RATE-DISTORTION IMPROVEMENT OF DIRECTIONAL WAVELETS BY MEGABLOCKING

Arian Maleki[1], *Boshra Rajaei* [2] , *Hamid R. Pourreza* [2]

ABSTRACT

The inefficiency of separable wavelets in representing smooth edges has motivated the researchers to pursue new two dimensional transformations. One of the successful transformations in image compression is the directional wavelets. Although researchers have empirically shown that the directional wavelets outperform the separable wavelets in compression, there is no theoretical analysis to demonstrate this phenomena, specially when the directional wavelets are combined with partitioning algorithms such as quadtree. In this paper, we calculate the rate-distortion performance of the directional wavelets on a class of images. Our analysis shows that the quadtree partitioning deteriorates the performance. Therefore we propose another scheme, called megablocking. Our theoretical and simulation results confirm that megablocking outperforms the quadtree approach.

Index Terms— Rate-distortion, quadtree, wavelet transforms, image coding, directional transforms.

1. INTRODUCTION

In the last decade, several schemes have been proposed to overcome the limitations of the traditional separable wavelets by incorporating directional representations. Most of the well-known transformations such as ridgelets [1], curvelets [2], contourlets [3], and shearlets [4] provide theoretically optimal approximation power on different edge models. However, they are highly overcomplete and therefore not useful for compression purposes. In this paper, we adopt the rate-distortion framework. This framewrok is better suited to the compression problem than the approximation power.

Le Pennec and Mallat used a similar approach [5] to prove that bandelets are nearly optimal. However, their analysis does not provide any insight on why the simpler directional wavelets are more successful and why the quadtree does not perform well in practice.

Other approaches such as wedgelets [6, 7] and platelets [8] address the compression problem on edge-like image models, as well. However, there are two main disadvantages with these approaches. 1. They are not yet optimal for more complicated geometries that are clearly present in images [9]. 2. They perform poorly in dealing with textures. There have been efforts to improve the performance of these algorithms on natural images such as [10]. But still these algorithms do not outperform the current state of the art.

Another area of research which has shown the most promise for the compression is the directional wavelets. The main idea is to apply the wavelet along the direction of the edges and get much fewer large wavelet coefficients. Many different approaches have been considered for directional wavelets including an application of shearing step to the image before a separable wavelet [11], subsampling and filtering along digital lines using lattices [12], and finally using lifting schemes for applying the directional wavelets [13]-[16]. All these approaches use a partitioning scheme to partition an image into blocks with just one dominant direction and apply the wavelets in that direction. In this paper, our goal is to analyze these algorithms theoretically. As a consequence of our analysis, we will show that the combination of the quadtree partitioning and directional wavelets proposed in several papers is not optimal in the rate-distortion setting [13, 16]. We will then propose the megablocking (first proposed in [7] for wedgelets) and claim that it improves the performance of the directional wavelets. The performance of our new scheme will be tested on natural images in the simulation section.

2. DIRECTIONAL WAVELETS

Consider a two dimensional function $f(t_1, t_2) \in L_2(\mathbb{R}^2)$. Directional wavelets, as the name suggests, is the application of the wavelet in two directions θ_1 and θ_2. We use the following notation for defining the wavelet coefficients,

$$W^{\theta_1,\theta_2}_{j_1,j_2,n_1,n_2} = \iint f(\cos(\theta_1)t_1 + \cos(\theta_2)t_2, \sin(\theta_1)t_1 + \sin(\theta_2)t_2)$$
$$\psi_{j_1,n_1}(t_1)\psi_{j_2,n_2}(t_2)dt_1dt_2, \qquad (1)$$

where ψ_{j_1,n_1} is applied in the direction of θ_1 and ψ_{j_2,n_2} is applied in the direction of θ_2. Clearly, we may replace either one or both of the ψ functions with ϕ, the father wavelet, to get the other coefficients. For the sake of brevity we do not rewrite those equations and refer the reader to [20]. Since images have complicated geometries, we have to partition an image to blocks and choose a direction for each block. A popular approach for partitioning images is called quadtree [5, 6, 16]. Quadtree partitioning approach starts from the whole image and recursively partitions each block to four smaller blocks if a certain criteria is met. We will explain our criteria in Section 5.1. After finding the partitions, the best direction is chosen for each partition and the wavelet is applied in that direction.

3. RATE-DISTORTION ANALYSIS

3.1. Our framework

Consider a class $\mathcal{F}$ of functions $f \in L^2(\mathbb{R}^2)$. Also, consider a compression scheme with an encoder $\mathcal{E} : \mathcal{F} \rightarrow \{1, 2, \ldots, 2^R\}$ and a decoder $\mathcal{D} : \{1, 2, \ldots, 2^R\} \rightarrow \mathcal{F}$. The distortion of a compression scheme on $\mathcal{F}$ at bit rate R, is the distortion of the least favorable function in this class, i.e.,

$$D(R) = \sup_{f\in\mathcal{F}} \|f - \mathcal{D}(\mathcal{E}(f))\|_2. \qquad (2)$$

[1] Department of Electrical Engineering, Stanford University.

[2] Department of Computer Engineering, Ferdowsi University of Mashhad.

978-1-4577-0539-7/11/$26.00 ©2011 IEEE

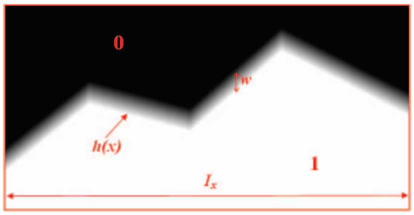

Fig. 1. A sample function in $\mathcal{H}^3$.

The performance of the best compression scheme under this framework is characterized by Kolmogorov ϵ- entropy [17]. In our analysis we are interested in high bit rate performance of different algorithms and the constants derived in this paper are not the best possible constants.

3.2. Edge model

In this paper we focus on piecewise linear models for edges. Let $\mathcal{PL}^Q(I_x, A)$ be the space of piecewise linear functions on the interval I_x, where Q is the number of singularity points and A is an upper bound on the magnitude of these functions. Let $I_x = [0, d]$ and $A = d$. For any $h(x) \in \mathcal{PL}^Q(I_x, d)$, we define a two-dimensional function

$$f_h(x,y) = \begin{cases} 1 & \text{if } y \leq h(x), \\ (h(x) - y + w)/w & \text{if } h(x) \leq y \leq h(x) + w, \\ 0 & \text{if } y > h(x) + w, \end{cases} \tag{3}$$

where $h(x)$ is the edge and w is the width of the edge. For the simplicity of the notation, all the linear pieces of the edge are assumed to have the same width. This may be relaxed without any major change to our analysis. Now we consider the class of two dimensional functions

$$\mathcal{H}^Q = \{f_h(x) \ : \ h(x) \in \mathcal{PL}^Q([0,d], d)\}. \tag{4}$$

3.3. Rate-distortion analysis of separable wavelets

For the sake of brevity we do not explain the definition of the separable wavelets and we refer the reader to [18] for the definitions. The following theorem shows the rate-distortion behavior of the separable wavelets on $\mathcal{H}^0$. In all the theorems of this section we assume that, the wavelets have finite support of length ℓ and their first moment is equal to zero [18].

Theorem 3.1. *On $\mathcal{H}^0$, a coding scheme based on separable wavelets and uniform quantization of the wavelet coefficients results in the following rate distortion at high bit rates:*

$$D(R) = O(R^{-\frac{3}{2}} \sqrt{\log_2 R}).$$

See [20] for the proof.
Clearly, except for the constants the decay rate of the distortion-rate function remains the same for H^Q as well.

3.4. Rate-distortion analysis of the directional wavelets

For the directional wavelets the first step is to code the directions. Suppose that we assign b' bits to θ. Then the precision of the scheme in θ is $\Delta\theta = \pi/2^{b'}$. The next step is to apply the directional wavelets in the quantized direction closest to the direction of the edge and code the wavelet coefficients again with uniform quantizers.

3.4.1. Analysis of one piece

Theorem 3.2. *The coding scheme that uses the directional wavelets with uniform quantization, achieves the following rate distortion on $\mathcal{H}^0$,*

$$D(R) = O(2^{-c_3\sqrt{R}}),$$

where c_3 is a constant that just depends on d and ℓ the length of the wavelet filters.

Refer to [20] for the proof.

3.4.2. Quadtree

Now consider H^Q space and invoke the directional wavelets based on a quadtree partitioning scheme. We first fix the depth of the tree and up to that depth, each block that has more than one piece in it is divided. The next theorem shows the rate distortion performance of this algorithm.

Theorem 3.3. *On $\mathcal{H}^Q$ the directional wavelet coder with uniform quantization and quadtree achieves the following rate distortion*

$$D(R) = O(\sqrt[6]{R} 2^{-c_3 \sqrt[3]{R}}).$$

Refer to [20] for the proof.

4. MEGABLOCKING: A SOLUTION TO THE SUBOPTIMALITY OF QUADTREES

Comparing Theorem 3.3 with Theorem 3.2 leads us to a conclusion that quadtree partitioning degrades the performance of the directional wavelets. In this section we prove that megablocking approach fixes this issue. Megablocks are formed by first applying quadtree partitioning and then merging neighboring blocks which contain no singularity points of the edge. We have explained the details of the megablocking approach in the simulation section. For the sake of brevity we do not repeat the details here.

Theorem 4.1. *On $\mathcal{H}^Q$ the coding scheme that uses the directional wavelets, uniform quantization and megablocking achieves the following rate distortion,*

$$D(R) = O(2^{-c_5\sqrt{R}}),$$

where c_5 is constant that just depends on d, ℓ, and Q.

Refer to [20] for the proof.
Clearly, megablocking does not deteriorate the performance of the directional wavelets and in that sense this partitioning scheme is optimal for the directional wavelets.

5. IMPLEMENTATION

Our implementation includes the following steps: First, the image is partitioned using the quadtree algorithm and the best direction is selected for each block. The next step is to align the directions and create megablocks to find a better partition of the image. Finally, the wavelet is applied to the megablocks in the specified directions. In this section we will explain each step briefly.

5.1. Direction selection using quadtree partitioning

For each pixel, the best direction is the one that minimizes the prediction error. However, the pixel-wise direction assignment is not practical for image compression. Therefore, we first partition input image into non-overlapping blocks using the well-known quadtree algorithm. In this algorithm each block is divided into four sub-blocks if the smaller partitions reduce the total cost. Following the proposal of [14, 16] for a given block $\mathbf{B}_b$ we consider the Lagrangian cost function defined as

$$C(\mathbf{B}_b) = \min_{d}\{\|\mathbf{B}_b - \hat{\mathbf{B}}_b^d\| + \lambda_1 R_d\}, \tag{5}$$

where λ_1 is a constant that will specify the complexity of the partitions, R_d denotes the number of bits spent on coding direction d and finally $\hat{\mathbf{B}}_b^d$ is the low pass approximation of $\mathbf{B}_b$ in the direction of d. Direction d is chosen from a finite set of quantized directions. The best direction is also defined as,

$$d_b^{\star} = \arg\min_{d}\{\|\mathbf{B}_b - \hat{\mathbf{B}}_b^d\| + \lambda_1 R_d\}, \tag{6}$$

5.2. Megablocking

The goal of this section is to address the implementation and technical challenges of the megablocking idea. Let us define a megablock more formally. Suppose that we have a partition of an image with blocks called $\mathbf{B}_b$. We use the notation $\mathcal{N}(\mathbf{B}_b)$ for the neighbors of the block $\mathbf{B}_b$. Two blocks are called similar, depicted by $\mathbf{B}_b \sim \mathbf{B}_{b'}$, if their directions are the same. A path $\mathbf{B}_{b_1} \leftrightarrow \mathbf{B}_{b_2} \leftrightarrow \ldots \mathbf{B}_{b_N}$ is a sequence of blocks such that $\mathbf{B}_{b_{i-1}} \in \mathcal{N}(\mathbf{B}_{b_i})$ for every $i \in \{2, \ldots, N\}$.

Definition 1. Two blocks $\mathbf{B}_b$ and $\mathbf{B}_{b'}$ are connected if and only if there is a path of similar blocks between them.

Definition 2. A *megablock* is a union of two or more blocks such that any pair of blocks are connected.

Finally a megablock $\mathcal{M}_b$ is called maximal iff there is no other megablock $\mathcal{M}_{b'}$ with $M_b \subset M_{b'}$.

One of the main issues in megablocking is that, in many regions of natural images there is no dominant direction and therefore the direction chosen is somewhat ‘random’. Therefore, there is a need for finding unreliable blocks (the ones without considerable cost difference while predicting along various directions) and change their directions accordingly. To do this, we will propose a global alignment scheme that will be explained later in this section.

5.2.1. Alignment step

Usually there are two types of unreliable blocks in the presence of noise: 1. small blocks and 2. blocks without dominant directional features e.g. blocks of smooth regions. Aligning the directions of such blocks and their neighbors may result in larger megablocks and improves the performance. Clearly we need a global alignment scheme, i.e. an alignment scheme that deals with all the blocks at the same time. If d_i is the current direction of each block and $\hat{d}_i$ is the direction after the alignment, we use the following optimization for calculating $\hat{d}_i$.

$$\min_{\hat{\mathbf{d}}} \sum_i w_i(\hat{d}_i - d_i)^2 + \lambda_2 \sum_j \sum_{i\in N(B_j)} |B_j|(\hat{d}_i - \hat{d}_j)^2 \tag{7}$$

where $|B_i|$ is the size of the block B_i, λ_2 is the Lagrange multiplier, and w_i is a measure of the reliability of the direction d_i and is defined as

$$w_i = |C_i^+ - C_i^-| \tag{8}$$

where C_i^+ and C_i^- are, respectively, the minimum and the maximum cost of predicting block i according to (5).

5.2.2. Creating megablocks

If a partition of an image with corresponding directions of the blocks are given, in this step we start joining the blocks until all the megablocks are maximal. For coding the new partition we use the following scheme. We define two block types: 1. *inner* blocks, which all 4-neighbors are belonging to the same megablock, and 2. *boundary* blocks with at least one neighbor from another megablock. Now scanning the blocks in a left to right and top to bottom order, we code each boundary block by 0 and each inner block by 1. Notice that scanning of blocks with different sizes is performed according to their origins. The above process is also reversible and the decoder can retrieve the structure perfectly. This approach uses one bit per block to code the structure of the megablock.

6. EXPERIMENTAL RESULTS

In this section, we compare the performance of the proposed megablocking scheme with quadtree-based directional wavelets, JPEG2000 and also DA-DWT proposed in [14]. In our implementation the choice of λ_1 is the same for these algorithms and is a decreasing linear function of bit rate. λ_2 is fixed and is empirically set to 0.3. We encode wavelet coefficients by TCE embedded bit-plane coder [19]. We also invoke run-length and variable run-length coding to encode megablocks and their related directions, respectively.

Due to the lack of space, we only focus on megablocking performance in dealing with noisy images since this may seem as a challenging situation for the alignment scheme. More advantages of our scheme can be found in [20]. Quadtree-based algorithms are extremely sensitive to even small values of noise. They result in lots of small blocks and, consequently, significant portion of bit budget is spent on coding the side information. Megablocking, however, reduces the overhead by changing unreliable directions and forming larger megablocks. Figure 2 gives the PSNR of the four mentioned algorithms for *Lena* and *Monarch* images as a function of PSNR of input noisy images. Images are corrupted by zero-mean Gaussian white noise. Accordingly, the proposed megablocking approach outperforms the quadtree-based algorithm by 1.80 and 2.23 dB, on

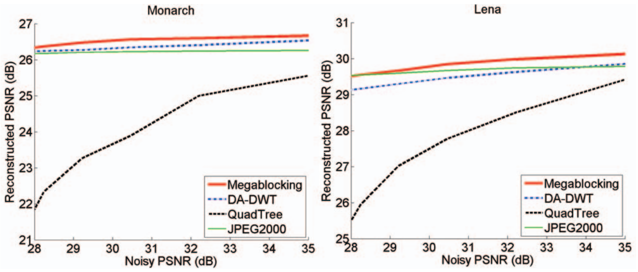

Fig. 2. Compression performance of the proposed megablocking, DA-DWT, the quadtree-based algorithm and the JPEG2000 at 0.1 bpp.

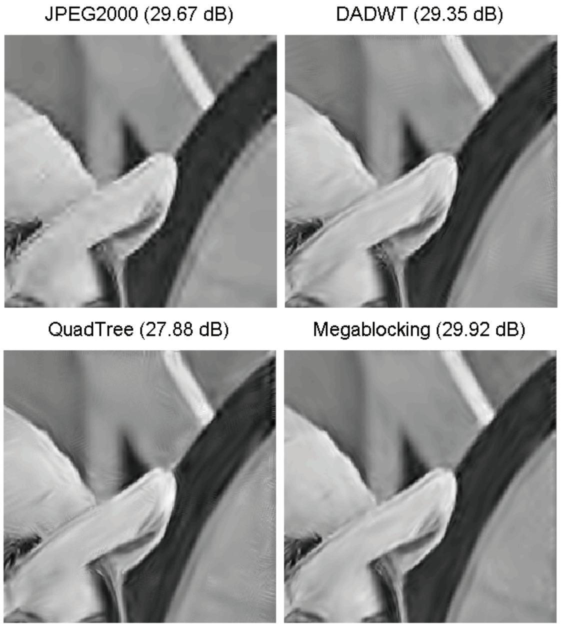

Fig. 3. Reconstructions of noisy *Lena* image at 0.1 bpp. PSNR of the original image is 30.01dB.

average, in the case of *Lena* and *Monarch*, respectively. It also performs better than JPEG2000 by up to 0.95 dB for *lena* and 0.49 dB for *Monarch*. Although the improvement over DA-DWT is up to .50 and .22 dB, respectively, for *Lena* and *Monarch*, the reconstruction from megablocking obviously better represents image geometries as shown in Figure 3. In this figure, the reconstruction results of four algorithms for noisy *Lena* at 0.1 bpp is provided. JPEG2000 clearly introduces ringing artifacts to image geometries. In DA-DWT and quadtree results the visual quality is affected by brushstroke-like artifacts. Reconstruction from megablocking, on the other hand, is clearly superior to JPEG2000 (the ringing effects are totally disappeared) and in comparison with DA-DWT and quadtree, megablocking better represents image geometries (the brush-like effects are mostly absent).

7. CONCLUSION

In this paper, we present a rate-distortion analysis for the directional wavelets transform. Our analysis led us to a new scheme for partitioning images, called megapartitioning. Theoretical and simulation results confirmed that our new partitioning scheme is able to outperform the quadtree partitioning and DA-DWT which are two of the recent successful implementations of the directional wavelets.

8. REFERENCES

[1] E. J. Candès and D. L. Donoho, “Ridgelets: a key to higher-dimensional intermittency?” *Phil. Trans. R. Soc. Lond. A.*, pp. 2495–2509, 1999.

[2] E. J. Candès and D. L. Donoho, “Curvelets – a surprisingly effective nonadaptive representation for objects with edges,” in *Curve and Surface Fitting*, Vanderbilt University Press, 1999.

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

[4] G. Kutyniok and D. Labate, “Construction of Regular and Irregular Shearlet Frames,” *J. Wavelet Theo. and Appl.*, Vol.1, pp. 1-10, 2007.

[5] E. Le Pennec and S. Mallat, “Sparse geometrical image approximation with bandelets, *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[6] D. L. Donoho, “Wedgelets: Nearly minimax estimation of edges,” *Annals of Stat.*, vol. 27, pp. 859–897, 1999.

[7] R. Shukla, P. L. Dragotti, M. N. Do, M. Vetterli, “Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images,” *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 1–17, March 2005.

[8] R. Willett and R. Nowak, “Platelets: a multiscale approach for recovering edges and surfaces in photonlimited medical imaging,” *IEEE Trans. on Medical Imag.*, vol. 22, 2003.

[9] A. Maleki, M. Shahram, and G. Carlsson, “Near optimal coder for image geometries With adaptive partitioning,” *Proc. IEEE Int. Conf. on Image Proc.*, San Diego, 2008.

[10] M. B. Wakin, J. K. Romberg, H. Choi, R. G. Baraniuk , “Image compression using an efficient edge cartoon + texture model,” *IEEE Data Compression Conference*, Utah, April 2002.

[11] D. S. Taubman, “Directionality and scalibility in image and video compression,” Ph.D. thesis, University of California Berkeley, 1994.

[12] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, “Directionlets: Anisotropic multi-directional representation with separable filtering”, *IEEE Trans. Image Process.*, vol. 15, pp. 1916–1933, July 2006.

[13] C.-L. Chang, A. Maleki, and B. Girod, “Adaptive wavelet transform for image compression via directional quincunx lifting,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Shanghai, China, Oct. 2005.

[14] C.-L. Chang and B. Girod, “Direction-adaptive discrete wavelet transform for image compression,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1289–1302, May 2007.

[15] V. Chappelier and C. Guillemot, “Oriented wavelet transform for image compression and denoising,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2892–2903, Oct. 2006.

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

[17] A. N. Kolmogorov, V. M. Tihomirov, “ϵ-entropy and ϵ-capacity of sets in functional spaces,” *American Math. Society Translation* (Ser. 2), vol. 17, pp. 277–364, 1961.

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

[19] C. Tian and S. S. Hemami, “An embedded image coding system based on tarp filter with classification,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Montral, QC, Canada, May 2004, vol. 3, pp. 49–52.

[20] A. Maleki, B. Rajaei, and H. R. Pourreza “Rate-Distortion improvement of directional wavelets by megablocking,” *Preprint*.