PAPER

Optimal Permutation Based Block Compressed Sensing for Image Compression Applications

Yuqiang CAO[†a], **Weiguo GONG**[†b], **Bo ZHANG**[††c], ***Nonmembers***, **Fanxin ZENG**[††d], ***Member***, ***and*** **Sen BAI**[††e], ***Nonmember***

SUMMARY Block compressed sensing (CS) with optimal permutation is a promising method to improve sampling efficiency in CS-based image compression. However, the existing optimal permutation scheme brings a large amount of extra data to encode the permutation information because it needs to know the permutation information to accomplish signal reconstruction. When the extra data is taken into consideration, the improvement in sampling efficiency of this method is limited. In order to solve this problem, a new optimal permutation strategy for block CS (BCS) is proposed. Based on the proposed permutation strategy, an improved optimal permutation based BCS method called BCS-NOP (BCS with new optimal permutation) is proposed in this paper. Simulation results show that the proposed approach reduces the amount of extra data to encode the permutation information significantly and thereby improves the sampling efficiency compared with the existing optimal permutation based BCS approach.

key words: *block compressed sensing, optimal permutation, image compression, image coding*

1. Introduction

Recently, compressed sensing (CS) has attracted more and more attention in signal processing field [1]–[3], which states that sparse or compressible signals can be exactly recovered from a small number of random measurements. Natural images are well known to be compressible in some transform domain, such as wavelet domain, discrete cosine transformation (DCT) domain, and overcomplete dictionaries. This means that natural images can be compressed by using CS. However, it has been shown that traditional CS combined with ordinary quantization is not a good compression technique in view of compression efficiency [4]. Nonetheless, the CS-based image compression has become a hot topic in recent years, because it has been demonstrated very efficiency in some special image compression applications, such as robust image coding [5] and encrypted image compression [6].

Usually, CS is applied to one-dimensional (1D) signals. In order to use CS to encode two-dimensional (2D) image signals, we can reshape the 2D image to a long vector firstly, and then sample the vector-reshaped image. An example of such sampling scheme with a random measurement matrix is the single-pixel camera proposed in [7]. Since each measurement is a linear combination of all pixels of the image to be sensed, we call the method proposed in [7] as global CS. However, the global CS faces two main challenges. First, the computational complexity of codec is very expensive. Second, it requires massive storage space for the random measurement matrix.

To solve the above problems, block CS (BCS) [8]–[12] is proposed. The basic idea of BCS is that we can divide the 2D image into small blocks and then sample these vector-reshaped blocks individually by using a same measurement matrix. BCS schemes save the memory storage for the sampling matrix and computational complexity for codec very significantly, which makes the application of CS theory in image compression applications more practical. However, in traditional BCS schemes [8]–[12], the sampling rate of each block is always identical without considering the sparsity level differences among the blocks. Therefore, it is ineffective to encode the blocks of nature images via a fixed sampling rate directly in view of sampling efficiency.

One of effective methods to solve the above problem is the permutation-based BCS [13]–[16]. The central concept is straightforward: we rearrange the 2D signal which is composed of the transform coefficients of the nature image by using permutation strategy and then sample the permutated 2D signal by using block-based sampling. A good permutation strategy can make the nonzero entries evenly distributed among the blocks, i.e., the maximum block sparsity level of the 2D signal after permutation is far smaller than that before, thereby improving sampling efficiency. Thus, the crux is finding a good permutation strategy. There are two feasible permutation strategies: random permutation [13]–[15] and optimal permutation [16]. It has been shown that the optimal permutation proposed in our past work [16] has better performance than random permutation, but it brings a large amount of extra data to encode the permutation information because it needs to know the permutation information to accomplish signal reconstruction. When the extra data is taken into consideration, the improvement in sampling efficiency of this method is limited.

In order to solve this problem, an improved optimal permutation based BCS strategy called BCS-NOP (BCS with new optimal permutation) is proposed in this paper.

Manuscript received March 12, 2017.
Manuscript revised September 3, 2017.
Manuscript publicized October 20, 2017.

[†]The authors are with Chongqing University, Chongqing, 400044 China.
[††]The authors are with Chongqing Communication Institute, Chongqing, 400035 China.
a) E-mail: caoyuqiang163@163.com
b) E-mail: wggong@cqu.edu.cn
c) E-mail: zhangboswjtu@163.com
d) E-mail: baisencq@126.com
e) E-mail: fzengx@yahoo.com.cn

DOI: 10.1587/transinf.2017EDP7087

Firstly, we propose a new block-based sampling model for image signals in which optimal permutation strategy and reweighted sampling strategy are applied simultaneously. In our proposed method, the sampling efficiency depends on the permutation matrices used in permutation operation. Then, an optimal permutation matrix generating algorithm is proposed, which can be used to reduce the maximum block sparsity level of the 2D signal significantly. As a result, better sampling efficiency and/or reconstructed-images quality can be achieved. Finally, simulation results show that the proposed approach reduces the amount of extra data to encode the permutation information significantly and thereby improves the sampling efficiency compared with the existing optimal permutation based BCS approach.

The rest of this paper is organized as follows. Section 2 introduces the theoretical backgrounds of CS. A BCS-NOP method for image compression applications is proposed in Sect. 3. Simulation results are presented in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2. Overview of Compressed Sensing

2.1 Traditional Compressed Sensing

For the convenience of elaboration, we denote the set of all K-sparse vectors by

$$\Sigma_K = \{\boldsymbol{x} \in R^{N\times 1} \mid \|\boldsymbol{x}\|_0 \leq K\} \tag{1}$$

where $\|\cdot\|_0$ stands for l_0-norm of a vector, i.e., the number of nonzero entries of the vector.

The basic idea of CS theory is that we can recover sparse signals from only a small set of random measurements. Consider a K-sparse signal $\boldsymbol{x}$, and let $\boldsymbol{\Phi}$ be an $M \times N$ ($M \ll N$) measurement matrix, then the measurement vector of $\boldsymbol{x}$ can be obtained by

$$\boldsymbol{y} = \boldsymbol{\Phi x} \tag{2}$$

The CS theory asserts that $\boldsymbol{y}$ can be used to recover any "sparse enough" signal efficiently provided that the measurement matrix satisfies restricted isometry property (RIP) [1], [2].

The reconstruction of sparse signals from random measurements can be achieved by solving the following l_1-norm minimization problem

$$\hat{\boldsymbol{x}} = \arg\min \|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi x} \tag{3}$$

This is a convex optimization problem that conveniently reduces to a linear program known as basis pursuit (BP) [17]. Other related reconstruction algorithms, including orthogonal matching pursuits (OMP) [18], iterative gradient projection algorithm [19], and iterative hard thresholding (IHT) algorithm [20], can also be used to solve the l_1-norm minimization problem effectively.

2.2 Block Compressed Sensing

Consider an image $\boldsymbol{D} \in R^{\sqrt{N}\times\sqrt{N}}$ with N pixels, which can be represented by sparsifying transformation

$$\boldsymbol{X} = \boldsymbol{\Psi D \Psi}^{\mathrm{T}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1\sqrt{N}} \\ x_{21} & x_{22} & \cdots & x_{2\sqrt{N}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\sqrt{N}1} & x_{\sqrt{N}2} & \cdots & x_{\sqrt{N}\sqrt{N}} \end{bmatrix} \tag{4}$$

where $\boldsymbol{\Psi} \in R^{\sqrt{N}\times\sqrt{N}}$ is an sparsifying basis, $(\cdot)^{\mathrm{T}}$ denotes the transposition of a matrix, $\boldsymbol{X} \in R^{\sqrt{N}\times\sqrt{N}}$ is the coefficient matrix of $\boldsymbol{D}$ and x_{ij} is the element of $\boldsymbol{X}$ located in (i, j).

A 2D signal is said to be sparse if most of the coefficients of $\boldsymbol{X}$ are zero or they can be discarded without much loss of "information." Let $\boldsymbol{X}_K$ be the 2D signal where only the K largest coefficients of $\boldsymbol{X}$ are kept and the rest are set to zero. When properly selecting K, $\boldsymbol{X}$ can be well approximated by $\boldsymbol{X}_K$. Natural images are known to be sparse or compressible when represented on an appropriated sparsifying basis, such as wavelet basis, DCT basis or overcomplete dictionaries. Therefore, natural images can be compressed by using CS.

Usually, CS is applied to 1D signals. In order to use CS to encode 2D image signals, we can reshape the 2D image to a long vector firstly, and then sample the vector-reshaped image. Since each measurement is a linear combination of all pixels of the image to be sensed, we call this method as global CS. However, the global CS faces two main challenges. First, the computational complexity of codec is very expensive. Second, it requires massive storage space for the random measurement matrix.

The BCS scheme is proposed to solve the above problems. In the BCS scheme, the 2D signal $\boldsymbol{X}$ is divided into small blocks with size of $\sqrt{n} \times \sqrt{n}$. The block matrix form of $\boldsymbol{X}$ can be written as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_{\sqrt{L}} \\ \boldsymbol{X}_{\sqrt{L}+1} & \boldsymbol{X}_{\sqrt{L}+2} & \cdots & \boldsymbol{X}_{2\sqrt{L}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{X}_{L-\sqrt{L}+1} & \boldsymbol{X}_{L-\sqrt{L}+2} & \cdots & \boldsymbol{X}_L \end{bmatrix} \tag{5}$$

where $\boldsymbol{X}_i$ ($i = 1, 2, \ldots, L$) is i-th block of $\boldsymbol{X}$ and $L = N/n$. Without any loss of generality, we assume that L is an integral.

Let K_i be the sparsity level of $\boldsymbol{X}_i$. Then, the block sparsity level of $\boldsymbol{X}$ can be denoted by a sparsity level vector $\boldsymbol{K} = [K_1, K_2, \ldots, K_L]$. Let $\bar{K} = \lceil K/L \rceil$ be the average block sparsity level and $K_{\max} = \|\boldsymbol{K}\|_\infty$ be the maximum block sparsity level, here $\lceil\cdot\rceil$ denotes the ceiling function, and $\|\cdot\|_\infty$ stands for the Chebyshev norm of a vector. The Chebyshev norm of a vector is equal to the largest magnitude of the entries in the vector.

Let $\boldsymbol{x}_i$ be the vectorized signal of the i-th block through raster scanning. Then, the measurement vector of $\boldsymbol{x}_i$ can be obtained by

$$\boldsymbol{y}_i = \boldsymbol{\Phi}_B \boldsymbol{x}_i \tag{6}$$

where $\boldsymbol{\Phi}_B \in R^{m\times n}$ is a Gaussian random matrix.

If $\boldsymbol{\Phi}_B$ satisfies RIP with order $K_{\max}$, we can recover each block of $\boldsymbol{X}$ separately by solving l_1-norm minimization problem, and then recover the original image $\boldsymbol{D}$ via inverse sparsifying transformation. This reconstruction scheme is called separate reconstruction, because the reconstruction of $\boldsymbol{X}$ is done block-by-block. In order to reduce the computational complexity in decoder side, we adopt separate reconstruction in this paper.

3. The Proposed Method

As stated above, $\boldsymbol{\Phi}_B$ should satisfy RIP with order $K_{\max}$ to guarantee all blocks of $\boldsymbol{X}$ can be recovered precisely by using CS reconstruction. Therefore, when separate reconstruction is used in BCS, the maximum block sparsity level of the 2D signal needs to be taken into consideration with the purpose of improving the sampling efficiency.

3.1 A Motivated Example

In this section, we briefly introduce permutation matrix firstly, and then give a motivated example for permutation-based BCS strategy.

A permutation matrix $\boldsymbol{P} \in \mathrm{R}^{n\times n}$ is a square matrix that has exactly one entry 1 in each row and each column and 0s elsewhere else. Every permutation of the numbers $(1, 2, \ldots, n)$ corresponds to a unique permutation matrix. Therefore, there are $n!$ permutation matrices of size n. Let $J = (J_1, J_2, \ldots, J_n)$ be a permutation of the numbers $(1, 2, \ldots, n)$, then we can use J to produce a permutation matrix $\boldsymbol{P} = [\boldsymbol{e}_{J_1}, \boldsymbol{e}_{J_2}, \ldots, \boldsymbol{e}_{J_n}]$, here $\boldsymbol{e}_{J_i} \in R^n$ denotes a column vector with 1 in the J_i-th position and 0 in every other position. We denote $\boldsymbol{P}_J$ be the permutation matrix produced by J, here the subscript 'J' is used for indicating that $\boldsymbol{P}_J$ is produced by J. We omit the subscript in this paper whenever it will not lead to misunderstanding. It is well known that, when a permutation matrix $\boldsymbol{P}$ is multiplied with a matrix $\boldsymbol{M} \in \mathrm{R}^{n\times n}$ from the left, it will permute the rows of $\boldsymbol{M}$, and when a permutation matrix $\boldsymbol{P}$ is multiplied with $\boldsymbol{M}$ from the right, it will permute the columns of $\boldsymbol{M}$.

In order to better understand why permutation can be used to reduce the maximum block sparsity level of the 2D signal, let us consider the following example.

Example 1: When $N = 64$, the following matrix

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

can be divided into small blocks with size of 4×4. We can rewrite $\boldsymbol{X}$ into its block matrix form

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \\ \boldsymbol{X}_3 & \boldsymbol{X}_4 \end{bmatrix} \tag{8}$$

where, $\boldsymbol{X}_i \in R^{4\times 4}$, $i = 1, 2, \ldots, 4$ is the i-th block of $\boldsymbol{X}$. Apparently, the block sparsity vector of $\boldsymbol{X}$ is $\boldsymbol{K} = [16, 6, 6, 0]$ and the average block sparsity level of $\boldsymbol{X}$ is $\bar{K} = 7$. Since the nonzero entries of $\boldsymbol{X}$ aren't evenly distributed among the blocks, the maximum block sparsity level $K_{\max} = 16$ is far greater than the average sparsity level $\bar{K} = 7$.

In order to make the nonzero entries of the permutated 2D signal evenly distributed among the blocks, we can process $\boldsymbol{X}$ by the following permutation operations: (1) exchanging 2-th with 8-th row, and 3-th with 5-th row of $\boldsymbol{X}$, respectively; (2) exchanging 2-th with 8-th column, and 4-th with 6-th column of $\boldsymbol{X}$, respectively. Thus, we can get a new 2D signal as

$$\boldsymbol{X}^{\dagger} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{9}$$

The above permutation operations can be achieved by multiplying $\boldsymbol{X}$ with permutation matrices. Let $\boldsymbol{P}_1 = [\boldsymbol{e}_1, \boldsymbol{e}_8, \boldsymbol{e}_5, \boldsymbol{e}_4, \boldsymbol{e}_3, \boldsymbol{e}_6, \boldsymbol{e}_7, \boldsymbol{e}_2]^{\mathrm{T}}$ and $\boldsymbol{P}_2 = [\boldsymbol{e}_1, \boldsymbol{e}_8, \boldsymbol{e}_3, \boldsymbol{e}_6, \boldsymbol{e}_5, \boldsymbol{e}_4, \boldsymbol{e}_7, \boldsymbol{e}_2]$ be the corresponding row permutation matrix and column permutation matrix, respectively. Then, the above permutation operations can be rewritten into the following matrix form

$$\boldsymbol{X}^{\dagger} = \boldsymbol{P}_1 \boldsymbol{X} \boldsymbol{P}_2 \tag{10}$$

We can also rewrite $\boldsymbol{X}^{\dagger}$ into its block matrix form

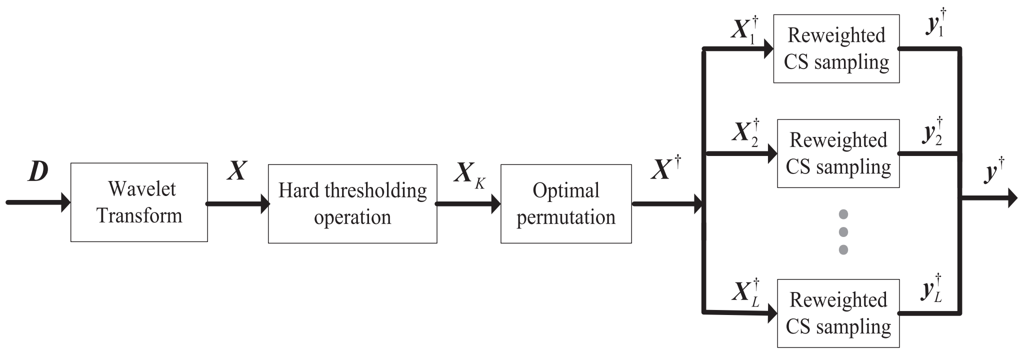
$$\boldsymbol{X}^{\dagger} = \begin{bmatrix} \boldsymbol{X}_1^{\dagger} & \boldsymbol{X}_2^{\dagger} \\ \boldsymbol{X}_3^{\dagger} & \boldsymbol{X}_4^{\dagger} \end{bmatrix} \tag{11}$$

here, $\boldsymbol{X}_i^{\dagger} \in R^{4\times 4}$, $i = 1, 2, \ldots, 4$ is the i-th block of $\boldsymbol{X}^{\dagger}$. Obviously, the block sparsity vector of $\boldsymbol{X}^{\dagger}$ is $\boldsymbol{K} = [7, 7, 7, 7]$. The nonzero entries of $\boldsymbol{X}^{\dagger}$ are evenly distributed among the blocks. As a result, the maximum block sparsity level is equal to the average block sparsity level.

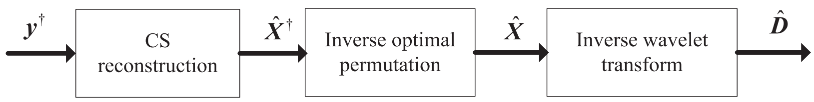
In conclusion, by selecting appropriate permutation matrices in permutation operation, we can make the nonzero entries of the permutated 2D signal evenly distributed among the blocks, which leads to reduce the maximum block sparsity level of the 2D signal significantly. Therefore, we can get better reconstruction performance by adding appropriate permutation operation prior to sampling.

3.2 Overview

In order to make the nonzero entries of the permutated 2D signal evenly distributed among the blocks, a BCS with optimal permutation (BCS-OP) is proposed in our past work [16]. Although BCS-OP improves the sampling efficiency significantly, it still faces some challenges. The most important one is that it brings a large amount of extra data to encode the permutation information because it needs to



(a) Block diagram of encoding



(b) Block diagram of decoding

Fig. 1 Block diagram of the proposed method

know the permutation information to accomplish signal reconstruction. To solve this problem, a new optimal permutation strategy which can significantly reduce extra data of encoding the permutation information is proposed in this paper. In addition, reweighted sampling proposed in [11] is also incorporated into our method with purpose of further improving the sampling efficiency. The proposed method mainly contains four steps:

Step 1 Use orthogonal wavelet basis (Without any loss of generality, we illustrate our algorithm by taking wavelet basis as the sparsifying basis of natural image signals. However, for other types of sparsifying basis, such as DCT, curvelet, or overcomplete dictionaries, our method remains effective.) to transform original image signals. Suppose that $\boldsymbol{X}$ is the corresponding wavelet coefficient matrix.

Step 2 Set small coefficients of the wavelet coefficient matrix to zero by adaptively selecting hard threshold. Let $\boldsymbol{X}_K$ be the 2D signal where only the K largest coefficients of $\boldsymbol{X}$ are kept and the rest are set to zero. Instead of using the constant threshold, we select $K = C_1 M/(\log(N/M) + 1)$ in adaptive manner with the purpose of guaranteeing that the large entries of $\boldsymbol{X}$ can be reconstructed precisely by CS reconstruction, where $M = mL$ is the total number of measurements. We use $C_1 = 0.6$ for the experimental results to follow latter, which we have found to perform well in practice.

Step 3 Process $\boldsymbol{X}_K$ with the following optimal permutation operation

$$\boldsymbol{X}^{\dagger} = \boldsymbol{P}_1 \boldsymbol{X}_K \boldsymbol{P}_2 \tag{12}$$

where $\boldsymbol{P}_1 \in R^{\sqrt{N}\times\sqrt{N}}$ and $\boldsymbol{P}_2 \in R^{\sqrt{N}\times\sqrt{N}}$ are permutation matrices, and $\boldsymbol{X}^{\dagger} \in R^{\sqrt{N}\times\sqrt{N}}$ is the permutated coefficient matrix. By selecting a pair of appropriate permutation matrices in permutation operation, we can make the nonzero entries of the permutated 2D signal evenly distributed among the blocks, which results in improving sampling efficiency. The optimal permutation matrices generating algorithm will be discussed detailed in Sect. 3.3.

Step 4 Sample $\boldsymbol{X}^{\dagger}$ by using reweighted CS. We divide $\boldsymbol{X}^{\dagger}$ into small blocks with size of $\sqrt{n} \times \sqrt{n}$ firstly, and then sample each block with the same measurement matrix. Instead of using the conventional sampling scheme, we consider the following reweighted sampling [11] on each block:

$$\boldsymbol{y}_i^{\dagger} = \mathrm{orth}(\boldsymbol{\Phi}_B)\boldsymbol{W}\boldsymbol{x}_i^{\dagger} \tag{13}$$

where the operation orth stands for orthogonalizing the columns of a matrix, $\boldsymbol{W}$ is a diagonal weighting matrix with weighting coefficients $\{|\boldsymbol{x}_i^{\dagger}(1)|, |\boldsymbol{x}_i^{\dagger}(2)|, \ldots, |\boldsymbol{x}_i^{\dagger}(n)|\}$ on the diagonal and zeros elsewhere, $\boldsymbol{x}_i^{\dagger} \in R^n$ represent the vectorized signal of the i-th block of $\boldsymbol{X}^{\dagger}$ through raster scanning, and $\boldsymbol{y}_i^{\dagger} \in R^m$ is the measurement vector of $\boldsymbol{x}_i^{\dagger}$.

We also illustrate the block diagram of encoding in Fig. 1 (a). Since the decoding procedure is the inverse operation of the encoding, we omit it for the purpose of avoiding duplication. The block diagram of decoding is shown in Fig. 1 (b).

Remark. The general advantage of BCS is reducing the computational complexity in both sampling and reconstruction process. The BCS deployed in transform domain like we described above retains this advantage for reconstruction, but will slightly increase the computational complexity of sampling process. However, it has been shown that the BCS deployed in transform domain can improve the sampling efficiency significantly [12]–[16]. Thus, it is deserved to deploy BCS in the transform domain in terms of the sampling efficiency.

3.3 A New Optimal Permutation Strategy

In Sect. 3.2, a novel permutation-based BCS strategy for 2D image signals is proposed. By selecting a pair of appropriate permutation matrices, the maximum block sparsity level of the permutated 2D signal can be reduced significantly, so better sampling efficiency can be obtained. However, for a given 2D signal $\boldsymbol{X}$ (For presentation simplicity, we abuse $\boldsymbol{X}$ and $\boldsymbol{X}_K$ in this paper whenever it will not lead to misunderstanding.), how to solve a pair of appropriate permutation matrices is not discussed yet. We will carefully study this problem in this section.

Let the column sparsity level vector of $\boldsymbol{X}$ and $\boldsymbol{X}^\dagger$ be $\boldsymbol{k} \in R^{\sqrt{N}}$ and $\boldsymbol{k}^\dagger \in R^{\sqrt{N}}$, respectively. Let the block sparsity level vector of $\boldsymbol{X}^\dagger$ be $\boldsymbol{K}^\dagger \in R^L$. The maximum block sparsity level and the average block sparsity level of $\boldsymbol{X}^\dagger$ can be denoted as $K^\dagger_{\max} = \|\boldsymbol{K}^\dagger\|_\infty$ and $\bar{K}^\dagger = \lceil \|\boldsymbol{K}^\dagger\|_1/L \rceil$, respectively.

In practice, it is desired that, after a permutation, the nonzero entries of the permuted 2D signal are evenly distributed among the blocks. It can be described by the following minimization problem

$$\begin{aligned} &\min \mu = \sum_{i=1}^{L} (K_i^\dagger - \bar{K}^\dagger)^2 \\ &\text{s.t. } \boldsymbol{X}^\dagger = \boldsymbol{P}_1 \boldsymbol{X} \boldsymbol{P}_2 \end{aligned} \tag{14}$$

We can calculate the value of μ over all possible $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to find a pair of optimal permutation matrices. Since there are $\sqrt{N}!$ permutation matrices of size $\sqrt{N}$, this method essentially require exhaustive searches, and a procedure which clearly is combinatorial in nature and has exponential complexity. This computational intractability led us to develop an alternative solving method. We will discuss the solving method in the following.

It is well known that when a permutation matrix $\boldsymbol{P}$ is multiplied with a matrix from the right (or left), it will permute the columns (or rows) of the matrix. Therefore, solving the optimal column (or row) permutation matrix is equivalent to find a column (or row) permutation which can make the nonzero entries of $\boldsymbol{X}^\dagger$ evenly distributed among the blocks. For easy of elaboration, let's take solving optimal column permutation matrix $\boldsymbol{P}_2$ as an example to illustrate in detail. The optimal row permutation matrix can be solved in the same way. Firstly, we initialize $\boldsymbol{X}^\dagger$ with a zero matrix with size of $\sqrt{N} \times \sqrt{N}$, and divide it into small blocks with block size of $\sqrt{N} \times \sqrt{n}$ (It is important to note that the block size isn't $\sqrt{n} \times \sqrt{n}$ here). Secondly, we allot the columns of $\boldsymbol{X}$ into the blocks of $\boldsymbol{X}^\dagger$. Since the number of columns of $\boldsymbol{X}$ is $\sqrt{N}$, we can allot the $\sqrt{N}$ columns of $\boldsymbol{X}$ into the blocks of $\boldsymbol{X}^\dagger$, each block with $\sqrt{n}$ columns in iteration manner. The total number of iterations is $\sqrt{n}$. At each iteration, we extract $\sqrt{L}$ columns from $\boldsymbol{X}$ successively and allot them to each block of $\boldsymbol{X}^\dagger$, each block with one column. As a rule of thumb, we allot the columns with few nonzero entries into the blocks with large sparsity level. The elaborated procedure can be described as follows.

Algorithm 1 Optimal Column Permutation Matrix Generating Algorithm (OCPMGA)

Input:

The column sparsity level of the 2D sparse signal $\boldsymbol{k} = \left[k_1, k_2, \ldots, k_{\sqrt{N}}\right]$, the number of entries in each block n and the number of blocks L.

Procedure:

(1) Initialize the column index set of the blocks $J_0^1 = J_0^2 = \cdots = J_0^{\sqrt{L}} = \varnothing$, the sparsity level accumulation of the blocks $S_0^1 = S_0^2 = \cdots = S_0^{\sqrt{L}} = 0$, and the iteration counter $T = 1$.

(2) Sort the elements of $\boldsymbol{k}$ in descending order. The sorted vector and its corresponding index vector are denoted by $\boldsymbol{b} = \left[b_1, b_2, \ldots, b_{\sqrt{N}}\right]$ and $\boldsymbol{q} = \left[q_1, q_2, \ldots, q_{\sqrt{N}}\right]$, respectively, here q_i is the original index of b_i in $\boldsymbol{k}$, i.e. $b_i = k_{q_i}$.

(3) Sort $\left\{S_T^1, S_T^2, \ldots, S_T^{\sqrt{L}}\right\}$ in ascending order, i.e. $S_T^{w_1} \leq S_T^{w_2} \leq \cdots \leq S_T^{w_{\sqrt{L}}}$, and record the index vector $\boldsymbol{w} = \left[w_1, w_2, \ldots, w_{\sqrt{L}}\right]$.

(4) For each block j, update the column index set and the sparsity level accumulation by $J_T^{w_j} = J_{T-1}^{w_j} \cup \left\{q_{(T-1)\sqrt{L}+j}\right\}$ and $S_T^{w_j} = S_{T-1}^{w_j} + b_{(T-1)\sqrt{L}+j}$, respectively.

(5) Increase T, and return to Step 3 if $T \leq \sqrt{n}$.

(6) Using the column index set of all the blocks to generate the column permutation $J = \left[J_T^1 \quad J_T^2 \quad \cdots \quad J_T^{\sqrt{L}}\right]$.

(7) Using J to generate an optimal column permutation matrix $\boldsymbol{P}_2^* = \boldsymbol{P}_J$.

Output:

Optimal column permutation matrix $\boldsymbol{P}_2^*$

3.4 An Example of the Proposed Strategy

Now, in order to better understand why the proposed optimal permutation strategy can reduce the maximum block sparsity level of 2D sparse signals, we consider the following example.

Example 2: Suppose that the block size is 4×4. Now we use the proposed optimal permutation strategy to rearrange the 2D signal $\boldsymbol{X}$ of Example 1.

Step 1 (Generating optimal column permutation matrix) Use Algorithm 1 to calculate optimal column permutation matrix. Obviously, we can easily obtain the column sparsity level vector of the 2D signal $\boldsymbol{k} = [7, 6, 5, 4, 3, 2, 1, 0]$. Sorting the entries of $\boldsymbol{k}$ in descending order, the sorted vector of $\boldsymbol{k}$ and the corresponding index vector are $\boldsymbol{b} = [7, 6, 5, 4, 3, 2, 1, 0]$ and $\boldsymbol{q} = [1, 2, 3, 4, 5, 6, 7, 8]$, respectively.

For the convenience of elaboration, we describe the iteration process of Algorithm 1 in Table 1. At each iteration, we allot two columns of the original 2D signal $\boldsymbol{X}$ to the blocks of $\boldsymbol{X}^\dagger$. In order to make the nonzero entries of $\boldsymbol{X}^\dagger$ evenly distributed among the blocks, we allot the columns with few nonzero elements into the blocks with large sparsity level. After four times of iteration, all columns of $\boldsymbol{X}$ are allocated into the blocks of $\boldsymbol{X}^\dagger$. The corresponding column permutation is J. Finally, we can use J to generate an optimal column permutation matrix $\boldsymbol{P}_2^* = [\boldsymbol{e}_1, \boldsymbol{e}_4, \boldsymbol{e}_5, \boldsymbol{e}_8, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_6, \boldsymbol{e}_7]$.

Step 2 (Generating optimal row permutation matrix) Using the similar method as Step 1 (we omit details for the purpose of avoiding duplication), we also can generate an optimal row permutation matrix $\boldsymbol{P}_1^* = [\boldsymbol{e}_1, \boldsymbol{e}_4, \boldsymbol{e}_5, \boldsymbol{e}_8, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_6, \boldsymbol{e}_7]^{\mathrm{T}}$.

Step 3 (optimal permutation) Process $\boldsymbol{X}$ with the following permutation operation

$$\boldsymbol{X}^\dagger = \boldsymbol{P}_1^* \boldsymbol{X} \boldsymbol{P}_2^* = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

Obviously, the block sparsity level vector of $\boldsymbol{X}^\dagger$ is $\boldsymbol{K} = [6, 8, 8, 6]$. The nonzero entries of $\boldsymbol{X}^\dagger$ are almost evenly distributed among the blocks. As a result, the maximum block sparsity level is reduced significantly by using our proposed permutation strategy.

3.5 Extra Data of the Proposed Strategy

By adding optimal permutation operation prior to sampling, the maximum block sparsity level of the permutated 2D signal can be reduced significantly, so better reconstruction performance can be obtained. However, we have to encode the permutation matrices in encoder side since we need to know permutation matrices to accomplish signal reconstruction, which brings some extra data overhead. Fortunately, a permutation matrix can be unique expressed by a permutation, so the amount of extra data is very small.

In our proposed strategy, it needs to additionally encode two permutations with length $\sqrt{N}$ while the former strategy proposed in [16] needs to additionally encode a permutation with length $\sqrt{NL}$. If the extra data is encoded into bit stream, $2\lceil \log_2 \sqrt{N} \rceil \cdot \sqrt{N}$ bits are needed for our proposed method while $\lceil \log_2 \sqrt{NL} \rceil \cdot \sqrt{NL}$ bits are needed for the former strategy proposed in [16]. Therefore, the proposed method can reduce the amount of extra data significantly compared with the former permutation strategy.

Table 1 Iteration computational process of OCPMGA

Block matrices		Initialization	Iteration Steps			
		$T=0$	$T=1$	$T=2$	$T=3$	$T=4$
$X_1^\dagger$	J_T^1	$\varnothing$	1	1, 4	1, 4, 5	1, 4, 5, 8
	S_T^1	0	7	11	14	14
$X_2^\dagger$	J_T^2	$\varnothing$	2	2, 3	2, 3, 6	2, 3, 6, 7
	S_T^2	0	6	11	13	14
J		$J = (1,4,5,8,2,3,6,7)$				
P^*		$P_2^* = P_J = [e_1, e_4, e_5, e_8, e_2, e_3, e_6, e_7]$				

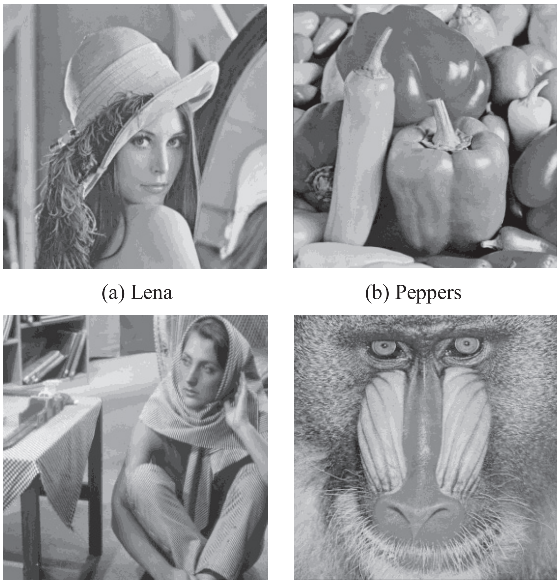
4. Simulation Results

In this section, we will report the overall, extensive experimental results to verify the compression performance of the proposed method. In our first experiment, we compare the permutation performance of the new optimal permutation strategy with that of other permutation strategies used in BCS, i.e. random permutation strategy described in [13]–[15] and the existing optimal permutation strategy described in our past work [16]. The wavelet coefficient matrix of Lena with dimension of 256 × 256 is used in the experiment. Here, the sparsifying basis used in the experiment is the well-known Daubechies 9/7 wavelet transform. The block size is selected to be 16 × 16. The maximum block sparsity levels of the coefficient matrix before and after permutation are shown in Table 2. Since the average block sparsity level is the low bound of the maximum block sparsity level, we also include the average block sparsity level in the table. The best K-term approximation is chosen according to different magnitude thresholds, i.e., keeping wavelet coefficients whose magnitudes are not less than the magnitude threshold and setting the remaining to be zeros. It can be seen from the table that the new optimal permutation has the suboptimal permutation performance among the three permutation strategies.

Permutation strategies can decrease the maximum block sparsity level significantly, which implies that better reconstruction performance can be achieved by adding permutation operation prior to sampling. In our second experiment, we evaluate the reconstruction performance of the proposed method and compare it with that of the other BCS methods. The experiments are tested on several gray-level natural images with size 512 × 512, such as, Lena, Peppers, Barbara, and Mandrill. These test images are shown in Fig. 2. We consider the following configurations: (1) traditional BCS [9]; (2) BCS with random permutation (BCS-RP); (3) BCS-OP; (4) BCS-NOP. The re-orthogonalizing Gaussian matrix is used for sensing matrix. Each CS sample is quantized to 8 bits for all of schemes. As discussed above, we need to reserve some space to encode the permutation information for Scheme (3) and Scheme (4). The extra data to encode the permutation information for different methods are tabulated in Table 3. As can be seen from the table, the proposed method can reduce the amount of extra data significantly compared with the former permutation

Table 2 Comparison of the maximum block sparsity levels for different permutation schemes

Magnitude Threshold		30	40	50
Average Block Sparsity Level		15	11	9
Maximum Block Sparsity Level	Random permutation	53	30	25
	Optimal permutation	**16**	**11**	**10**
	New optimal permutation	22	20	16



(a) Lena (b) Peppers (c) Barbara (d) Mandrill

Fig. 2 Four original test images used in the simulation experiment.

Table 3 Comparison of the extra bits for different permutation schemes

Permutation methods	Block size	
	16×16	32×32
Optimal permutation [16]	229, 376 bits	106, 496 bits
The proposed method	9, 216 bits	9, 216 bits

strategy.

The smoothed projected Landweber [9] reconstruction algorithm is used for image reconstruction in Scheme (1). In Scheme (2), Scheme (3) and Scheme (4), the reconstruction is implemented by using IHT algorithm in block-by-block manner, which is fast and effective. Peak signal-to-noise ratio (PSNR) is employed to evaluate the objective quality of the reconstructed images, which is defined as

$$\mathrm{PSNR} = 20\log_{10} R/\mathrm{RMSE} \tag{16}$$

with

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{\sqrt{N}}\sum_{i=1}^{\sqrt{N}}(X_{ij} - \hat{X}_{ij})^2} \tag{17}$$

where X_{ij} and $\hat{X}_{ij}$ denote the pixel values of the reconstructed image and the original image, respectively. R is the maximum value of the image gray level range.

The PSNR performance versus the compression ratio for block sizes with 16×16 and 32×32 are shown in Table 4 and Table 5, respectively. From the comparison result, we can draw some conclusions as follow. Firstly, all of three permutation-based BCS schemes can get better reconstructed-images quality than traditional BCS without permutation under the same compression ratio, which means that permutation-based BCS schemes can improve the sampling efficiency. Secondly, under the same compression ratio, our proposed algorithm provides a substantial gain of PSNR compared with BCS-RP algorithm, especially at the low sampling rates. Since the maximum block sparsity level after the proposed optimal permutation is smaller than that after random permutation, it shouldn't be surprising that Scheme (4) has the better reconstruction performance than Scheme (2). Finally, although the maximum block sparsity level after the proposed optimal permutation

Table 4 Comparison of PSNR (in dB) for different permutation-based BCS schemes with block size of 16×16

BCS algorithms	Compression Ratio				
	0.2	0.25	0.3	0.35	0.4
Lena					
traditional BCS	30.95	32.14	33.13	34.11	34.91
BCS-RP	31.01	32.35	33.58	35.24	35.87
BCS-OP	31.25	33.03	34.21	35.89	36.93
BCS-NOP	**31.88**	**34.29**	**35.99**	**37.73**	**38.65**
Peppers					
traditional BCS	31.65	32.67	33.51	34.27	34.88
BCS-RP	31.96	33.84	34.02	35.57	35.91
BCS-OP	32.15	34.02	34.86	35.99	36.31
BCS-NOP	**32.43**	**34.55**	**35.57**	**36.39**	**36.84**
Barbara					
traditional BCS	23.71	24.35	25.15	25.91	26.69
BCS-RP	24.12	25.58	26.72	29.84	31.22
BCS-OP	24.34	26.45	27.65	30.76	32.03
BCS-NOP	**24.98**	**27.12**	**28.59**	**31.33**	**33.00**
Mandrill					
traditional BCS	21.79	22.34	22.89	23.43	23.93
BCS-RP	21.98	22.51	23.14	24.13	24.78
BCS-OP	22.13	22.89	23.95	24.71	25.58
BCS-NOP	**22.35**	**23.17**	**24.20**	**25.06**	**26.01**

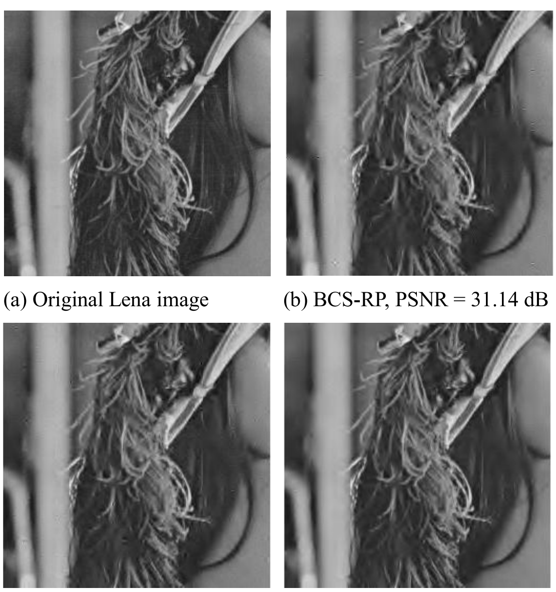
Table 5 Comparison of PSNR (in dB) for different permutation-based BCS schemes with block size of 32×32

BCS algorithms	Compression Ratio				
	0.2	0.25	0.3	0.35	0.4
Lena					
traditional BCS	30.75	31.91	32.91	33.81	34.66
BCS-RP	31.22	35.17	36.84	38.03	38.75
BCS-OP	33.42	35.46	37.01	38.08	39.02
BCS-NOP	**34.15**	**35.88**	**37.11**	**38.22**	**39.13**
Peppers					
traditional BCS	31.04	32.65	32.48	34.13	34.74
BCS-RP	33.51	35.02	35.96	36.75	37.21
BCS-OP	33.73	35.01	35.86	36.69	37.23
BCS-NOP	**33.91**	**35.11**	**36.03**	**36.78**	**37.42**
Barbara					
traditional BCS	23.53	24.20	24.82	25.51	26.21
BCS-RP	26.30	28.53	30.38	32.02	33.47
BCS-OP	26.84	28.75	30.51	32.06	33.52
BCS-NOP	**27.18**	**29.05**	**30.70**	**32.29**	**33.69**
Mandrill					
traditional BCS	21.55	22.10	22.63	23.16	23.68
BCS-RP	22.11	23.36	24.49	24.90	26.15
BCS-OP	23.03	23.62	24.56	25.34	27.19
BCS-NOP	**23.18**	**24.12**	**25.04**	**25.94**	**27.93**

(a) Original Lena image (b) BCS-RP, PSNR=31.14 dB

(c) BCS-OP, PSNR=33.11 dB (d) BCS-NOP, PSNR=34.26 dB

Fig. 3 Reconstructed Lena images under sampling rate $M/N = 0.2$.



(a) Original Lena image (b) BCS-RP, PSNR = 31.14 dB

(c) BCS-OP, PSNR=33.11 dB (d) BCS-NOP, PSNR=34.26 dB

Fig. 4 More details of the reconstructed images (Lena).

is larger than that after the former optimal permutation, it gets a substantial gain in terms of PSNR compared with Scheme (3). The reason is that the extra data of Scheme (3) is too large. When the extra data is taken into consideration, the improvement in sampling efficiency of this method is limited. The proposed method reduces the amount of extra data to encode the permutation information significantly, so it provides a substantial gain of PSNR compared with Scheme (3).

The visual quality comparison is also illustrated in Fig. 3 and Fig. 4, where the 512×512 gray-level Lena.bmp

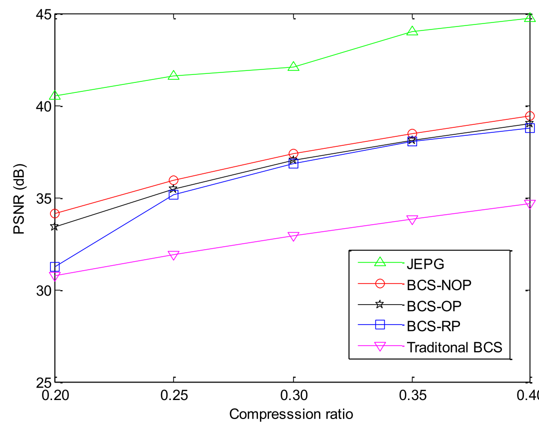


Fig. 5 Performance comparison of several compression methods.

is used for test. The block size is selected to be 32×32. Figure 3 displays some reconstructed images obtained by using the three evaluated permutation-based BCS methods under sampling rate $M/N = 0.2$. More Details of the reconstructed images can be found in Fig. 4. According to the simulation results, we can draw a conclusion that the visual quality of the reconstructed image by our method is best among all of the three methods, especially in edge details.

We also compare the proposed scheme with JPEG compression [21] and other CS-based image compression methods in Fig. 5. Lena with dimension of 512×512 is used in the test. For CS-based image compression methods, the block size is selected to be 32×32. From the figure, we can see that the proposed scheme is worse than JPEG compression in terms of the rate–distortion performance, but it narrows the gap between JEPG and the CS-based image compression methods. In fact, the CS-based image compression method is not a competitive technique compared with JPEG standard in view of compression efficiency. Nonetheless, the CS-based image compression has become a hot topic in recent years because of two reasons. Firstly, CS is a good candidate for robust image coding. A representative work was presented by Deng et al. in [5], in which the compressive measurements can be viewed as a number of descriptions mainly because of their democracy properties. We recommend the readers to see [5] for more details. Secondly, CS-based image coding is very promising method in encrypted image compression applications. It has been suggested in [6] that CS framework leads to an encryption scheme, where the sensing matrix can be considered as an encryption key. Therefore, the CS-based image compression methods like proposed in this paper is useful in contexts where robust coding [5] and encrypted compression [6] are very important, but the subsequent storage and communication of quantized samples is less constricted.

5. Conclusions

We present an improved optimal permutation based BCS called BCS-NOP in this paper. As a result, the proposed method can significantly reduce the extra data amount to encode the permutation information and thereby improves the code efficiency compared with the method in [16], as demonstrated by simulation tests.

Acknowledgments

This research is funded by the National Natural Science Foundation of China (No. 61272043), the Program for New Century Excellent Talents in University of China (No. NCET-11-0873), and the Key Project of Chongqing Natural Science Foundation (No. cstc2013jjB40009).

References

[1] D.L. Donoho, "Compressed sensing," IEEE Trans. Inf. Theory, vol.52, no.4, pp.1289–1306, April 2006.
[2] E.J. Candes and T. Tao, "Decoding by Linear Programming," IEEE Trans. Inf. Theory, vol.51, no.12, pp.4203–4215, Dec. 2005.
[3] E.J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," IEEE Trans. Inf. Theory, vol.52, no.2, pp.489–509, Feb. 2006.
[4] V.K. Goyal, A.K. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," IEEE Trans. on Signal Processing Magazine, vol.25, no.2, pp.48–56, Feb. 2008.
[5] C. Deng, W. Lin, B. Lee, and C.T. Lau, "Robust image coding based upon compressive sensing," IEEE Trans. Multimedia, vol.14, no.2, pp.278–290, Feb. 2012.
[6] A. Orsdemir, H.O. Altun, G. Sharma, and M.F. Bocko, "On the security and robustness of encryption via compressed sensing," Proc. of IEEE military communication Conference (MILCOM), San Diego, CA, pp.1–7, Nov. 2008.
[7] F.D. Marco, A.D. Mark, T. Dharmpal, N.L. Jason, S. Ting, F.K. Kevin, and G.B. Richard, "Single-pixel imaging via compressive sampling," IEEE Trans. on Signal Processing Magazine, vol.25, no.2, pp.83–91, April 2008.
[8] L. Gan, "Block compressed sensing of natural images," Proc. of 2007 15th International Conference on Digital Signal Processing, Cardiff, UK, pp.403–406, July 2007.
[9] S. Mun and J.E. Fowler, "Block compressed sensing of images using directional transforms," Proc. of International Conference Image Processing, Cario, Egypt, pp.3021–3024, Nov. 2009.
[10] L. Gan, T.T. Do, and T.D. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," Proc. of the European Signal Processing Conference, Lausanne, Switzerland, pp.1–5, Aug. 2008.
[11] Y. Yang, O.C. Au, L. Fang, X. Wen, and W. Tang, "Reweighted compressive sampling for image compression," Proc. International Conference Picture Coding Symposium (PCS), pp.89–92, May 2009.
[12] J.E. Flower, S. Mun, and E.W. Tramel, "Multiscale block compressed sensing with smoother projected Landweber reconstruction," Proc. of the 19th European Signal Processing Conference, Barcelona, Spain, pp.564–568, Aug. 2011.
[13] Z. Gao, C. Xiong, C. Zhou, and H. Wang, "Compressive sampling with coefficients random permutations for image compression," Proc. of International Conference on Multimedia and Signal Processing, Guilin, China, pp.321–324, May 2011.
[14] H. Fang, S.A. Vorobyov, and H. Jiang, "Permutation enhanced parallel reconstruction for compressive sampling," Proc. of International Conference on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico, pp.393–396, Dec. 2015.
[15] Z. Gao, C. Xiong, L. Ding, and C. Zhou, "Image representation using block compressive sensing for compression applications," Journal of Visual Communication and Image Representation, vol.24, no.7, pp.885–894, July 2013.
[16] B. Zhang, Y. Liu, J. Zhuang, and L. Yang, "A novel block compressed sensing based on matrix permutation," Proc. of the 30th International Conference on Visual Communications and Image Processing, Chengdu, China, pp.1–4, Nov. 2016.
[17] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," SIAM review, vol.43, no.1, pp.129–159, Jan. 2001.
[18] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory, vol.53, no.12, pp.4655–4666, Dec. 2007.
[19] G. Chen, D. Li, and J. Zhang, "Iterative gradient projection algorithm for two-dimensional compressive sensing sparse image reconstruction," Signal Processing, vol.104, pp.15–26, Nov. 2014.
[20] J.D. Blanchard, J. Tanner, and K. Wei, "Conjugate gradient iterative hard thresholding: observed noise stability for compressed sensing," IEEE Trans. Signal Process., vol.63, no.2, pp.528–537, Feb. 2015.
[21] W.B. Pennbaker and J.L. Mitchell, JPEG Still Image Data Compression Standard, Springer-Verlag, New York, 1993.

Yuqiang Cao received the B.E. and the M.E. degree in Electronics and Information Engineering, Signal and Information Processing from Chongqing Communication Institute, Chongqing, China, in 2001 and 2004, respectively. Currently, he is pursuing his doctoral degree at Chongqing University, Chongqing, China. His research interests include image processing, digital watermarking and information hiding.

Weiguo Gong received Ph.D. degrees in Computer Engineering from the Tokyo Institute of Technology, Japan, in 1996. He is a Professor in the Key Laboratory of Optoelectronic Technology and System of Ministry of Education, Chongqing University, Chong, China. His research interests include image processing, pattern recognition and computer vision.

Bo Zhang received the B.E. degree in Electronics and Information Engineering from Southwest Jiaotong University, Chengdu, China, in 2010 and the M.E. degree in signal processing from Chongqing Communication Institute, Chongqing, China, in 2013. He is currently an assistant of Chongqing Communication Institute. His research interests include signal processing and communication. Currently, he has a particular interest in compressed sensing, image processing and wireless sensor network.

Fanxin Zeng received the B.E. degree in mathematics from Chongqing Normal University, China, in 1985, and the Ph.D. degree in electronic circuit and system from the Chongqing University, China, in 2014. He is a Professor in the Department of Information Engineering, Chongqing Communication Institute. His research interests include channel coding, sequential design and spread spectrum communication.

Sen Bai received the B.S. degree in mathematics from Sichuan University, China, in 1985, and the M.S. and Ph.D. degrees in applied mathematics, control theory and control engineering from the Chongqing University, China, in 1998 and 2002, respectively. He is a Professor in the Department of Information Engineering, Chongqing Communication Institute. His research interests include image processing, digital watermarking and information hiding.