Submitted to: IEEE Transactions on Image Processing, September 1998

Multi-layered Image Representation: Application to Image Compression

F.G. Meyer [1], A.Z. Averbuch [2], and R.R. Coifman [3]

[1]Radiology and Computer Science, Yale University, New Haven, CT

[2]School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel

[3]Mathematics, Yale University, New Haven, CT

e-mail: `francois.meyer@yale.edu`

EDICS Category: 2-WAVP

Abstract

The main contribution of this work is a new paradigm for image representation and image compression. We describe a new multi-layered representation technique for images. An image is parsed into a superposition of coherent layers: smooth-regions layer, textures layer, etc. The multi-layered decomposition algorithm consists in a cascade of compressions applied successively to the image itself and to the residuals that resulted from the previous compressions. During each iteration of the algorithm, we code the residual part in a lossy way: we only retain the most significant structures of the residual part, which results in a sparse representation. Each layer is encoded independently with a different transform, or basis, at a different bitrate; and the combination of the compressed layers can always be reconstructed in a meaningful way. The strength of the multi-layer approach comes from the fact that different sets of basis functions complement each others: some of the basis functions will give reasonable account of the large trend of the data, while others will catch the local transients, or the oscillatory patterns. By selecting different analyzing functions, we allow different features to be discovered in the image. The reconstructed quality is related to the number of layers, and to the relative compression rate of each layer. This multi-layered representation has a lot of beautiful applications in image understanding, and image and video coding. We have implemented the algorithm and we have studied its capabilities. The results achieved by this method are better then those achieved by standard techniques such as JPEG, or zero-tree wavelet coders.

1 Introduction

The underlying assumption behind standard compression methods is that the basis (e.g. the Discrete Cosine Transform (DCT) basis, or a wavelet basis) used for compression is well adapted to most images. However, if one wishes to obtain a very sparse representation of an image it becomes critical that the basis functions bear a strong resemblance to the image. If the geometric properties (shape, smoothness, periodicity, etc) of the basis functions match the corresponding characteristics of the data to be analyzed, then the coefficients in such a basis are meaningful, and provide rich information about the data being analyzed. Unfortunately basis functions that compactly code smooth images (such as wavelets with many vanishing moments [2, 14, 26, 27, 28]) are ill suited to represent oscillatory patterns. Reciprocally, oscillatory analyzing functions such as the DCT can efficiently code textures, but they are ill equipped to represent smooth images. In order to obtain the most economical representation of an image composed of flat smooth regions and textures, one would typically need a mixture of functions chosen among wavelets, and trigonometric transforms. Unfortunately, not one single transform can offer this rich blend of smooth and oscillatory analyzing functions.

To alleviate this issue much larger libraries of waveforms, called wavelet packets [10], have been developed. In the two dimensional case wavelet packets provide an overcomplete library of patterns that vary in scale, oscillation, and location [29]. In order to select among all the wavelet packets a basis adapted to the content of the image, these patterns are matched to the image, and a selection of best matches that are sufficient for an efficient reconstruction is made [11]. This adapted basis is called the "best-basis" [11]. Unfortunately when coding images that contain a mixture of smooth and textured features, the best basis algorithm is always trying to find a compromise between two conflicting goals: describe the large scale smooth regions, and describe the local textures. For this reason, the best wavelet packet basis rarely provides the optimal transform to compress an image. A second drawback of standard transform coding techniques is the rapid degradation of the image quality when the bit rate drops. DCT-based methods (such as JPEG) cannot describe efficiently low frequency trends occurring at a large scale in the image. At high compression rates the blocks used for compression start to be visible in the reconstructed picture. Similarly, wavelet-based compression methods fail to preserve textures at low bitrate.

In order to address all these problems, our approach follows a completely different direction:

instead of forcing all images to adapt to one single basis, we use a collection of libraries of bases to represent a single image. The main contribution of this paper is a new paradigm for image representation and image compression. We describe a new multi-layered representation technique for images. An image is parsed into a superposition of coherent layers: smooth-regions layer, textures layer, etc. Each layer is encoded independently with a different transform, or basis, at a different bitrate, and the combination of the compressed layers can always be reconstructed in a meaningful way. The strength of the multi-layer approach comes from the fact that different sets of basis functions complement each others ; some of the basis functions will give reasonable account of the large trend of the data, while others will catch the local transients, or the oscillatory patterns. By selecting different analyzing functions, we allow different features to be discovered in the image. Furthermore, with each new basis we encode and compensate for the artifacts created by the quantization of the coefficients of the previous bases. The reconstructed quality is related to the number of layers, and to the relative compression rate of each layer.

The rest of the paper is organized as follows. In the next section we provide a general description of the principles of the algorithm. In Section 3 we discuss a key feature of the algorithm: the libraries of bases. We give two important examples: the wavelet packets, and smooth local cosine bases. In Section 4 we describe a new scheme to order and quantize the coefficients of wavelet packets, and local cosine transforms. Results of experiments are presented in Section 5.

2 Multi-layered image compression

2.1 Cascade of compressions: the "multi-layered" paradigm

A block diagram of the multi-layered algorithm is shown in Figure 1. The multi-layered compression algorithm consists in a cascade of compressions applied successively to the image itself and to the residuals that resulted from the previous compressions. An initial main approximation is obtained by compressing the input image with a wavelet basis. This first approximation preserves the general shape of the image, and captures the trend in the intensity function. We then reconstruct the compressed part, and we calculate the error between the original and compressed data. This compression error defines the first remainder, or *residual*. Residuals are composed of edges, and textures, and are compressed with wavelet packets or local trigonometric bases. These bases

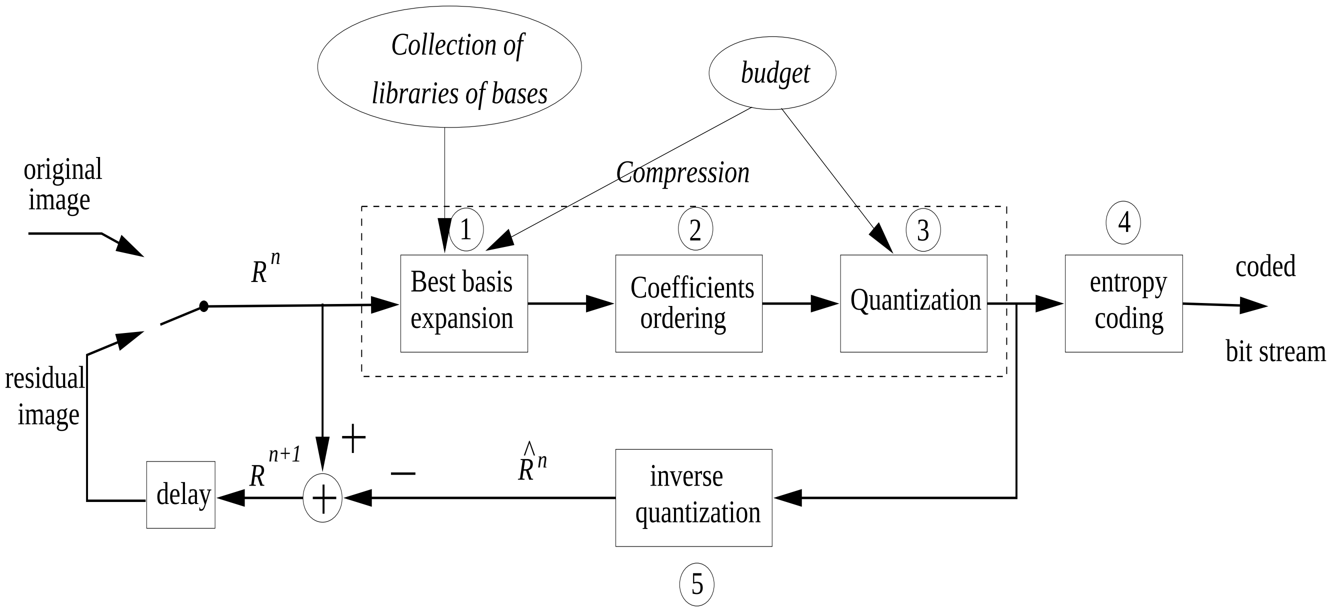

Figure 1: Block diagram of the multi-layered compression algorithm. In the first pass of the algorithm, the switch (on the left) is turned toward the original image (i.e the image that we want to compress). In the subsequent refinement passes the switch is turned toward the residual image. The block in the dotted line compresses either the original image, or the residual. This single pass compression consists of three parts: (1) best basis selection, and calculation of the coefficients of the image using the best basis, (2) ordering of the coefficients, and (3) quantization of the stream of coefficients. Finally, in (4) the quantized coefficients are entropy coded. The residual error is calculated, and is fed back to the compression algorithm.

are well adapted to texture coding [6, 16, 20]. Once the first residual is compressed, one defines the second residual as being the compression error of the first residual. The algorithm keeps on compressing the successive residuals until we reach a residual that contains no more structure. We describe now in details the different stages of the algorithm. We consider a sequence of libraries of orthonormal bases $(\mathcal{L}_i)_{i\in\mathbb{Z}}$ where each $\mathcal{L}_i$ is a library of functions that provides very large sub-collection of orthogonal bases. In this work $\mathcal{L}_0$ contains only one single wavelet basis, and $\mathcal{L}_1$ is a library of wavelet packets, or a library of local cosine transforms. The choice of the libraries, and their arrangement are parameters of the algorithm, and need to adjusted only once for large classes of images.

2.1.1 Initialization

Let I be an image. We first compress I over the library $\mathcal{L}_0$, using the budget b_0. The approximation is performed under a budget constraint, and the result of the approximation should be described with at most b_0 bits. Let $\hat{R}_0$ be the result of the compression, after decompression. $\hat{R}_0$ is an approximation of the original image I, and we have

$$I = \hat{R}^0 + R^1 \tag{1}$$

where R^1 is the approximation error. At this point we refine the approximation of I by calculating an approximation of the residual R^1. This is achieved by compressing the residual. But in order to discover different features in the image, we use a different library to compress R^1. We use a budget of b_1 bits to compress R^1. A best basis, $\left\{\Phi_j^1,\ j \in E_1\right\}$, that provides the optimal compression $\hat{R}^1$ of R^1, is constructed from elements of the library $\mathcal{L}_1$:

$$R^1 = \hat{R}^1 + R^2 \qquad \text{with} \qquad \hat{R}^1 = \sum_{j \in E_1} q_j^1 \Phi_j^1 \tag{2}$$

where $\left\{q_j^1,\right\}_{j \in E_1}$ are the quantized coefficients, and E_1 is the set of indices of the basis functions that constitute the best basis. We now reconstruct a second approximation $\hat{I}^1$ of I

$$\hat{I}^1 = \hat{R}^0 + \hat{R}^1 \tag{3}$$

where $\hat{I}^1$ is an image that can be encoded with $b_0 + b_1$ bits.

2.1.2 Main loop of the algorithm

Figure 1 shows the main loop of the algorithm. Let us assume that we have carried the approximation of I up to step $n-1$. Let R^n be the residual of the approximation at step $n-1$. A best basis, $\{\Phi_k\}_{k \in E_n}$, that provides the optimal approximation $\hat{R}^n$ of R^n with b_n bits, is constructed from the library $\mathcal{L}_n$:

$$R^n = \hat{R}^n + R^{n+1} \qquad \text{with} \qquad \hat{R}^n = \sum_{j \in E_n} q_j^n \Phi_j^n. \tag{4}$$

where E_n is the set of indices of the basis functions that constitute the best basis. Finally, we reconstruct an approximation of I using the $n+1$ compressed residual images $\hat{R}^0, \hat{R}^1, \hat{R}^2, \ldots$

$$\hat{I}^n = \sum_{k=0}^{n} \sum_{j \in E_k} q_j^k \Phi_j^k \tag{5}$$

where $\hat{I}^n$ is an image that can be compressed with a budget of $\sum_{k=0}^{n} b_k$ bits. The coefficients q_j^k of the non-linear approximation (5) are not the inner products with the basis functions ; they are the quantized versions of the inner products with the basis functions.

2.2 Completion of the algorithm

We stop the iteration when the residual is noisy and does not contain any apparent structure. This noisy part constitutes the incoherent noise that was present in the image. At this point no more structure can be pulled out of the residual image.This can be made explicit by looking at the statistical distribution of the coefficients of the expansion, as explained in the next section. After the final compression, one might still want to be able to save the final residual. This happens for legal reasons, or simply in order to be able to come back to the raw data. This functionality is fully available in the multi-layer algorithm. The final residual can be entropy coded. Because during each iteration the amplitude of the final residual decreases, the amplitude of the residual is usually small.

2.3 Toy example

Figures 2,3 and 4 illustrate the algorithm. Figure 2-left shows the original image *Barbara*. The first iteration of the algorithm is shown in Figure 2-right, where the smooth regions of the image are described with a few wavelet coefficients (compression ratio= 254). The remainder, or residual $R^1 = I - \hat{R}^0$, is shown in Figure 3-left. R^1 contains edges, as well as textured regions. $\hat{R}^0$ constitutes the first part of the multi-layered image. During the second pass of the algorithm, we compress the residual R^1 with an adapted local cosine basis (compression factor = 36.61). Figure 3-right shows the optimal geometry of the LCT basis adapted to the residual R^1. The result of the compression, $\hat{R}^1$, is shown in Figure 4-left, and the second residual $R^2 = R^1 - \hat{R}^1$, is shown on the right. $\hat{R}^1$ constitutes the second layer. Clearly, most of the textures, and most of the edges have been removed from the image, and are encoded into this second layer $\hat{R}^1$. Most structures have been encoded in the first two layers, and the final residual R^2 appears as random noise. When $\hat{R}^0$ and $\hat{R}^1$ are added together, we obtain an image which is compressed by a factor of 32 (see Figure 15-right).

Figure 2: Left: original image. Right: the first layer $\hat{R}^0$. The first layer is a very sparse representation of the smooth trends in the image; it was obtained by compressing the original image in a wavelet basis by a factor of 254.

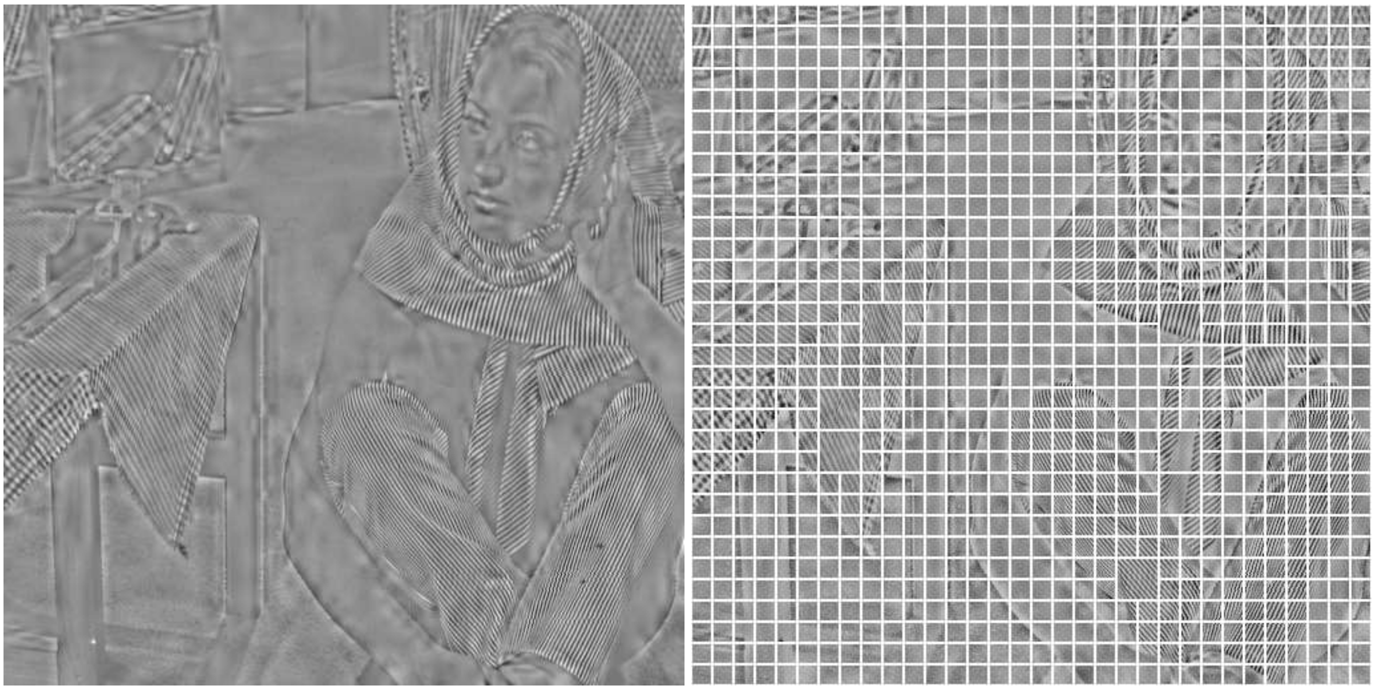

Figure 3: Left: the first residual part R^1. Right: optimal geometry of the LCT basis adapted to the residual R^1.

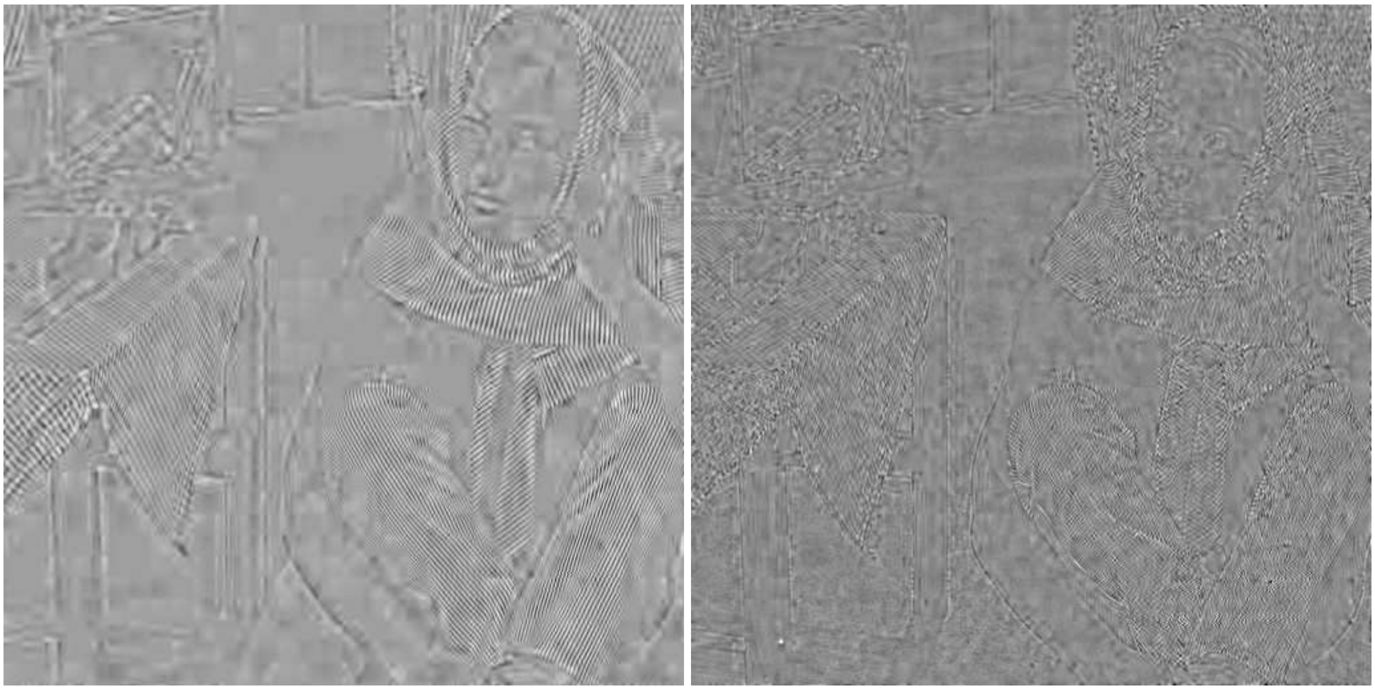

Figure 4: Left: the second layer $\hat{R}^1$. The second layer is a sparse representation of the textures in the image; it was obtained by compressing the first residual R^1 by a factor of 36.61, with an adapted local cosine basis (see Figure 3-right for the geometry of the best basis). Right: the second residual part R^2.

2.3.1 Textural layers should be coded with adapted local cosine bases or wavelet packets

Visual inspection of the residual image R^1 in Figure 3-right suggests that this textural layer should be coded with two-dimensional oscillatory patches, such as 2-D local trigonometric bases, or 2-D wavelet packets. In order to corroborate this visual and geometric intuition, we estimated the s-tatistical distribution of the wavelet coefficients of R^1, and the statistical distribution of the cosine coefficients of the same image R^1. Figure 5-left shows the modulus of the wavelet coefficients of: (i) the original image I, (ii) the wavelet-compressed image $\hat{R}^0$, (ii) and the residual R^1. The coefficients were sorted by decreasing order of magnitude, and only the first 65,536 coefficients are displayed. Figure 5-right compares the two different expansions of the residual R^1: (1) the best LCT coefficients, and (2) the wavelet coefficients. As Figure 5-right shows, the decay of the LCT coefficients is much more rapid, indicating that the projection on this basis is more interesting. Indeed, with less coefficients the LCT expansion captures more features (has a stronger correlation) than the wavelet expansion. In general, a flat distribution indicates that the basis is "uninteresting" because the energy of the image is spread across all frequencies in a way similar

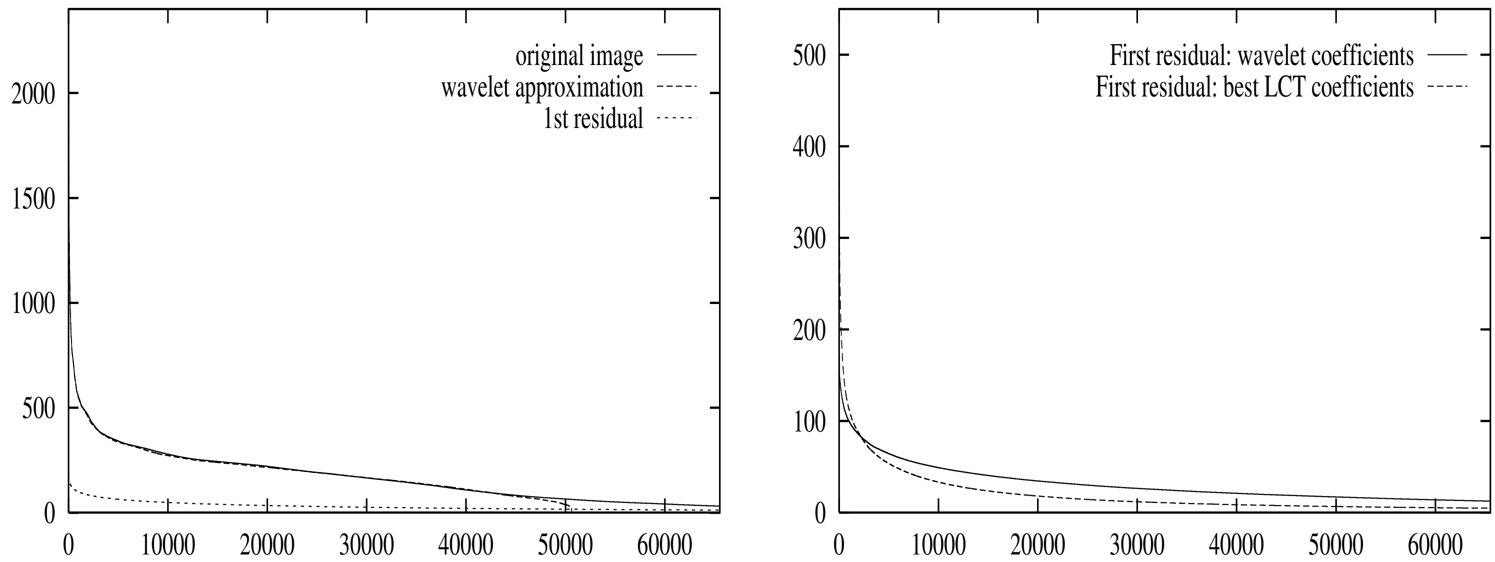

Figure 5: Left: modulus of the wavelet coefficients of: (i) the original image I, (ii) the wavelet-compressed image $\hat{R}^0$, (ii) and the residual R^1. The coefficients were sorted by decreasing order of magnitude, and only the first 65,536 coefficients are displayed. Right: modulus of the coefficients of R^1 in the best LCT basis, and modulus of the wavelet coefficients of the same image R^1 (only the first 65,536 coefficients are displayed).

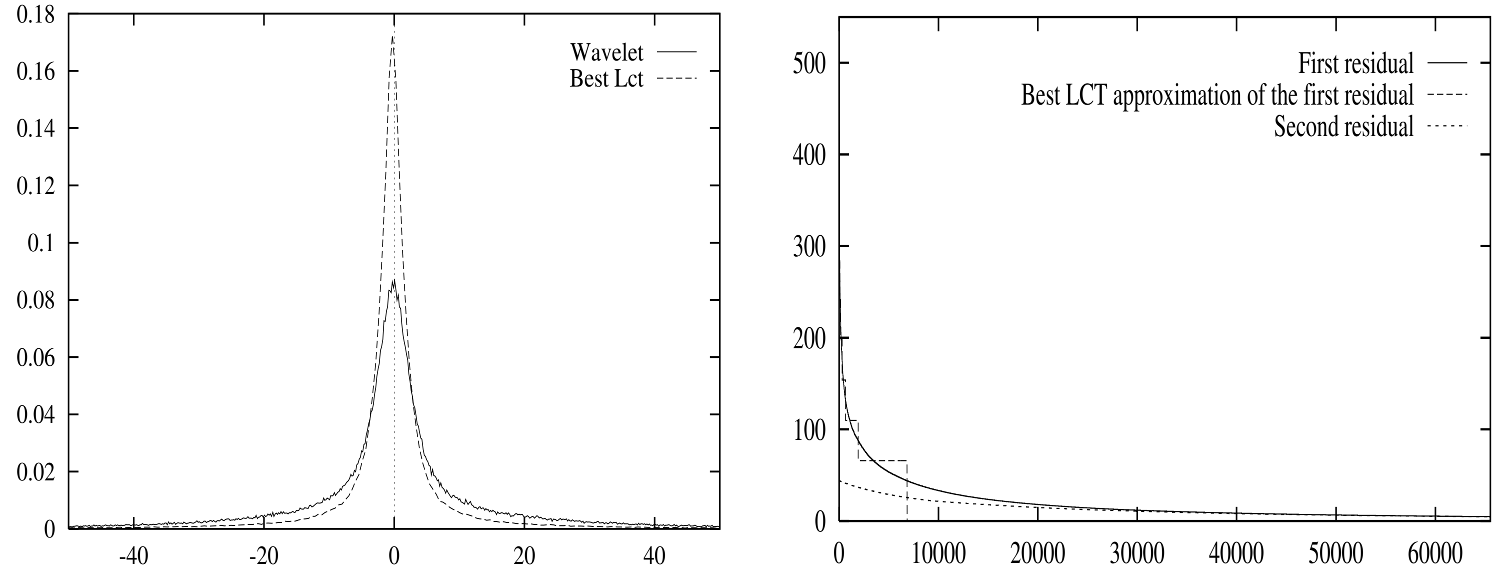

Figure 6: Left: the histogram of the coefficients of R^1 in the best LCT basis, and the histogram of the wavelet coefficients of the same image R^1. Right: modulus of the coefficients of the two layers, $\hat{R}^0$, $\hat{R}^1$, and the residual R^2. Only the first 65,536 coefficients are displayed. The distributions become more and more flat, indicating that the final layer does not contain any structure.

to white noise (wavelet coefficients, as well as LCT coefficients characterize the frequency content of the image). In order to characterize the shape of the probability density function of the distribution of the wavelet and cosine coefficients, we constructed estimates of the two distributions using histograms. Figure 6-left shows the histograms of the LCT coefficients, and the wavelet coefficients of the same image R^1. The variance of the distribution of the LCT coefficients is much smaller, and therefore the LCT coefficients will be easier to compress than the wavelet coefficients. Figure 6-right shows the modulus of the coefficients of the two layers, $\hat{R}^0$, $\hat{R}^1$, and the final residual R^2. Only the first 65,536 coefficients are displayed. The distribution of the coefficients becomes more and more flat, indicating that there are less and less features inside each layer. Another example of a multi-layered representation is shown in Figures 7, and 8. The original image, shown in Figure 7-left, contains a mixture of very flat, and highly textural regions. The processing was performed on the R,G,B components separately (no effort was made to take advantage of the YIQ coordinate system). The first layer, $\hat{R}^0$, was obtained with a wavelet compression (ratio = 150) of each R,G,B component. The first residual, R^1, (not shown here) contains textural information present in the image. The second layer, $\hat{R}^1$, shown in Figure 8-left, was obtained by compressing (ratio = 10) the first residual using an adapted brushlet basis [22]. Brushlets capture the oscillating patterns of the textures at many different orientations. The final residual, R^2, is shown in Figure 8-right. Because all smooth regions (that describe the general shape of the mandrill), and textures (that describe the fur and the whiskers) have been previously extracted in the first two layers, the second residual contains only structures that are "dot-like". This example demonstrates very clearly how an image is decomposed as a superposition of coherent layers. Within a given layer all the features have a similar scale, and similar smoothness properties, etc.

3 Libraries of bases

One of the key tenet of the multi-layered representation is a mechanism to obtain a specialized, or tailored basis $\mathcal{B}_i = \{\Phi_\gamma, \gamma \in G_i\}$ to encode each layer R^i:

$$R^i(x, y) = \sum_{\gamma \in G_i} \alpha_\gamma \Phi_\gamma(x, y). \tag{6}$$

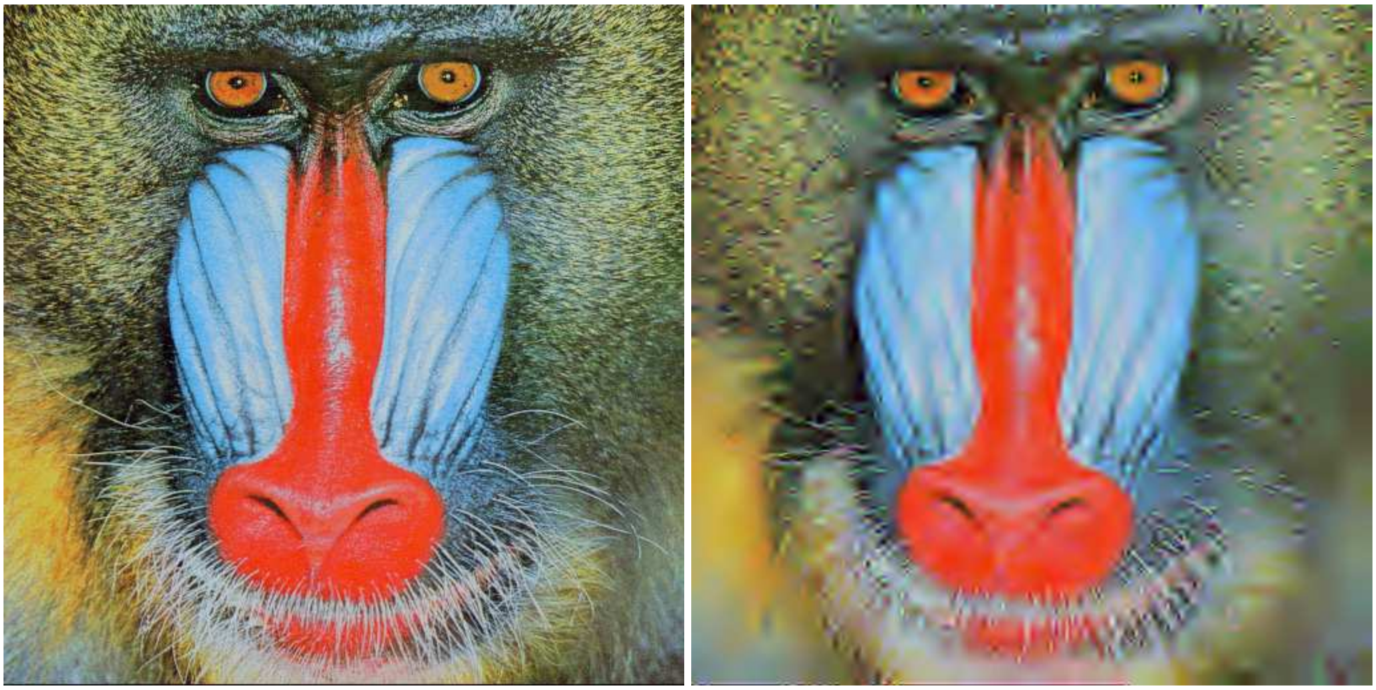

Figure 7: Left: original image. Right: the first layer $\hat{R}^0$ is a very sparse representation of the shape of the mandrill, and of the smooth trends in the image. Color information is almost entirely coded in this layer. This first layer was obtained by compressing each of the R,G,B components by a factor of 150, using a wavelet basis. All textures have been removed.

Figure 8: Left: the second layer $\hat{R}^1$ is a sparse representation of the textures in the image. It was obtained by compressing by a factor of 10 the first residual R^1 using an adapted brushlet basis. Right: final residual R^2.

$\mathcal{B}_i$ is "adapted" to R^i in the sense that each function in $\mathcal{B}_i$ is chosen from a very large, and highly redundant library of waveforms $\mathcal{L}_i = \{\Phi_\gamma,\ \gamma \in \Gamma\}$. The library $\mathcal{L}_i$ is *overcomplete* because the number of basis functions Φ_γ is much greater than the effective dimension of the input space. The representation in the library is thus not unique, and this redundancy allows us to choose among many possible bases that basis for which the coding of the coefficients α_γ requires the least number of bits. We describe in this section two examples of such libraries:

- **local trigonometric functions** [3]. Local trigonometric transforms provide an adaptive segmentation of the spatial domain in terms of oscillating patterns. An image is decomposed into blocks of different sizes within which a local Fourier expansion, or a DCT, is performed. Instead of abruptly cutting blocks in the image, we use a new family of *smooth orthogonal projectors* [3, 9, 29].

- **wavelet packets** [10]. Loosely speaking, wavelet packets make it possible to adaptively tile the frequency domain into different bands of arbitrary size. Wavelet packets have been used to characterize textures in images [6, 16, 20, 24]. However, an elementary 2-D wavelet packet always displays a "criss-cross" pattern, which comes from its two symmetric peaks in the Fourier domain. As a result one needs a combination of several wavelet packets to characterize a single 2-D pattern oscillating along one direction [22].

Another example of library is the collections of

- **brushlets** [22]. Brushlets are new families of steerable wavelet packets that adaptively segment the Fourier plane to obtain the most concise and precise representation of the image in terms of oriented textures with all possible directions, frequencies, and locations.

Because each library is overcomplete, it is possible to obtain a very sparse representation of an image by specializing, or "tailoring", its representation. Even though one could work with other overcomplete representations that do not necessarily contain orthogonal bases [7, 18], the libraries of orthonormal bases offers many advantages: (i) they provide very large sub-collection of orthogonal bases, (ii) in an orthogonal basis the decomposition, and the reconstruction can be performed using very fast algorithms, which are numerically exact and stable [10], (iii) there exist some fast algorithms that can be applied in real time, to select the optimal decomposition over the library

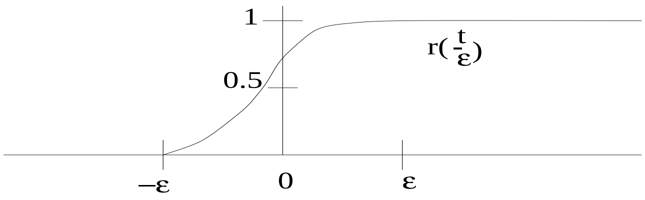

Figure 9: Ramp function r.

[11]. We describe in the following two important examples of libraries of bases: the library of smooth local cosine bases, and the wavelet packet library.

3.1 Adaptive smooth local cosine transforms

The main defect of windowed Fourier based compression methods (such as the DCT) is due to the limitation put on the size of the blocks, and the inability to adapt the patterns to the content of the picture. In order to address this problem, we adaptively select the size of the block according to the local content of the image. Another defect becomes apparent when reaching high compression rates: the blocks used for compression start to be visible in the reconstructed picture. An answer to this problems is to replace the characteristic function of each interval, that abruptly cuts off a block in the image, by a new family of smooth projections. Combining these two ideas result in a new library of trigonometric bases, that can be tailored to any image. First, we review the construction of smooth Localized Cosine Transforms (LCT) [3, 9, 11] in one dimension. We consider a cover of $\mathbb{R} = \bigcup_{n=-\infty}^{n=+\infty} [a_n, a_{n+1}[$, and we write $l_n = a_{n+1} - a_n$, and $c_n = (a_n + a_{n+1})/2$. We then perform a local Fourier analysis inside each block. To obtain a better frequency localization, we do not cut abruptly the signal, but we use a smooth window function to localize the segment of interest. Around each a_n we define a neighborhood of radius ε, and a ramp function, r (see Figure 9), such that

$$r(t) = \begin{cases} 0 & \text{if} \quad t \leq -1 \\ 1 & \text{if} \quad t \geq 1 \end{cases} \qquad \text{and} \qquad r^2(t) + r^2(-t) = 1, \qquad \forall t \in \mathbb{R} \tag{7}$$

We define the smooth orthogonal projection $P_{I_n} f$ of a one-dimensional signal f as follows [3, 9]:

$$P_{I_n} f(x) = \begin{cases} r^2(x-a_n) f(x) + r(x-a_n) r(a_n - x) f(2a_n - x) & \text{if} \quad a_n - \varepsilon \le x < a_n + \varepsilon \\ f(x) \quad \text{if} \quad a_n + \varepsilon \le x < a_{n+1} - \varepsilon \\ r^2(a_{n+1} - x) f(x) - r(x - a_{n+1}) r(a_{n+1} - x) f(2a_{n+1} - x) & \text{if} \quad a_{n+1} - \varepsilon \le x < a_{n+1} + \varepsilon \end{cases} \tag{8}$$

The projection of contiguous intervals is equal to the sum of the projections [3, 9]:

$$P_{I_n} \oplus P_{I_{n+1}} = P_{I_n \cup I_{n+1}}, \tag{9}$$

and this beautiful result permits to decompose $L^2(\mathbb{R})$ as follows:

$$L^2(\mathbb{R}) = \bigoplus_{n \in \mathbb{Z}} P_{I_n} L^2(\mathbb{R}) \tag{10}$$

Furthermore, the projection is as smooth as the original function [3]. We consider the collection of cosine functions:

$$c_{j,n} = \cos\left[\frac{2j+1}{2} \frac{\pi}{|I_n|}(x - a_n)\right] \tag{11}$$

In their discrete form these functions correspond to the DCT-IV:

$$X(j) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)(2j+1)\pi}{4N}\right] \tag{12}$$

We multiply the $c_{j,n}$ by the ramps, on each side of I_n, and we obtain a basis of $P_{I_n} L^2(\mathbb{R})$ [3]:

$$w_{j,n} = \sqrt{\frac{2}{|I_n|}}\, r(x - a_n)\, r(a_{n+1} - x) \cos\left[\frac{2j+1}{2} \frac{\pi}{|I_n|}(x - a_n)\right] \tag{13}$$

3.1.1 Implementation by folding

In practice, in order to expand a function f into the basis $\{w_{j,n}\}$ we do not calculate the correlation between f and the basis functions $w_{j,n}$. Instead we transform f restricted to $[a_n - \varepsilon, a_{n+1} + \varepsilon]$ into a smooth function onto $[a_n, a_{n+1}]$ with the adequate polarity (even around a_n, and odd around a_{n+1}), and we expand it into the basis $\{c_{j,n}\}$. To do this we fold the overlapping parts of the

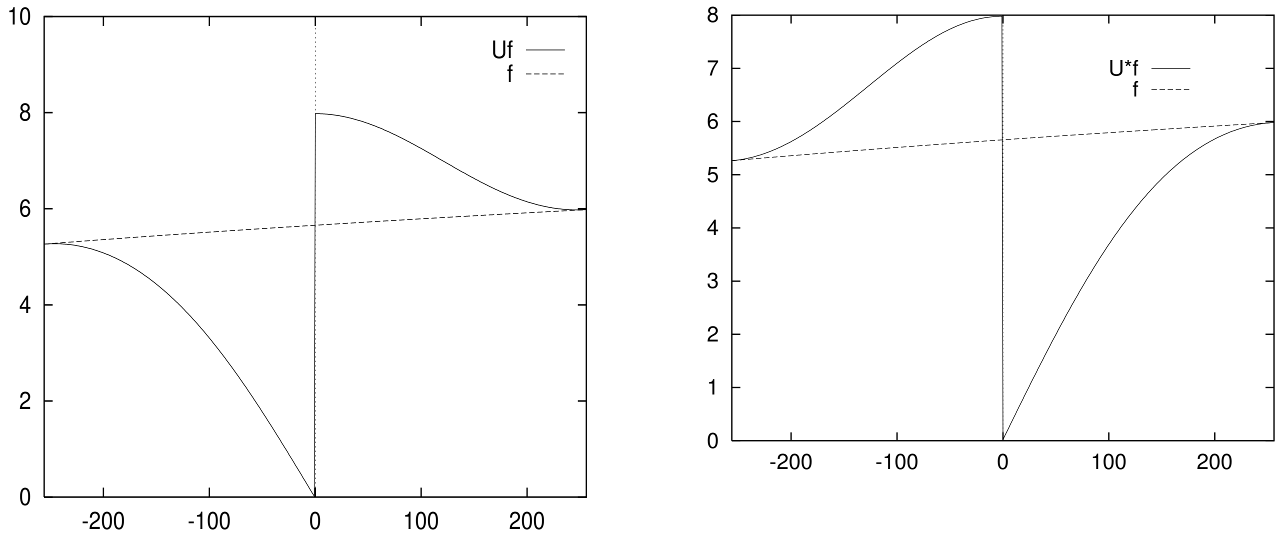

Figure 10: Left: result of the folding operator. Right: result of the unfolding operator.

ramps $r(x - a_n)$ and $r(a_{n+1} - x)$ back into the interval, across the endpoints of the interval, with some folding and unfolding operators. The advantage of the procedure is that we can pre-process the data with the folding operators, and then use a fast DCT to calculate the expansion into the basis $\{c_{j,n}\}$.

Unitary folding and unfolding

We define the unitary folding operator U_{a_n} and its adjoint the unfolding operator $U^*_{a_n}$ as follows:

$$U_{a_n} f(t) = \begin{cases} r(\frac{a_n - t}{\varepsilon}) f(t) - r(\frac{t - a_n}{\varepsilon}) f(2a_n - t), & \text{if} \quad a_n - \varepsilon < t < a_n, \\ r(\frac{t - a_n}{\varepsilon}) f(t) + r(\frac{a_n - t}{\varepsilon}) f(2a_n - t), & \text{if} \quad a_n < t < a_n + \varepsilon, \\ f(t), & \text{otherwise;} \end{cases} \tag{14}$$

$$U^*_{a_n} f(t) = \begin{cases} r(\frac{a_n - t}{\varepsilon}) f(t) + r(\frac{t - a_n}{\varepsilon}) f(2a_n - t), & \text{if} \quad a_n - \varepsilon < t < a_n, \\ r(\frac{t - a_n}{\varepsilon}) f(t) - r(\frac{a_n - t}{\varepsilon}) f(2a_n - t), & \text{if} \quad a_n < t < a_n + \varepsilon, \\ f(t), & \text{otherwise.} \end{cases} \tag{15}$$

Figure 10 shows the result of the folding and unfolding operators. The algorithm to expand a function f into the basis $w_{j,n}$ is thus:

1. Apply the transform U_{a_n} to f, apply $U_{a_{n+1}}$ to $U_{a_n} f$;

2. Expand $U_{a_{n+1}} U_{a_n} f$, which is now living on $[a_n, a_{n+1}]$, into the DCT IV basis $\{c_{j,n}\}$.

The reconstruction algorithm that recovers the function f is:

1. Apply an inverse DCT-IV to the coefficients. This gives us $U_{a_{n+1}} U_{a_n} f$, which is living on $[a_n, a_{n+1}]$.

2. Apply $U^*_{a_n}$, apply $U^*_{a_{n+1}}$: this gives us $P_{I_n} f$

3. Add the contiguous overlapping components $P_{I_n} f$. This gives f back.

3.1.2 Choice of the bell function

If one is ready to relax condition (7), then one can extend the choice of the bell functions, and improve the quality of the reconstructed image [1]. In this setting we have one bell function for the decomposition, or analysis, and one bell function for the reconstruction, or synthesis. In other words we construct biorthogonal bases of smooth localized cosine transforms [19, 22]. We use the bells of Matviyenko [19], that were designed to efficiently compress trigonometric functions.

3.1.3 Adaptive segmentation

We adaptively select the size and location of the windows $[a_n, a_{n+1})$. We divide the support of f into two intervals, and we consider the local cosine basis associated with each interval. We then further decompose each interval into two sub-intervals, and consider the local cosine bases associated with this finer subdivision. By applying this decomposition recursively we obtain a homogeneous binary tree-structured decomposition as shown in Figure 11-left. For each interval, or node of the binary tree, we calculate the set of coefficients in the subblock. In two dimensions we define two partitions of $\mathbb{R}$, $\bigcup_{n=-\infty}^{n=+\infty}[a_n, a_{n+1}[$, and $\bigcup_{k=-\infty}^{k=+\infty}[b_m, b_{m+1}[$. We then consider the tiling obtained by the lattice cubes $[a_n, a_{n+1}[\otimes[b_m, b_{m+1}[$. We consider the separable tensor products of the bases $w_{n,k}$, and $w_{m,j}$. The sequence $w_{n,k} \otimes w_{m,j}$ is an orthonormal basis for $L^2(\mathbb{R}^2)$. As in the one dimensional case we can build a tree decomposition of an image. We divide the image into four sub-squares, and we consider the local cosine basis associated with this tiling. We then further decompose each square into four sub-squares, and consider the local cosine bases associated with this finer tiling. By applying this decomposition recursively we obtain a homogeneous quadtree-

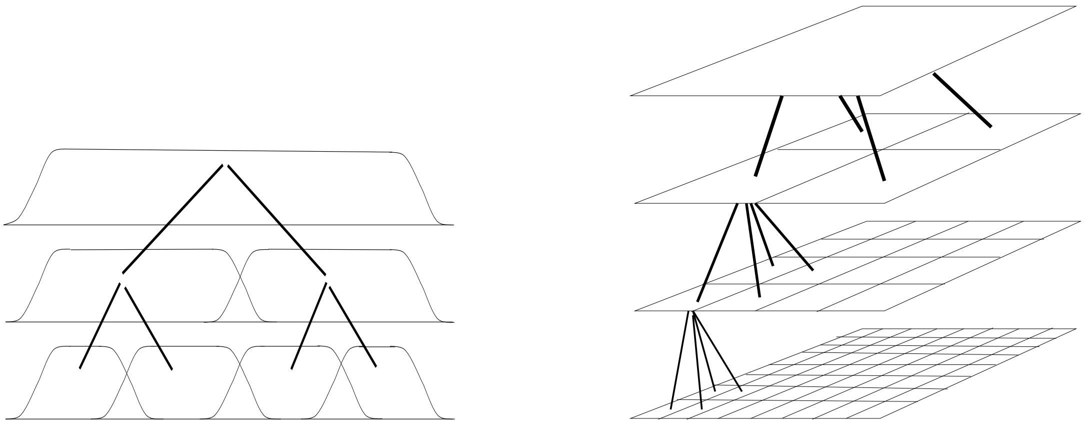

Figure 11: Left: binary tree. Right: quadtree. Within each block the signal or the image is expanded into a local DCT. The tree is then pruned in order to keep those coefficients that require the least number of bits to code.

structured decomposition as shown in figure 11-right. Anisotropic segmentations can also be obtained as explained in [4].

3.2 The wavelet packet library

3.2.1 Wavelet packets in one dimension

We first describe the wavelet packet library in one dimension. We present next a fast algorithm to calculate the wavelet packet coefficients. We consider two conjugate quadrature filters, the low pass filter $\{h_n\}$, and the high pass filter $\{g_n\}$. We have $g_n = (-1)^n \bar{h}_{1-n}$, and they satisfy

$$2\sum_{n\in\mathbb{Z}} h_n h_{n+2k} = \delta_{0,k} \quad \text{and} \quad \sum h_n = 1 \tag{16}$$

in such a way that we have perfect reconstruction. We consider the discrete signal $f = \{f_n\}$, $n = 0, \dots, N-1$. The wavelet packet coefficients $w_{n,\,j,\,l}$ are defined by the following recursions:

$$w_{0,0,k} = f_k \tag{17}$$

$$w_{2n,j,l} = \sum_k h_{k-2l} w_{n,j+1,k} \tag{18}$$

$$w_{2n+1,j,l} = \sum_k g_{k-2l} w_{n,j+1,k}. \tag{19}$$

The indices are interpreted as follows:

- j is the scaling index: the size of the support of the corresponding wavelet packet is approx-

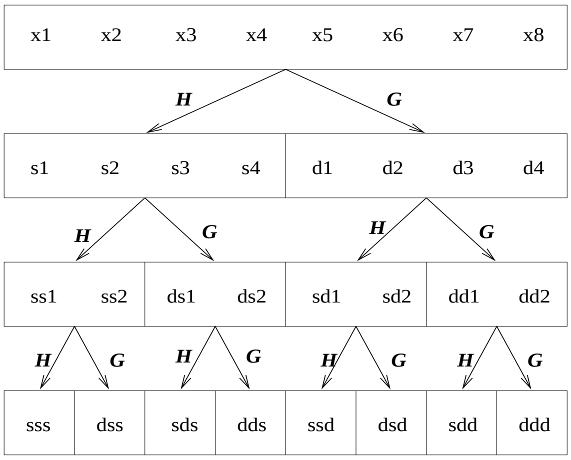

Figure 12: Wavelet packet tree. At each node of the tree, we apply a convolution and a decimation with the lowpass filter H, and the highpass filter, G. The prefix "s" stands for the sum, or lowpass filter ; and "d" stands for difference, or highpass filter.

imatively 2^{-j},

- l is the localization parameter: the corresponding wavelet packet is roughly located at $l\,2^{-j}$,
- n is the frequency index: the wavelet packet has approximatively n oscillations (a frequency $2^j\,n$).

Similarly to the LCT library, the wavelet packet library organizes itself into a binary tree, as shown in Figure 12, where the nodes of the tree represent subspaces with different time-frequency localization characteristics. The wavelet basis is obtained by iterating the decomposition process on the low frequency bands only, without further decomposing the high frequency component at each level of the tree. Clearly the wavelet packet library provides an over-complete description of the signal f. We need to know how to assemble the elements of the library to obtain an orthogonal basis. Loosely speaking, wavelet packets make it possible to adaptively tile the frequency domain into different bands of arbitrary size. Although the situation is slightly more complex, we can say that if a collection of functions in the library provides a tile of the time-frequency plane, then this set of functions is an orthonormal basis:

Theorem 1 *[10] If a subset $E \subset \mathbb{N} \times \mathbb{Z}$ has the property that the union of intervals $[2^j n, 2^j(n+1)\ )$, $j \in \mathbb{N},\ 0 \le n < 2^j$, $(n,j) \in E$, is a disjoint cover of $[0,\infty)$, then the set of wavelet packet coefficients $w_{n,\,j,\,l},\quad (n,j) \in E$ are the coefficients of f in an orthonormal basis.*

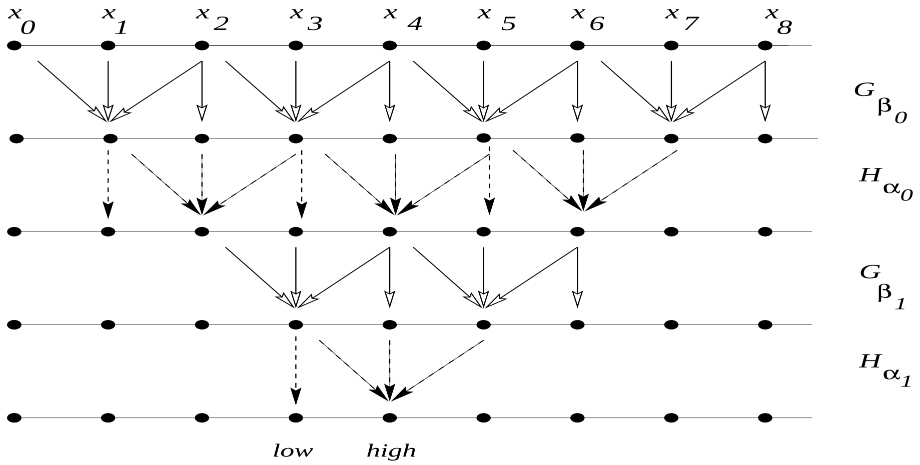

Figure 13: Three layers of 3-1 filters from which the 9-7 low-high filters are constructed

3.2.2 Fast convolution: 2-D factorization

The number of operations needed to compute the low-pass and high pass coefficients are reduced significantly by factoring the lowpass and highpass filters into smaller filters. Instead of using biorthogonal filters of size $(2m+1, 2m-1)$ for each point, we iteratively apply a ladder of m filters of size $(3, 1)$ (see Figure 13). In one dimension this method reduces the number of multiplications and additions by a factor of about 2. In 2-D the factorization reduces the number of multiplications and additions by a factor of 4 [20]. For all experiments we have used the factorized version of the biorthogonal filters 9-7 [8]. More details about the factorization are in [20, 21].

3.3 Best basis algorithm

Both the wavelet packets, and the local cosine transforms, come in large redundant libraries, and certain special sub-collections in these libraries amount to orthogonal bases. The heart of the matter is then the selection of a subset of basis functions $\mathcal{B} = \{\Phi_\gamma, \gamma \in G\}$ of $\mathcal{L}$ such that the expansion (6) is optimal for a criterion Q. Such a subset is called the *best adapted basis* [11, 29]. The best basis is expressed as the solution of the following optimization problem:

$$\max_{\mathcal{B} \subset \mathcal{L}} Q(\mathcal{B} = \{\Phi_\gamma,\ \gamma \in G\}) \quad \text{subject to} \sum_{\gamma \in G} \alpha_\gamma \Phi_\gamma = R^i \tag{20}$$

At first sight, the numerical solution of (20) appears to be non trivial: a constrained nonlinear minimization problem, with possibly many local minima. However, as explained in the previous sections, a beautiful property of the wavelet packet and local cosine libraries is that they are organized in a hierarchical fashion, and the problem (20) can then be solved with fast algorithms [11]. The "best basis" paradigm permits a rapid (order $N \log(N)$, where N is the number of pixels in the image) search among the large collection of orthogonal bases to find that basis which permits the best approximation for a given budget. The best-basis which minimizes this criterion is searched in this binary tree using a "divide and conquer" algorithm: at each node, the cost is compared with the cost of the union of its two children's nodes and if the node's cost is smaller than the children's costs, the node is retained; otherwise, the children nodes are retained instead of the node itself. This process is recursively applied from the bottom to the top of the tree. Ramchandran and Vetterli [24] have proposed to select the best basis according to the rate distortion criterion. Each node of the wavelet packet tree is associated with the best scalar quantizer for that node. Then the best basis and the best set of quantizers are obtained using a pruning algorithm. The pruning procedure needs to be iterated several times to find the optimal slope on the rate distortion curve, at which all the quantizers of the best basis operate. We note that the results published in [24] correspond to hypothetical compression rates, since the first order entropy was chosen to measure the rate. In principle the approach of [24] is optimal for scalar quantizers. In practice, their approach is computationally intensive since it requires to search for the best basis several times in order to find the optimal slope on the rate distortion curve.

Our approach is much faster, and requires only one single pruning of the wavelet packet tree. Firstly, we de-noise the image: we threshold the coefficients in the tree to remove those coefficients whose magnitude are below a given threshold. The threshold is defined as the amplitude of the smallest non-zero coefficients that can be reconstructed after inverse quantization. Discarding small coefficients permits to choose the best basis from the set of coefficients that will really contribute to the reconstruction of the image. Secondly, we measure the compactness of a basis with the first-order entropy. For each node of the tree, we calculate an histogram of the coefficients. This provides us with an approximation $\{p_i\}$ of the probability density function of the coefficients. The cost of the node is defined as: $-\sum_i p_i \log(p_i)$. We have tried several other cost measures. After de-noising, the first order entropy provides a very good measure of the overall budget required to encode the coefficients. Another excellent cost measure, which is faster to calculate, is defined as follows. Let λ be a given threshold, of the same order of magnitude as the quantization step, the cost of the node is the number of coefficients larger than λ.

4 Quantization

4.1 Frequency ordering of the coefficients

We explain here how to organize, and quantize the coefficients of the residual R^i, for each iteration i of the algorithm. We use the same ordering, and the same quantization methods for wavelet packets, and local trigonometric libraries. In the case of a wavelet basis several authors have exploited the multiscale structure of the wavelet coefficients to describe with quadtrees large regions where the quantized coefficients are equal to zero. Partitioning the coefficients into subtrees of significant, and insignificant coefficients, provides an extremely powerful description of the wavelet coefficients [12, 15, 26, 27, 31]. Such partitioning techniques take full advantage of the self similar structure of natural images across scales [13]. An adapted local cosine or wavelet packet basis does not have a multiscale structure, and therefore one cannot exploit the powerful zero-tree techniques. Nevertheless, we organize the coefficients in such a way that we take advantage of the expected decay of the coefficients. Our scanning method exploits the fact that if an image is smooth, then the amplitude of the coefficients decreases as the frequency of the basis function increases. This result is certainly true if we use local cosine bases. In the case of wavelet packet we have the following result [20, 21]:

Lemma 1 *[20] If f is a C^r regular function, then $\exists C > 0$ such that the wavelet packet coefficients of f, $w_{n,\,j,\,l}$, satisfy*

$$\forall q \geq 0, \quad \forall n = 2^q, \dots, 2^{q+1} - 1 \qquad |w_{n,\,j,\,l}| \leq C\; 2^{-q/2}\; 2^{-r(q+j)} \tag{21}$$

This bound on the coefficient tells us two things. On the one hand, as the scale j increases, the wavelet packet coefficients have a geometric decay 2^{-rj} — a decay similar to the wavelet coefficients. On the other hand, as the frequency $\nu = 2^j n$ increases, the wavelet packet coefficients decay faster than ν^{-r} — a decay similar to the decay of Fourier coefficients. This result tells us that we should organize the coefficients by increasing frequency. The organization of the coefficients is defined as follows:

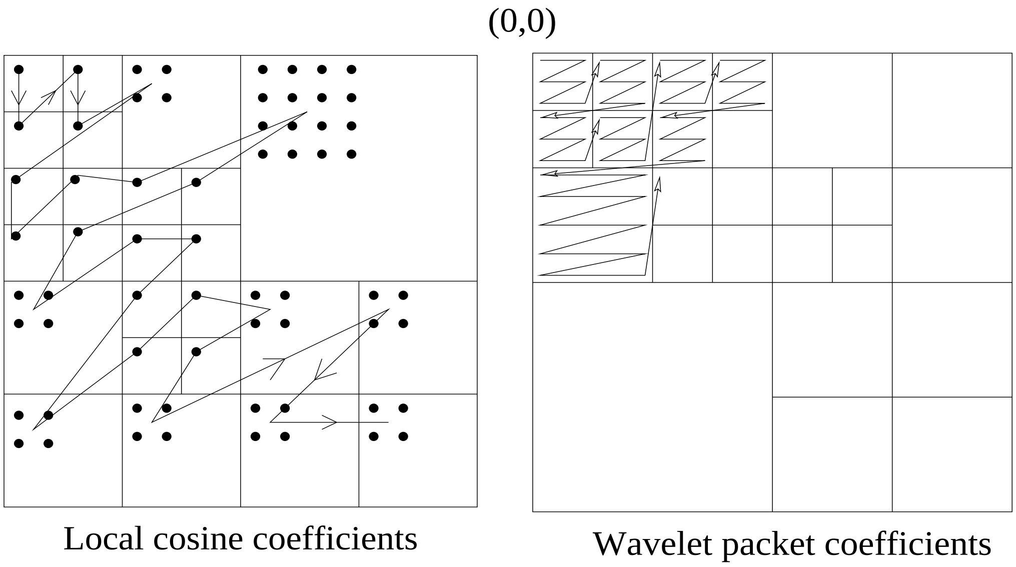

Figure 14: Left: ordering of the LCT coefficients. We visit all the blocks, and pick in each block all those coefficients that have a similar frequency (because the blocks may have different size, we may take more than one coefficient per block). We start with the smallest frequency (DC coefficients), and we continue until the largest frequency. Right: ordering of the Wavelet packets. We organize the coefficients in the natural frequency order and not in the Paley order, and we send them in this order.

- Local cosine basis: instead of sending all the coefficients of one DCT block after another (as is done for JPEG), we visit all the blocks, and pick in each block all those coefficients that have a similar frequency (because the blocks may have different size, we may take more than one coefficient per block). We start with the smallest frequency (DC coefficients), and we continue until the largest frequency. As shown in Figure 14-left, this scheme requires visiting the blocks several times (as many times as the size of the smallest block). However the generated stream of coefficient is fully scalable: the decoder can decode only a smaller set of coefficients, and reconstruct a meaningful image:
  - at a reduced spatial resolution (spatial scalability);
  - at a reduced quality (quality scalability).
- Wavelet packets: we organize the coefficients in the natural frequency order and not in the Paley order [11], and we send them in this order (see Figure 14-right). Again we create a fully quality scalable stream of coefficients.

4.2 Scalar quantization, and bitplane encoding

Once the coefficients are ordered, we quantize them with a uniform scalar quantizer. The quantizer step is the same for all coefficients. Once the coefficients are quantized, a stream of bits is generated by bitplane encoding the absolute value of the quantized coefficients. If a coefficient is not quantized to zero, then its sign is also encoded. The bit stream contains long stretches of zeros and thus we use a zero-runlength coder to encode the stream. The parameters of the quantization are optimized in order to reach an exact budget. The uniform quantizer is characterized by two parameters: Δ the cell size, and $\Delta\, 2^{-l}$ the radius of the dead-zone around zero. The bitplane encoding consists in transmitting the digits 0 and 1 of the binary representation of each quantized coefficient. The bitplane encoding is characterized by two parameters: l the index of the smallest bit encoded, and m the minimum number of planes needed to code the absolute value of the largest coefficients. A coefficient x is quantized into $q(x)$ using the following rule:

- if $|x| < \Delta\; 2^l$ then $q(x) = 0$
- if $|x| \geq \Delta\; 2^l$ then we code the sign of x, and we define the quantized coefficient $q(x)$ as follows:

$$|x| = \Delta\; q(x) + r(x) \quad \text{with} \quad 0 \leq r(x) < \Delta\; 2^l \tag{22}$$

In order to reach the targeted budget, we optimize the value of Δ, and l, the two parameters of the quantization. The sequence of 0 and 1 generated by the bitplane encoding is then encoded using a runlength coding technique. The best basis geometry is described by a quadtree, which is entropy coded with an adaptive arithmetic coder [30]. The number of bytes required to encode the geometry is always a small fraction of the total budget.

5 Experiments

We have implemented the coder and decoder, and an actual bit stream is generated by the coder. For each experiment we generated a compressed file with a size equal to the targeted budget. We present the results of the multi-layer compression algorithm, using the following test images: 512×512 8-bbp "Barbara", and 512×512 8-bpp "Houses". These images are difficult to compress because they contain a mixture of large smooth regions, and long oscillatory patterns. In order to evaluate the performance of our algorithm, we compared it to one of the best wavelet coder that was available to us: the SPIHT wavelet coder of Amir Said and William A. Pearlman [26]. A comparison with other wavelet coders (e.g. [27, 28, 31]) would result in different but comparable results. The performance of the algorithm is summarized in Table 1. We work with 8 bit images,

Figure 15: Barbara, compression ratio: 32. Left: SPIHT (wavelet) . Right: Multi-layer.

and we define the Peak Signal to Noise Ratio (PSNR) of the compressed image I_c as $PSNR = 10\ \log_{10} \frac{255^2}{\frac{1}{N^2}\sum_{i,j=0}^{N-1}|I(i,j)-I_c(i,j)|^2}$.

Barbara

Figure2-left shows the original image Barbara. Figure15-left shows the result of a compression of 32 using SPIHT, and on the right is the result of the multi-layer coder for the same compression ratio. In this example we used two layers: a first compression with wavelets, and a second compression with an adapted local cosine basis. To better evaluate the visual quality of the compression, we have magnified a detail of the image: the right leg of Barbara. This detailed view is shown in Figure 16. Clearly the texture on the pants of Barbara is very well preserved with the multi-layered representation. As Figure 15 shows, the textures on the tablecloth and on the chair are entirely preserved. Furthermore, the multi-layer coder does not introduce any unpleas-

Figure 16: Detail on the right leg of Barbara, compression ratio: 32. Left: SPIHT (wavelet) . Right: Multi-layer.

ant ringing artifacts. A quantitative comparison of SPIHT and the multi-layer coder is provided in Table 1. The multi-layer coder clearly outperforms SPIHT on the image Barbara, both in terms of PSNR and visual quality.

Houses

Figure 17-left shows the result of a compression of 24, using SPIHT, and on the right is the result of the multi-layer coder at the same compression rate. In this example we used two layers: a first compression with wavelets, and a second compression with an adapted wavelet packet basis. Figure 17 clearly shows that the multi-layer coder has kept all the details on the shutters, that have been erased by SPIHT. A quantitative comparison of SPIHT and the multi-layer coder is provided in Table 1. In this case, the multi-layer outperforms SPIHT on the image Houses in terms of visual quality.

Figure 17: Houses, compression ratio: 32. Left: SPIHT (wavelet) . Right: Multi-layer.

6 Discussion and conclusion

6.1 Relation to other work

It is possible to draw some connections between our algorithm and other related ideas. There are several different methods that are related to the multi-layered coding technique.

Hybrid Video Coding. Existing video compression standards, such as MPEG1-2, H26[1-3] [25] rely on a hybrid coding scheme: each frame in the video is encoded with two layers:

1. the first layer generated by motion prediction,
2. the second layer is the residual error after motion prediction. These error images are textural images that are usually coded with 8x8 DCT blocks. More efficient methods that rely on the “matching pursuit” technique [17], a concept described in the following, are being explored [23].

Document Image Compression. In order to efficiently transmit and store documents that include text and high quality images, one needs to separate text and images, and apply a different compression technique for text and images. The *DjVu* algorithm [5] is an example of this paradigm, where documents are decomposed into two layers:

Barbara				Houses			
Rate (bpp)	Compression	SPIHT	Multi-layer	Rate (bpp)	Compression	SPIHT	Multi-layer
1	8	36.41	36.58	1	8	30.84	30.44
0.67	12	33.40	33.76	0.67	12	28.07	27.88
0.5	16	31.39	31.97	0.5	16	26.15	26.30
0.4	20	30.10	30.70	0.4	20	25.06	25.27
0.333	24	29.13	29.73	0.333	24	24.33	24.43
0.286	28	28.27	28.94	0.286	28	23.75	23.75
0.25	32	27.58	28.27	0.25	32	23.17	23.24
0.20	40	26.65	27.25	0.20	40	22.33	22.46
0.154	52	25.79	26.19	0.16	50	21.65	21.76
0.125	64	24.86	25.37	0.125	64	20.98	20.95
0.10	88	24.25	24.61	0.10	88	20.37	20.33

Table 1: Left: coding results for 8bpp. 512x512 Barbara. Right: coding results for 8bpp. 512x512 Houses

1. images are coded with a wavelet basis,
2. text is coded with a technique dedicated to fax, or bi-level images.

The decomposition is performed with a segmentation algorithm.

Best Orthogonal Basis. This is the original best basis algorithm developed by Coifman and Wickerhauser in [11]. If the signal is composed of highly non orthogonal components, then the method may not yield a sparse representation. As explained previously, if the image is composed of a mixture of libraries, then the best-basis will not provide a sparse representation.

Matching pursuit. This technique was developed by Mallat and Zhang [18] in order to provide an adaptive representation of signals. The matching pursuit algorithm is a greedy algorithm that selects at each iteration the waveform that best correlates with a large "library" of waveforms. Matching pursuit has a myopic view, and therefore cannot select a set of features all at once. As opposed to the best basis algorithm, the library need not be composed of orthogonal "atoms", and thus the final signal representation is not constructed with orthonormal waveforms. The algorithm may therefore yield a representation that is redundant.

Basis Pursuit. This technique was developed by Chen and Donoho [7] to provide a representation of a signal with the minimum l^1 norm of the coefficients. Basis pursuit has shown to be very useful to obtain very sparse representations of signals. Unfortunately, unlike the best basis algorithm, Basis Pursuit cannot be applied to real time signal processing [7].

Another limitation of the basis pursuit, and matching pursuit techniques is the requirement to use a unique dictionary, or library to find interesting projections.

6.2 Future work

The question of how many residuals, and how much budget (how many bits b_i) should be allocated to each residual R^i remains open. Another related question is the order in which we use the libraries. At the moment our approach consist in using two layers: wavelets, and wavelet packets or local trigonometric transforms. We use an exhaustive search to find the optimal allocation of the budget between the two layers. In the future we intend to address this problem with a comprehensive methodology, based on a variational approach, that will provide a clear understanding on how to adjust these parameters.

6.3 Conclusion

We have addressed the problem of efficiently coding images that contain a mixture of smooth and textured features. We have shown that a new solution to the image coding problem is provided by "multi-layered" representations. Any image is parsed into a superposition of coherent layers: smooth-regions layer, textures layer, etc. A coder based on this new paradigm was studied: it offers the advantage of being scalable, both in term of spatial resolution, and in terms of quality of reconstruction. The evaluation of the algorithm indicates that this new coder outperforms the best wavelet coding algorithms [26, 27], both visually and in term of the quadratic error. Furthermore in error-prone environment at low-bitrate (such as wireless networks), this decomposition permits to efficiently protect the first layer (which corresponds to a very small number of bits), and could provide robust transmission over mobile channels.

References

[1] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli, *Local cosine transform –a method for the reduction of the blocking effect in JPEG*, J. of Math. Imag. and Vision **3** (1993), 7–38.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, *Image coding using wavelet transform*, IEEE Trans. on Image Processing **1** (1992), no. 2, 205–220.

[3] P. Auscher, G. Weiss, and M.V. Wickerhauser, *Local sine and cosine bases of Coifman and Meyer*, Wavelets-A Tutorial, Academic Press, 1992, pp. 237–256.

[4] N.N. Bennett, *Fast algorithms for best anisotropic bases*, To appear in Applied and Computational Harmonic Analysis, 1998.

[5] L. Bottou, P. Haffner, P.G. Howard, P. Simard, Y. Bengio, and Y. Le Cun, *High quality document image compression with DjVu*, To appear in Journal of Electronic Imaging, 1998.

[6] T. Chang and C.C.J. Kuo, *Texture analysis and classification with tree-structured wavelet transform*, IEEE Trans. on Image Processing **2,(4)** (1993), 429–441.

[7] S.S. Chen, *Basis pursuit*, Ph.D. thesis, Stanford University, Dept. of Statistics, November 1995.

[8] A. Cohen, I. Daubechies, and J. Feauveau, *Bi-orthogonal bases of cqmpactly supported wavelets*, Comm. Pure Appl. Math. **45** (1992), 485–560.

[9] R.R. Coifman and Y. Meyer, *Remarques sur l'analyse de Fourier à fenêtre*, C.R. Acad. Sci. Paris I (1991), 259–261.

[10] ———, *Size properties of wavelet packets*, Wavelets and their Applications (Ruskai et al, ed.), Jones and Bartlett, 1992, pp. 125–150.

[11] R.R. Coifman and M.V. Wickerhauser, *Entropy-based algorithms for best basis selection*, IEEE Trans. on Information Theory **38** (1992), no. 2, 713–718.

[12] G. Davis and S. Chawla, *Image coding using optimized significance tree*, IEEE Data Compression Conference -DCC'97, 1997, pp. 387–396.

[13] G.M. Davis, *A wavelet-based analysis of fractal image compression*, IEEE Trans. on Image Processing **7** (1998), no. 2, 141–154.

[14] R.A. DeVore, B. Jawerth, and B.J. Lucier, *Image compression through wavelet transform coding*, IEEE Trans. on Information Theory **38,(2)** (1992), 719–746.

[15] A.S. Lewis and G. Knowles, *Image compression using the 2-D wavelet transform*, IEEE Trans. on Image Processing **1,(2)** (1992), 244–250.

[16] J. Li, P.Y. Cheng, and C.C.J. Kuo, *An embedded wavelet packet transform technique for texture compression*, SPIE Vol 2569, 1995, pp. 602–613.

[17] S. Mallat, *A wavelet tour of signal processing*, Academic Press, 1998.

[18] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. on Signal Processing **41(12)** (1993), 3397–3415.

[19] G. Matviyenko, *Optimized local trigonometric bases*, Applied and Computational Harmonic Analysis **3** (1996), 301–323.

[20] F.G. Meyer, A.Z. Averbuch, and J-O. Strömberg, *Fast adaptive wavelet packet image compression*, submitted to IEEE Trans. on Image Processing, 1998. Also available from `http://noodle.med.yale.edu/meyer/~profile.html`.

[21] ______, *Fast wavelet packet image compression*, IEEE Data Compression Conference -DCC'98, 1998.

[22] F.G. Meyer and R.R. Coifman, *Brushlets: a tool for directional image analysis and image compression*, Applied and Computational Harmonic Analysis (1997), 147–187.

[23] R. Neff and A. Zakhor, *Very low bit-rate video coding based on matching pursuits*, IEEE Trans. Circ. & Sys. for Video Tech. **7, 1** (1997), 158–171.

[24] K. Ramchandran and M. Vetterli, *Best wavelet packet bases in a rate-distortion sense*, IEEE Trans. on Image Processing **2** (1993), no. 2, 160–175.

[25] K.R. Rao and J.J. Hwang, *Techniques and standards for image, video, and audio coding*, Prentice Hall, 1996.

[26] Amir Said and William A. Pearlman, *A new fast and efficient image codec based on set partioning in hierarchical trees*, IEEE Trans.on Circ.& Sys. for Video Tech. **6** (1996), 243–250.

[27] J.M. Shapiro, *Embedded image coding using zerotrees of wavelet coefficients*, IEEE Trans. on Signal Processing **41** (1993), no. 12, 3445–3462.

[28] P. Sriram and M.W. Marcellin, *Image coding using wavelet transforms and entropy-constrained treillis quantization*, IEEE Trans. on Image Processing **4** (1995), 725–733.

[29] M.V. Wickerhauser, *Adapted wavelet analysis from theory to software*, A.K. Peters, 1995.

[30] I.H. Witten, R.M. Neal, and J.G. Cleary, *Arithmetic coding for data compression*, Communications of the ACM **30,6** (1987), 520–540.

[31] Z. Xiong, K. Ramchandran, and M.T. Orchard, *Space-frequency quantization for wavelet image coding*, IEEE Trans. on Image Process. **6** (1997), no. 5, 677–693.