Compressive Sensing Based Image Processing and Energy-Efficient Hardware Implementation with Application to MRI and JPEG 2000

by

Nandini Ramesh Kumar

in fulfilment of the requirements of

PHD DISSERTATION

Faculty of Health,Engineering & Sciences

University of Southern Queensland

2014

Abstract

In the present age of technology, the buzzwords are low-power, energy-efficient and compact systems. This directly leads to the date processing and hardware techniques employed in the core of these devices. One of the most power-hungry and space-consuming schemes is that of image/video processing, due to its high quality requirements. In current design methodologies, a point has nearly been reached in which physical and physiological effects limit the ability to just encode data faster. These limits have led to research into methods to reduce the amount of acquired data without degrading image quality and increasing the energy consumption.

Compressive sensing (CS) has emerged as an efficient signal compression and recovery technique, which can be used to efficiently reduce the data acquisition and processing. It exploits the sparsity of a signal in a transform domain to perform sampling and stable recovery. This is an alternative paradigm to conventional data processing and is robust in nature. Unlike the conventional methods, CS provides an information capturing paradigm with both sampling and compression. It permits signals to be sampled below the Nyquist rate, and still allowing optimal reconstruction of the signal. The required measurements are far less than those of conventional methods, and the process is non-adaptive, making the sampling process faster and universal.

In this thesis, CS methods are applied to magnetic resonance imaging (MRI) and JPEG 2000, which are popularly used imaging techniques in clinical applications and image compression, respectively. Over the years, MRI has improved dramatically in both imaging quality and speed. This has further revolutionized the field of diagnostic medicine. However, imaging speed, which is essential to many MRI applications still remains a major challenge. The specific challenge addressed in this work is the use of non-Fourier based complex measurement-based data acquisition. This method provides the possibility of reconstructing high quality MRI data with minimal measurements, due to the high incoherence between the two chosen matrices. Similarly, JPEG2000, though providing a high compression, can be further improved upon by using compressive sampling. In addition, the image quality is also improved. Moreover, having a optimized JPEG 2000 architecture reduces the overall processing, and a faster computation when combined with CS.

Considering the requirements, this thesis is presented in two parts. In the first part: (1) A complex Hadamard matrix (CHM) based 2D and 3D MRI data acquisition with recovery using a greedy algorithm is proposed. The CHM measurement matrix is shown to satisfy the necessary condition for CS, known as restricted isometry property (RIP). The sparse recovery is done using compressive sampling matching pursuit (CoSaMP); (2) An optimized matrix and modified CoSaMP is presented, which enhances the MRI performance when compared with the conventional sampling; (3) An energy-efficient, cost-efficient hardware design based on field programmable gate array (FPGA) is proposed, to provide a platform for low-cost MRI processing hardware. At every stage, the design is proven to be superior with other commonly used MRI-CS methods and is comparable with the conventional MRI sampling.

In the second part, CS techniques are applied to image processing and is combined with JPEG 2000 coder. While CS can reduce the encoding time, the effect on the overall JPEG 2000 encoder is not very significant due to some complex JPEG 2000 algorithms. One problem encountered is the big-level operations in JPEG 2000 arithmetic encoding (AE), which is completely based on bit-level operations. In this work, this problem is tackled by proposing a two-symbol AE with an efficient FPGA based hardware design. Furthermore, this design is energy-efficient, fast and has lower complexity when compared to conventional JPEG 2000 encoding.

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Nandini Ramesh Kumar

0050109401

Signature of Candidate

Date

ENDORSEMENT

Signature of Supervisor/s

Date

Acknowledgments

First and foremost, I would like to express my gratitude to my principal supervisor A/Prof. Wei Xiang. My research would not have been possible without his support and guidance. I am thankful to my associate supervisor A/Prof. John Leis, for his support during my PhD tenure.

I would also like to thank USQ for providing a three-year post graduate research scholarship, without which my research term would be hectic. Also, thanks to the Computational Engineering and Science Research Centre (CESRC) for the top-up scholarship. A heartfelt thanks to my colleagues, who have extended their knowledge and resources in MRI. It would have been difficult to complete my thesis without these resources. The discussions that we had are invaluable. Special thanks to Ms. Juanita Ryan, who has always been there and extended unconditional support whenever needed. Thanks to my friends and family, for having faith in my capabilities. Their constant motivation and support has given me the confidence to do this research.

Last, but not the least, I would like to thank and dedicate this thesis to my husband Ramesh, who has been my pillar of strength. I am greatly indebted for his unconditional love, patience and all the sacrifices, even though it has not been easy being away from family and friends. A big thankyou to my friends in Toowoomba, for helping me stay positive and giving me the 'can do' attitude.

Nandini Ramesh Kumar

University of Southern Queensland
2014

Associated Publications

The following publications were produced during the period of candidature:

[1] Nandini Ramesh Kumar, Wei Xiang and Yafeng Wang, “An FPGA-based fast two-symbol processing architecture for JPEG 2000 arithmetic coding”, in *Proc. 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 14-19 Mar. 2010, Dallas, TX. USA, pp. 1282-1285.

[2] Nandini Ramesh Kumar, Wei Xiang and Yafeng Wang, “Two-symbol FPGA architecture for fast arithmetic encoding in JPEG 2000”, *Journal of Signal Processing Systems*, Vol. 69, No. 2, pp. 213-224, 2012.

The work in these papers is presented in Chapter 7.

[3] Nandini Ramesh Kumar, Wei Xiang and Jeffrey Soar, “A Novel Image Compressive Sensing Method Based on Complex Measurements”, in *Proc. International Conference on Digital Image Computing Techniques and Applications (DICTA)*, 6-8 Dec. 2011, Noosa, QLD, AUSTRALIA, pp. 175-179.

The work in the paper is presented in Chapter 6.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

2D	Two Dimensional
3D	Three Dimensional
AE	Arithmetic Encoding
ALM	Adaptive Logic Module
ALUT	Adaptive Look-Up Table
AMD	Advanced Micro Devices
ASIC	Application Specific Integrated Circuit
BC	Bit-plane Coding
BP	Basis Pursuit
BPC	Bit Plane Coder
BPDN	Basis Pursuit DeNoising
BPIC	Basis Pursuit with Integrated Constraints
CG	Conjugate Gradient
CHM	Complex Hadamard Matrix
CORDIC	COordinate Rotation DIgital Computer
CoSaMP	Compressive Sampling Matching Pursuit
CPC	Column Processing Core
CPU	Central Processing Unit
CRAD	Complex RADemacher function
CRAM	Configuration Random Access Memory
CS	Compressive Sensing
CX-D	Context-Decision (pair)
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
FFT	Fast Fourier Transform
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FOCUSS	FOCal Underdetermined System Solution
FOV	Field Of View
GHz	Giga Hertz
GPSR	Gradient Projection for Sparse Reconstruction
GPU	Graphics Processing Unit

ICBM	International Consortium of Brain Mapping
IDWT	Inverse Discrete Wavelet Transform
IHT	Iterative Hard Thresholding
IST	Iterative Shrinkage/Thresholding
JPEG	Joint Pictures Expert Group
LPS	Least Probable Symbol
LUT	Look-Up Table
MHz	Mega Hertz
MP	Matching Pursuit
MPS	Most Probable Symbol
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imaging
MSB	Most Significant Bit
NLPS	Next Least Probable Symbol
NMR	Nuclear Magnetic Resonance
NMPS	Next Most Probable Symbol
OMP	Orthogonal Matching Pursuit
PE	Processing Element
PFFT	Partial Fast Fourier Transform
POCS	Projection Onto Convex Set
PSNR	Peak Signal-to-Noise Ratio
QRD	matrix Q and R Decomposition
RAM	Random Access Memory
RF	Radio Frequency
RIP	Restricted Isometry Property
RPC	Row Processing Core
SART	Simultaneous Algebraic Reconstruction Technique
SNR	Signal-to-Noise Ratio
SP	Subspace Pursuit
TSMC	Taiwan Semiconductor Manufacturing Company
TV	Total Variation
UCHM	Unitary Complex Hadamard Matrix

Chapter 1

Introduction

Information processing is a dominant part in any field that uses present technologies. Computational and analytical tools are being continuously developed for the extraction of information from data, and are fast becoming irrelevant in the face of large problem sizes necessitated by todays applications. Therefore, the challenge is to devise new and computationally efficient set of information processing tools that can effectively cope with this huge set of data.

Any compressible signal can be well approximated using sparse representation and hence could be exploited for reduction in complexity of encoding. Compressive sensing (CS) provides a dramatic reduction in sampling rates and computation complexity in data compression. It aims to measure sparse and compressible signals close to their intrinsic information rate rather than their Nyquist rate. It addresses the shortcomings of the traditional transform-based methods by directly acquiring compressed samples. It uses the concept that a small group of non-adaptive linear projections of a sparse signal contain enough information to reconstruct the complete data and also preserve the originality of the signal. An appropriate way to obtain linear measurements is by using incoherent sampling in a transform domain that is equipped with fast transform algorithms. Hence, the CS theory has been rapidly gaining more attention in image/video processing due to the requirement for processing large data. Most signals exhibit a sparse representation in some basis (e.g., Fourier, wavelet domain). Since most problems can be formulated using a set of linear equations, CS theory is finding use in most practical applications.

1.1 Research Problem

In present clinical practice, magnetic resonance imaging (MRI) is one of the most popular imaging modalities due to its excellent depiction of soft tissues, and inherent absence of emitted ionizing radiation. The traditional approach of MRI data acquisition is to sample at Nyquist rate followed by use of coding methods for compression. Recent trends have advanced to 3D-MRI and generally require faster acquisition techniques to achieve clinical practicality. Unfortunately, such accelerations may result in a compromise of image quality, in terms of spatial and temporal resolution, signal-to-noise ratio (SNR) etc.

Despite its many advantages, a fundamental limitation of MRI is the linear relation between the number of measured data samples and net scan time. Increased scan duration presents a number of practical challenges in clinical imaging including higher susceptibility to physiological motion artifacts, diminished clinical throughput, and added patient discomfort. This shows the importance of imaging speed in MRI applications. However, the speed at which data can be collected in MRI is fundamentally limited by physical (gradient amplitude and slew-rate) and physiological (nerve stimulation) constraints [3]. Therefore, the prime concern is to seek methods to reduce the amount of acquired data without degrading the image quality.

In the past several years, the practical performance of CS theory, has been successfully demonstrated for a large range of clinical applications including non-Cartesian and 3D-MR angiography (MRA) [3,4], and time-resolved imaging [5]. Several groups in the MRI community have proposed novel numerical techniques for robust CS with specific focus on MR image reconstruction including nonlinear conjugate gradient (CG) [3], interior point (IP) [6], Bregman iteration or inverse scale space [7], and iterative reweighted least squares or FOCUSS [8,9] methods. Since there has been very few comparison of the computational performance of these techniques on large-scale problems to date, the best approach that can meet the clinical demands is still an open question. Moreover, to improve the speed, it becomes necessary that the processing algorithms are also computationally sped up. There have various attempts where central processing unit (CPU), graphic processing unit (GPU), field programmable gate array (FPGA) and application specific integrated circuit (ASIC) [10–15] being used for this purpose, which have time durations ranging from minutes to hours. Again, though there are solutions for reduction in computation time, the desire for the least computational time needs to be ascertained.

In conventional digital image sampling and compression system, natural image signals are sampled according to Shannon sampling theory and quantized into discrete digital signals. Once processed using image encoder, only a part of the transform coefficients are retained, while most of the other sampled data will be discarded. This is further entropy encoded, particularly in JPEG 2000, which is a cumbersome process. All the data is entropy encoded bit-by-bit and this leads to a slower system and high complexity. Hence, combining CS with image encoder can be one solution that could lead to having very few data transformed data (e.g., less than 10%) when compared to conventional image transforms. It is a known fact that, due to CPU dependencies, software-based processing is much slower than hardware-based processing. Moreover, with modern image/video technologies low-complexity, low-energy and small-size hardware systems are most preferable. This, further adds up to the necessity of having a image encode-decode system that addresses the aforementioned issues.

Considering the issues related to MRI and natural imaging, a solution is necessary that will help in overcoming these issues. This points to compressive sensing, which has been successful in providing reconstruction of data from very few samples. This study focusses on applying CS techniques for encoding and reconstruction, in order to overcome the drawbacks persisting in MRI and natural imaging. Specifically, this work focuses on; i) To reduce the MRI data acquisition without compromising quality of the processed data and; ii) CS-based JPEG 2000 optimized system that provides a high compression ratio and image quality, and also have a low-complexity and low-energy consumption.

Even though there exist many CS methods for 2D/3D-MRI processing, to the best of my knowledge, these methods either provide a GPU-based implementation of purely software-based system. While these are feasible solutions, it is rather more practical to have a purely hardware system. And, while thinking of a hardware solution, it is of utmost importance that the processing blocks have low-complexity, have low-energy consumption and most importantly maintain the data quality. The key innovations addressed in this work is the use of complex matrices for CS. In this methodology the acquisition is minimal compared to the conventional Fourier transform used in MRI. For an 256×256 image data, the required measurements is about 5K to 8K. The reconstruction quality is maintained. The idea behind this innovation is to provide a low-cost, energy efficient and low MRI scan time. This would make MRI scanning more user-friendly when compared to the existing MRI scanning. A similar challenge exists in JPEG 2000 processing, and one of the computation intensive block is the arithmetic coding. Furthermore, encoding/decoding large images in real-time requires efficient architecture, and hence the solution lies in using CS-based methods.

1.2 Contributions

The primary essence of this work is to provide a low-complexity, low-cost, energy-efficient encoding-decoding system for MRI and natural imaging, while overcoming the processing speed issues. Though the application of CS techniques varies for MRI and natural imaging, the core CS principles are same. In case of MRI processing, CS is used for data acquisition and reconstruction of images. For CS-based natural imaging, the aim is to reduce the transform data and embed the CS principles in JEPG 2000.

Specifically the contributions are summarized as follows:

1. **2D and 3D MRI processing using CS-based complex measurements**: A non-Fourier based 2D and 3D MRI data acquisition and reconstruction is designed. To enable CS techniques in this work, a complex Hadamard matrix proposed. This structure is used for the first time in this work and is elaborated in Chapter 3. The data acquisition is performed using complex Hadamard matrix and the transform used is the Daubechies-4 wavelet [16]transform since they are highly incoherent. The combination of the complex Hadamard and wavelet transform, called the sensing matrix is shown to satisfy restricted isometry property. A specific bound for CoSaMP reconstruction is derived with respect to 3D-MRI. This bound is optimized for 3D-MRI performance. Furthermore, comparison is drawn with an existing non-Fourier 3D CS algorithm. The simulation platform is built and performance of the system is shown in terms of signal-to-noise ratio. This method is discussed in Chapter 3.

2. **Optimization of Complex Hadamard Matrix for Enhanced 2D/3D-MRI Performance**: Chapter 4 mainly deals with the optimization algorithm and its effectiveness for 3D-MRI. An optimized version of the complex Hadamard matrix is presented and verified for CS properties, which is the primary requirement for any matrix to be used for CS-based processing. This is primarily a structured unitary matrix and hence termed as unitary complex Hadamard matrix. Furthermore, when used with the optimized CoSaMP(discussed in Chapter 3), ensures an enhanced 3D-MRI reconstruction. The numerical results are demonstrated for 3D-MRI and a comparison shows a significant increase in signal-to-noise ratio.

3. **Performance efficient CS-based FPGA hardware architecture for MRI processing**: In Chapter 5, a fast and efficient CS-based hardware is designed. The main features of this implementation is its pipeline structure and efficient memory organization. These features aim at providing reduced complexity and increase the speed, which is one of the issues in MRI. A simulation platform is built and the tabulated results show the same signal-

to-noise ratio as when done through software methods. A comparison with other existing architectures, shows the efficiency of this architecture.

4. **Low-complexity energy-efficient CS-based natural image processing hardware**: A low-complex and energy-efficient pipelined hardware-based architecture is presented in Chapter 6. The main aim is to provide an architecture that can fit easily with the existing JPEG 2000 and is based on CS principles. The idea behind the use of CS is to drastically reduce the transform coefficients. This in turn makes the encoder less complex and easily portable on low-power devices.

5. **Two-symbol arithmetic encoding architecture for efficient entropy coding in CS-based JPEG 2000**: Chapter 7 mainly deals with entropy encoding. Here, a high-performance two-symbol arithmetic encoding hardware is presented. Most of the JPEG 2000 entropy coders are based on processing one symbol per clock cycle. This bit-by-bit serial operation is computationally intense and requires huge hardware resources. Alongside, the energy efficiency is goes down significantly due to its serial nature. This issue is dealt with in this chapter and results compared with some of the existing hardware architectures.

1.3 Organization

This dissertation begins with a review on the relevant topics used in this work in Chapter 2. This includes compressive sensing, magnetic resonance imaging, field programmable gate arrays and JPEG 2000. Chapters 3, 4 and 5 deal with CS-based MRI processing, optimizations and their hardware architecture design. Chapters 6 and 7 are mainly on JPEG 2000 hardware architecture design and application of CS for an improved efficiency.

In Chapter 3 a CS-based MRI processing method is detailed, by deriving a proof for restricted isometry property and bound for CoSaMP. Simulation results for both 2D-MRI and 3D-MRI are presented separately, in Sections 3.4.1 and 3.4.2, respectively.

Chapter 4 deals with optimizing the earlier proposed matrix and reconstruction algorithm, for an enhanced 3D-MRI performance. The derivation for the restricted isometry property is given in Section 4.3. Finally, the simulation results and discussions are presented in Section 4.4.

A fast and efficient hardware-based architecture is designed in Chapter 5. The system blocks are presented in Section 5.4, with the related FPGA architecture in Section 5.3. Again, for the purpose of validation and comparison, simulations results are shown in Section 5.5 and followed with discussions.

Chapter 6 details on the aspects of the hardware structure of the transform block of JPEG 2000 combined with CS techniques. The system model is presented in Section 6.3 and the related CS processing is detailed in Section 6.4. The designed hardware architecture with the details of pipelining and timing diagrams are presented in Section 6.5, for both encoder and decoder. The simulated results are discussed in Section 6.6.

In Chapter 7, a two-symbol arithmetic encoder for JPEG 2000 is presented. The detailed architecture and process is explained in Sections 7.3 and 7.4. The combined JPEG 2000 architecture from the Chapters 6 and 7 is shown Section 7.5, which is targeted for an FPGA. The simulation results for the arithmetic encoder is tabulated in Section 7.6.

Finally, Chapter 8 summarizes the results of this dissertation and possible future directions for this work.

Chapter 2

Background

In this chapter, an overview of various theoretical aspects dealt in this thesis are provided. In Section 2.1, a brief description on compressive sensing theory is provided. In Section 2.2 we discuss magnetic resonance imaging with respect to single slice and 3D MRI. Section 2.5 and 2.4 provide an overview of the field programmable gate arrays and JPEG 2000 image processing, respectively.

2.1 Compressive Sensing Theory

Compressed sensing (CS) [17–20] offers a framework for simultaneous sensing and compression of finite-dimensional vectors, that rely on the reduction of linear dimensions. Specifically, in CS we do not acquire signal x directly but rather acquire $M < N$ linear measurements $y = \Phi x$ using an $M \times N$ CS matrix Φ, where y is the measurement vector. Ideally, the matrix Φ is designed to reduce the number of measurements M as much as possible while allowing the recovery of a wide class of signals x from their measurement vectors y. However, the fact that $M < N$ renders matrix Φ rank-deficient, meaning that it has a non-empty nullspace. This in turn, implies that for any particular signal $x_0 \in \mathbb{R}^N$, an infinite number of signals x will yield the same measurements $y_0 = \Phi x_0 = \Phi x$ for the chosen CS matrix Φ.

The motivation behind the design of matrix Φ is, therefore, to allow for distinct signals x, x' within a class of signals of interest to be uniquely identifiable from their measurements $y = \Phi x$, $y' = \Phi x'$, even though $M \ll N$. We must therefore make a choice on the class of signals that we aim to recover from CS measurements.

2.1.1 Sparsity

Sparsity is the signal structure behind many compression algorithms that employ transform coding, and is the most prevalent signal structure used in CS. Sparsity also has a rich history of applications in signal processing problems in the last century (particularly in imaging), including denoising, deconvolution, restoration etc [21–23].

To introduce the notion of sparsity, we rely on a signal representation in a given basis $\{\psi_i\}_{i=1}^N$ for $\mathbb{R}^N$. Every signal $x \in \mathbb{R}^N$ is representable in terms of N coefficients $\{\theta\}_{i=1}^N$ as $x = \sum_{i=1}^N \psi_i \theta_i$; arranging the ψ_i as columns into the $N \times N$ matrix ψ and the coefficients θ_i into the $N \times 1$ coefficient vector θ, we can write succinctly that $x = \psi\theta$, with $\theta \in \mathbb{R}^N$. Similarly, if we use ψ containing N unit-norm column vectors of length L with $L \times N$ (i.e., $\psi \in \mathbb{R}^{L\times N}$), then for any vector $x \in \mathbb{R}^L$ there exist infinitely many decompositions $\theta \in \mathbb{R}^N$ such that $x = \psi\theta$. In a general setting, we refer to ψ as the sparsifying dictionary [24]. These concepts are extendable to complex signals as well [25, 26]. We say that a signal x is K-sparse in the basis ψ if there exists a vector $\theta \in \mathbb{R}^N$ with only $K \ll N$ nonzero entries such that $x = \psi\theta$. We call the set of indices corresponding to the nonzero entries the support of θ and denote it by $\text{supp}(\theta)$. We also define the set Σ_K that contains all signals x that are K-sparse. A K-sparse signal can be efficiently compressed by preserving only the values and locations of its nonzero coefficients, using $O(Klog_2N)$ bits: coding each of the K nonzero coefficients locations takes log_2N bits, while coding the magnitudes uses a constant amount of bits that depends on the desired precision, and is independent of N. This process is known as transform coding, and relies on the existence of a suitable basis Ψ that renders signals of interest sparse or approximately sparse.

For signals that are not exactly sparse, the amount of compression depends on the number of coefficients of θ that we keep. Consider a signal x whose coefficients θ, when sorted in order of decreasing magnitude, decay according to the power law

$$|\theta(\mathfrak{I}(n))| \leq Sn^{-1/r}, n = 1, \dots, N, \tag{2.1}$$

where $\mathfrak{I}$ indexes the sorted coefficients. Due to the rapid decay of their coefficients, such signals are well-approximated by K-sparse signals. The best K-sparse approximation error for such a signal obeys

$$\sigma_\psi(x, K) := arg \min_{x' \in \Sigma_K} \|x - x'\|_2 \leq \mathbf{CSK}^{-s}, \tag{2.2}$$

with $s = \frac{1}{r} - \frac{1}{2}$ and $\mathbf{C}$ denoting a constant that does not depend on N [27]. That is, the signal's best approximation error in an l_2-norm sense, has a power law decay with exponent s as K increases. We dub such a signal s-compressible. When ψ is an orthonormal basis, the best sparse approximation of x is obtained by hard thresholding the signal's coefficients, so that only the K coefficients with largest magnitudes are preserved.

2.1.2 Design of Measurement Matrices

The main design criteria for the CS matrix Φ is to enable the unique identification of a signal of interest x from its measurements $y = \Phi x$. Clearly, when we consider the class of K-sparse signals Σ_K, the number of measurements $M > K$ for any matrix design, since the identification problem has K unknowns even when the support $\Omega =$supp(x) of the signal x is provided. In this case, we simply restrict the matrix Φ to its columns corresponding to the indices in Ω, , denoted by Φ_Ω, and then use the pseudoinverse to recover the nonzero coefficients of x:

$$x_\Omega = \Phi_\Omega^\dagger y. \tag{2.3}$$

Here x_Ω is the restriction of the vector x to the set of indices Ω, and $M^\dagger = (M^T)^{-1}M^T$ denotes the pseudoinverse of the matrix M. The implicit assumption in (2.3) is that Φ_Ω has full column-rank so that there is a unique solution to $y = \Phi_\Omega x_\Omega$.

We begin by determining properties of Φ that guarantee that distinct signals $x, x' \in \Sigma_K, x \neq x'$, lead to different measurement vectors $\Phi x \neq \Phi x'$. In other words, we want each vector $y = \mathbb{R}^M$ to be matched to at most one vector $x \in \Sigma_K$ such that $y = \Phi x$. A key relevant property of the matrix in this context is its spark.

Definition 1. *[28] The spark* spark(Φ) *of a given matrix Φ is the smallest number of columns of Φ that are linearly dependent.*

The spark is related to the Kruskal Rank from the tensor product literature; the matrix Φ has Kruskal rank spark$(\Phi) - 1$. This definition allows us to pose the following straightforward guarantee.

Theorem 1. *[28] If* spark$(\Phi) > 2K$ *, then for each measurement vector $y \in \mathbb{R}^M$ there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$.*

It is easy to see that spark $\in [2, M+1]$, so that Theorem 1 yields the requirement $M \geq 2K$.

While Theorem 1 guarantees uniqueness of representation for K-sparse signals, computing the spark of a general matrix Φ has combinatorial computational complexity, since one must verify that all sets of columns of a certain size are linearly independent. Thus, it is preferable to use properties of Φ that are easily computable to provide recovery guarantees. The coherence of a matrix is one such property.

Definition 2. *[29] The coherence $\mu(\Phi)$ of a matrix Φ is the largest absolute inner product between any two columns of Φ:*

$$\mu(\Phi) = \max_{1 \leq i \neq j \leq N} \frac{|\langle \Phi_i, \Phi_j \rangle|}{||\Phi_i||_{l2} ||\Phi_j||_2} \tag{2.4}$$

It can be shown that $\mu(\Phi) \in \left[\sqrt{\frac{N-M}{M(N-1)}}, 1 \right]$; the lower bound is known as the Welch bound [30,31]. Note that when $N \gg M$, the lower bound is approximately $\mu(\Phi) \geq 1/\sqrt{M}$. One can tie the coherence and spark of a matrix by employing the Gershgorin circle theorem.

Theorem 2. *[32] The eigenvalues of an $m \times m$ matrix M with entries $M_{i,j}, 1 \leq i, j \leq m$, lie in the union of m discs $d_i = d_i(c_i, r_i), 1 \leq i \leq m$, centered at $c_i = M_{i,i}$ with radius $r_i = \Sigma_{j \neq i} |M_{i,j}|$.*

Applying this theorem on the Gram matrix $G = \Phi_\Omega^T \Phi_\Omega$ leads to the following result.

Lemma 1. *[28] For any matrix Φ,*

$$\text{spark}(\Phi) \geq 1 + \frac{1}{\mu(\Phi)}. \tag{2.5}$$

By merging Theorem 1 with Lemma 1, we can pose the following condition on Φ that guarantees uniqueness.

Theorem 3. *[28, 33, 34] If*

$$K < \frac{1}{2}(1 + \frac{1}{\mu(\Phi)}, \tag{2.6}$$

then for each measurement vector $y \in \mathbb{R}_M$ there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$.

Theorem 3, together with the Welch bound, provides an upper bound on the level of sparsity K that guarantees uniqueness using coherence $K = O(\sqrt{M})$. The prior properties of the CS matrix provide guarantees of uniqueness when the measurement vector y is obtained without error. Hardware considerations introduce two main sources of inaccuracies in the measurements: inaccuracies due to noise at the sensing stage (in the form of additive noise $y = \Phi x + n$), and inaccuracies due to mismatches between the CS matrix used during recovery, Φ, and that implemented during acquisition, $\Phi' = \Phi + \Delta$ (in the form of multiplicative noise [35, 36]). Under these sources of error, it is no longer possible to guarantee uniqueness; however, it is desirable for the measurement process to be tolerant to both types of error. To be more formal, we would like the distance between the measurement vectors for two sparse signals $y = \Phi x$, $y' = \Phi x'$ to be proportional to the distance between the original signal vectors x and x'. Such a property allows us to guarantee that, for small enough noise, two sparse vectors that are far apart from each other cannot lead to the same (noisy) measurement vector. This behavior has been formalized into the restricted isometry property (RIP).

Definition 3. *[37] A matrix Φ has the (K, δ)-restricted isometry property ((K, δ)-RIP) if, for all $x \in \Sigma_K$,*

$$(1 - \delta)\|x\|_2^2 \leq \|\Phi x\|_2^2(1 + \delta)\|x\|^2{}_2. \tag{2.7}$$

In words, the (K, δ)-RIP ensures that all submatrices of Φ of size $M \times K$ are close to an isometry, and therefore distance-preserving. We will show later that this property suffices to prove that the recovery is stable to presence of additive noise n. In certain settings, noise is introduced to the signal x prior to measurement. Recovery is also stable for this case; however, there is a degradation in the distortion of the recovery by a factor of N/M [38–40].

Furthermore, the RIP also leads to stability with respect to the multiplicative noise introduced by the CS matrix mismatch Δ [35,36]. The RIP can be connected to the coherence property by using, once again, the Gershgorin circle theorem (Theorem 2).

Lemma 2. *[41] If Φ has unit-norm columns and coherence $\mu = \mu(\Phi)$, then Φ has the (K, δ)-RIP with $\delta \leq (K - 1)\mu$.*

One can also easily connect RIP with the spark. For each K-sparse vector to be uniquely identifiable by its measurements, it suffices for the matrix Φ to have the $(2K, \delta)$-RIP with $\delta > 0$, as this implies that all sets of $2K$ columns of Φ are linearly independent, i.e., spark$(\Phi) > 2K$ (Theorems 1 and 3). We will see later that the RIP enables recovery guarantees that are much stronger than those based on spark and coherence. However, checking whether a CS matrix Φ satisfies the (K, δ)-RIP has combinatorial computational complexity.

Now that we have defined relevant properties of a CS matrix Φ, we discuss specific matrix constructions that are suitable for CS. An $M \times N$ Vandermonde matrix V constructed from N distinct scalars has spark$(V) = M + 1$ [27]. Unfortunately, these matrices are poorly conditioned for large values of N, rendering the recovery problem numerically unstable. Similarly, there are known matrices Φ of size $M \times M^2$ that achieve the coherence lower bound

$$\mu\Phi = 1/\sqrt{M}, \tag{2.8}$$

such as the equiangular tight frames [31]. It is also possible to construct deterministic CS matrices of size $M \times N$ that have the (K, δ)-RIP for $K = O(\sqrt{M} logM/log(N/M))$ [42]. These constructions restrict the number of measurements needed to recover a K-sparse signal to be $M = O(K^2 logN)$, which is undesirable for real-world values of N and K. Fortunately, these bottlenecks can be defeated by randomizing the matrix construction. For example, random matrices Φ of size $M \times N$ whose entries are independent and identically distributed (i.i.d.) with continuous distributions have spark$(\Phi) = M + 1$ with high probability. It can also be shown that when the distribution used has zero mean and finite variance, then in the asymptotic regime (as M and N grow) the coherence converges to $\mu\Phi = 2\sqrt{logN/M}$ [43,44]. Similarly, random matrices from Gaussian, Rademacher, or more generally a subgaussian distribution have the (K, δ)-RIP with high probability if

$$M = O(Klog(N/K)/\delta^2). \tag{2.9}$$

A Rademacher distribution gives probability 1/2 to the values ±1. A random variable X is called subgaussian if there exists $c > 0$ such that $\mathbb{E}(e^{Xt}) \leq e^{c^2t^2/2}$ for all $t \in R$. Examples include the Gaussian, Bernoulli, and Rademacher random variables, as well as any bounded random variable.

Finally, we point out that while the set of RIP-fulfilling matrices provided above might seem limited, emerging numerical results have shown that a variety of classes of matrices Φ are suitable for CS recovery, including subsampled Fourier and Hadamard transforms [45,46].

2.1.3 CS Recovery Algorithms

We now focus on solving the CS recovery problem, given y and Φ, find a signal x within the class of interest such that $y = \Phi x$ exactly or approximately. When we consider sparse signals, the CS recovery process consists of a search for the sparsest signal x that yields the measurements y. By defining the l_0 norm of a vector $\|x\|_0$ as the number of nonzero entries in x, the simplest way to pose a recovery algorithm is using the optimization

$$\widehat{x} = \arg\min_{x \in \mathbb{R}^N} \|x\|_0. \tag{2.10}$$

Solving (2.10) relies on an exhaustive search and is successful for all $x \in \Sigma_K$ when the matrix Φ has the sparse solution uniqueness property (i.e., for M as small as $2K$). However, this algorithm has combinatorial computational complexity, since we must check whether the measurement vector y belongs to the span of each set of K columns of Φ, $K = 1, 2, \ldots, N$. Our goal, therefore, is to find computationally feasible algorithms that can successfully recover a sparse vector x from the measurement vector y for the smallest possible number of measurements M.

An alternative to the l_0 norm used in (2.10) is to use the l_1 norm, defined as $\|x\|_1 = \sum_{n=1}^{N} |x(n)|$. The resulting adaptation of (2.10), known as basis pursuit (BP) [22], is formally defined as

$$\widehat{x} = \arg\min_{x\in\mathbb{R}^N} \|x\|_1 subjecttoy = \Phi x. \tag{2.11}$$

Since the l_1 norm is convex, (2.11) can be seen as a convex relaxation of (2.10). Thanks to the convexity, this algorithm can be implemented as a linear program, making its computational complexity polynomial in the signal length [47]. The optimization (2.11) can be modified to allow for noise in the measurements $y = \Phi x + n$; we simply change the constraint on the solution to

$$\widehat{x} = \arg\min_{x\in\mathbb{R}^N} \|x\|_1 subjectto \|y - \Phi x\|_2 \leq \epsilon, \tag{2.12}$$

where$\epsilon \geq \|n\|_2$ is an appropriately chosen bound on the noise magnitude. This modified optimization is known as basis pursuit with inequality constraints (BPIC) and is a quadratic program with polynomial complexity solvers [47]. The Lagrangian relaxation of this quadratic program is written as

$$\widehat{x} = \arg\min_{x\in\mathbb{R}^N} \|x\|_1 + \lambda \|y - \Phi x\|_2, \tag{2.13}$$

and is known as basis pursuit denoising (BPDN). There exist many efficient solvers to find BP, BPIC, and BPDN solutions; for an overview, see [48]. Oftentimes, a bounded-norm noise model is overly pessimistic, and it may be reasonable instead to assume that the noise is random. For example, additive white Gaussian noise $n \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ is a common choice. Approaches designed to address stochastic noise include complexity-based regularization [49] and Bayesian estimation [50]. These methods pose probabilistic or complexity-based priors, respectively, on the set of observable signals. The particular prior is then leveraged together with the noise probability distribution during signal recovery. Optimization-based approaches can also be formulated in this case; one of the most popular techniques is the Dantzig selector [51]:

$$\widehat{x} = \arg\min_{x\in\mathbb{R}^N} \|x\|_1 suchthat \|\Phi^T(y - \Phi x)\|_\infty \leq \lambda\sqrt{\log N\sigma}, \tag{2.14}$$

where $\|\cdot\|_\infty$ denotes the l_∞-norm, which provides the largest-magnitude entry in a vector and λ is a constant parameter that controls the probability of successful recovery.

An alternative to optimization-based approaches, are greedy algorithms for sparse signal recovery. These methods are iterative in nature and select columns of Φ according to their correlation with the measurements y determined by an appropriate inner product. For example, the matching pursuit and orthogonal matching pursuit algorithms (OMP) [24, 52] proceed by finding the column of Φ most correlated to the signal residual, which is obtained by subtracting the contribution of a partial estimate of the signal from y. The OMP method is formally defined

Algorithm 1 Orthogonal Matching Pursuit

```
Input: CS matrix $\Phi$, measurement vector $y$
Output: Sparse representation $\widehat{x}$
Initialize:$\widehat{x}_0$, $r = y$, $\Omega = \emptyset, i = 0$
while halting criterion false do
    $i \leftarrow i + 1$
    $b \leftarrow \Phi^T r$ {form residual signal estimate}
    $\Omega = \Omega \bigcup supp(\mathfrak{T}(b, 1))$ {update support with residual}
    $\widehat{x}_i|_\Omega \leftarrow \Phi_\Omega^\dagger y, \widehat{x}_i|_{\Omega^C} \leftarrow 0$ {update signal estimate}
    $r \leftarrow y - \Phi\widehat{x}_i$ {update measurement residual}
end while
return $\widehat{x} \leftarrow \widehat{x}_i$
```

as Algorithm 1, where $\mathfrak{T}(x, K)$ denotes a thresholding operator on x that sets all but the K entries of x with the largest magnitudes to zero, and $x|_\Omega$ denotes the restriction of x to the entries indexed by Ω. The convergence criterion used to find sparse representations consists of checking whether $y = \Phi x$ exactly or approximately; note that due to its design, the algorithm cannot run for more than M iterations, as Φ has M rows. Other greedy techniques that are a similar, or rather derived from OMP include CoSaMP [53], and Subspace Pursuit (SP) [54]. Another variant is known as iterative hard thresholding (IHT) [55]: starting from an initial signal estimate $\widehat{x}_0 = 0$, the algorithm iterates a gradient descent step followed by hard thresholding, i.e.,

$$\widehat{x} = \mathfrak{T}(\widehat{x}_{i-1} + \Phi^T(y - \Phi\widehat{x}_{i-1}), K), \tag{2.15}$$

until a convergence criterion is met.

2.1.4 CS Recovery Guarantees

Many of the CS recovery algorithms above come with guarantees on their performance. We group these results according to the matrix metric used to obtain the guarantee.

Theorem 4. *[37, 53–55] Let the signal $x \in \Sigma_K$ and write $y = \Phi x + n$. The outputs $\widehat{x}$ of the CoSaMP, SP, IHT, and BPIC algorithms, with Φ having the (cK, δ)-RIP, obey*

$$\|x - \widehat{x}\|_2 \leq C_1\|x - x_K\|_2 + C_2\frac{1}{\sqrt{K}}\|x - x_K\|_1 + C_3\|n\|_2, \tag{2.16}$$

where $x_K = \arg\min_{x' \in \Sigma_K} \|x - x'\|_2$ is the best K-sparse approximation of the vector x when measured in the l_2 norm. The requirements on the parameters c,δ

of the RIP and the values of C_1, C_2, and C_3 are specific to each algorithm. For example, for the BPIC algorithm, $c = 2$ and $\delta = \sqrt{2} - 1$ suffice to obtain the guarantee in (2.16).

The type of guarantee given in Theorem 4 is known as uniform instance optimality, in the sense that the CS recovery error is proportional to that of the best K-sparse approximation to the signal x for any signal $x \in \mathbb{R}^N$. In fact, the formulation of the CoSaMP, SP and IHT algorithms was driven by the goal of instance optimality, which has not been shown for older greedy algorithms like MP and OMP. Theorem 4 can also be adapted to recovery of exactly sparse signals from noiseless measurements.

Corollary 1. *Let the signal $x \in \Sigma_K$ and write $y = \Phi x$. The CoSaMP, SP, IHT, and BP algorithms can exactly recover x from y if Φ has the (cK, δ)-RIP, where the parameters c, δ of the RIP are specific to each algorithm.*

The error in Theorem 4 is proportional to the noise magnitude $\|n\|_2$, and the bounds can be tailored to random noise with high probability.

Theorem 5. *[51] Let the signal $x \in \Sigma_K$ and write $y = \Phi x + n$, where $n \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. Suppose that $\lambda = \sqrt{2}(1 + 1/t)$ in (2.14) and that Φ has the $(2K, \delta_{2K})$ and $(3K, \delta_{3K})$-RIPs with $\delta_{2K} + \delta_{3K} < 1$. Then, with probability at least $1 - N^t/\sqrt{\pi \log N}$, we have*

$$\|\widehat{x} - x\|_2 \leq C(1 + 1/t)^2 K \sigma^2 \log N. \tag{2.17}$$

The main difference between the guarantees that rely solely on coherence and those that rely on the RIP and probabilistic sparse signal models is the scaling of the number of measurements M needed for successful recovery of K-sparse signals. According to the bounds (2.8) and (2.9), the sparsity level that allows for recovery with high probability in Theorems 4 and 5 is $K = O(M)$ instead of $K = O(\sqrt{M})$ for deterministic guarantees.

2.1.5 Structure of CS Matrices

While most initial work in CS has emphasized the use of randomized CS matrices whose entries are obtained independently from a standard probability distribution, such matrices are often not feasible for real-world applications due to the cost of multiplying arbitrary matrices with signal vectors of high dimension. In fact, very often the physics of the sensing modality and the capabilities of sensing devices limit the types of CS matrices that can be implemented in a specific application. Furthermore, in the context of analog sampling, one of the prime motivations for CS is to build analog samplers that lead to sub-Nyquist sampling rates. These involve actual hardware and therefore structured sensing devices. Hardware considerations require more elaborate signal models to reduce the number of measurements needed for recovery as much as possible. In this section, we review available alternatives for structured CS matrices; in each case, we provide known performance guarantees, as well as application areas where the structure arises. In Section VI we extend the CS framework to allow for analog sampling, and introduce further structure into the measurement process. This results in new hardware implementations for reduced rate samplers based on extended CS principles. Note that the survey of CS devices given in this section is by no means exhaustive [56]; our focus is on CS matrices that have been investigated from both a theoretical and an implementation point of view.

2.1.5.1 Subsampled Incoherent Bases

The key concept of a frames coherence can be extended to pairs of orthonormal bases. This enables a new choice for CS matrices: one simply selects an orthonormal basis that is incoherent with the sparsity basis, and obtains CS measurements by selecting a subset of the coefficients of the signal in the chosen basis [57]. We note that some degree of randomness remains in this scheme, due to the choice of coefficients selected to represent the signal as CS measurements.

Formally, we assume that a basis $\Phi \in \mathbb{R}^{N\times N}$ is provided for measurement purposes, where each column of $\Phi = [\Phi_1, \Phi_2, \ldots \Phi_N]$ corresponds to a different basis element. Let $\overline{\Phi}$ be an $N \times M$ column submatrix of Φ that preserves the basis vectors with indices Γ and set $y = \overline{\Phi}^T x$. Under this setup, a different metric arises to evaluate the performance of CS.

Theorem 6. *The mutual coherence of the N-dimensional orthonormal bases Φ and Ψ is the maximum absolute value of the inner product between elements of the two bases:*

$$\mu(\Phi, \Psi) = \max_{1\leq i,j\leq N} |\langle \Phi_i, \Psi_j \rangle|, \tag{2.18}$$

where Ψ_j denotes the jth column, or element, of the basis Ψ. The mutual coherence $\mu(\Phi, \Psi)$ has values in the range $[N^{-1/2}, 1]$. For example, $\mu(\Phi, \Psi) = N^{-1/2}$ when Φ is the discrete Fourier transform basis, or Fourier matrix, and Ψ is the canonical basis, or identity matrix, and $\mu(\Phi, \Psi) = 1$ when both bases share at least one element or column.

There are two main categories of applications where subsampled incoherent bases are used. In the first category, the acquisition hardware is limited by construction to measure directly in a transform domain. The most relevant examples are magnetic resonance imaging (MRI) [58] and tomographic imaging [59], as well as optical microscopy [60]; in all of these cases, the measurements obtained from the hardware correspond to coefficients of the images 2D continuous Fourier transform, albeit not typically selected in a randomized fashion. Since the Fourier functions, corresponding to sinusoids, will be incoherent with functions that have localized support, this imaging approach works well in practice for sparsity/compressibility transforms such as wavelets [57], total variation [59], and the standard canonical representation [60]. The second category involves the design of new acquisition hardware that can obtain projections of the signal against a class of vectors. The goal of the matrix design step is to find a basis whose elements belong to the class of vectors that can be implemented on the hardware. For example, a class of single pixel imagery based on optical modulators [61, 62] can obtain projections of an image against vectors that have binary entries. Example bases that meet this criterion include the Walsh-Hadamard and noiselet bases [63]. The latter is particularly interesting for imaging applications, as it is known to be maximally incoherent with the Haar wavelet basis. In contrast, certain elements of the Walsh-Hadamard basis are highly coherent with wavelet functions at coarse scales, due to their large supports. Permuting the entries of the basis vectors (in a random or pseudorandom fashion) helps reduce the coherence between the measurement basis and a wavelet basis.

2.1.5.2 Structurally Subsampled Matrices

In certain applications, the measurements obtained by the acquisition hardware do not directly correspond to the sensed signals coefficients in a particular transform. Rather, the observations are a linear combination of multiple coefficients of the signal. The resulting CS matrix has been termed a structurally subsampled matrix [64].

Consider a matrix of available measurement vectors that can be described as the product $\Phi = \mathbf{RU}$, where $\mathbf{R}$ is a $P \times N$ mixing matrix and $\mathbf{U}$ is a basis. The CS matrix $\overline{\Phi}$ is obtained by selecting M out of P rows at random, and normalizing the columns of the resulting subsampled matrix. There are two possible downsampling stages: first, $\mathbf{R}$ might offer only $P < N$ mixtures to be available as measurements; second, we only preserve $M < P$ of the mixtures available to represent the signal. This formulation includes the use of subsampled incoherent bases simply by letting $P = N$ and $\mathbf{R} = \mathbf{I}$, i.e., no coefficient mixing is performed. To provide theoretical guarantees we place some additional constraints on the mixing matrix $\mathbf{R}$.

Compressive ADCs are one promising application of CS, using this bases. A first step in this direction is the architecture known as the random demodulator (RD) [65]. The RD employs structurally subsampled matrices for the acquisition of periodic, multitone analog signals whose frequency components belong in a uniform grid. Such signals have a finite parametrization and therefore fit the finite-dimensional CS setting.

2.1.5.3 Subsampled Circulant Matrices

The use of Toeplitz and circulant structures [66, 67] as CS matrices was first inspired by applications in communications including channel estimation and multi-user detection where a sparse prior is placed on the signal to be estimated, such as a channel response or a multiuser activity pattern. When compared with generic CS matrices, subsampled circulant matrices have a significantly smaller number of degrees of freedom due to the repetition of the matrix entries along the rows and columns.

A circulant matrix $\mathbf{U}$ is a square matrix where the entries in each diagonal are all equal, and where the first entry of the second and subsequent rows is equal to the last entry of the previous row. Since this matrix is square, we perform random subsampling of the rows to obtain a CS matrix $\Phi = \mathbf{RU}$, where $\mathbf{R}$ is an $M \times N$ subsampling matrix, i.e., a submatrix of the identity matrix. We dub Φ a subsampled circulant matrix. Even when the sequence defining $\mathbf{U}$ is drawn at random from the distributions described, the particular structure of the subsampled circulant matrix $\Phi = \mathbf{RU}$ prevents the use of the proof techniques used in standard CS, which require all entries of the matrix to be independent. However, it is possible to employ different probabilistic tools to provide guarantees for subsampled circulant matrices. The results still require randomness in the selection of the entries of the circulant matrix.

There are several sensing applications where the signal to be acquired is convolved with the sampling hardwares impulse response before it is measured. Additionally, because convolution is equivalent to a product operator in the Fourier domain, it is possible to speed up the CS recovery process by performing multiplications by the matrices Φ and Φ^T via the fast fourier transform (FFT). In fact, such an FFT-based procedure can also be exploited to generate good CS matrices [66].

2.2 Magnetic Resonance Imaging

The MRI signal is generated by protons in the body, mostly those in water molecules. A strong static field B_0 polarizes the protons, yielding a net magnetic moment oriented in the direction of the static field. It is this net magnetic moment, or simply magnetization, which is manipulated and produces the nuclear magnetic resonance (NMR) signal. The field direction and its perpendicular plane are often referred to as the longitudinal direction and the transverse plane. The interaction of the magnetization M with an external magnetic field B is governed by the Bloch equation,

$$\frac{dM}{dt} = M \times \gamma B + \frac{M_0 - M_z}{T_1} + \frac{M_{xy}}{T_2}, \tag{2.19}$$

where M_0, M_z and M_{xy} are the equilibrium, longitudinal and transverse magnetization and γ, T_1 and T_2 are constants and are specific to different materials and types of tissues.

Applying a radio frequency (RF) excitation field B_1 to the net magnetization tips it and produces a magnetization component M_{xy}, transverse to the static field. The magnetization precesses at characteristic frequency $f_0 = \frac{\gamma}{2\pi} B_0$. Here f_0 denotes the precession frequency, B_0 the static field strength, and $\gamma/2\pi$ is a constant ($42.57MHz/T$) [37]. A typical $1.5T$ clinical MR system has a frequency of about 64 MHz. The transverse component of the precessing magnetization produces a signal detectable by a receiver coil. The transverse magnetization at a position r and time t is represented by the complex quantity $m(r,t) = |m(r,t)| \cdot e^{-i\phi(r,t)}$, where $|m(r,t)|$ is the magnitude of the transverse magnetization and $\phi(r,t)$ is its phase. The phase indicates the direction of the magnetization on the transverse plane. The transverse magnetization $m(r)$ can represent many different physical properties of tissue. One very intuitive property is the proton density of the tissue, but other properties, like relaxation, can be emphasized as well. The image of interest in MRI is $m(r)$, the image of the spatial distribution of the transverse magnetization.

Magnetization that is excited to the transverse plane precesses at the Larmor frequency. The precession creates a changing magnetic flux, which in turn (according to Faraday's law) induces a changing voltage in a receiver coil tuned to the Larmor frequency. This voltage is the MR signal that is used for imaging. The received signal is the cumulative contribution from all the excited magnetization in the volume. With only the homogeneous B_0 field present, the system does not contain any spatial information. The received signal is a complex harmonic with a single frequency peak centered at the Larmor frequency. The spatial distribution information comes from three additional fields that vary spatially. Three gradient coils, G_x, G_y and G_z create a linear variation in the longitudinal magnetic field strength as a function of spatial position. For example, when G_x is applied, the magnetic field will vary with position $B(x) = |B_0| + G_x x$. As a result, the resonance frequency of the magnetization will vary in proportion to the gradient field. This variation is used to resolve the spatial distribution.

The main difference between a 2D and 3D MRI sequence is that, in a 2D sequence, each RF pulse excites a narrow slice. Whereas in a 3D sequence, each RF pulse excites the entire imaging volume and encoding (e.g., phase encoding) is used to discriminate spatially [68]. Moreover, greater sensitivity is achieved with a 3D sequences since each acquisition represents an average of the entire sampled volume. However, the use of 3D MRI acquisitions implies long imaging times.

2.2.1 Imaging

In general, a B_1 RF field at the resonance frequency excites the whole volume. It is possible through the use of the gradients to selectively excite a smaller portion of it, for example only exciting a slice. The general idea is that only magnetization precessing close to the resonance frequency is affected by the RF field, whereas magnetization at distant frequencies is not affected. When a gradient field is applied, the resonance frequency varies with position. If during that time, a B_1 RF field with a limited bandwidth (for example a sinc shaped envelope pulse) is applied, only magnetization at a slice location corresponding to that frequency band is excited. Exciting a slice limits the imaging spatial encoding to two dimension. Exciting a slab or a volume requires three dimensional encoding. MR systems can encode spatial information by superimposing the gradient fields on top of the strong static field.

There is a Fourier relation between the received MR signal and the magnetization distribution and that the magnetization distribution can be decoded by a spectral decomposition. To see this Fourier relation more concretely consider the following: the gradient induced variation in precession frequency causes a location dependent phase dispersion to develop. The additional frequency contributed by gradient fields can be written as

$$f(r) = \frac{\gamma}{2\pi} G(t) \cdot r, \tag{2.20}$$

where $G(t)$ is a vector of the gradient fields' amplitudes. The phase of magnetization is the integral of frequency starting from time zero,soon after the RF excitation:

$$\phi(r,t) = 2\pi \int_0^t \frac{\gamma}{2\pi} G(s) \cdot r ds = 2\pi r \cdot k(t), where k(t) \equiv \frac{\gamma}{2\pi} G(s) ds. \tag{2.21}$$

The receiver coil integrates over the entire volume, producing a signal

$$s(t) = \int_R m(r) e^{-i2\pi k(t)\cdot r} dr. \tag{2.22}$$

This is the signal equation for MRI, that is, the received signal at time t is the Fourier transform of the object $m(r)$ sampled at the spatial frequency $k(t)$. Such information is fundamentally encoded and very different than traditional optical imaging where pixel samples are measured directly. The design of an MRI acquisition method centers on developing the gradient waveforms $G(t)$ that drive the MR system. These waveforms, along with the associated RF pulses used to produce the magnetization, are called a pulse sequence. The integral of the $G(t)$ waveforms traces out a trajectory $k(t)$ in spatial frequency space, or K-space.

2.2.2 Image Acquisition

Constructing a single MR image commonly involves collecting a series of frames of data, called acquisitions. In each acquisition, an RF excitation produces new transverse magnetization, which is then sampled along a particular trajectory in K-space. In principle, a complete MR image can be reconstructed from a single acquisition by using a K-space trajectory that covers a whole region of K-space [69]. This is commonly done in applications such as imaging brain activation. However, for most applications this results in inadequate image resolution and excessive image artifacts. Magnetization decays exponentially with time. This limits the useful acquisition time window. Also, the gradient system performance and physiological constraints limit the speed at which K-space can be traversed. These two effects combine to limit the total number of samples per acquisition. As a result, most MRI imaging methods use a sequence of acquisitions; each one samples part of K-space. The data from this sequence of acquisitions is then used to reconstruct an image.

Traditionally the K-space sampling pattern is designed to meet the Nyquist criterion, which depends on the resolution and field of view (FOV). Image resolution is determined by the sampled region of K-space: a larger region of sampling gives higher resolution. The supported field of view (FOV) is determined by the sampling density within the sampled region: larger objects require denser sampling to meet the Nyquist criterion. Violation of the Nyquist criterion causes the linear reconstruction to exhibit artifacts. The appearance of such artifacts depends on the details in the sampling pattern.

There is considerable freedom in designing the K-space trajectory for each acquisition. By far the most popular trajectory uses straight lines from a Cartesian grid. Most pulse sequences used in clinical imaging today are Cartesian. Reconstruction from such acquisitions is wonderfully simple: apply the inverse Fast Fourier Transform (FFT). More importantly, reconstructions from Cartesian sampling are robust to many sources of system imperfections. While Cartesian trajectories are by far the most popular, many other trajectories are in use, including sampling along radial lines and sampling along spiral trajectories. Radial acquisitions are less susceptible to motion artifacts than Cartesian trajectories [70], and can be significantly undersampled [71], especially for high contrast objects [72,73]. Spirals make efficient use of the gradient system hardware, and are used in real-time and rapid imaging applications [74]. Reconstruction from such non-Cartesian trajectories is more complicated, requiring filtered back-projection algorithms [75] or K-space interpolation schemes (e.g. gridding [76]).

2.2.3 Non-Fourier MRI Mathematical model

In this section, a overview of non-Fourier MRI acquisition is provided. The advantage of using a non-Fourier acquisition over conventional Fourier-based is that, non-Fourier coding can reduce the acquired signal space while maximizing the amount of pertinent image information that is captured. It partially encodes the field-of-view (FOV) by employing non-sinusoidal spatial encoding profiles induced via RF excitation. MRI sampling other than the Fourier has been used for effectively volume imaging of the heart [77], increasing effective relaxation times [78] etc. This non-Fourier based encoding can be derived from some of the well-known mathematical basis, such as Hadamard [79] and wavelet [78] that are also popular in signal processing. Imaging without the Fourier transform partially encodes the FOV by employing non-sinusoidal spatial encoding profiles induced via radio-frequency (RF) excitation. In general MR imaging, the received signal can be described by

$$f(k) = \int_V \rho(r) e^{i2\pi k.r} dr, \tag{2.23}$$

where $\rho(r)$ is the excited spin density function throughout the sample volume V, r is the spatial position of the spins, and k is a reciprocal spatial term corresponding to the applied gradients.

To obtain a non-Fourier based theory, we adopt and briefly review the theory from [80] for a 2D spin-echo experiment.(2.23) can be represented as

$$f(k_y, k_x) = \int_{-\alpha}^{\alpha} \int \int \rho(x, y, z) e^{i2\pi(k_x x + k_y y)} dx dy dz, \tag{2.24}$$

where 2α is the thickness of the excited slice, with the readout, phase encode, and slice-select gradients as G_x, G_y and G_z respectively. Fig.2.1 illustrates the direction of excitations applied for each of these gradients. Slice selection can be additionally performed by the slice-selective 180° refocusing RF pulse. With a known FOV, the readout and phase encoding gradient manipulations produce samples at $k_x = n\Delta k_x Z$ and $k_y = m\Delta k_y$ steps through K-space, such that $-N/2 < n \leq N/2$, $-M/2 < m \leq M/2$. In matrix form, the magnetic resonance system response can then be defined by placing the above mentioned samples in a $M \times N$ K-space matrix S, with readout samples placed in columns.

With the non-Fourier encoding methodology, the initial slice-selective RF pulse is replaced with a spatial excitation profile along the phase encode direction and

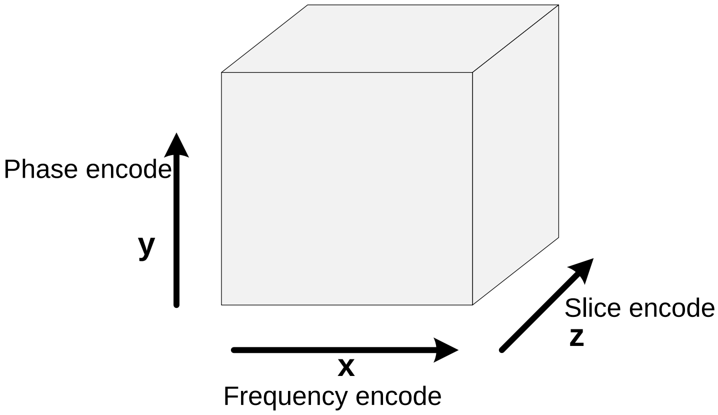


Figure 2.1: Illustration of 3D-MRI encoding directions.

eliminating G_y. An envelope for the RF is defined as $p(t) = \sum_{m=1}^{M} p_m \prod((t - m\Delta t)/\Delta t)$, where $\prod(t)$ is zero except in the interval $0 \leq t < 1$. If this RF is low flip ($\theta(r) < 30°$) [81] and is applied in a phase encode gradient with duration $M\Delta t$, and then followed by a re-phasing for half area, each constituent hard pulse p_m excites some magnetization that remains undisturbed by subsequent hard pulses and precedes under the influence of the remaining gradients. With no other y applied, each hard pulse generates a Fourier sample $k_m = (1/2M - m)G_y\Delta t$, scaled by the complex value p_m. In the low flip-angle approximation, the signal received due to this arbitrary RF pulse is a superposition of the individual hard pulse contributions [82]

$$a(p, k_x) = \int_{-\alpha}^{\alpha} \int \int \rho(x, y, z)(\sum_{m=1}^{M} p_m e^{i2\pi k_m y}) \\ e^{i2\pi(k_x x + k_y y)} dxdydz \tag{2.25}$$

$$= \sum_{m=1}^{M} (k_m, k_x), \tag{2.26}$$

where p is a row vector containing the p_m, i.e., $p = (p_1, \cdots, p_M)$. With sufficient gradient strength, the k_m can precisely reflect the phase encodes k_y of the Fourier basis. The Fourier transform term in Equation.(2.25) is the spatial profile of transverse magnetization generated by the RF pulse $\tilde{p}(y) \approx F\{p\}$. Equation.(2.26) can be rewritten in matrix-vector form as $a = pS$, when the length-M input vector p describes the RF excitation waveform,a is the length-N output response vector of sampled data, and S is the $M \times N$ K-space matrix corresponding to the spin distribution. One may now consider MR image encoding using arbitrary RF inputs p. Given an arbitrary invertible matrix P, we can use its rows as the RF pulse of each repetition of non-Fourier encoding. Populating the sampled responses into the rows of matrix A, the MR imaging can be expressed as [82]

$$A = PS, \tag{2.27}$$

which yields the K-space matrix by using an appropriate inverse

$$S_t = P^{\dagger}A. \tag{2.28}$$

Finally, the inverse transform of S_t yields the desired image. Based on (2.28), many non-Fourier transforms for input vectors have been studied [82] [79] and also used [77] [83] for MRI.

2.3 CS-based MRI Processing

MRI scanning time mainly depends on the number of samples taken during acquisition. Therefore, any application of CS to MRI should provide improvement in image acquisition speed. Since current MRI scanning time lasts at least 30 minutes, fast MRI will reduce patient discomfort and image distortion due to patient movement during acquisition.

State of the art development of CS-based MRI can be divided into three categories, 1) Fourier transform using CS [2,84–90], where the conventional Fourier transform is maintained; 2) Use of sparse matrix Ψ in combination with the conventional Fourier transform and perform CS reconstruction [61,91–96]. These methods are a step closer to having a complete CS based MRI system; and 3) CS-based non-Fourier data acquisition and corresponding reconstruction method that use random encoding in place of the Fourier encoding along the phase encoding direction [97,98]. These CS-based MRI processing are simulated and tested on real MRI scanners as an add-on component.

Even though there has been extensive research on CS-based MRI, there are no commercial products available as yet. The dependencies on the other hardware components of the MRI scanner are high.

2.4 Image Processing with JPEG 2000

JPEG 2000 is an image compression standard and coding system, created in the year 2000 by the joint photographic experts group (JPEG) committee. This standard supersedes their original discrete cosine transform-based JPEG standard with a newly designed, wavelet-based method, called the lifting wavelet transform. The aim of JPEG 2000 is not only improving compression performance over JPEG but also adding important features such as image scalability and editability. Variable rates (very high and very low) are supported and the ability to handle a very large range of effective bit-rates is one of the strengths of JPEG 2000.

A performance comparison graph is shown in Fig. 2.2. For low compression ratios, JPEG produces slightly better images, whereas for medium to high compression ratios one can attain higher quality with JPEG 2000. It also provides excellent compression performance and is used in many applications like printing, photography and medical imaging.

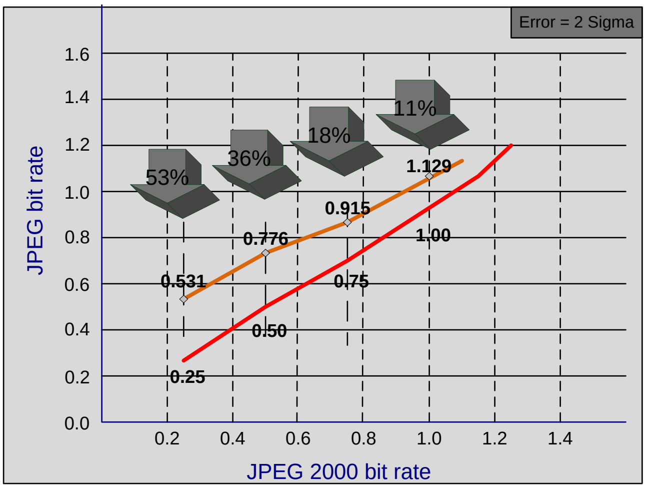


Figure 2.2: Performance comparison of JPEG 2000 vs. JPEG.

In JPEG 2000 encoder as shown in Fig. 2.3, an image is first level-shifted and then a component transform is performed to obtain three color components. A discrete wavelet transform is applied to these color samples and transform coefficients are obtained. After performing the discrete wavelet transform, each sub-band is divided into code-blocks, which are then independently processed by an embedded block coding with optimized truncation (EBCOT) tier-1 engine. The EBCOT tier-1 engine has two most computational intensive components, namely, bit-plane coding (BC) and arithmetic encoding (AE). The AE module implements binary, shift-based arithmetic coding to efficiently encode the symbols that it receives from the EBCOT engine. The context that is sent by the EBCOT engine provides an extra meaning to the symbol that needs to be encoded. The bit-plane coder processes the bit planes as coding passes and generates a sequence of symbols called the context (CX) and decision (D) pair. The D bit is also referred to as symbol.For example, a binary '1' symbol that originates from a significance propagation pass is different from a '1' symbol that is generated from a magnitude refinement pass. Consequently, each of these symbols are accompanied by distinct context labels and encoded in a very different manner. The previous sequence of symbols dominates the current state of a coding context and each of these context states are stored in form of tables. The context states are unique in nature and determines the probability of a less probable symbol in the AE module. The state of a given context is updated using a fixed state transition table. This probability estimation and encoding method is termed MQ-coding.

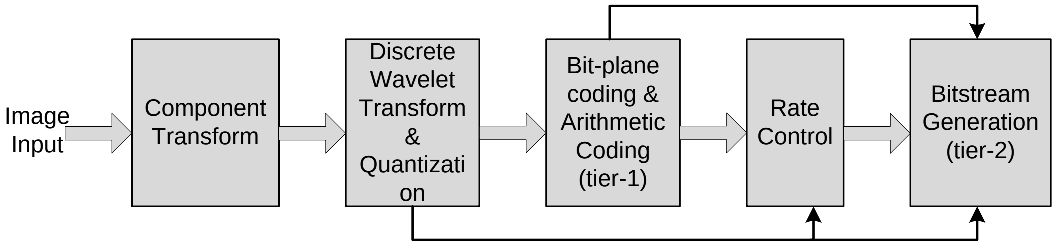


Image Input
Component Transform
Discrete Wavelet Transform & Quantizati on
Bit-plane coding & Arithmetic Coding (tier-1)
Rate Control
Bitstream Generation (tier-2)

Figure 2.3: JPEG 2000 encoder block diagram.

2.5 Field Programmable Gate Array Architecture

Altera was the first to introduce the 8-input fracturable look-up table (LUT) with the Stratix II family in 2004. At its core is the adaptive logic module (ALM) with 8 inputs, which can implement a full 6-input LUT (6-LUT) or select 7-input functions. The ALM can also be efficiently partitioned into independent smaller LUTs, providing the performance advantage of larger LUTs and the area efficiency of smaller LUTs. The Stratix series of FPGAs also excels in routing through the MultiTrack interconnect. As a result, Altera FPGA architecture is at least one generation ahead of the competition, and routing architecture is two generations ahead.

The key to the high-performance, area-efficient architecture is the ALM. It consists of combinational logic, two registers, and two adders as shown in Fig. 2.4. The combinational portion has eight inputs and includes a LUT that can be divided between two adaptive LUTs (ALUTs) using Alteras patented LUT technology. An entire ALM is needed to implement an arbitrary six-input function, but because it has eight inputs to the combinational logic block, one ALM can implement various combinations of two functions. A LUT is typically built out of SRAM bits to hold the configuration memory (CRAM) LUT-mask and a set of multiplexers to select the bit of CRAM that is to drive the output. To implement a t-input LUT; a LUT that can implement any function of t inputs$2t$ SRAM bits and a $2t : 1$ multiplexer are needed.

The key to high-performance Stratix IV FPGAs is the area-efficient ALM. It has 8 inputs with a fracturable look-up table (LUT) that can be divided into two adaptive LUTs (ALUTs) using Altera's patented LUT technology. Each ALM is capable of: (1) A full 6-input LUT or select 7-input LUT; (2) Two independent outputs of multiple combinations of smaller LUT sizes for efficient logic packing; (3) Implementing complex logic-arithmetic functions without additional resources. The fracturable LUT, two full adders, two registers, and additional

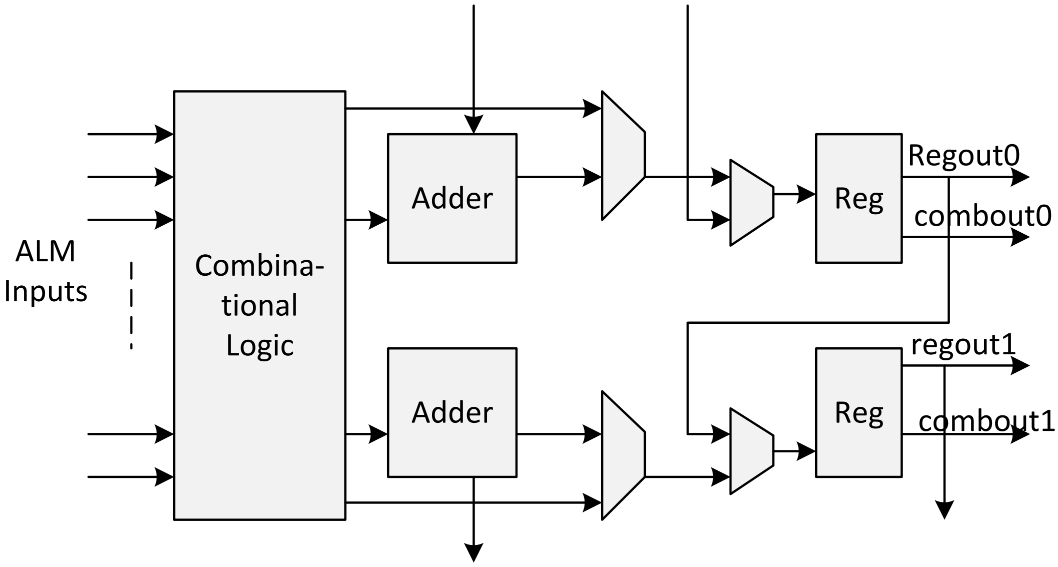


Figure 2.4: Internal structure of an ALM.

logic enhancements that enable the ALM to be partitioned into two independent LUTs for maximizing efficiency, make Stratix IV FPGAs the fastest and biggest 40-nm FPGAswith no wasted logic. This is the major advantage of Stratix IV FPGA architecture when the applications need high-speed and low-complexity FPGA. Stratix IV devices are 35 percent faster and can effectively pack 80 percent more logic compared to the nearest competing logic cell, thereby cutting costs by packing more logic in a smaller, less expensive device.

Chapter 3

2D and 3D MRI Processing Using CS-Based Complex Measurements

3.1 Introduction

In recent times, compressive sensing (CS) has proved its potential to reduce data acquisition time for magnetic resonance images (MRI). For a CS-based MRI imaging scheme to be effective, the signal of interest should be sparse or compressible in a known representation, and the measurement scheme should have good mathematical properties with respect to this representation. Although the Fourier transform has been commonly used for MRI data, it does not strongly satisfy CS mathematical properties. This limits the achievable time reduction factors necessary for 3D-MRI.

In this chapter, the aim is to exploit the sparsity which is implicit in MR images, and develop an approach based on exploiting the spatial and temporal redundancies. This, to some extent, would degrade the signal-to-noise ratio (SNR), but is worth when the amount of acquired data can be reduced. Implicit sparsity means transform sparsity, i.e., the underlying object of interest happens to have a sparse representation in a known and fixed mathematical transform domain. To begin with, consider the identity transform, so that the transform domain is simply the image domain itself. Here sparsity means that there are relatively few significant pixels with nonzero values. For example, angiograms are extremely sparse in pixel representation. More complex medical images may not be sparse in pixel representation, but they do exhibit transform sparsity, since they have a sparse representation in terms of spatial finite differences, their wavelet coefficients, or other transforms.

Based on compressive sensing methods, the attempt is to provide the following contributions for 2D and 3D MRI:

- Non-Fourier based MRI data acquisition using the complex Hadamard matrix and show that, when used with Daubechies-4 wavelet transform satisfies the RIP. This complex Hadamard matrix structure is proposed and used for the first time;
- Complex measurements based CoSaMP reconstruction, whose computational complexity is less than the original CoSaMP;
- Comparison of our proposed method with the conventional Fourier sampling and also its efficiency with respect to computational complexity. Furthermore, we demonstrate our proposed method through the measure of peak signal-to-noise ratio (PSNR) and compare with the commonly used orthogonal matching pursuit (OMP) [99] algorithm; and
- Compare results with the NFCS-3D Fista [2] method, and show that our proposed method has higher PSNR for a 3D phantom, when implemented on similar lines as outlined in [2].

3.2 Related Work

Conventional MRI based processing relies on the Fourier transform for data acquisition, including 3D and dynamic MRI [2,87–90]. In many instances, it is observed that the Fourier matrices are not necessarily well suited for CS reconstruction for arbitrary sparse matrix Ψ [97]. Since Fourier encoding is not universal, the incoherent condition is only weakly satisfied with respect to sparse transforms. Some research also suggests that, by using additional slice-selective excitation in a wavelet basis, it is possible to improve 3D image CS reconstruction [3]. For example, a wavelet transform in a coarse scale has its energy concentrated rather than spread out in the Fourier domain, which suggests the incoherence condition is barely satisfied [57]. This shows that the use of matrices other than the Fourier ones could possibly lead to better results.

Several matrices have been proposed in the literature for CS, such as independent identically distributed Gaussian matrix [100], and Bernoulli matrices as in [101] [102]. Their main advantage is that they are universally incoherent with any sparse signal and thus, the number of compressed measurements required for exact reconstruction is almost minimal. However, they inherently have two major drawbacks in practical applications, namely, huge memory buffering for storage of matrix elements and high computational complexity due to their completely unstructured nature [59]. Another group of matrices based on Fourier and

Hadamard were also proposed [103] where it was called the partial fast Fourier transform (PFFT) and scrambled block Hadamard ensemble respectively. PFFT exploits the fast computational property of fast Fourier transform (FFT) and thus, significantly reduces the complexity of a sampling system. However, it is only incoherent with signals which are sparse in the time domain, severely narrowing its scope of applications.

A few reconstruction algorithms have been in popular use for image processing in CS. To name a few, orthogonal matching pursuit (OMP) [104], a modified version of gradient projection for sparse reconstruction (GPSR) [105]. Though these algorithms are fast, they require a large number of samples which could be time-consuming to acquire. Also, algorithms like GPSR and its many varied versions are computationally burdensome. There are a few more reconstruction algorithms like compressive sampling matching pursuit (CoSaMP) [53] that have been suggested for image/video processing. The CoSaMP algorithm considers the shortcomings of other existing reconstruction algorithms and is computationally effective.

Some 3D-MRI methods [92, 106, 107] have been recently proposed in the literature. A forward-backward splitting based reconstruction for 3D-MRI is proposed in [2], for highly undersampled sequences. All of these tackle the problem of reconstruction with respect to Fourier data acquisition. As already discussed above, using the Fourier transform has some drawbacks when used for CS. In [108], CS-based 3D-MRI reconstruction using many-core graphic processing units (GPU) architectures are proposed that can achieve fast data acquisition than using 3D FFT and reconstruction with quasi-Newton algorithm [109].

Hence, considering the drawbacks of the Fourier-based data acquisition for 2D and 3D MRI, a non-Fourier based acquisition is inevitable. This method would overcome the drawbacks of Fourier-based methods and when combined with a suitable reconstruction algorithm yields outputs that are comparable with the conventional 2D and 3D MRI.

3.3 Compressive Sensing for 2D and 3D MRI

The properties that enable CS for MRI is the sparsity of the transform data and the coded nature of MR acquisition. The three key factors of CS is transform sparsity, mutual incoherence and non-linear reconstruction, and MRI processing obeys these properties [58]. Hence, data modeling is done as per CS theory as follows: An orthonormal basis where a real-valued, finite-length, discrete signal x in $\mathbb{R}^N$ is represented by an $N \times 1$ column vector $\{\psi_i\}_{i=1}^{N}$, since an image can be vectorized into an one-dimensional array. This signal x can be expressed using an $N \times N$ basis matrix $\Psi = [\psi_1|\psi_2|\ldots|\psi_N]$ as

$$x = \sum_{i=1}^{N} \psi_i \alpha_i = \Psi\alpha, \tag{3.1}$$

with vector ψ_i as columns, α is the $N \times 1$ column vector of the coefficients

$$\alpha_i = \langle x, \psi_i \rangle = \psi_i^* x, \tag{3.2}$$

where ψ_i^* is the transpose of ψ_i.

Signal x is said to be k-sparse if only k of the α_N coefficients in (3.1) are non-zero and the rest are zeroes. For the purpose of direct signal acquisition, an $M \times N$ matrix Φ that has measurement vectors ϕ_j^* as rows, and the $M < N$ inner products between α and vectors $\{\phi_j\}_{j=1}^{M}$ as $y_j = \langle \alpha, \phi_j \rangle$. Arranging y_j measurements in $M \times 1$ vector form and then substituting (3.2) for α, the following equation for y is obtained

$$y = \Phi\alpha = \Phi\Psi x = \Theta x, \tag{3.3}$$

where Θ is an $M \times N$ sensing matrix. The measurements y will be the random measurements that are sufficient for exact reconstruction of the MRI.

Random point k-space sampling in all dimensions is generally impractical as the k-space trajectories have to be relatively smooth because of hardware and physiological considerations. Instead, we aim to design a practical incoherent sampling scheme that mimics the interference properties of pure random undersampling as closely as possible yet allows rapid collection of data.

In this section, the theory of non-Fourier encoding of MRI and CS is combined and applied to MRI acquisition and reconstruction. Relating the proposed matrices to (3.3), the complex Hadamard matrix is the matrix Φ and we use Daubechies-4 wavelet as the matrix Ψ. This wavelet is used, since it satisfies the CS properties and has been popularly used in MRI [97]. The Daubechies-4 wavelet is generated based on these CS properties and for the purpose that the resulting sensing matrix satisfies the restricted isometry property (RIP).

3.3.1 Complex Hadamard Matrix

A complex Hadamard matrix (CHM) is defined as a square matrix composed of elements +1, -1, +i, and -i, whose row vectors are orthogonal. If H is a complex Hadamard matrix, then H^* represents the complex conjugate transpose of the matrix. It possesses a unique property known as the half-spectrum property, where only half of the complex spectrum is necessary to restore the original data completely. The existence of such a property is important for applications in signal processing for discrete or complex signals. Moreover, the CHM is preferred over the Fourier sampling, since it is requires less computations compared to Fourier, which is an important aspect when the system is required to be implemented as a hardware. In the discussions below, the analysis is confined to a 2×2 matrix where

$$H_m = H_2 \begin{bmatrix} 1 & i \\ -i & -1 \end{bmatrix} \otimes^m . \tag{3.4}$$

where $\otimes^m$ is the right hand side Kronecker product being applied m times.

Let H_M be a $M \times M$ complex Hadamard matrix and $h(j,k)$ be an element in it, where $0 \leq j,k \leq M-1$ and $N = 2^m$. Then, the transformation is given by

$$h(j,k) = (-1)^{\sum_{x=0}^{m-1} j_x + \frac{1}{2}(j_x \oplus k_m)}, \tag{3.5}$$

where $\langle j_{m-1}, j_{m-2}, \ldots, j_0 \rangle$ and $\langle k_{m-1}, k_{m-2}, \ldots, k_0 \rangle$ denote the respective binary representation of the decimal j and k respectively and $\oplus$ is the direct sum operator. Comparing with the real Walsh-Hadamard transform, if $w(j,k)$ denotes the element of the transform at row j and column k, it may be noted that $h(j,k) = w(j,k)$ iff $j_x + 3k_x = 4j_x, k_x$ [110].

Specifically, the CHM is generated based on the products of the row vectors of a complex Rademacher matrix as follows

$$H_N(m,k) = \prod_{r=0}^{n-1} R_n(r,k)^{b_r}, \tag{3.6}$$

where $R_n(r,k) = \text{CRAD}(r, \frac{4k+1}{2^{n+2}})$ is the (r,k) element of the complex Rademacher matrix R_n, $m = b_{n-1}2^{n-1} + \ldots + b_1 2^1 + b_0 2^0$ and $b_r = 0$ or 1. The complex Rademacher function (CRAD) over a normalized time base $0 \leq t \leq 1$ is given by

$$\text{CRAD}(0,t) = \begin{cases} 1, & t \in [0, \frac{1}{4}] \\ j, & t \in [\frac{1}{4}, \frac{1}{2}] \\ -1, & t \in [\frac{1}{2}, \frac{3}{4}] \\ -j, & t \in [\frac{3}{4}, 1] \end{cases}$$

and CRAD(r,t) is obtained by compressing CRAD(0,t) in the horizontal direction by a factor of 2^r [111].

Furthermore, the transform can be obtained by performing matrix factorization as follows

$$H_M = \begin{bmatrix} H_{M/2} & S_{2^{m-1}} H_{M/2} \\ H_{M/2} & -S_{2^{m-1}} H_{M/2} \end{bmatrix} \begin{bmatrix} X_e \\ X_0 \end{bmatrix} H_2 \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.7}$$

where H_2 is the boundary condition, $S_{2^k} = \begin{bmatrix} I_{2^{k-1}} & 0 \\ 0 & jI_{2^{k-1}} \end{bmatrix}$,
$X_e = [X(0), X(2), ..., X(M-2)]^T$ and
$X_0 = [X(1), X(3), ..., X(M-1)]^T$.

The complex Hadamard matrix adopted in our work is of the following form

$$H_4 = \Phi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}. \tag{3.8}$$

In our proposed system, Hadamard encoding derives the matrix H from the $M \times N$ CHM, whose rows are the spatial excitation $\tilde{p}(y)$. For reduced basis adaptive imaging, the encoding matrix $\hat{H}$ is $k \times M/f$, where $k \leq M/f$, reflecting the additional non-Fourier efficiency. In every case, the sampled signals form the matrix

$$A^{(f)} = \hat{P}S^{(f)}, \tag{3.9}$$

of size $k \times LN$, $k \leq M/f$. The non-Fourier inversion of (3.9) yields the individual coil subsampled k-sparse matrices as

$$S_e^{(f)} = \hat{P}^{\dagger} A^{(f)}. \tag{3.10}$$

The matrix $S^{(f)}$ can be separated by reversing the concatenation, and its constituents can be used with the MRI algorithm of choice to reconstruct the full $M \times N$ image of the field of view (FOV).

In low flip-angle approximation, the radio-frequency (RF) encoding matrix $\hat{P}$ is derived from the Fourier transform of each row of H_M. The rows of $\hat{P}$ are then used as RF excitations in consecutive experiment repetitions. For the 4-element array, once all repetitions are completed, the samples are arranged in the $128 \times (256 \times 4)$ composite response matrix $A^{(f)}$, which then represents the Hadamard spatially encoded FOV contents. Since Hadamard matrices are orthogonal, the inversion is achieved by multiplying this acquired composite response matrix by the Hermitian conjugate of the RF encoding matrix, i.e., $\hat{P}_{\dagger} = \hat{P}_{H_e}$ in (3.10). This results in the subsampled composite k-sparse matrix $S_e^{(f)}$ which is further divided into four parts, each sized 128×256, corresponding to each coil $l = 1, \ldots, 4$.

For a matrix to be used for CS, it is important that its necessary conditions are satisfied. In this discussion, we show the RIP of the sensing matrix Θ by proving that matrix Φ satisfies the RIP. Matrix Θcis the combination of Φ and

$$\Psi = \sqrt{2}\left(\frac{1+^{-j2\pi}}{2}\right)^2 \exp^{j2\pi} \tag{3.11}$$

as in (3.8). It is shown that matrix Φ satisfies the RIP and for this we follow the proof of the RIP for structured matrices. From

$$(1-\delta_s)||\alpha||_{l_2}^2 \leq ||\Phi\alpha||_{l_2}^2 \leq (1+\delta_s)||\alpha||_2^2, \tag{3.12}$$

it is shown that for an $M \times N$ matrix $\Phi \in \mathrm{RIP}(k, \delta_s)$ if the following inequality holds for some isometry constant $\delta_s \in (0, 1)$.

The following lemma and definition will also be used to arrive at the proof of Φ satisfying the RIP.

Lemma 3. [41]*: If Φ has unit-norm columns and the coherence parameter is μ, then $\Phi \in RIP(k, \delta_s)$ with $\delta_s \leq (k-1)\mu$ for k-sparse vectors.*

If Φ satisfies the RIP and the matrix being incoherent with Ψ, will ensure that the combined sensing matrix is bound by the RIP conditions. Further to this, the Daubechies-4 wavelet is mutually incoherent with Hadamard matrix [112]. The mutual coherence is considered for the following reason. In general, the signal of interest may not be sparse in a particular basis but in some orthonormal basis Ψ. Then, the signal of interest becomes Ψx with x being sparse and then we consider the matrix $\Theta = \Phi\Psi$ [113]. A low mutual coherence value indicates that a signal which is sparse in one basis has a dense representation in another base. Using two matrices with maximum mutual incoherence between leads to a sparse representation of signals, and higher the incoherence, lesser are the measurements required. This makes it possible to recover the signal correctly, and thus suitable for CS-based 3D-MRI processing.

Now we prove that there exists, for a certain constant δ_s, a matrix Φ satisfying the RIP of order k. A matrix is said to have the RIP of order k if δ_s is very close to 1. The proof is as follows,

Theorem 7. *Let $\Phi \in \mathbb{R}^{M \times N}$ is a complex Hadamard matrix, Ψ be the Daubechies-4 wavelet as in (3.11)and μ be the coherence parameter. Then $\Theta = \Phi\Psi$ can be used for CS, since Φ satisfies the RIP of order k with a constant δ_s*

Proof. If k is fixed to be as $k < N$ and $0 < \delta_s < 1$ and from Lemma 3, then for complex Hadamard matrix Φ we have,

$$1 - \delta_s \leq \frac{||\Phi x||_2}{||x||_2} \leq 1 + \delta_s, \tag{3.13}$$

for all $x \in \mathbb{R}^N$ with probability of atleast

$$1 - 2\left(\frac{12}{\delta_s}\right)^k e^{-m\mu(\frac{\delta_s}{2})}. \tag{3.14}$$

As Φ is linear, we only need to consider the cases where $||x||_2 = 1$. For unit vectors, that is for all $x \in \mathbb{R}^N$ we have from [114] that

$$\left(\frac{12}{\delta_s}\right)^k 2e^{-m\mu\frac{\delta_s}{2}}. \tag{3.15}$$

Rearranging the probability terms in (3.12), we get

$$1 - \frac{\delta_s}{2} \leq \frac{||\Phi\alpha||_2}{||\alpha||_2} \leq 1 + \frac{\delta_s}{2}, \tag{3.16}$$

with probability more than

$$1 - 2\left(\frac{12}{\delta_s}\right)^k e^{-m\mu(\frac{\delta_s}{2})}, \tag{3.17}$$

for all $\alpha \in \mathbb{R}^N$. Now we define the smallest number such that $||\Phi x||_2 \leq (1+\alpha)||x||_2$. To show that $\alpha \leq \delta_s$, we have, for any unit vector $x \in \mathbb{R}^N$ there exists α such that $||x - \alpha||_2 \leq \frac{\delta_s}{4}$. Let v_x be a vector such that $||x - v_x||_2 \leq \frac{\delta_s}{4}$. Then

$$||\Phi x||_2 \leq ||\Phi v_x||_2 + ||\Phi(x - v_x)||_2 \leq 1 + \frac{\delta_s}{2} + (1+\alpha)\frac{\delta_s}{4}. \tag{3.18}$$

For a smallest α, $||\Phi x||_2 \leq (1+\alpha)||x||_2$ for all $x \in \mathbb{R}^N$, it is required that

$$\alpha \leq \frac{\delta_s}{2} + (1+\alpha)\frac{\delta_s}{2} \Longrightarrow \alpha \leq \frac{3\delta_s}{4-\delta_s} \leq \delta_s. \tag{3.19}$$

This proves that

$$\frac{||\Phi x||_2}{||x||_2} \leq 1 + \delta_s, \tag{3.20}$$

for all $x \in \mathbb{R}^N$.

And, the lower bound is given by,

$$\begin{aligned} ||x||_2 &\geq ||\Phi v_x||_2 - ||\Phi(x - v_x)||_2 \\ &\geq 1 - \frac{\delta_s}{2} - (1+\delta_s)\frac{\delta_s}{4} \geq 1 - \delta_s. \end{aligned} \tag{3.21}$$

This completes the proof that Φ satisfies the RIP.

Utilizing the relationship between μ and δ_s and Applying the Welch bound inequality [30] to (2), $\mu \in \left[\sqrt{\frac{N-M}{M(N-1)}}, 1\right]$, where the lower bound is also known as the Welch bound and, when $N \gg M$, $\mu = \frac{1}{\sqrt{M}}$.

Moreover, there exists a universal lower bound [115]

$$\mu \gg \left(\sqrt{\frac{\log N}{M \log(M/\log N)}}\right) \geq \frac{1}{\sqrt{M}} \tag{3.22}$$

for $2\log N \leq M \leq N/2$ and all Θ. Hence, by estimating δ_s in terms of $\mu(\Theta)$ we cannot construct an $M \times N$ matrix of order larger than $\sqrt{M}$ and $\delta_s < 1$. Therefore, from Lemma 2 and (2), we obtain

$$\delta_s = (k-1)\left(\sqrt{\frac{\log N}{M \log(M/\log N)}}\right) \tag{3.23}$$

as a constant for matrix Θ satisfying the RIP of order k. ■

For the reconstruction of the acquired MRI data, the CoSaMP [53] algorithm outlined in Algorithm 2 is used. The inputs are sampling matrix Θ, noisy sample vector e, sparsity level k and output is an k-sparse approximation of the target signal x. Note that the Algorithm 2 is different from the standard CoSaMP [53] in the sense that the sampling matrix Θ is actually the sensing matrix. Hence, the CoSaMP used in this work performs complex measurements based reconstruction whereas CoSaMP [53] has a randomly generated matrix as the sampling matrix.

Algorithm 2 CoSaMP for MRI

```
$z = 0, x^z = 0$ {Initialization}
while halting criterion false do
    $v \leftarrow e - \Theta x^z$ {Updating samples}
    $y \leftarrow \Theta^* v$ {Proxy signal formation}
    $\Omega \leftarrow supp(y_2 k)$
    $T \leftarrow \Omega \cup supp(x^z)$ {Merge supports}
    $a \mid T \leftarrow \Theta_T e$ {Signal estimation using least squares}
    $a \mid T_c \leftarrow 0$
    $x^{z+1} \leftarrow a_k$
    $z = z + 1$
end while
$x \leftarrow x^z$
```

One of the main reasons of using CoSaMP reconstruction is that it provides rigorous bounds on computational costs and storage [53]. Moreover, it holds a temporal solution with k non-zero entries, and in each iteration it adds an additional set of $2k$ (instead of k) candidate non-zeros that are most correlated with the residual. After the pruning step, only the largest k elements are taken and a constant number of iterations are sufficient until stopping criterion is met.

Having the measurement matrix Φ with the isometry constant δ_S and $y = \phi x + e$ is a vector of samples of an arbitrary signal contaminated with arbitrary noise, and e is the noise vector, then CoSaMP produces a k-sparse approximation a that satisfies,

$$\|x - a\|_2 \leq C \max\{\eta, \frac{1}{\sqrt{k}}\|x - x_{k/2}\|_1 + \|e\|_2\} \tag{3.24}$$

where η is the precision parameter and $x_{k/2}$ is the best $k/2$-sparse approximation to x.

3.3.2 Computational Complexity

In this section, we discuss the selection of CoSaMP reconstruction for CHM-based CS, based on its computational complexity and implementation suitability on a hardware platform. Table 3.1 shows the complexity of some of the well-known CS algorithms that are used in image and MRI reconstruction. The complexity is calculated based on an $M \times N$ matrix of a k-sparse basis and α is the redundancy. As observed, CHM-based CoSaMP has the lowest computational complexity of the order of $O(M \log N)$. Another important fact of CoSaMP in general is that it does not depend on the sparsity level k and redundancy parameter α. This greatly affects the running time and also the algorithm complexity and hence the choice of CoSaMP in our proposed method.

Table 3.1: Computational complexity of some popular CS reconstruction algorithms used for MRI. The complexity is based on a $M \times N$ matrix for a k-sparse basis, u is the median filter and α is the redundancy.

CS reconstruction algorithm	Complexity
Subspace Pursuit (SP) [54]	$O(MNk)$
Orthogonal Matching Pursuit (OMP) [99]	$O(\alpha N^2)$
Gradient Projection for Sparse Representation (GPRS) [105]	$O(N^2)$
Matching Pursuit(MP) [24]	$O(\alpha N^2)$
Regularized OMP [116]	$O(MNk)$
NFCS-3D [2]	$O(uN\alpha)$
CoSaMP [53]	$O(N \log N)$
CoSaMP with CHM (proposed)	$O(M \log N)$

Other than reduction in data acquisition time for 3D-MRI, we also consider that our proposed methods should be physically implementable with low-complexity and less hardware resources. In conventional methods, processing required for data acquisition is enormous due to its repetitive nature and leads to a slower system. One of the main issues is the number of multiplications required to obtain a single 2D slice of a 3D image. And, when 3D processing has to be undertaken, the processing increases N-fold. This also makes the MRI system bulkier, while trying to reduce the acquisition time through parallel processing techniques. One possible way of reducing the time required for processing is by designing systems, that require less hardware resources. Hardware resources is directly proportional to the processing elements required for computation and its complexity. A FPGA-based hardware implementation is demonstrated in Chapter 5.

3.4 Numerical Results

In this section, we present some of the simulation results that we conducted to demonstrate the effectiveness of our proposed method. All the algorithms and simulations are implemented in MATLAB and the tests are performed on a 2.8 GHz AMD Phenom processor with 8 GB RAM on a Microsoft Windows

7 operating system. The simulations are performed in two different sets, i.e., one for 2D-MRI and another for 3D-MRI. Several real MRI data sets obtained from [1] are simulated.

3.4.1 Simulations for 2D-MRI

Experiments are conducted in order to demonstrate the efficiency of the proposed measurement matrix, namely the CHM and its suitability for MRI images. Several MRI test images are simulated. In all the cases, the simulations are performed using the following three measurement matrices with a fixed number of data samples:

- Complex Hadamard matrix: is the proposed measurement matrix and has complex entries as discussed in Subsection 3.3.1;
- Random Fourier matrix: is used in most of the standard CS-based MRI systems [84] [94] [95]. A comparison will show that the proposed CHM outperforms this measurement matrix; and
- Random Gaussian matrix: is a commonly used measurement matrix for CS in general for images. We will demonstrate that our method outperforms this sampling.

Firstly, in order to justify the efficiency of the proposed matrix, simulations are performed and the rate-distortion performance graph is obtained as shown in Fig. 3.1. The graph is plotted for less than 2K measurements. It is noteworthy how the complex matrix performs in comparison to other matrices used for test, as can be observed from Fig. 3.1. At lower sampling rates, the PSNR using complex Hadamard matrix is significantly higher than that of Fourier or Gaussian matrices. This is one of the important aspects required in compressive sampling, since we aim to obtain high performance with minimal samples. There is a difference of approximately 10 dB between CHM and Gaussian sampling, and this is maintained throughout various sampling rates. Due to this nature of CHM, it is most suitable in reducing the complexity of the system by performing computation with fewer samples and still gaining reasonable quality.

Furthermore, tests are performed in order to check the suitability of the proposed reconstruction method with other popularly used reconstruction methods like gradient projection for sparse reconstruction (GPSR) [105], orthogonal matching pursuit (OMP) [99], L1-minimization and iterative shrinkage/threshold (IST) [117]. Fig. 3.2 shows the problem complexity and CPU time taken to perform the reconstruction. These reconstruction methods are evaluated with measurements taken using the proposed CHM.

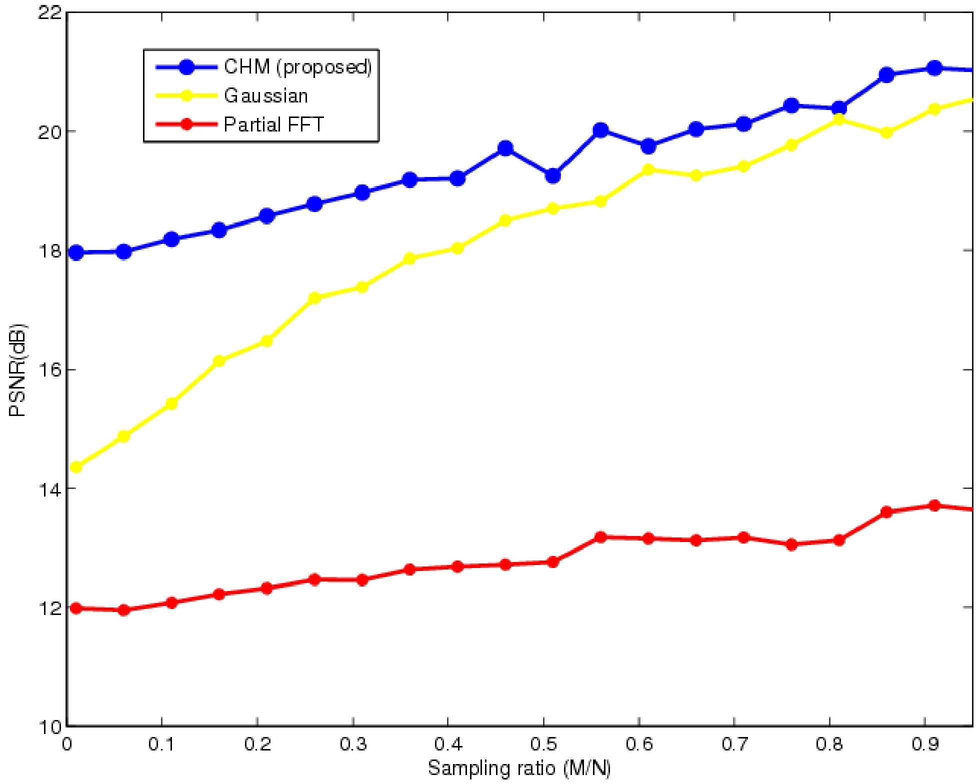


Figure 3.1: Rate distortion performance of various measurement matrices.

To validate our proposed system, three test images of 256×256 size are considered. The experiments are conducted using 4K and 10K measurements. To compare our system with other state-of-the-art methods, random Fourier and random Gaussian matrices are also tested in conjunction with CoSaMP reconstruction. All the algorithms are implemented in MATLAB and all the tests are performed on a 2.8 GHz AMD Phenom processor with 3 GB RAM. The running time of the proposed system is also noted. Each image is processed within 0.76 sec, which is the time taken from data acquisition to reconstruction of the image. Alongside, a similar processing of MRI data is performed using Fourier and Gaussian matrices for the purpose of final PSNR comparison. Hence, for each set of MRI data there are three CS systems with different measurement matrices executed. The observations are depicted in Figs. 3.3, 3.4 and 3.5. PSNR comparison for all the test images are presented in Table 3.2 and 3.3. From these data, it is evident that even for 4K samples, the proposed method has a PSNR of 25 dB and above for all the test cases. In the 4K range, it is about 3 dB higher than the random Fourier matrix and outperforms the random Gaussian matrix by approximately 10 dB.

Comparing the PSNR of the reconstructed figures, it can be noted that the proposed CHM measurement matrix outperforms by a PSNR of at least 10 dB in most cases, when compared to the Gaussian and Fourier sampling. The proposed method shows a PSNR of 40 dB for most of the images for just 10K samples,

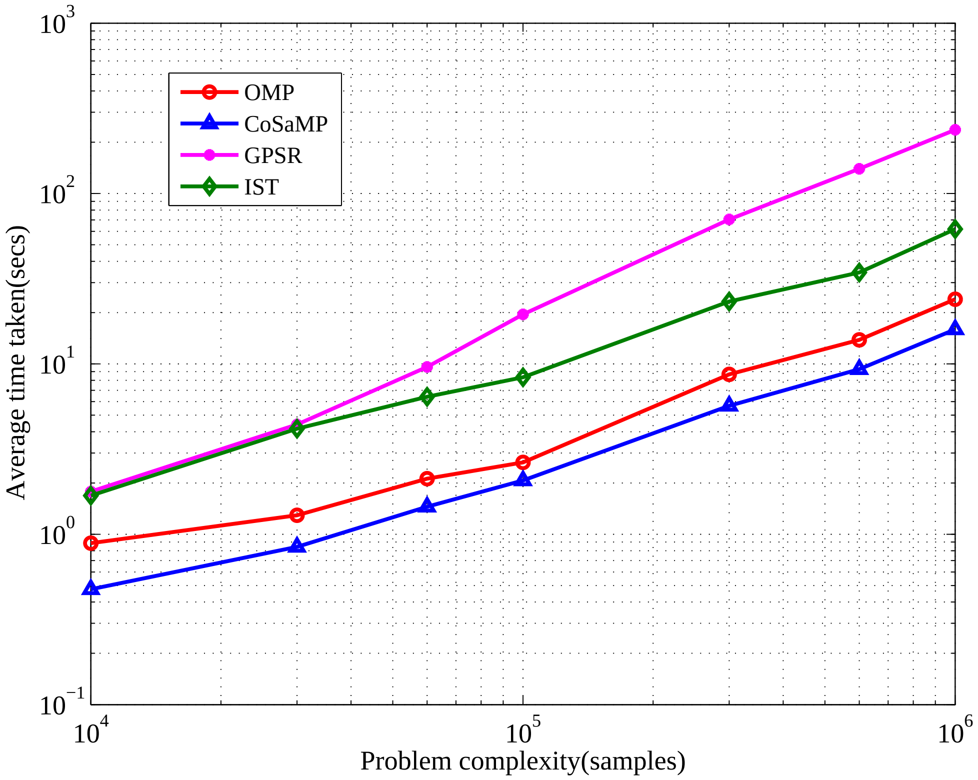
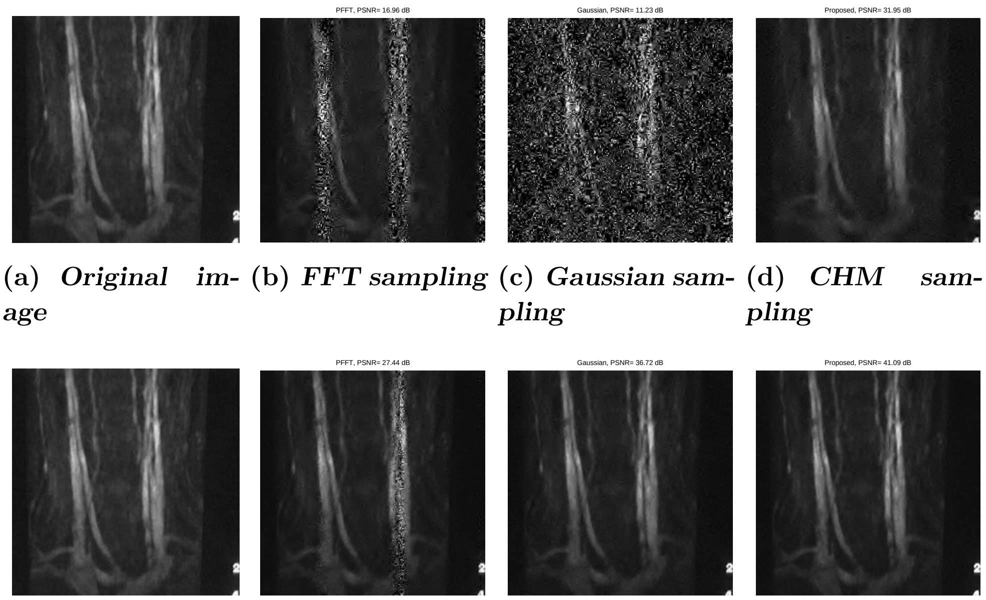


Figure 3.2: Runtime performance of reconstruction methods with respect to image complexity.



(a) ***Original image***

(b) ***FFT sampling***

(c) ***Gaussian sampling***

(d) ***CHM sampling***

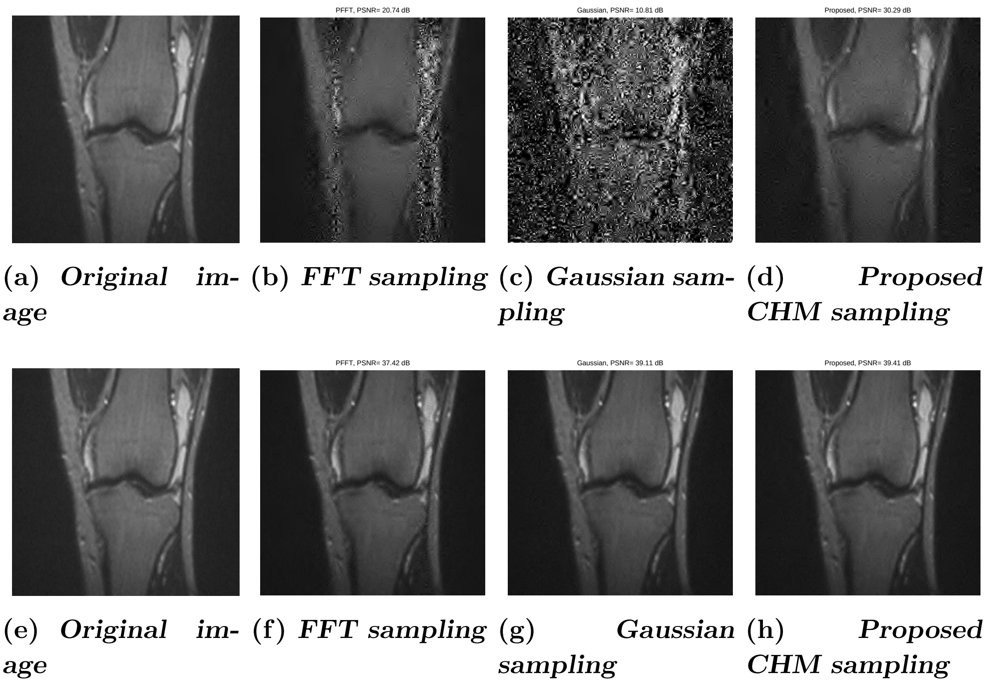
(e) ***Original image***

(f) ***FFT sampling***

(g) ***Gaussian sampling***

(h) ***CHM sampling***

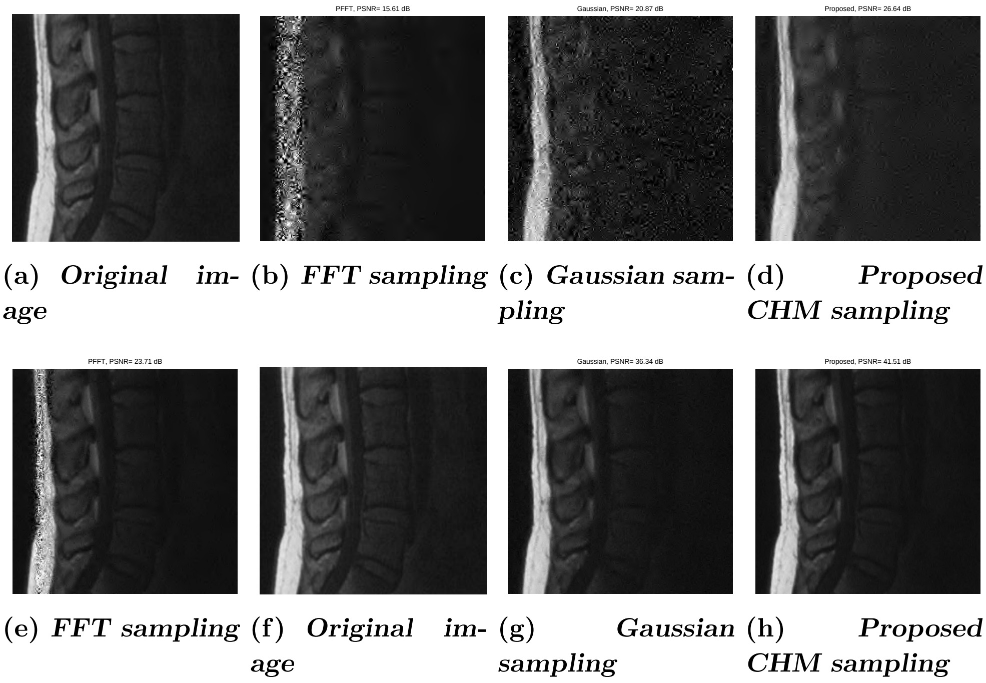
Figure 3.3: Reconstructed data for a 256×256 angio MRI image with 4K samples (from (b) to (d)) and 10K samples (from (f) to (h))



(a) ***Original image*** (b) ***FFT sampling*** (c) ***Gaussian sampling*** (d) ***Proposed CHM sampling***

(e) ***Original image*** (f) ***FFT sampling*** (g) ***Gaussian sampling*** (h) ***Proposed CHM sampling***

Figure 3.4: Reconstructed data for a 256×256 knee MRI image with 4K samples (from (b) to (d)) and 10K samples (from (f) to (h)).



(a) ***Original image*** (b) ***FFT sampling*** (c) ***Gaussian sampling*** (d) ***Proposed CHM sampling***

(e) ***FFT sampling*** (f) ***Original image*** (g) ***Gaussian sampling*** (h) ***Proposed CHM sampling***

Figure 3.5: Reconstructed data for a 256×256 spine MRI image with 4K samples (from (b) to (d)) and 10K samples (from (f) to (h)).

Table 3.2: PSNR performance comparison for 256 × 256 test images with 4K measurements.

Image	PSNR with	PSNR with	PSNR with
	random Fourier	random Gaussian	proposed
	matrix (dB)	matrix (dB)	CHM (dB)
“angio256”	16.96	11.23	31.95
“mri256”	23.15	12.15	28.43
“knee256”	20.74	10.81	30.29
“spine256”	15.61	20.87	26.64
“brain256”	8.89	9.75	26.68

Table 3.3: PSNR performance comparison for 256 × 256 test images with 10K measurements.

Image	PSNR with	PSNR with	PSNR with
	random Fourier	random Gaussian	proposed
	matrix (dB)	matrix (dB)	CHM (dB)
“angio256”	27.44	36.72	41.09
“mri256”	33.13	34.83	40.86
“knee256”	37.42	39.11	39.41
“spine256”	23.71	36.34	41.51
“brain256”	33.7	37.22	40.40

which is approximately 15% of the original image. The quality of image is also high for very small number of samples. As expected, the more samples taken, the higher the PSNR, but still the proposed matrix provides a higher quality in comparison with the other two measurement matrices. As in case of any image processing, reconstruction from more number of samples provides a higher PSNR, which can also be observed from the tabulated results. However, it is to be noted that, even by using just about 10% of the original image data, the proposed sampling method can yield good quality reconstruction.

3.4.2 Simulations for 3D-MRI

For the purpose simulations, the 3D-MRI images used are of size $256 \times 256 \times 160$. In addition, we also generate a 3D Shepp-Logan phantom using MATLAB as outlined in [2] for a fair comparison of our method with the Fourier based NFCS-3DFista [2].

We first compare the $PSNR_m$ for our method with the Fourier based NFCS-

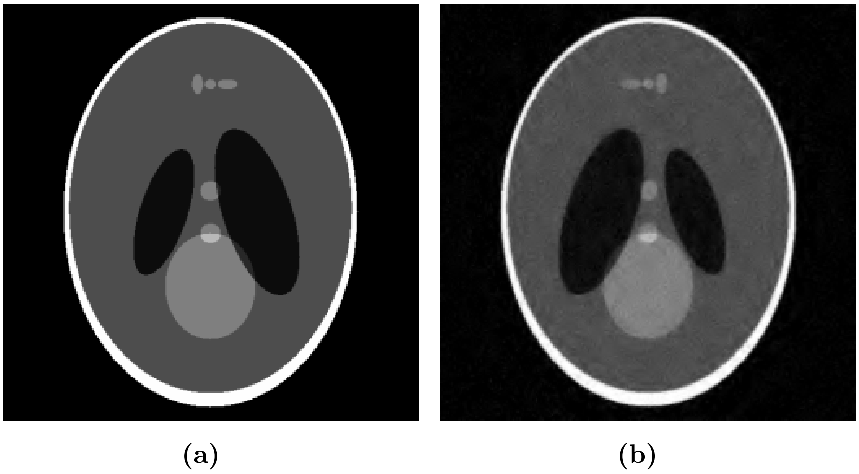


Figure 3.6: 3D Shepp-Logan phantom of size 256 × 256 × 11. Fig.1(a) is a single 2D original slice and Fig.1(b) shows the reconstruction from our proposed method.

3DFista, for a 3D phantom. $PSNR_m$ is where we take the mean of all the PSNR values obtained, which is given by [2]
$PSNR_m = \frac{1}{N}\sum_{i=1}^{N} PSNR_i$ where
$PSNR_i = 20 \log \frac{R}{RMSE}$ The above notations and equations to calculate PSNR are same as that used for NFCS-3DFista. Again, this is done so that all parameters are comparable.

A single slice of the generated 3D phantom and the reconstructed 2D image our proposed method is shown in Fig. 3.6. The results are tabulated in Table 3.4. The data in the frequency domain are acquired using random cartesian subsampling patterns. As depicted in the Table 3.4, for this pattern we obtain the $PSNR_m$ values at least 1 dB higher than that of [2] and have iterations much fewer than the NFCS-3DFista. Though the $PSNR_m$ is not very high, the iterations required makes our method more efficient. Furthermore, this also shows that our method would require less computation time when compared with NFCS-3DFista. All the values for NFCS-3DFista have been taken from the data presented in [2].

Next, we present the efficiency of our method with respect to conventional Fourier sampling and most popularly used OMP reconstruction. Fig. 3.7 shows the PSNR plotted with respect to the sampling rate (i.e., k/N). The data used for the graphs is real data supplied via the international consortium of brain mapping (ICBM) [1]. The performances of CHM and Fourier-based data acquisition with OMP reconstruction provide very similar results, and are much lower to the reconstruction using CoSaMP. There is at least 3 to 4 dB difference observed throughout various sampling rates when compared with Fourier-based CoSaMP reconstruction. Moreover, as determined previously in Fig. 3.1, CoSaMP performance is proven to be superior to the Fourier and Gaussian sampling. This can be easily observed in Fig. 3.7. From the same graph, we can also conclude that the CoSaMP algorithm, when used with Fourier transform can provide better re-

Table 3.4: Comparison of number of iterations and corresponding $PSNR_m$ for proposed method and NFCS-3D [2]. The acceleration factors vary from 4 to 16 times.

Accl. factor	Proposed		NFCS-3D [2]	
	$PSNR_m(dB)$	No. of iterations	$PSNR_m(dB)$	No. of iterations
x4	20.73	18	18.85	39
x8	17.93	56	15.63	73
x16	16.12	103	14.68	137

sults than used with OMP reconstruction. This is due to the provision of bounds in CoSaMP, which is not present in OMP. Furthermore, when CHM acquisition is combined with CoSaMP, the PSNR results are further improved. Hence, our proposed CHM-based CoSaMP is more suitable for CS-based 3D-MRI.

Furthermore, we perform multiple data acquisition with CHM with 10K, 20K and 30K samples. For better analysis and comparison of the proposed CoSaMP scheme with the OMP scheme, the data is reconstructed using both algorithms. We make a further combination of CHM with two different Φ matrices, namely, Daubechies-4 wavelet and the identity matrix commonly used in CS as the sparse matrix. The advantage of having wavelet coefficients is in terms of energy compaction in fewer coefficients, but does have a higher mutual coherence between the Φ and Ψ matrices.

By performing these simulations, we prove our choice of sparse matrix to be suitable when used in combination with CHM. The results are tabulated in Tables 3.5 and 3.6, for two different datasets. From the results, CoSaMP proves to be better than OMP, when used with CHM. There is a difference of about 2 to 3 dB in all cases. Considering that OMP has been extensively used in many CS-based MRI reconstruction, it is notable that the proposed reconstruction provides a better quality, which is of utmost importance in MRI.

Figs. 3.8 and 3.9 illustrate the performance of our proposed method with the conventional fast Fourier transform (FFT) based reconstruction without CS and the CHM-based OMP reconstruction, using identity matrix and Daubechies-4 wavelet respectively. To obtain this output, the image is reconstructed utilizing only 30K measurements, which is less than half of the fully sampled image. From the figures, it is evident that the difference of the reconstructed image from the proposed algorithm is better than that of OMP. Even with the use of different sparse matrices, the results for CoSaMP is superior to that of the OMP algorithm. From all the above illustrations we can conclude that, the choice of CoSaMP reconstruction algorithm is appropriate for CS-based 3D-MRI. And, when used with CHM, the output is in-par with conventional 3D-MRI using the FFT. A 3D-MRI reconstructed data from the proposed method for the two data sets are

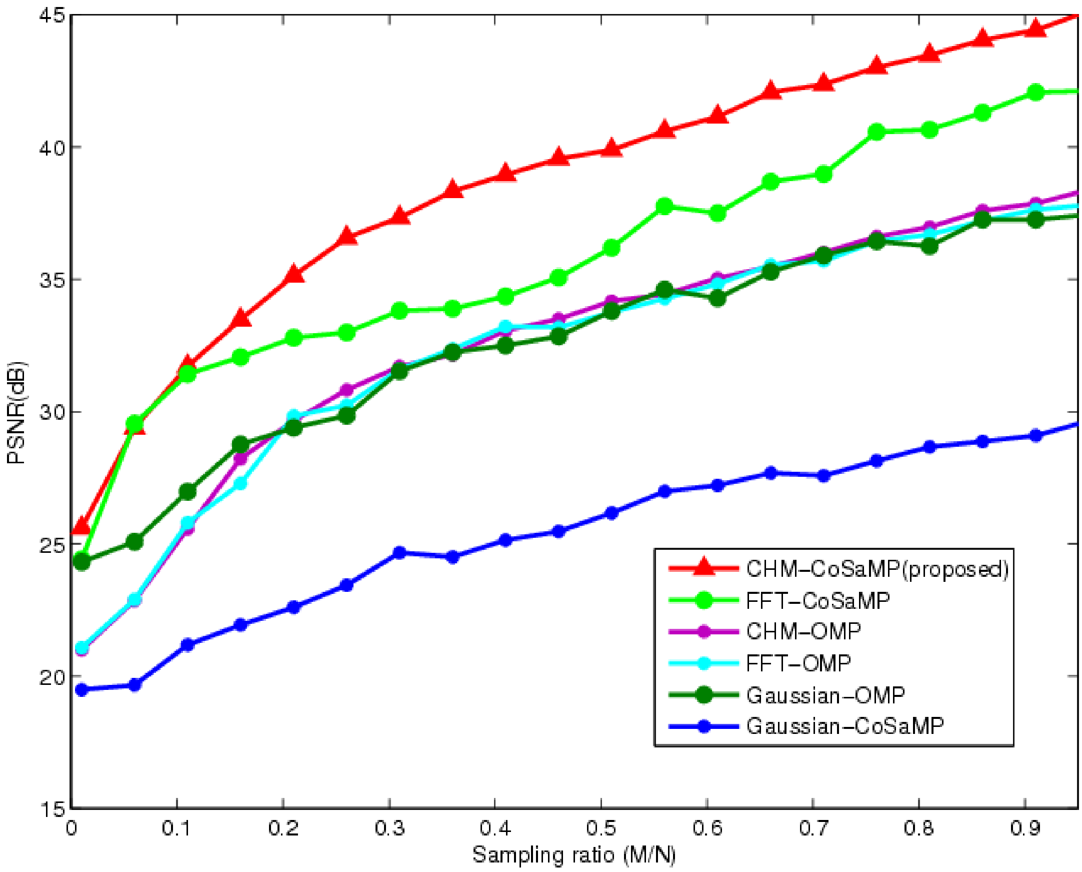


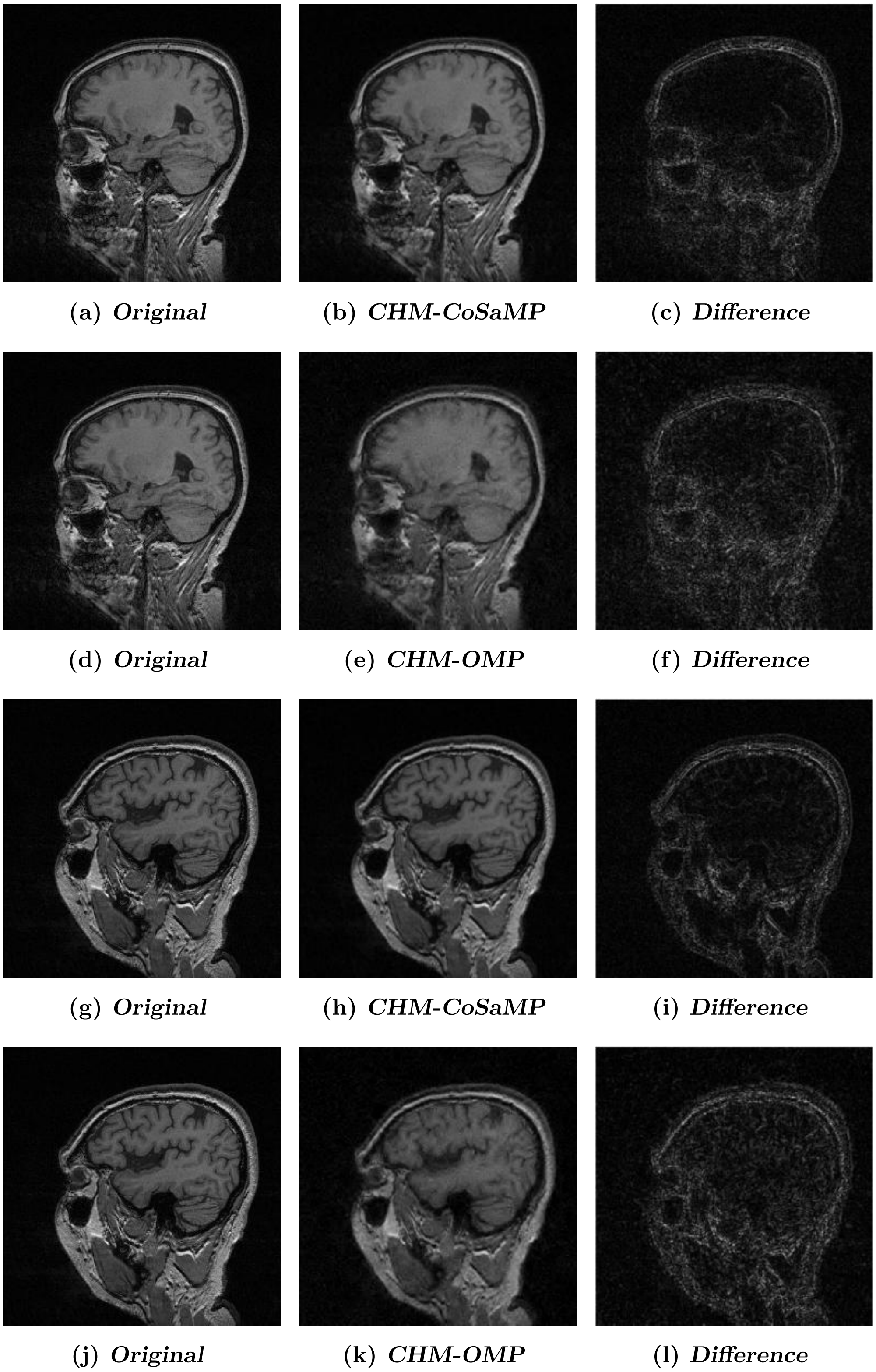
Figure 3.7: PSNR versus the sampling rate comparing the CHM with the Fourier and Gaussian sampling. The reconstruction is performed with proposed CoSaMP and with OMP each time.

also shown in Figs. 3.10 and 3.11.

Table 3.5: PSNR performance for dataset-1.

	PSNR (dB) with 10K measurements	PSNR (dB) With 20K measurements	PSNR (dB) With 30K measurements
CHM-CoSaMP-Identity	26.84	32.18	43.29
CHM-OMP-Identity	23.15	32.60	38.43
CHM-CoSaMP-Wavelet	27.32	33.04	44.32
CHM-OMP-Wavelet	24.61	32.87	38.10

Overall, from the simulation results the main idea of using a new non-Fourier basis for 3D-MRI is demonstrated. The results observed in every stage points to the fact that, matrices other than the conventional Fourier transform can possibly be used and also provide good results. This can also provide faster 3D-MRI scans and obtain accuracies close to the conventional non-CS method of MRI scans.



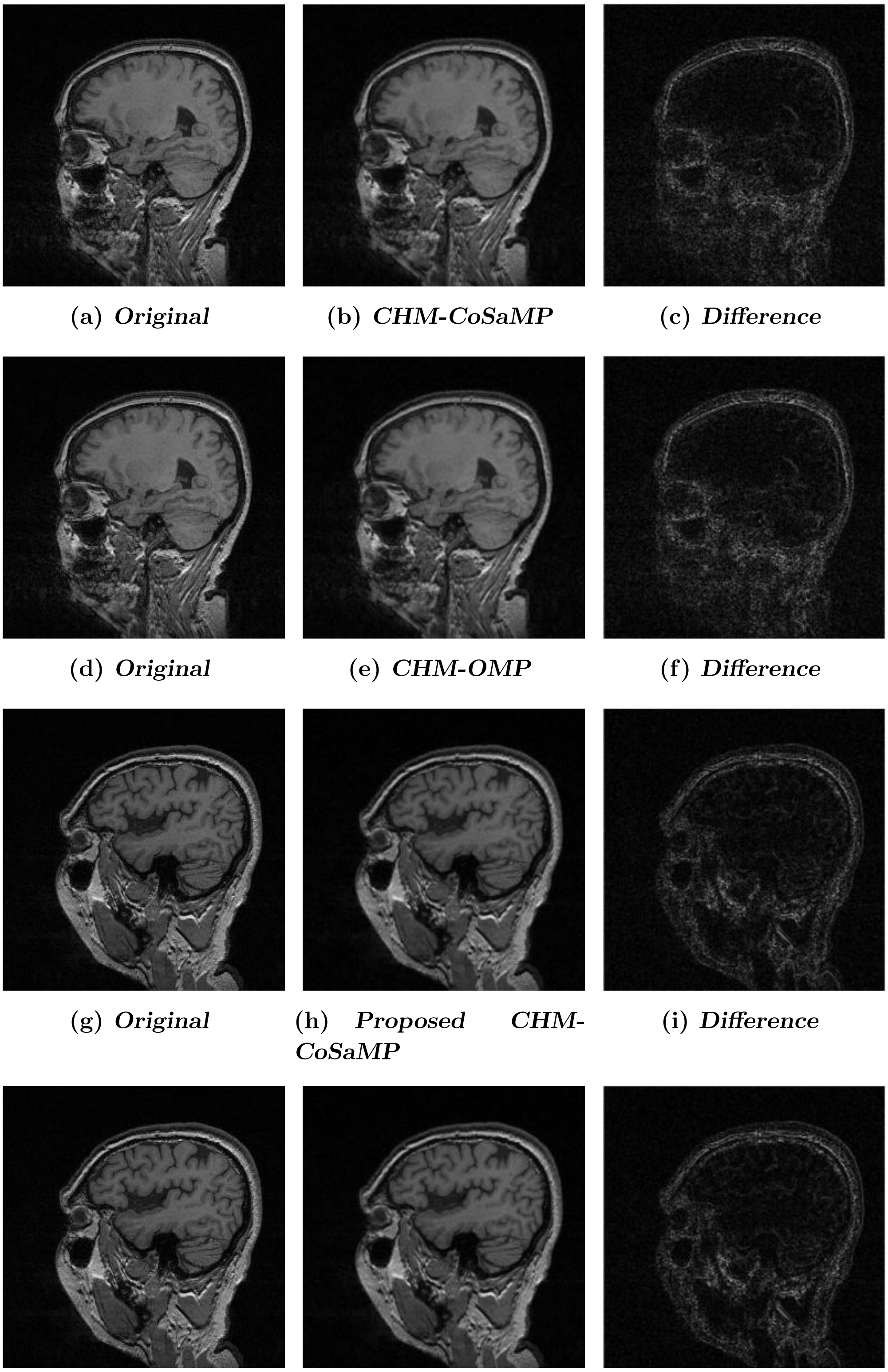
(a) ***Original*** (b) ***CHM-CoSaMP*** (c) ***Difference***

(d) ***Original*** (e) ***CHM-OMP*** (f) ***Difference***

(g) ***Original*** (h) ***CHM-CoSaMP*** (i) ***Difference***

(j) ***Original*** (k) ***CHM-OMP*** (l) ***Difference***

Figure 3.8: Fully sampled and reconstructed images for a 256×256 single 2D slice of a 3D dataset. The Ψ matrix used is the identity matrix and CHM is the Φ matrix. The reconstruction is performed with CoSaMP as in Fig(b) and with OMP as in Fig (c).



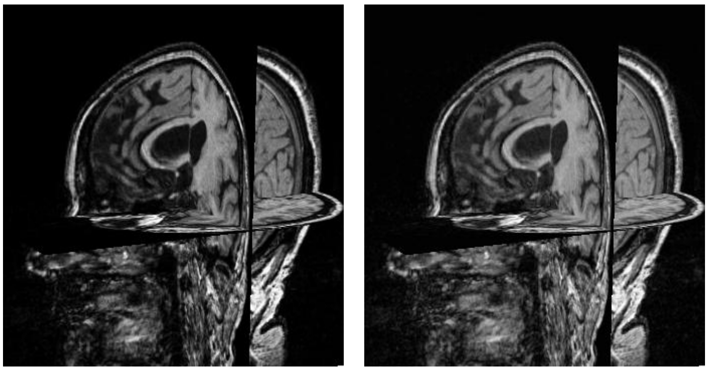
(a) ***Original*** (b) ***CHM-CoSaMP*** (c) ***Difference***

(d) ***Original*** (e) ***CHM-OMP*** (f) ***Difference***

(g) ***Original*** (h) ***Proposed CHM-CoSaMP*** (i) ***Difference***

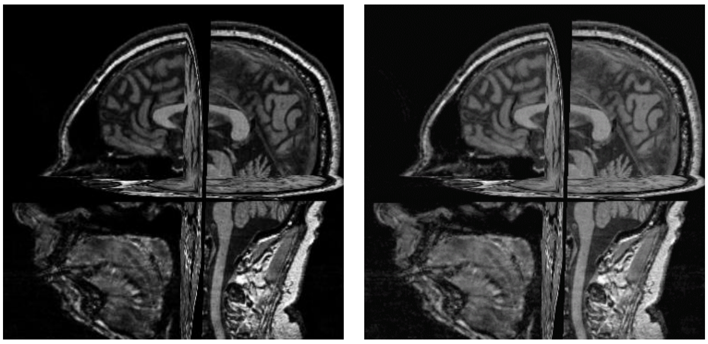
(j) ***Original*** (k) ***CHM-OMP*** (l) ***Difference***

Figure 3.9: Fully sampled and reconstructed images for a 256×256 single 2D slice of a 3D dataset. The Ψ matrix used is the Daubechies-4 wavelet and CHM is the Φ matrix. The reconstruction is performed with CoSaMP as in Fig(b) and with OMP as in Fig (c).



(a) ***Original 3D model*** (b) ***Proposed***

Figure 3.10: Reconstructed 3D-MRI model from fully sampled and Proposed methods.



(a) ***Original 3D model*** (b) ***Proposed***

Figure 3.11: Reconstructed 3D-MRI model from fully sampled and Proposed methods.

Table 3.6: PSNR performance for dataset-2.

	PSNR (dB) With 10K measurements	PSNR (dB) With 20K measurements	PSNR (dB) With 30K measurements
CHM-CoSaMP-Identity	27.98	34.07	44.35
CHM-OMP-Identity	26.15	32.68	43.72
CHM-CoSaMP-Wavelet	27.61	34.94	44.38
CHM-OMP-Wavelet	26.09	32.24	42.42

3.5 Summary

In this paper, a CS-based 3D-MRI implementable encoding scheme superior to conventional Fourier encoding is demonstrated. An efficient approach of compressive sampling for MRI using complex Hadamard measurements and CoSaMP reconstruction for MRI is proposed. A new measurement matrix called the complex Hadamard matrix is proposed and shown to satisfy the restricted isometry property using the Daubechies-4 wavelet transform. This is a sufficient condition for use in CS. CoSaMP in combination with complex Hadamard is used for the first time for the purpose of 3D-MRI reconstruction and shown to be suitable for the same. The results are compared with the state-of-the-art Fourier basis, and it is observed that the PSNR of the proposed method is better than the existing CS reconstruction methods.

Moreover, the fact that 3D-MRI can be very well represented in the complex Hadamard basis using relatively fewer coefficients is presented. To justify the use of this combined system, simulations are performed on real clinical data of healthy subjects, and compared with the conventional sampled data. The reconstruction image quality is indicated by the PSNR, which proves that our method is comparable with conventional 3D-MRI. We also observe that the data acquisition and reconstruction using our method is faster than conventional method, with comparable image quality.

Chapter 4

Optimization of Complex Hadamard Matrix for Enhanced 2D/3D-MRI Performance

In the previous chapter, a combination of the simple complex Hadamard matrix (CHM) with CoSaMP was proposed with minimal update of the bounds for reconstruction. For an efficient practical setting, it is necessary that an optimized structured matrix be defined, which strongly satisfies the CS conditions. One of the main properties is the incoherence property. The fact that small mutual coherence between the measurement matrix and the sparsifying matrix is a requirement for achieving successful CS reconstruction. Therefore, designing measurement matrices with smaller coherence is desired.

It is well-known that, any random matrix satisfies the RIP, where the entries are generated by a probability distribution such as the Gaussian or Bernoulli process, or from randomly chosen partial Fourier ensembles. This has been widely studied and applied in most practical cases. But, the use of structured CS matrices implies that existing RIP results pertaining to such matrices are not applicable in their case. In the past, researchers have often resorted to numerical simulations to prove the efficacy of structured CS matrices arising in various practical settings [118] [119]. Since this thesis deals with the complex Hadamard matrix, which is a structured matrix, proving its efficiency is a challenging task. Furthermore, its usability in practical situations is explored.

Hence, in this chapter we deal with this challenge and provide the following contributions

1. Generate a new CHM matrix based on unitary matrix principles for a improved MRI performance;

2. Show its suitability for CS-based applications by proving the RIP and incoherence property;

3. Study the practical implications with respect to the CHM and also with some of the existing structured matrices.

4.1 Related Work

The first family of sensing matrices for $l1$-based reconstruction algorithms consisted of random Gaussian/Bernoulli matrices, more generally, sub-Gaussian random matrices [120]. Their main advantage is that they are universally incoherent with any sparse signal and thus, the number of compressed measurements required for exact reconstruction is almost minimal. However, they inherently have two major drawbacks for practical applications, namely, huge memory buffering for storage of matrix elements and high computational complexity due to their completely unstructured nature [57]. The second family is partial Fourier [57], and more generally, randomizing rows of any orthonormal matrix. One of the most commonly used matrix is a partial Fourier matrix that exploits the fast computational property of the FFT and thus reducing the complexity of a sampling system. However, a partial Fourier matrix is only incoherent with signals which are sparse in the time domain, severely narrowing its scope of applications. Recently, random filtering was proposed empirically in [119] as a potential sampling method for fast low-cost compressed sensing applications. Unfortunately, this method currently lacks a theoretical foundation for quantifying and analyzing its performance.

4.2 Matrix Formulation

In the previous chapter, the complex Hadamard matrix was used for MRI data acquisition. The sensing matrix used was a combination of the CHM and Daubechies-4 wavelet transform. The aim is to formulate a modified CHM so that more efficient MRI processing can be achieved. This also includes exact CS reconstruction. The CoSaMP algorithm used, is the same as outlined in Section 4.3.1.

Towards this end, a unitary CHM (UCHM) with structurally permuting the CHM matrix is generated. For rest of the discussions, this matrix will be termed as UCHM for simplicity.

Definition 4. *The complex Hadamard matrix H_N of order $N = 2^n$ is unitary if it is a square matrix with elements $\{\pm 1, \pm j\}$, and H_N is orthogonal in the complex* *domain, the matrix is generated based on the property,*

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} H_N(p,k)H_N^*(q,k) = \begin{cases} N, & forp = q \\ 0, & forp \neq q \end{cases} \tag{4.1}$$

where H_N^* denotes the conjugate transpose of matrix H_N, and p,q and k are the row and column indices respectively.

One of the reasons that has led to the widespread applicability of CS theory in various application areas is the revelation that certain probabilistic constructions of matrices satisfy the RIP (Chapter 2: Definition 3) with high probability.

4.3 Restricted Isometry Property

In this section, the RIP can be established for the generated UCHM using Rademacher sequence.

Theorem 8. *Let the elements of the generating sequence $\Theta_p = \{a_i\}_{i=1}^p$ be independent and identically distributed realizations of Rademacher random variables taking values ± 1 with probability $1/2$. Choose a subset Ω of cardinality $n \equiv= |\Omega|$ uniformly at random from the set $[1 \dots m]$. Finally, let U be any $p \times p$ unitary matrix, and Θ be the $n \times p$ matrix obtained by sampling n rows of X corresponding to the indices in Ω and renormalizing the resulting columns by $\sqrt{m/n}$. Then for each integer $p, S > 2$, and for any $z > 1$ and $\delta_S \in (0,1)$, there exist absolute constant C such that whenever*

$$n \geq Cz\mu_U^2 S \log^3 p \log^2 S \tag{4.2}$$

the matrix $\Theta \in RIP(K, \delta_S)$ with probability exceeding $1.20 \max\{\exp(-C\delta_S^2 z), p^{-1}\}$.

Proof. We begin by recalling the result established in [59], which states that if matrices in a particular class satisfy RIP with probability exceeding $1 - \eta$ for the Bernoulli sampling model, then it follows that matrices belonging to the same class satisfy the RIP with probability exceeding $1 - 2\eta$ for the uniformly permuted sampling model.

Next, consider the Banach space $\mathcal{B} \equiv (\mathbb{C}^p \times p, \|\cdot\|_{T,S})$ and define variables $\{Y_i\}_{i=1}^p$ and $\{\tilde{Y}_i\}_{i=1}^p$ that take values in $\mathcal{B}$ as follows

$$Y_i \equiv \frac{m}{n} e_i x_i x_i^H - \frac{1}{p} I_p, \tag{4.3}$$

$$\tilde{Y}_i \equiv \frac{m}{n}(e_i x_i x_i^H - e_i' x_i' x_i'^H), i = 1 \dots p$$

where, $\{e_i\}$ are the Bernoulli random variables arising in the Bernoulli sampling model, $\{x_i^H\}$ denote the rows of X, and $\{e_i'\}$, $\{x_i'\}$ are independent copies of $\{e_i\}$ and $\{x_i\}$ respectively. In other words, each random variable $\tilde{Y}_i \equiv Y_i - Y_i'$ is a symmetric version of the corresponding variables Y_i where Y_i' denotes an independent copy of Y_i. In particular, we have that $\sum_{i=1}^p \tilde{Y}_i$ in $\mathcal{B}$ is a symmetric version of $\sum_{i=1}^p Y_i$ and, as a consequence, the following symmetric inequalities hold for all $u > 0$ [121].

$$\mathbb{E}\left[\|\sum_{i=1}^{p}\tilde{Y}_i\|_{T,S}\right] \leq 2\mathbb{E}\left[\|\sum_{i=1}^{p}Y_i - \mathbb{E}[\sum_{i=1}^{p}Y_i]\|_{T,S}\right], \tag{4.4}$$

$$\begin{aligned}\Pr\left(\mathbb{E}\left[\|\sum_{i=1}^{p}Y_i\|_{T,S}\right] > 2\mathbb{E}\left[\sum_{i=1}^{p}Y_i\|_{T,S}\right] + u\right) \\ \leq 2\Pr\left[\|\sum_{i=1}^{p}\tilde{Y}_i\|_{T,S} > u\right].\end{aligned} \tag{4.5}$$

Specifically, for any integer $p > 2$ and $r = 2\log p$, we have Bernoulli sampling model

$$\begin{aligned}(\mathbb{E}[\|\Theta\|_{\max}^r])^{1/r} &\leq \sqrt{\frac{m}{n}}\,(\mathbb{E}[\|X\|_{\max}^r])^{1/r} \\ &\leq \sqrt{\frac{16\mu_U^2 \log p}{n}}.\end{aligned} \tag{4.6}$$

Substituting (4.7) in (4.6), we obtain

$$\Pr\left(\sqrt{\frac{m}{n}}\|X\|_{\max} > \sqrt{\frac{16e\mu_U^2\log p}{n}}\right) \leq \tag{4.7}$$

$$\textit{nonumber}\, \Pr\left(\|X\|_{\max} > \sqrt{e}(\mathbb{E}[\|X\|_{\max}^r])^{1/r}\right) \tag{4.8}$$

$$\begin{aligned}&\Pr\left(\|X\|_{\max}^r > e^{r/2}\mathbb{E}[\|X\|_{\max}^r]\right) \\ &\leq \frac{\mathbb{E}[\|X\|_{\max}^r]}{e^{r/2}\cdot\mathbb{E}[\|X\|_{\max}^r]} = p^{-1}\end{aligned}$$

obtained from a simple application of Markov's inequality. Next, define $B_1 \equiv \frac{16e\mu_U^2\log p}{n}$. Then from (4.9) we obtain

$$\Pr\left(\left\{\sqrt{\frac{m}{n}}\|X\|_{\max} > \sqrt{B_1}\right\}\bigcup\left\{\sqrt{\frac{m}{n}}\|X'\|_{\max} > \sqrt{B_1}\right\}\right) \leq 2p^{-1} \tag{4.9}$$

where X' is comprised of $\{x_i'^H\}$ as its rows, and a union bounding argument.

Further, we also have

$$\begin{aligned} \max_i \|\tilde{Y}_i\|_{T,S} &= \max_i \|\frac{m}{n}(e_i x_i x_i^H - e_i' x_i' x_i'^H)\|_{T,S} \\ &\leq \max_i \left\{\|\frac{m}{n}\|x_i x_i^H\|_{T,S} + \|\frac{m}{n} x_i' x_i'^H\|_{T,S}\right\} \\ &\leq \max_i \left\{S(\sqrt{\frac{m}{n}}\|x_i^H\|_\infty)^2 + S(\sqrt{\frac{m}{n}}\|x_i'^H + \|_\infty)^2\right\} \\ &\leq S(\frac{m}{n}\|X\|_{\max}^2 + \frac{m}{n}\|X'\|_{\max}^2) \end{aligned} \tag{4.10}$$

obtained from triangle inequality and from, $\|X\|_{\max} \equiv max_i \|x_i^H\|_\infty$ and $\|X'\|_{\max} \equiv \max_i \|x_i'^H\|_\infty$. It is then easy from (4.9) and (4.11) that we have $\max_i \|\tilde{Y}_i\|_{T,S} \leq 2SB_1$ with probability exceeding $1 - 2p^{-1}$.

Finally, define $E \equiv \{\max_i \|\tilde{Y}_i\|_{T,S} \leq 2Sb_1\}$. Based on this, whenever $n \geq C\varepsilon^{-2}\mu_U^2 S \log^3 p \log^2 S$ we have

$$\Pr(\tilde{Y} \geq 16q\varepsilon + 4rSB_1 + tE) < \left(\frac{C}{q}\right) + 2\exp\left(-\frac{t^2}{1024q\varepsilon^2}\right) \tag{4.11}$$

for any integer $r \geq q$, $t > 0$, and $\varepsilon \in (0,1)$. Next, choose $q = \lceil eC \rceil, t = 32\sqrt{q}\eta\varepsilon$, and $r = \lceil \frac{t}{2SB_1} \rceil$ for some $\eta > 1$. Further, define a new constant $C_1 \overset{\text{def}}{=} \max\{e\sqrt{q}, C\}$ and let $n \geq C_1\varepsilon^{-2}\mu_U^2 S \log^3 p \log^2 S$. Note that this choice of n ensures $r \geq q$, resulting in

$$\Pr(\tilde{Y} \geq (16q + 96\sqrt{q})\eta\varepsilon E) < \exp\left(-\frac{\sqrt{q}\eta\varepsilon n}{3\mu_U^2 S \log p}\right) + 2\exp(-\eta^2). \tag{4.12}$$

Noting that $\Pr(E^c \leq 2p^{-1}$ implies,

$$\Pr(\tilde{Y} \geq (16q + 96\sqrt{q}\eta)\varepsilon) < \exp\left(-\frac{\sqrt{q}\eta\varepsilon n}{3\mu_U^2 S \log p}\right) + 2\exp(-\eta^2) + 2p^{-1}. \tag{4.13}$$

Finally, what remains to be shown is that $Y = \|\sum_{i=1}^p Y_i\|_{T,S} = \|\Theta^H\Theta - I_p\|_{T,S} \leq \delta_S$ with high probability. Note that, if $n \geq C_1\varepsilon^{-2}\mu_U^2 S \log^3 p \log^2 S$ then $\mathbb{E}[Y] \leq \varepsilon$, we get from (4.6)

$$\Pr(Y \geq (2+16q+96\sqrt{q}\eta)\varepsilon) < 2\exp\left(-\frac{\sqrt{q}\eta\varepsilon n}{3\mu_U^2 S \log p}\right) + 4\exp(-\eta^2) + 4p^{-1}. \tag{4.14}$$

By defining $C' \equiv (2 + 16q + 96\sqrt{q})$ and $C'\eta\varepsilon > (2 + 16q + 96\sqrt{q}\eta)\varepsilon$ since $\eta > 1$. If we choose $\eta = \frac{\delta_S}{C'\varepsilon}$ then $\frac{\sqrt{q}\eta\varepsilon n}{3\mu_U^2 S \log p} > \eta^2$. Therefore, (4.14) can be simplified as

$$\Pr(Y \geq \delta_S) < 10\max\{\exp(-\frac{1}{C'\varepsilon^2}\delta_S^2), p^{-1}\} \tag{4.15}$$

This ends the proof that the UCHM satisfies the RIP. ■

4.3.1 Complex Hadamard based CoSaMP

In this section, the bounds are derived for the UCHM-based CoSaMP. Among many CS reconstruction algorithms, CoSaMP provides a stopping criterion so that the reconstruction procedure stops after a certain number of iterations. The running time bound indicates that, with each matrix multiplication the error reduces by a constant factor. Hence, the CoSaMP algorithm achieves linear convergence. The total running time is also proportional to the reconstruction signal-to-noise ratio. The selection of CoSaMP over OMP and other reconstruction methods, is due to its fast convergence, the algorithm complexity does not depend on the sparsity K [53] and its suitability for implementing on a hardware platform(e.g.,FPGA, GPU).

As the matrix changes, the reconstruction bounds change simultaneously. Consider an $M \times N$ sensing matrix Θ with the restricted isometry constant C and $y = \Theta x + e$ is a vector of samples of an arbitrary and e is noise, then CoSaMP produces a K-sparse approximation a that satisfies,

$$\|x - a\|_2 \leq C \max\{\eta, \frac{1}{\sqrt{K}}\|x - x_{s/2}\|_1 + \|e\|_2\}, \tag{4.16}$$

where η is the precision parameter and $x_{K/2}$ is the best $K/2$-sparse approximation to x.

Now, we derive the bounds for the CoSaMP algorithm. The reconstruction error is bounded by the product of a constant C and the noise power in the form $\|x - \hat{x}\|_2^2 \leq C.\|e\|_2^2$ [53]. We obtain the bound based on matrix Θ, that is a constant times $\|\Theta_*^{T_e} e\|$. Then, $\hat{x}^l$ is the result obtained at the lth iteration and T is the support.

Theorem 9. *For a K-sparse vector x, under the condition $\delta_{bk} \leq \delta$, solution of CoSaMP at the lth iteration satisfies*

$$\|x - \hat{x}^l\|_2 \leq 2^{-l}\|x\|_2 + (C-1)\|\Theta {*}^{T_e} e\|_2. \tag{4.17}$$

In addition, after

$$[log_2(\frac{\|x\|_2}{\|\Theta {*}^{T_e} e\|_2} \tag{4.18}$$

iterations, the algorithm leads to an accuracy bounded by

$$\|x - \hat{x}^l\|_2 \leq C\|\Theta_*^{T_e} e\|_2 \tag{4.19}$$

where $b = 4$, $\delta = 0.1$ and $C = \frac{29 - 14\delta_{4K} + \delta_{4K}^2}{(1-\delta_{4K})^2} \leq 34.1$

Algorithm 3 CoSaMP

```
  Require: $K$,$M$,$\Theta$,$y$,$a$ where $y = Mx + e$ and $x = \Theta\alpha$
  $K$ is the cardinality of $\alpha$ and $e$ is the additive noise, $a = 2$
  Result: $\hat{x}$ : $K$-sparse approximation of $x$
  Initialize the support $T^0 = \Theta$, the residual $y_r^0 = y$ and set t=0

  while stop criterion is not satisfied do
    $t = t + 1$
    Find new support elements: $T_\Delta = supp(\Theta^* M^* y_r^{t-1}, ak)$
    Update support: $\tilde{T}^t = T^{t-1} \bigcup T_\Delta$
    Compute a temporal estimate: $\alpha_p = (M\Theta_{\tilde{T}_t})\dagger y$
    Prune small entries: $T^t = supp(\alpha_K)$
    Calculate a new estimate: $\hat{x}^t = \Theta_{T_t}(\alpha_p)_{T_t}$
    Update the residual: $y_r^t = y - M\hat{x}^t$
  end while
From final solution $\hat{x} = \hat{x}^t$
```

Proof. Beginning from the following equation,

$$\|x - \hat{x}^l\|_2 \leq 0.5\|x - \hat{x}^{l-1}\|_2 + 16.6C\|\Theta_*^{T_e} e\|_2 \tag{4.20}$$

for $\delta_{4K} \leq 0.1$, and applying it recursively, we arrive at

$$\|x - \hat{x}^l\|_2 \leq 0.5^K\|x - \hat{x}^{l-K}\|_2 + 16.6\left(\sum_{j=0}^{K-1} 0.5^j\right) C\|\Theta_*^{T_e} e\|_2 \tag{4.21}$$

By setting $K = l$ it easily leads to (4.17), since $\|x - \hat{x}^0\|_2 = \|x\|_2$. Inserting the number of iterations l^* as in (4.18) to (4.17) yields

$$\begin{aligned} \|x - \hat{x}^l\|_2 &\leq 2^{-l}\|x\|_2 + 2 \cdot \frac{14 - 6\delta_{4K}}{(1 - \delta_{4K})^2}\|\Theta_*^{T_e} e\|_2 \\ &\leq \left(1 + 2 \cdot \frac{14 - 6\delta_{4K}}{(1 - \delta_{4K})^2}\right)\|\Theta_*^{T_e} e\|_2 \\ &\leq \frac{29 - 14\delta_{4K} + \delta_{4K}^2}{(1 - \delta_{4K})^2}\|\Theta_*^{T_e} e\|_2 \end{aligned} \tag{4.22}$$

Then, applying the condition $\delta_{4K} \leq 0.1$ to the above equation leads to the result. ■

The CoSaMP algorithm used for reconstruction is outlined in Algorithm 3. Unlike other reconstruction algorithms, CoSaMP requires that the sparsity level K be provided as part of its input. To reduce the running time, K can be varied along a geometric progression as $K = 1, 2, \ldots, M$. Furthermore, partially known supports can be incorporated in CoSaMP unlike the OMP which is one of the commonly used reconstruction algorithm in image processing.

4.4 Numerical Results

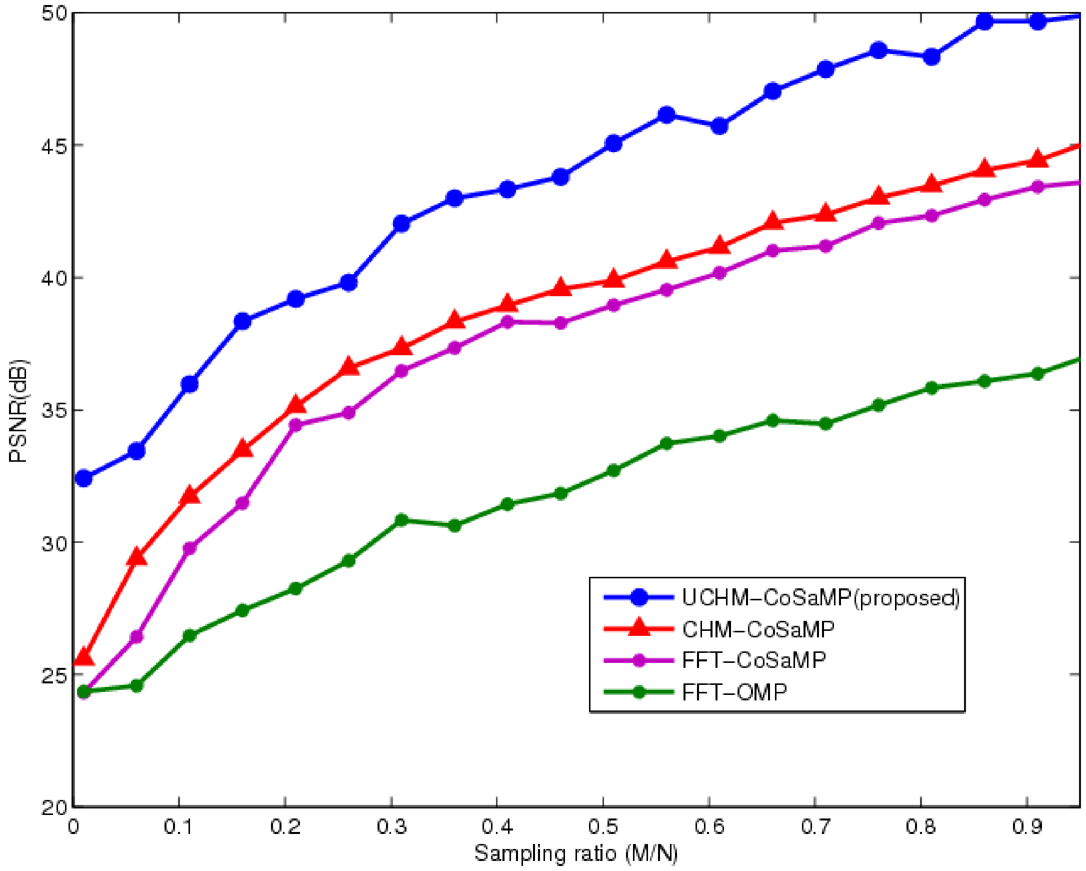
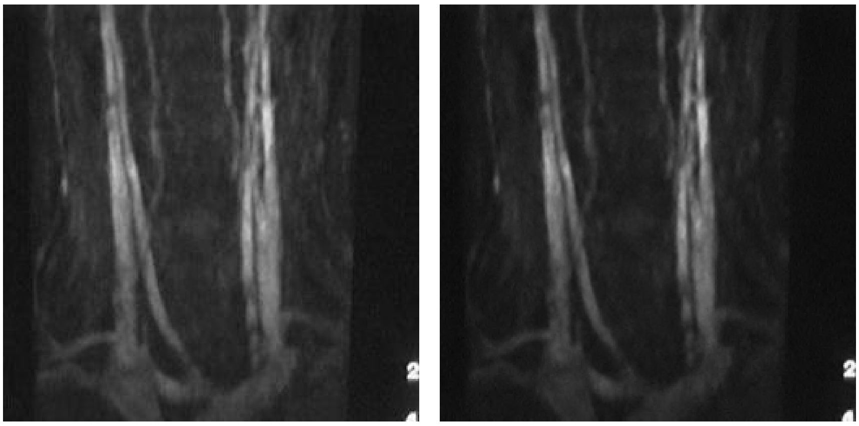


Figure 4.1: PSNR versus sampling rate graph comparing the proposed UCHM, CHM proposed in Chapter 3, FFT with CoSaMP reconstruction and FFT with OMP used in Chapter 3. Daubechies-4 wavelet is used as the matrix Φ

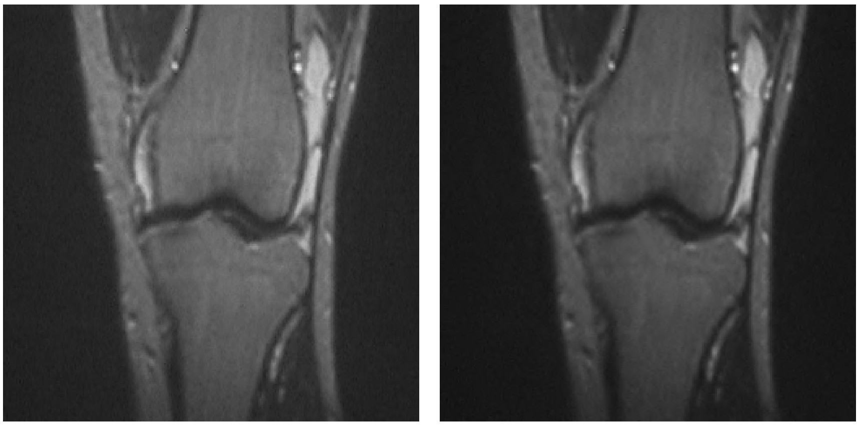
In this section, 2D-MRI and 3D-MRI processing with optimized complex Hadamard matrix UCHM and modified CoSaMP is compared with the CHM-based CoSaMP used in Chapter 3. The aim is to show the increase in peak signal-to-noise ratio and hence, the enhanced performance of UCHM-based 2D/3D-MRI.

Similar to the Chapter 3, the 3D-MRI images used are of size 256 × 256 × 160, supplied by the international consortium of brain mapping (ICBM) [1]. The simulation procedure followed is also identical. Fig. 4.1 depicts PSNR of various systems. First is the UCHM-CoSaMP, which is the modified version of CHM-CoSaMP system; second, the CHM-CoSaMP system; and the third and fourth are the FFT-based CoSaMP and OMP systems. Thought the aim is to compare the UCHM with CHM, we also consider the FFT-based system since the performance of the FFT-CoSaMP is closer to the CHM-CoSaMP system. For all cases, the sparse matrix Φ is the Daubechies-4 wavelet. From the graph, it can be clearly seen that the UCHM-CoSaMP system performance is superior to that of CHM-CoSaMP that was proposed in Chapter 3. The PSNR improvement is



(a) ***Original Image*** (b) ***Reconstructed Image***

Figure 4.2: Fully sampled and reconstructed images with 10K measurements for a 256×256 ‘angio’ 2D-MRI image.



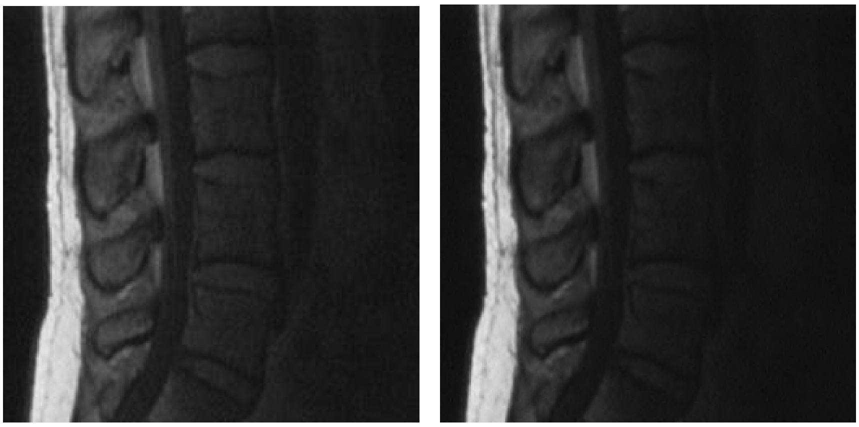
(a) ***Original Image*** (b) ***Reconstructed Image***

Figure 4.3: Fully sampled and reconstructed images with 10K measurements for a 256×256 ‘knee‘ 2D-MRI image.

approximately 5 dB throughout. For this we can conclude that, by modifying the CHM to imbibe unitary properties, the performance of the CS system has increased.

4.4.1 Simulation Results for 2D MRI

In order to have a fair comparison the 2D-MRI images used in Chapter 3, which are three test images of 256 × 256 size. The experiments are conducted for 10K measurements. Figs. 4.2, 4.3 and 4.4 depict three examples of 2D-MRI. The PSNR of these images with the proposed UCHM-CoSaMP system and the CHM-CoSaMP system for 10K measurements are shown in Table 4.1. It can be clearly seen that, the performance with the UCHM is atleast 11dB higher than CHM.



(a) ***Original Image*** (b) ***Reconstructed Image***

Figure 4.4: Fully sampled and reconstructed images with 10K measurements for a 256×256 ‘spine‘ 2D-MRI image.

Table 4.1: PSNR performance 2D-MRI images with 10K measurements.

Image	UCHM-based PSNR	CHM-based PSNR
‘angio’	52.39	41.09
‘knee’	50.01	40.86
‘spine’	48.93	39.41

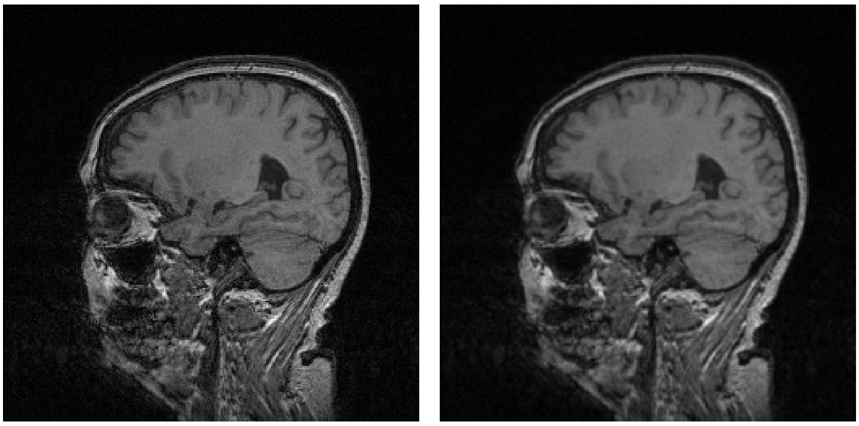
The values for the CHM-CoSaMP are taken from the Table 3.3 of Chapter 3.

4.4.2 Simulation Results for 3D MRI

Figs. 4.5 and 4.6 shows the visual comparison of the reconstructed data with the original data. The datasets used are again the same that are used perviously. The PSNR values of the proposed UCHM-CoSaMP systems and the CHM-CoSaMP system are tabulated in Tables 4.2 and 4.3. In both datasets, a PSNR difference of 4 to 5 dB is observed. This difference remains for almost all the datasets that are simulated for these systems.

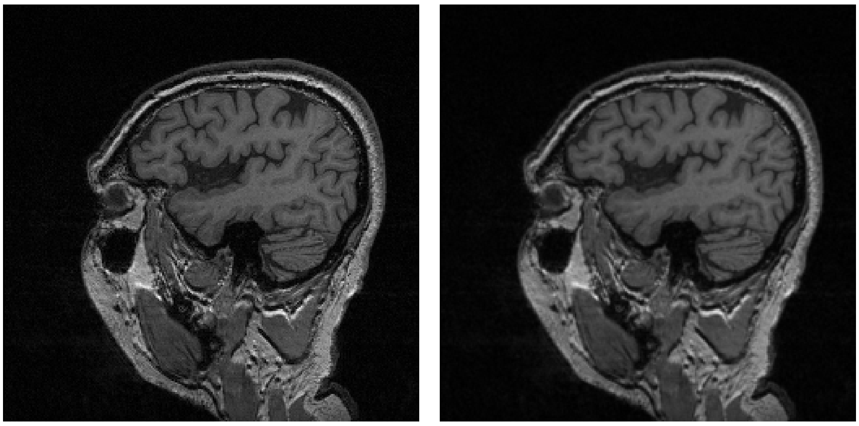
Table 4.2: PSNR performance 3D dataset-1.

Measurements	UCHM-based PSNR	CHM-based PSNR
6K measurements	28.61	-
10K measurements	31.88	27.32
20K measurements	46.03	33.04
30K measurements	48.97	44.32



(a) ***Original Image*** (b) ***Reconstructed Image***

Figure 4.5: Fully sampled and reconstructed images with 20K measurements for a 256×256 single 2D slice of 3D dataset-1.



(a) ***Original Image*** (b) ***Reconstructed Image***

Figure 4.6: Fully sampled and reconstructed images with 20K measurements for a 256×256 single 2D slice of 3D dataset-2.

Table 4.3: PSNR performance 3D dataset-2.

Measurements	UCHM-based PSNR	CHM-based PSNR
6K measurements	29.05	-
10K measurements	32.11	27.61
20K measurements	46.90	34.94
30K measurements	48.65	44.38

4.5 Summary

In this chapter, we introduced and analyzed a new version of complex Hadamard matrix called structured unitary complex Hadamard matrix. This matrix is a an optimized version of the CHM proposed in Chapter 3. It was shown with proof that this matrix satisfies RIP conditions, which is a sufficient condition to be used as a CS matrix. Furthermore, this method is simulated for 3D-MRI and results are observed to superior than the ones discussed in Chapter 3.

Chapter 5

Computation Efficient FPGA-Based Hardware Architecture for MRI Processing

5.1 Introduction

High-performance sparse signal recovery algorithms typically require a significant computational resources for the problem sizes occurring in most practical applications. While the computational complexity is not a major concern for applications where offline processing on central processing units (CPU) or graphics processing units (GPU) can be afforded (e.g., in MRI), it becomes extremely challenging when real-time processing with high throughput is required. Hence, to meet the stringent throughput, latency, and power-consumption constraints of real-time applications, developing dedicated hardware implementations, such as application specific integrated circuits (ASIC) or field-programmable gate arrays (FPGA), is of paramount importance.

Several studies showed that the performance of medical image processing algorithms, such as image registration and 3-D segmentation, can achieve significant improvements by implementing them on FPGA [10, 122] and GPU [123, 124]. However, GPU may not be suitable for applications that require irregular memory accesses [125]. On the other hand, FPGA may not be suitable for applications which have large and complex computational kernels that require double-precision floating point calculations due to limitations in silicon area. As a result, developers have to decide which architecture is suitable for their application such that they can achieve the most performance enhancement. FPGA provide several advantages for MR image processing. MR images contain large amounts of data and the algorithm requires frequent access to these data stored in memory. Another significant potential advantage of FPGA over GPU and CPU is low power consumption. GPU and CPU have a lower degree of parallelism in their architectures than FPGA and to achieve similar speeds, must have clock frequencies many times higher than FPGA (in the case of CPU, about 30 times higher). These high clock frequencies increase power consumption. As a result, FPGA can be considered a power-efficient alternative to other accelerators and typically do not require expensive cooling methods.

In order to overcome the above mentioned drawbacks, the following contributions are presented:

- Hardware based pipeline structure for the complex Hadamard matrix proposed in Chapter 3;
- Efficient memory organization for fast data access and processing;
- A fast hardware architecture for CS-based MRI encoding and reconstruction;

5.2 Related Work

While significant research efforts have been devoted to the design of high-performance and low-complexity sparse signal recovery algorithms, e.g., [24, 52, 53, 117], much less is known about their economical implementation in dedicated hardware. CS applied to most applications are computationally intensive due to iterative algorithms and require high-performance techniques to achieve near real-time solutions, but end up consuming enormous hardware resources. Power consumption and hardware size becomes a huge bottleneck, if CS needs to be used in practical applications. Hence, it is necessary to design hardware architectures that provides low power consumption, high throughput and near real-time solutions. Some of the ASIC implementations are reported in [126], where the authors compared several implementations of greedy pursuit algorithms for sparse channel estimation in wireless communication systems. A similar recovery algorithm specifically designed for signals acquired by the modulated wideband converter is implemented on FPGA in [127]. Another FPGA implementation for generic CS problems of dimension 32×128 is developed in [128]. All these implementations rely on algorithms that are well-suited for the recovery of highly sparse signals in hardware.

Traditionally, algorithms which directly calculate the image in a single backward reconstruction step, can be accelerated with GPU or FPGA [13–15, 129, 130]. However, when the number of samples is reduced, these methods generally generate very poor quality images. Thus, there is a strong motivation to accelerate iterative reconstruction methods for practical MRI systems. However, while there has been a substantial amount of previous work aimed at using the GPU [131–133] to accelerate iterative reconstruction approaches like simultaneous algebraic reconstruction technique (SART), there have been far fewer publications addressing FPGA implementations of iterative reconstruction. In [134], for example, backward projection was implemented on an FPGA, and the forward projection step was performed on a GPU. GPU and FPGA of course have very different features. GPU can have hundreds of parallel computing cores, and FPGA can support high performance logic customization for specific computations. A better performance design can be expected if an algorithm having significant computational diversity using the architecture advantages of GPU and FPGA are exploited. Moreover, the use of FPGA can help to significantly reduce the power consumption of the overall system.

The current literature for FPGA hardware-based MRI-CS is mainly targeted for filter algorithms [135],classifying images [136] and for CS reconstruction [137]. Moreover, the implementation is not completely FPGA-based. Multiples digital signal processing (DSP) cores are used in [137] and a combination of GPU and FPGA in others. Some implementations also utilize the high-speed feature of FPGA to control the complete system. In [107], the main kernel of the MRI system is FPGA-based and hence speeding up the data processing. In our proposed FPGA-based architecture, the complete system is implemented including the controller. This would make the complete MRI system portable on a Virtex or Xilinx FPGA. To the best of our knowledge, this is the first hardware architecture that implements a complete MRI-CS on a FPGA hardware.

5.3 System Architecture

A top level hardware block diagram is depicted in Fig. 5.1. As observed, the computational intensive component is the reconstruction process, which is an iterative process. It is composed of three major components, two for the core calculations, namely the least-squares component and multiply and sorting component, and one for data formation and control, namely the bus-control component. Among these three, different levels of parallelism are realized according to the priority and crucial levels of the algorithm. For example, the least-squares module is developed with extremely high parallelism and full pipeline, since it computes and updated estimates, and performs residue calculations. These are the important processing steps in the CoSaMP algorithm.

Hence, this architecture achieves a good trade-off between performance and resource consumption. Moreover, scalability is another remarkable aspect of our architecture that only the size of memory needs to be linearly expanded when

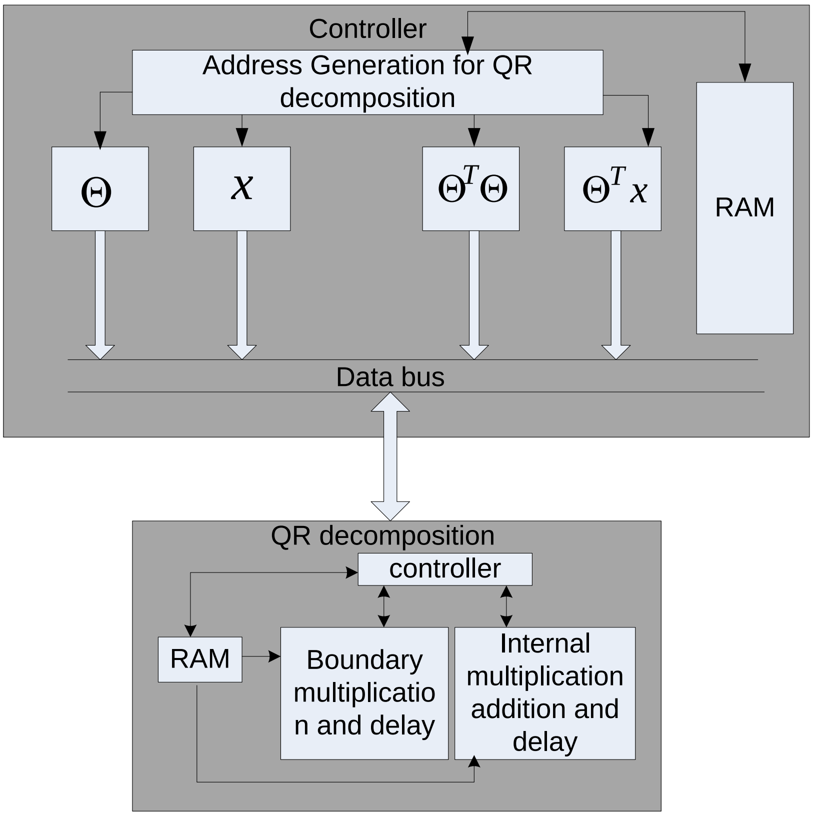

Controller
Address Generation for QR decomposition
Θ
x
$\Theta^T\Theta$
$\Theta^T x$
RAM
Data bus
QR decomposition
controller
RAM
Boundary multiplicatio n and delay
Internal multiplication addition and delay

Figure 5.1: Architecture of the hardware reconstruction process.

the length of the target signal is enlarged.

5.4 Hardware Process

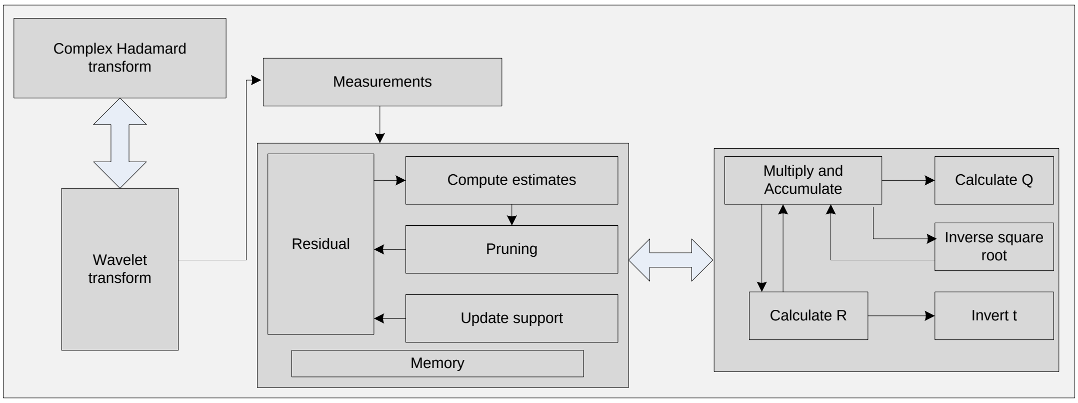


Figure 5.2: Block diagram for MRI hardware processing.

A generic CS system has two main blocks, namely, the compressive sampling and reconstruction blocks. CS utilizes two matrices for sparse representation and recovery, i.e., the Φ matrix and Ψ matrix. The proposed CHM is the Φ matrix and

Ψ is an identity matrix, and they are incoherent to each other. Using CoSaMP for reconstruction is most suited with CHM due to the reduction in running time, which is in the order of $O(N \log N)$. Fig. 5.2 depicts our proposed CS system architecture. The main components that are computationally intensive are CoSaMP reconstruction and QR decomposition associated with it.

In this paper, the optimization problem is dealt with a well-known greedy algorithm known as CoSaMP, which is an iterative reconstruction algorithm that offers rigorous bounds on computational costs and storage. It requires only matrix-vector multiplications with matrix Φ. It also provides a stopping criterion such that the reconstruction procedure stops after a fixed number of iterations. CoSaMP requires that the sparsity level K be provided as part of its input. For this purpose, when the signal length N is large, phase transition analysis suggests that most sparse signals can be recovered when $M \approx 2K \log N$. To reduce the running time, K can be varied along a geometric progression as $K = 1, 2, ..., M$ [53]. Therefore, CoSaMP is considered as an optimal choice for hardware implementation for sparse signal recovery. It is also to be noted that all greedy algorithms need square matrix calculations which are performed by iterations, resulting in high computational costs. This calls for a least squares method suitable for FPGA-based processing to be implemented in conjunction with CoSaMP.

CoSaMP requires matrix Θ, noise vector e, and sparsity level k as inputs. The output $\hat{x}^t$ of the system is an approximation of the original signal $\hat{x}$. If $\hat{x}^t$ is a k-sparse signal, then we need to find the M columns of Θ that contribute to y. At each iteration, we choose the column of Θ which is best correlated with the remaining part of y. We then determine its contribution, subtract it from y, and perform the next iteration on the residual vector. After finding the relevant columns of Θ, the values of the signal are found through solving a least square equation. After finding M columns of Θ which are closely related to y, the second stage is to solve the least square problem. This often involves finding the inverse of matrix H, where $H^{-1} = \Theta^T \Theta$. A set of eight multipliers are used in parallel to perform these steps. The critical problem in this method is to find the square root and the division in the final stage of each column processing. In this implementation, we use pipelined fixed-point inverse square root computation which utilizes only six clock cycles. This eliminates the division process and hence reduces the time and hardware area consumption.

The hardware architecture of the reconstruction algorithm of our proposed method is illustrated in Fig. 5.2. This architecture can be implemented on a field programmable gate array (FPGA), application specific integrated circuit (ASIC), graphic processing unit (GPU) etc. As shown, we have used QR decomposition in matrix computations for CoSaMP that involve complex matrix calculations. QR decomposition is a procedure where a complex matrix is decomposed into an orthogonal and a triangular matrix. Furthermore, this implementation of complex matrix provides a scalable architecture consuming only a small hardware area and memory utilization [138] [139].

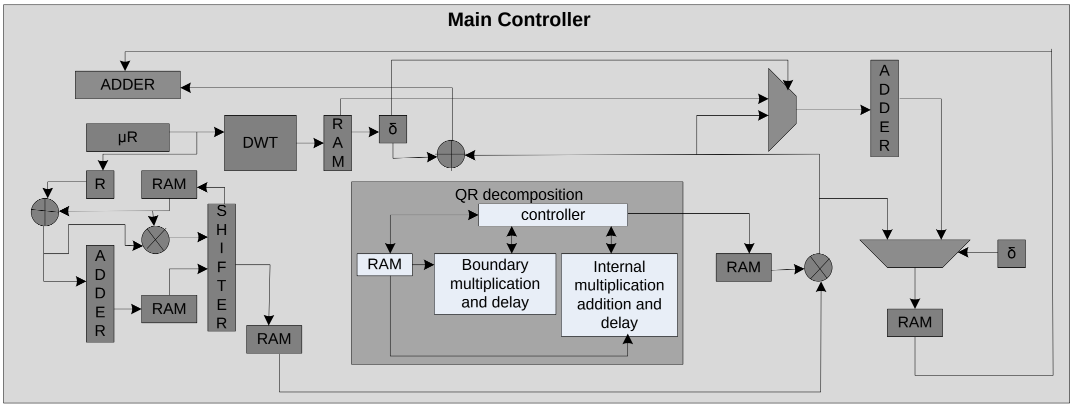


Figure 5.3: Internal structure of the reconstruction process.

Fig. 5.3 demonstrates the internal structure of the reconstruction process. The processing that takes place can be described as follows,

Multiply-sorting component: This component calculates the matching vector and gets the index collection by sorting. It is composed of a multiply module and a sorting module. As the realizations have some minor differences between the 1st and kth iterations, their working status should be switched by a temporary variable. In the multiply module, there are multiply-and-add sub-blocks in parallel to calculate the products in the every iteration. The result of multiply module is then transferred to the sorting module, which is made up of $3s$ comparators in serial. When all the N elements of vector y get through these comparators once, the indices of largest $3s$ is obtained. However, only the first $2s$ indices are needed for the rest of the iterations except in the 1st, where $3s$ is needed.

Least-squares component: This component solves the least squares problem by a highly parallelized and fully pipelined module. The module adopts QR decomposition algorithm, which is currently the fastest recursive algorithm, and is implemented in a linear systolic array. A systolic array is a pipeline arrangement of processing units, commonly used for parallel computing. The computed data is stored independently for each unit. Once the least squares calculation is complete, the result is transferred to a sorting module, which is composed of s comparators, to obtain the largest s elements. This is the new approximation of the target signal. This operation is the most crucial part in sparse recovery. QR decomposition has its unique advantages, which avoids burdensome matrix multiplications which are replaced by a series of rotations, and its excellent accuracy and stability.

5.4.1 QR Decomposition for Complex Hadamard Matrix

QR decomposition is a procedure where a matrix is decomposed into an orthogonal and a triangular matrix. This procedure is used in our FPGA implementation of complex matrix so that we can obtain a scalable architecture consuming only a small hardware area and memory utilization [138] [139].

The matrix D determines the orthogonal matrix Q and triangular matrix R, such that $D = QR$. Then the inverse of this matrix can be obtained by,

$$D^{-1} = (QR)^{-1} = R^{-1}Q^{-1} = R^{-1}Q^{H} \tag{5.1}$$

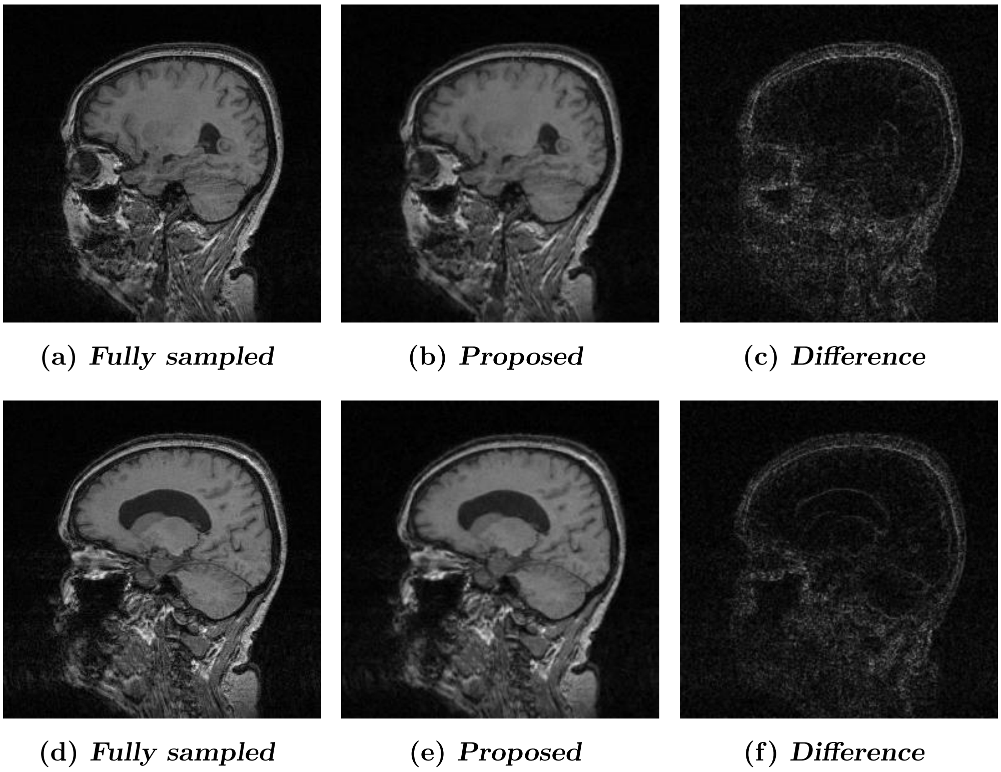
where Q^H is the Hermitian transpose of Q. Furthermore, this can be effectively implemented by a systolic array with processing elements based on the coordinate rotation digital computer (CORDIC) [140] algorithm. In our proposed system, we use the implementation from [139], which is based on the three angle complex rotation approach and enables significant reduction in latency.

All the matrix calculations perform fixed-point arithmetic, since this provides faster results and consumes less hardware. Since most of the processing involves multiplications, adder-shifter combinations are used wherever feasible. RAMs are used in the design to obtain an efficient sequential process and in turn provide a nominal operating frequency of about 82 MHz. By adopting this implementation procedure, we attempt to reduce the computational complexity and also speed up the 3D-MRI process. The system is ported on a FPGA and provides a processing time of 17μ secs per slice [141].

5.4.2 Data Processing on Hardware

In this paper, the hardware has been implemented for $N = 256$ and a sparsity of $K = 8$. Each data uses 24-bit (10 integer bits and 14 fractional bits) fixed-point format. It is observed that a larger number of fractional bits do not actually influence the result and our fixed-point format computation is comparable to the floating point simulation. To perform the dot product, 64 multipliers are operated in parallel and the results are added together. Multiply and addition are divided into 3 pipeline stages to decrease the logic output delays. Multiplication takes place in the first stage of the pipeline. In the second stage, eight additions are performed in parallel each adding eight values. These results are added to produce the final output in the third stage. It is fully pipelined so that the data is available at each clock cycle. Once the index that has close correlation to y is found, the residual is updated by subtracting it from the correlation of the columns of Φ.

The CoSaMP reconstruction requires as inputs, the CHM matrix Φ, noise vector e, and sparsity level K. The output α of the system is an approximation to the



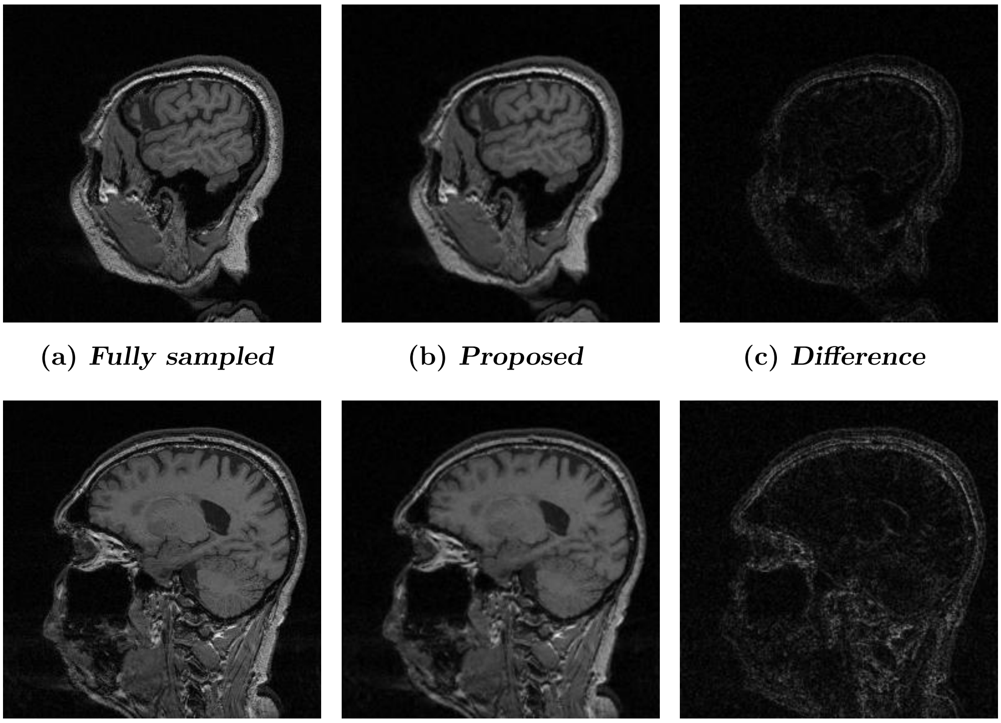
(a) ***Fully sampled*** (b) ***Proposed*** (c) ***Difference***

(d) ***Fully sampled*** (e) ***Proposed*** (f) ***Difference***

Figure 5.4: Reconstructed images of a two slices ((a) and (d)) from a total MRI scan samples of dataset-1 [1].

original signal x. If x is a K-sparse signal, then we need to find the K columns of Φ that contribute to y. At each iteration, we choose the column of Φ which is best correlated with the remaining part of y. We then determine its contribution, subtract it from y, and perform the next iteration on the residual vector. After finding the relevant columns of Φ, the values of the signal are found through solving a least squares equation.

After finding k columns of Φ which are closely related to y, the second stage is to solve the least square problem. This often involves finding the inverse of a matrix C, where $C = \Phi^T\Phi$. The main purpose of this is to solve for α'. Here, we use QR decomposition in a similar way as was used in the compression process. A set of eight multipliers are used in parallel to perform these steps. The critical problem in this method is to find the square root and the division in the final stage of each column processing. In this implementation, we use a pipelined fixed-point inverse square root computation which utilizes only six clock cycles. This eliminates the division process and hence reduces the time and hardware area consumption.



(a) ***Fully sampled*** (b) ***Proposed*** (c) ***Difference***

(d) ***Fully sampled*** (e) ***Proposed*** (f) ***Difference***

Figure 5.5: Reconstructed images of a two slices ((a) and (d))from a total MRI scan samples of dataset-2 [1].

5.5 Simulation Results

The proposed hardware architecture is implemented using the Verilog hardware description language, and synthesized using Altera Stratix IV E series FPGA [142]. This design utilizes 60% of resources of this FPGA capacity. It runs on a single clock frequency of 82 MHz and has three pipeline stages. This helps to overcome some of the bottlenecks caused by the multiplication and addition combinatorial logic in the design. The overall process takes about 810 cycles for data acquisition using the CHM and reconstruction of a 256 × 256 MRI image. Hence the total processing time is 22μ seconds. The major bottleneck of this architecture lies in the reconstruction process, where the residual needs to be computed, entailing complex matrix multiplications.

To provide a better insight into the efficiency of our architecture, we measure the reconstruction time alone and observe the processing time is 17μ seconds. This is about 7μ seconds faster than the architecture in [128], which is among the fastest architectures in the literature and implemented on a Xilinx FPGA. Furthermore, to obtain fair comparison, we also simulate and synthesize our design on a Xilinx Virtex 5 FPGA [143] and obtain a processing time of about 20.4μ seconds, which is still about 4μ seconds faster than the design in [128].

The comparison of our implementation with some of the existing FPGA and non-FPGA hardware solutions are shown in Table 5.1. All the tabulated architectures use 256 × 256 image for processing, and the running time is with respect to the reconstruction process. All the architectures use the conventional MRI sampling process based on the Fourier transform. Moreover, all the existing architecture are merely for reconstruction and not a complete CS system. Most of them implement the orthogonal matching pursuit (OMP) [128] [144] and total variance (TV) [145] reconstruction algorithms, while our proposed architecture is based upon the CoSaMP algorithm, which provides a pre-set stopping criterion that is not available in OMP and TV. This stopping criterion results in a guaranteed quality of the final approximation. The operating frequency comparison with that in [128] shows a difference of 45 MHz and the reason for this is optimization of the logic blocks for the place-and-route hardware process. This means the routing of data and control paths are near-optimal and utilizes minimal hardware resources, which can be observed during synthesis. Furthermore, our design is optimized to obtain an operating frequency of 82 MHz using sequential logic rather than combinatorial logic. An un-optimized architecture of our proposed design can also provide a similar low-frequency design, but will consume most of the FPGA hardware resources. Alongside a novel architecture, we also ensure that a low-cost optimal power design is provided.

Table 5.1: Hardware comparison with existing CS reconstruction architectures.

Architecture	Device	Frequency (MHz)	Time taken (secs)
Proposed 1	Stratix IV FPGA	82	17μ
Proposed 2	Virtex 5 FPGA	67	20.4μ
OMP [128]	Virtex 5 FPGA	39	24μ
OMP [144]	Virtex 5 FPGA	85 and 69	27.14μ
TV [145]	-	-	38m
OMP [146]	Intel core i7	-	68m
OMP [12]	GPU	-	37.5m

To validate the proposed system, several test data from [1] are considered. The data obtained is of a healthy male and female of age between 18-65 years of age. All the reconstructed images are of the size of 256 × 256. Some of the random samples are provided in Figs. 5.4 and 5.5 The reconstructed image is compared with the fully sampled Fourier MR image and difference is shown. The difference between our proposed reconstruction using the CHM and fully sampled image is also shown. It is observed that, there is still more improvement required, nonetheless has a good quality in comparison with the original. Unfortunately, due to the lack of PSNR results available in the literature, we could not perform a PSNR comparison with the existing architectures. It can be noted that, even after using fixed-point logic for all the arithmetic calculations, the PSNR is approximately 42 dB. A perceptual comparison of the proposed hardware outputs with the original image is also conducted, which demonstrates that the proposed design is able to provide images that are close to the original ones in reconstruction quality.

5.6 Practical Application

An important factor affecting the performance of CS-based MRI recovery is the sampling trajectory chosen in the frequency domain. Pure random sampling is impractical, due to hardware and physiological constraints. This directly impacts the RIP and coherence of the measurement matrix [147]. Hence it is suitable to use a structured matrix for a CS-based MRI.

This work is suitable for commercial implications provided that some hardware system related contingencies like the analog detectors, digitizers are resolved. These MRI scanner building blocks are designed to be used with Fourier transform. This would imply that MRI scanners currently available in market would need to undergo changes to accommodate the compressive sensing based module.

Compared to current MRI scanning time [148], the improvement expected is about 25% based on the simulation results. In saying that, the major roadblock would be the cost of changing the existing MRI scanners to suit CS techniques.

5.7 Summary

In this chapter, we present a complete hardware architecture of a CS-based data acquisition and reconstruction. The system is implemented on a Altera Stratix IV E series FPGA and verified for MRI suitability. This hardware is tested with various real data samples, sampled using the CHM and then reconstructed using CoSaMP. Furthermore, the performance is compared with existing software and GPU based implementations. QR decomposition is used for the implementation of the CHM in order to provide a fast, scalable and pipelined processing.

Chapter 6

Low-complexity Energy-Efficient CS-based Natural Image Processing Hardware

6.1 Introduction

In this chapter, we aim to provide a low-complexity energy-efficient framework for image processing based on CS principles. In the previous chapters, complex Hadamard matrix (CHM) and CoSaMP for MRI data was proposed, and proved superior when compared to some of the popularly used CS methods. We extend this concept for image processing with minimal modifications applied to the measurement matrix Φ, hence maintaining the originality of the matrix. Furthermore, the proposed concept will be incorporated in the discrete wavelet transform (DWT) of JPEG 2000, to provide an energy efficient hardware architecture.

Conventionally, after acquisition of an image, transform is performed on the image using pixel values. Afterwards, many coefficients that carry negligible energy are discarded prior to entropy coding. Therefore, much of the acquired information is discarded during this process although the image is fully acquired. In this Chapter, an alternative coding paradigm to conventional image compression is proposed based on CS principles. Two-dimensional discrete wavelet transform (DWT) is applied for sparse representation. Unlike in the JPEG 2000 encoder, the DWT coefficients are not directly encoded, but re-sampled with equal importance of information instead. At the decoder side, CS reconstruction is incorporated in the JPEG 2000 decoder. The recovery quality depends on the number of received CS measurements, and not which of the measurements that are received.

Most of the work in the literature on CS-based image processing (sometimes termed as compressive imaging (CI)), use measurement matrices that are either random or structured, but the use of complex matrices is not known. The complex Hadamard matrix is chosen not only for the reason that it satisfies the RIP conditions, but also for its suitability of implementation on hardware platforms. Moreover, we also investigate the behavior of the CHM with respect to natural images. Traditionally, the performance metrics for signal processing are latency and throughput. However, with the growing industry of portable, mobile devices, it has become increasingly important that systems are not only fast, but also energy-efficient. One such high computation requirement is for imaging applications. Due to this reason, an FPGA-based system presents a very viable solution. Currently, only a few commercially available FPGAs provide both millions of gates and low-power features. At the same time, matching the image compression algorithms to completely use these FPGA features is necessary. Thus, instead of low-level hardware optimization techniques, algorithmic techniques for minimizing energy dissipation is viable.

Keeping in mind the requirements of providing an efficient framework, the following contributions are made:

1. A complete framework of compression and reconstruction of natural images based on CS principles is proposed.
2. The proposed complex measurement matrix is combined with the most popular CS reconstruction algorithms, and compared with CoSaMP. The use of random matrices with CoSaMP is also demonstrated to verify the efficiency of the proposed framework.
3. A low-complexity energy-efficient hardware architecture based on FPGA is presented so that the complete JPEG 2000 is energy-efficient.

6.2 Related Work

The key elements of compressive imaging are the measurement matrix and reconstruction algorithm. The measurement matrix is selected based on a sufficient condition that satisfies the restricted isometric property. Several matrices have been proposed in the literature for image/video CS, such as independent identically distributed Gaussian matrix [100], and Bernoulli matrices [101] [102]. Their main advantage is that they are universally incoherent with any sparse signal and thus, the number of compressed measurements required for exact reconstruction is almost minimal. However, they inherently have two major drawbacks for practical applications namely, huge memory buffering for storage of matrix elements and high computational complexity due to their completely unstructured nature [59].

The authors in [100], [149] reduce the sampling rate for image/video signals by allocating the CS measurements according to the sparsity of images/frames. They exploit inter-frame correlation to predict local sparsity for image blocks [100]. However, the approach cannot be applied in single-image acquisition and can have poor performance in recovering high speed objects. Further, they utilize the information within one image block to help to predict the sparsity of neighboring blocks, but is difficult to implement in a parallel-processing system.

Another class of matrices based on Fourier and Hadamard were also proposed [103], where it is called the scrambled block Hadamard ensemble. Partial Fourier transform [103] has fast computational property and thus significantly reduces the complexity of a sampling system. However, it is only incoherent with signals which are sparse in the time domain, severely narrowing its scope of applications. Random Fourier matrices in the wavelet domain applied to the whole image are proposed in [150]. Simultaneously, this need to send the sampled data until the whole image is measured, which are not suitable for image reconstruction applications with limited storage and complexity. The authors in [151] proposed block compressive sensing for natural images, using the techniques of hard thresholding and projection onto the convex set (POCS). Here, image acquisition is conducted through the same measurement operator in a block-by-block manner, motivated by the success of the block DCT coding framework used in JPEG. However, the used frame expansions are not adaptive for all blocks.

JPEG 2000 is one of the most commonly used compression standard for image processing. Inspite of being efficient compared to JPEG standard, it has some shortcomings. The DWT block which is a 9/7 lifting wavelet transform is computationally intensive and requires fast algorithms to cope with real-time applications. This make the system highly complex and also consumes high power. Incorporating CS processing with the conventional image coders have also been explored in [152–154]. In [152], the discrete cosine transform (DCT) is used for smooth regions, and a bi-orthogonal wavelet transform is used for uneven regions. The CS acquisition is split as low and high frequency components in [153], while the results of [154] are suitable for lossless compression. All these perform reconstruction using OMP, which is not suitable for hardware implementation due to its computational complexity rising from the iterative bounds.

Hence to address the discrepancies of the existing architectures, a low-complexity energy-efficient architecture is investigated. This architecture incorporates compressive sensing technique, which requires few samples for exact recovery, and a low-power consuming transform and memory design that would provide a platform for real-time processing.

6.3 System Model

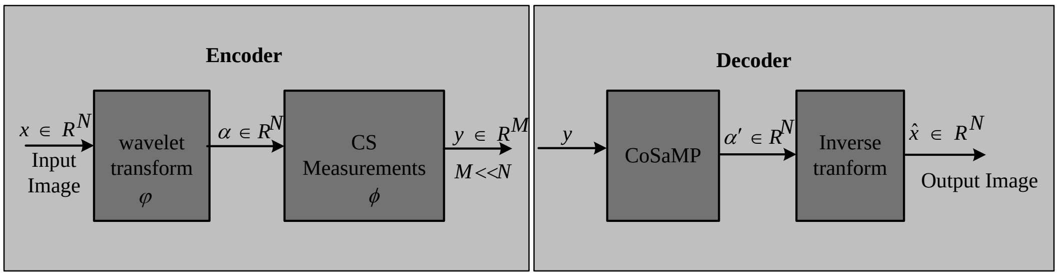


Figure 6.1: System model with for CS processing.

The CS encoder consists of an image transform, quantization and a measurement matrix block. As shown in Fig. 6.1, the image data x is transformed using one of the most commonly used image transforms, such as the discrete wavelet transform used in JPEG 2000. The transformed data α is CS sampled using matrix Φ to obtain k-sparse measurements. The CS decoder consists of two blocks, i.e., the CS reconstruction and inverse image transform blocks. The reconstruction is performed by linearly optimizing a set of equations using matrix Φ. Once the measurements are reconstructed, the original signal is obtained from matrix Ψ and these reconstructed measurements.

The wavelet transform is adopted since the wavelets have time-frequency location and multi-resolution characteristics, and therefore can decompose the image signal into a number of sub-band signals in different spatial resolution, frequency and directional characteristics. The wavelet transform also overcomes the block artifacts which are usually present when other transforms are used instead. The compressive sensing matrix employed in this processing is CHM, which is similar to the one used for 2D/3D MRI in Chapter 3, but with the columns of the matrix randomized. The proposed matrix Φ satisfies the RIP and hence it is possible to recover the signal correctly, and thus suitable for CS-based image processing. Relating the proposed matrices to (3.3), the DWT and CHM are denoted by Ψ and Φ, respectively. If Φ is a structurally random matrix, its rows are not stochastically independent because they are randomized from the same random seed vector and thus are correlated. This is the main difference between a structurally random matrix and a sub-Gaussian matrix. Relaxing the independence among its rows enables a structurally random matrix to have some particular structure with fast computation.

When considering natural images for processing, it is extremely rare that an image can have non-zero values and therefore CS cannot be applied as is. It is a known fact that, any image can have a sparse representation in a certain transform domain, which is the 9/7 irreversible DWT in our system. This transform is used for lossy compression in JPEG 2000 [155]. The main advantage over the discrete cosine transform (DCT) is that, DCT previously carries out a division into squared blocks, while the DWT works in its totality. Moreover the decomposition into subbands gives a higher flexibility in terms of scalability in resolution and distortion.

At the decoder, CoSaMP reconstruction is performed followed by the inverse discrete wavelet transform (IDWT). Since the process is for lossy compression, noise component is included during the CoSaMP reconstruction iterations. This is to ensure a perfect reconstruction of the image.

6.4 CS Processing

The matrix Φ used in image processing varies when compared to use in MRI processing. In the case of MRI, the CHM is used for data acquisition and the matrix is structured. In case of image processing, measurement samples are acquired only after the image transform. Though the basis of the matrix Φ still remains to be a CHM, the diagonal entries are randomized to provide better quality results for a variety of images. The matrix Φ is defined as

$$\Phi = \sqrt{\frac{N}{M}}\mathbf{R}, \tag{6.1}$$

where $\mathbf{R} \in N \times N$ is a diagonal random matrix whose diagonal entries are random variables R_i with identical distribution $P(R_i = \pm 1) = 1/2$. This diagonal matrix of random variables flips signals sample signs locally. The scaling coefficient $\sqrt{\frac{N}{M}}$ is to normalize the transform so that energy of the measurement vector is almost similar to that of the input signal vector. Once randomized, the entries are i.i.d Bernoulli random variables.

With (6.1), the framework can recover k-sparse signals exactly as per the following:

Theorem 10. *Recovery of k-sparse signals exactly, with a probability of at least $1 - \delta$, if the number of measurements are $M \geq O(\frac{N}{B}klog^2(\frac{N}{\delta}))$. For the DWT, the number of measurement needed is on the order of $O(Klog^2(\frac{N}{\delta}))$.*

Proof. This is similar to the corollary of Candes et. al [[57], Theorem 1.1]. It can be said so, due to the fact that Φ being an orthonormal matrix (when randomized with $\mathbf{R}$) representing the mutual coherence between Φ and Ψ. The mutual coherence once again is similar to the work of Do et. al [[156], Theorem III.A]. ■

Having the measurement matrix Φ with restricted isometry constant δ and $y = \phi x + e$ is a vector of samples of an arbitrary and e is the noise vector, then CoSaMP produces a k-sparse approximation to a that satisfies

$$\|x - a\|_2 \leq C.\max\{\eta, \frac{1}{\sqrt{k}}\|x - x_{s/2}\|_1 + \|e\|_2\} \tag{6.2}$$

where η is the precision parameter and $x_{k/2}$ is a best $k/2$-sparse approximation to x.

6.5 Hardware Architecture

The hardware architecture details the structure of the encoder and decoder structure and interface of DWT and randomized CHM. The main aim is to have a low-complexity and energy efficient design. This can be achieved by processing techniques such as pipelining and a combination of parallel-pipeline processing for arithmetic elements. Pipelining is an efficient design practice for both time and energy performance. In FPGA designs with large data, throughput is another important factor in power dissipation. Pipelining is a technique in which increasing the power dissipation may decrease the overall energy dissipation. Moreover, CS further contributes in having a energy efficient design by using up only a small percent of samples when compared with conventional image processing methods. In this section, an encoder and decoder design that includes pipelining and parallel processing is presented.

6.5.1 Encoder

Fig. 6.2 outlines the flowchart for the encoder. The processing is performed row-wise first and then column-wise. This ensure that the arithmetic computations are re-used and hence reduce the complexity and in turn lead to a energy efficient design.

With reference to Fig. 6.3, once the input coefficients and quantization steps are available, the DWT is performed. Initially the pixels of a row are fetched into the row processor. The 4 lifting steps of the DWT are applied to all pixels in that row. The lifting structure is show in Fig. 6.4. Specifically, the diagram shows the signal flow for samples $x0$ to $x8$. A pair of samples at equal positions is weighted by negative coefficients α and added to the intermediate sample. The next lifting step combines the results of the summations in pairs using the coefficients β. The third and fourth lifting step act in a similar manner using the weights γ and δ. The property of integer-to-integer mapping, which will be essential for lossless compression, is simply imposed by properly rounding the intermediate values to

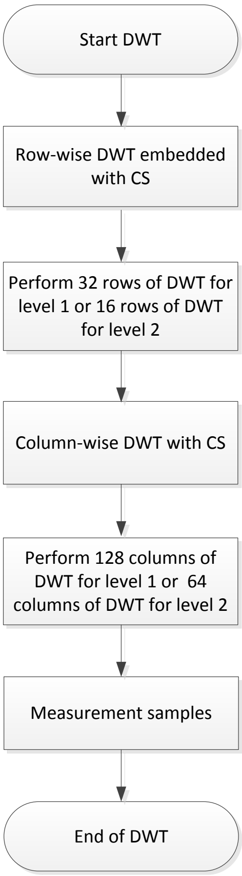

Start DWT
Row-wise DWT embedded with CS
Perform 32 rows of DWT for level 1 or 16 rows of DWT for level 2
Column-wise DWT with CS
Perform 128 columns of DWT for level 1 or 64 columns of DWT for level 2
Measurement samples
End of DWT

Figure 6.2: Flow chart for the proposed encoder architecture.

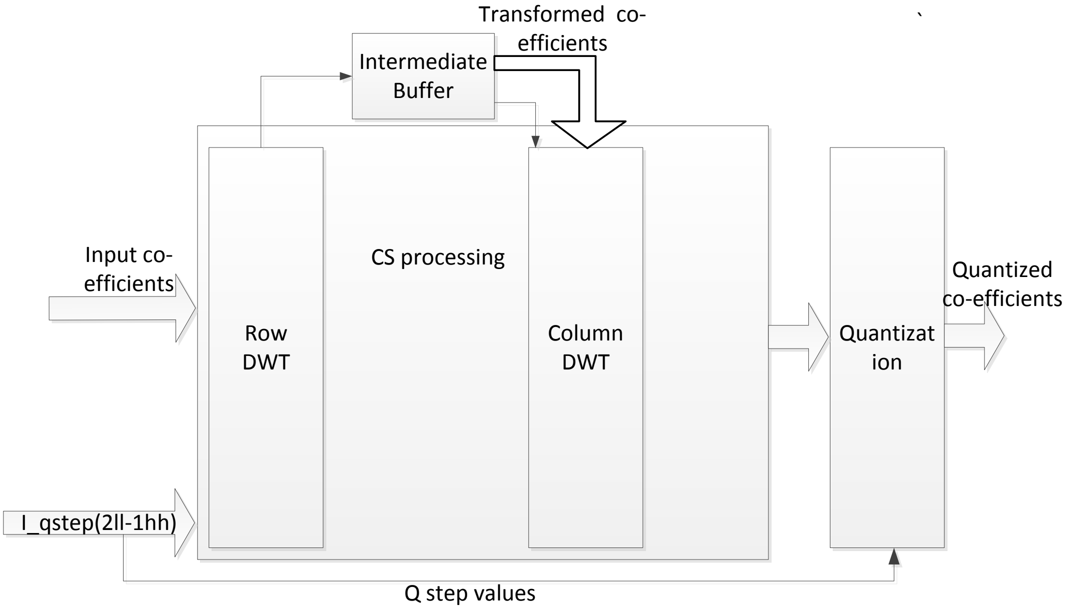


Figure 6.3: Detailed block diagram of the proposed encoder.

integer values. The result after all the lifting steps is an interleaved sequence of a low-pass filter output and a high-pass filter output.

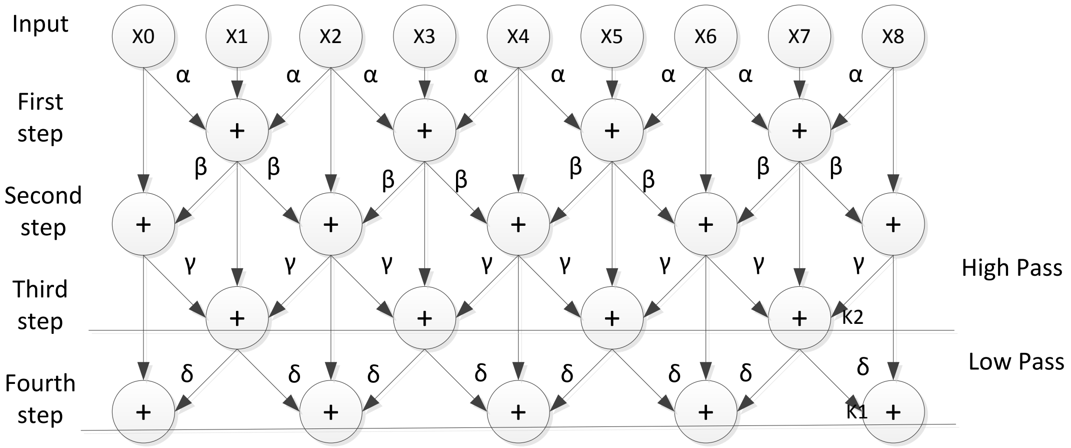


Figure 6.4: 9/7 lifting wavelet structure. α, β, γ and δ are the lifting parameters.

In the column DWT, the 4 steps of the DWT is performed sequentially. There are two column processors performing column DWT simultaneously. When the 2D-DWT of first level decomposition is done the coefficients of the 1HL, 1LH and 1HH are compressive sensed. This ensures that the samples required for further processing is very minimal and this affects the transmission to a greater extent. The next step is quantization and is pipelined with the second level of DWT decomposition. The whole processing is pipelined and the last to perform is the 2LL to 2HH sub-bands quantization. The timing diagram for the encoder processing is depicted in Fig. 6.5, which shows the pipeline structure employed.

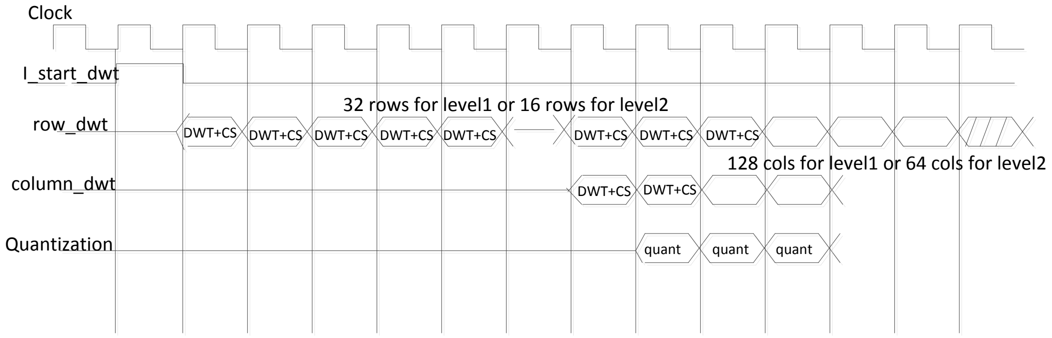

Clock
I_start_dwt
row_dwt
column_dwt
Quantization
32 rows for level1 or 16 rows for level2
DWT+CS
128 cols for level1 or 64 cols for level2
quant

Figure 6.5: Timing diagram of DWT with CS.

6.5.2 Decoder

Fig. 6.6 depicts the proposed decoder that incorporates CS reconstruction with the typical JPEG 2000 decoder structure. The main component is the IDWT

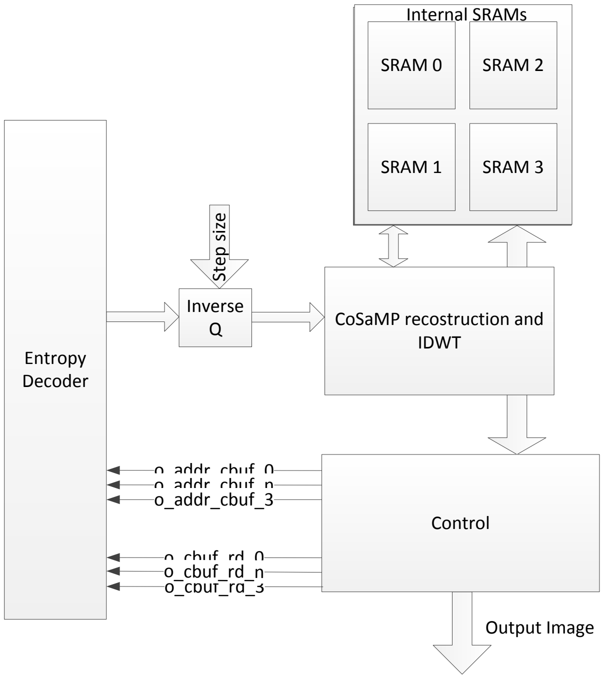


Figure 6.6: Block diagram of the proposed decoder.

combined with the CoSaMP algorithm. In the IDWT architecture, only one core component works sequentially. Inverse quantization is performed only at the start of processing. The IDWT processing is sequential i.e., Level 2 column processing core (CPC) then the Level 2 row processing core (RPC), which is followed by Level 1 CPC and RPC. An address mechanism is used for reading from the external code block memory and to write into the internal wavelet memory. This mechanism is necessary for proper processing, since the input to the core (i.e., for CPC and RPC) is in the form (H L H L . . .), which is different from the data sequence stored in the memory.

For each sub-band, the data is multiplied with the quantization factor. The data is of 10 bits, with the MSB being the sign bit. For multiplication purposes, 9 magnitude bits are used and the output is converted into its 2's complement based on the value of the sign bit. The output is concatenated with 0's or 1's to convert to 16 bits prior to the IDWT processing. The core is based on the 9/7 lifting scheme shown in Fig. 6.7. The IDWT filtering algorithm basically consists of four lifting steps, and hence the computation is performed in four stages. These four stages are realized as a single combinational circuit. The intermediate results generated at all stages are temporarily stored for further pipelined processing. A particular component with the IDWT block, which performs both column and row processing is based on a flag bit. Its input is fed through multiplexors, which are controlled by the address generation mechanism modules.

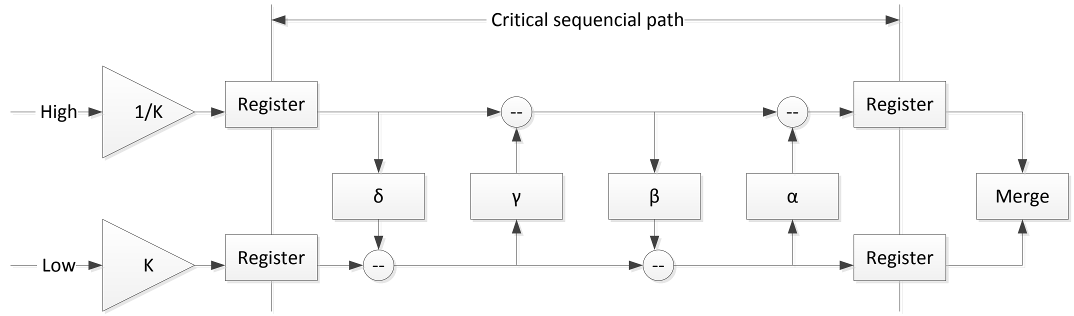


Figure 6.7: Inverse lifting wavelet structure.

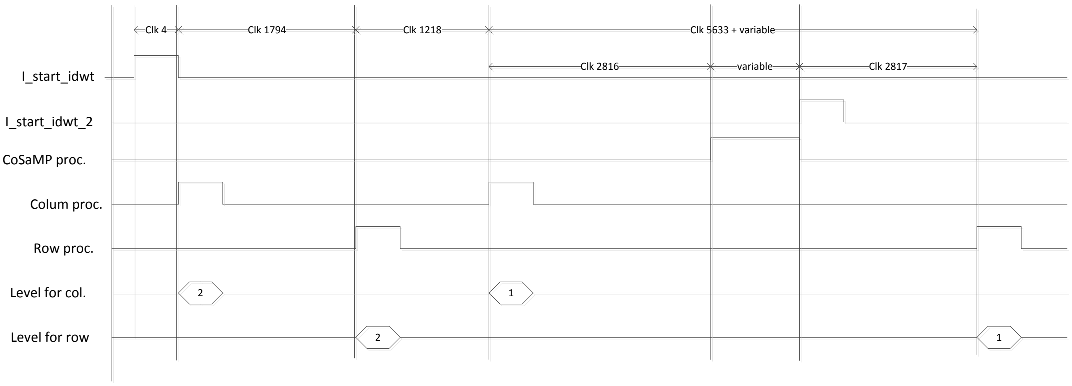


Figure 6.8: Timing diagram of the decoder.

6.5.3 Energy-efficient Processing

Memory blocks are the one of the most frequently used block and it also has high power dissipation. The other block is the transform block, which requires multiplications. In order to enable a low-power storage for the proposed design, power consumption for various types of memories is analyzed based on Altera FPGAs. Fig. 6.9 illustrates the power dissipation for three possible bindings for storage Altera FPGAs based on the number of data entries; namely, registers, slice based RAM, and block RAM. For large storage elements, those with more than 30 entries block RAM shows an advantage in power dissipation over other memory implementations. Hence, in the proposed image processing systems, we consider the use of block RAMs. In addition, the memory mapping of these RAMs are managed in a way that only a minimal number of accesses are required. The mapping approach consists of two algorithms that obtain a power-efficient mapping of logical memories to FPGA embedded memory blocks. Since most embedded memory block dynamic power is a result of clock-induced pre-charging, some specific cases are identified where user specified RAM read and write enable signals can be automatically converted or combined with the corresponding read and write clock enable signals. In some cases, memory banking is done. As a result of this banked mapping, only one embedded memory block is clocked per

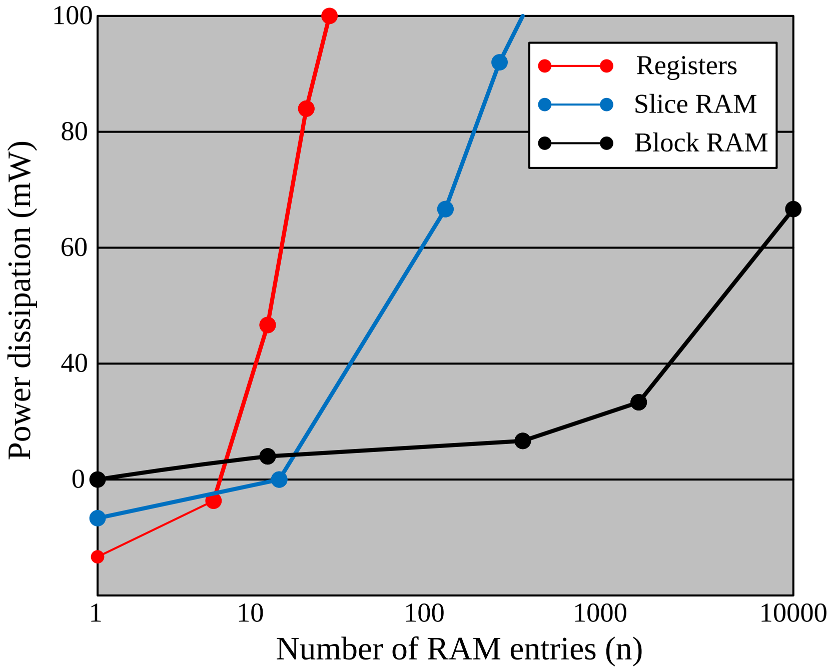


Figure 6.9: Graph showing power consumption with various FPGA storage options.

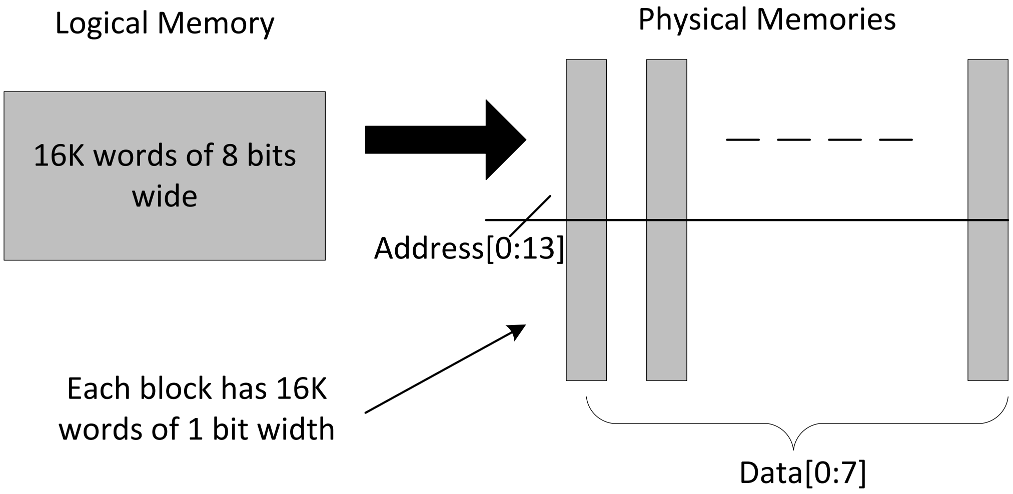


Figure 6.10: $16K \times 8$ bits memory mapping without selection logic.

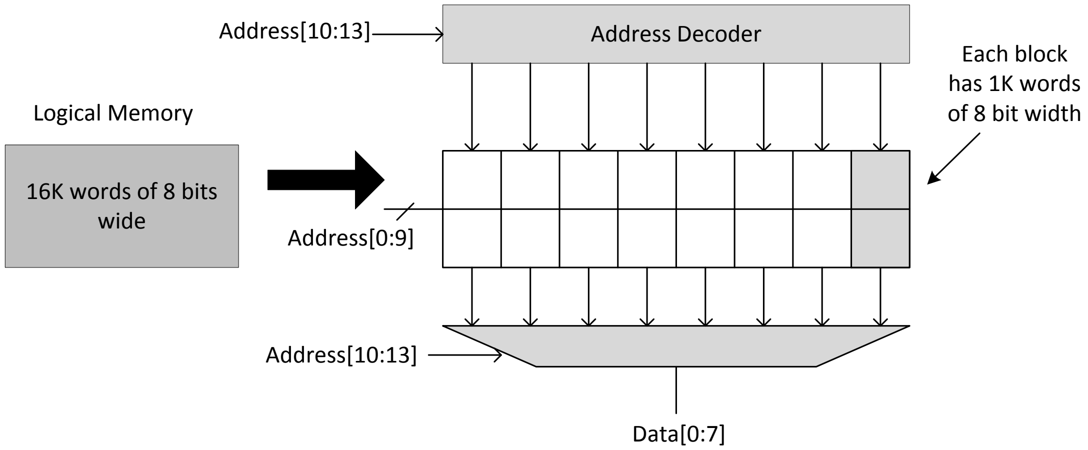


Figure 6.11: $16K \times 8$ bits memory mapping with banking and address selection logic.

access and to perform this some supporting logic is added. In all the cases, the correct functional behavior is ensured.

The conversion of user-specific read and write enable signals to respective clock enables primarily reduces power by eliminating line pre-charging when embedded memory block data access is not required, and maintains the functionality. The combining of the data enable and clock enable signals, forms a new combined clock enable signal, which can be attached to the memory port clock enable input. Figs. 6.10 and 6.11 depict two different memory mapping alternatives used in the proposed system. In the mapping in Fig. 6.11, the width of each physical memory block matches the width of the logical memory, whereas the depth of each physical memory block is reduced compared to its logical memory counterpart. This mapping requires the inclusion of address decoding circuitry to determine which memory block contains the requested data. In addition, a multiplexer is required on the read port to select the requested word during read requests. Although dynamic power is consumed by the added address decoder and multiplexer, all but one of the embedded memory blocks are disabled during RAM accesses, saving considerable dynamic power. Unused memory blocks are disabled by connecting the outputs of the address decoder to memory block clock enable signals.

Other than memory blocks, multiplication logic blocks also consume more power when compared to other blocks in the design. To provide an overall energy-efficient design, restructuring the multiplication block is done in our proposed system. The matrix multiplication algorithm considers two $n \times n$ input matrices A and B, and computes the product $C = A \times B$. The architecture utilizes parallelism and pipelining, also includes logic for the output matrix C. The output of first product is overlapped with the computation of the next and hence

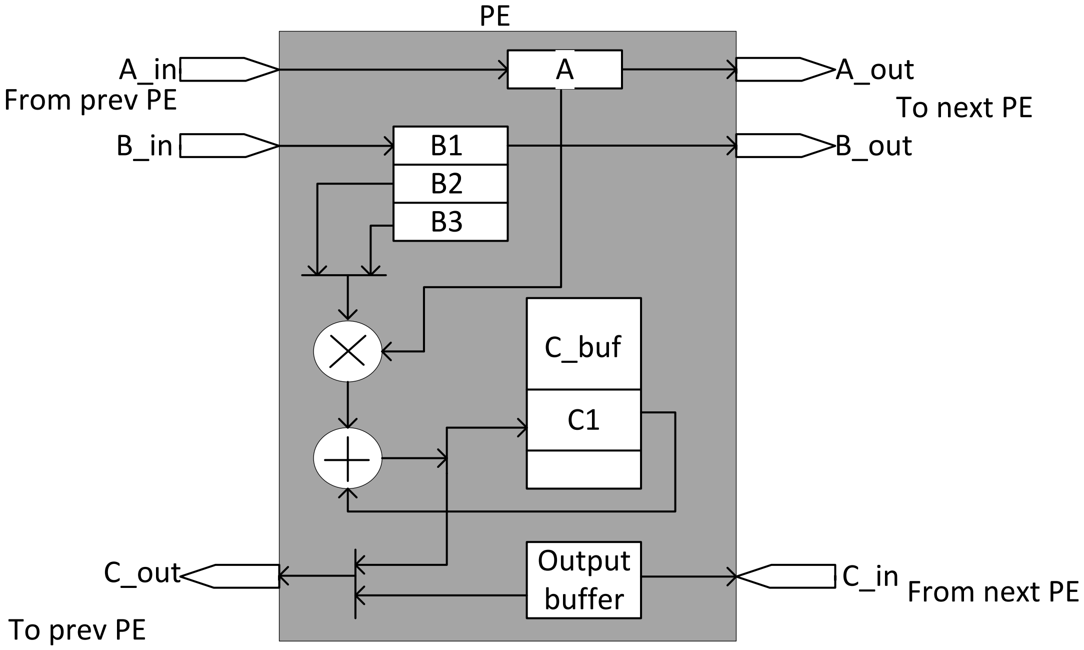


Figure 6.12: Single processing element (PE) of a multiplication block.

no wait cycles are wasted. Thus, a throughput of one data sample per clock cycle is achieved. Fig. 6.12 shows the architecture of a single processing element (PE) of a linear array of multiplication block. The algorithm computes the product efficiently, both in terms of latency and energy, by cleverly moving the entries of the input matrices through the linear array. The entries from matrix A are fed into the linear array in column-major order from the block memory, while the entries from matrix B are fed into the linear array in row-major order. Furthermore, the entries from matrix A does not begin until n cycles after the entries from matrix B. The PE computes the sums of products, which are entries for matrix C. In the figure, A, $B1$, $B2$, and $B3$ are the temporary storage registers, A_in, B_in are the inputs and C_in is the output. Since there are multiple PEs, each PE input will have the i-th row and the k-th column of matrix A and the k-th row and the j-th column of matrix B, and the corresponding output will be i-th row and j-th column of matrix C. The execution time is $2n-1$, since processing is pipelined and input from left to right. This linear array based design ensures that connections are only made between neighboring PE and further ensures that only short interconnects are used. All outputs flow from right to left and each PE is always active, which maximizes the throughput. For this architecture, when $n > 24$, block-wise matrix multiplication with blocks of size (n/p) is used, where p is the number of PEs. This technique decreases throughput but saves area and energy.

6.6 Numerical Results

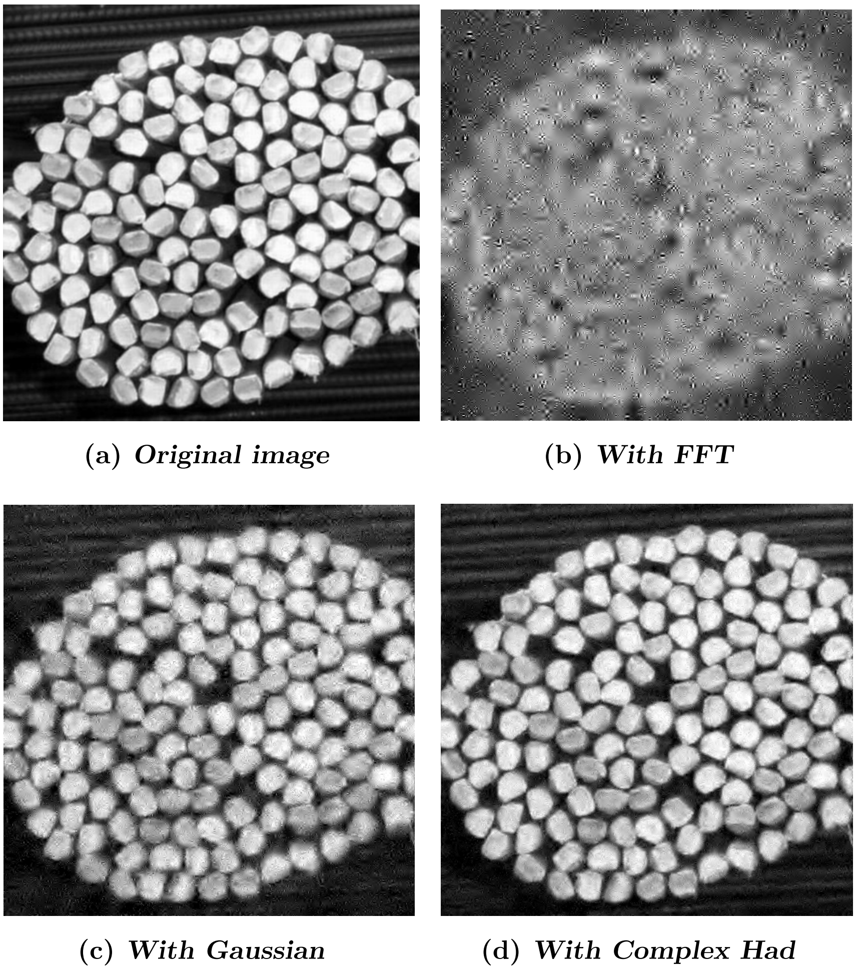
This section deals with two simulation aspects of CS-based natural image processing. Firstly, the comparison of various popular reconstruction algorithms against the choice of CoSaMP reconstruction is performed. Secondly, simulations are conducted to demonstrate the performance with respect to the conventional JPEG 2000 system. Alongside, energy efficiency in terms of latency ad resource utilization is compared.

Table 6.1 presents the computation complexity for block processing of an image. This computation is based on the running time of a 256 × 256 block size in a FPGA hardware environment. The running time is indicated through the matrix vector multiplications because in hardware-based implementations, higher the multiplications/divisions more complex is the system. Additionally, the systems is pipelined to have optimal utilisation of resources. The Gaussian based processing uses the Gaussian elimination method [157] and hence the complexity is $O(N^3)$ [158]. FFT uses the hardware based on CooleyTukey algorithm [159] with a computation complexity of $O(N \log N)$ [160]. In case of CHM, the complexity is calculated based on the time required to read the data, perform DWT and simultaneously perform CS processing. The overall computation time is $N + 1$ cycles, with DWT complexity being $O(N)$. The added complexity to this is the CHM based CoSaMP which is $O(M \log N)$. Therefore, due to pipelining and reusing arithmetic hardware, the system complexity is $O(N)$.

Table 6.1: Computational complexity for CHM, random FFT, random Gaussian in block processing. The complexity is based on a $M \times N$ matrix for a k-sparse basis. CoSaMP reconstruction is used in all cases.

CS Algorithm	Complexity
CHM	$O(N)$
FFT	$O(N \log N)$
Gaussian	$O(N^3)$

To demonstrate the system, a couple of test images were used. Fig. 6.13 shows one of the test images that is reconstructed using the random FFT, random Gaussian and CHM with 2K measurements. The visual quality can be easily compared with the original test image, and shows that the reconstruction quality with CHM is far superior than random FFT and random Gaussian. The PSNR for this test image with various measurements is also tabulated in Table 6.2. From this we can conclude that by using less number of measurements, the reconstructed image has a better quality than its counterparts. The CHM shows a consistent higher performance by approximately 3dB than Gaussian for most of the measurements.



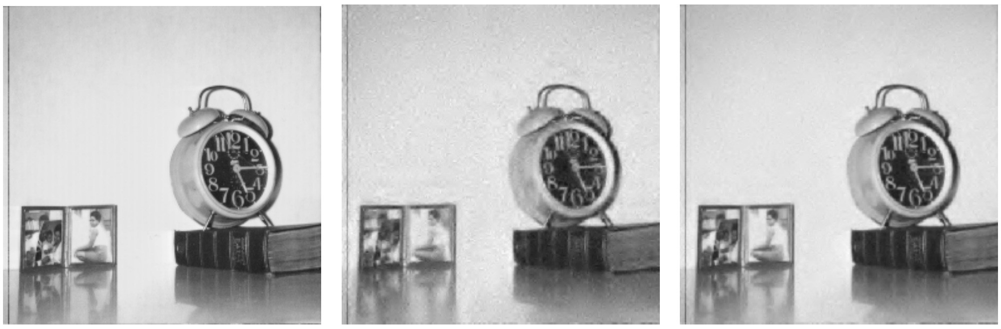
(a) ***Original image*** (b) ***With FFT***

(c) ***With Gaussian*** (d) ***With Complex Had***

Figure 6.13: Original and reconstructed test images.

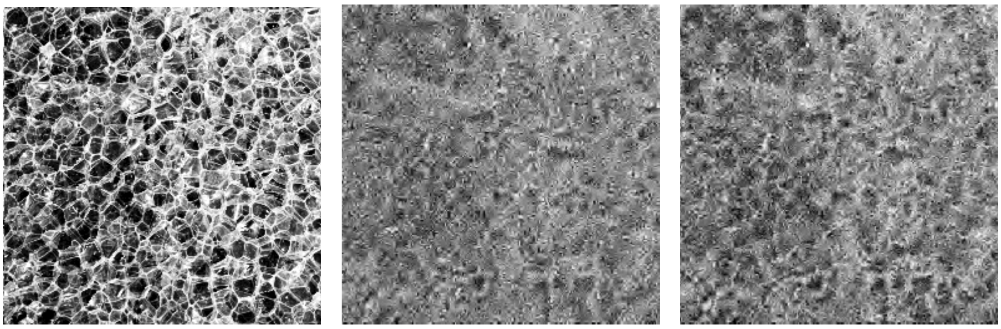
Table 6.2: PSNR performance using a 256 × 256 test image.

Measurements M	FFT dB	Gaussian dB	Proposed dB
1K	9.24	16.36	17.22
2K	10.27	18.17	21.35
3K	11.07	18.36	23.96



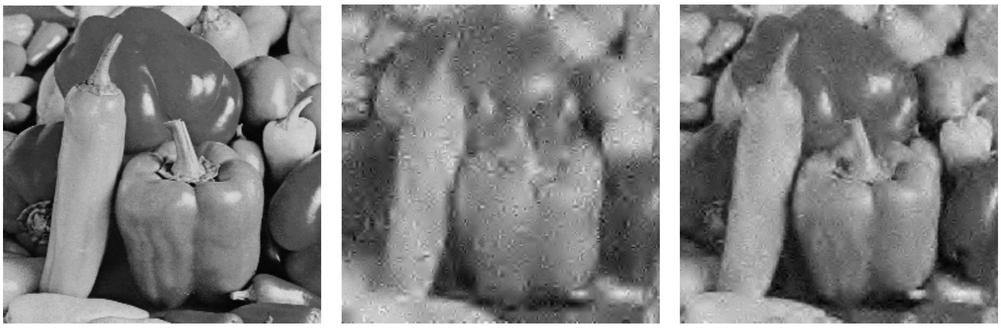
(a) ***Original image*** (b) ***With 3K samples*** (c) ***With 8K samples***

Figure 6.14: Original and reconstructed output of a 256 × 256 image-1.



(a) ***Original image*** (b) ***With 3K samples*** (c) ***With 8K samples***

Figure 6.15: Original and reconstructed output of a 256 × 256 image-2.



(a) ***Original image*** (b) ***With 3K samples*** (c) ***With 8K samples***

Figure 6.16: Original and reconstructed output of a 256 × 256 image-3.

Table 6.3: PSNR performance of various test images considered.

	PSNR with 3K samples	PSNR with 8K samples
Image-1	27.72	33.28
Image-2	22.96	29.36
Image-3	25.45	33.06

Figs. 6.14, 6.15 and 6.16 show the reconstruction data of three test images using proposed encoding and decoding with 3K and 8K measurements respectively.

Comparing the PSNR in Table 6.2, it can be noted that more the measurements, better is the image quality. Each test image has different kind of complexity involved. For example, the image-1 has a large part with a plain background, and hence the PSNR value is greater than the other test images. Similarly, since image-2 has irregular structure throughout, its PSNR at 3K measurements is comparatively low.

6.7 Summary

In this paper, a novel approach of compressive sampling using complex measurements is proposed. This matrix is compared with some of the existing methods and the performance is observed to be at least 3 dB higher. The proposed framework provides an advantage that, it needs very low measurements to represent the image. Alongside, it also yields a high quality output which is close to the conventional JPEG 2000 processing.

This architecture provides four important features: (i) It is universal with a wide range of sparse signals; (ii) The number of measurements required for exact reconstruction is nearly optimal; (iii) It has very low complexity and fast computation based on block processing; and (iv) Minimal computation/memory requirement and high quality of reconstruction.

Chapter 7

Two-symbol Arithmetic Encoding Architecture For Efficient Entropy Coding in CS-Based JPEG 2000

7.1 Introduction

The JPEG 2000 encoder architecture includes the building blocks of component transform, discrete wavelet transform, quantization, embedded block coding with optimized truncation (EBCOT) [155] [161], and the rate allocation. The main blocks that are computationally complex and clock hungry, are the DWT and the EBCOT blocks. In EBCOT, AE processing is serial in nature and hence increases the system latency. In Chapter 6, an efficient design for transform incorporating CS principles was demonstrated. In this chapter, the aim is to provide an efficient arithmetic encoding technique for JPEG 2000.

Nowadays, almost every multimedia application necessitates good compression techniques, needs to provide efficient solutions, and requires an excellent visual quality. To support these features, the algorithms that are employed are computationally intensive and complex. The JPEG 2000 standard [155] is an image compression standard, featuring low bit-rate, lossy and lossless coding, region of interest and error resilience. JPEG 2000 is superior to the original JPEG standard in the sense of both performance and functionality [161].

The contribution in this chapter is outlined as follows

- A two-symbol hardware architecture is designed and the step-by-step process is explained;

- The critical path is analyzed for the proposed architecture;
- A complete CS-based JPEG 2000 encoder hardware architecture is presented.

7.2 Related Work

The AE algorithm is serial in nature and hence the major throughput bottleneck of JPEG 2000. The standard [155] provides a reference AE implementation, which processes one CX-D pair at a time. Previous work has proposed several methods where one CX-D pair is processed with FIFO (first in, first out) inserted between BC and AE. However, this implementation can alleviate the problem only to a very small extent and results in an increase in hardware resources due to the inclusion of the FIFO module. Other single symbol processing methods (e.g., [162]) have doubled the frequency of operation than that of BC, which can offer the required AE performance but with drawbacks of clock-domain crossing issues and tedious methods to solve them. There are methods where separate AE modules are used for each of the three passes of the BC module. This kind of implementation reduces coding efficiency and the correlation of successive symbols that have not been considered appropriately. In [163], the bit-plane coder, FIFO and the AE modules are designed to achieve a high-speed low-power EBCOT module and there is a 27% improvement in power consumption. However, this implementation requires many hardware resources and although two CX-D pairs are read at once, they are not processed in every clock cycle.

A split arithmetic encoder is proposed in [164], which operates at 9.25 MHz and provides a 55% increase in performance compared to the standard architecture, but fails to provide a solution for AE which can handle the BC throughput effectively. The AE module stalls many times waiting for the context update tables and this causes a major bottleneck in delivering an efficient solution. A dual context modeling architecture is proposed in [165], where the AE module is processed in four pipeline stages and operates at 185 MHz. Since a pass switching technique is used, the coding efficiency is reduced drastically.

In [166], AE is implemented to operate at twice the frequency of the bit-plane coder, but only 25% improvement in throughput is observed. Chen *et.al* [167] propose a parallel AE implementation which encodes 50 Msymbols/sec at 100 MHz. However, this architecture lacks an optimized hardware implementation. Another implementation of AE has four MQ-coders used in parallel to match up the speed of the BC module generating CX-D pairs [168] [169]. This implementation consumes a large hardware area.

As mentioned earlier, some implementations have been proposed to double the AE engine operating frequency than that of BC and provide a solution to address the EBCOT bottleneck [162]. However, these methods still fail to provide the required throughput. Some of the methods [170] [171] are one-symbol AE engines showing about 24% increase in execution time [170] as opposed to the conventional architecture. However, method in [171] does not include all the procedures of AE. According to [172], there can be situations where 12 CX-D pairs are generated at a single clock cycle from the BC module and thus multi-symbol AE processing is necessary. A multi-symbol AE module can reduce the input storage significantly and also increase the overall performance. If AE can process more than one symbol per clock cycle, the AE bottleneck can be reduced drastically and a reduction in memory can be achieved.

There are a few two-symbol architectures available that provide some good results, but a complete two-symbol per clock cycle solution still lacks. For instance, one of them uses the inverse multiple branch selection method [173]. In [174], a throughput of 52 Msymbols/sec is achieved at a cost of increased hardware and memory storage. Noikaew *et. al* [175] uses a prediction process to determine the upper bound and index values but the throughput is only 62 Msymbols/sec. It also does not provide the code update procedure for two-symbol processing. Parallel processing techniques have also been used to arrive at a two-symbol architecture in [176], but with constraints on the interval register. In this case, the two-symbol update is possible only if the value of the interval register is less than two.

After considering both the advantages and disadvantages of the existing architectures, we propose a new two-symbol architecture, which is capable of encoding two CX-D pairs in every clock cycle and overcomes the interval and code update issue, while also providing a higher throughput and an operating frequency of 100 MHz. The coding efficiency is not affected and the memory is also kept minimum. Furthermore, the critical path is observed to be 9.4 ns and hence the AE engine can operate at a higher frequency above 100 MHz. This factor is of high importance in hardware implementations and also provides room for future optimization. Our proposed architecture is fast and efficient in the sense of the interval and code update , byte output, renormalization, and flush procedures. The proposed architecture is able to eliminate the AE bottleneck in JPEG 2000 and also increases the performance of the EBCOT engine as a whole.

7.3 Arithmetic Encoding System Model

The arithmetic coder is based on the statistical binary arithmetic coding technique, also known as the MQ-coder. The bit-plane coder provides the CX and

D information to the AE stage for further processing. The AE stage executes in a sequential process, where a series of CX-D pairs are coded using context-based probability estimation. The D bit is either a logic 0 or 1. The CX bits provide significant information about a single bit and its neighbors. With each binary decision, the current probability interval is subdivided into two sub-intervals, and the code stream is modified (if necessary) so that it points to the base (the lower bound) of the probability sub-interval assigned to the symbol. Since the coding process involves the addition of binary fractions rather than the concatenation of integer codewords, the more probable binary decisions are always coded at the cost of less than one bit per decision. The MQ-coder is capable of producing at most two code bytes at once. A symbol can belong to one of two possible categories i.e., most probable symbol (MPS) and least probable symbol (LPS), based on the probability of their occurrence. The new interval is obtained from the sub-interval corresponding to the new symbol. AE can be described by the following classical equations

MPS coding:

$$\begin{aligned} C &= C + A \times Qe \\ A &= A - A \times Qe \end{aligned} \tag{7.1}$$

LPS coding:

$$\begin{aligned} C &= C \\ A &= A \times Qe \end{aligned} \tag{7.2}$$

where C is the base of the current interval, A is the length of the current interval, and Qe is the estimated probability (Qe).

To avoid complex multiplications, a simple trick is used to simplify the above equations. A is bounded to lie in the range of (0.75, 1.5). When A falls below the lower bound of the range, it is doubled until A returns to the range. This process is termed renormalization. Each time as A doubles,C needs to be doubled. Since A is of the order of unity [155], (7.2) and (7.3) can be simplified as MPS coding:

$$\begin{aligned} C &= C + Qe \\ A &= A - Qe \end{aligned} \tag{7.3}$$

LPS coding:

$$\begin{aligned} C &= C \\ A &= Qe \end{aligned} \tag{7.4}$$

The current interval is split in a recursive manner until all the symbols of a code-block are received from the bit-plane coder. The interval length division is

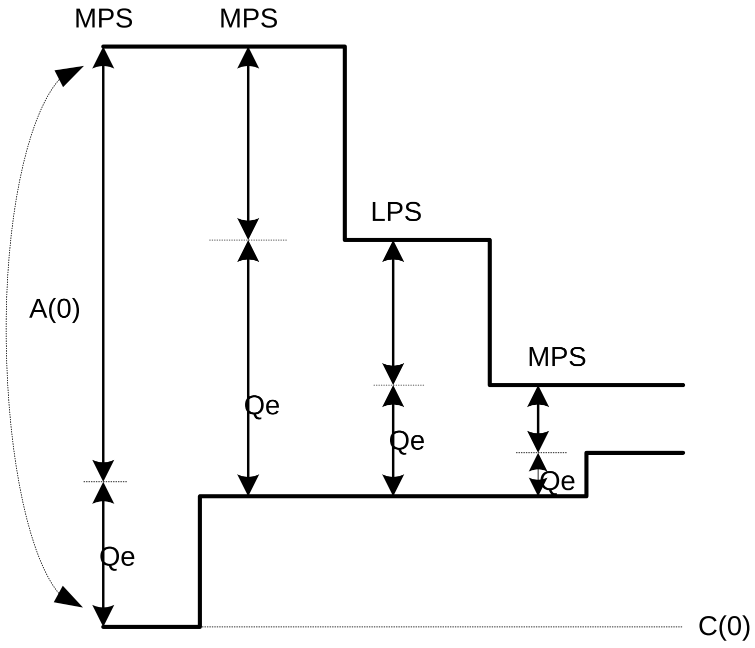


Figure 7.1: Interval length division.

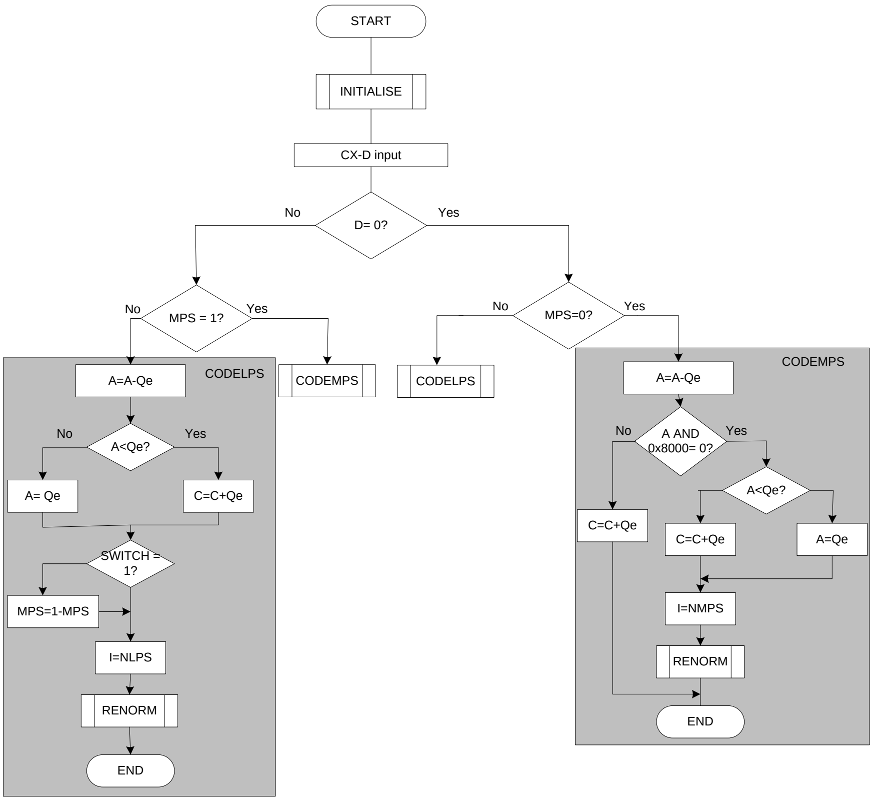
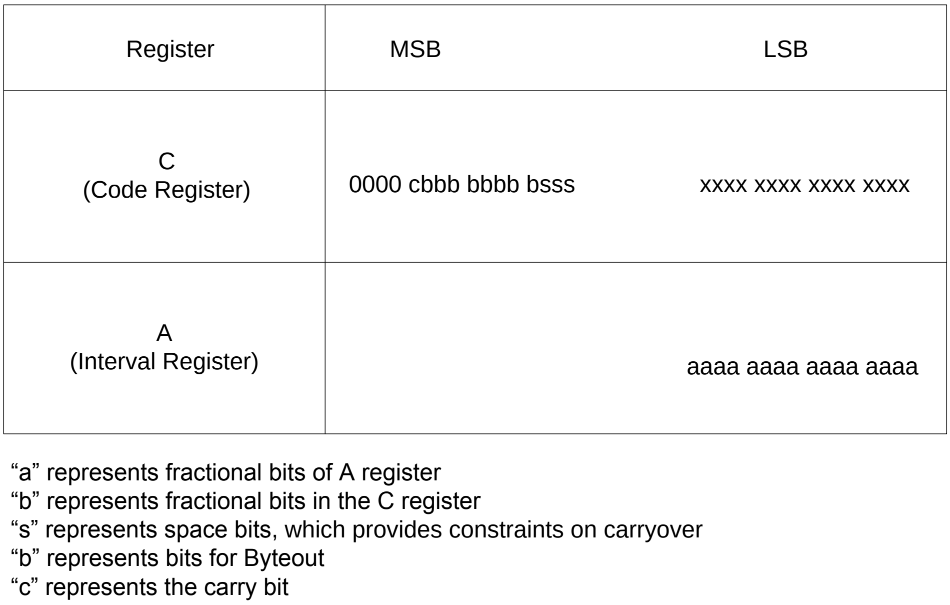


Figure 7.2: AE flowchart.

shown in Fig. 7.1 whereas the AE flowchart is illustrated in Fig. 7.2. JPEG 2000 uses 19 contexts for any given type of bits, and each of these contexts has an associated probability state that identifies the MPS and the index (I). The MPS and I point to a probability estimation table, which determines the Qe for the LPS, the next index values (NMPS, NLPS), and the probable symbol change of the MPS (SWITCH). The AE algorithm mainly deals with updating a set of registers based on the MPS and LPS. These registers are A, C, Ct and B. The structures of the registers A and C are depicted in Fig. 7.3 [177].



Register	MSB	LSB
C (Code Register)	0000 cbbb bbbb bsss	xxxx xxxx xxxx xxxx
A (Interval Register)		aaaa aaaa aaaa aaaa

"a" represents fractional bits of A register
"b" represents fractional bits in the C register
"s" represents space bits, which provides constraints on carryover
"b" represents bits for Byteout
"c" represents the carry bit

Figure 7.3: Structure of registers A and C.

Register A is a 16-bit interval register that contains the value of the current interval as required by AE, and register C is the code register containing the partial coded bits at every stage of encoding. Register A is initialized to 0x8000 to signal the beginning of the interval. Since the AE algorithm is implemented in fixed-point integer arithmetic, the initial value of A (0x8000) is equivalent to the decimal value of 0.75. Register C is of 28 bits, of which the lower 16 bits represent the lower bound of the interval and the upper 12 bits are used as a buffer for overflow [176].

Probability estimation Qe and the status of MPS are used to update registers A and C. Whenever the value of A falls below 0.75, the renormalization procedure is invoked and both registers are shifted left till A becomes greater than 0.75. Simultaneously, register Ct is decremented by the number of shifts occurred in these registers. The initial value of Ct is 0xC and B is 0x00. This procedure repeats continually until all the CX-D pairs of the code block are processed. During this process, whenever Ct becomes zero, the previous valid value in B, if any, is transferred to the output byte stream that forms the final encoded stream.

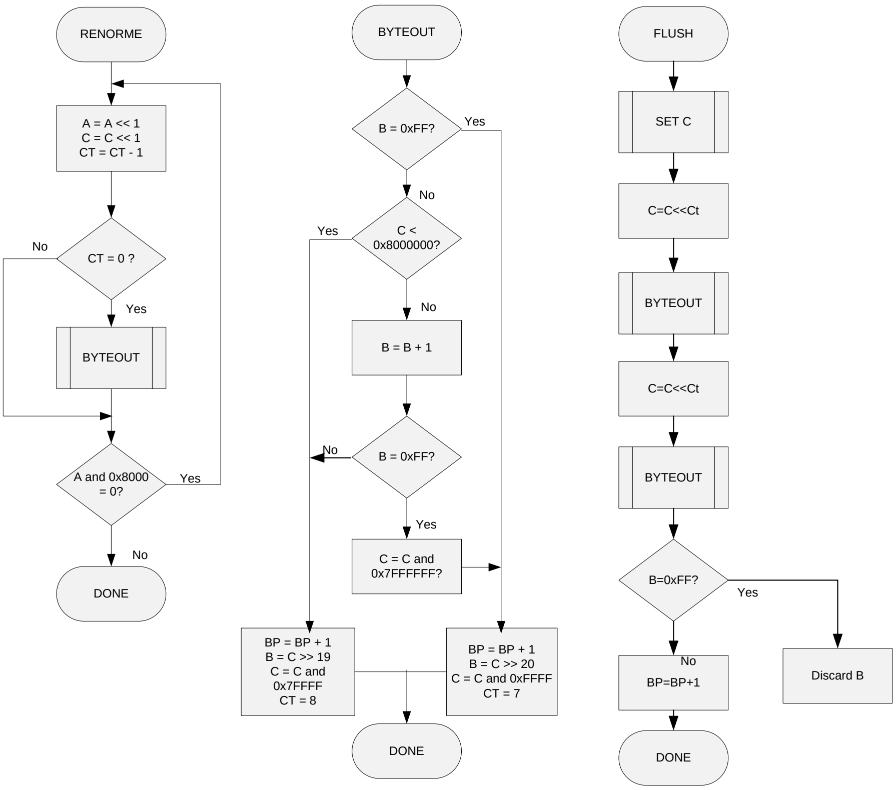


Figure 7.4: (a) Renormalization flowchart; (b) Byte-out flowchart; and (c) Flush procedure flowchart.

The byte-out procedure is then performed in parallel and register B is updated with the new value. Until the completion of the code block, A and C accumulate all the coding bits. In order to remove dependency and to provide error resilience in the bit stream, AE undergoes a termination process after every code block. This is done in a separate process dubbed flush. The renormalization, byte-out and flush flowcharts are illustrated in Fig. 7.4 [155].

Theoretically, the renormalization procedure executes a maximum of 15 times simultaneously, and hence the byte-out procedure can occur only twice at the same time. This means that at any given time we can have only 2 bytes generated at once. The byte-out procedure always outputs the previous generated bytes and stores the present bytes to output during the next cycle. Since the markers in the byte stream have a value of 0xFF, in order to distinguish them from legitimate code bytes, a bit-stuffing procedure is carried out in register B during which Ct is updated with a value of 0x7, since the stuffed bit takes up a single bit space.

7.4 Two-symbol Arithmetic Encoding Architecture

The block diagram of the proposed two-symbol AE architecture is depicted in Fig. 7.5. It processes two CX-D pairs every clock cycle. The architecture involves two main stages, namely, the interval update and code update stage. The other units include the probability estimation tables, index prediction, memory storage, and the AE controller. The following sections provide details on the two main stages.

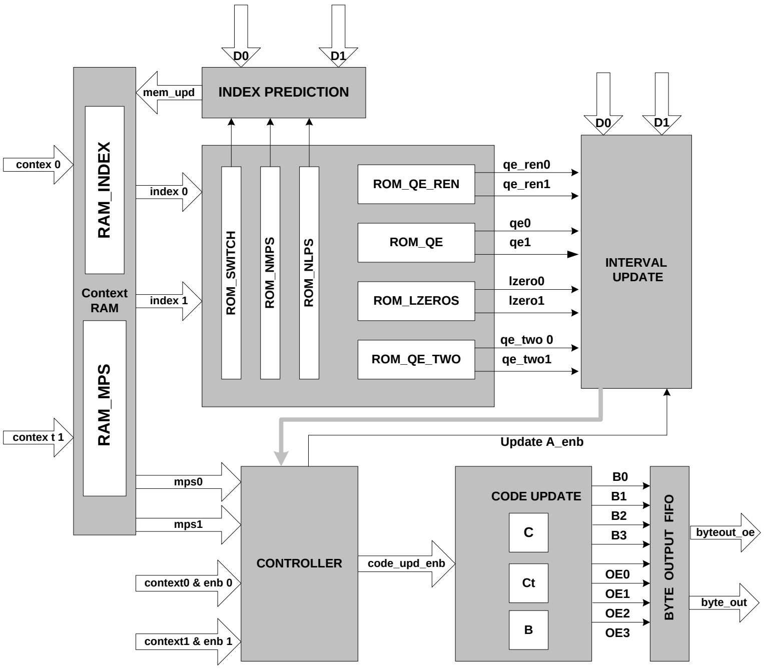


Figure 7.5: Block diagram of the proposed two-symbol AE architecture.

7.4.1 Interval Update Stage

The Interval Update stage as shown in Fig. 7.6 has the value of register A predicted beforehand. Since we process two symbols simultaneously, two register A predictions are performed in pipeline. Register A update mainly depends on the three MSB bits of the present register A, the left-shifted value of Qe, and the decision bit MPS or LPS. Register A can have three kinds of updated values, namely, 1) $A - Qe$ without renormalization; 2) $A - Qe$ with subsequent renormalization; and 3) Qe with renormalization only once at the end. Since A has to be greater than 0x8000, it can be renormalized twice at the most. This condition considers the type of updates A can have, the minimum value of A and the maximum value of Qe (0x5601). Hence, $A - Qe$ will always be greater than

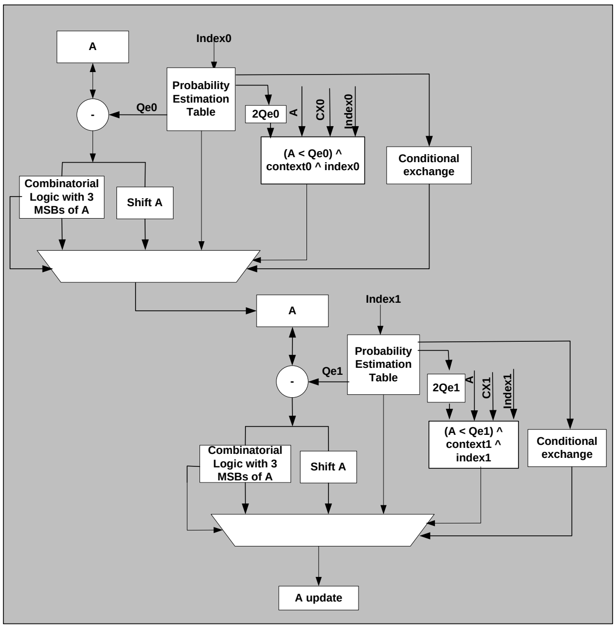

A
Index0
Probability Estimation Table
Qe0
-
2Qe0
A
CX0
Index0
(A < Qe0) ^ context0 ^ index0
Conditional exchange
Combinatorial Logic with 3 MSBs of A
Shift A
A
Index1
Probability Estimation Table
Qe1
-
2Qe1
A
CX1
Index1
(A < Qe1) ^ context1 ^ index1
Conditional exchange
Combinatorial Logic with 3 MSBs of A
Shift A
A update

Figure 7.6: Interval (A) Update procedure.

0x29FF, meaning there are two zeroes present at the MSB. Therefore, there can only be three ways of renormalization. When A is updated with Qe, the values are obtained from the probability estimation table. The renormalization of A will be with either one shift, two shifts or no renormalization at all. The prediction of A is carried out in two stages where the first stage updates A using the first CX-D pair information and the corresponding probability estimation tables. Once the intermediate A value becomes available, the net update of A is performed using the second CX-D pair and the second set of probability estimation tables. If the two contexts are the same, the updated index and MPS of the first symbol will be used as input for A updation of the second symbol.This whole process is carried out in pipeline with other AE stages, hence enabling two-symbol processing each time. A similar two-stage update A architecture is presented in [175]. Our architecture, however, is different due to the use of different conditions and short combinatorial paths, so that the A update is sped up. Another major difference is the way the A update module is embedded in the pipeline stages. This enables us to implement effective pipelining of the AE module and avoids any intermediate stalls in two-symbol processing.

7.4.2 Code Update Stage

Fig. 7.7 depicts the block diagram of the Code Update procedure, which includes updating registers C, Ct and B. Carry propagation and bit-stuffing are handled in the same module. The renormalization and byte-out procedures are also performed in parallel with the C update, reducing the critical path to a large extent.

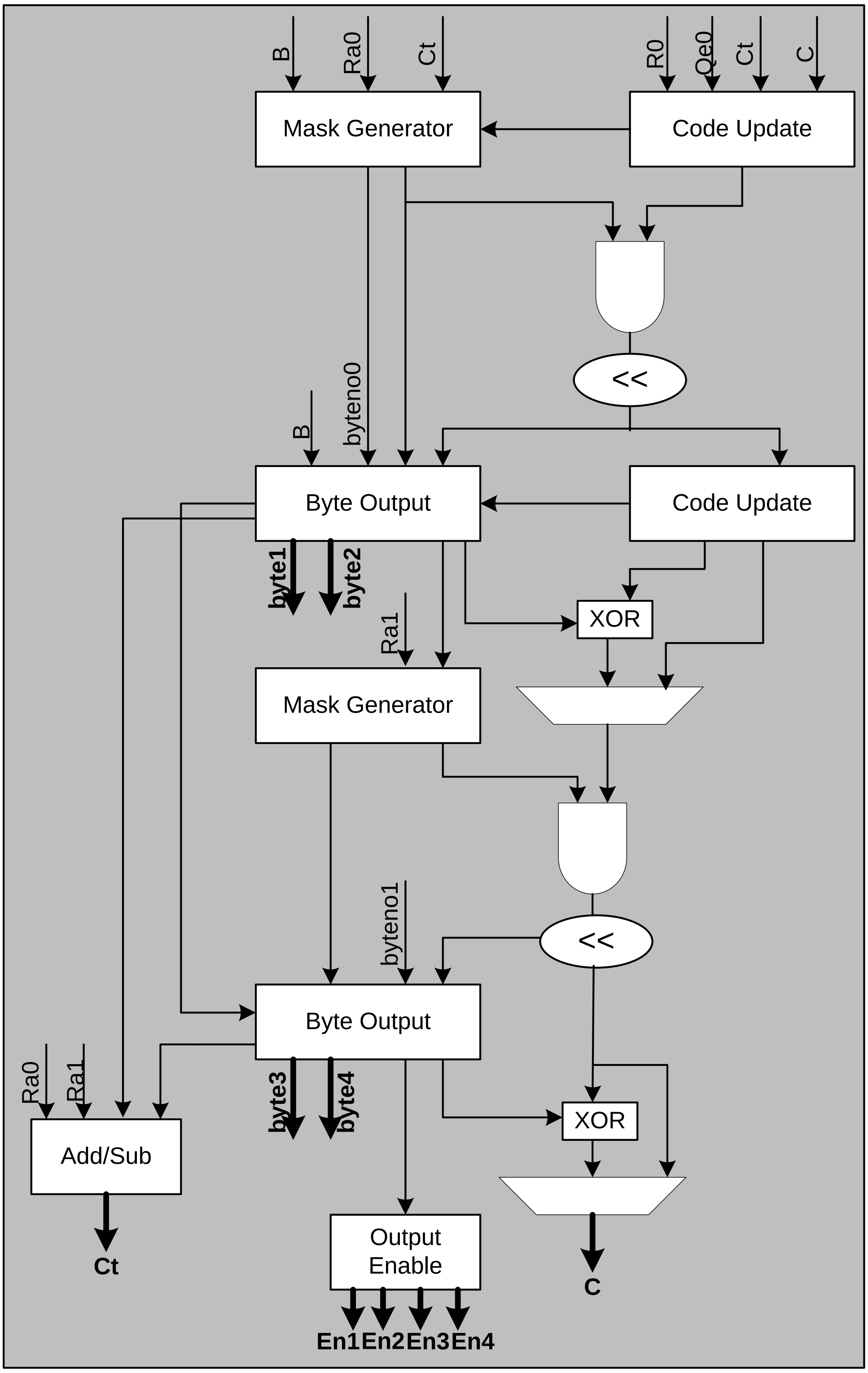


Figure 7.7: Code (C) Update procedure.

In the standard architecture, the renormalization and byte-out procedures execute sequentially and are achieved by serial shifters and lengthy conditional logic for generating output bytes. Having just a parallel architecture for register A update and a sequential procedure for register C update will be useless, since the delay in processing will be too long and also two-symbol processing will be limited by the register C update procedure. Hence, we perform the register C update and byte-out procedures in parallel. Renormalization is done in a single clock in comparison to the looping procedure in the standard architecture. Due to this, there can be a possibility of generating a 16-bit value at once and therefore have two register B values at the same time. This procedure is performed for two sets of contexts, and hence the same procedure of renormalization can generate another 16-bit output. In the worst scenario, there would be a maximum of four bytes generated at the same time, and therefore there is a need to have a method to output four bytes simultaneously. To achieve this, we use a mask generator to generate corresponding output enable signals. The byte-out procedure occurs whenever register Ct becomes zero. When this occurs, the byte already available in register B is outputted and the most significant byte of register C is moved to B. The mask generator generates the required mask, while the register C update module performs the required updates for the first symbol. The update is determined by R, which provides the information whether the shift amount required has to be a value from the leading zeroes (lzeroes) table or the value determined by the MSBs of $A - Qe$. The carry bits generated from the register C update module (Fig. 7.8)(a)) are used for the mask generation as shown in Fig. 7.8(b). If the value of B is 0xFE or 0xFF, a decision is made whether or not bit-stuffing is required, since these values correspond to byte stream markers. After encoding all the symbols of each code block, a flush procedure is performed as described in the standard. During flush, a maximum of 3 bytes can be generated simultaneously.

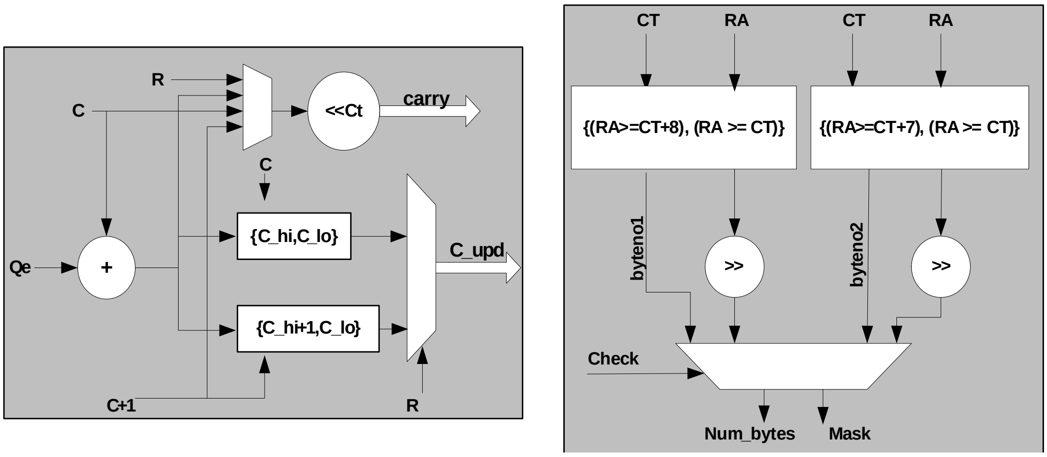


Figure 7.8: (a) Register C update module; (b) Mask Generation module.

7.4.3 Probability Estimation Table and State Update

In the JPEG 2000 standard [155], there are only 4 tables defined, i.e., the Qe table, NMPS table, NLPS table and SWITCH table. To facilitate two-symbol processing, we define three more tables, namely, the lzeroes table, renormalized Qe table and $2{*}Qe$ table. The lzeroes table records the count of zeroes in the MSB of the corresponding Qe value in the table. This is used during the renormalization of interval register A. The renormalized Qe table stores the renormalized result of the corresponding Qe value in the probability table. By using this table, some of the combinatorial logic is reduced and a value can be directly selected when Qe is to be updated as the new A. When $A \geq$ 0x8000, it is replaced with the corresponding value in the $2 * Qe$ table. The probability states are updated every time after the interval and code update occurs. These updated values are used when the next CX-D pair is processed.

7.4.4 Index Prediction

The index pointer is used to pick the right set of values for registers A and C update. Since two-symbol processing is implemented, the index corresponding to the second CX-D pair is predicted beforehand. This is done by using the previous MPS value and the index. A default index and MPS is available during the start of process as defined in the standard [155].

7.4.5 Critical Path Analysis

Critical path analysis is a very powerful approach for identifying bottlenecks in concurrent architectures. In conventional JPEG 2000 hardware implementation, critical paths are seen starting from bit-plane processing in BPC and goes upto the byte-out procedure of AE. Simple pipelining in BPC and AE can substantially shorten the critical paths, but this will incur the problem of having to incorporate a chain of storage elements, which can lead to a further increase in resource utilization. Hence, in our proposed AE architecture, we employ pipeline stages in a manner that the storage is kept minimal. We have also replaced some of the arithmetic operations with shift operations, wherever possible. For instance, two consecutive shifting operations required for C update contribute the critical path. If we decrease the number of consecutive shift operations in updating register C, faster extraction of the output bit-stream may be feasible, eventually resulting in a shorter critical path delay.

7.5 Combined System With CS

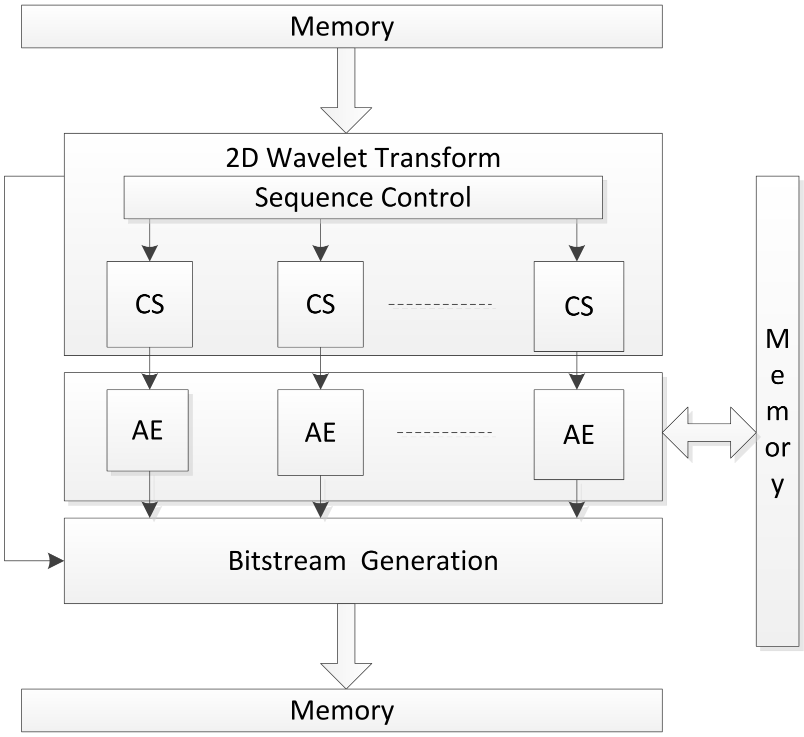


Figure 7.9: Top level block diagram of CS-based JPEG 2000 encoder.

Fig. 7.9 depicts the block diagram of a CS-based JPEG 2000 encoder system. The 2D wavelet transform block with sequence control and CS, is the same system that was proposed in Chapter 6. The proposed two-symbol AE is interfaced with the 2D wavelet transform block as shown in the figure. The sequence control block mainly performs the data control of the transformed coefficients and transfers them to AE via CS block, for sequential processing. Every block processing uses SRAMs, for storing the symbols temporarily.

7.6 Simulation Results

The proposed two-symbol architecture is implemented using the Verilog hardware definition language and synthesized on an Altera Stratix II FPGA. The hardware implementation cost is shown in Table 7.1. Since the proposed architecture has 4 pipeline stages, combinatorial logic is reduced and sequential logic is used wherever feasible. This helps in shortening the critical paths, which is advantageous in this design. Having a low critical path timing, i.e., 9.4 ns, shows that our circuit can operate at fairly high frequencies. The hardware utilization of the AE module and its control units is 1.2K ALUTs of Stratix II FPGA. The memory

Table 7.1: Hardware implementation cost.

ALUT	1267
Registers	1321
Pipeline stages	4
Throughput	212 Msymbols/sec
frequency	106.2 MHz
Critical path	9.4 ns

Table 7.2: Number of clock cycles for 512×512 image, code block size 64×64, lossless compression.

Image Name	Clock Cycles
Lena	839943
Peppers	916049
Baboon	989841
Jet	795816

used for this implementation is a FIFO for 16 CX-D pairs, where CX is of five bits and D is one bit. Hence, a memory unit of 32×8 bits is used.

The proposed AE module is evaluated using 10 images of various sizes ranging from 512×512 to 16384×16384. All the tested images are of full color (4:4:4). To test the functionality of the AE module, a complete JPEG 2000 system is constructed using Verilog and ported onto a FPGA. The design is optimized in such a way that the operating frequencies of every module in the system operate well above 100 MHz. This also ensures that there are no critical paths that affect the AE module. Multiple AE engines are not required to keep in pace with the throughput of the BC engine and we have a single operating frequency for the EBCOT engine. The frequency is kept at 100 MHz, although our AE engine can operate above this frequency. Some of the standard 512×512 test images with clock cycle consumption are summarized in Table 7.2 for which the AE engine was tested for throughput. We observe that the encoding time is strongly dependent on the number of symbols to be encoded. Our AE engine can encode 212 Msymbols/sec at 106.2 MHz with lossless coding and also processes two symbols for every clock cycle at all conditions of AE processing. There are no stall conditions encountered, since the bit-plane coder generates enough CX-D pairs that the AE engine can continuously process. Due to this, the memory requirement at the input of AE is drastically reduced.

It is observed that the AE engine processes two symbols for every clock cycle irrespective of the interval and code registers, unlike the constraints on A in [176]. Our proposed AE design is different when compared to that of [176] by the

Table 7.3: Frequency and throughput comparison among the proposed two-symbol architecture and other one-symbol and two-symbol architectures in the literature.

Type	Device	Frequency (MHz)	Throughput (Msymbol/s)	Hardware	Critical path (ns)
Two-symbol (Proposed)	Stratix II	106.2	212.4	1.2K ALUT 1321 registers (4.8K gates)	9.4
Conventional[1] (One-symbol)	Stratix II	42	42	4536 ALUT 6689 registers	17
One-symbol[4]	0.35um ASIC	150	150	7.2K gates	5.37
Two-symbol [5]	Stratix	88	22	8.5K LE	
One-symbol [7]	0.35um ASIC	180	150	13.6 K	
Two-symbol [9]	FPGA	26.29	52.58		
Two-symbol [10]	Spartan 3	125.68	62.84		
Two-symbol [11]	0.35um ASIC	90.9	180(cond)	7.7K gates	11
One-symbol [14]	Virtex II Pro	112			
One-symbol [15]	FPGA	55	54	152K gates	
One-symbol [16]	0.35um ASIC	200	200	6.9K gates	4.82
Two-symbol [17]	0.18um ASIC	200	-	18.7K gates	
One-symbol [19]	0.18um ASIC	100	-	56K gates	
One-symbol [20]	Virtex II Pro	120	-		
One-symbol [22]	0.18um ASIC	200	-	3.2K gates	
One-symbol [25]	ASIC	50	-	11K gates	10
Two-symbol [27]	LX80 Stratix	48.3	96.6	6974 Slices	

Table 7.4: Throughput per cycle

Technology	Technology	Throughput per cycle
Proposed	FPGA	2 symbols
Two-symbol [11]	ASIC	2 symbols
Two-symbol [17]	ASIC	1.2 symbol
Two-symbol [9]	FPGA	1.9 symbol
One-symbol [24]	FPGA	1 symbol

following: (i) we use two sets of pre-calculated tables, as well as a table for $2 * Qe$, which makes it possible to use a less number of shifters by comparison; (ii) the pipeline stages that incorporate A and C updates are specifically designed to achieve shorter critical paths; (iii) the mask that is used to output bytes does not use the present generated byte, but rather the MSB of the register C; (iv) as seen in Table 7.3, though our implementation is on a Stratix II FPGA, its throughput is comparable or even better than that of the design in [176], an ASIC-based design.

A comparison of performance, cost and the throughput with previously proposed methods is carried out. The comparison is shown in Table 7.3. The resource consumption mentioned in the table is only with respect to the AE module and not the complete EBCOT engine. The critical path in our implementation is observed in the C update module, during the flush procedure. The flush procedure information is not described in any of the available two-symbol methods due to intensive computation which reduces the throughput of the system. Since we consider this procedure as part of our AE implementation, our proposed method is more efficient than the existing methods.

Comparing the results in Table 7.3, it can be concluded that the throughput of our method more than doubles compared to that of conventional one-symbol methods, if operated at similar frequencies. However, the conventional method [155] cannot operate at high frequency due to the combinatorial paths in its implementation. Table 7.4 presents a comparison of throughput per cycle with respect to the technology used and type of architecture. It is observed that among the available FPGA implementations in the references, our design processes two symbols every clock cycle compared to others [174,178], though it is same as that of [176] which is an ASIC-based design.

7.7 Summary

A new two-symbol architecture for arithmetic coding in JPEG 2000 is proposed in this paper, which is able to encode two symbols every clock cycle. The processes for interval update, code update, index prediction, mask generation and efficient renormalization are described. The byte-out procedure is implemented to output four bytes at a time, so that the proposed AE engine is able to constantly maintain the processing of two symbols per cycle. It also keeps the critical paths minimal. This architecture is highly optimized for timing and cost. It operates at 106.2 MHz achieving upto 212 Msymbols/sec. The results show that our two-symbol architecture is fast and efficient. The performance of our two-symbol architecture doubles that of the conventional one-symbol methods, in terms of throughput and is about 50% faster than the existing two-symbol methods. The hardware utilization is minimal and hence the architecture is cost effective. The design is synthesized on an Altera Stratix II FPGA. This architecture may be improved to process multiple symbols and to further enhance the performance of the design.

JPEG 2000 is one of the most popular image compression standards offering significant performance advantages over previous image standards. The high computational complexity of the JPEG 2000 algorithms makes it necessary to employ methods that overcome the bottlenecks of the system and hence an efficient solution is imperative. One such crucial algorithm in JPEG 2000 is arithmetic coding and is completely based on bit level operations. In this paper, an efficient hardware implementation of arithmetic coding is proposed which employs efficient pipelining and parallel processing for intermediate blocks. The idea is to provide a two-symbol coding engine, which is efficient in terms of performance, memory and hardware. This architecture is implemented in the Verilog hardware definition language and synthesized using the Altera field programmable gate array. The only memory unit used in our design is a FIFO (first in, first out) of 256 bits to store the context-decision (CX-D) pairs at the input, which is negligible compared to existing arithmetic coding hardware designs. Our simulation and synthesis results demonstrate that the operating frequency of the proposed architecture is greater than 100 MHz and it achieves a throughput of 212 Msymbols/sec, doubling the throughput of conventional one-symbol implementations while enabling at least 50% throughput increase compared to existing two-symbol architectures.

Chapter 8

Conclusions

This dissertation mainly focusses on providing a compressive sensing based solution. A complex Hadamard matrix in combination with CoSaMP has been used to achieve the required goals. Furthermore, this work, though is based on CS, provides solution to two different applications. The first being 2D and 3D-MRI processing and the second one is natural image processing. At various stages, the proposed method have proven to be superior to the existing CS methods. In addition, low-complexity and energy-efficient hardware architectures are designed, to provide a flexibility of use in practical scenarios.

Specifically, in the first part, a MRI data acquisition and reconstruction system is designed based on CS principles. Firstly, a complex Hadamard matrix is used for data acquisition. A modified CoSaMP for MRI is presented and the system is verified for many real datasets. The system is further compared with an existing 3D-MRI system from literature, based on a phantom. The proposed system performs better than the existing one. The results are validated based on the signal-to-noise ratio.

Next, the already proposed system is optimized to enhance the performance and increase efficiency for 3D-MRI. This was necessary due to the high complexity and huge data processing requirements in 3D-MRI. In conjunction, an new matrix based on CHM is defined, termed as unitary CHM. This matrix satisfies restricted isometry property and is proved in this work.

Finally for MRI, a FPGA-based hardware architecture is proposed. This architecture is less complex and high performance compared to existing solutions available for MRI. From simulations, it is observed that the SNR results remain the same, while providing high throughput.

In the second part, the focus shifts to natural image processing. Here, CS techniques are applied to a JPEG 2000 encoder-decoder. The motivation behind combining CS with an image encoder-decoder, was to provide a system that drastically reduced the transform samples. This target is achieved by using CHM with the traditional JPEG 2000 wavelet transform, known as 9/7 lifting wavelet. The results after the use of CS show that, a high SNR valued reconstruction is possible with very low number of measurement samples. This proposed system is hardware-based and provides an energy-efficient solution.

Since AE is a serial processing block in the JPEG 2000 encoder, it is of utmost importance to have an efficient solution, especially when a high performance CS-based transform is conducted. Hence, a two-symbol arithmetic encoder for JPEG 2000 is developed to increase the overall encoder efficiency. Furthermore, this is integrated with the CS-based JPEG 2000 transformation to obtain an efficient encoder architecture.

This work is suitable for commercial implications provided that some hardware system related contingencies are resolved. This would imply that MRI scanners currently available in market would need to undergo changes to accommodate the compressive sensing based module. Compared to current MRI scanning time [148], the improvement expected is about 25% based on the simulation results. In saying that, the major roadblock would be the cost of changing the existing MRI scanners to suit CS techniques. In the case of JPEG 2000, the whole CS-based encoder/decoder is FPGA/ASIC-based and ready to be used commercially.

8.1 Future Work

This discussion concludes with some recommendations of possible future work, which are extensions of the problems considered in this thesis:

- **Further reduction in hardware system complexity:** Though the hardware architectures that are proposed in this thesis are less complex and energy-efficient, there is still room for hardware optimization. Once the target hardware (e.g., FPGA, ASIC) is chosen, and efficient pipelining, place and route will provide a better performance. The system can be further optimized based on the target devices and applications.
- **Application of CHM to general medical images:** Since the CS-based techniques are usually generic in nature, the proposed method is not bound to only MRI-images. There is a scope for using this method for any kind of medical image.
- **Optimizing the system for diffusion MRI and functional MRI:** Diffusion MRI allows mapping of the diffusion process of molecules, which

is mainly water, in tissues and in-vivo, and functional MRI measures the brain activity by detecting associated changes in blood flow. Having a CS system specifically aiming at these types of MRI, will provide pathways for having reduced cost and less computationally intensive MRI hardware. In turn, this can also increase the speed of MRI process, whose scanning time causes patient discomforts.

- **Two-symbol architecture for JPEG 2000 decoder:** Following in similar lines with the encoder, there is a possibility of having a two-symbol arithmetic decoding process. The JPEG 2000 entropy decoding has a huge dependency on the inverse transform, and having a CS-based reconstruction process is to be investigated. If a solution to this is arrived at, the JPEG 2000 would be a quite simple. This can be advantageous in various imaging devices and applications.

References

[1] Loni-ICBM. (2012, Apr.) International consortium of brain mapping. [Online]. Available: http://www.loni.ucla.edu/ICBM/

[2] L. Montefusco, D. Lazzaro, S. Papi, and C. Guerrini, “A fast compressed sensing approach to 3D MR image reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1064–1075, May 2011.

[3] M. Lustig, D. Donoho, and J. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, Dec 2007.

[4] K. Block, M. Uecker, and J. Frahm, “Undersampled radial MRI with multiple coils. iterative image reconstruction using a total variation constraint,” *Magnetic Resonance in Medicine*, vol. 57, no. 1, pp. 1086–1098, 2007.

[5] M. Lustig, J. Santos, D. Donoho, and J. Pauly, “k-t SPARSE: High frame rate dynamic MRI exploiting spatio-temporal sparsity,” in *Proc. of the International Society for Magnetic Resonance in Medicine*, Seattle, WA, 2006.

[6] S. Kim, K. Koh, M. Lustig, and S. Boyd, “An interior-point method for large-scale $l1$-regularized least squares,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[7] L. He, T. Chang, S. Osher, T. Fang, and P. Speier, “MR image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods,” UCLA, Los Angeles, USA, Tech. Rep., 2006.

[8] J. Ye, S. Tak, Y. Han, and H. Park, “Projection reconstruction MR imaging using FOCUSS,” *Magnetic Resonance in Medicine*, vol. 57, pp. 764–775, 2007.

[9] H. Jung, J. Ye, and E. Kim, “Improved k-t blast and k-t SENSE using FOCUSS,” *Physics in Medicine and Biology*, vol. 52, pp. 3201–3226, 2007.

[10] J. Jiang, W. Luk, and D. Rueckert, “FPGA-based computation of freeform deformations in medical image registration,” in *Proc. of Field-Programmable Technology*, 2003, pp. 234–241.

[11] J. Trzasko and A. Manduca, "Highly undesrampled magnetic resonance image reconstruction via homotopic l_0-minimization," *IEEE Transactions on Medical Imaging*, vol. 28, no. 1, pp. 106–121, Jan 2009.

[12] M. Andrecut, "Fast GPU implementation of sparse signal recovery from random projections," *Engineering Letters*, vol. 17, no. 3, pp. 151–158, 2009.

[13] N. Gac, S. Mancini, M. Desvignes, and D. Houzet, "High speed 3-D tomography on CPU, GPU, and FPGA," *EURASIP Journal on Embedded Systems*, vol. 2008, pp. 1–12, 2008.

[14] S. Coric, M. Leeser, E. Miller, and M. Trepanier, "Parallel-beam backprojection: An fpga implementation optimized for medical imaging," in *Proc. of International Symposium on Field-Programmable Gate Arrays*, 2002, pp. 217–226.

[15] J. Li, C. Papachristou, and R. Shekhar, "An FPGA-based computing platform for real-time 3-D medical imaging and its application to conebeam CT reconstruction," *Journal of Imaging Science and Technology*, vol. 49, pp. 237–245, 2005.

[16] I. Daubechies, "Ten lectures on wavelets," *SIAM*, 1992.

[17] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr 2006.

[18] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–120, July 2007.

[19] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[20] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[21] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[22] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[23] A. M. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[24] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[25] S. Foucart, “A note on guaranteed sparse recovery via l_p-minimization,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 97–103, Aug. 2009.

[26] S. Foucart and R. Gribonval, “Real versus complex null space properties for sparse vector recovery,” *Comptes Rendus de l’Acadmie des Sciences Paris*, vol. 348, pp. 863–865, 2010.

[27] A. Cohen, W. Dahmen, and R. A. DeVore, “Compressed sensing and best k-term approximation,” *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, Jan. 2009.

[28] D. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via $l1$ minimization,” *Proc. of Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar 2003.

[29] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, “Compressed sensing with coherent and redundant dictionaries,” *Candes, E., and Eldar, Y., and Needell, D., and Randall, P.*, 2010.

[30] L. Welch, “Lower bounds on the maximum cross correlation of signals (corresp.),” *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 397–399, may 1974.

[31] T. Strohmer and R. Heath, “Grassmanian frames with applications to coding and communication,” *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, Nov. 2003.

[32] S. A. Gersgorin, “U ber die abgrenzung der eigenwerte einer matrix,” *Izv. Akad. Nauk SSSR Ser. Fiz.-Mat.*, vol. 6, pp. 749–754, 1931.

[33] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[34] R. Gribonval and M. Nielsen, “Sparse representations in unions of bases,” *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[35] M. Herman and T. Strohmer, “General deviants: An analysis of perturbations in compressed sensing,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 342–349, Apr. 2010.

[36] Y. Chi, L. L. Scharf, A. Pezeshki, and R. Calderbank, “Sensitivity to basis mismatch in compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2182–2195, May 2011.

[37] E. J. Candes, “Compressive sampling,” in *Proc. of the International Congress of Mathematicians*, vol. 3, Madrid, Spain, 2006, pp. 1433–1452.

[38] S. Aeron, V. Saligrama, and M. Zhao, “Information theoretic bounds for compressed sensing,” *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5111–5130, 2010.

[39] Z. Ben-Haim, T. Michaeli, and Y. Eldar, “Performance bounds and design criteria for estimating finite rate of innovation signals,” *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 4993–5015, 2012.

[40] E. Arias-Castro and Y. Eldar, “Noise folding in compressed sensing,” *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 478–481, 2011.

[41] T. Cai, X. Guangwu, and J. Zhang, “On recovery of sparse signals via l_1 minimization,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3388–3397, july 2009.

[42] R. A. DeVore, “Deterministic constructions of compressed sensing matrices,” *Journal of Complexity*, vol. 23, no. 4-6, pp. 918–925, Aug. 2007.

[43] D. Donoho, “For most large underdetermined systems of linear equations, the minimal 1_1-norm solution is also the sparsest solution,” *Communication on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[44] E. Candes and Y. Plan, “Near-ideal model selection by 1_1 minimization,” *Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, Oct. 2009.

[45] D. Donoho and J. Tanner, “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing,” *Philosophical Transactions of the Royal Society A*, vol. 367, no. 1906, pp. 4273–4293, Nov. 2009.

[46] C. Dossal, G. Peyre, and J. Fadili, “A numerical exploration of compressed sampling recovery,” *Linear Algebra and its Applications*, vol. 432, no. 7, pp. 1663–1679, Mar. 2010.

[47] S. Boyd and L. Vanderberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.

[48] J. Tropp and S. Wright, “Computational methods for sparse solution of linear inverse problems,” *Proceeedings of IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[49] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, Sep. 2006.

[50] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[51] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, Dec. 2008.

[52] Y. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. of the Conference Record of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1993.

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

[54] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[55] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, Nov. 2008.

[56] M. A. Neifeld and J. Ke, "Optical architectures for compressive imaging," *Applied Optics*, vol. 46, no. 22, pp. 5293–5303, July 200.

[57] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Apr 2007.

[58] M. Lustig, D. Donoho, and J. Santos, "Compressed sensing MRI," *IEEE Transactions on Signal Processing*, vol. 25, no. 2, pp. 72–82, Mar 2008.

[59] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[60] S. Gazit, A. Szameit, Y. C. Eldar, and M. Segev, "Super-resolution and reconstruction of sparse sub-wavelength images," *Optics Express*, vol. 17, pp. 23 920–23 946, 2009.

[61] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communication on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, Aug 2006.

[62] 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 Processing Magazine*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[63] R. Coifman, F. Geshwind, and Y. Meyer, “Noiselets,” *Applied and Computational Harmonic Analysis*, vol. 10, pp. 27–44, Mar. 2001.

[64] W. U. Bajwa, A. M. Sayeed, and R. Nowak, “A restricted isometry property for structurally-subsampled unitary matrices,” in *Proc. of 7th Annual Allerton Conference on Communication, Control, and Computing*, Jan 2010, pp. 1005–1012.

[65] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, “Beyond nyquist: Efficient sampling of sparse bandlimited signals,” *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, Jan. 2010.

[66] J. D. Haupt, W. U. Bajwa, G. Raz, and R. Nowak, “Toeplitz compressed sensing matrices with applications to sparse channel estimation,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5862–5875, Jun. 2010.

[67] H. Rauhut, “Compressive sensing and structured random matrices,” in *Proc. of the International Congress of Mathematicians, Radon Series Comp. Appl. Math*, vol. 9, De Gruyter, 2010, pp. 1–92.

[68] J. Johnson, Y. ZainWadghiri, and D. Turnbull, “2D multislice and 3D MRI sequences are often equally sensitive,” *Magnetic Resonance Imaging*, vol. 41, pp. 824–828, Oct. 1999.

[69] M. . A. Bernstein, F. . K. King, and X. J. Zhou, *Handbook of MRI pulse sequences.* MA: Elsevier Academic Press, 2004.

[70] G. H. Glover and J. M. Pauly, “Projection reconstruction technique for reduction of motion effects in MRI,” *Magnetic Resonance in Medicine*, vol. 28, pp. 275–289, 1992.

[71] K. Scheffler and J. Hennig, “Reduced circular field-of-view imaging,” *Magnetic Resonance in Medicine*, vol. 40, no. 3, pp. 474–480, 1998.

[72] A. V. Barger, W. F. Bloch, Y. Toropov, T. M. Grist, and C. A. Mistretta, “Time-resolved contrast-enhanced imaging with isotropic resolution and broad coverage using an undersampled 3D projection trajectory,” *Magnetic Resonance in Medicine*, vol. 48, no. 2, pp. 297–305, 2002.

[73] D. C. Peters, F. R. Korosec, T. M. Grist, W. F. Block, J. E. Holden, K. K. Vigen, and C. A. Mistretta, “Undersampled projection reconstruction applied to MR angiography,” *Magnetic Resonance in Medicine*, vol. 43, no. 1, pp. 91–101, 2000.

[74] C. H. Meyer, B. S. Hu, D. G. Nishimura, and A. Macovski, "Fast spiral coronary artery imaging," *Magnetic Resonance in Medicine*, vol. 28, no. 2, pp. 202–213, 1992.

[75] E. M. Haacke, R. W. Brown, M. R. Thompson, , and R. Venkatesan, *Magnetic Resonance Imaging: Physical Principles and Sequence Design.* New York, 1st edition: Wiley-Liss, 1999.

[76] J. I. Jackson, C. H. Meyer, D. G. Nishimura, and A. Macovski, "Selection of a convolution function for fourier inversion using gridding," *IEEE Transactions on Medical Imaging*, vol. 10, no. 3, pp. 473–478, 1991.

[77] C. Cunningham, G. Wright, and M. Wood, "High-order multiband encoding in the heart," *Magnetic Resonance in Medicine*, vol. 48, pp. 689–698, 2002.

[78] J. D. Healy and J. Weaver, "Two applications of wavelet transforms in magnetic resonance imaging," *IEEE Transactions on Information Techonology*, vol. 38, no. 2, pp. 840–860, 1992.

[79] C. Oh, P. H.W., and Z. Cho, "Line-integral projection reconstruction (LPR) with slice encoding techniques: Multislice regional imaging in NMR tomography," *IEEE Transactions on Medical Imaging*, no. 3, pp. 170–178, 1984.

[80] D. Mitsouras, W. Hoge, F. Rybicki, W. Kyriakos, A. Edelman, and G. Zientara, "Non-fourier encode parallel MRI using multiple receiver coils," *Magnetic Resonance Imaging*, vol. 52, no. 2, pp. 321–328, 2004.

[81] W. Hinshaw and A. Lent, "An introduction to NMR imaging: From the Bloch equation to the imaging equation," 1983, pp. 338–350.

[82] L. Panych, G. Zientara, and F. Jolesz, "MR image encoding by spatially selective RF excitation: An analysis using linear response models," *Int Journal Imaging Systems and Technology*, vol. 30, no. 10, pp. 143–150, 1999.

[83] L. Panych, L. Zhao, F. Jolesz, and R. Mulkern, "Dynamic imaging with multiple resolutions along phase-encode and slice-select dimensions," *Magnetic Resonance in Medicine*, vol. 45, no. 6, pp. 940–947, 2001.

[84] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," in *Proc. of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Aug 2009, pp. 262–265.

[85] N. Vaswani, "LS-CS-Residual (LS-CS): Compressive sensing on least squares residual," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4108–4120, Aug 2010.

[86] S. Ravishankar and Y. Bresler, "Adaptive sampling design for compressed sensing MRI," in *Proc. of 33rd Annual Conference of IEEE Engineering in Medicine and Biology Society*, Dec 2011, pp. 3751–3753.

[87] Y. Kim, M. Nadar, and A. Bilgin, "Dynamic compressive magnetic resonance imaging using a Gaussian scale mixtures model," in *Proc. of IEEE International conference on Image processing*, Dec 2011, pp. 2293–2296.

[88] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magnetic Resonance in Medicine*, vol. 59, no. 2, pp. 365–373, Feb 2008.

[89] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye, "k-t FOCUSS: a general compressed sensing framework for high resolution dynamic MRI," *Magnetic Resonance in Medicine*, vol. 61, no. 1, pp. 103–116, Jan 2009.

[90] J. Jim and T. Tao, "Dynamic MRI with compressed sensing imaging using temporal correlations," in *Proc. of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Jun 2008, pp. 1613–1616.

[91] M. Guerquin-Kern, M. Haberlin, K. P. Pruessmann, and M. Unser, "A fast wavelet-based recontruction method for magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 30, no. 9, pp. 1649–1660, Sep. 2011.

[92] C. Haider, H. Hu, N. Campeau, J. Huston, and S. Riederer, "3D high temporal and spatial resolution contrast-enhanced MR angiography of the whole brain," *Magnetic Resonance in Medicine*, vol. 60, no. 3, pp. 749–760, Sept 2008.

[93] C. Qiu, W. Lu, and N. Vaswani, "Real-time dynamic MR image recontruction using Kalman filtered compressed sensing," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing.*, May 2009, pp. 393–396.

[94] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[95] M. Rudelson and R. Vershyinn, "On sparse reconstruction from Fourier and Gaussian measurements," *Communication on Pure and Applied Mathematics*, vol. 61, no. 8, pp. 1025–1045, Aug 2008.

[96] D. Donoho and J. Tanner, "Exponential bounds implying construction of compressed sensing matrices, error-correcting codes, and neighborly polytopes by random sampling," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 2002–2016, Apr 2010.

[97] J. Haldar, D. Hernando, and Z. Liang, “Compressed-sensing MRI with random encoding,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 4, pp. 893–903, 2011.

[98] F. Sebert, Y. M. Zou, and L. Ying, “Compressed sensing MRI with random B1 field,” in *Proc. of International Society of Magnetic Resonance in Medicine Scientific Meeting*, 2008, pp. 31–51.

[99] J. Tropp and A. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[100] V. Stankovic, L. Stankovic, and S. Cheng, “Compressive video sampling,” in *Proc. of Eusipco*, 2008.

[101] F. Yang, S. Wang, and C. Deng, “Compressive sensing of image reconstruction using multi-wavelet transforms,” in *Proc. of International Conference on Information Systems*, Oct.

[102] G. Zhang, S. Jiao, X. Xu, and L. Wang, “Compressed sensing and reconstruction with bernoulli matrices,” in *Proc. of International Conference on Information and Automation*, 2010, pp. 455–460.

[103] X. Xiaochun, Lixin, L. G., Zhenhui, and L. Zhaoshen, “Compressive sensing of image reconstruction using multi-wavelet transforms,” in *Proc. of the IEEE Youth Conference on Information, Computing and Telecommunication*, 2009, pp. 114–117.

[104] J. Tropp and A. Gilbert, “Signal recovery from partial information via orthogonal matching pursuit, information theory,” *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, 2007.

[105] M. Figueiredo, R. Nowak, and S. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[106] Y. Kim, S. Narayanan, and K. Nayak, “Accelerated three-dimensional upper airway MRI using compressed sensing,” *Magnetic Resonance in Medicine*, vol. 61, pp. 434–440, 2009.

[107] J. Chen, J. Cong, L. Vese, J. Villasenor, M. Yan, and Y. Zou, “A hybrid architecture for compressive sensing 3-D CT reconstruction,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 616–625, 2012.

[108] D. Kim, J. Trzasko, M. Smelyanskiy, C. Haider, P. Dubey, and A. Manduca, “High-performance 3D compressive sensing MRI reconstruction using

many-core architectures," *International Journal of Biomedical Imaging*, vol. 2011, pp. 1–11, 2011.

[109] A. Van Den Bos, "Complex gradient and hessian," *IEE Proceedings of Vision, Image and Signal Processing*, vol. 141, no. 6, pp. 380–383, 1994.

[110] S. Rahardja and B. Falkowski, "Digital signal processing with complex hadamard transform," in *Proc. of International Conference on Signal Processing*, 1998, pp. 533–536.

[111] A. Aung, B. P. Ng, and S. Rahardja, "Sequency-ordered complex Hadamard transform: properties, computational complexity and applications," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3562–3571, Aug 2008.

[112] B. Adcock, C. Hansen, and B. Roman, "Breaking the coherence barrier: A new theoy for compressed sensing," *Numerical Analysis*, 2013.

[113] L. Carin, L. Dehong, and G. Bin, "Coherence, compressive sensing, and random sensor arrays," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 4, pp. 28–39, Aug 2011.

[114] G. Lorentz, M. Golitschek, and Y. Makovoz, *Constructive Approximation: Advanced Problems.* Springer-Berlin, 1996.

[115] V. I. Levenstein, "Bounds for packings of metric spaces and some of their applications," *Problemy Kibernet*, vol. 40, pp. 43–110, 1983.

[116] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[117] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[118] W. Li and J. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 927–939, 2007.

[119] J. A. Tropp, M. B. Wakin, M. F. Duarte, D. Baron, and R. G. Baraniuk, "Random filters for compressive sampling and reconstruction," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, pp. 872–875.

[120] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, "Uniform uncertainty principle for bernoulli and subgaussian ensembles," *Preprint*, Aug. 2006.

[121] M. Ledoux and M. Talagrand, *Probability in Banach Spaces.* New York: Springer-Verlag, 1991.

[122] P. Dillinger, J. F. Vogelbruch, J. Leinen, S. Suslov, R. Patzak, H. Winkler, and K. Schwan, "FPGA base real-time image segmentation for medical systems and data processing," in *Proc. of Real Time Conference*, 2005, pp. 161–165.

[123] A. Khamene, R. Chisu, W. Wein, N. Navab, and F. Sauer, "A novel projection based approach for medical image registration," in *Proc. of Biomedical Image Registration*, 2006, pages =.

[124] A. E. Lefohn, J. E. Cates, and R. T. Whitaker, "Interactive, GPU-based level sets for 3D segmentation," in *Proc. of Medical Image Computing and Computer-Assisted Intervention*, 2003, pages =.

[125] S. Che, J. Li, J. Sheaffer, K. Skadron, and J. Lach, "Accelerating compute-intensive applications with GPUs and FPGAs," in *Proc. of Symposium on Application Specific Processors*, 2008, pp. 101–107.

[126] P. Maechler, P. Greisen, N. Felber, and A. Burg, "Matching pursuit: Evaluation and implementation for LTE channel estimation," in *Proc. of IEEE International Symposium on Circuits and Systems*, Paris, France, June 2010, pp. 589–592.

[127] M. Mishali, R. Hilgendrouf, E. Shoshan, I. Rivkin, and Y. Eldar, "Generic sensing hardware and real-time reconstruction for structured analog signals," in *Proc. of IEEE International Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, May 2011, pp. 1748–1751.

[128] A. Septimus and R. Steinberg, "Compressive sampling hardware reconstruction," in *Proc. of IEEE International Symposium on Circuits and Systems*, Paris, France, June 2010, pp. 3316–3319.

[129] H. Scherl, B. Keck, M. Kowarschik, and J. Hornegger, "Fast GPUbased CT reconstruction using the common unified device architecture (CUDA)," in *Proc. of IEEE Nuclear Science Symposium Conference Record*, vol. 6, 2007.

[130] J. Xu, N. Subramanian, A. Alessio, and S. Hauck, "Impulse C vs. VHDL for accelerating tomographic reconstruction," in *Proc. of 18th IEEE Annual International Symposium on Field-Programmable Custom Computational Machines*, 2010, pp. 171–174.

[131] B. Keck, H. Hofmann, H. Scherl, M. Kowarschik, and J. Hornegger, "GPU-accelerated SART reconstruction using the CUDA programming environment," in *Proc. of SPIE*, 2009.

[132] J. Chen, M. Yan, L. A. Vese, J. Villasenor, A. Bui, and J. Cong, "EM+TV for reconstruction of cone-beam CT with curved detectors using GPU," in *Proc. of Interentional Meeting Fully Three-Dimensional Image Reconstruct. Radiol. Nucl. Med.*, 2011, pp. 363–366.

[133] F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Transactions on Nuclear Science*, vol. 52, no. 3, pp. 654–663, 2005.

[134] D. Stsepankou, K. Kommesser, J. Hesser, and R. Manner, "Real-time 3-D cone beam reconstruction," in *Proc. of Nucl. Sci. Symp. Conf. Rec.*, 2004, pp. 3648–3652.

[135] S. Christel, M. Vignesh, and A. Kandaswamy, "An efficient FPGA implementation of MRI image filtering and tumour characterization using Xilinx system generator," *International Journal of VLSI design and Communication Systems*, vol. 2, no. 4, pp. 95–109, 2011.

[136] M. Bin Othman, N. Abdullah, and N. Bin Ahmad Rusli, "An overview of MRI brain classification using FPGA implementation," in *Proc. of IEEE Symposium on Industrial Electronics Applications (ISIEA)*, 2010, pp. 623–628.

[137] J. Stanislaus and T. Mohsenin, "Low-complexity FPGA implementation of compressive sensing reconstruction," in *Proc. of International Conference on Computing, Networking and Communications, Multimedia Computing and Communications Symposium*, 2013.

[138] M. Karkooti, J. Cavallaro, and C. Dick, "FPGA implementation of matrix inversion using QRD-RLS algorithm," in *Proc. of Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2005, pp. 1625–1629.

[139] A. Maltsev, V. Pestretsov, R. Maslennikov, and A. Khoryaev, "Triangular systolic array with reduced latency for QR-decomposition of complex matrices," in *Proc. of IEEE International Symposium on Circuits and Systems*, Island of Kos, Greece, May 2006, pp. 385–388.

[140] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," 1998, pp. 191–200.

[141] N. Kumar and W. Xiang, "An efficient low-complexity FPGA-based architecture for 3D-MRI proceessing," *ACM Transactions on Embedded Compution Systems*, To be submitted.

[142] *Stratix IV Device Handbook, Volume 1*, Handbook, Altera Corporation, 2011.

[143] *Virtex-5 FPGA User Guide, Volume 5.4*, User guide, Xilinx, 2012.

[144] J. Stanislaus and T. Mohsenin, "High performance compressive sensing reconstruction hardware with QRD process," in *Proc. of IEEE International Symposium on Circuits And Systems*, Seoul, Korea, May 2012.

[145] Y. Chen and X. Zhang, "High-speed architecture for image reconstruction based on compressive sensing," in *Proc. of IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, USA, 2010.

[146] A. Borghi, J. Darbon, A. Peyronett, T. Chan, and S. Osher, "Compressive sensing algorithm for parallel many-core architectures," UCLA, Los Angeles, USA, Tech. Rep., Sept. 2008.

[147] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications.* Cambridge University Press, 2012.

[148] Queensland XRay.

[149] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive image sampling with side information," in *Proc. of the IEEE International Conference on Image Processing*, 2009, pp. 3037–3040.

[150] E. J. Candes and J. Romberg, "Practical signal recovery from random projections," in *preprint*, 2005.

[151] L. Gan, "Block compressed sensing of natural images," in *Proc. of 15th International Conference on Digital Signal Processing*, 2007, pp. 403–408.

[152] S. She, Z. Luo, Y. Zhu, H. Zou, and Y. Chen, "Spatially adaptive image reconstruction via compressive sensing," in *Proc. of the 7th Asian Control Conference*, 2009, pp. 1570–1575.

[153] C. Deng, W. Lin, B. Lee, and C. Lau, "Robust image coding based upon compressive sensing," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 278–290, 2012.

[154] A. A.S., Z. P.B., and M. Moniri, "Stereo image representation using compressive sensing," in *Proc. of 3DTV Conference*, 2011, pp. 1–4.

[155] *JPEG2000 Part 1 020719*, ISO/IEC JTC1/SC29/WG1 N1636R Final Publication Draft, July 2002.

[156] T. Do, L. Gan, N. Nguyen, and T. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, 2012.

[157] X. Fang and H. George, "On the worst-case complexity of integer Gaussian elimination," *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pp. 28–31, 1997.

[158] R. Giryes and M. Elad, "RIP-based near-oracle performance guarantees for SP,CoSaMP and IHT," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1465–1468, 2012.

[159] J. Cooley and J. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathemetics Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[160] T. Do, T. Tran, and L. Gan, “Fast compressive sampling with structurally random matrices,” in *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 3369–3372.

[161] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamenetals, Standard and Practice.* Boston: Kluwer, 2002.

[162] B. Min, S. Yoon, J. Ra, and D. S. Park, “Enhanced renormalization algorithm in MQ-coder of JPEG2000,” in *Proc. of International Symposium on Information Technology Convergence*, 2007, pp. 213–216.

[163] L. Yijun and B. Magdy, “A three-level parallel high-speed low-power architecture for EBCOT of JPEG 2000,” *IEEE Transcations on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1153–1163, 2006.

[164] H. Damecharla, K. Varma, J. Carletta, and A. Bell, “FPGA implementation of a parallel EBCOT tier-1 encoder that preserves coding efficiency,” in *Proc. of GLSVLSI*, 2006, pp. 266–271.

[165] J. S. Chiang, C. H. Chang, Y. S. Lin, C. Y. Hsieh, and C. H. Hsia, “High-speed EBCOT with dual context-modelling coding architecture for JPEG2000,” in *Proc. of International Symposium on Circuits and Systems*, 2004, pp. 610–613.

[166] J. Chiang, C. Chang, C. Hsieh, and C. Hsia, “High efficiency EBCOT with parallel coding architecture for JPEG2000,” *IEEE Transcations on Circuits and Systems for Video Technology*, pp. 1–14, 2006.

[167] L. F. Chen, T. L. Huang, T. M. Chou, and Y. K. Lai, “Analysis and architecture design for memory efficient parallel embedded block coding architecture in JPEG 2000,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. 964–967.

[168] T. Saidani, M. Atri, and R. Tourki, “Implementation of JPEG 2000 MQ-coder,” in *Proc. of International Conference on Design and technology of Integrated Systems in Nanoscale Era*, 2008, pp. 1–4.

[169] I. F. Nesamani and C. Vasanthanayaki, “Implemenation of simplified architecture of JPEG 2000 MQ coder,” in *Proc. of International Conference on Control, Automation, Communication and Energy Conservation*, 2009, pp. 1–6.

[170] Y. Z. Zhang, C. Xu, W. T. Wang, and L. B. Chen, "Performance analysis and architecture design for EBCOT encoder in JPEG2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1336–1347, 2007.

[171] a. C. W. H. Ong, K. K., Tseng, Y. C., Y. S. Lee, and C. Y. Lee, "A high throughput context-based adaptive codec for JPEG2000," in *Proc. of the IEEE International Symposium on Circuits and Systems*, 2002, pp. 133–136.

[172] S. Rathi and Z. Wang, "FPGA implementation of a parallel EBCOT tier-1 encoder that preserves coding efficiency," in *Proc. of the IEEE Workshop on Signal Processing Systems*, 2007, pp. 595–599.

[173] P. Grzegorz, "A high-performance architecture for embedded block coding in JPEG 2000," *IEEE Transcations on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1182–1191, 2005.

[174] M. Dyer, D. Taubman, and S. Nooshabadi, "Improved throughput arithmetic coder for JPEG2000," in *Proc. of the International Conference on Image Processing*, 2004, pp. 2817–2820.

[175] N. Noikaew and O. Chitsobhuk, "Dual symbol processing for MQ arithmetic coder in JPEG2000," in *Proc. of the Congress on Image and Signal Processing*, 2008, pp. 521–524.

[176] Y. W. Chang, H. C. Fang, and L. G. Chen, "High performance two-symbol arithmetic encoder in JPEG2000," in *Proc. of the IEEE International Symposium on Consumer Electronics*, 2004, pp. 101–104.

[177] T. Acharya and P. Tsai, *JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures.* Hoboken: John Wiley and Sons, 2005.

[178] Y. Z. Zhang, C. Xu, and L. B. Chen, "A dual-symbol coding arithmetic coder architecture design for high speed EBCOT coding engine in JPEG2000," in *Proc. of 6th International Conference On ASIC*, 2005, pp. 610–613.