HAL
open science

Learning Adapted Dictionaries for Geometry and Texture Separation

Gabriel Peyré, Jalal M. Fadili, Jean-Luc Starck

To cite this version:

Gabriel Peyré, Jalal M. Fadili, Jean-Luc Starck. Learning Adapted Dictionaries for Geometry and Texture Separation. SPIE Wavelets XII, Aug 2007, San Diego, CA, United States. pp.67011T. hal-00365601

HAL Id: hal-00365601

https://hal.science/hal-00365601

Submitted on 3 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Adapted Dictionaries for Geometry and Texture Separation

Gabriel Peyré[a], Jalal Fadili[b] and Jean-Luc Starck[c]

[a]CEREMADE,
Université Paris Dauphine,
Pl. De Lattre De Tassigny
75775 Paris Cedex 16
(FRANCE)

[b]GREYC,
ENSICAEN,
6, Bd du Maréchal Juin
14050 Caen Cedex
(FRANCE)

[c]Service d'Astrophysique,
CEA/Saclay,
Orme des Merisiers, Bat 709,
91191 Gif-sur-Yvette Cedex
(FRANCE)

ABSTRACT

This article proposes a new method for image separation into a linear combination of morphological components. This method is applied to decompose an image into meaningful cartoon and textural layers and is used to solve more general inverse problems such as image inpainting. For each of these components, a dictionary is learned from a set of exemplar images. Each layer is characterized by a sparse expansion in the corresponding dictionary. The separation inverse problem is formalized within a variational framework as the optimization of an energy functional. The morphological component analysis algorithm allows to solve iteratively this optimization problem under sparsity-promoting penalties. Using adapted dictionaries learned from data allows to circumvent some difficulties faced by fixed dictionaries. Numerical results demonstrate that this adaptivity is indeed crucial to capture complex texture patterns.

Keywords: Sparsity, image separation, inpainting, learning dictionaries

1. REGULARIZED INVERSE PROBLEM WITH SEVERAL LAYERS

Many problems in image processing can be cast as inverting a linear system $f = \mathcal{K}u + \varepsilon$ where $u \in \mathbb{R}^N$ is the data to recover, $f \in \mathbb{R}^m$ is the observed image and ε is a gaussian white noise of known variance. The linear mapping $\mathcal{K} : \mathbb{R}^N \mapsto \mathbb{R}^m$ is typically ill-behaved since it models an acquisition process that encounters loss of information. This entails ill-posedness of the inverse problem.

In order to regularize such an inversion problem, one has to add some prior knowledge on the typical structures of the original image u. This prior information accounts for the smoothness of the solution and can range from uniform smoothness assumption to more complex knowledge of the geometrical structures of u.

This paper supposes that the original image can be written as

$$u = \sum_{s=1}^{S} u_s$$

In words, u is a linear combination of S layers (the so-called morphological components), where each u_s accounts for a different kind of features of the original data u. Each of these layers has its own prior assumption and multiplying the number of priors should help to recover intricate image structures such as smooth areas, edges and textures of natural images.

This regularized inversion with several layers can be formalized within a variational framework as the optimization of $\{u_s\}_{s=1}^S$ in order to solve

$$\min_{u_1,\ldots,u_S} \frac{1}{2}\left\| f - \mathcal{K}\sum_{s=1}^{S} u_s \right\|_{\ell^2}^2 + \mu \sum_{s=1}^{S} E_s(u_s), \tag{1}$$

where each energy $E_s : \mathbb{R}^N \mapsto \mathbb{R}^+$ favors images with some specific kind of structures.

Contact: gabriel.peyre@ceremade.dauphine.fr

Typical inverse problems that can be handled within this framework are for instance:

- *Denoising*: in this case, $S = 1$, $N = m$, $\mathcal{K} = \mathrm{Id}_N$ and one wants to remove the additional noise ε.
- *Image separation*: similarly to the denoising, $\mathcal{K} = \mathrm{Id}_N$, but one considers $S > 1$ layers which allows to split the image into meaningful components.
- *Deconvolution*: in this case, $\mathcal{K}$ is a blurring kernel and f lacks the high frequency content of u.
- *Super-resolution*: in this case, $\mathcal{K}$ is a blurring kernel of width σ followed by a sub-sampling by a factor σ. The dimension is typically $m = N/\sigma$.
- *Inpainting*: in this case, $\mathcal{K}$ is an operator that removes some pixels of the image.

This paper focuses on image separation and inpainting and thus assumes that $m = N$.

2. VARIATIONAL AND SPARSITY-BASED ENERGIES

This section recalls several classical typical energies E_s that come either from modeling of images using functional spaces or using sparse coding in various transforms. These two kinds of energies are closely related, and this paper uses the framework of sparse expansion to derive new kinds of energies that use sparse expansions of image patches in adapted local dictionaries.

2.1 Variational Energies and Functional Spaces

Denoising. The simplest problem corresponds to image denoising where one looks for only one component $f \approx u_1$ capturing structures such as the contours of the image. The residual $f - u_1$ is only capturing the noise and the parameter μ should be fixed in order to fit this noise level. The Rudin-Osher-Fatemi[1] model corresponds to the use of the following energy

$$E_1(u_1) \overset{\text{def.}}{=} \|u_1\|_{\mathrm{TV}} \overset{\text{def.}}{=} \int |\nabla_x u_1| \mathrm{d}x,$$

where the total variation norm $\|\cdot\|_{\mathrm{TV}}$ imposes that the contours of u_1 have a small overall length and removes from u_1 the oscillations due to noise and texture.

Structure/texture decomposition. Yves Meyer extends this model[2] in order to include an additional component $f = u_1 + u_2$ where u_2 captures the oscillating structure of textures. It corresponds to the energy

$$E_1(u_1) \overset{\text{def.}}{=} \|u_1\|_{\mathrm{TV}} \qquad \text{and} \qquad E_2(u_2) \overset{\text{def.}}{=} \|u_2\|_{\mathrm{G}} = \min_{u_2 = \mathrm{div}(g)} \|g\|_{\ell^\infty}, \tag{2}$$

where, for $g = (g_1, g_2)$, one defines $\|g\|_{\ell^\infty} \overset{\text{def.}}{=} \sup_x \sqrt{g_1(x)^2 + g_2(x)^2}$. The G-norm $\|\cdot\|_{\mathrm{G}}$ is close to being a dual norm to the total variation and thus favors highly oscillating functions. This model is implemented in practice with iterative optimization schemes such as the one proposed by Aujol et al.[3]

2.2 Sparsity-based Energy for a Global Dictionary

An alternative to these energies based on functional spaces consists in exploiting a linear decomposition of each component $u_s = D_s x_s$ in some dictionary $D_s \in \mathbb{R}^{N \times m_s}$, $m_s \geqslant N$. The sparsity of such a decomposition can be measured using the ℓ^1 norm

$$\|x_s\|_{\ell^1} \overset{\text{def.}}{=} \sum_j |x_s[j]|.$$

For redundant dictionaries $m_s > N$, such a set of coefficients $x_s \in \mathbb{R}^{m_s}$ is not unique in general, but one can define an energy related to the decomposition of x_s in D_s as

$$E_s(u_s) \overset{\text{def.}}{=} \min_{x \in \mathbb{R}^{m_s}} \frac{1}{2}\|u_s - D_s x\|_{\ell^2}^2 + \lambda \|x\|_{\ell^1}, \tag{3}$$

where the parameter λ allows an approximate reconstruction $D_s x_s \approx u_s$.

Such a sparsity-based energy requires the use of a dictionary D_s that can efficiently capture the relevant structures of the layer u_s. Many dictionaries have been proposed to capture several features of natural images.

- Wavelets[4] are used intensively in image compression and allow to capture in an optimal way images with bounded variations. The ℓ^1 norm of wavelets coefficients $\|{D_\mathrm{W}}^\mathrm{T} v\|_{\ell^1}$, where D_W is a tight frame of translation invariant wavelets, is closely related to the total variation norm $\|v\|_\mathrm{TV}$, see.[5]
- For cartoon-like images, that are regular outside a set of regular edge curves, wavelets are not anymore an optimal representation. The tight frame of curvelets, introduced by Candès and Donoho,[6] is able to efficiently capture the contours of images.
- Locally oscillating and stationary textures can be handled with a redundant tight frame D_C of local cosines.[4] The norm $\|{D_\mathrm{C}}^\mathrm{T} v\|_{\ell^1}$ of an image in this frame is related to the G-norm of Meyer $\|v\|_\mathrm{G}$ defined in equation (2).

This article focuses on the dictionaries D_W and D_C that are redundant tight frames. They can capture respectively the bounded variation and the oscillating parts of the image. Such dictionaries have been used by Starck et al.,[7,8] and give rise to a decomposition similar to the one of Meyer (2). This article shows how to add to these fixed dictionaries additional dictionaries learned from a set of exemplars in order to capture complex textural patterns.

One can note that other dictionaries can enhance over the result of local cosines to capture warped oscillating patterns. For instance the waveatoms of Demanet and Ying[9] and the brushlets of Meyer and Coifman[10] have been designed for this goal.

2.3 Sparsity-based Energy for a Local Dictionary

This article not only uses the dictionaries $D_s \in \mathbb{R}^{N \times m_s}$ defined on the whole set of N pixels, but also local dictionaries to capture fine scale structures of the textures. Such a dictionary $D_s \in \mathbb{R}^{n \times m_s}$ is used to represent patches $R_k(u_s) \in \mathbb{R}^n$ of $n \stackrel{\text{def.}}{=} \tau \times \tau$ pixels extracted from a component u_s,

$$\forall\, 0 \leqslant k_{1,2} < \sqrt{N} \quad \text{and} \quad -\tau/2 \leqslant i_{1,2} < \tau/2, \quad R_k(u_s)[i] = u_s(k_1 + i_1, k_2 + i_2),$$

where $k = k_1\sqrt{N} + k_2$ with k_i the index of the pixel (k_1, k_2).

Similarly to the energy (3) associated to a global dictionary, we can define an energy $E_s(u_s)$ associated to a local dictionary D_s. This energy allows to control the sparsity of the decomposition of all the patches $R_k(u_s)$ in D_s. This energy $E_s(u_s)$ is defined as

$$E_s(u_s) = \min_{\{x_s^k\}_k \in \mathbb{R}^{m_s \times N}} \frac{1}{n} \sum_{k=0}^{N-1} \frac{1}{2} \|R_k(u_s) - D_s x_s^k\|_{\ell^2}^2 + \lambda \|x_s^k\|_{\ell^1}. \tag{4}$$

In this energy, each x_s^k corresponds to the coefficient of the decomposition of the patch $R_k(u_s)$ in the dictionary D_s. The weight $1/n$ in the energy (4) compensates for the redundancy factor introduced by the overlap between the patches $R_k(u_s)$. This normalization allows to re-scale the local energy (4) with the global one (3).

The use of local dictionaries to perform signal processing has been proposed by Elad and co-workers in order to use the output of their K-SVD learning scheme.[11] They have tackled the problem of image denoising[12] and inpainting.[13] This article proposes a common framework to include both local and global dictionaries and uses this framework to solve more general problems involving morphological component separation.

2.4 Images vs coefficients, Analysis vs Synthesis

The variational formulation (1) proposed in this paper directly seeks for the components $\{u_s\}_{s=1}^S$. Alternative formulations of inverse problems in redundant dictionaries or union of bases search instead for the coefficients x_s of each component in the dictionary D_s

$$\min_{x_1,\ldots,x_S} \frac{1}{2} \Big\| f - \mathcal{K} \sum_{s=1}^S D_s x_s \Big\|_{\ell^2}^2 + \lambda \sum_{s=1}^S \|x_s\|_{\ell^1}, \tag{5}$$

see for instance.[7,14] In this formulation, the optimized components are given by $u_s \stackrel{\text{def.}}{=} D_s x_s$ where the $\{x_s\}_{s=1}^S$ are the solution of (5).

In the case of orthogonal dictionaries D_s, the two formulations (1) and (5) are closely related. However, since we consider highly redundant dictionaries, which correspond to non bijective transforms, the two results are expected to differ significantly. Indeed, in the formulation (5) (coefficients domains) the set of feasible solutions $\{u_s\}_{s=1}^S$ is confined to the column space of the merged dictionary, whereas in the formulation (5) (image domains) $\{u_s\}_{s=1}^S$ are allowed to be arbitrary vectors in $\mathbb{R}^N$. This difference disappears when the merged dictionary is made of orthonormal bases.

One may then legitimately ask the question whether it is better to optimize with respect to coefficients or images. Our belief is that there is not general answer to this delicate question. In the setting of local dictionaries as exposed in section 2.3, using an energy such as (5) is unfeasible since the dimension of $x_s = \{x_s^k\}$ is $m_s N \gg N$. Furthermore we argue that the objects of interest are the image layers. Hence, it appears natural to optimize our energy functional in terms of images as we proposed in equation (1).

3. MORPHOLOGICAL COMPONENT ANALYSIS

The morphological component analysis (MCA) algorithm[7,8] allows to solve iteratively the variational separation problem (1) for sparsity-based energies E_s as defined in equation (3). The main feature of this algorithm is that the regularization parameter λ decreases through the iterations until it reaches its final value $\lambda_{\min}$ that corresponds to the targeted level of regularization. The parameter $\lambda_{\min}$ is typically set to $3-4\times$ standard deviation of the noise for noisy data, and set to 0 for a noiseless separation. This decay of λ has a flavor of deterministic annealing and is also closely related to homotopy continuation[15] and path following methods.[16–18] It allows to speed-up the convergence and leads to a simple update rule at each iteration using a non-linear thresholding.

For the decomposition of an image into its geometrical and textured parts, the original approach[7,8] uses fixed dictionaries of wavelets D_{W} (and also curvelets in order to enhance the extraction of the contours) and local cosines D_{C}. This article extends the MCA algorithm in order to deal with energies E_s associated to local dictionaries D_s as defined in equation (4).

3.1 MCA with Local and Global Dictionaries

The new MCA-type algorithm minimizes iteratively the energy (1) by adding to the decomposition variables $\{u_s\}_{s=1}^S$ auxiliary variables $\{x_s\}_{s=1}^S$ corresponding to the coefficients of the decomposition of each u_s. For a global dictionary $D_s \in \mathbb{R}^{N\times m_s}$ these coefficients are packed in a vector $x_s \in \mathbb{R}^{m_s}$. But for a local dictionary $D_s \in \mathbb{R}^{n\times m_s}$ these coefficients are a collection of vectors $\{x_s^k\}_{k=0}^{N-1} \in \mathbb{R}^{m_s\times N}$.

The energy minimized by the MCA algorithm is

$$\mathcal{E}(\{u_s\}_s, \{x_s\}_s) \stackrel{\text{def.}}{=} \frac{1}{2}\|f - \mathcal{K}\sum_s u_s\|_{\ell^2}^2 + \mu \sum_s \mathcal{E}_s(u_s, x_s) \tag{6}$$

where each energy $\mathcal{E}_s$ is defined differently according to whether D_s is a global or a local dictionary

$$\mathcal{E}_s(u_s, x_s) \stackrel{\text{def.}}{=} \begin{cases} \frac{1}{2}\|u_s - D_s x_s\|_{\ell^2}^2 + \lambda\|x_s\|_{\ell^1}, & \text{(global dictionary)} \\ \frac{1}{n}\sum_k \frac{1}{2}\|R_k(u_s) - D_s x_s^k\|_{\ell^2}^2 + \lambda\|x_s^k\|_{\ell^1}. & \text{(local dictionary)} \end{cases}$$

The MCA algorithm operates by minimizing successively $\mathcal{E}$ on each component u_s and then on each set of coefficient x_s while maintaining all remaining variables fixed, for $s = 1, \ldots, S$. For each index s, the MCA scheme performs a step of computation of the coefficients x_s followed by an update of the component u_s. These two steps slightly differ depending on the nature (global or local) of the dictionary D_s.

The minimization of a convex functional such as $\mathcal{E}$ by a block-coordinate relaxation iterative algorithm is a classical optimization strategy. The difficulty of this approach for the optimization of $\mathcal{E}$ stems from the non-differentiability of the $\|x_s\|_{\ell^1}$ norms. When the D_s are global dictionaries corresponding to frames, the authors in[14] proved the convergence of the MCA algorithm using a fundamental result due to Tseng.[19] We will later use a similar argument to prove the convergence of our separation algorithm.

Global dictionaries considered in this paper are tight frames and thus lead to a fast convergence of the iterations. Local dictionaries are however in general badly conditioned because of the overlap of the patches $R_k(u_s)$ which slow down the practical optimization process and requires more iterations for the algorithm to converge.

3.2 Computation Algorithm

For each index s, the MCA algorithm proceeds according to two different steps for the minimization of $\mathcal{E}$.

Step 1 – Update of the coefficients x_s. This computation requires the minimization of $\mathcal{E}_s(u_s, x_s)$ with respect to x_s, while keeping u_s fixed. The corresponding minimization differs depending on whether the dictionary is local or global

$$\begin{cases} x_s \overset{\text{def.}}{=} \underset{x}{\operatorname{argmin}} \, \frac{1}{2}\|u_s - D_s x\|_{\ell^2}^2 + \lambda \|x\|_{\ell^1}, & \text{(global dictionary)} \\ x_s^k \overset{\text{def.}}{=} \underset{x}{\operatorname{argmin}} \, \frac{1}{2n}\|R_k(u_s) - D_s x\|_{\ell^2}^2 + \lambda \|x\|_{\ell^1}. & \text{(local dictionary)} \end{cases} \tag{7}$$

For a local dictionary, the coefficients $x_s = \{x_s^k\}_k$ are thus computed by considering independently each patch $R_k(u_s)$.

The problem of equations (7), known as sparse coding, requires solving a convex non-smooth optimization functional. There are several options to solve for the coefficients x_s (global dictionary) or x_s^k (local dictionary):

- This sparse coding problem corresponds to the basis pursuit denoising (BPDN),[20] which can be solved exactly with (perturbed) linear programming. Linear programming is however too slow for large-scale applications as in image processing, and one has to resort to an iterative thresholding approach to minimize (7).
- This minimization can be solved approximately with soft thresholding (written here for a global dictionary)

$$x_s = S_\lambda({D_s}^{\mathrm{T}} u_s) \quad \text{where} \quad S_\lambda(x_s) = \{s_\lambda(x_s[i])\}_i \quad \text{with} \quad s_\lambda(x) \overset{\text{def.}}{=} \begin{cases} x - \operatorname{sign}(x)\lambda & \text{if} \quad |x| > \lambda, \\ 0 & \text{if} \quad |x| \leqslant \lambda. \end{cases} \tag{8}$$

  This thresholding approach is in fact the exact solution in the case of an orthogonal dictionary D_s. For a tight frame $D_s {D_s}^{\mathrm{T}} = \mathrm{Id}_N$, this non-linear thresholding is no more the exact solution of (7). It is in fact the first iteration of the following proximal scheme to solve (7),

$$x_s^{(t+1)} = S_{\lambda\gamma_t}\left(x_s^{(t)} + \gamma_t {D_s}^{\mathrm{T}}(u_s - D_s x_s^{(t)})\right) \quad \text{with} \quad 0 < \inf_t \, \gamma_t \leqslant \sup_t \, \gamma_t < 2/B_s \tag{9}$$

  where B_s is the upper-bound of the frame D_s. Such proximal iterations have been proved to converge to a (non-necessarily unique) global minimizer of (7), see for instance.[21,22] Since the global dictionaries D_{W} and D_{C} we consider are tight frames, we use the thresholding (8) to compute the coefficients $x_s \in \mathbb{R}^{m_s}$. A similar approach has been used in the original MCA algorithm[7,8] and has been proved to converge to the global solution of the problem as long as tight frames are involved.[23]
- The local dictionaries we consider might deviate substantially from being tight frames. In particular, they can become ill-conditioned. An option to better approximate the solution of (8) is to perform $q > 1$ thresholdings with a smaller threshold λ/q, which corresponds to several proximal steps (8). In practice, we use a greedy algorithm such as orthogonal matching pursuit (OMP)[4] to recover the sparse set of coefficients x_s^k for the local dictionaries.

Step 2 – Update of the components u_s. Each component u_s is updated by minimizing

$$\|r_s - \mathcal{K} u_s\|_{\ell^2}^2 + \mu \mathcal{E}_s(u_s, x_s) \qquad \text{where} \qquad r_s = f - \mathcal{K} \sum_{\ell \neq s} u_\ell, \tag{10}$$

where the coefficients x_s and the other components $u_{s' \neq s}$ are held fixed. This leads to the following update rule

$$u_s = \left(\mathcal{K}^{\mathrm{T}}\mathcal{K} + \mu \mathrm{Id}_N\right)^{-1} \left(\mathcal{K}^{\mathrm{T}} r_s + \mu \hat{u}_s\right) \tag{11}$$

where the reconstructed $\hat{u}_s$ is computed differently depending if the dictionary is global or not

$$\hat{u}_s \stackrel{\text{def.}}{=} \begin{cases} D_s x_s, & \text{(global dictionary)} \\ \sum_k {R_k}^{\mathrm{T}} (D_s x_s^k). & \text{(local dictionary)} \end{cases}$$

To derive the expression for a local dictionary, we used the fact that

$$\frac{1}{n} \sum_k {R_k}^{\mathrm{T}} R_k = \mathrm{Id}_N,$$

where a special care should be taken at the boundaries of the image.

Global+local MCA algorithm. The MCA algorithm, given in listing 1, performs iteratively these two steps for each index $s = 1 \dots S$. The regularization parameter λ decays at each iteration. The number of iterations q allows to define the step $\delta = (\lambda_{\max} - \lambda_{\min})/q$, which is fixed to $q = 100$ in the numerical simulations of this paper.

Listing 1 Morphological components analysis with both global and local dictionaries.

Initialization: $\forall\, s, u_s = 0$, $\lambda = \lambda_{\max}$.
Loop: While $\lambda > \lambda_{\min}$,
Loop: For each $s = 1, \dots, S$,
- *(Compute the coefficients)* The coefficients x_s are computed by solving the sparse coding (7) with the thresholding (8) (global dictionary) or by OMP (local dictionary).
- *(Update the components)* The component u_s is computed using equation (11).

End For
(Update the threshold) $\lambda \leftarrow \lambda - \delta$.
End While

3.3 Convergence properties

Note first that the objective functional $\mathcal{E}(\{u_s\}_s, \{x_s\}_s)$ defined in (6) is a coercive proper function. Thus, minimizing (6) has at least one solution. The convergence proof of the cyclic algorithm in listing 1 follows, after identifying our problem with the one considered by the author in.[19] Then using the same notation as in[19] (for a global dictionary, the same holds for a local dictionary), we can write:

$$\begin{aligned} \mathcal{E}(\{u_s\}_s, \{x_s\}_s) &= \mu f_0(\{u_s\}_s, \{x_s\}_s) + f_1(\{u_s\}_s) + \mu \sum_s f_s^x(x_s) \\ \text{where } f_0(\{u_s\}_s, \{x_s\}_s) &= \sum_s \frac{1}{2} \|u_s - D_s x_s\|_{\ell^2}^2, \\ f_1(\{u_s\}_s) = \frac{1}{2} \|f - \mathcal{K} \sum_s u_s\|_{\ell^2}^2 \quad &, \quad f_s^x(x_s) = \lambda \|x_s\|_{\ell^1},\ \forall s = 1, \dots, S \end{aligned}$$

It is not difficult to see that f_0 has a non-empty open domain and is continuously differentiable on its domain. Thus f_0 satisfies Assumption A.1 in.[19] Moreover, $\mathcal{E}(\{u_s\}_s, \{x_s\}_s)$ is continuous on its effective domain, with bounded level sets. As f_1 and f_s^x are convex, and f_0 is strictly convex, $\mathcal{E}(\{u_s\}_s, \{x_s\}_s)$ is strictly convex (hence pseudoconvex). Thus, Lemma 3.1 and Theorem 4.1(a) of[19] imply that the sequence of iterates provided by the above cyclic MCA algorithm is defined, bounded and every accumulation point is a stationary point of $\mathcal{E}(\{u_s\}_s, \{x_s\}_s)$. This accumulation point is the unique global minimizer by strict convexity.

4. DICTIONARY LEARNING

4.1 Adaptive Representations

In order to efficiently represent complex natural images, one needs representation tools that can adapt to the regular part, the contours and the textured areas of such images. Fixed dictionaries such as wavelets or local cosines allow to capture only simple patterns, but more adaptive representations have been developed to handle more complex image models.

A class of adaptive methods consists in using a family of orthogonal bases and look for the best basis in this family using combinatorial optimization algorithms. The wedglets[24] and the bandlets[25,26] allow to better represent contours than a traditional wavelet dictionary D_{W}. For oscillating textures, a proper tree of the wavelet packets bases[4] with an appropriate tiling of the frequency domain can do good a job at sparsity of oscillatory patterns; see the waveatoms.[9] Cosine packets allow a dyadic partition of the spatial domain[4] according to a quad-tree structure. Grouplets bases[27] are able to approximate efficiently oscillating and turbulent textures.

Contrarily to these approaches, which are able to handle only a particular kind of images or textures, other approaches can adapt to the content of images through a learning process. By optimizing a sparsity criterion, such algorithms allow to optimize a local dictionary for a set of exemplar patches. This kind of methods naturally fit into the MCA framework developed in section 3 by considering local dictionaries applied to all the patches of the image to process. The remaining part of this section explains the procedure of computing such adapted local dictionaries.

4.2 Dictionary Optimization and Learning

The energy (4) defined on the patches of $n = \tau \times \tau$ pixels requires the use of a dictionary D_s adapted to the local structures of the textures. In order to obtain such a dictionary, Olshausen and Field[28] proposed to learn the matrix D_s to optimally represent a set $Y = \{y^k\}_{k=1}^p \in \mathbb{R}^{n\times p}$ of $p \gg n$ patches. They showed that when these patches are extracted from a set of natural images, the obtained dictionary is similar to the dictionary D_{W} of redundant wavelets.

Other formalisms have been proposed for the adaptation of a dictionary to a set of exemplars, see for instance,[11,29,30] where each method defines its own energy together with an optimization procedure.

Learning with fixed layer exemplar. Ultimately, the goal of the learning stage is to optimize a set of local dictionaries $\{D_s\}_s$ in order to minimize the morphological separation energy $\mathcal{E}$ of equation (6). However, in many applications, one already has an exemplar $\tilde{u}_s$ of some typical layer one would like to extract. Consequently, we first consider the problem of optimizing a single dictionary D_s in order to sparsify a set of patches $\{y^k = R_k(\tilde{u}_s)\}_k$ extracted from the exemplar $\tilde{u}_s$.

For the learning stage to be efficient, we suppose that the exemplar $\tilde{u}_s$ corresponds to a single homogeneous texture. The dictionary D_s trained from the patches $\{y^k\}_k$ is thus optimized to represent typical patterns of this texture and the component u_s extracted from f in the decomposition (1) is likely to contain these patterns.

Fixing the layer u_s in energy (6) thus leads to the following optimization problem

$$\min_{D_s\in\mathbb{R}^{n\times m_s},\{x^k\}_k\in\mathbb{R}^{m_s\times p}} \sum_{k=1}^{p} \frac{1}{2}\|y^k - D_s x^k\|_{\ell^2}^2 + \lambda\|x^k\|_{\ell^1}.$$

which is close to the K-SVD method optimization problem proposed by Aharon et al.[11] In order for this problem to be well-posed, one has to impose the additional normalization $\|d_j\|_{\ell^2} = 1$ on the columns of the dictionary $D_s \in \mathbb{R}^{n\times m_s}$.

This optimization problem is non-linear and non-convex and the algorithm K-SVD allows to find a local minimum by iteratively cycling between the computation of the coefficients $\{x^k\}_k$ and the dictionary D_s. Listing 2 details these two steps. Figure 1 (d) shows a learned dictionary. One can see that the learned atoms d_j do a good job at capturing the patterns of the exemplar texture depicted on figure 1 (a).

Learning both the dictionaries and the decomposition. Optimizing the energy (6) over both the decomposition variables $\{u_s, x_s\}_s$ and the dictionaries $\{D_s\}_s$ is a non-convex difficult problem. In particular, it is sensitive to the initialization of the dictionaries. This minimization can be carried over using a block-coordinate descent method that iterates on the following steps:

- *(MCA decomposition)* The dictionaries $\{D_s\}_s$ are fixed, and the MCA algorithm, listing 1, is used to compute the layers $\{u_s\}_s$.
- *(Dictionary update)* The layers $\{u_s\}_s$ are fixed, and the K-SVD algorithm, listing 2, is used to compute the optimized dictionaries $\{D_s\}_s$.

Similarly to subsection 3.3, this alternating block-relaxation minimization scheme has at least one solution by coercivity. But this time, when also optimizing (6) with respect to the dictionaries $\{D_s\}_s$, $\mathcal{E}$ is no longer convex, neither pseudoconvex in every pair of coordinate blocks. Thus, we are only able to claim that the algorithm will converge to a stationary point using Lemma 3.1 and Theorem 4.1(b) of.[19]

Listing 2 K-SVD algorithm.

Initialization: The dictionary is initialized with the cosine transform $D_s = D_{\mathrm{C}}$ and $x^k \leftarrow {D_s}^{\mathrm{T}} y^k$.
Loop: While D_s has not converged,
- D_s *fixed*: the coefficients $\{x^k\}_k$ are updated for each sample y^k by minimizing

$$x^k \leftarrow \operatorname*{argmin}_x \frac{\mu}{2} \|y^k - D_s x\|_{\ell^2}^2 + \|x\|_{\ell^1}.$$

  This optimization can be solved approximately using a pursuit algorithm (e.g. OMP), see.[4]
- $\{x^k\}_k$ *fixed*: for each atom d_j, define $I_j = \{k \setminus x^k[j] \neq 0\}$ the set of indices pointing to patches that use the atom d_j. This atom d_j is updated by minimizing

$$d_j \leftarrow \operatorname*{argmin}_g \min_x \sum_{k \in I_j} \|\tilde{y}^k - x[k]g\|_{\ell^2}, \qquad \text{where} \qquad \tilde{y}^k = y^k - \sum_{\ell \neq j} x^k[\ell] d_\ell.$$

  This minimization is equivalent to a rank-1 approximation of the matrix containing the signals $\tilde{y}^k$ for $k \in I_j$, which can in turn be solved with a singular value decomposition (SVD).

---

5. APPLICATIONS

5.1 Image decomposition

The image decomposition problem corresponds to $\mathcal{K} = \mathrm{Id}_N$. One thus looks for an approximate decomposition $f \approx \sum_s u_s$. The component update (11) is written as the convex combination

$$u_s = (1+\mu)^{-1}(r_s + \mu \hat{u}_s).$$

Image decomposition with exemplars. This application uses a first dictionary $D_1 = D_{\mathrm{W}}$ of redundant wavelets in order to capture the geometric part of the input image f. The second dictionary D_2 is either:

- a redundant cosine transform $D_2 = D_{\mathrm{C}}$. The image decomposition result with the D_{W} and D_{C} global dictionaries, as proposed by,[7,8] is illustrated on figure 1 (e);
- or learned by using the method described in section 4 from a known texture $\tilde{u}_2$ that is close to the texture we would like to remove from f.

The input image is generated as $f = u_1 + u_2$ where u_2 is a texture visually similar to $\tilde{u}_2$. The MCA algorithm described in section 3 is applied to the image f shown on figure 1 (c). Figure 1 (e) and (f) compare the result of two decompositions using $D_{\mathrm{W}} - D_{\mathrm{C}}$ (e), and $D_{\mathrm{W}}-$learned (f). One can see from (e) that the overlaying texture u_2 is not sparse enough in the cosine frame and this decomposition does not extract any textural patterns. These patterns are however perfectly recovered from the adapted decomposition as can be clearly seen on figure 1 (f).

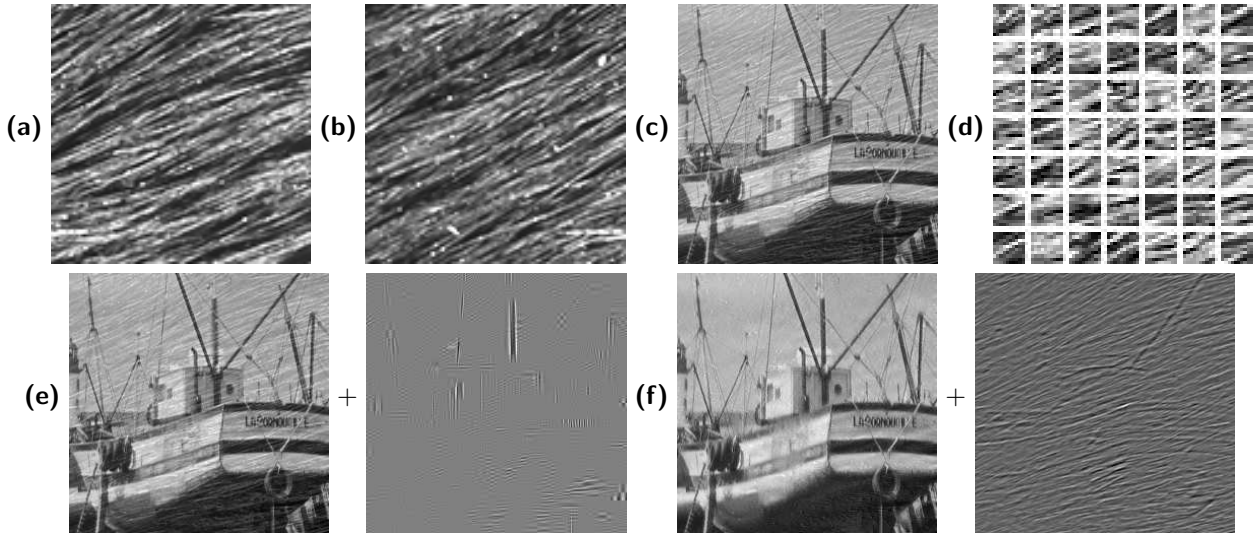

Figure 1. *(a) Known texture $\tilde{u}_2$, (b) texture to extract u_2, (c) input image to decompose $f = u_1 + u_2$, (d) learned dictionary D_2, (e) decomposition using translation invariant wavelets $D_1 = D_W$ and a cosine dictionary $D_2 = D_C$, (f) decomposition using translation invariant wavelets $D_1 = D_W$ and a learned dictionary D_2.*

Image decomposition with both dictionary and decomposition optimization. In this second decomposition, we compute a separation $f = \sum_{s=1}^{4} u_s$ using the wavelet dictionary $D_1 = D_{\mathrm{W}}$ and three learned dictionaries $\{D_s\}_{s=2}^{4}$. Each of these dictionaries D_2, D_3, D_4 is optimized automatically along with the computation of the decomposition as described in the end of subsection 4.2. The important issue is thus the initialization of these dictionaries. These initial dictionaries are obtained by training them with the K-SVD on patches extracted around pixels interactively selected by the user, see figure 2. This figure show the ability of the learned dictionaries to adapt to the orientation of the textures on the shirt.

Figure 3 exemplifies another decomposition with a single textured component. The dictionary was initialized from patches extracted around the center of the image.

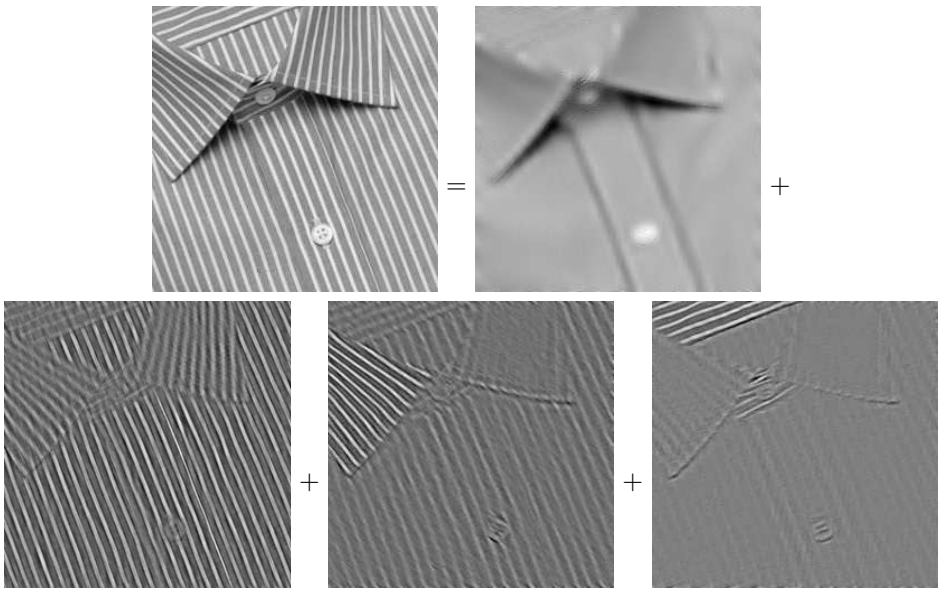

Figure 2. *Decomposition using a wavelet dictionary and three learned dictionaries.*

Figure 3. *Decomposition using a learned dictionary where its initialization is obtained from patches extracted around the center of the image.*

5.2 Inpainting

Inpainting is to restore missing image information based upon the still available (observed) cues from destroyed or deliberately masked subregions of the image f. Inpainting of non-textured images has been traditionally approached by diffusion equations that progressively fills the missing pixels. The original work of Masnou and Morel makes use of the continuation law of the level sets.[31] Following their work, several authors proposed high order PDEs, see for instance[32–34] and anisotropic diffusion[35] for non-texture inpainting.

The MCA framework has been also used to solve the inpainting problem[14,36] for images containing simple textural content such as locally parallel oscillations. Our extension of the MCA framework to local dictionaries allows to inpaint more complex textures as long as one is able to correctly train the dictionary on the existing parts of the image.

The inpainting problem corresponds to a diagonal operator

$$\mathcal{K} = \mathrm{diag}(\eta_i) \qquad \text{where} \qquad \eta_i = \begin{cases} 1 \text{ if } i \notin \Omega \\ 0 \text{ if } i \in \Omega \end{cases}$$

where $\Omega \subset \{0, \ldots, N-1\}$ denotes the set of missing pixels. The component update (11) becomes

$$u_s[i] = \begin{cases} (1+\mu)^{-1}(r_s[i] + \mu \hat{u}_s[i]) & \text{if } i \notin \Omega, \\ \hat{u}_s[i] & \text{if } i \in \Omega. \end{cases}$$

As shown in,[14] an important feature of the MCA approach to solving the inpainting is that the noise level can be estimated at each iteration using the known pixels inside Ω. In this article however, we use the MCA algorithm as given in listing 1 with a linear decay of the threshold λ and minimum threshold $\lambda_{\min} = 0$ since it works well in the inpainting problem we consider.

Figure 4 shows an example of inpainting to restore an image where 65% of the pixels are missing. This figure compares the inpainting done with a wavelet and a local DCT transform with the inpainting done with an additional layer with a learned dictionary. This dictionary is trained from an exemplar texture as already done for the experiment of figure 1. One can see that the inpainting result with a learned dictionary is able to recover the missing fine scale details of the texture, which is not the case with the use of only a local DCT to represent the texture. Figure 5 shows another example of inpainting with large holes that requires a learned dictionary in order to respect the patterns of the overlaid texture.

REFERENCES

1. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Phys. D **60**(1-4), pp. 259–268, 1992.
2. Y. Meyer, Oscillating Patterns in Image Processing and Nonlinear Evolution Equations, American Mathematical Society, Boston, MA, USA, 2001.
3. J. F. Aujol, G. Aubert, L. Blanc-Feraud, and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component," Journal of Math. Im. and Vision **22**, pp. 71–88, Jan. 2005.

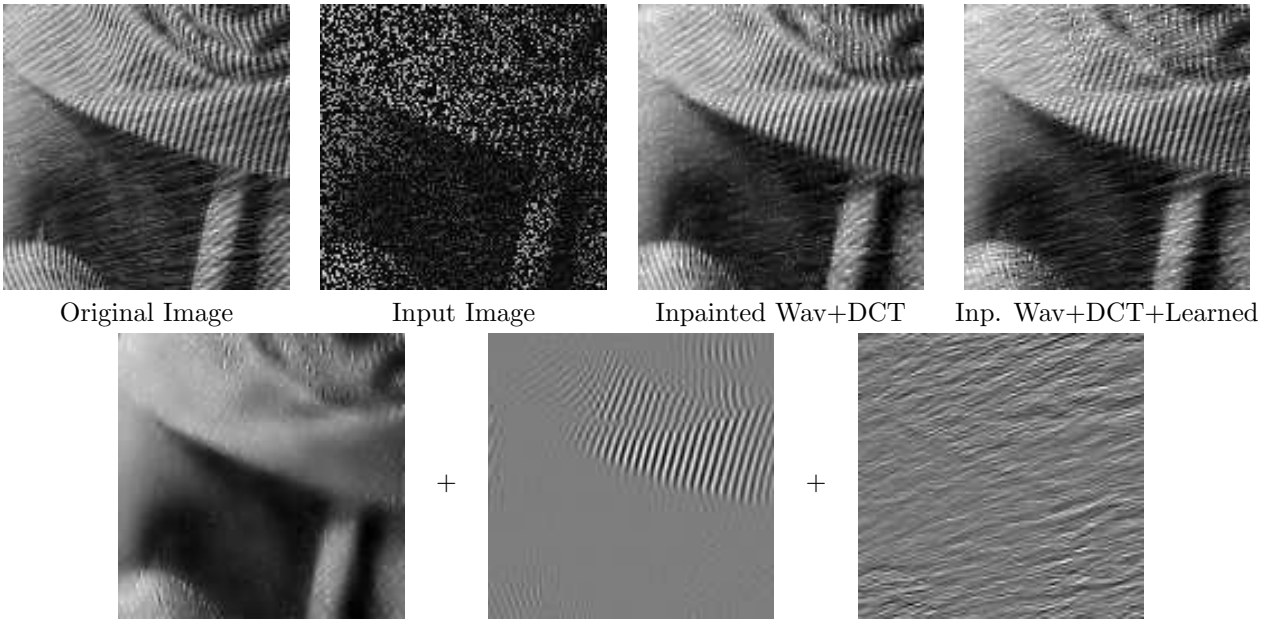

Figure 4. *Top: image inpainting of 65% randomly removed pixels, comparison of the two methods. Bottom: the three layers for the inpainting with a learned dictionary.*

Figure 5. *Image inpainting with a user defined mask.*

4. S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, San Diego, 1998.
5. G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk, "On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides," SIAM Journal on Numerical Analysis **42**(2), pp. 686–713, 2004.
6. E. Candès and D. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," Comm. Pure Appl. Math. **57**(2), pp. 219–266, 2004.
7. J.-L. Starck, M. Elad, and D. Donoho, "Image decomposition via the combination of sparse representation and a variational approach," IEEE Transaction on Image Processing **14**(10), pp. 1570–1582, 2005.
8. J.-L. Starck, M. Elad, and D. Donoho, "Redundant multiscale transforms and their application for morphological component analysis," Advances in Imaging and Electron Physics **132**, 2004.
9. L. Demanet and L. Ying, "Wave atoms and sparsity of oscillatory patterns," Preprint , 2006.
10. F. G. Meyer and R. R. Coifman, "Brushlets: A tool for directional image analysis and image compression," Journal of Appl. and Comput. Harmonic Analysis **5**, pp. 147–187, 1997.

11. M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," IEEE Trans. On Signal Processing **54**(11), pp. 4311–4322, 2006.
12. M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. on Image Processing **15**(12), pp. 3736–3745, 2006.
13. J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," IMA Preprint Series 2139 , 2006.
14. M. Fadili and J.-L. Starck, "Em algorithm for sparse representation-based image inpainting," in IEEE International Conference on Image Processing, Vol. II, pp. 61–63, 2005.
15. N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," Neural Networks **11**(2), pp. 271–282, 1998.
16. B. Efron, T. Hastie, I. Johnstone, and T. Tibshirani, "Least angle regression," Annals of Statistics **32**(2), p. 407?499, 2004.
17. M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," IMA Journal of Numerical Analysis **20**, pp. 389–403, July 2000.
18. D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," Technical Report , 2006.
19. P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," Journal of Optimization Theory and Applications **109**(3), pp. 475–494, 2001.
20. S. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," SIAM Journal on Scientific Computing **20**(1), pp. 33–61, 1998.
21. I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," Comm. Pure Appl. Math **57**, pp. 1413–1541, 2004.
22. P. L. Combettes and V. R. Wajs, "Gsignal recovery by proximal forward-backward splitting," SIAM Journal on Multiscale Modeling and Simulation **4**(4), 2005.
23. M. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," The Computer Journal (to appear) , 2006.
24. D. Donoho, "Wedgelets: Nearly-minimax estimation of edges," Ann. Statist **27**, pp. 353–382, 1999.
25. E. Le Pennec and S. Mallat, "Bandelet Image Approximation and Compression," SIAM Multiscale Modeling and Simulation **4**(3), pp. 992–1039, 2005.
26. S. Mallat and G. Peyré, "Orthogonal bandlet bases for geometric images approximation," To appear in Com. Pure and Applied Mathematics , 2006.
27. S. Mallat, "Geometrical grouplets," Submitted , 2006.
28. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive-field properties by learning a sparse code for natural images," Nature **381**, pp. 607–609, June 1996.
29. M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," Neural Comput. **12**(2), pp. 337–365, 2000.
30. K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Dictionary learning algorithms for sparse representation," Neural Comput. **15**(2), pp. 349–396, 2003.
31. S. Masnou, "Disocclusion: a variational approach using level lines," IEEE Trans. Image Processing **11**, pp. 68–76, Feb. 2002.
32. V. Caselles, J. M. Morel, and C. Sbert, "An axiomatic approach to image interpolation," IEEE Trans. Image Processing **7**, pp. 376–386, Mar. 1998.
33. M. Bertalmìo, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in Siggraph 2000, pp. 417–424, 2000.
34. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," IEEE Trans. Image Processing **10**, pp. 1200–1211, Aug. 2001.
35. D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDEs: Acommon framework for different applications," IEEE Trans. Pattern Anal. Mach. Intell **27**(4), pp. 506–517, 2005.
36. M. Elad, J.-L. Starck, D. Donoho, and P. Querre, "Simultaneous cartoon and texture image inpainting using morphological component analysis (mca)," Journal on Applied and Computational Harmonic Analysis **19**, pp. 340–358, 2005.