Visual Information Processing using Redundant Dictionaries.

Pierre Vandergheynst and Pascal Frossard*
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute
CH-1015 Lausanne

Abstract

This paper presents a image coder based on Matching Pursuit expansion of visual signal over a redundant dictionary of functions. The dictionary is built on anisotropic refinement of a edge-like functions, that allows to efficiently capture two-dimensional features like contours, that are predominant in natural images. Besides interesting compression performance, the coder provides a bitstream that presents interesting scalability properties, that allow to finely adapt to rate or resolution constraints in visual applications.

1. Introduction

Transform coding is at the heart of modern image compression standards such as JPEG and its descendant, JPEG2000. The properties of such systems are well-known. They depend crucially on efficient encoding schemes that use the transform as a decorrelating mapping, so that scalar quantization can be applied in an optimal setting. The choice of the transform is also motivated by the desire to obtain sparse approximations of signal, i.e., good approximations with few transform coefficients. Recently though, redundant transforms have been investigated because they offer greater flexibility in the design of the transform and thus are able to produce very sparse expansions. In this paper, we illustrate this principle by showing an image codec based on a redundant library of waveforms. This scheme shows good coding performances, but also provides interesting features such as rate scalability or geometric adaptivity, which are very interesting for various application scenarios.

*This work has been partly supported by the Swiss National Science Foundation.

2. Matching Pursuit Image Coding

2.1. Overview

The encoder can be summarized as follows. The input image is compared to a redundant library of functions, using a Matching Pursuit algorithm. Iteratively, the index of the function that best matches the (residual) signal is sent to an entropy coding stage. The corresponding coefficient is quantized, and eventually entropy coded. The output of the entropy coder block forms the compressed image bitstream. The decoder performs inverse entropy coding, inverse quantization, and finally reconstructs the compressed image by summing the dictionary functions, multiplied by their respective coefficients.

Alternative image representation methods based on Matching Pursuit, have been proposed in the literature. One of the first papers that proposed to use Matching Pursuit for representing images is [1]. This first work does however not propose a coder implementation, and the dictionary is different than the one proposed in this chapter. Matching Pursuit has been used for coding the motion estimation error in video sequences [2], in a block-based implementation. This coder, contrarily to the one proposed below, makes use of sub-blocks, which, in a sense, limits the efficiency of the expansion. In the same time, it has been designed to code the residual error of motion estimation, which presents very different characteristics than edge-dominated natural images. The coder presented in the remainder takes benefit of the properties of both redundant expansions, and anisotropic functions, to offer efficient and flexible compression of natural images.

2.2. Generating functions of the dictionary

The dictionary used by our coder is structured. It is built by applying geometric transformations to a generating mother function g. By varying the parameters of this function, we generate an overcomplete set of functions spanning the input image space. The choice of the generating function, g, is driven by the idea of efficiently approximating contour-like singularities in 2-D. To achieve this goal, the atom is a smooth low resolution function in the direction of the contour, and behaves like a wavelet in the orthogonal (singular) direction. In other words, the dictionary is composed of atoms that are built on Gaussian functions along one direction and on second derivative of Gaussian functions in the orthogonal direction, that is :

$$g(\vec{p}) = \frac{2}{\sqrt{3\pi}}(4\,x^2 - 2)\;\exp(-(x^2+y^2))\,, \tag{1}$$

where $\vec{p} = [x,y]$ is the vector of the image coordinates, and $||g|| = 1$. The choice of the Gaussian envelope is motivated by the optimal joint spatial and frequency localization of this kernel. The second derivative occurring in the oscillatory component is a trade-off between the number of vanishing moments used to filter out smooth polynomial parts and ringing-like artifacts that may occur after strong quantization. It is also motivated by the presence of second derivative-like filtering in the early stages of the human visual system [3].

The generating function described above is however not able to efficiently represent the low frequency characteristics of the image at low rates. There are two main options to capture these features: (i) to perform a low-pass filtering of the image and send a quantized and downsampled image or (ii) to use an additional dictionary capable of representing the low frequency components. This second approach has also the advantage of introducing more *natural* artifacts at very low bit rate, since it tends to naturally distribute the available bits between the low and high frequencies of the image. A second subpart of the proposed dictionary is therefore formed by Gaussian functions, in order to keep the optimal joint space-frequency localization. The second generating function of our dictionary can be written as :

$$g(\vec{p}) = \frac{1}{\sqrt{\pi}}\exp(-(x^2+y^2))\,, \tag{2}$$

where the Gaussian has been multiplied by a constant in order to have $||g(\vec{p})|| = 1$.

2.3. Anisotropy and orientation

Anisotropic refinement and orientation is eventually obtained by applying meaningful geometric transformations to the generating functions of unit L^2 norm, g, described here-above. These transformations can be represented by a family of unitary operators $U(\gamma)$, and the dictionary is thus expressed as :

$$\mathscr{D} = \{U(\gamma)g,\; \gamma \in \Gamma\}\,, \tag{3}$$

for a given set of indexes Γ. Basically this set must contain three types of operations: (i) Translations $\vec{b}$, to move the atom all over the image, (ii) Rotations θ, to locally orient the atom along contours and (iii) Anisotropic scaling $\vec{a} = (a_1, a_2)$, to adapt to contour smoothness. A possible action of $U(\gamma)$ on the generating atom g is thus given by :

$$U(\gamma)g = \mathscr{U}(\vec{b},\theta)D(a_1,a_2)g \tag{4}$$

where $\mathscr{U}$ is a representation of the Euclidean group,

$$\mathscr{U}(\vec{b},\theta)g(\vec{p}) = g\big(r_{-\theta}(\vec{p}-\vec{b})\big)\,, \tag{5}$$

r_θ is a rotation matrix, and D acts as an anisotropic dilation operator :

$$D(a_1,a_2)g(\vec{p}) = \frac{1}{\sqrt{a_1 a_2}}g\big(\frac{x}{a_1},\frac{y}{a_2}\big)\,. \tag{6}$$

It is easy to prove that such a dictionary is overcomplete using the fact that, under the restrictive condition $a_1 = a_2$, one gets 2-D continuous wavelets as defined in [4]. It is also worth stressing that, avoiding rotations, the parameter space is a group studied by Bernier and Taylor [5]. The advantage of such a parametrization is that the full dictionary is invariant under translation and rotation. Most importantly, it is also invariant under isotropic scaling, e.g. $a_1 = a_2$. These properties will be exploited for spatial transcoding in the next sections.

2.4. Dictionary

Since the structured dictionary is built by applying geometric transformations to a generating mother function g, the atoms are therefore indexed by a string γ composed of five parameters: translation $\vec{b}$, anisotropic scaling $\vec{a}$ and rotation θ.

For practical implementations, all parameters in the dictionary must be discretized. For the Anisotropic Refinement (AR) Atoms sub-dictionary, the translation parameters can take any positive integer value smaller than the image dimensions. The rotation parameter varies by increments of $\frac{\pi}{18}$, to ensure the overcompleteness of the dictionary. The scaling parameters are uniformly distributed on a logarithmic scale from one up to an eighth of the size of the image, with a resolution of one third of octave. Finally, to further constrain the dictionary size, the atoms are always smaller along the second derivative of the Gaussian function than along the Gaussian itself, thus maximizing the similarity of the dictionary elements with edges in images. For the Gaussian (low frequency) sub-dictionary, the translation parameters vary exactly in the same way as for the AR atoms, but the scaling is isotropic and varies from $\frac{min(W,H)}{32}$ to $\frac{min(W,H)}{4}$ on a logarithmic scale with a resolution of one third of octave (W and H are image width and height respectively). Lastly, due to isotropy, rotations are obviously useless for this kind of atoms.

2.5. Quantization and Coding

The encoder presented here performs quantization *a posteriori*, which does not allow to re-inject quantization error in the Matching Pursuit algorithm, contrarily to [8]. But in this case, the signal expansion does not depend on the quantization, and hence the coding rate. *A posteriori* quantization and coding allow for one single expansion to be encoded at different target rates. It uses a quantization method specifically adapted to the Matching Pursuit expansion characteristics, that takes benefit from the fact that the Matching Pursuit coefficient energy is upper-bounded by an exponential curve, decaying with the coefficient order. The quantization algorithm strongly relies on this property, and the exponential upper-bound directly determines the quantization range of the coefficient magnitude, while the coefficient sign is reported on a separate bit. The number of quantization steps is then computed as the solution of a rate-distortion optimization problem [9].

Efficient coding of Matching Pursuit parameters has been proposed in [2], with a smart scanning of atom positions within image blocks. The coder presented in this section however aims at producing fully scalable image streams. Such a requirement truly limits the options in the entropy coding stage, since the atom order is given by the magnitude of their coefficients, as discussed in the previous paragraph. The scalable encoder therefore implements an adaptive arithmetic coding, with independent contexts for position, scale, rotation and coefficient parameters. The core of the arithmetic coder is based on [6], with the probability update method from [7]. As the distribution of the atom parameters (e.g., positions or scales) is dependent on the image to be coded, the entropy coder first initializes the symbol probabilities to a uniform distribution. The encoded parameters are then sent in their natural order, which results in a progressive stream, that can eventually be cut at any point to generate rate scalable streams.

3. Coding Performance

The objective of this section is to emphasize the potential of redundant expansions for low rate compression of natural images, even though the Matching Pursuit encoder is not fully optimized yet. Figure 1 presents a comparison between detailed views of images compressed with Matching Pursuit, and respectively JPEG-2000[1]. It can be seen that the PSNR rating is in favor of JPEG-2000, which is not completely surprising since a lot of research efforts are being put in optimizing the encoding in JPEG-2000 like schemes. Interestingly, however, the image encoded with Matching Pursuit is visually more pleasant than the JPEG-2000 version. The coding artifacts are quite different, and the degradations due to Matching Pursuit are less annoying to the Human Visual System, than the ringing due to wavelet coding at low rate. JPEG-2000 has difficulties to approximate the 2-D oriented contours, which are generally the most predominant components of natural images. And this is clearly one of the most important advantages of the Matching Pursuit coder built on anisotropic refinement, which is really efficient to code edge-like features.

Finally, the proposed encoder performs reasonably well in terms of rate-distortion performance, especially at low rates, as it can be seen on Figure 2. When the rate increases, the saturation of the quality can be explained by the limitations of redundant transforms for high rate approximations. Hybrid coding schemes could provide helpful solutions for high rate coding.

[1] All results have been generated with the Java implementation available at http://jj2000.epfl.ch/, with default settings

4. High adaptivity

As outlined in the previous section, one of the main advantages of the MP coder is to provide highly flexible streams at no additional cost. This is very interesting in nowadays visual applications involving transmission and storage, like database browsing or pervasive image and video communications. The challenge in adaptive coding is to build a stream decodable at different resolutions without any significant loss in quality by comparison to non-adaptive streams. In other words, adaptive coding is efficient if the stream does not contain data redundant to any of the target resolutions.

4.1. Spatial adaptivity

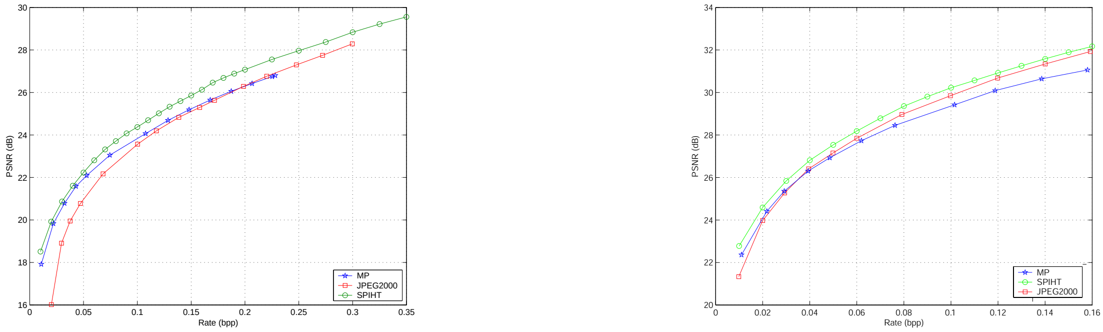
Due to the structured nature of our dictionary, the Matching Pursuit stream provides inherent spatial adaptivity. The group law of the similitude group of $\mathbb{R}^2$ indeed applies [4] and allows for invariance with respect to *isotropic* scaling of α, rotation of Θ and translation of $\vec{\beta}$. Therefore, when the compressed image $\hat{f}$ is submitted to any combination of these transforms (denoted here by the group element η), the indexes of the MP stream can simply be transformed with help of the group law :

$$\begin{aligned} \mathscr{U}(\eta)\hat{f} &= \sum_{n=0}^{N-1} \langle g_{\gamma_n} | \mathscr{R}^n f \rangle \, \mathscr{U}(\eta) g_{\gamma_n} \\ &= \sum_{n=0}^{N-1} \langle g_{\gamma_n} | \mathscr{R}^n f \rangle \, \mathscr{U}(\eta \circ \gamma_n) g \, . \end{aligned} \tag{7}$$

In the above expression $\gamma_n = (\vec{a}_n, \theta_n, \vec{b}_n)$ represents the parameter strings of the atom encoded at iteration n, with scaling $\vec{a}_n$, rotation θ_n and translation $\vec{b}_n$, and $\eta = (\alpha, \Theta, \vec{\beta})$

(a) MP: 31.0610 dB (b) JPEG-2000 : 31.9285 dB

Figure 1. Detail view, *Lena* (512 x 512) encoded at 0.16bpp.



Cameraman, 256 x 256 *Lena*, 512 x 512

Figure 2. Distortion-rate performance for JPEG-2000, SPIHT and the proposed MP coder, for common test images.

represents the geometric transformation that is applied to the set of atoms. The decoder can apply the transformations to the encoded bitstream simply by modifying the parameter strings of the unit-norm atoms, according to the group law of similitude, where

$$(\vec{a},\theta,\vec{b}) \circ (\alpha,\Theta,\vec{\beta}) = \left(\alpha\cdot\vec{a},\theta+\Theta,\vec{b}+\alpha\cdot r_{\Theta}\vec{\beta}\right). \quad (8)$$

In other words, if $\eta_\alpha = (\alpha,0,0)$ denotes the isotropic scaling by a factor α, the bitstream of an image of size $W \times H$, after entropy decoding, can be used to build an image at any resolution $\alpha W \times \alpha H$ simply by multiplying positions and scales by the scaling factor α (from Eq. (8) and (4)). The coefficients have also to be scaled with the same factor to preserve the energy of the different components. Finally, the scaled image is obtained by :

$$\mathcal{U}(\eta_\alpha)\hat{f} = \alpha \sum_{n=0}^{N-1} c_{\gamma_n} g_{\eta_\alpha\circ\gamma_n}. \quad (9)$$

The modified atoms $g_{\eta_\alpha\circ\gamma_n}$ are simply given by Eq. (4) to (6), where $\vec{b}$ and $\vec{a}$ are respectively replaced by $\alpha\,\vec{b}$ and $\alpha\,\vec{a}$. It is worth noting that the scaling factor α can take any positive real value, as long as the scaling is isotropic. The simple spatial adaption procedure is illustrated in Fig. 3, where the encoded image of size 256×256 has been rescaled with irrational factors $\sqrt{\frac{1}{2}}$ and $\sqrt{2}$.

4.2. Rate adaptivity

Matching Pursuit offers an intrinsic multiresolution advantage, which can be efficiently exploited for rate adaptivity. The coefficients are by nature exponentially decreasing so that the stream can simply be truncated at any point to provide a SNR-adaptive bitstream, while ensuring that the most energetic atoms are kept. The simplest possible rate adaption algorithm that uses the progressive nature of the Matching Pursuit stream works as follows. Assume an image has been encoded at a high target bit-rate R_b. The encoded stream is then restricted to lower bit budgets r_k, $k = 0,\ldots,K$ by simply dropping the bits r_k+1 to R_b. This simple rate-adption, or filtering operation is equivalent to dropping the last iterations in the MP expansion, focusing on the highest energy atoms.

Figure 4 illustrates the rate adaptivity performance of the MP encoder. Images have been encoded with MP at a rate of 0.17 bpp and truncated to lower rates r_k. For comparison, the bitstream has also been encoded directly at the different target rates r_k. Both optimal and truncated rate-

Figure 3. *Lena* image of size 256×256 encoded with MP at 0.3bpp (center), and decoded with scaling factors of $\sqrt{\frac{1}{2}}$ (left) and $\sqrt{2}$ (right).

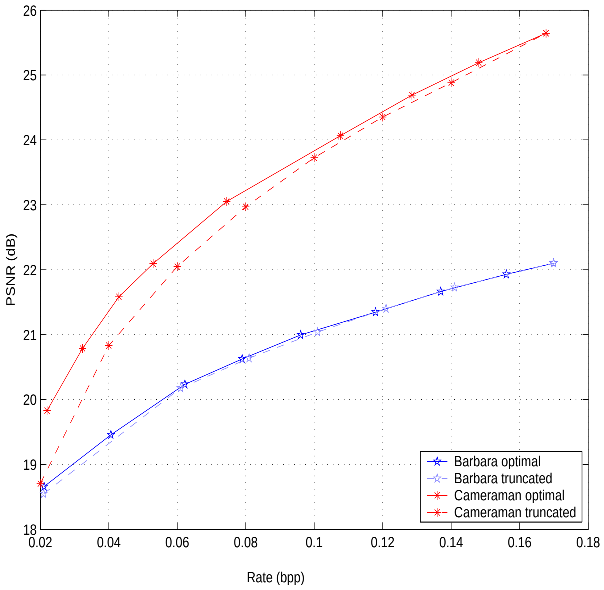


Figure 4. Rate-distortion characteristics for MP encoding of the 256x256 *Barbara* and *Cameraman* images at 0.17 bpp, and truncation/decoding at different (smaller) bit rates.

distortion curves are quite close, which shows that a simple rate adaption method, though quite basic, is very efficient.

5. Conclusions

A image coder based on Matching Pursuit signal expansion over a redundant dictionary of atoms has been presented in this paper. A specific dictionary, based on anisotropic refinement of edge-like functions, has been shown to provide sparse image representation, since it concentrates on representing contour-like characteristics that are dominant in natural images. Besides interesting compression properties at low rate, the proposed coder is able to generate a flexible and scalable stream, and thus represents an interesting alternative to coders based on classical transforms.

References

[1] B. F. and M. S., "Matching pursuit of images," in *Proc. IEEE Int. Conf. Image Proc.*, vol. I, Washington DC, October 1995, pp. 53–56.

[2] Neff R. and Zakhor A., "Very Low Bit-Rate Video Coding Based on Matching Pursuits," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 158–171, February 1997.

[3] M. D., *Vision*. San Francisco: Freeman, 1982.

[4] Antoine J.-P., Murenzi R. and Vandergheynst P., "Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns," *Applied and Computational Harmonic Analysis*, vol. 6, pp. 314–345, 1999.

[5] Bernier D. and Taylor K., "Wavelets from square-integrable representations," *SIAM Journal on Mathematical Analysis*, vol. 27, no. 2, pp. 594–608, 1996.

[6] Witten I.H., Neal R.M. and Cleary J.G., "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, June 1987.

[7] Duttweiler D.L.and Chamzas C., "Probability estimation in arithmetic and adaptive-huffman entropy coders," *Image Processing, IEEE Transactions on*, vol. 4 Issue: 3, pp. 237–246, Mar 1995.

[8] De Vleeschouwer C. and Zakhor, A., "In-loop atom modulus quantization for matching pursuit and its application to video coding," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1226–1242, October 2003.

[9] Frossard P., Vandergheynst P., Figueras i Ventura R. M. and Kunt M., "A Posteriori Quantization of Progressive Matching Pursuit Streams," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 525–535, February 2004.