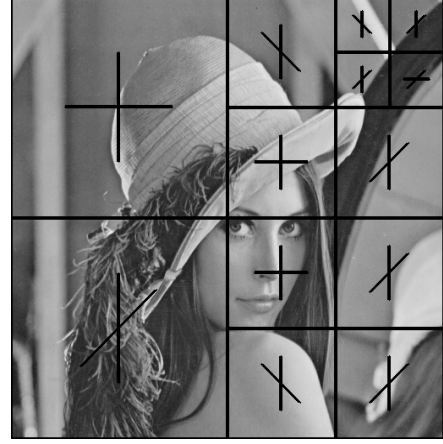
The proof of the theorem is given in Appendix I.

Notice that asymmetric segmentation is used here in the iteration, that is, an image is partitioned into vertical strips of equal widths. The number of segmentation steps depends on the asymmetry factor, the number of approximants, the number of transform directions, and the first derivative of the C^2 curve (see the proof of Theorem 1). In particular, when the optimal asymmetry factor $\rho^* = \alpha$ is used, the number of segmentation steps does not increase with the number of approximants. However, in reality, because of the discreteness of the transform, this asymmetry factor cannot be exactly achieved and, in general, the number of segmentation steps has to be increased with the number of approximants. Notice that the S-AWT($\Lambda, 3, 2$) approximates well the optimal transform, while retaining iterative segmentation (there are other possible transforms with the asymmetry factor even closer to optimal but we choose this one for the sake of simplicity). It follows from the proof of Theorem 1 that the number of required transform directions grows with the number of segmentation steps as $O[2^{\beta s}]$. Table 3 gives the achievable approximation and segmentation rates for the S-AWT($\Lambda, 3, 2$) and different values of β .

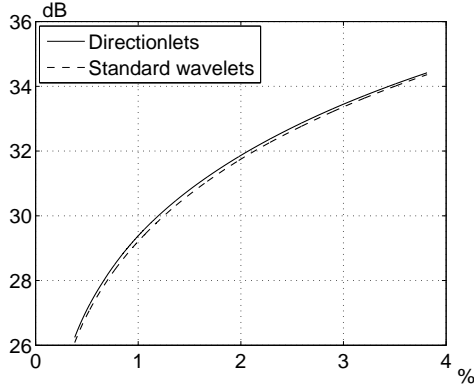
Although the obtained approximation rate is slower than the ones obtained in the other related results, we want to emphasize that the S-AWT($\Lambda, 3, 2$) is *critically sampled* and uses only *separable processing*. This is important for compression because, in the case of orthogonal 1-D filter banks, the Lagrangian optimization-based algorithms still can be applied, making it easier to achieve very good compression.



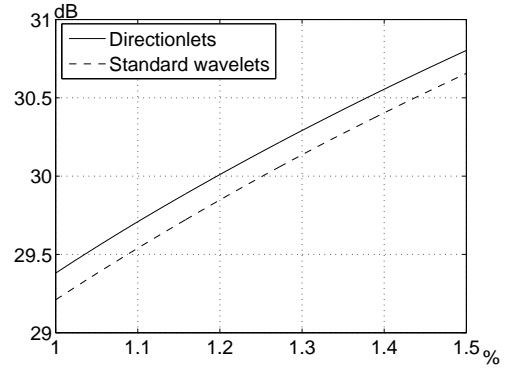
(a)



(b)



(c)



(d)

Figure 14: Non-linear approximation performance: *Lena* image. (a) The original image. (b) The optimal segmentation and the optimal choice of transform directions in each segment. (c) The NLA results obtained by retaining a portion of transform coefficients. (d) A magnified interval of the results shown in (c). Directionlets outperform the standard WT in terms of the approximation power, especially at low approximation rates.

Table 3: Dependence of the approximation rate $\text{MSE} = O[N^{-e_1^*}]$ and the number of segmentation levels $s = \eta_1 \log_2(N)$ on the growth rate of the number of transform directions β in the case of the S-AWT($\Lambda, 3, 2$).

β	2	1	0.5	0.25
η_1	1/51	1/26	2/27	4/29
e_1^*	1.55	1.50	1.41	1.24

Recall from Section 3.3 that the S-AWT($\Lambda, 3, 2$) is applied in the $|\det(\mathbf{M}_\Lambda)|$ cosets separately. The separate filtering and subsampling in the cosets affect the order of decay of the MSE, but only up to a constant factor and, thus, the rate of decay remains the same.

Figure 13 illustrates the gain obtained by NLA using the S-AWT($\Lambda, 2, 1$) with spatial segmentation applied on an image from the class C^2/C^2 when compared to the results of NLA obtained using the standard WT. Furthermore, we present the results of NLA of a natural image *Lena* shown in Figure 14(a). The optimal segmentation and the choice of transform directions in each segment is illustrated in Figure 14(b). Notice that the optimal transform directions follow the dominant directions in segments, as expected from the optimization method. A comparison between the NLA results obtained by directionlets and the standard WT is shown in Figure 14(c). The gain is obtained because directionlets are capable of



Figure 15: The NLA results of Lena: (a,b,c) Reconstructions of the image Lena for 0.5%, 1.0% and 1.5% retained transform coefficients, respectively, using directionlets. The quality of the obtained images is 27.10dB, 29.38dB and 30.80dB, respectively. (d,e,f) The reconstructions at the same approximation rates using the standard WT. The quality of the images is 26.93dB, 29.21dB and 30.66dB, respectively. The reconstructions obtained using directionlets are better than the ones obtained by the standard WT both numerically and visually. Moreover, the artifacts that appear in the reconstructions obtained using directionlets are also oriented in the dominant directions, making them less objectionable.

better capturing locally dominant directions in the image. In Figure 15, the shown reconstructed images are obtained for different number of retained transform coefficients using the two transforms. Both the numerical results and the visual quality of the reconstructions obtained by directionlets are better than the same in the case of the standard WT. Furthermore, the artifacts that appear in the reconstructions for a very small number of retained coefficients are oriented in the transform directions and aligned with the dominant directions in segments. Hence, these artifacts are less perceptually annoying than in the case of the standard WT.

5. Space-Frequency Quantization with Directionlets

We now show how directionlets can be successfully combined with the standard space-frequency quantization (SFQ) compression approach, resulting in an adaptive compression method using Lagrangian optimization. First, we review the standard SFQ proposed by Xiong et al. (1997). Then, we motivate and explain our spatial segmentation algorithm and the local adaptation of transform directions in each image segment. Furthermore, we explain the construction of our adaptive compression algorithm and, finally, we describe a post-processing of the reconstructed images in order to remove the blocking effect, which appears at low bit-rate compression due to the independent treatment of each block. We also analyze computational complexity of the entire method and show the corresponding results. Notice that most of the ideas and results in this section have been previously presented in Velisavljević et al. (2007).

5.1. Space-Frequency Quantization

The SFQ image compression method for images was originally proposed by Xiong et al. (1997). In the follow-up work by Xiong et al. (1998), the authors analyzed the extension of the SFQ using wavelet

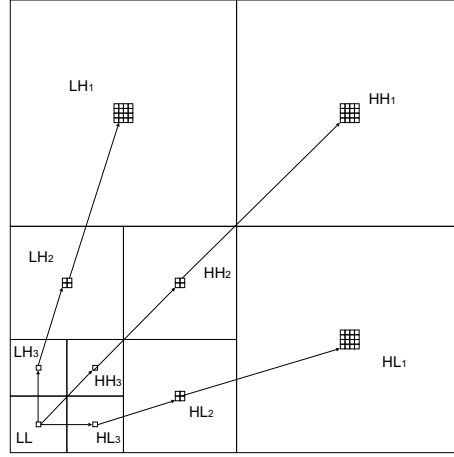


Figure 16: The wavelet coefficients are grouped in tree-structures to exploit the multi-scale correlation. Each coefficient has four children in the next finer scale in the 2×2 region that corresponds to the same spatial location. The exceptions are the coefficients at the coarsest scale, which have only three children, and the ones in the finest scale, which have no children.

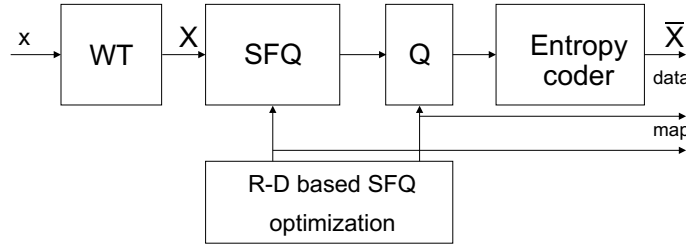


Figure 17: The standard SFQ encoding consists of four blocks. First, the standard 2-D WT is applied on the input image x . Then, the SFQ encoder decides which subset of the wavelet coefficients should be discarded based on the R-D optimization. The retained coefficients are quantized using the optimal quantization step size, which is also chosen based on the R-D optimization. Finally, the output stream of coding symbols is entropy coded. The data \bar{X} is transmitted together with the map information and the quantization step size as a side information.

packets. Here, we briefly revisit the SFQ and explain the basic concept.

The main idea behind SFQ is to minimize a MSE distortion measure of the reconstructed image for a given bit-rate constraint. The algorithm exploits the multi-scale correlation among wavelet coefficients produced by the standard 2-D WT. The coefficients are structured in multi-scale trees so that one tree consists of the coefficients from different transform scales at the same spatial location (see Figure 16). Each tree has a root at the corresponding coefficient from the coarsest scale. These coefficients have three children represented as nodes in the next finer scale. In turn, each of these children-nodes have four children in the next subsequent scale with the same orientation grouped in the 2×2 region at the same spatial location. The tree-structure grows iteratively through scales until the finest scale is reached, where the corresponding nodes have no children. The same tree-structure is used by Shapiro (1993), where it is referred to as *zerotrees*, whereas a similar one is exploited by Said and Pearlman (1996).

In the process of the SFQ encoding, a subset of wavelet coefficients is discarded (set to zero), whereas the rest is quantized using a single uniform scalar quantizer. The main tasks of the SFQ are (a) to select the subset of coefficients that should be discarded and (b) to choose which quantization step size should be used to quantize the retained coefficients. In both tasks, Lagrangian optimization (see Appendix II) is used to select the optimal solution in the R-D sense. The locations of the retained coefficients are encoded and sent as a map information, whereas the quantized magnitudes are entropy coded. The block diagram of the encoder is shown in Figure 17.

The optimization process consists of three phases: (a) space-frequency tree pruning, (b) predicting the map and (c) joint optimization of the quantizers. Notice that even though the optimal result of the tree

pruning is influenced by the bit-rate spent for predicting and encoding the map in (b), the optimization process in (a) is assumed to be independent and is updated in the subsequent phase. Notice also that the MSE is measured directly in the transform domain to reduce the computational complexity, with no loss of optimality due to the orthogonality.

The optimization process in the first phase (a) is defined as follows. For all nodes in the full depth multi-scale tree, check bottom-up if it is cheaper to keep or to zero out the descendant nodes in a R-D sense. The cost of pruning the descendants is equal to the sum of squares of the descendant magnitudes (since no bit-rate is spent). The cost of keeping and encoding the descendants is the Lagrangian cost obtained combining the entropy-based estimate of the bit-rate and the quantization MSE. The process is iterated on the resulting pruned multi-scale tree using the updated entropy-based estimates of the bit-rates until the convergence is reached, that is, until no new node is pruned. As a result of the pruning process, we can assign a binary map to each node defining if the node has children or not.

In the second phase (b), the locations of the retained nodes are encoded as a map information using a generalized version of the predictive scheme proposed by Lewis and Knowles (1992), where each subband is processed independently. First, the variance of each parent node is calculated as the energy of a 3×3 block of coefficients from the same subband centered at the corresponding coefficient using the quantized values. Then, the variances in each subband are ordered in decreasing magnitude. The coefficients with the variance above the threshold T_h are considered significant, thus having children and requiring no map information. Similarly, the coefficients with the variance below the threshold T_l are considered insignificant having no children and also requiring no map information. The map information is required only for the nodes with the variance between T_h and T_l . The values of T_h and T_l are chosen so that the corresponding total Lagrangian cost is minimized and the optimal values are sent as side information for each subband.

Finally, in the last phase (c), the previous optimization process (the phases (a) and (b)) is run exhaustively for each value of the quantization step size q from the list $\{q : q = 7.5 + 0.1 \cdot k, k = 1, 2, \dots, 245\}$. The value that minimizes the Lagrangian cost is optimal and applied to all retained wavelet coefficients. Similarly, the coefficients from the root of the multi-scale trees (the scaling coefficients) are quantized using another quantization step size q_{LP} that is also optimized exhaustively using the values from the same list. The quantized coefficients are entropy encoded using an adaptive entropy coder of Witten et al. (1987).

Notice that the probability density function of the LP scaling coefficients is significantly different when compared to the one of the wavelet coefficients (which is peaked at zero). Thus, independent optimization of the LP quantization step size is reasonable. In contrast, the wavelet coefficients at different scales have different sizes of the peak at zero (Antonini et al., 1992). However, after the thresholding imposed by tree pruning (where many coefficients with small magnitudes are discarded), the peaks are smoothed and the probability density functions at different scales are more similar. For that reason, only one quantization step size is used for all retained wavelet coefficients.

5.2. Spatial Segmentation

As explained in Sections 1 and 4, the requirement for basis functions to capture efficiently directional information in images is to have DVMs along more than only the standard (horizontal and vertical) directions. Since directionlets can maximally have 2 DVMs, image is segmented into smaller segments with one or a few dominant directions.

Even though there exist many ways to segment image, we choose the iterative quad-tree segmentation (Leonardi and Kunt, 1985, Strobach, 1991, Sullivan and Baker, 1994, Shukla et al., 2005) as the simplest method in the sense of encoding efficiency (only one bit per quad-tree node is required to transmit the information on the segmentation process). Notice that we are aware of a recent analysis of other more general segmentation methods (like the one by Huang et al., 2006), but the implementation of these methods in our image compression is left for future work.

Therefore, we apply the S-AWT independently to each segment in an image. The two transform directions per segment are chosen separately to match the two local and most dominant directions. This choice is based on Lagrangian R-D optimization and is explained in more details in the sequel.

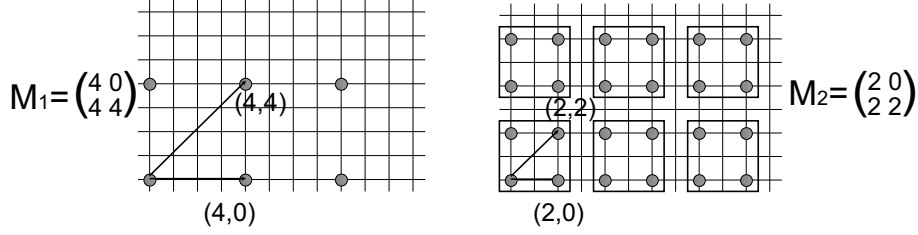


Figure 18: The multiscale grouping of wavelet coefficients. Children are grouped in squares 2×2 and joint to the corresponding parent. The shape of the children groups is not affected by the transform directions. The example shows grouping in the case of the transform directions determined by the vectors $(1, 0)$ and $(1, 1)$ at the two consequent scales. In the left-hand figure, the coefficients are obtained after two filtering and subsampling steps resulting in the generator matrix M_1 . In the right-hand figure, the children-coefficients are obtained after only one transform step and the corresponding generator matrix is M_2 .

5.3. Compression Algorithm

The implementation of the S-AWT is described in Section 3. Here, we present several details of the implementation of our transform and, then, we explain our compression algorithm.

Even though the original construction of the S-AWT allows for asymmetry, we do not make use of it here. The main reason for this is the quad-tree spatial segmentation, which is *symmetric*. As we noticed in the experiments and contrasting with the case analyzed in Section 4 and by Velisavljević et al. (2006) (where an asymmetric segmentation was used in the analysis of asymptotic behavior in NLA), an *asymmetric* transform applied to *symmetric* segments results in a less efficient representation. Furthermore, an asymmetric transform requires more bits to specify the choice of the transform directions when compared to the corresponding symmetric transform. More precisely, because of asymmetry of the transform directions, one more bit per transform is needed to specify which transform direction (out of two) is enforced in the asymmetric construction. For those reasons, we restrict the S-AWT only to symmetric realizations. However, notice that we still allow for different directions in the construction.

As mentioned earlier, the construction of the S-AWT is based on integer lattices. If the corresponding lattice partitions the cubic space \mathbb{Z}^2 into more cosets, then the iterated processing is applied in all cosets separately. However, since some neighbor pixels can be classified in different cosets, the separate filtering does not exploit efficiently the correlation among those pixels, thus, leading to less sparse representation in the transform domain. For that reason, we use only integer lattices that generate one coset, that is, lattices Λ such that $|\det(M_\Lambda)| = 1$. Even though this constraint seems to be severe at a first glance, this still allows for enough combinations of transform directions, as we show next.

In the practical implementation of the compression algorithm, we use 4 transform directions: 0° , 90° , 45° and -45° . These directions are represented by the following vectors: $(1, 0)$, $(0, 1)$, $(1, 1)$ and $(-1, 1)$, respectively. Notice that these four directions allow for six different linearly independent pairs of transform directions. Here, by linear independence of a pair of directions, we mean that the two corresponding vectors are linearly independent. However, the combination of 45° and -45° leads to two cosets in the lattice and is therefore discarded. Thus, the other five combinations are used in the optimization process.

The optimal number of transform levels depends on the size of the compressed images and segments, their content and complexity. However, for the various images we use in this paper, the best number of levels is always 5. In the filtering process, the 1-D biorthogonal "9-7" filter bank is used because of the good visual quality of the reconstructed images. To prevent a strong border effect, the symmetric extension is applied when the filters are convolved with the pixels that are surrounding the original support of the image and also the segments. Since the "9-7" filter bank consists of linear-phase filters, the symmetric extension preserves critical sampling in the transform domain.

Recall that the S-AWT provides a similar multi-resolution analysis as the standard 2-D WT. Thus, directionlets can also be grouped in multi-scale hierarchical trees, where all coefficients (except the ones in the finest scale) have children - the coefficients in the next finer scale at the same spatial location (revisit Figure 16). The multi-scale grouping of directionlets allows for more efficient capturing of the preserved correlation among the coefficients across scales. Notice that, even though the transform can be

Algorithm 1 The full adaptive SFQ compression algorithm using directionlets.

Step 0: Set $S_l \leftarrow 0$,

Step 1: If $S_l < S_{l,max}$, then:

- Apply quad-tree segmentation in the original domain,
- For each of the 4 segments go recursively to *Step 1* with $S_l \leftarrow S_l + 1$,

Step 2: For each pair of transform directions from the list $\{(0^\circ, 90^\circ), (0^\circ, 45^\circ), (0^\circ, -45^\circ), (90^\circ, 45^\circ), (90^\circ, -45^\circ)\}$:

- Apply S-AWT to each segment using symmetric construction and build the hierarchical trees,
- Quantize the LP coefficients using all values $q_{LP} \in \{5.0 + 0.5 \cdot k, k = 1, 2, \dots, 245\}$ and find the one that minimizes the Lagrangian cost,
- For each $q_{HP} \in \{5.0 + 0.5 \cdot k, k = 1, 2, \dots, 245\}$:
 - Prune the trees so that the resulting tree minimizes the Lagrangian cost (phase (a) from Section 5.1),
 - For each subband from coarser to finer scales (phase (b) from Section 5.1):
 - Order coefficients in descending order of variances computed as the sum of squares in the 3×3 neighborhood,
 - Optimize T_h and T_l ,
 - Compute and record the resulting Lagrangian cost,
- Choose the best q_{HP} that minimizes the Lagrangian cost,

Step 3: Choose the best pair of transform directions that minimizes the Lagrangian cost,

Step 4: If $S_l < S_{l,max}$, then:

- If the sum of the Lagrangian cost of the current segment and the cost of the side bits is smaller than the sum of the Lagrangian costs of its children-segments and the costs of their side bits, then keep only the current segment and discard the children-segments,
- Otherwise, keep its children-segments and set the Lagrangian cost of the current segment to be the sum of the Lagrangian costs of the children-segments and the costs of their side bits,

Step 5: Encode the quantized coefficients and map information for each segment using an adaptive arithmetic coder.

applied along non-standard directions, the corresponding sampling structure is always rectangular since the equivalent sublattices Λ can be described by the generator matrices $\mathbf{M}_\Lambda = n \cdot \mathbf{I}$, where \mathbf{I} is the identity matrix. For that reason, the children are grouped in squares of the size 2×2 with the sides aligned in the horizontal and vertical directions. An example of parent-children grouping is shown in Figure 18 for the transform directions determined by the vectors $(1, 0)$ and $(1, 1)$.

As mentioned earlier, the list of allowed quantization step sizes q in the original SFQ algorithm is restricted to the values $\{7.5 + 0.1 \cdot k, k = 1, 2, \dots, 245\}$. Since we also address the low bit-rate compression, we have to adapt this list by allowing coarser quantization (that is, larger values for q). The list of step sizes that we consider is optimized ad-hoc and given by $\{5.0 + 0.5 \cdot k, k = 1, 2, \dots, 245\}$.

The compression algorithm consists of several embedded optimization phases. First, spatial segmentation is applied on the entire image in the original domain and, then, S-AWT is applied on each segment separately using all allowed combinations of transform directions. For each segment and combination of transform direction, the optimal encoding is found following the same philosophy as in the optimization phases proposed by Xiong et al. (1997) and referred to in Section 5.1 as phases (a) and (b), adapting it to our additional freedom of directions. In Algorithm 1, we present the full compression method without repeating these phases in detail. We focus only on the optimization phases that handle the spatial segmentation and choice of directions.

The maximal depth of spatial segmentation, $S_{l,max}$, is chosen *a priori*. In our experiments, we

set $S_{l,max} = 3$. Notice that the jump in *Step 1* is not a loop, but a recursive call, where a newly generated smaller segment is forwarded as an argument for each call. The optimal choices of the spatial segmentation, transform directions for each segment and the quantization step sizes are encoded as side information that is added to the output bit stream. The cost of these side information bits is added to the total Lagrangian cost of encoding segments and is used when the optimal segmentation is calculated.

Notice also that the standard SFQ can be considered as a special case of the proposed algorithm for $S_{l,max} = 0$ and the list of the allowed transform directions reduced to a single entry $(0^\circ, 90^\circ)$.

5.4. Deblocking

In Section 4, we explained the need of spatial segmentation of images and independent adaptation of transform directions in each segment. However, since the segments are transformed separately, the resulting reconstruction may be affected by a blocking effect, which is visible as sharp artificial edges along the segment boundaries. This effect is especially severe in the case of compression at low bit-rates because the differences between the reconstructions of neighbor segments are larger.

The same issue appeared in the original JPEG standard in the 90's and it was analyzed by Pennebaker and Mitchell (1992). Since then, there have been many successful deblocking algorithms, like the ones by Gopinath et al. (1994) or Xiong et al. (1997). Here, we use the second deblocking algorithm proposed by Xiong et al. (1997), which is based on the overcomplete wavelet representation of Mallat and Zhong (1992). In this method, the wavelet coefficients produced by two levels of the three-band (one LP band, horizontal and vertical HP bands) non-subsampled WT are thresholded, whereas the corresponding scaling coefficients are smoothed. Then, the corresponding inverse transform is applied to the processed scaling and wavelet coefficients.

The original deblocking algorithm of Xiong et al. (1997) was proposed for the JPEG standard, where the size of segments was small and the tiling was fine resulting in a significant improvement in both the corresponding MSE and the visual quality of the reconstructed images. In our method, the segments are larger than in the case of the JPEG standard and, for that reason, the number of pixels affected by the blocking effect is smaller. Thus, the impact of the deblocking algorithm on the MSE is negligible (the improvement obtained in our experiments is less than 0.05dB). However, the visual quality of the deblocked images is improved significantly, as will be shown in Section 5.6.

5.5. Computational Complexity

Let us first calculate the order of computational complexity of the S-AWT (expressed in terms of the total number of multiplications and additions) and compare it to the computational complexity of other previously proposed directional methods (this analysis has also been presented by Velisavljević et al., 2006). Then, we compare the computational complexity of our compression method to the complexity of the standard SFQ.

Assuming that the subsampling operations do not carry any computational cost, each 1-D filtering operation is performed in $O[L \cdot N]$ multiplications and $O[L \cdot N]$ additions, where N is the number of input samples and L is the length of the applied filter. Then, in one iteration block of the S-AWT, there are

$$O \left[\sum_{j=0}^{n-1} 2^j \cdot LN^2 / 2^j \right] = O [n \cdot LN^2]$$

multiplications and additions. Furthermore, assume that the S-AWT consists of J iterations. Then, the total number of operations is given by

$$O \left[2 \cdot \sum_{j=0}^{J-1} nLN^2 \cdot 2^{-jn} \right] = O \left[2 \cdot \frac{2^n}{2^n - 1} nLN^2 (1 - 2^{-Jn}) \right], \quad (8)$$

which results in $O[LN^2]$.

Notice that the S-AWT can also be implemented using the lifting scheme proposed by Daubechies and Sweldens (1998), in which case the total number of operations is reduced. However, this decrease of computational complexity affects only the constant, whereas the order remains the same.

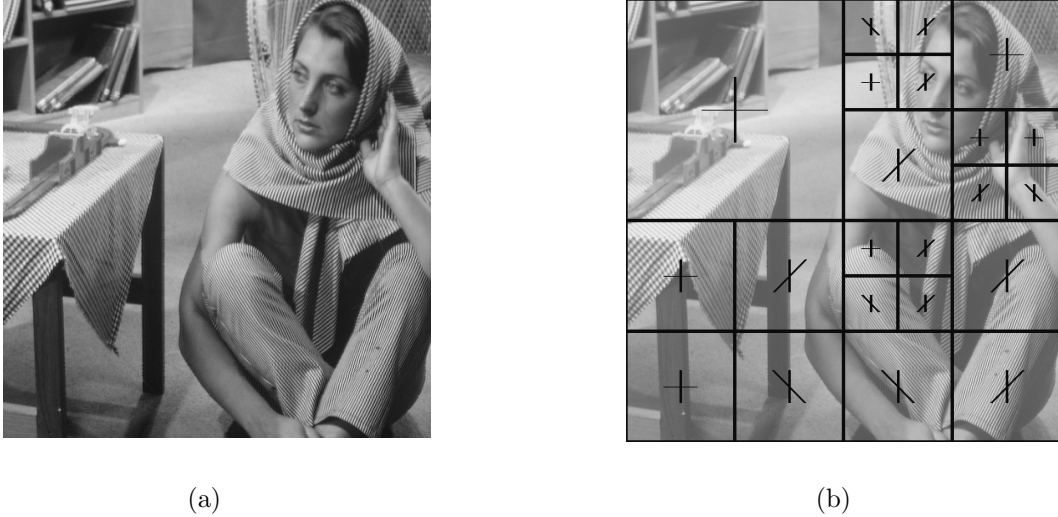


Figure 19: The *Barbara* image. (a) The original image. (b) The optimal segmentation and the optimal choice of transform directions in each segment. This solution is obtained as a result of the optimization process for the compression at the target bit-rate 0.12bpp.

The computational complexity of the S-AWT is substantially lower than the complexity of the other transforms. More precisely, bandelets (LePennec and Mallat, 2000, 2003) require $O[N^2(\log_2 N)^2]$ operations. The implementation of wedgeprints (Wakin et al., 2006) requires to build a large dictionary of linear edges and, thus, the complexity of processing grows rapidly with the size of the image, as $O[N^2 \log_2 N]$. Furthermore, contourlets (Do and Vetterli, 2005) have a complexity of the order $O[L_1 L_2 N^2]$, where the implemented filters are purely 2-D and have the size $L_1 \times L_2$. Thus, the number of operations is also higher than in the case of directionlets in the sense that $L_1 \cdot L_2 \approx L^2 \gg L$.

The complexity of our compression method is increased only by a constant factor as compared to the complexity of the standard SFQ method. The increase is generated by both the deblocking algorithm and the two steps in the optimization algorithm, that is, (a) optimization over spatial segmentation and (b) optimization over directions.

The deblocking algorithm carries more multiplication and addition operations because of the implemented forward and inverse undecimated WT. Assuming that thresholding and smoothing the oversampled transform coefficients do not carry any additional computational cost, the number of multiplications and additions is given by $12LN^2$ (2 decomposition levels with 3 undecimated subbands at each level in both the forward and inverse transforms), where the length of the filters L is smaller than in the case of the S-AWT. Thus, the order of operations remains the same and is given by $O[N^2]$.

The two optimization phases contribute to the total complexity in the two multiplicative constants. The optimization over spatial segmentation increases the complexity $(S_{l,max} + 1)$ times, whereas the optimization over directions contributes with the constant equal to the total number of allowed pairs of transform directions. Notice that these constants have small values in our experiments and do not depend on the image size.

Therefore, the total complexity of the compression method is increased only up to a constant, which does not depend on the image size, when compared to the complexity of the standard SFQ. Proposition 9 formalizes these conclusions.

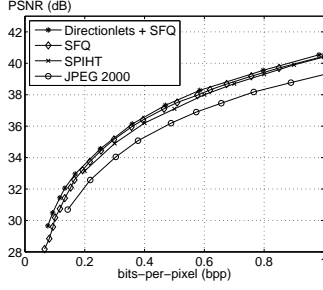
Proposition 9. *Given an $N \times N$ image,*

(a) *The number of operations required by the S-AWT with n transform steps in each transform level is of the order $O[LN^2]$, where L is the length of the filter used in the transform.*

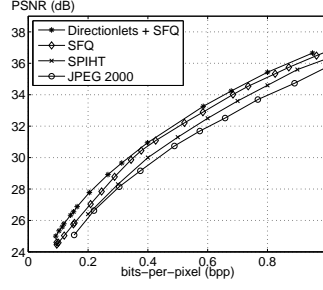
(b) *The number of operations required by the SFQ compression method using directionlets is given by $O[N^2]$.*

5.6. Results

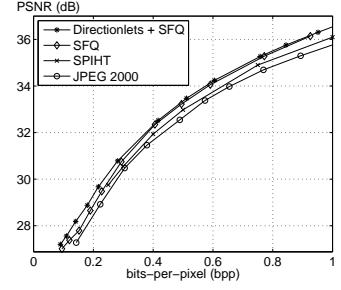
As mentioned in Section 5.2, spatial segmentation allows for local adaptation of transform directions to dominant directions in images. The choice of optimal segmentation and optimal transform directions



(a) Lena

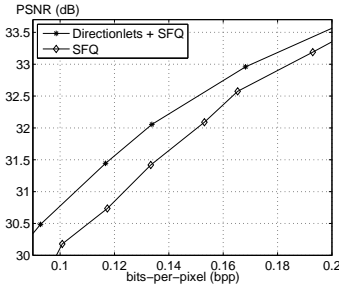


(b) Barbara

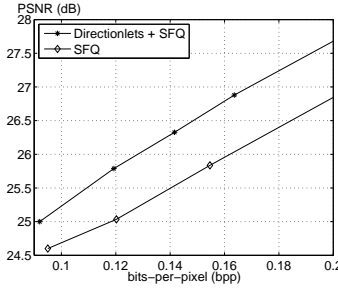


(c) Boat

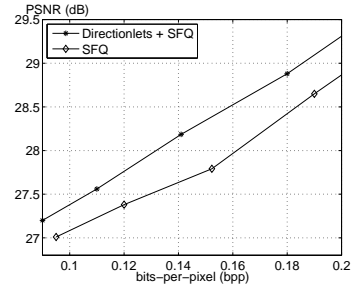
Figure 20: The results of compression expressed in terms of PSNR using four methods: JPEG-2000, SPIHT (Said and Pearlman, 1996), the standard SFQ (Xiong et al., 1997) and the SFQ combined with directionlets. The methods are applied to three test images: (a) Lena, (b) Barbara and (c) Boat. Our method outperforms the standard SFQ and the other two methods in all these examples.



(a) Lena



(b) Barbara



(c) Boat

Figure 21: Our compression method outperforms significantly the standard SFQ at low bit-rates (below 0.2bpp). The comparison of the results is shown for the three test images: (a) Lena, (b) Barbara and (c) Boat.

in the R-D sense depend on the target bit-rate and the Lagrangian multiplier. An example of the optimal solution in case of NLA is shown in Figure 14(b) for the image Lena compressed at 0.051bpp. In Figure 19(b), we show a similar example for the image Barbara at the operational bit-rate 0.12bpp. Notice that the transform directions are well adapted to the dominant directions in segments of the images in all these examples.

Both the standard SFQ and our adaptive method are applied to the images Lena, Barbara and Boat. The comparison of the numerical quality in terms of PSNR of the compressed images is shown in Figure 20. Notice that our method outperforms the standard SFQ as well as the other standard methods, like SPIHT (Said and Pearlman, 1996) and JPEG-2000. This gain is especially noticeable at low operational bit-rates. The results for the low bit-rate interval (below 0.2bpp) are magnified and shown in Figure 21 for compression of the same test images.

Recall that the standard SFQ (and also our method) does not produce an embedded bitstream and, thus, it is not bit-rate scalable, unlike SPIHT or JPEG-2000. The optimal encoding bitstream is found for each preselected bit-rate. However, in spite of this lack of rate-scalability, the standard SFQ, as well as our method, outperforms the two standard methods.

Notice also that, as mentioned in Section 5.4, spatial segmentation leads to a perceptible and annoying blocking effect in the reconstructed images, especially for low bit-rates. The proposed deblocking method successfully removes this effect. An example of compression of the image Barbara at 0.12bpp without and with deblocking is shown in Figure 22. Even though the deblocking method does not improve significantly the numerical result (only 0.04dB in the case shown in Figure 22), it leads to a less disturbing visual distortion in the compressed images.

Therefore, the deblocked reconstructions are better than the images obtained by the standard SFQ not only in terms of the numerical measurement, but also in terms of the visual quality. The remaining artifacts are oriented along dominant directions and are thus less annoying than in the case when the



Figure 22: Deblocking after compression. (a) Blocking effect is severe at very low bit-rates. The coefficients in neighbor segments are quantized and encoded separately and artificial sharp edges are noticeable at the boundaries of segments. This reconstruction of the image Barbara is obtained at 0.12bpp for the optimal segmentation shown in Figure 19 and PSNR=25.29dB. (b) The deblocking algorithm explained in Section 5.4 successfully removes the blocking effect. Even though the PSNR is improved by only 0.04dB, the edges between segments are less visually perceptible.

standard WT is used. The corresponding comparison for the test images Lena, Barbara and Boat are shown in Figures 23, 24 and 25, respectively, at the bit-rates 0.10bpp and 0.15bpp.

The amount of side information required to encode the optimal spatial segmentation, transform directions and quantizer step sizes is negligible. To show that, we analyze each particular requirement.

In the case of quad-tree spatial segmentation, one bit is sent to transmit the binary decision on further segmentation for each segment-node in the quad-tree, except the ones at the maximally allowed segmentation level. The total number of bits depends on the optimal solution, but it is upper bounded by $\sum_{m=0}^{S_{l,max}-1} 4^m$. For the examples shown above ($S_{l,max} = 3$), the upper bound is 21 bits.

Transform directions are encoded only for terminal segment-nodes in the quad-tree. Notice that the number of the terminal segments is upper bounded by $4^{S_{l,max}}$. Therefore, the number of required bits is upper bounded by $4^{S_{l,max}} \cdot \log_2 K$, where K is the number of possible combinations of transform directions. For the presented examples, $K = 5$ and $S_{l,max} = 3$ and, thus, the upper bound equals 149 bits.

Finally, two quantizer step sizes (for the LP and HP subbands) are chosen from a set of Q values. Therefore, the number of encoding bits is given by $\lceil 2 \cdot \log_2 Q \rceil$ and, for the given examples, it is equal to 16 bits.

Thus, the side information is encoded using maximally 186 bits, which is still negligible as compared to the number of encoding bits used in *Step 2* of Algorithm 1.

6. Directional Image Interpolation

The problem of resolution enhancement of images commonly refers to generating missing image pixels at high-resolution (HR) from the available low-resolution (LR) image information. This task is often required in magnification, which is an essential part of software zooming, focusing regions of interest, resolution conversion (e.g. in printer drivers), etc. The main challenge is to preserve sharpness of images after resolution enhancement.

The traditional magnification approaches based on bicubic or spline interpolation (Thévenaz et al., 2000) are used because of fast computation, easy implementation and no *a priori* knowledge assumption. However, these methods generate *blurred* HR images from their LR counterparts. The goal in this paper is to propose a method that reduces this blurring effect at HR.

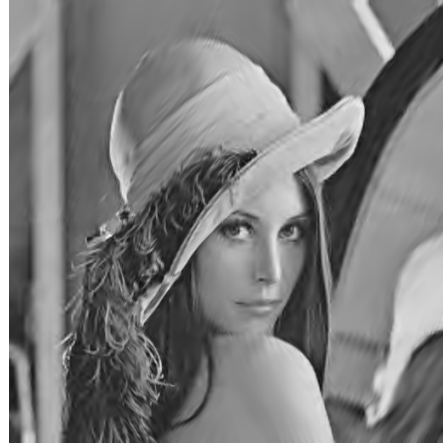
Several recent methods improve the visual quality of the interpolated images by exploiting the corre-



(a)



(b)



(c)



(d)



(e)

Figure 23: Compression of Lena. (a) The original image. (b,c) The image is compressed using the adaptive SFQ with directionlets at the bit-rates 0.10bpp (with the compression ratio 1 : 80) and 0.15bpp (1 : 53), respectively. The numerical quality of the reconstructions is 30.92dB and 32.56dB. (d,e) The image is compressed using the standard SFQ at the same bit-rates. The quality is lower and is equal to 30.17dB and 32.09dB, respectively. The visual quality is also improved because the artifacts are oriented along locally dominant directions and are, thus, less annoying.



(a)



(b)



(c)



(d)



(e)

Figure 24: Compression of Barbara. (a) The original image. (b,c) The image is compressed using the adaptive SFQ with directionlets at the bit-rates 0.10bpp and 0.15bpp, respectively. The numerical quality of the reconstructions is 25.34dB and 26.55dB. (d,e) The image is compressed using the standard SFQ at the same bit-rates. The quality is lower and is equal to 24.58dB and 25.75dB, respectively.



(a)



(b)



(c)



(d)



(e)

Figure 25: Compression of Boat. (a) The original image. (b,c) The image is compressed using the adaptive SFQ with directionlets at the bit-rates 0.10bpp and 0.15bpp, respectively. The numerical quality of the reconstructions is 27.10dB and 28.36dB. (d,e) The image is compressed using the standard SFQ at the same bit-rates. The quality is lower and is equal to 26.16dB and 27.66dB, respectively.

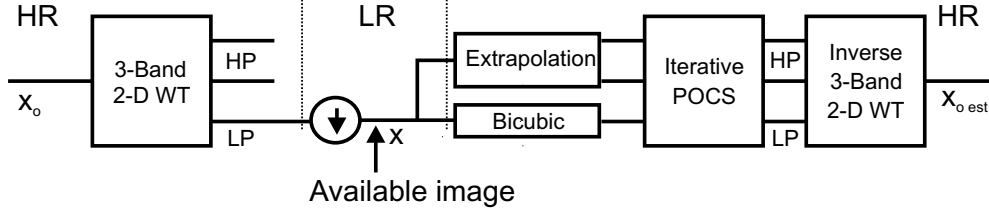


Figure 26: Block diagram of the interpolation algorithm proposed by Chang et al. (2006).

lation among image pixels and modeling it using the Markov random field either in the wavelet (Kinebuchi et al., 2001, Temizel, 2007) or in the pixel domain (Li and Nguyen, 2007). Furthermore, Li and Nguyen (2007) characterize pixels as edge and non-edge and apply different interpolation algorithms to them. Edge-adaptivity and geometric regularity are also exploited by Allebach and Wong (1996) and Li and Orchard (2001). In the latter, the edge direction is extracted from the LR covariance matrices and is used to estimate their HR counterparts. However, the computation of the covariance matrices is limited only to the first four neighbor pixels. As a result, the reconstructed edges in the interpolated HR image are still blurred when compared to the edges in the original image.

Another adaptive interpolation method has been proposed by Chang et al. (2006). This method makes use of the multiscale 2-D WT to capture and characterize edges, which induce peaks in the wavelet subbands. The characterization involves estimation of location, magnitude and evolution of the corresponding peaks across wavelet scales determined by the local Lipschitz regularity of edges (Mallat and Zhong, 1992, Mallat, 1997). This information is used to estimate the corresponding wavelet subbands in the HR multiscale decomposition and to generate the HR image by applying the inverse 2-D WT. The preserved characterization of edges at HR allows for sharpness and a good visual quality of the reconstructed images. However, notice that the implemented WT is a separable transform constructed only along the *horizontal and vertical directions* (Mallat and Zhong, 1992). Thus, it fails to characterize efficiently edges along different directions.

Here, we first review the algorithm of Chang et al. (2006) in more details. Then, we explain how we exploit directionality and build directionally adaptive interpolation method. Finally, we show the experimental results.

6.1. Locally adaptive wavelet-based interpolation

This algorithm is based on the assumption that the LR version is obtained from the HR original image as a LP output of the 3-band 2-D WT (1 LP and 2 HP subbands), which is also used by Mallat and Zhong (1992). The main idea is to estimate the corresponding missing high-resolution LP and two HP subbands from the available LR image so that the inverse 3-band 2-D WT applied to these subbands provides a sharp reconstructed HR image (see Figure 26).

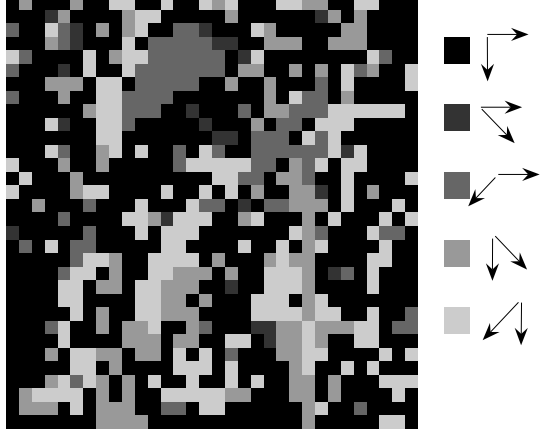
The process of estimation of the 3 wavelet subbands consists of two phases: (a) initial estimate and (b) iterative projections onto convex sets (POCS). In the first phase, the initial estimates of all the 3 subbands at HR are computed. The LP subband is simply obtained by the bicubic interpolation of the LR image. However, since the HP subbands play an important role in obtaining sharp reconstructed image, they are generated using a more sophisticated method. First, a multiscale 3-band 2-D WT is applied to the LR image with 3 levels of decomposition. Then, extrema of the magnitudes of the wavelet coefficients are located in each row and column of the HP subbands to determine the position of sharp variation points (SVP). The extrema of the magnitudes at different scales $j = 1, \dots, J$ related to a single SVP indexed by m follow the scaling relation (Mallat and Zhong, 1992)

$$|W^{(j)} f(x_m)| = K_m 2^{j\alpha_m}, \quad (9)$$

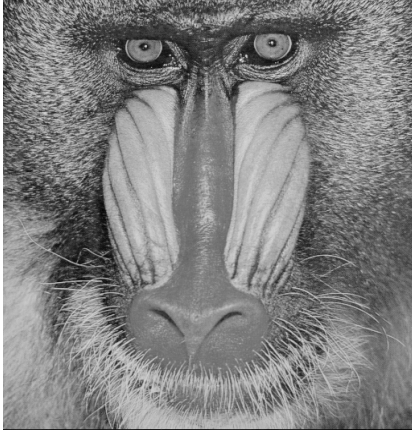
where K_m and α_m are scaling constant and local Lipschitz regularity factor assigned to the m th SVP, respectively. These two parameters are estimated from the determined extrema in the wavelet subbands by linear regression and they are used to extrapolate the corresponding coefficient values in the high-resolution HP subbands. The other HP coefficients that do not correspond to any SVP are filled by a simple linear interpolation along rows and columns.



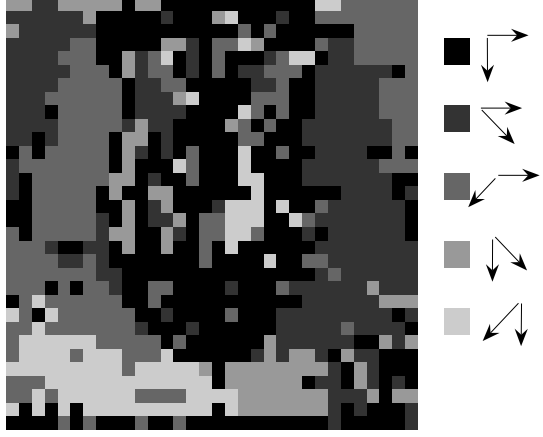
(a)



(b)



(c)



(d)

Figure 27: The transform directions are chosen within each spatial segment of the size 16×16 so that the energy in the HP subbands is minimized allowing for the best matching with locally dominant directions in image. The set of chosen directions form the directional map. (a) The original image Lena. (b) The corresponding directional map. (c) The original image Baboon. (d) The corresponding directional map.

In the second phase, the estimated wavelet subbands are iteratively projected onto 3 convex sets determined by the following properties: (a) the 3 wavelet subbands must belong to the subspace of the wavelet transform, (b) the subsampled LP subband must be consistent with the LR image and (c) the HP subbands must be consistent with the extracted SVP information. The final estimation of the wavelet subbands is transformed back to the original domain using the corresponding inverse 3-band 2-D WT to obtain the interpolated HR image.

This algorithm outperforms other wavelet-based interpolation algorithms. However, despite this success, it fails to preserve sharpness of edges oriented along the directions other than horizontal or vertical. For that reason, we implement directionlets instead of the 3-band WT with adaptation to locally dominant directions across the image domain.

Unlike in the case of compression, where critically sampled transforms are required, an overcomplete version of directionlets is used here. This version is obtained by removing the subsampling operations from the transform construction. Hence, such a transform preserves the same number of coefficients at

each scale equal to the number of pixels in the original domain. Moreover, the transform is shift-invariant and it makes easy to process rotated image rows and columns, which have different lengths.

Similarly to the compression method, the first step of the interpolation algorithm is spatial segmentation and local adaptation of transform directions. In our method, we apply this adaptation by computing directional maps, as explained next.

6.2. Directional map

The image is first divided into spatial segments of size 16×16 pixels (different segment sizes do not influence significantly the final results). Directionlets are then applied in each segment along each pair of directions from the set $\mathcal{D} = \{(0^\circ, 90^\circ), (0^\circ, 45^\circ), (0^\circ, -45^\circ), (90^\circ, 45^\circ), (90^\circ, -45^\circ)\}$ using the biorthogonal "9-7" 1-D filter bank. Notice that the pairs of directionlets in the set \mathcal{D} are chosen so that the cubic lattice is not divided into more cosets, as explained earlier and by Velisavljević et al. (2007) in detail. To avoid a blocking effect in the transform caused by many small segments, the pixels from the neighbor segments are used for filtering across the segment borders.

The best pair of directions $d_n^* \in \mathcal{D}$ is chosen for each segment indexed by n as

$$d_n^* = \arg \min_{d \in \mathcal{D}} \sum_i |W_{n,i}^{(d)}|^2, \quad (10)$$

where the wavelet coefficients $W_{n,i}^{(d)}$ are produced by applying directionlets to the n th segment along the pair d of directions. The directional map determined by the sequence $\{d_n^*\}$ minimizes the energy in the HP subbands and provides the best matching between transform and locally dominant directions across segments. For the reason of simplicity of implementation, the pair $(0^\circ, 90^\circ)$ is assigned by default to smooth segments with no apparent dominant direction (i.e. with low variation of the energy in the HP subbands for $d \in \mathcal{D}$). Two examples of directional map are shown in Figure 27 for the images Lena and Baboon.

The concept of directional map is used in the new interpolation algorithm to improve the extraction of edge information and the estimation of the HR wavelet subbands, as presented next.

6.3. Interpolation algorithm

The proposed novel interpolation algorithm uses the same concept as the previous method by Chang et al. (2006) (revisited also in Section 6.1) with several modifications related to the implementation of directionlets instead of the 3-band 2-D WT. Similarly, the goal is, first, to estimate the corresponding wavelet subbands at HR and, then, to apply the inverse transform to obtain a reconstructed HR image.

The estimation of the wavelet subbands also consists of two phases: (a) initial estimate and (b) iterative POCS. In the initial estimate, the LP subband is bicubic-interpolated from the LR image, whereas the HP subbands are generated from the extracted SVP information. However, as opposed to the 3-band 2-D WT, directionlets produce three HP subbands per scale denoted as HL, LH and HH according to the order of the LP and HP filtering in the two transform steps. In the case of the subbands HL and LH, the search for SVP and the extraction of the SVP parameters are performed along the first and second transform directions, respectively (instead of the horizontal and vertical directions in the previous method), whereas, in case of the subband HH, this process is applied along any of the two directions. Owing to the properties of the applied transform, the extrema of the magnitudes of the directionlets coefficients $|W_s^{(j)} f(x_m)|$ at scales $j = 1, \dots, J$, for $s \in \{\text{HL}, \text{LH}, \text{HH}\}$, follow the scaling relation (Mallat, 1997)

$$|W_s^{(j)} f(x_m)| = K_m 2^{j(\alpha_m+1)}, \text{ for } s \in \{\text{HL}, \text{LH}\}, \quad (11)$$

$$|W_s^{(j)} f(x_m)| = K_m 2^{j(2\alpha_m+1)}, \text{ for } s = \text{HH}. \quad (12)$$

By contrast to Chang et al. (2006), the SVP parameters (that is, the scaling constant K_m and local Lipschitz regularity factor α_m) are estimated in all the three HP subbands by linear regression using Eq. (11) and (12), instead of Eq. (9).

Algorithm 2 Interpolation algorithm based on directionlets.

Step 1: Directional map

- Apply directionlets to each 16×16 block using the pairs of transform directions from the set \mathcal{D} and compute the optimal pair of directions using Eq. (10),

Step 2: Initial estimate

- Compute the LP subband at high-resolution using bicubic interpolation,
- Determine the SVP in the HP subbands using the transform along the directions computed in *Step 1*; estimate the SVP parameters; compute the corresponding HP subbands,

Step 3: Iterative POCS (repeat this step K times)

- Project all the subbands onto the directionlets subspace using a pair of inverse and forward transform,
- Keep the subsampled version of the LP subband consistent with the low-resolution image so that the coefficients at even locations are equal to the original low-resolution pixels,
- Keep the SVP parameters in the HP subbands consistent with the initial edge estimation,

Step 4: Reconstruction

- Apply one step of inverse directionlets on the estimated subbands using the directional map computed in *Step 1*.
-

The initially estimated HR subbands are iteratively refined in the second phase by projection onto three convex sets. The sets are defined by similar properties as in the original algorithm, with a modification for the first set that the subbands must belong to the corresponding subspace of directionlets, instead of the 3-band WT.

Notice that the two SVP parameters that correspond to the same location estimated in different HP subbands are correlated, since they are produced by the same SVP. This correlation can be exploited to further improve the estimation of the high-resolution HP subbands. However, this issue will be addressed in future work.

The estimated HR subbands are transformed back to the original domain using inverse directionlets and the computed directional map. Notice also that the same transform is used in both the computation of directional map (as explained in Section 6.2) and the initial estimate of the HP subbands and, thus, this transform can be applied only once. This fact is exploited to reduce the overall computational complexity to the same order as the complexity of the initial interpolation algorithm. The entire interpolation algorithm is summarized in Algorithm 2 (see also Velisavljević, 2008).

Notice that the value of K in *Step 3* in the experiments is $K = 5$.

6.4. Results

The performance of the new method is compared to the performance of both the bicubic interpolation and the previous locally adaptive wavelet-based method proposed by Chang et al. (2006) applied to three test images: Lena, Baboon and Kodak star chart. To compare the interpolated images to the reference ones in terms of peak signal-to-noise-ratio (PSNR), the original HR images are first LP filtered and subsampled to obtain the LR versions and, then, the resulting images are interpolated back to HR. Furthermore, to emphasize the difference in the visual quality obtained using different interpolation methods, the images are interpolated twice, that is, the resolution at HR is 4 times larger than that at LR.

The LR versions of the images Lena and Baboon have 128×128 pixels, whereas the LR version of the image Kodak star chart has 256×256 pixels. The target HR versions have 512×512 and 1024×1024 pixels, respectively. The images are interpolated using three methods: the bicubic interpolation, the wavelet-based interpolation and the method based on directionlets and the resulting PSNR is shown in Table 4. Notice that the improvement in the numeric quality of interpolation induced by directionlets



(a)



(b)



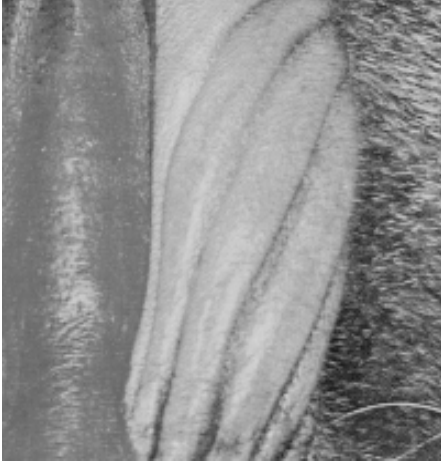
(c)



(d)

Figure 28: The image Lena is interpolated using three methods and a magnified detail is shown. (a) The original HR image, (b) bicubic interpolation, (c) locally adaptive wavelet-based interpolation and (d) directionally adaptive interpolation based on directionlets. The novel method based on directionlets outperforms the previous ones by providing both higher numerical and visual quality and sharper edges in the interpolated images, especially when locally dominant directions are neither horizontal nor vertical.

is significant in case of the images Lena and Kodak star chart, whereas it is rather weak in case of the image Baboon because of a complicated texture. However, the visual quality of the interpolated images is strongly enhanced in all cases as compared to the results of the standard wavelet-based method. To highlight this result, magnified details of all the 3 interpolated images are shown in Figures 28, 29 and 30. The edges are consistently sharper in the images obtained using the novel method, especially in the case when the dominant orientation is neither horizontal nor vertical.



(a)



(b)



(c)

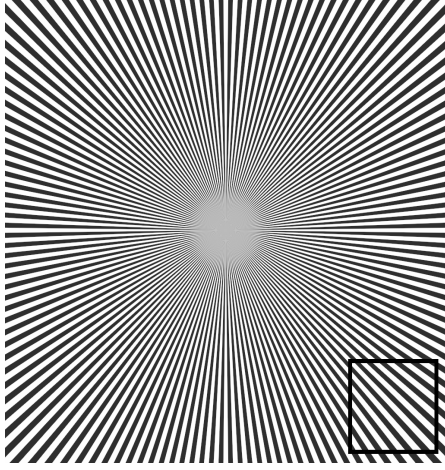


(d)

Figure 29: The image Baboon is interpolated from 128×128 to 512×512 pixels using 3 methods. A magnified area of the image is shown. (a) A detail of the original image shown in Figure 27(c). (b) Bicubic interpolation. (c) Wavelet-based interpolation. (d) Interpolation with directionlets. Notice that the visual quality of the image interpolated using the novel method is improved as compared to the standard one, even though the numeric improvement is negligible (as presented in Table 4).

7. Conclusions

Sparse image representation of images is required in many areas of image processing. The wavelet transform has achieved a great success because it is capable of providing sparsity in case of smooth regions in images. However, it fails to capture efficiently discontinuities characterized by geometric regularity in different directions. We proposed a transform with asymmetric oriented basis functions (called directionlets) that imposes directional vanishing moments along different directions. This property allows for a sparser representation of the oriented features in images than the representation obtained by the standard wavelet transform. At the same time, directionlets retain separability, conceptual and computational simplicity and critical sampling. Furthermore, using directionally adaptive schemes, critically



(a)



(b)



(c)



(d)

Figure 30: The image Kodak star chart is interpolated from 256×256 to 1024×1024 pixels using 3 methods. (a) The entire original image with a marked area that is magnified. This area is shown in the interpolated images using (b) bicubic interpolation, (c) wavelet-based interpolation and (d) the interpolation with directionlets. The diagonal lines are reconstructed with the best quality using the novel method.

sampled directionlets are able to improve the performance of non-linear approximation and compression of images, whereas an oversampled shift-invariant version is successfully applied to image interpolation.

Appendix I. Proof of Theorem 1

Recall first that a C^2 curve can be locally represented by the Taylor series expansion, that is, by a quadratic polynomial

$$y(x) = ax^2 + bx + c, \quad (13)$$

Table 4: Results of interpolation of 3 test images using 3 methods: bicubic, wavelet-based and the interpolation based on directionlets.

	Lena	Baboon	Kodak star chart
Bicubic	26.80dB	20.27dB	13.57dB
WT	28.59dB	20.97dB	15.95dB
Directionlets	29.65dB	20.93dB	16.51dB

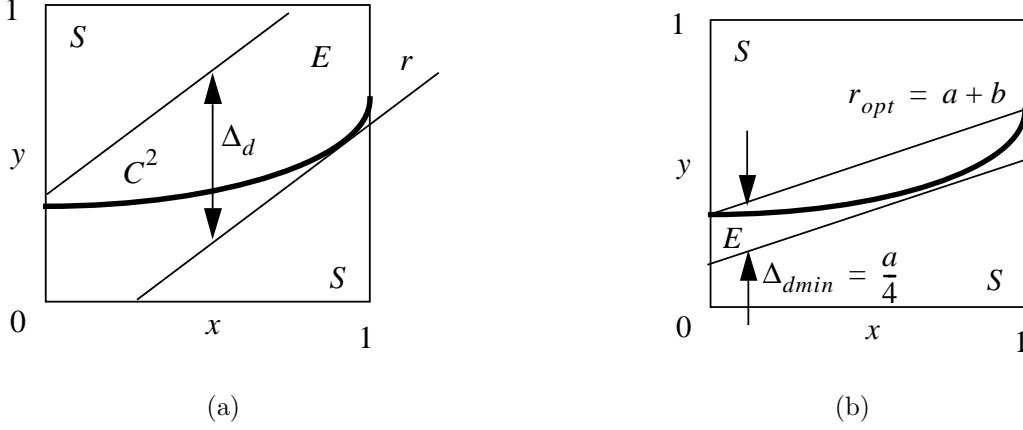


Figure 31: The 2-D function $f(x_1, x_2)$ is C^2 smooth on the unit square away from a C^2 discontinuity curve. The curve can be locally approximated by a quadratic polynomial $y(x) = ax^2 + bx + c$. The E-type transform coefficients intersect the curve and have a slower decay of magnitudes across scales than the S-type coefficients, which correspond to the smooth regions. (a) The S-AWT produces the E-type coefficients within the strip along the slope r . (b) The width of the strip Δ_d is minimized for $r = a + b$.

where a and b are related to the second and first derivative of the curve (curvature and linear component), respectively. Without loss of generality, we assume that the C^2 discontinuity curve is *Horizon* (Donoho, 1999) on the unit square $[0, 1]^2$.

Since the smooth regions of the function $f(x_1, x_2)$ are C^2 , assume that the 1-D filters used in the S-AWT are orthogonal and have at least two vanishing moments. Let the transform be applied along the class of straight lines defined by

$$\{y(x) = rx + d : d \in \mathbb{R}\}. \quad (14)$$

Here, the slope r determines the transform direction, whereas the alignment direction is vertical. Equalizing Eq. (13) and (14) we can write

$$d(x) = ax^2 + (b - r)x + c.$$

The transform coefficients of the S-AWT that intersect the discontinuity curve are called *E-type coefficients*. The number of the E-type coefficients at the scale j is given by $N_e^{(0)}(j) = O[2^{n_2 j} \Delta_d]$. Here, n_2 is the number of transforms applied along the vertical direction, $\Delta_d = \max_{0 \leq x \leq 1} d(x) - \min_{0 \leq x \leq 1} d(x)$ is the width of the strip along the transform direction that contains the curve (see Figure 31), and zero in the superscript of $N_e^{(0)}(j)$ denotes that no segmentation has been applied yet. The transform direction with the slope

$$r = a + b \quad (15)$$

minimizes the width Δ_d (and, thereof, $N_e^{(0)}(j)$) on the unit square. In that case the number of the E-type coefficients is given by

$$N_e^{(0)}(j) = O\left[\frac{a}{4} 2^{n_2 j}\right].$$

Notice that an increment in the scale index j is equivalent to a step to a finer scale.

The transform coefficients of the S-AWT, which do not intersect the discontinuity curve are called *S-type coefficients*. The number of the S-type coefficients depends on the number of transforms n_1 and

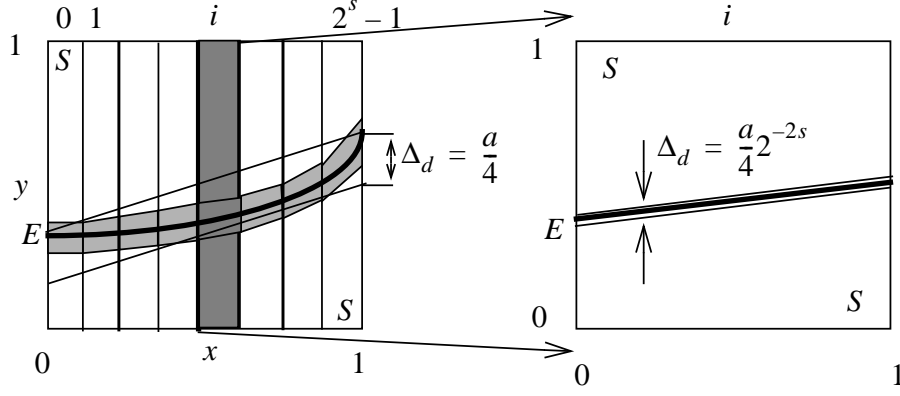


Figure 32: Asymmetric segmentation partitions the unit square into 2^s equally wide vertical strips. After rescaling, the curvature parameter a (related to the second derivative of the C^2 curve) is reduced in each segment by the factor 2^{2s} . Since there are 2^s segments that intersect the discontinuity, the total number of the E-type transform coefficients is reduced by 2^s . At the same time, the total number of the S-type coefficients is increased by the same factor.

n_2 at a scale along the transform and vertical directions, respectively, as

$$N_s^{(0)}(j) = 2^{(n_1+n_2)j} - N_e^{(0)}(j) = O \left[2^{(n_1+n_2)j} - \frac{a}{4} 2^{n_2 j} \right].$$

An asymmetric spatial segmentation is applied on the unit square. It partitions the unit square into vertical strips using the dyadic rule, that is, there are 2^s vertical strips at the s th level of segmentation, where the width of each is 2^{-s} (Figure 32). The optimal transform direction, according to Eq. (15), is chosen for each segment independently. Since each segment is rescaled again to the unit square, the number of the E-type transform coefficients in a segment is reduced and is given by

$$O \left[\frac{a}{4} 2^{n_2 j} \cdot 2^{-2s} \right].$$

The total number of the E-type coefficients is given by the sum across all the segments, that is,

$$N_e(j, s) = \sum_{k=0}^{2^s-1} O \left[\frac{a}{4} 2^{n_2 j-2s} \right] = O \left[\frac{a}{4} 2^{n_2 j-s} \right]. \quad (16)$$

Similarly, the total number of the S-type coefficients is given by

$$N_s(j, s) = \sum_{k=0}^{2^s-1} O \left[2^{(n_1+n_2)j} - \frac{a}{4} 2^{n_2 j-2s} \right] = O \left[2^{(n_1+n_2)j+s} - \frac{a}{4} 2^{n_2 j-s} \right]. \quad (17)$$

Notice that the exact number of the two types of coefficients given by Eq. (16) and (17) depends on the length of the 1-D filters used in the transform. However, the dependence is only up to a constant and, thus, the order of growth of these numbers across scales remains the same.

The magnitudes $|w_e(j)|$ of the E-type coefficients decay across scales as $O[2^{-(n_1+n_2)j/2}]$. The S-type coefficients correspond to the smooth regions of the function $f(x_1, x_2)$ and their magnitudes $|w_s(j)|$ are upper bounded by $O[2^{-n_3 j/2}]$. Notice that, since the 1-D HP filters have vanishing moments, the decay of the magnitudes of the S-type coefficients is faster than the one of the E-type coefficients, that is, $n_3 > n_1 + n_2$.

We estimate n_3 considering that the applied 1-D wavelets have at least two vanishing moments. It is shown by Mallat (1997) that, the decay of the magnitudes $|w_s(j)|$ in a smooth region after two consecutive transforms with alternated transform directions is 2^{-3} . Therefore, the decay rate n_3 is given by

$$n_3 = 6 \cdot \min(n_1, n_2) + |n_2 - n_1| = \begin{cases} n_1 + 5n_2, & n_1 \geq n_2 \\ 5n_1 + n_2, & n_1 \leq n_2 \end{cases}. \quad (18)$$

To approximate the function $f(x_1, x_2)$, we keep all the coefficients with the magnitudes larger than or equal to the threshold 2^{-m} , where $m \geq 0$, and discard (set to zero) the others. The retained coefficients can be divided into two groups:

- (1) The E-type coefficients at the scales $0 \leq j \leq 2m/(n_1 + n_2)$,
- (2) The S-type coefficients at the scales $0 \leq j \leq 2m/n_3$.

From Eq. (16), Eq. (17) and decays of the magnitudes across scales, we compute the order of the total number of retained coefficients $N(m, s)$ and the corresponding MSE. The number $N(m, s)$ is the sum of the retained E and S-type coefficients:

$$\begin{aligned} N(m, s) &= \sum_{j=0}^{2m/(n_1+n_2)} N_e(j, s) + \sum_{j=0}^{2m/n_3} N_s(j, s) \\ &= O \left[2^{\frac{2n_2}{n_1+n_2}m-s} \right] + O \left[2^{\frac{2(n_1+n_2)}{n_3}m+s} \right]. \end{aligned} \quad (19)$$

The MSE is given by

$$\begin{aligned} \text{MSE}(m, s) &= \sum_{j=2m/(n_1+n_2)+1}^{+\infty} N_e(j, s) |w_e(j)|^2 + \sum_{j=2m/n_3+1}^{+\infty} N_s(j, s) |w_s(j)|^2 \\ &= O \left[2^{-\frac{2n_1}{n_1+n_2}m-s} \right] + O \left[2^{-\frac{2(n_3-n_1-n_2)}{n_3}m+s} \right]. \end{aligned} \quad (20)$$

Assuming that the number of segmentation levels depends on the exponent m of the threshold as $s = \eta m$, where the *segmentation rate* $\eta \geq 0$, we distinguish the two cases, as follows:

- (1) The terms in Eq. (19) and (20) produced by the E-type coefficients dominate, in which case we have

$$\eta \leq \eta^* = \frac{n_2}{n_1 + n_2} - \frac{n_1 + n_2}{n_3} = \frac{1}{\rho + 1} - \frac{\rho + 1}{\rho + 5},$$

where $\rho = n_1/n_2 \geq 1$. Then the MSE decays as

$$\text{MSE} = O \left[N^{-e_1} \right], \text{ where } e_1 = \frac{2n_1 + \eta(n_1 + n_2)}{2n_2 - \eta(n_1 + n_2)} = \frac{2\rho + \eta(\rho + 1)}{2 - \eta(\rho + 1)}.$$

- (2) The terms in Eq. (19) and Eq. (20) produced by the S-type coefficients dominate, that is, $\eta \geq \eta^*$ and

$$\text{MSE} = O \left[N^{-e_2} \right], \text{ where } e_2 = \frac{2(n_3 - n_1 - n_2) - \eta n_3}{2(n_1 + n_2) + \eta n_3} = \frac{8 - \eta(\rho + 5)}{2(\rho + 1) + \eta(\rho + 5)}.$$

Plugging Eq. (18) in the relations above and knowing that the segmentation rate η is a non-negative value, we obtain the maximal decay rate $\text{MSE} = O[N^{-\alpha}]$, with $\alpha = (\sqrt{17} - 1)/2 \approx 1.562$. The optimal rate is attained for the asymmetry ratio $\rho^* = n_1/n_2 = \alpha \approx 1.562$ and the segmentation rate $\eta^* = 0$.

Notice that the analysis above is based on two assumptions: (a) the optimal transform direction given by Eq. (15) is chosen and (b) the C^2 curve is globally represented by a quadratic polynomial given by Eq. (13). Here, we address these two assumptions showing that they do not constrain severely the approximation rate.

- (a) Assume that the transform direction is given by the suboptimal slope $r = a + b + \epsilon$, where $|r| \leq 1$. Then it can be shown that $\Delta_d = a/4 + |\epsilon|/2 + \epsilon^2/4a = O[a]$ for $|\epsilon| \leq a$ and $\Delta_d = |\epsilon| + o[\epsilon]$ for $|\epsilon| > a$. Furthermore, assume that ϵ decays exponentially with the number of segmentation steps, that is, $\epsilon \sim 2^{-\beta s}$, where $\beta > 0$. If $\beta < 2$, then the expression of $N_e(j, s)$ given by Eq. (16) becomes $O[2^{n_2 j - (\beta-1)s}]$ and the optimal segmentation rate η^* is multiplied by the factor $2/\beta$. In that case the exponent e_1 is given by

$$e_1 = \frac{2\rho + (\beta - 1)\eta(\rho + 1)}{2 - (\beta - 1)\eta(\rho + 1)},$$

whereas the exponent e_2 is unchanged. However, even though some of these parameters are changed, the optimal approximation rate remains the same, that is, $\text{MSE} = O[N^{-\alpha}]$ if $\rho^* = \alpha$ and $\eta^* = 0$. On the other hand, the required number of transform directions is finite now and behaves as $1/|\epsilon| \sim 2^{\beta s}$.

- (b) The analysis that leads to the approximation rate holds only for the case when the slope of the tangent direction (or, equivalently, the first derivative) of the C^2 curve is in the interval $[-1, 1]$. However, the first derivative of a general C^2 curve is not constrained on that interval and, therefore, the optimal approximation rate cannot be achieved in the same way as in the case of a quadratic polynomial. In order to be able to achieve the same rate we need to introduce an initial number of segmentation steps prior to the iteration. Recall that one step of asymmetric segmentation attenuates twice the first derivative of the C^2 curve because one step of the asymmetric segmentation is equivalent to stretching the abscissa by the factor 2. Thus, it suffices to apply enough segmentation steps so that the maximal magnitude of the first derivative is less than or equal to 1. Then, the iterated segmentation and transform are continued on each of these initial segments and this construction results in the same optimal approximation rate. Notice that the necessity for reducing the magnitude of the first derivative below 1 is caused by the assumption that the C^2 curve is Horizon. However, if this assumption is not satisfied, then an appropriate combination of initial segmentation steps and transposition of the axes can rescale the curve so that each segment of the curve is Horizon. Therefore, the optimal approximation rate can be achieved in the case of a general C^2 curve.

For the compression application, the retained coefficients have to be indexed and quantized. For a given MSE (or *distortion*) each of these operations carries a cost in terms of the required bits.

The N retained S-AWT coefficients within a spatial segment can be organized in an embedded tree-structure, similar to the structures produced by the standard WT and exploited in the other compression algorithms, like zerotrees (Shapiro, 1993), SPIHT (Said and Pearlman, 1996), SFQ (Xiong et al., 1997, 1998). The main difference between the tree-structures of the standard WT and S-AWT is in the number of descendants of each transform coefficient. While this number is fixed in the standard WT, it depends on the number of transform steps applied at each scale in the S-AWT. However, the S-AWT tree-structure allows also for indexing the retained coefficients using approximately 1 bit per transform coefficient.

A variable length coding scheme allocates l bits to encode coefficients with magnitudes in the interval $[2^{-m}2^{l-1}, 2^{-m}2^l]$. Thus, using Eq. (19) and the optimal choice for n_1, n_2, n_3 , and η , the total number of encoding bits R is given by:

$$\begin{aligned} R(m) &= N(m, 0) + \sum_{l=1}^{\infty} N(m-l, 0) \\ &= O[2^{\frac{\alpha}{2}m}] + \sum_{l=1}^{\infty} 2^{\frac{\alpha}{2}(m-l)} = O[2^{\frac{\alpha}{2}m}]. \end{aligned} \quad (21)$$

The distortion D consists of two components: (a) the MSE resulting from the truncation of small coefficients in the approximation given by Eq. (20), and (b) distortion caused by the quantization of the retained coefficients. The second component is given by $N(m, 0) \cdot 2^{-2m}$ and, thus, the total distortion is

$$D(m) = \text{MSE}(m, 0) + N(m, 0) \cdot 2^{-2m} = O\left[2^{-\frac{\alpha^2}{2}m}\right]. \quad (22)$$

The R-D behavior follows from Eq. (21) and Eq. (22) and it is given by

$$D(R) = O[R^{-\alpha}].$$

Appendix II. Lagrangian Optimization

The optimization tool based on Lagrange multipliers is explained in detail by Ramchandran (1993) and Ramchandran and Vetterli (1993). Here, we give a brief overview on the use of the tool in the compression algorithm.

The goal of the optimization process is to minimize the average distortion D of a set of signal blocks subject to a total bit-rate R_{budget} assuming that (a) the operational R-D curve (which consists of the set of operational points in the R-D coordinate system characterizing a compression algorithm) is independent for each signal block and (b) the total bit-rate and distortion is obtained as the sum of the bit-rates and distortions in each signal block. The second assumption holds only in the case of orthogonal transforms applied in each signal block and, thus, this tool cannot be used in the case of overcomplete transforms

(frames) and is only used suboptimally for biorthogonal transforms. However, notice that in the case of some widely used biorthogonal filter banks (like "9-7") the error is negligible and we still use this optimization method.

For the i th signal block, we denote the chosen quantizer as Q_i and the obtained bit-rate and distortion as $R_i(Q_i)$ and $D_i(Q_i)$. This problem is often called resource (bit) allocation, although its applicability is more general (e.g. optimal segmentation, choice of different bases, etc.). The problem of independent resource allocation is stated in the form of a constrained minimization, as follows:

$$\min_{Q_i} \sum_{i=1}^N D_i(Q_i), \text{ such that } \sum_{i=1}^N R_i(Q_i) \leq R_{budget}. \quad (23)$$

As shown by Shoham and Gersho (1988), Ramchandran (1993) and Ramchandran and Vetterli (1993), the constrained optimization problem given by Eq. (23) can be shown to be equivalent to an unconstrained problem given by

$$J_i(\lambda) = \min_{Q_i} \sum_{i=1}^N D_i(Q_i) + \lambda R_i(Q_i). \quad (24)$$

In Eq. (24), the bit-rate R_i and the distortion D_i are incorporated into the Lagrangian cost J_i for a given Lagrange multiplier $\lambda \geq 0$. The multiplier trades off distortion for bit-rate and the above minimization is performed for each value of λ . Notice that, since at R-D optimality all signal blocks must operate at the same slope point λ on their R-D curves (otherwise, one can redistribute bits among signal blocks until the optimum is reached) the minimization of the Lagrangian cost J_i is performed *independently* in each signal block. Thus, the optimization problem in Eq. (24) is unconstrained and can be solved in an algorithmically simpler way than the one in Eq. (23).

Notice also that different values of the Lagrange multiplier λ correspond to different points on the R-D curve. Therefore, λ taking values from 0 to ∞ is equivalent to moving operating point across the whole R-D curve. Since the optimization problems given by Eq. (23) and (24) are equivalent, the solution to the unconstrained problem of Eq. (24) is also the solution to the constrained problem of Eq. (23) for the particular case of $R_{budget} = \sum_{i=1}^N R_i$ (see Shoham and Gersho, 1988, for the formal proof).

Each value of the total bit-rate R_{budget} used in Eq. (23) and (24) corresponds to a different value of the multiplier λ . Given a target R_{budget} , the optimal constant slope λ^* is not known *a priori*. However, even though the search for the corresponding λ^* increases the complexity of the whole optimization algorithm, there exist low computational cost algorithms, such as the bisection algorithm (Press et al., 1988, Shoham and Gersho, 1988, Ramchandran and Vetterli, 1993).

References

- [1] Allebach, J., and Wong, P.W. (1996). Edge-directed interpolation. *Proc. IEEE International Conference on Image Processing (ICIP1996)* **3**. Lausanne, Switzerland, pp. 16-19.
- [2] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Trans. Image Process.* **1**(2), pp. 205-220.
- [3] Bamberger, R.H., and Smith, M.J.T. (1992). A Filter Bank for the Directional Decomposition of Images: Theory and Design. *IEEE Trans. Signal Process.* **40**(4), pp. 882-893.
- [4] Bresenham, J.E. (1965). Algorithm for Computer Control of a Digital Plotter. *IBM Systems Journal* **4**(1), pp. 25-30.
- [5] Candès, E.J., and Donoho, D.L. (1999). Curvelets and Curvilinear Integrals. *Tech. Report*. Department of Statistics, Stanford University.
- [6] Candès, E.J., and Donoho, D.L. (1999). Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges. *Curve and Surface Fitting: Saint-Malo 1999* (A. Cohen, C. Rabut, and L. L. Schumaker, Ed.). Vanderbilt University Press.
- [7] Candès, E.J., and Donoho, D.L. (2002). New tight frames of curvelets and optimal representations of objects with smooth singularities. *Tech. Report*. Department of Statistics, Stanford University.

- [8] Chan, T.S., and Yip, R.K.K. (1996). Line Detection Algorithm. *Proc. IEEE International Conference on Pattern Recognition (ICPR1996)* **2**. Banff, Canada, pp. 126-130.
- [9] Chang, S.G., Cvetković, Z., and Vetterli, M. (2006). Locally adaptive wavelet-based image interpolation. *IEEE Trans. Image Process.* **15**(6), pp. 1471-1485.
- [10] Chang, C.-L., and Girod, B. (2007). Direction-adaptive discrete wavelet transform for image compression. *IEEE Trans. Image Process.* **16**(5), pp. 1289-1302.
- [11] Chen, T., and Vaidyanathan, P.P. (1993). Multidimensional multirate filters and filter banks derived from one-dimensional filters. *IEEE Trans. Signal Process.* **41**(5), pp. 1749-1765.
- [12] Cohen, A., and Daubechies, I. (1993). Non-separable bidimensional wavelet bases. *Revista Matemática Iberoamericana* **9**(1), pp. 51-137.
- [13] Cohen, A., and Matei, B. (2001). Compact representation of images by edge adapted multiscale transforms. *Proc. IEEE International Conference on Image Processing (ICIP2001)* **1**. Thessaloniki, Greece, pp. 8-11.
- [14] Conway, J.H., and Sloane, N.J.A. (1998). *Sphere packings, lattices and groups*. Springer, New York, NY.
- [15] da Cunha, A.L., Zhou, J., and Do, M.N. (2006). The Nonsubsampled Contourlet Transform: Theory, Design, and Applications. *IEEE Trans. Image Process.* **15**(10), pp. 3089-3101.
- [16] Daubechies, I., and Sweldens, W. (1998). Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.* **4**(3), pp. 247-269.
- [17] DeVore, R.A. (1998). Nonlinear approximation. *Acta Numer* **7**, pp. 51-150.
- [18] Ding, W., Wu, F., and Li, S. (2004). Lifting-based wavelet transform with directionally spatial prediction. *Proc. Picture Coding Symposium (PCS2004)*. San Francisco, CA.
- [19] Do, M.N., and Vetterli, M. (2005). The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *IEEE Trans. Image Process.* **14**(12), pp. 2091-2106.
- [20] Donoho, D.L., Vetterli, M., DeVore, R.A., and Daubechies, I. (1998). Data compression and harmonic analysis. *IEEE Trans. Inf. Theory* **44**(6), pp. 2435-2476.
- [21] Donoho, D.L. (1999). Wedgelets: Nearly-minimax Estimation of Edges. *Ann. Statist.* **27**, pp. 859-897.
- [22] Dragotti, P.L., and Vetterli, M. (2001). Footprints and edgeprints for image denoising and compression. *Proc. IEEE International Conference on Image Processing (ICIP2001)* **2**. Thessaloniki, Greece, pp. 237-240.
- [23] Dragotti, P.L., and Vetterli, M. (2003). Wavelet footprints: theory, algorithms, and applications. *IEEE Trans. Signal Process.* **51**(5), pp. 1306-1323.
- [24] Foley, J.D., Dam, A.V., Feiner, S.K., and Hughes, J.F. (1990). *Computer graphics: Principles and Practice*. Adilson-Wesley Publishing Company, Reading, MA.
- [25] Gerek, O.N., and Çetin, A.E. (2006). A 2-D orientation-adaptive prediction filter in lifting structures for image coding. *IEEE Trans. Image Process.* **15**(1), pp. 106-111.
- [26] Gopinath, R.A., Lang, M., Guo, H., and Odegard, J.E. (1994). Wavelet-based post-processing of low bit rate transform coded images. *Proc. IEEE International Conference on Image Processing (ICIP1994)* **2**. Austin, TX, pp. 913-917.
- [27] Granlund, G.H., and Knutsson, H. (1990). Compact associative representation of visual information. *Proc. on 10th International Conference on Pattern Recognition* **2**. Atlantic City, NJ, pp. 200-207.
- [28] Hirakawa, K., and Parks, T.W. (2005). Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans. Image Process.* **14**(3), pp. 360-369.

- [29] Huang, Y., Pollak, I., Do, M.N., and Bouman, C.A. (2006). Fast search for best representations in multitree dictionaries. *IEEE Trans. Image Process.* **15**(7), pp. 1779-1793.
- [30] Kinebuchi, K., Muresan, D.D., and Parks, T.W. (2001). Image interpolation using wavelet based hidden Markov trees. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2001)* **3**. Salt Lake City, UT, pp. 1957-1960.
- [31] Kingsbury, N. (2001). Complex Wavelets for Shift Invariant Analysis and Filtering of Signals. *Journal of Appl. and Comput. Harmonic Analysis* **10**(3), pp. 234-253.
- [32] Kovačević, J. (1991). Filter Banks and Wavelets: Extensions and Applications. *Ph.D. Thesis*. Graduate School of Arts and Sciences, Columbia University, New York, NY.
- [33] Kovačević, J., and Vetterli, M. (1992). Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for \mathbb{R}^n . *IEEE Trans. Inf. Theory* **38**(2), pp. 533-555.
- [34] Leonardi, R., and Kunt, M. (1985). Adaptive split and merge for image analysis and coding. *Proc. SPIE* **594**.
- [35] LePennec, E.L., and Mallat, S. (2000). Image compression with geometric wavelets. *Proc. IEEE International Conference on Image Processing (ICIP2000)* **1**. Vancouver, Canada, pp. 661-664.
- [36] LePennec, E., and Mallat, S. (2003). Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.* **14**(4), pp. 423-438.
- [37] Lewis, A., and Knowles, G. (1992). Image compression using 2-D wavelet transform. *IEEE Trans. Image Process.* **1**(2), pp. 244-250.
- [38] Li, M., and Nguyen, T. (2007). Markov random field model-based edge-directed image interpolation. *Proc. IEEE International Conference on Image Processing (ICIP2007)* **2**. San Antonio, TX, pp. 93-96.
- [39] Li, X., and Orchard, M.T. (2001). New edge-directed interpolation. *IEEE Trans. Image Process.* **10**(10), pp. 1521-1527.
- [40] Mallat, S. (1997). *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA.
- [41] Mallat, S., and Zhong, S. (1992). Characterization of signals from multiscale edges. *IEEE Trans. Pattern Analysis and Machine Intelligence* **14**(7), pp. 710-732.
- [42] Mehrseresht, N., and Taubman, D. (2006). Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression. *Proc. IEEE International Conference on Image Processing (ICIP2006)*. Atlanta, GA, pp. 1593-1596.
- [43] Meyer, F.G., and Coifman, R.R. (1997). Brushlets: a tool for directional image analysis and image compression. *Journal of Appl. and Comput. Harmonic Analysis* **4**(2), pp. 147-187.
- [44] Milner, J. (2002). *Mondrian*. Phaidon, London, UK.
- [45] Muresan, D.D., and Parks, T.W. (2000). Prediction of image detail. *Proc. IEEE International Conference on Image Processing (ICIP2000)* **2**. Vancouver, Canada, pp. 323-326.
- [46] Nowak, R.D., and Baraniuk, R.G. (1999). Wavelet-based transformations for nonlinear signal processing. *IEEE Trans. Signal Process.* **47**(7), pp. 1852-1865.
- [47] Pennebaker, W.B., and Mitchell, J.L. (1992). *JPEG Still image data compression standard*. Van Nostrand Reinhold, New York, NY.
- [48] Phoong, S.-M., Kim, C.W., Vaidyanathan, P.P., and Ansari, R. (1995). A new class of two-channel biorthogonal filter banks and wavelet bases. *IEEE Trans. Signal Process.* **43**(3), pp. 649-665.
- [49] Press, W.K., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, UK.

- [50] Ramchandran, K. (1993). Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast. *Ph.D. Thesis*. Columbia University, New York, NY.
- [51] Ramchandran, K., and Vetterli, M. (1993). Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Process.* **2**(2), pp. 160-175.
- [52] Romberg, J.K., Wakin, M., and Baraniuk, R. (2002). Multiscale wedgelet image analysis: fast decompositions and modeling. *Proc. IEEE International Conference on Image Processing (ICIP2002)* **3**. Rochester, NY, pp. 585-588.
- [53] Romberg, J.K., Wakin, M., and Baraniuk, R. (2003). Approximation and compression of piecewise smooth images using a wavelet/wedgelet geometric model. *Proc. IEEE International Conference on Image Processing (ICIP2003)* **1**. Barcelona, Spain, pp. 49-52.
- [54] Rosiène, C.P., and Nguyen, T.Q. (1999). Tensor-Product Wavelet vs. Mallat Decomposition: A Comparative Analysis. *Proc. IEEE International Symposium on Circuits and Systems (ISCAS1999)* **3**. Orlando, FL, pp. 431-434.
- [55] Rosiles, J.G., and Smith, M.J.T. (2003). A low complexity overcomplete directional image pyramid. *Proc. IEEE International Conference on Image Processing (ICIP2003)* **1**. Barcelona, Spain, pp. 1049-1052.
- [56] Said, A., and Pearlman, W.A. (1996). A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Trans. Circuits and Systems for Video Technology* **6**(3), pp. 243-250.
- [57] Simoncelli, E.P., Freeman, W.T. Adelson, E.H., and Heeger, D.J. (1992). Shiftable multiscale transforms. *IEEE Trans. Inf. Theory* **38**(2), pp. 587-607.
- [58] Shapiro, J.M. (1993). Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Trans. Signal Process.* **41**(12), pp. 3445-3462.
- [59] Shoham, Y., and Gersho, A. (1988). Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoustics, Speech, and Signal Process.* **36**(9), pp. 1445-1453.
- [60] Shukla, R., Dragotti, P.L., Do, M.N., and Vetterli, M. (2005). Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. *IEEE Trans. Image Process.* **14**(3), pp. 343-359.
- [61] Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001). The JPEG 2000 still image compression standard. *IEEE Signal Processing Mag.*, **18**(5), pp. 36-58.
- [62] Strobach, P. (1991). Quadtree structured recursive plane decomposition coding of images. *IEEE Trans. Signal Process.* **39**(6), pp. 1380-1397.
- [63] Sullivan, G.J., and Baker, R.L. (1994). Efficient quadtree coding of images and video. *IEEE Trans. Image Process.* **3**(3), pp. 327-331.
- [64] Taubman, D., and Zakhori, A. (1994). Orientation adaptive subband coding of images. *IEEE Trans. Image Process.* **3**(4), pp. 421-437.
- [65] Tay, D.B.H., and Kingsbury, N.G. (1993). Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables. *IEEE Trans. Image Process.* **2**(4), pp. 466-480.
- [66] Temizel, A. (2007). Image resolution enhancement using wavelet domain hidden Markov tree and coefficient sign estimation. *Proc. IEEE International Conference on Image Processing (ICIP2007)* **5**. San Antonio, TX, pp. 381-384.
- [67] Thévenaz, P., Blu, T., and Unser, M. (2000). Interpolation revisited. *IEEE Trans. Medical Imaging* **19**(7), pp. 739-758.
- [68] Velisavljević, V., Beferull-Lozano, B., Vetterli, M., and Dragotti, P.L. (2003). Discrete Multi-directional Wavelet Bases. *Proc. IEEE International Conference on Image Processing (ICIP2003)* **1**. Barcelona, Spain, pp. 1025-1028.

- [69] Velisavljević, V. (2005). Directionlets: anisotropic multi-directional representation with separable filtering. *Ph.D. Thesis*. School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland.
- [70] Velisavljević, V., Beferull-Lozano, B., Vetterli, M., and Dragotti, P.L. (2005). Approximation power of directionlets. *Proc. IEEE International Conference on Image Processing (ICIP2005)* **1**. Genova, Italy, pp. 741-744.
- [71] Velisavljević, V., Beferull-Lozano, B., Vetterli, M., and Dragotti, P.L. (2006). Directionlets: Anisotropic Multi-directional Representation with Separable Filtering. *IEEE Trans. Image Process.* **15**(7), pp. 1916-1933.
- [72] Velisavljević, V., Beferull-Lozano, B., Vetterli, M., and Dragotti, P.L. (2006). Low-rate reduced complexity image compression using directionlets. *Proc. IEEE International Conference on Image Processing (ICIP2006)* **1**. Atlanta, GA, pp. 1601-1604.
- [73] Velisavljević, V., Beferull-Lozano, B., and Vetterli, M. (2007). Space-Frequency Quantization for Image Compression With Directionlets. *IEEE Trans. Image Process.* **16**(7), pp. 1761-1773.
- [74] Velisavljević, V. (2008). Edge-preservation resolution enhancement with oriented wavelets. *Proc. IEEE International Conference on Image Processing (ICIP2008)*. San Diego, CA, pp. 1252-1255.
- [75] Vetterli, M., and Kovačević, J. (1995). *Wavelets and Subband Coding*. Prentice Hall PTR, New Jersey, NJ (available online at <http://waveletsandsubbandcoding.org>).
- [76] Viscito, E., and Allebach, J.P. (1991). The Analysis and Design of Multidimensional FIR Perfect Reconstruction Filter Banks for Arbitrary Sampling Lattices. *IEEE Trans. Circuits and Systems* **38**(1), pp. 29-41.
- [77] Wakin, M., Romberg, J.K., Hyeokho, C., and Baraniuk, R. (2002). Rate-distortion optimized image compression using wedgelets. *Proc. IEEE International Conference on Image Processing (ICIP2002)* **3**. Rochester, NY, pp. 237-240.
- [78] Wakin, M., Romberg, J.K., Choi, H., and Baraniuk, R. (2006). Wavelet-domain Approximation and Compression of Piecewise Smooth Images. *IEEE Trans. Image Process.* **15**(5), pp. 1071-1087.
- [79] Wang, D., Zhang, L., and Vincent, A. (2005). Improvement of JPEG2000 Using Curved Wavelet Transform. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2005)* **2**. Philadelphia, PA, pp. 365-368.
- [80] Watson, A.B. (1987). The cortex transform: rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing* **39**(3), pp. 311-327.
- [81] Westerink, P.H. (1989). Subband coding of images. *Ph.D. Thesis*. Information Theory Group, Department of Electrical Engineering, Delft University of Technology, Delft, Netherlands.
- [82] Witten, I.H., Neal, R.M., and Cleary, J.G. (1987). Arithmetic coding for data compression. *Communications of the ACM* **30**(6), pp. 520-540.
- [83] Xiong, Z., Orchard, M.T., and Zhang, Y.-Q. (1997). A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations. *IEEE Trans. Circuits and Systems for Video Technology* **7**(2), pp. 433-437.
- [84] Xiong, Z., Ramchandran, K., and Orchard, M.T. (1997). Space-frequency Quantization for Wavelet Image Coding. *IEEE Trans. Image Process.* **6**(5), pp. 677-693.
- [85] Xiong, Z., Ramchandran, K., and Orchard, M.T. (1998). Wavelet Packet Image Coding Using Space-Frequency Quantization. *IEEE Trans. Image Process.* **7**(6), pp. 892-898.
- [86] Zuidwijk, R.A. (2000). Directional and Time-Scale Wavelet Analysis. *SIAM Journal on Mathematical Analysis* **31**(2), pp. 416-430.