Space-Frequency Quantization for Image Compression With Directionlets

Vladan Velisavljević, Baltasar Beferull-Lozano, *Member, IEEE*, and Martin Vetterli, *Fellow, IEEE*

***Abstract*—The standard separable 2-D wavelet transform (WT) has recently achieved a great success in image processing because it provides a sparse representation of smooth images. However, it fails to efficiently capture 1-D discontinuities, like edges or contours. These features, being elongated and characterized by geometrical regularity along different directions, intersect and generate many large magnitude wavelet coefficients. Since contours are very important elements in the visual perception of images, to provide a good visual quality of compressed images, it is fundamental to preserve good reconstruction of these directional features. In our previous work, we proposed a construction of *critically sampled perfect reconstruction* transforms with *directional vanishing moments* imposed in the corresponding basis functions along different directions, called *directionlets*. In this paper, we show how to design and implement a novel efficient space-frequency quantization (SFQ) compression algorithm using directionlets. Our new compression method outperforms the standard SFQ in a rate-distortion sense, both in terms of mean-square error and visual quality, especially in the low-rate compression regime. We also show that our compression method, does not increase the order of computational complexity as compared to the standard SFQ algorithm.**

***Index Terms*—Directional transforms, directional vanishing moments (DVMs), image coding, image orientation analysis, image segmentation, multiresolution analysis, nonseparable transforms, wavelet transforms (WTs).**

I. Introduction

PROVIDING efficient transform-based representations of images is an important problem in many areas of image processing, like approximation and compression. An efficient representation requires *sparsity*, that is, most of the information has to be contained in a few large-magnitude coefficients.

The standard 2-D wavelet transform (WT) has become very successful in recent years because it provides a sparse multiresolution representation of natural images due to the presence of vanishing moments in the high-pass (HP) filters (enforced by imposing zeros at $\omega = 0$) [3]. This method is conceptually simple and has a low computational complexity because of the simple separable 1-D filtering and subsampling operations. For these reasons, the 2-D WT has been adopted in the image compression standard JPEG-2000.

However, the performance of the 2-D WT is limited by the *spatial isotropy* of the basis functions and the construction only along the *horizontal and vertical directions*, which does not provide enough directionality. For this reason, the standard 2-D WT fails to provide a sparse representation of oriented 1-D discontinuities (edges or contours) in images [1]. These features are characterized by a geometrical coherence that is not properly captured by the isotropic wavelet basis functions. Thus, to provide an efficient representation of contours, the basis functions are required to be *anisotropic* and to have directional vanishing moments (DVMs) along more than the two standard directions. Several previous approaches, like curvelets [4], contourlets [5], bandelets [6], [7], and wedgeprints [8], have already addressed this nontrivial task. However, these methods have *higher complexity* than the standard 2-D WT and require *nonseparable* filtering and filter design. Furthermore, these transforms are often *oversampled*, thus, making it nontrivial to have efficient image compression methods. Another directional method that resides on content-based adaptation of transform directions has already been reported in [9], where image is segmented and the segments are separately resampled and transformed so that the dominant directions are aligned with the horizontal or vertical direction. Similarly, in [10], the WT is applied along curves such that the energy in the HP subband is minimized.

Several recently proposed directional approaches use the lifting scheme [11] in image compression algorithms. This scheme is exploited in [12], where transform directions are adapted pixel-wise throughout images. A similar adaptation is used in [13] and [14], but with more (9 and 11, respectively) different directions. In addition, the method in [13] uses the pixel values at fractional coordinates obtained by interpolation. Lifting is also implemented in [10] and in [15], where the 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.

In our previous work [2], [16], [17], we designed *critically sampled* anisotropic basis functions with DVMs across *any* two directions with rational slopes, which we called *directionlets*. Our basis construction retains the separable processing and the computational simplicity of the standard 2-D WT. We showed that directionlets outperform the standard 2-D WT in nonlinear approximation (NLA) of images while keeping a similar complexity. In [17], we also analyzed the approximation power of

Manuscript received September 7, 2006; revised March 28, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giovanni Poggi.

V. Velisavljević is with the Deutsche Telekom Laboratories, 10587 Berlin, Germany (e-mail: vladan.velisavljevic@telekom.de).

B. Beferull-Lozano is with the Group of Information and Communication Systems, Instituto de Robótica-Escuela Técnica Superior de Ingenieria, Universidad de Valencia, 46980, Paterna (Valencia), Spain (e-mail: baltasar.beferull@uv.es).

M. Vetterli is with the School of Computer and Communication Sciences, EPFL, CH-1015 Lausanne, Switzerland, and also with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: martin.vetterli@epfl.ch).

Digital Object Identifier 10.1109/TIP.2007.899183

1057-7149/$25.00 © 2007 IEEE

directionlets when applied to piecewise smooth synthetic images with smooth twice continuously differentiable discontinuity curves.

The improvement in the efficiency of image representation provided by directionlets motivated us to implement our transform in the wavelet-based image compression methods that exploit the correlation of wavelet coefficients across scales [18]–[20]. We showed in [21] that directionlets can be structured in hierarchical multiscale trees of transform coefficients, similarly to the wavelet zerotrees originally proposed in [18]. Furthermore, we demonstrated that directionlets outperform the standard 2-D WT in the zerotree compression method [18] at low operational bit rates.

Our main goal is to design and implement a compression method based on space-frequency quantization (SFQ) [20] using directionlets instead of the WT. In the SFQ, the best hierarchical wavelet trees are chosen using Lagrangian optimization, that is, the hierarchical trees are pruned such that the total Lagrangian cost is minimized. Furthermore, the best quantizer step size is also found using Lagrangian optimization and the chosen quantizer is applied to all wavelet coefficients retained in the trees. Since directionlets retain orthogonality inherited from the 1-D wavelet filter-banks, thus conserving the mean-square error (MSE) in the transform domain,[1] they can be conveniently applied in the various compression methods based on Lagrangian optimization. In this paper, we demonstrate that directionlets lead to a significant gain in the compression performance, especially at low operational bit rates. The compressed images obtained using our method are better than the images obtained using the standard SFQ both in terms of MSE and visual quality. Furthermore, because of directional adaptation, the artifacts that appear as a result of compression of images are aligned with locally dominant orientations across the image domain and are, thus, less visually annoying. At the same time, our method retains the same order of computational complexity as the standard method.

In Section II, we briefly review principles of the standard SFQ and the construction of directionlets. Then, in Section III, we provide the motivation for spatial segmentation of images and present the details of the SFQ compression method based on directionlets. The segmentation process can cause some blocking effect in the compressed images, which is especially noticeable at low bit rates. For that reason, in the same section, we provide a deblocking algorithm used to remove the blocking effect and also we analyze the computational complexity of the compression method. We show that the computational complexity of our compression algorithm is not substantially increased when compared to the complexity of the SFQ method. In Section IV, we present the results of compression for several standard test images. Finally, in Section V, we provide several conclusions.

II. Background and Related Work

Here, we give a brief overview of the related work. First, we review the basic definitions and properties of the SFQ compression method explained in [20] and based on the joint rate-distortion (R-D) optimization using the Lagrangian method.[2] Then, we revisit the construction of directionlets presented in [2].

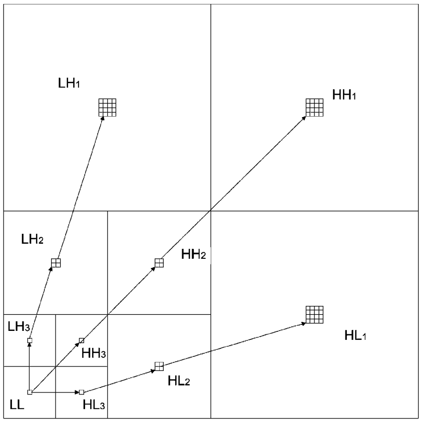


Fig. 1. Wavelet coefficients are grouped in tree structures to exploit the multiscale correlation. Each coefficient has four children in the next finer scale in the 2 × 2 region that corresponds to the same spatial location. The exceptions are the coefficients at the coarsest scale, which have only three children, and the ones in the finest scale, which have no children.

A. Space-Frequency Quantization

The SFQ image compression method for images was originally proposed in [20]. In the follow-up work [23], the authors analyzed the extension of the SFQ using wavelet packets. However, despite the improvement obtained along this direction, in this paper, we focus only on the adaptation of directionlets to the first implementation of the SFQ using the standard WT given in [20].[3] Here, we briefly revisit 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 multiscale correlation among wavelet coefficients produced by the standard 2-D WT. The coefficients are structured in multiscale trees so that one tree consists of the coefficients from different transform scales at the same spatial location (see Fig. 1). 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 in [18], where it is referred to as *zerotrees*, whereas a similar one is exploited in [19].

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

[1] Actually, if the used 1-D wavelet filter-bank is biorthogonal, then directionlets are also biorthogonal and the mean-square error is not exactly preserved. However, in practice, the frequently used biorthogonal filter-banks, like "9-7" [22], are very close to orthogonal and the difference between the mean-square error in the original and transform domains is negligible.

[2] A review of the main principles of Lagrangian optimization is given in Appendix I.

[3] We leave the implementation using wavelet packets for the future work.

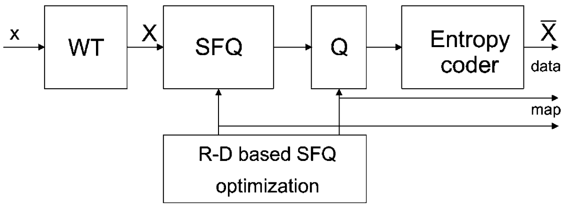


Fig. 2. Standard SFQ encoding consists of four blocks. First, the standard 2-D WT is applied on the input image x. Then, the SFQ encoder decides which subset of the wavelet coefficients should be discarded based on the R-D optimization. The retained coefficients are quantized using the optimal quantization 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.

both tasks, Lagrangian optimization (see Appendix I) 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 Fig. 2.

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 multiscale 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 multiscale 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 from [24], 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.

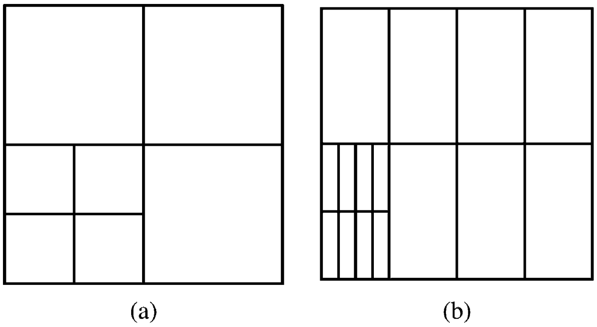


Fig. 3. Frequency decomposition of the standard wavelet transform (WT) and anisotropic wavelet transform (AWT). (a) The frequency decomposition of the standard 2-D WT. The number of transforms along the horizontal and vertical directions is equal at each scale and, thus, the transform is isotropic. The corresponding basis functions have symmetric square support. (b) The frequency decomposition of the AWT. The transform is anisotropic because the transform steps along one direction are applied more times than the ones along the other direction.

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 $\{q{:}q = 7.5 + 0.1 \cdot k, k = 1, 2, \ldots, 245\}$.[4] The value that minimizes the Lagrangian cost is optimal and applied to all retained wavelet coefficients. Similarly, the coefficients from the root of the multiscale 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 [25].

Notice that the probability density function of the low-pass (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 [22]. 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.

B. Directionlets

In our previous work [2], [16], we showed that the standard 2-D WT fails to provide an efficient representation of oriented and elongated objects, like contours. For that reason, we proposed a transform construction with anisotropic basis functions built along different directions, which we called *directionlets*. Here, we review the construction and basic properties of directionlets.

Recall that, in the standard 2-D WT, the number of 1-D transforms along the horizontal and vertical directions is the same at each scale, that is, the standard 2-D WT is isotropic [see Fig. 3(a)]. Consequently, the corresponding basis functions are also isotropic [with the square supports; see Fig. 4(a)]. However, elongated objects in images produce many large magnitude wavelet coefficients and this is the main reason for a lower

[4] The interval is optimized ad-hoc.

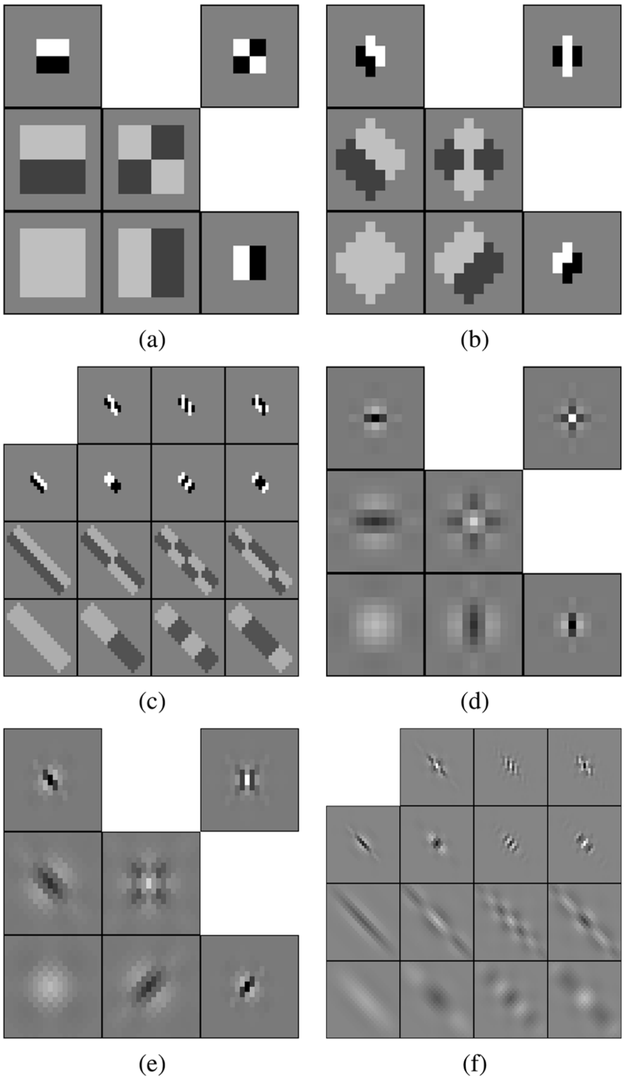


Fig. 4. Basis functions (directionlets) of the standard and anisotropic wavelet decomposition for different transform directions and 1-D filter-banks. The standard wavelet frequency decomposition is shown in Fig. 3(a), whereas the anisotropic decomposition is presented in Fig. 3(b). (a) The standard 2-D WT using the Haar 1-D filter-bank. (b) The skewed isotropic 2-D WT with the transform directions along 45° and $-45°$ using the Haar 1-D filter-bank. (c) The skewed AWT (S-AWT) with the transform directions along 45° and $-45°$ using the Haar 1-D filter-bank. (d) The standard 2-D WT using the biorthogonal 9-7 1-D filter-bank. (e) The skewed isotropic 2-D WT with the transform directions along 45° and $-45°$ using the biorthogonal 9-7 1-D filter-bank. (f) The S-AWT with the transform directions along 45° and $-45°$ using the biorthogonal 9-7 1-D filter-bank. In all cases, directionlets are critically sampled and separable.

efficiency of the standard 2-D WT in representation of these objects.

In contrast, in the anisotropic WT (AWT), the transform steps along one out of two directions are applied more than along the other. The frequency decomposition shown in Fig. 3(b) corresponds to an example of the AWT, where the number of the horizontal transform steps is twice the number of the vertical transform steps. This leads to anisotropic basis functions, elongated along the chosen transform direction [the horizontal for the example in Fig. 3(b)] at each scale.

The HP filters in both the standard 2-D WT and the AWT have vanishing moments only along the horizontal and vertical directions. To impose vanishing moments along more directions, we proposed in [2] and [16] a transform construction along a general pair of directions with rational slopes based on integer lattices.

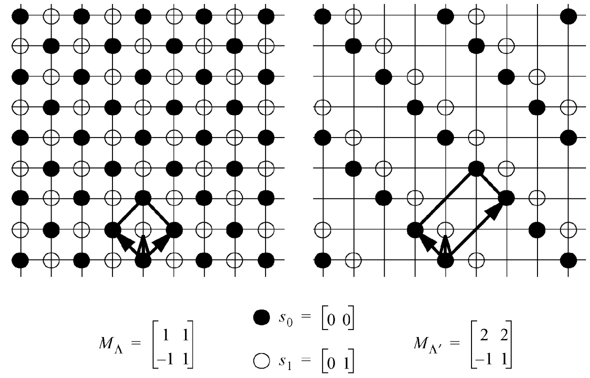

$M_\Lambda = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$
● $s_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}$
○ $s_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$
$M_{\Lambda'} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$

Fig. 5. Lattice Λ determined by the generator matrix $\mathbf{M}_\Lambda = (\mathbf{d}_1, \mathbf{d}_2)^T$ partitions the cubic lattice $\mathbb{Z}^2$ into two cosets (black and white circles). The transform (including filtering and subsampling) is performed along the first vector $\mathbf{d}_1$ of the matrix $\mathbf{M}_\Lambda$. The retained pixels in the subsampling operation belong to the sublattice Λ' determined by the generator matrix $\mathbf{M}_{\Lambda'} = (2\mathbf{d}_1, \mathbf{d}_2)^T$. The transform operations are performed in each coset separately.

Recall that a full-rank integer lattice Λ consists of the pixels obtained as linear combinations of two linearly independent integer vectors $\mathbf{d}_1$, and $\mathbf{d}_2$, where the coefficients are also integers. These two vectors form a generator matrix $\mathbf{M}_\Lambda = (\mathbf{d}_1, \mathbf{d}_2)^T$ associated to the integer lattice. Any integer lattice is a sublattice of the cubic integer lattice $\mathbb{Z}^2$. Recall also that the cubic lattice $\mathbb{Z}^2$ can be partitioned into $|\det(\mathbf{M}_\Lambda)|$ cosets of the lattice Λ.

The 1-D WT (including both filtering and subsampling operations) is applied on the pixels of the lattice Λ located along the direction determined by the first vector $\mathbf{d}_1$ in the corresponding generator matrix $\mathbf{M}_\Lambda$. The pixels retained after the subsampling operation belong to the sublattice Λ' of the original lattice Λ determined by the generator matrix $\mathbf{M}'_\Lambda = (2\mathbf{d}_1, \mathbf{d}_2)^T$. Such a subsampling operation allows for alignment of the retained pixels in the direction determined by the second vector $\mathbf{d}_2$ and efficient iteration of transform steps. Recall that, if the cubic lattice $\mathbb{Z}^2$ is partitioned into more cosets, then the processing is performed in each coset separately. An example of the lattice-based transform is shown in Fig. 5 for the transform directions $+45°$ and $-45°$ and two cosets.

The transform obtained as a combination of the lattice-based filtering and subsampling and the frequency decomposition used in the AWT results in the skewed AWT (S-AWT). The basis functions of the S-AWT, which we call *directionlets*, are elongated along one of the two transform directions (not necessarily horizontal or vertical). Several examples of directionlets are shown in Fig. 4 with different constructions and transform directions. Notice that, in the case $|\det(\mathbf{M}_\Lambda)| = 1$, directionlets inherit resolution-scalability from the standard 2-D WT, that is, the lowest frequency subband can be considered as a low-resolution version of the original image.

Notice also that, as shown in [2] and [16], anisotropy and the directionality power of directionlets allow for more efficient representation of elongated objects generating fewer large magnitude transform coefficients. Here, we want also to emphasize that directionlets are fully separable since all the basic operations (filtering and subsampling) are 1-D. Furthermore, the computational complexity of the transform is low and all the standard wavelet theory carries over. At the same time and very importantly, directionlets are critically sampled allowing for efficient compression algorithms based on Lagrangian optimization, as explained in the next section.

III. Space-Frequency Quantization Using Directionlets

Now, we show how directionlets can be successfully combined with the standard SFQ compression approach, resulting in an adaptive compression method using Lagrangian optimization.

First, we motivate and explain our spatial segmentation algorithm and its local adaptation of transform directions in each image segment. Then, we explain the construction of our adaptive compression algorithm. 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.

A. Spatial Segmentation

As explained in Section I, 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. However, images have geometrical oriented features that vary over space. Directionality, thus, can be considered as a local feature, defined in a small neighborhood.

Since directionlets can have up to 2 DVMs across any two directions with rational slopes, we have to adapt the choice of directions locally to each neighborhood. Thus, this implies a need for *spatial segmentation* as a way of partitioning an image into smaller segments with one or a few dominant directions per segment.

Even though there exist many ways to segment image, we choose the iterative quad-tree segmentation [26]–[29] 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).[5]

Therefore, we apply the S-AWT independently for 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.

B. Compression Algorithm

The implementation of the S-AWT is described in [2] and reviewed in Section II-B. 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 anisotropy, we do not make use of it here. The main reason for this is the quad-tree spatial segmentation, which is *isotropic*. As we noticed in the experiments and contrasting with the case analyzed in [2] (where an anisotropic segmentation was used in the analysis of asymptotic behavior in nonlinear approximation), an *anisotropic* transform applied to *isotropic* segments results in a less efficient representation. Furthermore, an anisotropic transform requires more bits to specify the choice of the transform directions when compared to the corresponding isotropic 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 anisotropic construction. For those reasons, we restrict the S-AWT only to isotropic 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 $|\det(\mathbf{M}_\Lambda)| = 1$. 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 four 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.[6] 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 [22] 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.[7]

Recall that the S-AWT provides a similar multiresolution analysis as the standard 2-D WT. Thus, directionlets can also be grouped in multiscale 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 Fig. 1). The multiscale 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 applied along nonstandard directions, the

[5] Notice that we are aware of a recent analysis of other more general segmentation methods [30], but the implementation of these methods in our image compression is left for future work.

[6] Here, by linear independence of a pair of directions, we mean that the two corresponding vectors are linearly independent.

[7] Since the "9-7" filter-bank consists of linear-phase filters, the symmetric extension preserves critical sampling in the transform domain.

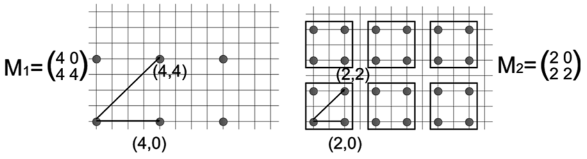


Fig. 6. Multiscale grouping of wavelet coefficients. Children are grouped in squares 2 × 2 and joint to the corresponding parent. The shape of the children groups is not affected by the transform directions. The example shows grouping in the case of the transform directions determined by the vectors (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 $\mathbf{M}_1$. In the right-hand figure, the children-coefficients are obtained after only one transform step and the corresponding generator matrix is $\mathbf{M}_2$.

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.[8] 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 Fig. 6 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 $\{7.5 + 0.1 \cdot k, k = 1, 2, \ldots, 245\}$. 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 $\{5.0 + 0.5 \cdot k, k = 1, 2, \ldots, 245\}$.

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 in [20] and referred to in Section II-A 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 variable $\mathrm{maxSlevel}$ is chosen *a priori*. In our experiments, we set $\mathrm{maxSlevel} = 3$. 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 $\mathrm{maxSlevel} = 0$ and the list of the allowed transform directions reduced to a single entry (0°, 90°).

Algorithm 1 The Full Adaptive SFQ Compression Algorithm Using Directionlets

```
Step 0: Set $\mathrm{Slevel} \leftarrow 0$,

Step 1: If $\mathrm{Slevel} < \mathrm{maxSlevel}$, then:
    * Apply quad-tree segmentation in the original domain.
    * For each of the 4 segments go recursively to Step 1 with
      $\mathrm{Slevel} \leftarrow \mathrm{Slevel} + 1$.

Step 2: For each pair of transform directions from the list
$\{(0°, 90°), (0°, 45°), (0°, -45°), (90°, 45°), (90°, -45°)\}$:
    * Apply S-AWT to each segment using isotropic
      construction and build the hierarchical trees.
    * Quantize the LP coefficients using all values
      $q_{\mathrm{LP}} \in \{5.0 + 0.5 \cdot k, k = 1, 2, \ldots, 245\}$ and find the one
      that minimizes the Lagrangian cost.
    * For each $q_{\mathrm{HP}} \in \{5.0 + 0.5 \cdot k, k = 1, 2, \ldots, 245\}$:
        ◦ Prune the trees so that the resulting tree minimizes
          the Lagrangian cost [phase a) from Section II-A].
        ◦ For each subband from coarser to finer scales [phase
          b) from Section II-A]:
        — Order coefficients in descending order of variances
          computed as the sum of squares in the $3 \times 3$
          neighborhood.
        — Optimize $T_h$, and $T_l$.
        ◦ Compute and record the resulting Lagrangian cost.
    * Choose the best $q_{\mathrm{HP}}$ that minimizes the Lagrangian cost.

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

Step 4: If $\mathrm{Slevel} < \mathrm{maxSlevel}$, 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.
```

C. Deblocking

In Section III-A, 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

[8] The proof is trivial and we do not show it here.

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 old JPEG standard in the 90s [31]. Since then, there have been many successful deblocking algorithms [32], [33]. In this paper, we use the deblocking algorithm proposed in [33], which is based on the overcomplete wavelet representation proposed in [34]. In this method, the wavelet coefficients produced by two levels of the three channel (one LP channel, horizontal and vertical HP channels) undecimated 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 from [33] 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 0.05 dB). However, the visual quality of the deblocked images is improved importantly, as shown in Section IV.

D. 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 in [21]). 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

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

which results in $O(LN^2)$.

Notice that the S-AWT can also be implemented using the lifting scheme [11], 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.

The computational complexity of the S-AWT is substantially lower than the complexity of the other transforms. More precisely, bandelets [6] require $O(N^2(\log_2 N)^2)$ operations. The implementation of wedgeprints [8] 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 [5] 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 $12LN^2$ (two decomposition levels with three 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 $(\mathrm{maxSlevel} + 1)$ 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 1 formalizes these conclusions.

Proposition 1: 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 $O(N^2)$.

IV. Results

In this section, we apply our compression algorithm using directionlets to several standard natural test images of the size 512×512. The results are compared to the results of the standard SFQ in terms of both numerical and visual quality.

First, we compare the behavior of the standard WT and the adaptive S-AWT choosing best directions per segment in NLA

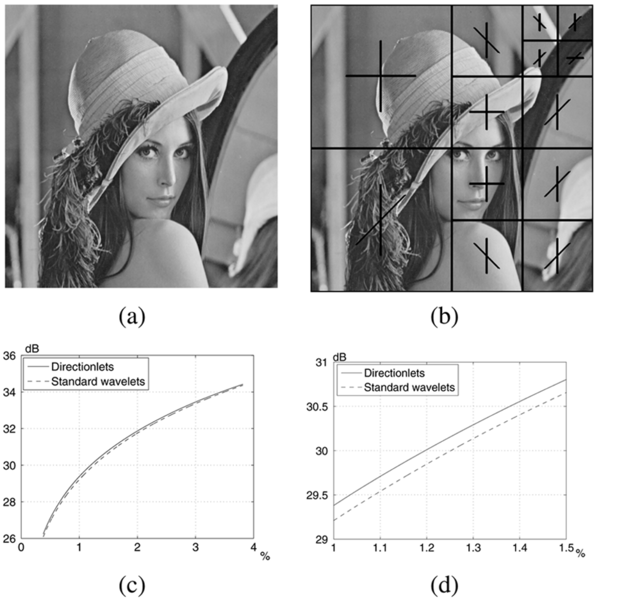


Fig. 7. Nonlinear approximation performance: *Lena* image. (a) The original image. (b) The optimal segmentation and the optimal choice of transform directions in each segment. (c) The NLA results obtained by retaining a portion of transform coefficients. (d) A magnified interval of the results shown in (c). Directionlets outperform the standard WT in terms of the approximation power.

of images. Then, we show the numerical results of the compression methods and reconstructed images for compression at different bit rates.

A. Nonlinear Approximation

The main task of approximation is to represent a signal by a subset of transform coefficients, while the rest of them is set to zero. We distinguish between linear approximation (LA) and nonlinear approximation (NLA). In the first, the indices of the retained coefficients are fixed, whereas, in the latter, they are adapted to the content of the signal.

The quality of approximation is commonly measured in terms of MSE, that is, for a signal $\mathbf{x}$ and its approximation $\hat{\mathbf{x}}$, the MSE is given by $|\mathbf{x}-\hat{\mathbf{x}}|^2$.[9] If an orthogonal transform is used in the NLA, then the optimal strategy to minimize the MSE is to retain the largest-magnitude transform coefficients [1].

Notice that the MSE decays as the number of retained coefficients grows. The asymptotic rate of decay of the MSE, as the number of retained coefficients tends to infinity, is a very important approximation property of the transform. In [2], [16], and [17], we showed that directionlets outperform the standard WT in terms of this asymptotic rate of decay in the case of a 2-D piecewise smooth signal with a 1-D C^2 smooth discontinuity curve. Here, we present the results of NLA for natural images and compare them to the results obtained using the standard WT.

As explained in Section III-A, directionality in images is a local characteristic, and, thus, spatial segmentation and adaptation of transform directions in segments is required to achieve a good representation. In the implementation of directionlets in NLA, we optimize spatial segmentation using Lagrangian optimization, similar to the method analyzed in Section III for compression.[10] In each segment, the best transform directions are chosen (again, in the sense of Lagrangian optimization) and a portion of the largest-magnitude transform coefficients is retained. Then, the corresponding inverse transform is applied and the reconstruction is compared to the original image in terms of PSNR.

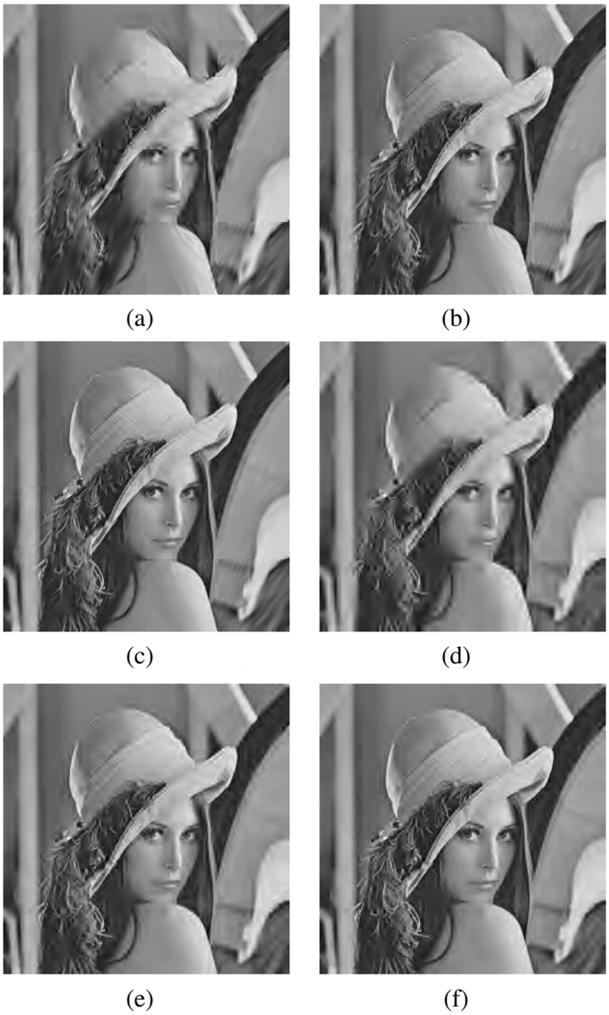
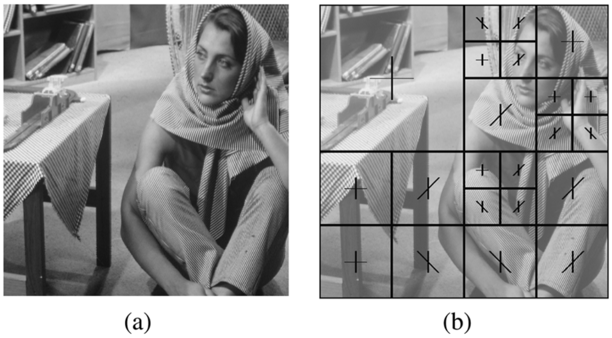


Fig. 8. NLA results of Lena: (a)–(c) Reconstructions of the image Lena for 0.5%, 1.0%, and 1.5% retained transform coefficients, respectively, using directionlets. The quality of the obtained images is 27.10, 29.38, and 30.80 dB, respectively. (d)–(f) The reconstructions at the same approximation rates using the standard WT. The quality of the images is 26.93, 29.21, and 30.66 dB, 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.

We show the results of NLA of the image *Lena*. The original image is shown in Fig. 7(a), whereas the optimal segmentation and the choice of transform directions in each segment is illustrated in Fig. 7(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 Fig. 7(c). The gain is obtained because directionlets are capable

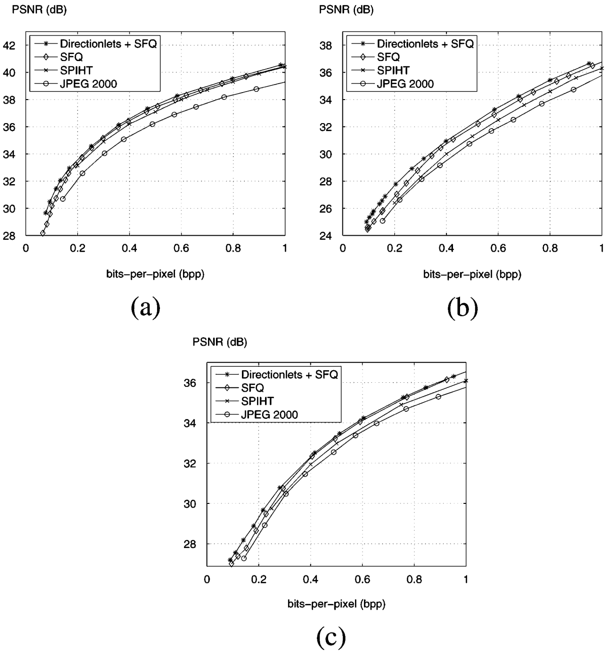
[9] The quality of images is usually evaluated by calculating the *peak-signal-to-noise-ratio* (PSNR), where PSNR=$10 \cdot \log_{10}(255^2/\mathrm{MSE})$.

[10] In the case of NLA, rate and distortion used in Lagrangian optimization are replaced by the number of retained coefficients and energy of the discarded coefficients, respectively.



(a) (b)

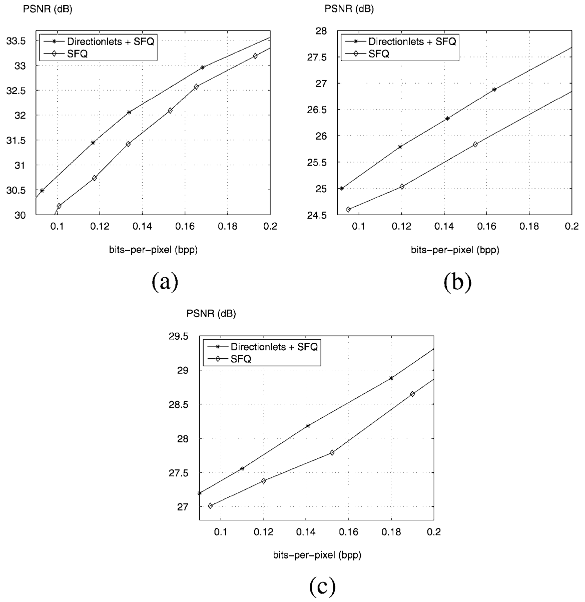
Fig. 9. *Barbara* image. (a) The original image. (b) The optimal segmentation and the optimal choice of transform directions in each segment. This solution is obtained as a result of the optimization process for the compression at the target bit rate 0.12 bpp.



(a) (b)

(c)

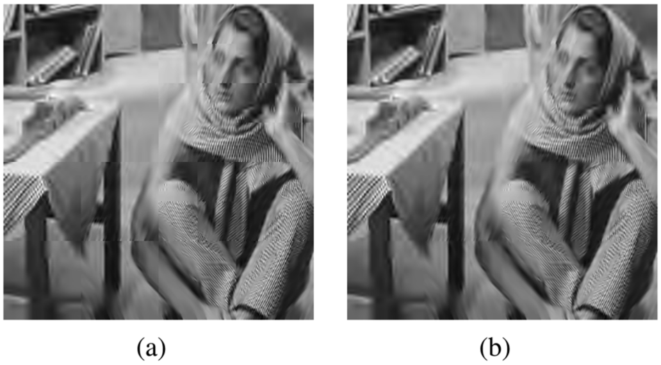
Fig. 10. Results of compression expressed in terms of PSNR using four methods: JPEG-2000, SPIHT [19], the standard SFQ [20], and the SFQ combined with directionlets. The methods are applied to three test images: (a) Lena, (b) Barbara, and (c) Boat. Our method outperforms the standard SFQ and the other two methods in all these examples.



(a) (b)

(c)

Fig. 11. Our compression method outperforms significantly the standard SFQ at low bit rates (below 0.2 bpp). The comparison of the results is shown for the three test images: (a) Lena, (b) Barbara, and (c) Boat.



(a) (b)

Fig. 12. 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 0.12 bpp for the optimal segmentation shown in Fig. 9 and PSNR $= 25.29$ dB. (b) The deblocking algorithm explained in Section III-C successfully removes the blocking effect. Even though the PSNR is improved by only 0.04 dB, the edges between segments are less visually perceptible.

of better capturing locally dominant directions in the image. In Fig. 8, the reconstructed images shown 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.

Notice that some remaining blocking effect is visible in the images approximated by directionlets because of the segmentation. However, even though this effect can be successfully removed by the deblocking method explained in Section III-C, here, we only compare the approximation performances and leave this correction for the next section.

B. Compression

Compression using orthogonal transforms is an extension of NLA that involves quantizing and indexing the retained coefficients. The quantized values are commonly compressed in an entropy coder, whereas the indexes are encoded using a predictive tree structure. We explained the compression method based on the SFQ and directionlets in details in Section III and, here, we present the results and compare them to the results of the standard SFQ.

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

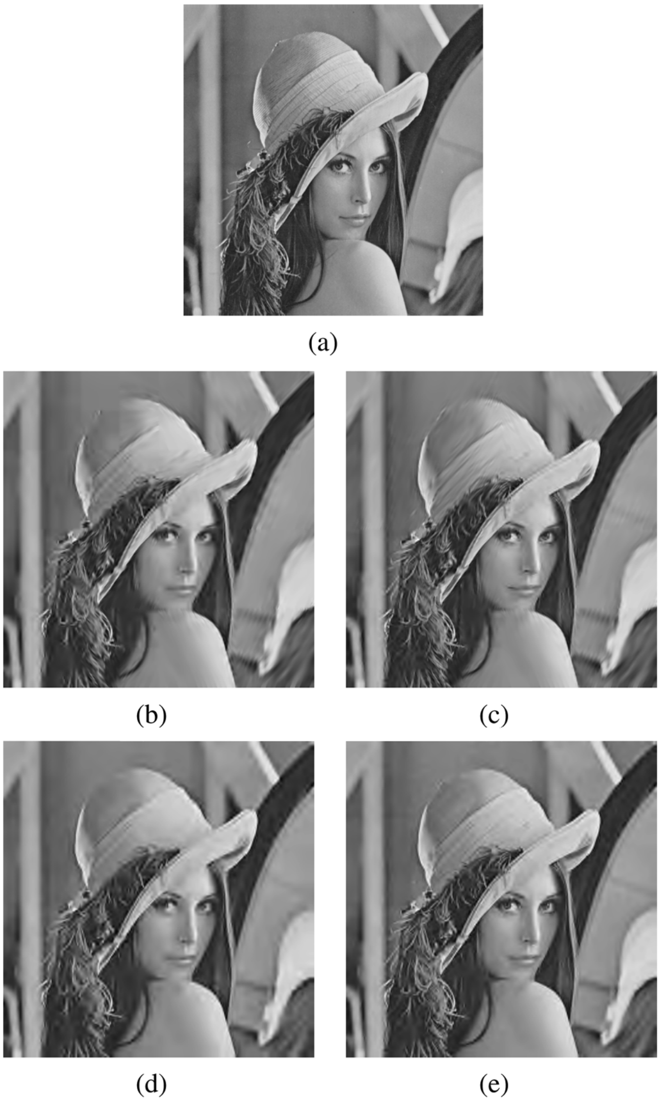

(a)
(b) (c)
(d) (e)

Fig. 13. Compression of Lena. (a) The original image. (b), (c) The image is compressed using the adaptive SFQ with directionlets at the bit rates 0.10 bpp (with the compression ratio 1:80) and 0.15 bpp (1:53), respectively. The numerical quality of the reconstructions is 30.92 and 32.56 dB. (d), (e) The image is compressed using the standard SFQ at the same bit rates. The quality is lower and is equal to 30.17 and 32.09 dB, respectively. The visual quality is also improved because the artifacts are oriented along locally dominant directions and are, thus, less annoying.

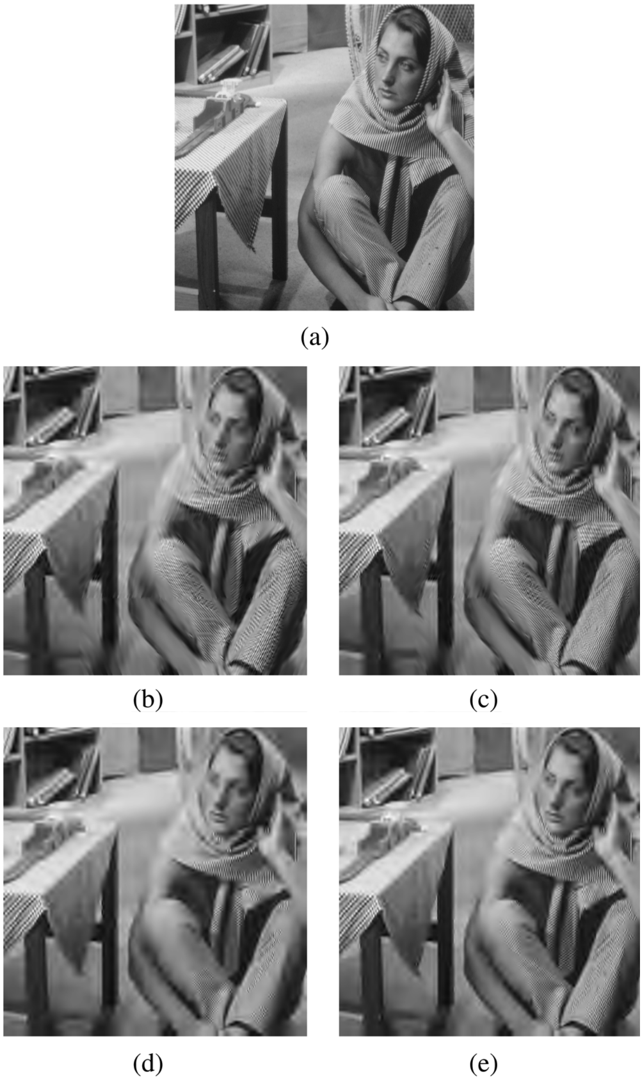

(a)
(b) (c)
(d) (e)

Fig. 14. Compression of Barbara. (a) The original image. (b), (c) The image is compressed using the adaptive SFQ with directionlets at the bit rates 0.10 and 0.15 bpp, respectively. The numerical quality of the reconstructions is 25.34 and 26.55 dB. (d), (e) The image is compressed using the standard SFQ at the same bit rates. The quality is lower and is equal to 24.58 and 25.75 dB, respectively.

transform directions in the R-D sense depend on the target bit rate and the Lagrangian multiplier. An example of the optimal solution is shown in the previous section in Fig. 7(b) for the image Lena compressed at 0.051 bpp. In Fig. 9(b), we show a similar example for the image Barbara at the operational bit rate 0.12 bpp. 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 Fig. 10. Notice that our method outperforms the standard SFQ as well as the other standard methods, like SPIHT [19] and JPEG-2000. This gain is especially noticeable at low operational bit rates. The results for the low bit-rate interval (below 0.2 bpp) are magnified and shown in Fig. 11 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 III-C, 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 0.12 bpp without and with deblocking is shown in Fig. 12. Even though the deblocking method does not improve significantly the numerical result (only 0.04 dB in the case shown in Fig. 12), it leads to a less disturbing visual distortion in the compressed images.

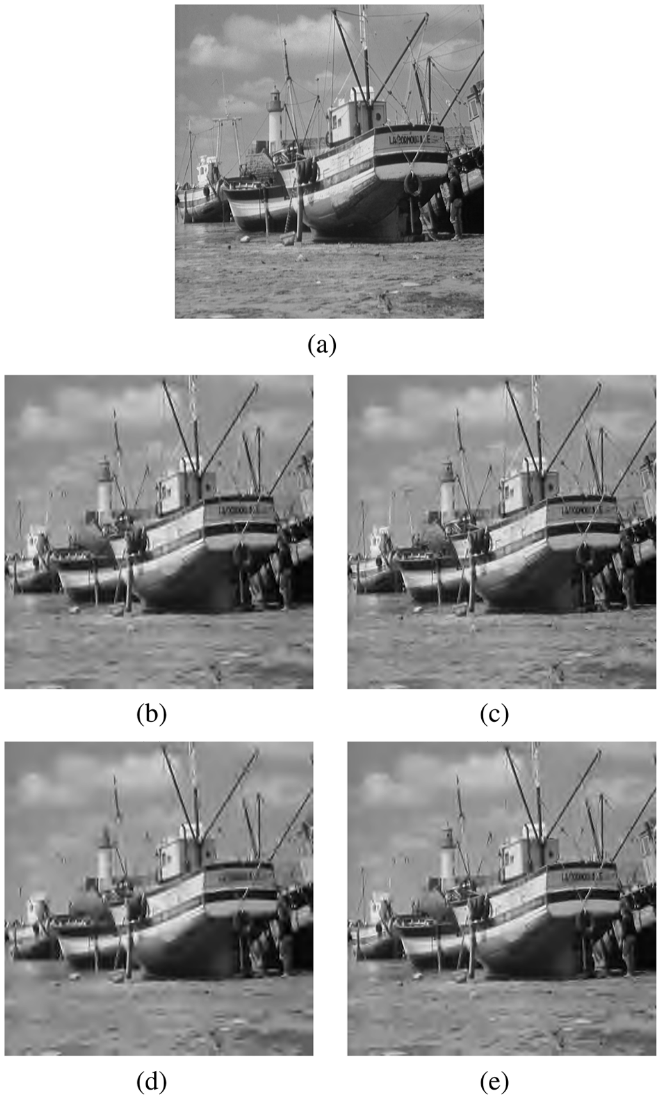


Fig. 15. Compression of Boat. (a) The original image. (b), (c) The image is compressed using the adaptive SFQ with directionlets at the bit rates 0.10 and 0.15 bpp, respectively. The numerical quality of the reconstructions is 27.10 and 28.36 dB. (d), (e) The image is compressed using the standard SFQ at the same bit rates. The quality is lower and is equal to 26.16 and 27.66 dB, respectively.

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 standard WT is used. The corresponding comparison for the test images Lena, Barbara and Boat are shown in Figs. 13–15, respectively, at the bit rates 0.10 and 0.15 bpp.

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 $\sum_{m=0}^{M-1} 4^m$, where M is the maximal number of segmentation levels. For the examples shown above $(M = 3)$, the upper bound is 21 bits.

Transform directions are encoded only for terminate segment-nodes in the quad-tree. Notice that the number of the terminate segments is upper bounded by 4^M. Therefore, the number of required bits is upper bounded by $4^M \cdot \log_2 K$, where K is the number of possible combinations of transform directions. For the presented examples, $K = 5$, and $M = 3$, 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.

V. Conclusion

We have proposed a novel adaptive image compression algorithm that combines the SFQ method proposed in [20] and directionlets. In our algorithm, image is segmented using the quad-tree segmentation method and transform directions are adapted to dominant directions in each segment. The spatial segmentation and the choice of transform directions are optimized in a R-D sense using Lagrangian optimization.

We showed in our previous work that directionlets are capable of providing a more efficient representation of elongated oriented features in images, like edges or contours, than the standard 2-D WT [2]. For that reason, our compression method outperforms the standard SFQ in terms of both the numerical and visual quality of compressed images at the same bit rate. The gain is especially significant at low operational bit rates, as we have demonstrated in Section IV.

Since we use spatial segmentation in our method, the reconstructed images are affected by blocking effect. Hence, we apply a postprocessing deblocking method that successfully removes the sharp edges along segment boundaries, as we have explained in Section III-C. We have shown that the improvement of the visual quality of reconstructed images is significant, although the difference in the numerical quality is negligible.

We have also analyzed the total computational complexity of our method. We have shown in Section III-D that the order of complexity of our method is of same as the order of complexity of the standard SFQ algorithm.

Appendix I
Lagrangian Optimization

The optimization tool based on Lagrange multipliers is explained in detail in [35], [36]. 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[11] is

[11] The R-D curve consists of the set of operational points in the R-D coordinate system that characterize a compression algorithm.

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.[12]

For the ith 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:

$$\min_{Q_i} \sum_{i=1}^{N} D_i(Q_i), \quad \text{such that } \sum_{i=1}^{N} R_i(Q_i) \leq R_{\text{budget}}. \qquad (2)$$

The constrained optimization problem given by (2) can be shown to be equivalent to an unconstrained problem given by [35]–[37]

$$J_i(\lambda) = \min_{Q_i} \sum_{i=1}^{N} D_i(Q_i) + \lambda R_i(Q_i). \qquad (3)$$

In (3), 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,[13] the minimization of the Lagrangian cost J_i is performed *independently* in each signal block. Thus, the optimization problem in (3) is unconstrained and can be solved in an algorithmically simpler way than the one in (2).

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 (2) and (3) are equivalent, the solution to the unconstrained problem of (3) is also the solution to the constrained problem of (2) for the particular case of $R_{\text{budget}} = \sum_{i=1}^{N} R_i$ (see [37] for the formal proof).

Each value of the total bit rate R_{budget} used in (2) and (3) 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 [36]–[38].

References

[1] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic, 1997.

[2] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic multi-directional representation with separable filtering," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1916–1933, Jul. 2006.

[3] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[4] E. J. Candès and D. L. Donoho, "Curvelets—A surprisingly effective nonadaptive representation for objects with edges," in *Curve and Surface Fitting*, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Nashville, TN: Vanderbilt Univ. Press, 1999.

[5] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[6] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[7] G. Peyré and S. Mallat, "Discrete bandelets with geometric orthogonal filters," presented at the IEEE Int. Conf. Image Processing, Genova, Italy, Sep. 2005.

[8] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, "Wavelet-domain approximation and compression of piecewise smooth images," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1071–1087, May 2006.

[9] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images," *IEEE Trans. Image Process.*, vol. 3, no. 4, pp. 421–437, Jul. 1994.

[10] D. Wang, L. Zhang, and A. Vincent, "Improvement of JPEG 2000 using curved wavelet transform," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Philadelphia, PA, May 2005, pp. 365–368.

[11] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl*, vol. 4, no. 3, pp. 247–269, 1998.

[12] Ö. N. Gerek and A. E. Çetin, "A 2-D orientation adaptive prediction filter in lifting structures for image coding," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 106–111, Jan. 2006.

[13] W. Ding, F. Wu, and S. Li, "Lifting-based wavelet transform with directionally spatial prediction," presented at the Picture Coding Symp., San Francisco, CA, Dec. 2004.

[14] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform via directional lifting and bandeletization," presented at the IEEE Int. Conf. Image Processing, Atlanta, GA, Oct. 2006.

[15] N. Mehrseresht and D. Taubman, "Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression," presented at the IEEE Int. Conf. Image Processing, Atlanta, GA, Oct. 2006.

[16] V. Velisavljević, "Directionlets: Anisotropic multi-directional representation with separable filtering," Ph.D. dissertation, School Comput. Commun. Sci., EPFL, Lausanne, Switzerland, 2005.

[17] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Approximation power of directionlets," presented at the IEEE Int. Conf. Image Processing, Genova, Italy, Sep. 2005.

[18] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3463, Dec. 1993.

[19] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, Jun. 1996.

[20] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 677–693, May 1997.

[21] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Low-rate reduced complexity image compression using directionlets," presented at the IEEE Int. Conf. Image Processing, Atlanta, GA, Oct. 2006.

[22] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[23] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Wavelet packet image coding using space-frequency quantization," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 892–898, Jun. 1998.

[24] A. Lewis and G. Knowles, "Image compression using 2-D wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 4, pp. 244–250, Apr. 1992.

[25] I. H. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, Jun. 1987.

[26] R. Leonardi and M. Kunt, "Adaptive split and merge for image analysis and coding," *Proc. SPIE*, vol. 594, 1985.

[27] P. Strobach, "Quadtree structured recursive plane decomposition coding of images," *IEEE Trans. Signal Process.*, vol. 39, no. 6, pp. 1380–1397, Jun. 1991.

[12] However, in the case of some widely used biorthogonal filter-banks (like "9-7" [22]), the error is negligible and we still use this optimization method.

[13] Otherwise, one can redistribute bits among signal blocks such that the signal blocks with larger slope λ gets more bits and, thus, contribute less to the total distortion. This redistribution can be continued until the optimum is reached, that is, until all signal blocks work at the same slope λ.

[28] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 327–331, May 1994.

[29] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 343–359, Mar. 2005.

[30] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman, "Fast search for best representations in multitree dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1779–1793, Jul. 2006.

[31] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1992.

[32] R. A. Gopinath, M. Lang, H. Guo, and J. E. Odegard, "Wavelet-based post-processing of low bit rate transform coded images," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, pp. 913–917.

[33] Z. Xiong, M. T. Orchard, and Y.-Q. Zhang, "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 433–437, Apr. 1997.

[34] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 710–732, Jul. 1992.

[35] K. Ramchandran, "Joint optimization techniques in image and video coding with applications to multiresolution digital broadcast," Ph.D. dissertation, Columbia Univ., New York, 1993.

[36] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Process.*, vol. 32, no. 2, pp. 160–175, Apr. 1993.

[37] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[38] W. K. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1988.

Vladan Velisavljević received the B.S. and M.S. degrees from the School of Electrical Engineering, University of Belgrade, Serbia, Yugoslavia, in 1998 and 2000, respectively, and the Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, in 2005.

From 1999 to 2000, he was a member of academic staff at the University of Belgrade. In 2000, he joined EPFL to work on his Ph.D. degree in the field of image processing with the Audiovisual Communications Laboratory, School of Computer and Communication Sciences. Since 2006, he has been with Deutsche Telekom Laboratories—Technische Universitäat Berlin, Germany.

His research interests include image and video processing and compression, wavelet theory, multiresolution signal processing, distributed image/video processing, sensor networks.

Baltasar Beferull-Lozano (S'01–M'02) was born in Valencia, Spain, in 1972. He received the M.Sc. degree in physics from the Universidad de Valencia (UV), Valencia, Spain, in 1995 (First in Class Honors) and the M.Sc. and Ph.D. degrees in electrical engineering from University of Southern California (USC), Los Angeles, in 1999 and 2002, respectively. His Ph.D. work was supported by a National Graduate Doctoral Fellowship from the Ministry of Education of Spain.

From January 1996 to August 1997, he was a Research Fellow Assistant with the Department of Electronics and Computer Science, UV, and from September 1997 to September 2002, he was a Research Fellow Assistant in the Department of Electrical Engineering, the NSF Research Integrated Media Systems Center and the Signal and Image Processing Institute (SIPI), USC. He has also been with AT&T Shannon Laboratories. From October 2002 to June 2005, he was a Research Associate in the Department of Communication Systems at the Swiss Federal Institute of Technology, Lausanne, Switzerland, and a Senior Researcher within the Swiss National Competence Center in Research on Mobile Information and Communication Systems. From July 2005 to November 2005, he was a Visiting Professor at the Universidad Politécnica de Valencia. Since December 2005, he has been a Research Professor in Instituto de Robótica-Escuela Técnica Superior de Ingeniería at Universidad de Valencia. He has served as a member of the Technical Program Committees for several ACM & IEEE International Conferences. His research interests are in the general areas of signal and image processing, distributed signal processing and communications for wireless networks, information theory, and communication theory.

Dr. Beferull-Lozano received several awards while at USC, including the Best Ph.D. Thesis paper Award in April 2002 and the Outstanding Academic Achievement Awards in April 1999 and April 2002.

Martin Vetterli (F'95) received the Engineering degree from Swiss Federal Institute of Technology, Zurich, the M.S. degree from Stanford University, Stanford, CA, and the Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL).

In 1986, he joined Columbia University, New York, first with the Center for Telecommunications Research, and then with the Department of Electrical Engineering, where he was an Associate Professor of electrical engineering. In 1993, he joined the University of California, Berkeley, where he was Full Professor until 1997. Since 1995, he has been a Professor with EPFL, where he was the Head of the Communication Systems Division (1996/1997) and is currently the Head of the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He has also been Vice-President for International Affairs at EPFL since October 2004. His research interests are in the areas of applied mathematics, signal processing, and communications.