

Article

A Regularized Weighted Smoothed L_0 Norm Minimization Method for Underdetermined Blind Source Separation

Linyu Wang, Xiangjun Yin, Huihui Yue * and Jianhong Xiang

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; wanglinyu@hrbeu.edu.cn (L.W.); yinxiangjun@hrbeu.edu.cn (X.Y.); xiangjianhong@hrbeu.edu.cn (J.X.)

* Correspondence: yuehuihui@hrbeu.edu.cn

Received: 22 October 2018; Accepted: 30 November 2018; Published: 4 December 2018



Abstract: Compressed sensing (CS) theory has attracted widespread attention in recent years and has been widely used in signal and image processing, such as underdetermined blind source separation (UBSS), magnetic resonance imaging (MRI), etc. As the main link of CS, the goal of sparse signal reconstruction is how to recover accurately and effectively the original signal from an underdetermined linear system of equations (ULSE). For this problem, we propose a new algorithm called the weighted regularized smoothed L_0 -norm minimization algorithm (WReSL0). Under the framework of this algorithm, we have done three things: (1) proposed a new smoothed function called the compound inverse proportional function (CIPF); (2) proposed a new weighted function; and (3) a new regularization form is derived and constructed. In this algorithm, the weighted function and the new smoothed function are combined as the sparsity-promoting object, and a new regularization form is derived and constructed to enhance de-noising performance. Performance simulation experiments on both the real signal and real images show that the proposed WReSL0 algorithm outperforms other popular approaches, such as SL0, BPDN, NSL0, and L_p -RLS and achieves better performances when it is used for UBSS.

Keywords: image reconstruction; nullspace measurement matrix; regularized least squares problem; smoothed L_0 -norm; sparse signal recovery; UBSS; weighted function

1. Introduction

The problem that UBSS [1,2] needs to address is how to separate multiple signals from a small number of sensors. The essence of this problem is to solve the optimal solution of the undetermined linear system of equations (ULSE). Fortunately, as a new undersampling technique, compressed sensing (CS) [3–5] is an effective way to solve ULSE, which makes it possible to apply CS to UBSS.

The model of CS is shown in Figure 1. According to this figure, it can be seen that CS boils down to the form,

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{b}, \quad (1)$$

where $\Phi = [\phi_1, \phi_2, \dots, \phi_n] \in \mathbb{R}^{m \times n}$ is a sensing matrix with the condition of $m \ll n$ and $\phi_i \in \mathbb{R}^m$, $i = 1, 2, \dots, n$, which can be further represented as $\Phi = \psi \varphi$, while ψ is a random matrix, and φ is the sparse basis matrix. $\mathbf{y} \in \mathbb{R}^m$ is the vector of measurements. Moreover, $\mathbf{b} \in \mathbb{R}^m$ denotes the additive noise.

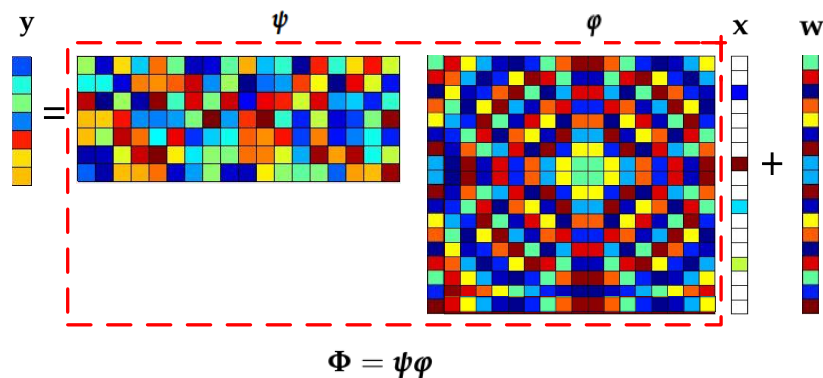


Figure 1. Frame of compressed sensing (CS).

To solve the ULSE in Equation (1), we try to recover the sparse signal \mathbf{x} from the given $\{\mathbf{y}, \Phi\}$ by CS. According to CS, this problem is transformed into solving the L_0 -norm minimization problem.

$$(P_0) \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_0, \text{ s.t. } \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon. \quad (2)$$

where ϵ denotes error. This rather wonderful attempt is actually supported by a brilliant theory [6]. Based on this theory, in the noiseless case, it is proven that the sparsest solution is indeed a real signal when \mathbf{x} is sufficiently sparse and Φ satisfies the restricted isometry property (RIP) [7]:

$$1 - \delta_K \leq \frac{\|\Phi \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \leq 1 + \delta_K, \quad (3)$$

where K is the sparsity of signal \mathbf{x} and $\delta_K \in (0, 1)$ is a constant. In Equation (2), the L_0 -norm is nonsmooth, which leads an NP-hard problem. In practice, two alternative approaches are usually employed to solve the problem [8]:

- Greedy search by using the known sparsity as a constraint;
- The relaxation method for the P_0 .

For greedy search, the main methods are based on greedy matching pursuit (GMP) algorithms, such as orthogonal matching pursuit (OMP) [9,10], stage-wise orthogonal matching pursuit (StOMP) [11], regularized orthogonal matching pursuit (ROMP) [12], compressive sampling matching pursuit (CoSaMP) [13], generalized orthogonal matching pursuit (GOMP) [14,15], and subspace pursuit (SP) [16,17] algorithms. The objective function of these algorithms is given by:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2, \text{ s.t. } \|\mathbf{x}\|_0 \leq K. \quad (4)$$

As shown in the above equation, the features of GMP algorithms can be concluded as:

- Using sparsity as prior information;
- Using the least squares error as the iterative criterion.

The advantage of GMP algorithms is that the computational complexity is low, but the reconstruction accuracy is not high in the noise case.

At present, the relaxation method for P_0 is widely used. The relaxation method is mainly divided into two categories: the constraint-type algorithm and the regularization method. The constraint-type algorithm can also be divided into L_1 -norm minimization methods and smoothed L_0 -norm

minimization methods. The representative algorithm of the former is the BPalgorithm [18], and the latter is the smoothed L_0 -norm minimization (SL0) algorithm. For the SL0 algorithm, the objective function can be expressed as:

$$(P_F) \arg \min_{\mathbf{x} \in \mathbb{R}^n} F_\sigma(\mathbf{x}), \quad \text{s.t. } \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon. \quad (5)$$

$$\lim_{\sigma \rightarrow 0} F_\sigma(\mathbf{x}) = \lim_{\sigma \rightarrow 0} \sum_{i=1}^n f_\sigma(x_i) \approx \|\mathbf{x}\|_0.$$

where $F_\sigma(\mathbf{x})$ is a smoothed function, which approximates the L_0 -norm when $\sigma \rightarrow 0$. Compared with L_1 or L_p , a small σ is selected to make the function close to L_0 -norm [8]; therefore, $F_\sigma(\mathbf{x})$ are closer to the optimal solution.

Based on the idea of approximation, Mohimani used a Gauss function to approximate the L_0 -norm [19], which is described as:

$$f_\sigma(x_i) = 1 - \exp\left(-\frac{x_i^2}{2\sigma^2}\right). \quad (6)$$

According to the equation, we can know:

$$f_\sigma(x_i) \approx \begin{cases} 1 & \text{if } x_i \gg \sigma \\ 0 & \text{if } x_i \ll \sigma. \end{cases} \quad (7)$$

when σ is a small enough positive value, the Gauss function is almost equal to the L_0 -norm. Furthermore, the Gauss function is differentiable and smoothed; hence, it can be optimized by optimization methods such as the gradient descent (GD) method. Zhao proposed another smoothed function: the hyperbolic tangent (tanh) [20],

$$f_\sigma(x_i) = \frac{\exp\left(\frac{x_i^2}{2\sigma^2}\right) - \exp\left(-\frac{x_i^2}{2\sigma^2}\right)}{\exp\left(\frac{x_i^2}{2\sigma^2}\right) + \exp\left(-\frac{x_i^2}{2\sigma^2}\right)}. \quad (8)$$

This smoothed function makes a closer approximation to the L_0 -norm than the Gauss function, as shown in [19], with the same σ ; hence, it performs better in sparse signal recovery. Indeed, a large number of simulation experiments confirmed this view.

Another relaxation method is the regularization method. For CS, sparse signal recovery in the noise case is a very practical and unavoidable problem. Fortunately, the regularization method makes the solution of this problem possible [21,22]. The regularization method can be described as a "relaxation" approach that tries to solve the following unconstrained recovery problem:

$$(P_v) \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \lambda v(\mathbf{x}), \quad (9)$$

where $\lambda > 0$ is the parameter that balances the trade-off between the deviation term $\|\Phi \mathbf{x} - \mathbf{y}\|_2^2$ and the sparsity regularizer $v(\mathbf{x})$. The sparse prior information is enforced via the regularizer $v(\mathbf{x})$, and a proper $v(\mathbf{x})$ is crucial to the success of the sparse signal recovery task: it should favor sparse solutions and make sure the problem P_v can be solved efficiently in the meantime.

For regularization, various sparsity regularizers have been proposed as the relaxation of the L_0 -norm. The most popular algorithms are the convex L_1 -norm [22,23] and the nonconvex L_p -norm to the p^{th} power [24,25]. In the noiseless case, the L_1 -norm is equivalent to the L_0 -norm, and the L_1 -norm is the only norm with sparsity and convexity. Hence, it can be optimized by convex optimization methods. However, according to [8], in the noisy case, the L_1 -norm is not exactly equivalent to the L_0 -norm, so the effect of promoting sparsity is not obvious. Compared to the L_1 -norm, the nonconvex L_p -norm

to the p^{th} power makes a closer approximation to the L_0 -norm; therefore, L_p -norm minimization has a better sparse recovery performance [8].

In view of the above explanation, in this paper, a compound inverse proportional function (CIPF) function is proposed as a new smoothed function, and a new weighted function is proposed to promote sparsity. For the noise case, a new regularization form is derived and constructed to enhance de-noising performance. The experimental simulation verifies the superior performance of this algorithm in signal and image recovery, and it has achieved good results when applied to UBSS.

This paper is organized as follows: Section 2 introduces the main work of this paper. The steps of the ReRSL0 algorithm and the selection of related parameters are described in Section 3. Experimental results are presented in Section 4 to evaluate the performance of our approach. Section 5 verifies the effect of the proposed weighted regularized smoothed L_0 -norm minimization (WRReSL0) algorithm in UBSS. Section 6 concludes this paper.

2. Main Work of This Paper

In this paper, based on the P_F in Equation (9), we propose a new objective function, which is given by:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{W}H_\sigma(\mathbf{x}), \quad \text{s.t. } \|\Phi\mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon. \quad (10)$$

According to this equation, We not only propose a smoothed function approximating the L_0 -norm, but also propose a weighted function to promote sparsity. This section focuses on the relevant contents of $\mathbf{W} = [w_1, w_2, \dots, w_n]^T$ and $H_\sigma(\mathbf{x})$.

2.1. New Smoothed Function: CIPF

According to [26], some properties of the smoothed functions are summarized in the following:

Property: Let $f : \mathbb{R} \rightarrow [-\infty, +\infty]$ and, define $f_\sigma(r) \approx f_\sigma(r/\sigma)$ for any $\sigma > 0$. The function f has the property, if:

- f is real analytic on (r_0, ∞) for some r_0 ;
- $\forall r \geq 0, f''(r) \geq -\epsilon_0$, where $\epsilon_0 > 0$ is some constant;
- f is convex on \mathbb{R} ;
- $f(r) = 0 \leftrightarrow r = 0$;
- $\lim_{r \rightarrow +\infty} f(r) = 1$.

It follows immediately from **Property** that $\{f_\sigma(r)\}$ converges to the L_0 -norm as $\sigma \rightarrow 0^+$, i.e.,

$$\lim_{\sigma \rightarrow 0^+} f_\sigma(r) = \begin{cases} 0 & \text{if } r = 0 \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

Based on **Property**, this paper proposes a new smoothed function model called CIPF, which satisfies **Property** and better approximates the L_0 -norm. The smoothed function model is given as:

$$f_\sigma(r) = 1 - \frac{\sigma^2}{\alpha r^2 + \sigma^2}. \quad (12)$$

In Equation (12), α denotes a regularization factor, which is a large constant. By experiments, the factor α is determined to be 10, which is a good result of the simulation. σ represents a smoothed factor, and when it is smaller, it will make the proposed model closer to the L_0 -norm. Obviously,

$\lim_{\sigma \rightarrow 0} f_\sigma(r) = \begin{cases} 0, & r = 0 \\ 1, & r \neq 0 \end{cases}$ or approximately $f_\sigma(r) \approx \begin{cases} 0, & |r| \ll \sigma \\ 1, & |r| \gg \sigma \end{cases}$ is satisfied. Let:

$$H_\sigma(\mathbf{x}) = \sum_{i=1}^n f_\sigma(x_i) = n - \sum_{i=1}^n \frac{\sigma^2}{\alpha x_i^2 + \sigma^2} \quad (13)$$

where $H_\sigma(\mathbf{x}) \approx \|\mathbf{x}\|_0$ for small values of σ , and the approximation tends to equality when $\sigma \rightarrow 0$.

Figure 2 shows the effect of the CIPF model approximating the L_0 -norm. Obviously, the CIPF model makes a better approximation.

In conclusion, the merits of the CIPF model can be summarized as follows:

- It closely approximates the L_0 -norm;
- It is simpler in form than that in the Gauss and tanh function models.

These merits make it possible to reduce the computational complexity on the premise of ensuring the accuracy of sparse signal reconstruction, which is of practical significance for sparse signal reconstruction.

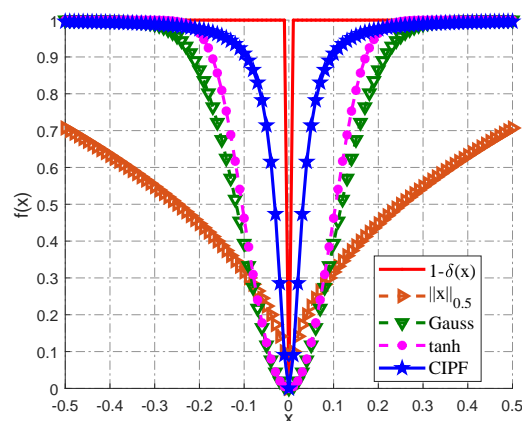


Figure 2. Different functions used in the literature to approximate the L_0 -norm; some of them are plotted in this figure, and the $L_{0.5}$ -norm is displayed for comparison. CIPF, compound inverse proportional function.

2.2. New Weighted Function

Candès et al. [27] proposed the weighted L_1 -norm minimization method, which employs the weighted norm to enhance the sparsity of the solution. They provided an analytical result of the improvement in the sparsity recovery by incorporating the weighted function with the objective function. Pant et al. [28] applied another weighted smoothed L_0 -norm minimization method, which uses a similar weighted function to promote sparsity. The weighted function can be summarized as follows:

- Candès et al.: $w_i = \begin{cases} \frac{1}{|x_i|} & x_i \neq 0 \\ \infty & x_i = 0 \end{cases}$;
- Pant et al.: $w_i = \frac{1}{|x_i| + \zeta}$, ζ is a small enough positive constant.

From the two weighted functions, we can find a phenomenon: a large signal entry x_i is weighted with a small w_i ; on the contrary, a small signal entry x_i is weighted with a large value w_i . By analysis, the large w_i forces the solution \mathbf{x} to concentrate on the indices where w_i is small, and by construction, these correspond precisely to the indices where \mathbf{x} is nonzero.

Combined with the above idea, we propose a new weighted function, which is given by:

$$w_i = e^{-\frac{|x_i|}{\sigma}}, \quad \text{s.t. } i = 1, 2, \dots, n. \quad (14)$$

As for Candès et al., when the signal entry is zero or close to zero, the value of w_i will be very large, which is not suitable for computation by a computer. Although Pant et al. noticed the problem and improved the weighted function to avoid it, the constant ζ depends on experience. Actually, the proposed weighted function can avoid the two problems. Moreover our weighted function can

be satisfied with the phenomenon. When the small signal entry x_i can be weighted with a large w_i and a large signal entry x_i can be weighted with a small w_i , this can make the large signal entry and small signal entry closer. In this way, the direction of optimization can be kept as consistent as possible, and the optimization process tends to be more optimal. Therefore, the proposed weighted function can have a better effect.

3. New Algorithm for CS: WReSL0

3.1. WReSL0 Algorithm and Its Steps

Here, in order to analyze the problem more clearly, we rewrite Equation (10) as follows:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{W}H_\sigma(\mathbf{x}), \quad \text{s.t. } \|\Phi\mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon.$$

where $H_\sigma(\mathbf{x}) = \mathbf{I} - \frac{\sigma^2}{\alpha\mathbf{x}^2 + \sigma^2}$ ($\mathbf{I} \in \mathbb{R}^N$ is a unit vector) is a differentiable smoothed accumulated function. The weighted function $\mathbf{W} = e^{-\frac{|\mathbf{x}|}{\sigma}}$. Therefore, we can obtain the gradient of CIPF, which is written as:

$$\mathbf{G} = \frac{\partial H_\sigma(\mathbf{x})}{\partial \mathbf{x}} = \frac{2\alpha\sigma^2\mathbf{x}}{(\alpha\mathbf{x}^2 + \sigma^2)^2} \quad (15)$$

According to Equation (15), as in [28], we can obtain:

$$\mathbf{W}\mathbf{G} = \left(e^{-\frac{|\mathbf{x}|}{\sigma}} \right)^T \frac{2\alpha\sigma^2\mathbf{x}}{(\alpha\mathbf{x}^2 + \sigma^2)^2} \quad (16)$$

Solving the problem of ULSE is to solve the optimization problem in Equation (10). As for this problem, there are many methods, such as split Bregman methods [29–31], FISTA [32], alternating direction methods [33], gradient descent (GD) [34], etc. In order to reduce the computational complexity, this paper adopts the GD method to optimize the proposed objective function.

Given σ , a small target value σ_{\min} , and a sufficiently large initial value σ_{\max} , after referring to the annealing mechanism in simulated annealing [35], this paper proposes a monotonically-decreasing sequence $\{\sigma_t | t = 2, 3, \dots, T\}$, which is generated as:

$$\sigma_t = \sigma_{\max}\theta^{-\gamma(t-1)}, \quad \text{s.t. } t = 1, 2, 3, \dots, T. \quad (17)$$

where $\gamma = \frac{\log_\theta(\sigma_{\max}/\sigma_{\min})}{T-1}$, θ is a constant that is larger than one, and T is the maximum number of iterations. Using such a monotonically-decreasing sequence can avoid the case of too small of a σ leading to the local optimum.

Similar to SL0, WReSL0 also consists of two nested iterations: the external loop, which begins with a sufficiently large value of σ , i.e., σ_{\max} , responsible for the gradually decreasing strategy in Equation (17), and the internal loop, which for each value of σ , finds the maximizer of $H_\sigma(\mathbf{x})$ on $\{\mathbf{x} | \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon\}$.

According to the GD algorithm, the internal loop consists of the gradient descent step, which is given by:

$$\hat{\mathbf{x}} = \mathbf{x} + \mu\mathbf{d}, \quad (18)$$

where $\mathbf{d} = \mathbf{g}$ and μ denotes a step size factor. This part is similar to SL0, followed by solving the problem:

$$\arg \min_{\mathbf{x}^* \in \mathbb{R}^n} \|\mathbf{x}^* - \hat{\mathbf{x}}\|_2^2, \quad \text{s.t. } \|\Phi\mathbf{x}^* - \mathbf{y}\|_2^2 \leq \epsilon \quad (19)$$

where \mathbf{x}^* denotes the optimal solution. By regularization, this form can be converted to another form as follows,

$$\arg \min_{\mathbf{x}^* \in \mathbb{R}^n} \|\mathbf{x}^* - \hat{\mathbf{x}}\|_2^2 + \lambda \|\Phi \mathbf{x}^* - \mathbf{y}\|_2^2. \quad (20)$$

where λ is the regularization parameter, which is adapted to balance the fit of the solution to the data \mathbf{y} and the approximation of the solution to the maximizer of $H_\sigma(\mathbf{x})$. Weighted least squares (WLS) can be used to solve this problem, and the solution is:

$$\mathbf{x}^* = \left[\begin{array}{c} \mathbf{I}_n \\ \Phi \end{array} \right]^H \left[\begin{array}{cc} \mathbf{I}_n & 0 \\ 0 & \lambda \mathbf{I}_m \end{array} \right] \left[\begin{array}{c} \mathbf{I}_n \\ \Phi \end{array} \right]^{-1} \left[\begin{array}{c} \mathbf{I}_n \\ \Phi \end{array} \right]^H \left[\begin{array}{cc} \mathbf{I}_n & 0 \\ 0 & \lambda \mathbf{I}_m \end{array} \right] \left[\begin{array}{c} \hat{\mathbf{x}} \\ \mathbf{y} \end{array} \right]. \quad (21)$$

By calculation, Equation (21) is equivalent to:

$$\mathbf{x}^* = (\mathbf{I}_n + \lambda \Phi^H \Phi)^{-1} (\hat{\mathbf{x}} + \lambda \Phi^H \mathbf{y}) \quad (22)$$

where \mathbf{I}_n and \mathbf{I}_m are both identity matrices of size $n \times n$ and $m \times m$, respectively. Therefore, we can obtain:

$$\begin{aligned} \mathbf{x}^* - \hat{\mathbf{x}} &= (\mathbf{I}_n + \lambda \Phi^H \Phi)^{-1} (\hat{\mathbf{x}} + \lambda \Phi^H \mathbf{y}) - \hat{\mathbf{x}} \\ &= (\mathbf{I}_n + \lambda \Phi^H \Phi)^{-1} (\hat{\mathbf{x}} + \lambda \Phi^H \mathbf{y} - (\mathbf{I}_n + \lambda \Phi^H \Phi) \hat{\mathbf{x}}) \\ &= (\mathbf{I}_n + \lambda \Phi^H \Phi)^{-1} (\hat{\mathbf{x}} + \lambda \Phi^H \mathbf{y} - \hat{\mathbf{x}} - \lambda \Phi^H \Phi \hat{\mathbf{x}}) \\ &= -(\lambda^{-1} \mathbf{I}_n + \Phi^H \Phi)^{-1} \Phi^H (\Phi \hat{\mathbf{x}} - \mathbf{y}) \end{aligned}$$

According to the above analysis and derivation, we can get:

$$\mathbf{x}^* = \hat{\mathbf{x}} - (\lambda^{-1} \mathbf{I}_n + \Phi^H \Phi)^{-1} \Phi^H (\Phi \hat{\mathbf{x}} - \mathbf{y}) \quad (23)$$

The initial value of the internal loop is the maximizer of $H_\sigma(\mathbf{x})$ obtained for σ_{\max} . To increase the speed, the internal loop is repeated a fixed and small number of times (L). In other words, we do not wait for the GD method to converge in the internal loop.

According to the explanation above, we can conclude the steps of the proposed WReSL0 algorithm, which are given in Table 1. As for σ , it can be shown that function $H_\sigma(\mathbf{x})$ remains convex in the region where the largest magnitude of the component of \mathbf{x} is less than σ . As the algorithm starts at the original value $\mathbf{x}^{(0)} = \Phi^H (\Phi \Phi^H)^{-1} \mathbf{y}$, the above choice of σ_1 ensures that the optimization starts in a convex region. This greatly facilitates the convergence of the WReSL0 algorithm.

Table 1. Weighted regularized smoothed L_0 -norm minimization (WReSL0) algorithm using the GD method.

-
- Initialization:
 - (1) Set $L, \mu = \sigma / (2\alpha), \hat{\mathbf{x}}^{(0)} = \Phi^H (\Phi \Phi^H)^{-1} \mathbf{y}$.
 - (2) Set $\sigma_{\max} = \sqrt{\alpha} \max |\mathbf{x}|, \sigma_{\min} = 0.01$, and $\sigma_t = \sigma_{\max} \theta^{-\gamma(t-1)}$, where $\gamma = \frac{\log_\theta(\sigma_{\max}/\sigma_{\min})}{T-1}$, and T is the maximum number of iterations.
 - while $t < T$, do
 - (1) Let $\sigma = \sigma_t$.
 - (2) Let $\mathbf{x} = \hat{\mathbf{x}}^{(t-1)}$.
 - for $l = 1, 2, \dots, L$
 - (a) $\mathbf{x} \leftarrow \mathbf{x} - \mu \left(e^{-\frac{|\mathbf{x}|}{\sigma}} \right)^T \frac{2\alpha\sigma^2\mathbf{x}}{(\alpha\mathbf{x}^2 + \sigma^2)^2}$.
 - (b) $\mathbf{x} \leftarrow \mathbf{x} - (\lambda^{-1} \mathbf{I}_n + \Phi^H \Phi)^{-1} \Phi^H (\Phi \hat{\mathbf{x}} - \mathbf{y})$
 - (3) Set $\hat{\mathbf{x}}^{(t-1)} = \mathbf{x}$.
 - The estimated value is $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(t)}$.
-

3.2. Selection of Parameters

The selection of parameters μ and σ will affect the performance of the WReSL0 algorithm; thus, this paper discusses the selection of these two above parameters in this section.

3.2.1. Selection of Parameter μ

According to the algorithm, each iteration consists of a descent step $x_i \leftarrow x_i - \mu \left(e^{-\frac{|x_i|}{\sigma}} \right) \frac{2\alpha\sigma^2 x_i}{(\alpha x_i^2 + \sigma^2)^2}$, $1 \leq i \leq n$, followed by a projection step. If for some values of i , we have $|x_i| \gg \sigma$, then the algorithm does not change the value of x_i in that descent step; however, it might be changed in the projection step. If we are looking for a suitably large μ , a suitable choice is to make the algorithm force all those values of x satisfying $|x_i| \lesssim \sigma$ toward zero. Therefore, we can get:

$$x_i - \mu \left(e^{-\frac{|x_i|}{\sigma}} \right) \frac{2\alpha\sigma^2 x_i}{(\alpha x_i^2 + \sigma^2)^2} \approx 0 \quad (24)$$

and:

$$\left(e^{-\frac{|x_i|}{\sigma}} \right) \xrightarrow{x_i \rightarrow 0} 1 \quad (25)$$

Combining Equations (24) and (25), we can further obtain:

$$x_i - \mu \frac{2\alpha\sigma^2 x_i}{(\alpha x_i^2 + \sigma^2)^2} \approx 0 \quad (26)$$

By calculation, we can obtain:

$$\mu \approx \frac{(\alpha x_i^2 + \sigma^2)^2}{2\alpha\sigma^2} \xrightarrow{x_i \rightarrow 0} \frac{\sigma^2}{2\alpha} \quad (27)$$

According to the above derivation, we have come to the conclusion that $\mu \approx \frac{\sigma^2}{2\alpha}$. Therefore, we can set $\mu = \frac{\sigma^2}{2\alpha}$.

3.2.2. Selection of Parameter σ

According to Equation (17), the descending sequence of σ is generated by $\sigma_t = \sigma_{\max} \left(\frac{\sigma_{\min}}{\sigma_{\max}} \right)^{\frac{t-1}{T-1}}$ (it is obtained through simplification of Equation (17)). Parameter σ_{\min} and parameter σ_{\max} should be appropriately selected. The selection of σ_{\min} and σ_{\max} is discussed below.

For the initial value of σ , i.e., σ_{\max} , here, let $\tilde{x} = \max\{|\mathbf{x}|\}$; suppose there is a constant b , in order to make the algorithm converge quickly; let parameter σ_{\max} satisfy:

$$H_{\sigma}(\tilde{x}) = 1 - \frac{\sigma_{\max}^2}{\alpha \tilde{x}^2 + \sigma_{\max}^2} \leq b \Rightarrow \sigma_{\max} \geq \left(\sqrt{\frac{1-b}{b}} \alpha \right) \tilde{x}. \quad (28)$$

From the equation, we can see that constant b satisfies $\frac{1-b}{b} \geq 0$; thus $0 < b \leq 1$, and here, we define constant b as 0.5. Hence, $\sigma_{\max} = \sqrt{\alpha} \max\{|\mathbf{x}|\}$.

For the final value σ_{\min} , when $\sigma_{\min} \rightarrow 0$, $H_{\sigma_{\min}}(\mathbf{x}) \rightarrow \|\mathbf{x}\|_0$. That is, the smaller σ_{\min} , the more $H_{\sigma_{\min}}(\mathbf{x})$ can reflect the sparsity of signal \mathbf{x} , but at the same time, it is also more sensitive to noise; therefore, the value σ_{\min} should not be too small. Combining [19], we choose $\sigma_{\min} = 0.01$.

4. Performance Simulation and Analysis

The numerical simulation platform is MATLAB 2017b, which is installed on a computer with a Windows 10, 64-bit operating system. The CPU of the simulation computer is the Intel (R) Core (TM)

i5-3230M, and the frequency is 2.6 GHz. In this section, the performance of the WReSL0 algorithm is verified by signal and image recovery in the noise case.

Here, some state-of-the-art algorithms are selected for comparison. The parameters are selected to obtain the best performance for each algorithm: for the BPDN algorithm [36], the regularization parameter $\lambda = \sigma_N \sqrt{2 \log(n)}$; for the SL0 algorithm [19], the initial value of smoothed factor $\delta_{\max} = 2 \max\{|x|\}$, the final value of smoothed factor $\delta_{\min} = 0.01$, scale factor is set as step size $L = 5$, and the attenuation factor $\rho = 0.8$; for the NSL0 algorithm [20], the initial value of smoothed factor $\delta_{\max} = 4 \max\{|x|\}$, the final value of smoothed factor $\delta_{\min} = 0.01$, the step size $L = 10$, and the attenuation factor $\rho = 0.8$; for L_p -RLS algorithm [24], the number of iterations $T = 80$, the norm initial value $p_1 = 1$, the norm final value $p_T = 0.1$, the initial value of regularization factor $\epsilon_1 = 1$, the final value of regularization factor $\epsilon_T = 0.01$, and the algorithm termination threshold $E_t = 10^{-25}$; for the WReSL0 algorithm, the initial value of smoothed factor $\sigma_{\max} = \sqrt{c} \max\{|x|\}$, the final value of smoothed factor $\sigma_{\min} = 0.01$, the iterations $T = 30$, the step size $L = 5$, and the regularization parameter $\lambda = 0.1$. All experiments are based on 100 trials.

4.1. Signal Recovery Performance in the Noise Case

In this part, we discuss signal recovery performance in the noise case. We add noise \mathbf{b} to the measurement vector \mathbf{y} ; moreover, $\mathbf{b} = \delta_N \mathbf{\Omega}$, $\mathbf{\Omega}$ is randomly formed and follows the Gaussian distribution of $\mathcal{N}(0, 1)$. For signal recovery under noise conditions, we evaluate the performance of algorithms by the normalized mean squared error (NMSE) and the CPU running time (CRT). NMSE is defined as $\|x - \hat{x}\|_2 / \|x\|_2$. CRT is measured with *tic* and *toc*. In order to analyze the de-noising performance of the WReSL0 algorithm in context closer to the real situation, we constructed a certain signal as an experimental object in the experiments in this section. The signal is given by:

$$\begin{cases} x_1 = \alpha_1 \sin(2\pi f_1 T_s t) \\ x_2 = \beta_1 \cos(2\pi f_2 T_s t) \\ x_3 = \alpha_2 \sin(2\pi f_3 T_s t) \\ x_4 = \beta_2 \cos(2\pi f_4 T_s t) \\ \mathcal{X} = x_1 + x_2 + x_3 + x_4 \end{cases} \quad (29)$$

where $\alpha_1 = 0.2$, $\alpha_2 = 0.1$, $\beta_1 = 0.3$, and $\beta_2 = 0.4$. $f_1 = 50$ Hz; $f_2 = 100$ Hz; $f_3 = 200$ Hz; and $f_4 = 300$ Hz. Here, \mathbf{t} is a sequence with $\mathbf{t} = [1, 2, 3, \dots, n]$, and T_s is sampling interval with the value of $\frac{1}{f_s}$. f_s is the sampling frequency with the value of 800 Hz. The object that needs to be reconstructed can be expressed as:

$$\mathbf{y} = \mathbf{\Phi} \mathbf{x} + \delta_N \mathbf{\Omega}. \quad (30)$$

where $\mathbf{x} \in \mathbb{R}^n$ is a sparse signal in the frequency domain, and it is the Fourier transform expression of \mathcal{X} , $\mathbf{y} \in \mathbb{R}^m$. Here, let $n = 128$, $m = 64$. Moreover, $\mathbf{\Phi}$ can be represented as $\mathbf{\Phi} = \mathbf{\psi} \mathbf{\phi}$; here, $\mathbf{\psi}$ is a random matrix generated by a Gaussian distribution, and $\mathbf{\phi}$ is a sparse basis matrix generated by Fourier transform. Here, $\mathbf{\phi}$ can be given by Fourier $\mathbf{I}_{n \times n}$, and $\mathbf{I}_{n \times n}$ is a unit matrix. This target signal \mathcal{X} is sparse in Fourier space; hence, the signal \mathcal{X} can be recovered from given $\{\mathbf{y}, \mathbf{\Phi}\}$ by CS recovery methods.

Figure 3 shows the signal recovery effect. Obviously, BPDN and SL0 do not perform well, while NSL0, L_p -RLS and the proposed WReSL0 perform quite well. This verifies that the regularization mechanism has a good de-noising effect. Figure 4 shows the frequency spectrum of the recovered signal by the selected algorithms. The spectrum of the signal recovered by our proposed WReSL0 algorithm is almost the same as the original signal, while other algorithms fail to achieve this effect.

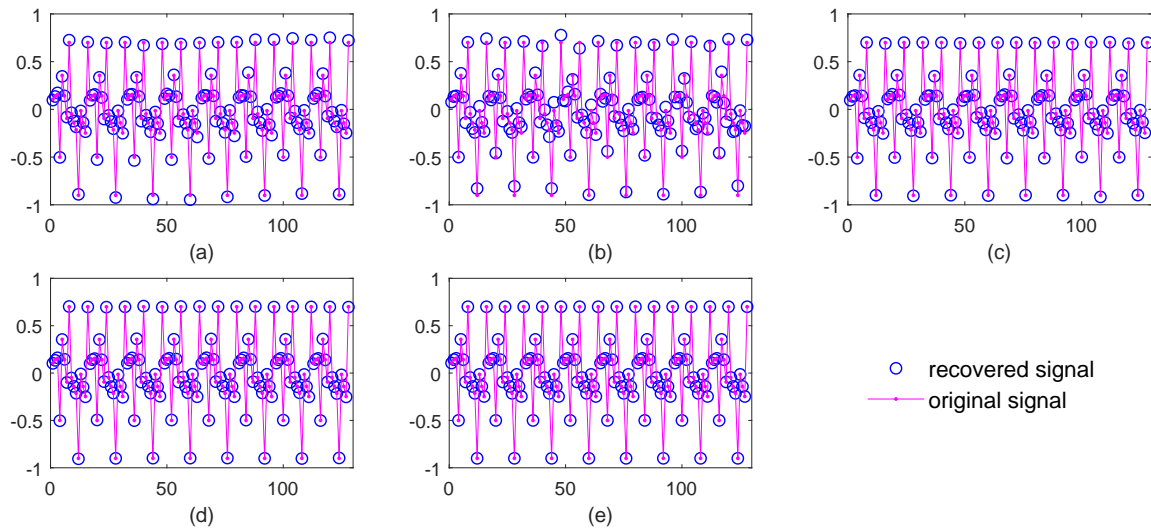


Figure 3. Signal recovery effect by BPDN, SL0, NSL0, L_p -RLS, and weighted regularized smoothed L_0 -norm minimization (WReSL0) when noise intensity $\delta_N = 0.2$. (a) signal recovery by the BPDN algorithm; (b) signal recovery by the SL0 algorithm; (c) signal recovery by NSL0 algorithm; (d) signal recovery by the L_p -RLS algorithm; (e) signal recovery by the WReSL0 algorithm.

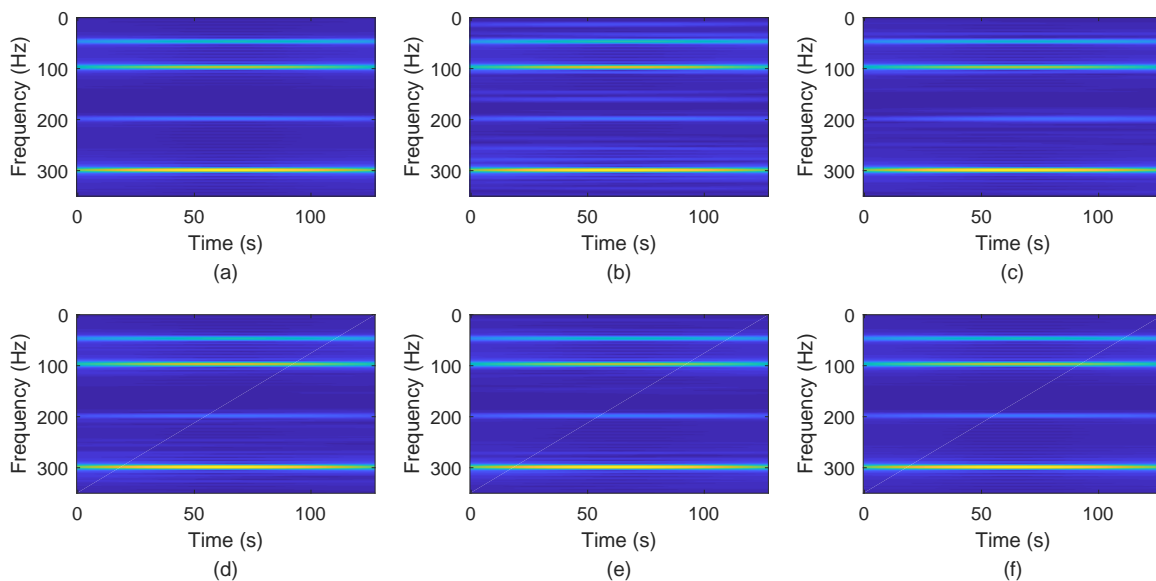


Figure 4. Frequency spectrum analysis of the original signal and the signal recovered by BPDN, SL0, NSL0, L_p -RLS, and WReSL0 when noise intensity $\delta_N = 0.2$. (a) original signal; (b) signal recovery by the BPDN algorithm; (c) signal recovery by the SL0 algorithm; (d) signal recovery by the NSL0 algorithm; (e) signal recovery by the L_p -RLS algorithm; (f) signal recovery by the WReSL0 algorithm.

Table 2 shows the CRT of all algorithms. The n changes according to a given sequence [170, 220, 270, 320, 370, 420, 470, 520]. From the table, for any n , SL0 has the shortest computation time, followed by WReSL0, NSL0, and L_p -RLS, and BPDN has the longest computation time. The BPDN algorithm is generally implemented by the quadratic programming method, and the computational complexity of this method is very high, thus resulting in a large increase in the overall computation time of the algorithm. Furthermore, in L_p -RLS, the iterative process adopts the conjugate

gradient method with high complexity, while NSL0 and WReSL0 do not. Compared with NSL0, WReSL0 is more prominent in the decrease of computation time.

Table 2. Signal CPU running time (CRT) analysis for BPDN, SL0, NSL0, L_p -RLS, and the proposed WReSL0 with signal length changes according to the sequence [170,220,270,320,370,420,470,520] when $\delta_N = 0.2$.

Signal Length (n)	CPU Running Time (Seconds)				
	BPDN	SL0	NSL0	L_p -RLS	WReSL0
170	0.195	0.057	0.091	0.194	0.063
220	0.289	0.139	0.230	0.350	0.142
270	0.495	0.229	0.426	0.505	0.291
320	0.767	0.320	0.639	0.712	0.509
370	1.059	0.456	0.926	0.982	0.892
420	1.477	0.613	1.133	1.491	1.017
470	1.941	0.796	1.478	2.118	1.344
520	2.619	1.038	2.089	2.910	1.882

The performance of each algorithm under different noise intensities is shown in Figure 5. When $\delta_N = 0$, SL0 outperforms other algorithms, but with the increase of δ_N , the effect of SL0 becomes worse and worse. This result further illustrates that the traditional constrained sparse recovery algorithm does not have the performance of anti-noising. For BPDN, NSL0, L_p -RLS, and WReSL0, they all applied the regularization mechanism, and they are indeed superior to SL0 in the noise case. Therefore, the proposed WReSL0 in this paper has the best de-noising performance.

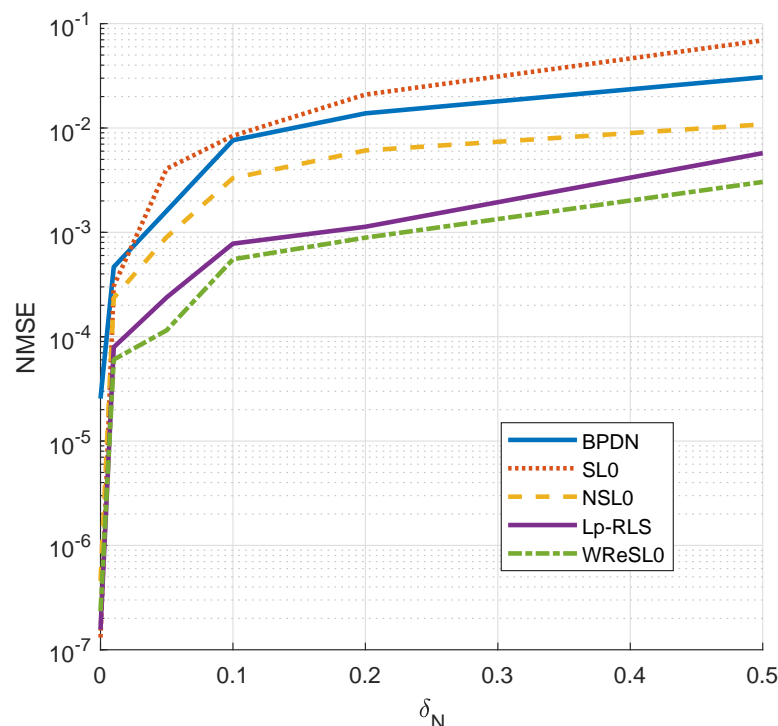


Figure 5. NMSE analysis by BPDN, SL0, NSL0, L_p -RLS, and WReSL0 when noise intensity δ_N changes according to the sequence [0, 0.1, 0.2, 0.3, 0.4, 0.5].

4.2. Image Recovery Performance in the Noise Case

Real images are considered to be approximately sparse under some proper basis, such as the DCT basis, DWT basis, etc. Here, we choose the DWT basis to recover these images. We compare the recovery performances based on the four real images in Figure 6: boat, Barbara, peppers, and Lena. The size of these images is 256×256 ; the compression ratio (CR; defined as m/n) is 0.5; and the noise δ_N equals 0.01. We still choose SL0, BPDN, NSL0, and L_p -RLS to make comparisons. For image recovery, the object of image processing is given by:

$$\mathbf{Y} = \Phi \mathbf{X} + \mathbf{B} \quad (31)$$

Here, \mathbf{Y} , \mathbf{X} , \mathbf{B} are matrices, and among these, $\mathbf{Y}, \mathbf{B} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{n \times n}$. In order to meet the basic requirements of CS, we perform the following processing:

$$\mathbf{Y}_i = \Phi \mathbf{X}_i + \mathbf{B}_i \quad s.t. \quad i = 1, 2, \dots, n. \quad (32)$$

where \mathbf{Y}_i , \mathbf{X}_i , \mathbf{B}_i are the column vectors of \mathbf{Y} , \mathbf{X} , \mathbf{B} , respectively. $\mathbf{B}_i = \delta_N \mathbf{\Omega}$, $\mathbf{\Omega}$ obeys the Gaussian distribution $\mathcal{N}(0, 1)$.

To perform image recovery, we evaluate it by the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM). PSNR is defined as:

$$\text{PSNR} = 10 \log(255^2 / \text{MSE}) \quad (33)$$

where $\text{MSE} = \|x - \hat{x}\|_2^2$, and SSIM is defined as:

$$\text{SSIM}(p, q) = \frac{(2\mu_p + \mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}. \quad (34)$$

Among these, μ_p is the mean of image p , μ_q is the mean of image q , σ_p is the variance of image p , σ_q is the variance of image q , and σ_{pq} is the covariance between image p and image q . Parameters $c_1 = z_1 L$ and $c_2 = z_2 L$, for which $z_1 = 0.01$, $z_2 = 0.03$, and L is the dynamic range of pixel values. The range of SSIM is $[-1, 1]$, and when these two images are the same, SSIM equals one.



Figure 6. Original images: (a) boat; (b) Barbara; (c) peppers; (d) Lena.

Figure 7 shows the recovery effect of boat and Barbara with noise intensity $\delta_N = 0.01$. For boat and Barbara, the recovered images by SL0 and BPDN have obvious water ripples, while recovered images by other algorithms have no such water ripples. Similarly, for peppers and Lena, the recovered images by SL0 and BPDN are blurred compared with the recovered images by other algorithms. The NSL0, L_p -RLS, and WReSL0 algorithms are also effective at noisy image recovery. For the NSL0, L_p -RLS, and WReSL0 algorithms, their recovery effects are very similar. In order to further analyze the advantages and disadvantages of the algorithms, we analyze the PSNR and SSIM of the images recovered by these algorithms, and the results are shown in Tables 3 and 4. By observation and analysis, L_p -RLS performs

better than NSL0, and at the same time, WReSL0 outperforms L_p -RLS. Hence, the WReSL0 proposed by this paper is superior to the other selected algorithms in image processing.



Figure 7. Image recovery effect by the BPDN, SL0, NSL0, L_p -RLS, and WReSL0 algorithms with noise intensity $\delta_N = 0.01$. In (a–d), from left to right, are: image recovered by the BPDN, SL0, NSL0, L_p -RLS, and WReSL0 algorithms.

Table 3. PSNR and SSIM analysis of recovered images (boat and Barbara) by SL0, BPDN, NSL0, L_p -RLS, and WReSL0.

Items	Barbara		Boat	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM
SL0	27.983	0.981	26.959	0.969
BPDN	28.834	0.984	27.376	0.971
NSL0	31.296	0.991	31.247	0.988
L_p -RLS	31.786	0.992	31.797	0.989
WReSL0	32.244	0.993	32.369	0.991

Table 4. PSNR and SSIM analysis of recovered images (peppers and Lena) by SL0, BPDN, NSL0, L_p -RLS, and WReSL0.

Items	Peppers		Lena	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM
SL0	28.677	0.982	30.334	0.987
BPDN	29.542	0.985	29.875	0.983
NSL0	31.373	0.991	32.639	0.993
L_p -RLS	33.757	0.994	34.051	0.995
WReSL0	34.231	0.996	34.653	0.997

5. Application in Underdetermined Blind Source Separation

The problem of UBSS stems from cocktail reception, which is shown in Figure 8. Suppose the source signal matrix $\mathbf{S}(t) = [s_1(t), s_2(t), \dots, s_m(t)]^T$, the mixed matrix (Sensors) \mathbf{A} is $m \times n$ ($m \ll n$) matrix, the Gaussian noise $\mathbf{G}(t) = [g_1(t), g_2(t), \dots, g_m(t)]^T$ is generated by Gaussian distribution, and the observed mixed signal matrix $\mathbf{X}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$; therefore, the general mathematical models of UBSS can be summarized as:

$$\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{G}(t) \quad (35)$$

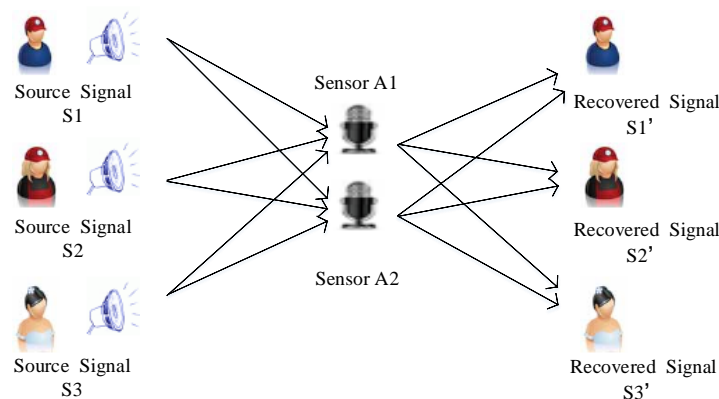


Figure 8. Schematic diagram of cocktail reception signal mixing.

In fact, each signal has L data collected; therefore, $\mathbf{X} \in \mathbb{R}^{m \times L}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ ($m \ll n$), $\mathbf{S}(t) \in \mathbb{R}^{n \times L}$, and $\mathbf{G} \in \mathbb{R}^{m \times L}$, and \mathbf{G} can be represented as $\delta_N \mathbf{W}$ (\mathbf{W} obeys $\mathcal{N}(0, 1)$). The purpose of UBSS is to use the mixed signal matrix $\mathbf{x}(t)$ to estimate the source signal matrix $\mathbf{s}(t)$. In fact, this is the process of solving the underdetermined linear system of equations (ULSE). For this problem, we can use the two-step method to solve it, which is shown in Figure 9.

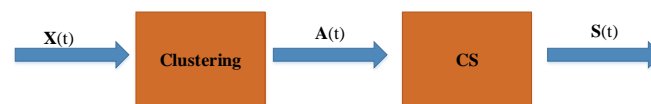


Figure 9. Schematic diagram of two-step method for UBSS.

From Figure 9, firstly, we get the mixed matrix by the clustering method and then use CS technology to separate the signal, so as to restore the original signal.

5.1. Process Analysis of CS Applied to UBSS

5.1.1. Solving the Mixed Matrix by the Potential Function Method

In this section, we choose the potential function method to solve the mixed matrix \mathbf{A} . To verify the performance of the proposed WReSL0 algorithm better, we choose four simulated signals and four real images to organize experiments in this section.

Suppose there are four source signals, which are:

$$\left\{ \begin{array}{l} s_1(t) = 5 \sin(2\pi f_1 t) \\ s_2(t) = 5 \sin(2\pi f_2 t) \\ s_3(t) = 5 \sin(2\pi f_3 t) \\ s_4(t) = 5 \sin(2\pi f_4 t) \\ \mathbf{S} = [s_1(t), s_2(t), s_3(t), s_4(t)]^T \end{array} \right. \quad (36)$$

where $f_1 = 310$ Hz, $f_2 = 210$ Hz, $f_3 = 110$ Hz, and $f_4 = 10$ Hz. The length of each source signal s_i ($i = 1, 2, 3, 4$) is 1024, and the sample frequency is 1024 Hz. These four signals are shown in Figure 10.

The four source images are the classic standard test images: boat, Barbara, peppers, and Lena, which are in Figure 6.

Suppose there are two sensors that receive signals and another two sensors that receive images. Mixed matrices \mathbf{A} and \mathbf{B} are set as:

$$\mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 0.9930 & 0.9941 & 0.1092 & 0.9304 \\ 0.2116 & 0.0757 & 0.9647 & 0.3837 \end{bmatrix} \quad (37)$$

$$\mathbf{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} 0.9354 & 0.9877 & -0.6730 & 0.1097 \\ 0.3535 & 0.07846 & 0.7396 & 0.9940 \end{bmatrix}$$

By this mixed matrix and added Gaussian noise ($\delta_N = 0.1$), we can get the two mixed signals, which are shown in Figure 11, and the two mixed images, which are shown in Figure 12. Then, we can get the estimated mixed matrix $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ by clustering by the potential function method [37]. As shown in Figure 13, the potential function method can cluster well. By clustering, we get the estimated values of \mathbf{A} and \mathbf{B} , as follows:

$$\begin{aligned} \hat{\mathbf{A}} &= \begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \end{bmatrix} = \begin{bmatrix} 0.9792 & 0.9969 & 0.1097 & 0.9239 \\ 0.2028 & 0.0785 & 0.9940 & 0.3827 \end{bmatrix} \\ \hat{\mathbf{B}} &= \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} = \begin{bmatrix} 0.9478 & 0.9431 & -0.6483 & 0.1130 \\ 0.3476 & 0.0765 & 0.7075 & 0.9979 \end{bmatrix} \end{aligned} \quad (38)$$

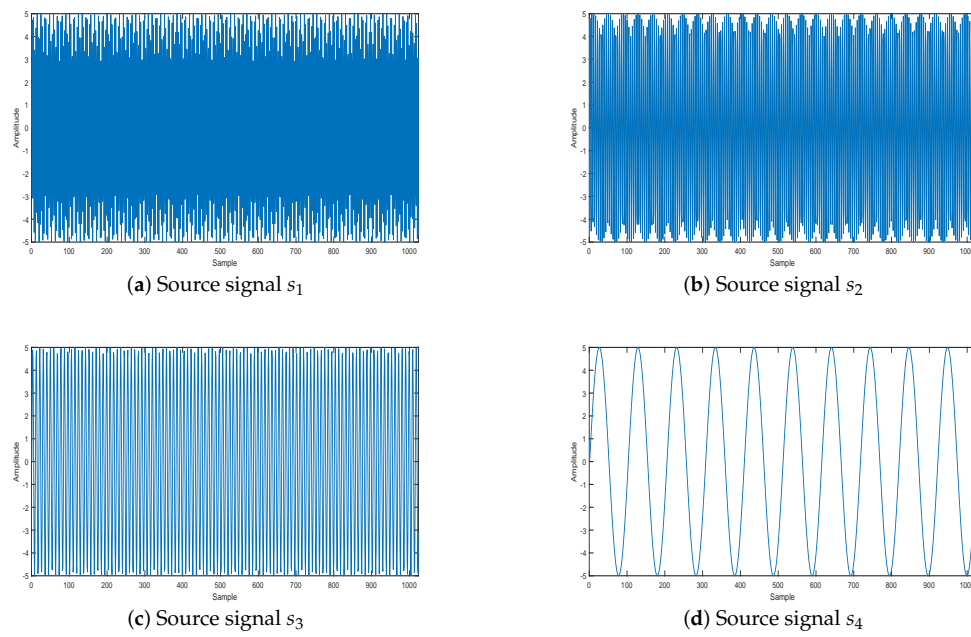


Figure 10. Source signal.

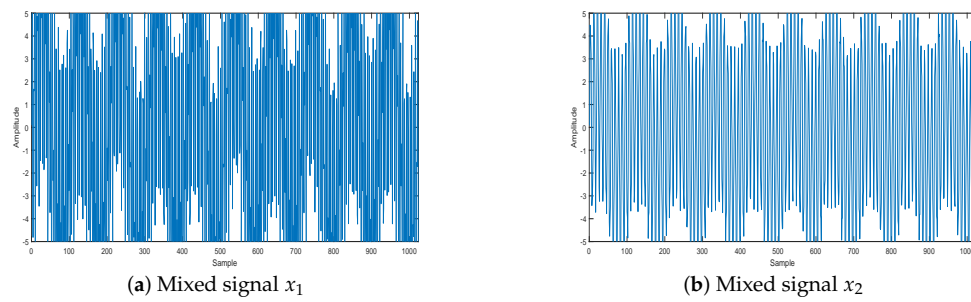


Figure 11. Mixed signal by sensors.

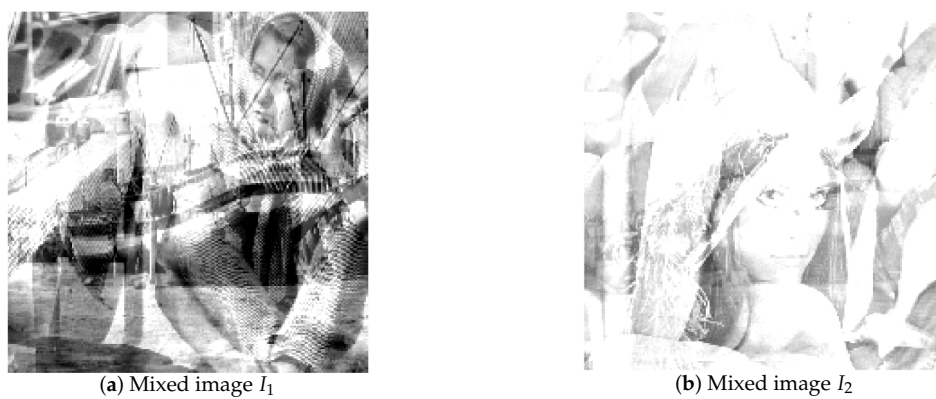


Figure 12. Mixed image by sensors.

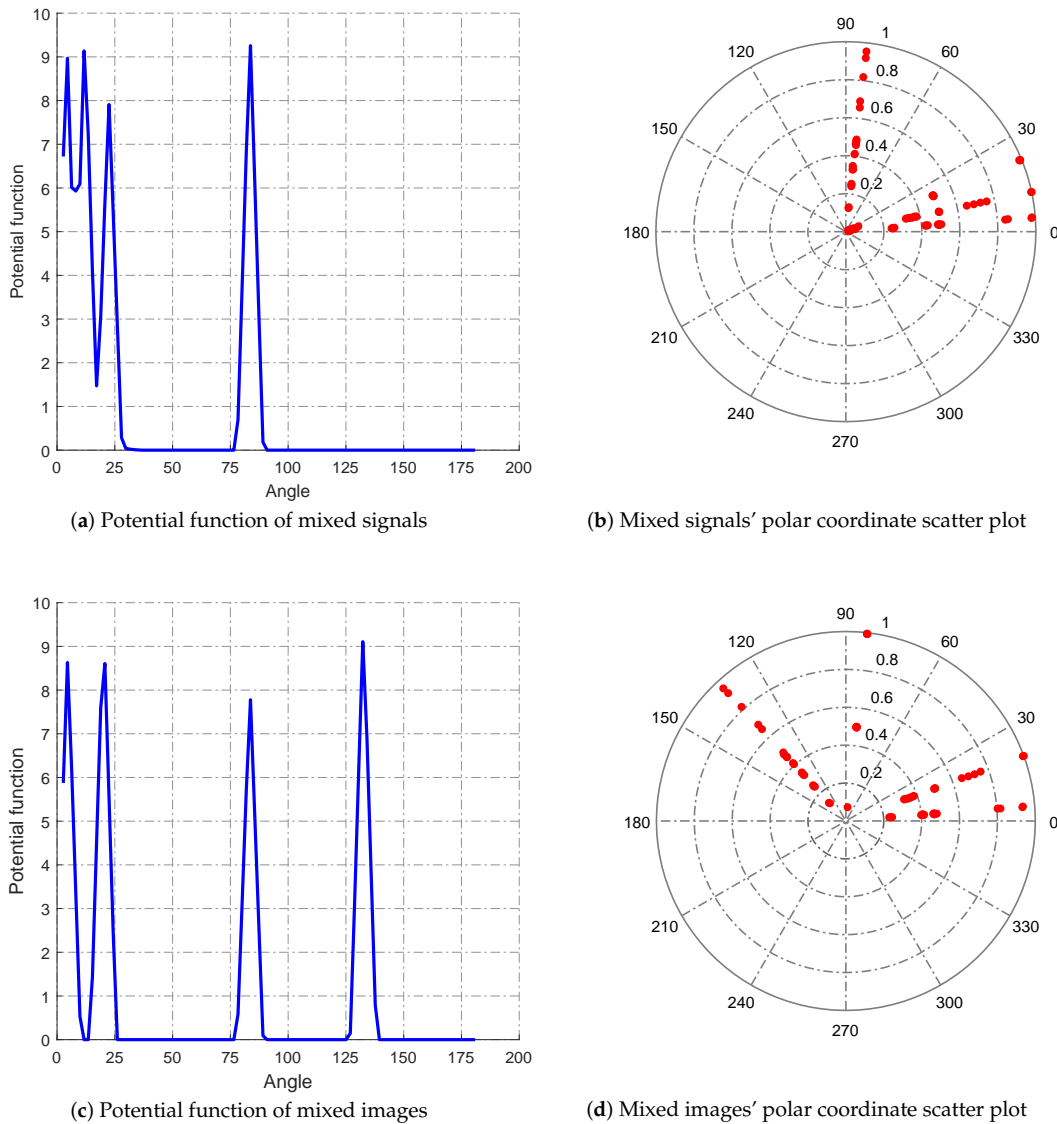


Figure 13. Clustering analysis.

By calculation, the error of solving the mixed matrix is $\frac{\|A-\hat{A}\|_F}{\|A\|_F} \times 100\% = 1.763\%$ and $\frac{\|B-\hat{B}\|_F}{\|B\|_F} \times 100\% = 3.64\%$. This error range is much smaller than the classical k-means and fuzzy c-means, thus laying a foundation for the reconstruction of compressed sensing.

5.1.2. Using CS to Separate Source Signals

The next problem is to get $S(t)$ from known $A(t)$ and $X(t)$. Here, we solve this problem by CS. The solution process is similar to the image reconstruction process. The difference is that the sparse basis used here is the Fourier basis. Then, we apply the proposed RWeSL0 algorithm to this process. First, we transform the obtained $x(t)$ into column vectors:

$$x(t) = [x_1(t), x_2(t)]^T \Rightarrow \tilde{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{39}$$

Then, we use the Fourier (for the sparse signal) or DWT (for the image) basis for sparse representation and extend the matrix and the valuated mixed matrix to obtain the sensing matrix.

$$\begin{aligned} \tilde{\mathbf{A}} &= \hat{\mathbf{A}} \otimes \mathbf{I}_{L \times L}, \text{ or } \tilde{\mathbf{B}} = \hat{\mathbf{B}} \otimes \mathbf{I}_{L \times L} \\ \Psi &= \text{Fourier}(\mathbf{I}_{L \times L})/\sqrt{L}, \text{ or } \Psi = \text{DWT}(\mathbf{I}_{L \times L})/\sqrt{L} \\ \tilde{\Psi} &= \begin{bmatrix} \Psi & 0 & \dots & 0 \\ 0 & \Psi & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & \Psi \end{bmatrix} \\ \Phi &= \tilde{\mathbf{A}}\tilde{\Psi}, \text{ or } \Phi = \tilde{\mathbf{B}}\tilde{\Psi} \end{aligned} \tag{40}$$

For this equation, \otimes denotes the Kronecker product sign, $\text{Fourier}(\cdot)$ represents the Fourier transform, and DWT represents the discrete wavelet transform. Therefore, the CS-UBSS model can be described as:

$$\begin{aligned} \hat{\mathbf{X}}(t) &= \tilde{\mathbf{A}}(t)\mathbf{S}(t) + \mathbf{G}(t) \\ &= \tilde{\mathbf{A}}(t)\tilde{\Psi}\Theta(t) + \mathbf{G}(t) \\ &= \Phi\Theta(t) + \mathbf{G}(t) \end{aligned}$$

or

$$\hat{\mathbf{X}}(t) = \tilde{\mathbf{B}}(t)\mathbf{S}(t) + \mathbf{G}(t)$$

$$\begin{aligned} &= \tilde{\mathbf{B}}(t)\tilde{\Psi}\Theta(t) + \mathbf{G}(t) \\ &= \Phi\Theta(t) + \mathbf{G}(t) \end{aligned} \tag{41}$$

where Θ is the Fourier transform or DWT of $\mathbf{S}(t)$, so Θ is a sparse signal. As for UBSS in the images, firstly, each image matrix needs to be transformed into a row vector, then the four row vectors form a matrix $\mathbf{S}(t)$. At the same time, the sparse basis in Equation (40) needs to be replaced by DWT.

Then, we can recover the source signal by CS. In summary, the above can be described as the flowchart in Figure 14.

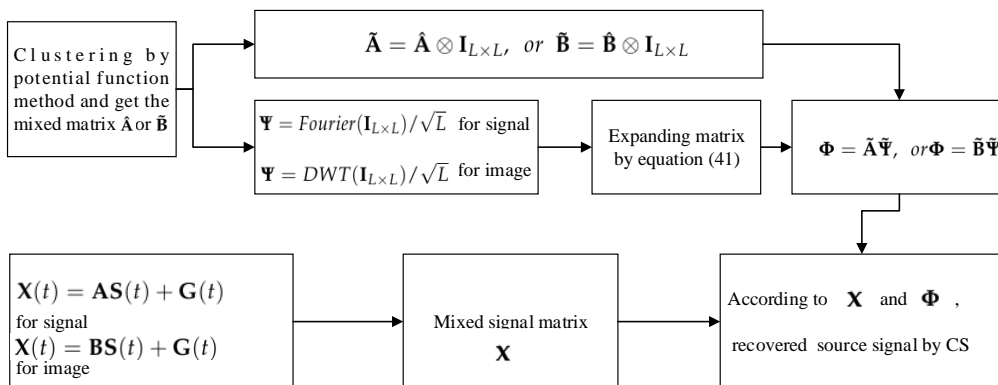


Figure 14. Flowchart of UBSS by CS.

5.2. Performance Analysis of the WReSL0 Algorithm Applied to UBSS

5.2.1. The Effect of the WReSL0 Algorithm Applied to UBSS

In this section, we evaluate the effect of the WReSL0 algorithm applied to UBSS by the separation of signals and spectrum analysis.

The effect of the separation of signals is shown in Figure 15: the source signals are well separated, and the separation signals and the original signals are very similar. Figure 16 displays the error between the original source signal and the recovered source signal. It indicates that the error between the original source signal and the recovered source signal is fairly small, and the WReSL0 algorithm can better deal with the problem of UBSS. In addition, We get the time-frequency diagram of the restored signal by short-time Fourier transform. Figure 17 is the time-frequency diagram. From this figure, we find that each signal has the same frequency as the original signal, and it also validates the rationality of the proposed algorithm for UBSS.

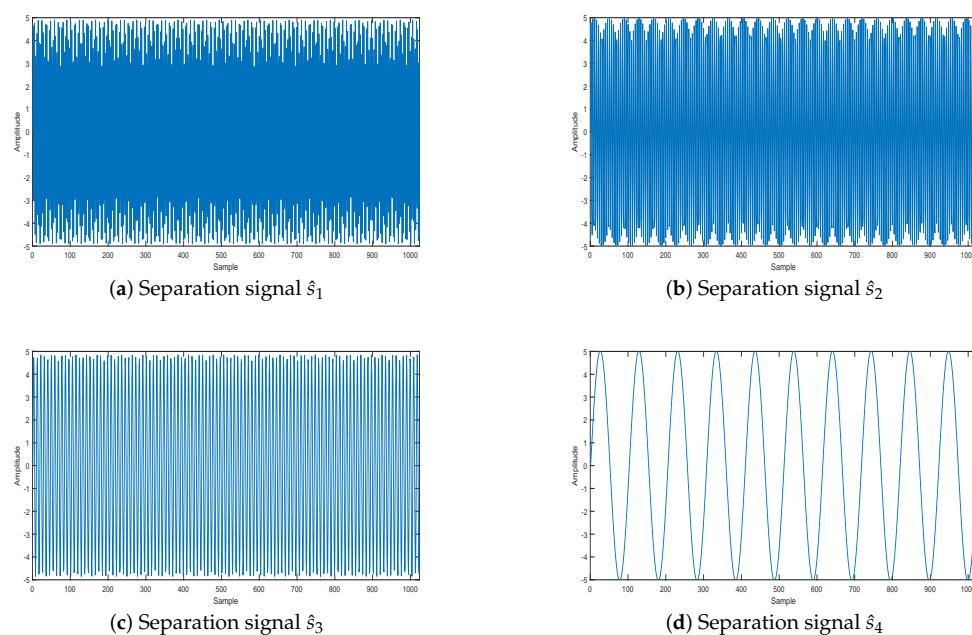


Figure 15. Separation signal.

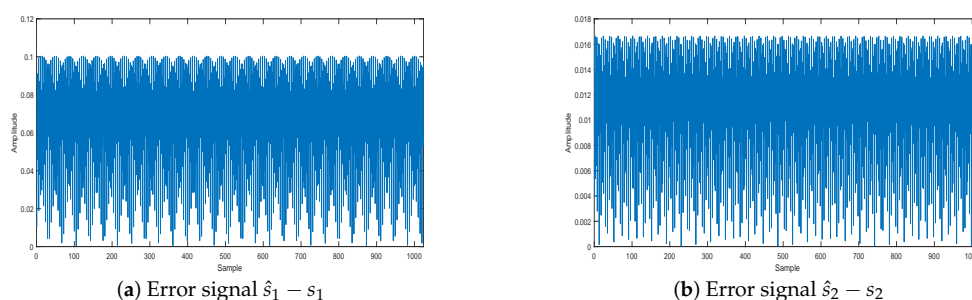


Figure 16. Cont.

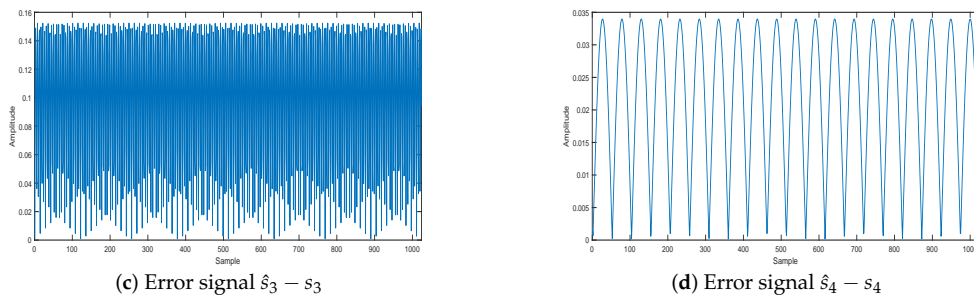


Figure 16. Separation signal error analysis.

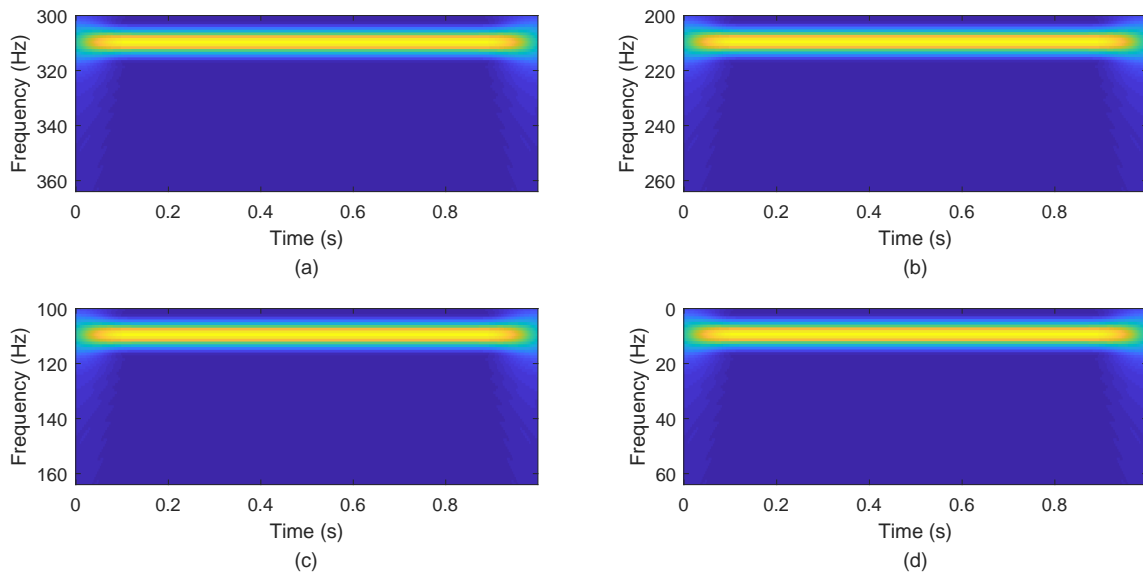


Figure 17. Separation signals' frequency spectrum. Subfigures (a–d) show the frequency spectrums of separation signals $\hat{s}_1, \hat{s}_2, \hat{s}_3,$ and \hat{s}_4 .

5.2.2. Performance Comparisons of the Selected Algorithms

Here, we use the SL0, NSL0, and L_p -RLS algorithms and the classical shortest path method (SPM) [38] to make a comparison in different noise cases. In order to analyze the situation of signal recovery clearly, we apply average SNR (ASNR) (for the signal) and average peak SNR (APSNR) (for the image) to evaluate. Let the original source signal be s_i and the recovered source signal be \hat{s}_i , so ANSR is defined as:

$$ASNR = \frac{1}{n} \sum_{i=1}^n SNR_i \tag{42}$$

$$SNR_i = 20 \log \frac{\|\hat{s}_i - s_i\|_2}{\|s_i\|_2},$$

and PSNR is defined as:

$$APSNR = \frac{1}{n} \sum_{i=1}^n PSNR_i \tag{43}$$

$$PSNR_i = 10 \log \frac{255^2 \times M \times N}{\|\hat{s}_i - s_i\|_2}$$

where M and N are the width and height of the image.

The ASNR comparisons are shown in Table 5. From the table, we can see that ASNR attenuates sharply when δ_N increases from 0.15–0.2. The reason is that the error of the valuated mixed matrix \hat{A} increases obviously, which leads those CS recovery algorithms to perform poorly. In fact, from this table, our proposed RWeSL0 algorithm performs well when δ_N is less than 0.15, and when δ_N is greater than 0.15, the L_p -RLS algorithm performs best, followed by our proposed RWeSL0 algorithm.

The APSNR comparisons are shown in Table 6. In this table, It is clear that APSNR is not high, and it drops greatly when δ_N increases from 0.15–0.2. From Figure 18, we can see that these separated images seem to be enveloped in mist, which leads to a low APSNR. Therefore, we will try our best to improve this problem in the future.

In summary, the CS technique can be used in UBSS and performs well especially for the signal recovery. Our proposed WReSL0 algorithm can perform well in UBSS for the signal recovery when the noise is small; and regarding image recovery, we will develop this in the future.

Table 5. Average SNR (ASNR) analysis for separated signals by SPM, SL0, NSL0, L_p -RLS, and the proposed WReSL0 with δ_N changing according to sequence [0,0.1,0.15,0.18,0.2] with 100 runs.

Oise Intensity (δ_N)	Error of \hat{A} (%)	ASNR (dB)				
		SPM	SL0	NSL0	L_p -RLS	WReSL0
0	1.763	45.443	41.576	42.324	38.412	39.993
0.1	1.763	36.788	35.278	36.034	37.091	39.295
0.15	1.763	31.407	30.754	32.930	35.332	38.975
0.18	112.6	26.355	24.063	25.437	28.305	26.650
0.2	126.3	11.201	9.974	12.358	17.549	15.581

Table 6. APSNR analysis for separated images by SPM, SL0, NSL0, L_p -RLS, and the proposed WReSL0 with δ_N changing according to the sequence [0,0.1,0.15,0.18,0.2] with 100 runs.

Noise Intensity (δ_N)	Error of \hat{B} (%)	APSNR (dB)				
		SPM	SL0	NSL0	L_p -RLS	WReSL0
0	3.64	16.447	19.211	20.035	16.372	18.483
0.1	3.64	15.639	16.305	17.327	15.407	17.849
0.15	3.64	13.407	14.754	14.930	14.932	17.351
0.18	133.2	9.355	11.063	11.437	10.305	11.650
0.2	142.4	5.201	5.974	6.358	3.549	5.581

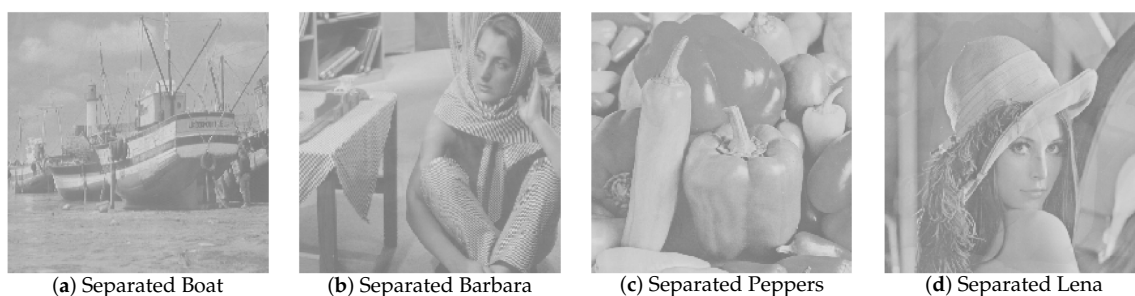


Figure 18. Separated images: (a) boat; (b) Barbara; (c) peppers; (d) Lena.

6. Conclusions

In this paper, we propose the WReSL0 algorithm to recover the sparse signal from given $\{y, \Phi\}$ in the noise case. The WReSL0 algorithm is constructed under the GD method, in which the update process of x in the inner loop adopts the regularization mechanism to enhance the de-noising performance. As a key part of the WReSL0 algorithm, a weighted smoothed function

$\mathbf{W}^T H_\sigma(\mathbf{x})$ is proposed to promote sparsity and provide the guarantee of robust and accurate signal recovery. Furthermore, We deduced the value of μ and the initial value σ_{\max} to ensure the optimization performance of the algorithm. Performance simulation experiments on both real signals and real images show that the proposed WReSL0 algorithm performs better than the L_1 or L_p regularization methods and the classical L_0 regularization methods. Finally, we apply the proposed WReSL0 algorithm to solve the problem of UBSS and also make comparisons with the classical SPM, SL0, NSL0, and L_p -RLS algorithms. Experiments show that this algorithm has some advanced performance. In addition, we would also like to apply the the proposed algorithm to other CS applications such as the RPCA [39], SAR imaging [40], and other de-noising methods [41].

Author Contributions: All authors have made great contributions to the work. L.W., X.Y., H.Y., and J.X. conceived of and designed the experiments; X.Y. and H.Y. performed the experiments and analyzed the data; X.Y. gave insightful suggestions for the work; X.Y. and H.Y. wrote the paper.

Funding: This research received no external funding.

Acknowledgments: This paper is supported by the National Key Laboratory of Communication Anti-jamming Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhang, C.Z.; Wang, Y.; Jing, F.L. Underdetermined Blind Source Separation of Synchronous Orthogonal Frequency Hopping Signals Based on Single Source Points Detection. *Sensors* **2017**, *17*, 2074. [[CrossRef](#)]
- Zhen, L.; Peng, D.; Zhang, Y.; Xiang, Y.; Chen, P. Underdetermined blind source separation using sparse coding. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *99*, 1–7. [[CrossRef](#)] [[PubMed](#)]
- Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
- Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *2*, 21–30. [[CrossRef](#)]
- Badeńska, A.; Błaszczuk, Ł. Compressed sensing for real measurements of quaternion signals. *J. Frankl. Inst.* **2017**, *354*, 5753–5769. [[CrossRef](#)]
- Candès, E.J. The restricted isometry property and its implications for compressed sensing. *C. R. Math.* **2008**, *910*, 589–592. [[CrossRef](#)]
- Cahill, J.; Chen, X.; Wang, R. The gap between the null space property and the restricted isometry property. *Linear Algebra Its Appl.* **2016**, *501*, 363–375. [[CrossRef](#)]
- Huang, S.; Tran, T.D. Sparse Signal Recovery via Generalized Entropy Functions Minimization. *arXiv* **2017**, arXiv:1703.10556.
- Tropp, J.A.; Gilbert, A.C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **2007**, *12*, 4655–4666. [[CrossRef](#)]
- Determe, J.F.; Louveaux, J.; Jacques, L.; Horlin, F. On the noise robustness of simultaneous orthogonal matching pursuit. *IEEE Trans. Signal Process.* **2016**, *65*, 864–875. [[CrossRef](#)]
- Donoho, D.L.; Tsai, Y.; Starck, J.L. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **2012**, *2*, 1094–1121. [[CrossRef](#)]
- Needell, D.; Vershynin, R. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE J. Sel. Top. Signal Process.* **2010**, *2*, 310–316. [[CrossRef](#)]
- Needell, D.; Tropp, J.A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM* **2010**, *12*, 93–100. [[CrossRef](#)]
- Jian, W.; Seokbeop, K.; Byonghyo, S. Generalized orthogonal matching pursuit. *IEEE Trans. Signal Process.* **2012**, *12*, 6202–6216. [[CrossRef](#)]
- Wang, J.; Kwon, S.; Li, P.; Shim, B. Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis. *IEEE Trans. Signal Process.* **2016**, *64*, 1076–1089. [[CrossRef](#)]
- Dai, W.; Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory* **2009**, *5*, 2230–2249. [[CrossRef](#)]

17. Goyal, P.; Singh, B. Subspace pursuit for sparse signal reconstruction in wireless sensor networks. *Procedia Comput. Sci.* **2018**, *125*, 228–233. [[CrossRef](#)]
18. Liu, X.J.; Xia, S.T.; Fu, F.W. Reconstruction guarantee analysis of basis pursuit for binary measurement matrices in compressed sensing. *IEEE Trans. Inf. Theory* **2017**, *63*, 2922–2932. [[CrossRef](#)]
19. Mohimani, H.; Babaie-Zadeh, M.; Jutten, C. A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed L0 Norm. *IEEE Trans. Signal Process.* **2009**, *57*, 289–301. [[CrossRef](#)]
20. Zhao, R.; Lin, W.; Li, H.; Hu, S. Reconstruction algorithm for compressive sensing based on smoothed L0 norm and revised newton method. *J. Comput.-Aided Des. Comput. Graph.* **2012**, *24*, 478–484.
21. Ye, X.; Zhu, W.P. Sparse channel estimation of pulse-shaping multiple-input–multiple-output orthogonal frequency division multiplexing systems with an approximate gradient L_2 -SL0 reconstruction algorithm. *Let Commun.* **2014**, *8*, 1124–1131. [[CrossRef](#)]
22. Nowak, R.D.; Wright, S.J. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 586–597.
23. Long, T.; Jiao, W.; He, G. RPC estimation via ℓ_1 -norm-regularized least squares (L1LS). *IEEE Trans. Geosci. Remote Sens.* **2015**, *8*, 4554–4567. [[CrossRef](#)]
24. Pant, J.K.; Lu, W.S.; Antoniou, A. New improved algorithms for compressive sensing based on ℓ_p norm. *IEEE Trans. Circuits Syst. II Express Br.* **2014**, *3*, 198–202. [[CrossRef](#)]
25. Wipf, D.; Nagarajan, S. Iterative Reweighted and Methods for Finding Sparse Solutions. *IEEE J. Sel. Top. Signal Process.* **2016**, *2*, 317–329.
26. Zhang, C.; Hao, D.; Hou, C.; Yin, X. A New Approach for Sparse Signal Recovery in Compressed Sensing Based on Minimizing Composite Trigonometric Function. *IEEE Access* **2018**, *6*, 44894–44904. [[CrossRef](#)]
27. Candès, E.J.; Wakin, M.B.; Boyd, S.P. Enhancing sparsity by weighted L1 minimization. *J. Fourier Anal. Appl.* **2008**, *14*, 877–905. [[CrossRef](#)]
28. Pant, J.K.; Lu, W.S.; Antoniou, A. Reconstruction of sparse signals by minimizing a re-weighted approximate L0-norm in the null space of the measurement matrix. In Proceedings of the IEEE International Midwest Symposium on Circuits and Systems, Seattle, WA, USA, 1–4 August 2010; pp. 430–433.
29. Aggarwal, P.; Gupta, A. Accelerated fmri reconstruction using matrix completion with sparse recovery via split bregman. *Neurocomputing* **2016**, *216*, 319–330. [[CrossRef](#)]
30. Chu, Y.J.; Mak, C.M. A new qr decomposition-based rls algorithm using the split bregman method for L1-regularized problems. *Signal Process.* **2016**, *128*, 303–308. [[CrossRef](#)]
31. Hu, Y.; Liu, J.; Leng, C.; An, Y.; Zhang, S.; Wang, K. Lp regularization for bioluminescence tomography based on the split bregman method. *Mol. Imaging Biol.* **2016**, *18*, 1–8. [[CrossRef](#)]
32. Liu, Y.; Zhan, Z.; Cai, J.F.; Guo, D.; Chen, Z.; Qu, X. Projected iterative soft-thresholding algorithm for tight frames in compressed sensing magnetic resonance imaging. *IEEE Trans. Med. Imaging* **2016**, *35*, 2130–2140. [[CrossRef](#)]
33. Yang, L.; Pong, T.K.; Chen, X. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. *Mathematics* **2016**, *10*, 74–110. [[CrossRef](#)]
34. Antoniou, A.; Lu, W.S. *Practical Optimization: Algorithms and Engineering Applications*; Springer: New York, NY, USA, 2007.
35. Samora, I.; Franca, M.J.; Schleiss, A.J.; Ramos, H.M. Simulated annealing in optimization of energy production in a water supply network. *Water Resour. Manag.* **2016**, *30*, 1533–1547. [[CrossRef](#)]
36. Goldstein, T.; Studer, C. Phasemax: Convex phase retrieval via basis pursuit. *IEEE Trans. Inf. Theory* **2018**, *64*, 2675–2689. [[CrossRef](#)]
37. Wei-Hong, F.U.; Ai-Li, L.I.; Li-Fen, M.A.; Huang, K.; Yan, X. Underdetermined blind separation based on potential function with estimated parameter’s decreasing sequence. *Syst. Eng. Electron.* **2014**, *36*, 619–623.
38. Bofill, P.; Zibulevsky, M. Underdetermined blind source separation using sparse representations. *Signal Process.* **2001**, *81*, 2353–2362. [[CrossRef](#)]
39. Su, J.; Tao, H.; Tao, M.; Wang, L.; Xie, J. Narrow-band interference suppression via rpca-based signal separation in time–frequency domain. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *99*, 1–10. [[CrossRef](#)]

40. Ni, J.C.; Zhang, Q.; Luo, Y.; Sun, L. Compressed sensing sar imaging based on centralized sparse representation. *IEEE Sens. J.* **2018**, *18*, 4920–4932. [[CrossRef](#)]
41. Li, G.; Xiao, X.; Tang, J.T.; Li, J.; Zhu, H.J.; Zhou, C.; Yan, F.B. Near—Source noise suppression of AMT by compressive sensing and mathematical morphology filtering. *Appl. Geophys.* **2017**, *4*, 581–589. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).