Permutation Meets Parallel Compressed Sensing: How to Relax Restricted Isometry Property for 2D Sparse Signals

Hao Fang, *Student Member, IEEE*, Sergiy A. Vorobyov, *Senior Member, IEEE*, Hai Jiang, *Member, IEEE* and Omid Taheri, *Student Member, IEEE*

***Abstract*—Traditional compressed sensing considers sampling a 1D signal. For a multidimensional signal, if reshaped into a vector, the required size of the sensing matrix becomes dramatically large, which increases the storage and computational complexity significantly. To solve this problem, the multidimensional signal is reshaped into a 2D signal, which is then sampled and reconstructed column by column using the same sensing matrix. This approach is referred to as *parallel compressed sensing*, and it has much lower storage and computational complexity. For a given reconstruction performance of parallel compressed sensing, if a so-called *acceptable permutation* is applied to the 2D signal, the corresponding sensing matrix is shown to have a smaller required order of restricted isometry property condition, and thus, lower storage and computation complexity at the decoder are required. A zigzag-scan-based permutation is shown to be particularly useful for signals satisfying the newly introduced layer model. As an application of the parallel compressed sensing with the zigzag-scan-based permutation, a video compression scheme is presented. It is shown that the zigzag-scan-based permutation increases the peak signal-to-noise ratio of reconstructed images and video frames.**

***Index Terms*—Compressed sensing, parallel processing, permutation, multidimensional signal processing.**

I. Introduction

Compressed sensing (CS) theory states that the information contained in an L-length sparse signal $\mathbf{x}$ can be fully preserved with only $K \ll L$ measurements, which form a K-length vector $\mathbf{y}$ [1], [2]. This is done by the help of a $K \times L$ *sensing matrix* $\mathbf{A}$, i.e., $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ satisfies the restricted isometry property (RIP) of a certain order. The signal $\mathbf{x}$ can be recovered from the K measurements in $\mathbf{y}$ by solving, for example, the following ℓ_1-norm minimization problem [3]

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{1}$$

where $||\cdot||_1$ denotes the ℓ_1-norm of a vector. In addition, if signal $\mathbf{f}$ is not sparse itself, it may be represented as a sparse signal in some orthonormal basis $\mathbf{\Psi}$, i.e., $\mathbf{x} = \mathbf{\Psi}^T\mathbf{f}$ is a sparse signal. Here $(\cdot)^T$ denotes the transpose operation. Then given the *sensing matrix* $\mathbf{A}$ for $\mathbf{x}$ and the orthonormal basis $\mathbf{\Psi}$, the signal $\mathbf{f}$ can be measured using a $K \times L$ *measurement matrix* $\mathbf{\Phi} = \mathbf{A}\mathbf{\Psi}^T$, i.e., $\mathbf{y} = \mathbf{\Phi}\mathbf{f}$. It is equivalent to using $\mathbf{A}$ to sense $\mathbf{x}$ since $\mathbf{y} = \mathbf{A}\mathbf{\Psi}^T\mathbf{f} = \mathbf{A}\mathbf{x}$. Therefore, $\mathbf{x}$ and thus $\mathbf{f}$ can be recovered from $\mathbf{y}$ as long as $\mathbf{A}$ satisfies the RIP of a certain order.

Usually, CS is applied to 1D signals, or vector-reshaped multidimensional signals. When the length of the 1D signal or the vector-reshaped multidimensional signal is very large, a large size measurement matrix $\mathbf{\Phi}$, or equivalently, a large size sensing matrix $\mathbf{A}$, is needed. A dense measurement matrix $\mathbf{\Phi}$ or a dense sensing matrix $\mathbf{A}$ results in a high storage requirement and computational complexity in both the sampling and reconstruction processes. An example of such sampling and reconstruction scheme with a dense measurement matrix is the single-pixel camera proposed in [4]. The single-pixel camera acquires a group of measurements of a 2D image using different patterns of the digital micromirror device (DMD) array without collecting the pixels. Each pattern of the DMD array plays the role of a row in the measurement matrix $\mathbf{\Phi}$, and gives one measurement for the vector-reshaped 2D image. If the length of the vector-reshaped 2D image is very large, the encoder needs to store a large group of patterns, and the computational complexity of the reconstruction process is high.

To address the above problem related to the storage requirement and the computational complexity during the sampling process, a separable sensing operator framework is designed for compressive imaging in [5], where an imaging operator, that is, the measurement matrix for the whole image, can be separated into two dimensions, i.e., two smaller-sized measurement matrices. Using this approach, all dimensions of the image signal can be sampled sequentially using corresponding measurement matrix of a smaller size. Then the encoder needs to store only the smaller-sized measurement matrices. The separable sensing operator design significantly reduces

Manuscript received November 24, 2012; revised April 10, 2013 and August 23, 2013; accepted September 20, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiang-Gen Xia. This work is supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada. Some initial results based on which this paper is built have been presented by the authors at Asilomar 2012, Pacific Grove, California, USA.

H. Fang was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada, and is currently with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA; e-mail: hfang@uw.edu. S. A. Vorobyov is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (on leave) and the Department of Signal Processing and Acoustics, Aalto University, FI-00076 AALTO, Finland; e-mail: svor@ieee.org. H. Jiang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada; e-mail: hai1@ualberta.ca. O. Taheri was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada; email: otaheri@ualberta.ca.

the complexity of implementation, storage, and usage of the imaging operator. A general framework of the separable sensing operator for sampling multidimensional signals, named *Kronecker CS*, is proposed in [6]. In Kronecker CS, it is shown that the following two methods are equivalent: (i) sampling the vectorized multidimensional signal using a measurement matrix which is the Kronecker product of several smaller-sized measurement matrices that correspond to the measurement processes for different portions of the multidimensional signal; (ii) sampling the multidimensional signal sequentially using corresponding smaller-sized measurement matrices. Thus, it is enough to store only the smaller-sized measurement matrices instead of the Kronecker product of the smaller-sized measurement matrices.

Another solution to the aforementioned problem related to the sampling process is the block CS of [7]. The idea of the block CS is to divide a 2D signal into smaller blocks and sample individual vector-reshaped blocks, whereas all blocks need to be reconstructed as a whole. Essentially, a block-diagonal measurement matrix, instead of a dense measurement matrix, is used in the block CS of [7] to sample the vector-reshaped signal. As a result, the block CS can reduce the storage and computational complexity at the encoder side. Some improved reconstruction algorithms for the block CS scheme are presented in [8]. They help to further reduce the required number of rows in the measurement matrix $\mathbf{\Phi}$ for a given reconstruction error requirement. Based on the block CS architecture, a fast sampling operator is proposed in [9] using the block Hadamard ensemble, which can be easily implemented in the optical domain. Note that all the above works focus on the encoder side, aiming at reducing the storage and computational complexity of the sampling process. Joint reconstruction is employed in these schemes, and thus, the computational complexity at the decoder is still high.

In this paper, to achieve a low-complexity at both encoder and decoder sides, a parallel CS scheme is considered for the multidimensional sparse signal. The multidimensional signal is first rearranged into a 2D matrix, and then sampled column-by-column via CS using the same sensing matrix. In this way, the required size of the sensing matrix can be reduced significantly compared to the scheme that samples the vectorized signal. Furthermore, both sampling and reconstruction can be conducted for individual columns in parallel. Note that when a 2D signal sparse in the identity basis is considered, the measurement matrix is the same as the sensing matrix; and then the column-by-column sampling scheme is actually a special case of the sampling operator used in the Kronecker CS framework of [6] and the block CS framework of [7] and [8]. In the Kronecker CS framework, if one of the two sampling operators for the 2D signal is the identity matrix, then the 2D signal is sampled either row-by-row or column-by-column. In the block CS framework, if each column of the 2D signal is a block, then the block-by-block sampling is also the same as the column-by-column sampling. Several other works also use a similar column-by-column sampling setting at the encoder side for the signal which is sparse in the identity basis. In [4], the wavelet coefficients of a 2D image are rearranged in a specific order into a 2D matrix and sampled column-by-column via CS. If the same sampling operator is applied to each row, the row-by-row scan in [10] is similar to the column-by-column sampling scheme in the parallel CS framework. Another example is the multiple measurement vectors (MMV) model of [11], which considers a group of 1D signals that share the same sparsity profile. In the MMV model, a group of 1D signals is considered and all signals are sampled using the same dictionary, which is analogous to the sensing matrix in CS, while the group of 1D signals can be viewed as a virtual 2D signal.

It is known that the column-by-column reconstruction scheme in the parallel CS framework provides a poor reconstruction error performance, although it has low computational complexity [6]. Thus, different reconstruction schemes are proposed to provide a better reconstruction error performance [4], [6]–[8], [10], [11]. All these approaches employ a joint reconstruction, which has higher computational complexity compared to the basic column-by-column reconstruction scheme.

Another problem of the parallel CS scheme is that the sparsity levels[1] of the columns of the 2D-reshaped signal differ from one column to another. This problem has been partially studied in [12] and [13]. If we regard each column as a vectorized block, then the column-by-column sampling and reconstruction setting is similar to the block-based CS architecture proposed for video compression in [12] and [13]. Considering that some blocks are not sparse enough to apply CS, it is proposed in [12] and [13] to apply CS only to sparse blocks found by a block classification scheme. Another block classification scheme based on inter-frame correlation is proposed in [14].

In the parallel CS framework considered in this paper, we propose to permute the 2D signal before it is sampled. Using a so-called acceptable permutation, the maximal sparsity level of the columns of the permuted 2D signal can be reduced, and thus a sensing matrix with a weaker RIP condition can be used to sample all columns. Accordingly, the error performance of the reconstruction can be improved for a given number of measurements for the 2D signal. The introduction of permutation provides an alternative solution to those in [4], [6]–[8], [10], [11] in order to give a better reconstruction error performance compared to the basic column-by-column reconstruction scheme, while not increasing the computational complexity.

Permutations are studied in several papers related to CS, though the goals of permutations in the existing literature are very different from our goal here. In [15], a segmented CS architecture is proposed and it is shown that a similar improvement to that obtained by increasing the size of the measurement matrix can be achieved by using a virtual extended measurement matrix obtained by permuting the existing rows of the initial measurement matrix. In [16], it is shown that if nonzero entries of a sparse signal are clustered, the deterministic Delsarte-Goethals frame used as a sensing matrix does not work. Thus, it is proposed to apply permutations to the signal in order to avoid clustered nonzero entries. In

[1]Note that the sparsity level of a column is the number of nonzero entries in the column.

our paper, the goal for applying permutations is different. Specifically, the parallel CS framework considers sensing matrices that satisfy the RIP, and permutation is applied to the 2D-reshaped signal aiming at ensuring that the maximum sparsity level of the columns is reduced. We show that if a so-called acceptable permutation is conducted before sampling, the sensing matrix needs to satisfy the RIP of a smaller order than the sensing matrix of the parallel CS without any permutation.

In our paper, a group-scan-based permutation is introduced as an acceptable permutation for 2D signals which can be divided into a number of groups with entries in each group having the same probability to be large in magnitude. As a special case of such group-scan-based permutation, a zigzag-scan-based permutation is introduced and investigated for 2D signals satisfying a newly introduced layer model. A video compression scheme based on the parallel CS with the zigzag-scan-based permutation is also developed and investigated. It improves the peak signal-to-noise ratio (PSNR) of reconstructed frames compared to the parallel CS without permutation. This demonstrates the effectiveness of the zigzag-scan-based permutation in image compression.

In summary, this paper makes three contributions. First, we investigate properties of permutations when applied to parallel CS. Second, we introduce a group-scan-based permutation and as an example a zigzag-scan-based permutation, and we show that the zigzag-scan-based permutation is an acceptable permutation with a large probability for 2D signals satisfying a newly introduced layer model. Finally, the application of our parallel CS with the zigzag-scan-based permutation to video compression in wireless multimedia sensor networks is discussed. Some preliminary results have been reported in [17].

The remainder of the paper is organized as follows. Section II introduces the parallel CS scheme. Permutations are discussed in Section III. Section IV describes the video compression scheme that employs the parallel CS with the zigzag-scan-based permutation in application to wireless multimedia sensor networks. Simulation results are given in Section V. Finally, conclusions and further discussion are given in Section VI. This paper is reproducible research, and the software needed to generate the simulation results can be obtained from the IEEE Xplore together with the paper.

II. Parallel CS

Given any multidimensional sparse signal, we can rearrange it into a 2D matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$. A multidimensional signal and the corresponding 2D matrix $\mathbf{X}$ are called s-sparse or have sparsity level s if $\mathbf{X}$ has only s nonzero entries. The sparsity level of $\mathbf{X}$ can be denoted as a sparsity vector $\mathbf{s} = [s_1, s_2, \cdots, s_N]$, where s_j is the sparsity level of the j-th column of $\mathbf{X}$. In other words, the j-th column of $\mathbf{X}$ has only s_j nonzero entries. Apparently, $||\mathbf{s}||_1 = s$.

In terms of the 2D signal $\mathbf{X}$, the parallel CS scheme consists of sampling each column of $\mathbf{X}$ by the same sensing matrix $\mathbf{A}$ and reconstructing these columns individually and in parallel by using any 1D CS reconstruction algorithm. In this paper, for presentation simplicity, we consider 2D signals, i.e., rearrangement of a multidimensional signal into a 2D matrix is done in advance.

A. Theoretical Results on CS for 1D Signals

Most practical signals are not strictly sparse, but rather regarded as *compressible*, i.e., they have only a few large[2] entries. For a 1D or 2D signal, we define its *best s-term approximation* as a signal generated by keeping the s largest entries in the original signal and changing the remaining entries to zeros. Therefore, the best s-term approximation has sparsity level s. It is an optimal approximation using only s entries to approximate a compressible signal. However, such approximation requires knowledge about the values and locations of all entries in $\mathbf{x}$.

On the other hand, when CS is applied to $\mathbf{x}$, it is known that, if the sensing matrix $\mathbf{A}$ obeys the RIP of order s, the reconstruction via solving (1) is nearly as good as that using the best s-term approximation, as shown in the following Lemma 1 [3], [18]. Before Lemma 1, consider the following definition.

Definition 1: *[3] For every integer $s = 1, 2, \ldots,$ the s-restricted isometry constant δ_s of a given matrix $\mathbf{A}$ is defined as the smallest quantity such that the inequality*

$$(1 - \delta_s)||\mathbf{z}||_2^2 \leq ||\mathbf{A}\mathbf{z}||_2^2 \leq (1 + \delta_s)||\mathbf{z}||_2^2$$

holds for all sparse signals $\mathbf{z}$ with no more than s nonzero entries, where $||\cdot||_2$ denotes the ℓ_2-norm of a vector.

Lemma 1: *[18] Assume that $\delta_{2s} < \sqrt{2} - 1$ (s is a positive integer) for a sensing matrix $\mathbf{A}$. Then for a signal $\mathbf{x}$, the solution $\mathbf{x}^*$ to (1) obeys*

$$||\mathbf{x}^* - \mathbf{x}||_1 \leq G \cdot ||\mathbf{x} - \mathbf{x}^s||_1$$

and

$$||\mathbf{x}^* - \mathbf{x}||_2 \leq G \cdot ||\mathbf{x} - \mathbf{x}^s||_1 / \sqrt{s}$$

for some constant G.

In this paper, if the $2s$-restricted isometry constant of $\mathbf{A}$ satisfies $\delta_{2s} < \sqrt{2} - 1$, the matrix $\mathbf{A}$ is regarded as obeying the RIP of order s. Therefore, according to Lemma 1, an s-sparse signal can be exactly reconstructed via solving (1) if the sensing matrix $\mathbf{A}$ obeys the RIP of order s. For a compressible signal $\mathbf{x}$, if a sensing matrix $\mathbf{A}$, which obeys the RIP of order s, is used to sample $\mathbf{x}$, the reconstruction via solving (1) has an error bounded by the ℓ_1-norm of the approximation error when $\mathbf{x}^s$ is used to approximate $\mathbf{x}$. Note that, for reconstruction via solving (1), we do not need knowledge about the values and locations of all entries in $\mathbf{x}$, while such knowledge is needed if $\mathbf{x}^s$ is used to approximate $\mathbf{x}$.

B. New Theoretical Results on Parallel CS for 2D Signals

Based on Lemma 1, the following lemma gives a sufficient condition for exact reconstruction of a 2D s-sparse signal using parallel CS.

[2]In this paper, when we say a value is large or small, it means the magnitude of the value is large or small.

Lemma 2: *Consider a 2D s-sparse signal $\mathbf{X}$ with sparsity vector $\mathbf{s}$, if the RIP of order $||\mathbf{s}||_\infty$ holds for the sensing matrix $\mathbf{A}$, i.e., $\delta_{2||\mathbf{s}||_\infty} < \sqrt{2}-1$, where $||\cdot||_\infty$ stands for the Chebyshev norm of a vector*[3]*, then $\mathbf{X}$ can be exactly reconstructed using parallel CS scheme.*

Proof: The proof follows the same steps as the proof for the following Lemma 3. ■

For a 2D compressible signal $\mathbf{X}$, the following lemma gives a sufficient condition that the reconstruction error of the parallel CS is bounded by the ℓ_1-norm of the approximation error when the best s-term approximation of $\mathbf{X}$, denoted as $\mathbf{X}^s$, is used to approximate $\mathbf{X}$.

Lemma 3: *Let $\mathbf{X}^s \in \mathbb{R}^{M\times N}$, which has a sparsity vector $\mathbf{s} = [s_1, s_2, \cdots, s_N]$, be the best s-term approximation of $\mathbf{X} \in \mathbb{R}^{M\times N}$. If $s_j \geq 1$ for $j = 1, 2, \cdots, N$, and the sensing matrix $\mathbf{A}$ obeys the RIP of order $||\mathbf{s}||_\infty$, i.e., $\delta_{2||\mathbf{s}||_\infty} < \sqrt{2}-1$, then the signal $\hat{\mathbf{X}}$ reconstructed using parallel CS scheme obeys*

$$||\hat{\mathbf{X}} - \mathbf{X}||_1 \leq G \cdot ||\mathbf{X} - \mathbf{X}^s||_1$$

and

$$||\hat{\mathbf{X}} - \mathbf{X}||_2 \leq G \cdot ||\mathbf{X} - \mathbf{X}^s||_1$$

where G is a finite constant. Here for a matrix, $||\cdot||_1$ and $||\cdot||_2$ denote ℓ_1-norm and ℓ_2-norm of the vectorized matrix, respectively.

Proof: Due to the fact that for all $j \in \{1, 2, \cdots, N\}$, $||\mathbf{s}||_\infty \geq s_j$, and according to the definition of s-restricted isometry constant, we have $\delta_{2s_j} \leq \delta_{2||\mathbf{s}||_\infty} < \sqrt{2}-1$. Then, according to Lemma (1), we obtain that

$$||\hat{\mathbf{x}}_j - \mathbf{x}_j||_1 \leq G_j \cdot ||\mathbf{x}_j - \mathbf{x}_j^s||_1$$

and

$$||\hat{\mathbf{x}}_j - \mathbf{x}_j||_2 \leq G_j \cdot ||\mathbf{x}_j - \mathbf{x}_j^s||_1 \cdot s_j^{-1/2} \leq G_j \cdot ||\mathbf{x}_j - \mathbf{x}_j^s||_1$$

where $\mathbf{x}_j$, $\hat{\mathbf{x}}_j$, and $\mathbf{x}_j^s$ denote the j-th column of $\mathbf{X}$, $\hat{\mathbf{X}}$, and $\mathbf{X}^s$, respectively, and G_j is a finite constant. Therefore, by choosing $G = \max_j\{G_j\}$, we obtain that

$$\begin{aligned}||\hat{\mathbf{X}} - \mathbf{X}||_1 &= \sum_{j=1}^{N} ||\hat{\mathbf{x}}_j - \mathbf{x}_j||_1 \\ &\leq G \cdot \sum_{j=1}^{N} ||\mathbf{x}_j - \mathbf{x}_j^s||_1 = G \cdot ||\mathbf{X} - \mathbf{X}^s||_1\end{aligned}$$

and

$$\begin{aligned}||\hat{\mathbf{X}} - \mathbf{X}||_2 &= \sqrt{\sum_{j=1}^{N} ||\hat{\mathbf{x}}_j - \mathbf{x}_j||_2^2} \\ &\leq \sqrt{G^2 \cdot \sum_{j=1}^{N} ||\mathbf{x}_j - \mathbf{x}_j^s||_1^2} = G \cdot \sqrt{\sum_{j=1}^{N} ||\mathbf{x}_j - \mathbf{x}_j^s||_1^2} \\ &\leq G \cdot \sum_{j=1}^{N} ||\mathbf{x}_j - \mathbf{x}_j^s||_1 = G \cdot ||\mathbf{X} - \mathbf{X}^s||_1.\end{aligned}$$

This completes the proof. ■

To sum up, for parallel CS, the RIP condition for the sensing matrix $\mathbf{A}$ for a given reconstruction quality is related to $||\mathbf{s}||_\infty$. In Subsection III-A, it will be shown that the RIP condition can be relaxed by performing a so-called acceptable permutation before using the parallel CS.

III. Permutations

When parallel CS is applied to a 2D compressible signal[4] $\mathbf{X}$, the difference of sparsity levels among columns of $\mathbf{X}^s$ (which has sparsity vector $\mathbf{s}$) is not considered, and thus, the 'worst-case' sparsity level of the columns of $\mathbf{X}^s$, i.e., $||\mathbf{s}||_\infty$, needs to be taken into account when designing the sensing matrix. In this section, permutation is introduced such that by permuting[5] entries of $\mathbf{X}$, the maximum sparsity level of the columns of the best s-term approximation of the newly formed 2D signal is reduced.

Let $\mathrm{P}(\cdot)$ be a permutation operator which maps a matrix into another matrix by permuting its entries and $\mathrm{P}^{-1}(\cdot)$ be the corresponding inverse permutation operator. Then $\mathbf{X}^\dagger = \mathrm{P}(\mathbf{X})$ and $\mathbf{X} = \mathrm{P}^{-1}(\mathbf{X}^\dagger)$ where $\mathbf{X}^\dagger \in \mathbb{R}^{M\times N}$ is a permuted 2D signal.

With permutation before sampling, the parallel sampling process can be described as follows

$$\mathbf{y}_j = \mathbf{A}\mathbf{x}_j^\dagger \tag{2}$$

where $\mathbf{x}_j^\dagger$ is the j-th column of $\mathbf{X}^\dagger$, and $\mathbf{y}_j$ is the measurement vector of $\mathbf{x}_j^\dagger$. We can rewrite (2) in the matrix form as

$$\mathbf{Y} = \mathbf{A}\mathbf{X}^\dagger = \mathbf{A}\mathrm{P}(\mathbf{X})$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]$.

For signal reconstruction, all columns of $\mathbf{X}^\dagger$ can be reconstructed in parallel by any existing CS reconstruction algorithm. Let $\hat{\mathbf{X}}^\dagger$ be the reconstructed permuted signal. Then we can apply inverse permutation to $\hat{\mathbf{X}}^\dagger$ to obtain the reconstructed 2D signal $\hat{\mathbf{X}}$, that is, $\hat{\mathbf{X}} = \mathrm{P}^{-1}(\hat{\mathbf{X}}^\dagger)$.

A. Discussion about Permutation

For any multidimensional signal, the permutation can be either applied after or included in the process of rearranging the multidimensional signal into a 2D matrix. The block-based CS employed in [7], [8], and [13] is a special case of the parallel CS, which can be interpreted as making each vectorized block as a column of a new 2D signal. Furthermore, the problem of difference of sparsity levels among blocks is addressed in [13] by employing a classification scheme to identify sparse blocks and dense blocks and then applying CS only to the sparse blocks. In our work, permutation is applied to a 2D compressible signal $\mathbf{X}$ or integrated into the process of rearrangement of a multidimensional signal to a 2D compressible signal $\mathbf{X}$ such that all columns of $\mathbf{X}^{\dagger s}$ (the

[3]The Chebyshev norm of a vector is equal to the largest magnitude of the entries in the vector.

[4]Without loss of generality, compressible signals are considered in the remainder of the paper, since a sparse signal can be regarded as a special case of a compressible signal.

[5]In this paper, "permute" means changing positions of entries in a 2D matrix, while not changing the dimension of the matrix.

best s-term approximation of the resulted 2D signal $\mathbf{X}^\dagger$) are sparse. Thus, the classification step of [13] is avoided.

Consider a compressible 2D signal $\mathbf{X}$ and its best s-term approximation $\mathbf{X}^s$ with sparsity vector $\mathbf{s}$, i.e., $||\mathbf{s}||_1 = s$. If the sensing matrix $\mathbf{A} \in \mathbb{R}^{K\times M}$ is constructed from Gaussian ensembles with

$$K \geq C \cdot ||\mathbf{s}||_\infty \log\left(M/||\mathbf{s}||_\infty\right) \tag{3}$$

for some constant C, then it will satisfy the RIP of order $||\mathbf{s}||_\infty$ [3]. Then according to Lemma 3, the signal $\hat{\mathbf{X}}$ reconstructed using parallel CS obeys the inequalities shown in Lemma (3).

Definition 2: *For a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ and its best s-term approximation $\mathbf{X}^s$ with sparsity vector $\mathbf{s}$, a permutation $P(\cdot)$ is called acceptable for $\mathbf{X}$ if the Chebyshev norm of the sparsity vector of the best s-term approximation of the signal $P(\mathbf{X})$ is smaller than $||\mathbf{s}||_\infty$.*

When permutation is applied before parallel CS, the signal after permutation is $\mathbf{X}^\dagger$, and the best s-term approximation of $\mathbf{X}^\dagger$ is denoted as $\mathbf{X}^{\dagger s}$ with sparsity vector $\mathbf{s}^\dagger$, i.e., we have $||\mathbf{s}^\dagger||_1 = s$. Consider that $M \gg ||\mathbf{s}||_\infty$ and $M \gg ||\mathbf{s}^\dagger||_\infty$, i.e., $\mathbf{X}^s$ and $\mathbf{X}^{\dagger s}$ are sparse enough. If $||\mathbf{s}^\dagger||_\infty < ||\mathbf{s}||_\infty$, it can be seen that for parallel CS with an acceptable permutation, the lower bound of K in (3) is smaller than that in the parallel CS without an acceptable permutation. In other words, for the sufficient condition in Lemmas 2 and 3, the condition "$\mathbf{A}$ obeys the RIP of order $||\mathbf{s}^\dagger||_\infty$" for parallel CS with an acceptable permutation is weaker than the condition "$\mathbf{A}$ obeys the RIP of order $||\mathbf{s}||_\infty$" for parallel CS without an acceptable permutation. To sum up, the RIP condition for a given reconstruction quality is weaker after permutation if $||\mathbf{s}^\dagger||_\infty$ is smaller than $||\mathbf{s}||_\infty$.

Since $||\mathbf{s}^\dagger||_1 = ||\mathbf{s}||_1 = s$, it is desired that, after a permutation, the s nonzero entries in the best s-term approximation of the permuted 2D signal are evenly distributed among the columns, which leads to minimum $||\mathbf{s}^\dagger||_\infty$. A permutation providing such even distribution is an optimal permutation defined below.

Definition 3: *For a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ and its best s-term approximation $\mathbf{X}^s$, if after a permutation, the best s-term approximation $\mathbf{X}^{\dagger s}$ of the resulted 2D signal $\mathbf{X}^\dagger$ has sparsity vector $\mathbf{s}^*$ satisfying $\max_i\{s_i^*\} - \min_i\{s_i^*\} \leq 1$, where s_i^* denotes the i-th entry of $\mathbf{s}^*$, then $\mathbf{s}^*$ is called an optimal sparsity vector of $\mathbf{X}^s$, and the corresponding permutation is call an optimal permutation of $\mathbf{X}$.*

Lemma 4: *For a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ and its best s-term approximation $\mathbf{X}^s$, there exists at least one optimal sparsity vector $\mathbf{s}^*$ of $\mathbf{X}^s$.*

Proof: Obviously, $||\mathbf{s}^*||_1 = s$. If $\lceil s/N\rceil = \lfloor s/N\rfloor = s/N$, we can immediately find an optimal sparsity vector $\mathbf{s}^*$ whose entries are all s/N. Here $\lceil\cdot\rceil$ denotes the ceiling function and $\lfloor\cdot\rfloor$ denotes the floor function. If $\lceil s/N\rceil \neq \lfloor s/N\rfloor$, we consider a permutation on $\mathbf{X}$ such that: for the best s-term approximation of the resulted 2D signal, there are $\lceil s/N\rceil$ nonzero entries in each of the first $s - N\lfloor s/N\rfloor$ columns, and the remaining nonzero entries are evenly distributed among the remaining columns. Then each of the latter $N\lceil s/N\rceil - s$ columns has $\lfloor s/N\rfloor$ nonzero entries. Therefore, the sparsity vector of the best s-term approximation of the permuted 2D signal is an optimal sparsity vector. This completes the proof. ■

From the proof of Lemma 4, it follows that the optimal sparsity vector and the optimal permutation may not be unique, and the Chebyshev norm of an optimal sparsity vector of $\mathbf{X}^s$ is equal to $\lceil s/N\rceil$.

In most scenarios, finding an optimal permutation may not be practical. Then an acceptable permutation defined in Definition 2 can be used instead.

B. Group-scan-based and Zigzag-scan-based Permutations

The following observation is of interest. For a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$, consider a permuted signal $\mathbf{X}^\dagger$ and its best s-term approximation $\mathbf{X}^{\dagger s}$. For any $i \in \{1, 2, \cdots, N\}$, if all entries in the i-th row of $\mathbf{X}^{\dagger s}$ share the same probability to be nonzero, denoted as p_i, then all columns of $\mathbf{X}^{\dagger s}$ have the same expected sparsity level, given as $\sum_{i=1}^M p_i$.

Example 1: When $M = N = 4$, if after a permutation, the entries in the 1st, 2nd, 3rd, and 4th rows of $\mathbf{X}^{\dagger s}$ have respectively probabilities $p_1 = 0.9$, $p_2 = 0.3$, $p_3 = 0.2$ and $p_4 = 0.1$ to be nonzero, then for the sparsity vector of $\mathbf{X}^{\dagger s}$, denoted as $\mathbf{s}^\dagger = [s_1^\dagger, s_2^\dagger, s_3^\dagger, s_4^\dagger]$, we have

$$\mathrm{E}\left\{\max_j\{s_j^\dagger\} - \min_j\{s_j^\dagger\}\right\} = 1.3881$$

and

$$\Pr\left\{\max_j\{s_j^\dagger\} - \min_j\{s_j^\dagger\} \leq 1\right\} = 0.6003$$

where $\mathrm{E}\{\cdot\}$ means expectation and $\Pr\{\cdot\}$ means probability of an event. Thus, the permutation in this example is optimal with probability 0.6003.

For the best s-term approximation $\mathbf{X}^s$ of a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$, consider that entries in $\mathbf{X}^s$ can be divided into several non-overlapped groups, and in each group all entries share the same probability to be nonzero. Based on the observation in the two preceding paragraphs, a permutation, named *group-scan-based permutation*, can work as follows: 1) preform group-by-group scan[6] of the 2D compressible signal $\mathbf{X}$ into a vector, and 2) row-wisely reshape the resulted vector into a new $M \times N$ 2D signal. In this way, all nonzero entries of the best s-term approximation of the new 2D signal are likely to be evenly distributed among all columns.

Definition 4: *For a 2D signal $\mathbf{X} \in \mathbb{R}^{M\times N}$, let $\mathbf{X}(i, j)$ denote the entry in the position (i, j). The m-th ($1 \leq m < M + N$) layer of $\mathbf{X}$ is the group of all entries $\mathbf{X}(i, j)$'s satisfying $i + j - 1 = m$.*

Example 2: When $M = N = 4$, the following matrix $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_6 & x_7 \\ x_3 & x_5 & x_8 & x_{13} \\ x_4 & x_9 & x_{12} & x_{14} \\ x_{10} & x_{11} & x_{15} & x_{16} \end{bmatrix} \tag{4}$$

[6]That is, first scan all entries in the first group, then scan all entries in the second group, and so on.

has 7 layers, including $\{x_1\}$, $\{x_2, x_3\}$, $\{x_4, x_5, x_6\}$, $\{x_7, x_8, x_9, x_{10}\}$, $\{x_{11}, x_{12}, x_{13}\}$, $\{x_{14}, x_{15}\}$, $\{x_{16}\}$, respectively. The layers of $\mathbf{X}$ are parallel to each other.

For a 2D compressible signal $\mathbf{X}$, if all entries within each layer of its best s-term approximation $\mathbf{X}^s$ have similar probabilities to be nonzero (an example when this condition is satisfied is to be given later in this subsection), we propose the following zigzag-scan-based permutation, which is a special example of the group-scan-based permutation.

Define the *zigzag-scan-based permutation* P: $\mathbb{R}^{M\times N} \to \mathbb{R}^{M\times N}$ for a 2D signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ as $\mathrm{P}(\mathbf{X}) = \mathrm{R}(\mathrm{Z}(\mathbf{X}))$, where R: $\mathbb{R}^{MN} \to \mathbb{R}^{M\times N}$ is the row-wisely reshaping function which row-wisely turns a vector into an $M \times N$ matrix and Z: $\mathbb{R}^{M\times N} \to \mathbb{R}^{MN}$ is the zigzag scan function which turns a matrix into a "zigzag" sequence vector.

Correspondingly, define the *inverse zigzag-scan-based permutation* P^{-1}: $\mathbb{R}^{M\times N} \to \mathbb{R}^{M\times N}$ for a 2D signal $\mathbf{X}^\dagger \in \mathbb{R}^{M\times N}$ as $\mathrm{P}^{-1}(\mathbf{X}^\dagger) = \mathrm{Z}^{-1}(\mathrm{R}^{-1}(\mathbf{X}^\dagger))$, where R^{-1}: $\mathbb{R}^{M\times N} \to \mathbb{R}^{MN}$ is a vectorization function which row-wisely turns a matrix into a vector and Z^{-1}: $\mathbb{R}^{MN} \to \mathbb{R}^{M\times N}$ is the inverse zigzag scan function which turns a "zigzag" sequence into an $M \times N$ matrix.

Example 3: The matrix $\mathbf{X}$ given in (4) becomes a "zigzag" sequence after zigzag scan, i.e.,

$$\mathrm{Z}(\mathbf{X}) = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, \cdots, x_{16}]$$

and then becomes the permuted signal $\mathbf{X}^\dagger$ after row-wisely reshaping, that is,

$$\begin{aligned}\mathbf{X}^\dagger &= \mathrm{P}(\mathbf{X}) = \mathrm{R}(\mathrm{Z}(\mathbf{X}))\\ &= \mathrm{R}([x_1, x_2, x_3, x_4, x_5, x_6, x_7, \cdots, x_{16}])\\ &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4\\ x_5 & x_6 & x_7 & x_8\\ x_9 & x_{10} & x_{11} & x_{12}\\ x_{13} & x_{14} & x_{15} & x_{16}\end{bmatrix}\end{aligned}$$

and again becomes a "zigzag" sequence after vectorization, that is,

$$\mathrm{R}^{-1}(\mathbf{X}^\dagger) = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, \cdots, x_{16}]$$

and then returns to the original signal $\mathbf{X}$ after inverse zigzag scan, i.e., $\mathrm{P}^{-1}(\mathbf{X}^\dagger) = \mathrm{Z}^{-1}(\mathrm{R}^{-1}(\mathbf{X}^\dagger)) = \mathbf{X}$.

Thus, according to the analysis at the beginning of this subsection, if entries in each layer of $\mathbf{X}^s$ share similar probabilities to be nonzero, after the zigzag-scan-based permutation, all nonzero entries of the permuted $\mathbf{X}^s$ tend to be evenly distributed among all columns.

Definition 5: *Consider a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ and its best s-term approximation $\mathbf{X}^s$. For given transition layer indices r_0, r_1, r_2 and a decay factor α, we say that $\mathbf{X}$ follows the (r_0, r_1, r_2, α)-layer model if for any entry in the m-th layer of $\mathbf{X}^s$, its probability to be nonzero is given as*

$$p_m = \begin{cases} 0 & 1 \le m \le r_0\\ 1 & r_0+1 \le m \le r_1\\ e^{-\alpha(m-r_0-1)} & r_1+1 \le m \le r_2\\ 0 & r_2+1 \le m \le M+N-1.\end{cases}$$

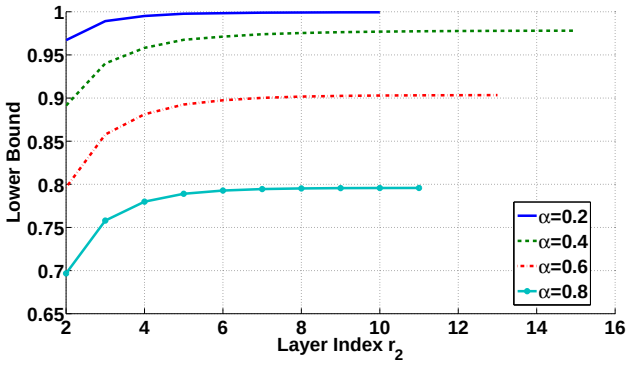


Fig. 1: Lower bound of $\Pr\{\mathrm{P} \text{ is acceptable}\}$ in (5) for $r_0 = 0, r_1 = 1$.

Based on the (r_0, r_1, r_2, α)-layer model, we have the following proposition for the zigzag-scan-based permutation.

Proposition 1: *If a 2D compressible signal $\mathbf{X} \in \mathbb{R}^{M\times N}$ follows the (r_0, r_1, r_2, α)-layer model with $r_2 \ge 2r_1 - 3r_0 - 1$ and $0 \le r_0 < r_1 < r_2 \le \min\{M, N\}$, the zigzag-scan-based permutation $P(\cdot)$ is an acceptable permutation with a large probability given as*

$$\begin{aligned}&\Pr\{\mathrm{P} \text{ is acceptable}\} = \Pr\left\{||\mathbf{s}||_\infty > ||\mathbf{s}^\dagger||_\infty\right\}\\ &\ge 1 - \left[\prod_{m=r_1+1}^{r_2}\left(1-e^{-\alpha(m-r_0-1)}\right)^m\right]\cdot\prod_{j=1}^{r_2}\left\{1+\sum_{k=k_j+1}^{\min\{\lceil (r_0+r_2+1)/2\rceil,\, r_2-r_0, r_2-j+1\}}\right.\\ &\left.\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{1}{(e^{\alpha(a_1-1-r_0)}-1)\cdots(e^{\alpha(a_{k-k_j}-1-r_0)}-1)}\right\}\end{aligned} \tag{5}$$

where $\mathbf{s}$ and $\mathbf{s}^\dagger$ are the sparsity vectors of the best s-term approximation of $\mathbf{X}$ and $\mathbf{X}^\dagger$, respectively, $\mathbf{X}^\dagger$ is a 2D signal after the zigzag-scan-based permutation, and for $1 \le j \le r_2$, $\mathcal{A}_j \triangleq \{m_j, m_j+1, \cdots, r_2\}$, $m_j = \max\{r_1+1, j\}$, and

$$k_j = \begin{cases} r_1 - r_0, & 1 \le j \le r_0\\ r_1 - j + 1, & r_0+1 \le j \le r_1\\ 0, & r_1+1 \le j \le r_2.\end{cases}$$

Proof: See Appendix for the proof. ■

Figs. 1–3 show the value of the lower bound on $\Pr\{\mathrm{P} \text{ is acceptable}\}$ in (5) under different α and r_2 for 1) $r_0 = 0$, $r_1 = 1$; 2) $r_0 = 0, r_1 = 2$; and 3) $r_0 = 3, r_1 = 5$. It can be seen that the lower bound on $\Pr\{\mathrm{P} \text{ is acceptable}\}$ is large enough in general. For other r_0 and r_1, the results are similar.

From Proposition 1, it can be seen that the zigzag-scan-based permutation is an acceptable permutation for a very broad class of signals. The knowledge of exact locations of the nonzero entries of the best s-term approximation $\mathbf{X}^s$, i.e., the knowledge of the support of the 2D signal $\mathbf{X}^s$, is not needed.

As an example, we show that the zigzag-scan-based permutation is particularly useful for 2D discrete cosine transform (DCT2) coefficient matrices of 2D piecewise smooth image signals. Since the DCT2 coefficient matrix of a piecewise smooth image signal typically has most of its large entries

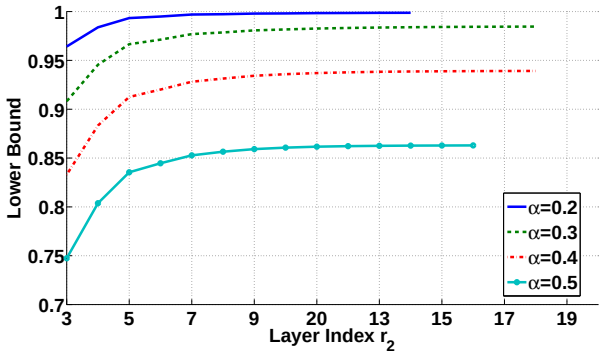


Fig. 2: Lower bound of $\Pr\{\text{P is acceptable}\}$ in (5) for $r_0 = 0, r_1 = 2$.

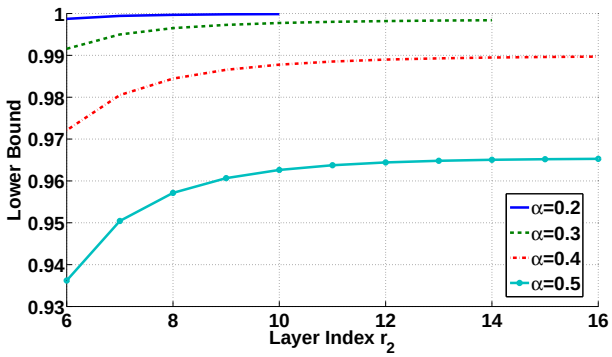
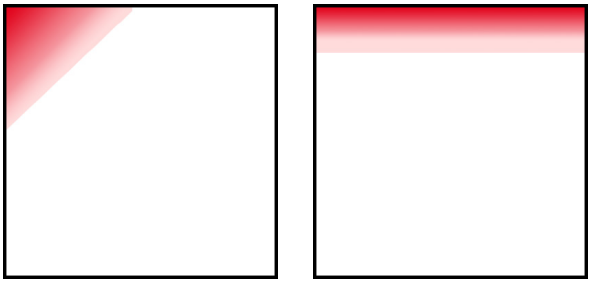


Fig. 3: Lower bound of $\Pr\{\text{P is acceptable}\}$ in (5) for $r_0 = 3, r_1 = 5$.

lie in the top left corner, and small entries lie in the bottom right corner because most of its energy is concentrated in low frequencies, the zigzag scan process is commonly used in image compression like JPEG [19]. Thus, the DCT2 coefficient matrices of piecewise smooth image signals satisfy the (r_0, r_1, r_2, α)-layer model with $r_0 = 0$. It will also be shown via simulation in Subsection V-B. Thus, the proposed zigzag-scan-based permutation has a large probability to be an acceptable permutation when parallel CS is applied to the DCT2 coefficient matrices. Note that the knowledge of the layer indices r_0, r_1, r_2 and the decay factor α of the layer model is not needed when applying the zigzag-scan-based permutation to the DCT2 coefficient matrices.

Fig. 4 shows the difference before and after the zigzag-scan-based permutation when the 2D signal is the DCT2 coefficient matrix of an image. The energy, which can be loosely viewed as an interpretation of the sparsity vector if all non-zero entries of the 2D signal have magnitude of the same order, is distributed more evenly among columns after the zigzag-scan-based permutation.

One advantage of the zigzag-scan-based permutation is that it is a pre-defined permutation, and thus, the encoder and decoder know it in advance without any additional communication to each other. In Subsection V-B, we will also show by simulation that the zigzag-scan-based permutation is an acceptable permutation for DCT2 coefficient matrices of several typical images.



(a) Before the zigzag-scan-based permutation

(b) After the zigzag-scan-based permutation

Fig. 4: Energy distribution of a DCT2 coefficient matrix before and after the zigzag-scan-based permutation.

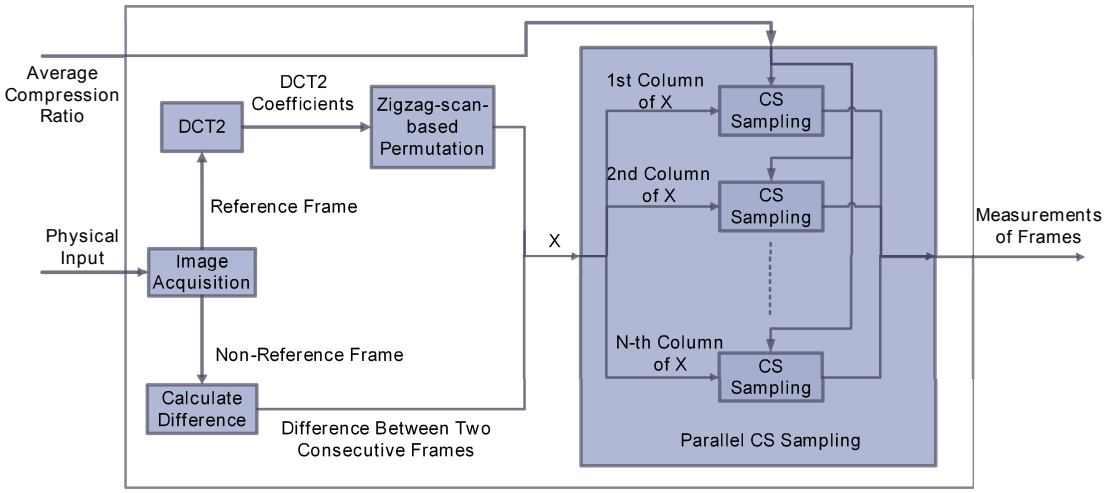
IV. Example of Video Compression via Parallel CS with Permutations in Wireless Multimedia Sensor Networks

As an application example, we design a pair of CS video encoder and decoder based on the parallel CS with the proposed zigzag-scan-based permutation. This CS video encoder and decoder can be plugged into the application layer of the compressive distortion minimizing rate control (C-DMRC) system [20]. As discussed in [20], in wireless multimedia sensor networks, the C-DMRC system is preferred compared to traditional video coding standards such as MPEG and H.264, since the C-DMRC system has less-complex video encoder and can tolerate much higher bit error rates. The CS video encoder and decoder in the C-DMRC system are built based on the block CS architecture proposed in [9]. Thus, as we discussed in Section I, the CS video decoder in the C-DMRC system requires a joint reconstruction. By replacing the CS video encoder and decoder at the application layer of the C-DMRC system with the CS video encoder and decoder based on parallel CS architecture, the computational complexity of the video decoder can be reduced and the reconstruction process can be parallelized.

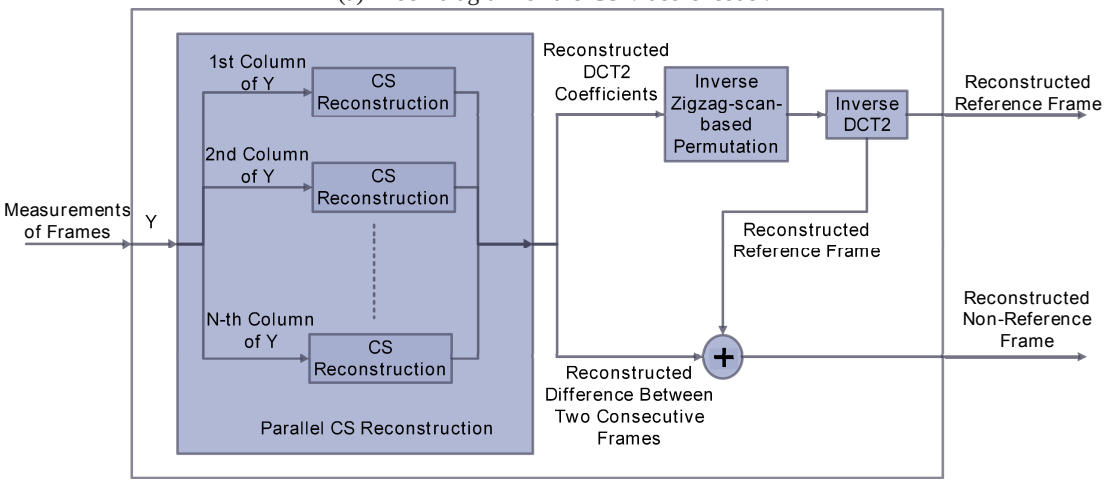
In this example, frames with odd indices and even indices are taken as reference frames and non-reference frames, respectively.[7] The block diagram of the CS video encoder is shown in Fig. 5(a). The average compression ratio[8] is computed by the rate controller at the transport layer according to current network status, e.g., the end-to-end round trip time and the estimated sample loss rate of the network, and it controls the number of measurements for a video frame. For every pair of a reference frame and its following non-reference frame, the rate controller gives an average compression ratio. According to this average compression ratio, the compression ratios of the reference and non-reference frames in a pair are obtained. At the output of the CS video encoder, we have the frame measurements. The image acquisition device turns the physical input into video frames and outputs the video frames to different processing blocks according to the frame index.

[7] More sophisticated index assignment schemes for the reference frame and non-reference frame can be used as well.

[8] Here the compression ratio is the ratio of the number of measurements to the number of entries in the target signal.



(a) Block diagram of the CS video encoder.



(b) Block diagram of the CS video decoder.

Fig. 5: Block diagram of the CS video encoder and decoder.

The procedure for encoding the reference frame is as follows: 1) compute DCT2 on the reference frame; 2) perform the zigzag-scan-based permutation on the DCT2 coefficient matrix; 3) perform parallel compressed sampling of the permuted DCT2 coefficient matrix. The procedure for encoding the non-reference frame is as follows: 1) compute the difference between the non-reference frame and the preceding reference frame; 2) perform parallel compressed sampling of this difference. The outputs of all CS sampling processors are combined.[9] For the non-reference frames, no permutation is performed since the difference between two consecutive frames, especially in videos with slow motion, is sparse enough so that the sparsity level of each column is too small to have significant difference from column to column. Thus, the permutation does not bring significant improvement, which we have checked by simulations in [17].

Considering that the sparsity level of the difference between a non-reference frame and its preceding reference frame is smaller than that of the DCT2 coefficient matrix of the reference frame, the compression ratio of the non-reference frames should be higher than that of the reference frames, i.e., fewer measurements are assigned to the non-reference frames. In our experiment in Section V, we set the ratio of measurements being 4:1, i.e., the number of measurements for reference frames is 4 times that for non-reference frames. For example, if current average compression ratio given by the rate controller is 0.5, then the compression ratio of the reference frame is 0.8 and the compression ratio of the non-reference frame is 0.2. Other ratios can be set according to the motion intensity of the video.

The block diagram of the CS video decoder is shown in Fig. 5(b). To decode a reference frame at the receiver side, the following steps are performed: 1) parallel CS reconstruction of the permuted DCT2 coefficient matrix from the measurements of the reference frame; 2) the inverse zigzag-scan-based permutation on the reconstructed permuted DCT2 coefficient matrix; 3) inverse DCT2 on the reconstructed DCT2 coefficient matrix. To decode a non-reference frame, the following steps are performed: 1) parallel CS reconstruction of the difference between the non-reference frame and its preceding reference frame from the measurements of the non-reference frame;

[9]Quantization is omitted in the example presented here, but it has to been done in a practical video coding scenario.

2) add the reconstructed difference between the non-reference frame and its preceding reference frame to the corresponding reconstructed reference frame. For parallel CS reconstruction, any ℓ_1-norm minimization solver, e.g., the basis pursuit algorithm, can be used.

V. Simulation Results

A. Empirical Performance of Parallel CS with Permutation

Our first experiment considers an s-sparse 2D signal $\mathbf{X} \in \mathbb{R}^{N\times N}$ whose sparsifying basis is the identity basis. To compare the reconstruction error performance of the parallel CS with permutation framework with that of other frameworks including the Kronecker CS framework [6], we consider the following four schemes to acquire K measurements of $\mathbf{X}$.

(i) Sample the vectorized 2D signal using a dense Gaussian sensing matrix $\mathbf{A}_{\rm g} \in \mathbb{R}^{K\times N^2}$, i.e., $\mathbf{y}_{\rm g} = \mathbf{A}_{\rm g}\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \mathrm{vec}(\mathbf{X})$ is the vectorized signal $\mathbf{X}$ and $\mathbf{y}_{\rm g}$ is called as a vector of *global measurements*. Here the subscript 'g' stands for "global".

(ii) Sample the 2D signal using Kronecker sampling operator, i.e., $\mathbf{Y}_{\rm K} = \mathbf{A}_1\mathbf{X}\mathbf{A}_2^T$, or equivalently, $\mathbf{y}_{\rm K} = \mathbf{A}_{\rm K}\bar{\mathbf{x}}$, where both $\mathbf{Y}_{\rm K}$ and $\mathbf{y}_{\rm K}$ are called *Kronecker measurements*, $\mathbf{A}_1$ and $\mathbf{A}_2$ are dense Gaussian sensing matrices of size $\sqrt{K}\times N$, and $\mathbf{A}_{\rm K} \in \mathbb{R}^{K\times N^2}$ is the Kronecker product of $\mathbf{A}_1$ and $\mathbf{A}_2$. Here the subscript 'K' stands for "Kronecker". In our simulation, considering that $\sqrt{K}$ is not always an integer, we substitute $\sqrt{K}$ by $\lceil\sqrt{K}\rceil$. Thus, for this scheme, totally $\lceil\sqrt{K}\rceil^2$ measurements are acquired.

(iii) Sample the 2D signal column-by-column with permutation, i.e., using the same dense Gaussian sensing matrix $\mathbf{A}_{\rm ip} \in \mathbb{R}^{K/N\times N}$ to sample each column of the permuted 2D signal. In other words, $\mathbf{Y}_{\rm ip} = \mathbf{A}_{\rm ip}\mathrm{P}^*(\mathbf{X})$, where $\mathbf{Y}_{\rm ip}$ is a vector of *independent measurements with permutation* and $\mathrm{P}^*(\cdot)$ is the optimal permutation operator. In our experiment, we use K that makes K/N an integer. Here the subscript 'ip' stands for "independent with permutation".

(iv) Sample the 2D signal column-by-column using the same dense Gaussian sensing matrix $\mathbf{A}_{\rm i} \in \mathbb{R}^{K/N\times N}$, i.e., $\mathbf{Y}_{\rm i} = \mathbf{A}_{\rm i}\mathbf{X}$, where $\mathbf{Y}_{\rm i}$ is a vector of *independent measurements without permutation*. Here the subscript 'i' stands for "independent". As before, we use K that makes K/N an integer.

We set $N = 20$ and the sparsity level $s = 40$. Joint reconstruction is used for Schemes (i) and (ii), while column-by-column reconstruction is used for Schemes (iii) and (iv). We let the number of measurements K vary from 20 to N^2. For Schemes (iii) and (iv), we let $\mathbf{A}_{\rm ip} = \mathbf{A}_{\rm i}$. For each value of K, we average 100 iterations by generating s-sparse signals $\mathbf{X}$ with independent and identically distributed Gaussian entries and with support following a uniform distribution among all supports of size s. We then measure the probability of successful reconstruction for each value of K, where the success is declared if the signal estimate $\hat{\mathbf{X}}$ obeys $||\mathrm{vec}(\mathbf{X}-\hat{\mathbf{X}})||_2 \le 10^{-3}||\mathrm{vec}(\mathbf{X})||_2$, as used in [6].

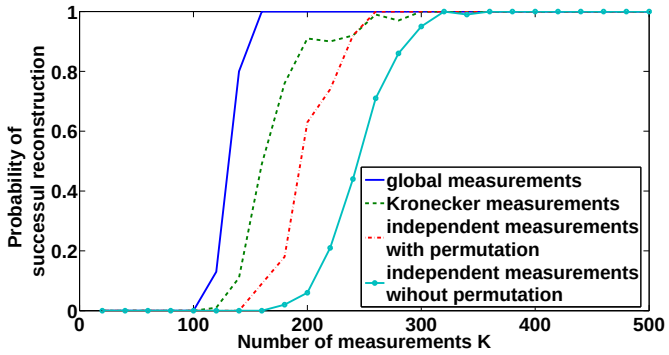


Fig. 6: Empirical performance of different sampling and reconstruction schemes.

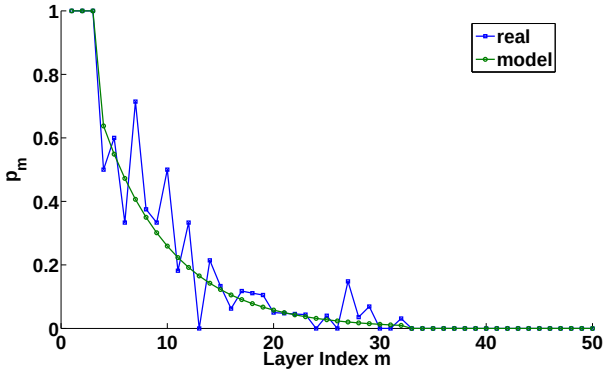


Fig. 7: p_m of real image "Boat.tiff" and p_m of the layer model with $r_0 = 0, r_1 = 3, r_2 = 32$, and $\alpha = 0.15$.

Fig. 6 shows the results for these four schemes. It can be seen that Scheme (i) has the best reconstruction error performance, while Scheme (iv) has the worst one. Both Schemes (ii) and (iii) have better reconstruction error performance than Scheme (iv). In addition, Scheme (ii) has slightly better performance compared to Scheme (iii). However, to achieve 90% successful reconstruction, the number of measurements needed is almost the same for both of the schemes. From this observation, we conclude that the parallel CS with permutation framework can achieve similar reconstruction error performance as the Kronecker CS framework, while it has much lower computational complexity since the column-by-column reconstruction is used in the parallel CS with permutation framework.

B. The Layer Model and the Zigzag-scan-based Permutation

In this example, we check the layer model for the DCT2 coefficient matrix of the gray image: Boat (512 × 512). The format used in the simulation is tagged image file format (TIFF). The best s-term approximation is obtained by keeping all DCT2 coefficients with magnitudes not less than 1000 and changing the remaining to zeros. In Fig. 7, the x-axis is the layer index m, and y-axis is the probability of an entry in the m-th layer of the best s-term approximation $\mathbf{X}^s$ of the DCT2 coefficient matrix to be nonzero, calculated as $p_m = (1/m)\sum_{i+j-1=m}\mathrm{I}(\mathbf{X}^s(i,j) \neq 0)$, where $\mathrm{I}(\cdot)$ is the

TABLE I: Comparison of $||\mathbf{s}||_\infty$ before and after the zigzag-scan-based permutation.

Image	Magnitude Threshold			
	400	600	800	1000
Boat	33 vs. 2	23 vs. 2	19 vs. 2	16 vs. 1
Cameraman	13 vs. 2	8 vs. 2	7 vs. 1	4 vs. 1
Lena	14 vs. 3	11 vs. 2	8 vs. 1	7 vs. 1
Peppers	27 vs. 3	15 vs. 2	11 vs. 2	11 vs. 2

indicator function. The p_m's versus layer index m for the real image "Boat.tiff" and the result of the (r_0, r_1, r_2, α)-layer model with $r_0 = 0, r_1 = 3, r_2 = 32, \alpha = 0.15$ are shown in Fig. 7. It can be seen that the two curves are close to each other. Similar results are also achieved for other images. Then according to Proposition 1, the zigzag-scan-based permutation is an acceptable permutation with an overwhelming probability for DCT2 coefficient matrices of such images.

The changes of $||\mathbf{s}||_\infty$ of the best s-term approximation of the DCT2 coefficient matrix before and after the zigzag-scan-based permutation are shown in Table I. The DCT2 coefficient matrices are taken from four test images: Boat (512×512), Cameraman (256×256), Lena (512×512), Peppers (512×512). The best s-term approximation is chosen according to different magnitude thresholds, i.e., keeping DCT2 coefficients whose magnitudes are not less than the magnitude threshold and setting the remaining to be zeros. Table I shows that $||\mathbf{s}||_\infty$ decreases significantly after the zigzag-scan-based permutation, which is consistent with Proposition 1.

C. Image Compression via Parallel CS with the Zigzag-Scan-Based Permutation

The performance of image compression via parallel CS with the zigzag-scan-based permutation is shown by compressing the DCT2 coefficients of four images: Boat, Cameraman, Lena, and Peppers. The PSNR is employed to show the reconstruction performance. We compare the performances of the parallel CS scheme for the configurations: 1) without permutation; 2) with the zigzag-scan-based permutation. Entries of the sensing matrix $\mathbf{A} \in \mathbb{R}^{K \times M}$ are drawn from Gaussian ensembles, with variance being $1/K$. The parallel CS reconstruction is implemented using basis pursuit algorithm by the CVX optimization toolbox.[10] Note that other reconstruction algorithms than the basis pursuit can also be used. PSNR performance for different methods is shown in Fig. 8 versus the compression ratio.

From Fig. 8, we can see that under the same compression ratio, the zigzag-scan-based permutation helps to improve the PSNR by around 2∼4 dB for all images. Consequently, it shows that the PSNR performance is indeed improved significantly after permutation.

Note that for the block CS framework of [7] and [8], if DCT2 is performed on each block of the image, instead of the whole image, parallel reconstruction can also be achieved. On the other hand, in the parallel CS with permutation framework, the DCT2 is performed on the whole image. To compare the reconstruction error performance of these two schemes where parallel sampling and reconstruction can be achieved, we study the PSNR values for the image Lena (256×256). For the block CS framework, the block dimension is 16×16 and the DCT2 is performed on every block. Moreover, all vectorized blocks are sampled independently. For the parallel CS with permutation, the DCT2 is performed on the whole 2D image and the DCT2 coefficients matrix is sampled column-by-column. Both schemes use parallel reconstruction, either block-by-block or column-by-column, and the reconstruction is implemented using basis pursuit algorithm. The comparison between the two schemes is shown in Table II. For the block CS framework, Table II also shows the PSNR values obtained by using the smoothed projected Landweber (SPL) algorithm proposed in [8]. It can be seen from Table II that the parallel CS with permutation scheme outperforms the block-by-block sampling and reconstruction scheme, although both of them can achieve parallel reconstruction. Compared to the block CS with SPL reconstruction scheme, there is a 4 dB degradation in terms of PSNR when using the parallel CS with permutation scheme. However, notice that the SPL algorithm is a joint reconstruction scheme. This degradation is thus expected since the permutation, though can improve the error performance of the column-by-column reconstruction, can hardly achieve better reconstruction error performance than that of the joint reconstruction, which has a higher computational complexity.

[10]The toolbox can be downloaded at http://cvxr.com/cvx.

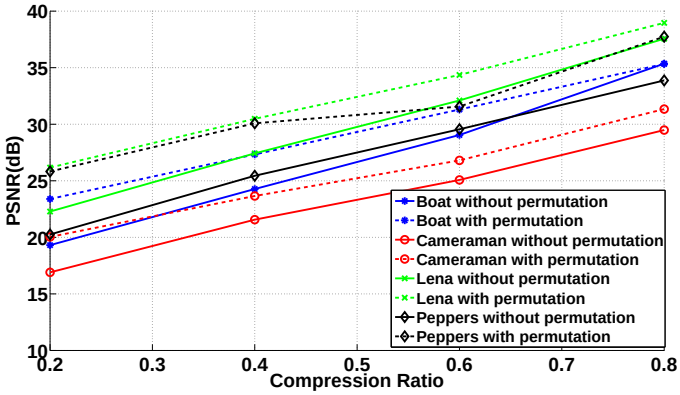


Fig. 8: PSNR for the parallel CS scheme with/without the zigzag-scan-based permutation in image compression.

TABLE II: Comparison among block CS-based frameworks and parallel CS with permutation framework.

Compression Ratio		0.1	0.2	0.3	0.4	0.5
PSNR (dB)	Block CS (block reconstruction)	18.38	20.89	23.64	26.14	27.36
	Block CS (SPL reconstruction)	24.51	27.68	29.75	31.39	33.06
	Parallel CS with permutation	20.38	23.18	25.67	27.60	29.50

TABLE III: Comparison between the block-by-block sampling and reconstruction with zigzag scan scheme of [21] and parallel CS with the zigzag-scan-based permutation scheme.

Compression Ratio		0.75	0.5	0.625	0.375
PSNR (dB)	[21]	34.89	30.94	31.17	27.68
	Parallel CS with permutation	35.19	29.50	32.14	27.17

This result coincides with our result in Subsection V-A as well.

In [21], it is proposed to sample and reconstruct a 2D image block-by-block. The DCT2 coefficients read in zigzag scan order of each block are divided into a group of important components and a group of unimportant components. In this way, the error performance of the simple block-by-block reconstruction can be improved, and a similar computational complexity as that of the parallel CS with permutation scheme can be achieved. To make a fair comparison between the *nonuniform sampling* with zigzag scan scheme in [21] and the parallel CS with permutation scheme, we use "Lena" image of size 256×256, and the block size for the nonuniform sampling with zigzag scan scheme is selected to be 16×16. In this way, the length of each vectorized block is equal to the length of each column of the image. Orthogonal matching pursuit[11] (OMP) as described in [21] is used for reconstruction in the nonuniform sampling with zigzag scan scheme. Similar to the experiment of [21], for compression ratios 0.75 and 0.5, the important/unimportant coefficients rate is 1/3; for compression ratios 0.625 and 0.375, the important/unimportant coefficients rate is 1/7. It can be seen from Table III that there is no absolute winner between the method of [21] and our scheme. Both schemes have similar reconstruction error performance.

D. Video Compression via Parallel CS with the Zigzag-scan-based Permutation

The test video sequences in this example are three standard YUV video sequences: Akiyo, Foreman, Coastguard. The format used in the simulation is quarter common intermediate format (QCIF). The performance of the proposed video compression scheme is shown by compressing the luminance components of the first 10 frames, i.e., 5 reference frames and 5 non-reference frames. The average PSNRs for reference and non-reference frames are used as performance metrics. All settings are the same as in the example in Fig. 8 in Subsection V-C. PSNR performance for different methods is shown in Figs. 9 and 10 versus the average compression ratio, that is, $(\text{compression ratio of reference frames} + \text{compression ratio of non-reference frames})/2$.

From Fig. 9, we can see that under the same average compression ratio, the zigzag-scan-based permutation helps to improve the PSNR of reference frames by around 3∼9 dB for Akiyo, 5∼6 dB for Foreman and 4∼8 dB for Coastguard. Fig. 10 shows that the zigzag-scan-based permutation also

[11]We use the OMP algorithm implemented in SparseLab package that can be downloaded at http://sparselab.stanford.edu.

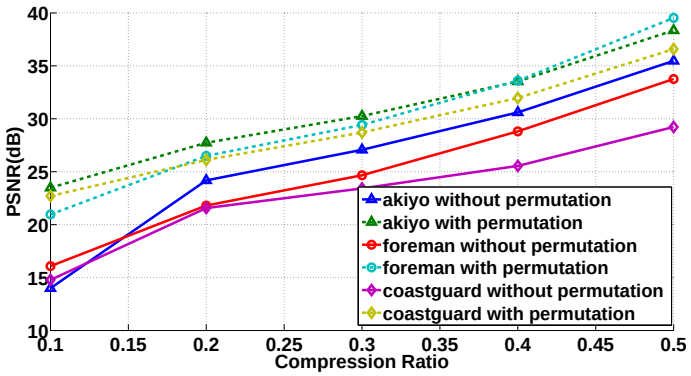


Fig. 9: Average PSNR of reconstructed reference frames.

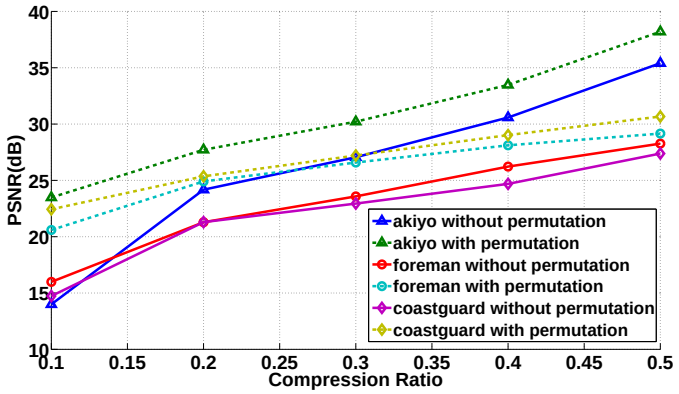


Fig. 10: Average PSNR of reconstructed non-reference frames.

improves the PSNR performance of non-reference frames by around 3∼9 dB for Akiyo, 2∼5 dB for Foreman and 3∼7 dB for Coastguard. The improved PSNR for non-reference frames is a bit lower than that of the preceding reference frames because the reconstruction of non-reference frames relies on both the reconstruction of their corresponding preceding reference frames and the reconstruction of the difference between non-reference frames and their corresponding preceding reference frames.

To show the advantage of the video compression scheme proposed in Section IV, we compare the total time of reconstructing one pair of reference and non-reference frames using (i) the video encoder and decoder employed in the C-DMRC system proposed in [20] and (ii) the video encoder and decoder proposed in Section IV. We also show the PSNRs of reconstructed reference and non-reference frames for both schemes. The video sequence used in the simulation is the standard YUV sequence Akiyo (QCIF format). The measurement matrices used in the C-DMRC system and our scheme are the scrambled block Hadamard matrix with block length equal to 32 and the random Gaussian matrix, respectively. For the reference frame, the DCT2 basis is used as the sparsifying basis in both schemes. The CS reconstruction algorithm is implemented using the l_1-magic package.[12] To eliminate the effects of randomness, we run 200 trials for each average compression ratio and show the average PSNR and the total

[12]The package is available at http://users.ece.gatech.edu/~justin/l1magic.

TABLE IV: Total reconstruction time and PSNR of reconstructed video frames using the video encoder and decoder employed in [20].

Average Compression Ratio		0.1	0.2	0.3	0.4	0.5
Reconstruction Time (seconds)		55.32	47.34	37.23	37.08	30.49
PSNR (dB)	Reference Frame	24.43	27.52	29.79	32.53	36.24
	Non-reference Frame	24.44	27.53	29.73	32.27	35.32

TABLE V: Total reconstruction time and PSNR of reconstructed video frames using the video encoder and decoder proposed in Section IV.

Average Compression Ratio		0.1	0.2	0.3	0.4	0.5
Reconstruction Time (seconds)		12.85	14.30	20.17	18.40	18.67
PSNR (dB)	Reference Frame	24.17	27.30	30.32	33.79	38.34
	Non-reference Frame	24.17	27.29	30.29	33.71	38.10

reconstruction time. The results are shown in Tables IV and V.

It can be seen from Tables IV and V that the reconstruction time using the video encoder and decoder proposed in Section IV is less than that for the video encoder and decoder employed in [20], especially when the compression ratio is low. In addition, if there are multiple decoding processors simultaneously reconstructing the columns of a video frame as shown in Fig. 5(b), the reconstruction time of the video decoder proposed in Section IV can be further reduced approximately to the total reconstruction time divided by the number of decoding processors. It can also be observed in Table IV that the time for reconstruction using the video encoder and decoder employed in [20] decreases as the average compression ratio increases. This is because the reconstruction algorithm converges faster as the number of measurements increases. According to Table V, the time for reconstruction using the video encoder and decoder proposed in Section IV is less sensitive to the compression ratio. In addition, we can see that as compared to the video encoder and decoder employed in [20], the PSNR of reconstructed video frames for the video encoder and decoder proposed in Section IV is larger, when the average compression ratio is larger than 0.3, and it is almost the same (less than 0.3dB degradation), when the average compression ratio is smaller than 0.3.

VI. Conclusion and Discussion

A parallel CS scheme with permutation has been proposed. It has been proved that with a so-called acceptable permutation, the RIP condition for the sensing matrix in the parallel CS can be relaxed. The group-scan-based permutation has been introduced. As an example, the zigzag-scan-based permutation for 2D signals satisfying the (r_0, r_1, r_2, α)-layer model, such as DCT2 coefficient matrices of 2D images, has been analyzed. The application to image and video compression has been discussed as well. In the simulations, it has been shown that the zigzag-scan-based permutation for DCT2 coefficient matrices of images is an acceptable permutation. In addition, the simulation results have shown that the proposed scheme improves the reconstruction performance of images and videos in terms of PSNR significantly.

It is worth mentioning that the zigzag-scan-based permutation is designed for signals satisfying the proposed layer model. If a signal has most of its large entries clustered around one or more fixed locations, the more general group-scan-based permutation is applicable. Similar to the zigzag-scan-based permutation for the layer model, a lower bound on the probability that the group-scan-based permutation is an acceptable permutation can be derived given a mathematical model for the distribution pattern of large entries in the signal.

Comparing the proposed approach to other existing alternatives, the following discussion is of interest. The column-by-column sampling and reconstruction considered in the parallel CS framework has been studied experimentally in several works [6], [10]. For example, the DMD-based compressive hyperspectral imaging operator presented in [6] is a parallel system where each spectral image is measured with the same sampling operator. If each spectral image is reconstructed one-by-one, the reconstruction can be parallelized. However, in this case, the Kronecker CS system becomes essentially the same as the block-by-block sampling and reconstruction scheme where DCT2 is performed on every block. This block-by-block sampling and reconstruction scheme has been discussed in Subsection V-C. Thus, in a similar manner, the parallel CS with permutation proposed in this paper outperforms the parallel sampling and reconstruction scheme for the Kronecker CS.

For a 2D signal which is sparse in the identity basis, if the same sampling operator is applied to each row, the row-by-row scan in [10] is similar to the column-by-column sampling scheme in the parallel CS framework. To achieve a better reconstruction error performance, an algorithm is proposed in [10], which iteratively improves the current estimate of the 2D sparse signal by modelling statistical dependencies between neighboring rows. We have proposed a different approach here, that is, permutation, to achieve a better reconstruction error performance than the basic column-by-column reconstruction. The introduction of permutation into the column-by-column sampling makes the sampling operator in this paper different from that of [10], and improves the reconstruction error performance of the column-by-column reconstruction.

It is obvious that the parallel CS without permutation has low computational complexity. However, its reconstruction error performance is not favorable. Usually, the joint sampling and joint reconstruction scheme has better error performance, but at the same time, it has much higher computational complexity at both encoder and decoder. Consequently, other

approaches have been studied which can achieve better reconstruction error performance than the parallel CS without permutation scheme and have lower computational complexity than the joint sampling and joint reconstruction scheme. For example, the block CS framework uses the block-by-block sampling and joint reconstruction, with low-complexity at sampling and high-complexity at reconstruction [7], [8]. For the Kronecker CS framework [6], separable sampling operators are used to achieve low complexity at the encoder, but the joint reconstruction still requires high-complexity at the decoder. Obviously, to achieve better performance than the parallel CS without permutation, both the block CS and the Kronecker CS frameworks require higher computational complexity of reconstruction, although their sampling operators have similar complexity to the column-by-column sampling operator. Our work has provided an alternative solution. We have shown that by introducing permutation into the parallel CS framework, the reconstruction error performance can indeed be improved, while the computational complexity remains the same as that of the parallel CS without permutation framework. It should be noted that there is a trade-off between the reconstruction error performance and the computational complexity. Compared to the parallel CS without permutation, the one with permutation results in a better reconstruction error performance and similar computational complexity. Compared to the joint reconstruction, the parallel CS with permutation may have worse reconstruction error performance, but enjoys much lower computational complexity.

To sum up, the main contribution of this work is the introduction of permutation for the parallel CS framework. The use of permutation can improve the error performance of the column-by-column reconstruction significantly. Although the parallel CS without permutation is a special case of the Kronecker CS, it has been shown in this paper that by using permutation, the parallel CS reconstruction can achieve similar error performance to the Kronecker CS, while it has much lower computational complexity. Compared to the block-based CS system [7], [8] with parallel reconstruction, we have shown that our scheme has better reconstruction error performance. Therefore, overall we conclude that the parallel CS with permutation framework provides an effective solution for the column-by-column sampling and reconstruction with low computational complexity.

Appendix: Proof of Proposition 1

Proof: Denote the j-th entry of the sparsity vector $\mathbf{s}$ as s_j, i.e., the sparsity level of the j-th column of $\mathbf{X}^s$ is s_j. Since $\mathbf{X}$ follows the (r_0, r_1, r_2, α)-layer model, the nonzero entries in $\mathbf{X}^s$ are all in layers of $\mathbf{X}^s$ whose indices range from r_0+1 to r_2. After performing the zigzag-scan-based permutation on $\mathbf{X}^s$, the number of nonzero entries in any column is not more than $u = \lceil (r_0+r_2+1)(r_2-r_0)/2N \rceil$. Therefore, $u \geq ||\mathbf{s}^\dagger||_\infty$. Let $l = \lceil (r_0+r_2+1)/2 \rceil$. Since $r_2 \leq \min\{M, N\}$ and $r_2 \geq 2r_1 - 3r_0 - 1$, we have $l \geq u$ and $l \geq r_1 - r_0$.

As a result, the probability that the zigzag-scan-based permutation of a 2D signal satisfying the (r_0, r_1, r_2, α)-layer model is an acceptable permutation can be expressed as

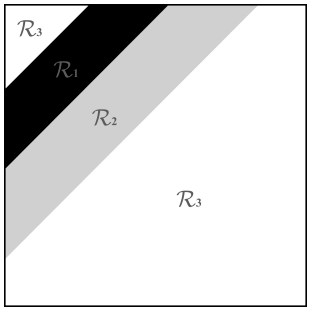


Fig. 11: Regions in the (r_0, r_1, r_2, α)-layer model.

$$\Pr\{\mathrm{P} \text{ is acceptable}\} = \Pr\{||\mathbf{s}||_\infty > ||\mathbf{s}^\dagger||_\infty\} \tag{6a}$$

$$= \sum_{t=1}^{r_2-r_0} \Pr\{||\mathbf{s}||_\infty = t, ||\mathbf{s}^\dagger||_\infty \leq t-1\} \tag{6b}$$

$$\geq \sum_{t=u+1}^{r_2-r_0} \Pr\{||\mathbf{s}||_\infty = t, ||\mathbf{s}^\dagger||_\infty \leq t-1\}$$

$$= \sum_{t=u+1}^{r_2-r_0} \Pr\{||\mathbf{s}||_\infty = t\} \tag{6c}$$

$$= \Pr\{||\mathbf{s}||_\infty \geq u+1\} = 1 - \Pr\{||\mathbf{s}||_\infty \leq u\}$$

$$\geq 1 - \Pr\{||\mathbf{s}||_\infty \leq l\}. \tag{6d}$$

To derive (6a), we use the fact that an acceptable permutation must result in $||\mathbf{s}^\dagger||_\infty < ||\mathbf{s}||_\infty$. For deriving (6b), we use the fact that the maximal sparsity level among columns of the best s-term approximation $\mathbf{X}^s$ is upper bounded by $(r_2 - r_0)$, i.e., $||\mathbf{s}||_\infty \leq r_2 - r_0$, which immediately follows from the (r_0, r_1, r_2, α)-layer model. For deriving (6c), we use the fact that $u \geq ||\mathbf{s}^\dagger||_\infty$. Finally, for deriving (6d), we use the fact that $u \leq l$. Based on (6d), we focus on the cumulative distribution function of $||\mathbf{s}||_\infty$.

Since the events that $s_j \leq l$ for different j's are independent with each other, we have

$$\Pr\{||\mathbf{s}||_\infty \leq l\} = \prod_{j=1}^{N} \Pr\{s_j \leq l\}. \tag{7}$$

Moreover, since the position (i, j) of an entry in $\mathbf{X}^s$ indicates the index m of the layer where the entry is located, i.e., $m = i + j - 1$, we can define three regions in $\mathbf{X}^s$:

$$\begin{aligned}
\mathcal{R}_1 &= \{(i,j) \in \mathbb{Z}^2 | r_0 + 1 \leq i + j - 1 \leq r_1\} \\
\mathcal{R}_2 &= \{(i,j) \in \mathbb{Z}^2 | r_1 + 1 \leq i + j - 1 \leq r_2\} \\
\mathcal{R}_3 &= \{(i,j) \in \mathbb{Z}^2 | 1 \leq i + j - 1 \leq r_0\} \cup \\
&\quad \{(i,j) \in \mathbb{Z}^2 | r_2 + 1 \leq i + j - 1 \leq M + N - 1\}.
\end{aligned}$$

These regions are separated by three transition layers, i.e., the r_0-th layer, the r_1-th layer and the r_2-th layer. These regions are shown in Fig. 11. Therefore, according to Definition 5, all entries of $\mathbf{X}^s$ are nonzero with probability 1 in region $\mathcal{R}_1$. In region $\mathcal{R}_3$, all entries of $\mathbf{X}^s$ are zero with probability 1. In region $\mathcal{R}_2$, the probability of an entry to be nonzero decreases exponentially with decay factor α as the layer index m increases.

$$
\begin{aligned}
&\Pr\{\text{P is acceptable}\}\\
&\geq 1-\prod_{j=1}^{r_2}\left\{\prod_{m=m_j}^{r_2}(1-p_m)+\sum_{k=k_j+1}^{\min\{l,r_2-r_0,r_2-j+1\}}\left[\prod_{m=m_j}^{r_2}(1-p_m)\right]\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\right\}\\
&=1-\prod_{j=1}^{r_2}\left[\prod_{m=m_j}^{r_2}(1-p_m)\right]\cdot\prod_{j=1}^{r_2}\left\{1+\sum_{k=k_j+1}^{\min\{l,r_2-r_0,r_2-j+1\}}\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\right\}\\
&\overset{(a)}{=}1-\left[\prod_{j=1}^{r_1}\prod_{m=r_1+1}^{r_2}(1-p_m)\cdot\prod_{j=r_1+1}^{r_2}\prod_{m=j}^{r_2}(1-p_m)\right]\cdot\prod_{j=1}^{r_2}\left\{1+\sum_{k=k_j+1}^{\min\{l,r_2-r_0,r_2-j+1\}}\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\right\}\\
&=1-\left[\prod_{m=r_1+1}^{r_2}(1-p_m)^{r_1}\cdot\prod_{m=r_1+1}^{r_2}(1-p_m)^{m-r_1}\right]\cdot\prod_{j=1}^{r_2}\left\{1+\sum_{k=k_j+1}^{\min\{l,r_2-r_0,r_2-j+1\}}\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\right\}\\
&=1-\left[\prod_{m=r_1+1}^{r_2}(1-p_m)^{m}\right]\cdot\prod_{j=1}^{r_2}\left\{1+\sum_{k=k_j+1}^{\min\{l,r_2-r_0,r_2-j+1\}}\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\right\}.
\end{aligned}
$$

For the j-th column of $\mathbf{X}^s$, if $r_2+1\leq j\leq N$, all entries of the column are in region $\mathcal{R}_3$ and thus are all zeros. Then we have $\Pr\{s_j\leq l\}=1$ since $l\geq r_1-r_0\geq 1$. According to (7), we have

$$\Pr\{||\mathbf{s}||_\infty\leq l\}=\prod_{j=1}^{r_2}\Pr\{s_j\leq l\}=\prod_{j=1}^{r_2}\sum_{k=0}^{l}\Pr\{s_j=k\}. \quad (8)$$

Consequently, we focus on the probability distribution of s_j for the first r_2 columns of $\mathbf{X}^s$, i.e., $\Pr\{s_j=k\}$ for all $0\leq k\leq l$ and $1\leq j\leq r_2$.

Let k_j denote the number of entries in the j-th column ($1\leq j\leq r_2$) of $\mathbf{X}^s$ that are in region $\mathcal{R}_1$, i.e.,

$$k_j=\begin{cases} r_1-r_0, & 1\leq j\leq r_0\\ r_1-j+1, & r_0+1\leq j\leq r_1\\ 0, & r_1+1\leq j\leq r_2.\end{cases} \quad (9)$$

Meanwhile, in the j-th column ($1\leq j\leq r_2$) of $\mathbf{X}^s$, $m_j=\max\{r_1+1,j\}$ and r_2 are the starting and ending layer indices of region $\mathcal{R}_2$, respectively.

In (8), for $1\leq j\leq r_2$, i.e., for the first r_2 columns of $\mathbf{X}^s$, we consider the following three cases depending on the value of k: 1) $k=k_j$; 2) $k_j+1\leq k\leq\min\{r_2-r_0,r_2-j+1\}$; and 3) $k\leq k_j-1$ or $k\geq\min\{r_2-r_0+1,r_2-j+2\}$.

For the first case, i.e., $k=k_j$, it can be seen that the event that $s_j=k$ happens when the entries of the j-th column of $\mathbf{X}^s$ that are in region $\mathcal{R}_2$ are all zeros. Therefore, we have

$$\Pr\{s_j=k\}=\prod_{m=m_j}^{r_2}(1-p_m). \quad (10)$$

For the second case, i.e., $k_j+1\leq k\leq\min\{r_2-r_0,r_2-j+1\}$, the event that $s_j=k$ means that the j-th column of $\mathbf{X}^s$ has $(k-k_j)$ nonzero entries in the region $\mathcal{R}_2$. Denote layer indices of these $(k-k_j)$ nonzero entries as $a_1,a_2,\cdots,a_{k-k_j}$ with $a_1<a_2<\cdots<a_{k-k_j}$. So $a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\overset{\triangle}{=}\{m_j,m_j+1,\cdots,r_2\}$. We have

$$
\begin{aligned}
&\Pr\{s_j=k\}\\
&=\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}p_{a_1}\cdots p_{a_{k-k_j}}\prod_{\substack{m=m_j\\ m\neq a_i\\ i=1,\cdots,k-k_j}}^{r_2}(1-p_m)\\
&=\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}\prod_{m=m_j}^{r_2}(1-p_m)\\
&=\left[\prod_{m=m_j}^{r_2}(1-p_m)\right]\sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j\\ a_1<a_2<\cdots<a_{k-k_j}}}\frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}. \quad (11)
\end{aligned}
$$

For the third case, i.e., $k\leq k_j-1$ or $k\geq\min\{r_2-r_0+1,r_2-j+2\}$, since $k_j\leq s_j\leq\min\{r_2-r_0,r_2-j+1\}$ for $1\leq j\leq r_2$, the event that $s_j=k$ never happens, i.e.,

$$\Pr\{s_j=k\}=0. \quad (12)$$

According to (6d), (8), (10)–(12) and the fact that $l\geq r_1-r_0\geq k_j$ (from (9)), we have the inequality shown on the top of this page, where equality (a) follows from the fact that $m_j=r_1+1$ for $1\leq j\leq r_1$ and $m_j=j$ for $r_1+1\leq j\leq r_2$. Using the facts that $l=\lceil(r_0+r_2+1)/2\rceil$ and $p_m=e^{-\alpha(m-r_0-1)}$ for $r_1+1\leq m\leq r_2$, we obtain (5). ■

REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[3] ——, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[4] J. Wu, F. Liu, L. C. Jiao, X. Wang, and B. Hou, "Multivariate compressive sensing for image reconstruction in the wavelet domain: Using scale mixture models," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3483–3494, Dec. 2011.

[5] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 449–452, June 2009.

[6] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 494–504, Feb. 2012.

[7] L. Gan, "Block compressed sensing of natural images," in *Proc. Int. Conf. Digital Signal Process.*, Cardiff, UK, July 2007, pp. 403–406.

[8] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. Int. Conf. Image Process.*, Cario, Egypt, Nov. 2009, pp. 3021–3024.

[9] L. Gan, T. T. Do, and T. D. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," in *Proc. 16th European Signal Process. Conf.*, Lausanne, Switzerland, Aug. 2008.
[10] G. Coluccia, S. K. Kuiteing, A. Abrardo, M. Barni, and E. Magli, "Progressive compressed sensing and reconstruction of multidimensional signals using hybrid transform/prediction sparsity model," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 340–352, Sep. 2012.
[11] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, July 2005.
[12] V. Stanković, L. Stanković, and S. Cheng, "Compressive sampling of binary images," in *Proc. Cong. on Image and Signal Process.*, Sanya, Hainan, China, May 2008, pp. 7–11.
[13] ——, "Compressive video sampling," in *Proc. 16th European Signal Process. Conf.*, Lausanne, Switzerland, Aug. 2008.
[14] Z. Liu, A. Y. Elezzabi, and H. V. Zhao, "Maximum frame rate video acquisition using adaptive compressed sesnsing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1704–1718, Nov. 2011.
[15] O. Taheri and S. A. Vorobyov, "Segmented compressed sampling for analog-to-information conversion: Method and performance analysis," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 554–572, Feb. 2011.
[16] M. F. Duarte, S. Jafarpour, and A. R. Calderbank, "Performance of the Delsarte-Goethals frame on clustered sparse vectors," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 1998–2008, Apr. 2013.
[17] H. Fang, S. A. Vorobyov, H. Jiang, and O. Taheri, "2D signal compression via parallel compressed sensing with permutations," in *Proc. 46th Annual Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, California, USA, Nov. 2012, pp. 1925–1929.
[18] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[19] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.
[20] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1060–1072, June 2012.
[21] C. Zhou, C. Xiong, R. Mao, and J. Gong, "Compressed sensing of images using nonuniform sampling," in *Proc. Int. Conf. Intell. Computation Technology and Automat.*, Shenzhen, China, Mar. 2011, pp. 483–486.

Hao Fang (S'12) received the B.Eng. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2011 and the M.Sc. degree in electrical and computer engineering from the University of Alberta, Edmonton, Alberta, Canada, in 2013.

He is currently working towards his Ph.D. degree at the University of Washington, Seattle, USA. His research interests include signal processing, speech/language processing, compressed sensing, and image/video processing.

Sergiy A. Vorobyov (M'02-SM'05) received the M.Sc. and Ph.D. degrees in systems and control from Kharkiv National University of Radio Electronics, Ukraine, in 1994 and 1997, respectively.

He is a Professor with the Department of Signal Processing and Acoustics, Aalto University, Finland and is currently on leave from the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He has been with the University of Alberta as Assistant Professor from 2006 to 2010, Associate Professor from 2010 to 2012, and Full Professor from 2012. Since his graduation, he also held various research and faculty positions at Kharkiv National University of Radio Electronics, Ukraine; the Institute of Physical and Chemical Research (RIKEN), Japan; McMaster University, Canada; Duisburg-Essen University and Darmstadt University of Technology, Germany; and the Joint Research Institute between Heriot-Watt University and Edinburgh University, U.K. He has also held short-term visiting positions at Technion, Haifa, Israel and Ilmenau University of Technology, Ilmenau, Germany. His research interests include statistical and array signal processing, applications of linear algebra, optimization, and game theory methods in signal processing and communications, estimation, detection and sampling theories, and cognitive systems.

Dr. Vorobyov is a recipient of the 2004 IEEE Signal Processing Society Best Paper Award, the 2007 Alberta Ingenuity New Faculty Award, the 2011 Carl Zeiss Award (Germany), the 2012 NSERC Discovery Accelerator Award, and other awards. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2006 to 2010 and for the IEEE SIGNAL PROCESSING LETTERS from 2007 to 2009. He was a member of the Sensor Array and Multi-Channel Signal Processing Committee of the IEEE Signal Processing Society from 2007 to 2012. He is a member of the Signal Processing for Communications and Networking Committee since 2010. He has served as the Track Chair for Asilomar 2011, Pacific Grove, CA, the Technical Co-Chair for IEEE CAMSAP 2011, Puerto Rico, and the Tutorial Chair for ISWCS 2013, Ilmenau, Germany.

Hai Jiang (M'07) received the B.Sc. and M.Sc. degrees in electronics engineering from Peking University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree (with an Outstanding Achievement in Graduate Studies Award) in electrical engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2006.

Since July 2007, he has been a faculty member with the University of Alberta, Edmonton, Alberta, Canada, where he is currently an Associate Professor at the Department of Electrical & Computer Engineering. His research interests include radio resource management, cognitive radio networking, and cross-layer design for wireless multimedia communications.

Dr. Jiang is an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the IEEE WIRELESS COMMUNICATIONS LETTERS. He served as a Co-Chair for the Wireless and Mobile Networking Symposium at the IEEE International Conference on Communications (ICC) in 2010. He received an Alberta Ingenuity New Faculty Award in 2008 and a Best Paper Award from the IEEE Global Telecommunications Conference (GLOBECOM) in 2008.

Omid Taheri (S'10) received the B.Sc. and M.Sc. degrees in electrical engineering from Isfahan University of Technology, Isfahan, Iran, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 2012.

His research interests are in digital signal processing with emphasis on compressive sampling, analog-to-information conversion, and sparse channel estimation.