Signal Processing 125 (2016) 9–20

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

An efficient algorithm for designing projection matrix in compressive sensing based on alternating optimization

Tao Hong [a,*], Huang Bai [a], Sheng Li [a], Zhihui Zhu [b]

[a] Zhejiang Key Laboratory for Signal Processing, College of Information Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang, PR China

[b] Department of Electrical Engineering and Computation Science, Colorado School of Mines, Golden, CO 80401, USA

ARTICLE INFO

Article history:
Received 26 August 2015
Received in revised form
28 November 2015
Accepted 26 December 2015
Available online 12 January 2016

Keywords:
Compressive sensing (CS)
Alternating optimization
Robust projection matrix
Mutual coherence

ABSTRACT

This paper considers the problem of optimally designing the projection matrix $\boldsymbol{\Phi}$ for a certain class of signals which can be sparsely represented by a specified dictionary $\boldsymbol{\Psi}$. The optimal projection matrix is proposed to minimize the distance between the Gram matrix of the equivalent dictionary $\boldsymbol{\Phi\Psi}$ and a set of relaxed Equiangular Tight Frames (ETFs). An efficient method is derived for the optimal projection matrix design with a given Gram matrix. In addition, an extension of projection matrix design is derived for the scenarios where the signals cannot be represented exactly sparse in a specified dictionary. Simulations with synthetic data and real images demonstrate that the obtained projection matrix significantly improves the signal recovery accuracy of a system and outperforms those obtained by the existing algorithms.

1. Introduction

Compressive sensing, also known as compressed sensing (CS), has attracted a great deal of researchers' attention in signal processing community and applied mathematics since its introduction at the early of this century [1–3]. Excellent comments on the development of CS in the signal processing community can be found in [4–6].

CS framework involves a class of mathematic problems of reconstructing a signal[1] $\boldsymbol{x} \in \Re^{N\times 1}$ from its limited linear observations $\boldsymbol{y} \in \Re^{M\times 1}$ obtained through the projection matrix $\boldsymbol{\Phi} \in \Re^{M\times N}$:

$$\boldsymbol{y} = \boldsymbol{\Phi x} \tag{1}$$

with $M \ll N$. There are infinite number of candidate signals $\tilde{\boldsymbol{x}}$ for which $\boldsymbol{y} = \boldsymbol{\Phi}\tilde{\boldsymbol{x}}$ because the number of the measurements M is smaller than the dimension N of the signal. Additional conditions about $\boldsymbol{x}$ should be assumed to solve such an underdetermined linear system $\boldsymbol{y} = \boldsymbol{\Phi x}$ for a unique $\boldsymbol{x}$. The CS framework assumes the signal $\boldsymbol{x}$ can be represented as a linear combination of (a small number of) columns from $\boldsymbol{\Psi} \in \Re^{N\times L}$:

$$\boldsymbol{x} = \boldsymbol{\Psi\theta}$$

where $\boldsymbol{\Psi}$ is called a dictionary and $\boldsymbol{\theta} \in \Re^{L\times 1}$ is referred to as a coefficient vector. A dictionary is called redundant or overcomplete if $L > N$. All dictionaries considered in this paper are assumed to be overcomplete. A lot of works [7–11] have shown that most of the natural signals can be expressed by few atoms from a properly chosen dictionary.

The signal $\boldsymbol{x}$ is said to be S-sparse in $\boldsymbol{\Psi}$ if $\|\boldsymbol{\theta}\|_0 = S$. Here $\|\boldsymbol{\theta}\|_0$ denotes the number of non-zero elements in $\boldsymbol{\theta}$. Strictly speaking, $\|\cdot\|_0$ is not a true norm,[2] as the

* Corresponding author.
E-mail addresses: hongtao@zjut.edu.cn (T. Hong), zzhu@mines.edu (Z. Zhu).

[1] In this paper, we focus on the discrete time signal with finite-length.

[2] For simplicity, l_0 is used to represent the zero norm in this paper.

http://dx.doi.org/10.1016/j.sigpro.2015.12.015
0165-1684/

definition of a true l_p-norm with $p \geq 1$ of a vector $\boldsymbol{v}$ is

$$\|\boldsymbol{v}\|_p \triangleq \left(\sum_{k=1}^{N} |\boldsymbol{v}(k)|^p\right)^{1/p}$$

where $\boldsymbol{v}(k)$ is the k-th element[3] of $\boldsymbol{v}$.

Assuming $\boldsymbol{\theta}$ is sparse (which means S is small compared to L), the CS system recovers $\boldsymbol{\theta}$ (and hence $\boldsymbol{x}$) by resolving the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_0 \\ &\text{s.t.} \boldsymbol{y} = \boldsymbol{D}\boldsymbol{\theta} \\ &\boldsymbol{D} \triangleq \boldsymbol{\Phi}\boldsymbol{\Psi} \end{aligned} \tag{2}$$

Under some conditions [6], the NP-hard problem (2) can be solved efficiently with theoretical guarantee by several numerical algorithms, such as basis pursuit (BP) [12], orthogonal matching pursuit (OMP) [13], least absolute shrinkage and selection operator (LASSO) [14] and other algorithms proposed in [10,15–18].

One of the important concepts in CS is the mutual coherence $\mu(\boldsymbol{D})$ [6], which denotes the coherence between any two columns of the equivalent dictionary $\boldsymbol{D}$. It has been proved in [11] that an S-sparse signal $\boldsymbol{x}$ can be perfectly recovered from the observed signal $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\theta}$ through (2) if the following condition holds:

$$S < \frac{1}{2}\left[1 + \frac{1}{\mu_{mx}(\boldsymbol{D})}\right], \tag{3}$$

where $\mu_{mx}(\boldsymbol{D})$ represents the maximal mutual coherence.

Noting that (3) indicates the set of signals that can be recovered exactly by solving (2), one may design $\boldsymbol{\Phi}$ such that $\mu_{mx}(\boldsymbol{D})$ is as small as possible so that the CS system can recover the signal $\boldsymbol{x}$ with a relatively small measurements $\boldsymbol{y}$. In [19], $\mu_{mx}(\boldsymbol{D})$ was minimized directly by a subgradient optimization approach but has extremely high computational complexity. Instead of minimizing μ_{mx} directly, Elad [20] suggested reducing the average mutual coherence $\mu_{av}(\boldsymbol{D})$ and proposed an iterative algorithm based on shrinkage operation and the singular value decomposition (SVD). The simulations carried out in [19] show that the method proposed in [20] yields a little inferior performance in terms of signal recovery accuracy than their method.

In CS framework, it is desired to find an optimal projection matrix that the correlation between any two different columns of the equivalent dictionary is minimized. The direct idea is to check the *Gram matrix*. As defined by $\boldsymbol{G} \triangleq \boldsymbol{D}^{\mathcal{T}}\boldsymbol{D}$, [4] the *Gram matrix* for an ideal equivalent dictionary is an identity matrix. Based on this desirableness, an approximate solution is obtained in [21] by solving

$$\min_{\boldsymbol{\Phi}} \|\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^2 - \boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^2 \boldsymbol{\Gamma}^{\mathcal{T}} \boldsymbol{\Gamma} \boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^2\|_F^2 \tag{4}$$

where $\|\cdot\|_F$ denotes the *Frobenius* norm and $\boldsymbol{\Psi} = \boldsymbol{U}_{\boldsymbol{\Psi}}[\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}\ \boldsymbol{0}]\boldsymbol{V}_{\boldsymbol{\Psi}}^{\mathcal{T}}$ is an SVD of the dictionary $\boldsymbol{\Psi}$, and $\boldsymbol{\Gamma} = \boldsymbol{\Phi}\boldsymbol{U}_{\boldsymbol{\Psi}}$.[5]

[3] MATLAB notations are used in this paper. In this connection, $\boldsymbol{Q}(i,j)$ means the (i,j)-th element of matrix $\boldsymbol{Q}$, while $\boldsymbol{Q}(k,:)$ and $\boldsymbol{Q}(:,k)$ denote the k-th row and column vector of $\boldsymbol{Q}$, respectively.

[4] $\mathcal{T}$ denotes the transpose operator, and this notation will be assumed in the rest of the paper.

[5] In this paper, $\boldsymbol{0}$ denotes a matrix or vector with all of its elements equal to zero.

An iterative optimization method was developed to solve (4) in [21] with high efficiency. However, this approach does not have a clear physical meaning as it has lost the original purpose of making the Gram matrix as close to the identity matrix as possible, and hence, its performance can be improved. Recently, in [22], the optimal projection matrix was investigated by solving the following problem:

$$\min_{\boldsymbol{\Phi}} \|\boldsymbol{I} - \boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}\|_F^2 \tag{5}$$

A closed-form solution was given in [22] with the assumption that the dictionary matrix has full row rank.

Note that the methods proposed in [19,20] cannot reach the optimal solution for μ_{mx} and μ_{av}, respectively. Researchers found (3) is a worst-case bound and cannot reflect the average signal recovery performance. So the researchers tend to find a framework to control the values of μ_{mx} and μ_{av} simultaneously and then Equiangular Tight Frames (ETF) for projection matrix design is utilized in these years [23]. See also [24–26]. In [26], a gradient descent algorithm was developed to update the projection matrix with a target Gram. In this work, a strategy similar to [21] is developed for addressing this problem. The proposed algorithm denoted as Algorithm 2 has a fast convergence than the gradient in [26] in terms of the values of μ_{av} and μ_{mx}. In addition, the proposed algorithm can also be perfectly used to solve (5).

It should be noted that most of the existing projection design methods assuming the signal is exactly sparse,[6] e.g. [19–22,25–27]. In order to relax this constraint on the signal, a new objective function is proposed for optimal projection matrix design. In the case when the signal is not exactly sparse (which is true for most of practical signals, e.g., images), the recovery performance can be improved by using the proposed projection matrix.

The main contribution of this paper is two-fold:

- An iterative alternating minimization algorithm is proposed for reducing the coherence of the equivalent dictionary and then yields an optimal projection matrix. The simulation results illustrate that the proposed method yields a projection matrix which has a better performance in terms of signal recovery accuracy than the methods in [26].
- An innovated cost function is given to design $\boldsymbol{\Phi}$ when the signal has sparse representation error. As can be seen, such a projection matrix can significantly improve the signal recovery accuracy and outperforms the existing approaches when the signal cannot be represented exactly sparse.

The rest of this paper is arranged as follows. An equivalent cost function for (5) is shown in Section 2. It should be noted that the cost function is different from the one proposed in [21,22,26]. Due to the special structure, a novel algorithm is developed to efficiently solve the optimization problem. The detailed discussions about this

[6] In this paper, we call $\boldsymbol{x}$ is exactly sparse in $\boldsymbol{\Psi}$ means there exists some $\boldsymbol{\theta}$ with $\|\boldsymbol{\theta}\|_0 = S$ exactly. In this case, we say there is no sparse representation error of $\boldsymbol{x}$.

novel algorithm can be found in Section 3. An extension of designing projection matrix to the case with sparse representation error is shown in Section 4. Detailed discussions about how this cost function is developed and how this problem is efficiently solved can be also found in Section 4. In Section 5, simulations are given to illustrate the effectiveness and efficiency of the methods shown in this paper. The conclusions are given in Section 6 to end this paper.

2. Preliminaries and problem formulation

Following [20], we define the maximal mutual coherence of the matrix $\boldsymbol{D}$ as

$$\mu_{mx}(\boldsymbol{D}) \triangleq \max_{1 \le i \ne j \le L} \frac{|\boldsymbol{D}(:,i)^{\mathcal{T}} \boldsymbol{D}(:,j)|}{\|\boldsymbol{D}(:,i)\|_2 \|\boldsymbol{D}(:,j)\|_2}. \tag{6}$$

Denote

$$\boldsymbol{S}_c \triangleq \mathrm{diag}(\boldsymbol{G}(1,1)^{-1/2} \cdots \boldsymbol{G}(k,k)^{-1/2} \cdots \boldsymbol{G}(L,L)^{-1/2})$$

Then, the Gram matrix of $\overline{\boldsymbol{D}} = \boldsymbol{D}\boldsymbol{S}_c$, denoted as $\overline{\boldsymbol{G}}$, is normalized such that $\overline{\boldsymbol{G}}(k,k) = 1, \forall k$. Obviously, $\mu_{mx}(\boldsymbol{D}) = \max_{i \ne j} |\overline{\boldsymbol{G}}(i,j)|$. The lower and upper bounds of $\mu_{mx}(\boldsymbol{D})$ are [23]

$$\underline{\mu} \le \mu_{mx}(\boldsymbol{D}) \le 1$$

with $\underline{\mu} \triangleq \sqrt{\frac{L-M}{M(L-1)}}$. The average mutual coherence used in [20] is given by

$$\mu_{av}(\boldsymbol{D}) \triangleq \frac{\sum_{\forall (i,j) \in H_{av}} |\overline{\boldsymbol{G}}(i,j)|}{N_{av}} \tag{7}$$

where N_{av} is the number of entries in the index set $H_{av} \triangleq \{(i,j): |\overline{\boldsymbol{G}}(i,j)| \ge \zeta \& i \ne j\}$ with $0 \le \zeta < 1$ that we can choose as desired. Both μ_{av} and μ_{mx} are used as the measures of projection matrix design in this paper.

Frames are an important concept in signal analysis. We refer the reader to [29,30] for the detailed discussions on frames. Here, we only introduce two important classes of frames: tight frames and ETFs, which are necessary for our later analysis. The sequence of column vectors $\boldsymbol{D}(:,j)$ of matrix $\boldsymbol{D}$ is a frame for the Hilbert space $\Re^N$ if there exist two constants $0 < \alpha \le \beta < +\infty$ such that

$$\alpha \|\boldsymbol{v}\|_2 \le \|\boldsymbol{D}^{\mathcal{T}} \boldsymbol{v}\|_2 \le \beta \|\boldsymbol{v}\|_2 \tag{8}$$

for all $\boldsymbol{v} \in \Re^{N \times 1}$. Such a frame is called α-tight if $\alpha = \beta$ in (8).

A normalized frame $\{\boldsymbol{D}(:,j)\}$ (i.e., $\|\boldsymbol{D}(:,j)\|_2 = 1, \forall j$) is said to be equiangular if

$$|\boldsymbol{D}(:,i)^{\mathcal{T}} \boldsymbol{D}(:,j)| = c, \quad \forall i \ne j$$

where c is a positive constant. As shown in [23], a matrix $\boldsymbol{D}$ with $\|\boldsymbol{D}(:,j)\|_2 = 1, \forall j$ achieves $\mu_{mx}(\boldsymbol{D}) = \underline{\mu}$ *if and only if* $\{\boldsymbol{D}(:,j)\}$ is ETF, moreover, $\mu_{mx}(\boldsymbol{D}) = \underline{\mu}$ can only hold if $L \le M(M+1)/2$. It is clear that the orthogonal bases form a special class of ETFs. An ETF has a very nice average mutual coherence behavior and has been used in optimal dictionary design [24]. In addition, such an idea was also extended to the optimal projection matrix design [25,26].

The projection matrix design problem based on the ETF structure can be formulated as [25,26] [7]

$$\begin{aligned} &\min_{\boldsymbol{G}_t, \boldsymbol{\Phi}} \|\boldsymbol{G}_t - \boldsymbol{\Psi}^{\mathcal{T}} \boldsymbol{\Phi}^{\mathcal{T}} \boldsymbol{\Phi} \boldsymbol{\Psi}\|_F^2 \\ &\boldsymbol{G}_t \in H_\zeta \\ &H_\zeta = \{\boldsymbol{G}_t | \boldsymbol{G}_t = \boldsymbol{G}_t^{\mathcal{T}}, \mathrm{diag}(\boldsymbol{G}_t) = \boldsymbol{1}, \max_{i \ne j} |\boldsymbol{G}_t(i,j)| \le \zeta\} \end{aligned} \tag{9}$$

where ζ is a parameter to control the maximal non-diagonal value in $\boldsymbol{G}_t$, diag $(\boldsymbol{G}_t)$ denotes the diagonal entries of $\boldsymbol{G}_t$ and $\boldsymbol{1}$ represents a vector of proper dimension with its elements all equal to 1. It can be found that (9) is a highly nonconvex problem. The alternating method has been widely used to address such problems [23,24,26]. The basic idea is to update $\boldsymbol{\Phi}$ and $\boldsymbol{G}_t$ alternatively. Firstly, the projection matrix $\boldsymbol{\Phi}$ is fixed and $\boldsymbol{G}_t$ is updated. Secondly, $\boldsymbol{G}_t$ is fixed and the projection matrix $\boldsymbol{\Phi}$ is updated. This proceeding will be repeated until a stop criterion is reached. At the first stage, we directly project the Gram matrix of $\boldsymbol{\Phi}\boldsymbol{\Psi}$ to the set H_ζ. The key problem is the second step. Abolghasemi et al. [26] proposed a gradient descent method to address the problem in this step. For convenience, the procedures of solving (9) in [26] is denoted as Algorithm 1 and shown as follows.

Algorithm 1. AFS.

Initialization:
Sparsify dictionary matrix $\boldsymbol{\Psi}$, initialize $\boldsymbol{\Phi}_0$ to a random matrix, and the number of iterations: $Iter_{outer}$.
Output:
Projection matrix $\boldsymbol{\Phi}$.
Iteration:

```
Φ ← Φ_0
for l = 1 to Iter_outer do
    (G ← Ψ^T Φ^T Φ Ψ)
    Update G_t:
    diag(G_t) ← 1
    for all i ≠ j do
        G_t(i,j) ← { G(i,j)              if |G(i,j)| ≤ ζ
                   { ζ · sign(G(i,j))    otherwise
    where sign (·) is a sign function.
    end for
    Update Φ:
    Φ ← Φ̃ = arg min_Φ ||G_t − Ψ^T Φ^T Φ Ψ||_F^2
end for
```

(10)

Note that the gradient descent algorithm can only reach a local minimum for (10). Moreover, as pointed out by the authors of [26], the accuracy of the solution for (10) will affect the final performance in terms of signal recovery accuracy. Hence, in order to reach a good solution, a large number of iterations is needed for the gradient descent algorithm while the step size must be chosen carefully. Meanwhile, the complexity of the algorithm is increased. According to these observations, a new algorithm is proposed in this paper which can yield a better solution for (10) with a much lower complexity when the dimension of the problem is not very large. In order to carry out the proposed algorithm, some manipulations should be taken to cast (10) to a processable equation. The related manipulations are shown as follows.

[7] In the case of $\boldsymbol{G}_t = \boldsymbol{I}$, this problem is equivalent to (5).

Without loss of generality, assume that the rank and the SVD of $\boldsymbol{\Psi} \in \Re^{M\times L}$ are $\overline{N}$ and

$$\boldsymbol{\Psi} = \boldsymbol{U}_{\boldsymbol{\Psi}} \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\Psi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}_{\boldsymbol{\Psi}}^{\mathcal{T}},$$

respectively. So (10) can be casted to

$$\begin{aligned} &\left\| \boldsymbol{V}_{\boldsymbol{\Psi}}^{\mathcal{T}} \boldsymbol{G}_t \boldsymbol{V}_{\boldsymbol{\Psi}} - \begin{bmatrix} \Sigma_{\boldsymbol{\Psi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{\Phi}}^{\mathcal{T}} \overline{\boldsymbol{\Phi}} \begin{bmatrix} \Sigma_{\boldsymbol{\Psi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right\|_F^2 \\ &= \left\| \begin{bmatrix} \overline{\boldsymbol{G}}_t^{11} & \overline{\boldsymbol{G}}_t^{12} \\ \overline{\boldsymbol{G}}_t^{21} & \overline{\boldsymbol{G}}_t^{22} \end{bmatrix} - \begin{bmatrix} \Sigma_{\boldsymbol{\Psi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Phi}}_1^{\mathcal{T}} \\ \overline{\boldsymbol{\Phi}}_2^{\mathcal{T}} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\Phi}}_1 & \overline{\boldsymbol{\Phi}}_2 \end{bmatrix} \begin{bmatrix} \Sigma_{\boldsymbol{\Psi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right\|_F^2 \\ &\Leftrightarrow \left\| \overline{\boldsymbol{G}}_t^{11} - \boldsymbol{\Sigma}_{\boldsymbol{\Psi}} \overline{\boldsymbol{\Phi}}_1^{\mathcal{T}} \overline{\boldsymbol{\Phi}}_1 \boldsymbol{\Sigma}_{\boldsymbol{\Psi}} \right\|_F^2 \\ &= \left\| \overline{\boldsymbol{G}}_t^{11} - \boldsymbol{\Omega}^{\mathcal{T}} \boldsymbol{\Omega} \right\|_F^2 \end{aligned}$$

Due to these manipulations, we conclude that the solution of (10) is equivalent to solving

$$\min_{\boldsymbol{\Omega}} \left\| \overline{\boldsymbol{G}}_t^{11} - \boldsymbol{\Omega}^{\mathcal{T}} \boldsymbol{\Omega} \right\|_F^2 \tag{11}$$

In the next section, an efficient method is derived to solve (11).

3. An algorithm for the optimum projection matrix

In this section, we first investigate (11) which is the main part in the optimum projection matrix design problem. As to be seen, a novel iterative method is proposed to solve (11). Based on this method, we propose a new algorithm for designing optimum projection matrix.

3.1. An efficient method for solving (11)

The strategy is to update $\boldsymbol{\Omega}$ row by row, which has also been used in [7,21]. Rewrite $\boldsymbol{\Omega}^{\mathcal{T}}\boldsymbol{\Omega} = \sum_i \omega_i \omega_i^{\mathcal{T}}$ with ω_i being the i-th row of $\boldsymbol{\Omega}$. Now, (11) in terms of ω_j can be reformulated as

$$\varrho(\boldsymbol{\omega}_j) = \| \overline{\boldsymbol{G}}_t^{11} - \sum_{i\neq j} \boldsymbol{\omega}_i \boldsymbol{\omega}_i^{\mathcal{T}} - \boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathcal{T}} \|_F^2 = \| \boldsymbol{E}_j - \boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathcal{T}} \|_F^2 \tag{12}$$

where $\boldsymbol{E}_j = \overline{\boldsymbol{G}}_t^{11} - \sum_{i\neq j} \boldsymbol{\omega}_i \boldsymbol{\omega}_i^{\mathcal{T}}$. The derivative of $\varrho(\boldsymbol{\omega}_j)$ with respect to ω_j is

$$\frac{\partial \varrho(\boldsymbol{\omega}_j)}{\partial \boldsymbol{\omega}_j} = -4\boldsymbol{E}_j \boldsymbol{\omega}_j + 4\boldsymbol{\omega}_j \boldsymbol{\omega}_j^{\mathcal{T}} \boldsymbol{\omega}_j \tag{13}$$

The optimal ω_j should satisfy $\frac{\partial \varrho(\boldsymbol{\omega}_j)}{\partial \boldsymbol{\omega}_j} = 0$, which leads to $-4\boldsymbol{E}_j\boldsymbol{\omega}_j + 4\boldsymbol{\omega}_j\boldsymbol{\omega}_j^{\mathcal{T}}\boldsymbol{\omega}_j = 0$. Equivalently, we have

$$\boldsymbol{E}_j \boldsymbol{\omega}_j = \| \boldsymbol{\omega}_j \|_2^2 \boldsymbol{\omega}_j \tag{14}$$

According to (14), we know ω_j and its energy are one of the eigenvector and the corresponding eigenvalue of $\boldsymbol{E}_j$, respectively. Additionally, the eigenvalue decomposition (ED) of $\boldsymbol{E}_j$ can be expressed as

$$\boldsymbol{E}_j \triangleq \boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^{\mathcal{T}} = \sum_k \lambda_{kj} \boldsymbol{u}_{kj} \boldsymbol{u}_{kj}^{\mathcal{T}}$$

where $\boldsymbol{\Lambda}_j = \mathrm{diag}(\lambda_{1j}, \ldots, \lambda_{kj}, \ldots, \lambda_{\overline{N}j})$ with $\lambda_{kj} \geq \lambda_{(k+1)j}$, $\forall k$, λ_{kj} and $\boldsymbol{u}_{kj}$ are the k-th eigenvalue and the corresponding eigenvector of $\boldsymbol{E}_j$. According to (14), the optimal solution for (12) is $\boldsymbol{\omega}_j = \sqrt{\lambda_{1j}}\boldsymbol{u}_{1j}$ if λ_{1j} is bigger than 0. In practical, this procedure will be repeated until a given number of iterations is reached. This derivative forms the foundation of our method for solving (11), see Algorithm 2 with steps 10–18.

After obtaining $\boldsymbol{\Omega}$, $\overline{\boldsymbol{\Phi}}_1$ is updated by $\boldsymbol{\Omega}\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^{-1}$. Note that $\overline{\boldsymbol{\Phi}}_2$ has nothing to do with the objective function of (10) and is fixed with the initial one. To summary, $\boldsymbol{\Phi}$ is obtained by

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Omega}\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^{-1} & \overline{\boldsymbol{\Phi}}_2 \end{bmatrix} \boldsymbol{U}_{\boldsymbol{\Psi}}^{\mathcal{T}}$$

3.2. Optimum projection matrix design

The same alternating framework as Algorithm 1 is used to handle (9) in this subsection. For convenience, the proposed algorithm for projection matrix design is listed in a pseudocode style and shown in Algorithm 2.

Algorithm 2. Optimum projection matrix design.

Initialization:
Sparsify dictionary matrix $\boldsymbol{\Psi}$, initialize $\boldsymbol{\Phi}_0$ to a random matrix, and the number of iterations: $Iter_{outer}$ and $Iter_{inner}$.
Output:
Projection matrix $\boldsymbol{\Phi}$.
Iteration

1: $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}_0$
2: **for** $l = 1$ **to** $Iter_{outer}$ **do**
3: $\boldsymbol{G}_t \leftarrow \boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}$
4: Update $\boldsymbol{G}_t$:
5: $\mathrm{diag}(\boldsymbol{G}_t) \leftarrow \boldsymbol{1}$
6: for all $i \neq j$ **do**
7: $\boldsymbol{G}_t(i,j) \leftarrow \begin{cases} \boldsymbol{G}_t(i,j) & \text{if } |\boldsymbol{G}_t(i,j)| \leq \zeta \\ \zeta \cdot \mathrm{sign}(\boldsymbol{G}_t(i,j)) & \text{otherwise} \end{cases}$
8: **end for**
9: Update $\boldsymbol{\Phi}$: solving $\min_{\boldsymbol{\Phi}} \| \boldsymbol{G}_t - \boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi} \|_F^2$
10: **for** $k = 1$ **to** $Iter_{inner}$ **do**
11: **for** $J = 1$ **to** M **do**
12: Compute $\boldsymbol{E}_J$
13: Compute λ_{1J} and $\boldsymbol{u}_{1J}$ of $\boldsymbol{E}_J$
14: **if** $\lambda_{1J} > 0$ **then**
15: $\boldsymbol{\omega}_J \leftarrow \sqrt{\lambda_{1J}}\boldsymbol{u}_{1J}$
16: **end if**
17: **end for**
18 **end for**
19: $\overline{\boldsymbol{\Phi}}_1 \leftarrow \boldsymbol{\Omega}\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^{-1}$
20: $\boldsymbol{\Phi} \leftarrow \overline{\boldsymbol{\Phi}}\boldsymbol{U}_{\boldsymbol{\Psi}}^{\mathcal{T}}$
21: **end for**

Comment 3.1:

- The parameter ζ in (9) is set to the lower bound of μ_{mx}, i.e., $\underline{\mu} \triangleq \sqrt{\frac{L-M}{L(M-1)}}$. As shown in [26], different ζ can yield different results. Setting ζ to this lower bound, even it is not the optimal value, can also receive a moderate result. Thus the lower bound is used throughout this paper.
- It should be pointed out that all of the diagonal entries in $\boldsymbol{G}_t$ are set to 1 directly (i.e., we do not normalize the equivalent dictionary during each iteration). The experiments show that the algorithm can also work well without the normalization stage.
- The proposed method for solving (11) shares the same concept as K-SVD. However, as illustrated in step 10 of Algorithm 2, we repeat the iterative procedure for

several more times. We find this step can make the algorithm more robust, that is, any initial value $\boldsymbol{\Phi}$ can converge to the same objective value. Typically, simulations show that $Iter_{inner}=2$ always yields a satisfactory result. Algorithm 2 has a faster convergence than the gradient algorithm shown in [26] in terms of μ_{av}, μ_{mx} versus iteration number. It should be noted that the proposed Algorithm 2 of problems[8] which have the form of $\varrho(\boldsymbol{\Omega})=\|\boldsymbol{G}_0-\boldsymbol{\Omega}^{\mathcal{T}}\boldsymbol{\Omega}\|_F^2$. In [22], the authors proposed a closed-form solution for (11) when $\boldsymbol{G}_t=\boldsymbol{I}$. We find Algorithm 2 can converge to the same solution as suggested in [22] when $\boldsymbol{G}_t=\boldsymbol{I}$. We also show some experiments to indicate this concern. However, the strict proof is left for the future work.

- Our algorithm is an iterative method. The main complexity in Algorithm 2 for each iteration is located at steps 3, 12, 13, 15, 19 and 20. In step 13, the power method can be used, so the complexity reduces to only $O(Iter_{inner}M\overline{N}^2)$ rather than $O(Iter_{inner}M\overline{N}^3)$ for complete ED. For other steps, the flops required are $O(MNL+L^2M)$, $O(Iter_{inner}\overline{N}^2)$, $O(Iter_{inner}\overline{N})$, $O(Iter_{inner}\overline{N}^2)$ and $O(N^2)$, respectively. Note that $Iter_{inner}$ is small and $N\le L$, the complexity for our algorithm is approximately equal to $O(L^2MIter_{outer})$.

4. Extension to the case with sparse representation error

Learning a dictionary from a set of training data is to solve the following problem:

$$\boldsymbol{\Psi}\triangleq\arg\min_{\boldsymbol{\Psi},\{\boldsymbol{\theta}_k\}}\sum_{k=1}^{P}\|\boldsymbol{x}_k-\boldsymbol{\Psi}\boldsymbol{\theta}_k\|_2^2\quad\text{s.t. }\|\boldsymbol{\theta}_k\|_0\le S \tag{15}$$

where $\boldsymbol{x}_k$, $k=1,2,\ldots,P$ are the training sequence of signals and S is the given sparsity level. There exist many efficient algorithms to consider the above problem, such as K-SVD, MOD [7,8], etc. Clearly, the sparse representation error $\boldsymbol{e}_k\triangleq\boldsymbol{x}_k-\boldsymbol{\Psi}\boldsymbol{\theta}_k$ is not nil. Let $\boldsymbol{X}$ and $\boldsymbol{\Theta}$ be the training data and sparse coefficient matrix with

$$\boldsymbol{X}(:,k)=\boldsymbol{x}_k,\boldsymbol{\Theta}(:,k)=\boldsymbol{\theta}_k,\quad\forall k$$

The sparse representation error matrix is then represented by

$$\boldsymbol{E}\triangleq\boldsymbol{X}-\boldsymbol{\Psi}\boldsymbol{\Theta}$$
$$\boldsymbol{e}_k=\boldsymbol{E}(:,k),\quad\forall k$$

If $\boldsymbol{e}_k=\boldsymbol{0}$, we say $\boldsymbol{x}_k$ is exactly sparse. In most practical applications, given the whole training data, dictionary learning is an offline problem. So the sparse representation error matrix $\boldsymbol{E}$ can be obtained when the dictionary learning procedure is finished.

In CS framework, the observed signal $\boldsymbol{Y}$ is of form $\boldsymbol{Y}=\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\Theta}+\boldsymbol{\Phi}\boldsymbol{E}$, where $\boldsymbol{\Phi}\boldsymbol{E}$ is the sparse representation error of $\boldsymbol{Y}$ in $\boldsymbol{\Phi}\boldsymbol{\Psi}$. Note that though $\hat{\boldsymbol{X}}\triangleq\boldsymbol{\Psi}\boldsymbol{\Theta}$ is a satisfactory approximate of $\boldsymbol{X}$, $\|\boldsymbol{\Phi}\boldsymbol{E}\|_F$ can be very large if $\boldsymbol{\Phi}$ is not properly chosen. In order to retrieve $\hat{\boldsymbol{X}}$ from $\boldsymbol{Y}$ accurately, the value $\|\boldsymbol{\Phi}\boldsymbol{E}\|_F$ should be considered in the projection matrix design. Therefore, it is desired to design $\boldsymbol{\Phi}$ in the same way as suggested by (9) and at the same time, to reduce $\|\boldsymbol{\Phi}\boldsymbol{E}\|_F$ as much as possible. To deal with this vector optimization problem, we tend to find the Pareto optimal solution of the following problem:

$$\begin{aligned}&\min_{\boldsymbol{G}_t,\boldsymbol{\Phi}}\|\boldsymbol{G}_t-\boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}\|_F^2+\lambda\|\boldsymbol{\Phi}\boldsymbol{E}\|_F^2\\&\quad\boldsymbol{G}_t\in H_{\underline{\mu}}\\&\quad H_{\underline{\mu}}=\{\boldsymbol{G}_t|\boldsymbol{G}_t=\boldsymbol{G}_t^{\mathcal{T}},\mathrm{diag}(\boldsymbol{G}_t)=\boldsymbol{1},\max_{i\ne j}|\boldsymbol{G}_t(i,j)|\le\underline{\mu}\}\end{aligned}\tag{16}$$

where λ is a regularization parameter which quantifies the trade-off between closeness to ETF and sparse representation error minimization. Clearly, alternating optimization is still suitable to address this problem.

As shown in Appendix A, the problem of $\min_{\boldsymbol{\Phi}}\|\boldsymbol{G}_t-\boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}\|_F^2+\lambda\|\boldsymbol{\Phi}\boldsymbol{E}\|_F^2$ is equivalent to

$$\min_{\boldsymbol{\Phi}}\|\tilde{\boldsymbol{G}}_t-\boldsymbol{\Psi}_{sqr}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\|_F^2 \tag{17}$$

where $\tilde{\boldsymbol{G}}_t=\boldsymbol{\Psi}_{sqr}^{-1}\left(\boldsymbol{\Psi}\boldsymbol{G}_t\boldsymbol{\Psi}^{\mathcal{T}}-\frac{\lambda}{2}\boldsymbol{E}\boldsymbol{E}^{\mathcal{T}}\right)\boldsymbol{\Psi}_{sqr}^{-1}$ and $\boldsymbol{\Psi}_{sqr}=\boldsymbol{U}_{\boldsymbol{\Psi}}\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}\boldsymbol{U}_{\boldsymbol{\Psi}}^{\mathcal{T}}$ with $\boldsymbol{\Psi}_{sqr}\boldsymbol{\Psi}_{sqr}^{\mathcal{T}}=\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathcal{T}}\triangleq\boldsymbol{U}_{\boldsymbol{\Psi}}\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}^2\boldsymbol{U}_{\boldsymbol{\Psi}}^{\mathcal{T}}$. Note that (17) has the same form as (10) and hence can be solved with the method in Section 3.1. Thus, only by replacing $\boldsymbol{G}_t$ with $\tilde{\boldsymbol{G}}_t$ in step 9 of Algorithm 2, the modified algorithm can be utilized to solve (16). For convenience, such procedure for solving (16) is denoted as Algorithm 3. Additionally, Algorithm 3 can be also applied to address the related problems involving any symmetric $\boldsymbol{G}_t$. When $\zeta=0$, which indicates $\boldsymbol{G}_t=\boldsymbol{I}$, Algorithm 4 is used to denote the related procedure that obtains the projection matrix for this case.

5. The simulation experiments

In this section, we show several simulations to illustrate the performance of the proposed algorithms and to compare it with other algorithms. This section is divided into two parts. In the first part, we conduct experiments with synthetic data (which are exactly sparse) to show the merits of Algorithm 2 in terms of the measures μ_{av}, μ_{mx} and the recovery performance of the CS systems. Additional experiments are carried out to demonstrate that Algorithm 2 can converge to a global solution as suggested in **Comment 3.1**. The real images are utilized in the second part to manifest the influence of sparse representation error to the CS systems. The extension in Section 4 is used in this part to illustrate the importance of considering sparse representation error in the optimum projection matrix design for this case. The choice of λ is investigated in this part. For convenience, the projection matrices obtained with the algorithms [20,26,19] are marked by Elad, AFS and Lu, respectively.

The parameters in the above mentioned algorithms are shown as follows and are fixed in all of the experiments. The parameters t, γ for the algorithm proposed by Elad [20] are set to $\underline{\mu}$ and 0.75, respectively. For the algorithm in [26], parameters $\bar{\beta}$, K, ξ are set to 5×10^{-4}, 50 and 10^{-5}, respectively. For Lu, we use the same parameters as they are shown in [19]. The outer iteration number is set to 100 for all of the algorithms. $Iter_{inner}$ is set to 2 in Algorithms 2–4.

[8] In the next section, we will use this algorithm to address a new cost function which is derived to design an optimum projection matrix $\boldsymbol{\Phi}$ when the signal is not exactly sparse.

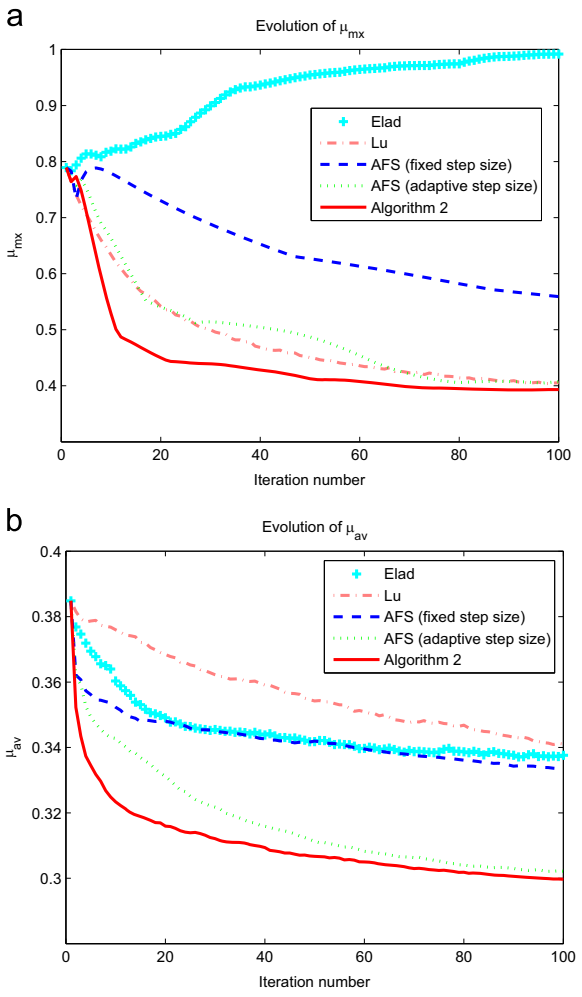

Fig. 1. Evolution of the maximal coherence μ_{mx} and average mutual coherence μ_{av} versus iteration number for $M=15$, $N=60$, $L=80$, respectively.

5.1. Synthetic experiments

An $N \times L$ dictionary $\boldsymbol{\Psi}$ with normally distributed elements[9] and an $M \times N$ random matrix $\boldsymbol{\Phi}_0$ as the initial condition are generated for our simulation.

The measure of μ_{mx} and μ_{av} versus the outer iteration number and the final values of μ_{mx} and μ_{av} obtained for the five CS systems are shown in Fig. 1 and Table 1, respectively.

Comment 5.1:

- As seen from Table 1, Algorithm 2 has the minimal μ_{av} and μ_{mx} compared with other algorithms. This demonstrates the argument that designing the projection matrix under ETF structure can simultaneously control μ_{av} and μ_{mx}. We can also find in the next subsection that the projection matrix obtained by Algorithm 2 yields a better performance in terms of signal recovery accuracy than the projection matrix obtained by Elad and Lu who minimize only μ_{av} or μ_{mx}.
- As shown in Fig. 1, Algorithm 2 has a faster convergence speed than other algorithms in terms of the decrease of μ_{av} and μ_{mx}. Although Algorithm 2 shares the same framework as AFS, the former has a faster convergence than AFS because Algorithm 2 finds a more accurate solution for (10). This can be observed from Fig. 2(a).[10]

Table 1
Evaluation with different measures for each of the five systems ($M=15$, $N=60$, $L=80$). The minimal μ_{mx} and μ_{av} is marked as bold.

Methods	μ_{mx}	μ_{av}
Elad	0.992	0.338
Lu	0.405	0.341
AFS (fixed)	0.559	0.333
AFS (adaptive)	0.407	0.302
Algorithm 2	**0.393**	**0.300**

It is interesting to note that the method illustrated in Section 3.1 for solving (11) converges to a global solution in some specific cases. To demonstrate this standpoint, we set $\boldsymbol{G}_t$ to an identity matrix and design a dictionary matrix which has a full row rank. In this case, [22] showed the global minimum of (5) is $L-M$. It is observed from Fig. 2(b)–(d) that our method converges to the same solution as suggested in [22].

As seen from Fig. 3, Algorithm 2 can make the objective function decrease monotonically which demonstrates that it is stable and the objective value converges.

Now, several experiments are given to examine the signal reconstruction accuracy of the five CS systems. A set of $P=1000$ S-sparse $L \times 1$ vectors $\{\boldsymbol{\theta}_k\}$ $(k=1,2,\ldots,P)$ is produced with a given $N \times L$ random dictionary $\boldsymbol{\Psi}$. Each non-zero entry of $\boldsymbol{\theta}_k$ is randomly positioned with a Gaussian distribution of independent identical distribution (i.i.d.) zero-mean and unit variance. The test signal vectors $\{\boldsymbol{x}_k\}$ are generated by $\boldsymbol{x}_k = \boldsymbol{\Psi}\boldsymbol{\theta}_k$, $\forall k$. The observed signal $\{\boldsymbol{y}_k\}$ is obtained using $\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k$, $\forall k$, where $\boldsymbol{\Phi}$ is the projection matrix taken from the above five projection matrices.

The recovered signal $\hat{\boldsymbol{x}}_k$ is obtained by using $\hat{\boldsymbol{x}}_k = \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_k$, where $\hat{\boldsymbol{\theta}}_k$ is the solution of (2). OMP algorithm is utilized to handle (2).[11] The recovered accuracy is quantified with the mean square error (MSE) defined as [7]

$$\sigma_{mse} \triangleq \frac{1}{N \times P} \sum_{k=1}^{P} \| \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k \|_2^2$$

To explain the performance, two distinct sets of experiments are carried out as follows.

Case I: $M=25$, $N=80$, $L=120$, Fig. 4(a) shows the signal reconstruction accuracy of the CS systems for signal sparsity S varying from 1 to 6.

[9] In this paper, all of the dictionary matrices are normalized that the l_2 norm of all columns in dictionary matrix is equal to 1.

[10] Note that both Algorithm 2 and AFS solve (10) iteratively. Only in Fig. 2(a)–(d), the iteration refers to the process of solving (10). In other figures, the iteration refers to the process of solving (9) and the iteration number represents the number of outer iterations in both Algorithms 1 and 2.

[11] Note that OMP is also harnessed in the next subsection to find the sparsest solution.

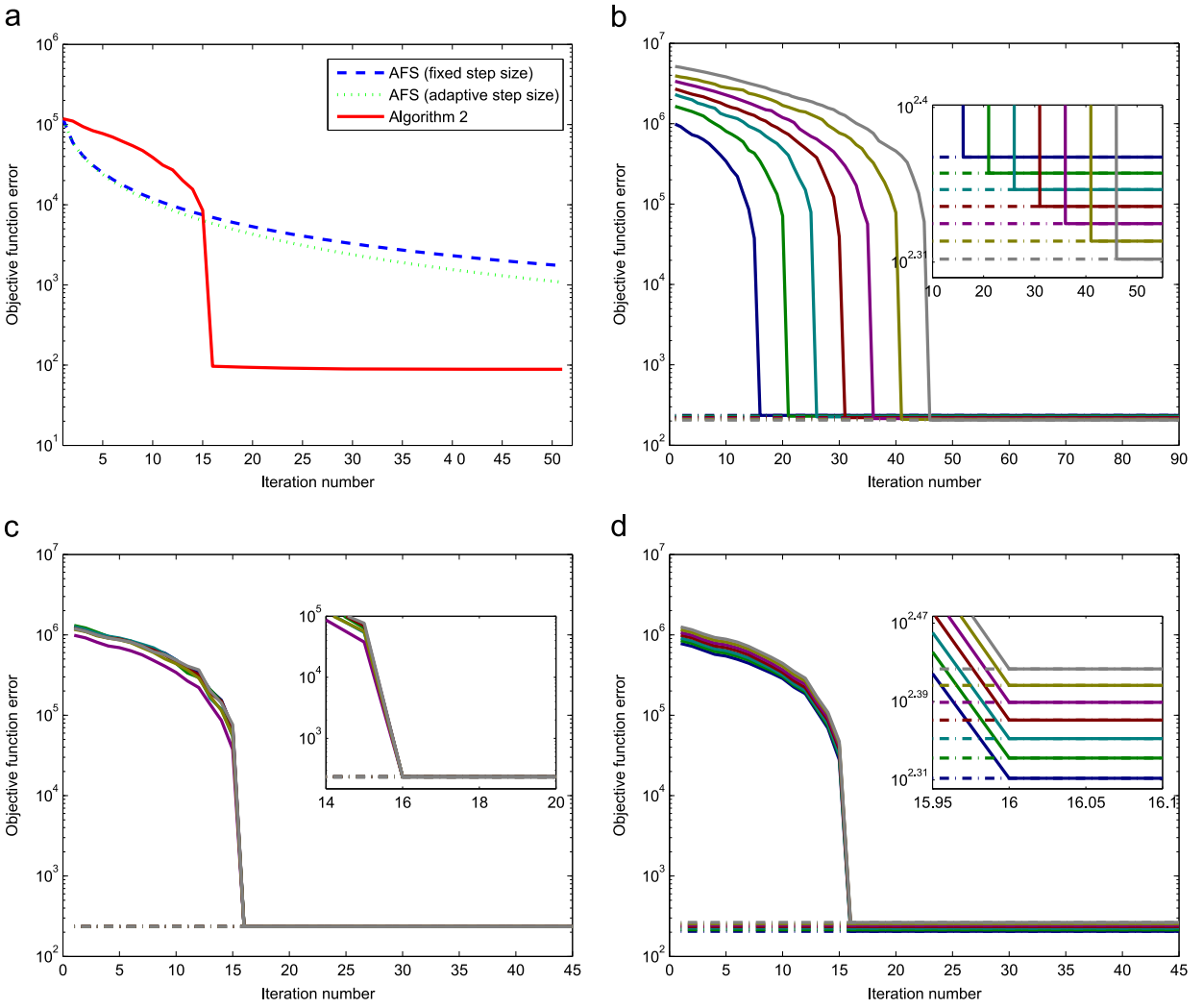

Fig. 2. (a) Evolution of the objective function error of (10) for one of the fixed $\boldsymbol{G}_t$ versus iteration number for $M=15$, $N=60$, $L=80$. (b) Evolution of the objective function error of (5) for $\boldsymbol{G}_t=\boldsymbol{I}$ for $N=100$, $L=250$ and M varying from 15 to 45. (c) Evolution of the objective function error of (5) for $\boldsymbol{G}_t=\boldsymbol{I}$ for $M=15$, $L=250$ and N varying from 60 to 120. (d) Evolution of the objective function error of (5) for $\boldsymbol{G}_t=\boldsymbol{I}$ for $M=15$, $N=100$ and L varying from 220 to 280.

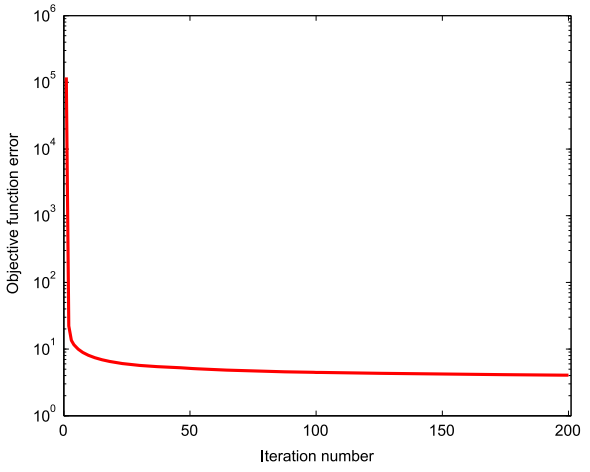

Fig. 3. Evolution of the objective function of (9) for $\|\boldsymbol{G}_t-\boldsymbol{\Psi}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Psi}\|_F^2$ versus iteration number for $M=15$, $N=60$, $L=80$.

Case II: $S=4$, $N=80$, $L=120$, Fig. 4(b) shows the signal reconstruction accuracy of CS systems for projection dimension M varying from 12 to 24.

As seen from Fig. 4(a) and (b), Algorithm 2 outperforms the others in terms of σ_{mse}. As shown in Table 1, Algorithm 2 and AFS (adaptive) have approximately the same values of μ_{mx} and μ_{av}, so they achieve similar recovery performance. However, Algorithm 2 has a faster convergence rate than AFS (adaptive) in decreasing μ_{mx} and μ_{av}, see Fig. 1.

5.2. Real image experiments

In this subsection, we investigate a different situation that there exists signal representation error. The real images are used to examine the performance of the above mentioned CS systems. The classical K-SVD algorithm is utilized to design the sparsifying dictionary $\boldsymbol{\Psi}$ [7]. The data of training and testing is extracted from the LabelMe database [33].

Training data: Firstly, 400 images are selected from the LabelMe training data set. Then 15 non-overlapping patches with size 8×8 are extracted randomly from each image. The patches are rearranged as 64×1 vectors. Finally, 6000 training samples are generated and utilized for sparsifying dictionary learning.

Testing data: The testing data is extracted from the LabelMe testing data set in the same way as for the training data. Once again, we extract 6000 testing samples $\{\boldsymbol{x}_k \in \Re^{N\times 1}\}$ with $N=64$. These testing samples will be used to test the choice of λ.

The recovery accuracy is evaluated in terms of peak signal-to-noise ratio (PSNR) given by [9]

$$\sigma_{psnr} \triangleq 10 \times \log_{10}\left[\frac{(2^r-1)^2}{\sigma_{mse}}\right] \text{dB}$$

with $r=8$ bits per pixel. The dimension of the overcomplete dictionary $\boldsymbol{\Psi}$ is set to 64×256 in the following two subsubsections.

5.2.1. The choice of λ

The trade-off parameter λ plays an important role in Algorithms 3 and 4. However, it is difficult to obtain analytical solution as the guidance for choosing λ. Hence, some experiments are carried out to predetermine a suitable λ for the practical design procedure. The testing data is utilized to conduct the experiments. We choose λ in the range of 0–1 with a step size of 0.01 while the related M varying from 15 to 20. Two different sparsity levels $S=4$ and $S=5$ are tested. Fig. 5 shows the PSNRs with its maximum value normalized to 1 versus λ and M.

As seen from Fig. 5, different sparsity level S, with fixed M and λ, results in different recovery performance. However, we can observe that the figures will become flat when λ exceeds a proper value. Thus though different images may have different optimal λ, a relatively large λ always yields a moderate performance. In practical, we cannot determine an optimal λ for every image. A general λ can be determined through Fig. 5 and we will see the λ chosen this way can also receive optimistic results in the next subsubsection.

In the next part, some natural images are taken to test the performance of Algorithms 3 and 4.

5.2.2. Application to image compression

Six natural images are taken to illustrate the performance of above mentioned CS systems. We choose two different sparsity levels $S=4,5$ and set the sampling dimension M to a fixed value 16. According to Fig. 5, we set the trade-off parameter λ in Algorithms 3 and 4 to 0.5 for all of the two different sparsity levels. The random projection matrix is also utilized to compare with other algorithms and is denoted as Rand. The method mentioned in [21] for designing projection matrix is used in this subsubsection to have a comparison with the extension method proposed in this paper and is denoted as DCS. As shown in [21], DCS has a well performance for the case when the signal representation error exists. So we add it here to have a comparison. For convenience, in what follows, we use Alg 2, Alg3 and Alg4 to represent Algorithm 2, Algorithm 3 and Algorithm 4, respectively. Their performance in terms of σ_{psnr} is shown in Table 2. In order to demonstrate the visual clearly, two images 'Elaine' and 'Boat' are shown in Figs. 6 and 7, respectively.[12]

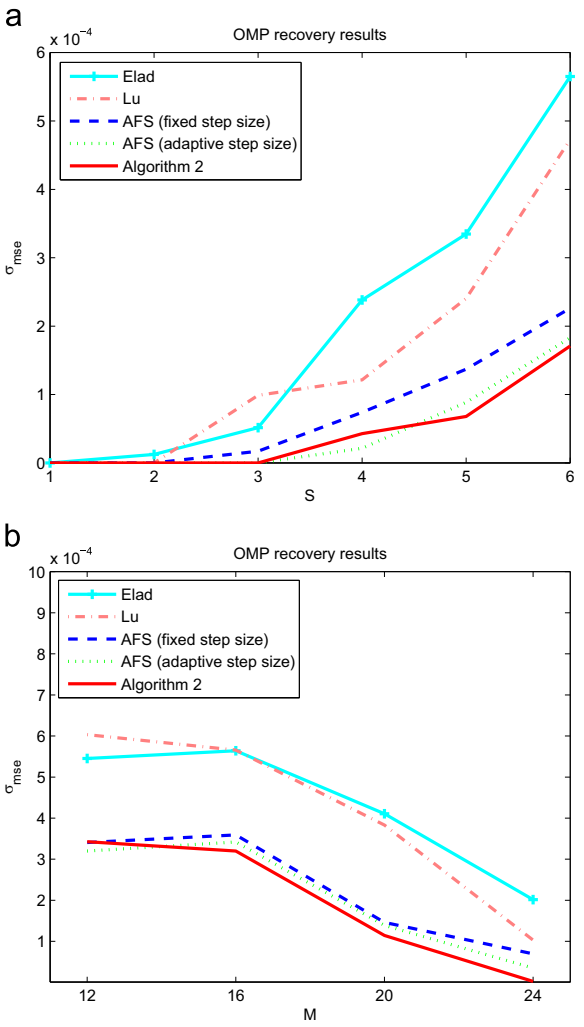

Fig. 4. (a) Performance evaluation: σ_{mse} versus signal sparsity level S varying from 1 to 6 for $M=25$, $N=80$, $L=120$. (b) Performance evaluation: σ_{mse} versus sampling dimension M varying from 12 to 24 for $S=4$, $N=80$, $L=120$.

As seen from Table 2, the PSNRs of the Alg3 are 2 dB better than Alg2 for the tested six images. This illustrates the effectiveness of the penalty function (i.e., sparse representation error) mentioned in Section 4. Clearly, Alg4 has the best performance in terms of PSNR among the CS systems. In this case where the signal is not exactly sparse, some optimum projection matrices, such as Elad, and

[12] Since Algorithm 2 shares the same framework as AFS (fixed or adaptive) and has better performance, we do not display the results for them here.

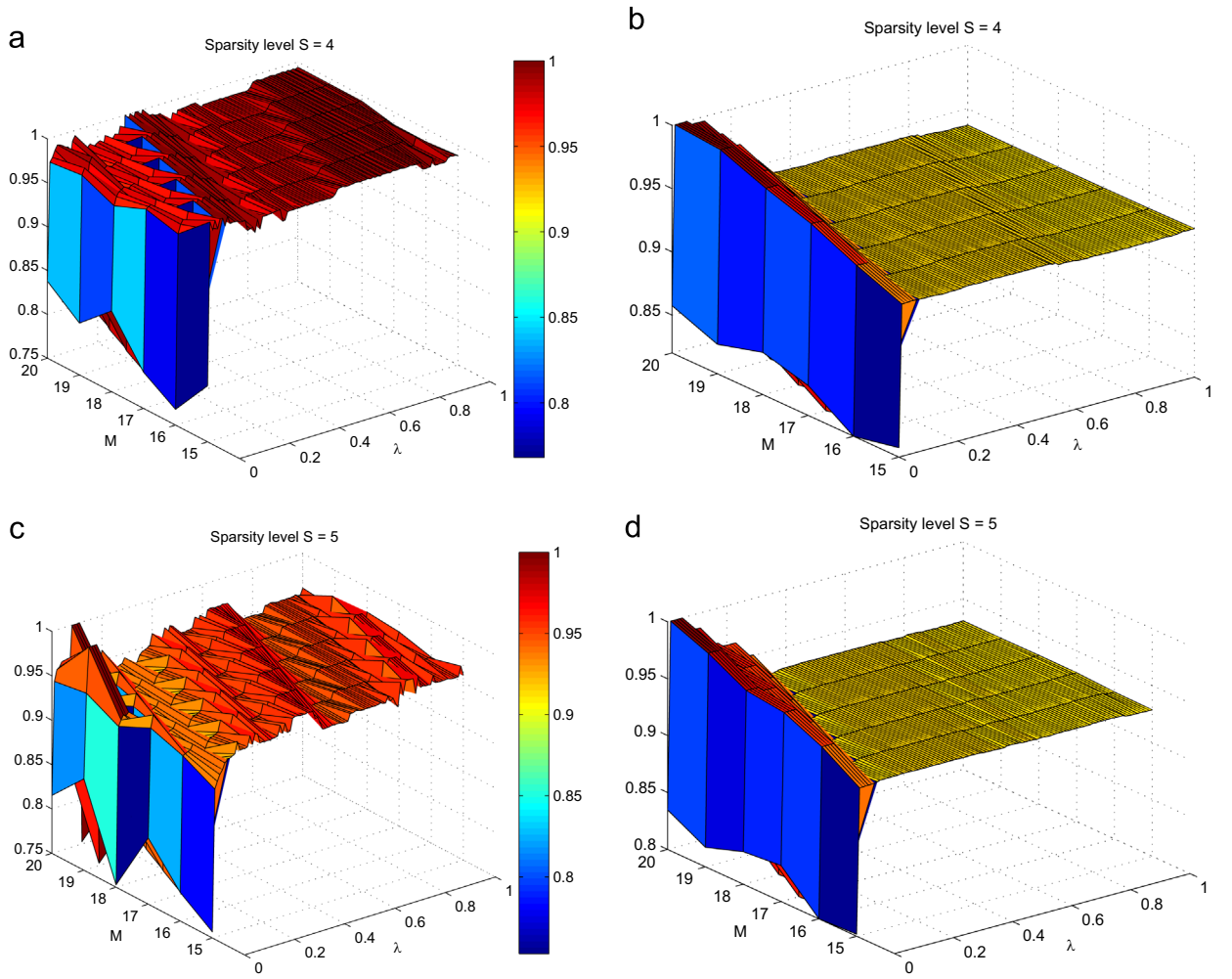

Fig. 5. (a) Testing patches' PSNRs versus λ and M for Algorithm 3 with sparsity level $S=4$. (b) Testing patches' PSNRs versus λ and M for Algorithm 4 with sparsity level $S=4$. (c) Testing patches' PSNRs versus λ and M for Algorithm 3 with sparsity level $S=5$. (d) Testing patches' PSNRs versus λ and M for Algorithm 4 with sparsity level $S=5$.

Table 2
σ_{psnr} for six natural images processed of $M=16$, $N=64$, $L=256$ with $S=4$ and $S=5$. The highest PSNR is marked as bold.

	Lena		Elaine		Boat		Man		Couple		Peppers	
Methods	$S=4$	$S=5$	$S=4$	$S=5$	$S=4$	$S=5$	$S=4$	$S=5$	$S=4$	$S=5$	$S=4$	$S=5$
Rand	27.67	27.19	27.93	27.70	24.83	24.58	26.10	25.77	24.03	23.77	23.25	22.96
Elad	27.26	27.04	23.73	23.78	23.72	23.39	25.10	24.90	23.19	22.80	22.96	22.78
Lu	28.03	28.05	26.16	26.84	24.70	24.68	26.06	26.21	23.66	24.01	23.72	23.64
DCS	28.87	28.30	29.50	29.44	26.17	26.09	27.47	27.33	24.92	25.05	24.53	24.34
Alg 2	26.53	26.76	23.07	23.33	22.89	23.08	24.31	24.46	22.23	22.24	22.56	22.31
Alg 3	28.94	29.30	28.68	28.95	26.18	25.99	27.32	27.43	25.31	25.80	24.81	25.44
Alg 4	**30.85**	**31.01**	**31.08**	**31.08**	**27.81**	**27.86**	**29.10**	**29.18**	**26.92**	**26.98**	**26.66**	**27.07**

Algorithm 2, yield inferior performance than the Rand one. This phenomenon indicates the importance of considering sparse representation error for projection matrix design. The same phenomena can also be observed from Figs. 6 and 7.

6. Conclusion

An efficient algorithm based on alternating optimization is proposed in this paper to design projection matrix. Algorithm 2 converges reasonably faster than other

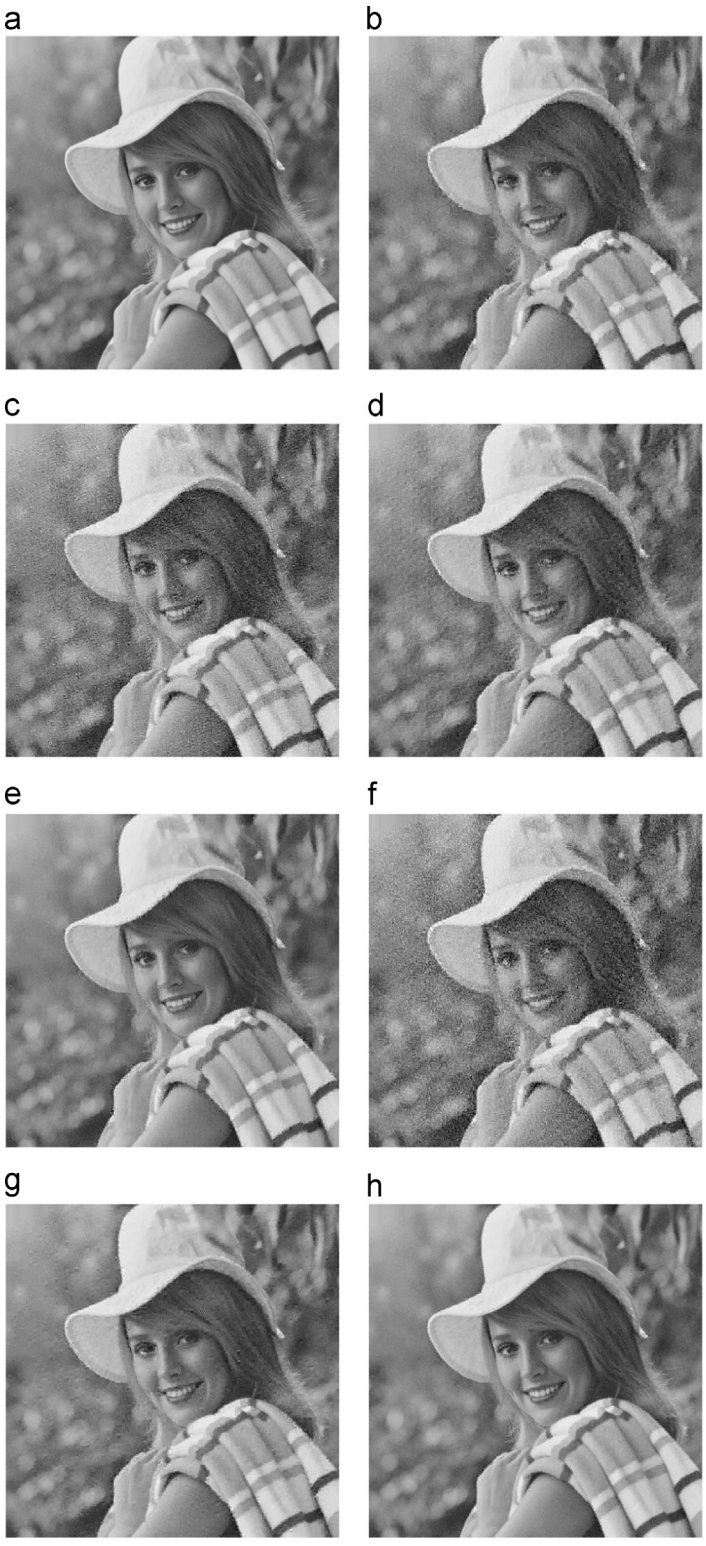

Fig. 6. 'Elaine' and its reconstructed images from the corresponding CS systems with $S = 4$. (a) The original. (b) Rand. (c) Elad. (d) Lu. (e) DCS. (f) Alg 2. (g) Alg 3. (h) Alg 4. The PSNRs are shown in Table 2.

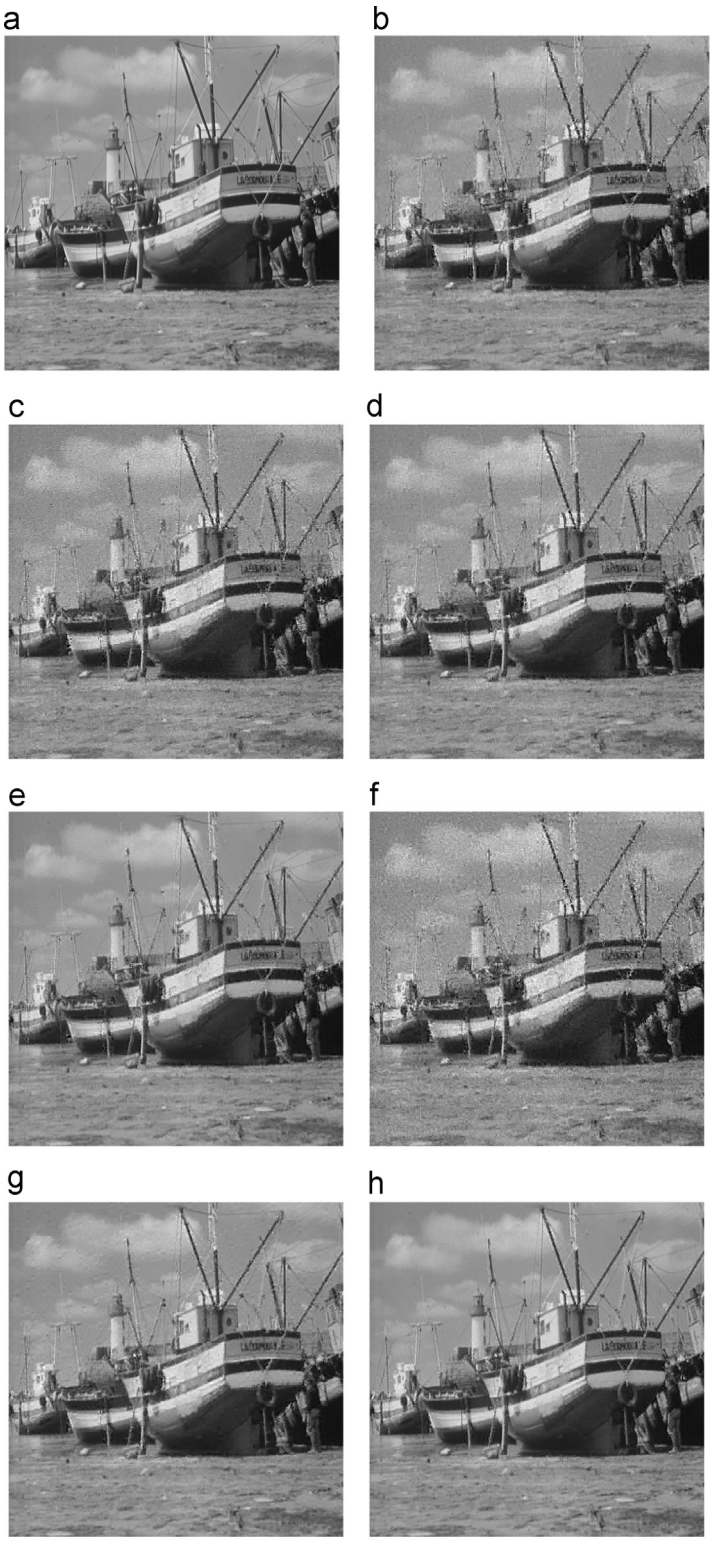

Fig. 7. 'Boat' and its reconstructed images from the corresponding CS systems with $S = 4$. (a) The original. (b) Rand. (c) Elad. (d) Lu. (e) DCS. (f) Alg 2. (g) Alg 3. (h) Alg 4. The PSNRs are shown in Table 2.

algorithms. In order to design a robust projection matrix for the case when the sparse representation error exists, an extension approach is introduced and can be efficiently solved by Algorithm 2. The simulation results demonstrate that Algorithm 2 and the extension methods outperform other algorithms. Moreover, the extension method gives a way to take the sparse representation error into consideration when the signal is not exactly sparse. The related problems about this direction still need more investigation.

Acknowledgment

This work was support by the NSFC-Grants 61273195, 61304124, 61473262, 61503330 and 61503339.

Appendix A. The proof of (17)

Proof. Suppose the dictionary $\boldsymbol{\Psi}$ is *full* rank, which is satisfied in most case. $\boldsymbol{\Psi}=\boldsymbol{U}_{\boldsymbol{\Psi}}[\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}\ \boldsymbol{0}]\boldsymbol{V}_{\boldsymbol{\Psi}}^{\mathcal{T}}$ represents an SVD of $\boldsymbol{\Psi}$. Let $\boldsymbol{\Psi}_{sqr}$ denote the square root of $\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathcal{T}}$; that is

$$\boldsymbol{\Psi}_{sqr}=\boldsymbol{U}_{\boldsymbol{\Psi}}\boldsymbol{\Sigma}_{\boldsymbol{\Psi}}\boldsymbol{U}_{\boldsymbol{\Psi}}^{\mathcal{T}}.$$

We have

$$\begin{aligned}
&\|\boldsymbol{G}_t-\boldsymbol{\Psi}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}\|_F^2+\lambda\|\boldsymbol{\Phi}\boldsymbol{E}\|_F^2\\
&\quad\Leftrightarrow -2\,\mathrm{tr}\left(\boldsymbol{Q}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\right)+\mathrm{tr}\left(\boldsymbol{\Psi}_{sqr}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\right)\\
&\qquad+\lambda\,\mathrm{tr}\left(\boldsymbol{E}\boldsymbol{E}^{\mathcal{T}}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\right)\\
&\quad=\mathrm{tr}\left(-2\boldsymbol{\Psi}_{sqr}^{-1}\left(\boldsymbol{Q}-\frac{\lambda}{2}\boldsymbol{E}^{\mathcal{T}}\boldsymbol{E}\right)\boldsymbol{\Psi}_{sqr}^{-1}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\right.\\
&\qquad\left.+\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\right)\\
&\quad\Leftrightarrow\|\boldsymbol{\Psi}_{sqr}^{-1}\left(\boldsymbol{Q}-\frac{\lambda}{2}\boldsymbol{E}^{\mathcal{T}}\boldsymbol{E}\right)\Psi_{sqr}^{-1}-\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\|_F^2\\
&\quad=\|\tilde{\boldsymbol{G}}_t-\boldsymbol{\Psi}_{sqr}\boldsymbol{\Phi}^{\mathcal{T}}\boldsymbol{\Phi}\boldsymbol{\Psi}_{sqr}\|_F^2
\end{aligned}$$

where $\boldsymbol{Q}=\boldsymbol{\Psi}\boldsymbol{G}_t\boldsymbol{\Psi}^{\mathcal{T}}$ and $\tilde{\boldsymbol{G}}_t=\boldsymbol{\Psi}_{sqr}^{-1}\left(\boldsymbol{Q}-\frac{\lambda}{2}\boldsymbol{E}^{\mathcal{T}}\boldsymbol{E}\right)\boldsymbol{\Psi}_{sqr}^{-1}$. This completes the proof.□

References

[1] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (February) (2006) 489–509.
[2] E.J. Candès, T. Tao, Near optimal signal recovery from random projections: universal encoding strategies, IEEE Trans. Inf. Theory 52 (December) (2006) 5406–5425.
[3] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (April) (2006) 1289–1306.
[4] M. Elad, Sparse and redundant representation modeling—what next? IEEE Signal Process. Lett. 19 (December) (2012) 922–928.
[5] T. Strohmer, Measure what should be measured: progress and challenges in compressive sensing, IEEE Signal Process. Lett. 19 (December) (2012) 887–893.
[6] E.J. Candès, M.B. Wakin, An introduction to compressive sampling, IEEE Signal Process. Mag. 25 (March) (2008) 21–30.
[7] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (November) (2006) 4311–4322.
[8] I. Tošić, P. Frossard, Dictionary learning, IEEE Signal Process. Mag. 28 (March) (2011) 27–38.
[9] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (December) (2006) 3736–3745.
[10] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer Science & Business Media, 2010.
[11] D.L. Donoho, M. Elad, Optimally sparse representation in general (nonorthonormal) dictionaries via l_1 minimization, Proc. Natl. Acad. Sci. 100 (March) (2003) 2197–2202.
[12] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (January) (1998) 33–61.
[13] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory 53 (December) (2007) 4655–4666.
[14] T. Robert, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Methodol. 58 (January) (1996) 267–288.
[15] G. Chen, D.F. Li, J.S. Zhang, Iterative gradient projection algorithm for two-dimensional compressive sensing sparse image reconstruction, Signal Process. 104 (November) (2014) 15–26.
[16] D.L. Donoho, H. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Inf. Theory 47 (November) (2001) 2845–2862.
[17] J. Zhang, C. Zhao, D.B. Zhao, W. Gao, Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization, Signal Process. 103 (October) (2014) 114–126.
[18] S. Stanković, I. Orović, L. Stanković, An automated signal reconstruction method based on analysis of compressive sensed signals in noisy environment, Signal Process. 104 (November) (2014) 43–50.
[19] W.-S. Lu, T. Hinamoto, Design of projection matrix for compressive sensing by nonsmooth optimization, in: IEEE International Symposium on Circuits and Systems (ISCAS), June 2014, pp. 1279–1282.
[20] M. Elad, Optimized projections for compressed sensing, IEEE Trans. Signal Process. 55 (December) (2007) 5695–5702.
[21] J.M. Duarte-Carvajalino, G. Sapiro, Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization, IEEE Trans. Image Process. 18 (July) (2009) 1395–1408.
[22] L.Z. Manor, K. Rosenblum, Y.C. Eldar, Sensing matric optimization for block-sparse decoding, IEEE Trans. Signal Process. 59 (September) (2011) 4300–4312.
[23] J.A. Tropp, I.S. Dhillon, R.W. Heath Jr., T. Strohmer, Designing structured tight frames via an alternative projection method, IEEE Trans. Inf. Theory 51 (January) (2005) 188–209.
[24] M. Yaghoobi, L. Daudet, M.E. Davies, Parametric dictionary design for sparse coding, IEEE Trans. Signal Process. 57 (December) (2009) 4800–4810.
[25] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, S. Sanei, On optimization of the measurement matrix for compressive sensing, in: European Signal Process Conference, EUSIPCO, Aalborg, Denmark, August 2010.
[26] V. Abolghasemi, S. Ferdowsi, S. Sanei, A gradient-based alternative minimization approach for optimization of the measurement matrix in compressive sensing, Signal Process. 94 (April) (2012) 999–1009.
[27] G. Li, Z.H. Zhu, D.H. Yang, L.P. Chang, H. Bai, On projection matrix optimization for compressive sensing systems, IEEE Trans. Signal Process. 61 (June) (2013) 2887–2898.
[29] O. Christensen, Introduction to Frames and Riesz Bases, Birkhäuser, Boston, MA, USA, 2003.
[30] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, 3rd ed. Academic Press, 2008.
[33] B. Russell, A. Torralba, W.T. Freeman, Labelman, The Open Annotation Tool MIT, Computer Science and Artificial Intelligence Laboratory [Online]. Available from: ⟨http://labelme.csail.mit.edu⟩.