

# Sparse Signal Reconstruction with Hierarchical Decomposition

Ming Zhong\*

Advisor: Dr. Eitan Tadmor<sup>†</sup>

*AMSC and CSCAMM*

University of Maryland College Park  
College Park, Maryland 20742 USA

November 8, 2012

## Abstract

In this project, we develop a *Hierarchical Decomposition* algorithm to solve the  $\ell_1$ -Regularized Least Square problem:  $\min_{\mathbf{x} \in \mathbf{R}^n} \{ \|\mathbf{x}\|_{\ell_1} \mid A\mathbf{x} = \mathbf{b} \}$ . With the new approach, we show a systemic approach on how to select a family of regularization parameters  $\lambda$ 's in order to improve accuracy while retaining the sparsity of our approximation.

## 1 Background: A Constrained $\ell_0$ Minimization

In the recent decades, there have been a considerable amount of interests put into solving a minimization problem originated in Compressed Sensing. The problem is asking for the possibility of which is to ask for effective and efficient approaches to encode a large and sparse signal (reconstruct) with a relatively small number of linear measurements (acquire). Mathematically speaking, we are looking for a solution of the following minimization problem:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ \|\mathbf{x}\|_{\ell_0} \mid A\mathbf{x} = \mathbf{b} \} \quad (1)$$

Where the  $\|\cdot\|_{\ell_0}$  measures the number of non-zero element in a vector  $\mathbf{x}$ ,  $A \in \mathbf{R}^{m \times n}$  is a  $m \times n$  matrix over real and a measurement vector  $\mathbf{b} \in \mathbf{R}^m$ , with  $m \ll n$ . Two possible scenarios exist for choosing the matrix  $A$ : it is either prescribed by a specific transformation or chosen by the user to recover  $\mathbf{x}$  using least amount of information possible. The system  $A\mathbf{x} = \mathbf{b}$  is under-determined; when  $A$  has full rank, an infinite number of solutions exists. One can then find a  $\mathbf{x}$  with the minimal  $\ell_0$ -norm. However, it is shown in [10] that (1) is NP hard and requires techniques

---

\*mzhong1@umd.edu

<sup>†</sup>tadmor@cscamm.umd.edu

from combinatorial optimization. Hence it is numerically appropriate to consider a relaxed (1) by using either the  $\ell_1$  or  $\ell_2$  norm; then (1) is changed into a convex optimization problem, making it possible for us to employ convex optimization techniques (namely linear programming) to recover the solution.

## 1.1 A Constrained $\ell_2$ Minimization

We will be using  $\|\cdot\|_p$  instead of  $\|\cdot\|_{\ell_p}$  from now on. Let us consider the following Constrained  $\ell_2$  minimization:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{\|\mathbf{x}\|_2 \mid A\mathbf{x} = \mathbf{b}\} \quad (2)$$

It is also called a minimum norm problem. An analytic solution is known as  $\mathbf{x}_0 = A^*(AA^*)^{-1}\mathbf{b}$ <sup>1</sup>, where  $A^*$  is the Hermitian transpose of  $A$  (apparently  $A\mathbf{x}_0 = AA^*(AA^*)^{-1}\mathbf{b} = \mathbf{b}$ ). Let  $\mathbf{x}$  be another solution to  $A\mathbf{x} = \mathbf{b}$ , then we want to show  $\mathbf{x} - \mathbf{x}_0$  is perpendicular to  $\mathbf{x}_0$ . To Justify our claim, observe the following:

$$\begin{aligned} \langle \mathbf{x} - \mathbf{x}_0, \mathbf{x}_0 \rangle_2 &= \langle \mathbf{x} - \mathbf{x}_0, A^*(AA^*)^{-1}\mathbf{b} \rangle_2 = \langle A(\mathbf{x} - \mathbf{x}_0), (AA^*)^{-1}\mathbf{b} \rangle_2 \\ &= \langle \mathbf{b} - \mathbf{b}, (AA^*)^{-1}\mathbf{b} \rangle_2 = 0 \end{aligned}$$

We basically show that  $\forall \mathbf{z} \in N(A)$ <sup>2</sup>,  $\mathbf{z}$  is perpendicular to  $\mathbf{x}_0$ . Hence by the Pythagorean Theorem in  $\mathbf{R}^n$ :

$$\|\mathbf{x}\|_2^2 = \|\mathbf{x} - \mathbf{x}_0 + \mathbf{x}_0\|_2^2 = \|\mathbf{x} - \mathbf{x}_0\|_2^2 + \|\mathbf{x}_0\|_2^2 \geq \|\mathbf{x}_0\|_2^2$$

Therefore  $\|\mathbf{x}\|_2 \geq \|\mathbf{x}_0\|_2$ , with equality realized only when  $\mathbf{x} = \mathbf{x}_0$ .  $\mathbf{x}_0$  is indeed the minimal solution. Unfortunately, in most cases, the solution from (2) is not going to be sparse, the non-zero elements fill up the whole vector  $\mathbf{x}_0$ <sup>3</sup>.

## 1.2 A Constrained $\ell_1$ Minimization

With solutions from the  $\ell_2$  norm not satisfying the sparsity requirement, we'd like switch our focus to the  $\ell_1$  norm instead. And in fact, it is widely accepted that restricting the  $\ell_1$  norm generally produces sparse solutions. Let us consider the following:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{\|\mathbf{x}\|_1 \mid A\mathbf{x} = \mathbf{b}\} \quad (3)$$

It is shown in [2, 3] that under the Robust Uncertainty Principles: when  $A$  has the Restricted Isometry Property and  $m = O(n^{\frac{1}{4}}[\log(n)]^{\frac{5}{2}})$ , one can recover the solution from (1) by finding the minimizer from (3).

**Remark 1.1.** *Using straightforward calculus of variation, one can derive a non-linear equation from (9), which is:*

$$\text{sgn}(\mathbf{x}) + \lambda(A^*A\mathbf{x} - A^*\mathbf{b}) = 0 \quad (4)$$

<sup>1</sup>Assuming  $A$  has full rank, otherwise a SVD technique is used to solve (2)

<sup>2</sup> $N(A) = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} = \mathbf{0}\}$

<sup>3</sup>Quotation needed!!

Where the  $\text{sgn}(a) = \begin{cases} 1, & a > 0 \\ -1, & a < 0 \end{cases}$  is defined for scalars; so for  $\text{sgn}(\mathbf{x})$  on vectors, we just take  $\text{sgn}(\cdot)$  on each component of  $\mathbf{x}$ ; however,  $\text{sgn}(a)$  is not defined at  $a = 0$ . Such delicacy makes (4) hard to solve directly; furthermore, the solution set of (4) is contained in the solution set of (9) due to nondifferentiable property of  $\text{sgn}(\cdot)$  at 0.

However the  $\ell_1$  problem is not well-posed. Let us begin the argument by considering a more general family of problems:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{J(\mathbf{x}) | A\mathbf{x} = \mathbf{b}\} \quad (5)$$

Where  $J(\cdot)$  is a energy functional defined on  $\mathbf{R}^n$ . Moreover  $J(\cdot)$  is continuous and convex. Let  $J(\mathbf{x})$  also be coercive, i.e., whenever  $\|\mathbf{x}\|_2 \rightarrow \infty$ ,  $J(\mathbf{x}) \rightarrow \infty$ . As it is stated in [12]<sup>4</sup>, the solution set of (5) is nonempty and convex. And if  $J(\mathbf{x})$  is strictly or strongly convex, the solution is unique. However  $J(\mathbf{x}) = \|\mathbf{x}\|_1$  is not strictly convex, the solution is not going to be unique, thus making the problem ill-posed.

### 1.3 Tikhonov Regularization

In order to efficiently solve ill-posed problems, a regularization type of method, called Tikhonov Regularization, was proposed and developed ([16, 17]). Such regularization method can be applied to a broader class of problems. Similar to the discussion in [13, 14], let us consider two topological spaces  $U$  and  $W$  and their corresponding norms  $\|\cdot\|_U$  and  $\|\cdot\|_W$ . And we intend to analyze the following regularization problem:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \left\{ \|\mathbf{x}\|_U + \frac{\lambda}{2} \|\mathbf{b} - A\mathbf{x}\|_W^q \right\} \quad (6)$$

It is proved in [13, 14] that if  $\mathbf{x}$  solves (6), then it will satisfy the following:

**Theorem 1.2** (Validation Principles).

$$\langle \mathbf{x}, A^*(\mathbf{b} - A\mathbf{x}) \rangle_U = \|\mathbf{x}\|_U \|A^*(\mathbf{b} - A\mathbf{x})\|_{U^*} \quad (7)$$

$$\|A^*(\mathbf{b} - A\mathbf{x})\|_{U^*} = \frac{1}{\lambda} \quad (8)$$

where  $U^*$  is the dual space of  $U$ . And  $(\mathbf{x}, \mathbf{r} = \mathbf{b} - A\mathbf{x})$  is called an extremal pair.

**Remark 1.3.** In this project, we will apply theorems from [13, 14] with  $U = \ell_1$ ,  $W = \ell_2$ ,  $q = 2$ , and  $A$  as any  $m \times n$  matrix over real or a linear operator defined by either the DCT or the FFT transformation.

Hence we can show that: a solution, say  $\mathbf{x}$ , from

$$\min_{\mathbf{x} \in \mathbf{R}^n} \left\{ \|\mathbf{x}\|_1 + \frac{\lambda}{2} \|\mathbf{b} - A\mathbf{x}\|_2^2 \right\} \quad (9)$$

---

<sup>4</sup>Another quotation needed

can be thought of an approximation to (3) with  $\|A^*(\mathbf{b} - A\mathbf{x})\|_\infty = \frac{1}{\lambda}$  by (8). According to [13], if  $\|A^*\mathbf{b}\|_\infty \leq \frac{1}{\lambda}$ , then the optimizer of (9) is  $\mathbf{0}$ . To avoid such trivial solution, we will pick a  $\lambda$  such that

$$\frac{1}{\|A^*\mathbf{b}\|_\infty} \leq \lambda \leq \frac{2}{\|A^*\mathbf{b}\|_\infty} \quad (10)$$

Generally speaking, as  $\lambda$  increases, the solution  $\mathbf{x}$  tends to be sparser ([7, 15]). Also with an increasing  $\lambda$ , we are avoiding the ill-posed part of (3) and emphasizing more on the regularized part of (3).

## 2 Hierarchical Decomposition

Following an alternate view point of (6) argued by the authors in [13], we employ a similar argument to (9).

### 2.1 Motivation

Let us consider the following J-functional:

$$J(\mathbf{b}, \lambda) := \operatorname{arginf}_{A\mathbf{x}+\mathbf{r}=\mathbf{b}} \left\{ \|\mathbf{x}\|_1 + \frac{\lambda}{2} \|\mathbf{r}\|_2^2 \right\} \quad (11)$$

And let  $\mathbf{x}_\lambda$  and  $\mathbf{r}_\lambda$  be an extremal pair such that they solve (11), that is:

$$\mathbf{b} = A\mathbf{x}_\lambda + \mathbf{r}_\lambda, \quad [\mathbf{x}_\lambda, \mathbf{r}_\lambda] = J(\mathbf{b}, \lambda) \quad (12)$$

then we have  $\|A^*\mathbf{r}_\lambda\|_\infty = \frac{1}{\lambda}$  by (8). While the pair  $[\mathbf{x}_\lambda, \mathbf{r}_\lambda]$  minimize  $J(\mathbf{b}, \lambda)$ , they also decompose  $\mathbf{b}$  into two parts: the recovered sparse signal  $\mathbf{x}_\lambda$  and residual  $\mathbf{r}_\lambda$ . Most of the time,  $\mathbf{r}_\lambda$  is viewed as noise under the scale  $\lambda$ . However, the difference between these two components is scale dependent - whatever is considered as noise at a given scale  $\lambda$  contains significant information when viewed under a refined scale, say  $2\lambda$ ,

$$\mathbf{r}_\lambda = A\mathbf{x}_{2\lambda} + \mathbf{r}_{2\lambda}, \quad [\mathbf{x}_{2\lambda}, \mathbf{r}_{2\lambda}] = J(\mathbf{r}_\lambda, 2\lambda) \quad (13)$$

We end up with a better two-scale approximation to  $\mathbf{b}$  given by  $\mathbf{b} \approx A(\mathbf{x}_\lambda + \mathbf{x}_{2\lambda})$ ; noise below scale  $\frac{1}{2\lambda}$  remains unresolved in  $\mathbf{r}_{2\lambda}$ . This process of (13) can continue. Beginning with an initial scale  $\lambda = \lambda_0$ ,

$$\mathbf{b} = A\mathbf{x}_0 + \mathbf{r}_0, \quad [\mathbf{x}_0, \mathbf{r}_0] = J(\mathbf{b}, \lambda_0)$$

we continue in this iterative manner for the decomposition of the dyadic refinement step (13),

$$\mathbf{r}_j = A\mathbf{x}_{j+1} + \mathbf{r}_{j+1}, \quad [\mathbf{x}_{j+1}, \mathbf{r}_{j+1}] = J(\mathbf{r}_j, 2^{j+1}\lambda_0), \quad j = 0, 1, 2, \dots \quad (14)$$

generating, after  $k$  such steps, the following **Hierarchical Decomposition** of  $\mathbf{b}$ :

$$\begin{aligned} \mathbf{b} &= A\mathbf{x}_0 + \mathbf{r}_0 \\ &= A\mathbf{x}_0 + A\mathbf{x}_1 + \mathbf{r}_1 \\ &= \dots \\ &= A\mathbf{x}_0 + A\mathbf{x}_1 + \dots + A\mathbf{x}_k + \mathbf{r}_k. \end{aligned} \quad (15)$$

We arrive at a new multiscale signal decomposition,  $\mathbf{b} \approx A(\mathbf{x}_0 + \mathbf{x}_1 + \dots + \mathbf{x}_k)$ , with a residual/noise  $\mathbf{r}_k$ . As  $k$  increases, the  $\mathbf{x}_k$ 's resolve signal with increasing sparsity and scales,  $\lambda_k := 2^k \lambda_0$ .

**Remark 2.1.** *The residual  $\mathbf{r}_k$  is decreasing since  $\|A^*r\|_\infty = \frac{1}{\lambda_k} = \frac{1}{2^k \lambda_0}$ , hence we are getting a better approximation to  $\mathbf{b}$ .*

## 2.2 The Algorithm

Starting out with an initial  $\lambda_0$  satisfying (10) and stopping at an optimal  $\lambda_j$  suggested in [11], we can decompose the measurement vector  $\mathbf{b}$  with signals on increasing level of sparsity,  $\mathbf{x}_j$ 's. We then present the following algorithm:

---

**Algorithm 1** Solve  $\min_{\mathbf{x} \in \mathbf{R}^n} \{ \|\mathbf{x}\|_1 + \frac{\lambda}{2} \|\mathbf{b} - A\mathbf{x}\|_2^2 \}$

---

**Require:**  $A$  and  $\mathbf{b}$ , pick  $\lambda_0$

**Ensure:**  $\mathbf{x} = \sum_{j=0}^J \mathbf{x}_j$ ;

Initialize:  $\mathbf{r}_0 = \mathbf{b}$ , and  $j = 0$ ;

**while** A certain  $\lambda_j$  is not found **do**

$\mathbf{x}_j(\lambda_j, \mathbf{r}_j) := \operatorname{argmin}_{\mathbf{x} \in \mathbf{R}^n} \{ \|\mathbf{x}\|_1 + \frac{\lambda_j}{2} \|\mathbf{r}_j - A\mathbf{x}\|_2^2 \}$ ;

$\mathbf{r}_{j+1} = \mathbf{r}_j - A\mathbf{x}_j$ ;

$\lambda_{j+1} = 2\lambda_j$ ;

$j = j + 1$ ;

**end while**

---

With a suitable solver for (9) the algorithm 1 is very intuitive to implement. Let us discuss in details about 2 special solvers for (9) in the following sections.

## 3 Implementation

We choose the Gradient Projection for Sparse Reconstruction method and A Fixed-Point Continuation method as built-in solvers for (9). We have singled these two methods out for their robustness and efficiency<sup>5</sup>.

### 3.1 Gradient Projection for Sparse Reconstruction

The **Gradient Projection for Sparse Reconstruction** algorithm is proposed in [4] and it is used to solve this following problem:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ \tau \|\mathbf{x}\|_1 + \frac{1}{2} \|\mathbf{b} - A\mathbf{x}\|_2^2 \} \quad (16)$$

---

<sup>5</sup>mention other methods and quotations.

We define  $(a)_+ = \begin{cases} a, & a \geq 0 \\ 0, & \text{otherwise} \end{cases}$ , and then let  $\mathbf{u} = (\mathbf{x})_+$ ,  $\mathbf{v} = (-\mathbf{x})_+$ . Putting  $\mathbf{u}$  and  $\mathbf{v}$  back into (16), and simplifying (16) further with  $\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$ ,  $\mathbf{y} = A^*\mathbf{b}$ ,  $\mathbf{c} = \tau\mathbf{1}_{2n} + \begin{bmatrix} -\mathbf{y} \\ \mathbf{y} \end{bmatrix}$ ,  $B = \begin{bmatrix} A^*A & -A^*A \\ -A^*A & A^*A \end{bmatrix}$ . we obtain:  $\min_{\mathbf{z} \in \mathbf{R}^{2n}} \{\mathbf{c}^*\mathbf{z} + \frac{1}{2}\mathbf{z}^*B\mathbf{z} \equiv \mathbf{F}(\mathbf{z}) \mid \mathbf{z} \geq 0\}$ . Note that  $\tau = \frac{1}{\lambda}$ . Before the algorithm is described, we have to introduce some extra parameters:  $\beta$  ( $0 < \beta < 1$ ) is used to control the gradient descent step size  $\alpha_0$ ,  $\mu$  ( $0 < \mu < \frac{1}{2}$ ) is used to make sure  $\mathbf{F}(\cdot)$  is decreased sufficiently from the "Armijo rule along the projection arc" in [1, p. 226] and  $\mathbf{g}^{(k)}$  is a projected gradient (defined component wise):

$$g_i^{(k)} = \begin{cases} (\nabla \mathbf{F}(\mathbf{z}^{(k)}))_i, & \text{if } z_i^{(k)} > 0 \text{ or } (\nabla \mathbf{F}(\mathbf{z}^{(k)}))_i < 0 \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

And with an initial guess of  $\mathbf{z}^{(0)} = \mathbf{0}$ , we have the following algorithm:

---

**Algorithm 2** Solve  $\min_{\mathbf{x} \in \mathbf{R}^n} \{\tau\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{b} - A\mathbf{x}\|_2^2\}$

---

**Require:**  $A$ ,  $\mathbf{b}$ ,  $\tau$ , and  $\mathbf{z}^{(0)}$ , pick  $\beta \in (0, 1)$  and  $\mu \in (0, 1/2)$ ;

**Ensure:**  $\mathbf{z}^{(K)}(\tau) := \min_{\mathbf{z} \in \mathbf{R}^{2n}} \{\mathbf{c}^*\mathbf{z} + \frac{1}{2}\mathbf{z}^*B\mathbf{z} \equiv \mathbf{F}(\mathbf{z}) \mid \mathbf{z} \geq 0\}$ ;

Initialize:  $k = 0$ ;

**while** A stopping criteria is not satisfied **do**

    Compute  $\alpha_0 = \min_{\alpha \in \mathbf{R}^1} \{\mathbf{F}(z^{(k)} - \alpha \mathbf{g}^{(k)})\}$ ;

    Let  $\alpha^{(k)}$  be the first in the sequence  $\alpha_0, \beta\alpha_0, \beta^2\alpha_0, \dots$ , such that  $\mathbf{F}((\mathbf{z}^{(k)} - \alpha^{(k)}\nabla \mathbf{F}(\mathbf{z}^{(k)}))_+) \leq \mathbf{F}(\mathbf{z}^{(k)}) - \mu \nabla \mathbf{F}(\mathbf{z}^{(k)})^*(\mathbf{z}^{(k)} - (\mathbf{z}^{(k)} - \alpha^{(k)}\nabla \mathbf{F}(\mathbf{z}^{(k)}))_+)$ ;

$\mathbf{z}^{(k+1)} = (\mathbf{z}^{(k)} - \alpha^{(k)}\nabla \mathbf{F}(\mathbf{z}^{(k)}))_+$ ;

$k = k + 1$ ;

**end while**

---

There are several types of stopping criteria available. For now we will take the most intuitive one:  $\|\mathbf{z} - (\mathbf{z} - \bar{\alpha}\nabla \mathbf{F}(\mathbf{z}))_+\|_2 \leq \nu$ , where  $\bar{\alpha}$  is a positive constant, and  $\nu$  is a pre-set tolerance. When such criteria is applied to GPSR, it is basically comparing the difference between the  $(k+1)^{th}$  and  $k^{th}$  iterates:  $\|\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\|_2$ . Although GPSR has increased the matrix-vector size from  $n$  to  $2n$ , the matrix vector multiplication can still be done at the  $O(n)$  level of complexity. We can simply the matrix-vector multiplication by  $B\mathbf{z} = \begin{bmatrix} A^*A(\mathbf{u} - \mathbf{v}) \\ -A^*A(\mathbf{u} - \mathbf{v}) \end{bmatrix}$ ,  $\mathbf{c}^*\mathbf{z} = \tau\mathbf{1}_n^*(\mathbf{u} + \mathbf{v}) - \mathbf{y}^*(\mathbf{u} - \mathbf{v})$ ,  $\mathbf{z}^*B\mathbf{z} = \|A(\mathbf{u} - \mathbf{v})\|_2^2$ ,  $\mathbf{F}(\mathbf{z}) = \tau\mathbf{1}_n^*(\mathbf{u} + \mathbf{v}) - \mathbf{y}^*(\mathbf{u} - \mathbf{v}) + \frac{1}{2}\|A(\mathbf{u} - \mathbf{v})\|_2^2$  and  $\nabla \mathbf{F}(\mathbf{z}) = \mathbf{c} + B\mathbf{z}$ . We also do not directly calculate the product  $A^*A$ .

**Remark 3.1.** Theorem 1 in [4] states that the sequence of  $\{z^{(k)}\}$  generated by the GPSR algorithm either terminates at a solution of  $\min_{\mathbf{z} \in \mathbf{R}^{2n}} \{\mathbf{c}^*\mathbf{z} + \frac{1}{2}\mathbf{z}^*B\mathbf{z} \equiv \mathbf{F}(\mathbf{z}) \mid \mathbf{z} \geq 0\}$  or converge to a solution of the aforementioned minimization problem at an  $R$ -linear rate.

### 3.2 Fixed-Point Continuation Method

In [6], the **Fixed-Point Continuation** algorithm is developed to solve the following:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ \|\mathbf{x}\|_1 + \frac{\lambda}{2} \|\mathbf{b} - A\mathbf{x}\|_M^2 \} \quad (18)$$

Where  $\|\mathbf{x}\|_M := \sqrt{\mathbf{x}^* M \mathbf{x}}$  and  $M$  is a Symmetric Positive Definite matrix. In this project, we take  $M = I$ . Let  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  and  $\mathbf{g} : \mathbf{R}^n \rightarrow \mathbf{R}^n$  defined as followed:

$$\begin{aligned} f(\mathbf{x}) &:= \frac{1}{2} \|\mathbf{b} - A\mathbf{x}\|_2^2 \\ \mathbf{g}(\mathbf{x}) &:= \nabla f(\mathbf{x}) = A^*(A\mathbf{x} - \mathbf{b}) \end{aligned}$$

And consider two mappings  $\mathbf{s}_\xi$  and  $\mathbf{h}$ , both from  $\mathbf{R}^n$  to  $\mathbf{R}^n$ , defined as (for any  $\eta > 0$ ):

$$\mathbf{h}(\mathbf{x}) := \mathbf{x} - \eta \mathbf{g}(\mathbf{x}) \quad (19)$$

$$\mathbf{s}_\xi(\mathbf{x}) := \mathbf{sgn}(\mathbf{x}) \odot \max\{|\mathbf{x}| - \xi, 0\} \quad (20)$$

where  $\xi = \frac{\eta}{\lambda}$  and  $\odot$  is component wise multiplication for vectors. And then consider the following fixed point equation:

$$\mathbf{x}^{k+1} = \mathbf{s}_\xi \circ \mathbf{h}(\mathbf{x}^k) \quad (21)$$

**Remark 3.2.** *As it is proven in [6], if  $\mathbf{x}_*$  solves (9), then  $\mathbf{0} \in \mathbf{sgn}(\mathbf{x}_*) + \lambda \mathbf{g}(\mathbf{x}_*)$  and vice versa. And if  $\mathbf{x}_*$  is a fixed point of (21), then  $\mathbf{0} \in \mathbf{sgn}(\mathbf{x}_*) + \lambda \mathbf{g}(\mathbf{x}_*)$  and vice versa. Thus establishing the equivalence between the fixed point of (21) and a minimizer of (9) through the signum equation.*

Let  $\rho_{max}$  be the maximum eigenvalue of the Hessian of  $f(\mathbf{x})$ , namely  $H(\mathbf{x}) = A^*A$ . As it is shown in [6], we have to maintain  $\eta \in (0, \frac{2}{\rho_{max}})^6$  in order to have convergence results. Under this setting, [6] proposes the following algorithm:

---

**Algorithm 3** Solve  $\mathbf{sgn}(\mathbf{x}) + \lambda \mathbf{g}(\mathbf{x}) = 0$

---

**Require:**  $A, \mathbf{b}, \lambda$ , pick  $\mathbf{x}^0$ , set  $\bar{\mu} = \lambda$ ;

**Ensure:**  $\mathbf{x}$ ;

Select:  $0 < \mu_1 < \mu_2 < \dots < \mu_L = \bar{\mu}$ ;

**for**  $\mu = \mu_1, \mu_2, \dots, \mu_L$  **do**

**while** A convergence test is not satisfied **do**

        Select  $\eta$  and set  $\xi = \frac{\eta}{\mu}$

$\mathbf{x} \leftarrow \mathbf{s}_\xi \circ \mathbf{h}(x)$ ;

**end while**

**end for**

---

Although the fixed point iteration is simple, the algorithm depends on a suitable choice of  $\eta$  and an appropriate sequence of  $\mu_i$ 's and an initial guess  $\mathbf{x}^0$ , and even the stopping criteria is problem dependent. All of these requirement makes the algorithm harder to apply to general problems.

---

<sup>6</sup>To have faster convergence, we have to have  $\eta \in [\frac{1}{\rho_{max}}, \frac{2}{\rho_{max}})$

### 3.3 Implementation Platform and Memory Allocation

Codes will be written in Matlab for GPSR, FPC, and the whole HD algorithm. When time permits, parallel codes will be written in C. The version of the Matlab running on my personal computer is: 7.12.0.635(R2011a). It is installed on a copy of the Windows 7 Home Premium operating system (64 bit). Validations and testing of GPSR and FPC will be run on my personal computer with AMD Phenom™N950 Quad-Core processor (clocked at 2.10 GHZ) and 4.00 GB (DDR3) memory. If the test problems are big enough, clusters at the CSCAMM will be used.  $Ax$  and  $A^*x$  can be defined as function calls instead of direct matrix-vector multiplication in order to save memory allocation.

## 4 Selection of Databases

Databases are not needed for this stage of testing yet.

## 5 Validation Principles

If  $\mathbf{x}$  is a solution of (9), then  $\mathbf{x}$  and  $A^*(\mathbf{b} - A\mathbf{x})$  form an extremal pair ([14]). They satisfy (7) and (8).

## 6 Testing

### 6.1 Compressed Sensing Cases

As it is suggested in [4], we are to test the code using  $m = 1024$ ,  $n = 4096$ , and the original signal  $\mathbf{x}_{true}$  has 160 randomly placed  $\pm 1$  spikes, and the matrix  $A$  is generated first by being filled with independent samples of a standard Gaussian distribution and then orthonormalizing the rows. The paper also suggested a initial  $\tau = 0.1\|A^*\mathbf{b}\|_\infty$ , and the measurement vector  $\mathbf{b}$  is corrupted with noise, hence  $\mathbf{b} = A\mathbf{x}_{true} + \zeta$ , where  $\zeta$  is a white Gaussian noise of variance  $\sigma^2$  with  $\sigma^2 = 10^{-4}$ . In the case of Compressed Sensing testing, we can have some pre-processing information about the parameters as suggested in [6]. Let  $\delta = \frac{m}{n}$  and  $\gamma = \frac{k}{m}$ , where  $k$  stands for the number of non-zero elements in the original signal  $\mathbf{x}_{true}$ . Then we can pick  $\eta = \min\{1 + 1.665(1 - \delta), 1.999\}$ ,  $\mathbf{x}^0 = \eta A^*M\mathbf{b}$ ,  $\mu_1 = \theta\|\mathbf{x}^0\|_\infty$  (where  $0 < \theta < 1$  is a user-defined constant), set  $\mu_i = \min\{\mu_1\omega^{i-1}, \bar{\mu}\}$  (where  $\omega > 1$  and  $L$  is the first integer  $i$  such that  $\mu_i = \bar{\mu}$ )<sup>7</sup>. And we can test the solutions with the following convergence test:

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2}{\max\{\|\mathbf{x}^k\|_2, 1\}} < xtol \quad \text{and} \quad \mu_i\|\mathbf{g}(\mathbf{x}^k)\|_\infty - 1 < gtol \quad (22)$$

As it was done in [6], we will take  $xtol = 10^{-4}$  and  $gtol = 10^{-2}$ . We will also try the algorithm with  $A$  being either a DCT or FFT transform matrix.

<sup>7</sup>[6] also suggests  $\theta = 0.99$  and  $\omega = 4$  or  $\theta = 0.9$  and  $\omega = 2$

## 6.2 Image Deconvolution, Deblurring

Plans with Image Processing experiments will be developed later after a better understanding of the vector representation of images is achieved, as well as the convolution operator.

## 7 Project Phases and Time lines

The following table provides specified time line for my project:

Phase I	08/29/2012 to 10/05/2012	Project Background Research, Project Proposal Presentation and Report
Phase II	10/06/2012 to 11/21/2012	Implementation of the GPSR algorithm and validation by theorem 1.2
Phase III	11/22/2012 to 12/20/2012	Implementation of the FPC algorithm and preparation for mid-year presentation and report
Phase IV	12/21/2012 to 01/22/2013	Validation of the FPC algorithm by theorem 1.2
Phase V	01/23/2013 to 02/22/2013	Implementation of the whole HD algorithm
Phase VI	02/23/2013 to 03/16/2013	Validation of the HD algorithm by theorem 1.2
Phase VII	03/17/2013 to 05/20/2013	Final Testing phase and preparation for end-of-year presentation and report

## 8 Milestones

Here are major milestones about the project:

Phase I	Presentation given on 10/02/2012 and Project Proposal written on 10/05/2012
Phase II	Implementation of the GPSR algorithm starts on 10/06/2012

## 9 Deliverables

Matlab codes, presentation slides (proposal presentation, mid-year presentation, end-of-year presentation), the complete project document, test databases (if any), and test results (both in text file and/or figures) will be delivered at the end of this year long sequence.

## References

- [1] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, Boston, 2<sup>nd</sup> ed., 1999.
- [2] E. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489 – 509.
- [3] D. L. DONOHO, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289 – 1306.
- [4] M. A. T. FIGUEIREDO, R. D. NOWAK, AND S. J. WRIGHT, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586 – 597.
- [5] J. J. FUCHS, *More on sparse representations in arbitrary bases*, IEEE Transactions on Information Theory, (2004), pp. 1341 – 1344.
- [6] E. T. HALE, W. YIN, AND Y. ZHANG, *A fixed-point continuation method for  $\ell_1$ -regularized minimization with applications to compressed sensing*, Technical Report TR07 - 07, CAAM.
- [7] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer-Verlag, New York, 2001.
- [8] S.-J. KIM, K. KOH, M. LUSTIG, S. BOYD, AND D. GORINVESKY, *A method for large-scale  $\ell_1$ -regularized least squares problems with applications in signal processing and statistics*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 606 – 617.
- [9] F. LASTNAME, F. I. LASTNAME, AND F. LASTNAME JR., *Conference paper MUN title*, in Proceedings of the Conference of Sample Conferences, Apr. 1996, pp. 100–110.
- [10] B. K. NATARAJAN, *Sparse approximation solutions to linear systems*, SIAM J. COMPUT, 24 (1995), pp. 227 – 234.
- [11] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling & Simulation, 4 (2005), pp. 460 – 489.
- [12] S. OSHER, Y. MAO, B. DONG, AND W. YIN, *Fast linearized bregman iteration for compressive sensing and sparse denoising*, Communications in Mathematical Sciences, 8 (2011), pp. 93 – 111.
- [13] E. TADMOR, S. NEZZAR, AND L. VESE, *A multiscale image representation using hierarchical (bv,  $l^2$ ) decomposition*, Multiscale Modeling & Simulation, 2 (2004), pp. 554 – 579.
- [14] ———, *Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation*, Communications in Mathematical Sciences, 6 (2008), pp. 281 – 307.

- [15] R. TIBSHIRANI, *Regression shrinkage and selection via lasso*, Journal of the Royal Statistical Society, Series B (Methodological), 58 (1996), pp. 267 – 288.
- [16] A. N. TKHONOV, *On the stability of inverse problems*, Doklady Akademii Nauk SSSR, 39 (1943), pp. 195 – 198.
- [17] —, *Solution of incorrectly formulated problems and the regularization method*, Doklady Akademii Nauk SSSR, 151 (1963), pp. 501 – 504.