University of Alberta

Parallel Sampling and Reconstruction with Permutation in Multidimensional Compressed Sensing

by

Hao Fang

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Signal and Image Processing

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2013

Abstract

The advent of compressed sensing provides a new way to sample and compress signals. In this thesis, a parallel compressed sensing architecture is proposed, which samples a two-dimensional reshaped multidimensional signal column by column using the same sensing matrix. Compared to architectures that sample a vector-reshaped multidimensional signal, the sampling device in the parallel compressed sensing architecture stores a smaller-sized sensing matrix and has lower computational complexity. Besides, the reconstruction of the multidimensional signal can be conducted in parallel, which reduces the computational complexity and time for reconstruction at the decoder side. In addition, when parallel sampling is not required but analog compressed sensing is desired, an alternative architecture proposed in this thesis, named parallel compressed sensing reconstruction architecture, can be used. In both proposed architectures, permutation is introduced and shown to enable the reduction of the required number of measurements for a given desired reconstruction error performance.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisors Dr. Sergiy A. Vorobyov and Dr. Hai Jiang for the continuous support of my M.Sc. study and resaerch, for their patience, motivation, enthusiasm and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Besides my supervisors, I would like to thank the rest of my thesis committee, Dr. Vicky Zhao and Dr. Ioanis Nikolaidis, for their insightful comments and questions.

I would also like to thank my labmates and friends, Saman Atapattu, Hao Cheng, Lei Dong, Jie Gao, Yichen Hao, Chunxing Jiang, Arash Khabbazibasmenj, Mahdi Shaghaghi, Omid Taheri, Lijie Huang, Maoling Wang, Qian Wang, Chunyan Wu, Yuan Yuan, Zhou Zhang and Yu Zheng for the valuable comments on my presentation and thesis and for their help during my stay in Edmonton.

Last but not the lease, I would like to thank my parents for their love and encouragement in all my pursuits.

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description	First use
ℓ_0	0-pseudonorm for finite-dimensional vector spaces	2
ℓ_1	1-norm for finite-dimensional vector spaces	2
$\mathbf{x}$	Signal of interest	4
L	Length of a vector	4
S	Sparsity level of a signal	4
K	Number of measurements	4
$\mathbf{y}$	Measurement vector of a signal	4
δ_t	t-restricted isometry constant	4
$\|\|\cdot\|\|_2$	ℓ_2-norm of a vector	5
$\zeta_{t,t'}$	(t,t')-restricted orthogonality constant	5
$\|\cdot\|$	Absolute value of a real number	5
$[\cdot]^T$	Transpose operator	5
$\|\|\cdot\|\|_1$	ℓ_1-norm of a vector	4
$[\cdot]^S$	Best S-term approximation of a signal	5
$\mathbf{\Psi}$	Sparsifying basis of a signal	6
$\boldsymbol{\theta}$	Projection of a signal onto the sparsifying basis	7
$\mathbf{\Phi}$	Measurement matrix	7
$\Pr\{\cdot\}$	Probability of an event	7
$\otimes$	Kronecker product	8
$[\cdot](i,j)$	The (i,j)-th element of a matrix	8
$\mathbf{s}$	Sparsity vector of a 2D signal	17
$[\cdot]_j$	The j-th element of a vector	17
$\|\|\cdot\|\|_\infty$	Chebyshev norm of a vector	17
$\mathrm{P}(\cdot)$	Permutation operator	19
$\mathrm{P}^{-1}(\cdot)$	Inverse permutation operator	19
$\lceil\cdot\rceil$	Ceiling function	21

List of Abbreviations

Abbreviation	Description	First use
2D	Two-dimensional	3
ADC	Analog-to-digital converter	1
BP	Basis pursuit	9
C-DMRC	Compressive distortion minimizing rate control	32
CCD	Charge-couple device	9
CMOS	Complementary metal-oxide-semiconductor	9
CS	Compressed sensing	2
DAC	Digital-to-analog converter	1
DCT	Discrete cosine transform	1
DCT2	Two-dimensional discrete cosine transform	28
DFT	Discrete Fourier transform	6
DMD	Digital micromirror device	2
DSP	Digital signal processing	1
DWT	Discrete wavelet transform	6
i.i.d.	Independent and identically distributed	7
JPEG	Joint photographic experts group	1
MMV	Multiple measurement vector	15
MPEG	Moving picture experts group	32
MSE	Mean square error	50
OMP	Orthogonal matching pursuit	5
PSNR	Peak signal-to-noise ratio	16
QCIF	Quarter common intermediate format	37
RIP	Restricted isometry property	4
TIFF	Tagged image file format	35

Chapter 1

Introduction

In digital signal processing (DSP), sampling is a fundamental step which converts signal from an analog form to a digital form. The conversion is usually achieved by an analog-to-digital converter (ADC), whose input is an analog signal and output is a digital signal. The corresponding inverse conversion, i.e., conversion from a digital signal to an analog signal, is often achieved by a digital-to-analog converter (DAC). Most ADC and DAC devices are designed based on the famous Nyquist-Shannon sampling theorem. In other words, in the ADC, the analog signal is sampled with the Nyquist rate, which is determined by the desired bandwidth of the reconstructed signal; in the DAC, the digital signal is converted to an analog signal by interpolation algorithm. Besides, quantization is usually done in the ADC, which converts the discrete-time signal to a digital signal. The word "digital" means that the signal takes only a discrete set of values. For example, a digital signal can be represented as 8 bit (256 levels), 16 bit (65536 levels), etc. The necessity of the ADC and DAC makes DSP more complex than analog signal processing. On the other hand, digital signal can be processed more easily and accepted by microprocessor-based devices, e.g., computers, smart phones, etc. Thus, DSP has been applied to many areas such as data compression, image processing, etc.

Typically, the raw digital data are redundant because it helps to achieve some acceptable reconstruction quality. Then to reduce resource usage such as data storage space and transmission capacity, data compression is introduced. A typical data compression technique for "natural" data like audio signals or images is transform coding. The joint photographic experts group (JPEG) standard is an application of transform coding to image compression [1]. The digital image signal acquired by the sampling devices is projected onto a basis, e.g., the discrete cosine transform (DCT) basis, where only a few significant coefficients are needed to preserve most energy/information of the signal. In this way, the signal can be

compressed by simply discarding other insignificant coefficients. By knowing the basis and the stored/transmitted coefficients, the digital signal can be reconstructed easily with a small error.

In traditional architectures for data compression mentioned above, the sampling and the compression processes are separated. Besides, the Nyquist rate sampling can be expensive to implement for wideband signals. So it is reasonable to ask whether there is a way that can either avoid the Nyquist rate sampling or combine the sampling and compression into one step while still achieving the same compression quality. The compression quality is determined by the error between the reconstructed digital signal and the digital signal acquired by the Nyquist rate sampling.

The advent of compressed sensing (CS) theory provides a solution to this problem. The core of CS theory demonstrates the possibility of recovering an analog signal from fewer samples than the Nyquist rate requires [2–4]. In the CS architecture, the signal is simultaneously compressed during the discretization, whereas in the traditional architecture mentioned above, the signal is compressed after the discretization. Accordingly, Nyquist rate sampling is avoided in the CS architecture. However, the reconstruction process of the CS architecture needs to solve an ℓ_0-pseudonorm minimization problem, which is an NP-hard problem. Although it is proved that solving an ℓ_1-norm minimization problem instead can be effective [5], the reconstruction process is usually more complex than that in the traditional architecture for data compression mentioned above (where the sampling process and the compression process are separated). Thus, CS architecture is preferred for less powerful sampling devices with more powerful reconstruction devices such as in wireless sensor networks.

1.1 Proposed research problem

In recent years, CS has been applied for sampling multidimensional signals such as image and video signals. Several imaging architectures based on CS have been proposed. For example, the single-pixel camera can acquire a group of samples of an image with only one optical sensor using different patterns of the digital micromirror device (DMD) array [6, 7]; the optically compressed image sensing uses random coded aperture to acquire random projections of an image [8].

An immediate way to apply CS to a multidimensional signal is sampling it as a "vector", as it is done in the single-pixel camera and the optically compressed image sensing. However, the sampling process then becomes more difficult as the length of the "vector"

increases since the sampling requires more storage and longer acquisition time. Besides, the complexity of the reconstruction process, which relies on solving an ℓ_1-norm minimization problem, increases significantly with the increase of the "vector" length. Sometimes, it is even impossible to use the CS architecture for multidimensional signal because of limited resources, e.g., processor capacity, storage, etc.

Therefore, this thesis is focused on studying the way to reduce the complexity of the sampling process and the reconstruction process in the CS architecture. To be specific, this thesis exploits the possibility of sampling and/or reconstructing all segments of a multidimensional signal in parallel by using the CS architecture. Furthermore, advantages brought by introducing permutation into the CS architecture are discussed in the thesis.

In summary, this thesis makes the following contributions. First, a parallel CS architecture is proposed, where a signal is divided into several segments and all segments are sampled and reconstructed in parallel. Second, properties of permutation are investigated when applied to the parallel CS architecture proposed in the thesis. Third, a group-scan-based, a zigzag-scan-based and a block-test-based permutations are introduced as specific examples of practically appealing permutations. As an example, the zigzag-scan-based permutation is shown to be an acceptable permutation with a large probability for two-dimensional (2D) signals satisfying an introduced layer model. The block-test-based permutation is discussed via simulation. Fourth, an application of the parallel CS architecture with permutation to video compression is discussed. Finally, as a complement, a new parallel CS reconstruction architecture is proposed, where a signal is sampled in centralized manner and reconstructed in parallel. Compared to the parallel CS architecture, the parallel CS reconstruction architecture integrates the permutation into the sampling process but sacrifices the ability of parallel sampling.

1.2 Organization of the thesis

Preliminary and background materials are presented in Chapter 2. In Chapter 3, a new parallel CS architecture is introduced and the properties of permutation when applied to the parallel CS architecture are analyzed. Then a new parallel CS reconstruction architecture is presented and discussed in Chapter 4. Chapter 5 concludes the thesis and discusses the future work that can be done based on this thesis.

Chapter 2

Preliminaries

In this chapter, some background on CS is given first. Traditional data compression technique is also presented here. The chapter is concluded with introduction of several existing CS based imaging architectures.

2.1 Background on CS

2.1.1 CS for sparse signals

An L-length signal $\mathbf{x} \in \mathbb{R}^L$ is called *S-sparse* if there are only S nonzero entries in $\mathbf{x}$. CS theory states that if $S \ll L$, i.e., the signal is sparse enough, then the information contained in an S-sparse signal $\mathbf{x} \in \mathbb{R}^L$ can be fully preserved with only $K (S \ll K \ll L)$ measurements, which form a measurement vector $\mathbf{y} \in \mathbb{R}^K$ [2, 4]. The measurement vector $\mathbf{y}$ is obtained by the help of a sensing matrix $\mathbf{A} \in \mathbb{R}^{K \times L}$, i.e.,

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \tag{2.1}$$

The S-sparse signal $\mathbf{x}$ can be reconstructed from the measurement vector $\mathbf{y}$ given in (2.1) by solving, for example, the following ℓ_1-norm minimization problem

$$\min_{\mathbf{z}} ||\mathbf{z}||_1 \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{A}\mathbf{z} \tag{2.2}$$

where $||\cdot||_1$ denotes the ℓ_1-norm of a vector.

The restricted isometry property (RIP) condition is a sufficient condition on $\mathbf{A}$ which guarantees the exact recoverability of an S-sparse signal $\mathbf{x}$ by solving (2.2). The RIP condition can be described in terms of the restricted isometry constants and the restricted orthogonality constants of $\mathbf{A}$, which are defined below.

Definition 1 [9]: *The t-restricted isometry constant δ_t of a given sensing matrix $\mathbf{A} \in \mathbb{R}^{K \times L}$*

is defined as

$$\delta_t := \min_{c \geq 0} c \qquad \text{s.t.} \qquad (1-c)||\mathbf{z}||_2^2 \leq ||\mathbf{A}\mathbf{z}||_2^2 \leq (1+c)||\mathbf{z}||_2^2 \tag{2.3}$$

for all S-sparse signals $\mathbf{z} \in \mathbb{R}^L$ with $S \leq t$. Here $||\cdot||_2$ denotes the ℓ_2-norm of a vector.

Definition 2 [9]: *The (t, t')-restricted orthogonality constant $\zeta_{t,t'}$ of a given sensing matrix $\mathbf{A} \in \mathbb{R}^{K \times L}$ is defined as*

$$\zeta_{t,t'} := \min_{c \geq 0} c \qquad \text{s.t.} \qquad |\mathbf{z}^T \mathbf{A}^T \mathbf{A} \mathbf{z}'| \leq c||\mathbf{z}||_2||\mathbf{z}'||_2 \tag{2.4}$$

for all S-sparse signals $\mathbf{z} \in \mathbb{R}^L$ with $S \leq t$ and all S'-sparse signals $\mathbf{z}' \in \mathbb{R}^L$ with $S' \leq t'$ where $t + t' \leq L$. Here $|\cdot|$ denotes the absolute value of a real number and the superscript T stands for the transpose.

Many works have been devoted to deriving the RIP condition on $\mathbf{A}$ [5, 9–15]. For example, the RIP condition derived in [9] is

$$\delta_S + \zeta_{S,S} + \zeta_{S,2S} < 1. \tag{2.5}$$

The RIP condition derived in [11], which merely depends on the $2S$-restricted isometry constant, is $\delta_{2S} \leq \sqrt{2} - 1$. Meanwhile, an S-sparse signal $\mathbf{x}$ can also be exactly recovered from the measurement vector $\mathbf{y}$ given in (2.1) using matching pursuit algorithm, or better orthogonal matching pursuit (OMP) [16]. Similar RIP conditions for using OMP algorithm to recover the sparse signal have also been derived [16–20]. A review of these conditions can be found in [20]. In this thesis, only the RIP conditions related to the recovery problem (2.2) are considered.

2.1.2 CS for compressible signals

In Section 2.1.1, strictly sparse signals have been considered. However, most practical signals are not strictly sparse. Rather, they can be regarded as *compressible*, i.e., they have only a few large[1] elements. A compressible signal $\mathbf{x}$ can be approximated using its *best S-term approximation*, denoted as $\mathbf{x}^S$. The best S-term approximate $\mathbf{x}^S$ is an S-sparse signal generated by keeping the S largest entries in $\mathbf{x}$ and changing the remaining entries to zeros. The best S-term approximation is regarded as an optimal approximation using only S elements since it preserves most energy of the compressible signal. However, such approximation requires knowledge about the values and locations of all elements in $\mathbf{x}$.

[1] In this thesis, when saying that a value is large or small, it means the magnitude of the value is large or small.

CS is still effective when applied to compressible signals. If the sensing matrix $\mathbf{A}$ obeys the RIP condition, the reconstruction via solving (2.2) is nearly as good as that using the best S-term approximation, as shown in the following Lemma 1 [21].

Lemma 1: *Assume that* $\delta_{2S} < \sqrt{2} - 1$ *for a sensing matrix* $\mathbf{A}$*. Then for a signal* $\mathbf{x}$*, the solution* $\mathbf{x}^*$ *to (2.2) obeys*

$$||\mathbf{x}^* - \mathbf{x}||_1 \leq G \cdot ||\mathbf{x} - \mathbf{x}^S||_1 \quad \textit{and}$$

$$||\mathbf{x}^* - \mathbf{x}||_2 \leq G' \cdot ||\mathbf{x} - \mathbf{x}^S||_1/\sqrt{S} \tag{2.6}$$

for some constants G *and* G'*.*

As mentioned before, $\delta_{2S} < \sqrt{2} - 1$ is the RIP condition. Sometimes, this RIP condition is called *RIP of order* S. According to Lemma 1, an S-sparse signal can be exactly reconstructed via solving (2.2) if this RIP condition is held. For a compressible signal $\mathbf{x}$, if the RIP condition is held for the sensing matrix $\mathbf{A}$, the reconstruction via solving (2.2) 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, if (2.2) is solved for reconstruction, there is no need to know the values and locations of all elements in $\mathbf{x}$, whereas such knowledge is required if $\mathbf{x}^S$ is used to approximate $\mathbf{x}$.

2.1.3 CS for general signals

Previous subsections discussed the situations when a signal itself is either sparse or compressible. This subsection discusses the situation when the projection of a signal onto some orthonormal basis is either sparse or compressible. Without loss of generality, only the sparse case is considered since the compressible case can be similarly analyzed. The orthonormal basis where the projection of a signal is sparse is called *sparsifying basis*. For example, the sparsifying basis of most audio signals can be the discrete Fourier transform (DFT) basis since the audio signals are usually sparse in frequency domain; the sparsifying basis of 2D piecewise smooth image signals can be either the DCT basis or the discrete wavelet transform (DWT) basis. One way to apply CS to such signal is to pre-project the signal onto its corresponding sparsifying basis and sample the projection which is sparse. Then for reconstruction, the reconstructed sparse projection is projected back onto the original basis to obtain the reconstructed signal. Obviously, it is not the preferred way, and we will elaborate about it more in the thesis.

Denote the sparsifying basis of a signal $\mathbf{x} \in \mathbb{R}^L$ as $\mathbf{\Psi} \in \mathbb{R}^{L \times L}$ and its corresponding

projection as $\boldsymbol{\theta} \in \mathbb{R}^L$, i.e.,

$$\mathbf{x} = \boldsymbol{\Psi}\boldsymbol{\theta}. \tag{2.7}$$

Consider the following two sampling schemes. One scheme uses a sensing matrix $\mathbf{A} \in \mathbb{R}^{K\times L}$ to sample $\boldsymbol{\theta}$, which is described above, i.e., $\mathbf{y} = \mathbf{A}\boldsymbol{\theta}$. The other one uses a measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{K\times L}$ to sample $\mathbf{x}$, i.e., $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$. If $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$, it is obvious that these two schemes are equivalent. Therefore, the signal $\mathbf{x}$ can be directly sampled using the measurement matrix $\boldsymbol{\Phi}$ without being projected onto the sparsifying basis $\boldsymbol{\Psi}$. When reconstructing the signal $\mathbf{x}$, $\boldsymbol{\theta}$ is firstly recovered and then projected back onto the original basis, which gives the reconstructed signal.

2.1.4 Sensing matrix and measurement matrix

According to the analysis in Subsection 2.1.3, the sensing matrix corresponds to the S-sparse signal, whereas the measurement matrix corresponds to the original signal whose projection on the sparsifying basis is an S-sparse signal. Obviously, if the original signal is S-sparse itself, i.e., the sparsifying basis is the identity basis, then the measurement and the sensing matrices are the same.

As mentioned in Subsection 2.1.1, the RIP condition is a condition on the sensing matrix. Thus, constructing a sensing matrix $\mathbf{A} \in \mathbb{R}^{K\times L}$ that satisfies the RIP condition is a branch of CS theory. The most popular matrix that satisfies the RIP condition is the Gaussian random matrix. In [9], it is shown that if the entries of $\mathbf{A}$ are independent and identically distributed (i.i.d.) Gaussian with zero mean and variance $1/K$ and the sparsity ratio S/L is small enough, then the RIP condition (2.5) holds with an overwhelming probability. Meanwhile, the following theorem derived in [22] gives a more general class of matrices that obey the RIP condition.

Theorem 1 [22]: *Assume that the probability distribution that generates entries of* $\boldsymbol{\Phi} \in \mathbb{R}^{K\times L}$ *satisfies the concentration inequality, i.e.,*

$$\Pr\left\{|||\boldsymbol{\Phi}\mathbf{z}||_2^2 - ||\mathbf{z}||_2^2| \geq \epsilon||\mathbf{z}||_2^2\right\} \leq 2e^{-Kc_0(\epsilon)} \tag{2.8}$$

for any $\mathbf{z} \in \mathbb{R}^L$ *and* $0 < \epsilon < 1$*, where* $\Pr\{\cdot\}$ *denotes the probability of an event,* $c_0(\epsilon)$ *is a positive constant that depends only on* ϵ*. For any orthonormal basis* $\boldsymbol{\Psi} \in \mathbb{R}^{L\times L}$ *and a given* $\epsilon \in (0, 1)$*, the* S*-restricted isometry constant of* $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ *is equal to* $\epsilon/2$ *with probability no less than* $1 - 2e^{-c_2(\epsilon)K}$ *if*

$$K \geq c_1(\epsilon)S\log(L/S) \tag{2.9}$$

where positive constants $c_1(\epsilon), c_2(\epsilon)$ depend only on ϵ.

Examples of distributions that satisfy the concentration inequality (2.8) are Gaussian distribution and Bernoulli distribution [22]. In addition, Theorem 1 points out the universality of the measurement matrix $\mathbf{\Phi}$ with respect to the sparsifying basis $\mathbf{\Psi}$. In other words, by constructing a measurement matrix $\mathbf{\Phi}$ satisfying (2.8) and (2.9), the corresponding sensing matrix $\mathbf{A}$ with respect to any orthonormal basis $\mathbf{\Psi}$ is guaranteed to obey the RIP condition. It should be noted that the RIP condition used in Theorem 1 depends only on the restricted isometry constants, whereas the RIP condition used in [9] depends on both the restricted isometry constants and the restricted orthogonality constants.

2.1.5 CS for multidimensional signals

One way to apply CS to multidimensional signals is by using the Kronecker product of matrices, named *Kronecker CS* [23]. The Kronecker product of two matrices $\mathbf{U} \in \mathbb{R}^{P\times Q}$ and $\mathbf{V} \in \mathbb{R}^{R\times S}$ is defined as

$$\mathbf{U}\otimes\mathbf{V} := \begin{bmatrix} \mathbf{U}(1,1)\mathbf{V} & \cdots & \mathbf{U}(1,Q)\mathbf{V} \\ \vdots & \ddots & \vdots \\ \mathbf{U}(P,1)\mathbf{V} & \cdots & \mathbf{U}(P,Q)\mathbf{V} \end{bmatrix} \tag{2.10}$$

where $\mathbf{U}(i,j)$ denotes the (i,j)-th element of $\mathbf{U}$.

Given a multidimensional signal $\mathbf{x} \in \mathbb{R}^{L_1\times L_2\times\cdots\times L_d}$, its vector-reshaped representation is $\bar{\mathbf{x}} \in \mathbb{R}^{\bar{L}}$ where $\bar{L} = \prod_{i=1}^{d} L_i$. Assuming that the sparsifying basis of the i-th dimension of $\mathbf{x}$ is $\mathbf{\Psi}_i \in \mathbb{R}^{L_i\times L_i}$, the sparsifying basis for the multidimensional signal $\bar{\mathbf{x}}$ can be obtained using Kronecker products as $\bar{\mathbf{\Psi}} = \mathbf{\Psi}_1\otimes\mathbf{\Psi}_2\otimes\cdots\otimes\mathbf{\Psi}_d \in \mathbb{R}^{\bar{L}\times\bar{L}}$. Then the projection of $\bar{\mathbf{x}}$ onto $\bar{\mathbf{\Psi}}$ is an $\bar{S}$-sparse signal $\bar{\boldsymbol{\theta}} \in \mathbb{R}^{\bar{L}}$, i.e., $\bar{\mathbf{x}} = \bar{\mathbf{\Psi}}\bar{\boldsymbol{\theta}}$, and there are only $\bar{S} \ll \bar{L}$ nonzero entries in $\bar{\boldsymbol{\theta}}$. Let $\mathbf{\Phi}_i$ and $\mathbf{A}_i$ be respectively the measurement and sensing matrices corresponding to the i-th dimension of $\mathbf{x}$, and let K_i be the number of rows in $\mathbf{\Phi}_i$ and $\mathbf{A}_i$. Then the corresponding measurement matrix $\bar{\mathbf{\Phi}} \in \mathbb{R}^{\bar{K}\times\bar{L}}$ and sensing matrix $\bar{\mathbf{A}} \in \mathbb{R}^{\bar{K}\times\bar{L}}$ for $\bar{\mathbf{x}}$ can be defined respectively as $\bar{\mathbf{\Phi}} = \mathbf{\Phi}_1\otimes\mathbf{\Phi}_2\otimes\cdots\otimes\mathbf{\Phi}_d$ and $\bar{\mathbf{A}} = \mathbf{A}_1\otimes\mathbf{A}_2\otimes\cdots\otimes\mathbf{A}_d$, and the total number of rows in $\bar{\mathbf{\Phi}}$ and $\bar{\mathbf{A}}_i$ is $\bar{K} = \sum_{i=1}^{d} K_i$. Therefore, the measurement vector $\bar{\mathbf{y}} \in \mathbb{R}^{\bar{K}}$ can be expressed as

$$\bar{\mathbf{y}} = \bar{\mathbf{\Phi}}\bar{\mathbf{x}} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}. \tag{2.11}$$

According to the analysis in Subsection 2.1.1, if $\bar{\mathbf{A}}$ satisfies some RIP condition, $\bar{\boldsymbol{\theta}}$ and $\bar{\mathbf{x}}$ can be recovered via solving the ℓ_1-norm minimization problem.

Here compressible signal is not considered since it can be analyzed similarly.

2.1.6 Reconstruction algorithms

To solve (2.2), several algorithms have been developed, including interior point methods [24], projected gradient methods [25], and iterative thresholding [26]. The computational complexity of the interior point method such as the basis pursuit (BP) algorithm is $\mathcal{O}(L^3)$, where L is the length of the vector $\mathbf{z}$ in (2.2) [27].

Another way to recover the sparse signal from the measurement vector is to use greedy algorithms, which iteratively approximate the projection and the support of the original signal. Greedy algorithms are very fast and easy to implement. OMP is the most popular greedy algorithm, which was introduced in [28]. Other examples of greedy algorithms are stagewise OMP [29], regularized OMP [30], and compressive sampling matching pursuit [31].

2.2 Traditional data compression technique

The procedure of traditional data compression technique, especially for conventional images or videos, can be briefly described as "sampling, processing, keeping the important information and throwing away the rest". This compression technique typically exploits a priori information about the data, e.g., an L-pixel image data can be well approximated as a sparse linear combination of $S \ll L$ wavelets or frequency components. Such compression technique is similar to representing an L-length signal with its best S-term approximation.

JPEG standard for image compression is a typical example of such compression technique [1]. Given a 2D digital image acquired by the charge-couple device (CCD) array or the complementary metal-oxide-semiconductor (CMOS) array of size $M \times N$, it is divided into several 8×8 blocks. Corresponding 64 DCT coefficients of each block are calculated and quantized. The quantized DCT coefficients are then ordered into the "zigzag" sequence via a zigzag scan, as shown in Fig. 2.1. The zigzag scan results in a "zigzag" sequence where low-frequency coefficients are placed before high-frequency components. This ordering helps to facilitate entropy coding since the low-frequency quantized coefficients are more likely to be nonzero, whereas the high-frequency quantized coefficients are more likely to be zero. Then, a simple way to compress data is to keep the first few coefficients in the "zigzag" sequence and discard the rest which are likely to be zero.

2.3 CS based imaging architectures

The image compression technique presented in Section 2.2 has two major potential disadvantages. First, it can be expensive to acquire large amounts of raw image data, particularly

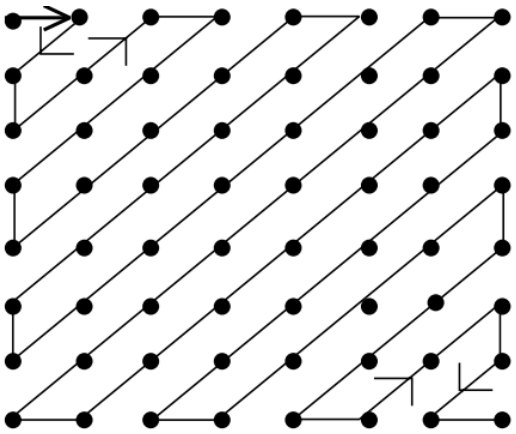


Fig. 2.1: The zigzag scan order of an 8×8 block.

at wavelengths where CCD and CMOS sensing technologies do not work well. Second, it can be computationally demanding to compress large amounts of raw image data.

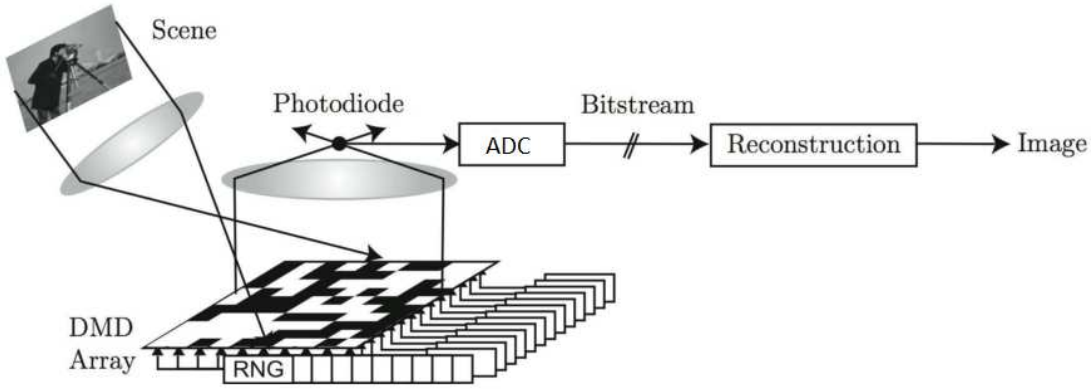


Fig. 2.2: The single-pixel camera architecture [6].

To address these two drawbacks, a new imaging architecture based on the CS theory has been proposed in [6], named *single-pixel camera*. As shown in Fig. 2.2, in the single-pixel camera architecture, an L-pixel image is formed on the plane of the DMD array of size L via a biconvex lens and it is reflected according to the DMD array pattern. The reflected light is then focused via the second biconvex lens onto the photodiode, which outputs a voltage that corresponds to a measurement of the image. The DMD array is controlled by the random number generators. Each different pattern of the DMD array acts as one row in the measurement matrix. By producing $K \ll L$ different patterns of the DMD array, K measurements of the image can be acquired.

The single-pixel camera collects K measurements of an image without first collecting

the L pixels of the image. Besides, there is only a single photodiode, which potentially enables imaging at new wavelengths that are too expensive to measure using CCD or CMOS technology. Other features of the single-pixel camera include (1) the universal measurement bases which are incoherent with arbitrary sparsifying bases, (2) encrypted measurements which are tied to a random seed that can be kept secure, and (3) scalable progressive reconstruction which yields better quality as more measurements are obtained.

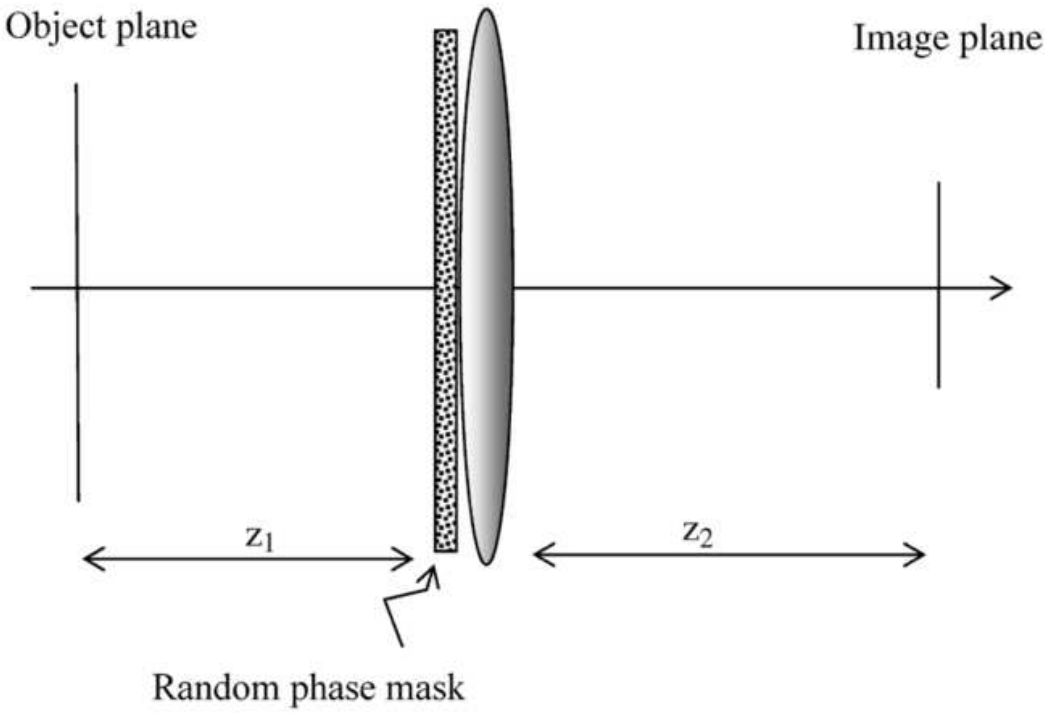


Fig. 2.3: One possible optical setup to implement the imaging using random coded aperture [8].

Another CS based imaging architecture is proposed in [8]. In this architecture, the measurement matrix is implemented by employing random aperture coding. One possible optical setup for such architecture is shown in Fig. 2.3. An L-pixel image is placed at a distance of z_1 from the lens. The image plane, which is the CCD array of size $K \ll L$, is placed at a distance of z_2 from the lens on the other side. A random Gaussian phase mask is attached to the lens, which randomly encodes the aperture. The scattered light from the random phase mask is collected by the lens and reaches the CCD array where a K-pixel image is formed.

The random coded aperture architecture and the single-pixel camera have their own advantages and disadvantages. The random coded aperture architecture requires only a

single shot, whereas the single-pixel camera requires successive random exposures to be taken. A CCD array of size K is needed in the random coded aperture architecture, whereas only one photodiode is needed in the single-pixel camera. Nevertheless, they both require fewer optical sensors than traditional imaging devices.

Chapter 3

Parallel Compressed Sensing with Permutation for 2D Signals[1]

Recently, the research interest in applying CS to sample multidimensional signals has increased significantly. 2D signals such as images and video frames are typical examples of multidimensional signals. A branch of CS theory, named *compressive imaging*, is introduced in [6], where the *single-pixel camera* is proposed. The single-pixel camera acquires a group of measurements of an image using different patterns of the DMD array, without collecting the pixels. Mathematically, each pattern of the DMD array plays the role of a row in the measurement matrix $\mathbf{\Phi}$, and the image is viewed as a vector. Consequently, the size of the DMD array is the same as the expected number of pixels in the image. The research is extended to sample color images by combining the Bayer color filter and the DMD array [34]. Furthermore, for reconstruction, the architecture proposed in [34] employs joint sparsity models to exploit the correlation among different color channels. However, as the expected number of pixels in the image increases, the number of columns and the required number of rows in the measurement matrix $\mathbf{\Phi}$ also increase. In other words, both the size and the required number of patterns of the DMD array increase. Therefore, the implementation cost and the complexity of the encoder increase significantly as well. For example, the storage of $\mathbf{\Phi}$ and computational complexity for acquiring measurements are unwieldy for any normal size images. A general framework for sampling multidimensional signals, named *Kronecker CS*, is proposed in [23]. In Kronecker CS, a multidimensional signal is vectorized and then sampled using a measurement matrix which is the Kronecker product of several smaller-sized measurement matrices that correspond to the measurement processes

[1]Some preliminary results in this chapter have been presented at Asilomar 2012, Pacific Grove, California, USA [32]. A version of this chapter has been submitted to IEEE Trans. on Signal Process. for publication [33].

for different portions of the multidimensional signal[2]. After finding such Kronecker product, the resulting measurement matrix clearly has a very large size. Thus, the problem related to the storage of $\mathbf{\Phi}$ and computational complexity for acquiring measurements arises in the Kronecker CS framework.

To address the above problem, instead of storing the Kronecker product of several measurement matrices, all portions of the multidimensional signal can be sampled sequentially using corresponding smaller-sized measurement matrices. Then the encoder needs to store only the smaller-sized measurement matrices. Using this approach, a separable sensing operator is designed for compressive imaging in [35], where an imaging operator (the measurement matrix for the whole image) can be separated into two dimensions. The separable sensing operator design significantly reduces the complexity of implementation, storage, and usage of the imaging operator. Another solution to the problem of storage and computational complexity is the block CS of [36]. The idea is to divide a 2D signal into smaller blocks and sample individual vectorized blocks, whereas all blocks need to be reconstructed as a whole. The block CS of [36] uses a block-diagonal measurement matrix for sampling the vectorized signal $\mathbf{x}$. 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 [37]. 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 [38] using the block Hadamard ensemble, which can be easily implemented in the optical domain.

The use of fast algorithms for computing measurements by taking advantage of the measurement matrix's structure is another way to address the problem of storage and computational complexity at the encoder side. For example, in [5] and [39], a scrambled Fourier ensemble is used as the sensing matrix $\mathbf{A}$ and the wavelet basis is used as the orthonormal basis $\mathbf{\Psi}$ in which the image projection is sparse. Thus, the sampling process can be implemented efficiently by first transforming a 2D image into the wavelet domain and then applying $\mathbf{A}$ to the wavelet coefficients by means of the fast Fourier transform. The sparsity structure of the multidimensional signal can be employed as well to reduce the storage and computational complexity at the encoder side. It is proposed in [40] to decompose the wavelet coefficients of a 2D signal into sparse components and dense components, and apply CS only to sparse components using a smaller-sized sensing matrix. In [41] and [42], the

[2]For example, for a 2D signal, the smaller-sized measurement matrices can correspond to the measurement processes for rows and for columns of the 2D signal, respectively.

statistical structure of the wavelet coefficients of a 2D image is considered for CS reconstruction of the image by using a scale mixture model. It is shown that fewer measurements are required in order to achieve a given reconstruction error performance. In addition, the scheme in [41] suggests to rearrange the wavelet coefficients into a new 2D matrix and sample each column of the matrix using the same sensing matrix $\mathbf{A}$ of a smaller size.

All above research works focus on the encoder side, aiming at reducing the implementation cost and the storage and computational complexity of the encoder. Joint reconstruction is employed in these schemes, and thus, the complexity at the decoder side is still high.

Taking into account the implementation cost and the storage and computational complexity of the decoder, a block-based CS architecture can be employed in video compression where all blocks can be sampled and reconstructed independently. In [43] and [44], it is proposed to apply CS only to sparse blocks found by a block classification scheme that considers the difference of sparsity levels among different blocks in an image or a video frame. Another block classification scheme based on inter-frame correlation is proposed in [45].

In this chapter, a parallel CS architecture is developed. A multidimensional sparse signal is considered, i.e., the signal is assumed to be sparse in the identity basis. 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 architecture that samples the vectorized signal. Furthermore, both sampling and reconstruction can be conducted for individual columns in parallel. Note that several works use a similar column-by-column sampling setting at the encoder side, e.g., the aforementioned scheme of [41]. The focus of [41] is on studying the scale mixture models used in CS for image reconstruction. The signal considered in [41] is the matrix of wavelet coefficients.

Another example is the multiple measurement vectors (MMV) model of [46], which considers a group of signals that share the same sparsity profile. In the MMV model, signals from a group of signals are sampled using the same dictionary, which is analogous to the sensing matrix in CS, while the group of signals can be viewed as a virtual 2D signal. Joint reconstruction is then used for the MMV model. Compared to the aforementioned works [41] and [46], which consider some specific sparse signals and require joint reconstruction at the decoder side, we address a more general setting at the encoder side and develop a parallel reconstruction at the decoder side. Moreover, we derive some analytical results related to the parallel CS scheme. Although joint reconstruction of multiple vectors, e.g., reconstruction of multiple vectors via sum-of-norm minimization, can bring some benefits

[47], it is shown that the uniform-recovery rate in the sum-of-norm minimization case cannot exceed that for the case of individual reconstruction of each vector [48]. Besides, there are problems that cannot be solved by the joint recovery via sum-of-norm minimization, but can be solved by reconstructing each vector individually and independently [48].

In the parallel CS architecture proposed in this chapter, a 2D signal may be permuted before it is sampled. It is because the permutation may provide benefits, for example, in computation and storage. 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 [49], 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 [50], it is shown that if nonzero entries of a sparse signal are clustered, the deterministic Delsarte-Goethals frame used as sensing matrix does not work. Thus, it is proposed to apply permutations to the signal in order to avoid clustered nonzero entries. In this chapter, the goal for applying permutations is different. Specifically, the parallel CS architecture considers sensing matrices that satisfy the RIP, and permutation is applied to 2D-reshaped signal aiming at ensuring that all columns of such signal have similar sparsity levels. 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. Thus, the storage and computational complexity can be further reduced. In this chapter, a group-scan-based permutation is introduced for 2D signals which can be divided into a number of groups with elements 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. Another example of the group-scan-based permutation, named block-test-based permutation, is introduced as well. A video compression scheme based on the parallel CS architecture with the zigzag-scan-based permutation and/or the block-test-based permutation is investigated. It improves the peak signal-to-noise ratio (PSNR) of reconstructed frames compared to the parallel CS architecture without permutation. This demonstrates the effectiveness of the zigzag-scan-based permutation and the block-test-based permutation in image and video compression.

In summary, the work in this chapter makes four main contributions. First, we propose a parallel CS scheme, which reduces the required size of sensing matrix and can be conducted

at both the encoding and decoding sides in a parallel (column-by-column) manner. Second, we investigate properties of permutations when applied to parallel CS. Third, we introduce a group-scan-based permutation, a zigzag-scan-based permutation and a block-test-based permutation. As an example 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, an application of the proposed parallel CS with the zigzag-scan-based permutation to video compression in wireless multimedia sensor networks is discussed.

The remainder of this chapter is organized as follows. Subsection 3.1 introduces the parallel CS scheme. Permutations are discussed in Subsection 3.2. Subsection 3.3 describes the video compression scheme that employs parallel CS with the zigzag-scan-based permutation in application to wireless multimedia sensor networks. Simulation results are given in Subsection 3.4. Finally, Subsection 3.5 concludes this chapter.

3.1 Parallel CS

Any multidimensional sparse signal can be rearranged into a 2D matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$. A multidimensional signal and the corresponding 2D matrix $\mathbf{X}$ are called to be S-sparse or to have sparsity level S if $\mathbf{X}$ has only S nonzero entries. The sparsity level of $\mathbf{X}$ can also be denoted in 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 $\mathbf{X}$ and $||\mathbf{s}||_1 = S$. In other words, the j-th column of $\mathbf{X}$ has only s_j nonzero entries.

In terms of the 2D signal $\mathbf{X}$, the parallel CS architecture consists of sampling each column of $\mathbf{X}$ by the same sensing matrix $\mathbf{A}$ and reconstructing these columns individually and in parallel. In this chapter, for presentation simplicity, only 2D signals are considered, i.e., rearrangement of a multidimensional signal into a 2D matrix is done in advance.

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

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., $\delta_{2||\mathbf{s}||_\infty} < \sqrt{2} - 1$, then $\mathbf{X}$ can be exactly reconstructed by the parallel CS architecture. Here $|| \cdot ||_\infty$ stands for the Chebyshev norm of a vector.*[3]

Proof. The proof follows the same steps as the proof for the following Lemma 3, and therefore, is omitted here for brevity. □

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

For a 2D compressible signal $\mathbf{X}$, the following lemma gives a sufficient condition that the reconstruction error in the parallel CS architecture 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 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 in the parallel CS architecture obeys*

$$||\hat{\mathbf{X}} - \mathbf{X}||_1 \le G \cdot ||\mathbf{X} - \mathbf{X}^S||_1 \quad \textit{and}$$
$$||\hat{\mathbf{X}} - \mathbf{X}||_2 \le G' \cdot ||\mathbf{X} - \mathbf{X}^S||_1 \tag{3.1}$$

where G and G' are finite constants.

Proof. Note that for all $1 \le j \le N$, $||\mathbf{s}||_\infty \ge s_j$. Then according to the definition of t-restricted isometry constant, we have $\delta_{2s_j} \le \delta_{2||\mathbf{s}||_\infty} < \sqrt{2}-1$. In turns, according to (2.6), the following inequalities hold for every j

$$||\hat{\mathbf{x}}_j - \mathbf{x}_j||_1 \le G_j \cdot ||\mathbf{x}_j - \mathbf{x}_j^S||_1$$

and

$$||\hat{\mathbf{x}}_j - \mathbf{x}_j||_2 \le G'_j \cdot ||\mathbf{x}_j - \mathbf{x}_j^S||_1 \cdot s_j^{-1/2}$$

where $\hat{\mathbf{x}}_j$ and $\mathbf{x}_j^S$ denote the j-th column of $\hat{\mathbf{X}}$ and the j-th column of $\mathbf{X}^S$, respectively, and G_j and G'_j are finite constants. Therefore, by choosing $G = \max_j\{G_j\}$ and $G' = \max_j\{G'_j\}$, it can be obtained that

$$||\hat{\mathbf{X}} - \mathbf{X}||_1 = \sum_{i=1}^{N} ||\hat{\mathbf{x}}_j - \mathbf{x}_j||_1 \le G \cdot \sum_{j=1}^{N} ||\mathbf{x}_j - \mathbf{x}_j^S||_1 = G \cdot ||\mathbf{X} - \mathbf{X}^S||_1$$

and

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

This completes the proof. □

To sum up, in the parallel CS architecture, the RIP condition for the sensing matrix $\mathbf{A}$ for a given reconstruction quality is related to the Chebyshev norm of the sparsity vector, i.e., $||\mathbf{s}||_\infty$. In Subsection 3.2.1, it will be shown that the RIP condition can be relaxed by performing a so-called acceptable permutation before parallel sampling the signal.

3.2 Permutation

When the parallel CS architecture 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. 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 $\mathbf{X}$ all columns of the best S-term approximation of the newly formed 2D signal would share similar sparsity levels.

Let $\mathrm{P}(\cdot)$ be a permutation operator which maps a matrix into another matrix by permuting its elements 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 performed before sampling, the parallel sampling process can be described as follows

$$\mathbf{y}_j = \mathbf{A}\mathbf{x}_j^\dagger \tag{3.2}$$

where $\mathbf{x}_j^\dagger$ is the j-th column of $\mathbf{X}^\dagger$, and $\mathbf{y}_j$ is the measurement vector of $\mathbf{x}_j^\dagger$. The equation (3.2) can be rewritten in the matrix form as

$$\mathbf{Y} = \mathbf{A}\mathbf{X}^\dagger = \mathbf{A}\mathrm{P}(\mathbf{X}) \tag{3.3}$$

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 reconstructed 2D signal $\hat{\mathbf{X}}$, i.e.,

$$\hat{\mathbf{X}} = \mathrm{P}^{-1}(\hat{\mathbf{X}}^\dagger). \tag{3.4}$$

3.2.1 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-

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

[5]In this chapter, when we say "permute", it means exchanging entries in a 2D matrix, while not changing the dimension of the matrix.

based CS employed in [36], [37], and [44] is a special case of the parallel CS architecture, 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 [44] by employing a classification scheme to identify sparse blocks and dense blocks and then applying CS only to the sparse blocks. In the parallel CS architecture, 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 best S-term approximation of the resulted 2D signal) are sparse. Thus, the classification step which is necessary in [44] can be avoided.

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

$$K \geq C \cdot ||\mathbf{s}||_\infty \log\left(M/||\mathbf{s}||_\infty\right) \tag{3.5}$$

for some constant C, then it will satisfy the RIP of order $||\mathbf{s}||_\infty$ [9]. Thus, according to Lemma 3, the signal $\hat{\mathbf{X}}$ reconstructed in the parallel CS architecture obeys (3.1).

Definition 3: *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* $\mathrm{P}(\cdot)$ *is called acceptable for* $\mathbf{X}$ *if the Chebyshev norm of the sparsity vector of the best* S*-term approximation of the permuted 2D signal* $\mathrm{P}(\mathbf{X})$ *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 $\mathbf{s}^{\dagger}$ (then we have $||\mathbf{s}^{\dagger}||_1 = S$). Consider that $M \gg ||\mathbf{s}||_\infty$ as well as $M \gg ||\mathbf{s}^{\dagger}||_\infty$, 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 in the parallel CS architecture with an acceptable permutation, the lower bound of K in (3.5) is smaller than that in the parallel CS architecture without an acceptable permutation. In other words, for the sufficient condition in Lemmas 2 and 3, the condition that "$\mathbf{A}$ obeys the RIP of order $||\mathbf{s}^{\dagger}||_\infty$" for the parallel CS architecture with an acceptable permutation is weaker than the condition that "$\mathbf{A}$ obeys the RIP of order $||\mathbf{s}||_\infty$" for the parallel CS architecture 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 an acceptable permutation, the S nonzero elements in the best S-term approximation of the permuted 2D signal are evenly distributed among the columns, which leads to minimum $||\mathbf{s}^{\dagger}||_\infty$. The permutation that

allows to achieve minimum $||\mathbf{s}^\dagger||_\infty$ is called an optimal permutation and it is formally defined below.

Definition 4: *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* $\binom{N}{S-\lfloor S/N\rfloor}$ *optimal sparsity vectors* $\mathbf{s}^*$ *of* $\mathbf{X}^S$*, where* $\binom{n}{k}$ *is the binomial coefficent "n choose k" and* $\lfloor\cdot\rfloor$ *denotes the floor function.*

Proof. Obviously, $||\mathbf{s}^*||_1 = S$. If $\lceil S/N\rceil = \lfloor S/N\rfloor = S/N$ (where $\lceil\cdot\rceil$ denotes the ceiling function), we can immediately find an optimal sparsity vector $\mathbf{s}^*$ whose entries are all S/N, and this is the only optimal sparsity vector under this case. 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 elements in each of the first $S - N\lfloor S/N\rfloor$ columns, and the remaining nonzero elements are evenly distributed among the remaining columns. Then each of the last $N\lceil S/N\rceil - S$ 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. Obviously, there are totally $\binom{N}{S-\lfloor S/N\rfloor}$ optimal sparsity vectors. This completes the proof. □

From the proof of Lemma 4, it follows that optimal sparsity vector and 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. Thus, an acceptable permutation (see Definition 3) can be used instead.

3.2.2 Group-scan-based permutation

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 $1 \leq i \leq N$, if all elements 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 $\sum_{i=1}^M p_i$.

For example, when $M = N = 4$, if after a permutation the elements 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

$$\Pr\left\{\max_j\{s_j^\dagger\} - \min_j\{s_j^\dagger\} \leq 1\right\} = 0.6003 \tag{3.6}$$

where $\mathrm{E}\{\cdot\}$ means expectation 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 elements in $\mathbf{X}^S$ can be divided into several non-overlapped groups, and in each group all elements share the same probability to be nonzero. Based on the observation at the beginning of this subsection, 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 columns of the best S-term approximation of the new 2D signal are likely to have similar sparsity levels.

3.2.3 Zigzag-scan-based permutation

Definition 5: *For a 2D signal $\mathbf{X} \in \mathbb{R}^{M\times N}$, let $\mathbf{X}(i,j)$ denote the element in the position (i,j). The m-th ($1 \leq m < M+N$) layer of $\mathbf{X}$ is the group of all elements $\mathbf{X}(i,j)$'s satisfying $i+j-1=m$.*

For example, when $M = N = 4$, the following matrix

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_6 & x_7 \\ x_3 & x_5 & x_8 & x_{13} \\ x_4 & x_9 & x_{12} & x_{14} \\ x_{10} & x_{11} & x_{15} & x_{16} \end{bmatrix} \tag{3.7}$$

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 elements in 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), then the following zigzag-scan-based permutation can be applied, 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

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

which row-wisely turns a vector into an $M \times N$ matrix and Z: $\mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{MN}$ is the zigzag scan function which turns a matrix into a "zigzag" sequence.

Correspondingly, define the *inverse zigzag-scan-based permutation* P^{-1}: $\mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{M\times N}$ 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}: $\mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{MN}$ is a vectorization function which row-wisely turns a matrix into a vector and Z^{-1}: $\mathbb{R}^{MN} \rightarrow \mathbb{R}^{M\times N}$ is inverse zigzag scan function which turns a "zigzag" sequence into an $M \times N$ matrix.

For example, the matrix $\mathbf{X}$ given in (3.7) 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,

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

and then returns to the original 2D signal $\mathbf{X}$ after inverse zigzag scan, that is,

$$\mathrm{P}^{-1}(\mathbf{X}^\dagger) = \mathrm{Z}^{-1}(\mathrm{R}^{-1}(\mathbf{X}^\dagger)) = \mathbf{X}.$$

Thus, according to the analysis in Subsection 3.2.2, if elements in each layer of $\mathbf{X}^S$ share similar probabilities to be nonzero, after the zigzag-scan-based permutation, all columns of the permuted $\mathbf{X}^S$ tend to have similar sparsity levels.

Definition 6: *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 the probability of the event E_m that an element in the m-th layer of $\mathbf{X}^S$ is nonzero follows the probability distribution*

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

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* $r_2 \geq 2r_1 - 3r_0 - 1$ *and* $0 \leq r_0 < r_1 < r_2 \leq \min\{M, N\}$*, the zigzag-scan-based permutation* $\mathrm{P}(\cdot)$ *is an acceptable permutation with a large probability that is given as*

$$\begin{aligned}
&\Pr\{\mathrm{P} \text{ is acceptable}\} = \Pr\left\{||\mathbf{s}||_\infty > ||\mathbf{s}^\dagger||_\infty\right\} \\
&\geq 1 - \left[\prod_{m=r_1+1}^{r_2} (1-p_m)^m\right] \cdot \prod_{j=1}^{r_2} \left\{1 + \sum_{k=k_j+1}^{\substack{\min\{\lceil (r_0+r_2+1)/2\rceil,\\ r_2-r_0, r_2-j+1\}}} \right. \\
&\left. \sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j \\ a_1<a_2<\cdots<a_{k-k_j}}} \frac{1}{(e^{\alpha(a_1-1-r_0)}-1)\cdots(e^{\alpha(a_{k-k_j}-1-r_0)}-1)}\right\}
\end{aligned} \tag{3.8}$$

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* $1 \leq j \leq r_2$*,* $\mathcal{A}_j := \{m_j,\ m_j+1,\ \cdots,\ r_2\}$*,* $m_j = \max\{r_1+1, j\}$*, and*

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

Proof. Denote the j-th element 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 elements 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 maximal number of nonzero entries in any column is $u = \lceil (r_0+r_2+1)(r_2-r_0)/2N \rceil$. 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

$$\Pr\{\mathrm{P} \text{ is acceptable}\} = \Pr\{||\mathbf{s}||_\infty > ||\mathbf{s}^\dagger||_\infty\} \tag{3.9a}$$

$$= \sum_{t=1}^{r_2-r_0} \Pr\{||\mathbf{s}||_\infty = t, ||\mathbf{s}^\dagger||_\infty \leq t-1\} \tag{3.9b}$$

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

$$= \sum_{t=u+1}^{r_2-r_0} \Pr\{||\mathbf{s}||_\infty = t\} \tag{3.9c}$$

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

$$\geq 1 - \Pr\{||\mathbf{s}||_\infty \leq l\}. \tag{3.9d}$$

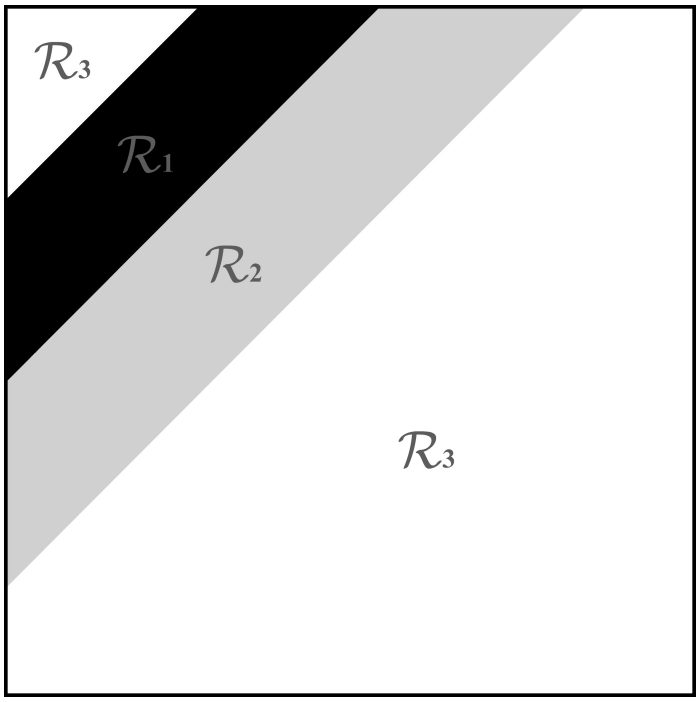


Fig. 3.1: Regions in the (r_0, r_1, r_2, α)-layer model .

To write (3.9a), we have used the fact that an acceptable permutation must result in $||\mathbf{s}^\dagger||_\infty < ||\mathbf{s}||_\infty$. For deriving (3.9b), we have used 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 (3.9c), we have used the fact that $u \geq ||\mathbf{s}^\dagger||_\infty$. Finally, for deriving (3.9d), we have used the fact that $u \leq l$. Based on (3.9d), 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

$$\Pr\{||\mathbf{s}||_\infty \leq l\} = \prod_{j=1}^{N} \Pr\{s_j \leq l\}. \tag{3.10}$$

Moreover, since the position (i, j) of an element in $\mathbf{X}^S$ indicates the index m of the layer where the element 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. Theses regions are shown in Fig. 3.1. Therefore, according to Definition 6, all elements of $\mathbf{X}^S$ are nonzero with probability 1 in region $\mathcal{R}_1$. In region $\mathcal{R}_3$, all elements of $\mathbf{X}^S$ are zero with probability 1. In region $\mathcal{R}_2$, the probability of an element to be nonzero decreases exponentially with decay factor α as the layer index m increases. We use p_m to denote the probability of an element in the m-th layer of $\mathbf{X}^S$ to be nonzero.

For the j-th column of $\mathbf{X}^S$, if $r_2 + 1 \leq j \leq N$, all elements of the column are in region $\mathcal{R}_3$ and thus are all zeros. We have $\Pr\{s_j \leq l\} = 1$ since $l \geq r_1 - r_0 \geq 1$. According to (3.10), we have

$$\Pr\{||\mathbf{s}||_\infty \leq l\} = \prod_{j=1}^{r_2} \Pr\{s_j \leq l\} = \prod_{j=1}^{r_2} \sum_{k=0}^{l} \Pr\{s_j = k\}. \tag{3.11}$$

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 elements in the j-th column of $\mathbf{X}^S$ that are in the region $\mathcal{R}_1$, i.e.,

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

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 (3.11), for $1 \leq j \leq r_2$, i.e., the first r_2 columns of $\mathbf{X}^S$, we consider 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 elements of the j-th column of $\mathbf{X}^S$ that are in the region $\mathcal{R}_2$ are all zeros. Therefore, we have

$$\Pr\{s_j = k\} = \prod_{m=m_j}^{r_2} (1 - p_m). \tag{3.12}$$

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 elements in the region $\mathcal{R}_2$. Denote the indices of these $(k - k_j)$ nonzero elements as $a_1, a_2, \cdots, a_{k-k_j}$ with $a_1 < a_2 < \cdots < a_{k-k_j}$.

So $a_1, a_2, \cdots, a_{k-k_j} \in \mathcal{A}_j \stackrel{\triangle}{=} \{m_j, m_j+1, \cdots, r_2\}$. We then have

$$\begin{aligned}
&\Pr\{s_j = k\} \\
&= \sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j \\ a_1<a_2<\cdots<a_{k-k_j}}} p_{a_1}\cdots p_{a_{k-k_j}} \prod_{\substack{m=m_j \\ m\neq a_i \\ i=1,\cdots k-k_j}}^{r_2} (1-p_m) \\
&= \sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j \\ a_1<a_2<\cdots<a_{k-k_j}}} \frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})} \prod_{m=m_j}^{r_2}(1-p_m) \\
&= \Big[\prod_{m=m_j}^{r_2}(1-p_m)\Big] \sum_{\substack{a_1,a_2,\cdots,a_{k-k_j}\in\mathcal{A}_j \\ a_1<a_2<\cdots<a_{k-k_j}}} \frac{p_{a_1}\cdots p_{a_{k-k_j}}}{(1-p_{a_1})\cdots(1-p_{a_{k-k_j}})}. \qquad (3.13)
\end{aligned}$$

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

$$\Pr\{s_j = k\} = 0. \qquad (3.14)$$

According to (3.9d) and (3.11)-(3.14) and the fact that $l \ge r_1 - r_0 \ge k_j$, we have

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

where to obtain (a) we have used the fact that $m_j = r_1 + 1$ for $1 \le j \le r_1$ and $m_j = j$ for $r_1 + 1 \le j \le r_2$. Using the facts that $l = \lceil (r_0 + r_2 + 1)/2 \rceil$ and $p_m = e^{-\alpha(m-r_0-1)}$ for $r_1 + 1 \le m \le r_2$, we obtain (3.8). This completes the proof. □

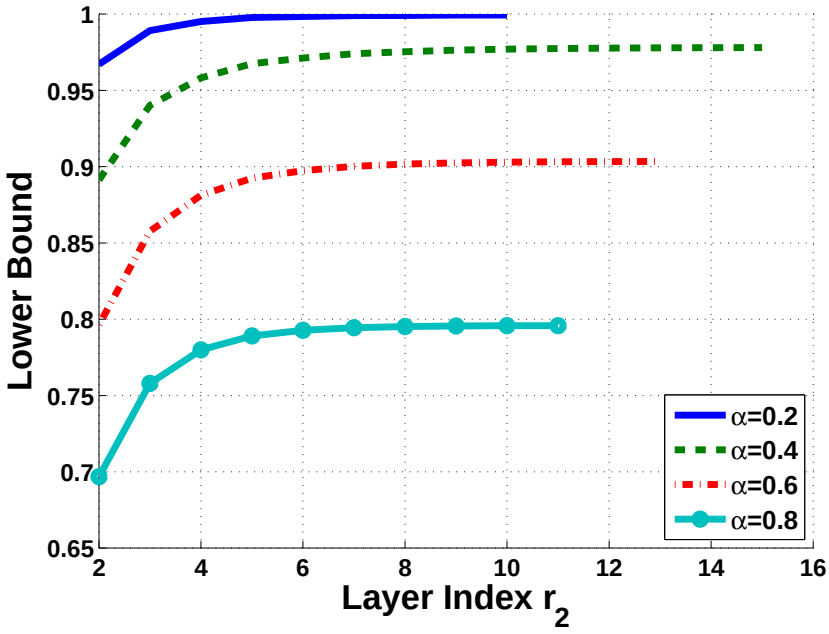


Fig. 3.2: Lower bound of $\Pr\{\text{P is acceptable}\}$ in (3.8) for $r_0 = 0, r_1 = 1$.

Figs. 3.2–3.4 show the value of the lower bound on the probability $\Pr\{\text{P is acceptable}\}$ in (3.8) 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 $\Pr\{\text{P is acceptable}\}$ is large enough in general. For other values of 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-DCT (DCT2) coefficient matrices of 2D piecewise smooth image signals. Since most of the large elements in the DCT2 coefficient matrix of a piecewise smooth image signal typically lie in the top left corner, and small elements lie in the bottom right corner because most of the energy of such image signal is concentrated in low frequencies, the zigzag scan process is commonly used in image compression like JPEG [1]. Thus, the DCT2 coefficient matrices of piecewise smooth image signals satisfy the (r_0, r_1, r_2, α)-layer model with $r_0 = 0$ (which will also be shown via simulations in Subsection 3.4.1), and 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

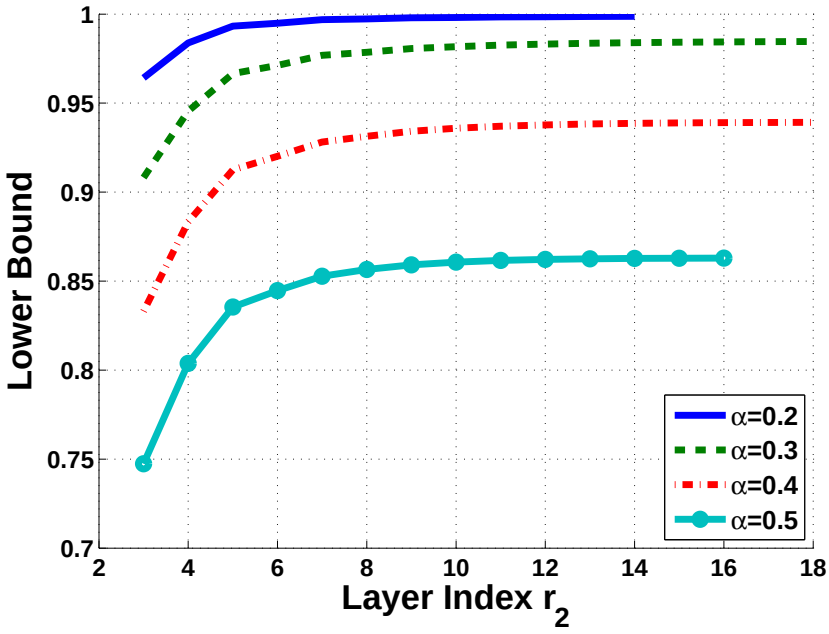


Fig. 3.3: Lower bound of $\Pr\{\text{P is acceptable}\}$ in (3.8) for $r_0 = 0, r_1 = 2$.

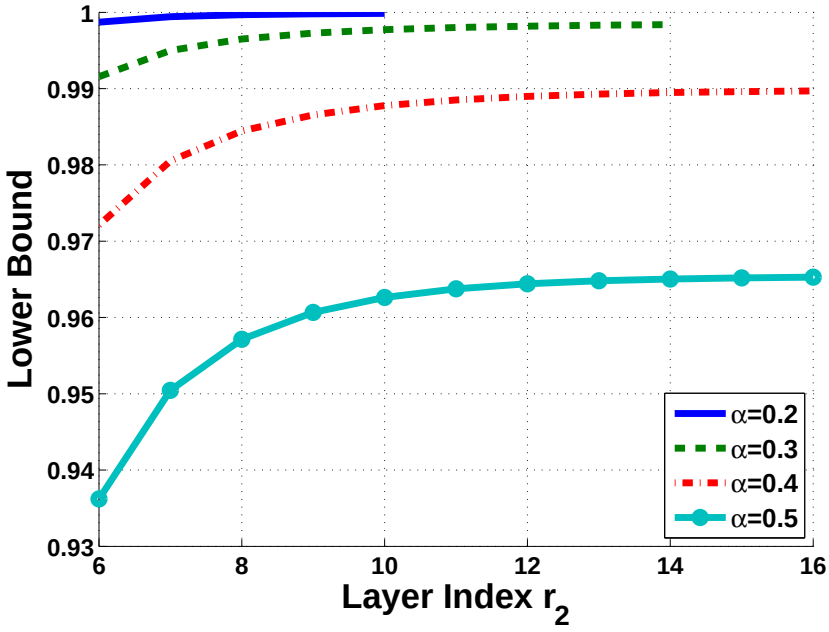
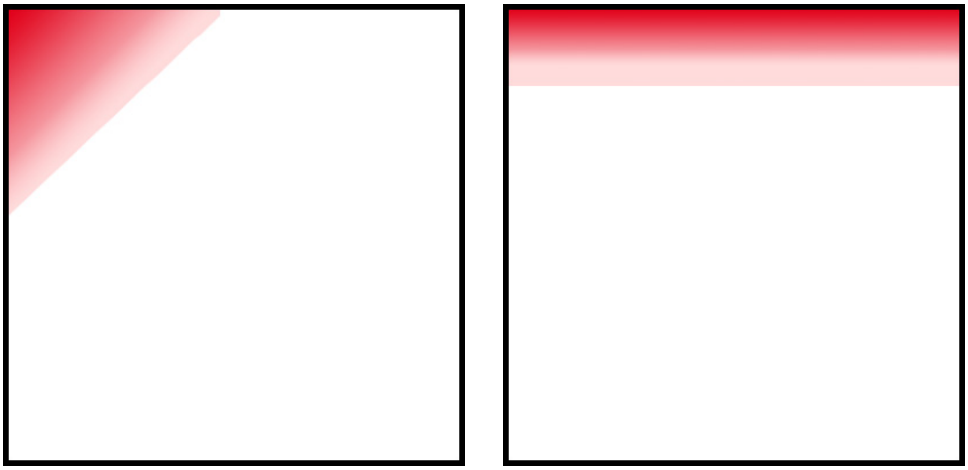


Fig. 3.4: Lower bound of $\Pr\{\text{P is acceptable}\}$ in (3.8) for $r_0 = 3, r_1 = 5$.

of the layer indices 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. 3.5 shows the difference before and after the zigzag-scan-based permutation when



(a) Before the zigzag-scan-based permutation

(b) After the zigzag-scan-based permutation

Fig. 3.5: Energy distribution of a DCT2 coefficient matrix before and after the zigzag-scan-based permutation.

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. In Subsection 3.4.1, we will also show by simulations that the zigzag-scan-based permutation is an acceptable permutation for DCT2 coefficient matrices of several typical images.

3.2.4 Block-test-based permutation

In this subsection, another permutation based on the notion of group-scan-based permutation is proposed, named *block-test-based permutation*. Consider a 2D signal whose nonzero entries are clustered in some regions, e.g., the difference between two consecutive frames in a low-motion video sequence, whose nonzero entries are likely to be clustered in regions where the moving objects in the video frames are located. If the 2D signal is divided into a number of blocks, then all blocks are classified into two groups: one group includes blocks that have nonzero entries, and the other group includes blocks with all entries being zeros. Then in the group-scan-based permutation, blocks in the first group are scanned first, followed by blocks in the second group. The classification is called *block test*, and the corresponding

permutation is named *block-test-based permutation*.

For clarity, $\mathbf{X}[i] \in \mathbb{R}^{b_1 \times b_2}$ is used to denote the i-th block of $\mathbf{X} \in \mathbb{R}^{M \times N}$ when divided into B non-overlapping blocks of size $b_1 \times b_2$, i.e.,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}[1] & \mathbf{X}[2] & \cdots & \mathbf{X}[\frac{1}{M}B] \\ \mathbf{X}[\frac{1}{M}B+1] & \mathbf{X}[\frac{1}{M}B+2] & \cdots & \mathbf{X}[\frac{2}{M}B] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}[\frac{M-1}{M}B+1] & \mathbf{X}[\frac{M-1}{M}B+2] & \cdots & \mathbf{X}[B] \end{bmatrix}.$$

A block $\mathbf{X}[i]$ is said to pass the *block test* if $\mathbf{X}[i]$ has sparsity larger than 0. Define the set of block indices that pass the block test as

$$\mathcal{B} = \{i \mid \mathrm{I}(||\mathbf{X}[i]||_1 > 0), 1 \leq i \leq B\}$$

where $\mathrm{I}(\cdot)$ is the indicator function.

For $i_1 < i_2 < \cdots < i_m$ and $j_1 < j_2 < \cdots < j_n$, define the block test function T: $\mathbb{R}^{M \times N} \to \mathbb{R}^{MN}$ of $\mathbf{X}$ as

$$\mathrm{T}(\mathbf{X}) = [\mathrm{V}(\mathbf{X}[i_1]), \mathrm{V}(\mathbf{X}[i_2]), \cdots, \mathrm{V}(\mathbf{X}[i_m]), \mathrm{V}(\mathbf{X}[j_1]), \mathrm{V}(\mathbf{X}[j_2]), \cdots, \mathrm{V}(\mathbf{X}[j_n])]$$

where $i_k \in \mathcal{B}$ for $1 \leq k \leq m = |\mathcal{B}|$, $j_l \in \bar{\mathcal{B}}$ for $1 \leq l \leq n = B - m$, and V: $\mathbb{R}^{b_1 \times b_2} \to \mathbb{R}^{b_1 b_2}$ is a vectorization function. By knowing $\mathcal{B}$ and block dimensions b_1 and b_2, the inverse block test function $\mathrm{T}^{-1} : \mathbb{R}^{MN} \to \mathbb{R}^{M \times N}$ can recover the "block" sequence $\mathrm{T}(\mathbf{X})$ into the original matrix, i.e.,

$$\mathrm{T}^{-1}(\mathrm{T}(\mathbf{X})) = \mathbf{X}.$$

Define the *block-test-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 $\mathrm{P}(\mathbf{X}) = \mathrm{R}(\mathrm{T}(\mathbf{X}))$, where R: $\mathbb{R}^{MN} \to \mathbb{R}^{M \times N}$ is a row-wisely reshape function which turns a vector into a matrix and T: $\mathbb{R}^{M \times N} \to \mathbb{R}^{MN}$ is block test function which turns a matrix into a "block" sequence.

Correspondingly, define the *inverse block-test-based permutation* $\mathrm{P}^{-1} : \mathbb{R}^{M \times N} \to \mathbb{R}^{M \times N}$ for a 2D signal $\mathbf{X}^\dagger \in \mathbb{R}^{M \times N}$ as $\mathrm{P}^{-1}(\mathbf{X}^\dagger) = \mathrm{T}^{-1}(\mathrm{R}^{-1}(\mathbf{X}^\dagger))$, where $\mathrm{R}^{-1} : \mathbb{R}^{M \times N} \to \mathbb{R}^{MN}$ is a vectorization function which turns a matrix into a vector and $\mathrm{T}^{-1} : \mathbb{R}^{MN} \to \mathbb{R}^{M \times N}$ is inverse block test function which turns a "block" sequence into a matrix by knowing $\mathcal{B}$ and block dimensions b_1 and b_2.

The block-test-based permutation is introduced for 2D block sparse signals, whose nonzero entries occur in clusters. The block sparse 1D signals are consider in [51], [52], and [53]. In this section, it is extended into 2D signals. The definition of 2D block J-sparse signals is given as follows.

Definition 7: *A 2D signal* $\mathbf{X} \in \mathbb{R}^{M \times N}$ *is called block* J*-sparse if* J *blocks of* $\mathbf{X}$ *have one or more nonzero entries.*

In this thesis, the block-test-based permutation is optionally used for difference between a non-reference frame and its preceding reference frame in the compressed video sensing scheme as described in Section 3.3. It will be shown in Section 3.4.3 that the best S-term approximation of the difference between a non-reference frame and its preceding reference frame is block J-sparse signal with $J \ll B$, and the block-test-based permutation is an acceptable permutation for these signals.

3.3 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 parallel CS with the 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 [54]. In wireless multimedia sensor networks, the C-DMRC system is preferred compared to traditional video coding standards such as the moving picture experts group (MPEG) standards, 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 [38]. Thus, as we discussed at the beginning of this chapter, 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 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. 3.6. The average compression ratio is the ratio of the number of measurements to the total number of pixel values for a pair of a reference frame and a non-reference frame. The average compression ratio 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

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

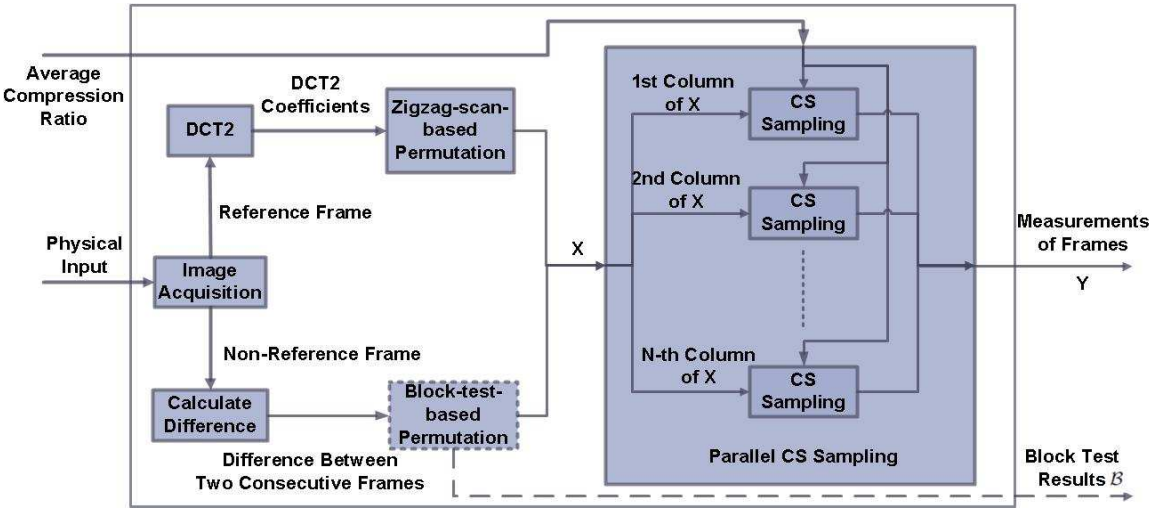


Fig. 3.6: Block diagram of the CS video encoder.

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.

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) optionally perform the block-test-based permutation on the difference coefficient matrix; 3) perform parallel compressed sampling of this (permuted) difference. The outputs of all CS sampling processors are combined.[8] If the block-test-based permutation is performed, the block test results will be sent with the measurements of frames. The block-test-based permutation is an optional process (shown by the dashed line border in Fig. 3.6) because according to our simulation, it improves only a little the performance of the reconstruction of the non-reference frames. Details would be discussed in Subsection 3.4.4.

Considering that the sparsity level of the difference between the 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

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

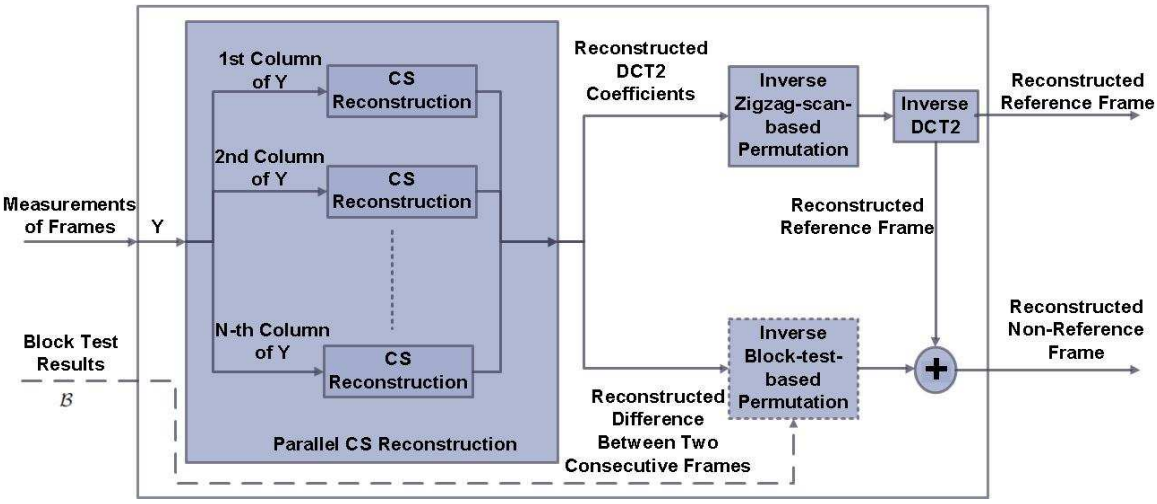


Fig. 3.7: Block diagram of the CS video decoder.

that of the reference frames, i.e., fewer measurements are assigned to the non-reference frames. In our experiment in Section 3.4.4, we set the ratio of measurements being 4:1, i.e., the number of measurements for the reference frames is 4 times that for the 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. 3.7. To decode a reference frame at the receiver side, the following steps are performed: 1) parallel CS reconstruction from the measurements of the reference frame; 2) the inverse zigzag-scan-based permutation; 3) inverse DCT2 on the reconstructed DCT2 coefficient matrix. To decode a non-reference frame, the following steps are performed: 1) parallel CS reconstruction from the measurements of the non-reference frame; 2) the block-test-based permutation if the block test result of the current non-reference frame is available; 3) adding 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 BP algorithm, can be used.

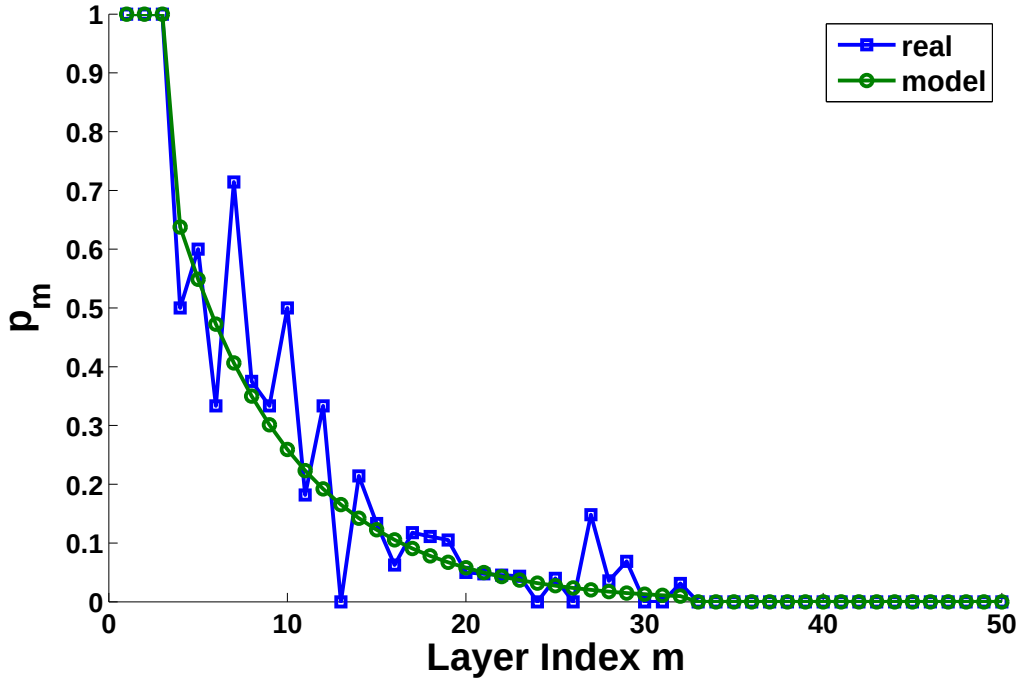


Fig. 3.8: Layer model of Boat.tiff.

3.4 Simulation results

3.4.1 The layer model and the zigzag-scan-based permutation

We first 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. 3.8, the x-axis is the layer index m, and y-axis is the probability of an element in the m-th layer of the best S-term approximation $\mathbf{X}^S$ of the DCT2 coefficient matrix to be nonzero, calculated as $p_m = (1/m)\sum_{i+j-1=m} \mathrm{I}(\mathbf{X}^S(i,j) \neq 0)$. 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. 3.8. 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 for DCT2 coefficient matrices of such images with an overwhelming probability.

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 3.1. The DCT2 coefficient matrices are taken from four test images: Boat (512×512), Lena (512×512),

Table 3.1: 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

Cameraman (256×256), Peppers (512×512). The best S-term approximation is chosen according to different magnitude thresholds by keeping DCT2 coefficients whose magnitudes are not less than the magnitude threshold and setting the remaining to be zeros. Table 3.1 shows that $||\mathbf{s}||_\infty$ decreases significantly after the zigzag-scan-based permutation, which is consistent with Proposition 1.

3.4.2 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, Lena, Cameraman, and Peppers. The PSNR is employed to show the reconstruction performance. We compare the performances of the parallel CS scheme for the configurations: 1) with no 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 $1/K$. The parallel CS reconstruction is implemented using BP algorithm by the CVX optimization toolbox.[9]

Other reconstruction algorithms than the BP can also be used. PSNR performance for different methods is shown in Fig. 3.9 versus the compression ratio, which is the ratio of the number of measurements to the total number of elements in the DCT2 coefficient matrix.

From Fig. 3.9, we can see that under the same compression ratio, the zigzag-scan-based permutation helps to improve the PSNR by around 4 dB for all images. Consequently, it shows that the PNSR performance is indeed improved significantly after permutation.

3.4.3 Block sparse model and the block-test-based permutation

In Tables 3.2 and 3.3, the block sparse model for the DCT2 coefficient matrices of the four different gray images and the matrices of the difference between the non-reference frame and its preceding reference frame in several video sequences have been checked. In this

[9]The toolbox was downloaded at http://cvxr.com/cvx.

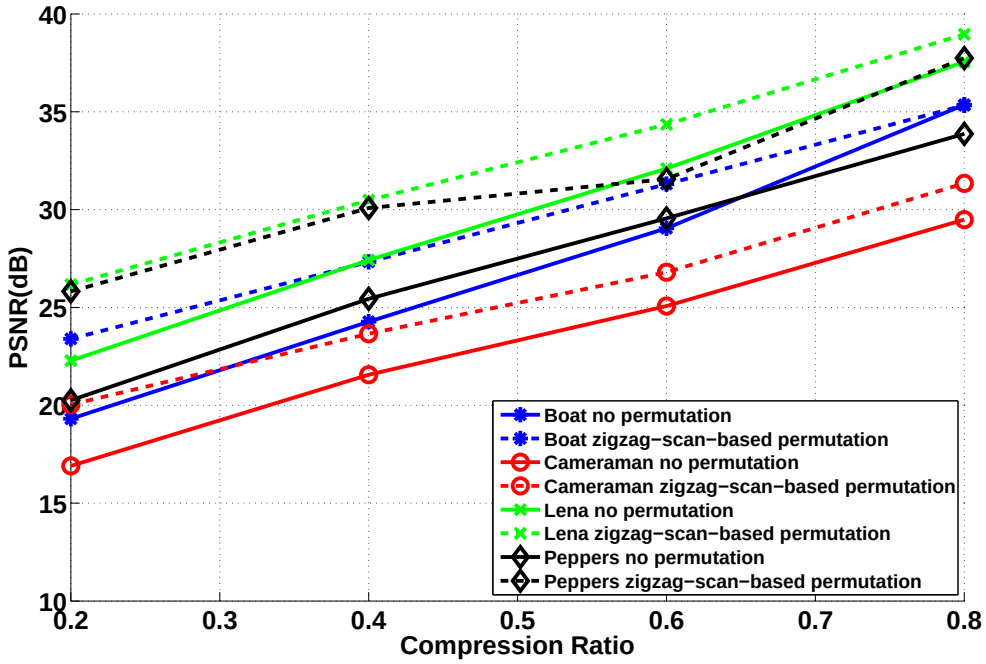


Fig. 3.9: PSNR for the parallel CS scheme with/without the zigzag-scan-based permutation in image compression.

simulation, the best S-term approximation is chosen so that all elements have absolute value not less than the magnitude threshold shown in the table. The block size is 8×8. Then, for each image, there are $B = 4096$ blocks, except that Cameraman has only $B = 1024$ blocks. For each video frame, there are 396 blocks. According to our simulation, the block sparsity J is much less than B for either the DCT2 coefficient matrices of images or the matrices of the difference between the non-reference frame and its preceding reference frame of videos.

The changes of $||\mathbf{s}||_\infty$ of the best S-term approximation before and after the block-test-based permutation are also shown in Tables 3.4 and 3.5. It can be seen from the tables that $||\mathbf{s}||_\infty$ decreases, although sometimes slightly, after the block-test-based permutation.

3.4.4 Video compression via parallel CS with 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 PSNR for the reference frames and non-reference frames

Table 3.2: Block sparsity of the DCT2 coefficient matrices of images.

Image	**Magnitude Threshold**			
	400	600	800	1000
Boat	33	23	19	16
Cameraman	13	11	7	4
Lena	14	11	8	7
Peppers	27	15	11	11

Table 3.3: Block sparsity of the matrices of the difference between two consecutive frames in video sequences.

Akiyo	**Magnitude Threshold**	2	4	6	8
	Block Sparsity	18	3	2	1
Carphone	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	51	31	19	8
Claire	**Magnitude Threshold**	20	30	40	50
	Block Sparsity	12	6	4	1
Coastguard	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	13	6	2	1
Foreman	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	42	19	9	1
Salesman	**Magnitude Threshold**	5	10	15	20
	Block Sparsity	11	5	3	1

is used as performance metric. All settings are the same as in the example in Subsection 3.4.2. The following parallel CS schemes are compared: 1) the scheme with no permutation; 2) the scheme with only the zigzag-scan-based permutation on the reference frames; 3) the scheme with the zigzag-scan-based permutation on the reference frames and the block-test-based permutation on the non-reference frames. PSNR performance for different methods is shown in Figs. 3.10–3.12 versus the average compression ratio, that is computed as (compression ratio of reference frames + compression ratio of non-reference frames)/2.

From Fig. 3.10, we can see that under the same average compression ratio, the zigzag-scan-based permutation helps to improve the PSNR of the reference frames by around 3∼9 dB for Akiyo, 5∼6 dB for Foreman, and 4∼8 dB for Coastguard.

Fig. 3.11 shows that the zigzag-scan-based permutation also improves the PSNR performance of the 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 the non-reference frames is a bit lower

Table 3.4: Comparison of $||\mathbf{s}||_\infty$ before and after the block-test-based permutation (DCT2 coefficient matrices of images).

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

Table 3.5: Comparison of $||\mathbf{s}||_\infty$ before and after the block-test-based permutation (matrices of the difference between two consecutive frames in video sequences).

Akiyo	**Magnitude Threshold**	2	4	6	8
	Block Sparsity	18 vs. 9	3 vs. 1	2 vs. 1	1 vs. 1
Carphone	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	51 vs. 16	31 vs. 17	19 vs. 4	8 vs. 2
Claire	**Magnitude Threshold**	20	30	40	50
	Block Sparsity	12 vs. 4	6 vs. 3	4 vs. 1	1 vs. 1
Coastguard	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	13 vs. 8	6 vs. 3	2 vs. 1	1 vs. 1
Foreman	**Magnitude Threshold**	20	40	60	80
	Block Sparsity	42 vs. 18	19 vs. 16	9 vs. 2	1 vs. 1
Salesman	**Magnitude Threshold**	5	10	15	20
	Block Sparsity	11 vs. 6	5 vs. 3	3 vs. 1	1 vs. 1

than that of the preceding reference frames because the reconstruction of the non-reference frames relies on both the reconstruction of their preceding reference frames and the reconstruction of the difference between the non-reference frames and their preceding reference frames.

Moreover, Fig. 3.12 demonstrates that the block-test-based permutation scheme slightly improves the PSNR performance. This is because the PSNR performance for the non-reference frames can hardly be better than that for the reference frames due to the fact that the reconstruction of the non-reference frames is based on the reconstruction of the reference frames. Furthermore, the difference between the reference and non-reference frames is already sparse enough such that the space for improvement due to the block-test-based permutation in reconstruction of the non-reference frames is limited. Thus, we recommend to use the block-test-based permutation as an optional process. However, the simulation

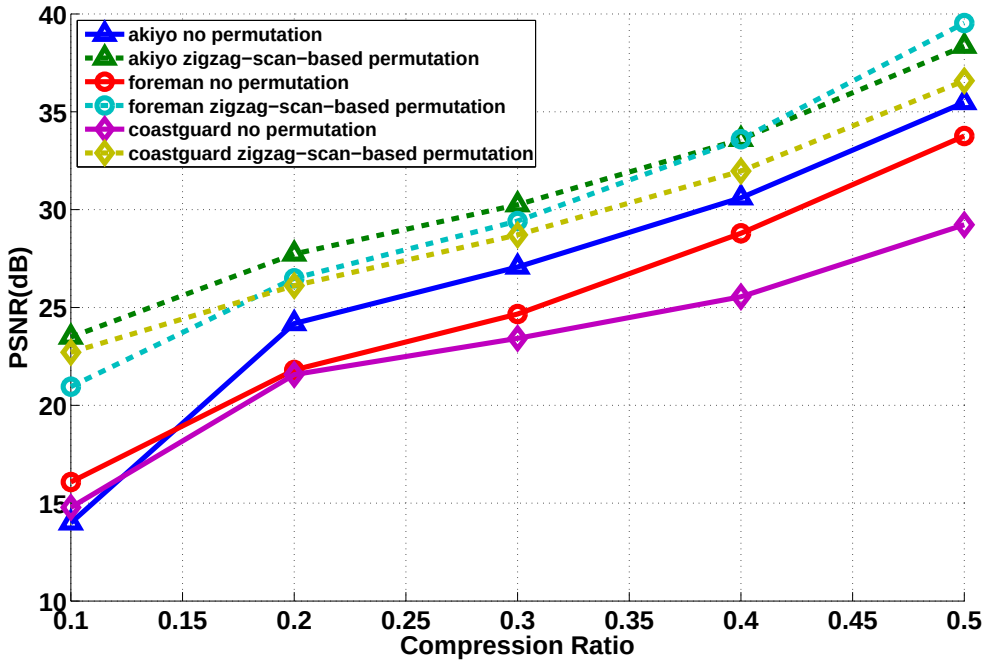


Fig. 3.10: Average PSNR of the reference frames.

result still proves that the block-test-based permutation works. Therefore, we can use it for 2D compressible signals in other scenarios, e.g., to encode fast motion videos.

To show the advantage of the video compression scheme proposed in Section 3.3, we compare the total time of reconstructing one pair of the reference and non-reference frames using (i) the video encoder and decoder employed in the C-DMRC system proposed in [54] and (ii) the video encoder and decoder proposed in Section 3.3. We also show the PSNRs of the 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 (block length equal to 32) and the random Gaussian matrix, respectively. The CS reconstruction algorithm is implemented using the l_1-magic package.[10] To eliminate the effects of randomness, we run 200 trials for each average compression ratio and show the average PSNR and total reconstruction time. The results are shown in Tables 3.6 and 3.7.

It can be seen from Tables 3.6 and 3.7 that the reconstruction time using the video encoder and decoder proposed in Section 3.3 is less than that for the video encoder and decoder employed in [54], especially when the compression ratio is low. In addition, if there

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

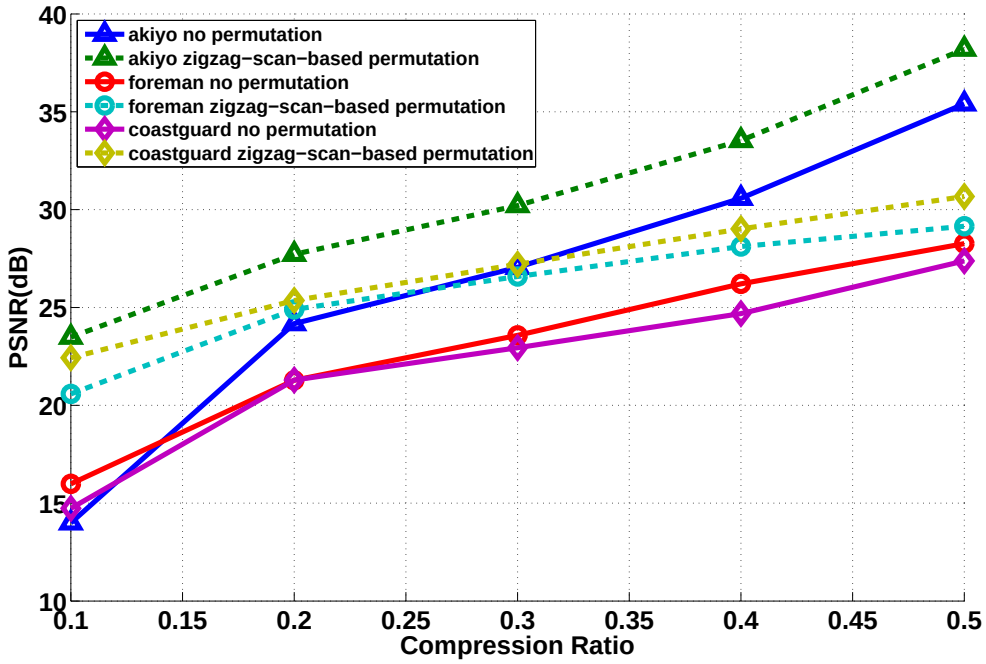


Fig. 3.11: Average PSNR of the non-reference frames (comparison for the zigzag-scan-based permutation).

are multiple decoding processors simultaneously reconstructing the columns of a video frame as shown in Fig. 3.7, the reconstruction time can be further reduced approximately to the total reconstruction time divided by the number of decoding processors. It can also be observed in Table 3.6 that the time for reconstruction using the video encoder and decoder employed in [54] decreases as the average compression ratio increases. This is because the reconstruction algorithm converges faster as the number of measurements increases. According to Table 3.7, the time for reconstruction using the video encoder and decoder proposed in Section 3.3 is less sensitive to the compression ratio. In addition, we can see that compared to the video encoder and decoder employed in [54], the PSNR of reconstructed video frames for the video encoder and decoder proposed in Section 3.3 is larger when the average compression ratio is larger than 0.3, and it is almost the same (less than 0.3 dB degradation) when the average compression ratio is smaller than 0.3.

3.5 Chapter summary

In this chapter, a parallel CS architecture with permutation has been proposed. It has been proved that with a so-called acceptable permutation, the RIP condition for the sensing

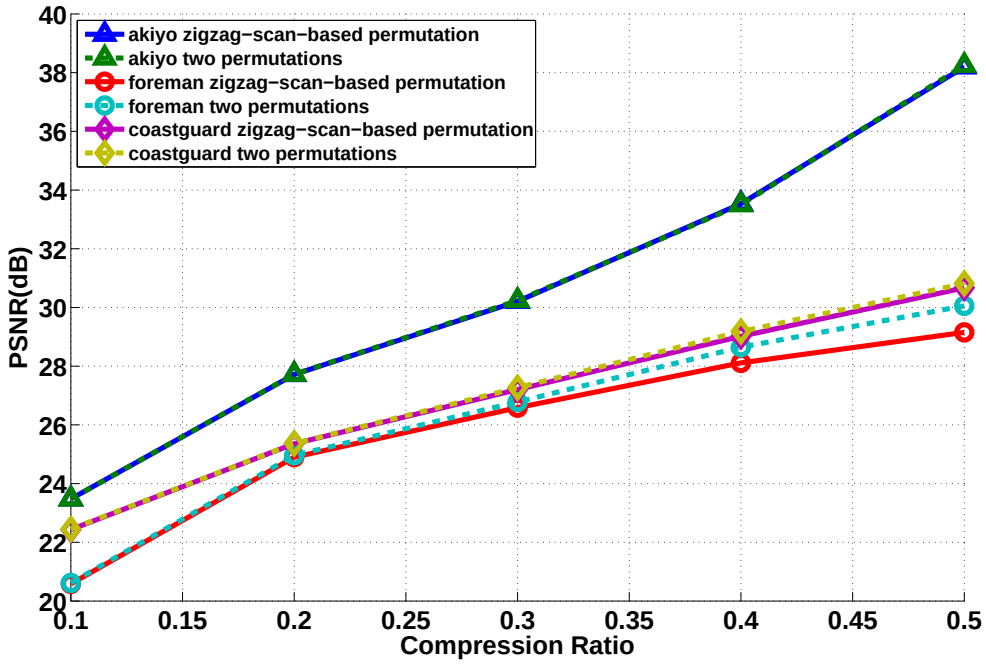


Fig. 3.12: Average PSNR of the non-reference frames (comparison for the block-test-based permutation).

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 the DCT2 coefficient matrices of 2D images, has been analyzed. Meanwhile, the block-test-based permutation is introduced for 2D signals satisfying the block sparse model. The application to image and video compression has been discussed as well. In the simulations, it has been shown that both the zigzag-scan-based and the block-test-based permutations for the DCT2 coefficient matrices of images are acceptable permutations. The zigzag-scan-based permutation is shown to be effective and bring a significant performance improvement, while the block-test-based permutation is shown to be effective, though not necessarily significant. In addition, the simulation results have shown that the permutation improves the reconstruction performance of images and videos in terms of PSNR.

Table 3.6: Total reconstruction time and PSNR of reconstructed video frames using the video encoder and decoder employed in [54].

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 3.7: Total reconstruction time and PSNR of reconstructed video frames using the video encoder and decoder proposed in Section 3.3.

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

Chapter 4

Parallel Compressed Sensing Reconstruction[1]

Block-diagonal measurement matrix has been used in several papers on distributed sampling scenario, i.e., different segments of the signal $\mathbf{x}$ can be sampled in parallel [36], [38]. It is further proved in [56] and [57] that given any sparsifying basis $\mathbf{\Psi}$, the corresponding sensing matrix $\mathbf{A}$ for a block-diagonal random measurement matrix $\mathbf{\Phi}$ would always satisfy the RIP condition. In the aforementioned works, the reconstruction has to be conducted in a centralized manner, i.e., the projection $\boldsymbol{\theta}$ has to be reconstructed as a whole. Thus, the reconstruction has high computational complexity, especially when the length of the signal $\mathbf{x}$ is very large. In applications where the time for reconstruction is a crucial evaluation criterion, parallel reconstruction is desired. In the parallel CS architecture presented in Chapter 3, the projection $\boldsymbol{\theta}$ has to be obtained and "explicitly" permuted at the encoder side. However, when the signal $\mathbf{x}$ is an analog signal, obtaining $\boldsymbol{\theta}$ requires a high-rate sampling of $\mathbf{x}$ first, whereas one major advantage of CS is the ability to compress $\mathbf{x}$ directly using the measurement matrix $\mathbf{\Phi}$ without such high-rate sampling. Besides, the permutation performed at the encoder side in Chapter 3 increases the workload of the encoder.

To address the above problem of the parallel CS architecture, we propose in this chapter a parallel CS reconstruction architecture as a centralized sampling and parallel reconstruction architecture. It is efficient in applications where centralized sampling is acceptable and fast reconstruction is preferred, e.g., the single-pixel camera of [7]. In this architecture, a block-diagonal sensing matrix is employed. Thus, different segments of the projection $\boldsymbol{\theta}$ can be reconstructed in parallel, which helps to significantly reduce the computational complexity and the time needed in reconstruction. Furthermore, we show that the error performance of the parallel reconstruction architecture can be improved by permuting $\boldsymbol{\theta}$,

[1] A version of this chapter has been submitted to IEEE Trans. on Signal Process. Lett. for publication [55].

and the permutation on $\boldsymbol{\theta}$ can be implicitly implemented by a new measurement matrix. Therefore, given a fixed permutation for $\boldsymbol{\theta}$, signal $\mathbf{x}$ can be directly compressed using the new measurement matrix at the encoder side. In this way, the projection $\boldsymbol{\theta}$ is not required to be obtained and explicitly permuted at the encoder side, and thus, the high-rate sampling can be avoided when $\mathbf{x}$ is an analog signal. Compared to existing CS architectures with centralized sampling, our only modification on the CS encoder is on the measurement matrix. Thus, the parallel CS reconstruction architecture proposed in this letter is suitable for most existing CS acquisition architectures, e.g., the single-pixel camera.

The remainder of this chapter is organized as follows. Subsection 4.1 explains the parallel CS reconstruction architecture and investigates the permutations. Simulation results are presented in Subsection 4.2. Subsection 4.3 concludes the work.

4.1 Parallel CS reconstruction

Given a multidimensional signal $\mathbf{x} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$, its vector-reshaped representation is $\bar{\mathbf{x}} \in \mathbb{R}^{\bar{N}}$ where $\bar{N} = \prod_{i=1}^{d} N_i$. Denote $\boldsymbol{\theta}$ as the projection of $\mathbf{x}$ onto some orthonormal sparsifying basis, and denote the vector-reshaped representation of $\boldsymbol{\theta}$ as $\bar{\boldsymbol{\theta}}$. Using the Kronecker product property of sparsifying basis [23], $\bar{\boldsymbol{\theta}}$ is the projection of $\bar{\mathbf{x}}$ on an orthonormal sparsifying basis $\bar{\boldsymbol{\Psi}} \in \mathbb{R}^{\bar{N} \times \bar{N}}$, i.e., $\bar{\mathbf{x}} = \bar{\boldsymbol{\Psi}}\bar{\boldsymbol{\theta}}$. Thus, the measurement vector for $\bar{\mathbf{x}}$ can be obtained by $\bar{\mathbf{y}} = \bar{\boldsymbol{\Phi}}\bar{\mathbf{x}} = \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Psi}}\bar{\boldsymbol{\theta}} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}$ where $\bar{\boldsymbol{\Phi}}$ is the Kronecker product measurement matrix and $\bar{\mathbf{A}} \triangleq \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Psi}}$ is the sensing matrix. The decoder needs to reconstruct $\bar{\boldsymbol{\theta}}$ from $\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}$.

In the parallel CS reconstruction architecture proposed in this chapter, our objective is to design the sensing matrix $\bar{\mathbf{A}}$ such that different segments of $\bar{\boldsymbol{\theta}}$ can be reconstructed in parallel. We partition $\bar{\boldsymbol{\theta}}$ into M segments such that

$$\bar{\boldsymbol{\theta}} = [\underbrace{\theta_1 \cdots \theta_{l_1}}_{\boldsymbol{\theta}^T[1]} \underbrace{\theta_{l_1+1} \cdots \theta_{l_1+l_2}}_{\boldsymbol{\theta}^T[2]} \cdots \underbrace{\theta_{l_1+l_2+\cdots+l_{M-1}+1} \cdots \theta_{\bar{N}}}_{\boldsymbol{\theta}^T[M]}]^T$$

where θ_i denotes the i-th element of the projection $\bar{\boldsymbol{\theta}}$, $(\cdot)^T$ stands for the transpose operation, and $\boldsymbol{\theta}[i]$ denotes the i-th segment of the projection $\bar{\boldsymbol{\theta}}$ with length l_i. So we have $\bar{N} = \sum_{i=1}^{M} l_i$. Besides, we design the sensing matrix of size $K \times \bar{N}$ as a block-diagonal matrix, i.e.,

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}[1] & 0 & \dots & 0 \\ 0 & \mathbf{A}[2] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}[M] \end{bmatrix} \tag{4.1}$$

where $\mathbf{A}[i] \in \mathbb{R}^{K_i \times l_i}$ and $\sum_{i=1}^{M} K_i = K$.

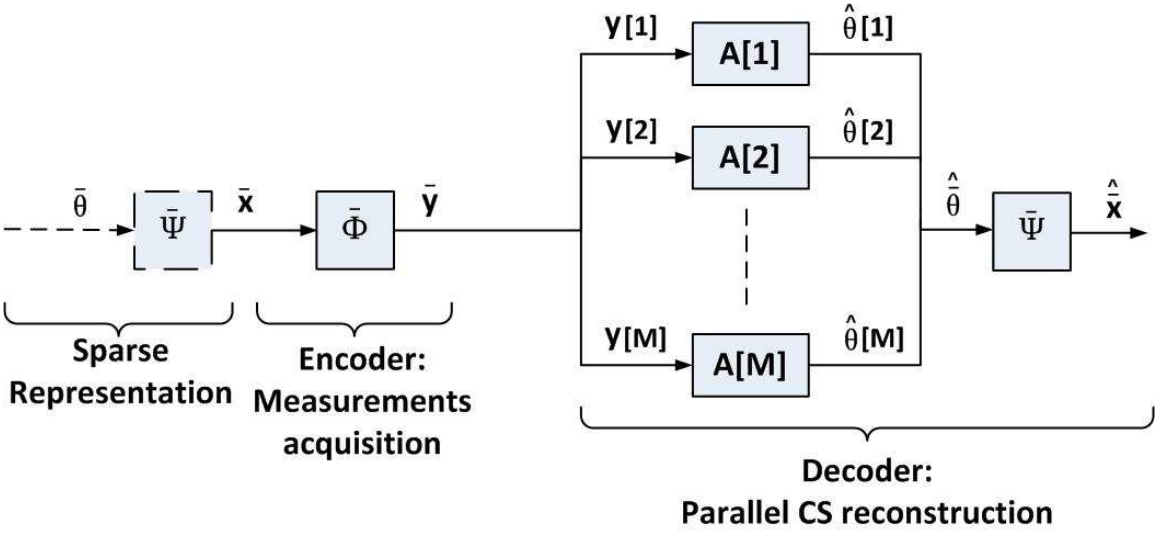


Fig. 4.1: Block diagram of the system employing the parallel CS reconstruction architecture.

The block diagram of the system employing the parallel CS reconstruction architecture is given in Fig. 4.1. In practice, during the acquisition of measurements, $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\Psi}}$ are not required to be obtained or stored, which is shown in the diagram by the dashed line and dashed line border of the corresponding block. At the encoder side, the measurement matrix is given by $\bar{\boldsymbol{\Phi}} = \bar{\mathbf{A}}\bar{\boldsymbol{\Psi}}^T$ and the measurement vector is given by $\bar{\mathbf{y}} = \bar{\boldsymbol{\Phi}}\bar{\mathbf{x}}$. When $\bar{\mathbf{x}}$ is an analog signal, the measurements acquisition process can be implemented via hardware without the digitalization of $\bar{\mathbf{x}}$ by high-rate sampling [58]. At the decoder side, since $\bar{\mathbf{A}}$ is block-diagonal, the measurement vector $\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}$ can be divided into M segments, i.e., $\mathbf{y}[i] = \mathbf{A}[i]\boldsymbol{\theta}[i]$ $(i = 1, 2, \ldots, M)$ where $\mathbf{y}[i]$ denotes the i-th measurement sub-vector. In this way, all segments of the sparse signal $\bar{\boldsymbol{\theta}}$ can be reconstructed in parallel, and the signal can be recovered via $\hat{\bar{\mathbf{x}}} = \bar{\boldsymbol{\Psi}}\hat{\bar{\boldsymbol{\theta}}}$ where $\hat{\bar{\mathbf{x}}}$ and $\hat{\bar{\boldsymbol{\theta}}}$ are the reconstructed signal and its projection on corresponding sparsifying basis, respectively. In the block diagram shown in Fig. 4.1, the block $\mathbf{A}[i]$ represents a CS decoding processor. The inputs of the CS decoding processor are the measurement sub-vector $\mathbf{y}[i]$'s and the outputs are the reconstructed segments $\hat{\boldsymbol{\theta}}[i]$'s, which are then stacked in one vector $\hat{\bar{\boldsymbol{\theta}}}$.

Denote the sparsity level of projection $\bar{\boldsymbol{\theta}}$ as S, i.e., there are only $S \ll \bar{N}$ nonzero entries in $\bar{\boldsymbol{\theta}}$, and denote the sparsity level of $\boldsymbol{\theta}[i]$ as S_i. Thus, $\sum_{i=1}^{M} S_i = S$. It is known that in order to reconstruct exactly $\boldsymbol{\theta}[i]$, the sensing matrix $\mathbf{A}[i]$ needs to satisfy the RIP condition determined by S_i [9]. Smaller S_i indicates looser RIP condition, and thus K_i, that is, the number of measurements for $\boldsymbol{\theta}[i]$, can be smaller. Therefore, the design of the sensing matrix $\bar{\mathbf{A}}$ depends on the sparsity levels of the $\boldsymbol{\theta}[i]$'s.

4.1.1 Computational considerations

We briefly compare the computational complexity of the parallel CS reconstruction architecture with that of the architectures employing centralized reconstruction. Several solvers exist to reconstruct $\bar{\boldsymbol{\theta}} \in \mathbb{R}^{\bar{N}}$ from $\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}$, including for example, the basis pursuit (BP) algorithm based on interior point methods that have computational complexity $\mathcal{O}(\bar{N}^3)$ [27]. Therefore, for multidimensional signals, since $\bar{N}$ can be dramatically large, the reconstruction process becomes rather slow.

For the parallel CS reconstruction architecture in Fig. 4.1, each decoding processor only needs to reconstruct a segment of the sparse signal $\bar{\boldsymbol{\theta}}$. Thus, the computational complexity of the i-th decoding processor is only $\mathcal{O}(l_i^3)$, and the total computational complexity is $\mathcal{O}(\sum_{i=1}^{M} l_i^3)$. The total computational complexity is minimized to $\mathcal{O}(\bar{N}^3/M^2)$ when $l_i = l = \bar{N}/M$ for all i. Thus, by using the parallel CS reconstruction architecture, the computational complexity is much lower.

4.1.2 Permutation

To ensure that all decoding processors in Fig. 4.1 have the same configurations, we assume $l_i = l$ and $\mathbf{A}[i] = \mathbf{A}_0$ for all i. Therefore, to reconstruct exactly all $\boldsymbol{\theta}[i]$'s, $i = 1, 2, \ldots, M$, $\mathbf{A}_0$ needs to satisfy the RIP condition determined by $\max_i \{S_i\}$. However, considering the difference of sparsity levels among all $\boldsymbol{\theta}[i]$'s, $i = 1, 2, \ldots, M$, the above setting is not efficient because fewer measurements are actually needed for some segments with smaller S_i. This problem can be solved by applying permutation on $\boldsymbol{\theta}$ such that all segments have similar sparsity level. Then, the required number of measurements to achieve a given error performance can be reduced.

For a permutation π of $\bar{N}$ elements: $\{1, \ldots, \bar{N}\} \to \{1, \ldots, \bar{N}\}$ with $\pi(i)$ denoting the new index of the original i-th element after the permutation, its permutation matrix is $\mathbf{P}_\pi \in \mathbb{R}^{\bar{N} \times \bar{N}}$, whose entries are all 0 except that the $\pi(i)$-th entry in the i-th row is 1.

Fig. 4.2 describes the parallel CS reconstruction architecture with permutation on the projection $\bar{\boldsymbol{\theta}}$. At the encoder side, signal $\bar{\mathbf{x}}$ is first projected onto the sparsifying basis $\bar{\boldsymbol{\Psi}}$, which gives the projection $\bar{\boldsymbol{\theta}} = \bar{\boldsymbol{\Psi}}^T \bar{\mathbf{x}}$. Then permutation $\mathbf{P}_\pi$ on the projection $\bar{\boldsymbol{\theta}}$ is applied, which gives the permuted projection $\bar{\boldsymbol{\theta}}^\dagger = \mathbf{P}_\pi \bar{\boldsymbol{\theta}}$. The measurement vector $\bar{\mathbf{y}}^\dagger$ of permuted projection $\bar{\boldsymbol{\theta}}^\dagger$ is then given using the sensing matrix $\bar{\mathbf{A}}$ as $\bar{\mathbf{y}}^\dagger = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}^\dagger$. At the decoder side, all segments $\boldsymbol{\theta}^\dagger[i]$ can be reconstructed in parallel from $\mathbf{y}^\dagger[i] = \mathbf{A}[i]\boldsymbol{\theta}^\dagger[i]$ where $\mathbf{y}^\dagger[i]$ and $\boldsymbol{\theta}^\dagger[i]$ are the i-th segment of $\bar{\mathbf{y}}^\dagger$ and the i-th segment of $\bar{\boldsymbol{\theta}}^\dagger$, respectively. The inverse permutation is performed after parallel reconstruction of all segments of $\bar{\boldsymbol{\theta}}^\dagger$, i.e., $\hat{\bar{\boldsymbol{\theta}}} = \mathbf{P}_\pi^{-1} \hat{\bar{\boldsymbol{\theta}}}^\dagger = \mathbf{P}_\pi^T \hat{\bar{\boldsymbol{\theta}}}^\dagger$.

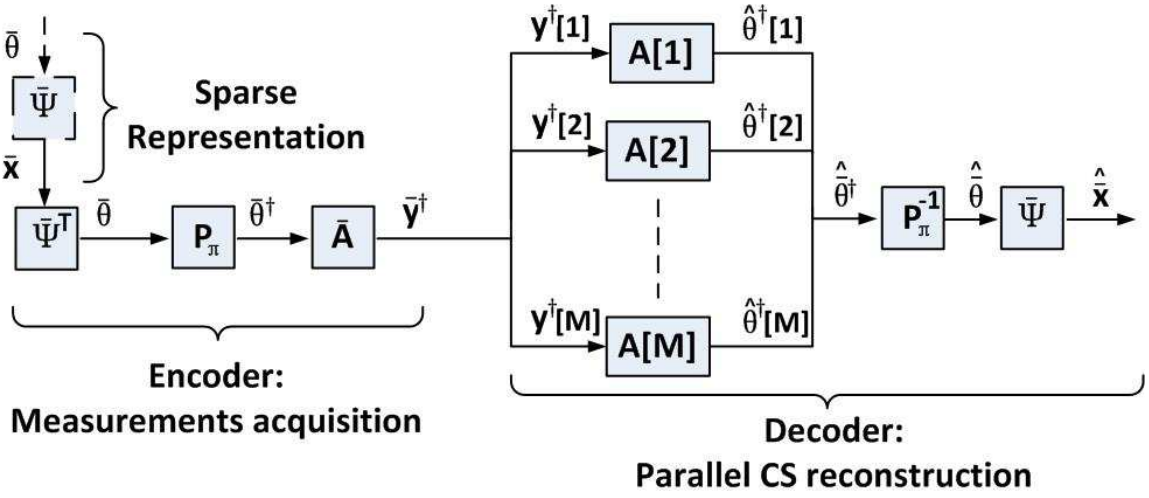


Fig. 4.2: Block diagram of the system employing the parallel CS reconstruction architecture with permutation on the projection $\bar{\boldsymbol{\theta}}$.

As shown in Fig. 4.2, to introduce permutation into the parallel CS reconstruction architecture, the digital projection $\bar{\boldsymbol{\theta}}$ is obtained during the measurements acquisition process, which requires a high-rate sampling of $\bar{\mathbf{x}}$ if $\bar{\mathbf{x}}$ is an analog signal. However, one major advantage of CS is the ability to avoid the high-rate sampling. Besides, the permutation has to be done during the measurements acquisition process, which increases the workload on the encoder. To solve the above two problems, we have the following lemma.

Lemma 5: *Consider a signal* $\bar{\mathbf{x}} \in \mathbb{R}^{\bar{N}}$*, its projection* $\bar{\boldsymbol{\theta}}$ *onto the sparsifying basis* $\bar{\boldsymbol{\Psi}}$*, a sensing matrix* $\bar{\mathbf{A}}$*, and a permutation matrix* $\mathbf{P}_\pi$*. The measurements acquired in Fig. 4.2, i.e.,* $\bar{\mathbf{y}}^\dagger = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}^\dagger$*, can also be acquired by sampling* $\bar{\mathbf{x}}$ *directly using a measurement matrix* $\bar{\boldsymbol{\Phi}}^\dagger$ *given by* $\bar{\boldsymbol{\Phi}}^\dagger = \bar{\mathbf{A}}\mathbf{P}_\pi\bar{\boldsymbol{\Psi}}^T$.

Proof. We have the following observation

$$\bar{\mathbf{y}}^\dagger = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}}^\dagger = \bar{\mathbf{A}}(\mathbf{P}_\pi\bar{\boldsymbol{\Psi}}^T\bar{\mathbf{x}}) = (\bar{\mathbf{A}}\mathbf{P}_\pi\bar{\boldsymbol{\Psi}}^T)\bar{\mathbf{x}}. \tag{4.2}$$

Therefore, the same measurements $\bar{\mathbf{y}}^\dagger$ can be given using a new measurement matrix $\bar{\boldsymbol{\Phi}}^\dagger = \bar{\mathbf{A}}\mathbf{P}_\pi\bar{\boldsymbol{\Psi}}^T$. □

Based on Lemma 5, the encoder in Fig. 4.2 can be replaced by the encoder shown in Fig. 4.3, which avoids the high-rate sampling. In addition, since the new measurement matrix $\bar{\boldsymbol{\Phi}}^\dagger$ is pre-generated and stored at the encoder side, no permutation process is actually required at the encoder side to obtain $\bar{\mathbf{y}}^\dagger$. The only difference between the encoder in

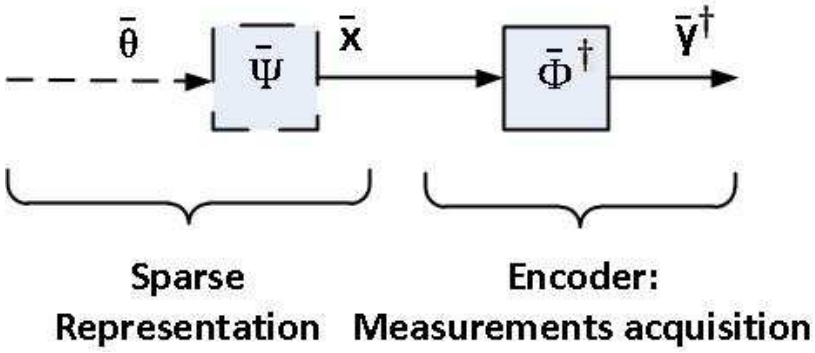


Fig. 4.3: Block diagram of the encoder that is equivalent to the encoder in Fig. 4.2.

Fig. 4.1 and that in Fig. 4.3 is the measurement matrix $\bar{\mathbf{\Phi}} = \bar{\mathbf{A}}\bar{\mathbf{\Psi}}$ in Fig. 4.1 and the new measurement matrix $\bar{\mathbf{\Phi}}^{\dagger}$ in Fig. 4.3..

Denote the sparsity level of $\boldsymbol{\theta}^{\dagger}[i]$ as $S_i^{\dagger}$. Recall that we assume $l_i = l$, $\mathbf{A}[i] = \mathbf{A}_0$. If $\max_i\{S_i^{\dagger}\} < \max_i\{S_i\}$, the required RIP condition for $\mathbf{A}_0$ to reconstruct $\bar{\boldsymbol{\theta}}^{\dagger}$ is weaker than that to reconstruct $\bar{\boldsymbol{\theta}}$. Thus, with permutation on $\bar{\boldsymbol{\theta}}$, fewer measurements are needed to achieve the same reconstruction error performance. Note that the exact positions of nonzero entries of $\bar{\boldsymbol{\theta}}$ are not known, and optimal permutation which results in uniform sparsity levels among all segments of $\bar{\boldsymbol{\theta}}^{\dagger}$ is not practical. Thus, permutation design in practice must be based on a sparsity model of the projection $\bar{\boldsymbol{\theta}}$, as introduced in Chapter 3 for video compression application. If no sparsity model is known, the best choice of permutation is a random permutation, which we consider here. Since we assume $l_i = l$, $\bar{\boldsymbol{\theta}}$ can be rewritten as a matrix $\mathbf{\Theta} \in \mathbb{R}^{l \times M}$ by letting $\boldsymbol{\theta}[i]$ be the i-th column of $\mathbf{\Theta}$. Let C_i be the number of nonzero entries in $\mathbf{v}_i$, which is the i-th row of $\mathbf{\Theta}$. There are $M!$ different permutations that can be applied to $\mathbf{v}_i$. With a random permutation from the $M!$ possibilities, a nonzero entry in $\mathbf{v}_i$ is permuted to the columns of $1, 2, \ldots, M$ with equal probability of $1/M$. Therefore, considering the permuted i-th row, denoted as $\mathbf{v}_i^{\dagger}$, the average number of nonzero entries in every column is C_i/M. Assuming that the permutations applied to different rows are independent from each other, the average number of nonzero entries in each column of the resulted matrix is $\sum_{i=1}^{l} C_i/M$. Therefore, in average, every column of the resulted matrix has the same sparsity level. For example, if the projection $\bar{\boldsymbol{\theta}}$ has length 1000 and we set $l = 100$ and $M = 10$, for a randomly generated sparse signal $\mathbf{\Theta}$ with $S = 60$ nonzero entries, after random permutation, the mean and standard deviation of the sparsity level of each column obtained via 10^5 trails are 6 and 2.28, respectively. Therefore, random permutation results in an acceptable sparsity distribution among segments.

4.1.3 Application of the parallel CS reconstruction architecture

In most existing CS acquisition devices, the measurement matrix is pre-generated and stored in the encoder. The decoder "stores" a corresponding sensing matrix for reconstruction. The reconstruction in CS is known to have high computational complexity compared to the sampling process, especially when the dimension of the signal is very large. Thus, when the computational complexity and time for reconstruction is a crucial evaluation criterion and centralized sampling is acceptable, parallel CS reconstruction is very useful. For example, the compressive imaging via coded aperture architecture proposed in [8] uses a pre-designed random coded aperture as the measurement matrix; the single-pixel camera uses a digital micromirror device array with pre-designed random patterns as the measurement matrix [7]. For such acquisition devices, the parallel CS reconstruction architecture can be applied immediately by designing the measurement matrix as described above in this section.

4.2 Simulation results

We compare the error performance and the reconstruction time among three different schemes: the centralized CS reconstruction, i.e., $M = 1$; the parallel CS reconstruction, i.e., $M \geq 2$, without permutation; and the parallel CS reconstruction with permutation. The reconstruction time includes the sum of reconstruction time of the decoding processors, as well as the average reconstruction time and the worst reconstruction time of the decoding processors.

Our simulations are performed using Matlab on a laptop computer with Intel Dual Core CPU at 2.70 GHz and 8 GB of memory. The sparse projection $\bar{\boldsymbol{\theta}}$ is a random binary sequence of length $\bar{N} = 1200$ with $S = 60$ nonzero entries, which are randomly distributed across the signal. To ensure that all decoding processors have the same configuration, we set $l_i = l$ and $\mathbf{A}[i] = \mathbf{A}_0 \in \mathbb{R}^{K_0 \times l}$ for all i, where $K_0 = K/M$. Entries of $\mathbf{A}_0$ are drawn from Gaussian ensembles with variance $1/K_0$. The reconstruction algorithm that we use in each decoding processor is the BP algorithm. The goal of our simulation is to show the maximum improvement that can be brought by introducing permutation. So, $\mathbf{P}_\pi$ is selected to ensure that all segments of $\bar{\boldsymbol{\theta}}^\dagger$ have the same sparsity level, i.e., $S_i^\dagger = S/M$ for all i, although such permutation may not be practical. We set M to 1, 2, 3, 4, 5 and 10. For $M = 1$, the parallel CS reconstruction boils down to the centralized CS reconstruction. In our simulation, we run 500 trials for each combination of (M, K) and average the results over the trails.

Fig. 4.4 and Fig. 4.5 shows the mean square error (MSE) (normalized to the signal en-

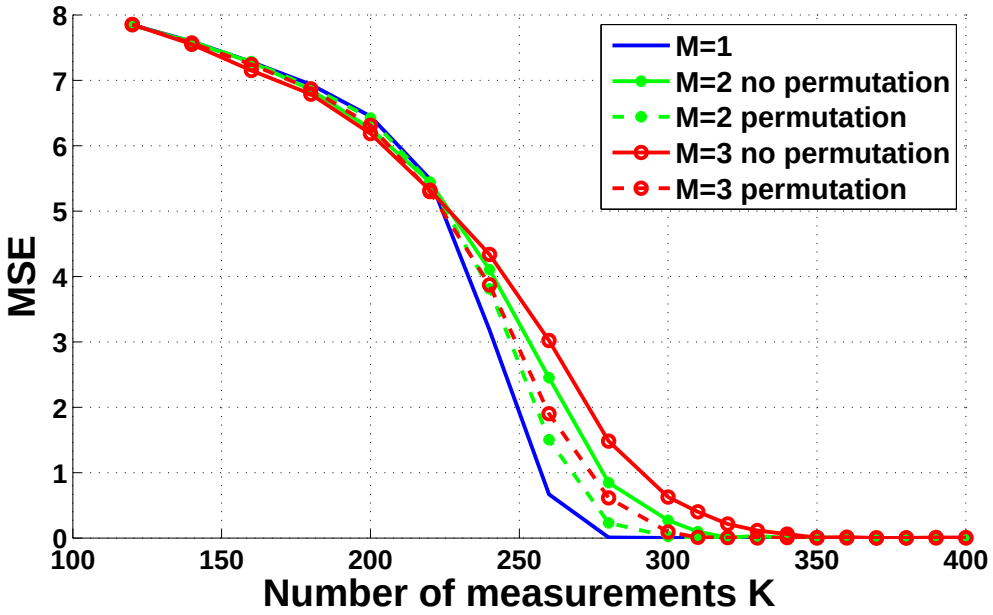


Fig. 4.4: Reconstruction error performance vs. the number of measurements K for $M = 1, 2, 3$.

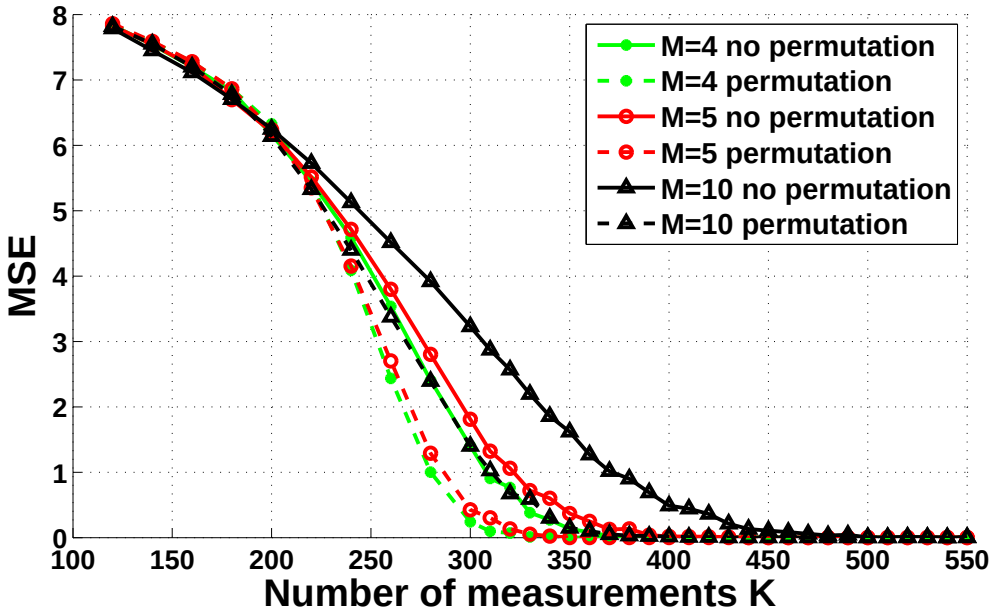


Fig. 4.5: Reconstruction error performance vs. the number of measurements K for $M = 4, 5, 10$.

Table 4.1: Minimum number of measurements required for exact reconstruction and corresponding reconstruction time (in seconds).

M	Permutation	K	t_{total}	t_{average}	t_{worst}
1	N/A	340	5.9085±2.5978	5.9085	5.9085
2	No	370	1.5741±0.0900	0.7870	0.8701
2	Yes	330	1.4208±0.0928	0.7104	0.7819
3	No	380	1.0059±0.0990	0.3353	0.4178
3	Yes	330	0.9331±0.1021	0.3110	0.3861
4	No	400	0.8889±0.2012	0.2222	0.3104
4	Yes	360	0.8801±0.1255	0.2200	0.3090
5	No	450	0.9549±0.1614	0.1910	0.2833
5	Yes	350	0.9222±0.0847	0.1844	0.2739
10	No	560	1.1764±0.1596	0.1176	0.1967
10	Yes	430	1.1771±0.1436	0.1177	0.1944

ergy) of the reconstructed signal in the three aforementioned schemes versus the number of measurements. It can be seen that the MSE for a fixed number of measurements increases as the number of segments M increases. In other words, the minimum number of measurements required for exact reconstruction increases as M increases. It is reasonable since the required number of measurements per segment does not linearly decrease when M increases. Besides, it is shown in Fig. 4.4 and Fig. 4.5 that for a fixed K, the MSE can be reduced with the permutation. The minimum number of measurements required for exact reconstruction is also reduced with the permutation.

Table 4.1 shows the minimum number of measurements and reconstruction time required for exact reconstruction. The total reconstruction time t_{total} is the time used to reconstruct all segments, and we show its mean and standard deviation in the table. The average reconstruction time $t_{\text{average}} = t_{\text{total}}/M$ is the average time used to reconstruct each segment. The worst reconstruction time t_{worst} is the maximal reconstruction time used to reconstruct the 'worst' segment, given as $t_{\text{worst}} = \max_i \{t_i\}$ where t_i denotes the reconstruction time for the i-th segment. From Table 4.1, the total reconstruction time decreases as M increases from 1 to 4. When M further increases, the total reconstruction time may increase. This is because more measurements are required for exact recovery. However, the total reconstruction time, and the average and the worst reconstruction time of the parallel CS reconstruction are much less than those of the centralized CS reconstruction.

4.3 Chapter summary

The parallel reconstruction architecture for CS has been proposed, where all segments of the projection of a signal onto the sparsifying basis can be reconstructed in parallel. Both the analysis and simulations have shown that via the parallel CS reconstruction, the computational complexity and the reconstruction time can be reduced significantly. Permutation has been employed to reduce the minimum number of measurements required for exact reconstruction in the parallel CS reconstruction architecture. It is also proved that the permutation can be implicitly done via using a permuted version of the measurement matrix. Consequently, the measurements of the permuted projection of the signal can be obtained by directly sampling the signal without any permutation being actually performed.

Chapter 5

Conclusion

In this chapter, a summary of the contributions of the thesis are given. In addition, the future work that can be done based on the thesis is discussed.

5.1 Summary of contributions

In this thesis, two CS architectures have been proposed to sample multidimensional signals. Both of them help to reduce the complexity of the sampling process and/or the reconstruction process.

In the parallel CS architecture proposed in Chapter 3, both the sampling process and the reconstruction process can be parallelized. Thus, it significantly reduces the storage requirements and computational complexity. The RIP condition on the sensing matrix in the parallel CS architecture has been discussed. Moreover, the acceptable permutation has been shown to enable relaxation of the RIP condition on the sensing matrix. Several acceptable permutations have been proposed in Chapter 3, including the group-scan-based, the zigzag-scan-based and the block-test-based permutations. The group-scan-based permutation has provided a guideline for designing acceptable permutations. The zigzag-scan-based permutation has been shown to be an acceptable permutation for 2D signals satisfying the (r_0, r_1, r_2, α)-layer model, such as the DCT2 coefficient matrix of 2D images. The block-test-based permutation has been proposed for block sparse signals. As an example, a video compression scheme that uses the parallel CS architecture with permutation has been proposed. In the simulations, the zigzag-scan-based permutation has been shown to be an acceptable permutation for the DCT2 coefficient matrix of 2D images. Besides, the block-test-based permutation has been shown to be an acceptable permutation for both the DCT2 coefficient matrix of 2D images and the difference of two frames of a video. The simulation results for the video compression scheme that is based on the parallel CS architecture with

the zigzag-scan-based permutation and the block-test-based permutation have shown that the zigzag-scan-based permutation improves significantly the reconstruction performance of the reference frames in a video in terms of PSNR, whereas the block-test-based permutation improves, though insignificantly, the reconstruction performance of the non-reference frames in a video in terms of PSNR.

The permutation in the parallel CS architecture has to be performed before the signal being sampled in parallel and it requires the projection of the signal onto the sparsifying basis. This issue has been addressed in Chapter 4. In the parallel CS reconstruction architecture proposed in Chapter 4, the sampling process is not parallelized. It enables the permutation to be performed implicitly during the sampling process via using a permuted version of the measurement matrix. In this way, the measurements of the permuted projection can be obtained by directly sampling the signal without any permutation being actually performed. The computational complexity for the reconstruction process in the parallel CS reconstruction architecture has been analyzed. The simulation has been also performed to compare the empirical performance of the centralized CS reconstruction architecture, the parallel CS reconstruction architecture without permutation, and the parallel CS reconstruction architecture with permutation. It has been shown that the reconstruction time can be reduced significantly as the number of segments increases. It has also been shown that although the minimum number of measurements required for exact reconstruction increases as the number of segments increases, the permutation helps to reduce this number.

5.2 Future work

5.2.1 Extension of permutation

The group-scan-based permutation gives only a guideline for designing an acceptable permutation. It is more challenging to develop an analytical result similar to that derived for the zigzag-scan-based permutation in Proposition 1 in Chapter 3 since there is no prior information about the positions of nonzero entries. Even knowing that the signal is block sparse does not help for analyzing the probability that the block-test-based permutation is acceptable. So more theoretical analysis about this permutation can be done in future using a different machinery from the one used in the thesis for the zigzag-scan-based permutation analysis. Moreover, it is possible to adaptively choose permutation while sampling a stream of signals by estimating the sparsity level of each segment in the signal.

5.2.2 Practical video coding scheme

In this thesis, a rough video compression scheme has been proposed and studied. However, the practical video coding scheme has much more details to be considered, for example, the quantization, the source encoding and the channel encoding, etc. Besides, the inter-frame coding scheme can be more powerful by introducing, for example, motion estimation and compensation. Thus, it is interesting to design a practical video coding scheme based on the parallel CS architecture or the parallel CS reconstruction architecture with permutation as a practical extension of the results of this thesis.

5.2.3 Optimal segment size for the parallel CS reconstruction architecture

From the simulation example in Chapter 4, it can be noticed that there should exist an optimal value for the number of segments or the size of each segment, considering the trade-off between the reconstruction time and the required number of measurements. Thus, the optimal segment size can be studied in future. For example, the reconstruction time can be formulated as a function of the segment size, and the required number of measurements can be given via information theoretical analysis.

Bibliography

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.

[2] E. J. Candès, "Compressive sampling," in *Proc. Int. Cong. Math.*, Madrid, Spain, Aug. 22–30, 2006, pp. 1433–1452.

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

[4] 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.

[5] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[6] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. Picture Coding Symp.*, Beijing, China, Apr. 24–26, 2006.

[7] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[8] A. Stern and B. Javidi, "Random projections imaging with extended space-bandwidth product," *J. Display Technology*, vol. 3, no. 3, pp. 315–320, Sep. 2007.

[9] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4214, Dec. 2005.

[10] ——, "The Dantzig selector: statistical estimation when p is much larger than n," *Ann. Statist.*, vol. 35, no. 6, pp. 2313–2351, 2007.

[11] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9–10, pp. 589–592, May 2008.

[12] A. Cohen, W. Dahmen, and R. Devore, "Compressed sensing and best k-term approximation," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 211–231, Jul. 2008.

[13] M. Rudelson and R. Vershynin, "On sparse reconstruction from fourier and gaussian measurements," *Comm. Pure Appl. Math.*, vol. 61, no. 8, pp. 1025–1045, 2008.

[14] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proc. IEEE Workshop Stat. Signal Process.*, Madison, WI, USA, Aug. 26–29, 2007, pp. 294–298.

[15] T. T. Cai, G. Xu, and J. Zhang, "On recovery of sparse signals via ℓ_1 minimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3388–3397, Jul. 2009.

[16] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[17] M. A. Davenport and M. B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4395–4401, Sep. 2010.

[18] E. Liu and V. N. Temlyakov, "The orthogonal super greedy algorithm and applications in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2040–2047, Apr. 2012.

[19] Q. Mo and Y. Shen, "A remark on the restricted isometry property in orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3654–3656, Jun. 2012.

[20] J. Wang and B. Shim, "Improved recovery bounds of orthogonal matching pursuit using restricted isometry property," *arXiv: 1211.4293v1*, Nov. 2012.

[21] 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.

[22] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008.

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

[24] E. J. Candès, 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.

[25] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[26] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[27] S. Boyd and L. Vanderberghe, *Convex Optimization*. New York: Cambridge Univ. Press, Dec. 2008.

[28] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, California, USA, Nov. 1–3, 1993, pp. 40–44.

[29] D. L. Donoho, Y. Tsaig, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[30] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Computational Math.*, vol. 9, no. 3, pp. 317–334, Jun. 2009.

[31] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. and Computational Harmonic Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[32] 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. 4–7, 2012, pp. 1925–1929.

[33] ——, "Permutation meets parallel compressed sensing: How to relax restricted isometry property for 2D sparse signals," *arXiv:1305.4980*, May 2013.

[34] P. Nagesh and B. Li, "Compressive imaging of color images," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 1261–1264.

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

[36] L. Gan, "Block compressed sensing of natural images," in *Proc. Int. Conf. Digital Signal Process.*, Cardiff, UK, Jul. 1–4, 2007, pp. 403–406.

[37] S. Mun and J. Fowler, "Block compressed sensing of images using directional transforms," in *Proc. Int. Conf. Image Process.*, Cario, Egypt, Nov. 7–10, 2009, pp. 3021–3024.

[38] 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. 25–29, 2008.

[39] E. J. Candès and J. K. Romberg, "Signal recovery from random projections," in *Proc. SPIE-IS&T Electronic Imaging*, San Jose, CA, USA, Jan. 17–18, 2005, pp. 76–86.

[40] B. Han, F. Wu, and D. Wu, "Image representation by compressed sensing," in *Proc. 15th IEEE Int. Conf. Image Process.*, San Diego, California, USA, Oct. 12–15, 2008, pp. 1344–1347.

[41] 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.

[42] Y. Kim, M. S. Nadar, and A. Bilgin, "Wavelet-based compressed sensing using a Gaussian scale mixture model," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3102–3108, Jun. 2012.

[43] V. Stanković, L. Stanković, and S. Cheng, "Compressive sampling of binary images," in *Proc. Cong. on Image and Signal Process.*, Sanya, Hainan, China, May 27–30, 2008, pp. 7–11.

[44] ——, "Compressive video sampling," in *Proc. 16th European Signal Process. Conf.*, Lausanne, Switzerland, Aug. 25–29, 2008, pp. 2–6.

[45] 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.

[46] 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, Jul. 2005.

[47] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4634–4643, Dec. 2006.

[48] E. van den Berg and M. P. Friedlander, "Theoretical and empirical results for recovery from multiple measurements," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2516–2527, May 2010.

[49] 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, 2011.

[50] 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.

[51] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[52] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[53] Y. C. Eldar and M. Mishali, "Block sparsity and sampling over a union of subspaces," in *Proc. Int. Conf. Digital Signal Process.*, Santorini, Greece, Jul. 5–7, 2009, pp. 1–8.

[54] 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, Jun. 2012.

[55] H. Fang, S. A. Vorobyov, and H. Jiang, "Parallel compressed sensing reconstruction," *IEEE Signal Process. Lett.*, submitted.

[56] A. Eftekhari, H. L. Yap, C. J. Rozell, and M. B. Wakin, "The restricted isometry property for random block diagonal matrices," *arXiv:1210.3395v1*, Oct. 2012.

[57] J. Y. Park, H. L. Yap, C. J. Rozell, and M. B. Wakin, "Concentration of measure for block diagonal matrices with applications to compressive signal processing," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5859–5875, Dec. 2011.

[58] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulatoin," in *IEEE Int. Symp. Circuits and Systems*, New Orleans, LA, May27–30, 2007, pp. 1959–1962.