# Image compression using an edge adapted redundant dictionary and wavelets

Lorenzo Peotta, Lorenzo Granai, Pierre Vandergheynst *

*Signal Processing Institute*
*Swiss Federal Institute of Technology*
*EPFL-STI-ITS-LTS2*
*ELD 241, Station 11*
*CH-1015 Lausanne, Switzerland*

## Abstract

Low bit rate image coding is an important problem regarding applications such as storage on low memory devices or streaming data on the internet. The state of the art in image compression is to use 2-D wavelets. The advantages of wavelet bases lie in their multiscale nature and in their ability to sparsely represent functions that are piecewise smooth. Their main problem on the other hand, is that in 2-D wavelets are not able to deal with the natural geometry of images, i.e they cannot sparsely represent objects that are smooth away from regular submanifolds. In this paper we propose an approach based on building a sparse representation of the edge part of images in a redundant geometrically inspired library of functions, followed by suitable coding techniques. Best N-terms non-linear approximations in general dictionaries is, in most cases, a NP-hard problem and sub-optimal approaches have to be followed. In this work we use a greedy strategy, also known as Matching Pursuit to compute the expansion. The residual, that we suppose to be the smooth and texture part, is then coded using wavelets. A rate distortion optimization procedure chooses the number of functions from the redundant dictionary and the wavelet basis.

*Key words:* Image compression, Sparse representation, Greedy approximation, Matching Pursuit, Wavelet, Image representation, Coding, Redundant dictionaries

* {lorenzo.peotta, lorenzo.granai, pierre.vandergheynst}@epfl.ch

# 1  Introduction

The state of the art in image compression is based on transform coding using orthonormal basis such as DCT and Wavelets. These schemes have achieved high compression ratios thanks to the huge research work that has been performed in efficiently coding the transform coefficients and parameters. Nevertheless these traditional approaches suffer some severe limitations. Wavelets for example fail to capture regularities of contours, since they are not able to sparsely represent one-dimensional singularities of 2-D signals [1]. Recently a lot of research effort was aimed at representing a natural image exploiting its inherent geometrical structure. Concerning image compression, interesting approaches that follow this idea can be found for example in [2], [3] and [4].

In this work we aim at obtaining an efficient encoding by approximating the edges of an image by a sum of two-dimensional, non-separable functions. The residual, that we suppose constituted by the smooth part of the image and textures, is then coded via wavelets. The edge-oriented dictionary is built by anisotropically scaling, orienting and bending a generating function, resulting in an overcomplete basis set. Highly non-linear approximation in redundant dictionaries is, in general, a NP-hard problem. However this does not impair the possibility of finding good sparse representations in particular classes of dictionaries. There are different approaches that find a sub-optimal solution to this problem like Basis Pursuit [5] (BP), Orthogonal Matching Pursuit [6] (OMP) and Matching Pursuit [7] (MP). In this paper we use a technique based on the greedy Matching Pursuit algorithm that in general is not sparseness-preserving [8]. However experimental results showed its ability to give highly sparse decompositions of 1-D signals [9], images [10,11] and videos [12,13]. Moreover very recent theoretical results prove that, under certain conditions, MP iteratively selects the atoms that give the sparsest representation. This recovery condition is based on the quasi-incoherence of the dictionary and the sparseness of the signal [14–16]. Combining the MP approximation properties with an accurate design of the dictionary makes it possible to achieve high compression ratios, catching the most visually relevant structures of natural images. Since at very low bit rate these structures mainly consist of object contours, we designed a dictionary that can represent edges working on the output of a Laplacian pyramid. The low-frequency part and textures are coded using a simple wavelet coder. The balance between the functions approximating the contours and wavelet coefficients is performed in a rate distortion sense. Numerical and visual comparisons with the state of the art show the quality of the results; in particular ringing artifacts typical of wavelets are avoided.

This paper is organized as follows: Section 2 explains the adopted coding scheme composed by a high pass filtering that gives a detail version of an image, a greedy decomposition for the high frequencies and a wavelet based

coding of the residual. Section 3 recalls the basic principles of the Matching Pursuit algorithm, while the dictionary design and the atom selection method are illustrated in Section 4 and 5 respectively. Section 6 presents a rate distortion study of our representation method and Section 7 shows results and comparisons. Finally, conclusions can be found in Section 8.

## 2   Representation Method

Natural images present different primitives with peculiar characteristics. We can easily distinguish smooth parts (low frequency components, $I_{smooth}$), edges ($I_{edge}$) and textures ($I_{texture}$). Formally, we propose here the following model for an image $I$:

$$I \simeq I_{smooth} + I_{edge} + I_{texture}. \tag{1}$$

According to this model, we describe here a compression scheme that codes each of the primitives of Eq. 1 with a different technique. Such a scheme is presented in Fig. 1.

The dictionary, described in Section 4, has been designed to match the object contours. These discontinuities have most of their energy at high frequencies. Therefore, before coding the edges, the image is decomposed with wavelets and reconstructed keeping all the sub-bands but the low-pass. This step is equivalent to a high-pass filtering and it is labeled as "W HP" in Fig 1. The high frequency content of the signal is thus decomposed over the edge-oriented dictionary using the Matching Pursuit algorithm. After that the MP projection coefficients have been quantized, a residual image is computed by subtracting the quantized MP reconstruction from the original input image. This residual contain the low frequencies of the signal, the textures and the artifacts introduced by MP (the latter also include quantization errors). As can be seen in Fig. 1 the residual is decomposed with wavelets. The wavelet functions, used for both decomposing the residual and computing the HP input for MP, are the Cohen-Daubechies-Feauveau 9,7 [17]. At this point we have low pass wavelet coefficients (projection on the scaling function) representing $I_{smooth}$, atoms from the edge-oriented dictionary representing $I_{edge}$ and the high frequency wavelet coefficients representing $I_{texture}$ and correcting the MP artifacts, if any. Note that the quantization of the MP indexes and projections is fixed. The coefficients of the coarse version representing the smooth part of the image are quantized in a differential way (DPCM), while the wavelet coefficients are subject to a dead-zone quantization with the dead-zone step twice the quantization step. The quantization steps for the coarse signal and wavelet coefficients are independent.

All these parameters are subject to a Rate-Distortion (RD) optimization that establishes the number of atoms to code, the quantization step for the DPCM

of the coarse version and the step for the wavelet dead-zone quantization. All the parameters and the quantized coefficients are entropy coded using an adaptive arithmetic coding algorithm.
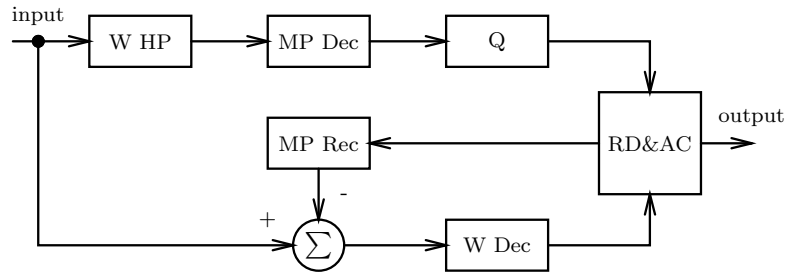


Fig. 1. Encoding scheme: W HP is the high-pass filtering using Wavelets, MP Dec and MP Rec are respectively the Matching Pursuit decomposition and reconstruction, Q represents the quantization operation, W Dec is the Wavelet decomposition, RD is the Rate-Distortion optimization and AC stands for Arithmetic Coding.

## 3 Matching Pursuit Algorithm

In this section we recall the basics of the iterative process used for the selection of waveforms that represent the signal structures. A more detailed explanation of the Matching Pursuit algorithm can be found in [7].
Let $D = \{g_\gamma\}_{\gamma \in \Gamma}$ be a dictionary of unit norm vectors $g_\gamma$ called atoms, where $\Gamma$ is the set of possible indexes. The function $f$ is first decomposed as follows:

$$f = \langle g_{\gamma_0}, f \rangle g_{\gamma_0} + Rf, \tag{2}$$

where $Rf$ is the residual component after having approximated $f$ in the direction of $g_{\gamma_0}$. Since $Rf$ and $g_{\gamma_0}$ are orthogonal, it follows that

$$\|f\|^2 = |\langle g_{\gamma_0}, f \rangle|^2 + \|Rf\|^2. \tag{3}$$

To minimize $\|Rf\|$ we must choose $g_{\gamma_0}$ such that the absolute value of the projection $|\langle g_{\gamma_0}, f \rangle|$ is maximal. Applying iteratively such a procedure, after $N$ iterations we obtain:

$$f = \sum_{n=0}^{N-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} + R^N f, \tag{4}$$

where $R^0 f = f$ and $R^n f$ is the residual after the $n^{th}$ step; it can be proved [18] that $R^n f$ converges to zero when n tends to infinity. The convergence is exponential in the case of finete dimensional signal spaces. Following (3) we can write:

4

$$\|f\|^2 = \sum_{n=0}^{N-1} |\langle g_{\gamma_n}, R^n f \rangle|^2 + \|R^N f\|^2. \tag{5}$$

Equation (5) expresses the energy conservation of the MP. The convergence of MP depends on the structure of the dictionary, the search strategy and the signal $f$ that has to be approximated. In [18] it is shown that there exist two real numbers $\alpha, \beta \in ]0, 1]$ such that for all $n \geq 0$ the following relation is valid:

$$\|R^{n+1} f\| \leq (1 - \alpha^2 \beta^2)^{1/2} \cdot \|R^n f\|, \tag{6}$$

where $\alpha$ is an optimality factor related to the strategy adopted to select the best atom in the dictionary, while $\beta$ depends on the dictionary, representing its ability to capture the features of the input function $f$ [19]. Equation (6) gives a simple upper bound of the decay of the approximation error.

## 4 The Dictionary

The dictionary used to represent the detail version of the image is composed of a set of functions, named atoms, built by applying the following four types of transformation to the generating function $\phi(\vec{x}) : \mathbb{R}^2 \to \mathbb{R}$ with $\vec{x} = (x_1, x_2)$.

$a)$ Translation $\mathcal{T}_{\vec{b}}$, to move the atom all over the image:

$$\mathcal{T}_{\vec{b}} \, \phi(\vec{x}) = \phi(\vec{x} - \vec{b}). \tag{7}$$

$b)$ Rotation $\mathcal{R}_\theta$, to locally orient the atom along contours:

$$\mathcal{R}_\theta \, \phi(\vec{x}) = \phi(r_\theta(\vec{x})), \tag{8}$$

where $r_\theta$ is a rotation matrix

$$r_\theta(\vec{x}) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{9}$$

$c)$ Since, in general, images not only contain straight edges, we are adapting the atoms to the shape of natural contours with a bending transformation $\mathcal{B}_r$. Roughly speaking, this operation arches the $x_2$-axis with radius $r$, formally $\mathcal{B}_r \, \phi(\vec{x}) = \phi(\beta_r(\vec{x}))$. Figure 2 shows how the $\beta_r(\cdot)$ operation acts, and Figure 4 shows the result of bending a generating function. The transformation

5

$\beta_r : \mathbb{R}^2 \longrightarrow (-\infty, r] \times \mathbb{R}$ is not linear and it is defined as

$$\beta_r(\vec{x}) = \begin{cases} \begin{bmatrix} r - \sqrt{(x_1 - r)^2 + x_2^2} \\ \arctan(\frac{x_2}{r - x_1}) \end{bmatrix} & \text{if } x_1 < r \\[4mm] \begin{bmatrix} r - |x_2| \\ \text{sign}(x_2)\left(x_1 - r + r\,\frac{\pi}{2}\right) \end{bmatrix} & \text{if } x_1 \geq r \end{cases} . \qquad (10)$$

Applying the bending to a continuous function $\phi(\vec{x})$ we obtain $\mathcal{B}_r\phi(\vec{x})$, which is in general discontinuous on the semi-axis $[r, +\infty)$. When $\phi(\vec{x})$ is continuous and satisfies the conditions

$$\begin{aligned} \phi(r, x_2) &= const & \text{for} \quad -r \leq x_2 \leq r \quad \text{and} \\ \phi(r, x_2) &= \phi(r, -x_2) & \forall x_2 \in \mathbb{R}, \end{aligned} \qquad (11)$$

it follows that $\mathcal{B}_r\phi(\vec{x})$ is continuous for all $\vec{x} \in \mathbb{R}^2$. The definition of the bending transformation is driven by the desire to keep the wavelet like behavior of the generating function (see Subsection 4.1) perfectly orthogonal to the smooth direction of edges.

In practice the bending transformation does not introduce discontinuities in the atoms, since the generating functions are close to zero for $x_1 = r$.
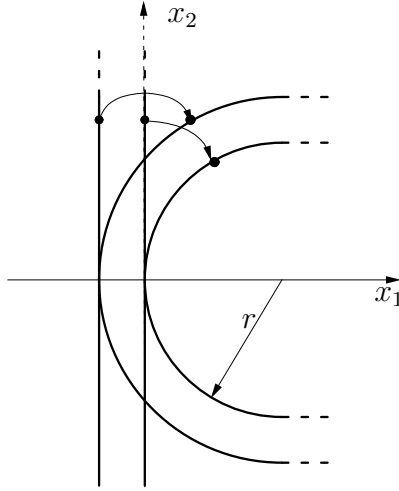


Fig. 2. Bending operation $\mathcal{B}_r$ that arches the $x_2$-direction with radius $r$.

d) Anisotropic scaling $\mathcal{S}_{a_1, a_2}$, to adapt to contour smoothness

$$\mathcal{S}_{\vec{a}}\ \phi(\vec{x}) = \mathcal{S}_{a_1, a_2}\ \phi(x_1, x_2) = \phi\left(\frac{x_1}{a_1}, \frac{x_2}{a_2}\right). \qquad (12)$$

Atoms are generated varying the parameters $\vec{b}, \theta, r, \vec{a}$ of the four previous transforms in the following order:

$$g_{(\vec{b},\theta,r,\vec{a})}(\vec{x}) = \mathcal{T}_{\vec{b}} \ \mathcal{R}_\theta \ \mathcal{B}_r \ \mathcal{S}_{\vec{a}} \ \phi(\vec{x}). \tag{13}$$

Finally the waveforms obtained are normalized:

$$g^{\text{norm}}_{(\vec{b},\theta,r,\vec{a})}(\vec{x}) = \frac{g_{(\vec{b},\theta,r,\vec{a})}(\vec{x})}{\|g_{(\vec{b},\theta,r,\vec{a})}(\vec{x})\|}. \tag{14}$$

The dictionary used by the MP-algorithm can be written as in equation (15), where all the parameters are discretized:

$$\mathcal{D} = \{g^{\text{norm}}_{(\vec{b},\theta,r,\vec{a})}(\vec{x})\}_{\vec{b},\theta,r,\vec{a}} \ . \tag{15}$$

The radius $r$ is discretized using a dyadic grid, while for the position $\vec{b}$ a uniform grid is kept. The two scaling factors are discretized in a uniform way; the range of the scaling factor along $x_2$ is bigger than the one along $x_1$ and depends on the radius parameter (it can not exceed $\frac{\pi}{2}$ times the radius). The rotation step $\theta$ is inversely proportional to the scale $a_2$.

## 4.1 Generating functions

The choice of the generating function $\phi(x_1, x_2)$ is driven by the idea of efficiently approximating the high frequencies of contours, like singularities in 2-D. Therefore, the atom must be a smooth low resolution function in the direction of the contour and approximate the edge transition in the orthogonal (singular) direction.

In order to be able to well represent either roof and ramp edges [20] we adopted two different generating functions, doubling in this way the size of the dictionary.

The first function, called $\phi_1(\vec{x})$ is a combination of a generalized Gaussian and its first derivative. In the $x_1$-direction (which is the singular-direction) it is the first derivative of a Gaussian, while in the $x_2$-direction (which is the contour-direction) it is a generalized Gaussian, see Fig. 3:

$$\phi_1(x_1, x_2) = 2x_1 \ e^{-(x_1^2 + x_2^4)}. \tag{16}$$

The second generating function $\phi_2(\vec{x})$, shown in Fig. 3, is a combination of a Gaussian and its second derivative. It is a slight variation of the function
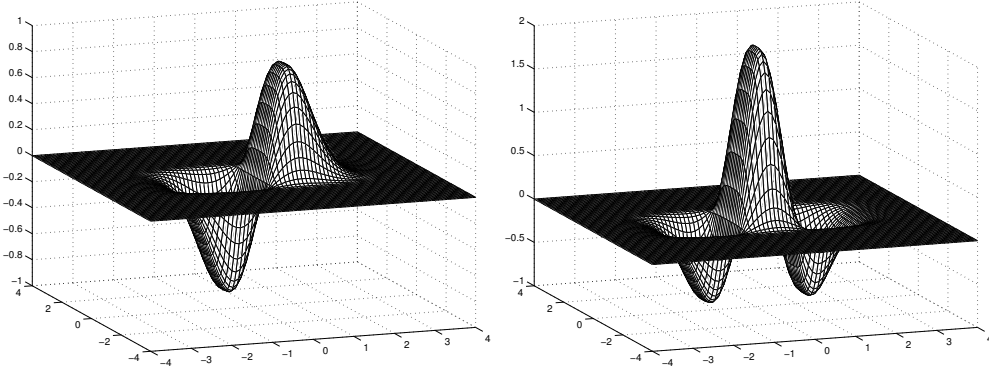
Fig. 3. Generating functions: $\phi_1(x_1, x_2)$ on the left, $\phi_2(x_1, x_2)$ on the right.

introduced in [21]; this choice is motivated by the optimal joint spatial and frequency localization of the gaussian kernel:

$$\phi_2(x_1, x_2) = (4x_1^2 - 2) \cdot e^{-(x_1^2 + x_2^4)}. \tag{17}$$

We chose the shape parameter of the generalized gaussian equal to four in the $x_2$-direction in order to have a faster decay in the space domain. It turns out that atoms can better approximate segments of edges and visual artifacts are reduced.

Figure 4 shows three atoms generated using the two different generating functions in both space and frequency domain. The bending transformation is also shown. It can be seen that the function $\phi_2$ is more compact in the frequency domain, whereas $\phi_1$ reaches lower frequencies.

It is worth mentioning that the atoms we generate are similar to the basis functions obtained by Olshausen and Field while studying the spatial receptive fields of simple cells in mammalian striate cortex [22]. These functions, as well as our atoms, are characterized by being localized in space, oriented and bandpass. An additional property of the overcomplete dictionary we propose is given by the possibility of bending the generating function. This characteristic does not appear in the functions obtained by Olshausen and Field. This can be explained by considering that the image patches they were analyzing are too small to observe such a phenomenon.

### 4.2 Size of the Dictionary

Taking into account all the atom parameters and the two generating functions, the dictionary can be written as:

$$\mathcal{D} = \{g_{(\phi, \vec{b}, \theta, r, \vec{a})}^{\mathrm{norm}}(\vec{x})\}_{\phi, \vec{b}, \theta, r, \vec{a}} \cdot \tag{18}$$
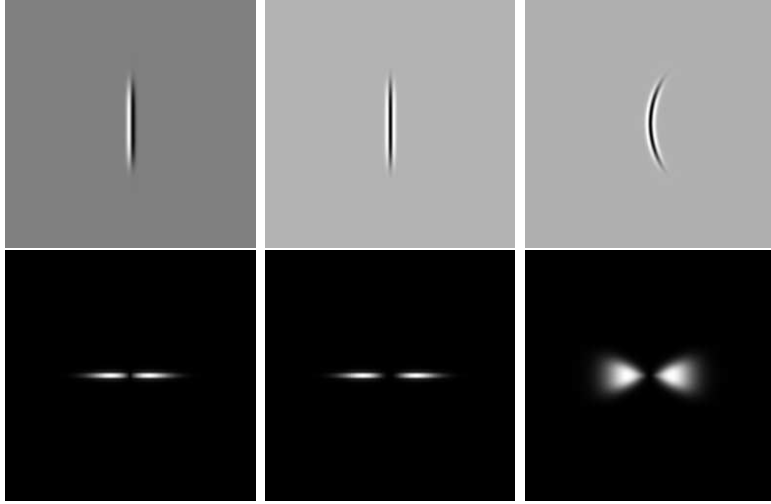
8

Fig. 4. Three atoms: on the top space domain, on the bottom frequency domain represented in a logarithmic scale. The first atom is generated from $\phi_1$, the others from $\phi_2$. The effect of bending and anisotropically scaling the atom can be observed.

Here $\phi \in \{\phi_1, \phi_2\}$ is the index that specifies which function has been chosen to create the atom, while the other values are the same as in equation (15). Finally we obtain a highly redundant dictionary, with a redundancy factor $d \simeq 17000$. The size of the dictionary increases dramatically allowing big scaling factor along $x_2$, indeed the number of rotations is proportional to the scale parameter $a_2$. A dictionary including elongate atoms is able to better represents long edge structures. In fact we can say that designing high redundant dictionaries, we are trying to increase the $\beta$ parameter of equation (6).

Even if big dictionaries can be built with a small coherence [23], our dictionary has high coherence since we adopt a geometric oriented design. Thus, we can not assure that MP recovers the best sparse approximation of the signal [14]. Nevertheless we notice a fast energy decay of the residual at first iterations, which means that the dictionary copes well with natural data. MP is able to select good atoms, at least during first iterations.

## 5 Searching Algorithm

Matching Pursuit is used to decompose the detail version of the image in its most important features. This greedy algorithm selects at each iteration an atom from the dictionary such that the projection coefficient $|\langle g_{\gamma_n}, R^n f \rangle|$ is maximum. To find such $g_{\gamma_n}$ we use a full search algorithm that computes the inner products between the residual and all the functions of the dictionary. Since the dictionary is composed of all the translations of the transformed generating functions (TGF), see Eq. (13), it is clear that all the inner products between the TGF translated all over the residual and the residual itself,

correspond to the convolution of the TGF with the residual. To speed up the search, we compute the convolutions like products in the frequency domain. The Fourier transform of all the TGF is computed only once and stored.

The complexity of a MP decomposition of a signal of $n$ samples results to be of the order of

$$C \cdot N \cdot d \cdot n \log_2 n, \tag{19}$$

where $N$ is the number of chosen atoms, the constant $C$ depends on the strategy adopted for atom selection and $d$ depends on the size of the dictionary. In fact, $d = |\mathcal{D}|/n$ is the redundancy of the dictionary and it corresponds to the size of $\mathcal{D}$ without considering translations. In particular we use a modified Matching Pursuit algorithm for which $C \ll 1$. At each iteration, $n_k - n_{k-1}$ atoms are selected and used to decompose the residual. Like in Eq. (4) we can write:

$$f = \sum_{k=0}^{K-1} \left( \sum_{n=n_k}^{n_{k+1}-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} \right) + R^N f, \tag{20}$$

with $n_0 = 0$ and $n_K = N$. At the $k^{th}$ iteration all the atoms of the dictionary are sorted according to the absolute values of the projection coefficients. Afterwords, starting from the one with highest projection, all the atoms that are quasi-orthogonal are selected. We adopted this algorithm in order to obtain an important reduction in computational load. In fact selecting on average $\overline{n}_k$ atoms at once it turns out that MP only needs $N/\overline{n}_k$ iterations, reducing in this way the number of inner products which constitute the most computationally demanding part of the algorithm. For example, decomposing images of size $256 \times 256$ pixels we observed a speed-up of around 20. The drawback of this method is that there is no more a guaranty that at each iteration the best atom will be selected as in the case of the full search MP. However the resulting loss in image quality is negligible [10].

This method is similar to the fast MP implementation described by Mallat [18]. When you control the structure of the dictionary, you can update the projection of the atoms on the residual at the next step taking into account the correlation between atoms, according the following updating formula:

$$\langle g_\gamma, R^{n+1} f \rangle = \langle g_\gamma, R^n f \rangle - \langle g_{\gamma_n}, g_\gamma \rangle \langle g_{\gamma_n}, R^n f \rangle. \tag{21}$$

In particular the projection of the atoms orthogonal to the selected ones will not change.

In order to further speed up the atom selection, another algorithm, based on a tree-based pursuit decomposition, may be taken into account [24]. But since the quality loss is not negligible, especially at the range of bit rate that we are interesting in, we decided not to use this searching method.

## 6  Rate Distortion

As described in Section 2 , the number of atoms to code and the quantization step for the wavelet coefficients are chosen based on a Rate-Distortion optimization. In order to study the RD of our representation method, we take into account the image model (1) proposed in Section 2.

We indicate with $\tilde{I}_{edge}(m)$ the $m$-terms approximation of the edge part of the image obtained by the MP decomposition of $I_{edge}$ over the contour oriented dictionary. Thereby the input of the wavelet decomposition $(I_{st})$, supposed to contain the smooth and texture part, is obtained by subtracting $\tilde{I}_{edge}$ from the original image:

$$I_{st}(m) = I - \tilde{I}_{edge}(m) = I_{smooth} + I_{texture} + (I_{edge} - \tilde{I}_{edge}(m)). \qquad (22)$$

Since $\tilde{I}_{edge}(m)$ is the superposition of $m$ atoms with quantized projections, the final error is given by

$$I_{err} = I - \tilde{I}_{st}(m, \Delta), \qquad (23)$$

where $\tilde{I}_{st}(m, \Delta)$ is the approximation of $I_{st}(m)$ given by the quantization of the wavelet coefficients. Let $R_{MP}$ the rate due to the atoms and $R_W$ the one due to the wavelet coefficients, the total rate is $R = R_{MP} + R_W$. It depends on the number of atoms used to approximate $I_{edge}$ and on the quantization steps of the wavelet coefficients. Before investigating the Rate-Distortion of our representation method, we study the rate related to the MP expansion, and we recall the RD theory concerning wavelet coding.

### 6.1  MP rate

Our signal approximation over $\mathcal{D}$ is represented by the atom indexes, positions and projections. The indexes or parameters that characterize the atoms shape are entropy coded using an adaptive arithmetic coding algorithm. Since the $x_2$-scale parameter depends on the radius, the arithmetic coder uses the conditional probability $p(x_2\text{-}scale|radius)$ to code the $x_2$-scale, and $p(rotation|x_2\text{-}scale)$ for the rotation parameter. In order to code the positions and projection coefficients, two different approaches can be taken into account. The first one consists of ordering the atoms in decreasing absolute projection values, then the projections can be quantized either in a differential way (DPCM) [10] or using an exponential quantizer [25]. The quantization is followed by arithmetic coding. The $x_1$ and $x_2$ coordinates of the atoms positions are then simply stored without any particular coding scheme.

The second approach performs a different sorting of the atoms in such a way to take advantage in coding their positions [12]. The atoms are ordered by
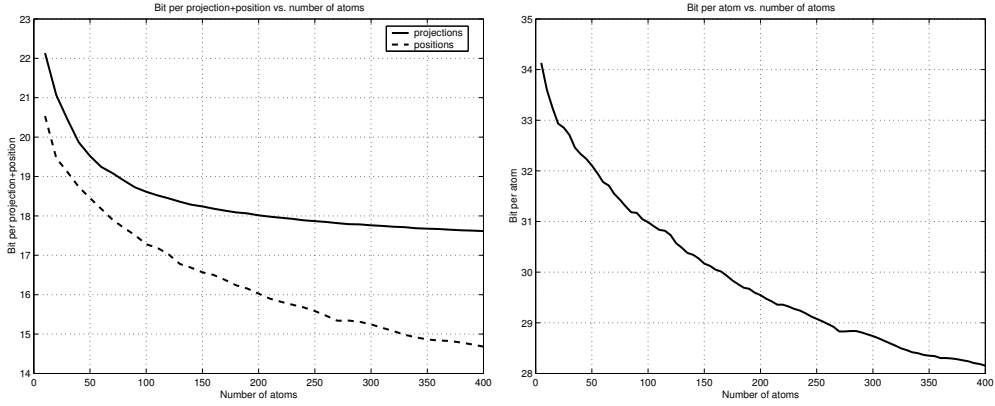
Fig. 5. Left: Comparison between projections oriented and positions oriented coding. Right: Total number of bits per atom as a function of the number of atoms used to approximate $I_{edge}$. The position oriented method is used to code positions and projections.

raster scanning, then the $x_1$ and $x_2$ coordinates are coded in a differential way followed by arithmetic coding. The drawback is that the SNR scalability is lost [11]. In this case a simple uniform quantization and arithmetic coding of the projections is performed.

As shown on the left-hand side of Figure 5, the position oriented coding method outperforms the projection coding, thus we chose to use the former to code atoms positions and projections. Finally the right-hand side of Figure 5 gives the total bit rate per atom $R_a(m)$ as a function of the number of atoms used to approximate the edge component $I_{edge}$. The exponential decay of $R_a(m)$ is due to the fact that increasing $m$, the grid representing the atoms positions becomes more dense and the entropy of the displacements between adjacent atoms decreaseS. At the limit we can say that the bits required to code an atom get close to the coding rate of the projection and shape parameters.

## 6.2 Wavelet rate distortion

The wavelet functions, used to decompose the residual (22), are the Cohen-Daubechies-Feauveau 9,7 with normalization $(\sqrt{2}, \sqrt{2})$. This biorthogonal wavelet basis is nearly orthogonal and thus we suppose that the distortion given by the quantization in the wavelet domain coincides with the distortion in the original domain. In order to have an hint on the real rate-distortion behavior, let us make the hypothesis of high resolution quantization, although it is not always satisfied in the compression domain. Using a uniform quantizer, we can

12

approximate the distortion or MSE as a function of the quantization step $\Delta$:

$$D_W = \frac{\Delta^2}{12}. \tag{24}$$

The rate, that corresponds to the entropy of the output indexes, depends on the quantization step and on the differential entropy

$$R_W = \frac{\sum_k N_k(h_k - \log_2 \Delta)}{N} = \frac{\sum N_k h_k}{N} - \log_2 \Delta, \tag{25}$$

where $h_k$ is the differential entropy, $N_k$ is the number of wavelet coefficients at resolution $k$ and $N$ is the size of the signal.

### 6.3 MP+Wavelet rate distortion

Now we can formulate the Rate-Distortion of our coder based on MP and wavelet decomposition. The final distortion depends on the quantization step of the wavelet coefficients, and for fine quantization we have that $D = D_W = \frac{\Delta^2}{12}$. The total rate $R = R_{MP}(m) + R_W(m, \Delta)$ depends on the number of atoms approximating $I_{edge}$ and obviously on the quantization step $\Delta$,

$$R_{MP}(m) = \frac{R_a(m)m}{N}, \tag{26}$$

$$R_W(m, \Delta) = \frac{\sum N_k h_k(m)}{N} - \log_2 \Delta. \tag{27}$$

It is important to notice that the differential entropy $h_k(m)$ associated to wavelet resolution level $k$, depends on the number of atoms that represent $\tilde{I}_{edge}(m)$. Indeed the statistic of $I_{st}(m) = I - \tilde{I}_{edge}(m)$ changes with $m$, especially at the resolution levels that contain the energy of the edge structures. The left-hand side of Figure 6 shows how the differential entropy depends on $m$. We thus write the Lagrangian cost

$$L(m, \Delta) = D(\Delta) + \lambda\left(R(m, \Delta) - R_{bpp}\right) = \tag{28}$$
$$\frac{\Delta^2}{12} + \lambda\left(\frac{R_a(m)m}{N} + \sum_k \frac{N_k}{N} h_w(m) - \log_2 \Delta - R_{bpp}\right),$$

where $R_{bpp}$ is the bit budget per pixel.

Differentiating with respect to $\Delta$ and $m$ (we neglect the integer constrain on $m$), we obtain

$$\frac{\partial L}{\partial \Delta} = \frac{2\Delta}{12} - \frac{\lambda}{\Delta \ln 2} \tag{29}$$
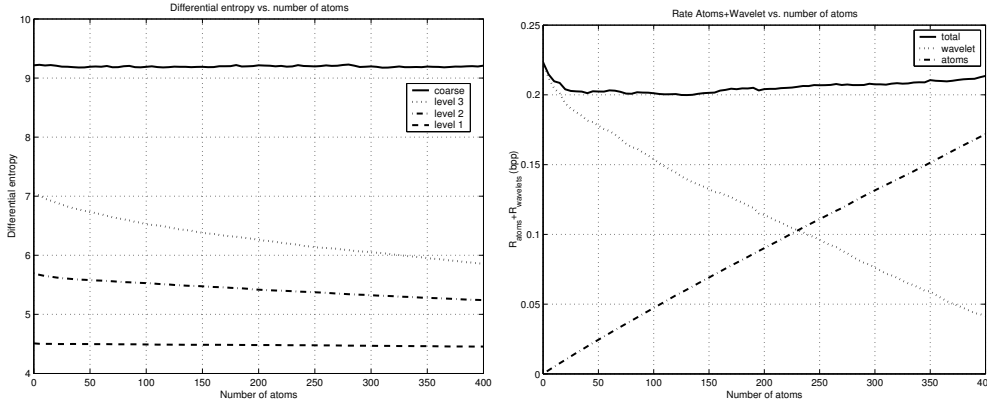
Fig. 6. Left: Differential entropy of the wavelet coefficients at different resolution levels. Right: bpp of our representation method as a functions of the number of atoms. The quantization step is fixed, and so the final distortion.

$$\frac{\partial L}{\partial m} = \lambda \left( \frac{R_a'(m)m + R_a(m)}{N} + \sum_k \frac{N_k}{N} h_w'(m) \right) \qquad (30)$$

and setting the derivative to 0 we obtain

$$\Delta = \sqrt{\frac{6\lambda}{\ln 2}} \qquad (31)$$

and

$$R_a'(m)m + R_a(m) + \sum_k N_k h_k'(m) = 0. \qquad (32)$$

Solving equation (32) we find the optimal number of atoms that minimizes the rate for a given distortion. It is important to notice that the solution of (32) does not depend on the final distortion $D$. This is due to the assumption of fine quantization of the wavelet coefficients. Once we get the optimal number of atoms $m_{opt}$, setting to zero the derivative of equation (28) with respect to $\lambda$, we obtain the quantization step as a function of the bit budget,

$$\Delta = 2^{\left( \frac{R_a(m_{opt})m_{opt}}{N} + \sum \frac{N_k}{N} h_k(m_{opt}) - R_{bpp} \right)}. \qquad (33)$$

Remark that this is true only at hight bit rate, and that the functions $R_a(m)$ and $h_k(m)$ have to be estimated. Figure 6 shows on the left-hand side the behavior of the differential entropy at different resolution levels, and on the right-hand side shows the total rate as a function of $m$ for a fixed step $\Delta$ (changing $\Delta$ corresponds to a vertical translation of the wavelet rate). The minimum rate is reached coding $m_{opt}$ atoms: for studied example this minimum occurs between 100 and 150 atoms.

In practice, at low bit rates the fine quantization hypothesis is not satisfied and the simple model for the wavelet RD does not fit its real behavior. Moreover, we are using a dead zone uniform quantizer, which improves the rate distortion at low bit rate quantization [26], and different steps of quantization for the

14

coarse and wavelet coefficients can be chosen. Implementing a numerical rate distortion optimization, it turns out that the optimal number of atoms changes depending on the bit rate. All the graphics in this section show the mean of the results obtained using the three standard images *Lena, cameraman* and *peppers* of size $256 \times 256$.

## 7 Results

This section provides some results obtained with our algorithm; in the following it will be simply called MPW. A comparison is made with the standard JPEG2000 [27,28]. Another point of comparison is a pure wavelet encoder that uses exactly the same coding options and RD optimization we adopted in the MPW coding scheme. In the following, we refer to the latter method as "Wavelets". Table 1 shows the PSNR vs. bit-rate results for the images *peppers* and *cameraman*. Both images have size $256 \times 256$ pixels. For this size, the computational time for MPW coding is around one hour, using a 2GHz processor.

Table 1
PSNR vs. bit rate for the images "cameraman" and "peppers": comparisons between MPW, JPEG2000 and a coding scheme based on DWT. See also Fig.7.

| cameraman ($256 \times 256$) | | | | peppers ($256 \times 256$) | | | |
|---|---|---|---|---|---|---|---|
| Rate (bpp) | MPW | JPEG2000 | Wavelets | Rate (bpp) | MPW | JPEG2000 | Wavelets |
| 0.05 | 22.50 | 21.00 | 21.50 | 0.06 | 23.13 | 21.36 | 22.10 |
| 0.10 | 25.06 | 23.63 | 23.79 | 0.10 | 25.33 | 23.83 | 23.90 |
| 0.15 | 26.45 | 25.23 | 24.96 | 0.15 | 27.03 | 25.89 | 25.68 |
| 0.20 | 27.38 | 26.45 | 26.10 | 0.20 | 28.25 | 27.25 | 27.00 |
| 0.25 | 28.11 | 27.38 | 27.02 | 0.25 | 29.09 | 28.62 | 28.05 |
| 0.30 | 28.76 | 28.53 | 27.80 | 0.30 | 29.91 | 29.79 | 29.00 |
| 0.35 | 29.27 | 29.57 | 28.51 | 0.35 | 30.56 | 30.62 | 29.80 |
| 0.50 | 30.61 | 31.16 | 30.21 | 0.50 | 32.24 | 32.85 | 31.89 |

At very low bit rate, our algorithm obtains good results because it is capable of catching the main features of a natural image with few functions. It is fair to observe that at less than 0.1 bpp the gap between JPEG2000 and MPW is also partly due to the bigger size of the JPEG2000 header: in fact one can notice that even our very simple "Wavelet" encoder outperforms the standard. The size of the JPG2000 header is 148 Byte, meanwhile MPW has a simple header of size 22 Byte. Regarding JPEG2000, if we take into account only the

bit stream due to the quantized wavelets coefficients, we observe an increase of about 0.6 dB, 0.4 dB and 0.2 dB respectively at 0.1 bpp, 0.2 bpp and 0.3 bpp. This difference disappears for bigger images (e.g. $512 \times 512$ pixels). In [10] we obtained similar (but slightly worse in terms of PSNR) results with a less elaborated approach also based on Matching Pursuit. However, thanks to the use of the wavelets for coding the residual (see Sec. 2), the gain we obtain here is not only limited at very low bit rates. Moreover, observing Figure 7, one can see that the proposed scheme outperforms JPEG2000 not only in terms of PSNR but also of visual quality. We compare results up to 0.5 bpp, where there is no relevant visual difference between the images compressed with our method and JPEG2000.

Other numerical results can be found in [29,30]. In particular we compress bigger images and we show an example of how the PSNR decreases using a less redundant dictionary.
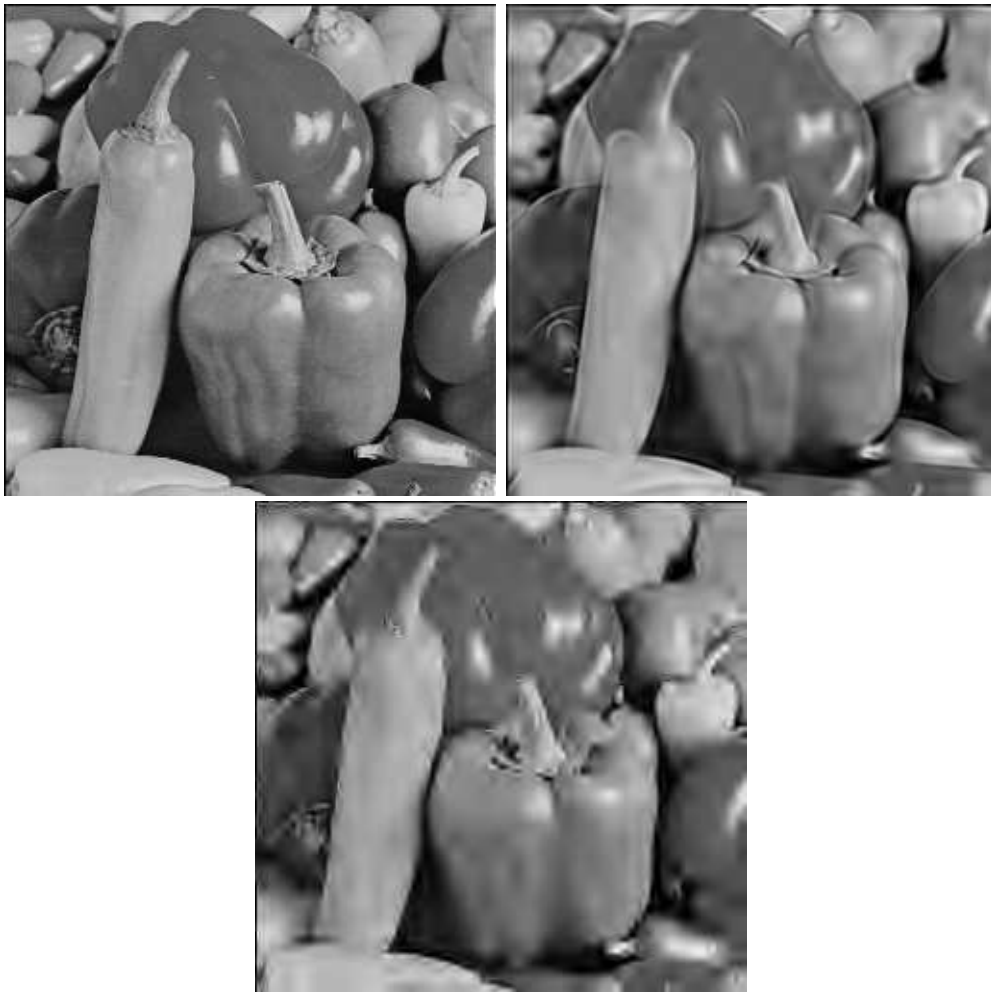
Fig. 7. Comparison between JPEG2000 and the proposed scheme based on MP at 0.15 bpp. Top left: original peppers image (256x256). Top right: image compressed with MPW (PSNR = 27.03). Bottom: image compressed with JPEG2000 (PSNR = 25.89).

## 8 Conclusions

In this paper we introduced a technique for coding natural images at low bit rates. The edges are approximated using elementary waveforms from a redundant dictionary, while the non-edge part of the image is represented by wavelets. The geometry inspired dictionary is build by applying anisotropic scaling, rotation and bending transformations to two generating functions which are smooth on the edge direction and wavelet like in the other. In order to overcome the combinatorial complexity in finding the sparsest approximation over a redundant dictionary, we use a pursuit algorithm based on MP. A rate distortion analysis, together with numerical entropy estimations, controls the number of edge-oriented atoms and the quantization of wavelets.

This coding scheme easily allows SNR scalability. Starting from a base layer given by the atoms representing contours, SNR enhancements can be obtained applying usual scalability properties of wavelet encoders [27].

Our coder outperforms the state of the art in terms of both PSNR and visual quality. In particular edges, which at very low bit rate constitute the more visually relevant features of an image, are no more affected by ringing artifacts typical of wavelets.

## References

[1] E. J. Candes, D. L. Donoho, Curvelets - a surprisingly effective nonadaptive representation for objects with edges, in: C. R. A. Cohen, L. Schumaker (Eds.), Curve and Surface Fitting: Saint Malo 1999, Vanderbilt University Press, Nashville, TN, 2000.

[2] M. Wakin, J. Romberg, H. Choi, R. Baraniuk, Wavelet-domain approximation and compression of piecewise smooth images, IEEE Trans. Image Processing Submitted.

[3] R. Shukla, P. Dragotti, M. Do, M. Vetterli, Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images, IEEE Transactions on Image Processing 14 (3).

[4] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Trans. Image Processing 14 (4) (2005) 423–438.

[5] S. S. Chen, D. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1) (1998) 33–61.

[6] Y. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximatiuon with application to wavelet decomposition, in: Proceedings of 27th Annual Asilomar Conference on Signal Systems and Computers, 1993.

[7] S. Mallat, Z. Zhang, Matching pursuit with time-frequency dictionary, IEEE Trans. on Signal Processing 41 (12) (1993) 3397–3415.

[8] R. DeVore, V. Temlyakov, Some remarks on greedy algorithms, Advances in Computational Mathematics 5 (1996) 173–187.

[9] R. Gribonval, E. Bacry, Harmonic decomposition of audio signals with matching pursuit, IEEE Trans. Signal Processing 51 (1) (2003) 101–111.

[10] L. Peotta, L. Granai, P. Vandergheynst, Very low bit rate image coding using redundant dictionaries, in: Proc. of 48th SPIE annual meeting, Vol. 2, San Diego, USA, 2003.
URL http://lts2www.epfl.ch/publications.html

[11] R. M. Figueras i Ventura, P. Vandergheynst, P. Frossard, Low rate and flexible image coding with redundant representations., IEEE Trans. Image Processing Accepted.

[12] R. Neff, A. Zakhor, Very low bit-rate video coding based on matching pursuit, IEEE Trans. Circuits Syst. Video Technol. 7 (1997) 158–171.

[13] L. Granai, E. Maggio, L. Peotta, P. Vandergheynst, Hybrid video coding based on bi-dimensional matching pursuit, Journal on Applied Signal Processing 2004 (17) (2004) 2705–2714.

[14] R. Gribonval, P. Vandergheynst, On the exponential convergence of matching pursuit in quasi-inchoerent dictionaries, IEEE Trans. Inform. Theory Submitted to.

[15] J. Tropp, Greed is good : Algorithmic results for sparse approximation, IEEE Trans. Inform. Theory 50 (10) (2004) 2231–2242.

[16] J. Tropp, Just relax: Convex programming methods for subset selection and sparse approximation, Tech. rep., Texas Institute for Computational Engineering and Sciences (2004).

[17] A. Cohen, I. Daubechies, J. C. Feauveau, Biorthogonal bases of compactly supported wavelets, Commun. Pure Appl. Math. 45 (1992) 485–560.

[18] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1998.

[19] P. Frossard, P. Vandergheynst, Redundancy in non-orthogonal transforms, in: Proc. ISIT, Washington DC, USA, 2001, p. 196.

[20] J. Beltran, F. Beltran, A. Estopanan, Multiresolution edge detection and classification: noise characterization, in: Proc. IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5, 1998, pp. 4476–4481.

[21] P. Vandergheynst, P. Frossard, Efficient image representation by anisotropic refinement in matching pursuit, in: Proc. IEEE International Conference on Acoustic, Speach and Signal Processing(ICASSP'01), Salt Lake City, USA, 2001.

[22] B. Olshausen, D. Field, Sparse coding with an overcomplete basis set: A strategy employed by V1?, Vison Research  (37) (1997) 3311–3325.

[23] R. Gribonval, M. Nielsen, Approximation with highly redundant dictionaries, in: Proc. of 48th SPIE annual meeting, Vol. 2, San Diego, USA, 2003.

[24] P. Jost, P. Vandergheynst, P. Frossard, Tree-based pursuit, Tech. rep., Swiss Federal Institute of Technology, Lausanne, Switzerland (July 2004).

[25] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, M. Kunt, A posteriori quantization of progressive matching pursuit streams, IEEE Trans. on Signal Processing 52 (2) (2004) 525–535.

[26] S. Mallat, F. Falzon, Analisys of low bit rate image transform coding, IEEE Trans. Signal Processing 46 (4) (1998) 1027–1042.

[27] M. Rabbani, R. Joshi, An overview of the JPEG2000 still image compression standard, Signal Processing: Image Communication 17 (2002) 3–48.

[28] `http://jpeg2000.epfl.ch/`.

[29] `http://lts2www.epfl.ch/~granai/research.htm`.

[30] `http://lts2www.epfl.ch/~peotta/`.