SCHOOL OF ENGINEERING - STI
SIGNAL PROCESSING INSTITUTE
Gianluca Monaci and Pierre Vandergheynst

CH-1015 LAUSANNE

Telephone: +4121 6936874
Fax: +4121 6937600
e-mail: gianluca.monaci@epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

LEARNING STRUCTURED DICTIONARIES FOR IMAGE REPRESENTATION

Gianluca Monaci and Pierre Vandergheynst

Swiss Federal Institute of Technology Lausanne (EPFL)

Signal Processing Institute Technical Report

TR-ITS-2004.010

April 14th, 2004

Learning structured dictionaries for image representation

Gianluca Monaci, Pierre Vandergheynst
Signal Processing Institute (ITS)
Swiss Federal Institute of Technology (EPFL)
Lausanne 1015, 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. Experiments have been carried out using images of different sizes in order to check the influence of this parameter on the learnt bases. 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.

Keywords

Sparse representation, image representation, redundant expansion, dictionary learning.

I. 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 examples, we can cite the wavelet transform and the *curvelet* transform. The success of the first one in image coding largely depends on its hierarchical tree structure [3], that however limits its flexibility, as witnessed by the limited freedom in choosing the properties of a wavelet basis. The frames of curvelet introduced by Candès and Donoho [4] exhibit an essentially optimal behavior in representing objects with C^2 singularities. They impose, however, strong constraints in the structure of the transform, that can pose problems when dealing with real-world images, that contain features much more complex then uniform regions and smooth edges.

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 [5][6] or independent components (ICA) [7][8] of natural images. The sparse approach, however, seems to be more plausible than the ICA one from a biological [5][9] and mathematical [10] points 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 [11] 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 [5]. Moreover, basis functions for images of different size have been learnt, in order to study the influence of this factor on the resulting atoms.

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 [12]. 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 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.

This report is structured as follows: in Section II, the image model adopted is introduced, as well as the basic dictionary of Anisotropic Refinement atoms. The learning process is described in Section III, while in Section IV is presented the method used for the construction of the tree. In Section V the experimental results are presented and discussed, and in Section VI conclusions are drawn, and possible future applications are depicted.

II. 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 $g_{\gamma_i}(x, y)$:

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

where c_i are the coefficients and N is the number of basis functions used to form the reconstruction of the image $I(x, y)$.

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. The dictionary $\mathcal{D}= \{U_\gamma g, \gamma \in \Gamma\}$ for a given set of indexes Γ, is thus built by applying the transformations U_γ to the function g. Basically, the required transformations are translations by t_x and t_y, rotations by θ and scaling by s_x and s_y. It is easy to demonstrate that the dictionary built in such a way is overcomplete [11].

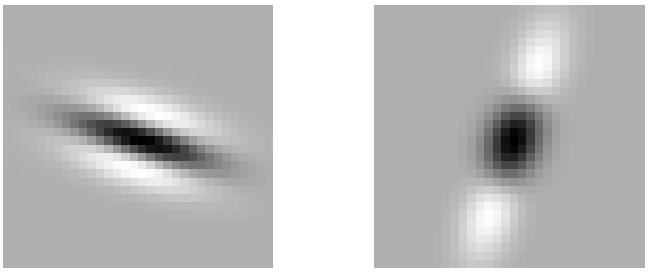
The generating function g should be able to efficiently represent edges on the 2-D plane and thus should behave like a smooth scaling function in one direction and like a wavelet in the orthogonal one. In this case, the function $g(x, y)$ is a Gaussian along one axis and the second derivative of a Gaussian along the other one:

$$g(x, y) = (2 - 4x^2) \exp(-(x^2 + y^2))\,. \tag{2}$$

This set of atoms has been chosen because, as described in [11], is able to represent very well contours and edges, and also because of the optimal spatial and frequency localization of the Gaussian kernel.

An Anisotropic Refinement (AR) 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:

$$g_\gamma(u, v) = \frac{C}{\sqrt{s_x s_y}}(2 - 4u^2) \exp(-(u^2 + v^2))\,, \tag{3}$$



(a) (b)

Fig. 1. Anisotropic Refinement atoms of Eq.(3). Positive values are depicted in white and negative values in black. (a) Atom with x scale s_x smaller than the y 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.

where C is a normalization constant and

$$u = \frac{cos\theta(x - t_x) + sin\theta(y - t_y)}{s_x}\,, \tag{4}$$

and

$$v = \frac{-sin\theta(x - t_x) + cos\theta(y - t_y)}{s_y}\,. \tag{5}$$

An example of Anisotropic Refinement atom is shown in Fig. 1(a).

It is interesting to remark that the atoms g_γ have the same characteristics of the waveforms employed in [4] to define the curvelet functions.

III. Learning the Basis

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 $I(x, y)$, but that also take into account the sparseness of the representation. The learning can thus be accomplished by minimizing an objective function composed of three terms:

$$E = \sum_{x,y}\left[I(x,y) - \sum_{i=0}^{N-1} c_i g_{\gamma_i}(x,y)\right]^2 + \lambda_1 \sum_{i=0}^{N-1} S(c_i) + \lambda_2 \sum_{i=0}^{N-1} P(s_{x_i}, s_{y_i})\,, \tag{6}$$

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 $i = 0, \ldots, N-1$ and N being 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, and indicates how accurate is the reconstruction. 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 the parameter k determines the slope of the $\arctan$ function. This term has been added to reduce the introduction of *pathological* atoms that do not have the desired characteristics of band-pass, edge-detector functions (see Fig. 1). When s_{x_i} is bigger then s_{y_i}, the argument of $\arctan$ is positive and the value of the function is high, while in the case in which s_{x_i} is smaller then s_{y_i}, the value of $\arctan$ is small. The parameters λ_1 and λ_2 are constant terms that attribute the importance of the second and the third term respectively, relative to the first.

The images used for the learning are those of the dataset of ten 512×512 pixels filtered images of Olshausen and Field [5]. Four examples of the test images are shown in Fig. 2. Experiments have been run on 16×16 and 32×32 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 atoms, thus, for each image the function E was minimized in a space of

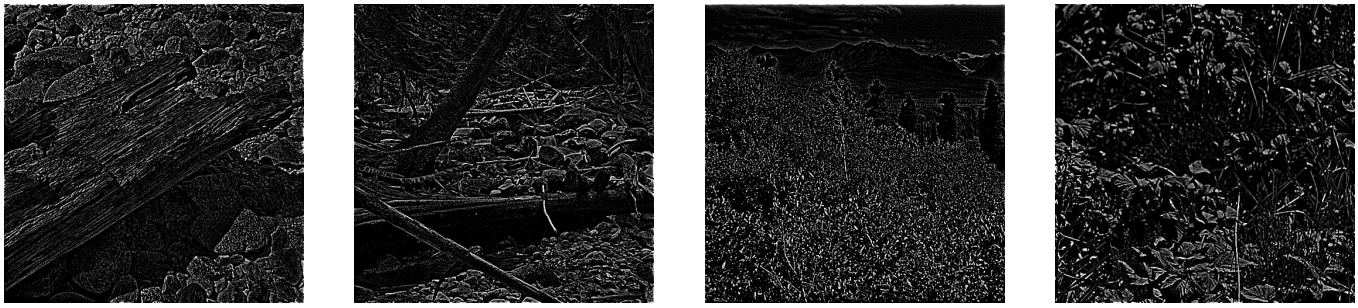


Fig. 2. Examples of four test images from the dataset of Olshausen and Field.

dimension $6 \times N$. In the first series of experiments with 16×16 pixels patches, images were reconstructed using 30 atoms ($N = 30$). The second set of experiments on 32×32 patches was run using 60 atoms ($N = 60$) for the reconstruction.

The optimization has been done on each patch individually using a Sequential Quadratic Programming (SQP) method [13]. The algorithm, beginning from a given starting point, generates at each iteration a direction d^0 of descent for the objective function, solving a standard quadratic program. The minimization stops when the norm of the vector d^0 is smaller than a threshold ε. In our experiments ε was set equal to 10^{-3}, since this value represented a good trade-off between learning speed and reconstruction accuracy.

The parameter λ_1 was imposed to be equal to $0.14\sigma_I$, where σ_I was the variance 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.

IV. Generation of the Tree

The resulting atoms have been grouped into clusters using the algorithm presented in [12]. 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:

$$c_{i,j} = \frac{\sum_{k \in G_{i,j}} g_{\gamma_k}}{\sqrt{\| \sum_{k \in G_{i,j}} g_{\gamma_k} \|}}\,, \tag{7}$$

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

$$d(g_{\gamma_l}, g_{\gamma_m}) = |\langle g_{\gamma_l}, g_{\gamma_m} \rangle|\,, \tag{8}$$

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

$$D_{i,j} = 1/n_{i,j} \sum_{k \in G_{i,j}} d(g_{\gamma_k}, c_{i,j})\,, \tag{9}$$

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:

$$Q_{G_{i,j}} = \frac{1}{M} \sum_{\omega=0}^{M-1} D_{i+1,jM+\omega}\,. \tag{10}$$

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. Here the value of ϵ is set equal to 10^{-6}.

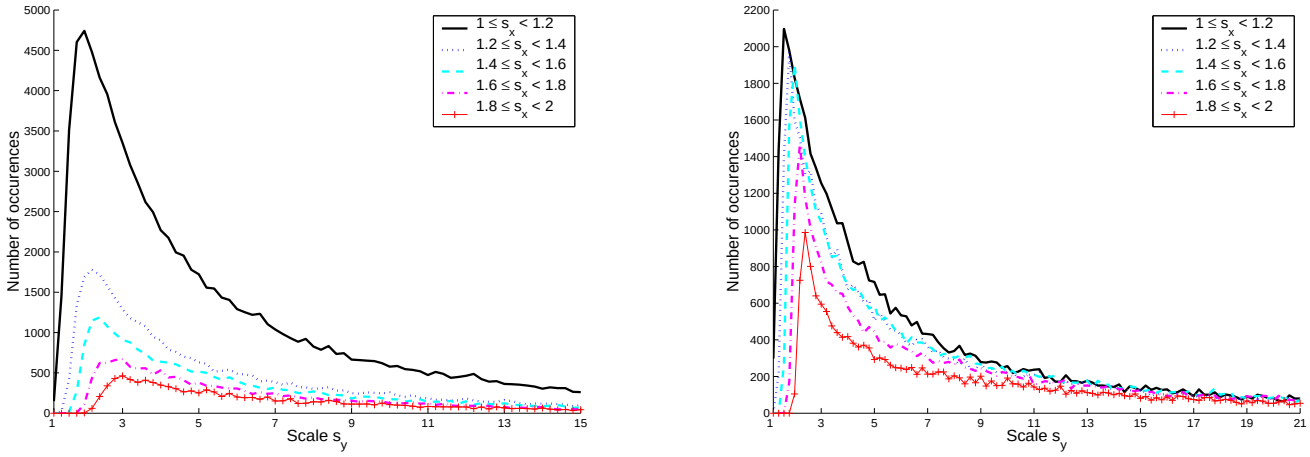


Fig. 3. Histograms of the scale s_y conditioned to different values of s_x. (a) Results for 16×16 image patches and (b) for 32×32 image patches.

V. Results

A. Learnt Dictionary

In the experiment with 16×16 pixels images and 30 atoms, the minimization of the functional E has been computed on 10000 images, thus obtaining 300000 atoms. We have however considered only the atoms lying on the image area and whose scales satisfied the inequality $s_x \leq s_y$, and we have drawn a joint histogram of the two scales. A large number of atoms has been found to have scale s_y around 2 and scale s_x of about 1. In fact the histogram bin with the higher number of occurrences is the one corresponding to $1 \leq s_x < 1.2$ and $1.8 \leq s_y < 2$. The learnt atoms have a mean value of the scale s_x equal to 1.1348 while the mean value of s_y is 4.1265; the mean anisotropy scale ratio s_y/s_x is 2.6246 with a standard deviation of 1.4508. The behavior of the histogram of s_y conditioned to different values of s_x is depicted in Fig. 3(a).

The results obtained with bigger images (32×32 pixels) behave similarly. In that case, 5000 images, reconstructed using 60 atoms each, have been analyzed. Thus, in this second experiment we have again 300000 atoms learned. The mean value of the scale s_x is 1.4555 and the mean value of s_y is 5.0276; the mean of the ratio s_y/s_x is now 2.5043 with a standard deviation of 1.9206. The behavior of the histogram of s_y conditioned to different values of s_x is depicted in Fig. 3(b).

In Fig. 4 are plotted the histograms of the anisotropy scale ratio s_y/s_x conditioned to various values of the scale s_x, for (a) 16×16, and (b) 32×32 image patches. It is evident the preference for small atoms with an anisotropy ratio between 2 and 4. Increasing the size (here the scale s_x) of the atoms, the scale ratio tends to be uniform.

Concerning the orientations of the atoms, in Fig. 5 the histograms of the parameter θ conditioned to different values of the scales s_x and s_y, for images 16×16 pixels, are shown. The behavior of the histograms for 32×32 images is similar. The peaks in the histograms corresponding to rotations θ equal to 0, $\pi/2$ and π are typical of the small scales, while atoms with larger scales exhibit a much uniform distribution of the rotations. 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.

As observed before, the AR atoms employed here have the same shape of the curvelet functions introduced in [4]. However, the learnt dictionary has substantially different properties with respect to the curvlet frame. The width and length of a cuvelet, that essentially correspond to s_x and s_y of the AR atom, obey the *Anisotropy Scaling Relation*:

$$\text{width} \approx \text{length}^2 .$$

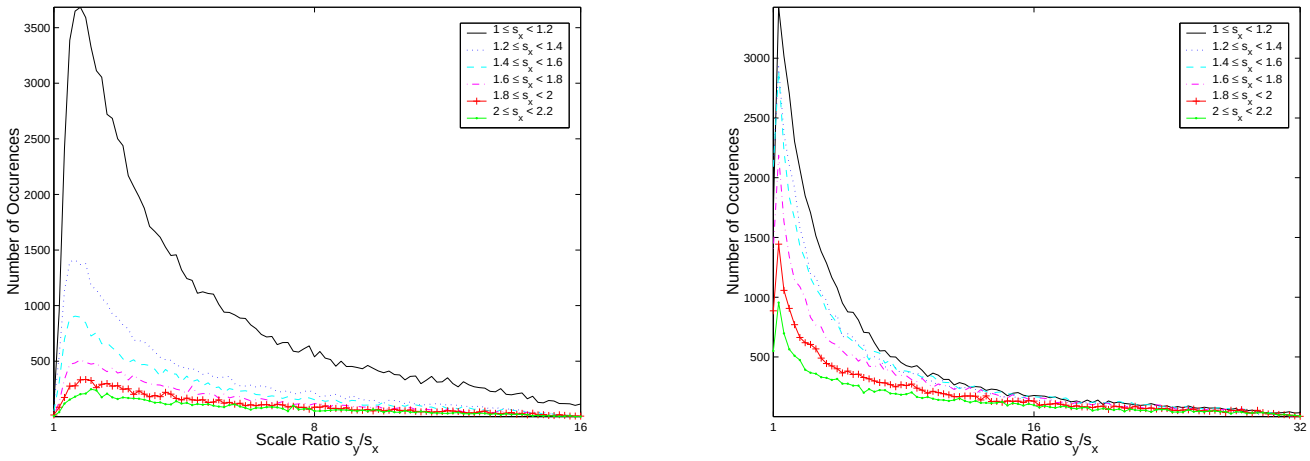


Fig. 4. Histograms of the ratio s_y/s_x conditioned to different values of s_x. (a) Results for 16×16 image patches and (b) for 32×32 patches.

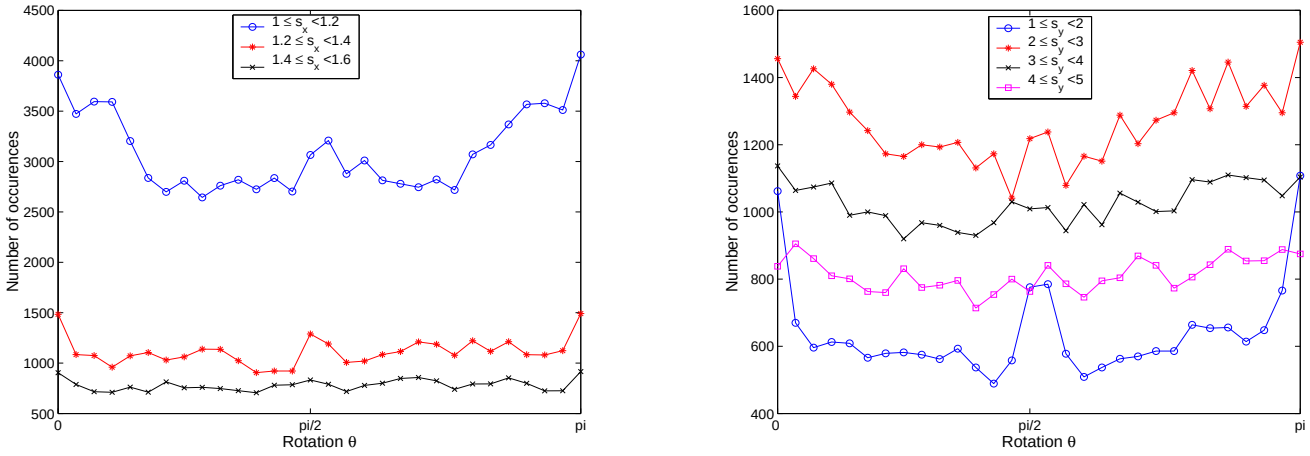


Fig. 5. Histograms of the rotation θ (a) for different values of the scale s_x, and (b) for different values of s_y. The diagrams refer to 16×16 pixels images.

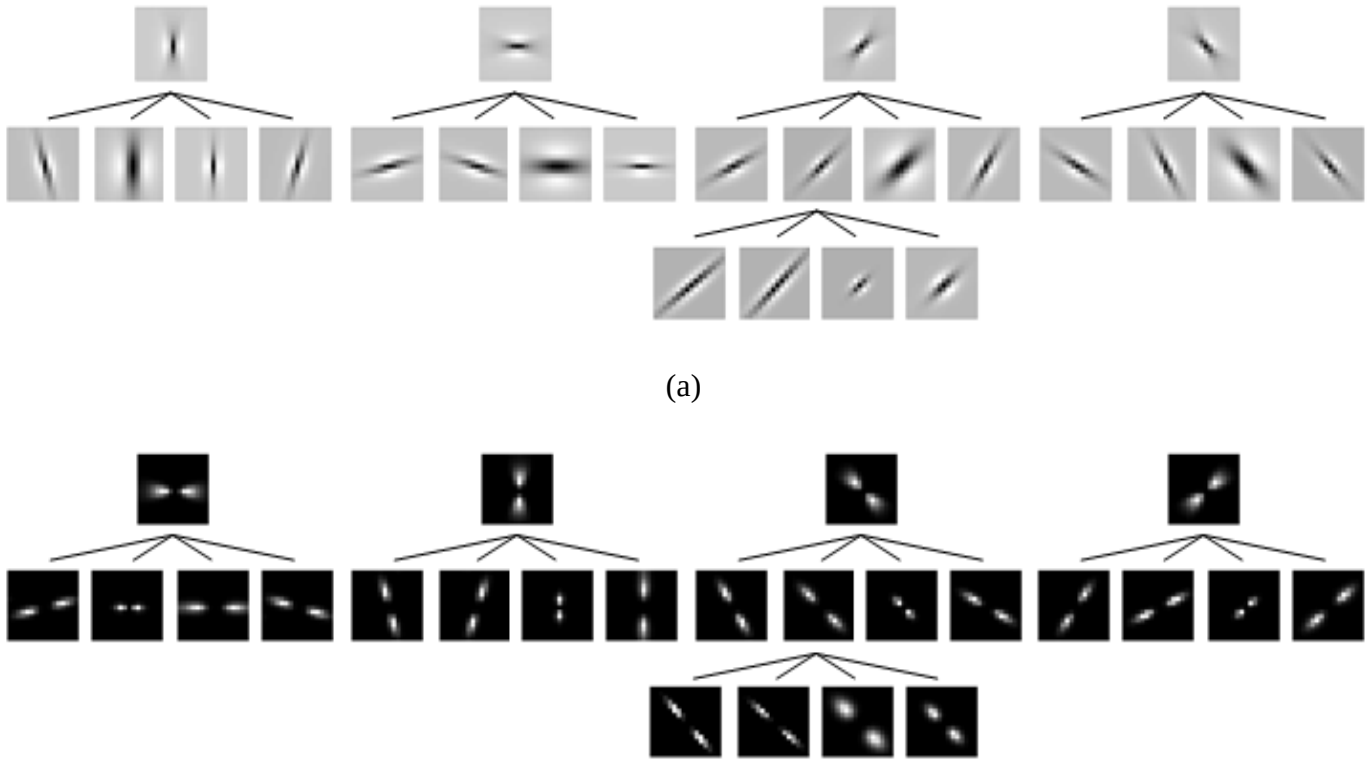
Moreover, each curvelet frame element has the *Directional Sensitivity Property*, that is:

$$\text{number of orientations} = 1/\sqrt{\text{width}}\,.$$

These two properties clearly do not hold in the dictionaries we have learnt. In our case, in fact, the scales s_x and s_y are in general not related by a parabolic law (see Fig. 3). The number of orientations found, moreover, do not decrease with the increase of the width (i.e. of s_x) of the atoms, on the contrary they seem to be more uniformly distributed (see Fig. 5). These observations seem to confirm the idea that real-world images are difficult to model with a tractable mathematical representation and motivate us in pursuing the promising direction of the learning approach to image representations.

B. Dictionary Organization

In order to organize the learnt dictionary in a hierarchical, tractable structure, the obtained atoms have been grouped using the algorithm described in Section IV, 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



(a)

(b)

Fig. 6. The first two layers of the tree and an example of a third layer sub-cluster. The basis functions (a) and the corresponding power spectra (b) are shown. The depicted functions are the centroids of the clusters obtained grouping the atoms learnt in the experiment with 16×16 pixels image patches. Each node has $M = 4$ children.

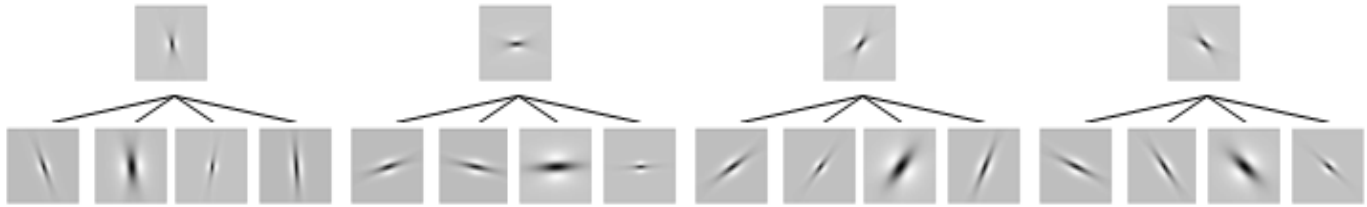
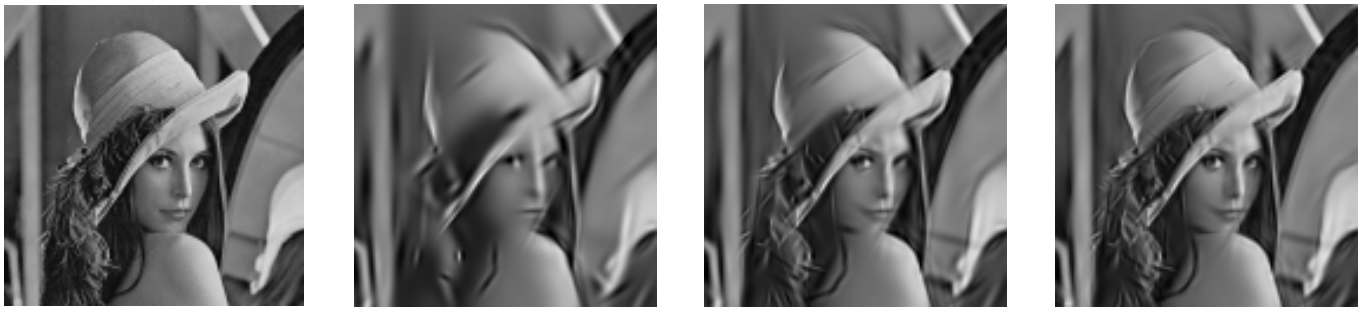


Fig. 7. The first two layers of the tree. The depicted functions are the centroids of the clusters obtained grouping the atoms learnt in the experiment with 32×32 pixels image patches. Each node has $M = 4$ children.

is depicted in Fig. 6. A substantially identical tree-structure has been derived clustering only 100000 learnt atoms.

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.

In Fig. 7 it is shown the tree obtained considering the set of atoms learnt from 32×32 image patches. In this case, we have clustered 100000 atoms. The resulting structure is essentially similar to the one obtained with 16×16 atoms.



(a) Original image (b) PSNR = 24.93 dB (c) PSNR = 28.63 dB (d) PSNR = 30.82 dB

Fig. 8. *Lena* 128×128. Original *Lena* image (a) and its reconstructions using respectively (b) 100, (c) 300 and (d) 500 atoms. Results for the 16×16 dictionary.

(a) Original image (b) PSNR = 24.24 dB (c) PSNR = 26.37 dB (d) PSNR = 27.96 dB

Fig. 9. *Barbara* 128×128. Original *Barbara* image (a) and its reconstructions using respectively (b) 100, (c) 300 and (d) 500 atoms. Results for the 16×16 dictionary.

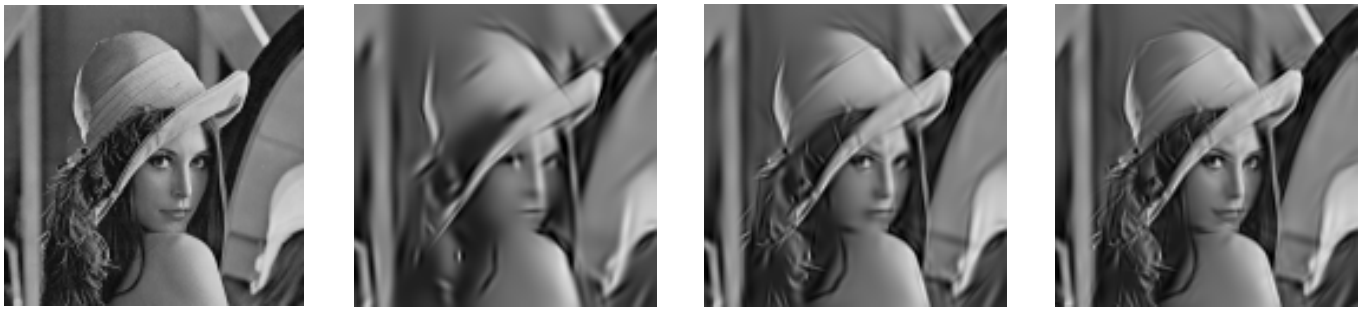
C. Image Representation

We take advantage of the hierarchical representation of the learnt dictionary, using a tree-based MP algorithm to generate sparse representations of images. The method, proposed in [12], finds at each step the best path through 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 MP method is much lower than that of a full search 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. 8 shows the 128×128 *Lena* test image reconstructed using 100, 300 and 500 atoms. Fig. 9 shows the 128×128 *Barbara* test image approximated with 100, 300 and 500 atoms. The original image has been down-sampled by a factor of 4 obtaining thus a 32×32 sub-image that has been interpolated with 2-D Gaussian functions in order to obtain a low-pass image. The difference between the original image and this low-pass version has been reconstructed with the tree-based MP algorithm and the result of the reconstruction has been added again to the low-pass part.

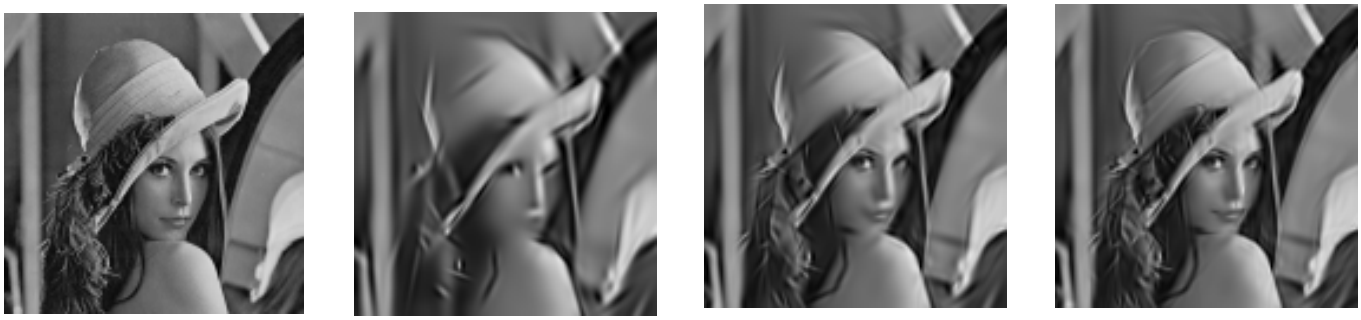
In order to speed up the construction of the tree and the search procedure, the algorithm described above has been applied to a reduced tree composed of 100000 elements. The results of the reconstruction of the 128×128 *Lena* image are shown in Fig. 10. As can be immediately observed, the results are qualitatively and quantitatively similar to those obtained using the entire dictionary.

We have also decomposed the image *Lena* using the tree constructed with 100000 atoms learnt on 32×32



(a) Original image (b) PSNR = 25.06 dB (c) PSNR = 28.59 dB (d) PSNR = 30.72 dB

Fig. 10. *Lena* 128×128. Original *Lena* image (a) and its reconstructions using respectively (b) 100, (c) 300 and (d) 500 atoms. The results have been obtained applying the tree-based MP algorithm to a sub-dictionary of 100000 16×16 atoms.



(a) Original image (b) PSNR = 25.42 dB (c) PSNR = 28.99 dB (d) PSNR = 31.25 dB

Fig. 11. *Lena* 128×128. Original *Lena* image (a) and its reconstructions using respectively (b) 100, (c) 300 and (d) 500 atoms. The results have been obtained applying the tree-based MP algorithm to a dictionary of 100000 atoms learnt from 32×32 image patches.

image patches (see Fig. 7). The results using 100, 300 and 500 atoms are shown in Fig. 11. In this case the performances of the reconstruction algorithm are slightly improved with respect to the results obtained with the 16×16 atoms dictionary. This is probably due to the wider range of values taken in this case by the scale parameters: bigger atoms are present in the dictionary, that can take into account larger image features.

VI. Conclusions

In this work 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. Universal sets of basis functions were than obtained, displaying various spatial and frequency localization behaviors. Dictionaries of various dimensions and composed of atoms with different size were built, demonstrating the robustness and flexibility of the proposed approach. The characteristics of the learnt sets of functions were described and discussed in the context of sparse image representation. Imposing a hierarchical structure on the learnt sets was achieved using a clustering approach. Finally, a fast tree-structured greedy algorithm was designed to benefit from the organization of the dictionary, and it was tested on the learnt dictionaries. Applications of this technique to image coding are foreseen, where encoding atom identities could also be performed in a tree-structured manner.

Acknowledgements

This work was supported by the Swiss NFS through the IM.2 National Center of Competence for Research. The authors would also like to thank Dr Michel Bierlaire, Philippe Jost and Oscar Divorra Escoda for fruitful discussions.

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] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representation of objects with C^2 singularities," Tech. Rep., Department of Statistics, Standford University, 2002.

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

[6] 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.

[7] 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.

[8] 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.

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

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

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

[12] 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.

[13] 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.