

EFFICIENT CIRCULANT MATRIX CONSTRUCTION AND IMPLEMENTATION FOR COMPRESSED SENSING

Feng Yi^{1,2,3}, Zaichen Zhang^{2,3}, Xiaohu You², and Chuan Zhang^{1,2,3,}*

¹Lab of Efficient Architectures for Digital-communication and Signal-processing (LEADS)

²National Mobile Communications Research Laboratory, Southeast University

³Quantum Information Center of Southeast University, Nanjing, China

Email: {fengyi, zczhang, xhyu, chzhang}@seu.edu.cn

ABSTRACT

The design of measurement matrices is an important part in compressed sensing (CS). Random matrices superior to incoherence are considered to be optimal measurement matrices to achieve successful recovery. However, they are deficient in memory cost. Structure matrices like circulant matrices are preferred for low-memory cost. Nevertheless, their recovery performance is greatly damaged because of element coherence. In this paper, a new method called different-spaced selection & different-spaced flipping (DSS & DSF) is proposed to modify structure matrices. Based on circulant matrices, regular extraction and symbol flipping imposed on columns of measurement matrices can increase randomness to a large scale. As a result, not only near optimal recovery but also much less memory cost can be achieved. Compared with Gaussian random matrices, the memory cost can be reduced to 4% when measurement matrices based on circulant matrices are in 128×512 dimensions. An efficient hardware design and VLSI implementation are also presented at the end of this paper.

Index Terms— Measurement matrix, compressed sensing (CS), structure matrix, memory cost, hardware implementation.

1. INTRODUCTION

The reasonable design of measurement matrices plays a significant role in compressed sensing (CS) [1] framework, because the product of the measurement matrix and basis matrix has to satisfy restricted isometry property (RIP), which guarantees a successful recovery. Random matrices such as Gaussian random matrices have been proved to meet the criterion and have low coherence [2]. However, they cost too much memory space to store all elements when implemented in hardware. Structured matrices [3] are then proposed to replace random matrices to make up memory cost, but they will lead to the recovery performance decline.

In this paper, to enhance the performance decline caused by high coherence, a more efficient method is proposed to modify structured matrices, whose aim is to scramble structured matrices and break up their internal structures. After the application, both near optimal recovery performance and 96% memory space reduce can be realized. A more detailed comparison among the sign-flipped scrambling method, extended-select scrambling method [4] and the proposed DSS & DSF method is provided. In addition, the architecture designs and VLSI implementation are introduced to make comparisons.

The remainder of this paper is organized as follows. Section II will introduce the preliminaries of CS theory, the characteristic of

structured matrices. Section III discusses the proposed method and its simulation results compared with other methods. The architecture design and VLSI implementation results are analysed in Section IV. Finally, Section V concludes the entire paper.

2. BACKGROUND

2.A. Preliminaries of CS

Signal sparsity is a key ingredient in CS framework. Generally, an input signal \mathbf{x} can be spread by basis matrix Ψ and coefficient vector α . Thus, almost all signals could be recognized as sparse by spreading under sparse basis matrix as follows. Both $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ and $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ are $N \times 1$ vectors, and basis matrix $\Psi = [\Psi_1, \Psi_2, \dots, \Psi_N]$ is in $N \times N$ dimensions.

$$\mathbf{x} = \sum_{i=1}^N \Psi_i \alpha_i = \Psi \alpha. \quad (1)$$

The signal \mathbf{x} is k -sparse, that means, there are k elements in the coefficient vector α are non-zero at most. Thus, the sparse signal \mathbf{x} can achieve recovery from the measurement vector \mathbf{y} and measurement matrix Φ . And its mathematical linear expression is presented as Eq. (2), where the measurement matrix Φ is $M \times N$ dimensions ($M < N$). The product of measurement matrix Φ and basis matrix Ψ is denoted by sensing matrix Θ , which is $M \times N$ in dimensions.

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \alpha = \Theta \alpha. \quad (2)$$

To ensure a successful recovery, there are two criterions for measurement matrix to meet. At first, measurement matrix Φ and basis matrix Ψ should satisfy the condition of low coherence. The coherence [5] between the measurement matrix Φ and the basis matrix Ψ comes to

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{1 \leq k, j \leq N} |\langle \phi_k, \psi_j \rangle|. \quad (3)$$

The equation requires vectors in matrix Φ are orthogonal to vectors in matrix Ψ . That is to say the value μ should be small enough correspondingly.

Secondly, the sensing matrix and k -sparse signal should meet the restricted isometry property (RIP). Thus, the ability of successful recovery is guaranteed. In that case, a matrix meets the criterion of low coherence when satisfying RIP [6], which is defined as

$$(1 - \delta_k) \|\alpha\|_2^2 \leq \|\Theta \alpha\|_2^2 \leq (1 + \delta_k) \|\alpha\|_2^2. \quad (4)$$

more interference is added to matrix Φ_s by having some columns for symbol flipping. Similar to previous operation, the columns chosen to be flipped are at the intervals I changing from 1 to 9. Repeat until the all the columns in matrix Φ_{fn} that satisfy the rules are processed. Matrix F ($N \times N$) is used to show the interference.

$$\Phi_{fn} = \Phi_s F. \quad (9)$$

The operation of symbol flipping can be achieved easily by multiply -1 , which looks like a simple operation but actually has a great effect because it breaks the inherent structure to a much larger degree.

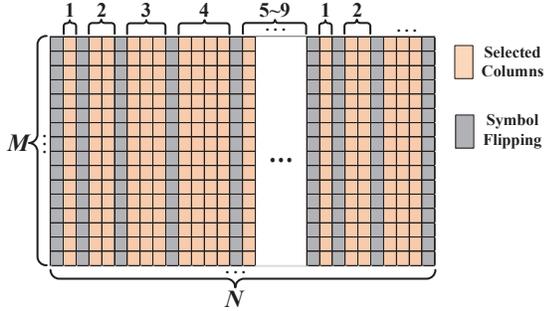


Fig. 3. Illustration of symbol flip rules for measurement matrix.

The final measurement matrix Φ_{fn} is optimized after applying DSS & DSF method. With these modifications, Φ_{fn} is much more complex than original circulant matrix apparently. All the operations can be expressed in the following mathematical expression in the end.

$$\Phi_{fn} = \Phi_{ext.cir} SF. \quad (10)$$

3.B. Simulation Results Comparison

In order to better describe our experiment, parameter settings are claimed as follows. M and N are set to be 128 and 512, respectively. The input signal x is generated randomly and the coefficient vector α has the property of k -sparse and k increases at the interval 3 ($k \in [1, 50]$). The IDCT matrix is employed as the basis matrix Ψ . Since OMP algorithm [8] has been used widely in signal recovery, all the simulation results discussed below are based on it.

$$y = \Phi_{fn} x = \Phi_{ext.cir} SF \Psi \alpha. \quad (11)$$

As we explained previously, Eq. (2) can be rewritten into the format of Eq. (11) after the measurement matrix Φ_{fn} is modified by DSS & DSF. Fig. 4 shows different SSR curves obtained by corresponding measurement matrix. SFS stands for measurement matrix modified by sign-flipped scrambling method and ESS is on behalf of measurement matrix modified by extended-select scrambling method. SFS+ESS is the combination of both two methods. Moreover, symbol I represents space between modified columns. Measurement matrices modified with the same method but different-spaced intervals are also compared in the simulation.

It is easy to notice from the Fig. 4 that all of the modification-s work since their SSR are higher than circulant matrix's, because they increase the randomness of the measurement matrices in different extent. Measurement matrix based on circulant matrix or

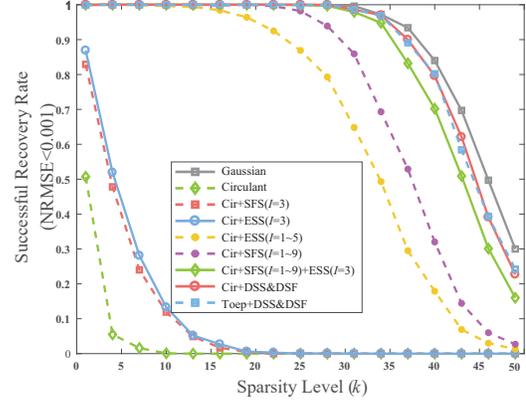


Fig. 4. Reconstruction performance comparison of proposed method and other methods with $M \times N = 128 \times 512$.

Toeplitz matrix has a remarkable effect after applying the recommended method, and their recovery performances are almost as good as Gaussian random matrix's. Thus, the proposed method also works on Toeplitz matrix. Moreover, measurement matrix with SFS+ESS comes secondly and circulant matrix without any changes certainly performs the worst.

4. HARDWARE ARCHITECTURE AND VLSI IMPLEMENTATION

4.A. Hardware Architecture

Circulating left-shift register arrays (RA) are used to design the recommended $M \times N_{ext}$ measurement matrix imposed on DSS & DSF method. A number of N_{ext} elements generated by random sequence are fed to the register arrays at the beginning. For each row produced, the elements in left-shift RAs rotate to the left once until M rows are all produced and the RAs are driven by clock signals. This property enables us have simple implementation as we did for LD-PC codes [9–12]. In addition, a condition check logic is needed to control the MUXs, deciding whether the elements selected from RAs are output directly or modified by multiplying -1 . Finally, the entire measurement matrix can be obtained from MUXs.

Because columns in $\Phi_{ext.cir}$ to be chosen are at repeated regular intervals changing from 1 to 5, the relationship between the two column ordinals is determined. Assuming that the symbol i stands for the column ordinal in Φ_s and symbol j stands for the column ordinal in $\Phi_{ext.cir}$, then

$$j = \lfloor \frac{i}{5} \rfloor \times 20 + \frac{(i\%5) \times (i\%5 + 1)}{2}. \quad (12)$$

MUXs along with not-gates can be anchored below the corresponding register arrays to select the columns satisfied the relationship in Eq. (12). Similarly, due to the intervals between columns increasing from 1 to 9, there is also a determined relationship between i and column ordinal p in Φ_{fn} ,

$$i = \lfloor \frac{p}{9} \rfloor \times 54 + \frac{(p\%9) \times (p\%9 + 1)}{2}. \quad (13)$$

A counter is necessary to help achieve sign flipping operation. When the counter equals to column ordinal j that satisfy the relationship in Eq. (13), the condition check logic will have the columns

output after multiplying -1 , otherwise outputting directly. Thus, the architecture achieves the proposed DSS & DSF method successfully. Fig. 5 shows the functional hardware architecture in detail. Considering both flipping rule and elements utilization ratio, it is reasonable to set N_{ext} as $4N$.

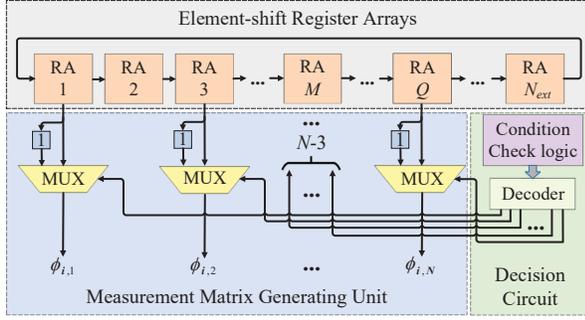


Fig. 5. Proposed architecture for generating measurement matrix based on circulant matrix with DSS & DSF.

4.B. Analysis of Memory Cost

According to conventional hardware architecture [13], to realize a generation device for Gaussian random matrix, the element memory device consists of M sub-memories. In this paper, the sub-memory is implemented by a ROM and its size is $N \times a$, and each ROM stores a specific row of the measurement matrix.

As analysed before, circulant matrix without any changes cost the least memory and the modified circulant matrix comes the second which is a little bit higher than original circulant matrix. There is no doubt that storing a Gaussian random matrix costs much greater than both of them.

Table 1. Comparison of Memory Cost and SSR.

Matrix type	Memory cost	Reduction		SSR	
		$(M, N) = (128, 512)$	$k = 25$	$k = 40$	
Circulant matrix	$N \times a$	99.2%	0	0	
Cir+SFS+DSS	$N_{ext} \times a$	96.1%	0.9986	0.6952	
Cir+DFS+DSS*	$N_{ext} \times a$	96.1%	1	0.7970	
Gaussian matrix	$M \times N \times a$	---	1	0.8390	

The table above exhibits the comparison of different matrix types on memory cost, reduction compared to Gaussian random matrix and recovery performance. Supposing that storing an element needs a bits, the memory cost of storing an $M \times N$ Gaussian random matrix needs $M \times N \times a$ bits in all while to output an $M \times N$ circulant matrix only requires $N \times a$ bits memory space by using the hardware architecture mentioned above. Although generating a matrix Φ_{fin} needs to store N_{ext} elements, which is 25% higher than basic circulant matrix, but it brings recovery performance improved greatly. Compared to Gaussian random matrix, it still saves almost 96% memory space. The measurement matrix imposed on SFS+DSS costs memory space as much as that with DFS+DSS, but its SSR performs worse. Overall, the proposed method comes the best and is more practical to realize.

4.C. Synthesis Results and Comparisons

The architecture designs are implemented in ISE Design Suite 14.7 compiler tools. Due to the limitations of compiler, the dimension of 128×512 will be hard for it to compile. Thus, a 16×64 measurement matrix is preferred to show the VLSI implementation and synthesis results. For the sake of accurate reconstruction [14], at least 6 bits are needed to value an element. Therefore, a is set as 6 bits in this VLSI implementation.

Table 2 presents the synthesis results. The frequency is set as 100 MHz in this implementation. In memory-based architecture, the storage of elements for Gaussian random matrix is 16 ROMs and the size of each ROM is width \times depth = 6×64 . However, there is little difference of the storage cost between the comparable SFS+DSS matrix and proposed DFS+DSS matrix. What's more, the number of LUTs used by proposed matrix is almost half of comparable matrix's. In addition to the placement of component, the architecture between the proposed matrix and comparable matrix [4] looks similar to each other, but the simulation results and implementation comparisons show the proposed matrix comes to the best among other matrix mentioned in this paper.

Table 2. Synthesis Results of Different Measurement Matrices.

	Gaussian Random Matrix	Comparable SFS+DSS Matrix	Proposed DFS+DSS Matrix
Architecture	Memory-based architecture	Similar architecture	Proposed architecture
Storage for elements	16 ROMs	765 RAs	857 RAs
Frequency	100 MHz	100 MHz	100 MHz
LUT	—	821	499

5. CONCLUSION

The quality of the measurement matrix's design largely determines the recovery performance of signal. Measurement matrix imposed on proposed DSS & DSF method works out remarkable compared to other methods in the simulation. Numerical results show that its recovery performance is very close to the performance of Gaussian random matrix. Moreover, nearly 96% memory space is saved when generating the proposed matrix by element shift register arrays compared to conventional Gaussian random matrix. What's more, the implementation cost between comparable SFS+DSS matrix and proposed matrix are not statistically significant. Simulation results and implementation comparisons show the proposed matrix comes to the best among other matrix mentioned in this paper. Future work will focus on more substitutional matrices or better hardware designs.

Acknowledgement

This work is supported in part by NSFC under grant 61501116, Jiangsu Provincial NSF under grant BK20140636, Huawei HIRP Flagship under grant YB201504, the Fundamental Research Funds for the Central Universities, the SRTP of Southeast University, State Key Laboratory of ASIC & System under grant 2016KF007, ICRI for MNC, and the Project Sponsored by the SRF for the Returned Overseas Chinese Scholars of MoE.

References

- [1] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.
- [2] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [3] W. Yin, S. Morgan, J. Yang, and Y. Zhang, "Practical compressive sensing with toeplitz and circulant matrices," in *Proc. VCIP*, 2010.
- [4] Y.-M. Lin, J.-F. Zhang, J. Geng, and A.-Y. A. Wu, "Structural scrambling of circulant matrices for cost-effective compressive sensing," *J. Signal Process. Syst.*, pp. 1–13, 2016.
- [5] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Probl.*, vol. 23, no. 3, p. 969, 2007.
- [6] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [7] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, 2007.
- [8] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [9] C. Zhang, L. Li, J. Lin, and Z. Wang, "Low-complexity shift-LDPC decoder for high-speed communication systems," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2008, pp. 1636–1639.
- [10] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 1, pp. 116–124, 2010.
- [11] C. Zhang and J. Sha, "Efficient network for non-binary QC-LDPC decoder," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, pp. 2617–2620.
- [12] C. Zhang and K. K. Parhi, "A network-efficient nonbinary QC-LDPC decoder architecture," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 6, pp. 1359–1371, 2012.
- [13] S. Pfetsch, T. Ragheb, J. Laska, H. Nejati, A. Gilbert, M. Strauss, R. Baraniuk, and Y. Massoud, "On the feasibility of hardware implementation of sub-nyquist random-sampling based analog-to-information conversion," in *Proc. ISCAS*, 2008, pp. 1480–1483.
- [14] E. G. Allstot, A. Y. Chen, A. M. R. Dixon, D. Gangopadhyay, and D. J. Allstot, "Compressive sampling of ECG bio-signals: Quantization noise and sparsity considerations," in *Proc. IEEE Biomedical Circuits and System Conf.*, Nov 2010, pp. 41–44.