

# Convergence of a Sparse Representations Algorithm Applicable to Real or Complex Data

Jean Jacques Fuchs, *Member, IEEE*

**Abstract**—Sparse representations has become an important topic in recent years. It consists in representing, say, a signal (vector) as a linear combination of as few as possible components (vectors) from a redundant basis (of the vector space). This is usually performed, either iteratively (adding a component at a time), or globally (selecting simultaneously all the needed components). We consider a specific algorithm, that we obtain as a fixed point algorithm, but that can also be seen as an iteratively reweighted least-squares algorithm. We analyze it thoroughly and show that it converges to the global optimum. We detail the proof in the real case and indicate how to extend it to the complex case. We illustrate the result with some easily reproducible toy simulations, that further illustrate the potential tracking properties of the proposed algorithm.

**Index Terms**—Convergence of numerical methods, fixed-point algorithms, iterative methods, minimization methods, spectral analysis.

## I. INTRODUCTION

THERE is growing interest in sparse representations, which is a technique that allows to decompose a signal into a small number of components chosen from an over-complete set of vectors or signals often called a dictionary or a redundant basis. The domains of applications are numerous, but one can essentially distinguish the case where there exists indeed an exact sparse representation possibly observed in additive noise and the case where one aims to represent a complex signal by a simplified model.

From a different—more abstract—point of view, one may also say that one seeks—either the sparsest exact representation of the signal in terms of the elementary components—or the representation of a given complexity that minimizes a certain approximation error—or the sparsest representation that yields an approximation error smaller than a specified threshold.

Theoretical results concerning the first of these questions have been obtained recently. Given an  $n \times m$  matrix  $A$  with  $m > n$  and a vector  $b$  that indeed admits an exact sparse representation, say  $b = Ax_o$ , it has been shown that, if the number of nonzero entries in  $x_o$  is smaller than a given bound, then  $x_o$  is the unique sparsest representation [1], [2]. Since searching for the sparsest representation is a nonpolynomial (NP) hard problem that can only be solved by exhaustive search, one

replaces the true search for the sparsest solution:  $\min_x \|x\|_0$  subject to  $Ax = b$  with  $\|x\|_0$  the number of nonzero entries in  $x$ , by the easy-to-solve linear program

$$\min_x \|x\|_1 \quad \text{subject to} \quad Ax = b \quad (1)$$

i.e., one minimizes the  $\ell_1$  norm of  $x$  instead of the sparsity itself. Here and below, we denote  $\|x\|_k$  the  $\ell_k$  norm of a vector  $x$ , defined as  $\|x\|_k = [\sum_1^m |x_j|^k]^{1/k}$  for  $k \geq 1$ . The problem is then to determine sufficient conditions for the two criteria to have the same unique solution, or, for the criterion (1) to recover the sparsest solution. These problems have been initiated in [1] and developed since by several other authors e.g., [2]–[4].

Since seeking the sparsest exact representation may be useless either because there is none or there is one but observed in additive noise, an approximate reconstruction is often preferable. It then makes sense to replace (1) by

$$\min_x \|x\|_1 \quad \text{subject to} \quad \|Ax - b\|_2^2 \leq \rho^2$$

with  $\rho^2$  the tolerance to be defined, or, somehow equivalently, by the following criterion:

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + h \|x\|_1, \quad h > 0. \quad (2)$$

One now seeks the representation with smallest  $\ell_1$ -norm that yields an approximation error smaller than a specified threshold. In practice, this is the standard situation when one seeks a compact representation of an audio or video signal either for transmission or simply signal reconstruction purposes [5], [6]. The algorithms, that allow to build such a sparse approximate representation such as matching pursuit [7], a suboptimal algorithm that selects sequentially the components, linear programming that minimizes the linear program also known as the basis pursuit criterion (1), or quadratic programming algorithms, that minimize (2) [8], [9], are quite time consuming and a lot of effort is steadily made to develop “fast” variants [10]–[12] and dedicated software architectures leading to fast implementations are developed [13]. Another class of algorithms which can be used to solve (2) is that of “iterative thresholding” or “iterative denoising”, which were proposed under an expectation-maximization framework in [14] and whose convergence proofs can be found in [15]. These algorithms also benefit from an initialization near the solution, so they can be used to track solutions, just as the method we propose in this manuscript.

Let us mention also, that more recently, (2) has become the criterion of choice in the “compressed sensing” domain [16]–[19]. See also the compressed sensing repository of Rice University <http://www.dsp.ece.rice.edu/cs/>, as well as the following sites <http://www.11-magic.org/>, [Manuscript received January 31, 2007; revised August 31, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yonina Eldar.](http://</a></p>
</div>
<div data-bbox=)

The author is with the IRISA/Université de Rennes 1, 5042 Rennes Cedex, France (e-mail: fuchs@irisa.fr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2007.909363

www.lx.it.pt/mtf/GPSR/, [http://www.stanford.edu/boyd/l1\\_ls/](http://www.stanford.edu/boyd/l1_ls/) and <http://www.sparselab.stanford.edu/>, to cite a few, that propose algorithms that solve (2).

As a matter of fact, most of these algorithms are generic, in the sense that—aside from sparsity—they do not exploit any other information that may be available such as proximity of the sought sparse representation to a known one. Indeed, in many applications, where a stream of representations is desired, temporal (in audio and video) or physical (in image) proximity is present and guarantees some kind of proximity in the representations of neighboring or adjacent signals. Taking into account the fact that such neighboring signals have sparse representations that are close, should certainly lead to faster implementations. Clearly gradient-based algorithms can be used for this purpose, but other algorithms, that allow to get sparse representations, and can be modified to possess tracking properties, do exist [20]–[24].

We propose to analyze thoroughly one of them in the sequel. In Section II, we further specify the criterion, present the optimality conditions and introduce the iterative optimization algorithm. We propose a self-contained analysis of its convergence properties in the real case in Section III and indicate how to extend the proof to the complex case in Section IV. For illustration purposes, we present in Section V, the output of the algorithm for two elementary simulations in the complex case.

## II. PROBLEM FORMULATION

### A. Criterion

We consider the criterion (2) introduced above

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + h \|x\|_1, \quad h > 0.$$

It is a specific instance of a more general family of penalized or regularized criteria that is currently often considered [8], [9], [21] and has many different interpretations. Its sparseness properties have been established and compared to those of (1) in [3] where it is further shown that as  $h$  decreases to zero, the optimum of (2) converges to the optimum of (1). The parameter  $h$  in (2) has indeed a nice physical interpretation [3], [9] that is best seen on its Lagrangian dual. The criterion (2) can be transformed into a quadratic program and thus solved using standard subroutines. Though it is only convex and not strictly convex, we will assume, in the sequel, that its optimum is attained at a unique point. This assumption is made to simplify the exposition. The nonstrict convexity is due to both the *linear* penalty term and the semi positive definiteness of the quadratic term. The minimum point has at most  $n$  nonzero components, this follows from the quadratic programming theory [25] but can also deduced from the developments given below.

### B. Optimality Conditions

Since the problem is convex, the first-order optimality conditions are sufficient and there are different means to obtain them [9], [26]. A quick way consists in introducing the subdifferential

of  $\|x\|_1$  [25], a set of vectors called the subgradients, denoted  $\partial\|x\|_1$ , as follows:

$$\begin{aligned} \partial\|x\|_1 &= \{u | u^T x = \|x\|_1, \|u\|_\infty \leq 1\} \\ &= \{u | u_i = \text{sign}(x_i) \text{ if } x_i \neq 0 \text{ and } |u_i| \leq 1 \text{ otherwise}\} \end{aligned}$$

where  $\text{sign}(x_i) = 1$  when  $x_i > 0$  and  $\text{sign}(x_i) = -1$  when  $x_i < 0$ . A necessary and sufficient condition (NSC) for  $x$  to be a global minimum of (2) is that there exists a subgradient of the criterion at  $x$  that is equal to zero [25]

$$\exists u \in \partial\|x\|_1 \quad \ni \quad A^T(Ax - b) + hu = 0. \quad (3)$$

This is an implicit set of equations that is difficult to solve. Under the assumption we made above, there is a unique couple  $\{x, u\}$ , say  $\{x_o, u_o\}$ , where  $u_o$  is a subgradient of  $\|x\|_1$  at  $x_o$  that satisfies it. Only a lengthy combinatorial search could lead to it.

It follows from (3) that if  $h \geq \|A^T b\|_\infty$ , the unique optimum is at the origin, while otherwise the optimum we denote  $x_o$  is a function of  $h$  that has at most  $n$  nonzero components. Here and quite generally below the subscript ‘ $o$ ’ is used to designate the sparse representation we aim to recover, it will also be associated with other quantities to emphasize their link with  $x_o$ . We hope that there will be no confusion with  $x_i$  which denotes the  $i$ -th component of the vector  $x$  and exists only for  $i \geq 1$ .

To learn more about  $x_o$  and for later use we split it into its nonzero components we denote  $\bar{x}_o$  and its zero components we denote  $\bar{\bar{x}}_o$ , we accordingly split the associated subgradient  $u_o$  and the  $A$  matrix, one then has, e.g.,  $Ax_o = \bar{A}_o \bar{x}_o$ . One can then decompose (3) into two parts

$$\bar{A}_o^T (\bar{A}_o \bar{x}_o - b) + h \bar{u}_o = 0, \quad \bar{\bar{A}}_o^T (\bar{\bar{A}}_o \bar{\bar{x}}_o - b) + h \bar{\bar{u}}_o = 0. \quad (4)$$

From the first part, since  $\bar{u}_o = \text{sign}(\bar{x}_o)$  is uniquely defined, one gets  $\bar{x}_o = \bar{A}_o^+ b - h (\bar{A}_o^T \bar{A}_o)^{-1} \text{sign}(\bar{x}_o)$  provided  $\bar{A}_o$  is full rank and substituting  $\bar{x}_o$  in the second part yields an explicit expression for  $\bar{\bar{u}}_o$ . At this point one can show that assuming the uniqueness of the optimum of (2) implies that  $\bar{A}_o$  is full rank and  $\|\bar{\bar{u}}_o\|_\infty < 1$ . If one of these two conditions were not satisfied one would indeed have a convex set of solutions.

### C. Iterative Algorithm

To get the unique optimum of (2) we propose to solve the equation  $A^T(Ax - b) + hu = 0$  using a *fixed-point* algorithm. The idea is to rearrange this equation in the form  $x = t(x)$  and, by successive substitution,  $x^k = t(x^{k-1})$  to build a sequence of points that converges towards the solution. To simplify the notations, we will often write  $x$  or  $x^-$  for the current point and  $x^+$  for the next point in the sequence.

Though it is as far as the optimum is concerned, let us assume that  $x$  in  $A^T(Ax - b) + hu = 0$  has no zero component, the subgradient  $u$  of  $\|x\|_1$  at  $x$  is then unique,  $u = s = \text{sign}(x)$ , where  $s$  is the vector of the signs of the components of  $x$ . This assumption is made to arrive at a implementable fixed point

algorithm and does not prevent the sequence to converge towards an optimum having many zero components, see the comments just before Section III-A. We denote  $X = \text{diag}(x)$  the square diagonal matrix built upon the components of  $x$  and  $|X| = \text{diag}(|x|)$  should not be confused with the determinant of the square matrix  $X$ , similarly  $S = \text{diag}(s)$ . One then has  $s = S\mathbf{1} = SX^{-1}X\mathbf{1} = SX^{-1}x = |X|^{-1}x$ . This allows to rewrite the equation  $A^T Ax + hs = A^T b$  as  $(A^T A + h|X|^{-1})x = A^T b$ , which leads to

$$x = (A^T A + h|X|^{-1})^{-1} A^T b \quad (5)$$

$$\begin{aligned} &= \{(A^T A|X| + hI)|X|^{-1}\}^{-1} A^T b \\ &= |X| (A^T A|X| + hI)^{-1} A^T b. \end{aligned} \quad (6)$$

One further transforms this relation by observing that  $(A^T A|X| + hI)A^T = A^T (A|X|A^T + hI)$  yields

$$A^T (A|X|A^T + hI)^{-1} = (A^T A|X| + hI)^{-1} A^T \quad (7)$$

and, using this last equality, we get:  $x = |X|A^T (A|X|A^T + hI)^{-1} b$ , and we therefore propose the following iterative algorithm:

$$\begin{aligned} x \rightarrow x^+ = t(x) &= |X|A^T (A|X|A^T + hI)^{-1} b, \\ &\text{with } |X| = \text{diag}(|x|) \end{aligned} \quad (8)$$

where  $x$  is the current point, also denoted  $x^k$ , and  $x^+$  the next point, also denoted  $x^{k+1}$ . We initialize this iteration at any point  $x^0$  that has no zero component. Note that while, the initial version (5), is ill conditioned in case  $x$  has a small component, no such difficulty arises with this last version (8), which is also preferable to (6) from a computational complexity point of view since we consider situations where  $m \gg n$ .

We obtained (8) by applying a fixed point approach to the NSC (3) but the equivalent algorithm (5) can also be obtained by applying an iterative reweighted least squares (IRLS) approach to (2), see [20]–[24]. Other iterative algorithms [27]–[30] have also been proposed for similar criteria having a different penalization term. They are usually applied to either strictly convex or convex differentiable criterion. Their convergence has been analyzed in several of these papers and is established in [32] under quite weak condition that, however, do not encompass (2). A link between these algorithms and the expectation-maximization (EM) algorithm [31] has also been made [23], [24], [33], [34] but again no convergence proof seems to be available when applied to (2). To our knowledge, the convergence of (8), when applied to (2), has never been established neither in the real case nor, a fortiori, in the complex case, that we consider in Section IV, although parts of the proof can be found in [20]–[24].

### III. CONVERGENCE ANALYSIS

In this section, we prove the following proposition.

*Proposition:* The sequence  $\{x^k\}$  generated by (8)

$$\begin{aligned} x^{k+1} &= |X^k|A^T (A|X^k|A^T + hI)^{-1} b, \\ &\text{with } |X^k| = \text{diag}(|x^k|) \end{aligned}$$

converges to a minimum point of the convex criterion (2) if no point of the sequence, including  $x^0$ , has a zero component.  $\square$

For simplicity, we assume that the minimum is attained at a unique point, but the proof can be adapted if this does not hold. The idea of the proof is the following. We first characterize the fixed points of the iteration, we then show that the sequence remains in a compact set and admits thus cluster points. We then prove that  $\|x^{k+1} - x^k\|$  converges to zero. This tells us that the cluster points are fixed points. We then show that the denumerable undesirable fixed points are all repelling.

*Comments:* The assumption that no component in the sequence  $\{x^k\}$  will be zero may seem awkward since the expected point of convergence has precisely many zero components. There is no contradiction. It is indeed perfectly possible for, say, a scalar sequence of reals to converge to zero without any of its elements to be zero, this is actually the *standard* situation. Clearly when implementing the recursion (8), a component can become *exactly* zero only by accident, i.e., with probability zero. The precise answer would consist in establishing that the set of initials points for which at least one point of the sequence has a zero component is of measure zero [35]. The answer is open. Finally, a definite way to keep any component from staying at zero, in case it has become zero by accident, consists in replacing (8) by

$$\begin{aligned} x \rightarrow x^+ = t(x) &= DA^T (ADA^T + hI)^{-1} b \quad \text{and} \\ d \rightarrow d^+ &= (1 - r)d + r|x^+| \end{aligned} \quad (9)$$

with  $d^0 = |x^0|$  and  $r \in ]0, 1]$  a relaxation parameter close to one. Its beneficial effect can be seen in the second equation that concerns only the absolute values of the components and does not prevent them from changing sign at any time and as often as desired.

#### A. Fixed Points

The fixed points of the iteration (8) satisfy  $x = t(x)$ , i.e.,  $x = |X|A^T (A|X|A^T + hI)^{-1} b$ , or, equivalently, after some rewriting

$$X (\mathbf{1} - SA^T (A|X|A^T + hI)^{-1} b) = 0. \quad (10)$$

The point  $x = 0$  is always a fixed point but there are other ones. To find another one, say  $x$ , we split it, as at the end of Section II-B, into its nonzero components  $\bar{x}$  and its zero components  $\bar{\bar{x}}$ , we accordingly split the matrix  $A$  and again one has, for instance  $Ax = \bar{A}\bar{x}$ .

The equations in (10) associated with the zero components vanish and those associated with the nonzero components become after simplification and some rearrangements

$$\bar{s} = \bar{A}^T (\bar{A}|\bar{X}|\bar{A}^T + hI)^{-1} b \quad (11)$$

using (7), this becomes

$$\begin{aligned} \bar{s} &= |\bar{X}|^{-1} (\bar{A}^T \bar{A} + h|\bar{X}|^{-1})^{-1} \bar{A}^T b \\ &= (\bar{A}^T \bar{A}|\bar{X}| + hI)^{-1} \bar{A}^T b \end{aligned}$$

which can be rewritten

$$\begin{aligned} \bar{A}^T \bar{A} |\bar{X}| \bar{s} + h \bar{s} &= \bar{A}^T b \\ \Leftrightarrow \bar{A}^T \bar{A} \bar{x} + h \bar{s} &= \bar{A}^T b \\ \Leftrightarrow \bar{A}^T (\bar{A} \bar{x} - b) &= h \bar{s}. \end{aligned} \quad (12)$$

In this last relation, one recognizes the NSC (3), (4) restrained to the set of equations associated with  $\bar{A}$  and  $\bar{s}$ . This means that, for the splitting associated with  $x_o$ , (11) or equivalently (12) admit the solution  $x = x_o$ . This establishes that the sought-for point  $x_o$ , is a fixed point of (8).

But there are many other ones as well. Indeed, remove one or several columns from  $A$ , and consider the new associated optimization problem. With its optimum is then associated a splitting of the new as well as of the initial problem that will lead to an equation like (11) or equivalently (12) that admits a solution with which one easily associates a fixed point of the initial problem.

There are thus quite a large, but denumerable, number of fixed points, except in the case  $h \geq \|A^T b\|_\infty$  for which the unique optimum is at the origin, and the above trick always yields this same point.

In summary, for  $h \geq \|A^T b\|_\infty$  there is a unique fixed point  $x = 0$  which is the optimum of (2) that is then independent of  $h$ . For  $h < \|A^T b\|_\infty$ , there are many fixed points in addition to the origin and  $x_o$ , the unique minimum of (2). Their number and locations of their zero components then depends upon  $h$ .

### B. Descent Algorithm

Denoting  $f(x)$  the criterion in (2), we prove that  $f$  decreases steadily along the sequence generated by (8), though we did not explicitly built it with this goal. Let us evaluate the residual  $r^+ = b - Ax^+$

$$\begin{aligned} r^+ &= \left[ I - A|X|A^T (A|X|A^T + hI)^{-1} \right] b \\ &= h (A|X|A^T + hI)^{-1} b \end{aligned}$$

substituting in (8), the iteration becomes

$$x^+ = |X|A^T r^+ / h = -|X|g^+ / h$$

where  $g = A^T(Ax - b) = -A^T r$  is the gradient of the quadratic part of (2). This is an implicit expression of little interest as such that we use below. To evaluate  $\Delta f = f(x^-) - f(x^+)$ , we replace the quadratic part of  $f$  by its second order Taylor expansion around  $x^+$  and add to it, the variation of  $h\|x\|_1 = hs^T x$  to get

$$\begin{aligned} \Delta f &= \frac{1}{2} (x^- - x^+)^T A^T A (x^- - x^+) + (g^+)^T (x^- - x^+) \\ &\quad + h (s^{-T} x^- - s^{+T} x^+). \end{aligned}$$

To establish that  $\Delta f \geq 0$ , we observe that the first and quadratic term is always  $\geq 0$  and we concentrate on the two last and linear terms, which we detail componentwise. One has  $\Delta f \geq \sum_{i=1}^m \Delta_i$  with

$$\Delta_i = g_i^+ (x_i^- - x_i^+) + h (s_i^- x_i^- - s_i^+ x_i^+).$$

We now take a close look at the  $i^{\text{th}}$  component  $x_i^-$  and  $x_i^+$  and the variation  $\Delta_i$ . We distinguish two cases  $x_i^- x_i^+ > 0$  and  $x_i^- x_i^+ < 0$ . We will use the relation  $|x_i^-| g_i^+ = -h x_i^+$  established above, then in the first case, for identical signs denoted  $s_i$ , one has

$$\begin{aligned} \Delta_i &= (g_i^+ + h s_i) (x_i^- - x_i^+) \\ &= \left( -\frac{h x_i^+ s_i}{x_i^-} + h s_i \right) (x_i^- - x_i^+) \\ &= \frac{h}{|x_i^-|} (x_i^- - x_i^+)^2 \end{aligned}$$

and, if  $x_i^- x_i^+ < 0$ , replacing  $s_i^+$  by  $-s_i^-$  one has, omitting the details

$$\begin{aligned} \Delta_i &= g_i^+ (x_i^- - x_i^+) + h s_i^- (x_i^- + x_i^+) \\ &= \frac{h s_i^-}{x_i^-} (x_i^{+2} + x_i^{-2}) \geq h |x_i^-|. \end{aligned}$$

Since both expressions are positive, this establishes that  $\Delta f$ , i.e.,  $\Delta f^k = f(x^{k-1}) - f(x^k) \geq 0$  for all  $k$  and, thus, that both  $\Delta f^k$  and  $\Delta_i^k$  converge towards 0 as  $k \rightarrow \infty$ , since  $\sum_k \Delta f^k$  is upperbounded.

Note that the monotonicity, established in this section, follows immediately when (5) or (8) is interpreted as a EM algorithm [23], [24], [33], [34], but since part of the results and notations introduced are used in Section III-C, we prefer to establish it from scratch, making, furthermore, the overall proof self-contained.

### C. Convergence Towards a Fixed Vector

Since  $f$  is convex, the fact that  $f$  decreases along the sequence  $x^k$  shows that the sequence  $\{x^k\}$  remains in a compact subset and has one or several cluster points.

We now establish that  $|x_i^- - x_i^+|$  converges towards 0, which in turn will imply that  $\|x^{k-1} - x^k\| \rightarrow 0$ . We concentrate on the  $i$ -th component of  $x^k$  and the associated scalar sequence  $\{x_i^k\}$ . If  $x_i^- x_i^+ > 0$ , we have

$$\Delta_i = \frac{h}{|x_i^-|} (x_i^- - x_i^+)^2 > (h/M) (x_i^+ - x_i^-)^2$$

with  $M = \max_k \|x^k\|_\infty$ , which exists since the sequence  $x^k$  is bounded. If  $x_i^- x_i^+ < 0$ , we have  $\Delta_i \geq h |x_i^-|$ .

Since  $\Delta_i$ , i.e.,  $\Delta_i^k$  converges towards zero, it follows that either  $x_i^k$  or  $|x_i^{k-1} - x_i^k|$  converges towards zero, depending upon whether the sign of  $x_i^k$  changes infinitely often or not. In both cases,  $|x_i^{k-1} - x_i^k|$  converges towards 0 which implies  $\|x^{k-1} - x^k\| \rightarrow 0$ , the result we wanted to establish.

This in turn implies that all the cluster points are fixed points and that  $x^k$  converges to one of them. Indeed if  $\{x^{k_n}\}$  is the subsequence that converges towards the cluster point  $x^{*n}$ , then  $\{t(x^{k_n})\}$  converges towards  $t(x^{*n})$ , but since  $\|x^{k_n} - x^{k_n+1}\| \rightarrow 0$ , one has  $x^{*n} = t(x^{*n})$  which shows that all the cluster points are fixed points. But since there are a finite number of fixed points, for  $k$  large enough, it implies that the sequence remains in the vicinity of one of them since  $\|x^{k+1} - x^k\| = \|t(x^k) - x^k\| \rightarrow 0$ , and thus converges towards it.

#### D. The Optimum is the Only Attractive Fixed Point

For  $\|A^T b\|_\infty < h$ , there is a unique fixed point at the origin and there is thus nothing to prove. The algorithm converges towards it. It is the case  $\|A^T b\|_\infty > h$  that needs investigations. We will establish that all fixed points, except the minimum of (2), are locally repelling. Remember, that we assume that no component in  $x^k$  is equal to zero for any  $k$ .

Let us start with the origin, that needs a special treatment. Let  $c > 0$  be such that  $(\|A^T b\|_\infty/h) = 1 + c > 1$  and denote  $i$  the component for which equality is reached. We consider (8) rewritten as

$$x^+ = |X|A^T \left( I + \frac{1}{h}A|X|A^T \right)^{-1} \frac{1}{h}b$$

and observe that, if it were  $x^+ = |X|A^T(1/h)b$ , then, since it could be rewritten  $x^+ = \text{diag}((1/h)SA^Tb)x$ , the origin would be unstable, since the  $i^{\text{th}}$  component  $x_i^k$  of  $x^k$  would diverge as  $(1+c)^k$ .

But, since in the neighborhood of the origin, the positive definite matrix  $(I + (1/h)A|X|A^T)^{-1}$  can be made as close to  $I$  as desired (in any norm) by reducing the size of the neighborhood, the above proof remains valid. One can, for instance, upper-bound the maximum of the absolute values of the components of  $I - (I + (1/h)A|X|A^T)^{-1}$  over all  $x$  in a ball around the origin and monitor this upper-bound. The origin can thus not be the fixed point towards which the algorithm converges, since as soon as  $x^k$  enters this ball, the  $i^{\text{th}}$  component increases and the sequence moves out of the ball.

Now we consider one of the fixed points, we denote  $x_f$ , different of the sought-for optimum  $x_o$  of (2) associated with the plain  $A$  matrix, the demonstration will be similar. This fixed point, just as  $x_o$ , has many zero components. Remember that  $x_f$  is deduced from the optimum of (2) in which a set of columns has been removed from the  $A$ -matrix, by adding zero components to this optimum at precisely the indices of the removed columns. We split  $x_f$  into  $\bar{x}_f$  whose components are all different from zero, and  $\bar{\bar{x}}_f = 0$ , we split accordingly  $A$  into  $\bar{A}_f$  and  $\bar{\bar{A}}_f$ . First observe that since  $x_f$  is a fixed point, for its components in  $\bar{x}_f$ , one has  $\bar{x}_f = |\bar{X}_f|\bar{A}_f^T(hI + \bar{A}_f|\bar{X}_f|\bar{A}_f^T)^{-1}b$ , which yields

$$b - \bar{A}_f\bar{x}_f = h(hI + \bar{A}_f|\bar{X}_f|\bar{A}_f^T)^{-1}b. \quad (13)$$

Now since  $\bar{x}_f$  corresponds to the nonzero components of the optimum of (2) with some columns deleted in  $A$ , one has from (4)

$$\bar{A}_f^T(b - \bar{A}_f\bar{x}_f) = h\bar{u}_f = h\text{sign}(\bar{x}_f) \quad (14)$$

and if one defines the vector  $\bar{\bar{\mu}}_f$  by  $\bar{A}_f^T(b - \bar{A}_f\bar{x}_f) = h\bar{\bar{\mu}}_f$ , it has at least a component greater than 1 in absolute value, since otherwise  $x_f$  would be the optimum of the original (2). Remember that the columns, that were deleted from  $A$ , are present in  $\bar{\bar{A}}_f$ . For later use, notice that combining this relation with (13), one has

$$\bar{\bar{A}}_f^T(hI + \bar{A}_f|\bar{X}_f|\bar{A}_f^T)^{-1}b = \bar{\bar{\mu}}_f. \quad (15)$$

We now take a point  $x^k$  in a ball centered at  $x_f$  but with no component equal to zero and prove that the sequence originated at  $x^k$  cannot converge towards  $x_f$ . The associated  $x^{k+1} = |X^k|A^T(hI + A|X^k|A^T)^{-1}b$  can be made arbitrarily close (by reducing the size of the ball) to say  $y^{k+1}$  given by

$$\begin{aligned} y^{k+1} &= |X^k|A^T(hI + A|X_f|A^T)^{-1}b \\ &= |X^k|A^T(hI + \bar{A}_f|\bar{X}_f|\bar{A}_f^T)^{-1}b \end{aligned}$$

which splitting this last relation into its two  $x_f$ -induced parts and, using (13), (14), and (15), becomes

$$\bar{y}_f^{k+1} = |\bar{X}_f^k|\bar{u}_f \quad \text{and} \quad \bar{\bar{y}}_f^{k+1} = |\bar{\bar{X}}_f^k|\bar{\bar{\mu}}_f.$$

Now since  $\bar{\bar{\mu}}_f$  has at least a component greater than one in absolute value, the corresponding component in  $\bar{\bar{y}}_f^{k+1}$  (which is close to zero) increases and the same holds true for the corresponding component in the true  $x^{k+1}$ . As soon as  $x^k$  enters the so-defined ball centered at  $x_f$ , at least one of its components (which is close to zero) increases in absolute value, and the sequence moves out of the ball.

The minimum of (2) is thus the only fixed point that is attractive and since  $x^k$  converges to a fixed point, this completes the proof.  $\square$

*Remark 1:* In practice, the iterations are stopped when a stopping criterion is satisfied and one will then set to zero all the components whose absolute value is smaller than a threshold function of the stopping criterion.

*Remark 2:* The inversion present in (8) can become time consuming if  $\min(m, n)$  is large but it is possible to implement it in an efficient way, [34].

#### IV. EXTENSION TO THE COMPLEX CASE

We now indicate how to extend the criterion, the algorithm and convergence proof to the complex case. One wants to solve, with respect to  $z \in C^m$

$$\min_z \frac{1}{2}\|Az - b\|_2^2 + h\|z\|_1, \quad h > 0 \quad (16)$$

where all the quantities may be complex, except  $h$ , which is a positive real number to be tuned by the user. We define  $\|z\|_2^2 = z^*z$  with  $z^*$  the complex conjugate transpose of  $z$  and  $\|z\|_1 = \sum_k |z_k|$  with  $|z_k|$  the modulus of the  $k^{\text{th}}$  component of  $z$ . We only detail the few parts that are significantly different from their real counterparts.

##### A. Optimality Conditions

Since the complex case is seldom considered, we establish the optimality conditions and summarize them in a Lemma.

*Lemma:*  $z$  is a minimum of (16) if and only if there exists a complex vector  $u$ , a *subgradient* of  $\|z\|_1$

$$\begin{aligned} \text{satisfying} \quad u_k &= \frac{z_k}{|z_k|}, \quad \text{if } k \in \{k|z_k \neq 0\} \\ \text{and} \quad |u_k| &\leq 1, \quad \text{if } k \in \{k|z_k = 0\}, \\ \text{such that} \quad A^*(Az - b) + hu &= 0. \end{aligned} \quad (17)$$

*Proof:* Denote  $f(z)$  the function (16) to be minimized; it is a function from  $C^m$  to  $R^+$ . Since  $f$  is convex, it is sufficient that the first order variation in  $f(z + \delta z)$  is  $\geq 0$ , at the optimal  $z$ . One first verifies that, for  $k \in \{k | z_k \neq 0\}$ , also denoted  $k \in act$

$$|z_k + \delta z_k| \simeq |z_k| + \frac{\Re(z_k^* \delta z_k)}{|z_k|}$$

where  $\Re(z)$  denotes the real part of  $z$ , while for  $k \in \{k | z_k = 0\}$ , also denoted  $k \in inact$ ,  $|z_k + \delta z_k| = |\delta z_k|$ . We now define  $v = A^*(Az - b)$ . One then has successively

$$\begin{aligned} f(z + \delta z) &\simeq f(z) + \frac{1}{2}(\delta z^* v + v^* \delta z) + h \sum_{k \in act} \frac{\Re(z_k^* \delta z_k)}{|z_k|} \\ &\quad + h \sum_{k \in inact} |\delta z_k| \\ &\simeq f(z) + \Re(v^* \delta z) + h \sum_{k \in act} \frac{\Re(z_k^* \delta z_k)}{|z_k|} \\ &\quad + h \sum_{k \in inact} |\delta z_k| \end{aligned}$$

distributing now the term  $\Re(v^* \delta z)$  in the two sums, and replacing  $v$  by  $-hu$  [see (17)], one gets

$$\begin{aligned} f(z + \delta z) &\simeq f(z) + h \sum_{k \in act} \left( -\frac{\Re(z_k^* \delta z_k)}{|z_k|} + \frac{\Re(z_k^* \delta z_k)}{|z_k|} \right) \\ &\quad + h \sum_{k \in inact} (-\Re(u_k^* \delta z_k) + |\delta z_k|) \\ &\simeq f(z) + h \sum_{k \in inact} [1 - |u_k| \cos(u_k, \delta z_k)] |\delta z_k|. \end{aligned}$$

This establishes the result, since these last terms are always greater or equal to zero because  $|u_k| \leq 1$  and hence:  $1 - |u_k| \cos(\dots) \geq 0$ .  $\square$

The optimality conditions (17) are thus identical to (3) but with the subgradient  $u$  of  $\|z\|_1$  now a complex vector with components of modulus smaller or equal to one for the zero components in  $z$  and equal to  $z_k/|z_k|$  for the nonzero components. It follows that the fixed-point algorithm (8) and the Proposition in Section III remain formally the same. One simply has to replace the absolute values by the modulus in (8). We will denote  $\{z^k\}$  the sequence of complex vectors generated by the fixed point algorithm.

Most parts in the convergence analysis remain similar and are easy to transpose. The main difference appears in the equivalent of Section III-C.

### B. Convergence Towards a Fixed Vector

It seems impossible to prove that  $\|z^{k-1} - z^k\| \rightarrow 0$ , and one has to resort to the sequence of magnitude vectors  $\{m^k\}$  with  $m^k = |z^k|$  to handle this difficulty. It admits similarly cluster points and can be seen as generated by  $m^{k+1} = |t(m^k)|$ . Moreover, since  $t(z) = t(m)$ , (8), one can associate with each fixed point, say,  $\tilde{z}$  of  $t(\cdot)$  satisfying  $\tilde{z} = t(\tilde{z})$  a fixed point  $\tilde{m} = |\tilde{z}|$  of  $m = |t(m)|$ :

$$\tilde{z} = t(\tilde{z}) = t(\tilde{m}) \Rightarrow \tilde{m} = |t(\tilde{m})| = |\tilde{z}|.$$

One then establishes several properties of  $\{m^k\}$ . One first proves that  $\|m^{k-1} - m^k\| \rightarrow 0$ . One then establishes that the cluster points of  $\{m_k\}$  are fixed points of  $m = |t(m)|$  and eventually that  $\{m^k\}$  converges towards one of its fixed points. All these proofs are quite similar to the one proposed in Section III-C. It finally remains to observe that this last property is transferable to  $\{z^k\}$ . Indeed, since the fixed points (or convex sets of fixed points) differ by the indices of their zero components, if  $m_k \rightarrow m^*$  then  $z_k \rightarrow z^*$  with  $z^* = t(m^*)$ . The sequel of the proof then follows the same path as in Section III-D. It is thus possible to extend the proof detailed in the real case, to establish that the Proposition given at the beginning of Section III also holds for complex data.  $\square$

*Remark 3:* Little attention has been devoted, so far, to the complex case, though there are potential applications in array processing and spectral analysis, for instance. Most algorithms dedicated to a fast resolution of (2) cannot be extended to handle the complex case. The optimization of (16) can be performed using standard descent algorithms or second-order cone programming (SOCP) techniques [36], [37], a particular type of the more general convex cone programming approaches. In the transcription to SOCP, the number of unknowns increases however drastically and the computational complexity is quite high. These techniques cannot, moreover, take advantage of prior knowledge on the optimum and implementing (8) may thus be quite interesting.

## V. SIMULATION RESULTS

To illustrate the performance of the proposed iterative algorithm (8), we present the results on some easily reproducible toy examples, which are, however, of some interest in practice. We consider the often considered example where one observes  $n$  samples of a sum of  $P$  real sinusoids in additive white Gaussian noise and one wants to recover the number  $P$  of sinusoids, and for each of them, amplitude, initial phase and frequency. To treat this problem, using the sparse representation approach in the real case is quite challenging, because the initial phases are difficult to handle. This difficulty, however, disappears when switching to the complex case, since the signal is then somehow *linear* with respect to the initial phases:  $a \cos(\omega t + \varphi) = \alpha e^{j\omega t} + \alpha^* e^{-j\omega t}$  with  $\alpha = (a/2)e^{j\varphi}$ . Though the  $n$ -dimensional observation vector  $b$  will be real, we take a complex  $n \times m$  matrix  $A$ , with  $m = 2m_f \gg n$ . The  $p^{\text{th}}$  column of  $A$ ,  $p \in (1, m_f)$ , has components equal to  $e^{2i\pi f_p k}$  with  $f_p = p/(2m_f)$  and  $k \in (0, n-1)$ . The remaining  $m_f$  columns are the complex conjugate of these first  $m_f$  columns. All these columns are normalized to one in Euclidean norm, i.e., divided by  $\sqrt{n}$ , as is recommended when using (2) or (16), [9].

With this complex  $A$  matrix, the idea is to recover each individual real sinusoid using just two complex conjugate columns, if its frequency is a multiple of  $(1/2m_f)$ , or using essentially two pairs of conjugate columns, otherwise. The optimum  $z_o$  of (16) should thus have between  $2P$  and  $4P$  nonzero components, from which one deduces both an estimate of  $P$ , and of the amplitudes and initial phases, the frequencies being encoded in the indices of the nonzero components. The  $h$  parameter in (16) allows to somehow fix a threshold on the reconstruction error, that is allowed. In the absence of noise on the observations, it can be

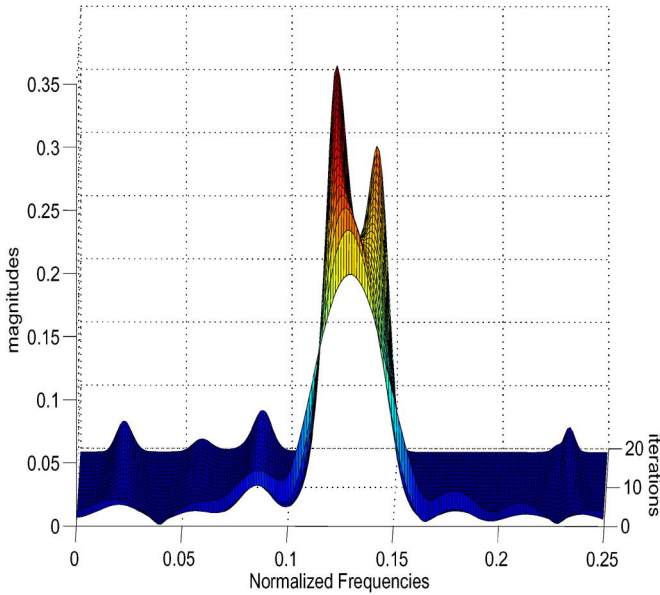


Fig. 1. Two closely spaced sinusoids: magnitudes of the 200 first weights over the 20 first iterations.

taken quite small and essentially a function of  $m_f$ , i.e., of the discretization step in frequency. In the presence of noise, the value of  $h$  will mainly depend upon  $m$  and the expected standard deviation of the noise [9].

#### A. A Basic Time Invariant Case

We consider a case where the difficulty is the resolution of two closely spaced sinusoids. We take  $n = 40$ ,  $m = 800$  and simulate  $P = 2$  sinusoids with common amplitude equal to one, common nonzero initial phase (to fix ideas) and additive white Gaussian noise with standard deviation .22, hence a 10-dB signal-to-noise ratio. The normalized frequencies are  $f_1 = .125$  and  $f_2 = .125 + (1/2n)$ . These sinusoids can thus only be separated by a high resolution technique. We present, in Fig. 1, the results for a typical noise realization. We plot the modulus of the first 200 components of  $z$  for the first 20 iterations, when the algorithm is initialized at  $z^0 = A^+b$ . The parameter  $h$  is taken equal to .1 but the sensitivity of the output, with respect to  $h$ , is quite low. The method separates the two sinusoids after a few steps, while the basic periodogram, close to  $z^0$ , does not separate them.

In case of regular sampling, there are of course many competing methods, but this approach can handle irregular sampling without additional complexity, and there, the competition is less severe.

#### B. A Time Varying Case

The  $A$  matrix is as above, but the observation vector  $b$  is time varying, it is modified at each iteration of the algorithm (8). It is the sum of three sinusoids of unit amplitude in additive white Gaussian noise with standard deviation .22. The first sinusoid remains constant over 15 iterations, disappears and emerges again at iteration 25 at a different frequency. The second remains constant and is present over the 40 iterations. The third has its frequency that increases linearly. The magnitudes of the first half

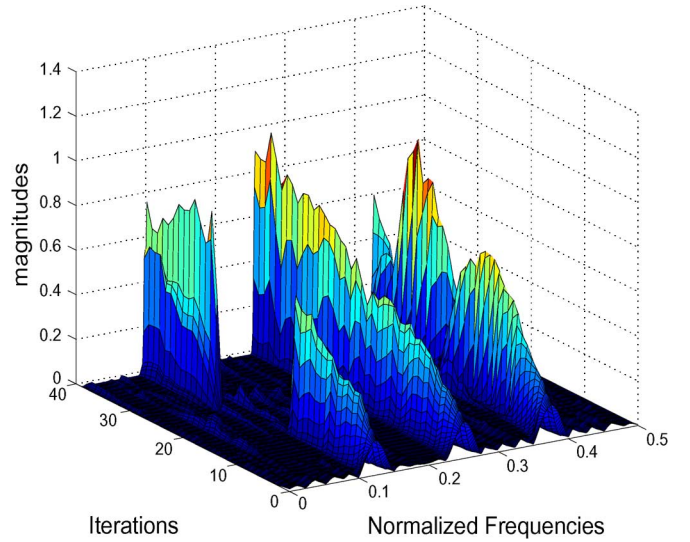


Fig. 2. Time-varying scenario with three sinusoids: magnitudes of the 400 first weights over 40 iterations.

of the  $z$  vector is presented in Fig. 2, for the different iterations. The algorithm tracks the time-varying frequency and adapts to the new emerging sinusoid without any difficulty. The fluctuations that are observed in Fig. 2 are due to the fact that the noise vector is also changed at each iteration. For completeness, let us specify that we take  $b^k = \cos((2\pi f_1)[0 : n - 1]) + \cos(.5\pi[0 : n - 1] + \varphi_2) + \cos((2\pi(.375 + .0012k)[0 : n - 1] + \varphi_3) + .22e^k$  where  $f_1$  is first equal to .125 and later to .10, and  $e^k$  is a standard Gaussian noise vector.

These two simulation results do indeed convey little information, they are just meant to describe the general behavior of the algorithm. Let us comment them briefly. Due to the presence of noise, one observes that there are indeed many nonzero components, in  $z^k$  at each iteration, their number can be reduced by increasing  $h$ . The iteration (8) has no real inherent tracking properties, these can be induced and enhanced by preventing the different components to become too small in modulus. A standard trick consists in replacing the small components in  $z^k$  or  $x^k$  by  $\epsilon = \pm 10^{-4}$ , for instance, or by intervening similarly on  $d$  in (9). Introducing Huber's function in place of the  $\ell_1$  penalization in (2), as proposed in [30], can also be considered. Here, however, no such modifications were implemented.

## VI. CONCLUDING REMARKS

The major contribution of the paper is the self-contained proof of the convergence of (8) towards the minimum of (2) and its extension to the case where the data are complex. The criterion (8) is one of the most frequently used to get sparse representations. While, in the real case, it can be transformed into a quadratic program, and many fast dedicated algorithms have been proposed to solve it, no such possibility exists in the complex case and only standard descent algorithms or second order cone programming approaches have been proposed in that case.

Since a major concern in the sparse representation community is to get faster implementations able to work on real time applications in which, quite systematically, the goal is to track

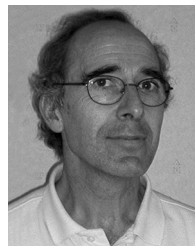
slowly time-varying sparse representations, there is obviously a need for algorithms able to take advantage of this feature. This is the case of the algorithm we analyze, as opposed to fast solvers of this same criterion which by their very nature (they add one component at a time and never erase an existing component) cannot be made adaptive. Modifying the analyzed algorithm, to enhance its tracking potentialities, is the object of current investigations.

#### ACKNOWLEDGMENT

The author thanks B. Delyon (Univ. de Rennes 1) and J. Idier (Ircycn, Nantes) for helpful discussions.

#### REFERENCES

- [1] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Info. Technol.*, vol. 47, no. 11, pp. 2845–2862, Nov. 2001.
- [2] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Info. Technol.*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.
- [3] J. J. Fuchs, "More on sparse representations in arbitrary bases," *IEEE Trans. Info. Technol.*, vol. 50, no. 6, pp. 1341–1344, Jun. 2004.
- [4] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximations," *IEEE Trans. Info. Technol.*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
- [5] M. Bertalmio *et al.*, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.
- [6] M. Zibulewsky and B. Pearlmutter, "Blind source separation by sparse decomposition on a signal dictionary," *Neural Comput.*, vol. 13, pp. 863–882, 2001.
- [7] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [8] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.
- [9] J. J. Fuchs, "On the application of the global matched filter to DOA estimation with uniform circular arrays," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 702–709, Apr. 2001.
- [10] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, Jan. 2003.
- [11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annal. Statist.*, vol. 32, pp. 407–499, Apr. 2004.
- [12] J. J. Fuchs, "Recovery of exact sparse representations in the presence of noise," in *Proc. IEEE ICASSP Conf.*, Montreal, QC, Canada, May 2004, vol. II, pp. 533–536.
- [13] S. Krstulovic and R. Gribonval, "MPTK: Matching pursuit made tractable," in *Proc. IEEE ICASSP Conf.*, Toulouse, France, May 2006.
- [14] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Inform. Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
- [15] I. Daubechies, M. Defriese, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. LVII, pp. 1413–1457, 2004.
- [16] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Technol.*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [17] E. Candes and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Found. Comput. Math.*, vol. 6, pp. 227–254, 2006.
- [18] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Technol.*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [19] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal coding strategies?," *IEEE Trans. Inform. Technol.*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [20] I. F. Gorodnitski and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A reweighted minimum-norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
- [21] M. D. Sacchi, T. J. Ulrych, and C. J. Walker, "Interpolation and extrapolation using a high-resolution discrete Fourier transform," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 31–38, Jan. 1998.
- [22] B. D. Rao *et al.*, "Subset selection in noise based on diversity measure minimization," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 760–770, Mar. 2003.
- [23] M. Davies and L. Daudet, "Sparsifying subband decompositions," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Pfalz, NY, Oct. 2003.
- [24] M. Figueiredo, "Adaptive sparseness for adaptive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.
- [25] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1987.
- [26] J. J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise," *IEEE Trans. Info. Technol.*, vol. 51, no. 10, pp. 3601–3608, Oct. 2005.
- [27] S. Alliney and S. A. Ruzinsky, "An algorithm for the minimization of mixed  $\ell_1$  and  $\ell_2$  norms with application to Bayesian estimation," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 618–627, Mar. 1994.
- [28] M. E. Davies and L. Daudet, "Sparse audio-representations using the MCLT," *EURASIP Signal Process. — Special Issue on Sparse Representations*, May 2005, submitted for publication.
- [29] R. H. Byrd and D. A. Payne, Convergence of the IRLS Algorithm for Robust Regression John Hopkins Univ., Baltimore, MD, Tech. Rep. 313, Jun. 1979.
- [30] R. Yarlagadda, J. B. Bednar, and T. L. Watt, "Fast algorithms for  $\ell_p$  deconvolution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 174–182, Feb. 1985.
- [31] C. F. J. Wu, "On the convergence of the EM algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 1983.
- [32] J. Idier, "Convex half-quadratic criteria and interactive auxiliary variables for image restoration," *IEEE Trans. Inf. Process.*, vol. 10, no. 7, pp. 1001–1009, Jul. 2001.
- [33] F. Champagne and J. Idier, "A connection between half-quadratic criteria and EM algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 9, pp. 709–712, Sep. 2004.
- [34] J. Bioucas-Dias, "Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors," *IEEE Trans. Inform. Technol.*, vol. 15, pp. 937–951, 2006.
- [35] L. Canavate, R. Canavate, and A. Marin, "On the convergence of the Weiszfeld algorithm," *Math. Program.*, vol. A93, pp. 327–330, 2002.
- [36] D. Malioutov, M. Cetin, and A. Willsky, "A sparse reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.
- [37] A. Ben Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia, PA: Philadelphia SIAM MPS, 2001.



**Jean Jacques Fuchs** (M'81) was born in France in 1950. He graduated from the "École Supérieure d'Électricité", Paris, in 1973 and received the M.S. degree in electrical engineering from Massachusetts Institute of Technology, Cambridge, in 1974 and the "Thèse d'Etat" in adaptive control and identification in 1982.

After a short period in industry with Thomson-C.S.F., he joined the "Institut de Recherche en Informatique et Systèmes Aléatoires" (IRISA) in 1976. Since 1983, he has been a

Professor at the "Université de Rennes 1". His research interests have shifted from adaptive control and identification towards signal processing. He is now involved in array processing and sparse representations.