LEARNING STRUCTURED DICTIONARIES FOR IMAGE REPRESENTATION

Gianluca Monaci, Pierre Vandergheynst

Signal Processing Institute (ITS)
Swiss Federal Institute of Technology (EPFL)
1015, Lausanne, Switzerland
e-mail: {gianluca.monaci, pierre.vandergheynst}@epfl.ch

ABSTRACT

The dictionary approach to signal and image processing has been massively investigated in the last two decades, proving very attractive for a wide range of applications. The effectiveness of dictionary-based methods, however, is strongly influenced by the choice of the set of basis functions. Moreover, the *structure* of the dictionary is of paramount importance regarding efficient implementation and practical applications such as image coding. In this work, an overcomplete code for sparse representation of natural images has been learnt from a set a real-world scenes. The functions found have been organized into a hierarchical structure. We take advantage of this representation of the dictionary, adopting a tree-structured greedy algorithm to build sparse approximations of images. Using this procedure, no a-priori constraint is imposed on the structure of the dictionary, allowing great flexibility in its design and lower computational complexity.

1. INTRODUCTION

For many applications in the field of signal and image processing, it is desirable to have an efficient, sparse representation of information, in particular for computational cost reasons. Redundant systems like Matching Pursuit (MP) [1] are able to produce such a sparse representation and allow for great freedom in designing dictionaries with prescribed properties, or adapted to particular signal structures or even to communication application requirements [2].

The effectiveness of approaches based on expanding the signal over a redundant set of functions, however, largely depends on the choice of the dictionary of functions itself. Thus, the question that arises at this point is how to built effective, meaningful sets of functions, that are able to generate sparse representations of images.

Until today, the methods developed to deal with natural images impose somehow a structure in the representation. This means that, being able to efficiently process, encode or transmit image data imposes strong constraints on the representation framework. As an example, we can cite the wavelet transform whose success in image coding largely depends on its hierarchical tree structure [3]. This however limits its flexibility, as witnessed by the limited freedom in choosing the properties of a wavelet basis.

In contrast, a promising approach consists in learning a set of visual primitives from training images, and then organize the learnt dictionary in a useful and meaningful structure. In the field of computational vision, several efforts have been done to try to deduce sets of functions that are able to efficiently represent natural images. Particularly interesting and successful methods are those designed to learn sparse codes [4, 5] or independent components (ICA) [6, 7] of natural images. The sparse approach, however, seems to be more plausible than the ICA one from a biological [4, 8] and mathematical [9] point of view.

In this work, we study the characteristics of real world scenes to build an *ad hoc* library of functions for the sparse representation of natural images. The image is assumed to be a linear superposition of functions belonging to an overcomplete library. The functions used in this study are *Anisotropic Refinement* atoms, that have been used in [10] as basis functions for a Matching Pursuit algorithm. Here the parameters of such waveforms are learnt from a set of natural images, using a method inspired by [4].

Once the learning process is accomplished, the resulting huge amount of data must be organized. Basically, we want to identify the essential, most significative structures underlying the learnt dictionary. This would allow to arrange it in a tractable structure. To this end, the obtained atoms are clustered and organized in a tree representation, like the one proposed in [11]. Atoms are grouped into clusters that represent subspaces of the whole learnt dictionary, which are as orthogonal as possible one to the others.

The obtained tree-structured dictionary allows to design a coarse-to-fine greedy algorithm to build sparse approximations of natural images. This algorithm has the non negligible advantage of being less complex and much faster than

Both authors acknowledge the support of the Swiss NFS through the IM.2 National Center of Competence for Research

0-7803-8554-3/04/$20.00 ©2004 IEEE.

a classical MP method.

The great advantage of the proposed approach is that no *a-priori* hypothesis on the structure of the dictionary is done, except for the shape of the basis waveforms. This permits a great flexibility in the design of the dictionary, that is thus able to adapt to the structures present in images.

2. IMAGE MODEL

As a first step, we define the image model used in this work. An image $I(x, y)$ is supposed to be represented as a linear summation of basis functions $g_{\gamma_i}(x, y)$:

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

where c_i are the coefficients and N is the number of basis functions used to reconstruct the image.

The functions $g_{\gamma_i}(x, y)$ are created by applying geometric transformations to a generating function $g(x, y)$ of unit L^2 norm. Basically, the required transformations are translations over the image plane by t_x and t_y, rotations by θ, and scaling by s_x and s_y. It is easy to demonstrate that the set of atoms built in such a way is overcomplete [10].

The generating function g should be able to represent well edges on the 2-D plane and thus should behave like a smooth scaling function in one direction and like a wavelet in the orthogonal one. In this case, the function g is a Gaussian along one axis and the second derivative of a Gaussian along the orthogonal one. This set of atoms has been chosen because it is able to efficiently represent contours and edges, as shown in [10]. An Anisotropic Refinement atom g_γ rotated by θ, translated by t_x and t_y, and anisotropically scaled by s_x and s_y can thus be written as:

$$g_\gamma(u, v) = \frac{C}{\sqrt{s_x s_y}}(2 - 4u^2)\exp(-(u^2 + v^2)), \tag{2}$$

where C is a normalization constant and

$$u = \frac{cos\theta(x - t_x) + sin\theta(y - t_y)}{s_x} \tag{3}$$

$$v = \frac{-sin\theta(x - t_x) + cos\theta(y - t_y)}{s_y}. \tag{4}$$

3. LEARNING OF THE DICTIONARY

Our aim is to learn the parameters of the atoms ($t_{x,i}$, $t_{y,i}$, θ_i, $s_{x,i}$ and $s_{y,i}$) that best represent an image, but that also enforce the sparseness of the representation. The learning can thus be accomplished by minimizing an objective function composed of three terms:

$$E = \sum_{x,y}\left[I(x, y) - \sum_{i=0}^{N-1} c_i g_{\gamma_i}(x, y)\right]^2 + \\ + \lambda_1 \sum_{i=0}^{N-1} S(c_i) + \lambda_2 \sum_{i=0}^{N-1} P(s_{x_i}, s_{y_i}), \tag{5}$$

with respect to the parameters t_{x_i}, t_{y_i}, θ_i, s_{x_i}, s_{y_i} and the coefficients c_i, with $i = 0, \ldots, N-1$ where N is the number of atoms considered for the reconstruction. The first term of the functional E represents the square error between the original image and the reconstructed one. The second term encourages a sparse representation of the data, giving a high penalty to large coefficients. In this case we set $S(x) = \log(1+x^2)$. The third part of the expression encourages, for each atom, the scale s_{x_i} to be smaller then s_{y_i}. Here we have chosen to set $P(x, y) = \arctan(k(x - y))$, where k determines the slope of the $\arctan$ function. This term has been inserted to reduce the introduction of *pathological* atoms that do not have the desired characteristics of band-pass, edge-detector functions. The parameters λ_1 and λ_2 are constant terms that determine the importance of the second and third terms respectively.

The images used for the learning are those of the dataset of ten 512 $\times$ 512 pixels filtered images of Olshausen and Field [4]. Experiments have been run on 16 $\times$ 16 patches, randomly sampled from the dataset. Only patches with a variance at least twice as large as that of the original set of images have been taken into account for the computation. Every image patch $I(x, y)$ was reconstructed using $N = 30$ atoms, each having 6 free parameters c_i, t_{x_i}, t_{y_i}, θ_i, s_{x_i}, s_{y_i}. For each image, thus, the function E was minimized in a space of dimension 6 $\times N$ = 180. The optimization was performed on each patch individually using a Sequential Quadratic Programming (SQP) method [12].

The parameter λ_1 was imposed to be equal to 0.14 σ_I, where σ_I is the standard deviation of the considered image patch, λ_2 was set to the same value of λ_1 and the parameter k was fixed to 5. Different combinations of the parameters have been tested with no significant changes in the results.

4. GENERATION OF THE TREE

The resulting atoms have been grouped into clusters using the algorithm presented in [11]. This method creates clusters in the initial dictionary and it organizes them in a hierarchical tree structure. Each node $N_{i,j}$ at level i and position j in the tree has M children and is characterized by the group of atoms $G_{i,j}$ 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:

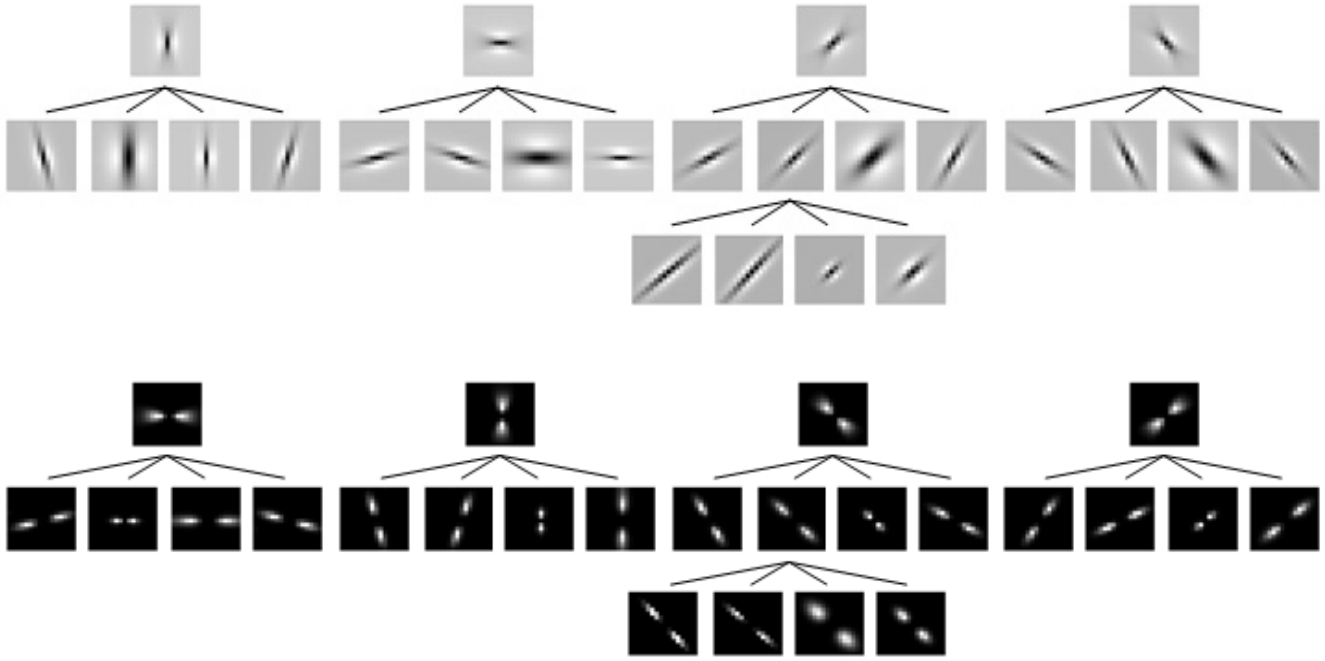

Fig. 1. The first two layers of the tree and an example of a third layer sub-cluster. Basis functions (up) and their corresponding power spectra (down) are shown.

$$c_{i,j} = \frac{\sum_{k \in G_{i,j}} g_{\gamma_k}}{\sqrt{\| \sum_{k \in G_{i,j}} g_{\gamma_k} \|}}, \tag{6}$$

where g_{γ_k} is the learnt anisotropic atom. The elements of the original learnt dictionary lie at the leaves of the tree, and each node represents a subspace of the dictionary, which is as orthogonal as possible to its siblings.

Defining the distance between two atoms as

$$d(g_{\gamma_l}, g_{\gamma_m}) = |\langle g_{\gamma_l}, g_{\gamma_m} \rangle| \,, \tag{7}$$

one can define the mean distance between $c_{i,j}$ and the atoms that it represents as

$$D_{i,j} = \frac{1}{n_{i,j}} \sum_{k \in G_{i,j}} d(g_{\gamma_k}, c_{i,j}) \,, \tag{8}$$

with $n_{i,j}$ being the cardinality of $G_{i,j}$. For a fixed set $G_{i,j}$, the quality of the clustering is defined as:

$$Q_{G_{i,j}} = \frac{1}{M} \sum_{\omega=0}^{M-1} D_{i+1,jM+\omega} \,. \tag{9}$$

The tree is built using a *k-means* algorithm that attempts to maximize for each group of atoms the quantity $Q_{G_{i,j}}$. The clustering process stops when $Q_{G_{i,j}}$ increases from one step of the *k-means* algorithm to the following one by a quantity that is smaller than a given ϵ.

5. RESULTS

The minimization of the functional E has been computed on 10000 images, thus obtaining 300000 atoms. The learnt atoms have a mean scale ratio s_y/s_x of 2.6246 with a standard deviation of 1.4508. The learnt functions exhibit an approximately uniform distribution of the rotations. At small scales, there is a slight preference for horizontal and vertical orientations, but this could be due to the fact that images are sampled using a square grid when digitalized: small atoms seem to be more influenced by the sampling structure.

The obtained atoms have been grouped using the algorithm described in Section 4, setting the number of children for each node to $M = 4$. The upper part of the tree resulting from the clustering of the atoms learnt from 16×16 image patches is depicted in Fig. 1.

The centroids are linear combinations of the atoms learnt and are thus functions well localized in space and frequency. The waveforms that represent the first level of the tree are edge-detector functions oriented along the four main directions of the image plane. Descending into the tree, the children of each node specialize in catching different image features at various scales and orientations.

We take advantage of the hierarchical representation of the learnt dictionary, using a Tree-Based Pursuit algorithm to generate sparse representations of images. The method, proposed in [11], finds at each step the best path through

(a) Original image (b) PSNR = 24.93 dB

(c) PSNR = 28.63 dB (d) PSNR = 30.82 dB

Fig. 2. *Lena* 128 $\times$ 128. Original *Lena* image (a) and its reconstructions using respectively (b) 100, (c) 300 and (d) 500 atoms.

the tree down to the leaves level, picking the best atom from the learnt dictionary. 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.

The complexity of this modified pursuit method is much lower than that of a full search MP method. Moreover, the learnt dictionary is completely general and can be used to reconstruct images of different types and sizes, and with variable quality. Fig. 2 shows the 128 $\times$ 128 *Lena* test image reconstructed using 100, 300 and 500 atoms.

6. CONCLUSION

In this paper we addressed the problem of efficiently representing images using sparse superposition of functions selected in a redundant dictionary. Meaningful atoms were designed through learning by minimizing a cost functional enforcing sparsity and good approximation power. A universal set of basis functions were then obtained, displaying various spatial and frequency localization behaviors. Imposing a hierarchical structure on the learnt set was achieved using a clustering approach. Finally, a fast tree-structured greedy algorithm was designed to benefit from the organization of the dictionary. Applications of this technique to image coding are foreseen, where encoding atom identities could also be performed in a tree-structured manner.

Acknowledgements

The authors would like to thank Dr Michel Bierlaire, Philippe Jost and Oscar Divorra for fruitful discussions.

7. 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] 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.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," in *IEEE Transactions on Signal Processing*, 2001, vol. 41, pp. 3445–3462.

[4] 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, `http://redwood.ucdavis.edu/bruno/sparsenet.html`.

[5] B. A. Olshausen, P. Sallee, and M. S. Lewiki, "Learning sparse image codes using a wavelet pyramid architecture," in *Advances in Neural Information Processing Systems*, 2001, vol. 13, pp. 887–893.

[6] A. J. Bell and T. J. Sejnowski, "The "independent" components of natural scenes are edge filters," in *Vision Research*, 1997, vol. 37, pp. 3327–3338.

[7] H. J. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," in *Proc. Royal Soc. Lond. B*, 1998, vol. 265, pp. 359–366.

[8] D. J. Field, "What is the goal of sensory coding?," in *Neural Computation*, 1994, vol. 6, pp. 559–601.

[9] D. L. Donoho, "Sparse component of images and optimal atomic decomposition," Tech. Rep., Statistics Departement, Stanford University, 2000.

[10] P. Vandergheynst and P. Frossard, "Efficient image representation by anisotropic refinement in matching pursuit," in *Proc. of IEEE, ICASSP*, Salt Lake City UT, 2001, vol. 3.

[11] L. Peotta, P. Jost, P. Vandergheynst, and P. Frossard, "Sparse approximation with sparse incoherent dictionaries," Tech. Rep. TR-ITS 2003.007, EPFL, 1015 Ecublens, 2003.

[12] 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, University of Maryland, College Park, 1997.