Image Compression with Learnt Tree-Structured Dictionaries

Gianluca Monaci, Philippe Jost, Pierre Vandergheynst
Signal Processing Institute (ITS)
Swiss Federal Institute of Technology (EPFL)
Lausanne 1015, Switzerland
{gianluca.monaci, philippe.jost, pierre.vandergheynst}@epfl.ch

***Abstract*— In the present paper we propose a new framework for the construction of meaningful dictionaries for sparse representation of signals. The dictionary approach to coding and compression proves very attractive since decomposing a signal over a redundant set of basis functions allows a parsimonious representation of information. This interest is witnessed by numerous research efforts that have been done in the last years to develop efficient algorithm for the decomposition of signals over redundant sets of functions. However, the effectiveness of such methods strongly depends on the dictionary and on its structure. In this work, we develop a method to learn overcomplete sets of functions from real-world signals. This technique allows the design of dictionaries that can be adapted to a specific class of signals. The found functions are stored in a tree structure. This data structure is used by a Tree-Based Pursuit algorithm to generate sparse approximations of natural signals. Finally, the proposed method is considered in the context of image compression. Results show that the learning Tree-Based approach outperforms state-of-the-art coding technique.**

I. Introduction

In order to compress a signal, it is desirable to represent information using as few terms as possible, satisfying constraints of reconstruction quality or bandwidth or hardware limitations. To succeed in this task, a flexible, parsimonious representation of the information has to be found. Projecting a signal on a suitable, overcomplete, set of functions can provide a sparse representation of the signal. The information can therefore be represented with few, salient components.

The Matching Pursuit (MP) algorithm [1] represents an interesting method to decompose a signal over a redundant dictionary. It implements a greedy procedure that iteratively projects a signal over a set of basis functions called *atoms*, and, at each step, finds the waveform that best matches the signal. The approximation power of such a technique largely depends on the choice of the employed set of atoms.

Dictionary design is a challenging research topic. One would like to be able to build dictionaries with limited complexity, preserving at the same time good approximation power and representation flexibility. Several works in the last years addressed this topic especially in the field of MP video coding [2], [3], [4], searching a tradeoff between coding efficiency and encoding complexity.

This work was supported by the Swiss NFS through the IM.2 National Center of Competence for Research.

In this work, we propose a new framework to build meaningful structured dictionaries for the sparse representation of natural signals. An overcomplete dictionary is firstly learnt from real-world signals, using a biologically inspired approach [5]. The searched set of atoms should be able to represent efficiently information with as few terms as possible. Then, the obtained set of functions is organized in a hierarchical tree structure, that allows the design of fast tree-based greedy algorithm [6]. No *a priori* constraint is imposed on the structure of the dictionary, allowing great flexibility in its design. The learning algorithm chooses the atoms that better represent the set of signals on which the training is performed. Thus, it is easy to adapt the set of basis waveforms to a given family of signals with common characteristics, strongly reducing the cardinality of the dictionary without considerable loss in terms of representation power.

In order to prove the effectiveness of this approach, we consider the proposed methodology in the context of compression of human faces images. The results obtained with the presented approach are shown, and they are compared with those provided by state-of-the-art coding technique.

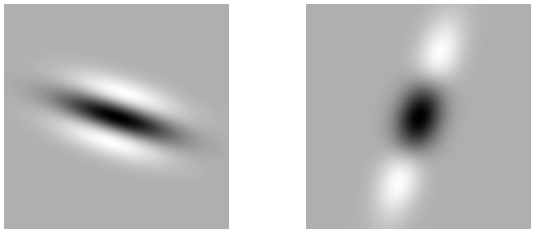
The paper is structured as follows: in Section II, the signal model adopted is introduced, and the learning algorithm is described, together with the method used for the construction of the tree. Section III presents the employed coding strategies and Section IV shows the experimental results. Finally, in Section V conclusions are drawn and possible future developments are depicted.

II. Building the Dictionary

The construction of the structured dictionary is performed in three main steps. Firstly, we define a linear, overcomplete model for the approximation of signals. Then we learn from a training dataset a redundant codebook that is able to accurately represent signals in a parsimonious way. Finally, we organize the learnt dictionary into a hierarchical tree structure, in order to effectively manage the set of learnt basis functions.

A. Signal Model

If we assume that a signal $s(\vec{x})$ can be represented as a linear superposition of functions $g_{\gamma_i}(\vec{x})$ belonging to an



(a) (b)

Fig. 1. Anisotropic Refinement atoms. Positive values are depicted in white and negative values in black. (a) Atom with scale s_x smaller than scale s_y. It is evident its edge-detector behavior. (b) *Pathological* atom: the s_x scale is bigger than the s_y scale and the function looses its edge-detector characteristic.

overcomplete dictionary $\mathcal{D}$, it is possible to write:

$$s(\vec{x}) \approx \sum_{i=0}^{N-1} c_i g_{\gamma_i}(\vec{x}), \tag{1}$$

where c_i are the coefficients and N is the number of waveforms used to approximate the signal. In the following of this work, we will consider bi-dimensional signals. Since we have to deal with natural images, the functions $g_{\gamma_i}(\vec{x})$ are *Anisotropic Refinement* (AR) atoms [7], obtained by applying a set of geometric transformations to the generating function $g(\vec{x})$ of unit L^2 norm. The dictionary $\mathcal{D}= \{U_\gamma g(\vec{x}), \gamma \in \Gamma\}$ for a given set of indexes Γ, is thus built by applying the transformations U_γ to the function g. The considered geometric transformations are anisotropic scaling s_x and s_y, translations t_x and t_y over the 2-D plane and rotation θ. The waveform g is a combination of a Gaussian with the second derivative of a Gaussian and has been chosen since it has been shown to be able to efficiently represent edges [7]. An example of Anisotropic Refinement atom is shown in Fig. 1(a).

B. Learning the Dictionary

The parameters that identify the geometric transformations of the basis waveforms are learnt from a set of natural images, using a method inspired by [5]. The learning is performed by minimizing the objective function

$$E = \sum_{\vec{x}} \left[s(\vec{x}) - \sum_{i=0}^{N-1} c_i g_{\gamma_i}(\vec{x}) \right]^2 + \\ + \lambda_1 \sum_{i=0}^{N-1} S(c_i) + \lambda_2 \sum_{i=0}^{N-1} P(i) \tag{2}$$

with respect to the coefficients c_i and the geometric parameters of the atoms s_{x_i}, s_{y_i}, t_{x_i} t_{y_i} and θ_i, with $i = 0, \ldots, N-1$ and N number of atoms used for the representation. The parameters λ_1 and λ_2 are weights that determine the contribution to E of the second and third term respectively. The first part of the functional E indicates the square error between the original image and the reconstructed one. The second term encourages the sparsity of the representation, attributing a high cost to coefficients having large values. In this case we set $S(x) = \log(1+x^2)$. Finally, the third part of the functional is a penalty term that takes into account the considered application. It is interesting to remark that this term can be composed of an arbitrary number of functionals that can incorporate conditions on any aspect of the desired dictionary. In the present case, we pose

$$P(i) = \arctan(k(s_{x_i} - s_{y_i})) + \alpha \text{Mask}(t_{x_i}, t_{y_i}). \tag{3}$$

The first part of the expression encourages, for each atom, the scale s_{x_i} to be smaller then s_{y_i}. This term attempts to reduce the introduction of *pathological* atoms that do not have the desired characteristics of band-pass, edge-detector functions (see Fig. 1(b)). The parameter k determines the slope of the inverse tangent and here is set equal to 5. The second part of $P(i)$ encourages the dictionary elements to be learnt from image locations that are considered more informative. Clearly, *a priori* information about the training images is required here, in order to correctly set this term. The function Mask(x, y) is defined such that it attributes penalty 0 to locations around the zones of interest and penalty 1 to positions outside these regions. The parameter α indicates the importance of this term with respect to the others.

Once one has fixed the number N of atoms to be learnt from a training image, the functional E has to be minimized in a space of dimension $6 \times N$, since each atom has 6 free parameters c_i, t_{x_i}, t_{y_i}, θ_i, s_{x_i}, s_{y_i}. The objective function is optimized using a Sequential Quadratic Programming (SQP) algorithm [8].

C. Dictionary Structuring

The interest of drawing a structure in the learnt dictionary comes from the fact that such an organization is required, in order to handle in a fast, efficient way the collected information. Another aspect comes from the fact that a well designed structure can reveal the intrinsic properties of a dictionary.

The learnt dictionary $\mathcal{D}$ is stored in a hierarchical way in a tree. A node holds a subspace of $\mathcal{D}$ as orthogonal as possible to its siblings. Each node $N_{i,j}$ at level i and position j has M children and is characterized by the list $L_{i,j}$ of indices of atoms contained in the subtree spanned by $N_{i,j}$. A centroid $c_{i,j}$ is assigned to the node $N_{i,j}$ that represents the functions of the dictionary present in the corresponding subtree. The centroid belongs to the span of the atoms it represents and has the following property:

$$\min_{i \in L_{l,n}} d(g_{\gamma_i}, c_{l,n}) \geq \max_{j \in \mathcal{D} \setminus L_{l,n}} d(g_{\gamma_j}, c_{l,n}). \tag{4}$$

In Eq. (4) the function $d(x, y)$ is a measure of the distance between two atoms, taking values between 0 and 1. Here we use $d(x, y) = 1 - | < x, y > |$.

Using the measure distance, it is naturally possible to compute the mean distance $D_{l,m}$ between a centroid and the

atoms it represents:

$$D_{l,m} = \frac{1}{n_{l,m}} \sum_{i \in L_{l,m}} d(g_{\gamma_i}, c_{l,m}), \tag{5}$$

where $n_{l,m}$ is the cardinality of $L_{l,m}$. The tree is built by making use of recursive calls to a *k-means* clustering algorithm that divides a group of atoms into a set of M clusters. The quality of a clustering, $Q_{l,m}$, is defined as the mean of all the distance between the centroids and the associated atoms:

$$Q_{l,m} = \frac{1}{M} \sum_{i=0}^{M-1} D(l+1, mM+i). \tag{6}$$

The computation is over when the decrease of $Q_{l,m}$ is smaller than a threshold $\epsilon = 10^{-6}$.

The tree organization of the dictionary allows the design of a fast, efficient and flexible algorithm to built sparse representations of images from tree-structured dictionaries.

III. Image Coding

The original image is decomposed into a low frequency part and a high frequency component. The low-pass approximation is obtained by filtering and downsampling the original image. It is differentially quantized using DPCM and then it is entropy coded with an arithmetic coding method. The high frequency part of the image is obtained by subtracting to the image itself the upsampled, filtered version of the low-pass approximation. This residual high frequency image is coded using a Tree-Based Pursuit technique.

The Tree-Based greedy algorithm finds at each step the best path through the tree-structured dictionary, choosing the atom from the codebook that best matches the input image. Let $R^N I$ be the residual image after N steps of the algorithm. The method firstly performs a full search over $R^N I$ for the set of M root nodes, returning the centroid c_B that best matches the residual image and its position (x_B, y_B). Then, a full search over a window of size $W \times W$ (here $W = 3$) around the position (x_B, y_B) is performed, considering the subtree referring to c_B. The algorithm executes the search descending through the tree down to the leaves level, where the atom that best matches $R^N I$ is found. Experiments show that this modified pursuit method is up to 150 times faster than a classical full-search algorithm, with a negligible loss of approximation accuracy [6].

The real coefficients obtained from the dictionary expansion are quantized using an exponential *a posteriori* method [9]. The quantized coefficients, the positions and the indexes of the atoms are then coded with an adaptive arithmetic coder.

The learning procedure allows to drastically reduce the cardinality of the dictionary, thus reducing the number of bits needed to index the atoms and speeding up the coding and decoding processes. At the same time, since the algorithm learns the most representative image features, it is possible to code visual information with considerable quality.

(a) (b) (c)

Fig. 2. (a) Example of image from the database of faces, (b) "sphered" training image and (c) corresponding function Mask(x, y). The values of the Mask function fill the linear gray scale, from 0 (black) to 1 (white).

IV. Experimental Results

The proposed approach has been applied to the compression of images representing human faces. The learning process is accomplished using images taken from the AT&T database of faces [10]. An example of database image is shown in Fig. 2(a). The pictures have been resized to 80×96 pixels and they have been filtered using a "sphering" function, in order to speed up the learning process. The filter that we use here is the same employed by Olshausen and Field [5], it has a frequency response

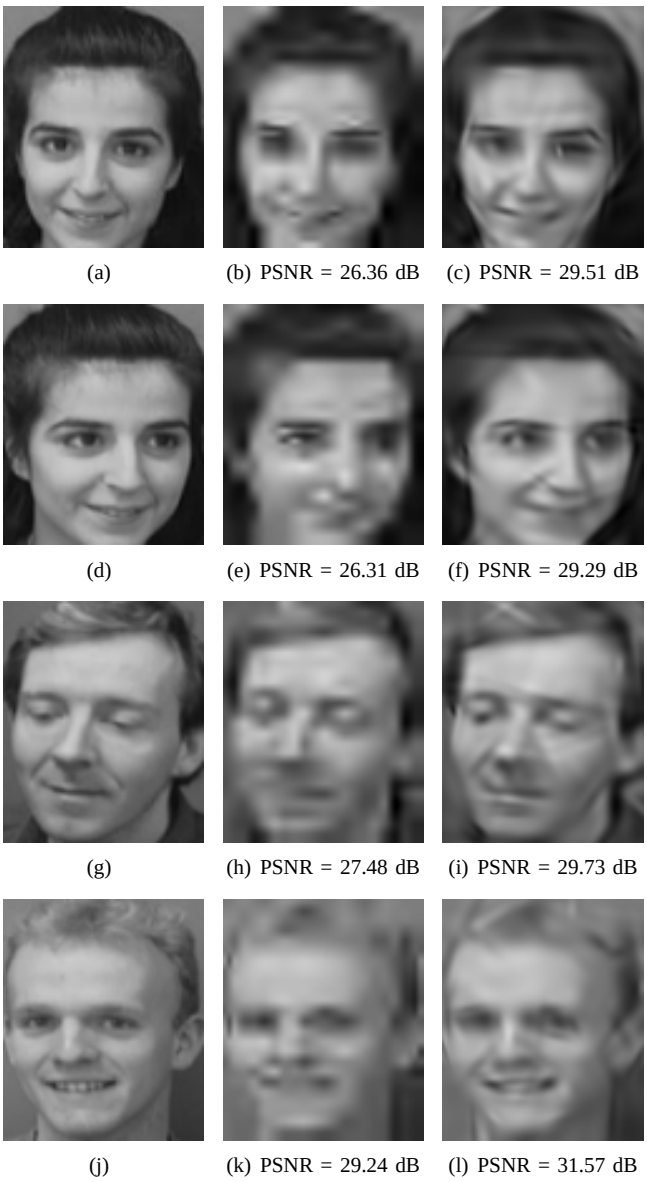
$$H(f) = f e^{-(f/f_0)^4}, \tag{7}$$

where f_0 is the cut off frequency of 200 cycles/picture. Since natural images generally exhibit a power spectrum that goes like $1/f^2$, there are great inequities in variance along the different directions of the image space. Thus, variance equalization can be accomplished using a circular symmetric filter with a frequency response that behave like f. Such a filter, however, will increase the high frequencies, that are typically corrupted by noise and artifacts due to the square sampling scheme used to digitalize images. Thus, we attenuate the highest frequencies using a circular exponential low-pass filter that goes like $e^{-(f/f_0)^4}$. An example of a training image preprocessed with the filter $H(f)$ is shown in Fig. 2(b).

Since in the present work we are dealing with human faces images, the zones of the eyes, of the nose and of the mouth are considered the most relevant areas. Exploiting this *a priori* knowledge about the considered class of images, the masking function of Eq. (3) is straightforwardly defined. An example of the function Mask(x, y) for the training image of Fig. 2(a) is shown in Fig. 2(c).

The parameter α in Eq. (3) is set equal to 3, while in Eq. (2) the parameter λ_1 is imposed to be equal to $0.14\sigma_I$, where σ_I is the standard deviation of the considered image, and λ_2 is set to the same value of λ_1. Experiments using different combinations of the weights λ_1, λ_2 and α have been run, without significant variations in the results.

A dictionary of AR atoms has been deduced from seven image faces taken from the dataset. From each image, $N = 82$ representative atoms have been learnt. The choice of the value



(a) (b) PSNR = 26.36 dB (c) PSNR = 29.51 dB

(d) (e) PSNR = 26.31 dB (f) PSNR = 29.29 dB

(g) (h) PSNR = 27.48 dB (i) PSNR = 29.73 dB

(j) (k) PSNR = 29.24 dB (l) PSNR = 31.57 dB

Fig. 3. Original images (a, d, g, j), and images coded at 0.3 bpp with JPEG2000 (b, e, h, k) and with the proposed Tree-Based scheme (c, f, i, l).

of N is based on heuristic and basically takes into account the computational complexity of the minimization algorithm (the functional E is minimized in a space of dimension $6 \times N$) and the number of image features one wants to extract. The resulting set of functions has been organized in a quaternary tree ($M = 4$), on which the Tree-Based Pursuit algorithm is based.

In Fig. 3, a set of results obtained with 80×96 images is shown, together with the corresponding PSNR values expressed in dB. The images are coded at a rate of 0.3 bpp using the standard JPEG2000 algorithm [11], [12] (b, e, h, k) and the proposed Tree-Based Pursuit scheme (c, f, i, l). The settings used for the JPEG2000 encoder are the standard one, with 5 decomposition levels for the Discrete Wavelet Transform.

The first image (Fig. 3(a)) was present in the training dataset, while the second one (Fig. 3(d)), that is a different picture of the same subject, and the last two images (Fig. 3(g, j)) belong to the database of faces but were not considered for the learning. Our method clearly outperforms JPEG2000 both in terms of PSNR and visual quality. As expected, the Tree-Based Pursuit scheme performs better when coding an image that belongs to the training set (3.15 dB of gain with respect to JPEG2000). Similar performances ($+2.98$ dB) are achieved on the second image, while in the last two examples the gain is of 2.25 dB and 2.33 dB.

V. Conclusions

In this work, we faced the problem of coding images at very low bit rates using sparse decomposition over redundant sets of atoms. A new framework for the construction of meaningful dictionaries has been introduced, that allow to employ efficient and effective image compression techniques. Firstly, a codebook of basis waveforms is learnt from a set of images by minimizing a cost function, imposing sparsity and good representation properties. The learnt atoms have been grouped into clusters in order to organize them in a hierarchical tree structure. Such a structure has allowed the design of a fast greedy Tree-Based Pursuit algorithm for the compression of images. The proposed scheme demonstrated to achieve good performances, that can however be improved introducing a coding strategy for the atoms, that would take into account the peculiar tree-structure of the dictionary.

References

[1] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," in *IEEE Transactions on Signal Processing*, 1993, vol. 41, pp. 3397–3415.

[2] R. Neff and A. Zakhor, "Dictionary approximation for matching pursuit video coding," in *Proceedings of IEEE ICIP*, 2000, vol. 2, pp. 828–831.

[3] C. de Vleeschouwer and B. Macq, "New dictionaries for matching pursuit video coding," in *Proceedings of IEEE ICIP*, 1998, vol. 1, pp. 764–768.

[4] Y.-T. Chou, W.-L. Hwang, and C.-L. Huang, "Gain-shape optimized dictionary for matching pursuit video coding," in *Signal Processing*, 2003, vol. 83, pp. 1937–1943.

[5] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," in *Vision Resesarch*, 1997, vol. 37, pp. 3311–3327.

[6] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-based pursuit," Tech. Rep. TR-ITS 2004.13, EPFL, 1015 Ecublens, 2004.

[7] P. Vandergheynst and P. Frossard, "Efficient image representation by anisotropic refinement in matching pursuit," in *Proceedings of IEEE ICASSP*, 2001, vol. 3, pp. 1757–1760.

[8] C. Lawrence, J. L. Zhou, and A. L. Tits, "User's guide for CFSQP *Version 2.5*," Tech. Rep. TR-94-16r1, Electrical Engineering Dept. and Institute for System Research, Univ. of Maryland, College Park, 1997.

[9] P. Frossard, P. Vandergheynst, R. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," in *IEEE Transactions on Signal Processing*, 2004, vol. 52, pp. 525–535.

[10] "AT&T Database of Faces," `http://www.uk.research.att.com/facedatabase.html`.

[11] "JPEG2000 implementation in Java," `http://jpeg2000.epfl.ch`.

[12] D. Santa-Cruz, T. Ebrahimi, J. Askelöf, M. Larsson, and C. Christopoulos, "JPEG2000 still image coding vs other standards," in *Proceedings of SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII*, 2000, vol. 4115, pp. 446–454.