EFFICIENT ZEROTREE-BASED IMAGE COMPRESSION WITH DIRECTIONLETS

Vladan Velisavljević[1], *Baltasar Beferull-Lozano*[2], *and Martin Vetterli*[3,4]

[1]Deutsche Telekom Laboratories, Germany
http://www.deutsche-telekom-laboratories.de/~velisavl/

[2]University of Valencia, Spain
http://www.uv.es/balbelo/

[3]EPFL, Switzerland,
[4]University of California, Berkeley, CA
http://lcavwww.epfl.ch/~vetterli/

ABSTRACT

*Directionlets are built as basis functions of critically sampled perfect-reconstruction transforms with directional vanishing moments (DVMs) imposed along different directions. Here, we combine the directionlets with the space-frequency quantization (SFQ) image compression method, originally based on the standard two-dimensional (2-D) wavelet transform (WT). We show that our new compression method outperforms the standard SFQ as well as the state-of-the-art image compression methods, such as SPIHT and JPEG-*2000*, in terms of the quality of compressed images, especially in a low-rate compression regime. We also show that the order of computational complexity remains the same, as compared to the complexity of the standard SFQ algorithm.*

1. INTRODUCTION

The standard two-dimensional (2-D) wavelet transform (WT) has become very successful in image compression 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 $\omega = 0$) [1]. This transform is conceptually simple and has a low computational complexity because of the simple separable one-dimensional (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 have directional vanishing moments (DVMs) along more than the two standard directions. Several previous approaches, like curvelets [2], contourlets [3] and bandelets [4], have already addressed this non-trivial task. However, these methods have *higher complexity* than the standard 2-D WT and require *non-separable* filtering and filter design. Furthermore, these transforms are often *oversampled*, thus, making it non-trivial to have efficient image compression methods.

Several recently proposed methods use the lifting scheme in image compression algorithms. This scheme is exploited in [5], where transform directions are adapted pixel-wise throughout images. A similar adaptation is used in [6] and [7], but with more different directions (9 and 11, respectively). In addition, the method in [6] uses the pixel values at fractional coordinates obtained by interpolation. 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 [8], anisotropic wavelet transforms have been constructed along different directions. The resulting basis functions, called *directionlets*, are *critically sampled* and have DVMs across *any* two directions with rational slopes. At the same time and very importantly, these transforms retain the separable processing (filtering and subsampling operations) and the computational simplicity of the standard 2-D WT. Directionlets have been successfully implemented in non-linear approximation of images providing a faster decay of mean-square error as compared to the standard 2-D WT. Furthermore, in [9], directionlets have been grouped in zerotrees, similarly to the multiscale grouping of standard wavelet coefficients in [10], while keeping a similar complexity.

Here, our main goal is to design and implement a compression method based on the space-frequency quantization (SFQ) [11] using directionlets instead of the standard 2-D WT. We show that our new method outperforms the standard SFQ as well as the state-of-the-art image coding algorithms, such as SPIHT [12] or JPEG-2000. At the same time, our method preserves the same order of computational complexity as the standard SFQ.

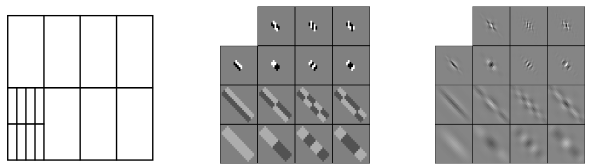
In Section 2, we briefly review the main principles of the construction of directionlets and also the standard SFQ method. Then, in Section 3, we present the details of our new compression method. We compare the results achieved by our method to the results obtained by the standard SFQ, SPIHT and JPEG-2000 in Section 4. Finally, we conclude in Section 5.

2. BACKGROUND AND RELATED WORK

2.1 Construction of Directionlets

The construction of directionlets has been explained in detail in [8]. Here, we only revisit the basic ideas.

Directionlets are constructed as basis functions of the so-called *skewed anisotropic wavelet transforms* (S-AWT). These transforms make use of the two concepts: anisotropy and directionality. Anisotropy is obtained by an unbalanced iteration of transform steps along two transform directions, that is, the transform is applied more along one than along the other direction. Directionality is a result of the construction along skewed transform directions built using integer lattices. The DVMs are imposed in the corresponding HP filters along any pair of directions. Two examples of directionlets are shown in Fig. 1(b) and (c). These basis functions are constructed using the frequency decomposition illustrated in Fig. 1(a) and the Haar and biorthogonal "9-7" 1-D filter-



(a) (b) (c)

Figure 1: The S-AWT allows for an anisotropic iteration of the filtering and subsampling operations applied along two different directions. (a) The decomposition in frequency for two iterations. The basis functions obtained from the (b) Haar and (c) biorthogonal ”9-7” 1-D scaling and wavelet functions.

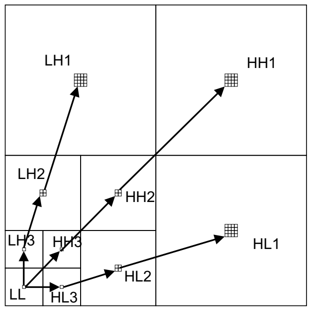


Figure 2: The wavelet coefficients are grouped in zerotrees to exploit the multi-scale correlation. The zerotrees have originally been proposed in [10].

banks, respectively.

2.2 Space-Frequency Quantization

The SFQ image compression method for images was originally proposed in [11]. Here, because of lack of space, we only briefly revisit the basic concept of the SFQ.

The main idea behind SFQ is to minimize a mean-square error (MSE) distortion measure of the reconstructed image for a given bit-rate constraint using Lagrangian optimization. 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 (*zerotrees*) so that one tree consists of the coefficients from different transform scales at the same spatial location (see Fig. 2). Each tree has a root at the corresponding coefficient from the coarsest scale. The same tree-structure is used in [10], whereas a similar one is exploited in [12].

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 (1) to select the subset of coefficients that should be discarded and (2) to choose which quantization step size should be used to quantize the retained coefficients. In both tasks, Lagrangian optimization is used to select the optimal solution in a rate-distortion (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. 3.

The optimization process consists of three phases: (a)

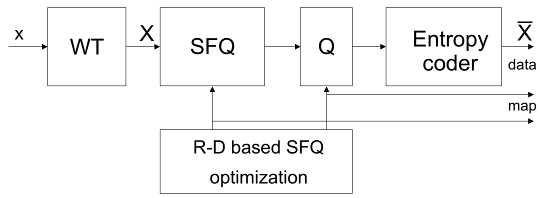


Figure 3: The standard SFQ encoding consists of four blocks: the 2-D WT, SFQ optimization, quantization and entropy coding. The task of the SFQ optimization is to pick the optimal subset of retained transform coefficients in a R-D sense. These coefficients are quantized in the subsequent step. The locations of retained coefficients are transmitted as a side information.

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.

In the first optimization phase (a), all nodes in the full depth multi-scale tree are checked bottom-up if it is cheaper in a R-D sense to keep or to zero out the descendant nodes. The process is iterated on the resulting pruned multi-scale tree until the convergence is reached, that is, until no new node is pruned. In the second phase (b), the locations of the retained nodes are encoded as a map information using a predictive scheme based on the variance of parent nodes. 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 an ad-hoc optimized list $\{q : q = 7.5 + 0.1 \cdot k,\ k = 1, 2, \ldots, 245\}$ for the scaling and wavelet coefficients and the value that minimizes the Lagrangian cost is chosen as optimal. The quantized coefficients are encoded using an adaptive entropy coder.

3. COMPRESSION ALGORITHM

Our compression algorithm is explained in more details in [14]. Here, because of lack of space, we present only brief ideas.

Images have geometrical oriented features that vary over space. For that reason, we have to adapt the DVMs of directionlets locally to each neighborhood. Recall that directionlets can have up to 2 DVMs. Thus, this implies a need for *spatial segmentation* as a way of partitioning image into smaller segments with one or a few dominant directions per segment. In our algorithm, we use the quad-tree segmentation, as the simplest method in the sense of encoding efficiency. The transform directions (and DVMs) are adapted independently in each spatial segment allowing for more efficient capturing of geometrical information. However, the separate processing of segments may cause some blocking effect in the compressed images, especially noticeable at low bit-rates. Hence, a post-processing is required to remove this effect, as explained in the sequel.

Next, we present the basic concept of our compression algorithm. Then, we give a brief overview of the deblocking algorithm originally proposed in [13] for JPEG compressed images and, finally, we analyze the computational complexity of the full method.

3.1 Definition of the Algorithm

Even though the construction of directionlets, as proposed in [8], allows for anisotropy and DVMs along any two directions with rational slopes, we apply two restrictions on the transform: (1) only the isotropic realizations are allowed and (2) the transform direction pairs are taken only from the set

$$\mathscr{D} = \{(0^\circ, 90^\circ), (0^\circ, 45^\circ), (0^\circ, -45^\circ), (90^\circ, 45^\circ), (90^\circ, -45^\circ)\}. \quad (1)$$

The reason for the first restriction is in a better compression performance with natural images in the case of isotropic segmentation (like quad-tree). The second restriction is imposed to prevent the constructions of directionlets that lead to more than one coset in the transform, since such constructions result in a less efficient image representation (see [8] for more details).

The depth of the multi-scale decomposition in the transform is ad-hoc optimized to 5 levels. The filtering operations are implemented using the 1-D biorthogonal "9-7" filter-bank [15]. Since a wider interval of the target compression bit-rates is allowed, as compared to the standard SFQ, the quantization step size is chosen from an extended list of values. The new extended list is given by $\mathscr{Q} = \{5.0 + 0.5 \cdot k,\ k = 1, 2, \ldots, 245\}$.

The compression algorithm consists of several embedded optimization phases based on minimization of the Lagrangian cost.[1] First, spatial segmentation is applied on the entire image in the original domain until a preselected maximal depth is reached and, then, the transform is applied on each segment separately using the transform directions from the list $\mathscr{D}$ given by (1). For each segment and combination of transform directions, the optimal encoding is found following the same principles as in the standard SFQ optimization phases [11] (referred to in Section 2.2 as phases (a) and (b)). The best transform directions that minimize the Lagrangian cost are found for each segment and the spatial quad-tree is pruned bottom-up to the optimal solution. Finally, the optimal quantization step size is chosen from the list $\mathscr{Q}$. The full algorithm is presented next.

Step 0: Set $Slevel \leftarrow 0$,

Step 1: If $Slevel < maxSlevel$, then:

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

Step 2: For each pair of transform directions from the list $\mathscr{D}$ given by (1):

- ∗ Apply directionlets to each segment using the isotropic construction and build the zerotrees,
- ∗ Quantize the LP coefficients using all values $q_{LP} \in \mathscr{Q}$ and choose the one that minimizes the Lagrangian cost,
- ∗ For each $q_{HP} \in \mathscr{Q}$, apply the standard SFQ, compute and record the resulting Lagrangian costs,
- ∗ 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 $Slevel < maxSlevel$, then:

- ∗ If the Lagrangian cost of the current segment is smaller than the sum of the Lagrangian costs of its children-segments, 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,

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

The variable *maxSlevel* determines the maximal segmentation depth and is chosen *a priori* (in our experiments, $maxSlevel = 3$). Notice that the jump in *Step 1* is not a loop, but a recursive call, where newly generated smaller segments are forwarded as arguments in 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.

3.2 Deblocking

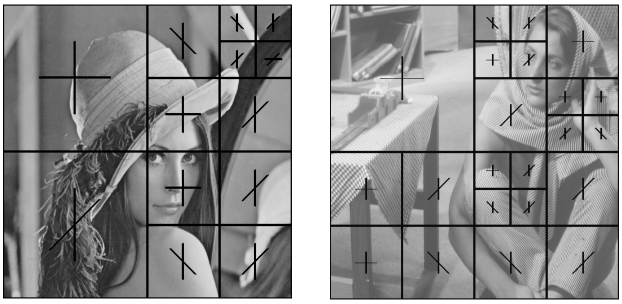
Because of the separated processing of spatial segments, the compressed images may be affected by a blocking effect, which is visible as sharp artificial edges along the segment boundaries. This effect is especially severe in the case of compression at low bit-rates. The same issue appeared in the JPEG standard in the 90's and, since then, there have been many successful deblocking algorithms. We use the algorithm proposed in [13], which is based on thresholding oversampled wavelet coefficients. The visual quality of the reconstructed images is importantly improved (as shown in Fig. 7), even though the impact on the MSE is negligible.

3.3 Computational Complexity

In [9], it has been shown that the order of computational complexity (or the order of the number of arithmetic operations) of applying directionlets to an $N \times N$ image using L-tap 1-D filters is given by $O(LN^2)$. Here, we show that our compression method increases the computational complexity of the standard SFQ only up to a constant and, thus, retains the same order.

The increase of the order is generated by two factors: (1) the two additional optimization phases (over spatial segmentation and directions) and (2) the deblocking algorithm. The two optimization phases contribute to the total complexity in the two multiplicative constants. The optimization over spatial segmentation increases the complexity $(maxSlevel + 1)$ times, whereas the optimization over directions contributes in the constant $|\mathscr{D}|$. Notice that these constants have small values in our experiments and do not depend on the image size. The deblocking algorithm carries more multiplication and addition operations because of the implemented forward and inverse overcomplete 2-D WT. However, the computa-

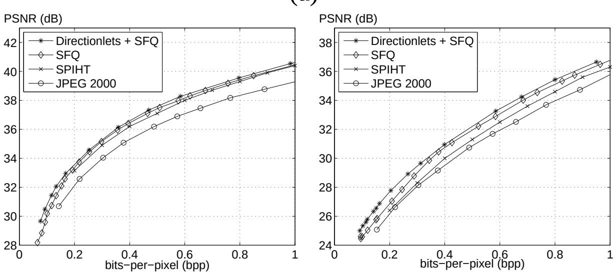
[1] Notice that directionlets retain orthogonality from the standard WT allowing for conservation of the mean-square error (MSE) in the transform domain. Thus, they can be fully implemented in Lagrangian optimization-based methods. Notice also that, although the conservation of the MSE does not hold exactly for the biorthogonal "9-7" filter-bank used in the experiments, the difference of the MSE is small enough and the optimization process is still valid.



(a) (b)

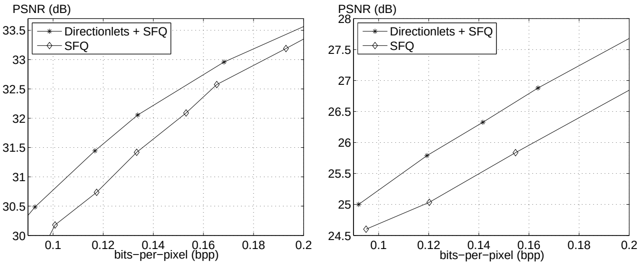
Figure 4: The optimal segmentation and choice of transform directions in each segment are found using Lagrangian optimization. These solutions are obtained for compression of the images (a) Lena at the target bit-rate 0.05bpp and (b) Barbara at 0.12bpp.

(a)



(b)

Figure 5: The numerical comparison of the compression performance in terms of PSNR. (a) The original images Lena and Barbara. (b) The results obtained by our method, the standard SFQ, SPIHT and JPEG-2000. Our method outperforms the standard SFQ as well as the other two methods.



(a) (b)

Figure 6: The numerical comparison of the compression performance at low bit-rates for the images (a) Lena and (b) Barbara. The gain that our method provides over the standard SFQ is significant.

(a)

(b)

Figure 7: The reconstructions of the two images are obtained by the compression at 0.1bpp using (a) the standard SFQ (30.17dB for Lena and 24.58dB for Barbara) and (b) our method (30.92dB and 25.34dB). Our new method provides better reconstructions than the standard SFQ at the same bit-rate. The artifacts are aligned with the locally dominant directions in the images and are less visually annoying.

tional complexity remains of the order $O(N^2)$. Thus, the total computational complexity is equal to the complexity of the standard SFQ multiplied by a constant, which does not depend on the image size.

4. RESULTS

We compare the performance of our compression method to the performance of the standard SFQ and the state-of-the-art methods SPIHT [12] and JPEG-2000 when applied to several standard test images. The comparison is given in terms of both the visual and numerical (PSNR) quality.

As explained in Section 3.1, the optimal spatial segmentation and transform directions are found using Lagrangian optimization. Fig. 4 shows a result of this optimization process in the case of the images Lena and Barbara compressed at the bit-rates 0.05bpp and 0.12bpp, respectively. Notice that the chosen directions are aligned to the locally dominant directions in the segments of the image.

The results of compression of the images Lena and Barbara using our method, the standard SFQ, SPIHT and JPEG-2000 are compared in Fig. 5. Our method outperforms all these methods in the entire bit-rate interval shown in the graphs. The gain is especially significant at low bit-rates (up to 0.8dB) and the results for that bit-rate interval are shown magnified in Fig. 6.

The corresponding reconstructions of the two images are shown in Fig. 7 for the compression at the bit-rate 0.1bpp using our method and the standard SFQ. The obtained PSNR factors are 30.92dB and 30.17dB for Lena and 25.34dB and 24.58dB for Barbara, respectively. Both the numerical and visual quality of the images obtained by our method are better than those obtained by the standard method. Moreover, the artifacts that appear in the low bit-rate compressed images are oriented along locally dominant directions and are, thus, less visually annoying.

5. CONCLUSIONS

We have proposed a novel adaptive image compression algorithm that combines the SFQ method proposed in [11] 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 segmentation and the choice of transform directions are optimized in a R-D sense using Lagrangian optimization. We showed that our method outperforms the standard SFQ and also the state-of-the-art image coding methods, like SPIHT or JPEG-2000, with no significant increase of computational complexity.

REFERENCES

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

[2] 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. Saint-Malo: Vanderbilt University Press, 1999.

[3] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Processing*, vol. 14, pp. 2091-2106, Dec. 2005.

[4] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Processing*, vol. 14, pp. 423-438, Apr. 2005.

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

[6] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *IEEE Trans. Image Processing*, vol. 16, no. 2, pp. 416–427, Feb. 2007.

[7] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform via directional lifting and bandeletization," in *Proc. IEEE International Conference on Image Processing (ICIP2006)*, Atlanta, GA, October 2006.

[8] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic multi-directional representation with separable filtering," *IEEE Trans. Image Processing*, vol. 15, pp. 1916-1933, July 2006.

[9] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Low-rate reduced complexity image compression using directionlets," in *Proc. IEEE International Conference on Image Processing (ICIP2006)*, Atlanta, GA, Oct. 2006.

[10] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3463, Dec. 1993.

[11] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing*, vol. 6, pp. 677-693, May 1997.

[12] 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, June 1996.

[13] 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.

[14] V. Velisavljević, B. Beferull-Lozano and M. Vetterli, "Space-Frequency Quantization for Image Compression with Directionlets," *IEEE Trans. Image Processing*, to appear.

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