

# Very low bit rate image coding using redundant dictionaries

Lorenzo Peotta, Lorenzo Granai and Pierre Vandergheynst

Signal Processing Institute, Swiss Federal Institute of Technology, Lausanne 1015, Switzerland

## ABSTRACT

Very 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 images in a redundant geometrically inspired library of functions, followed by suitable coding techniques. Best N-term 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. Finally the last step in our algorithm is an enhancement layer that encodes the residual image: in our simulation we have used a genuine embedded wavelet codec.

**Keywords:** Sparse representation, Greedy approximation, Matching Pursuit, Wavelet, Image Representation, Coding, Redundant dictionaries

## 1. INTRODUCTION

The state of the art in image coding is based on DCT and Wavelets. These schemes have achieved big compression ratios thanks to the huge research work that has been performed in efficiently coding the transform coefficients and parameters. Nevertheless, wavelets fail to capture regularities of contours, since they are not able to sparsely represent one-dimensional singularities. Efficient encoding can be obtained by decomposition of the image in a sum of two-dimensional, non-separable, functions. Anisotropy, orientation and bending of the basis functions, as well as their translation all over the image, generate a redundant basis that composes the building blocks of the signal expansion. Highly non-linear approximations in redundant dictionaries of functions is, in general, a NP-hard problem, anyhow this does not impair the possibility of finding very good sparse representations in particular classes of dictionaries. There are different approaches that find a sub-optimal solution to this problem like Basis Pursuit<sup>1</sup> (BP), Orthogonal Matching Pursuit<sup>2,3</sup> (OMP) and Matching Pursuit<sup>2</sup> (MP). In this paper we use a technique based on the greedy Matching Pursuit algorithm that gives good highly sparse decompositions of images. Combining this characteristic with an accurate design of the dictionary makes possible to achieve high compression ratios, catching the most visually relevant structures in an image. Since at very low bit rate these structures mainly consists of objects' contours, we designed a dictionary that can represent edges working on a detail version of an image obtained from a Laplacian Pyramid scheme. Finally if one wants to target a higher bit rate, the quality can be incremented adopting an enhancement layer that encodes the residual image: in our simulation we have used a wavelet based encoder that performs a space-frequency quantization of the transform coefficients. Numerical and visual comparisons with other algorithms show that good results can be obtained; 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 low pass filtering that gives a coarse version of an image, a greedy decomposition for the high frequencies and an wavelet based enhanced layer for the residual. Section 3 recalls the basic principles of the Matching Pursuit algorithm. The dictionary design and the atoms' selection method are illustrated in Section 4 and 5 respectively, while Section 6 presents the entropy coding for both the low frequency part and the atoms decomposition. In Section

---

Send correspondence to {lorenzo.peotta,lorenzo.granai,pierre.vandergheynst}@epfl.ch.

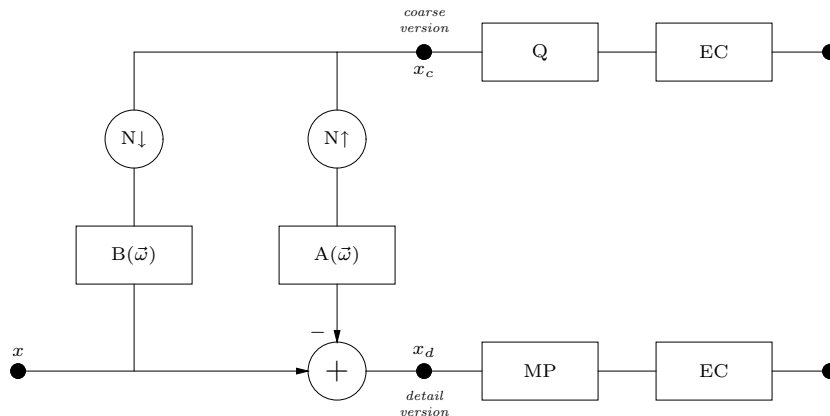
Web site: lts2www.epfl.ch

This work has been supported by the SNF grant 2100-066912.01/1 and SNF NCCR IM.2

7 the main results obtained are shown, comparing with the state of the art. Section 8 explains how the proposed algorithm can be extended in order to achieve higher bit rates. Finally conclusions and future works can be found in Section 9.

## 2. REPRESENTATION METHOD

The main aim of this work is the efficient coding of the ‘edge’ discontinuities of images and a new dictionary has been designed to match the shape of the objects. These discontinuities have most of their energy at high frequencies, therefore, before coding the edges, the image is decomposed into low frequencies and high frequencies using the Laplacian Pyramid scheme of Burt and Adelson.<sup>4</sup> From an original image, the Laplacian Pyramid scheme derives a coarse approximation by low pass filtering and downsampling. Based on this coarse version, it predicts the original (by upsampling and filtering) and calculates the difference as the high pass version or detail version. The coarse approximation is compressed with quantization followed by entropy coding, and the detail version is represented with the atoms from the new dictionary. The Matching Pursuit algorithm is used to decompose the detail version in its most important feature. The scheme is presented in Fig. 1.



**Figure 1.** Laplacian Pyramid. Only the encoding part is shown.  $B(\vec{\omega})$  is the 2-D lowpass filter,  $A(\vec{\omega})$  is the 2-D interpolation filter,  $N$  is the downsampling factor.  $Q$  represents the quantization operation,  $MP$  the Matching Pursuit and  $EC$  stands for Entropy Coding.

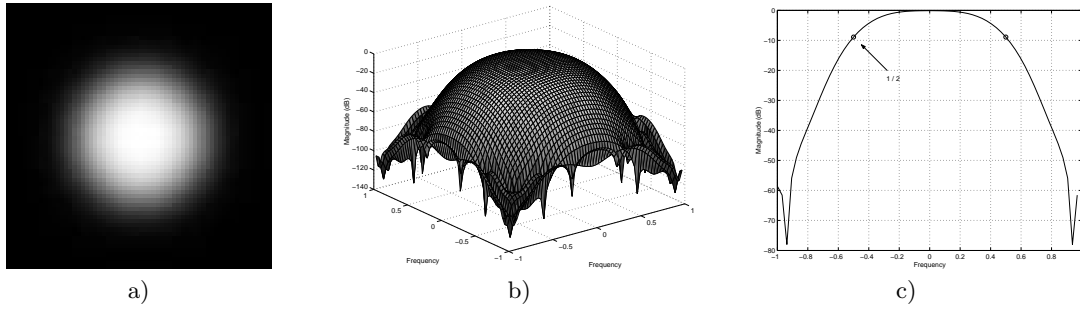
Even though better coding results may be obtained by choosing the two filters independently, here the filter  $B(\vec{\omega})$  and  $A(\vec{\omega})$  are set to be the same for simplicity. The filtering process is performed applying three times a low pass filter followed by downsampling. The filter used is a  $11 \times 11$  taps, symmetric low pass FIR filter, designed using the window method. The window used is gaussian with variance  $\sigma^2 = 2$  pixels, the normalized cut-off frequency is 0.45, while the downsampling factor is two.

One could use wavelets to obtain the coarse version  $x_c$ , see Fig. 1, but with the drawback of losing the circular symmetry that we can have with a true 2-D filter, see Fig. 2. This results in the detail version  $x_d$  being ‘sullied’ in the horizontal directions, and the Matching Pursuit could be unable to efficiently code the detail version.

The Matching Pursuit algorithm turns out to be extremely efficient at very high compression rates, losing its good performance when the number of iteration increases. This is partly due to  $MP$ ’s own properties<sup>2</sup> and partly to the fact that the dictionary that we are using is mainly aimed to detect edges, as will be explained in section 4. So, in order to be able to code images not only at very low bit rate we introduce a third step in the representation method that is based on wavelets and that will be illustrated in section 8.

## 3. MATCHING PURSUIT ALGORITHM

In this section we recall the basics of the iterative process used for the selection of the waveforms that represent the signal structures. A more detailed explanation of the Matching Pursuit algorithm can be found in Ref. 2.



**Figure 2.** Low pass filter: magnitude frequency responses, a) linear, b) and c) in dB.

Let  $D = \{g_\gamma\}_{\gamma \in \Gamma}$  be a dictionary of unitary norm vectors  $g_\gamma$  called atoms and  $\Gamma$  represent the set of possible indexes. The function  $f$  is first decomposed as follows:

$$f = \langle g_{\gamma_0}, f \rangle g_{\gamma_0} + Rf, \quad (1)$$

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

$$\|f\|^2 = |\langle g_{\gamma_0}, f \rangle|^2 + \|Rf\|^2. \quad (2)$$

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

$$f = \sum_{n=0}^{N-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} + R^N f, \quad (3)$$

where  $R^0 f = f$  and  $R^n f$  is the residual after the  $n^{\text{th}}$  step; it can be proved<sup>5</sup> that  $R^n f$  converges exponentially to zero when  $n$  tends to infinity.

As in (2) we can write:

$$\|f\|^2 = \sum_{n=0}^{N-1} |\langle g_{\gamma_n}, R^n f \rangle|^2 + \|R^N f\|^2. \quad (4)$$

Equation (4) expresses the energy conservation of the MP. The convergence of MP depends on both the dictionary and the search strategy. In Ref. 6 it has been shown that there are two real numbers  $\alpha, \beta \in ]0, 1]$  such that for all  $n \geq 0$  the following relation is valid:

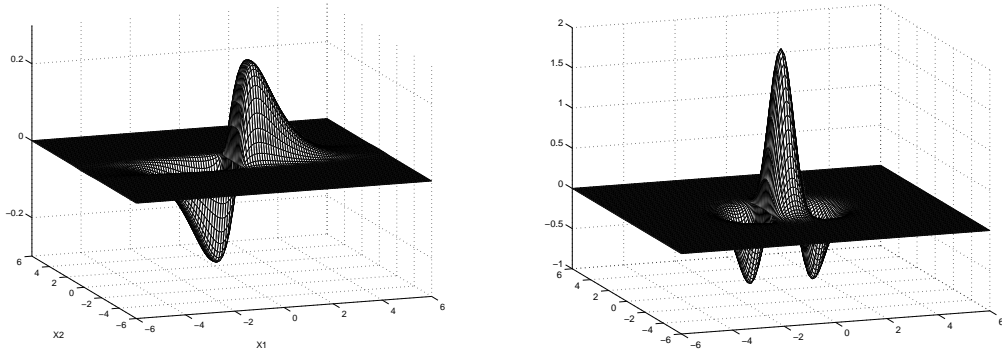
$$\|R^{n+1} f\| \leq (1 - \alpha^2 \beta^2)^{1/2} \cdot \|R^n f\|, \quad (5)$$

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  $f$ .<sup>7</sup>

#### 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 transformations to the generating function  $g(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$  with  $\vec{x} = (x_1, x_2)$ .





**Figure 4.** Generating functions:  $g_1(x_1, x_2)$  on the left,  $g_2(x_1, x_2)$  on the right

The dictionary used by the MP-algorithm can be written as in equation (13), where the range of all parameters is discretized:

$$\mathcal{D} = \{\text{atom}_{(\vec{b}, \theta, r, \vec{a})}^{\text{norm}}(\vec{x})\}_{\vec{b}, \theta, r, \vec{a}}. \quad (13)$$

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 an uniform way except for the first interval which is split in two parts; the range of the scaling factor along  $x_2$  is bigger than the one along  $x_1$  and depends on the radius parameter. The rotation step  $\theta$  depends linearly on  $a_2$  scale.

#### 4.1. Generating functions

The choice of the generating atom  $g(x_1, x_2)$  is driven by the idea of efficiently approximating the high frequencies of contour, 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<sup>8</sup> we adopted two different generating functions, doubling in this way the size of the dictionary.

The first function, called  $g_1(\vec{x})$  is a combination of generalized gaussians. In the  $x_1$ -direction (which is the singular-direction) it has a Laplacian shape with odd symmetry while in the  $x_2$ -direction (which is the contour-direction) it is a generalized gaussian, see Fig. 4. The deviation  $\sigma_x$  is set to 0.7:

$$g_1(x_1, x_2) = x_1 e^{-\left(\frac{|x_1|}{\sigma_{x_1}} + x_2^4\right)}. \quad (14)$$

The second generating function, shown in Fig. 4, is a combination of a gaussian and its second derivative introduced in Ref. 6; this choice is motivated by the optimal joint spatial and frequency localization of the gaussian kernel:

$$g_2(x_1, x_2) = (4x_1^2 - 2) \cdot e^{-(x_1^2 + x_2^2)}. \quad (15)$$

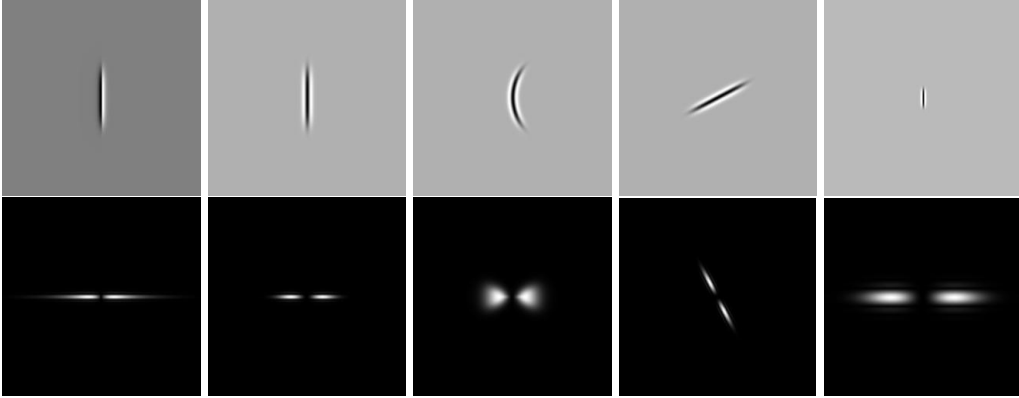
Figure 5 shows five atoms with different scaling factors, rotation and bending in both space and frequency domain. It can be seen that the function  $g_2$  is more compact in the frequency domain, whereas  $g_1$  is more spread and reaches higher frequencies.

#### 4.2. Size of the Dictionary

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

$$\mathcal{D} = \{\text{atom}_{(g, \vec{b}, \theta, r, \vec{a})}(\vec{x})\}_{g, \vec{b}, \theta, r, \vec{a}}. \quad (16)$$

Here  $g \in \{g_1, g_2\}$  is the index that specifies which function has been chosen to create the atom, while the other values are the same as in equation (13). Finally the number of functions that compose our dictionary is around 11000. This set of atoms results to be highly redundant.



**Figure 5.** Five atoms: on the top space domain, on the bottom frequency domain represented in a logarithmic scale. The first function starting from the left is generated from  $g_1$ , the others from  $g_2$ . The effect of bending, rotating and anisotropically scaling the atom can be observed.

## 5. SEARCHING ALGORITHM

The Matching Pursuit is used to decompose the detail version in its most important features. This greedy algorithm described in Section 3 at each iteration selects 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. (11), 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 dictionary is computed once and stored.

A slightly modified Matching Pursuit algorithm is also introduced. At each iteration,  $n_k$  atoms are selected and used to decompose the residual. Like in Eq. (3) we can write:

$$f = \sum_{k=0}^{K-1} \left( \sum_{n=n_k}^{n_{k+1}-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} \right) + R^N f, \quad (17)$$

with  $n_0 = 0$  and  $n_K = N$ . At  $k^{th}$  iteration, according to the absolute values of the projection coefficients, all the atoms of the dictionary are sorted. Starting from the one with highest projection, all the atoms that are quasi-orthogonal are selected. We adopted this algorithm in order to obtain a consistent reduction in computational load. In fact selecting on average  $\bar{n}_k$  atoms at once it turns out that MP only needs  $N/\bar{n}_k$  iterations. 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; in Fig. 6 it can be seen that the difference is visually imperceptible.

## 6. ENTROPY CODING OF THE COEFFICIENTS

In Section 2 we saw that the image is decomposed into a coarse version (low frequencies) and detail version (high frequencies), as shown in Fig.1. The coarse version is quantized in a differential way (DPCM) and the quantization indexes are entropy coded using an adaptive arithmetic coding algorithm. The detail version is coded with Matching Pursuit. The 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 algorithm uses the conditioned 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 of the atoms, two different approaches can be taken into account. The first one, that we use, consists of ordering the atoms in decreasing absolute projection values, then the projections are quantized in a differential way (DPCM) followed by arithmetic coding; the  $x_1$  and  $x_2$  coordinates of the atoms' positions are 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 the atoms' positions.<sup>9</sup> For example it is possible to code the  $x_1$  or  $x_2$  coordinates in a differential way followed by arithmetic coding. The drawback is that we can not perform the exponential quantization<sup>10</sup> of the absolute projections values anymore.

We developed another method oriented to the "position coding approach". A sorting based on the best path through all the atoms' positions is performed, where the cost to go from one position to another is based on the entropy of the  $x_1, x_2$ -displacement. This is clearly a Traveling Salesman Problem, where the crucial point is the definition of the cost matrix. Using an approach based on positions coding, it can be useful to adopt a scheme to quantize the projections based on the scaling parameters that specify the area of the atoms. The impact that this approach can have on compression efficiency should be deeply investigated.

## 7. RESULTS

In this section we present results obtained compressing the images "Lena" and "cameraman" with the algorithm previously presented. The size of both images is  $256 \times 256$  pixels. The bit allocation for the coarse and detail versions has not been optimized in a rate distortion sense, so the amount of data reserved for each part of the coded stream has been chosen empirically. Table 1 shows the PSNR vs. bit-rate comparing with JPEG2000<sup>11,12</sup>; it can be seen that MP clearly outperforms JPEG2000 in terms of PSNR at very low bit rate.

It is interesting to compare these results with more recent techniques such as SFQ<sup>13</sup> and WSFQ.<sup>14</sup> For example the former obtains a PSNR of 23.63 (dB) at 0.077 bpp while the latter obtains 23.66 (dB).

**Table 1.** PSNR vs. bit rate for the images "cameraman" and "Lena": comparison between the proposed algorithm based on MP and JPEG2000. See Fig. 6 also.

cameraman ( $256 \times 256$ )			Lena ( $256 \times 256$ )		
Rate (bpp)	MP	JPEG2000	Rate (bpp)	MP	JPEG2000
0.030	20.45	18.41	0.036	22.43	21.02
0.052	22.46	21.07	0.062	24.71	23.40
0.077	23.92	22.62	0.079	25.51	24.32
0.100	24.75	23.74	0.100	26.30	25.28
0.125	25.46	24.53	0.125	27.15	26.14
0.147	25.61	25.13	0.150	27.70	26.89

The results presented in Table 1 are obtained by decomposing the image with the algorithm that selects several atoms per iteration, see Eq.(17). Using the classical Matching Pursuit decomposition we obtain slightly better results in terms of PSNR with the drawback of a heavier computational load. For example at a rate of 0.077 bpp we have 24.01 (dB): a gain of around 0.1 (dB) that does not yield in a better visual quality.

One can see in Fig. 6 that the gain with respect to JPEG2000 is not only in terms of PSNR but also in terms of visual quality.

Figure 7 and Table 1 show the results for "Lena" image. Our representation method outperforms JPEG2000 in terms of PSNR and it can also be seen that the ringing artifacts typical of wavelets are avoided.



cameraman 256 x 256



classic MP at 0.077 bpp, PSNR = 24.01



JPEG2000 at 0.077 bpp, PSNR = 22.62



MP at 0.077 bpp, PSNR = 23.92



JPEG2000 at 0.147 bpp, PSNR = 25.13



MP at 0.147 bpp, PSNR = 25.61

**Figure 6.** Comparison between JPEG2000 and MP at low bit rate. On top right the image is decomposed with the classical MP method, see Eq. (3) and Section 5





MP at 0.079 bpp, PSNR = 25.51



JPEG2000 at 0.079 bpp, PSNR = 24.32

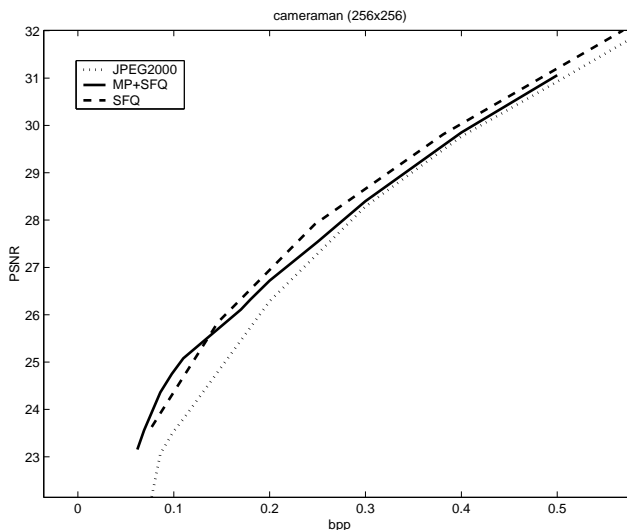
**Figure 7.** Lena coded with MP and JPEG2000 at 0.079 bpp

## 8. ENHANCED LAYER WITH WAVELETS

As mentioned in Section 2, the good performance of Matching Pursuit slowly gets worse as the bit rate increases. In fact, once that the smooth part of an image is represented by low-pass filtering and downsampling, the MP dictionary is able to well represent the edges; indeed in this way the other structures that compose a picture (mainly texture) are not taken into account and this limits the algorithm. Studying the histogram of the pixel values of the MP residual we found that it tends to a gaussian when the number of atoms is augmented. The signal has less and less structure to code and slowly becomes similar to noise. However, before arriving at this point, MP has already lost its efficiency and is no more competitive with respect to other techniques such as wavelets. So, in order to extend the range of efficiency of the algorithm presented in this paper we introduced an enhanced layer that codes the residual of MP with wavelets.

We adopted the scheme named space-frequency quantization (SFQ) and presented in Ref. 13: it is based on the joint application of the wavelet zero-tree quantization (setting to zero tree-structured sets of wavelet coefficients) and scalar quantization (a uniform scalar quantizer applied to all non-zeroed coefficients), optimized in a rate-distortion sense. The implementation that we used<sup>15</sup> gives results slightly worse than the ones reported in the cited work, but allows to reach the same rate-distortion slope. The wavelet functions used for decomposing the residual are the Cohen-Daubechies-Feauveau.9-7, even if it could be interesting to test other wavelet bases. In practice, after a certain number of iterations or a target PSNR is reached the residual image is passed to the SFQ encoder that is responsible to represent the signal's component that the low pass part and MP were not able to catch. In this way the good behavior of Matching Pursuit is exploited in the very low bit rate range as can be seen in the previous section; later good performances can be maintained in terms of both PSNR and visual quality. Figure 8 shows the R-D curve obtained for the image "cameraman" and compares it with the standard JPEG2000<sup>11,12</sup> and pure SFQ algorithm.<sup>13</sup> One can observe that the same asymptotic slope can be achieved for a bit rate higher than 0.2 bpp, i.e. where MP alone would be clearly outperformed by the cited methods.

An interesting evolution of this method should be to integrate the the SFQ algorithm performing an optimization in a rate distortion sense, so that the point where to switch from one algorithm to another will be better determined.



**Figure 8.** R-D curve obtained for the image "cameraman"; comparison of the proposed method with SFQ and JPEG2000

## 9. CONCLUSIONS

In this paper we have presented a new method for very low bit rate image coding. The signal is represented by a coarse version obtained by low pass filtering and detail version that corresponds to the high frequencies.

Decomposing the detail version on an overcomplete basis using a greedy algorithm (Matching Pursuit) we obtain a sparse representation of the signal. In order to be able to code images not only at very low bit rate, a third step based on wavelets is proposed. This technique outperforms the state of the art in terms of both PSNR and visual quality. In particular the edges, that at very low bit rate constitute the more visually relevant feature of an image, are no more affected by ringing artifacts typical of wavelets. It can be seen that taking into account the inherent geometry of images we obtain extremely high compression ratio. Moreover, exploiting the ability of the algorithm to represent the edges and reducing the low frequency information, we are able to catch the main characteristics of a picture: for example it can be useful for face detection and recognition.<sup>16</sup>

An other interesting quality of the Matching Pursuit algorithm is that it provides an intrinsically progressive stream. This valuable feature allows to fit restrictive bandwidth constraints simply transmitting the information related to the first atoms that are the most significant ones. Applications where rate scalability is an issue can profit from this property.<sup>17</sup>

An extension of this work should be to optimize the bit allocation between Matching Pursuit and SFQ in a rate distortion sense. As mentioned in Section 6 there are other possibilities to code the atoms position and the projection. The approach that we followed guarantees the scalability but it is not necessarily the best one concerning compression. A comparative investigation between the two approaches should be performed in order to select the best technique.

## REFERENCES

1. S. S. Chen, D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.* **20**(1), pp. 33–61, 1998.
2. S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionary," *IEEE Trans. on Signal Processing* **41**, pp. 3397–3415, dec 1993.
3. Y. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with application to wavelet decomposition," in *Proceedings of 27th Annual Asilomar Conference on Signal Systems and Computers*, Nov 1993.
4. P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.* **4**, pp. 532–540, April 1983.
5. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.
6. P. Vandergheynst and P. Frossard, "Efficient image representation by anisotropic refinement in matching pursuit," in *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP'01)*, (Salt Lake City, USA), 2001.
7. P. Frossard and P. Vandergheynst, "Redundancy in non-orthogonal transforms," in *Proc. ISIT*, (Washington DC, USA), 2001.
8. J. Beltran, F. Beltran, and A. Estopanan, "Multiresolution edge detection and classification: noise characterization," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, **5**, pp. 4476–4481, Oct 1998.
9. O. Al-Shaykh, E. Miloskavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits," *IEEE Trans. on CSVT* **9**, pp. 113–143, Feb 1998.
10. P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," in *To appear in IEEE Trans. of Signal Processing*, 2003.
11. <http://jpeg2000.epfl.ch/>.
12. D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Larsson, and C. Christopoulos, "JPEG2000 still image coding versus other standards," in *Proceedings of SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII*, pp. 446–454, SPIE, Aug 2000.
13. Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. on Image Processing* **6**(5), pp. 677–693, 1997.
14. M. B. Wakin, "Image compression using multiscale geometric edge models," Master's thesis, Rice University, Huston, Texas, May 2002.

15. J. E. Fowler, "Qccpack: An open-source software library for quantization, compression, and coding," in *Application of Digital Image Processing XXIII*, SPIE, Aug 2000.
16. L. Peotta, "Low bit rate image representation by following objects' geometry in matching pursuit," tech. rep., Swiss Federal Institute of Technology, Lausanne, Switzerland, Nov 2001.
17. P. Frossard, P. Vandergheynst, and R. M. Figueras i Ventura, "High flexibility scalable image coding," in *Proc. of VCIP*, (Lugano, Switzerland), 2003.