A Predual Proximal Point Algorithm solving a Non Negative Basis Pursuit Denoising model

François Malgouyres, Tieyong Zeng

Primal/dual implementation of a Basis Pursuit model

F. Malgouyres*and T. Zeng*

Feb. 23, 2007

Résumé

This paper contains a new algorithm solving a variant of the Basis Pursuit model. The variant aims at simplifying the parameter tuning. The algorithm is easy to implement and prove it converges to the actual solution of the model. Some experiments on image approximation show that it outperforms by far the existing algorithms.

AMS Classification : 41A45, 65D15, 49N30, 49N15, 65K05
Keywords : Basis Pursuit, algorithm, sparse representation

1 Introduction

1.1 Recollection on Basis Pursuit

The use of the Basis Pursuit norm [4] in image/signal processing is now a fairly developed field of research. To mention few contributions it is commonly used for compression, source separation [17] and feature selection in classification [3]. Many theoretical result have also been established supporting this model. Most of them aim at understanding the equivalence between the common Basis Pursuit model (see below) and the search for the sparsest decomposition (see, among others, [8, 9]). Other authors show that the Basis Pursuit model is an efficient way to simplify a complex data distribution (see [13, 12]).

In its most recent form [9, 13], the Basis Pursuit functional is defined by a finite subset of $\mathbb{R}^N$ (called dictionary) $(\psi_i)_{i\in I}$ and takes the form

$$\begin{cases} E(v) = \inf_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \\ \text{and } \sum_{i\in I} \lambda_i \psi_i = v, \end{cases}$$

for all $v \in \mathbb{R}^N$.

The strength of this functional is that its level sets are scaled versions of the convex hull of $(\psi_i)_{i\in I}$ (see [7, 13]). It is therefore possible to build a functional E that favor the apparition of specific structures ; and we have a complete control on these structures. This functional can then be used in optimization problems designed for specific applications.

The common model named Basis Pursuit takes the form

$$\min_{(\lambda_\psi)_{\psi\in\mathcal{D}}\in\mathbb{R}^{\mathcal{D}}} \| \sum_{\psi\in\mathcal{D}} \lambda_\psi \psi - v\|^2 + \lambda \sum_{\psi\in\mathcal{D}} |\lambda_\psi|, \tag{1}$$

for a dictionary $\mathcal{D}$, $\lambda > 0$, a datum $v \in \mathbb{R}^N$ and the standard l^2 norm on $\mathbb{R}^N$, $\|.\|$. It can be rewritten under the form

$$\min_{w\in\mathbb{R}^N} \|w - v\|^2 + \lambda E(w), \tag{2}$$

when E is defined with the dictionary $\mathcal{D} \cup \{-\psi, \psi \in \mathcal{D}\}$.

*LAGA/L2TI, Université Paris 13, 99, avenue Jean-Batiste Clement, 93430 Villetaneuse, FRANCE malgouy@math.univ-paris13.fr, zeng@math.univ-paris13.fr

Notice that the use of a general E in (2) permits to assign a sign to each feature. This might be of some importance in some applicative contexts. (Think about a source separation problem involving text. The letters are always black on a white background.)

Another issue which we wanted to improve in (1) concerns the choice of the parameter λ. For practical applications, it is always preferable to solve the model under the form

$$\begin{cases} \min_{(\lambda_\psi)_{\psi\in\mathcal{D}}\in\mathbb{R}^{\mathcal{D}}} \sum_{\psi\in\mathcal{D}} |\lambda_\psi|, \\ \text{under the constraints } \| \sum_{\psi\in\mathcal{D}} \lambda_\psi \psi - v\|^2 \leq \tau, \end{cases}$$

for a parameter $\tau > 0$. Indeed, τ can generally be tuned automatically, according to some prescribed precision or a known noise level.

All these considerations led us to consider a Basis Pursuit model written under the form

$$(D) \begin{cases} \inf_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \\ \text{and } \|v - \sum_{i\in I} \lambda_i \psi_i\| \leq \tau, \end{cases}$$

for a dictionary $\mathcal{D} = (\psi_i)_{i\in I}$, $\tau > 0$ and an initial datum $v \in \mathbb{R}^N$.

The purpose of the paper is to design an efficient algorithm for solving (D).

Throughout the paper, we will assume that the dictionary is such that

$$\forall w \in \mathbb{R}^N, \exists (\lambda_i)_{i\in I}, \forall i \in I, \lambda_i \geq 0 \text{ and } w = \sum_{i\in I} \lambda_i \psi_i$$

or equivalently (the equivalence is not trivial and relies, for instance, on the variant of the Farkas Lemma given in [16], Th. 22.3, pp.199) that

$$\{w, \forall i \in I, \langle w, \psi_i \rangle \leq 1\} \text{ is bounded.}$$

When this hypothesis holds, E is a norm.

1.2 Sketch of the paper

In section 2, we build two algorithms. This is done by solving a problem (P) whose dual problem is (D). The problem (P) is stabilized to improve convergence speed (see Section 2.1 and 2.2). Then, some calculations permits to give analytical formulas for some of the necessary computations of the algorithms (see Section 2.3). They also guarantee its convergence (see Section 2.4). Two versions of the algorithms are proposed in Section 2.5 and 2.6. They are easy to implement.

Then, some experiments are explained and commented in Section 3. The experiments are described in Section 3.1, the practical convergence of the proposed algorithms is studied in Section 3.2, a bibliography on existing algorithms is made in Section 3.3 and we compare our algorithms to the main existing algorithm. This comparison shows our algorithms (in particular one of them) outperforms, by far, the existing algorithms.

2 Building algorithms

2.1 Dual formulation

We consider the optimization problem below and will show that the corresponding dual problem takes the form (D) above. As in the preceding section $v \in \mathbb{R}^N$ is the initial datum, $\mathcal{D} = (\psi_i)_{i\in I}$ is a finite subset of $\mathbb{R}^N$ (called dictionary) and $\tau > 0$.

$$(P) \begin{cases} \min_{w\in\mathbb{R}^N} \|w\| - \frac{1}{\tau}\langle w, v\rangle \\ \text{under the constraints } \forall i \in I, \langle w, \psi_i\rangle \leq 1. \end{cases}$$

The Lagrangian of the problem (P) is

$$L(w, (\lambda_i)_{i\in I}) = \|w\| - \frac{1}{\tau}\langle w, v\rangle + \sum_{i\in I} \lambda_i(\langle w, \psi_i\rangle - 1).$$

As usual (see Th. 28.3, pp 281, in [16]), the unique solution w^* to (P) is also the first argument of any saddle point $(w^*, (\lambda_i^*)_{i\in I})$ of the form

$$\min_{w\in\mathbb{R}^N} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w, (\lambda_i)_{i\in I}),$$

where, we write,

$$\mathbb{R}^{+I} = \{(\lambda_i)_{i\in I} \in \mathbb{R}^I, \forall i \in I, \lambda_i \geq 0\}.$$

All along the paper, we denote

$$\mathcal{S} = \{(\lambda_i^*)_{i\in I} \in \mathbb{R}^{+I}, (\lambda_i^*)_{i\in I} = \arg \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w^*, (\lambda_i)_{i\in I})\}. \tag{3}$$

We know that $\mathcal{S} \neq \emptyset$ (see Cor. 28.2.1, pp. 278, in [16]) but cannot guarantee it is reduced to a single element.

Notice that, L is a saddle function (i.e. : convex in w and concave in $(\lambda_i)_{i\in I}$) which satisfies the hypotheses of Th. 37.6, pp. 397, in [16] ($L(., (\lambda_i)_{i\in I})$ and $-L(w, .)$ do not have any direction of recession). So, for any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$, $(w^*, (\lambda_i^*)_{i\in I})$ is a saddle point of the form

$$\begin{aligned}
\min_{w\in\mathbb{R}^N} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w, (\lambda_i)_{i\in I}) &= \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} \min_{w\in\mathbb{R}^N} L(w, (\lambda_i)_{i\in I}) \\
&= \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} \min_{w\in\mathbb{R}^N} \left(\|w\| - \langle w, \frac{1}{\tau} v - \sum_{i\in I} \lambda_i \psi_i \rangle \right) - \sum_{i\in I} \lambda_i.
\end{aligned}$$

Finally, notice that, denoting $F(w) = \|w\|$, we have

$$\min_{w\in\mathbb{R}^N} \left(\|w\| - \langle w, \frac{1}{\tau} v - \sum_{i\in I} \lambda_i \psi_i \rangle \right) = \begin{cases} -\infty & \text{, if } v - \sum_{i\in I} \tau\lambda_i\psi_i \notin \tau\partial F(0) \\ 0 & \text{, otherwise.} \end{cases} \tag{4}$$

Also, we know that

$$\partial F(0) = \{w \in \mathbb{R}^N, \|w\| \leq 1\}. \tag{5}$$

So we finally know that any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$ is solution to

$$\begin{cases} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} & -\sum_{i\in I} \lambda_i \\ \text{under the constraint } & \|v - \sum_{i\in I} \tau\lambda_i\psi_i\| \leq \tau, \end{cases}$$

which, modulo a trivial multiplication by τ is precisely the problem (D) considered in the preceding section.

As a conclusion, the problem (D) can be solved by any algorithm solving (P) which also provides a Kuhn-Tucker vector $(\lambda_i^*)_{i\in I}$. The point is that, in fact, most algorithms solving (P) also provide such a $(\lambda_i^*)_{i\in I}$.

In the following, we will only consider a small family of such algorithms. (Our motivation for considering this family will be clear after Section 2.3 and 2.4) This family is described in the next section.

2.2 A stabilized family of algorithms

We write

$$f\left((\lambda_i)_{i\in I}\right) = \min_{w\in\mathbb{R}^N} L(w, (\lambda_i)_{i\in I}).$$

As indicated in the previous section, (D) consists in maximizing f over $\mathbb{R}^{+I}$. Assuming that we know how to evaluate ∇f at any location $(\lambda_i)_{i\in I}$ such that $f\left((\lambda_i)_{i\in I}\right)$ is finite, we could in principle apply any gradient based algorithm to achieve that goal. A typical example is the Uzawa algorithm.

Now, at each iteration, the step size of such an algorithm will have to be such that f remains finite (see (4)). This will result in a slow and unstable algorithm.

In order to avoid this problem, we propose to stabilize the algorithm with the idea used in [1, 18]. Rephrased in our context, they first consider the intermediate problems :

$$(P_u) \begin{cases} \min_{w\in\mathbb{R}^N} \alpha\|w-u\|^2 + \|w\| - \frac{1}{\tau}\langle w, v\rangle \\ \text{under the constraints : } \forall i \in I, \langle w, \psi_i\rangle \leq 1, \end{cases}$$

for $\alpha > 0$ and $u \in \mathbb{R}^N$.

Taking $u^0 \in \mathbb{R}^N$, they show in the context of their paper (see Proposition 2, in [1]) that the algorithm

$$u^{m+1} = solve(P_{u^m}), \tag{6}$$

where "solve(P_{u^m})" is the unique solution to (P_{u^m}), converges to the unique solution to (P).

In our paper, we go one step further and adapt their idea to the problem of finding a saddle point. To do so, we write

$$f_{u^m}\left((\lambda_i)_{i\in I}\right) = \min_{w\in\mathbb{R}^N} L'(w, (\lambda_i)_{i\in I}, u^m),$$

with

$$L'(w, (\lambda_i)_{i\in I}, u^m) = \alpha\|w-u^m\|^2 + \|w\| - \langle w, \frac{1}{\tau}v - \sum_{i\in I}\lambda_i\psi_i\rangle - \sum_{i\in I}\lambda_i.$$

The family of algorithm which we consider in this paper is described in Table 1. A discussion similar to the one of the preceding section guarantees that the sequence $(u^m)_{m\in\mathbb{N}}$ built by such an algorithm equals the one of the scheme (6).

– Initialize u^0 – Repeat until convergence (loop in m) 1. Use a gradient based algorithm for solving $$(\lambda_i^m)_{i\in I} = \arg\max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} f_{u^m}\left((\lambda_i)_{i\in I}\right)$$ 2. Update $u^{m+1} = \arg\min_{w\in\mathbb{R}^N} L'(w, (\lambda_i^m)_{i\in I}, u^m)$.

TAB. 1 – General form of the algorithms. The gradient based algorithm still needs to be specified.

Notice that, beside the decompositions and recompositions, the only difficulty in the implementation of the above algorithm is the computations of the gradient ∇f_{u^m}, in step 1, and the resolution of the step 2.

Our interest for the algorithms above comes from the fact that, as will be shown in the next section, those two computations can be performed exactly. Essentially, the cost of the evaluation of ∇f_{u^m} is one decomposition and one recomposition in $(\psi_i)_{i\in I}$ and the cost for computing $\arg\min_{w\in\mathbb{R}^N} L'(w, (\lambda_i^m)_{i\in I}, u^m)$ is one recomposition in $(\psi_i)_{i\in I}$.

Moreover, we will show that ∇f_{u^m} is Lipschitz and we will provide its Lipschitz constant (which can be computed numerically). This will guarantee the convergence of the gradient based algorithms considered in step 1. By the way, we will consider two variants : a projected gradient ascent of constant step size (Step 1 is then a Uzawa algorithm solving (P_{u^m})) and a Nesterov Algorithm (which is argued to be optimal among gradient based algorithm) (see [15], Section 2.2).

Before, going into those details, let us first state the following proposition which guarantees that our dual approach actually provides an approximation of the actual solution we are looking for. Its proof is given in Appendix.

Proposition 1 *The sequences $(u^m)_{m\in\mathbb{N}}$ and $\left((\lambda_i^m)_{i\in I}\right)_{m\in\mathbb{N}}$ defined in Table 1 satisfy*

1. *$(u^m)_{m\in\mathbb{N}}$ converges to the solution w^* of (P).*
2. *$\lim_{m\to+\infty} \inf_{(\lambda_i^*)_{i\in I}\in\mathcal{S}} \|(\lambda_i^m - \lambda_i^*)_{i\in I}\| = 0$, where, we recall that, $\mathcal{S}$ is the optimal set of (D).*

2.3 Exact computation of the gradient and resolution of step 2

First, as is usual with the gradient of functions defined as a minimum, many terms cancels out[1] and we finally have

$$\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) = \left(\langle w^*, \psi_i\rangle - 1\right)_{i\in I},$$

where

$$w^* = \arg\min_{w\in\mathbb{R}^N} \alpha\|w-u^m\|^2 + \|w\| - \langle w, \frac{1}{\tau}v - \sum_{i\in I}\lambda_i\psi_i\rangle. \tag{7}$$

As a consequence, modulo a decomposition in $(\psi_i)_{i\in I}$, the computation of ∇f_{u^m} and the resolution of step 2 boils down to the same problem : The resolution of (7).

Let us first simplifies the notations and consider the problem

$$w^* = \arg\min_{w\in\mathbb{R}^N} \alpha\|w-u\|^2 + \|w\| + \langle w, r\rangle,$$

where u and r are in $\mathbb{R}^N$.

Let us begin with the situation where $\|w^*\| = 0$. Differentiating, we know that $2\alpha(w^*-u)+r \in \partial F(0)$, where $F(w) = \|w\|$. Using (5), we have

$$w^* = 0 \Rightarrow \|r - 2\alpha u\| \leq 1.$$

On the other hand, if we assume that $\|w^*\| \neq 0$, we know that

$$2\alpha(w^*-u) + \frac{w^*}{\|w^*\|} + r = 0.$$

This gives

$$\|w^*\|(2\alpha u - r) = (2\alpha\|w^*\| + 1)w^*. \tag{8}$$

Taking the norm of the above equality, we obtain

$$\|2\alpha u - r\| = 2\alpha\|w^*\| + 1, \tag{9}$$

which guaranties that $\|2\alpha u - r\| > 1$.

As a conclusion,

$$w^* = 0 \Leftrightarrow \|2\alpha u - r\| \leq 1,$$

and when $w^* \neq 0$, w^* can be computed, using (8) and (9), and is

$$w^* = \frac{\|2\alpha u - r\| - 1}{2\alpha\|2\alpha u - r\|}(2\alpha u - r)$$

We can rephrase this as

$$w^* = \begin{cases} 0 & \text{, if } \|2\alpha u - r\| \leq 1 \\ \frac{\|2\alpha u - r\| - 1}{2\alpha\|2\alpha u - r\|}(2\alpha u - r) & \text{, otherwise.} \end{cases}$$

As a conclusion, in the Step 1 of the algorithm described in Table 1, the gradient can be computed with :

$$\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) = \left(\langle w^*, \psi_i\rangle - 1\right)_{i\in I}, \tag{10}$$

where

$$w^* = \begin{cases} 0 & \text{, if } \|t\| \leq 1 \\ \frac{\|t\|-1}{2\alpha\|t\|}\, t & \text{, otherwise,} \end{cases} \tag{11}$$

with

$$t = 2\alpha u^m + \frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i.$$

Moreover, the step 2 of the algorithm of Table 1 is solved by applying (11) at (λ_i^m).

[1] Notice that the differentiation is not that trivial since, in L', the optimal w depends on $(\lambda_i)_{i\in I}$. However, as is common with such max min problems, the term $\frac{\partial L'}{\partial w}$ equals zero and it cancels the terms $\frac{\partial w}{\partial \lambda_i}$ which appear in the calculation of $\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right)$. For an example of such a calculation, see the proof of Th. 9.3.3, in [5]

2.4 Computing the Lipschitz constant of the energy gradient

The calculus of the preceding section permits to prove the following result.

Proposition 2 *For any* $(\lambda_i)_{i\in I}$ *and* $(\lambda'_i)_{i\in I}$ *in* $\mathbb{R}^I$,

$$\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\| \leq C\|(\lambda_i - \lambda'_i)_{i\in I}\|,$$

with $C = \frac{3\sqrt{M_1 M_2}}{2\alpha}$, *with*

$$M_1 = \sum_{i\in I} \|\psi_i\|^2,$$

and M_2 *is the largest eigenvalue of the Gram matrix* $(\langle\psi_i, \psi_j\rangle)_{i,j\in I}$.

Proof. Let $(\lambda_i)_{i\in I}$ and $(\lambda'_i)_{i\in I}$ be in $\mathbb{R}^I$. We have, from (10),

$$\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right) = \left(\langle w^* - w'^*, \psi_i\rangle\right)_{i\in I},$$

with

$$w^* = \begin{cases} 0 & \text{, if } \|t\| \leq 1 \\ \frac{\|t\|-1}{2\alpha\|t\|}\, t & \text{, otherwise,} \end{cases} \quad \text{and} \quad w'^* = \begin{cases} 0 & \text{, if } \|t'\| \leq 1 \\ \frac{\|t'\|-1}{2\alpha\|t'\|}\, t' & \text{, otherwise,} \end{cases}$$

where

$$t = 2\alpha u^m + \tfrac{v}{\tau} - \textstyle\sum_{i\in I} \lambda_i\psi_i \quad \text{and} \quad t' = 2\alpha u^m + \tfrac{v}{\tau} - \textstyle\sum_{i\in I} \lambda'_i\psi_i.$$

In order to prove the proposition, we are going to distinguish the different possible location of $\|t\|$ and $\|t'\|$ with respect to 1.

First, if both $\|t\| \leq 1$ and $\|t'\| \leq 1$: We obviously have

$$\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\| = 0,$$

which trivially satisfies the statement of the proposition.

The second case is when $\|t\| > 1$ and $\|t'\| \leq 1$ (the case $\|t\| \leq 1$ and $\|t'\| > 1$ is similar and will not be treated) : We then have

$$\begin{aligned} \|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\|^2 &= \sum_{i\in I} \langle w^* - w'^*, \psi_i\rangle^2 \\ &\leq \|w^* - w'^*\|^2 \sum_{i\in I} \|\psi_i\|^2 \\ &\leq M_1 \left(\frac{\|t\|-1}{2\alpha\|t\|}\right)^2 \|t\|^2 \\ &\leq \frac{M_1}{4\alpha^2}(\|t\|-1)^2, \end{aligned}$$

where M_1 is given in the proposition. We also have

$$\begin{aligned} \|t\| &= \|2\alpha u^m + \frac{v}{\tau} - \sum_{i\in I} \lambda'_i\psi_i - \sum_{i\in I}(\lambda_i - \lambda'_i)\psi_i\| \\ &\leq \|t'\| + \|\sum_{i\in I}(\lambda_i - \lambda'_i)\psi_i\| \\ &\leq 1 + \|\sum_{i\in I}(\lambda_i - \lambda'_i)\psi_i\|. \end{aligned}$$

So, we have

$$(\|t\|-1)^2 \leq M_2\|(\lambda_i - \lambda'_i)_{i\in I}\|^2,$$

where M_2 is given in the proposition.

We finally obtain

$$\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\|^2 \leq \frac{M_1 M_2}{4\alpha^2}\|(\lambda_i - \lambda'_i)_{i\in I}\|^2,$$

which means

$$\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\| \leq \frac{\sqrt{M_1 M_2}}{2\alpha}\|(\lambda_i - \lambda'_i)_{i\in I}\|.$$

Therefore, the statement of the proposition holds in that case.

We finally need to study the case where both $\|t\| > 1$ and $\|t'\| > 1$: We then have

$$\begin{aligned}
\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\|^2 &= \sum_{i\in I}\langle w^* - w'^*, \psi_i\rangle^2 \\
&\leq M_1\|w^* - w'^*\|^2 \\
&\leq M_1\left\|\frac{\|t\|-1}{2\alpha\|t\|}\, t - \frac{\|t'\|-1}{2\alpha\|t'\|}\, t'\right\|^2 \\
&\leq \frac{M_1}{(2\alpha\|t\|\|t'\|)^2}\left\|(\|t\|-1)\|t'\|t - (\|t'\|-1)\|t\|t'\right\|^2 \\
&\leq \frac{M_1}{(2\alpha\|t\|\|t'\|)^2}\left\|\|t\|\|t'\|(t-t') - \|t'\|t + \|t\|t'\right\|^2.
\end{aligned}$$

So, we finally obtain

$$\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\| \leq \frac{\sqrt{M_1}}{2\alpha}\|t - t'\| + \frac{\sqrt{M_1}}{2\alpha\|t\|\|t'\|}\left\|\|t\|t' - \|t'\|t\right\|.$$

But, we also have,

$$\begin{aligned}
\left\|\|t\|t' - \|t'\|t\right\| &= \left\|\|t\|(t'-t) + (\|t\| - \|t'\|)t\right\| \\
&\leq \|t\|\|t'-t\| + \Big|\|t\| - \|t'\|\Big|\|t\| \\
&\leq 2\|t\|\|t'-t\|
\end{aligned}$$

Since,

$$\|t' - t\| \leq \sqrt{M_2}\|(\lambda_i - \lambda'_i)_{i\in I}\|,$$

we finally have

$$\begin{aligned}
\|\nabla f_{u^m}\left((\lambda_i)_{i\in I}\right) - \nabla f_{u^m}\left((\lambda'_i)_{i\in I}\right)\| &\leq \left(\frac{\sqrt{M_1 M_2}}{2\alpha} + \frac{\sqrt{M_1 M_2}}{\alpha\|t'\|}\right)\|(\lambda_i - \lambda'_i)_{i\in I}\| \\
&\leq \frac{3\sqrt{M_1 M_2}}{2\alpha}\|(\lambda_i - \lambda'_i)_{i\in I}\|.
\end{aligned}$$

□

The above proposition is important since it guarantees that some gradient based algorithm with constant step size, used in the algorithms of Table 1, converges for some step size (see next sections). Together with Proposition 1, this ensures that the whole algorithm converges to the desired solution.

However, in order to chose the step size in these algorithms we need to have an estimate of the best possible constant Lipschitz constant. This can, of course be done experimentally be running the algorithm for several step-size, when all the other parameters are fixed.

A more flexible way to chose the step size is to use the formula expressing the bound C given in Proposition 2. With this regards, for most dictionaries, all its elements but M_2 are easy to calculate.

In order to estimate M_2, one can use a standard algorithm for computing the largest eigenvalue of an operator (see [5], Section 6). Notice with this respect that the application of the Gram matrix $(\langle\psi_i, \psi_j\rangle)_{i,j\in I}$ to a vector $(\lambda_i)_{i\in I}$ requires one recomposition and one decomposition in the dictionary $(\psi_i)_{i\in I}$.

Another possibility (the one we will use in our experiments), is to use the bound (obtained from Gershgorin Circle Theorem)

$$M_2 \leq \left(\max_{i\in I} \|\psi_i\|\right)\left(\sum_{i\in I} \|\psi_i\|\right). \tag{12}$$

Finally, as can easily be seen from (10) and (11), f_{u^m} does not satisfy any sort of ellipticity property. In particular, it is not elliptic.

2.5 Uzawa version of the algorithm

In this section, we present the algorithm obtained when the gradient based algorithm used to solve the step 1 of the algorithm described in Table 1 is a simple projected gradient ascent with constant time step. The step 1 is then an Uzawa algorithm solving the dual of (P_{u^m}) (thus the name of the version). Given Proposition 2, we know (see [15], Cor. 2.1.2, pp. 70, and Th. 2.2.8, pp. 88) that it converges as soon as the time step is in the range $(0, \frac{2}{C})$, where C is given in Proposition 2. Moreover, the "best time step" is $\rho = \frac{1}{C}$.

We also know (see [15]) that, for $\rho = \frac{1}{C}$ and $u^m \in \mathbb{R}^N$, there exists a constant $C_1 > 0$ (which depends on the initialization quality) such that

$$f_{u^m}\left((\lambda_i^k)_{i\in I}\right) - f_{u^m}^* \leq C_1 \frac{2C}{k+4},$$

where $(\lambda_i^k)_{i\in I}$ is the result at the k^{th} iteration of the algorithm and

$$f_{u^m}^* = \min_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} f_{u^m}\left((\lambda_i^k)_{i\in I}\right) \tag{13}$$

The final algorithm is described in Table 2.

- Initialize $(\lambda_i^0)_{i\in I}$, $u^0 \in \mathbb{R}^N$ and $\rho = \frac{1}{C}$.
- Repeat until convergence (loop in m)
  - Repeat until convergence (loop in k)
    1. Compute $w^k = 2\alpha u^m - \sum_{i\in I} \lambda_i^k \psi_i + \frac{1}{\tau} v$
    2. if $(\|w^k\| \leq 1)$, set $w^k = 0$
       otherwise, set $w^k \leftarrow a\, w^k$, with $a = \frac{\|w^k\|-1}{2\alpha\|w^k\|}$
    3. Update λ^{k+1},
       $$\forall i \in I, \lambda_i^{k+1} = \max\left(0, \lambda_i^k + \rho(\langle w^k, \psi_i\rangle - 1)\right)$$
  - update $u^{m+1} = w^k$ and, for all $i \in I$, $\lambda_i^0 = \lambda_i^{k+1}$.

TAB. 2 – Uzawa version of the algorithm : The step 1 of the algorithm described in Table 1 is solved by a projected gradient descent with constant step size.

In practice (and in the experiments presented in Section 3), we initialized $(\lambda_i^0)_{i\in I}$ and u^0 at 0. The constant C was estimated with Proposition 2 and (12). This gives :

$$C = \frac{3\sqrt{\left(\sum_{i\in I}\|\psi_i\|^2\right)\left(\sum_{i\in I}\|\psi_i\|\right)\left(\max_{i\in I}\|\psi_i\|\right)}}{2\alpha}. \tag{14}$$

2.6 Nesterov version of the algorithm

In this section, we present a version the algorithm where the step 1 of the algorithm described in Table 1 is solved with a Nesterov algorithm (see [15], p.90). In the community involved in complexity theory, this algorithm is said to be optimal for solving the problem we are interested in.

We know that (see [15], Th. 2.2.3, pp. 80 and the remark after the description of Algorithm 2.2.19, pp. 90), for $\rho = \frac{1}{C}$ and $u^m \in \mathbb{R}^N$ there exists some constants $C_2 > 0$ (which depends on the initialization quality) and $c < C$ (and, in practice, close to C) such that

$$f_{u^m}\left((\lambda_i^k)_{i\in I}\right) - f_{u^m}^* \leq C_2 \frac{4C}{(2\sqrt{C} + ck)^2},$$

where $(\lambda_i^k)_{i\in I}$ is the result at the k^{th} iteration of the Netsterov's algorithm and $f_{u^m}^*$ is given by (13).

So, in theory, the convergence speed of this algorithm is one order of magnitude faster than the Uzawa implementation of the preceding section. The convergence speed is indeed bounded by a term whose decay is of order $\frac{1}{k^2}$. It was $\frac{1}{k}$ in the preceding section. Notice that, in image processing, a Nesterov algorithm has also been used to minimize an approximation of the total variation (see [19]).

The final algorithm is described in Table 3.

- Initialize $(\lambda_i^0)_{i\in I}$, $u^0 \in \mathbb{R}^N$ and $\rho = \frac{1}{C}$.
- Repeat until convergence (loop in m)
  - Initialize $(\mu_i^0)_{i\in I} = (\lambda_i^0)_{i\in I}$ and $r_0 = 0.999$.
  - Repeat until convergence (loop in k)
    1. Compute $w^k = 2\alpha u^m - \sum_{i\in I} \mu_i^k \psi_i + \frac{1}{\tau} v$
    2. if $(\|w^k\| \leq 1)$, set $w^k = 0$
       otherwise, set $w^k \leftarrow a\, w^k$, with $a = \frac{\|w^k\|-1}{2\alpha\|w^k\|}$
    3. Update λ^{k+1} :
       $$\forall i \in I, \lambda_i^{k+1} = \max\left(0, \mu_i^k + \rho(\langle w^k, \psi_i\rangle - 1)\right)$$
    4. Compute $r_{k+1} = \frac{-r_k + \sqrt{r_k^2 + 4r_k}}{2}$ and $t_{k+1} = \frac{r_k(1-r_k)}{r_k^2 + r_{k+1}}$
    5. Update μ^{k+1} :
       $$\forall i \in I, \mu_i^{k+1} = \lambda_i^{k+1} + t_{k+1}\left(\lambda_i^{k+1} - \lambda_i^k\right)$$
  - Set for all $i \in I$, $\lambda_i^0 = \lambda_i^{k+1}$.
  - Update u^{m+1} :
    - Compute $u^{m+1} = 2\alpha u^m - \sum_{i\in I} \lambda_i^{k+1} \psi_i + \frac{1}{\tau} v$
    - if $(\|u^{m+1}\| \leq 1)$, set $u^{m+1} = 0$
      otherwise, set $u^{m+1} \leftarrow a\, u^{m+1}$, with $a = \frac{\|u^{m+1}\|-1}{2\alpha\|u^{m+1}\|}$

TAB. 3 – Nesterov version of the algorithm.

As for the Uzawa version of the algorithm, in practice (and in the experiments presented in Section 3), we initialized $(\lambda_i^0)_{i\in I}$ and u^0 at 0. The constant C was also estimated with (14).

3 Experimental results

In Section 3.1, we give all the details on the experimental data and the quantities which will be used to evaluate the quality of the algorithms.

We display in Section 3.2 some experiments on the convergence of the algorithms presented in this paper. In particular they emphasize on the influence of α on the convergence speed of the algorithms.

In Section 3.3, we describe the algorithm proposed, by M. Elad, in [10] to solve the Basis Pursuit model and make few comments on the other existing algorithms.

Finally, in Section 3.4, we compare the different implementations (Uzawa, Nesterov, Elad) of the Basis Pursuit.

3.1 Experiments description

All the experiments are made with the same dictionary : a translation invariant discrete local cosine dictionary. It consists of all the translations of the 64 small images displayed on Figure 1. All these small images are set to 0 outside of their support and the large images (those in $\mathbb{R}^N$) are assumed periodized outside of their original support. Notice also that the dictionary is symmetrized. This means that the dictionary also contains the opposite of the elements already defined with the translations. Doing so, we obtain a model which is closer to the more common definition of the Basis Pursuit (the one minimizing the sum of the absolute value of the coordinates).

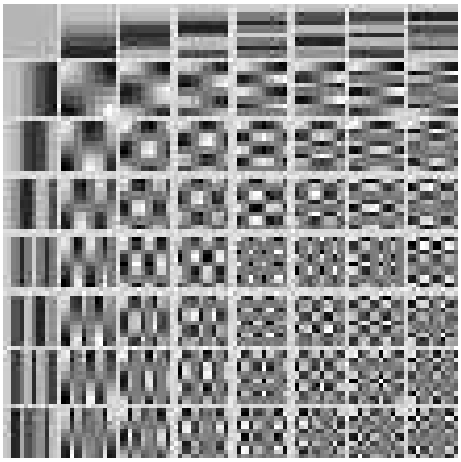

FIG. 1 – Small images defining the translation invariant discrete local cosine dictionary.

The decompositions and recompositions which are needed in the algorithms are computed with Fast Fourier Transforms, as is explained in [20].

To evaluate the quality of a decomposition $(\lambda_i)_{i\in I}$ approximating an image v, we consider three quantities :

$$l^0 = \frac{200}{\#I}\#\{i \in I, \lambda_i \neq 0\}, \tag{15}$$

where $\#$ denotes the cardinal of a set. Notice that we put 200 (not 100). This is due to the fact that, for every $i \in I$, there exists $j \in I$ such that $\psi_j = -\psi_i$. With the 200, l^0 represents the percentage of elements of I which appear in the decomposition, without regard to the sign of the corresponding coordinate.

Similarly, we consider

$$l^1 = \frac{2}{\#I}\sum_{i\in I}\lambda_i, \tag{16}$$

(remember that, for all $i \in I$, $\lambda_i \geq 0$) and

$$l^2 = \|\sum_{i\in I}\lambda_i\psi_i - v\|, \tag{17}$$

where, for any $u \in \mathbb{R}^N$,

$$\|u\| = \sqrt{\frac{1}{N}\sum_{l=0}^{N-1}u_l^2}.$$

Those are the quantities evaluated by the curves displayed on Figure 4,5,6,...

In all the curves presented in the paper (Figure 4,5,6,...) the scale of the x-axis is given by the iteration number (those indexed by k in Table 2 and Table 3). In terms of computational effort, this corresponds mainly to one decomposition and one recomposition in the dictionary $(\psi_i)_{i\in I}$.

Finally, in the experiments, we either have v equal to the image Barbara (see Figure 2) or v equal to an extracted part of it (see Figure 3).

FIG. 2 – Image used for the experiments.

3.2 Convergence of the Uzawa and Nesterov algorithms

First, we would like to point out that despite the theoretical guarantees concerning the convergence of these algorithms, we found few situations where the step size $\frac{1}{C}$ given in the algorithm description (see Table 2 and Table 3) is too large. However, we would like to point out that those rare cases are easy to detect since both l^2 and l^1 rapidly blow up. It is then possible to increase the estimation of the Lipschitz constant C and run the algorithm again. Since, this problem rarely occurs, we have not implemented such a loop. We would recommend to do it if the algorithm is used in a fully automatic way.

This issue being clarified, the algorithms have good convergence properties. As can be seen in the preceding sections, beside the parameters of the problem $\mathcal{D}$, τ and v, the only parameter of the algorithm is α (see the definition of (P_u)). Our experiments therefore aim at understanding its role on the convergence properties of the algorithms.

In fact, it plays the same role in both the Uzawa and the Nesterov implementations. So, we only display the curves for the Uzawa version of the algorithm. All the curves which we comment and display in this section concern experiments with the image on Figure 3, $\tau = 0.0445$ and the dictionary described in Section 3.1.

The first issue we would like to address is the convergence of l^2. In theory, it should converge to τ. This is actually the case in our experiments. We display on Figure 4 the curves representing l^2 as a function of the iteration number, for the 400 first iterations. Up to negligible fluctuations, the curves are constant for the iteration between 400 and 3000.

We see on Figure 4 (and this was confirmed in many other experiments for both the Uzawa and the Nesterov versions of the algorithm) that, as far as the l^2 criterion is concerned, a small value α is preferable.

We display on Figure 5 the curves representing the quantity l^1 as a function of the iteration number. Again, those curves are representative of many other experiments confirming the same statement : As far as the l^1 criterion is concerned, a large value α is preferable. Notice that adding more iterations does not permit to improve the result as much as a change of α.

The quantity l^0 is also of a particular interest, since people often use the Basis Pursuit model to obtain

FIG. 3 – Image extracted from Barbara (the image on Figure 2). It is used for the experiments displayed under the form of curves.

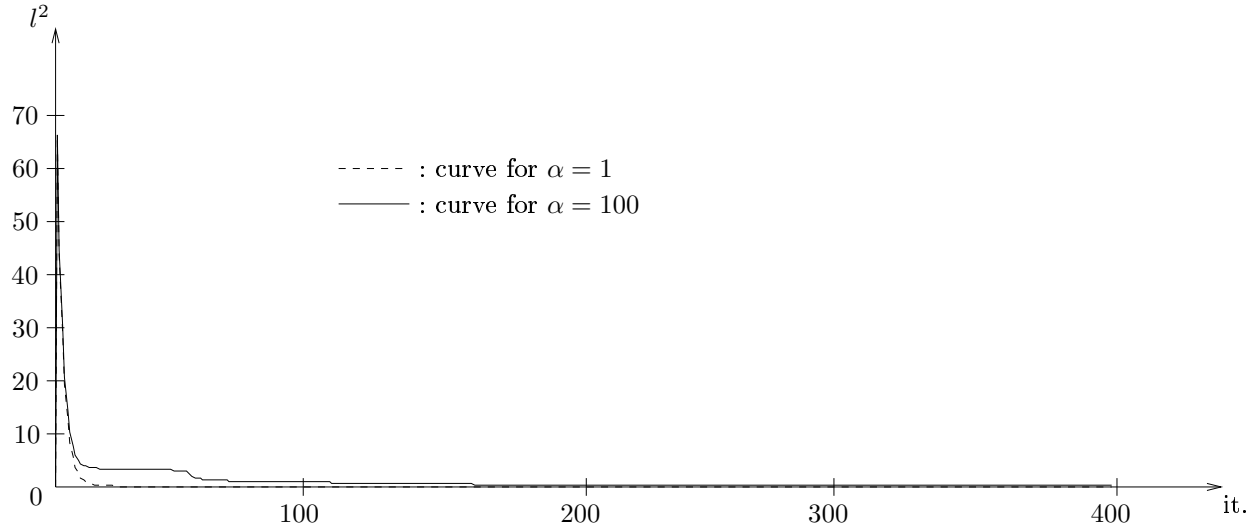

FIG. 4 – l^2 curves : The drawn curves give the criterion l^2 (see (17)) as a function of the iteration number, for $\alpha = 1$ and $\alpha = 100$. Typically, the curves for other values of α and the Nesterov version of the algorithm are similar. They always converge to the prescribed value (0.0445 in the experiment).

a result which is sparsely represented in the dictionary $\mathcal{D}$. We display on Figure 6 the curves representing the quantity l^0, as a function of the iteration number. These curves are, of course, very much correlated to those concerning the l^1 criterion. We get the conclusions : As far as the l^0 criterion is concerned, a large value α is preferable. Again, adding more iterations does not permit to improve the result as much as a change of α.

As a conclusion, it seems that a wise way to chose the parameter α is to chose α as large as possible, such that the convergence of the l^2 criterion is satisfactory, given the applicative context and the number of iterations allowed.

Also, we have not tried to make α evolve as a function of the iteration number (or some other extra criterion such as a test on the l^2 criterion). This should, of course, improve the results we obtained in our experiments.

3.3 Existing algorithms for solving the Basis Pursuit model

As already mentioned in the introduction, there are surprisingly few algorithms for solving Basis Pursuit based regularisation models. All those we found [6, 2, 10, 11, 14] deal with the model under its form :

$$\min_{(\lambda_i)_{i\in I}\in\mathbb{R}^I} \| \sum_{i\in I} \lambda_i \psi_i - v\|^2 + \lambda \sum_{i\in I} |\lambda_i|. \tag{18}$$

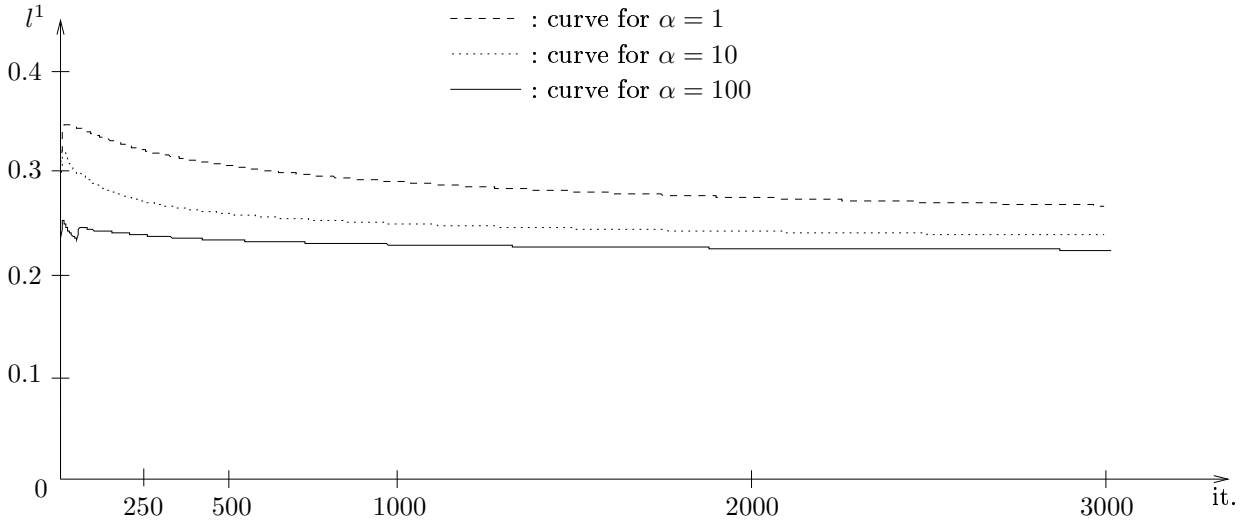

FIG. 5 – l^1 curves : The drawn curves give the criterion l^1 (see (16)) as a function of the iteration number, for $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$ and the Uzawa version of the algorithm. Typically, the curves for other values of α and the Nesterov version of the algorithm are similar.

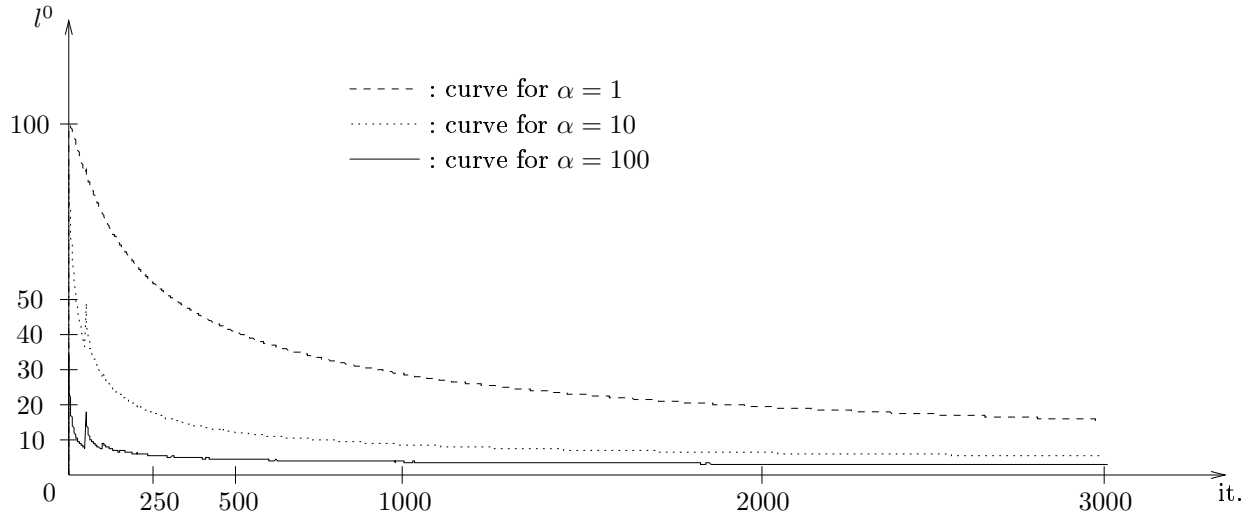

FIG. 6 – l^0 curves : The drawn curves give the l^0 criterion (see (15)) as a function of the iteration number, for $\alpha = 1$, 10 and 100. We observed the same behavior for the Nesterov version of the algorithm.

Let us denote, for all $(\lambda_i)_{i\in I} \in \mathbb{R}^I$

$$f\left((\lambda_i)_{i\in I}\right) = \|\sum_{i\in I}\lambda_i\psi_i - v\|^2 + \lambda\sum_{i\in I}|\lambda_i|,$$

and, for $\sigma > 0$ and $t \in \mathbb{R}$,

$$S_\sigma(t) = \begin{cases} t - \frac{\sigma}{2} & \text{, if } t \geq \frac{\sigma}{2} \\ 0 & \text{, if } |t| < \frac{\sigma}{2} \\ t + \frac{\sigma}{2} & \text{, if } t \leq -\frac{\sigma}{2}. \end{cases}$$

We describe in Table 4 the algorithm proposed by M. Elad in [10]. This is the existing algorithm to which we will compare our results.

- Initialize $(\lambda_i^0)_{i\in I}$.
- Repeat until convergence (loop in k)
  1. Compute d_i^k, for all $i \in I$:

$$d_i^k = S_{\frac{\lambda}{\|\psi_i\|^2}}\left(\lambda_i^k + \frac{1}{\|\psi_i\|_2^2}\langle v - \sum_{i\in I}\lambda_i^k\psi_i, \psi_i\rangle\right) - \lambda_i^k.$$

  2. Compute the optimal step :

$$t^k = \arg\min_{t\in\mathbb{R}} f\left((\lambda_i^k)_{i\in I} + t(d_i^k)_{i\in I}\right).$$

  3. Update λ^{k+1} :

$$\forall i \in I, \lambda_i^{k+1} = \lambda_i^k + t^k d_i^k.$$

TAB. 4 – The algorithm, solving (18), described in [10].

Let us explain this choice. First, remark that the algorithms proposed in [6, 2] are easy to explain, given Table 4. When all the elements of the dictionary are normalised, they indeed correspond to the algorithm of Table 4 where we always choose $t^k = 1$. According to our experiments, the introduction of t^k greatly stabilizes the algorithm.

Finally, the main innovation, in [11], is to replace t^k by a M-dimensional vector. Its computation is then performed by minimizing f over $\text{Span}\left((d^{k+1-m})_{m\in\{0,\dots M\}}\right)$. Although this obviously improves the convergence results, we have not implemented this algorithm. It seems indeed to provide only a relatively small improvement when compared to the algorithm described in Table 4 (see [11]). This improvement is made at the price of an important effort in the implementation of the algorithm.

Finally, the algorithm proposed in [14] is very elegant and has the advantage of being exact. However, it does require, at each iteration, the inversion of a matrix. The size of this matrix goes to the number of non-zero coordinates of the result. This restricts its use to applications where this number remains very small.

3.4 Comparison of the algorithms

We display on Figure 7, 8, 9, 10, 11 and 12 the curves corresponding to a comparison between the Uzawa and Nesterov versions of our algorithm (see Table 2 and 3) and the Elad Algorithm described in Table 4.

Concerning the choice of the parameters, the purpose of our paper is obviously not to answer the question : How to fix λ in the model (18) ? So our only choice is to follow the steps :

- Run the Elad algorithm for a given value λ.
- Compute τ : the l^2 criterion for the result of Elad algorithm.
- Run Uzawa or Nesterov version of the algorithm for this τ.

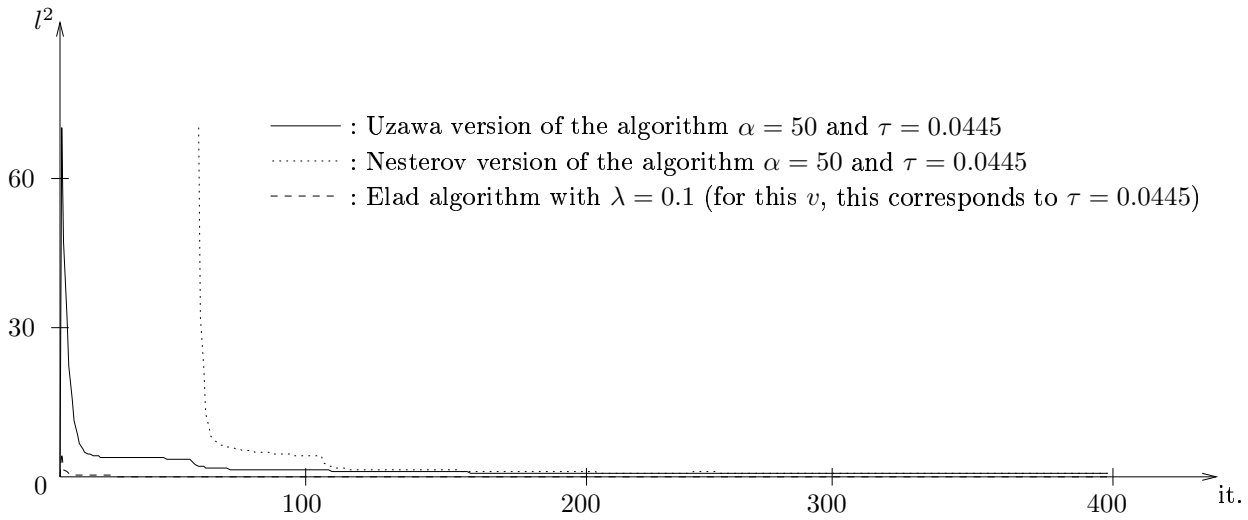

FIG. 7 – l^2 curves : The drawn curves give the criterion l^2 (see (17)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). For the Nesterov algorithm, the piece of curve corresponding to the iterations between 1 and 52 is not drawn since its values are around 800. The parameter $\tau = 0.0445$ corresponds to l^2 criterion applied to the result of the Elad Algorithm.

This results in an unfair comparison favoring the Elad algorithm.

We display on Figure 7, 8 and 9 the l^2, l^1 and l^0 criterion as a function of the iteration number. This experiment is made for $\lambda = 0.1$, in (18), which corresponds to $\tau = 0.0445$, in (D). Concerning the comparison between the Uzawa and the Nesterov versions of the algorithm, we find that, because of some small instability, the Nesterov version takes more time to converge. However, it seems to converge to the same solution.

Concerning the convergence of the l^2 criterion, Elad is much faster (see Figure 7). It is not clear whether we would find the same result when λ is tuned in order to reach a given precision level τ. (This would clearly depend on the strategy used to achieve this goal.)

The convergence of the l^1 and l^0 criterion is in favor of our implementation (see Figure 8 and 9). In particular, none of the coordinates are canceled by the Elad implementation of the Basis Pursuit, our implementation has less than 2.8% non-zero coordinates (after the 3000 iterations).

Finally, concerning the Elad implementation, we observe (this is corroborated by many other experiments) that, modulo negligible changes, it stops evolving after few iterations (say 20).

We also display the curves corresponding to the same experiment for $\lambda = 200$, in (18), which corresponds to $\tau = 14.9973$, in (D), for the small image displayed on Figure 3. Although the situation is completely different, we can draw, from these curves, exactly the same conclusions as in the previous case.

To conclude with these curves, notice that the Uzawa version of our algorithm reaches a fair level of convergence after few hundreds of iteration (i.e. few hundreds of decomposition/recomposition in the dictionary $(\psi_i)_{i\in I}$).

In the case $\lambda = 200$ (i.e. $\tau = 475.852$, when v is the large image displayed on Figure 2), the qualitative difference between the algorithms under study can be displayed under the form

$$\sum_{i\in I} \lambda_i \psi_i,$$

for the results $(\lambda_i)_{i\in I}$ of the algorithms. This image is indeed significantly different from the initial image v (this was not the case for $\lambda = 0.1$). We display on Figure 13 a sharpened part of the reconstruction of the result obtained by the Elad algorithm[2]. Mostly, it is blurry and many details are lost. The same sharpened part of the result of the Uzawa algorithm[2] is displayed on Figure 14. It is much sharper and much more

[2]The electronic images are available at *http ://www.math.univ-paris13.fr/∼malgouy/research.html*

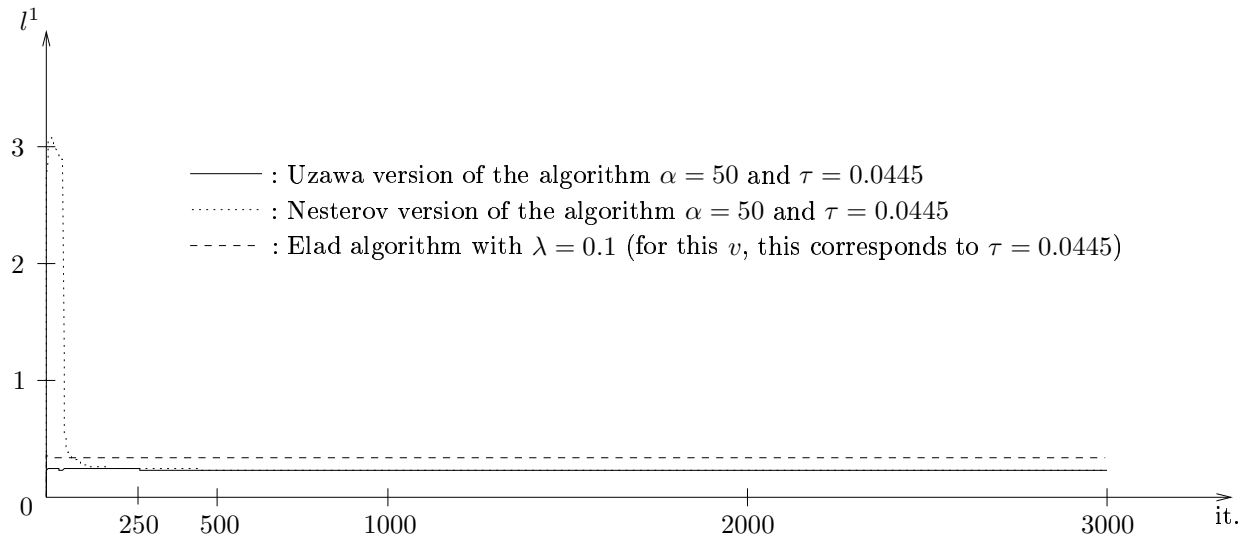

FIG. 8 – l^1 curves : The drawn curves give the criterion l^1 (see (16)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). The parameter $\tau = 0.0445$ corresponds to l^2 criterion applied to the result of the Elad Algorithm.

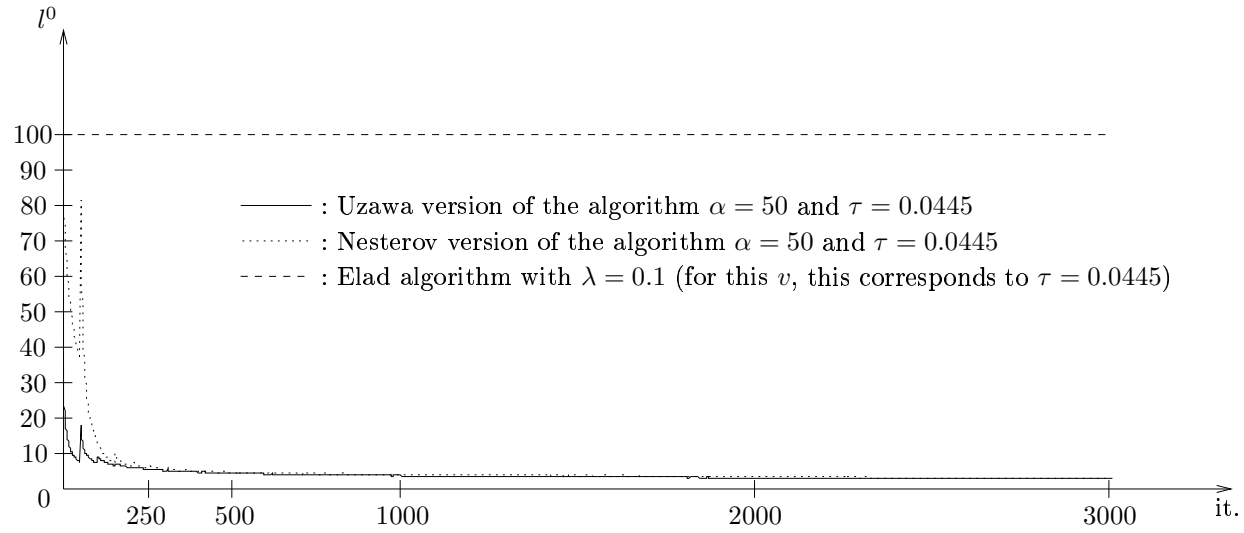

FIG. 9 – l^0 curves : The drawn curves give the criterion l^0 (see (15)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). The parameter $\tau = 0.0445$ corresponds to l^2 criterion applied to the result of the Elad Algorithm.

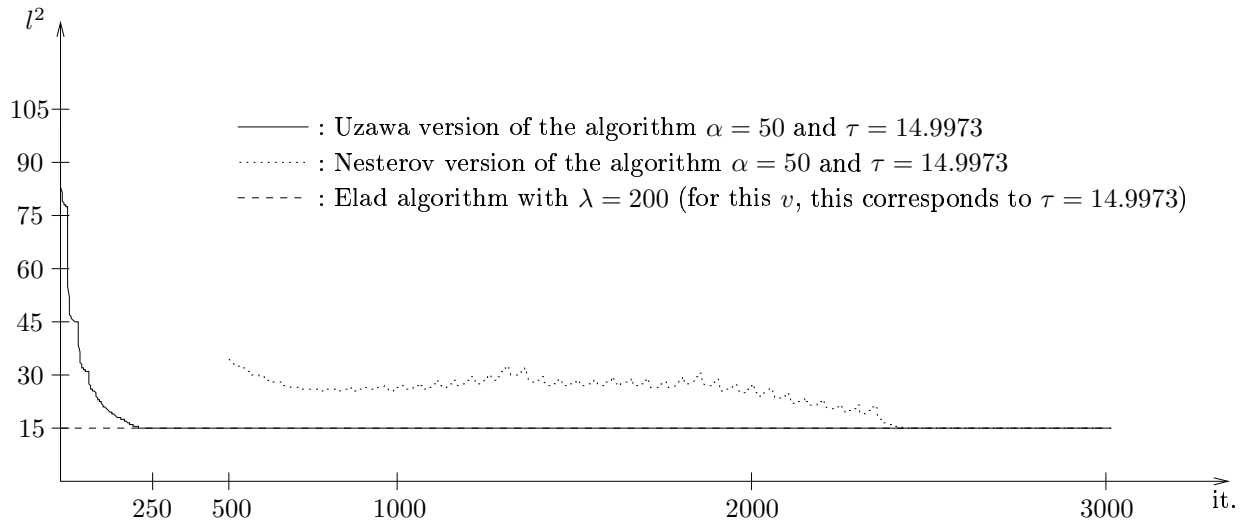

FIG. 10 – l^2 curves : The drawn curves give the criterion l^2 (see (17)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). For the Nesterov algorithm, the piece of curve corresponding to the iterations between 1 and 500 so that the difference between Uzawa and Elad algorithms is visible. The parameter $\tau = 14.9973$ corresponds to l^2 criterion applied to the result of the Elad Algorithm.

details are present. However, some drawbacks of the model also start to appear (this should be improved by modifying the dictionary). Some sort of "ringing" artifact appears next to contrasted edges. Its "width" corresponds to the size of the support of the elements in the dictionary (namely 8). Also, some details are extended out of their original support. For instance, the texture of the scarf is present on Barbara's chin.

Appendix : proof of Proposition 1

First, notice that, because of the construction of f_m and L', u^{m+1} as defined by Table 1 is solution to (P_{u^m}), where we recall that

$$(P_u) \begin{cases} \min_{w \in \mathbb{R}^N} \alpha \|w - u\|^2 + \|w\| - \frac{1}{\tau}\langle w, v\rangle \\ \text{under the constraints : } \forall i \in I, \langle w, \psi_i\rangle \leq 1. \end{cases}$$

The statement 1 is therefore a simple adaptation of the Proposition 2, in [1], to a different energy. A straightforward adaptation of their proof works and need not be written in details.

Let us focus on the proof of the second statement. Its proof decomposes into two stages :

1. For any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$,
$$\lim_{m\to+\infty} \sum_{i\in I} \lambda_i^m \psi_i = \sum_{i\in I} \lambda_i^* \psi_i.$$
(Notice that this guarantees that $\sum_{i\in I} \lambda_i^* \psi_i$ is independent on the particular choice of $(\lambda_i^*)_{i\in I} \in \mathcal{S}$. Which is not surprising.)
2. The sequence $\left((\lambda_i^m)_{i\in I}\right)_{m\in\mathbb{N}}$ is bounded in $\mathbb{R}^I$ and all its accumulation points solve (D).

In order to prove the first stage, we first consider the situation where (P) is such that $w^* = 0$. If this occur, we obviously have $\|v\| \leq \tau$ (look at (P)). In turns, this implies that, for all $i \in I$, $\lambda_i^* = 0$ (just look at (D)).

Now, for all $m \in \mathbb{N}$, if there exists $i \in I$ such that $\lambda_i^m > 0$, we can deduce from the Kuhn-Tucker relation (see Th. 28.3, pp 281, [16]) that $\langle u^{m+1}, \psi_i\rangle = 1$. This implies that

$$\sup_{i\in I} \langle u^{m+1}, \psi_i\rangle \geq 1.$$

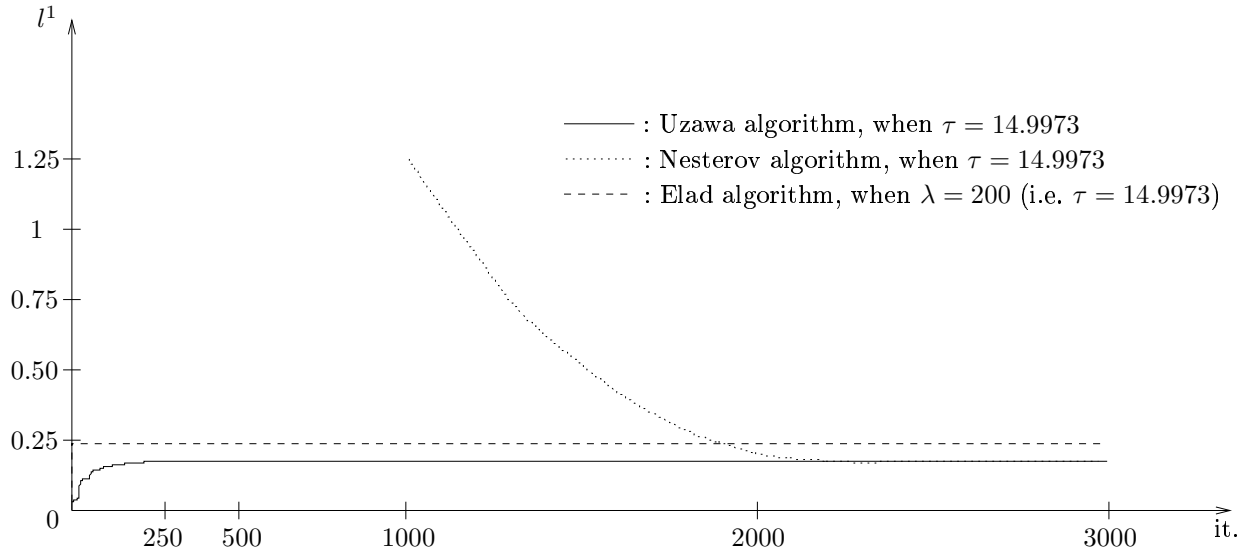

FIG. 11 – l^1 curves : The drawn curves give the criterion l^1 (see (16)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). For the Nesterov algorithm, the piece of curve corresponding to the iterations between 1 and 1000 so that the difference between Uzawa and Elad algorithms is visible. The parameter $\tau = 14.9973$ corresponds to l^2 criterion applied to the result of the Elad Algorithm. For Nesterov and Uzawa algorithms we took $\alpha = 50$.

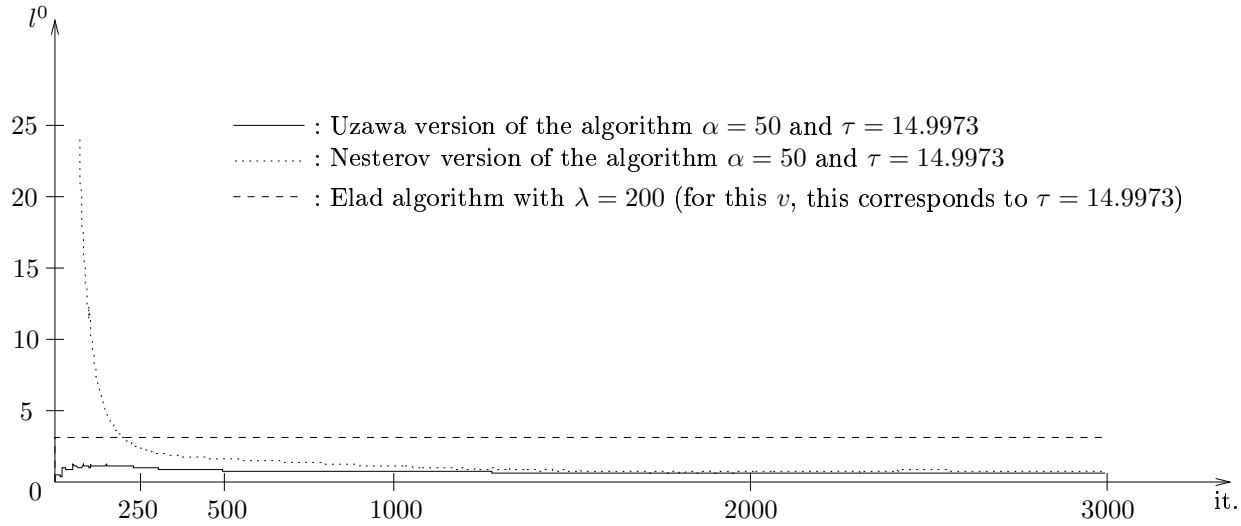

FIG. 12 – l^0 curves : The drawn curves give the criterion l^0 (see (15)) as a function of the iteration number, for the Elad Algorithm (see Table 4), the Nesterov and Uzawa versions of the algorithm (see Table 3 and 2). For the Nesterov algorithm, the piece of curve corresponding to the iterations between 1 and 70 is not drawn since its values reach 85%. The parameter $\tau = 14.9973$ corresponds to l^2 criterion applied to the result of the Elad Algorithm.

FIG. 13 – Part of the image obtained with Elad algorithm, for $\lambda = 200$ (i.e. $\tau = 475.852$). (The image has been **sharpened** for the display.)

As a consequence, since $\lim_{m\to+\infty} u^{m+1} = w^* = 0$, there exists $M > 0$ such that for all $m > M$ and for all $i \in I$, $\lambda_i^m = 0$.

As a conclusion, when $w^* = 0$,

$$\lim_{m\to+\infty} \sum_{i\in I} \lambda_i^m \psi_i = \sum_{i\in I} \lambda_i^* \psi_i.$$

Let us now assume that $w^* \neq 0$. We first remark that for any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$,

$$\frac{w^*}{\|w^*\|} - \frac{1}{\tau} v + \sum_{i\in I} \lambda_i^* \psi_i = 0.$$

Since u^{m+1} converges to w^*, for m large enough, u^{m+1} cannot be zero. Given the definition of u^{m+1}, we also know that

$$2\alpha(u^{m+1} - u^m) + \frac{u^{m+1}}{\|u^{m+1}\|} - \frac{1}{\tau} v + \sum_{i\in I} \lambda_i^m \psi_i = 0.$$

We finally obtain

$$\sum_{i\in I} (\lambda_i^* - \lambda_i^m)\psi_i = 2\alpha(u^{m+1} - u^m) + \frac{u^{m+1}}{\|u^{m+1}\|} - \frac{w^*}{\|w^*\|}.$$

Since $(u^m)_{m\in\mathbb{N}}$ converges to w^*, we have

$$\lim_{m\to+\infty} \sum_{i\in I} (\lambda_i^m - \lambda_i^*)\psi_i = 0.$$

In order to establish the second step of the proof, we first remark that because of the definition u^{m+1} and $(\lambda_i^{m+1})_{i\in I}$, we have for any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$ (as for any element of $\mathbb{R}^{+I}$),

$$L'(u^{m+1}, (\lambda_i^{m+1})_{i\in I}, u^m) \geq L'(u^{m+1}, (\lambda_i^*)_{i\in I}, u^m).$$

FIG. 14 – Part of the image obtained with Uzawa algorithm, for $\tau = 475.852$. (The image has been **sharpened** for the display.)

Using the definition of L', we obtain

$$\langle u^{m+1}, \sum_{i\in I} \lambda_i^{m+1}\psi_i\rangle - \sum_{i\in I}\lambda_i^{m+1} \geq \langle u^{m+1}, \sum_{i\in I}\lambda_i^*\psi_i\rangle - \sum_{i\in I}\lambda_i^*.$$

So,

$$\sum_{i\in I}\lambda_i^{m+1} \leq \sum_{i\in I}\lambda_i^* + \|u^{m+1}\| \left\| \sum_{i\in I}(\lambda_i^* - \lambda_i^{m+1})\psi_i \right\|. \tag{19}$$

Since $\lim_{m\to+\infty} u^m = w^*$ and $\lim_{m\to+\infty}\sum_{i\in I}(\lambda_i^* - \lambda_i^{m+1})\psi_i = 0$, we are sure that there exists $B > 0$, such that, for all $m \in \mathbb{N}$

$$\sum_{i\in I}\lambda_i^m \leq B.$$

Let $(\overline{\lambda}_i)_{i\in I}$ be an accumulation point of $((\lambda_i^m)_{i\in I})_{m\in\mathbb{N}}$, we obtain, using (19),

$$\sum_{i\in I}\overline{\lambda}_i \leq \sum_{i\in I}\lambda_i^*.$$

Now, since $\lim_{m\to+\infty}\sum_{i\in I}\lambda_i^m\psi_i = \sum_{i\in I}\lambda_i^*\psi_i$, we obviously have

$$\sum_{i\in I}\overline{\lambda}_i\psi_i = \sum_{i\in I}\lambda_i^*\psi_i.$$

Using the fact that $(\lambda_i^*)_{i\in I}$ solves (D), we finally have

$$\sum_{i\in I}\overline{\lambda}_i = \sum_{i\in I}\lambda_i^*,$$

which implies $(\overline{\lambda}_i)_{i\in I} \in \mathcal{S}$.

This concludes the proof.

Références

[1] A. Almansa, C. Ballester, V. Caselles, and G. Haro. A tv based restoration model with local constraints. IMA preprint series 2119, IMA, May 2006. Available at : http ://www.ima.umn.edu/preprints/may2006/may2006.html.

[2] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A l1-unified variational framework for image restoration. *Lecture notes in Computer Science*, 2004. Proc. ECCV 2004.

[3] M. Brown and N.P. Costen. Exploratory basis pursuit classification. *Pattern Recognition Letters*, 26 :1907–1915, 2005.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1) :33–61, 1999.

[5] P. Ciarlet. *Introduction to numerical linear algebra and optimisation.* Cambridge University Press, 1989.

[6] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problem with sparsity constraint. *Communication on Pure and Applied Mathematics*, 57(11) :1413–1457, Aug. 2004.

[7] D. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. Technical Report 2005-04, Dept of Statistics, Stanford University, January 2005.

[8] D. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representation in the presence of noise. *IEEE, Trans. on Information Theory*, 52 :6–18, 2006.

[9] D. Donoho and J. Tanner. Sparse nonegative solution of underdetermined linear equations by linear programming. Technical Report 2005-06, Stanford University, April 2005.

[10] M. Elad. Why simple shrinkage is still relevant for redundant transforms. *IEEE, Trans. on Information theory*, 52(12) :5559–5569, Dec. 2006.

[11] M. Elad, B. Matalon, and M. Zibulevsky. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Journal on Applied and Computational Harmonic Analysis*, 2007. To appear.

[12] F. Malgouyres. Projecting onto a polytope simplifies data distributions. Technical Report 2006-1, University Paris 13, January 2006.

[13] F. Malgouyres. Rank related properties for basis pursuit and total variation regularization. Technical Report ccsd-00020801, CCSD, March 2006. Accepted to Signal Processing (minor modifications).

[14] S. Maria and J.J. Fuchs. Application of the global matched filter to stap data : an efficient algorithmic approach. In *Proceedings of ICASSP 2006*, volume 4, pages 1013–1016, Toulouse, France, May 2006.

[15] Y. Nesterov. *Introductory lectures on convex optimization : A basic course.* Kluwer Academic Publishers, 2004.

[16] R.T. Rockafellar. *Convex analysis.* Princeton University Press, 1970.

[17] J-L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE, Trans. on Image Processing*, 14(10) :1570–1582, October 2005.

[18] L. Vese. A study in the bv space of a denoising-deblurring variational problem. *Applied Mathematics and Optimization*, 44 :131–161, 2001.

[19] P. Weiss, L. Blanc-Feraud, and G. Aubert. Sur la complexité et la rapidité d'algorithmes pour la minimisation de la variation totale sous contraintes. Technical report, INRIA, 2007.

[20] T. Zeng and F. Malgouyres. Using gabor dictionaries in a $tv - l^\infty$ model, for denoising. In *Proceedings of ICASSP 2006*, volume 2, pages 865–868, Toulouse, France, May 2006.