Sparse Signal Reconstruction using Weight Point Algorithm

Koredianto Usman[1,3*], Hendra Gunawan[2] & Andriyan B. Suksmono[1]

[1]School of Electrical Engineering & Informatics
Bandung Institute of Technology (ITB), Jalan Ganesha 10, Bandung 40132, Indonesia
[2]Faculty of Mathematics and Natural Sciences
Bandung Institute of Technology (ITB), Jalan Ganesha 10, Bandung 40132, Indonesia
[3] Faculty of Electrical Engineering, Telkom University, Jalan Telekomunikasi No.1,
Bandung 40257, Indonesia
*E-mail: koredianto.usman@telkomuniversity.ac.id

Abstract. In this paper we propose a new approach of the compressive sensing (CS) reconstruction problem based on a geometrical interpretation of l_1-norm minimization. By taking a large l_1-norm value at the initial step, the intersection of l_1-norm and the constraint curves forms a convex polytope and by exploiting the fact that any convex combination of the polytope's vertexes gives a new point that has a smaller l_1-norm, we are able to derive a new algorithm to solve the CS reconstruction problem. Compared to the greedy algorithm, this algorithm has better performance, especially in highly coherent environments. Compared to the convex optimization, the proposed algorithm has simpler computation requirements. We tested the capability of this algorithm in reconstructing a randomly down-sampled version of the Dow Jones Industrial Average (DJIA) index. The proposed algorithm achieved a good result but only works on real-valued signals.

Keywords: *compressive sampling; convex combination; convex polytope; sparse reconstruction; l_1-norm, weight point.*

1 Introduction

Compressive sampling (CS) is a new technique in signal processing that efficiently combines acquisition and compression in a single step. Among the pioneers in this subject is Donoho with his seminal paper from 2006 [1]. Other researchers jointly or independently contributed to the development of CS, among others Candes and Wakin [2], Candes and Tao [3], and Baraniuk [4]. Even though this technique was introduced in the early 2000s, its foundation was laid in the early 1980s in the form of several mathematical techniques such as the uncertainty principle and signal recovery [5], basis pursuit [6], and the time-frequency representation of signals [7].

CS consists of two main parts, i.e. a compression step and a reconstruction step. In practice, the compression step in CS can be viewed mathematically as a

Received December 2nd, 2016, 1st Revision August 15th, 2017, 2nd Revision January 4th, 2018, Accepted for publication January 16th, 2018.

multiplication of sampling or sensing matrix A with signal x to produce a compressed signal y, i.e. $y = Ax$. If the length of x is N, then by selecting A to be an $M \times N$ matrix with $M << N$, we can produce a much smaller dimension of y. After compression it is often required to reconstruct the original signal x back from the compressed signal y. This step is called CS reconstruction.

Since sparse or compressible signals have a wide range of applications, CS has been applied in many fields, such as radar and sonar [8-10], antenna beam forming [11,12], imaging [13,14], and video [15], to name a few. Nevertheless, there are still challenges, because CS reconstruction generally involves a heavy computational load and is time-consuming.

At the moment there are two main groups of CS reconstruction algorithms, i.e. convex optimization and the greedy algorithm. Convex optimization solves CS reconstruction by minimizing the l_1-norm of the available solutions. Convex optimization can be applied since l_1-norm is a convex function. Greedy algorithm, on the other hand, solves the CS reconstruction heuristically by choosing a local optimum at each intermediate step in the hope of finding the global optimum at the end. The greedy algorithm does not guarantee a correct solution, but it has fast computation. In terms of accuracy, the greedy algorithm suffers from reconstruction errors when the sensing matrix has high coherence. Convex optimization, on the other hand, provides a robust estimate but is much slower than the greedy algorithm.

In this paper, we propose a new CS reconstruction algorithm based on a geometric interpretation of l_1-norm minimization in the CS reconstruction problem. Using a bisection method, l_1-norm is reduced iteratively until the minimum value is obtained. In terms of accuracy and complexity, our proposed method stands between convex optimization and the greedy algorithm. It is faster than convex optimization and more robust than the greedy algorithm. This method offers an alternative way of solving CS reconstruction when a balance between computation time and accuracy is required.

This paper is arranged as follows. Section 2 explores related works, especially on convex optimization and the greedy algorithm. Section 3 describes the details of the proposed method. Section 4 reports our numerical experiments, consisting of a performance comparison of the proposed method with convex optimization and the greedy algorithm. A test case of how the proposed method performs on a randomly down-sampled version of the Dow Jones Industrial Average (DJIA) index is also presented in this section. Finally, Section 5 concludes the paper by summarizing the overall findings and giving an outlook on a future CS reconstruction strategy.

2 CS Reconstruction Problem and Related Works

The reconstruction of sparse signals can be traced back to the uncertainty principle and signal recovery. One of the early papers on this subject is [5] by Donoho and Stark. For a continuous signal, the uncertainty principle states that any continuous signal cannot be dense in both the time and the frequency domain simultaneously. If a signal *x(t)* is concentrated in time interval Δ_t and at frequency interval Δ_f, then shown in following Eq. (1)

$$\Delta_t \cdot \Delta_f \geq 1 \tag{1}$$

In the case of a discrete signal *x(n)*, if the signal has length *N* and it has nonzero values in time interval N_t and if *X(f),* a discrete Fourier transform of *x(n),* has N_f nonzero values, then

$$N_t \cdot N_f \geq N \tag{2}$$

Donoho and Stark derived another form of Eq. (2) in following Eq. (3):

$$N_t + N_f \geq N\sqrt{2} \tag{3}$$

The uncertainty principle in the time-frequency domain was generalized by Elad and Bruckstein [16] for presenting $\boldsymbol{x}$ on any basis. This generalization is called the generalized uncertainty principle (GUP). This principle states that if $\boldsymbol{x} \in \mathbb{R}^N$ can be represented in two orthonormal bases $\Phi \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\Psi} \in \mathbb{R}^{N\times N}$, then the following inequality holds:

$$|T| + |\Omega| \geq {}^{2}\!/_{\mu} \tag{4}$$

In Eq. (4), $|T|$ and $|\Omega|$ represent the number of nonzero elements of x in bases $|\boldsymbol{\Phi}|$ and $|\Psi|$ respectively. The value of μ corresponds to the highest inner product of vectors in Φ and Ψ, that is:

$$\mu = max_{i,j}\langle \phi_i, \psi_i \rangle \text{ with } i = 1, 2, \dots, N \text{ and } j = 1, 2, \dots, N \tag{5}$$

The μ value is also called the coherency between Φ and Ψ. The lower and upper bounds of μ are $1/\sqrt{N}$ and *1* respectively, where the lower bound is reached when either Φ or Ψ has vectors whose components are random numbers and the upper bound is reached when any two vectors in Φ and Ψ are identical. Another important finding of Elad and Bruckstein is related to the uniqueness of the representation of sparse vector *x* using an over-complete dictionary composed of $\Phi \cup \Psi$. After stating this uniqueness, let us briefly overview the CS principle and after that, we continue with related works on two reconstruction algorithms, i.e. convex optimization and the greedy algorithm.

Let $\boldsymbol{x}$ be a vector that represents the discrete time sparse signal of length N with k nonzero elements and $k << N$. A direct method to compress $\boldsymbol{x}$ is by pre-multiplying $\boldsymbol{x}$ with an $M \times N$ dimension of matrix $\boldsymbol{A}$, i.e. a sensing matrix with $M < N$, to produce a compressed signal $\boldsymbol{y}$ of length M:

$$y = Ax \tag{6}$$

The compression as represented by Eq. (6) can also be viewed as a system of a linear equation with an independent variable $\boldsymbol{x}$ and an dependent variable $\boldsymbol{y}$. The reconstruction problem is therefore how to obtain sparse signal x from sensing matrix A and compressed signal y. Since A has dimension $M \times N$ and $M < N$, Eq. (6) represents an underdetermined system of linear equations. Therefore, solving Eq. (6) for x produces an infinite number of possible answers. However, by knowing that the signal x is sparse, we can pick the solution from this infinite number of answers that has the least nonzero elements and take it as the solution of the reconstruction problem. This solution is called the sparsest solution, or P_0. Mathematically, we can write:

$$P_0 = min\ ||x||_0\ \ subject\ to\ \ Ax = y \tag{7}$$

In Eq. (7), $||x||_0$ denotes the zero-th order norm of x (or l_0-norm), which is equivalent to the support of vector x, i.e. the number of nonzero elements of $\boldsymbol{x}$ (also equal to signal sparsity k) in following Eq. (8).

$$||x||_0\ = supp(x) = k \tag{8}$$

Finding P_0 from Eq. (7) is, however, an NP-hard problem. That is, the solution can be found only by exhaustive search. Therefore, the requirement of minimizing l_0-norm is relaxed to minimizing the first order norm, that is:

$$P_1 = min\ ||x||_1\ \ subject\ to\ \ Ax = y \tag{9}$$

where $\|\boldsymbol{x}\|_1$ is the first order norm of $\boldsymbol{x}$. The formulation in Eq. (9) is also called basis pursuit (BP). However, relaxing P_0 to P_1 requires higher sparsity as indicated by Donoho and Huo [17]. Regarding the uniqueness of P_1 and P_0, Donoho & Huo and Elad & Bruckstein formulated the following proposition:

Proposition 1: *Let* Φ_1 *and* Φ_2 *be two orthonormal bases with coherency* μ *and let* $\Phi = \Phi_1 \cup \Phi_2$ *be a superset of* Φ_1 and Φ_2. *If a discrete time signal* s *can be* *represented as* s = $\Phi\alpha$, *where the* l_0-*norm of* α *fulfills* $\|\alpha\|_0 < (1/2)\{1+(\mu(\Phi_1, \Phi_2))^{-1}\}$, *then* α *is a unique solution of* P_1 *and also a unique solution of* P_0.

Proposition 1 is very important since it guarantees that Eq. (9) has a unique solution. Furthermore, since l_1-norm is a convex function, a standard method such as convex optimization can be used to solve Eq. (9) iteratively. Using convex optimization, BP is cast into a linear programming form, which can be written as [18]:

$$\begin{aligned} &\min \sum_i u_i \quad s.t. \quad x_i - u_i \leq 0, \\ &-x_i - u_i \leq 0, \\ &\boldsymbol{Ax} = \boldsymbol{y} \end{aligned} \tag{10}$$

This LP problem can be solved using the interior point method (IPM). Several software packages have been developed for this purpose, for example, Matlab with its optimization toolbox, CVX discipline programming by Boyd and Vandenberghe [19], and l_1-magic by Candes and Romberg [18].

Another class of CS reconstruction solvers uses the greedy algorithm. Using this approach, CS sampling as in Eq. (6) is viewed as a linear combination of each column in A determined by x to produce y. The greedy algorithm works reversely by finding a best fit vector in A one at a time, and repeats iteratively until best estimate of x is achieved. Among the most famous greedy algorithms are matching pursuit (MP) and orthogonal matching pursuit (OMP). MP was popularized by Mallat and Zhang [7] in connection with reconstruction using a time-frequency dictionary. OMP was developed rather independently from MP by Chen, *et al.* [20]. Basically, MP and OMP have the same main procedure, to which a least square step is added in OMP. Among the earliest applications of OMP in signal processing was wavelet decomposition [21]. The success of MP and OMP was followed by the introduction of several variants of them. These variants are, for example, the Regularized OMP (ROMP, [22]), Stagewise OMP (StOMP, [23]), and Compressive Sampling MP (CoSaMP, [24]). The initial suggestion to use OMP for CS reconstruction was made by Tropp and Gilbert [25]. In this section we review only MP and OMP to represent the use of the greedy algorithm.

As stated previously, the basic principle of MP in solving CS is by viewing the $M \times N$ sensing matrix A as a collection of N vectors $\alpha_1, \alpha_2, \cdots, \alpha_N$ of dimension M. That is:

$$A = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_N] \tag{11}$$

If we let the unknown x to be $x = [x_1 \quad x_2 \quad \cdots \quad x_N]$, then $Ax = y$ can be seen as a linear combination of each basis α_i with a corresponding weight of x_i to form y. This can be written as:

$$Ax = x_1\alpha_1 + x_2\alpha_2 + \cdots + x_N\alpha_N \tag{12}$$

Solving for x, MP works reversely by decomposing y into each component of α_i iteratively, where in each iteration the estimated weight $\hat{x}_i$ is chosen to be the largest contribution to y. The largest contribution is taken as the maximum dot product of α_j ($j = 1, 2, \cdots, N$) to y. After the first $\hat{x}_i$ has been chosen, the residue of A is calculated and the next estimate is calculated in a similar manner.

As already mentioned before, OMP has the same basic procedure as MP but it adds a least square step. OMP has better accuracy than MP, since the least square step ensures that the residue of the previous selected base is not recounted in the next step.

As compared to convex optimization, OMP has the advantage of having faster computation. However, OMP does not always produce a correct estimate, especially when the sensing matrix is composed of high-coherence bases. This happens because high-coherence bases create confusion as they are mixed up by the high coherency. A deeper analysis of the greedy algorithm can be found, for example, in DeVore and Temlyakov's paper [26] and also in Cohen's paper [27].

3 Proposed Method

Let us now consider Eq. (9) from a geometrical point of view. The objective function, i.e. $\|x\|_1$, is a rectangular shape and the objective function $\boldsymbol{Ax = y}$ forms a straight line in a *two*-dimensional coordinate system. The solution of Eq. (9) is the intersection of the curve $Ax = y$ with the curve $\|x\|_1 = k_{opt}$, which in Fig. 1(A) is denoted by x_s. If we take $k > k_{opt}$, then $Ax = y$ and $\|x\|_1 = k$ intersect at two points and if $k < k_{opt}$ then $Ax = y$ and $\|x\|_1 = k$ do not intersect (Fig. 1(A)). From this illustration, we conclude that the solution of Eq. (9) is obtained when $\|x\|_1 = k$ touches $Ax = y$. Capitalizing on this fact, we start deriving the proposed algorithm by analyzing a simple case of two and three dimensions of x and then generalize it for any N dimensions.

Two dimension case. Let $x = [x_1 \; x_2]^T \in R^2$ and $A = [a_1 \; a_2] \in R^2$. Therefore, $y = Ax$ produces $y = y_1 = a_1 x_1 + a_2 x_2$. The reconstruction problem is finding x, given y and A. Using BP, one needs to minimize $\|x\|_1$ fulfilling $Ax = y$. By substituting x and A, $Ax = y$ produces $a_1 x_1 + a_2 x_2 = y_1$, which is a straight line in the x_1-x_2 coordinate system.

In order to find k_{opt}, we start with initial step $k = k_0$, which is large enough so that curve $\|x\|_1 = k_0$ and $Ax = y$ intersect each other at points P_1 and P_2 (Fig. 2(a)). Let M_1 be a convex combination of P_1 and P_2, that is:

$$M_1 = \beta \cdot P_1 + (1-\beta) \cdot P_1 \quad ; \quad 0 \le \beta \le 1 \tag{13}$$

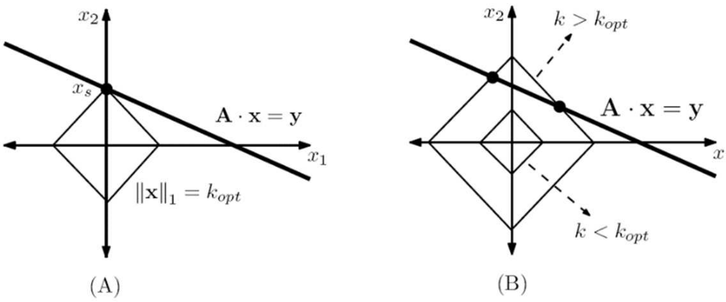

x_2
x_s
$\mathbf{A} \cdot \mathbf{x} = \mathbf{y}$
x_1
$\|\mathbf{x}\|_1 = k_{opt}$
(A)
x_2
$k > k_{opt}$
$\mathbf{A} \cdot \mathbf{x} = \mathbf{y}$
x_1
$k < k_{opt}$
(B)

Figure 1 Illustration of CS reconstruction with two variables using l_1-norm minimization. (A) Solution $\boldsymbol{x}_s$ fulfilling BP. (B) The case of $\|\boldsymbol{x}\|_1 = k > k_{opt}$ produces intersections, while $\|\boldsymbol{x}\|_1 = k < k_{opt}$ does not produce intersections.

As M_1 is a convex combination of P_1 and P_2, it lies between P_1 and P_2 (Figure 2(A)). In addition, since M_1 is inside $\|\boldsymbol{x}\|_1 = k_0$, a new norm $\|\boldsymbol{x}\|_1 = k_1$ that passes through M_1 will be smaller. That is, $k_1 \le k_0$ for any value of β in [0, 1]. The equality holds only if $\beta = 0$ or $\beta = 1$. We delay the proof of this statement until the general N-variables case has been discussed. After the new norm, k_1, has been calculated, we can repeat the previous steps to find M_2 (Figure 2(B)). These steps produce k_2, which is smaller than k_1. We repeat these steps until the j-th iteration, where the value of k_j is close enough to k_{opt}.

Three-dimensional case. Let x be $[x_1 \;\; x_2 \;\; x_3]^T$ and A is a 1×3 matrix, i.e. $[a_{11} \;\; a_{12} \;\; a_{13}]$. Equation $Ax = y$ produces $a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = y_1$, which is a plane in the x_1-x_2-x_3 coordinate system. Norm $\|x\|_1 = k_0$ is an octahedron in this three-dimensional coordinate system. At the initial step, we take k_0 sufficiently large so that $Ax = y$ intersects $\|x\|_1 = k_0$ as shown in Figure 3(A). Now, depending on the value of k_0, the intersection of $Ax = y$ and $\|x\|_1 = k_0$ may produce a polygon with four or more sides, which is called a polytope. Figure 3(B) illustrates the intersection of these curves to give a five-sided polytope. We denote P_1, P_2, P_3, P_4, and P_5 as the vertexes of the polytope as shown in Figure 3(B). Point M_1 is calculated as the convex combination of these vertexes as:

$$M_1 = \beta_1 \cdot P_1 + \beta_2 \cdot P_2 + \cdots + \beta_R \cdot P_R, \tag{14}$$

with

$$\beta_1 + \beta_2 + \cdots + \beta_R = 1, \tag{15}$$

and $0 \leq \beta_i \leq 1$, $i = 1, 2, \ldots, R$, where R is the number of vertexes in the polytope. Using similar reasoning as in the two-dimensional case, a new norm curve $\|x\|_1 = k_1$, which passes through M_1, has a smaller norm value than the previous k_0. The iteration can be restarted using a new value for norm k_1.

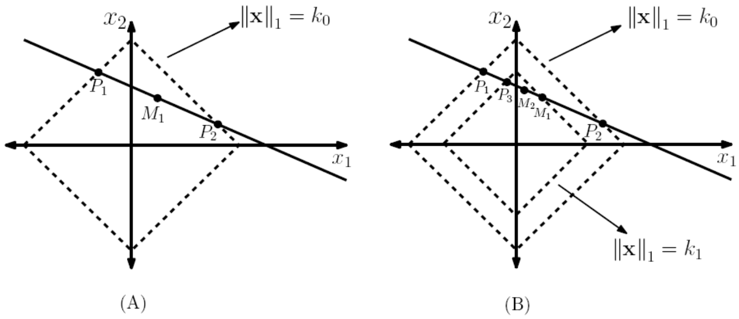


Figure 2 (A) Illustration at initial iteration, with k_0 is sufficiently large, the curve $Ax = y$ and $\|x\|_1 = k_0$ intersect at P_1 and P_2. The point M_1 is the convex combination of P_1 and P_2. (B) The norm at M_1 is chosen as the next value of k for the next iteration.

Generalization to N dimensions. As we can observe from the two- and three-dimensional cases, the intersection of $Ax = y$ and $\|x\|_1 = k_0$ with large enough k_0 forms a convex polytope. Selecting a point M as the convex combination of the vertexes of the polytope will then produce a smaller l_1-norm. This observation is stated in the following propositions.

Proposition 2: *Let* Ax = y *be a system of linear equations with* A $\in \mathbb{R}^{M\times N}$, x $\in \mathbb{R}^N$, *and* y $\in \mathbb{R}^M$. *If* Ax = y *intersects* $\|x\|_1 = \mathrm{k}_0$, *then the edges of the intersection form a convex polytope.*

Proof: Since $\|x\|_1 = k_0$ forms a closed convex set where each side is represented by $\sum_{i=1}^{N} f_i x_i = k_0$ with f_i is either +1 or -1, the intersection of each side of $\|x\|_1 = k_0$ with Ax $= y$ is a straight line. As $Ax = y$ spans unbounded in each x_i direction, plane $Ax = y$ cuts the sides of $\|x\|_1 = k_0$ from one side to other sides. Therefore, the intersection forms a closed set with straight lines as the boundary, which is a convex polytope.

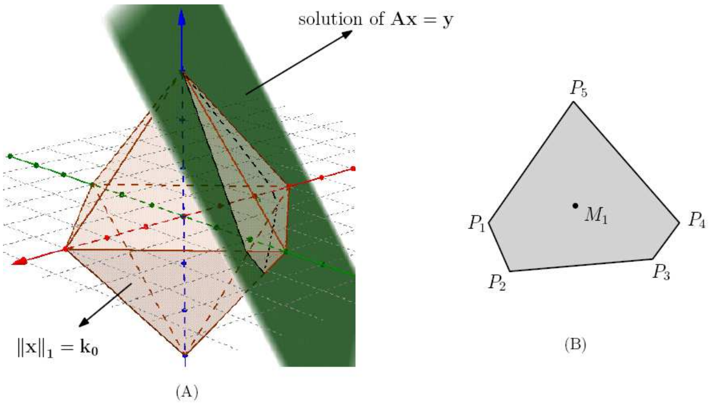

solution of $\mathbf{Ax} = \mathbf{y}$
$\|\mathbf{x}\|_1 = \mathbf{k}_0$
(A)
P_5
P_1
M_1
P_4
P_2
P_3
(B)

Figure 3 (A) In the case of three variables, $\|\boldsymbol{x}\|_1 = k_0$ forms an octahedron. The solution $Ax = y$ with a 1 × 3 matrix A is a plane. If k_0 is sufficiently large, $\|\boldsymbol{x}\|_1 = k_0$ intersects $Ax = y$. (B) The intersection plane is a convex polytope. A point M_1 is the convex combination of points P_i as a new point with a smaller norm.

Proposition 3: *Let* $\mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_\mathrm{R}$ *be the vertexes of the polytope as described in Proposition 2. Any point* M *as a convex combination of* $\mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_\mathrm{R}$ *has an* l_1*-norm that is smaller than the* l_1*-norm of each vertex* P_i.

Proof: Since each P_i is on the polytope, the l_1-norm on P_i is k_0, i.e. $\|P_i\|_1 = k_0$. Let M be a convex combination of P_i, therefore M satisfies Eq. (14) with the summation of β_i satisfying Eq. (15). Using Eq. (15), the value of l_1-norm at M is:

$$\begin{aligned} \|M\|_1 &= \|\beta_1 P_1 + \beta_2 P_2 + \cdots + \beta_N P_N\| \\ &\leq \|\beta_1 P_1\| + \|\beta_2 P_2\| + \cdots + \|\beta_N P_N\| \end{aligned} \tag{16}$$

Since $\beta_i \geq 0$ for all $i=1,2,..,N$ and $\|P_i\|_1 = k_0$, Eq. (16) can be simplified to Eq. (17):

$$\begin{aligned} \|M\|_1 &\leq \beta_1 \|P_1\| + \beta_2 \|P_2\| + \cdots + \beta_N \|P_N\| \\ &\leq \beta_1 k_0 + \beta_2 k_0 + \cdots + \beta_N k_0 = k_0(\beta_1 + \beta_2 + \cdots + \beta_N) = k_0 \end{aligned} \tag{17}$$

For practical purposes, we can choose the value of β_i for all $i=1, 2, ..., N$ to be equal to $1/N$. Therefore, point M_i is the weight point of the polytope. This selection of β_i may not be the best point for fast convergence, but it works well for most CS reconstructions.

After giving these two propositions and their proofs, we now can state the weight point algorithm for N dimensions of $\boldsymbol{x}$.

Weight Point Algorithm.

```
Input: $\boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{y} \in \mathbb{R}^{M}$
Output: $\boldsymbol{x} \in \mathbb{R}^{N}$
1.  Initiate $k \leftarrow k_0$ with $k_0$ sufficiently large; $\Omega \leftarrow \{\ \ \}$; loop index $i \leftarrow 1$, and
    $\mathbf{F} \leftarrow \begin{bmatrix} +1 & +1 & \cdots & +1 \\ +1 & +1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & 1 \end{bmatrix}$ an $2^N \times N$ dictionary matrix,
2.  assign $\boldsymbol{x_{hi}} = \boldsymbol{F}(i,:)$,
3.  construct $[\boldsymbol{A}; \boldsymbol{x_{hi}}] = [\boldsymbol{y}\ ; k]$,
4.  solve $[\boldsymbol{A}; \boldsymbol{x_{hi}}] = [\boldsymbol{y}\ ; k]$ to obtain the solution $\boldsymbol{\lambda_i}$ i.e. the vertexes of a
    polytope,
5.  update the set of solutions $\boldsymbol{\Omega} \leftarrow \{\boldsymbol{\Omega} \cap \lambda_i\}$,
6.  update the counter $i \leftarrow i + 1$,
7.  repeat step 2 to 6 until $i = N$,
8.  remove repeated columns in $\Omega$,
9.  calculate the weight point of the polytope: $\boldsymbol{M_i} = \sum_{j=1}^{R} \boldsymbol{\Omega}(:, j)/R$
10. calculate the $l_1$-norm of $M_i$ : $k_M = \|M\|_1$ and $\delta = |k - k_M|$, update $k \leftarrow k_{M,}$,
11. if $\delta \le \varepsilon$, with $\varepsilon$ is a small positive number, then assign $\boldsymbol{x} = \boldsymbol{M_i}$ and stop,
    else reset $i$ by assigning $i \leftarrow 1$ and repeat step 2 to 10.
```

Step 9 is the calculation of the convex combination of each vertex in the polytope using each value of $\beta_i = 1/R$ which is the weight point of the polytope. Therefore, we call this proposed method the Weight Point Algorithm. Step 4 is crucial because it finds the vertexes of the polytope. It is performed by finding the intersection point between each side of l_1-norm and the constraint. An efficient method to solve this problem is Householder QR-factorization (Golub and Van Loan [28]). Householder QR-factorization is a procedure to factorize any matrix A that is:

$$A = QR \tag{18}$$

where Q is an orthonormal matrix and R is an upper triangular matrix. This factorization is particularly useful for solving linear equations. The details of this method can be found in [28], for example.

4 Numerical Experiments

In order to measure the effectiveness of the proposed method we performed a computer simulation on the reconstruction of a CS problem. In this simulation, we measured the performance of the algorithm for amplitude reconstruction accuracy, reconstruction accuracy as a function of the coherency of A. We used CVX programming, which is a Matlab package for convex optimization and OMP as a representation of the greedy algorithm as the comparator. In the final experiment, we tested our proposed algorithm in a real-world application of data interpolation using the Dow Jones Industrial Average (DJIA) index.

In our experiments, we used mean absolute error (MAE) to measure the closeness between the reconstructed signal $\hat{\boldsymbol{x}}$ and the original signal $\boldsymbol{x}$. MAE is calculated as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N}|x_i - \hat{x}_i| \tag{19}$$

4.1 Amplitude Reconstruction Accuracy

This simulation was carried out to compare reconstruction accuracy in terms of amplitude similarity between the actual signal and the reconstructed signal. This comparison is important to see the capability of the algorithm of achieving a perfect reconstruction. In this simulation, we selected a sparse signal of 12 elements, i.e. $x = [0\ 2\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$. The non-zero elements of x are at the second and third positions. The sensing matrix A is composed of a column-wise Gaussian normalized random number with zero mean and unit variance, in other words, $A = [a_1\ a_2\ \dots\ a_N]$; $a_i \in \mathcal{N}(0,1)$. In this particular experiment, we compressed x to become y using a 10×12 sized matrix A. The simulation was repeated 100 times and the average of the result was collected. Figure 4 shows the reconstruction results of each method.

The proposed method together with OMP produced a perfect reconstruction both of nonzero and zero elements. CVX programming, on the other hand, even though it produced a result with a similar pattern to the original signal, it did not produce a perfect reconstruction. To quantify the amplitude reconstruction accuracy, we performed a simulation using a matrix A with a size of $M \times 12$. We took the values of M to be integers from 3 to 10. The value of M that corresponds to the number of rows in A indicates the compression ratio. A higher M corresponds to a lower compression ratio and therefore we expect better construction accuracy. Figure 5 shows the performance of the three algorithms.

The horizontal axis is the number of samples in the compressed signal $\boldsymbol{y}$, which is equivalent to M. As expected, the accuracy (i.e. a lower MAE) increased as the number of samples in $\boldsymbol{y}$ increased. In this simulation, we observed that our proposed method performed best at low compression ratios (with the number of samples more than 5). As the compression ratio increased, the performance of proposed method decreased and its MAE was higher compared to that of OMP and CVX programming, which indicates that the proposed method performed less well than OMP and CVX programming. This decrease in performance can be seen as the failure of the proposed method to converge to the correct solution as the number of constraints becomes smaller and is no longer sufficient to converge to the correct solution.

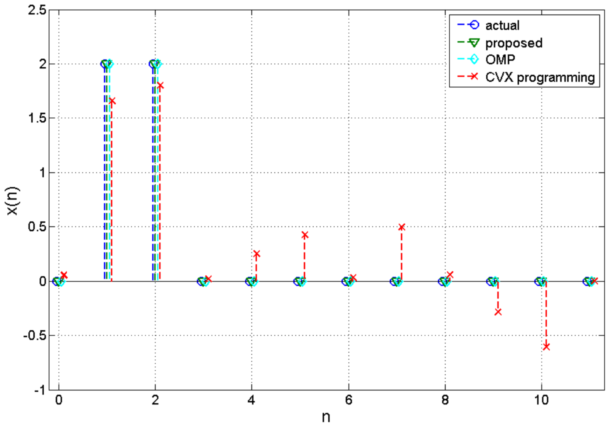


Figure 4 Reconstruction accuracy illustration of the three algorithms. The proposed method and OMP had the best accuracy, while CVX programming had some offset in nonzero and zero elements.

4.2 Performance as Function of Coherency of Sensing Matrix

OMP has been known to perform badly at high coherency of A. The proposed method, on the other hand, is expected to perform better since it does not process the iteration column-wise as in OMP. Instead, it views the reconstruction problem geometrically. This experiment was done to assess the capability of both algorithms and CVX programming as a function of coherency.

We used a similar setup as in the previous experiment (see Section 4.1) but adjusted the third column of A using the second column with the following assignment:

$$a_3 \leftarrow \mu a_2 + (1-\mu)a_3 \qquad (20)$$

In other words, the value of the third column is modified by taking influence from the second column. Here, μ $(0 \leq \mu \leq 1)$ performs as coherency control. Increasing the value of μ corresponds to a higher coherency in A since the third column becomes similar to the second column. In this particular simulation we ran the coherency tests under low and high compression ratios, represented by a 10 × 12 matrix and a 6 × 12 matrix of A respectively.

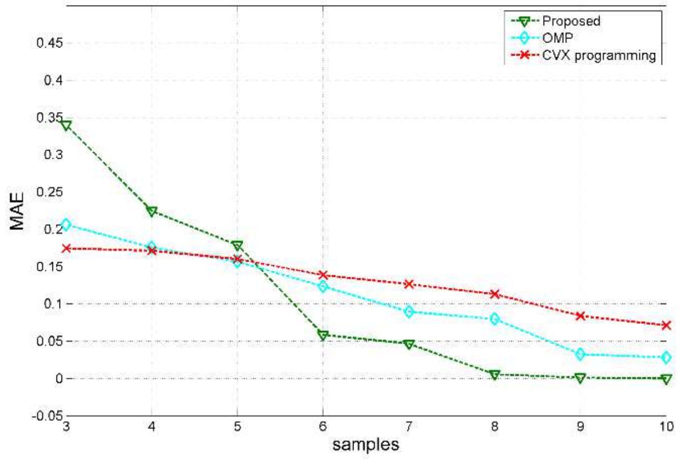


Figure 5 The performance of the three reconstruction methods for amplitude accuracy as indicated by MAE (lower MAE means better accuracy). The proposed method has a better slope and in practice performs best at a number of samples greater than 5.

The simulation results are depicted in Figures. 6(A) and (B). As can be observed in both figures, the proposed method had best accuracy at low to moderately high coherency (less than 0.6). OMP, on the other hand, already suffered at low coherency and got worse at higher values. It is interesting to note that CVX programming performed uniformly, relatively independent to the coherency of $\boldsymbol{A}$. The simulation results show that the proposed method performs better than OMP in the presence of coherency in sensing matrix A at either low or high compression ratios. The proposed method also has better performance compared to the CVX programming method at low to medium value of coherency. However, the proposed method's performance decreases as coherency increases, while CVX programming is not sensitive to coherency, hence, at coherency values greater than 0.8, CVX programming performs better.

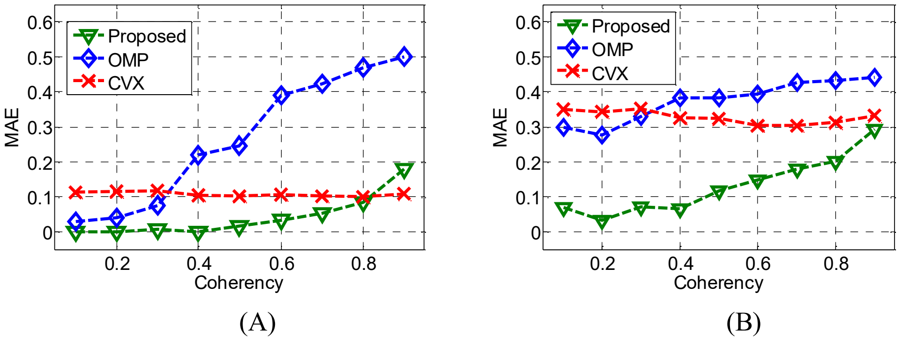


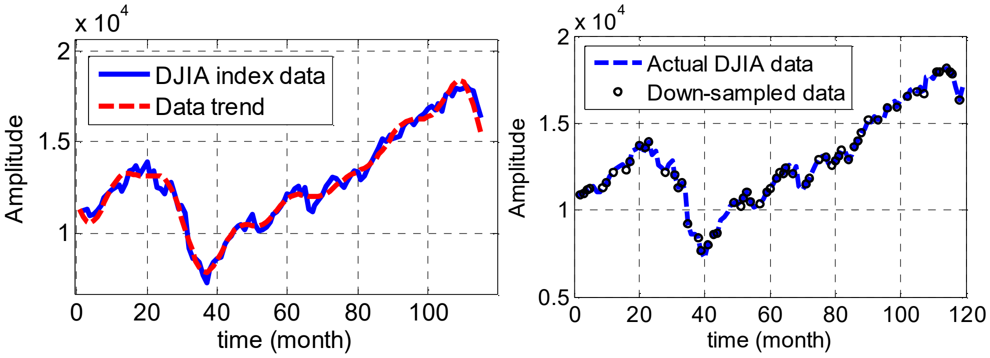
Figure 6 (A) Performance comparison between the three CS reconstruction algorithms as a function of base coherency of matrix A, which is a 10×12 matrix. (B) The simulation results using sensing matrix A, which is a 6×12 matrix.

4.3 Data Interpolation of DJIA Index

The Dow Jones Industrial Average (DJIA) index reflects the average transaction volume on the stock market of thirty large publicly owned companies in the USA. DJIA is one of the most influential index indicators for the US market. Normally, the data are reported at the beginning of every month. We tested the proposed algorithm using the DJIA for the period of January 2006 to October 2015, which corresponds to 119 data. In Figure 7(A), the solid line shows a graph of the DJIA index over this 119-month period.

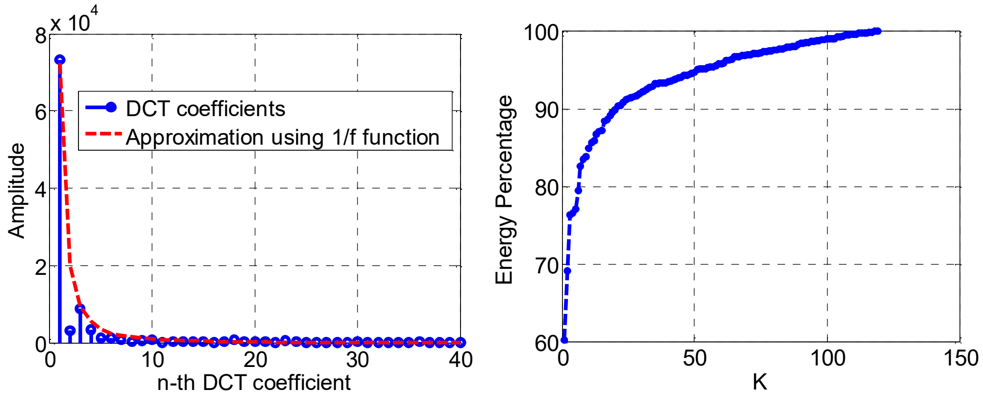
Previous research [29] indicates that the DJIA dataset can be modeled using fractional Brownian motion, which is characterized by the $1/f^H$ function, where f is frequency and H is the Hurst parameter. In other words, the DJIA dataset has decaying frequency components in the frequency domain. This phenomenon suggests that the DJIA dataset is sparse in the frequency domain, thus the CS method can be employed for reconstruction of random data missing in the DJIA index. In Fig. 7(A), the dash line shows the trend data of DJIA using its first ten frequency components. To simulate the missing data, we selected a compression matrix A of size $M \times N$ such that there was only one element of 1 in each row of A, while these values of 1 appeared randomly over the columns of A. By this arrangement, we obtained the compressed y to be a random down-sample of x (Figure 7(B)). For the reconstruction of missing data, we used the discrete cosine transform (DCT) instead of the discrete Fourier transform (DFT) to get real-valued frequency coefficients. Figure 8(A) shows the DCT coefficient plot of DJIA. The decaying trend of the coefficients of this particular dataset can be estimated using α / f^H with α = 640.03 and H = 1.8576 (the dash line in Fig. 8(A)).

Figure 8(B) shows the energy percentage of the DJIA dataset using the first ten DCT coefficients. As can be seen in this figure, the first ten coefficients (*K = 10*) already represent about 85% of the total energy of the signal. The energy contribution of the next coefficients does not significant anymore. For example, at *K = 20*, only about 5% energy is added from the previous *K = 10*.



(A) (B)

Figure 7 (A) The DJIA index from January 2006 to October 2015 (solid line) and its trend using the first ten frequency components (dash line). (B) The original DJIA data and its random down-sampled version using a 1:2 ratio.



(A) (B)

Figure 8 (A) DJIA's DCT coefficients and the 1/f approximation function. (B) The energy percentage of the DJIA dataset using the first ten DCT coefficients.

Using the proposed method, we reconstructed $\boldsymbol{x}$ under the assumption that $\boldsymbol{x}$ consists of only 10 low-frequency components (10 harmonics) and 20 low-frequency components on $\boldsymbol{x}$ (20 harmonics). The reconstruction results are shown in Figure 9. It is interesting to note that both assumptions had values close to the original $\boldsymbol{x}$. The reconstruction using a 10-harmonics assumption, however, had incorrect estimates on $\boldsymbol{x}$ at deep slope values (at time around 40) due to limited frequency components, thus it was unable to reconstruct high-frequency components in the deep slope part of signal. The reconstruction using a 20-harmonics assumption was successful in reconstructing this deep slope.

Figure 10 shows a comparison of the reconstructed data using proposed method, CVX programming, and OMP on the DJIA randomly down-sampled data. In this figure, it can be seen that the proposed method can reconstruct the missing data with a good fit, about similar to the reconstructed data from CVX programming. OMP on the other hand has correct tracking on the data trend, but it consistently deviates from the correct values in the form of spikes around the correct values. The down-sampling ratio in this simulation was 2:3.

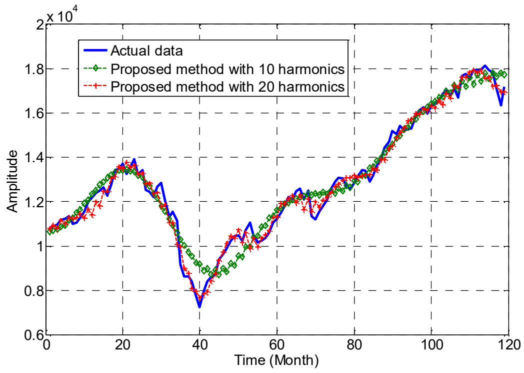


Figure 9 DJIA data and their reconstruction result using the proposed method under 10- and 20-harmonics assumption.

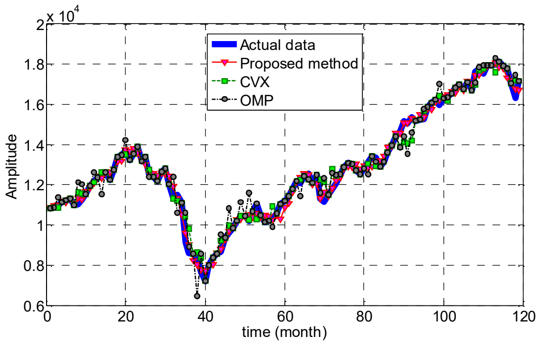


Figure 10 DJIA data and its random down-sampled data reconstruction using the proposed method, CVX programming and OMP.

5 Conclusion

A CS reconstruction algorithm was presented based on a geometrical interpretation of the minimization of l_1-norm in the solution of CS equation $Ax = y$. The basic principle of this method is the fact that a point M that is produced by convex combination of the vertexes of the polytope produced by the intersections of $Ax = y$ and $\|x\|_1 = k_0$ has a norm that is smaller than the previous k on the vertexes. The selection of equal weights for each component in convex combination leads to the physical interpretation that point M is the weight point of the polytope. Unlike OMP, the proposed method has better robustness in high coherency environments. Compared to convex optimization this method offers simpler computation. This method offers more choice when selecting a CS reconstruction algorithm, especially for applications that need a balance between accuracy and speed. Given these advantages, the proposed method works only for real-valued signals. It is necessary to generalize this method beyond the geometrical interpretation so that it provides the capability to solve reconstruction problems of complex-valued signals.

References

[1] Dohono, D., *Compressed Sensing*, IEEE Transactions on Information Theory, **52**(4), pp. 1289-1306, 2006.

[2] Candes, E. & Wakin, M.B., *Compressive Sampling*, Proceedings of the International Congress of Mathematicians, **25**(2), pp. 21-30, 2008.

[3] Candes, E. & Tao, T., *Decoding by Linear Programming*, IEEE Transactions on Information Theory, **51**(12), pp. 4203-4215, 2005.

[4] Baraniuk, R., *Compressive Sensing*, IEEE Signal Processing Magazine, **24**(4), pp. 118-121, Jul, 2007.

[5] Donoho, D. & Stark, P. B., *Uncertainty Principles and Signal Recovery*, Siam J. Appl. Math., **49**(3), pp. 906-931, Jun, 1989.

[6] Chen, S. S., Donoho, D. & Saunders, M.A., *Atomic Decomposition by Basis Pursuit*, SIAM Review, Society for Industrial and Applied Mathematics, **43**(1), pp. 129-159, 2001.

[7] Mallat, S.G. & Zhang, Z., *Matching Pursuits with Time-Frequency Dictionaries*, IEEE Transactions on Signal Processing, **41**(12), 1993.

[8] Chen, Y., Huang, J. & He, C., *High Resolution Direction-of-Arrival Estimation Based on Compressive Sensing with Noval Compression Matrix*, IEEE International Conference on Signal Processing, Communication, and Computing, 2012.

[9] Kim, J.M., Lee, O.K. & Ye, J.C., *Compressive MUSIC: Revisiting the Link Between Compressive Sensing and Array Signal Processing*, IEEE Transactions on Information Theory, **58**(1), 2012.

[10] Usman, K., Gunawan, H. & Suksmono, A.B., *Uniform Non-Exhaustive Search on Sparse Reconstruction for Direction of Arrival Estimation*, IEEE APWiBob, 2015.

[11] Xenakia, A., Gerstoft, P. & Mosegaard, K., *Compressive Beamforming*, J. Acoust. Soc. America, **136**(1), Jul, 2014.

[12] Edelmann, G.F. & Gaumond, C.F., *Beamforming using Compressive Sensing*, J. Acoust. Soc. America, **130**(4), 2011.

[13] Lustig, M., Donoho, D., Santos, J. & Pauly, J., *Compressed Sensing MRI*, IEEE Signal Processing Magazine, **25**(2), pp. 72-82, 2008.

[14] Suksmono, A. B., *Interpolation of PSF based on Compressive Sampling and Its Application in Weak Lensing Survey*, Monthly Notices of the Royal Astronomical Society, **443**(1), pp.919-926, 2014.

[15] Wahidah, I., Hendrawan, T., Mengko, T.L.R. & Suksmono, A.B. *Parameter Estimation for Coefficient Thresholding in Block-based Compressive Video Coding*, International Journal of Imaging and Robotics, **15**(3), pp. 66-72, 2015.

[16] Elad, M. & Bruckstein A.M., *A Generalized Uncertainty Principle and Sparse Representation in Pairs of Bases*, IEEE Transactions on Information Theory, **48**(9), pp. 2558-2567, 2002.

[17] Donoho. D.L. & Huo, X., *Uncertainty Principles and Ideal Atomic Decomposition*, IEEE Transactions on Information Theory, **47**(7), 2001.

[18] Candes, E. & Romberg, J., *l1-Magic: Recovery of Sparse Signals via Convex Programming*, Retrieved on 2 November 2016 from http://users.ece.gatech.edu/~justin/l1magic/, 2005.

[19] Boyd, S. & Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[20] Chen, S., Billings, S.A. & Luo, W., *Orthogonal Least Squares Methods and Their Application to Non-linear System Identification*, International Journal on Control, **50**(5), pp. 1873-1896, 1989.

[21] Pati, Y.C., Rezahfar, R. & Krishnaprasad, P.S., *Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition*, Annual Asilomar Conference on Signals Systems and Computers, **1**(3), 1993.

[22] Needell, D. & Vershynin, R., *Signal Recovery from Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit*, IEEE Journal of Selected Topics in Signal Processing, **4**(2), pp. 310-316, 2010.

[23] Donoho, D., Tsaig, Y., Drori, I. & Starck, J.L., *Sparse Solution of underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit (StOMP)*, IEEE Transactions on Information Theory, **58**(2), 2012

[24] Needell, D. & Tropp, J.A., *CoSaMP: Iterative Signal recovery from Incomplete and Inaccurate Samples*, Appl. Comput. Harmon. Anal., **26**, pp. 301-321, 2008.

[25] Tropp, J.A. & Gilbert, A.C., *Signal Recovery from Random Measurements Via Orthogonal Matching Pursuit*, IEEE Transactions on Information Theory, **53**(12), pp. 4655-4666, 2007.

[26] DeVore, R.A. & Temlyakov, V.N., *Some Remarks on Greedy Algorithms*, Advances in Computational Mathematics, **5**, pp. 173-187, 1996.

[27] Cohen, A., *Approximation by Greedy Algorithms*, Siam News, **40**(8), 2007.

[28] Golub, G.H. & Loan, C.V., *Matrix Computation*, 3rd Edition, The Johns Hopkins University Press, Baltimore, United States, 1996.

[29] Suksmono, A.B., *Reconstruction of Fractional Brownian Motion Signals From Its Sparse Samples Based on Compressive Sampling*, ICEEI, Bandung, Indonesia, July 2011.