

INVESTIGATION OF SPARSIFYING TRANSFORMS IN COMPRESSED SENSING FOR
MAGNETIC RESONANCE IMAGING WITH FASTTESTCS

by

Christopher Adams Baker

A Dissertation Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
in Engineering

at

The University of Wisconsin-Milwaukee

December 2016

ABSTRACT

INVESTIGATION OF SPARSIFYING TRANSFORMS IN COMPRESSED SENSING FOR MAGNETIC RESONANCE IMAGING WITH FASTTESTCS

by

Christopher Adams Baker

The University of Wisconsin-Milwaukee, 2016
Under the Supervision of Professor Guangwu Xu

The goal of this contribution is to achieve higher reduction factors for faster Magnetic Resonance Imaging (MRI) scans with better Image Quality (IQ) by using Compressed Sensing (CS). This can be accomplished by adopting and understanding better sparsifying transforms for CS in MRI. There is a tremendous number of transforms and optional settings potentially available. Additionally, the amount of research in CS is growing, with possible duplication and difficult practical evaluation and comparison. However, no in-depth analysis of the effectiveness of different redundant sparsifying transforms on MRI images with CS has been undertaken until this work. New theoretical sparsity bounds for the dictionary restricted isometry property constants in CS are presented with mathematical proof. In order to verify the sparsifying transforms in this setting, the experiments focus on several redundant transforms contrasting them with orthogonal transforms. The transforms investigated are Wavelet (WT), Cosine (CT), contourlet, curvelet, k-means singular value decomposition, and Gabor. Several variations of these transforms with corresponding filter options are developed and tested in compression and CS simulations. Translation Invariance (TI) in transforms is found to be a key contributing factor in producing good IQ because any particular translation of the signal will not effect the transform representation. Some transforms tested here are TI and many others are made TI by transforming small overlapping image patches. These transforms are tested by comparing different under-sampling patterns and reduction ratios with varying image types including MRI data. Radial, spiral, and various random patterns are implemented and demonstrate that the TIWT is very robust across all under-sampling patterns. Results of the TIWT simulations show

improvements in de-noising and artifact suppression over that of individual orthogonal wavelets and total variation ℓ_1 minimization in CS simulations. Some of these transforms add considerable time to the CS simulations and prohibit extensive testing of large 3D MRI datasets. Therefore, the FastTestCS software simulation framework is developed and customized for testing images, under-sampling patterns and sparsifying transforms. This novel software is offered as a practical, robust, universal framework for evaluating and developing simulations in order to quickly test sparsifying transforms for CS MRI.

©Copyright by Christopher Adams Baker, 2016
All Rights Reserved

To

My parents, Joyce and Thomas

My wife, Sara

My daughter, Annabelle

TABLE OF CONTENTS

1	Introduction to Compressed Sensing in MRI	1
1.1	Questions for Consideration in CS Research	5
1.2	Contribution	6
2	State of the Art in CS MRI	8
2.1	Under-Sampling for MRI	8
2.2	The Restricted Isometry Property in CS	9
2.3	Quantitative Measurements for Image Comparison	10
2.4	Compressed Sensing Software for MRI	12
3	Image Compression Techniques and Measures for IQ	14
3.1	Introduction	14
3.2	Methods	15
3.2.1	Compression Tests	16
3.3	Comparison of Image Measures	17
3.4	Results and Discussion	20
4	Sparsifying Transforms for CS MRI	27
4.1	Introduction	27
4.2	Determining Optimal Artifact Free Sparsifying Transforms for CS	27
4.3	Methods	30
4.3.1	Gabor Transform Experiments	31
4.4	Results and Discussion	32
4.4.1	Gabor Transform	42
5	MRI Sampling Trajectories for CS MRI Simulation	47
5.1	Introduction	47
5.2	Methods	48
5.2.1	MRI Under-Sampling Patterns Developed	48
5.3	Results	51
5.3.1	Comparing TIWT and TV	51
5.4	Discussion	55
5.4.1	Good Under-Sampling Patterns	55
5.4.2	Reconstruction of MIP	57

6	Optimization for CS Problems with FastTestCS	58
6.1	Introduction	58
6.2	Methods	59
6.2.1	“FastTestCS”, A Fast Software Framework for Testing Under-Sampling and Sparsifying Transforms in CS	59
6.3	Robust Simulation Framework	68
6.4	Results	72
6.4.1	Lack of Good Comparison Software	73
6.4.2	Fast Software Framework for Testing Sparsifying Transforms with CS “FastTestCS”	75
6.5	Discussion	79
6.5.1	Stopping Criteria	79
6.6	Other Algorithms	80
6.6.1	Computation Time with FastTestCS	80
6.6.2	Use of FastTestCS	81
7	Proof of Bounds on D-RIP Constants	82
7.1	Introduction	82
7.1.1	Contribution	82
7.2	New D-RIP Bounds	83
8	Conclusion	88
8.1	Future work	88
8.2	Best Practice Definition	88
8.3	More Sparsifying Transforms	89
8.3.1	FastTestCS: Enhancement, Versatility and Distribution	90
	Bibliography	91
	Appendix A The Gradient of the Regularization Function	95
	Appendix B MRI Dataset Parameters	97
	Appendix C Software References for Different Transforms Used	98
	Curriculum Vitæ	100

LIST OF FIGURES

1.1	Compressed sensing for MRI reconstruction	2
3.1	The process of combining the coefficients and compression with wavelets and contourlets.	17
3.2	Compression keeping 20% of pixels, on three types of images, multi-coil high sensitivity (from left to right and top to bottom)(a-c), multi-coil low sensitivity (d-f), and single coil (g-i). (a) No compression (b) Contourlet compression, (c) Wavelet compression, (d) No compression (e) Contourlet compression, (f) Wavelet compression (g) No compression, (h) Contourlet compression, (i) Wavelet compression.	22
3.3	Compression keeping 15% of pixels.(from left to right) (a) Contourlet + wavelet, (b) Wavelet (c) Contourlet.	23
3.4	TOF MIP detail comparison. (left to right, top to bottom) (a) Fully sampled MIP reconstruction, (b) Fully sampled individual TOF image from which compression is performed, (c) fully sampled detail of MIP, (d) Absolute value threshold, keeping 10% of pixels, (e) Contourlet compression, keeping 10% of pixels, (f) Wavelet compression, keeping 10% of pixels.	25
4.1	(left)Fully sampled original MRI brain image quadrant. (right) Wavelet compression and reconstruction showing artifacts where 80% of the smallest wavelet coefficients are removed by a hard threshold.	28
4.2	(left)Fully sampled original MRI brain image quadrant. (right) CS reconstruction with decimated wavelets showing artifacts where sampled lines of k-space are reduced aggressively by a factor of 3 (only 33% of "full" data used to reconstruct).	28
4.3	Unconstrained CS results with reduction factor of 33% (left to right and top to bottom) (a) Original image, (b) Without regularization, (c) With only ℓ_1 -norm regularization, (d) With only TV (e) With only curvelet regularization, (f) With only wavelet regularization, (g) With only contourlet regularization, (h) All combined regularization.	33
4.4	CS reconstruction with reduction factor of 30% (left to right and top to bottom) (a) Fully sampled (b) Linear reconstruction (c) K-SVD (d) OWT, (e) Over-complete TICT, (f) TICT, (g) OWT on small patches.	34

4.5	CS reconstructions, (a) Single channel T1-weighted brain fully sampled, (b) Initial guess linear reconstruction, reduction factor of 50%, (c) Reconstruction with an over-complete TICT untrained dictionary and a reduction factor of 50%, (d) Reconstruction with a trained dictionary (K-SVD) and a reduction factor of 50%. (e) Initial guess linear reconstruction, reduction factor of 20%, (f) Reconstruction with an over-complete TICT untrained dictionary and a reduction factor of 20%, (g) Reconstruction with a trained dictionary (K-SVD) and a reduction factor of 20%.	35
4.6	Single-channel T1-weighted brain MRI data (a) Linear reconstruction, reduction factor = 25%, (b) Linear reconstruction, reduction factor = 50%, (c) Fully sampled, (d and e) CS reconstruction with wavelet sparsifying transform on full image representation, reduction factor = 25% and 50%, respectively, (f and g) CS reconstruction with wavelet sparsifying transform on small and redundant patch representation, reduction factor = 25% and 50%, respectively.	36
4.7	CS reconstructions with a reduction factor of 40% (left to right and top to bottom, with the corresponding enlarged region shown above) utilizing different sparsifying transforms as regularization (a) Fully sampled reconstruction and (b) Reconstructions with TV, (c) DCT, (d) TICT, (e) WT, and (f) TIWT.	37
4.8	Comparison of linear reconstruction error (non CS) and CS reconstruction errors (NMSE) with different sparsifying transforms: TV, orthogonal DCT (OCT), TICT, orthogonal WT (OWT), and TI WT as the sparsifying transforms. Each point is an average of 10 results.	38
4.9	IQ (NMSE) vs redundancy. Patch size = 256×256 , 2D circular shift of 6 rows/columns, reduction factor = 2	41
4.10	Synthetically generated images, varying numbers of points or circles or sizes can be adjusted to increase or decrease image complexity for rerunning simulations with different images	43
4.11	Signal reconstruction test 1, one long pulse (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	43
4.12	Reconstruction test 2, 20 short pulses (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	44
4.13	Reconstruction test 3, Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	45
4.14	Reconstruction test 4, penguin (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	45
4.15	Signal reconstruction test 5, one long pulse + Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	46
4.16	Image reconstruction test 5, one long pulse + Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	46
4.17	Reconstruction test 6, T1 MRI, (a) Original , (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	46
4.18	Reconstruction test 7, MIP MRI, (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV	46

5.1	Fully sampled reconstruction simulation images: (a) Shepp-Logan phantom 256×256 (b) T1 MRI image 256×256 (c) TOF MRI MIP image $192 \times 512 \times 512$ 8 channel (d) TOF MRI (one slice) image 512×512	48
5.2	Under-sampling patterns at 25%: (a) Radial with 70 lines (b) Spiral (c) Random (d) 2D variable density (e) 2D variable density with 25 point fully sampled center radius (f) 1D variable density with 32 line fully sampled center	49
5.3	CS reconstructions of T1 MRI with 25% samples (a) Radial (b) Spiral (c) Random (d) 2D variable density (e) 2D variable density with 25 point fully sampled center radius (f) 1D variable density with 32 line fully sampled center	50
5.4	TOF MRI k-space data (1 receiver) and MIP reconstruction zoomed in (a) High resolution 512×512 data (b) Cropped low resolution 256×256 (c) High resolution reconstruction (d) Cropped low resolution reconstruction	52
5.5	TOF reconstruction with 38% random under-sampling, receiver 1, slice 1 (a) Fully sampled (b) Linear (c) TIWT CS (d) TV CS	53
5.6	TOF MIP reconstruction with 38% random under-sampling (a) Linear (b) TIWT CS (c) TV CS	53
6.1	Implementation of CS image reconstruction simulation	63
6.2	(left to right) (a) PDF of sampling function. Vertical scale of the plot is 0 to 1. (b) The resulting sampling pattern in k-space, where the white lines are acquisition lines.	64
6.3	Unconstrained CS with different transforms and regularization parameters (λ).	67
6.4	Unconstrained CS iteration comparison.	67
6.5	Unconstrained CS result for iteration comparison with wavelet regularization. (left to right) (a) 15 iterations, (b) 35 iterations, (c) 100 iterations.	68
6.6	Unconstrained CS results, (left to right, top to bottom) (a) Original image, (b) Without regularization, (c) With only ℓ_1 -norm regularization, (d) With only TV (e) With only curvelet regularization, (f) With only wavelet regularization, (g) With only contourlet regularization, (h) All transforms combined regularization.	69
6.7	CS with different transforms and soft threshold values	70
6.8	Unconstrained CS simulation results using optimized soft threshold (left to right, top to bottom)(a) Original image (b) Non-transformed (no threshold) (c) Wavelet (d) Contourlet (e) Curvelet	71

LIST OF TABLES

3.1	Compression to 20% of pixels for comparison between contourlet filter results and settings	20
3.2	Compression to 20% of pixels for comparison between wavelet filter results	21
3.3	Compression results	24
4.1	Comparison of sparsifying transform parameters in CS reconstruction	39
4.2	Sparsity and CS reconstruction errors of various signals (L - Linear, G - Gabor, TV - Total Variation, MSE - Mean Square Error)	44
5.1	Radial under-sampling pattern IQ comparison with MSE between CS reconstructed images and reduction factors (%) with 10 and 500 NLCG iterations	55
5.2	Spiral under-sampling pattern IQ comparison with MSE between images and reduction factors (%)	55
5.3	Random under-sampling pattern IQ comparison with MSE between images and reduction factors	56
5.4	1D variable density with small (32 line) center under-sampling pattern IQ comparison with MSE between images and reduction factors	56
5.5	1D variable density with large (60 line) center under-sampling pattern IQ comparison with MSE between images and reduction factors	56
5.6	2D variable density under-sampling pattern IQ comparison with MSE between images and reduction factors	56
5.7	2D variable density with small (25 pixel radius) center under-sampling pattern IQ comparison with MSE between images and reduction factors	57
6.1	Time measurement (seconds) comparison of 40 wavelet transforms (10 complex, forward and reverse), O = Orthogonal, with C++ and Matlab implementations.	73
6.2	Time and memory measurements when using various transforms in CS simulation vs. threads, Block Size (BS) and Sliding Distance (SD) for a single image reconstruction (256x256)	76

ACKNOWLEDGMENTS

The author wishes to thank the following from the Department of Computer Science and Engineering at the University of Wisconsin – Milwaukee: Professor Guangwu Xu, Academic Adviser, for guidance in research and publications, with mathematical assistance in the proof. Dr. David Klemer, former Adviser, for important guidance and advice. Dr. Leslie Ying, former Adviser, for substantial initial help and guidance in this research and publications. Additional thanks to Dr. Kevin King, Scientist at GE Healthcare, for providing guidance throughout this research, helpful review of many results and the MRI data used in the experiments.

Onward in Training

Oh, what could it be,
that the PhD has for me?
To work is good, for such is the day,
put my hands to it and come what may.
For I am eager to solve,
all that life has to resolve.
Where to start, where to end,
these experiments must extend.
To prove the theory, win the prize,
such a monumental work in disguise.
Reading books and searching papers,
taxing thoughts and extending labors.
Barriers, obstacles and misunderstanding,
are par for the course in journeying.
Work, work, work into the night,
work, work, work until it's right.
Spending time and going in debt,
is there recovery in what's left?
Life continues with a hitch,
a baby is born, what more could I wish?
Ah, the end in sight, the research done,
the defense in place, the degree won!
The many fruits of this labor are coming to be,
with lessons learned. Oh, rejoice with me!

Christopher Baker

Chapter 1

Introduction to Compressed Sensing in MRI

Magnetic Resonance Imaging (MRI) is a noninvasive diagnostic modality used to create images of 3-Dimensional (3D) biological tissue utilizing non-ionizing microwave radiation with magnetic fields that are 100,000 times more powerful than what is measured from the earth's natural magnetic field. An MRI signal arises from protons which are in water and has a wide range throughout our body which gives good imaging possibilities. High quality images are created when the MRI signal or k-space frequency data is fully sampled based on the Nyquist sampling criteria. However, when using this criteria the length of time required to acquire these samples can be a great challenge. Many advances in speed have occurred over the past few decades, however, many technical and physical limits have been reached. Less sampling time is desired in MRI to accelerate the data acquisition speed and enable more applications such as cardiac imaging. A typical high resolution acquisition can take five minutes which may negatively effect Image Quality (IQ), due to the movements of the patient and biologic function.

A promising theory that can potentially be applied to reduce MRI acquisition time is Compressed Sensing (CS) which allows under-sampling of k-space below what the Nyquist criteria requires without compromising IQ, see Figure 1.1. K-space is the sampling frequency domain of MRI; it is inversely related to the spatial image domain via the Fourier transform. Candès *et al.* introduce theory and experiments for the general CS framework [15]. Under-sampling for CS must be carefully designed so that artifacts and errors can be minimized and noise-like. MR scanner

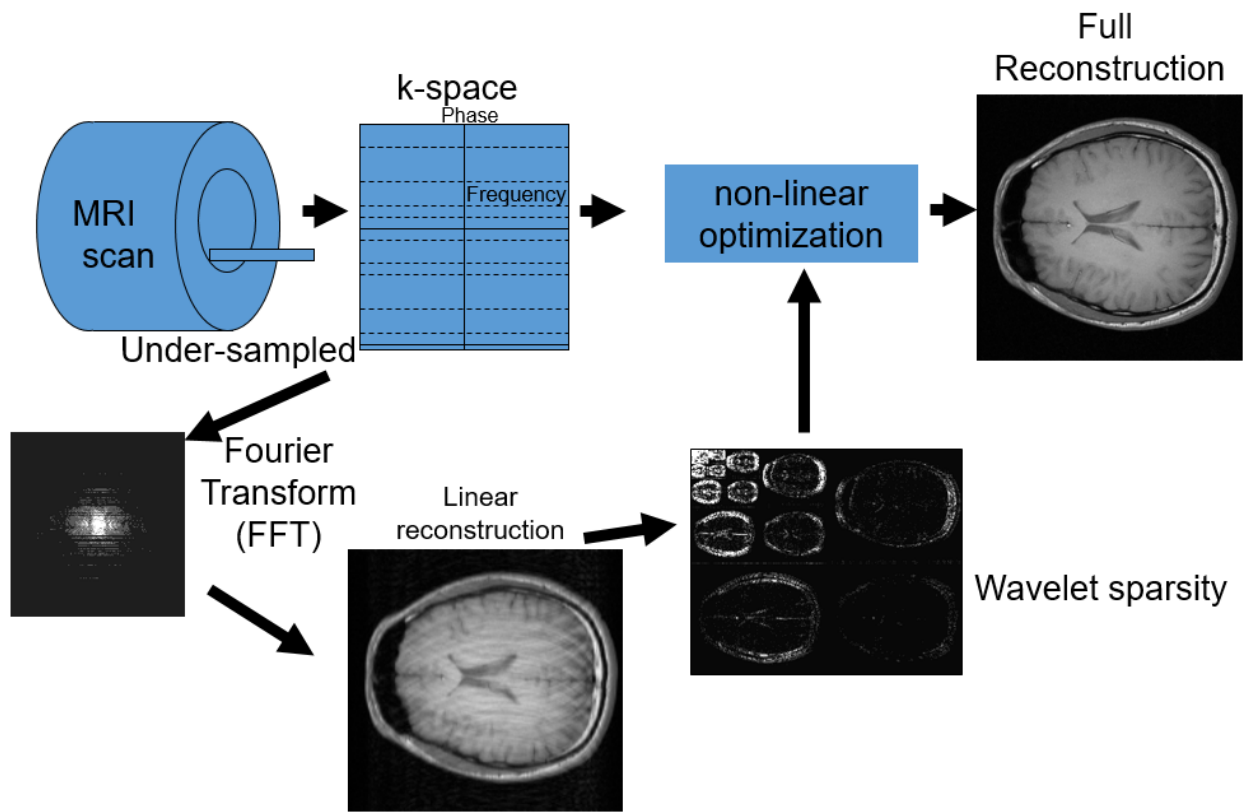


Figure 1.1: Compressed sensing for MRI reconstruction

data is redundant which allows measuring signals at a rate closer to their intrinsic information rate rather than the Nyquist rate.

The CS theory has been applied by Lustig *et al.* [40] to MRI reconstruction problems outlining a technique to produce a high quality MRI image while under-sampling k-space and thus improve the imaging speed. The application is made possible by the fact that MRI satisfies the two conditions of CS: the image is sparse in a certain transform domain and the under-sampling artifacts can be designed to be noise-like. Parallel imaging is another important speed up technique to combine the signals from multiple coils at the same time allowing some additional parallel sampling of k-space to boost the IQ. Parallel imaging can be combined with CS to have sampling speed further improved. Deshmane *et al.* discuss the effects of under-sampling and the advantages of using parallel imaging that allows faster acquisition [27].

High sparsity is a representation of an image or signal that has minimal information. When a p -dimensional signal is k -sparse, the signal has p elements, but only has k elements of valuable information. In cases where there is high sparsity, $k \ll p$, CS provides recovery of a signal with less samples required than Nyquist sampling. Therefore, CS produces images with the same quality in less time. MRI data and images are not typically sparse, however, a sparsifying transform is used which will produce a sparse representation in the transform coefficient domain and thereby allow the application of CS to recover the image.

If under-sampling data is acquired, the signal recovery in equation 1.1 is under-determined. The signal $\beta \in \mathbb{R}^p$ is recovered from incomplete measurements $y \in \mathbb{R}^n$ and a sampling matrix $\Phi \in \mathbb{R}^{n \times p}$. Error z is added to accommodate real world systems with noise. The error must be sufficiently small with respect to the measurement magnitude, where ϵ is at most proportional to the noise level [17].

$$y = \Phi\beta + z, \quad \text{where } \|z\|_2 \leq \epsilon \quad (1.1)$$

The solution to 1.1 can be obtained using either a combinatory algorithm, a greedy algorithm, or an ℓ_1 minimization for which convex minimization with linear programming applies. However, a combinatory algorithm is non-practical since there is an exponentially expanding number of search possibilities to find, $O(2^n)$, where there are n elements. When the set of possible elements

is small, the problem becomes a sparse recovery.

Greedy algorithms rely on an iterative approximation of the signal coefficients and support. Orthogonal Matching Pursuit (OMP) and iterative thresholding are some of the simplest greedy algorithms. OMP iteratively finds columns most correlated with the measurements and the estimated signal. Iterative thresholding will iterate on the steepest gradient descent step until it meets convergence criteria.

ℓ_1 minimization can also be used to find a sparse solution because the problem space has a convex shape, and therefore, can be minimized by finding the minimum point on the curve or cone. The ℓ_1 norm is defined

$$\|x\|_1 = \sum_{i=1}^p |x_i| \quad (1.2)$$

And the ℓ_2 norm defined here:

$$\|x\|_2 = \sqrt{\sum_{i=1}^p |x_i|^2} \quad (1.3)$$

The problem is now convex and can be minimized with quadratic programming, see equation 1.4.

$$\begin{aligned} & \text{minimize : } \|Wx\|_1 \\ & \text{subject to : } \|\Phi x - y\|_2 \leq \epsilon \end{aligned} \quad (1.4)$$

Where W represents the sparsifying transform and x is the signal approximation. In order to practically implement this sparse recovery problem, an unconstrained formulation is utilized as shown in equation 1.5.

$$\text{minimize } \|\Phi x - y\|_2 + \lambda \|Wx\|_1 \quad (1.5)$$

λ is a regularization parameter for balancing data consistency and sparsity. Equation 1.5 can be solved in many ways; for example, a Non-Linear Conjugate Gradient (NLCG) method can be used, as in [45] and [40].

1.1 Questions for Consideration in CS Research

Stated here are some of the questions and challenges for consideration when undertaking this research. What optimization techniques and sampling patterns for simulation are applicable to MRI? What computational hurdles may need to be crossed? Should certain transforms that are computationally quicker or more robust, be favored over others that are theoretically superior? What is the best experimental environment? Should ℓ_1 Magic be used? Should NLCG be used? Should constrained vs unconstrained optimization be used?

Another question to consider in applying sparsifying transforms is how robust they are against changes in under-sampling that could create unmanageable artifacts that affect diagnostic quality. The best sparsifying transform would produce consistent results with various under-sampling patterns across required image types. Practically, every real MRI image differs drastically in the size of the data collected, sparsity, edges, or detail level, and will produce different under-sampling artifacts even if the same sampling pattern is used. Are there transforms that can create great images and still be applicable for several image types, sampling patterns, sizes and under-sampling ratios?

One of the biggest variables with CS MRI is to determine the sampling pattern that will produce the best images. There are several types of MRI images: Multi channel, single channel, MIP, cine and other temporal, k-time space types. How should the many parallel imaging reconstruction techniques be integrated or compared? What sampling is practical on MRI machines? How do you test robustness of the sampling pattern for various image types? What about non-Cartesian sampling? How does re-gridding/re-sampling affect the CS algorithm? How does PSD impact different sampling schemes for the CS recon solution?

Since CS can have an effect of de-noising, does the sampling pattern need to be adjusted for this effect? Potentially too much smoothing could be introduced which would be very undesirable due to lack of sharp edges. What changes to the sampling pattern or optimization algorithm should be made? Is the type of sparsifying transform totally the source of this smoothing or are there other factors? What artifacts can become better dispersed or incoherent based on the

sampling pattern? Are certain artifacts particularly difficult for CS?

Does accuracy or IQ degrade after certain number of iterations? How robust are these minimization techniques and are they usable for all image types? Does image size affect the optimization technique? Does the number of iterations need to be defined or optimized? Does a balance between iteration time and IQ need to be established? Do different sparsifying transforms cause the number of iterations or stability of the minimization to change? What artifacts are present if too few iterations are performed?

What ℓ_1 minimizations or convex optimizations can and should be used for MRI? There are many, so is there a tradeoff – speed vs. accuracy? What about reconstruction times? Does the minimization algorithm need to be tested to verify it does not diverge from a solution? Are certain objective functions better than others? In the NLCG, the gradient of the regularization function is needed; there are different ways to calculate the gradient. Which one is best?

1.2 Contribution

In summary, the approach followed for this research looks at the current state of the art to see how to advance the science, with an in-depth look at each aspect of the problems and solutions. Lustig's work is a starting point for developing CS MRI, which used the orthogonal wavelet sparsifying transform. Preliminary experiments of this research showed improvement with other sparsifying transforms. Understanding the role of better sparsifying transforms in CS for MRI is key to developing the appropriate solution that improves the speed of the MRI.

Sparsifying transforms are proposed to provide advancement for the CS solution in MRI: Translational Invariant Wavelet Transform (TIWT), Gabor transform, three redundant patch transforms that use OMP with the CT or WT, and K-means Singular Value Decomposition (K-SVD). These dictionaries produce a higher IQ in CS MRI compared to other sparsifying transforms such as TV and non-redundant WT. Additionally, a software framework called FastTestCS is presented for developing and comparing other sparsifying transforms and under-sampling patterns. Next, new theoretical bounds for sparsity constraints in CS are introduced.

The outline is as follows: Chapter two provides a more detailed background to the chap-

ters that follow. Chapter three explains several image comparison techniques for identifying and measuring IQ of compressed and CS reconstructed images. The qualities and options for using these measures are discussed to give insight into quantitative image comparison methods that work well for identifying good sparsifying transforms. Chapter four covers an explanation of redundant sparsifying transforms compared with other transforms such as Gabor wavelets in CS for MRI. Problems in finding good sparsifying transforms in CS MRI are also identified. Chapter five develops various under-sampling patterns utilized for CS in MRI. Chapter six highlights complications and extensions that must be overcome when simulating CS MRI solutions. Insights for researchers and practical software engineering practices are advanced. FastTestCS, a tool for comparing simulations and running CS reconstructions with various sparsifying transforms, is formulated as a time saving solution. The NLCG and Cartesian CS sampling scheme is detailed for quick application to enhance standard Cartesian sampling in MRI. Chapter seven presents the sparsity bound developed for the Dictionary based Restricted Isometry Property (D-RIP). New techniques and solutions are utilized for proving this sparsity bound.

Chapter 2

State of the Art in CS MRI

2.1 Under-Sampling for MRI

For quick adoption of CS in MRI, patterns that follow existing sequences and trajectories with some under-sampling should be used. Both Cartesian and non-Cartesian under-sampling patterns, such as radial and spiral trajectories, have already been investigated for CS in MRI. Lustig *et al.* utilized a 1D variable density under-sampling [40]. Bigot *et al.* investigated a random block of sensing vectors [8]. Castets *et al.* developed an interleaved golden angle spiral for imaging at high magnetic fields [22]. Chan *et al.* utilized CS with different radial sampling schemes [23]. Polak *et al.* considered how the performance of CS is bounded when grouped sampling is necessary [46]. For MRI, sampling speed is somewhat constrained to human physiology and Pulse Sequence Design (PSD).

Radial and spiral sampling techniques are used for their advantages. For example, they reduce structural and motion artifacts, while keeping the requirement of artifact incoherence. King improved non-Cartesian spiral scanning with an anisotropic field of view where acquisition is fast and reconstruction improves IQ utilizing gridding [36]. Liu *et al.* developed a circular Cartesian under-sampling strategy for CS that does not require gridding [37]. Ersoz *et al.* also analyzed a Cartesian based radial sampling trajectory for MRI [30]. In many of these cases, due to computational simplicity, the Fast Fourier Transform (FFT) is desired, which operates on a Cartesian grid. Gridding and inverse gridding with interpolation are often needed to transform from a non-Cartesian sample space to a Cartesian space and back again, adding significant

computation when repeated for iterative optimization in CS. However, there are spiral and radial linogram Cartesian based trajectories which do not require gridding and interpolation which easily make use of the FFT directly. Therefore, careful selection of sampling patterns does have an impact on reconstruction time.

2.2 The Restricted Isometry Property in CS

Let $\Phi \in \mathbb{R}^{n \times p}$ and $\beta \in \mathbb{R}^p$ be a signal such that

$$y = \Phi\beta + z \quad (2.1)$$

with $\|z\|_2 \leq \varepsilon$. In CS, one can find a good stable approximation (in terms of ε and the tail of β consisting of $p - k$ smallest entries) of β from the measurement matrix Φ and the measurement y through solving an ℓ_1 -minimization, provided that Φ belongs to a family of well behaved matrices. A subclass of this family of matrices can be characterized by the well known Restrictive Isometry Property (RIP) of Candès, Romberg, and Tao, [17] and [20]. This property requires the following relation for Φ and an integer k , $1 \leq k \leq p$, the k Restricted Isometry Constant (RIC) δ_k is the smallest constant such that

$$\sqrt{1 - \delta_k} \|c\|_2 \leq \|\Phi c\|_2 \leq \sqrt{1 + \delta_k} \|c\|_2 \quad (2.2)$$

for every k -sparse vector c (namely, c has at most k non-zero components). Then, the matrix Φ satisfies the k RIP with RIC δ_k . The RIP constant is defined as the infimum of all possible δ for a given Φ .

$$\delta_K = \inf[\delta : (1 - \delta)\|c\|_2^2 \leq \|\Phi c\|_2^2 \leq (1 + \delta)\|c\|_2^2], \quad (2.3)$$

$$\forall k \leq K, \forall c \in \mathbb{R}^k$$

ℓ_1 minimization can recover a sparse signal under specific conditions or bounds on δ . Some bounds on δ have been determined in previous publications [17], [20], [11], and [14]. Notably, Cai and Zhang have established several sharp RIP bounds that cover the most interesting cases of δ_k and δ_{2k} [10][9] showing $\delta_k \leq \frac{1}{3}$, $\delta_{2k} < \frac{\sqrt{2}}{2}$.

A key requirement in this setting is a signal being sparse or approximately sparse. Indeed, many families of integrating signals have sparse representations under suitable bases. Recently, an interesting sparsifying scheme was proposed by Candès *et al.* [13]. In their scheme, instead of bases, tight frames are used to sparsify signals.

Let $D \in \mathbb{R}^{p \times d}$ ($d \geq p$) be a tight frame and $k \leq d$. Candès *et al.* [13] suggests that one use the following optimization to approximate the signal β :

$$\hat{\beta} = \operatorname{argmin}_{\gamma \in \mathbb{R}^p} \|D^* \gamma\|_1 \quad \text{subject to} \quad \|y - \Phi \gamma\|_2 \leq \varepsilon. \quad (2.4)$$

The traditional RIP is no longer effective in the generalized setting. Candès *et al.* defined the *Dictionary-restricted isometry property* D-RIP which generalizes RIP [13]. Here the formulation of D-RIP is used as in Lin *et al.* [51].

Definition 1 *The measurement matrix Φ obeys the D-RIP with constant δ_k if*

$$(1 - \delta_k) \|Dv\|_2^2 \leq \|\Phi Dv\|_2^2 \leq (1 + \delta_k) \|Dv\|_2^2 \quad (2.5)$$

holds for all k -sparse vectors $v \in \mathbb{R}^d$.

The RIP is now a special case of D-RIP (when the dictionary D is the identity matrix). For D being a tight frame, Candès *et al.* [13], proved that if $\delta_{2k} < 0.08$, then if $D^* \beta$ is approximately k -sparse, the solution to (2.4) is a good approximation of β . Lin *et al.* [38], improved this result to $\delta_{2k} < 0.472$ by using some techniques developed by Candès *et al.* [18].

2.3 Quantitative Measurements for Image Comparison

Quantitative measurements are used and listed here. However, it is important to note that quantitative measures do not directly correlate to a clinical setting of perceived IQ, where a radiologist's expertise is needed. Wang *et al.* present a helpful example of visual IQ differences in image which all have the same Mean Square Error (MSE), yet have quite different visual IQ. They introduce Structural SIMilarity (SSIM) as a better way of assessing IQ [50].

The MSE or Normalized MSE (NMSE) is defined as a measure of pixel intensity error between x_i and \hat{x}_i , which are the original and reconstructed image.

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} |x_i - \hat{x}_i|^2 \quad (2.6)$$

$$Normalized\ MSE = \frac{MSE}{\frac{1}{N} \sum_{i=0}^{N-1} |x_i|^2} \quad (2.7)$$

The smaller this value is, roughly correlates to less measured error and better IQ. MSE and NMSE is used extensively in this work as a consistent numeric measure of image differences.

The Standard Deviation (STD) where \bar{x}'_i is the mean of x'_i and x'_i is a small (15×15) block of pixels in a section of the image, which is relatively uniform.

$$\sigma = STD = \left(\frac{1}{N} \sum_{i=0}^{N-1} |x'_i - \bar{x}'_i|^2 \right)^{\frac{1}{2}} \quad (2.8)$$

$$SNR = 10 \log_{10} \frac{\sigma^2}{MSE} \quad (2.9)$$

Note that the whole image is used for STD when calculating the Signal to Noise Ratio (SNR) rather than a small (15×15) block. These measurements of quality (Eq's (2.6) - (2.9)) are presented in [49]. Closely related to MSE is the sum of error calculation. This is where the pixel by pixel difference is calculated between the golden image (x) vs. reconstructed image (\vec{x}) and totaled.

$$sum\ of\ error = \sum_m \sum_n |x_{mn} - \vec{x}_{mn}| \quad (2.10)$$

Another well-known measurement is SSIM, where several measures are combined to produce an index for measuring the similarity between two images x and y .

$$Contrast = \sigma_x \quad luminance = \frac{1}{N} \sum_{i=1}^N x_i \quad SSIM = \frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y} \quad (2.11)$$

We use a measure of noise suppression and edge preservation from [48]

$$Noise\ Suppression = \frac{\Gamma(s - \bar{s}, \hat{s} - \bar{\hat{s}})}{\sqrt{\Gamma(s - \bar{s}, s - \bar{s})\Gamma(\hat{s} - \bar{\hat{s}}, \hat{s} - \bar{\hat{s}})}} \quad (2.12)$$

where \bar{s} and $\bar{\hat{s}}$ are mean values of the Region of Interest (ROI) of the two images being compared.

$$\Gamma(s_1, s_2) = \sum_{i,j \in ROI} s_1(i, j) \cdot s_2(i, j) \quad (2.13)$$

$$Edge\ Preservation = \frac{\Gamma(\Delta s - \overline{\Delta s}, \widehat{\Delta s} - \overline{\widehat{\Delta s}})}{\sqrt{\Gamma(\Delta s - \overline{\Delta s}, \Delta s - \overline{\Delta s}) \Gamma(\widehat{\Delta s} - \overline{\widehat{\Delta s}}, \widehat{\Delta s} - \overline{\widehat{\Delta s}})}} \quad (2.14)$$

where Δs is the high passed filtered version of s obtained by a standard 3×3 approximation of the Laplacian operator.

2.4 Compressed Sensing Software for MRI

Several software packages have been developed and released in Matlab that can be used for CS simulation. Here are a few of noteworthy contributions:

- CVX [33]
- ℓ_1 -Magic [21]
- SparseMRI V0.2 [40]

CVX is Matlab software for disciplined Convex Programming. Several advances have been made to this software over the past decade. It solves the convex optimization problem allowing constrains and objectives with standard Matlab syntax. Although this software will support typical CS MRI problems, it will perform slow when scaling up to larger matrices which are needed for image processing.

ℓ_1 -Magic is specifically designed for solving CS problems by using nonlinear recover algorithms based on convex optimization. It does perform large scale optimization for images that would be relevant for MRI. Although the software source code is available, there are still challenges in developing new redundant sparsifying transforms that have stable convergence.

SparseMRI is a collection of Matlab functions that implement the CS algorithm described by Lustig *et al.* It performs ℓ_1 penalized non linear conjugate gradient reconstruction for MRI data.

This uses the orthogonal wavelet as a sparsifying transform in the CS reconstruction which can be considered as a first step for comparison of other sparsifying transforms.

Matlab is very versatile with a good purpose for developing algorithm prototypes for CS that can easily be distributed and consumed by the research community. However, with each of these pieces of software, the reconstruction times are longer compared to an optimized re-implementation in C++. Matlab does have an optional MEX feature allowing custom C++ methods to be developed that run within Matlab. Using MEX is a great way to speed up highly computational code and optimize it to bring additional speed up to these algorithms.

Chapter 3

Image Compression Techniques and Measures for IQ

3.1 Introduction

There have been some studies of the sparsity of typical medical images with a few standard compression methods such as CT and WT. Alexander analyzed Joint Picture Experts Group (JPEG) compression thresholds that do not compromise medical image IQ [2]. However, with CS, no major comparison of the effectiveness of different sparsifying transforms on MRI images has been undertaken. Studying the sparsity of data with a specific basis is helpful because the CS approach requires that the desired image have a sparse representation in a known transform domain. Sparsity is defined for a vector as the number of non-zero coefficients. A lesser number of coefficients needed to represent an image corresponds to a sparser representation. In CS MRI, an ideal sparsifying transform would produce images without artifacts, is not overly de-noising, is able to account for various sampling levels and would work for various MRI image types.

The stability of a sparsifying transform is critical for CS. Perhaps investigating IQ and detail preservation after transform coefficient truncation or noise reduction will give clues as to what transforms to use for CS. For image compression standards like JPEG, there are orthogonal transforms, such as wavelets and cosines, which represent images as sparse coefficients. These applications truncate small values while still allowing expansion to the original image after transmission[34].

How do you compare techniques in CS? A good solution is to compare resulting image differ-

ences objectively. However, there are many pitfalls in performing the correct image comparison. One can judge images from human perception, which may range from being subjective and varying, to standardized and professional. How does one measure subjective visual qualities? Using mathematical calculations to compare IQ has a range of results from being ineffective and misleading to precise and essential. Do certain levels of qualitative or quantitative measures need to be used for medical imaging? How does one determine the signal to be measured vs. noise for all 2D images? The most accurate image comparisons are between the “golden,” fully sampled image and the under sampled reconstruction to verify if there are artifacts introduced. However, this is not the real world CS case because the original data is already under sampled – there is no gold standard. Do additional precautions need to be made that guarantee the reconstruction is free from artifacts? One would be challenged to verify every possible image scenario, but is there a better approach?

Ultimately, since the user is the radiologist, what works well for them? Do compression experiments inform good CS results? Do certain sparsifying transforms that lend themselves more to MRI image compression also work for CS? What specific measures of IQ are needed or need to be avoided? What precautions should be made when under-sampling? To answer these questions, image compression tests are proposed with several IQ comparisons as a first step to guide future CS experiments.

3.2 Methods

To narrow the field of possible sparsifying transforms covered in many publications, compression tests are proposed to understand how well transforms and settings perform. The quality of a transform for compression can be measured by IQ tests where the representation of the smallest valued coefficients are truncated and then the image is reconstructed with only the larger coefficients. Quality can be seen visually by looking at noise, artifacts, distortion and detail preservation. Quantitatively, high fidelity can be measured with direct comparisons between the ideal image and the reconstruction with image subtraction or (MSE) and other calculations such as SSIM index [50], edge preservation and noise suppression[48].

The following sparsifying transforms are compared:

1. Orthogonal wavelets with all combinations of Wavelab filters and types
2. Standard redundant contourlets
3. Crisp contourlets
4. Wavelets with contourlets
5. Curvelets

The transforms tested are from publications and software available from various sources listed in the appendix.

3.2.1 Compression Tests

For each compression technique, a hard threshold is used where the smaller coefficients are set to zero depending on a desired threshold or percentage of compression. In using each of these transforms, the real and imaginary datasets are separately transformed and then re-combined in the transform domain with an absolute value calculation.

For the compression testing, Matlab and Wavelab are used extensively to loop through all the different Wavelab filters, transforms and settings of interest. Due to the Wavelab software having a similar API for each transform, setup is minimal. The integration of the other sparsifying transform software for curvelets, contourlets, and various thresholds is also straight forward. The steps involved are: Reading in the data from disk, applying an IFFT on the 2D image, applying a transform of choice, truncating the data, applying the inverse of the transform, and calculating various quantitative comparison measures between the original and the compressed image. All these simulations are performed timely on current laptop computers. Therefore, efficiency and performance of software are not issues considered for these tests.

A similar approach is used to perform a combined basis of wavelet + contourlet compression, see Figure 3.1. The process starts by transforming the input image with each transform independently. Combining the coefficients in the transform domain is performed by sorting the

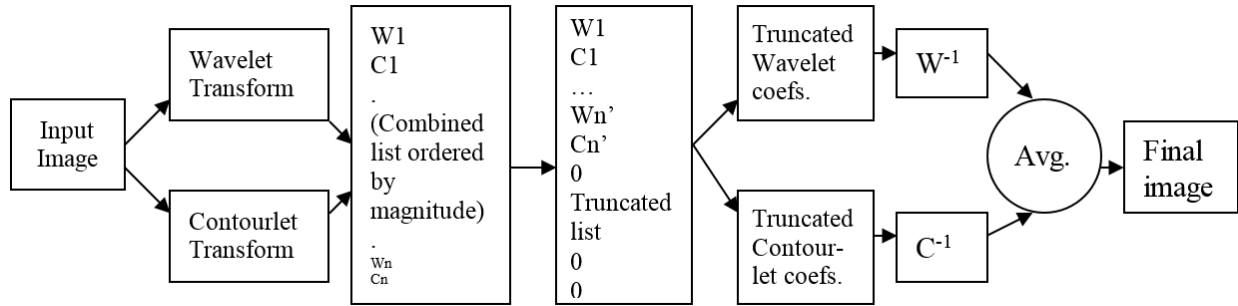


Figure 3.1: The process of combining the coefficients and compression with wavelets and contourlets.

magnitude. The combined list is truncated by setting the smaller coefficients in the list to zero. Then the combined coefficients are separated to use the respective inverse transform. Finally, an averaging of the two images is completed to produce the final image. The averaging process is a pixel-by-pixel average of a wavelet image pixel with the corresponding contourlet image pixel. This scheme is developed because it is the most similar to the way a CS compression will be performed. It is expected that for CS, when basis are combined, the coefficients are combined in a similar fashion.

For Time Of Flight (TOF) Maximum Intensity Projection (MIP) datasets, in addition to wavelet or contourlet compression, a pixel magnitude threshold is used where the image is compressed by just a hard threshold. This can be done because in TOF MIP datasets, the data of interest is of a much higher magnitude than the rest of the image.

Different image datasets are compared: A brain image, a TOF MIP image, and an individual coil image from a multi-coil brain dataset. A visual inspection of the smoothness, detail preservation and artifacts is performed.

3.3 Comparison of Image Measures

For the contourlet and wavelet transform, different parameters are adjusted to find which would achieve better compression. Both transforms use filters and there are differences depending on which filters are used and the number of coefficients for each filter. A trial and error search of available settings is performed. Some settings produce much better images as measured

quantitatively and qualitatively and these are then used for compression tests. An important factor in dealing with the contourlet transforms is that it produces an over complete set of coefficients. In the transform domain there are more coefficients than initial image values. Therefore, since the number of coefficients truncated is a percentage of the image coefficients, the contourlet compression is at a disadvantage to wavelet compression, which does not have an over complete set. This is also the case for curvelets, and the combined wavelet + contourlet transform. However, the purpose of these tests is to determine transform robustness during compression with the application of interest being CS which does not perform truncation.

MSE is one of the easiest measures to implement. From the quantitative perspective, MSE can be a great measure; however, it can easily become misleading. When MSE is used within the same type of image artifact, it is great at distinguishing between small levels of error. When different filters or sampling patterns and artifacts are introduced, however, then the use of MSE across the board is questionable.

Four other measurements are used and compared: SNR, edge preservation, noise suppression and SSIM. Edge preservation measurement comparisons tend to be the most helpful in observing good detail preservation; however, good results in edge measures do not always correlate to good noise suppression measurements. This observation bolsters the argument that quantitative IQ analysis must take into account multiple measures. SSIM is also investigated quite extensively; however, understanding its usage is more complicated, due to it already being a combination of several qualities of an image. It gives a result which is an index, based on how much structure is similar. Another variable in SSIM is the amount of Gaussian blur. When looking for edges or fine artifacts, one may prefer a very small amount of Gaussian blur in the SSIM measure. Whereas, when looking at noisy images, measures that are overly influenced by how much noise is in the image, may be quite misleading. Much variability in the SSIM measure is found, therefore, caution and understanding would be advised when using it.

The usefulness of SSIM is found in the following way: Use the SSIM index as a measure for how much structure is available to compare, then measure the SSIM map of the “Golden” image

g with itself,

$$G = SSIM_{map}(g, g) \quad (3.1)$$

and another map of the reconstructed image r with itself.

$$R = SSIM_{map}(r, r) \quad (3.2)$$

Normalize the maps to 1, as in eq. 3.3 and eq. 3.4, subtract the maps, and measure the max and the mean over the difference map, as in eq. 3.5 and eq. 3.6. Remember that the sensitivity of the difference map can be adjusted by setting of the amount of Gaussian blur in the SSIM map calculation.

$$\text{Golden Normalized SSIM Map} = N_1(i, j) = \frac{G(i, j)}{\max(G(i, j))} \quad (3.3)$$

$$\text{Reconstructed Normalized SSIM Map} = N_2(i, j) = \frac{R(i, j)}{\max(R(i, j))} \quad (3.4)$$

$$\text{Max SSIM} = \max(N_1 - N_2) \quad (3.5)$$

$$\text{Mean SSIM} = \text{mean}(N_1 - N_2) \quad (3.6)$$

The max SSIM tells the maximum amount any smaller structure is mis-aligned with the golden image. The mean SSIM gives an overall average measure of mis-alignment and image differences. These measures should be taken with a comparison to the overall SSIM of eq. 3.1 and eq. 3.2 to help understand the significance of the structural differences. For instance, an image without much structural difference does have a very low SSIM. Yet, the max SSIM and mean SSIM measures may be quite high if the small amount of structure is quite different. Due to normalizing the maps, the calculation is a relative measure of difference, compared to the amount of structure available.

Visual inspection is a good way to verify that these measures are being used properly. Some effective visual inspection criteria use comparison of identifiable features, noise, and artifact suppression. These visual inspections of image differences are performed throughout this research. Although, quantitative measures can give clues about IQ, visual analysis is more valuable because ultimately a person is going to read and evaluate these images visually.

Nlevs	Pfilter	Dfilter	SSIM	SNR	Notes
[0,0,0,4,5]	9-7	pkva	0.92	42.52	Previously used settings
[4,5]	9-7	pkva	0.911	41.89	
[4,5,0,0,0]	9-7	pkva	0.914	49.7	Almost critically sampled when 0's at end of nlevs
[3,3]	Sym4	9-7	0.845	52.8	Less noise, spot artifacts, and more detailed features compared to bior4.4 [3 3]
[3,3,0,0,0]	bior4.4	9-7	0.915	52.21	
[3,3]	bior4.4	9-7	0.916	51.63	
[2,0,0,0,0]	db4	9-7	0.911	51.35	
[2,0,0,0,0]	bior3.3	9-7	0.94	40.67	Blurry features compared to other options. These filters and settings measured the highest SSIM.

Table 3.1: Compression to 20% of pixels for comparison between contourlet filter results and settings

3.4 Results and Discussion

Tables 3.1 and 3.2 list compression results from comparing different settings for wavelets and contourlets. Many parameters do not make a great difference; however, using the wrong parameters does cause major differences and greater error. When comparing wavelets to contourlets, there is some difference visually and quantitatively when using the better filters for each transform, see Figure 3.2.

Wavelets appear to have a non-uniform de-noising effect that preserves some detail and smooths others, which produces a “block like” effect and may be less visually appealing. Multi-coil compression did not seem to differ much from single coil compression. Noise is impacted quite differently by different compression schemes and noise in multi-coil images appears more present than in single coil images, and therefore, needs to be considered more thoroughly for multi-coil datasets, see Figure 3.2 d-f.

Wavelet compression consistently performs better than the contourlet in quantitative comparisons of SNR and MSE (see Table 3.3), most likely because there are more contourlet coefficients than wavelet coefficients. For these experiments, a percentage of the original number of pixels

Type	SSIM	NMSE	SNR	Notes
Haar coarse level = 4	0.849	0.000194	44.9	Lower resolution result. Very large blocks in the place of detail.
Beylkin coarse level= 6	0.904	0.000152	45.9	Slightly more noise than others
Vaidyanathan coarse level= 6	0.907	0.000154	46.2	More artifacts are noticeable
Vaidyanathan coarse level= 5	0.92	0.000155	46.3	The change in course level produces a major change in SSIM, but it is not visually apparent.
Coiflet coarse level= 5, par = 3	0.906	0.000135	48.5	Has slight spot artifacts
Daubechies coarse level= 5, par = 8	0.908	0.000139	51.1	A comparison between parameters, par=8 vs par=4, appears to show a decrease in details of some features while the overall image is more smoothed out with less artifacts.
Symmlet coarse level= 5, par = 4	0.908	0.000134	52.7	Preserves more detail and is smoother. Indistinguishable difference from Battle.
Battle coarse level= 5, par=1	0.907	0.000134	54.1	Preserves more detail and is smoother. Indistinguishable difference from Symmlet.

Table 3.2: Compression to 20% of pixels for comparison between wavelet filter results

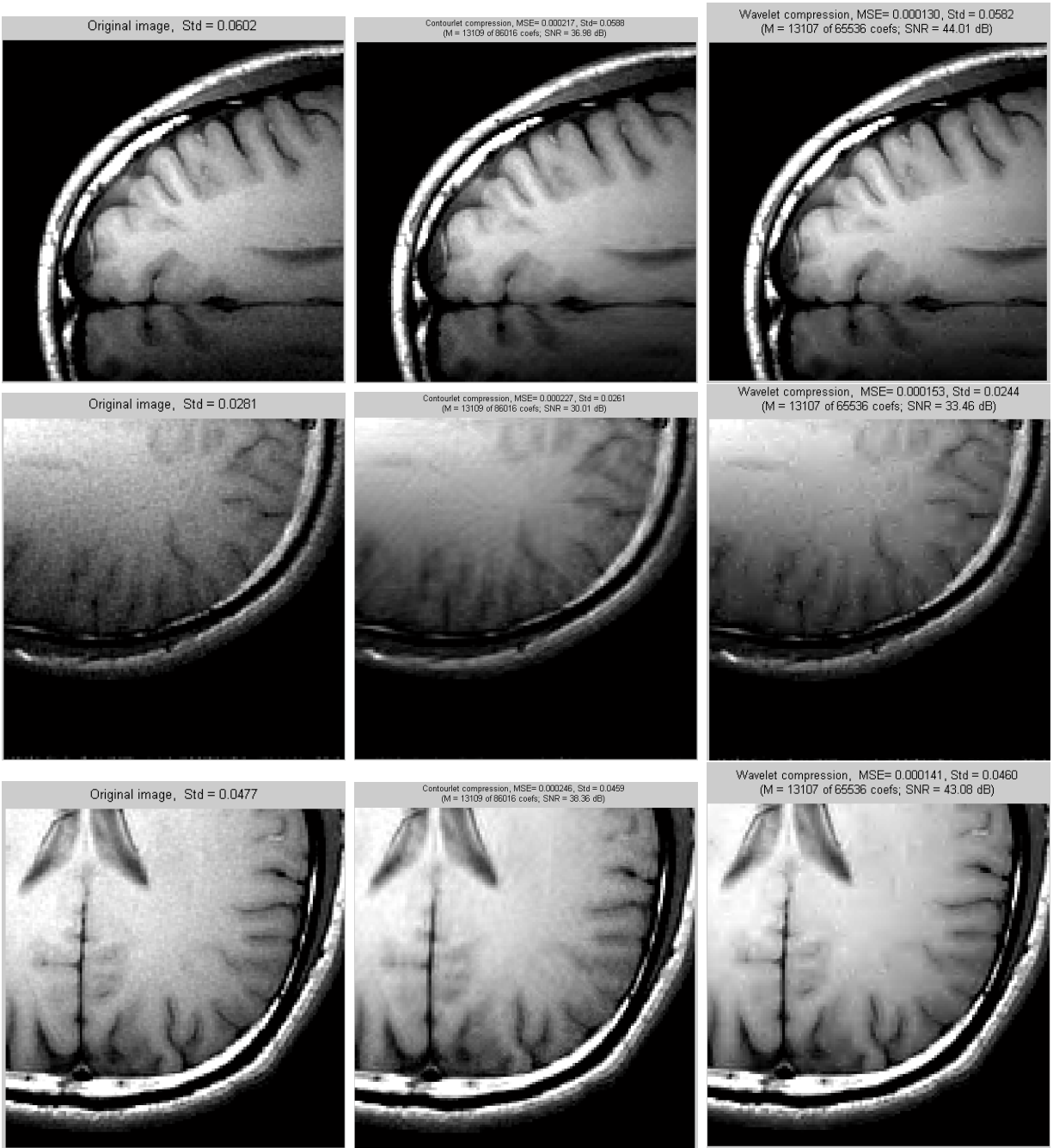


Figure 3.2: Compression keeping 20% of pixels, on three types of images, multi-coil high sensitivity (from left to right and top to bottom)(a-c), multi-coil low sensitivity (d-f), and single coil (g-i). (a) No compression (b) Contourlet compression, (c) Wavelet compression, (d) No compression (e) Contourlet compression, (f) Wavelet compression (g) No compression, (h) Contourlet compression, (i) Wavelet compression.



Figure 3.3: Compression keeping 15% of pixels.(from left to right) (a) Contourlet + wavelet, (b) Wavelet (c) Contourlet.

is chosen as the threshold, rather than a percentage of the number of coefficients. In contourlet compression, everything is smoothed; therefore, some fine details of features are lost. Crisp contourlets are also investigated for their use in compression; however, they consistently performed poorer quantitatively for compression ratios below 40% where major artifacts became apparent. The combination of wavelet and contourlet coefficients did produce good images compared to wavelets or contourlet alone, as seen in Figure 3.3; however, quantitatively, in Table 3.3 the MSE results are poorer. This performance is primarily due to the higher level of truncation performed in these experiments on redundant transforms. When using the combined dictionary, the number of coefficients is over a factor of two times the number of pixels y , therefore, a truncation of 50% would result in over 75% of the coefficients being truncated. When compressing TOF data and generating a MIP, less artifacts could be seen, so greater compression could be achieved. At compression to 10% of pixel values, differences in transforms can be found, as seen in Figure 3.4. An absolute value threshold of pixels produced artifacts at the noise level which are not smooth. This may be improved if a soft threshold is used rather than a hard threshold. Wavelet compression has an interesting streaking artifact where contourlets did not, although, some fine details are lost. In compression tests, the transforms performed better with MIP images than compression of standard body images. Standard images could be compressed to 40% of the data whereas TOF MIP images could be compressed by transform thresholding to about 20% without

Compression Transform	Compression % of retained pixels	STD	NMSE	SNR
No Compression	100	0.0477	0	N/A
Wavelet	60	0.0476	0.000017	56.8
Wavelet	50	0.0474	0.000030	56.4
Wavelet	40	0.0475	0.000050	52.4
Wavelet	30	0.0472	0.000081	51
Wavelet	20	0.0460	0.000141	43
Contourlet	60	0.0471	0.000054	50.8
Contourlet	50	0.0464	0.000077	50.3
Contourlet	40	0.0463	0.000110	45.2
Contourlet	30	0.0459	0.000159	42.5
Contourlet	20	0.0459	0.000246	38.7
Wavelet + Contourlet	60	0.0462	0.000083	49.7
Wavelet + Contourlet	50	0.0457	0.000106	46.4
Wavelet + Contourlet	40	0.0454	0.000139	45
Wavelet + Contourlet	30	0.0453	0.000192	41.3
Wavelet + Contourlet	20	0.0451	0.000289	37.3
CRISP Contourlet	60	0.0476	0.000024	48.2
CRISP Contourlet	50	0.0475	0.000057	44.7
CRISP Contourlet	40	0.0475	0.000131	39
CRISP Contourlet	30	0.0475	0.000404	36.7
CRISP Contourlet	20	0.0503	0.003097	22.7
Curvelet	60	0.0460	0.000200	37.3
Curvelet	50	0.0464	0.000261	35.7
Curvelet	40	0.0461	0.000352	33.6
Curvelet	30	0.0470	0.000492	31.6
Curvelet	20	0.0467	0.000751	29.5

Table 3.3: Compression results

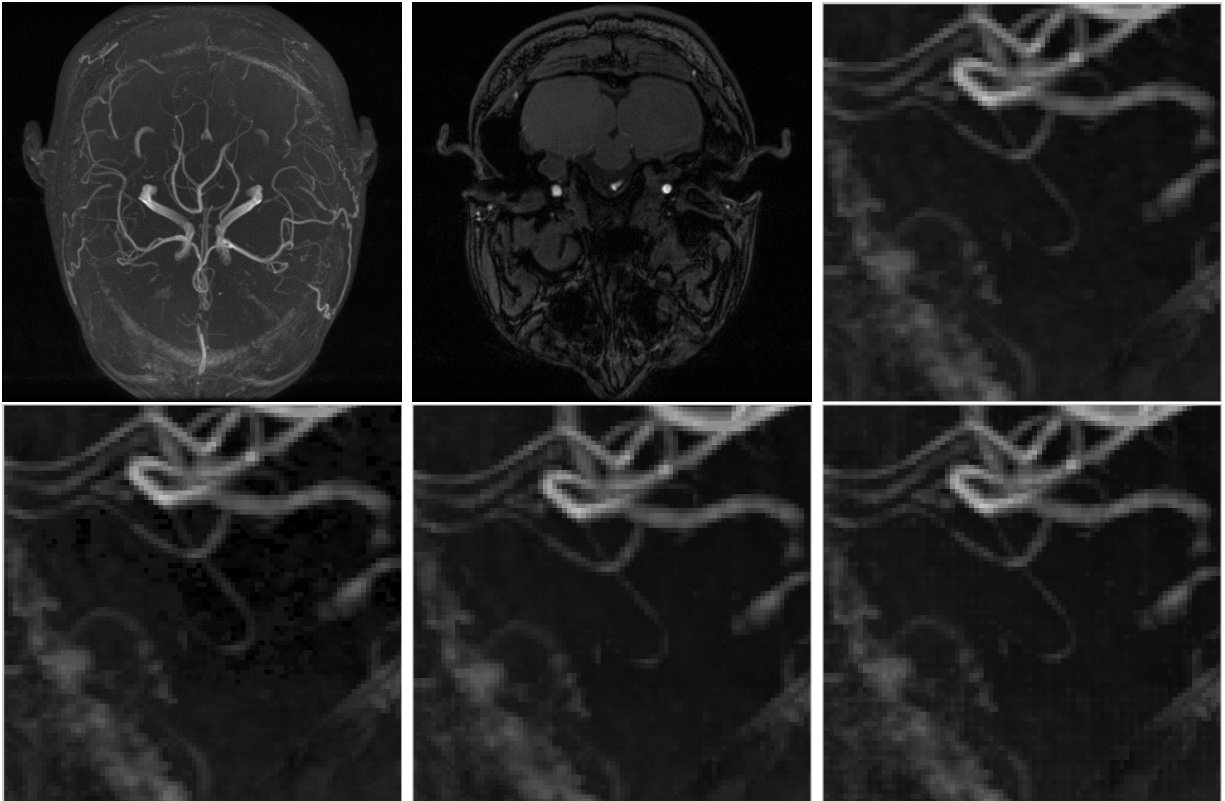


Figure 3.4: TOF MIP detail comparison. (left to right, top to bottom) (a) Fully sampled MIP reconstruction, (b) Fully sampled individual TOF image from which compression is performed, (c) fully sampled detail of MIP, (d) Absolute value threshold, keeping 10% of pixels, (e) Contourlet compression, keeping 10% of pixels, (f) Wavelet compression, keeping 10% of pixels.

visual loss when using contourlets or wavelets.

Chapter 4

Sparsifying Transforms for CS MRI

4.1 Introduction

Linear reconstructions of MRI data that are under-sampled may produce major artifacts because the solution is under-determined. Many sparsifying transforms have been proposed and used for data and image processing applications, including MRI. Sparsifying transforms that apply to image compression focus on non-redundant sparsifying transforms because the goal is compression and extra data is not desired. CS utilizes expansion; it may be that a redundant representation is more applicable, produce sparser coefficients, and have a better reconstruction. Candès *et al.* has generalized the CS RIP theory to utilize potentially redundant dictionaries, D-RIP [13].

4.2 Determining Optimal Artifact Free Sparsifying Transforms for CS

What sparsifying transforms for compression are applicable to CS MRI for optimum IQ? Is it a good idea to have orthogonal wavelets as a standard, which are considered a primary case proposed by Lustig *et al.* in Sparse MRI [40]? They are fast to compute compared to redundant transforms and can be used as a starting point comparison with any other sparsifying transforms. There are orthogonal wavelet implementations readily available in software for Matlab and C++. They do sparsify images and are used in image compression standards like JPEG. However, a problem occurs because orthogonal wavelets leave artifacts at very high levels of compression, as seen in Figure 4.1. These compression artifacts may carry over to CS artifacts, as seen in Figure

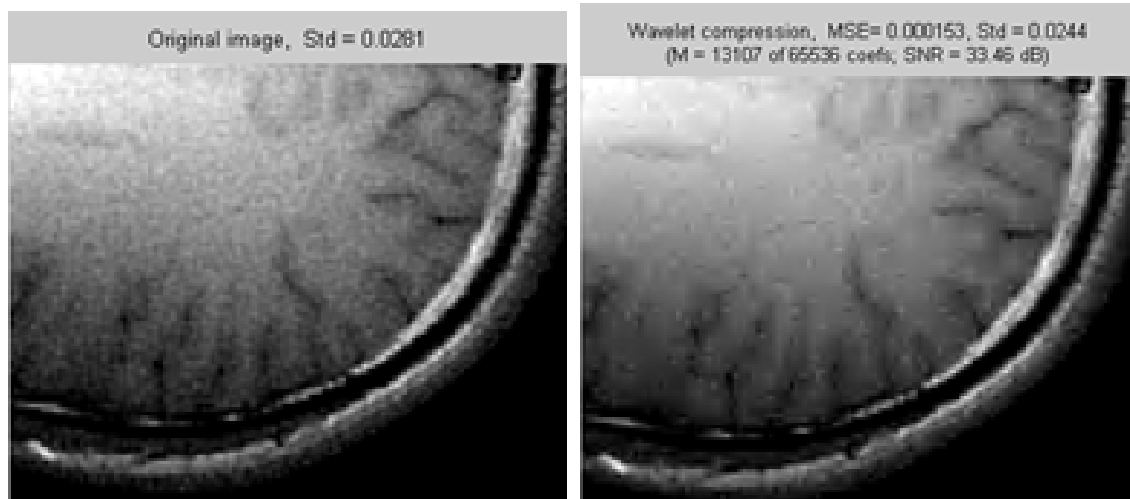


Figure 4.1: (left) Fully sampled original MRI brain image quadrant. (right) Wavelet compression and reconstruction showing artifacts where 80% of the smallest wavelet coefficients are removed by a hard threshold.

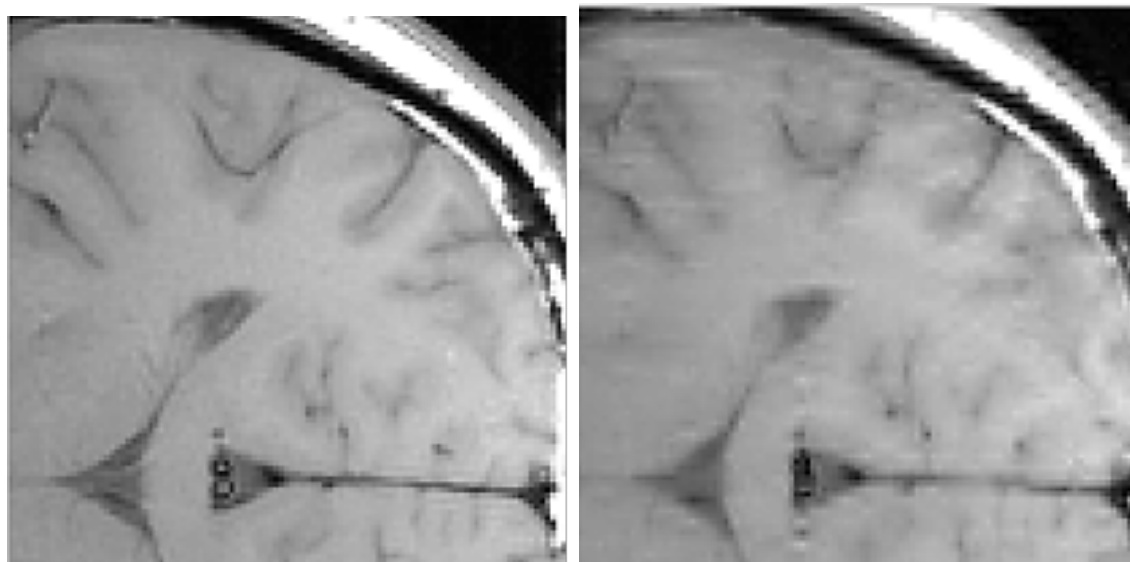


Figure 4.2: (left) Fully sampled original MRI brain image quadrant. (right) CS reconstruction with decimated wavelets showing artifacts where sampled lines of k-space are reduced aggressively by a factor of 3 (only 33% of "full" data used to reconstruct).

4.2.

We also see from Figures 4.1 and 4.2 that de-noising is a natural result of using an orthogonal wavelet sparsifying transform. At first thought, de-noising appears to be a positive effect of using CS, but radiologists may see a de-noised image as problematic and wonder if some structures have been smoothed out. A sparsifying transform ensures that ℓ_1 minimization is regularized and converges, but should de-noising be applied to MRI images? Since it is very important that end users are confident in the results, are there other sparsifying transforms that produce good results both visually and accurately?

What approach should be used for finding the best sparsifying transforms? For example, curvelets, proposed by Candès *et al.*, are an alternative to wavelets and may be better for dealing with curves or edges in a 2D image [19]. Also, according to Lustig *et al.* [40] when considering the CS MRI problem, using just one sparsifying transform is not as effective as combining sparsifying transforms. In Sparse MRI, Total Variation (TV) is used in combination with orthogonal wavelets for better results. What other combinations and transforms are as effective or better?

Redundancy can be achieved in various ways such as from circular shifting the full sized image, or from application to small, overlapping patches. Contourlets are sparsifying transforms that preserve geometries and are redundant, yet have a fast discrete computation[28]. Both “CRISP” contourlets and redundant contourlets are considered [39]. Learned dictionaries have been shown to provide good sparse representations. How much of a benefit can learned (or trained) dictionaries enhance a sparsifying transform? K-SVD is an extension of patches with the Discrete Cosine Transform (DCT) and the K-means algorithm to train a dictionary on the dataset using OMP to sparsify the coefficients[1]. TV is a calculation of the pixel by pixel finite differences in both horizontal and vertical directions across an image [42]. The TIWT is a redundant circularly shifted wavelet calculation that achieves translational invariance, which is lacking in the standard orthogonal wavelet[25].

A Gabor dictionary is used as a sparsifying transform in re-weighted ℓ_1 minimization [16]. One important advantage of a Gabor dictionary is that it is invariant in both time and frequency [31]. In a similar context, a TIWT is used with good results in MRI reconstruction[6]. Invariance

in the ℓ_1 minimization penalty term is advantageous for its ability to represent different signals sparsely.

4.3 Methods

To pinpoint changes between sparsifying transforms, each transform is implemented in either Matlab or FastTestCS. The CS algorithm is also mirrored in both simulation platforms and the transforms are selected individually. Also, only the 1D variable density under-sampling pattern is used for these experiments from Figure 5.2 f. Due to the long computation time of some transforms, they are implemented in FastTestCS from Matlab. The confidence in good sparsifying transforms is achieved through testing fidelity on different data sets. The following sparsifying transforms are tested in this framework.

Full image transforms:

1. Orthogonal Wavelet Transform (OWT)
2. OWT with small redundant patches
3. OWT with large redundant patches
4. Redundant contourlets with full image
5. Non-redundant contourlets with full image
6. Curvelets with full image
7. DCT with full image
8. Translational Invariant DCT with small redundant patches (TICT)
9. TIWT with full image
10. TIWT with small redundant patches
11. TIWT with large redundant patches

12. OMP with small redundant patches
13. K-SVD with small redundant patches
14. TV pixel by pixel finite differences

4.3.1 Gabor Transform Experiments

Comparison measurements of sparsity in Gabor and TV coefficients of various signals are used to verify if good and robust results can be achieved. A selection of 5 different real valued signals and/or images with variants are used to simulate the complexity of practical signals.

- Sinusoidal pulses (GHz range)
- Shepp-Logan phantom image
- Penguin image
- T1 weighted MRI image
- TOF MIP MRI image

The original images and signals are sized to be a total length of $16k$ values. For images, this is a 128×128 gray-scale image, shown in part (a) of Figures 4.11 through 4.18. These signals are not sparse in their native domain, but can become sparse when a transform is applied.

The use of the Gabor dictionary to reconstruct these signals utilizes a core optimization algorithm solver for the primal-dual interior point method provided by ℓ_1 -Magic [21]. The redundancy of the Gabor transform coefficients compared to the original signal is about 43 times. These results are compared to reconstructions of the same input signal using another CS optimization algorithm solver, TV minimization with quadratic constraint, which is also provided by ℓ_1 - Magic. These algorithms operate on the full sized image (or signal) without breaking up the data into segments. The under-sampling is performed by using a pseudo-random Gaussian measurement matrix, similar to Candès *et al.* [13], with a factor of 50%. The measurement matrix in these

Gabor experiments is quite different from a typical MRI trajectory that would have a binary representation which samples frequency points.

A Linear (L) MSE of the reconstruction is used as a reference, shown as (b) in Figures 4.11 through 4.18. This is calculated by the transpose of the measurement matrix as the pseudo inverse. Gabor (G) MSE identifies the error in the use of that dictionary in the CS reconstruction, shown in parts (c) of the figures. TV MSE is a measure of the error when TV weighted ℓ_1 minimization is performed and is shown in part (d) of each figure.

Sparsity measurements taken in the coefficient domain are based on a ratio of the count of values that are less than $\frac{1}{256}$ of the maximum coefficient, divided by the total number of the coefficients. This ratio then is a percentage of sparsity. Two sparsity measures are taken:

- % G Sparse - Sparsity of the Gabor transform coefficients of the fully sampled signal
- % TV Sparse - Sparsity of the TV calculation of the fully sampled signal

4.4 Results and Discussion

Based on compression test comparisons, one can optimize for settings and filters in different sparsifying transforms and identify good IQ, as seen in chapter 2. The use of the compression experiments help inform generally good transforms, however, those results did not directly correlate to the perfect transforms for use in CS. Because of this finding, one must additionally compare sparse transforms with a CS simulation to determine the best transforms.

Table 4.1 shows results of some of the best filters and settings for wavelets, curvelets and contourlets. The baseline for comparison is the non-optimized row indicated where no sparsifying transform is used, but rather the non-linear optimization is still performed. The best filter and settings for each transform are chosen from this table for the other comparison tests.

Some similar artifacts seen in compression do show up in CS image results when using the same sparsifying transforms. For instance, a blocky image artifact appears when using decimated wavelets for CS, in much the same way when using too much wavelet compression in an image reconstruction. To counter that, redundant transforms with TI qualities produce better IQ in many

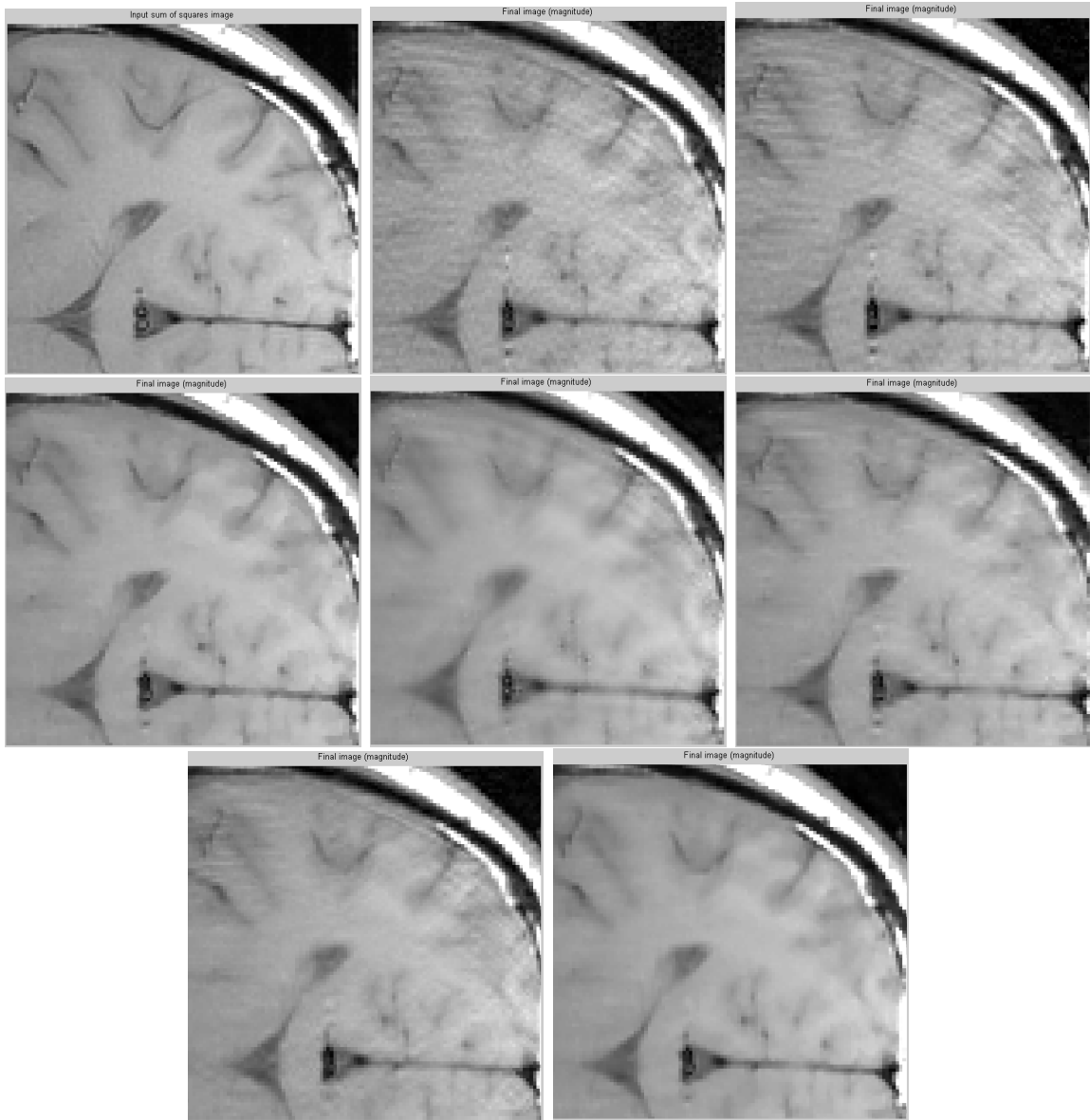


Figure 4.3: Unconstrained CS results with reduction factor of 33% (left to right and top to bottom) (a) Original image, (b) Without regularization, (c) With only ℓ_1 -norm regularization, (d) With only TV (e) With only curvelet regularization, (f) With only wavelet regularization, (g) With only contourlet regularization, (h) All combined regularization.

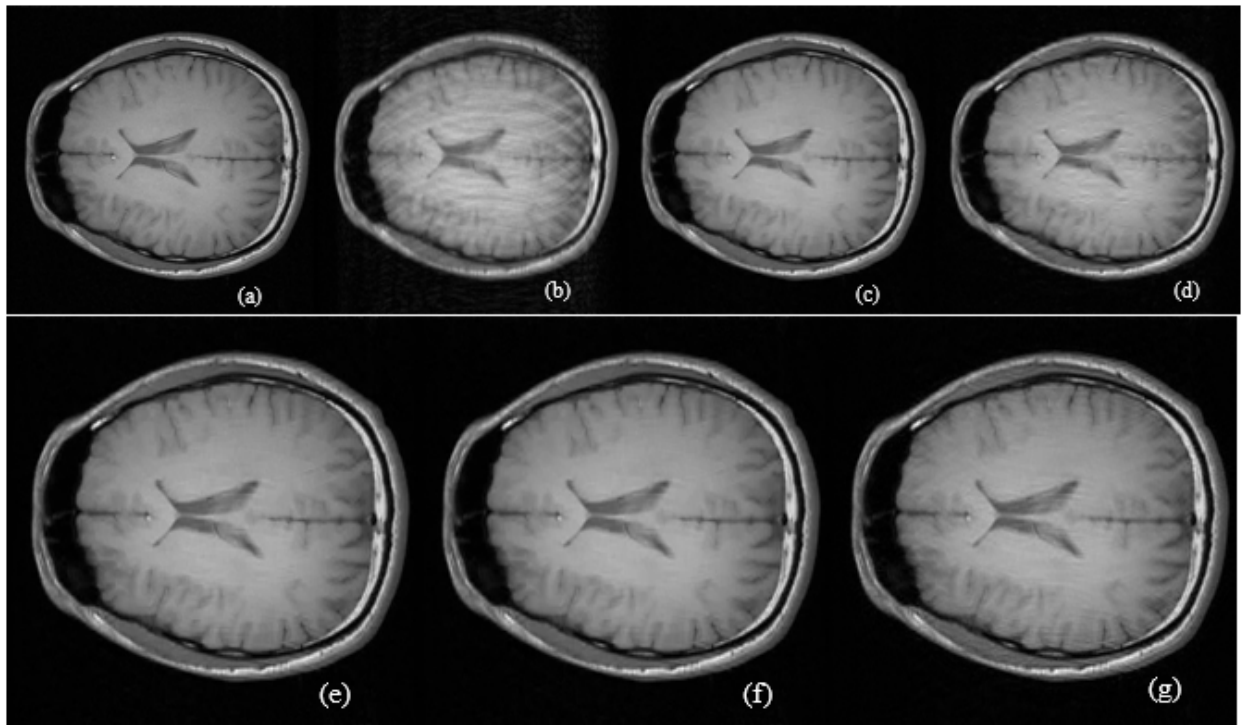


Figure 4.4: CS reconstruction with reduction factor of 30% (left to right and top to bottom) (a) Fully sampled (b) Linear reconstruction (c) K-SVD (d) OWT, (e) Over-complete TICT, (f) TICT, (g) OWT on small patches.

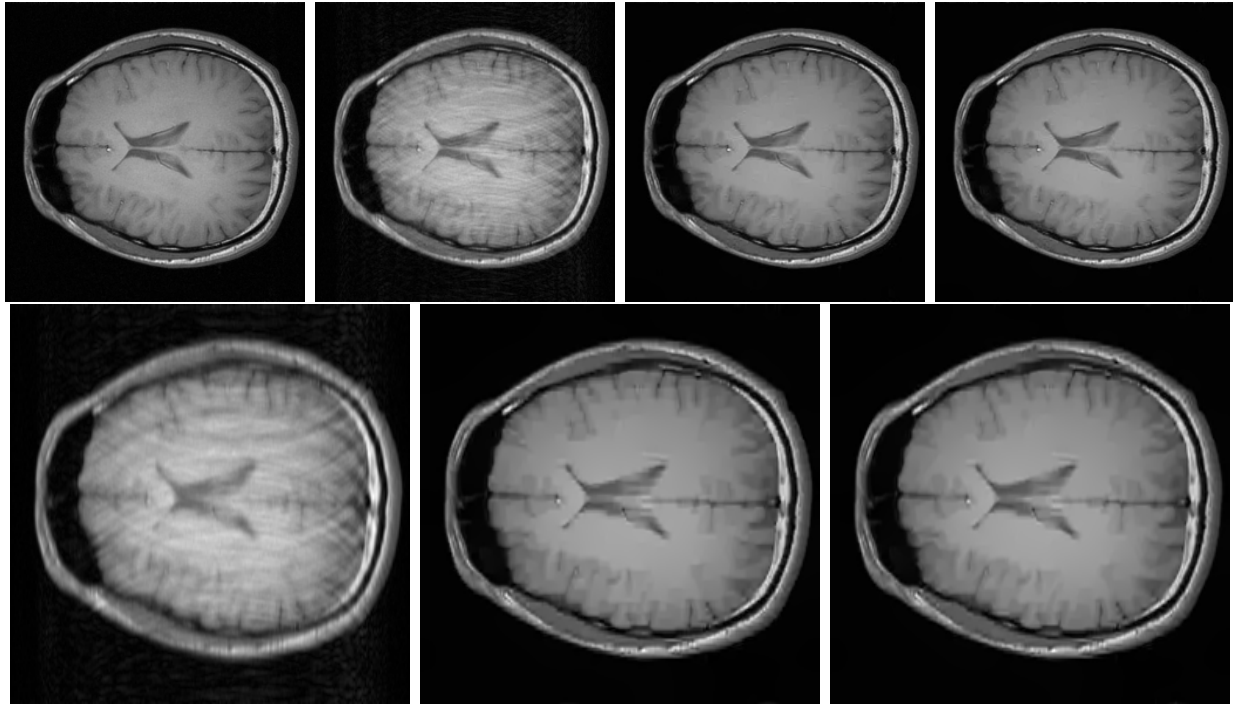


Figure 4.5: CS reconstructions, (a) Single channel T1-weighted brain fully sampled, (b) Initial guess linear reconstruction, reduction factor of 50%, (c) Reconstruction with an over-complete TICT untrained dictionary and a reduction factor of 50%, (d) Reconstruction with a trained dictionary (K-SVD) and a reduction factor of 50%. (e) Initial guess linear reconstruction, reduction factor of 20%, (f) Reconstruction with an over-complete TICT untrained dictionary and a reduction factor of 20%, (g) Reconstruction with a trained dictionary (K-SVD) and a reduction factor of 20%.

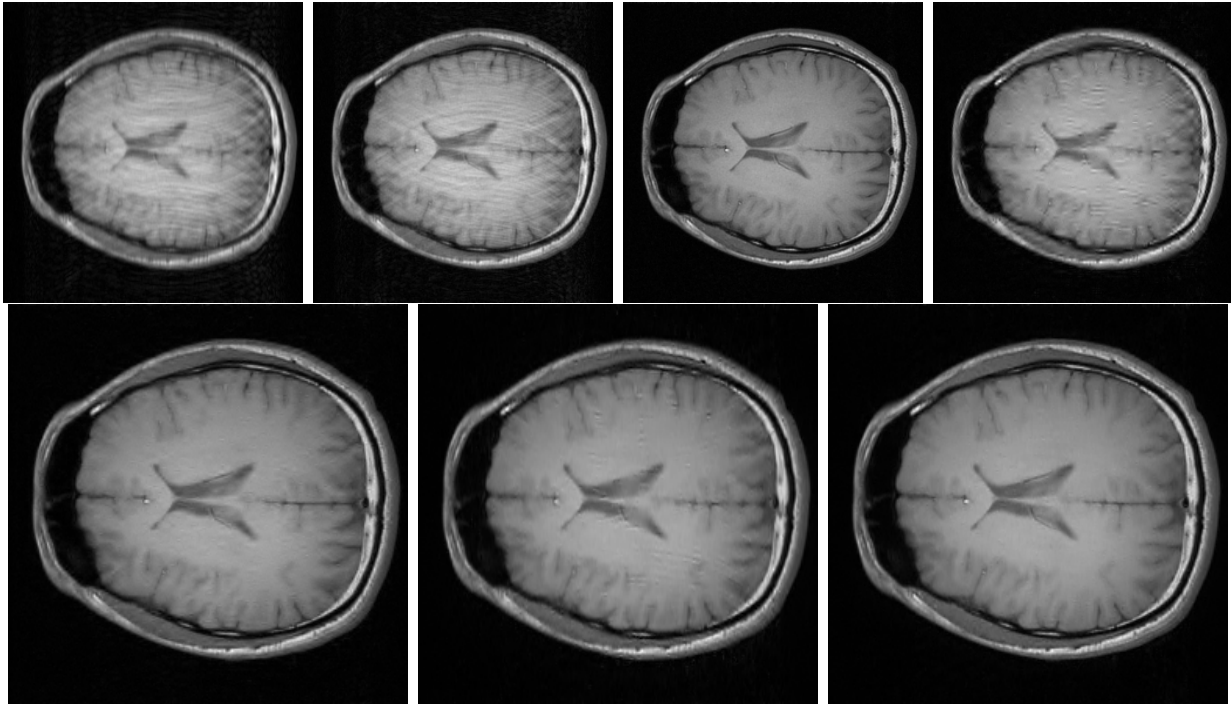


Figure 4.6: Single-channel T1-weighted brain MRI data (a) Linear reconstruction, reduction factor = 25%, (b) Linear reconstruction, reduction factor = 50%, (c) Fully sampled, (d and e) CS reconstruction with wavelet sparsifying transform on full image representation, reduction factor = 25% and 50%, respectively, (f and g) CS reconstruction with wavelet sparsifying transform on small and redundant patch representation, reduction factor = 25% and 50%, respectively.

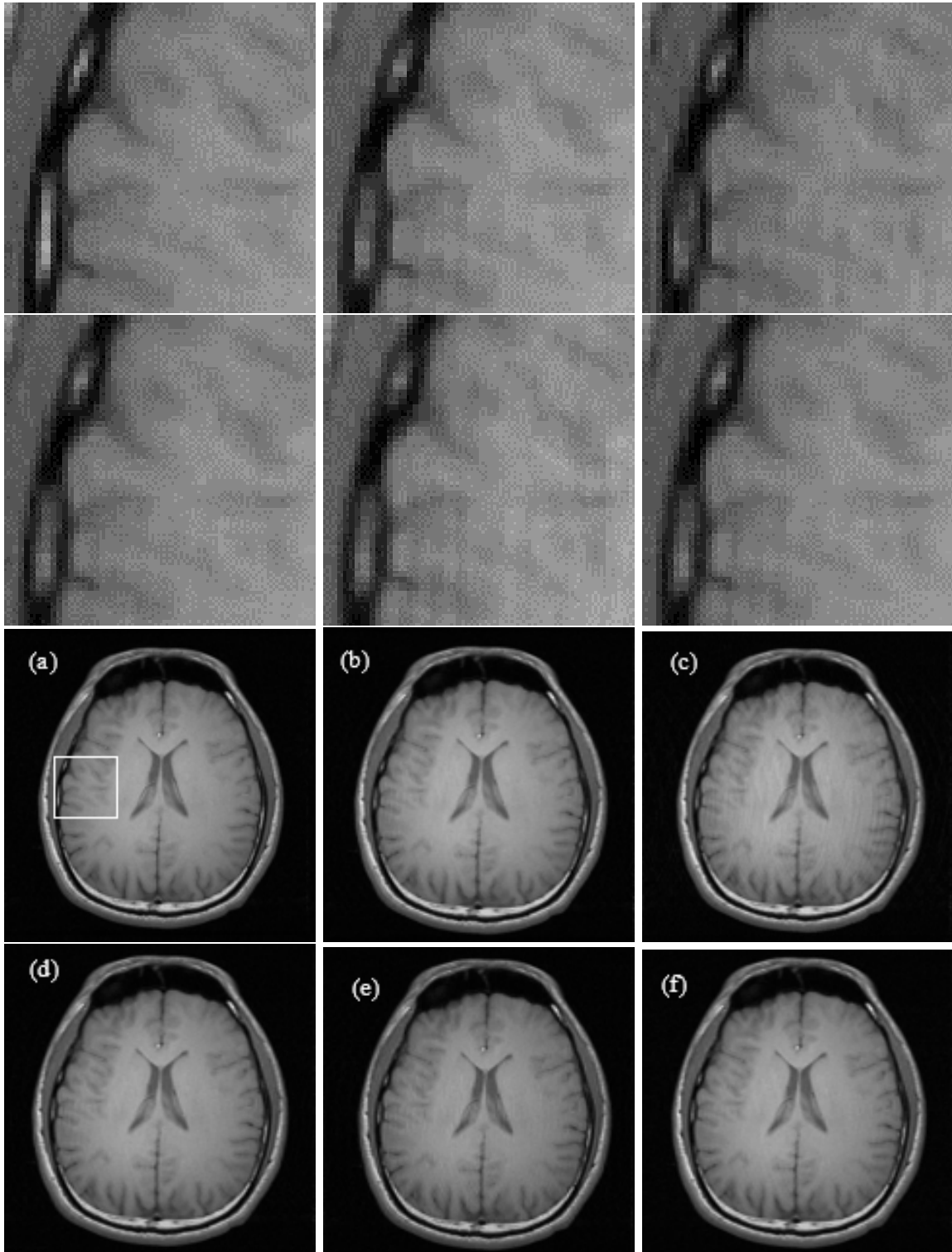


Figure 4.7: CS reconstructions with a reduction factor of 40% (left to right and top to bottom, with the corresponding enlarged region shown above) utilizing different sparsifying transforms as regularization (a) Fully sampled reconstruction and (b) Reconstructions with TV, (c) DCT, (d) TICT, (e) WT, and (f) TIWT.

Image Error vs Reduction Factor

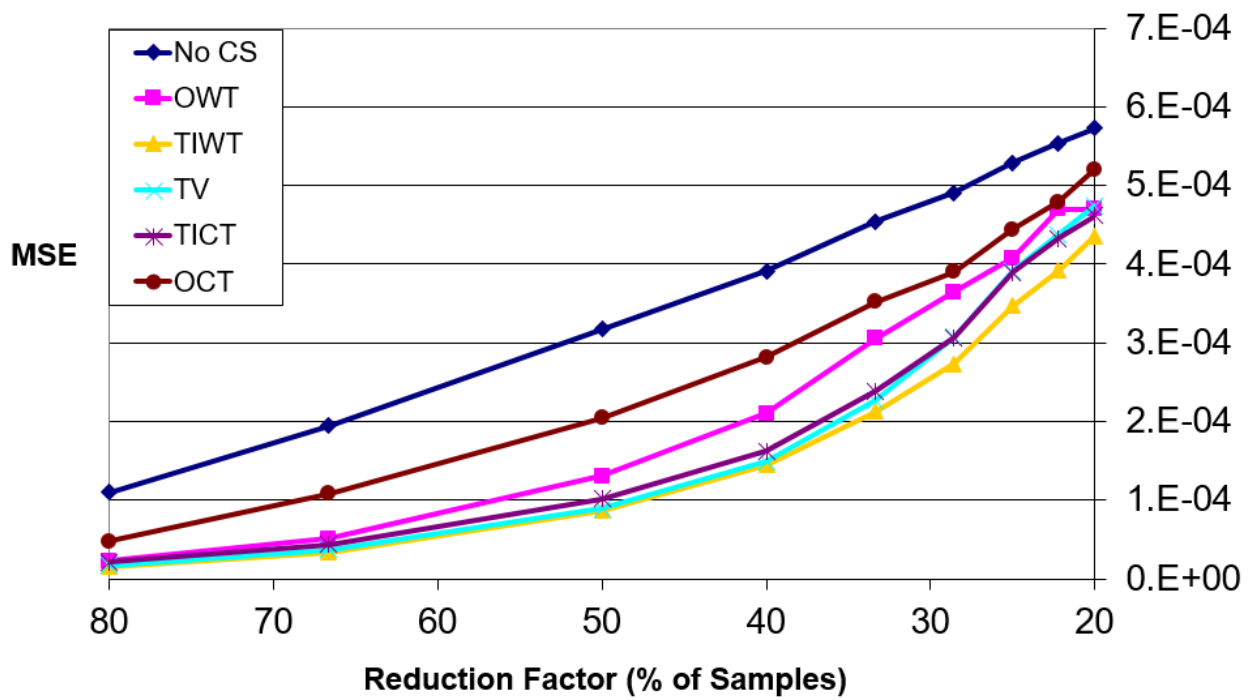


Figure 4.8: Comparison of linear reconstruction error (non CS) and CS reconstruction errors (NMSE) with different sparsifying transforms: TV, orthogonal DCT (OCT), TICT, orthogonal WT (OWT), and TI WT as the sparsifying transforms. Each point is an average of 10 results.

Type	Settings	SNR	SSIM	Notes
Non-optimized	NA	23.5	0.93	All artifacts and noise are present
Daubechies Wavelet	par = 4, L = 3	20.92	0.9494	A few more artifacts than the other wavelet types
Symmlet Wavelet	par = 4, L = 1	21.09	0.9509	A few more artifacts than the contourlet cases
Battle Wavelet	par = 1, L = 2	21.02	0.949	Good IQ
Curvelets	Real, defaults	20.35	0.9469	Good IQ
Curvelets Highest SNR	Real, Finest = curvelets, nbscales = 2, nbangles coarse = 16	21.03	0.938	More artifacts and less smooth compared to other curvelets
Curvelets High SSIM	Real, Finest = curvelets, nbscales = 5, nbangles coarse = 8	20.92	0.9502	Good IQ
Contourlets nlevs = [1 1 1 1]	pfilter = sym5, dfilter = pkva12	20.8	0.9474	Less artifact result compared to other contourlet filters, wavelets and curvelets
Contourlets nlevs = [1 1 1 1]	pfilter = db4, dfilter = pkva 12	20.76	0.9466	Good IQ
Curvelets	Complex, defaults	19.24	0.934	Much smoother and less detailed edges

Table 4.1: Comparison of sparsifying transform parameters in CS reconstruction

areas: Artifact suppression, noise reduction, detail preservation, and statistical quantification. TI transforms are more robust with less variance from run to run with different random sampling patterns.

Figure 6.4 shows that TV does contribute the most to an improved sum of error calculation. Visually, the artifacts seen in Figure 4.3.b appear to be reduced the most by TV regularization, seen in Figure 4.3.d. The other transforms (Figures 4.3.e,f,g) produce varying degrees of quality. The Wavelet regularization has fewer ring artifacts than the Contourlet or Curvelet regularization. The Curvelet regularization is much smoother than the Contourlet and Wavelet regularization. The Contourlet regularization appears to preserve some line features better, but more artifacts remain. The ℓ_1 -norm regularization did not contribute to improve the artifacts; however, some noise is reduced, as seen in a comparison of Figures 4.3.b and 4.3.c. A small additional improvement in MSE can be found when combining multiple regularization techniques with the appropriate lambda. This lambda will not be the same as the minimum lambda found in Figure 6.3 because lambda becomes a balancing parameter between the other regularization techniques. Finding the optimum lambdas for multiple regularization was not performed, therefore better results than Figure 4.3.h may be achievable.

Another group of CS reconstruction results involve transforms performed on small and overlapping patches of the image. The K-SVD learning algorithm is applied to these patches which have the benefit of TI. The smaller patch sized matrices also allow OMP and SVD to operate with more feasibility. The transforms tested are K-SVD, DCT, over-complete DCT, and wavelets. These results show an improvement over full image transforms in the CS reconstruction simulation. Figures 4.4 through 4.8 show comparisons of a wavelet reconstruction using the full image with different TI and patch transforms. The reduction factor is from 20% to 80%, the patch size is 4×4 , the redundancy factor of the dictionary for K-SVD and the over-complete DCT is 4, and the number of iterations is determined by a MSE convergence and visual inspection. These patch transform results show an improvement over the wavelet transform on a whole image by visually comparing Figure 4.4.d with c,e,f and g. When comparing KSVD with the training step of that algorithm removed, the reconstruction IQ is not substantially degraded. This leads to a

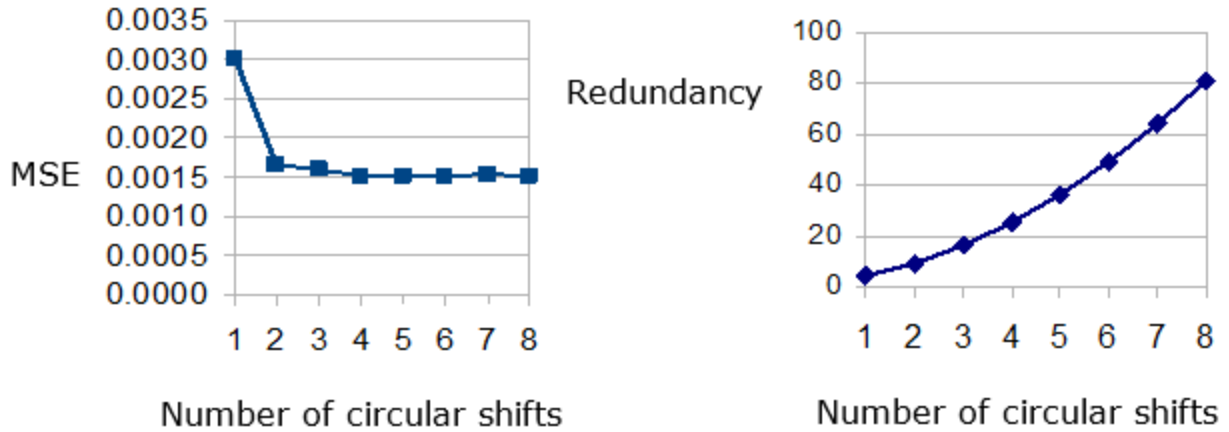


Figure 4.9: IQ (NMSE) vs redundancy. Patch size = 256×256 , 2D circular shift of 6 rows/columns, reduction factor = 2

valid trade off, if it is necessary to include the training step of KSVD when less computation time is desired.

In whole image comparisons, contourlets and wavelets perform equivalently. In general, over-complete dictionaries are better than critically sampled dictionaries, as seen from these results. With that in mind, the TIWT is one transform that performs well, and is moderately redundant. This is an advantageous middle ground between having too much redundancy and too little. With too much redundancy, the transforms take too long to perform. With too little, quality is compromised. With the TIWT, the implementation is faster and deserves more consideration because it may be more computationally feasible.

The TICT is a good sparsifying transform, therefore, it is compared with other transforms and settings. Measurements and comparisons are made for levels of redundancy. IQ vs sliding distance is measured by MSE and shows that IQ worsens as sliding distance increases. The sliding distance is the number of pixels shifted between overlapping patches. Both small patches and large overlapping patches are compared. Small patches are image portions from 4×4 to 16×16 pixels, either square or rectangular. Large patches are full sized images (256×256) that overlap by circularly shifting edge rows and columns. With large patches, a certain level of redundancy in overlapping patches is beneficial, but increasing redundancy is less impacting, as seen in Figure 4.9.

This result clearly shows that redundant sparsifying transforms perform better than orthogonal transforms. With the DCT being improved, other transforms may also be improved. When the orthogonal wavelet transform is implemented with patches, it performs even better. Overlapping patches with OMP provide a great way to improve CS and enhance IQ because OMP does operate efficiently on smaller data sets. The patches provide a good level of translational invariance, while also providing a way for parallelization which takes advantage of programming parallel threads. In conclusion, patches open up a way to improve many different sparsifying transforms.

In MRI image CS tests, several image types that give a basic broad coverage are tested: A single channel 256×256 , a multi-channel 256×256 , a multi-channel MIP, and multi-slab 3D. Simulated images are also explored; they are created in Matlab and randomly varied in complexity and type, see Figure 4.10. Sixty different images are created starting with a few small dots of many large circles of varying intensity and size. The CS MRI problem is tested with varying under-sampling factors and image sizes. Measurements of IQ are done with different sparsifying transforms. The results are interesting, due to extreme changes in the image. No single transform could consistently perform well across all images. This is most likely due to the different types of structures and changing sparsity in the images.

With each image, several under-sampling factors are tested. Generally, under-sampling greater than 66% reduces IQ to non-diagnostic; however, this is a subjective visual measure. Some under-sampling schemes and transforms minimize artifacts better than others with higher under-sampling. Measuring several levels of reduction factors gives confidence in transform robustness.

4.4.1 Gabor Transform

The goal is to show how well a sparse tight frame representation of various signals performs in CS reconstruction. Analysis is done of the Gabor dictionary as a sparsifying transform on non-sparse signals and images. A large range of reconstruction errors and sparsity levels are observed for different image types and signals. The use of the Gabor frame is contrasted with TV weighted ℓ_1 minimization in CS reconstruction. Table 4.2 quantifies MSE and sparsity for each reconstruction test. Figures 4.11 - 4.18 are the corresponding image and signal reconstructions.

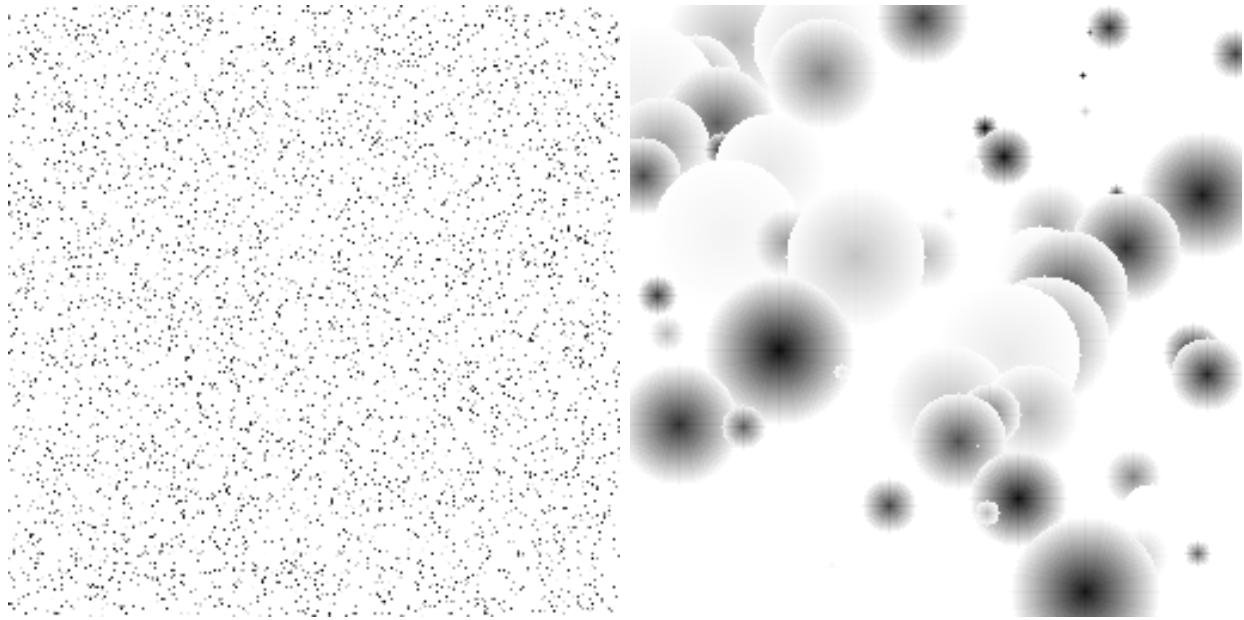


Figure 4.10: Synthetically generated images, varying numbers of points or circles or sizes can be adjusted to increase or decrease image complexity for rerunning simulations with different images

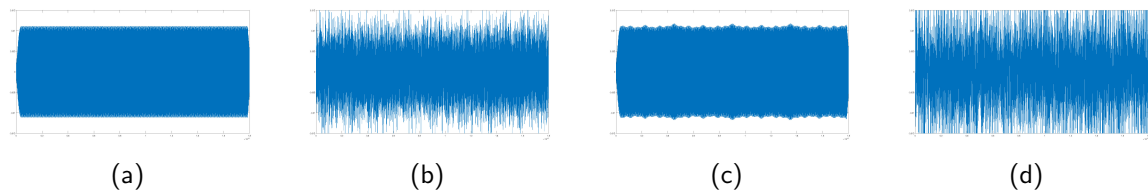


Figure 4.11: Signal reconstruction test 1, one long pulse (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

According to these measurements, sparsity in the coefficient domain will correlate to image reconstruction success. For example, test 1 measures a Gabor coefficient sparsity of over 99% and a reconstruction success which reduces the MSE by 36 times compared to the linear reconstruction. Whereas, with the same signal, which is not sparse at all in the TV domain, TV minimization actually increases the MSE when compared with the linear reconstruction, see Figure 4.11. It is important to note that the sparsity calculated on the Gabor coefficient set is on a much larger set of redundant coefficients than the non-redundant TV coefficients.

In test 2, the complexity and sparsity are adjusted by adding additional sinusoidal pulses which may overlap. The complexity of the pulses significantly increases to twenty pulses and the Gabor dictionary is able to sparsely represent the signal very well compared to TV minimization, see

Test	L MSE	G MSE	TV NMSE	%GSparse	%TVSparsity
1 (Pulse 1)	0.7045	0.0195	0.9411	99.5	0.19
2 (Pulse 2)	0.7075	0.0192	0.8023	94.8	2.6
3 (Shepp-Logan)	0.7062	0.2697	0.0	54.7	91.7
4 (Penguin)	0.7077	0.3011	0.0	54.1	87.6
5 (Pulse + Shepp-Logan)	0.7101	0.2541	0.2141	54.4	2.2
6 (T1 MRI)	0.7013	0.1445	0.0637	82.9	7.8
7 (TOF MRI)	0.7012	0.2555	0.1369	68.4	13.3

Table 4.2: Sparsity and CS reconstruction errors of various signals (L - Linear, G - Gabor, TV - Total Variation, MSE - Mean Square Error)

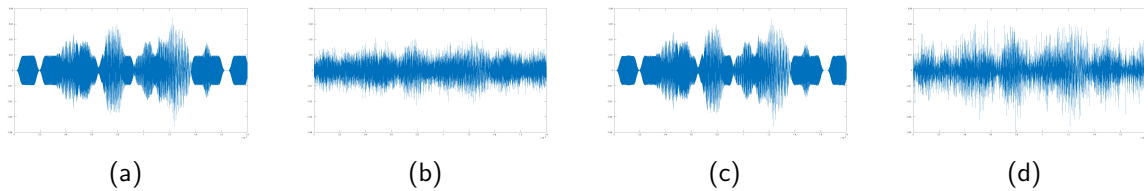


Figure 4.12: Reconstruction test 2, 20 short pulses (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

Figure 4.12. Similar to the results in test 1, the reconstruction MSE is reduced by 36 times.

In tests 3 and 4, images are chosen which are sparse in the TV domain but not in the Gabor domain. The TV reconstruction reduces the MSE to zero, compared to the Gabor reconstruction reducing by only a factor of 2.6 and 2.4 respectively. The penguin image is an example with a different background magnitude from the Shepp-Logan phantom, see Figures 4.13 and 4.14.

The signal in test 5 is a combination of a pulse from test 1 with the Shepp-Logan image from test 3. The same signal is plotted in the time domain for Figure 4.15 and in the image domain for Figure 4.16. Although both Gabor and TV CS reconstructions improve the result over the linear calculation, the error still remains quite large. It is noteworthy that the sparsity percentages are much lower for this case in both the TV and Gabor domains. This underscores the important connection between having a sparse representation and making a good reconstruction.

In the last experiments, tests 6 and 7, MRI images of the brain are either T1 weighted or TOF MIP. They appear not to be sparsely represented well in either the Gabor domain or in TV. The

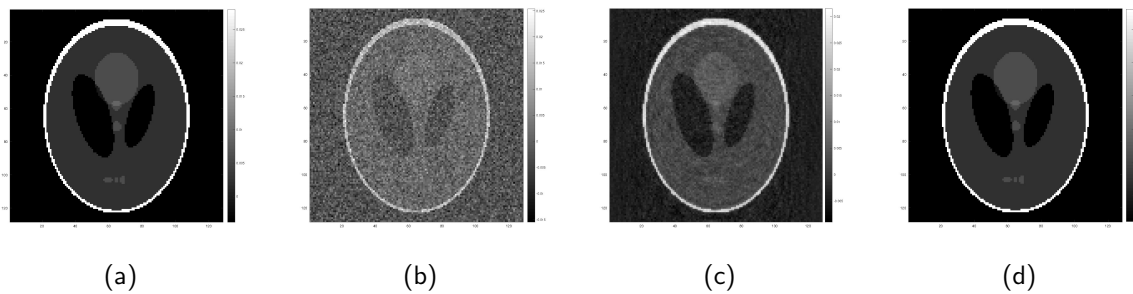


Figure 4.13: Reconstruction test 3, Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

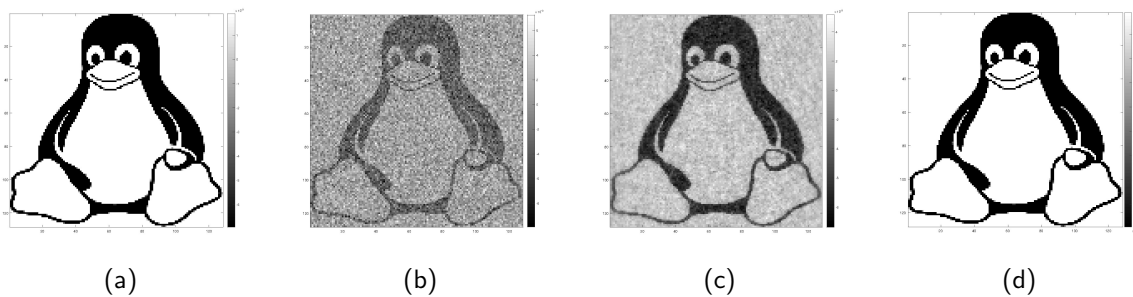


Figure 4.14: Reconstruction test 4, penguin (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

MSE result is poor in both reconstruction algorithms, see Figures 4.17 and 4.18. It is important to note that under-sampling is in the image domain and not in the native MRI signal domain of k-space. However, this is still an equivalent comparison for cases when the under-sampled k-space produces artifacts that are incoherent, as in this experiment. A requirement of CS reconstruction is that artifacts due to sampling are similar to uniform noise with an even distribution across the image.

It is also important to point out that the linear reconstructions, calculated with a pseudo-inverse, have a consistent MSE for all experiments, see L MSE in Table 4.2. This is not the case for this tight frame. These findings show remarkably good results for some periodic signals. The ability of the sparsifying transform to produce a high percentage of sparsity contributes greatly to the reduction of reconstruction error. However, the Gabor tight frame does not appear to be advantageous for the images investigated here. When using CS in these cases, it is vital to pick a dictionary which will effectively and sparsely represent the signal of interest.

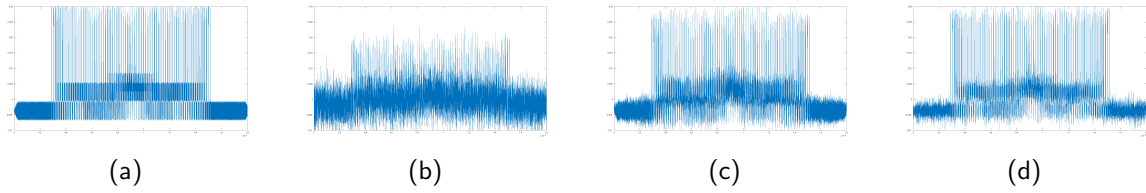


Figure 4.15: Signal reconstruction test 5, one long pulse + Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

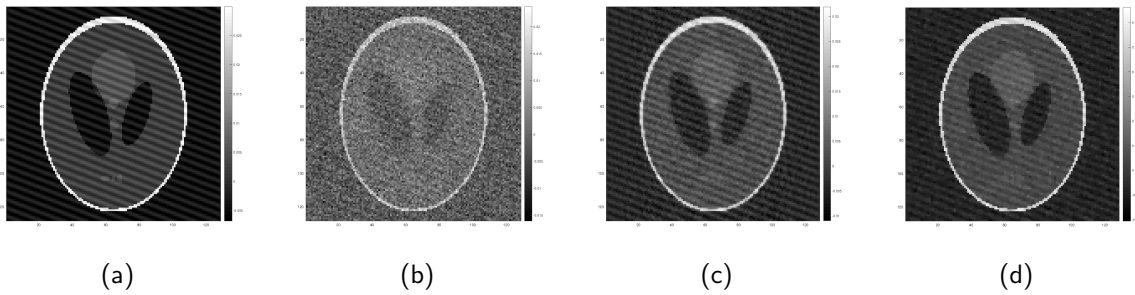


Figure 4.16: Image reconstruction test 5, one long pulse + Shepp-Logan phantom (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

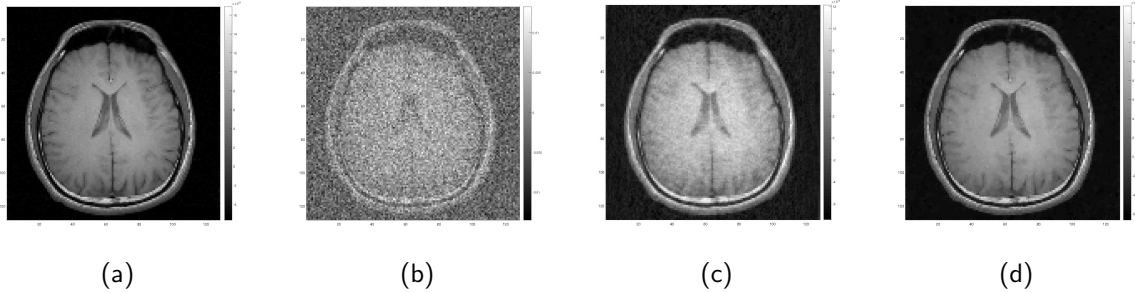


Figure 4.17: Reconstruction test 6, T1 MRI, (a) Original , (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

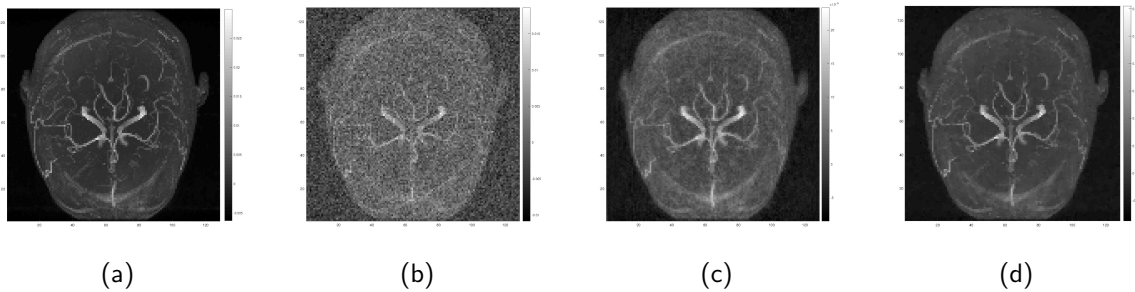


Figure 4.18: Reconstruction test 7, MIP MRI, (a) Original, (b) Linear pseudo inverse, (c) CS Gabor, (d) CS TV

Chapter 5

MRI Sampling Trajectories for CS MRI Simulation

5.1 Introduction

One of the main ingredients of CS for MRI is correct under-sampling. Candès *et al.* and Lustig *et al.* state a CS requirement of incoherence between the under-sampling domain and the sparse representation domain, [15] and [40]. This means that under-sampling in k-space must have minimal artifacts that are incoherent in the linear image reconstruction. A primary aspect of this requirement is determining the best under-sampling technique. Using a uniform random under-sampling matrix could seem appropriate, however, in MRI, k-space packs a majority of discernible frequency content at its origin which trails off at its periphery. Additionally, just sampling individual points of k-space randomly is non-standard for MRI and may not be as efficient. Sampling is costly in time, therefore, determining the shortest sample trajectory path is desired.

Under-sampling patterns for use in CS MRI are developed for comparing Total Variation (TV) with the TIWT sparsifying transforms. The six different under-sampling patterns analyzed are: Uniform random, 1D variable density with large and small center, 2D variable density with and without fully sampled center, uniform radial, and logarithmic spiral. The process of developing and comparing these different measurement matrices also involves processing a number of permutations. Through checking several types of applicable under-sampling patterns in simulation, the goal of proving the robustness of the TIWT is established. This provides proof that it can

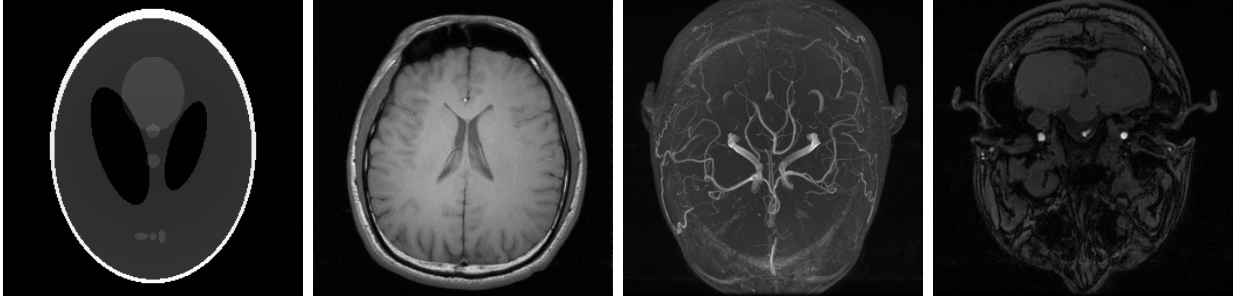


Figure 5.1: Fully sampled reconstruction simulation images: (a) Shepp-Logan phantom 256×256 (b) T1 MRI image 256×256 (c) TOF MRI MIP image $192 \times 512 \times 512$ 8 channel (d) TOF MRI (one slice) image 512×512

enhance the reconstruction quality of the CS MRI result, therefore, enabling the use of less MRI data to reduce acquisition time.

5.2 Methods

5.2.1 MRI Under-Sampling Patterns Developed

The radial under-sampling technique is calculated in a circular pattern, see Figure 5.2.a and equations 5.1 and 5.2. Each line has a unique angle a of π/L . Where L is the number of lines, N is the length of one dimension of the sampling space and n represents a point on that line.

$$x(a, n) = \begin{cases} \tan(a)(-\frac{N}{2} + n) + \frac{N}{2} & \text{if } a \leq 90^\circ, a > 270^\circ, \\ n & \text{if } 90^\circ < a \leq 270^\circ. \end{cases} \quad (5.1)$$

$$y(a, n) = \begin{cases} n & \text{if } a \leq 90^\circ, a > 270^\circ, \\ \cot(a)(-\frac{N}{2} + n) + \frac{N}{2} & \text{if } 90^\circ < a \leq 270^\circ. \end{cases} \quad (5.2)$$

The spiral technique presented here is logarithmic, meaning it will grow from the origin as the distance increases, see Figure 5.2.b and equations 5.3 and 5.4. Where a and b are constants that control the radius and expansion of the spiral and t is a point along the spiral

$$x(t) = ae^{bt} \cos(t) \quad (5.3)$$

$$y(t) = ae^{bt} \sin(t) \quad (5.4)$$

Several random distributions are utilized; a basic uniform random under-sampling is used, as in 5.2.c. The variable density random under-sampling in Figure 5.2.d uses a 2D radial distance x

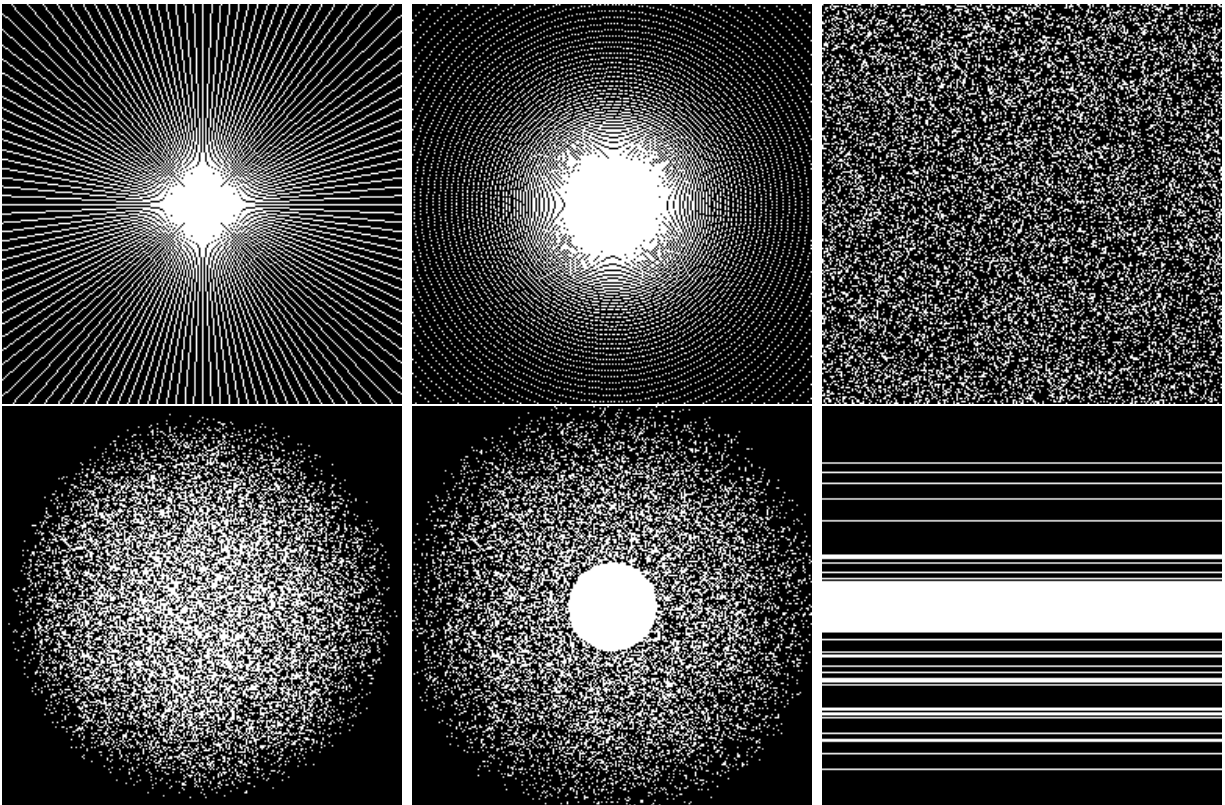


Figure 5.2: Under-sampling patterns at 25%: (a) Radial with 70 lines (b) Spiral (c) Random (d) 2D variable density (e) 2D variable density with 25 point fully sampled center radius (f) 1D variable density with 32 line fully sampled center

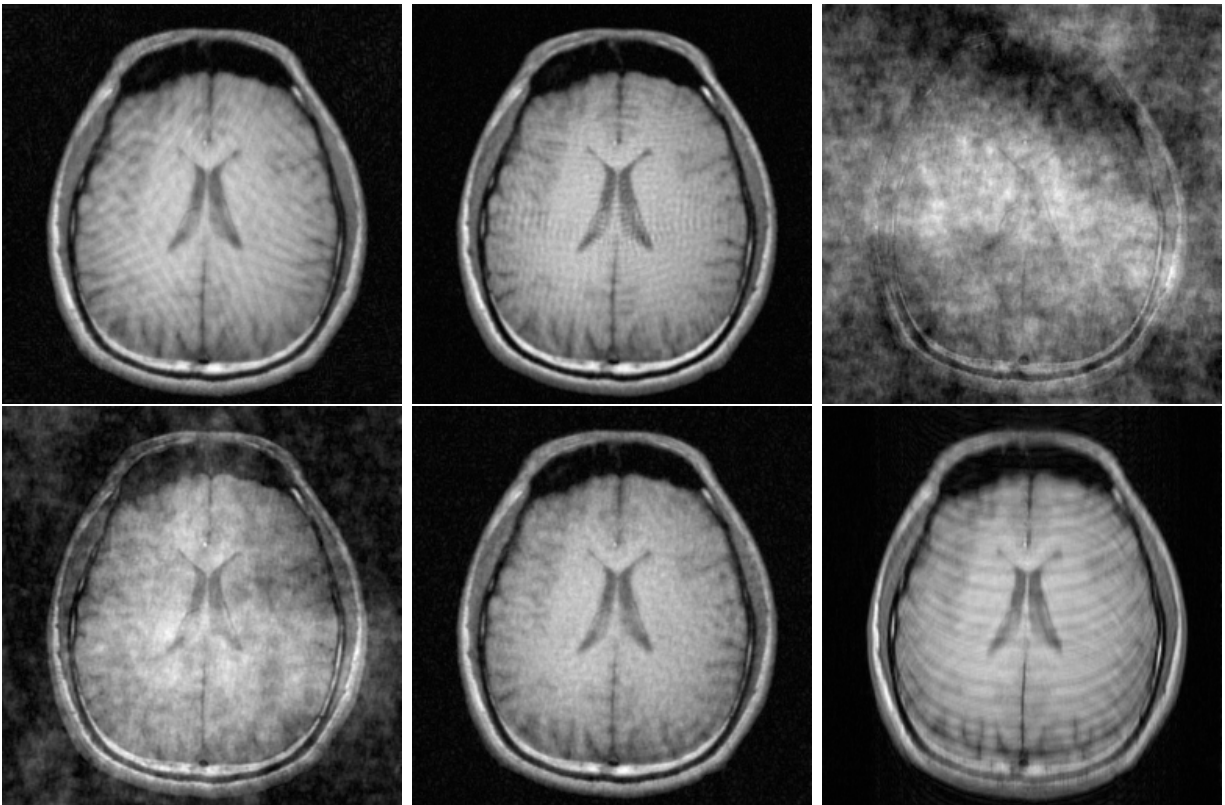


Figure 5.3: CS reconstructions of T1 MRI with 25% samples (a) Radial (b) Spiral (c) Random (d) 2D variable density (e) 2D variable density with 25 point fully sampled center radius (f) 1D variable density with 32 line fully sampled center

calculation between the origin and any particular point. This distance is then used in a calculation of a random variable based on a 1D Probability Density Function (PDF), that takes a standard deviation (σ) and a distance (x), see equation 5.5.

$$PDF(\sigma, x) = \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \quad (5.5)$$

The resulting random distribution matrix can be thresholded to any desired reduction factor %. Additionally, a fully sampled center can be applied to this technique by setting the probability of the points within the center diameter to 100%, see Figure 5.2.e.

A 1D variable density under-sampling technique with a fully sampled center is seen in Figure 5.2.f. This is achieved in much the same way as the 2D variable density calculation, using a 1D distance from the origin of k-space and sampling the entire line.

An analysis is carried out for the question, “Can CS reconstructions for MIP be performed at low or high resolution?” The input high resolution MRI TOF k-space dataset is 256×224 . It is centered by the highest peak in a 512×512 matrix, see Figure 5.4.a. Notice that this dataset is a partial Fourier encoding, where the center of k-space is fully sampled, and a portion of k-space is not sampled. For simulation of CS, this dataset presents a challenge for reconstruction at low resolution, see Figure 5.4.b. In order to use this data, it needs to be cropped at 256×256 having the peak centered. This does impact IQ of high resolution details, due to the fact that some fine details are deleted in the periphery of k-space. For the simulations here, shifting of the k-space peak avoids any cropping of data, and thus minimizes this negative impact to IQ. Therefore, a shifting of the under-sampling pattern to correlate with the center of k-space is also performed. For a real CS MRI acquisition, the under-sampling needs to be centered at the k-space peak to be effective and not lose important information at the k-space center and periphery.

5.3 Results

5.3.1 Comparing TIWT and TV

The Tables 5.1 through 5.7, show a comparison of TIWT vs TV CS with several sampling patterns for IQ analysis on three sample images from Figure 5.1. These images are diverse in several

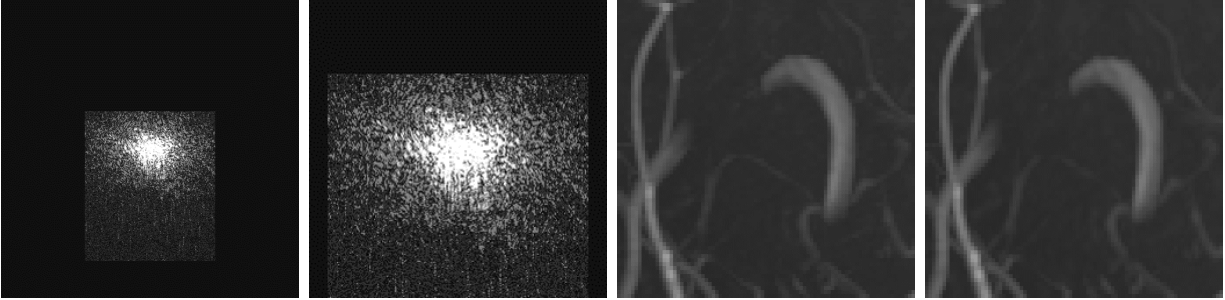


Figure 5.4: TOF MRI k-space data (1 receiver) and MIP reconstruction zoomed in (a) High resolution 512×512 data (b) Cropped low resolution 256×256 (c) High resolution reconstruction (d) Cropped low resolution reconstruction

practical qualities. The Shepp-Logan phantom is commonly used in analysis and demonstrates low noise and simple structures. The T1 and TOF MRI brain images portray practical medical images which have differences in noise and fine structure throughout. The 3D TOF additionally represents a sparse image domain which should lend well to CS due to the use of TV and wavelet ℓ_1 minimization functions.

In Table 5.1, there is a linear vs CS reconstruction IQ comparison with a variable density radial under-sampling pattern across the three images. There is also an analysis of stopping criteria where MSE results of 10 iterations are compared with 100 and 500 iterations. Note that for the phantom, the CS reconstruction with the TIWT has a much lower MSE than that of TV at 10 NLCG iterations and at a lower under-sampling rate of 25%. However, at 100 and 500 iterations the MSE is similar. This shows that 10 iterations is insufficient and performing more than 100 iterations may not change the image substantially. Through this data, there also appears to be different iteration limits to improving IQ for certain image types and transforms. When looking for a robust general solution, this experiment highlights how important it is to have an analysis to determine good iteration stopping criteria. Given this challenge, in order to have a consistent analysis, 100 iterations are chosen as the stopping criteria for all the other simulations.

In Table 5.2 for spiral under-sampling, comparisons can be made with radial sampling in terms of MSE. For instance, in the TOF MIP at 38% samples for TIWT and TV, the MSE is 10.4 and 33.0, respectively, which is similar to the results shown in Table 5.1. Looking at the same under-sampling ratios, a better IQ is achieved in an initial linear reconstruction. However, this does not

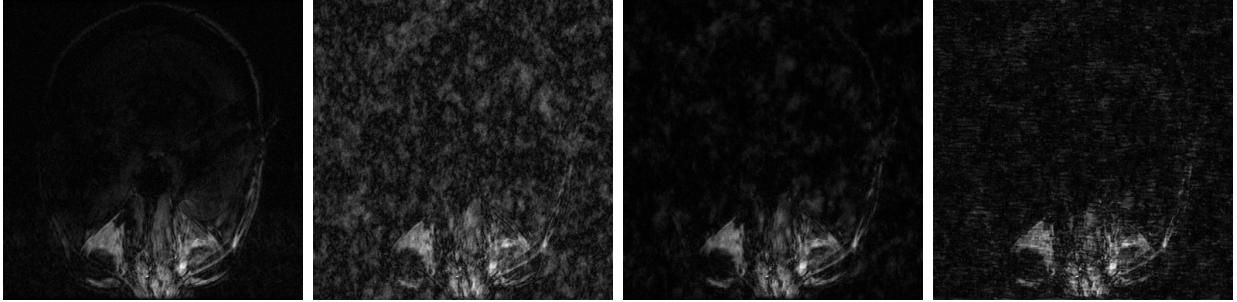


Figure 5.5: TOF reconstruction with 38% random under-sampling, receiver 1, slice 1 (a) Fully sampled (b) Linear (c) TIWT CS (d) TV CS

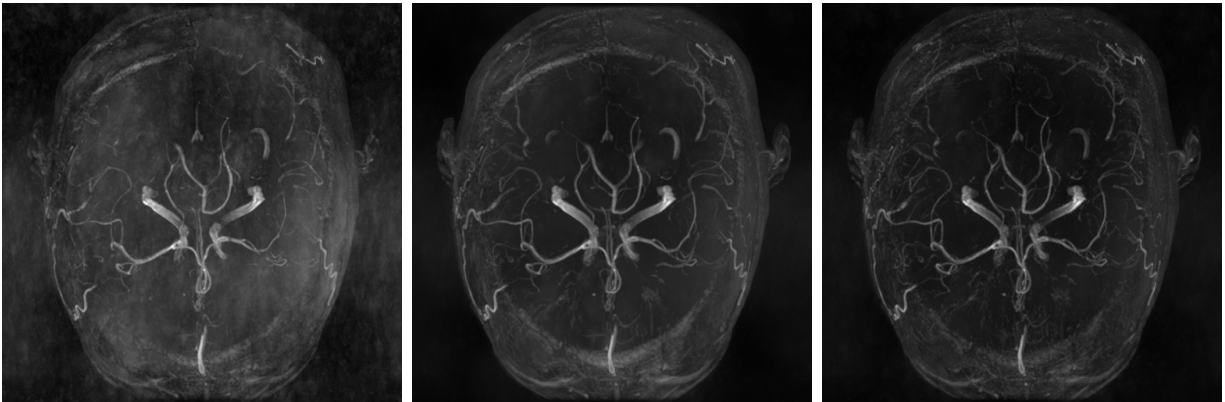


Figure 5.6: TOF MIP reconstruction with 38% random under-sampling (a) Linear (b) TIWT CS (c) TV CS

always correlate to a better CS reconstruction, as seen in some results of this table.

Table 5.3 shows results using a uniform random under-sampling pattern. Much higher MSE is observed in all linear and CS reconstructions. These results show it is important to have dense sampling at the center of k-space for reconstruction. Notice the range of MSE at lower % vs higher %, even at 75% samples, the MSE of the reconstruction is similar with radial under-sampling at 25%. However, even with this poor choice of under-sampling, the TIWT does operate effectively.

In Table 5.3, according to the MSE of the TOF MIP, it appears that the IQ does not improve very much when CS is utilized, and in some cases for TV, even degrades. Actually, this is not entirely the case, but rather an example of where IQ analysis with only MSE can be misleading. This can be seen by analyzing intermediate CS reconstructed TOF images that make up the MIP. Figures 5.5b and 5.6a show a 38% under-sampling linear reconstruction of the TOF slice and MIP. A fully sampled slice is shown in Figure 5.5a. Intense noise has clouded the TOF and MIP image.

After a TIWT or TV CS reconstruction, the noise is significantly reduced, see Figures 5.5c, 5.6b, 5.5d and 5.6c. Notice that comparing these results and the fully sampled MIP, Figure 5.1c, shows a large difference in background intensity. This is greatly reduced in the CS reconstruction and may be a desired trait because it intensifies the SNR.

Tables 5.4 and 5.5 represent the use of several 1D variable density under-samplings of k-space for good IQ and simple adoption into standard Cartesian MRI pulse sequences. In Table 5.5, a larger center of 2D fully sampled k-space contributes to a much better linear reconstruction, which in turn, produces an improved CS reconstruction over the smaller center shown in Table 5.4. Additionally, these consistent improvements do bolster a robust use of TIWT for CS.

Table 5.6 shows the use of a random variable density 2D under-sampling. Similar to the fully uniform random under-sampling cases of Table 5.3, there are drastic differences in MSE for reconstructions at different under-sampling percentages. Lower sampling results in less improvement in the CS reconstructions, however, this result improves when more samples are taken. Notice that for both these cases when a low percentage of samples are taken, the center of k-space is not fully sampled and the image reconstructions suffer a large loss of quality in major image structures and intensity. It appears that this under-sampling pattern should also be avoided.

Table 5.7 shows improved results over Table 5.6 by modifying the 2D under-sampling pattern with the addition of a center that is fully sampled. These results are very promising and show that this under-sampling pattern produces some of the best reconstructions. The TIWT also performs quite well with this sampling pattern.

Figures 5.4c and 5.4d represent an enlarged portion of the fully sampled TOF MIP. A high resolution (512×512) linear reconstruction is compared with a cropped k-space low resolution (256×256) reconstruction zipped to 512×512 for MIP calculation, The MSE difference between the two results is about 3.0, which is small compared to some under-sampled linear reconstructions. However, in a visual analysis of the figures, some very fine vessels are missing or blurred in the low resolution reconstruction causing an IQ impairment.

To mitigate this, the MIP simulations shown in the tables are all done by shifting the k-space data so that it is not cropped, but rather has the center of k-space and the sampling center match

Samples % (itr)	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25% (10)	355	55.2	145	449	65.6	362	194	135	140
25% (100)	355	4.68	3.00	449	65.3	59.5	194	27.2	81.5
25% (500)	355	0.450	0.033	449	64.0	49.7	194	25.3	67.9
38% (10)	197	22.6	25.7	256	12.1	50.0	132	36.0	73.4
38% (100)	197	1.34	0.0244	256	13.0	34.8	132	10.9	38.5
38% (500)	197	0.004	0.02	256	14.5	21.2	132	9.52	27.2
50% (10)	132	10.6	2.95	86.2	13.0	29.0	37.8	6.00	20.6
50% (100)	132	0.163	0.0112	86.2	15.6	14.6	37.8	3.45	15.9
50% (500)	132	0.0018	0.0122	86.2	14.0	23.1	37.8	3.69	14.4
75% (10)	53.6	3.11	0.021	4.6	1.94	2.51	3.67	0.92	2.66
75% (100)	53.6	0.033	0.0064	4.6	1.96	2.65	3.67	0.73	2.52
75% (500)	53.6	0.0007	0.0052	4.6	2.34	2.60	3.67	0.77	2.48

Table 5.1: Radial under-sampling pattern IQ comparison with MSE between CS reconstructed images and reduction factors (%) with 10 and 500 NLCG iterations

Samples %	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	216	28.9	1.91	354	18.8	25.3	138	39.6	85.8
38%	164	3.2	0.0221	80	12.0	14.0	70.2	10.4	33.0
50%	118	0.492	0.0194	34.6	6.75	19.6	30.9	4.81	14.8
75%	54.9	0.061	0.0077	2.98	2.0	2.92	4.12	0.93	2.23

Table 5.2: Spiral under-sampling pattern IQ comparison with MSE between images and reduction factors (%)

slightly off center. The MSE difference in this case is 0.11, which is negligible, and verifies the assumption that there is no visible difference in IQ between low and high resolution fully sampled reconstructions. With this accuracy, there is confidence in running these tests at low resolution with a shifted k-space.

5.4 Discussion

5.4.1 Good Under-Sampling Patterns

Here it is shown that the choice of good under-sampling patterns clearly has a correlation to the data being gathered. The center of k-space visually has a more intense magnitude of data, therefore, sampling density at the center contributes to a better SNR and a lower MSE. The

Samples	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	2629	1029	1128	3709	2077	3125	389	304	2684
38%	1910	837	1033	3251	2121	2823	281	196	310
50%	1518	789	922	1346	519	837	154	95.5	145
75%	1099	603	824	409	146	269	33.2	10.1	33.9

Table 5.3: Random under-sampling pattern IQ comparison with MSE between images and reduction factors

Samples	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	594	129	164	354	124	135	1128	457	631
38%	408	15.0	29.0	277	66.6	108	562	85.8	266
50%	291	1.23	13.5	252	26.7	35.6	444	50.8	216
75%	85.3	0.033	1.97	43.4	3.41	6.87	29.2	3.01	15.8

Table 5.4: 1D variable density with small (32 line) center under-sampling pattern IQ comparison with MSE between images and reduction factors

Samples	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	336	264	182	169	85.3	89.0	370	188	276
38%	222	60.7	75.8	145	31.7	57.9	217	82.6	114
50%	174	11.0	2.24	136	27.5	76.6	134	41.4	67.1
75%	64.6	0.014	0.386	28.5	2.66	4.72	12.9	1.94	7.70

Table 5.5: 1D variable density with large (60 line) center under-sampling pattern IQ comparison with MSE between images and reduction factors

Samples	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	1429	770	960	1956	848	1696	87.8	31.9	84.9
38%	1092	568	808	1618	768	1541	59.5	23.7	52.5
50%	910	581	750	1496	797	1505	22.2	4.03	11.8
75%	59.6	0.702	0.008	3.74	1.99	2.61	2.45	1.53	1.87

Table 5.6: 2D variable density under-sampling pattern IQ comparison with MSE between images and reduction factors

Samples %	Phantom			T1 MRI			TOF MIP		
	Lin	TI	TV	Lin	TI	TV	Lin	TI	TV
25%	286	65.4	3.42	280	31.7	27.1	187	32.3	56.1
38%	221	14.3	0.0211	186	11.1	20.4	124	4.96	37.4
50%	167	0.0328	0.0132	101	7.50	10.3	61.4	2.87	20.1
75%	54.2	0.0596	0.0035	10.3	2.32	3.25	4.37	0.84	2.69

Table 5.7: 2D variable density with small (25 pixel radius) center under-sampling pattern IQ comparison with MSE between images and reduction factors

radial, spiral, 1D and 2D variable density with the center fully sampled perform well, compared to the other patterns that did not sample with a dense center. Balancing IQ with how the under-sampling pattern dissipates is a complex analysis that requires visual as well as error analysis.

5.4.2 Reconstruction of MIP

CS reconstruction of TOF data used in a MIP can be performed at a lower acquisition resolution and then transformed to a higher resolution because there are no visible differences in smaller vessels. The analysis of IQ with MSE is challenging for MIPs because the error measurement may not pick up the changes in small vessels being of little numerical error. Additionally, changes in the background intensity of the image do contribute to larger error. Having a larger MSE may be misleading compared to what is observed with a visual analysis.

Special care must be taken for under-sampling with partially encoded Fourier k-space data. Visually, the image generated without small amounts of data at the periphery does have a negative impact on small vessel details of the MIP CS reconstruction. Therefore, adjusting the k-space center is needed to prevent cropping any data and have more meaningful IQ comparisons.

Chapter 6

Optimization for CS Problems with FastTestCS

6.1 Introduction

Since CS MRI reconstruction problems are extremely computational due to the iterative optimization with repeated transforms required, this reconstruction would have taken vastly longer to perform even a couple decades ago. Now many of these experiments and simulations are able to be implemented and performed fairly quickly with advances in computer technology and resources. Computationally, orthogonal transforms will operate faster than redundant or adaptive dictionaries which can add considerable time and memory to CS simulations and reconstructions. Sparsifying transforms that perform well in CS can be a wide variety from orthogonal to redundant to adaptive dictionaries, each with varying degrees of complexity and applicability. Along these lines of thought, Ning *et al.* investigate the patched-based trained directional wavelets and extend them into the T1 domain to enhance MRI image features [44]. Sadeghi *et al.* compare redundant dictionary learning algorithms for computational burden and quality [47]. Practical experimentation and application as proof of theory underlay engineering progress. The need for comparing other researcher's software code and algorithms is essential for prototyping ideas. There is a huge body of research in CS for MRI, so how does one evaluate what is the best practical technique for a CS MRI product? The advanced simulation software proposed here is necessary for trying techniques and making comparisons.

6.2 Methods

6.2.1 “FastTestCS”, A Fast Software Framework for Testing Under-Sampling and Sparsifying Transforms in CS

Novel software, FastTestCS, is introduced here as a CS testing framework for answering some of the questions posed in Chapter 1. It is a tool that can be customized for CS reconstructions using different images, sampling patterns, sparse transforms and optimization techniques. It also is written in the compiled language of C++ that allows for quick runtime of simulations. Additionally, CPU performance and time can be measured more in line with an expected fast product implementation. The FastTestCS executable is designed with a full set of versatile arguments to customize all the needed functionality to run several different simulations. It is modular, with simple methods that can be used similar to functions used in Matlab [43] or other object oriented programs.

FastTestCS is built upon the understanding that CS takes considerably more computation and memory than standard cases of MRI reconstruction. Wavelab [29], a collection of wavelet implementations for Matlab, used initially in testing the TIWT, is too slow. The abstraction and additional language internal processing are contributing factors. FastTestCS addresses the need to have a more efficient compiled programming environment over prototype Matlab software.

The speed of the TIWT is tested by comparing the transform reconstruction wall time difference between the Wavelab Matlab implementation vs. the transform written in C++. Two resolutions of an image are chosen, 256×256 and 512×512 . The test code is developed by isolating the transform setup, data memory transfers, computation and tear down of the method. These timing tests are performed without the overhead of other CS processing. This time measurement includes calculating 10 forward and inverse transforms on both real and imaginary parts of the image for a total of 40 calls to the TIWT method.

This number of transform calculations is comparable to the minimum that must be performed in a CS program. However, often the number of transforms are much greater, such is the case for the NLCG with a quadratic line search and Wolfe conditions. Many extra iterations must be performed to determine appropriate search directions and step sizes.

Four different implementations are tested for the standard orthogonal wavelet and the TIWT. The translational invariance is achieved by performing circular shifts of the data. The orthogonal wavelet provided by Matlab as "WAVEREC2" and "WAVEDEC2," is compared with the GSL C++ implementation "gsl_wavelet2d_nstransform_matrix." The TIWT, provided by Wavelab as "FWT2_TI" and "IWT2_TI," is compared with the FastTestCS implementation "TIDWT" and "ITIDWT."

The FastTestCS approach implements CS and several sparsifying transforms that run faster in C++ than in Matlab. Algorithms can be prototyped for accuracy first in Matlab, and then re-implemented in C++. The simulations are run in C++ in less time and in parallel across multiple cores and machines. Additionally, FastTestCS has image comparison tools such as MSE, as well as other useful image operations like "Normalize," "ReadImg" or "WriteData." It is a Microsoft Visual Studio project and can be compiled for Windows or Linux. It allows sharing code for easy algorithm prototyping, comparison and dataset manipulation. It is a portable and customizable simulation tool to benefit CS research.

There are many required permutations of several parameters and reconstruction tests to run in order to prove robustness. It is apparent that efficiency can be improved by parallelizing computations and simulations, compared with running on just one machine with one thread. Since some CS routines could run over a week to a month, it would be impractical to try many different options serially.

In response, FastTestCS executable is designed with very versatile arguments to implement all the needed functionality and parameters to run several different simulations from the command line. Multiple machines are used to easily distribute many simulations. Simple scripts are used to tar up the software and unique calling parameters, send the package to the specified machine, build and run the package, save the results and transfer the images and results back to a single archiving repository for review. With this method, there are no limits to the number of machines that can be used and simulations that can be achieved more quickly in parallel.

Distribution is achieved for these tests with a small cluster of five individual quad core machines, with one being the source and archive, in addition to having its own portion of processing.

Meta-data in the image identifies what simulation parameters produced the image for distinction. Once all the resulting images are together, quantitative and qualitative image comparisons are performed with a “golden” fully sampled image. An executable loads the reconstructed images and the golden image, and makes several calculations for each image such as MSE. The program can then output results to a comma separated table with one line for each image that is easily viewed in a spreadsheet.

MRI k-space data is generated by a General Electric (GE) Healthcare MRI scanner and saved as a “P”-file. GE provided Matlab software to read the P-file and make a k-space matrix. Due to this data being on a Cartesian grid, a fully sampled reconstruction is simply an inverse FFT. This fully sampled dataset is input to the under-sampling process used in FastTestCS reconstruction by way of a Comma Separated Value (CSV) file. The resulting CS reconstruction is compared to the original fully sampled dataset and measured by MSE to verify its fidelity.

An important way to analyze different parameters and IQ results of CS reconstructions is to keep all parameters the same in the CS algorithm and just change the transform or sampling pattern. The analysis and results here are done with individual transforms without a combination of other transforms. This is to measure the error of each sampling pattern independently and identify the transform quality.

The steps of the CS reconstruction simulation are as follows and in Figure 6.1:

1. Start with fully sampled k-space data.
2. Sample the k-space data using a desired under-sampling pattern. (ie. Randomly 1D sample the k-space data rows using a desired PDF. The middle of the PDF has a probability of 1 to fully sample at the middle of k-space. See Figure 6.1 for the PDF and an example resulting sampling pattern.)
3. Perform the iterative non-linear optimization technique.
4. Compare the error between the fully sampled image and the CS reconstruction to measure the quality of the reconstruction.

Regularization is a standard technique for preventing over-fitting and having noise overly affect the solution [35]. In CS, conjugate gradient regularization uses a sparsifying transform such as the Wavelet transform, W , to take advantage of the compressibility of the actual image. To balance the effect of regularization, λ is used as a weighting parameter. One solution is the unconstrained convex optimization problem shown in equation 6.1 [5].

$$x = \underset{x}{\text{minimize}} [\|Ax - y\|_2^2 + \lambda \|Wx\|_1] \quad (6.1)$$

The last term can be replaced or added to with other regularization approaches such as TV or a contourlet transform [27] instead of a wavelet transform. The transform chosen depends on the sparse domain(s) of the image, which can be measured by IQ from compression and CS simulations. One important property of this formulation in equation 6.1 is non-linearity; it is no longer a linear function of the vector x .

For reduced acquisition in MRI, data along the phase-encoding direction is usually under-sampled due to its direct reduction in acquisition time, but data along the frequency-encoding direction is still fully acquired. In Sparse MRI [40], variable-density random under-sampling along the phase-encoding direction is usually used for data acquisition.

The CS reconstruction optimization in FastTestCS is the NLCG with backtracking line search and Wolfe conditions, [45] and [40]. The formulation of the objective function $f(x)$ in equation 6.1 requires the calculation of the gradient ($\nabla f(x)$), which Lustig *et al.* [40] define as equation 6.2.

$$\nabla f(x) = 2\Phi^*(\Phi x - y) + \lambda \nabla \|Wx\|_1 \quad (6.2)$$

Where $\nabla \|Wx\|_1$ can be approximated with the *sign* function as in equation 6.3. The star (*) symbol represents the complex conjugate transpose.

$$\nabla \|Wx\|_1 = W^* \text{sign}(Wx_r) + W^* \text{sign}(Wx_i) \quad (6.3)$$

This unconstrained formulation is versatile when dealing with non-orthonormal transforms. The calculation of the gradients in the unconstrained approach requires the inverse of the transform. For an orthonormal transform, that is simply a multiplication of the coefficients by a

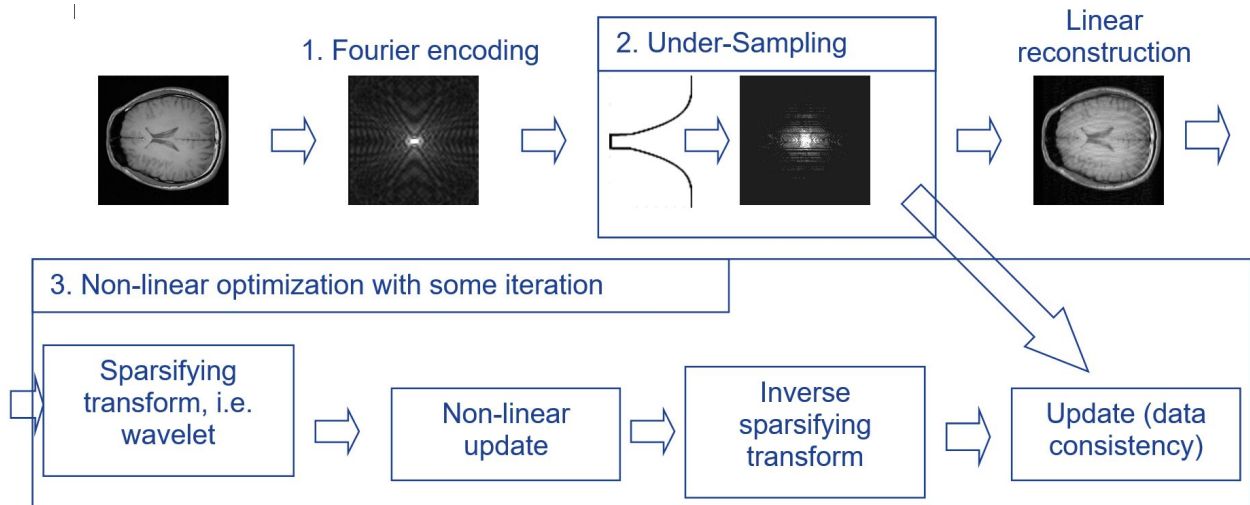


Figure 6.1: Implementation of CS image reconstruction simulation

transpose of the wavelet dictionary. However, for a non-orthonormal transform, like the TIWT and many of the other transforms, a separate calculation of the inverse transform is used rather than a transpose of the dictionary. In FastTestCS, a transform method is created that calls the forward or inverse of the transform. All coding for the transform is easily integrated, segregated and tested. Real and imaginary parts of MRI data are treated separately throughout all of the optimization.

Additionally, a combination of any two transforms could be created. This is done by performing a separate NLCG optimization at each iteration for each type, then the average of the two is created at the end. The IQ of the combined transforms perform better than the individual ones with the obvious impact of more CPU time. Much analysis is done on individual transforms without other transforms or a TV term. This is needed to measure the value of each transform independently and identify their qualities.

There are several other approaches to calculating $\nabla \|Wx\|_1$ which are also investigated. The ℓ_1 norm for the data matrix is defined as $\sum |Wx|$, however, when dealing with a complex data set, there are different ways of applying the sparsifying transform and calculating the gradient. A description is in the appendix.

Recall in equation (6.1) for the unconstrained CS formulation, there is an unknown constant, the weighting parameter lambda. In order to estimate the ideal lambda, a guess and check

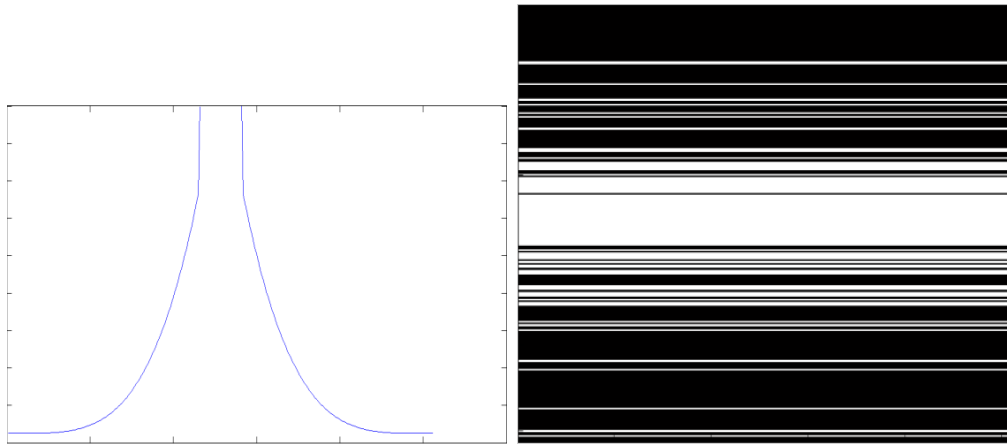


Figure 6.2: (left to right) (a) PDF of sampling function. Vertical scale of the plot is 0 to 1. (b) The resulting sampling pattern in k-space, where the white lines are acquisition lines.

method is implemented where the MSE measurement of the image quality is computed for each image that is produced with a given lambda. It is determined that there is a minimum MSE as a function of lambda. Therefore, the lambda that produced the lowest MSE is an optimized lambda and is used for comparisons between transforms.

Another non-linear reconstruction approach uses OMP, to calculate a sparse transform given a custom dictionary for the transform. A custom dictionary can be created and even trained to be more applicable to the image being reconstructed. OMP is an inexact transform because it will find a representation that is most sparse given an acceptable error tolerance. The lower the error tolerance, the more exact is the representation. To use OMP in the CS framework of Figure 6.1, replace the sparsifying transform, non-linear update, and inverse sparsifying transform steps, with the OMP sparsifying transform. Alternatively, one could keep the non-linear update step; however, that would be adding an unnecessary additional non-linear update to the process.

To train a dictionary, the K-SVD [1] approach is implemented, where the training set is the initial guess image that is produced from a linear reconstruction. K-SVD will produce a sparsifying transform dictionary, given a specified error tolerance. Once a dictionary is trained, it can be used with OMP.

OMP is implemented on small image patches because it cannot operate on large datasets all at once due to memory requirements. A full image is broken up into small patches that overlap

and are averaged when recombined so as to eliminate blocky artifacts. This technique will add much redundancy to the image and coefficient set if the overlap is large. The block size and overlap are important factors in IQ. Comparisons of these different settings are also performed. One additional parameter with this approach is the use of a redundant dictionary, which can vary from not being redundant to being many times redundant.

Due to the differences of a sparse representation calculated on a whole image vs patches, tests with transforms on small patches are also investigated. By using the unconstrained NLCG approach, rather than doing a whole image transform, the image is broken into small and redundant patches, then the NLCG is performed on the redundant representation.

Another parameter that needs to be understood in all of these optimization approaches is the number of iterations. The objective function is found to be a good indicator of convergence because it will decrease rapidly at first and then decrease at smaller increments. When these increments are small, the image is not changing much and the advantage of more iterations becomes of lesser value. Additionally, sometimes the objective function will increase, thus indicating that the image may not converge but rather diverge. In this case, it is important to stop the iterations at a point when objective function is a minimum and before it diverges.

When using OMP as the non-linear update step, there is no objective function, but there is a level of sparsity. The sparsity increases after each iteration, at first very drastically, but after a few iterations, it increases by a smaller amount. At this point, the image is not changing very much which may indicate a good stopping criteria.

Both the objective function and sparsity stopping criteria should be contrasted with actual image quality by visually inspecting the images a few iterations before and after one has determined to stop. Additionally, a quantitative measure can be plotted at each iteration to see where improvement ends. These approaches are a simulated verification of the stopping criteria because in a real situation, one will not have a fully sampled image to compare.

To find the best k-space trajectories to use, the standards set up by Lustig *et al.* in Sparse MRI are compared using a Cartesian grid, so results can be compared equally with other sparsifying transforms [41]. Just the sum of squares is used for parallel imaging, since our focus is on

sparsifying transforms, and not on parallel imaging technique comparisons. In using the PDF, there is a way to randomize the sampling pattern so as to test robustness with different seeded patterns, see Figure 6.2.

The unconstrained CS reconstruction formulation from equation 6.1 requires calculation of lambda. Lambda was determined by setting the number of non-linear CG iterations to 15 and the reduction factor to 33% and varying lambda for each transform. The result of this analysis is shown in Figure 6.3, where it can be seen that a minimum occurs for each sparsifying transform. This minimum can then be used for future tests that use a similar image type and reduction factor. The effect of lambda may need to be readjusted when the image is of a different type, has different artifacts or a different reduction factor. This is because more or less regularization may be needed to mitigate or enhance these image features so convergence of good IQ occurs.

The other factor that needs to be understood is the number of iterations. This can be determined partly from the plot of MSE after each iteration and from visual inspection of the image when MSE appears to be stable. This stability can be seen in Figure 6.4 after about 40 iterations. It should be noted that in certain cases, more iterations are needed to remove artifacts. Even if MSE does not appear to be improving much, artifacts can still be removed. This is observed in Figure 6.5 where at 35 iterations there are still artifacts, but at 100, they are lessened. Figure 6.4 shows that TV does contribute the most to an improved sum of error calculation. Visually, the artifacts seen in Figure 6.6.b appear to be reduced the most by TV regularization, seen in Figure 6.6.d as well. The other transforms (Figures 6.6.e,f,g) produce varying degrees of quality. The wavelet regularization has fewer ring artifacts than the contourlet or curvelet regularization. The curvelet regularization is much smoother than the contourlet and wavelet regularization. The contourlet regularization appears to preserve some line features better, but more artifacts remain. The ℓ_1 -norm regularization did not contribute to improve the artifacts; however, some noise is reduced, as seen in a comparison of Figures 6.6.b and 6.6.c. A small additional improvement in MSE can be found when combining multiple regularization techniques with the appropriate lambda. This lambda will not be the same as the minimum lambda found in Figure 6.4 because lambda becomes a balancing parameter between the other

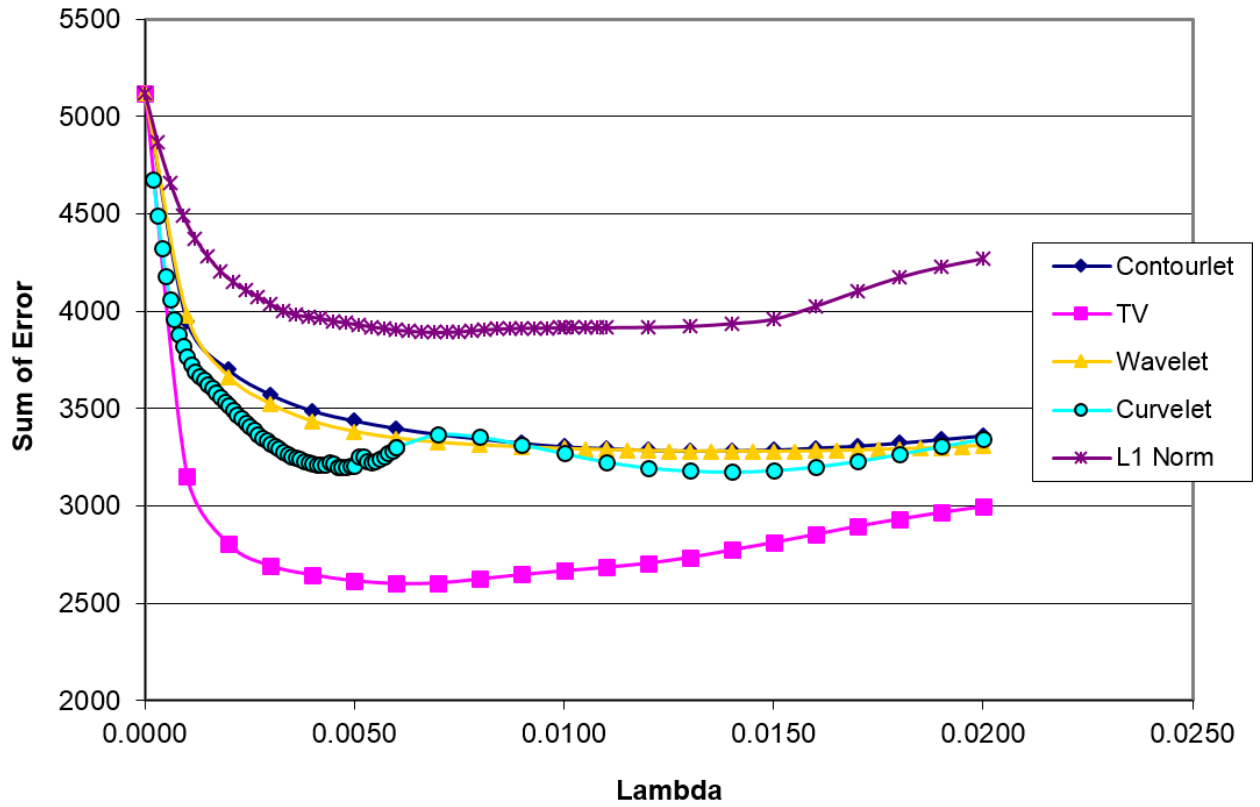


Figure 6.3: Unconstrained CS with different transforms and regularization parameters (lambda).

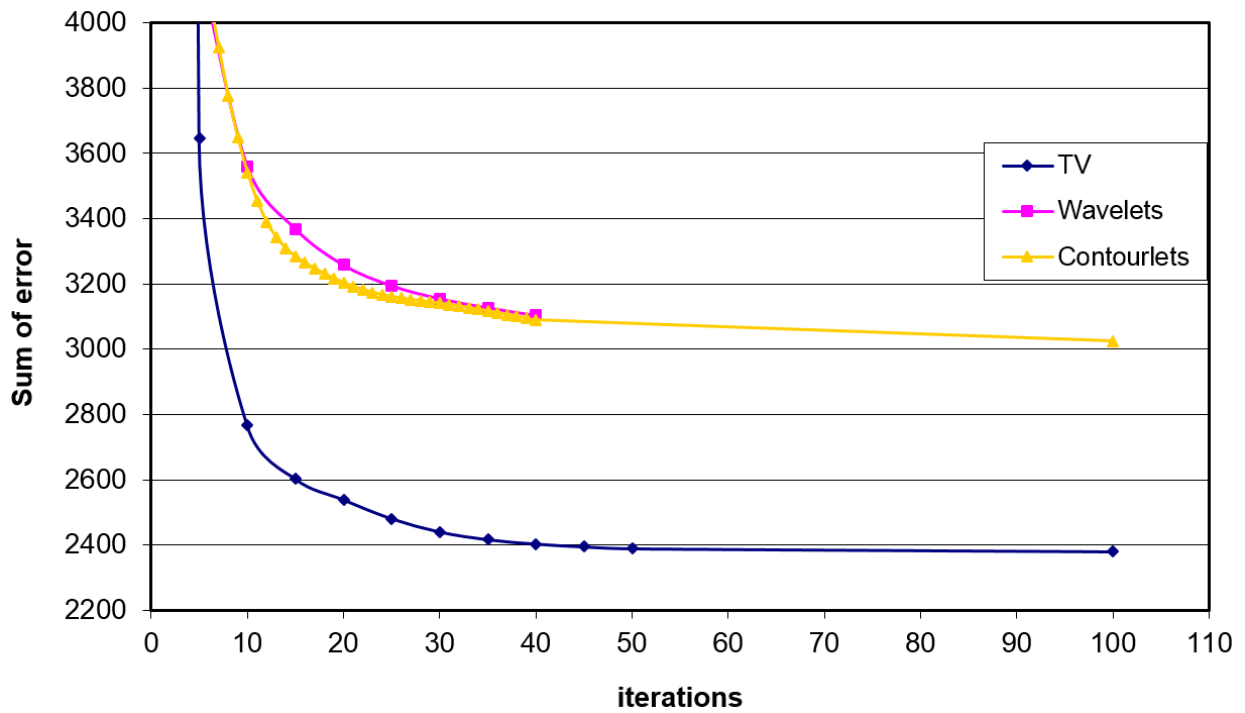


Figure 6.4: Unconstrained CS iteration comparison.

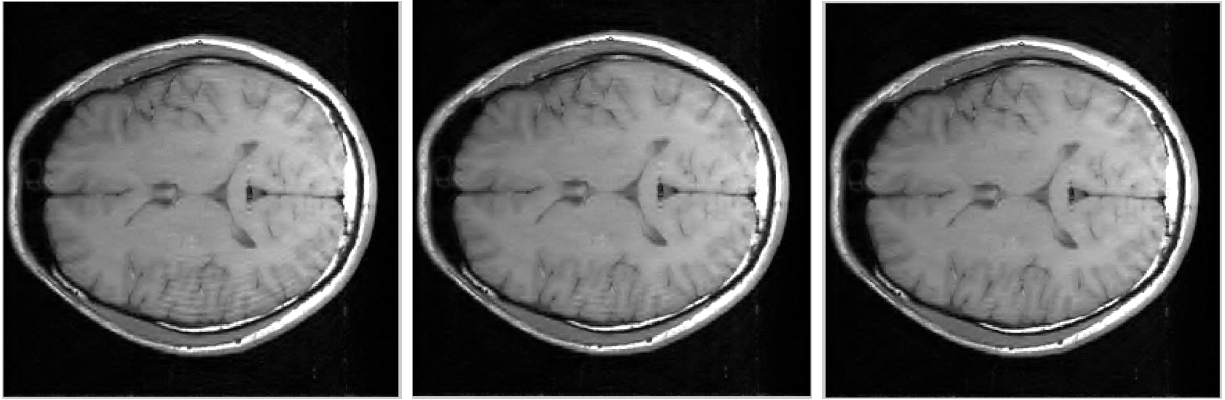


Figure 6.5: Unconstrained CS result for iteration comparison with wavelet regularization. (left to right) (a) 15 iterations, (b) 35 iterations, (c) 100 iterations.

regularization techniques. Finding the optimum lambdas for multiple regularization was not investigated, therefore better results than Figure 6.6.h may be achievable.

The unconstrained CS formulation from equation (6.1) can be performed without a weighing parameter; however, there may be some improvement when a soft threshold is used to weight the regularization. Figure 6.7 shows how different soft thresholds affect the sum of error for the constrained CS reconstruction. Figure 6.7 also shows how different contourlet settings (a change in n_{levs} is shown) can drastically affect the minimum error that can be achieved. One important observation from this graph is that for these cases, a soft threshold only minimally improves the reconstruction. Figure 6.8 shows images reconstructed with constrained CS using an optimized soft threshold. Here wavelets appear to have mitigated artifacts better than contourlets or curvelets. One factor that is not optimized for each transform is the number of iterations. This depends on how quickly an image converges for each transform, which can differ.

6.3 Robust Simulation Framework

Two main sampling schemes are investigated, Cartesian 2D and radial 2D. The Cartesian scheme is investigated more thoroughly. A random probability density under-sampling is used which samples lines, while fully sampling the center of k-space on the Cartesian grid. Having the probability density function allows for repeating runs to test the robustness of the sampling pattern with different random seeds. Each sparsifying transform is tested repeatedly with the

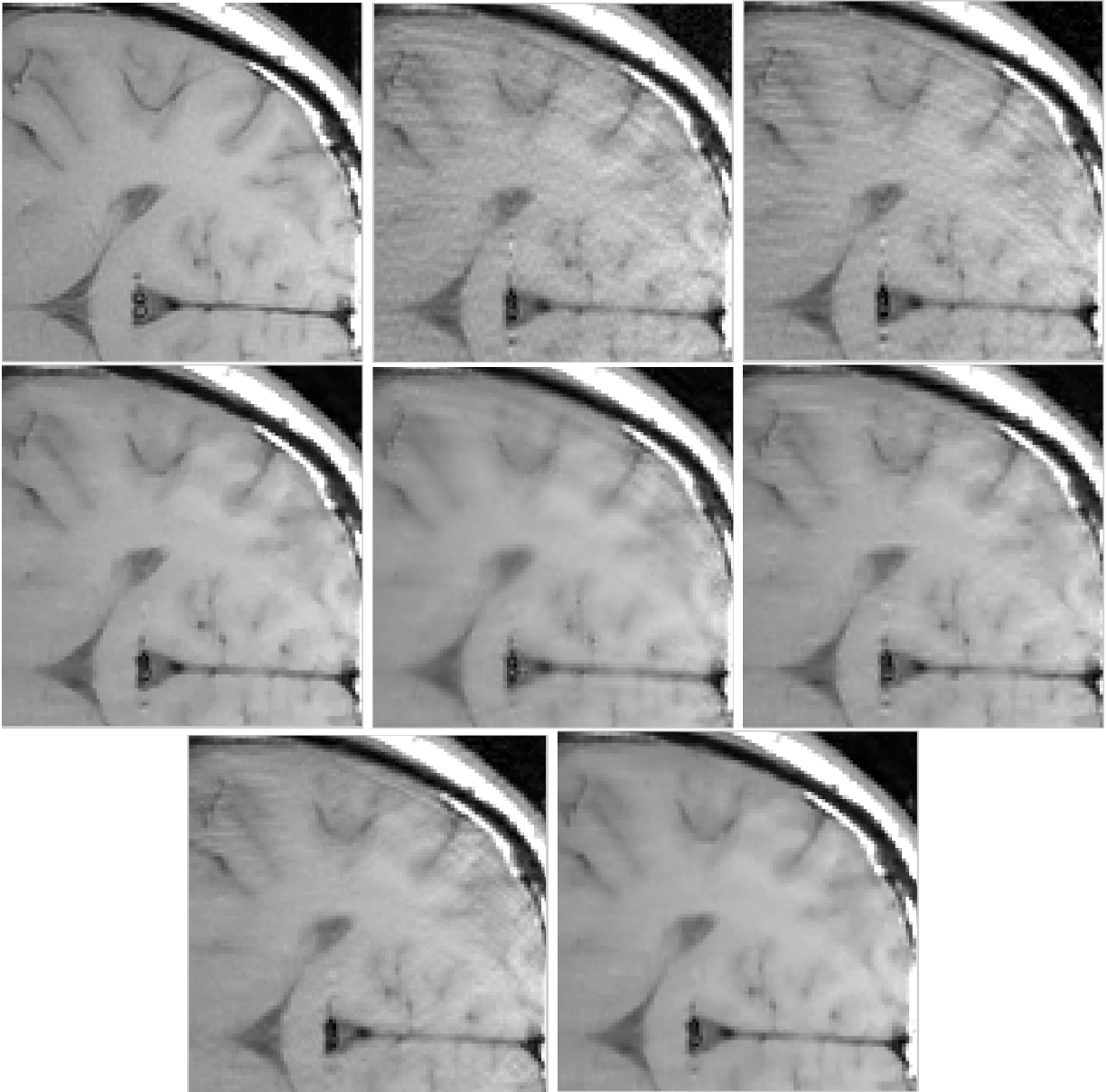


Figure 6.6: Unconstrained CS results, (left to right, top to bottom) (a) Original image, (b) Without regularization, (c) With only ℓ_1 -norm regularization, (d) With only TV (e) With only curvelet regularization, (f) With only wavelet regularization, (g) With only contourlet regularization, (h) All transforms combined regularization.

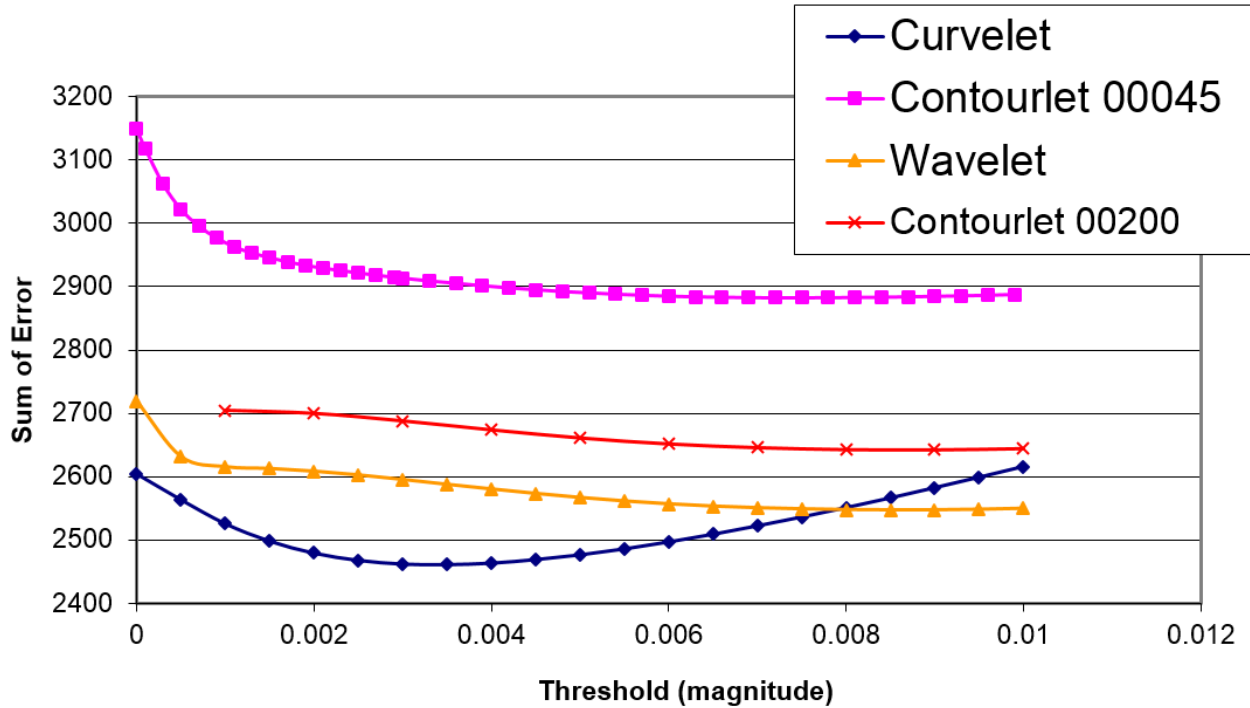


Figure 6.7: CS with different transforms and soft threshold values

same set of random k-space acquisitions at several levels of under sampling. Sum of squares is used to combine the parallel channel datasets.

The first step is to test these transforms and find the best optimization and sampling schemes for this research in Matlab. Due to limits of compute time, the NLCG constrained ℓ_1 minimization was re-implemented into FastTestCS, a C++ application. The probability density function is also re-implemented from Matlab to C++ for use with the NLCG CS application. OMP is also re-implemented in C++ and used for TICT in the NLCG CS application. Additionally, K-SVD is implemented in Matlab using code from the author[1]. Attempts are made to re-implement it in C++ without success. The Matlab CS NLCG problem is also able to be modified to perform K-SVD as the sparsifying transform. Due to K-SVD and OMP also being non-linear optimizations, an additional approach is taken that eliminates the non-linear update step typical in the CS NLCG implementation. From a theoretical perspective, there is no need to have two non-linear steps present in the optimization, only one is necessary. Therefore, tests were also performed using only K-SVD as the non-linear optimization and results proved this to be true.

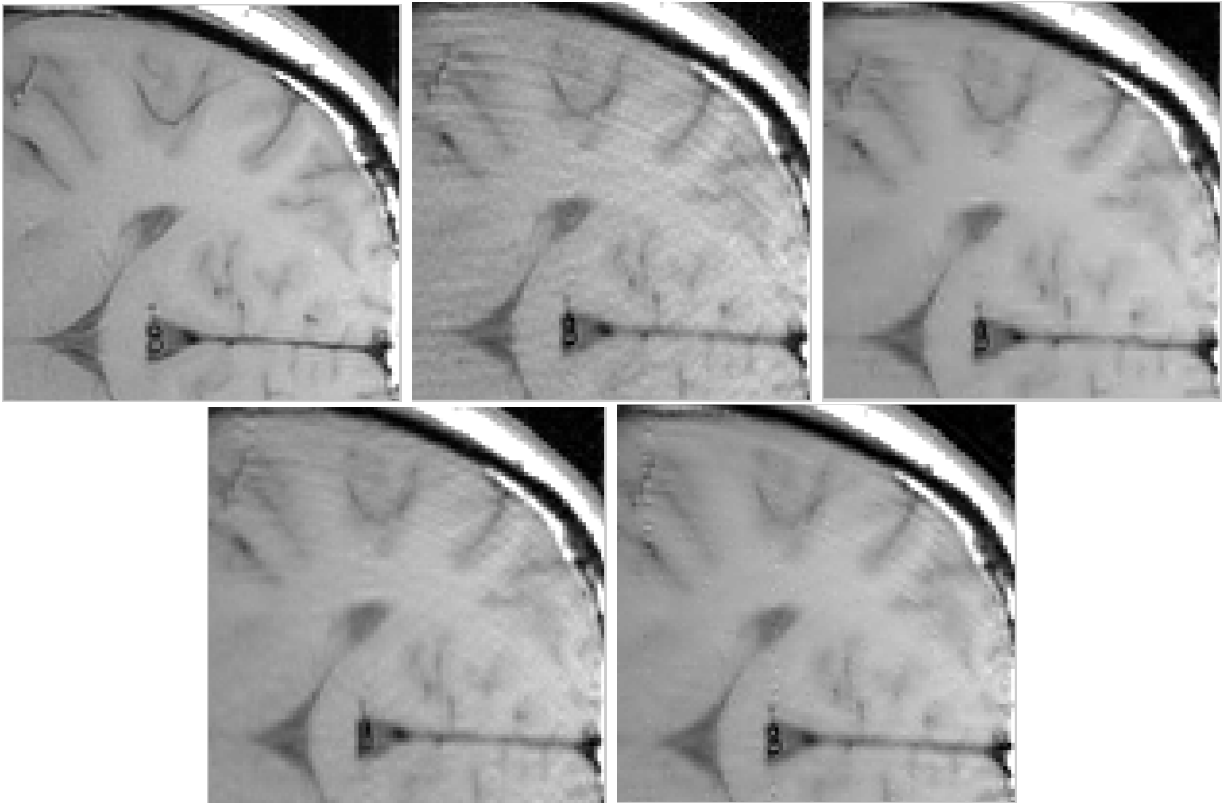


Figure 6.8: Unconstrained CS simulation results using optimized soft threshold (left to right, top to bottom)(a) Original image (b) Non-transformed (no threshold) (c) Wavelet (d) Contourlet (e) Curvelet

In terms of non-linear solver convergence, the unconstrained NLCG has the best robustness and ease of implementation. Every sparsifying transform did converge with this solver.

A radial sequence is investigated because of the quality of less coherent artifacts, which has been pointed out in the application of ℓ_1 -Magic. ℓ_1 -Magic is investigated for use in 2D CS with different sparsifying transforms; although many attempts are made, the optimization algorithms did not converge, therefore, no results are obtained.

In terms of de-noising and artifacts, having the center of k-space fully sampled produces better quantitative measures. The center of k-space is where lower frequency larger structures are maintained. When under-sampling the outer lines of k-space, more fine structures can be distorted, blurred or lost when using higher reduction factors. Therefore, minimizing based on MSE does not lead to the best sampling pattern.

In terms of using quantitative measures to find the best sparsifying transforms and settings, MSE did not always produce reliable results, but neither did any single measure. If quantitative measures are to be used, a balancing of multiple measures may prove most effective, similar to edge and noise measures, or SSIM, or a combination of all. These measures do identify a good set of sparsifying transform settings, but are not reliable enough to identify the best. A visual inspection of the results is always necessary. Often, a very low edge error does not produce a very low noise result. Potentially, these two measures are helpful together, because when both measures are equally good, the results are better.

In summary, a simple to use framework is developed for testing sparsifying transforms: The NLCG with ℓ_1 minimization. The gradient calculation applies the transform to the real and imaginary parts separately to produce the most stable results.

6.4 Results

Performance measurements show a big speed up in computation time with optimized C++ code vs Wavelab. See Table 6.1, where the orthogonal wavelet is optimized in a Matlab toolbox vs in C++ using the Gnu Scientific Library (GSL) [32] in FastTestCS, the computational time is very similar. However, when using the TIWT from Wavelab in Matlab vs C++ where the TIWT

Transform	Function	Language	256x256	512x512
O Wavelet	WAVEREC2 & WAVEDEC2	Matlab toolbox	0.75	2.9
O Wavelet	gsl_wavelet2d_ nstransform_matrix	C++ GSL FastTestCS	0.33	2.21
TIWT	FWT2_TI & IWT2_TI	Matlab Wavelab	342	1380
TIWT	TIDWT & ITIDWT	C++ GSL FastTestCS	4.6	22.83

Table 6.1: Time measurement (seconds) comparison of 40 wavelet transforms (10 complex, forward and reverse), O = Orthogonal, with C++ and Matlab implementations.

implementation is coded for FastTestCS, the difference is quite dramatic. FastTestCS is sixty times faster! This shows that coding some computationally time consuming algorithms in C++ is most beneficial.

Reconstructing the TOF MIP, all at a high resolution did come at a high computation cost of about $5\times$ longer, therefore, lower resolution reconstructions are performed. As a time comparison, a 512×512 8 channel 192 slice MIP CS reconstruction with the TIWT for 10 iterations takes 20 hours, whereas, the same reconstruction at 256×256 takes 4 hours, single threaded, with an Intel Core i7 CPU.

6.4.1 Lack of Good Comparison Software

Software engineering is sometimes a neglected discipline in the process of research and simulation. It may be at most an afterthought of the work for a publication which produces tables, images or other results through much computation.

Ideally, one would like to have produced software that meets software engineering principles with high quality such as robustness, maintainability, usability, re-usability, abstraction, modularity, stability, documentation, performance, modern language, streamlined, efficient, safe, accurate, multi-threaded, multiplatform, customizable, and distributable. Effort is required for such tasks and when these aspects are neglected, a need may arise in the development of the research software for one or a few of the missing aspects requiring extra re-work. This may be typical and

may explain why research software is very much an application specific prototype. It can have problems when needed for other projects, other than the one it is designed. After going through a few iterations of rewriting methods, Application Programming Interfaces (API) and designs, some understanding of the value of software engineering becomes apparent.

Additionally, one should identify and answer several questions: What is the state of the art? What implementation is currently used that requires advancement? How are comparisons done? How are simulations and results used and shared? What compute resources are needed? How much collaboration and teamwork is needed? What cross functional work is required?

Much MRI research software is written in Matlab; often, colleagues will share Matlab code. So, the use of Matlab software is essential to research development, because it allows for easy algorithm prototyping and dataset manipulation. Matlab can be used as a prototype to read raw data and generate plots, images, movies and other datasets.

For performance computation measurements, and future implementation, is Matlab really a good environment for the development of CS? How can research software be made that will work for prototype and then be useful to others for advancement or comparison?

Practical CPU considerations should also be understood. For example, is a learned dictionary practical when the computations are orders of magnitude higher than other sparsifying transforms?

With the understanding that CS takes considerably more computation and memory than standard cases of MRI reconstruction, a question could be asked: Should a different programming environment be used? There are several additional considerations. Current MRI products are produced and programmed in C++ and must use several software libraries. This may be a big transition for researchers that use Matlab where there is such a code base already written. Matlab also has many necessary software libraries already preloaded or easily integrated. However, with C++, one may be able to integrate their software more easily to product implementations when the same language is used. From a CPU performance perspective, C++ code may run up to $10,000\times$ faster than Matlab code, and this is the type of speed up required for CS. Matlab does have the ability to wrap C++ code to be used in Matlab, which does then run just as fast. It seems that the transition to at least coding some time consuming algorithms in C++ would be

most beneficial. Whether to code everything in C++ or some in Matlab and some in C++ takes some additional balancing.

Even with an ideal language implementation, compute resources often limit simulation and test speed. Research often involves trying several different options which will involve many iterations of the simulation. Contrast this with product software which must run once the right way and may not be as parallelizable or customizable. Therefore, product equivalent software and compute resources may not be ideal for this research. What should be used instead?

The CS research involved in investigating sparsity of transforms in CS first required implementing the transform and allow the API to modify several important parameters. For instance, after investigating the K-SVD transform, it is found that the image patch size can be modified in some cases to produce better results. It is important to try these cases to compare results with other parameter settings. Additionally, many results require statistical stability, so several iterations with different random seeds for the random under-sampling are required. Some sparsifying transforms work better at different reduction factors; therefore, trying several ranges of reduction factors is also required. For wavelets, several different filters could be used which would also result in different results and these parameters could be set up to be swept through as well. Image types are also a factor: There are various single and multi-channel image types and datasets which will perform differently with different transforms. The exponential expansion of the number of simulations quickly becomes overwhelming.

Consider new applications that patches can impact like DCT or wavelets. Consider how much computation does this required compared to other transforms? How does adding TI and redundancy to wavelets affect the CS MRI reconstruction and computational cost? TV is very fast to compute and can be easily combined with other sparsifying transforms for better IQ.

6.4.2 Fast Software Framework for Testing Sparsifying Transforms with CS “FastTestCS”

Several transforms for IQ improvements are evaluated while not neglecting performance, the analysis of the amount of time it takes to perform these sparsifying transforms in a clinical

Transform	Memory(MB)	Iterations	Time(sec)	Threads	BS	SD
WT	20	1	0.34	1	n.a.	n.a.
WT	20	20	3.9	1	n.a.	n.a.
TIWT	134	20	30.5	1	n.a.	n.a.
TV	24	20	9.5	1	n.a.	n.a.
DCT	24	20	17	1	4	4
TICT	156	20	178.3	4	4	1
TICT	169	20	397.1	1	4	1
TICT	324	20	2243	1	6	1
TICT	560	20	9915	1	8	1

Table 6.2: Time and memory measurements when using various transforms in CS simulation vs. threads, Block Size (BS) and Sliding Distance (SD) for a single image reconstruction (256x256)

setting. If these transforms take an impractically long time, how can it be applicable to be used in a product time frame of about a minute? CS reconstructions with decimated wavelets for single images can take minutes. However, when dealing with larger datasets such as multi-channel 3D volumes, this could result in the reconstruction taking over a day on a single dataset. For more advanced sparsifying transforms with dictionary learning, the time problem becomes much worse. Not only would clinicians be affected by the time element, but, researchers would find it insurmountable to perform many tests in Matlab comparing many transforms. To overcome this time hurdle, another language and implementation is pursued by optimizing code and re-implementing algorithms and transforms in C++. Speed up factors of $10\times$ to $10,000\times$ are seen. Once the CS algorithm and a few sparsifying transforms are converted to C++, CPU performance and time could be measured more in line with a fast product implementation. Now a decimated orthogonal wavelet transform in a single image reconstruction takes less than 10 seconds. Still other redundant transforms such as the TIWT takes on the order of $10\times$ longer than the decimated orthogonal wavelet transform of about a minute per image. A $100\times$ to $3000\times$ increase in time is measured when the translational invariant DCT using patches with OMP is used compared to the baseline time of decimated orthogonal wavelets, depending on the level of redundancy, see Table 6.2.

CS is an iterative approach which repeats the computations several times. Hundreds or

even potentially thousands of iterations are possible when forward and inverses of sparsifying transforms and FFT's are being performed to go back and forth from sparse space to image space to measurement space. If additional gridding is required, then another highly computative step must be a part of each iteration. For the investigation of sparsifying transforms, it leads to over-complete representations and dictionary learning. This means that the image was no longer a 256×256 matrix, for example, but rather could be redundant 5 fold or more, and much more with overlapping patches. A single 2D image could be broken up into about 65000 8×8 overlapping patches with a redundancy of 64 times. Additionally, when performing the K-SVD algorithm, at each sparse representation calculation of coefficients, there is an additional optimization step of OMP on each small patch. The computational increase in performing K-SVD is 100's of times more than a decimated 2D wavelet on the same image. When considering memory space, the space required for a basic 2D image in Matlab is 65k elements, which are complex doubles, at 16 bytes each, that is 1 megabyte. The processing involved with larger representations will require much more time and memory space. When dealing with 3D datasets, memory becomes even more of an issue. For example, a $256 \times 256 \times 256$ matrix, when gridded could be up to $512 \times 512 \times 512$ taking 2 GB of memory. On a 32 bit system, which is typical of the student edition of Matlab, 2 GB is the maximum capacity of RAM. In subsequent years, however, most modern workstations are 64-bit with support of +10 GB of RAM; therefore, these sizes of datasets are now more plausible.

The prototype Matlab software is used for generating all the results for the ISMRM abstract "The Investigation of Sparsifying Transforms in Compressed Sensing for Magnetic Resonance Imaging" [6], where several different sparsifying transforms and algorithms are compared and evaluated by simulation. Some of these sparsifying transforms are available through Wavelab, a collection of wavelets written in Matlab. Some other transforms are available through the authors' websites (see appendix). Finally, the remaining algorithms and transforms are hand coded in Matlab. The raw k-space data is generated by a GE scanner and saved as a raw data "P"-file. GE provided Matlab software to read the P-file and use a k-space matrix. Due to this data being on a Cartesian grid, reconstruction is simply an inverse Fourier transform, performed

with a Matlab function “IFFT”. The fully sampled dataset is input to the under-sampling process used in CS reconstructions. The resulting reconstruction is compared to the original fully sampled dataset to verify its fidelity.

It became apparent that after running these simulations on just one machine, Matlab would not be efficient. Since some CS routines such as K-SVD would take over a week to a month to run, it was impractical to try many different options. This research then takes an additional approach to implement CS and several sparsifying transforms that are in Matlab into C++. The path is quite clear now, one can create software by verifying the accuracy of the transform and algorithm first in Matlab and then re-implement it in C++, verifying between the two, and then run the simulations in C++.

Much Matlab code is converted to C++ to take advantage of more speedup. The orthogonal wavelets, TIWT, DCT, TICT with patches are re-implemented in C++. Some investigation is also done to determine the value of implementing code with GPUs and CPU's. It is determined that the versatility and advancements of multi-core CPUs is keeping pace with the advancing GPU offerings. The patch transform with OMP for TICT with parallel threads is implemented to take advantage of multi-core CPU's. When attempting GPU integration with Matlab and C++ for additional speedup, one needs to understand what transforms and operations are parallelizable. This research is also able to use advances in multi-core computer hardware to run multiple simulations at the same time on the same machine, to test several options more quickly than testing those options one by one.

This optimization and parallelization gives speed up benefits of 3 orders of magnitude in performance for many simulations. The OMP algorithm also sees this substantial speed improvement. Additionally, for K-SVD and TI-DCT which uses OMP, the small patch calculations can be performed in parallel. An additional speed up of about 2.5 is achieved with P-threads on a 4 core system.

Multiple machines are also used to distribute many simulations. Simple software clustering scripts are written to tar up the test executable and calling parameters, send the package to the specified machine, run the package, save the results and transfer the images and results

back to a single archiving machine for review. The executable is designed with a very versatile API to implement all the needed functionality to run several different simulations with all the needed parameters. With this method, there are no limits to the number of machines that can be used and simulations that can be achieved more quickly in parallel. For instance, a cluster was implemented to use 5 individual quad core machines, with one being the source and archive, in addition to having its own portion of processing. It would tar up 5 packages each with different calling parameters. Then, scripts would send the packages, one by one, to each of the machines, to execute their set of simulations, running 8 at a time, 1 per hyper threaded core, starting the next set of 8 simulations once the first 8 finished. Any number of simulations are automatically accommodated. The archiving machine then starts polling the other machines to see when they are done. If they are, it will retrieve the results and place them in the archive. Meta-data is used for distinction in the image, it identifies what simulation parameters produced the image. Once all the resulting images are together, quantitative and qualitative image comparisons are then performed with a “golden” fully sampled image. An executable loads the “golden” images, and makes several calculations for each image such as MSE, Edge, Noise, and SSIM. The program then outputs a comma separated Table with one line for each image that is then easily viewed in a spreadsheet.

6.5 Discussion

6.5.1 Stopping Criteria

The stopping criteria is determined by analyzing the IQ after a certain number of iterations. This proves to be a very challenging issue due to the fact that different image types will require a different number of iterations to produce the best reconstructions. Image types with very little noise require many more iterations, however, the reconstruction IQ improves greatly as well. This improvement is probably due to the fundamental aspect of CS reconstruction theory of sparsity. When higher sparsity can be achieved in transforming these images, CS has a greater probability of better reconstructions with less data. However, most real images do have noise.

Despite this dilemma, a solution can be observed where enough iterations can be achieved

that produces a result that will not change IQ significantly, even if more iterations are performed. Then this number of iterations can be used for other similar tests. This aspect of CS highlights the importance of performing an analysis for determining the best stopping criteria for specific image types and algorithms.

6.6 Other Algorithms

There is a way to get ℓ_1 -Magic to work with different sparsifying transforms. The code for ℓ_1 -Magic with the Gabor transform and reweighted ℓ_1 minimization is available (thanks to the author Michel Wakin). However, since the example is in Matlab, it needed to be converted to C++ for a quicker comparison with the other sparsifying transforms and optimization algorithms. Another option when using radial sampling is to implement gridding for either radial 2D or 3D with sparsifying transforms in the framework.

6.6.1 Computation Time with FastTestCS

The choice to develop FastTestCS came out of the necessity to speed up reconstructions using a more complex TIWT in CS with larger 3D data sets. Table 6.1 proves a considerable speedup is achieved by coding in a compiled language like C++ which is a more realistic time measurement of reconstruction than using less optimized software. Having faster software contributes to more research opportunities such as being able to compare more sampling patterns, finding better stopping criteria, and performing higher resolution experiments with more images. For instance, comparing the CS reconstruction time of the TOF MIP, which uses +1500 images (256×256), the Wavelab TIWT implementation in Matlab would have taken approximately 10 days to reconstruct, whereas, with FastTestCS, this same reconstruction takes 4 hours on a single CPU core. Additionally, by adding parallelization, with five quad-core machines, this reconstruction takes less than 15 minutes.

6.6.2 Use of FastTestCS

FastTestCS may be customized to include any sampling pattern and sparsifying transform to test different scenarios and compare reconstructions. Having consistent CS input data and algorithms to test various options provides a clear analysis and simulation tool with unlimited possibilities for future enhancements and tests. The standalone executable and command line interface provides a convenient way to parallelize on different CPU cores and machines to quickly test across various inputs and options in an automated way. Contact the author for software availability to obtain reproducible results, usage and advancement.

Chapter 7

Proof of Bounds on D-RIP Constants

7.1 Introduction

Proofs of some bounds on RIP constants have been provided by Candès *et al.* [20] and then improved for sharp bounds by Cai *et al.* [9]. An extension of the RIP is to accommodate redundant dictionaries with D-RIP for CS. D-RIP has been proposed by Candès *et al.* [13] with a proof of some bounds with cases of interest. The proof by Cai *et al.* [10] and [9] for sharp bounds in RIP does not apply to D-RIP; therefore, can the proof of Candès *et al.* [13] be improved to cover the sharp sparsity bounds for D-RIP? Bounds on D-RIP isometry constants have been proven by Candès *et al.* [13], specifically $\delta_{2k} \leq 0.08$. Counter examples for sharp bounds have also been reported in Cai *et al.* [10] and Davies *et al.* [26]. What new sharp bounds could be proven? This research aims at providing a proof and simulated experimental results that show how new sharp bounds can be achieved.

7.1.1 Contribution

The proof in Section 7.2 establishes an improved D-RIP bound which states that $\delta_{2k} < \frac{2}{3}$. This result was previously available [5] and it has been improved on by Wu and Li [51]. The main ingredient of the proof in Section 7.2 is a tool developed by Xu *et al.* [54]. This approach takes its inspiration from the clever ideas of Cai and Zhang [10] and [9].

7.2 New D-RIP Bounds

Theorem 1 *Let D be an arbitrary tight frame and let Φ be a measurement matrix satisfying D-RIP with $\delta_{2k} < \frac{2}{3}$. Then the solution $\hat{\beta}$ to (2.4) satisfies*

$$\|\beta - \hat{\beta}\|_2 \leq C_0 \varepsilon + C_1 \frac{\|D^* \beta - (D^* \beta)_{\max(k)}\|_1}{\sqrt{k}} \quad (7.1)$$

where the constants C_0 and C_1 depend on δ_{2k} , $(D^* \beta)_{\max(k)}$ is the vector $D^* \beta$ with all but the k largest components (in magnitude) set to zero.

Before proving this theorem, some remarks are helpful. Firstly, Cai and Zhang have obtained a sharp bound $\delta_{2k} < \frac{\sqrt{2}}{2}$ for the case $D = I$ [9]. Their work inspired pursuit of the bound in this proof. Secondly, following the ideas of Cai and Zhang [10] and [9], more general results (other D-RIP bounds) can be obtained in parallel.

In order to prove Theorem 1, the following ℓ_1 -norm invariant convex k -sparse decomposition by Xu and Xu [54], and Cai and Zhang [9] are needed.

The following description is taken from Xu *et al.* [54].

Lemma 1 *For positive integers $k \leq n$, and positive constant C , let $v \in \mathbb{R}^n$ be a vector with $\|v\|_1 \leq C$ and $\|v\|_\infty \leq \frac{C}{k}$. Then there are k -sparse vectors w_1, \dots, w_M with*

$$\|w_t\|_1 = \|v\|_1 \quad \text{and} \quad \|w_t\|_\infty \leq \frac{C}{k} \quad \text{for } t = 1, \dots, M, \quad (7.2)$$

such that

$$v = \sum_{t=1}^M x_t w_t \quad (7.3)$$

for some nonnegative real numbers x_1, \dots, x_M with $\sum_{t=1}^M x_t = 1$.

Now Theorem 1 is proven. This proof follows some ideas in the proofs of Theorems 1.1 and 2.1 by Cai *et al.* [9], incorporating some simplified steps. Some strategies from Cai *et al.*[11] and [12] are also used. This proof deals only with the δ_{2k} case so that the key ideas can be conveyed clearly. Let $h = \hat{\beta} - \beta$. For a subset $S \subset \{1, 2, \dots, d\}$, denoted by D_S the matrix D is restricted to the columns indexed by S (and replacing other columns by zero vectors). Let Ω denote the

index set of the largest k components of $D^*\beta$ (in magnitude), i.e., $(D^*\beta)_{\max(k)} = D_{\Omega}^*\beta$. With this notation there is $D_{\Omega^c}^*\beta = D^*\beta - (D^*\beta)_{\max(k)}$. As in Candès *et al.* [13], one can easily verify the following:

$$\|D_{\Omega^c}^*h\|_1 \leq 2\|D_{\Omega^c}^*\beta\|_1 + \|D_{\Omega}^*h\|_1 \quad (7.4)$$

$$\|\Phi h\|_2 < 2\varepsilon \quad (7.5)$$

Denote $v_i = \langle D_i, h \rangle$ for $i = 1, \dots, d$, where D_i is the i -th column of D , then

$$D^*h = (v_1, \dots, v_d)^\top. \quad (7.6)$$

By rearranging the columns of D if necessary, assume $|v_1| \geq |v_2| \geq \dots \geq |v_d|$. Let $T = \{1, 2, \dots, k\}$. In this case, have

$$D_T^*h = (v_1, \dots, v_k, 0, \dots, 0)^\top \text{ and } D_{T^c}^*h = (0, \dots, 0, v_{k+1}, \dots, v_d)^\top. \quad (7.7)$$

Assume that the tight frame D is normalized, i.e., $DD^* = I$, and $\|x\|_2 = \|D^*x\|_2$ for all $x \in \mathbb{R}^p$.

Thus there is the following useful relation:

$$\begin{aligned} \langle DD_T^*h, DD_{T^c}^*h \rangle &= \langle DD_T^*h, DD^*h - DD_T^*h \rangle = \langle DD_T^*h, h \rangle - \|DD_T^*h\|_2^2 \\ &= \langle D_T^*h, D^*h \rangle - \|DD_T^*h\|_2^2 \\ &= \|D_T^*h\|_2^2 - \|DD_T^*h\|_2^2. \end{aligned} \quad (7.8)$$

From the facts

$$\|D_{\Omega}^*h\|_1 \leq \|D_T^*h\|_1 \quad (7.9)$$

and

$$\|D_{\Omega}^*h\|_1 + \|D_{\Omega^c}^*h\|_1 = \|D_T^*h\|_1 + \|D_{T^c}^*h\|_1 = \|D^*h\|_1, \quad (7.10)$$

the relation

$$\|D_{\Omega^c}^*h\|_1 \leq 2\|D_{\Omega^c}^*\beta\|_1 + \|D_{\Omega}^*h\|_1 \quad (7.11)$$

yields

$$\|D_{T^c}^*h\|_1 \leq 2\|D_{\Omega^c}^*\beta\|_1 + \|D_T^*h\|_1 \quad (7.12)$$

Since

$$\|D_{TC}^* h\|_\infty \leq \frac{\|D_T^* h\|_1}{k} \leq \frac{2\|D_{\Omega^C}^* \beta\|_1 + \|D_T^* h\|_1}{k}, \quad (7.13)$$

use Lemma 1 to get the following ℓ_1 -invariant convex k -sparse decomposition of $D_{TC}^* h$:

$$D_{TC}^* h = \sum_{t=1}^M x_t w_t, \quad (7.14)$$

with each $w_t \in \mathbb{R}^d$ being k -sparse,

$$\|w_t\|_1 = \|D_{TC}^* h\|_1 \quad (7.15)$$

$$\|w_t\|_\infty \leq \frac{2\|D_{\Omega^C}^* \beta\|_1 + \|D_T^* h\|_1}{k}. \quad (7.16)$$

From this and the Cauchy-Schwartz inequality, there is

$$\sum_{t=1}^M x_t \|w_t\|_2 \leq \frac{2\|D_{\Omega^C}^* \beta\|_1 + \|D_T^* h\|_1}{\sqrt{k}} \leq \frac{2\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \|D_T^* h\|_2. \quad (7.17)$$

By the triangle inequality, $\|D_{TC}^* h\|_2 \leq \sum_{t=1}^M x_t \|w_t\|_2$ holds and thus

$$\|D_{TC}^* h\|_2 \leq \frac{2\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \|D_T^* h\|_2. \quad (7.18)$$

Note that

$$\|\beta - \hat{\beta}\|_2^2 = \|h\|_2^2 = \|D^* h\|_2^2 = \|D_{TC}^* h\|_2^2 + \|D_T^* h\|_2^2 \quad (7.19)$$

and

$$D^* \beta - (D^* \beta)_{\max(k)} = D_{\Omega^C}^* \beta. \quad (7.20)$$

In order to prove the theorem, it suffices to show that there are constants C'_0, C'_1 such that

$$\|D_T^* h\|_2 \leq C'_0 \varepsilon + C'_1 \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}}. \quad (7.21)$$

In fact, assuming (7.21) there is

$$\|h\|_2 = \sqrt{\|D_T^* h\|_2^2 + \|D_{TC}^* h\|_2^2} \quad (7.22)$$

$$\begin{aligned} &\leq \sqrt{\left(C'_0 \varepsilon + C'_1 \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}}\right)^2 + \left(\frac{2\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \|D_T^* h\|_2\right)^2} \\ &\leq C'_0 \varepsilon + C'_1 \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \frac{2\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \|D_T^* h\|_2 \end{aligned} \quad (7.23)$$

$$= 2C'_0 \varepsilon + 2(C'_1 + 1) \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} \quad (7.24)$$

Now moving to the proof of (7.21). Denote

$$\Pi := |\langle \Phi DD_T^* h, \Phi h \rangle| = |\langle \Phi DD_T^* h, \Phi DD^* h \rangle|. \quad (7.25)$$

First, as $D_T^* h$ is k sparse, hence $2k$ sparse. There is

$$\Pi \leq \|\Phi DD_T^* h\|_2 \|\Phi h\|_2 \leq \sqrt{1 + \delta_{2k}} \|D_T^* h\|_2 2\varepsilon. \quad (7.26)$$

On the other hand, as each $D_T^* h + w_t$ is $2k$ sparse, there is

$$\begin{aligned} \Pi &= |\langle \Phi DD_T^* h, \Phi DD_T^* h + \Phi DD_{T^c}^* h \rangle| = \left| \sum_{t=1}^M x_t \langle \Phi DD_T^* h, \Phi DD_T^* h + \Phi Dw_t \rangle \right| \\ &= \left| \sum_{t=1}^M x_t \left\langle \left(\Phi DD_T^* h + \frac{1}{2} \Phi Dw_t \right) - \frac{1}{2} \Phi Dw_t, \left(\Phi DD_T^* h + \frac{1}{2} \Phi Dw_t \right) + \frac{1}{2} \Phi Dw_t \right\rangle \right| \\ &= \sum_{t=1}^M x_t \left(\left\| \Phi DD_T^* h + \frac{1}{2} \Phi Dw_t \right\|_2^2 - \left\| \frac{1}{2} \Phi Dw_t \right\|_2^2 \right) \\ &\geq \sum_{t=1}^M x_t \left((1 - \delta_{2k}) \left\| DD_T^* h + \frac{1}{2} Dw_t \right\|_2^2 - (1 + \delta_{2k}) \left\| \frac{1}{2} Dw_t \right\|_2^2 \right) \end{aligned} \quad (7.27)$$

$$\begin{aligned} &= (1 - \delta_{2k}) \|DD_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \sum_{t=1}^M x_t \|Dw_t\|_2^2 + (1 - \delta_{2k}) \sum_{t=1}^M x_t \langle DD_T^* h, Dw_t \rangle \\ &= (1 - \delta_{2k}) \|DD_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \sum_{t=1}^M x_t \|Dw_t\|_2^2 + (1 - \delta_{2k}) \langle DD_T^* h, DD_{T^c}^* h \rangle \end{aligned} \quad (7.28)$$

$$\stackrel{(7.8)}{=} (1 - \delta_{2k}) \|DD_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \sum_{t=1}^M x_t \|Dw_t\|_2^2 + (1 - \delta_{2k}) (\|D_T^* h\|_2^2 - \|DD_T^* h\|_2^2)$$

$$= (1 - \delta_{2k}) \|D_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \sum_{t=1}^M x_t \|Dw_t\|_2^2$$

$$\geq (1 - \delta_{2k}) \|D_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \sum_{t=1}^M x_t \|w_t\|_2^2$$

$$\geq (1 - \delta_{2k}) \|D_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \left(\frac{2 \|D_{\Omega^c}^* \beta\|_1}{\sqrt{k}} + \|D_T^* h\|_2 \right)^2 \quad (7.29)$$

$$= (1 - \frac{3}{2} \delta_{2k}) \|D_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \left(\frac{4 \|D_{\Omega^c}^* \beta\|_1^2}{k} + \frac{4 \|D_{\Omega^c}^* \beta\|_1 \|D_T^* h\|_2}{\sqrt{k}} \right). \quad (7.30)$$

Combining this with (7.26) shows

$$(1 - \frac{3}{2} \delta_{2k}) \|D_T^* h\|_2^2 - \frac{1}{2} \delta_{2k} \left(\frac{4 \|D_{\Omega^c}^* \beta\|_1^2}{k} + \frac{4 \|D_{\Omega^c}^* \beta\|_1 \|D_T^* h\|_2}{\sqrt{k}} \right) \leq \sqrt{1 + \delta_{2k}} \|D_T^* h\|_2 2\varepsilon. \quad (7.31)$$

By making a perfect square, there is

$$\begin{aligned} & \left(\|D_T^* h\|_2 - \left(\frac{2\sqrt{1+\delta_{2k}}}{3(\frac{2}{3}-\delta_{2k})} \varepsilon + \frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} \right) \right)^2 \\ & \leq \left(\frac{2\sqrt{1+\delta_{2k}}}{3(\frac{2}{3}-\delta_{2k})} \varepsilon + \frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} \right)^2 + \left(\sqrt{\frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})}} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} \right)^2, \end{aligned} \quad (7.32)$$

which implies that

$$\begin{aligned} & \|D_T^* h\|_2 - \left(\frac{2\sqrt{1+\delta_{2k}}}{3(\frac{2}{3}-\delta_{2k})} \varepsilon + \frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} \right) \\ & \leq \frac{2\sqrt{1+\delta_{2k}}}{3(\frac{2}{3}-\delta_{2k})} \varepsilon + \frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}} + \sqrt{\frac{2\delta_{2k}}{3(\frac{2}{3}-\delta_{2k})}} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}}. \end{aligned} \quad (7.33)$$

and finally have (7.21):

$$\|D_T^* h\|_2 \leq \frac{4\sqrt{1+\delta_{2k}}}{3(\frac{2}{3}-\delta_{2k})} \varepsilon + \frac{4\delta_{2k} + \sqrt{6\delta_{2k}(\frac{2}{3}-\delta_{2k})}}{3(\frac{2}{3}-\delta_{2k})} \frac{\|D_{\Omega^C}^* \beta\|_1}{\sqrt{k}}. \quad (7.34)$$

This demonstrates the use of Lemma 1 to get a good result. This could be pursued further to general cases for an even better bound. Indeed, this has been done recently by Wu and Li [51] to improve the result of this proof which has been available previously [5].

Chapter 8

Conclusion

An in-depth coverage of several sparsifying transforms for CS in MRI is presented. A novel proof of the D-RIP sparsity constant bound, δ_{2k} , is given to address the need for understanding the use of redundant dictionaries in CS. A thorough investigation is presented for redundant sparsifying transforms in CS for MRI due to the improved quality they provide over the state of the art. This research covers compression tests with several settings and filters for determining transform robustness. Several good sparsifying transforms are simulated in a CS reconstruction with MRI data. Different under-sampling patterns, reduction factors and images are utilized for a broad range of tests. Results prove that redundant transforms that have a TI quality perform exceedingly well. Of note, the TIWT has a fast implementation and provides robust IQ improvements over the Orthogonal WT in CS MRI simulations.

FastTestCS simulation software is developed which performs CS simulations and reconstructions significantly faster than prototype Matlab software. Several simulations are compared with the various algorithms and settings that can be adjusted. It is available to aid in this research at <http://people.uwm.edu/gxu4uwm/fasttestcs>.

8.1 Future work

8.2 Best Practice Definition

Some future work would develop the best way for using a combination of all the quantitative image measures to give credible across the board evidence that IQ is measured accurately or

improved. Another look at the compression experiments that base the threshold percentage on the number of coefficients rather than the number of pixels might give additional insight into which transform will perform well. Those results may be more applicable for CS and encourage the use of more redundant transforms in CS.

8.3 More Sparsifying Transforms

A literature search found over 300 articles in just the journal MRM on CS MRI, and every few weeks a new article or two appears. This is a vast amount of research! It would be important to compare other new sparsifying transforms that work well in MRI. Here is a list of additional ideas and transforms for future investigation.

1. TI wavelets with patches.
2. Better contourlets?
3. Contourlets on patches?
4. Formulate a better simulation with dynamic dots which can be used to identify better conclusions why some transforms performed better than others.
5. Continue literature search for new developments in sparse transforms for consideration.
6. Develop a better understanding of the theory: Why do transforms on small patches perform better than transforms on full images?
7. Consider wavelet filter adjustment for optimal compression. This would be another type of learned transform. Are there other learned dictionaries that are important?
8. For unconstrained CS, analyze the combination of regularization techniques for optimal balancing between techniques, while keeping in mind the number of iterations affecting this balance.
9. Perform mean normalization on the reconstructed images for an improvement in contrast and error.

10. For patch transforms, change overlapping factors and observe block artifacts and IQ.
11. Compile a list of good filters and settings for wavelets, contourlets and curvelets.
12. Implement the combined wavelet+contourlet transform with patches.
13. Implement the curvelet transform on patches.
14. Analyze standard deviation of image error vs. reduction factor as a measure of robustness.

8.3.1 FastTestCS: Enhancement, Versatility and Distribution

To enhance the FastTestCS software framework, more functionality and algorithms can be added and a more standardized API. It may be beneficial to investigate GPU opportunities for additional speed up. For more product exposure and capability, integrating into a GE Orchestra reconstruction would be a great advancement.

BIBLIOGRAPHY

- [1] Aharon, Michal, Michael Elad, and Alfred Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." *IEEE Transactions on signal processing* 54, no. 11 (2006): 4311-4322.
- [2] Flint, Alexander C. "Determining Optimal Medical Image Compression: Psychometric and Image Distortion Analysis." *BMC Medical Imaging* 12, no. 1 (2012): 1.
- [3] Aybat, Necdet Serhat. "First Order Methods for Large-scale Sparse Optimization." PhD diss., *Columbia University*, 2011.
- [4] Baker, Christopher A. "Sparse Representation by Frames with Signal Analysis." *Journal of Signal and Information Processing* 7, no. 01 (2016): 39.
- [5] Baker, Christopher A., "A Note on Sparsification by Frames", *arXiv* :1308.5249v1 [cs.IT] 23 Aug 2013.
- [6] Baker, Christopher A., Kevin King, Dong Liang, and Leslie Ying. "Translational Invariant Dictionaries for Compressed Sensing in Magnetic Resonance Imaging." *In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, (2011): 1602-1605.
- [7] Baker, Christopher A., Kevin King, Dong Liang, and Leslie Ying. "Investigation of Sparsifying transforms for Compressed Sensing in MRI Reconstruction", *ISMRM*, (2009): 4583.
- [8] Bigot, Jérémie, Claire Boyer, and Pierre Weiss. "An Analysis of Block Sampling Strategies in Compressed Sensing." *IEEE Transactions on Information Theory* 62, no. 4 (2016): 2125-2139.
- [9] Cai, T. Tony, and Anru Zhang. "Sparse Representation of a Polytope and Recovery of Sparse Signals and Low-rank Matrices." *IEEE Transactions on Information Theory* 60, no. 1 (2014): 122-132.
- [10] Cai, T. Tony, and Anru Zhang. "Sharp RIP Bound for Sparse Signal and Low-rank Matrix Recovery." *Applied and Computational Harmonic Analysis* 35, no. 1 (2013): 74-93.
- [11] Cai, T. Tony, Lie Wang, and Guangwu Xu. "New Bounds for Restricted Isometry Constants." *IEEE Transactions on Information Theory* 56, no. 9 (2010): 4388-4394.
- [12] Cai, T. Tony, Guangwu Xu, and Jun Zhang. "On Recovery of Sparse Signals via Minimization." *IEEE Transactions on Information Theory* 55, no. 7 (2009): 3388-3397.
- [13] Candès, Emmanuel J., Yonina C. Eldar, Deanna Needell, and Paige Randall. "Compressed Sensing with Coherent and Redundant Dictionaries." *Applied and Computational Harmonic Analysis* 31, no. 1 (2011): 59-73.

- [14] Candès, Emmanuel J. "The Restricted Isometry Property and its Implications for Compressed Sensing." *Comptes Rendus Mathématique* 346, no. 9 (2008): 589-592.
- [15] Candès, Emmanuel J., and Michael B. Wakin. "An Introduction to Compressive Sampling." *IEEE Signal Processing Magazine* 25, no. 2 (2008): 21-30.
- [16] Candès, Emmanuel J., Michael B. Wakin, and Stephen P. Boyd. "Enhancing Sparsity by Reweighted ℓ_1 Minimization." *Journal of Fourier Analysis and Applications* 14, no. 5-6 (2008): 877-905.
- [17] Candès, Emmanuel, and Justin Romberg. "Sparsity and Incoherence in Compressive Sampling." *Inverse problems* 23, no. 3 (2007): 969.
- [18] Candès, Emmanuel J., Justin K. Romberg, and Terence Tao. "Stable Signal Recovery from Incomplete and Inaccurate Measurements." *Communications on Pure and Applied Mathematics* 59, no. 8 (2006): 1207-1223.
- [19] Candès, Emmanuel J., Laurent Demanet, David Donoho, and Lexing Ying. "Fast Discrete Curvelet Transforms." *Multiscale Modeling & Simulation* 5, no. 3 (2006): 861-899.
- [20] Candès, Emmanuel J., and Terence Tao. "Decoding by Linear Programming." *IEEE Transactions on Information Theory* 51, no. 12 (2005): 4203-4215.
- [21] Candès, Emmanuel, and Justin Romberg. "l1-magic: Recovery of sparse signals via convex programming." URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf (2005).
- [22] Castets, Charles R., William Lefrançois, Didier Wecker, Emeline J. Ribot, Aurélien J. Trotier, Eric Thiaudière, Jean-Michel Franconi, and Sylvain Miraux. "Fast 3D Ultrashort Echo-time Spiral Projection Imaging using Golden-angle: A Flexible Protocol for In Vivo Mouse Imaging at High Magnetic Field." *Magnetic Resonance in Medicine* (2016).
- [23] Chan, Rachel W., Elizabeth A. Ramsay, Edward Y. Cheung, and Donald B. Plewes. "The Influence of Radial Undersampling Schemes on Compressed Sensing Reconstruction in Breast MRI." *Magnetic Resonance in Medicine* 67, no. 2 (2012): 363-377.
- [24] Coifman, Ronald R., and David L. Donoho. *Translation-invariant de-noising*. Springer, New York, 1995.
- [25] Coifman, Ronald R., and David L. Donoho. "Translation-Invariant De-Noising", *Wavelets and Statistics, Lect. Notes in Statist.*, vol. 103, 125-150, Springer-Verlag, New York, 1995.
- [26] Davies, Michael E., and Rémi Gribonval. "Restricted Isometry Constants where ℓ_p Sparse Recovery can Fail for $0 < p \leq 1$ ", *IEEE Trans. Inf. Theory*, vol. 55, (2009): 2203-2214.
- [27] Deshmane, Anagha, Vikas Gulani, Mark A. Griswold, and Nicole Seiberlich. "Parallel MR Imaging." *Journal of Magnetic Resonance Imaging* 36, no. 1 (2012): 55-72.
- [28] Do, Minh N., and Martin Vetterli, Welland, Grant, ed. *Beyond Wavelets*. Vol. 10. Academic Press, 2003.
- [29] Donoho, D., A. Maleki, and M. Shahram. "Wavelab 850." *Software Toolkit for Time-frequency Analysis* (2006).
- [30] Ersoz, Ali, and L. Tugan Muftuler. "Undersampled Linogram Trajectory for Fast Imaging (ULTI): Experiments at 3T and 7T." *NMR in Biomedicine* 29, no. 3 (2016): 340-348.

- [31] Goh, Say Song. *Gabor and Wavelet Frames*. Vol. 10. World Scientific, 2007.
- [32] Gough, Brian. *GNU Scientific Library Reference Manual*. Network Theory Ltd., 2009.
- [33] Grant, Michael, Stephen Boyd, and Yinyu Ye. "CVX: Matlab Software for Disciplined Convex Programming." <http://cvxr.com>, (2008). IEEE, 2006.
- [34] Joint Photographic Experts Group. "JPEG Home Page." The JPEG Committee Home Page (2006).
- [35] Kim, Yookyung, M. Altbach, T. Trouard, and Ali Bilgin. "Compressed Sensing Using Dual-tree Complex Wavelet Transform." *In Proceedings of the International Society for Magnetic Resonance in Medicine*, vol. 17, (2009): 2814.
- [36] King, Kevin F. "Spiral Scanning with Anisotropic Field of View." *Magnetic Resonance in Medicine* 39, no. 3 (1998): 448-456.
- [37] Liu, Jing, and David Saloner. "Accelerated MRI with CIRCular Cartesian UnderSampling (CIRCUS): a Variable Density Cartesian Sampling Strategy for Compressed Sensing and Parallel Imaging." *Quantitative Imaging in Medicine and Surgery* 4, no. 1 (2014): 57-67.
- [38] Lin, Junhong, Song Li, and Yi Shen. "New Bounds for Restricted Isometry Constants with Coherent Tight Frames." *IEEE Transactions on Signal Processing* 61, no. 3 (2013): 611-621.
- [39] Lu, Yue, and Minh N. Do. "CRISP Contourlets: a Critically Sampled Directional Multiresolution Image Representation." *In Optical Science and Technology, SPIE's 48th Annual Meeting*, International Society for Optics and Photonics, (2003): 655-665.
- [40] Lustig, Michael, David Donoho, and John M. Pauly. "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging." *Magnetic Resonance in Medicine* 58, no. 6 (2007): 1182-1195.
- [41] Lustig, Michael, Juan M. Santos, David L. Donoho, and John M. Pauly. "kt SPARSE: High Frame Rate Dynamic MRI Exploiting Spatio-temporal Sparsity." *In Proceedings of the 13th Annual Meeting of ISMRM*, Seattle, (2006): 2420.
- [42] Ma, Shiqian, Wotao Yin, Yin Zhang, and Amit Chakraborty. "An Efficient Algorithm for Compressed MR Imaging using Total Variation and Wavelets." *In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1-8. IEEE, 2008.
- [43] MathWorks, Inc. *MATLAB: the Language of Technical Computing. Desktop Tools and Development Environment*, version 7. Vol. 9. MathWorks, 2005.
- [44] Ning, Bende, Xiaobo Qu, Di Guo, Changwei Hu, and Zhong Chen. "Magnetic Resonance Image Reconstruction Using Trained Geometric Directions in 2D Redundant Wavelets Domain and Non-convex Optimization." *Magnetic Resonance Imaging* 31, no. 9 (2013): 1611-1622.
- [45] Nocedal, Jorge, and Stephen Wright. "Conjugate Gradient Methods" in *Numerical Optimization*. Springer Science & Business Media, 2006.
- [46] Polak, Adam C., Marco F. Duarte, and Dennis L. Goeckel. "Performance Bounds for Grouped Incoherent Measurements in Compressive Sensing." *IEEE Transactions on Signal Processing* 63, no. 11 (2015): 2877-2887.

- [47] Sadeghi, Mostafa, Massoud Babaie-Zadeh, and Christian Jutten. "Learning Overcomplete Dictionaries Based on Parallel Atom-updating." *In 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, (2013): 1-5.
- [48] Sattar, Farook, Lars Floreby, Göran Salomonsson, and Benny Lovstrom. "Image Enhancement Based on a Nonlinear Multiscale Method." *IEEE Transactions on Image Processing* 6, no. 6 (1997): 888-895.
- [49] Vetterli, Martin, and Jelena Kovacevic. *Wavelets and Subband Coding*. No. LCAV-BOOK-1995-001. Prentice-Hall, 1995.
- [50] Wang, Zhou, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. "Image Quality Assessment: from Error Visibility to Structural Similarity." *IEEE Transactions on Image Processing* 13, no. 4 (2004): 600-612.
- [51] Wu, Fen-Gong, and Dong-Hui Li. "The Restricted Isometry Property for Signal Recovery with Coherent Tight Frames." *Bulletin of the Australian Mathematical Society* 92, no. 03 (2015): 496-507.
- [52] Wu, B., R. Watts, R. Millane, and P. Bones. "An Improved Approach in Applying Compressed Sensing in Parallel MR Imaging." <http://hdl.handle.net/10092/3780> (2009).
- [53] G. Xu, and Z. Xu, "Compressed Sensing Matrices from Fourier Matrices" arXiv:1301.0.73v1 [cs.IT] 3 Jan 2013.
- [54] Xu, Guangwu, and Zhiqiang Xu. "On the ℓ_1 -Norm Invariant Convex k -Sparse Decomposition of Signals." *Journal of the Operations Research Society of China* 1, no. 4 (2013): 537-541.

Appendix A

The Gradient of the Regularization Function

Calculating the gradient of the regularization function $\nabla \|Wx\|_1$ is necessary in using the conjugate gradient solver. There are three ways to calculate this gradient of a complex function and a transform:

1. Apply the transform to the complex data and then take the magnitude.

$$\begin{aligned}\sum |Wx| &= \sum |W(x_r + ix_i)| \\ \nabla \sum |Wx| &= \sum \partial \frac{|W(x_r + ix_i)|}{\partial x_r} + \partial \frac{|W(x_r + ix_i)|}{\partial x_i} \\ \nabla \sum |Wx| &= \sum W^T \frac{Wx_r}{|W(x_r + ix_i)|} + W^T \frac{Wx_i}{|W(x_r + ix_i)|}\end{aligned}$$

2. Apply the transform to the magnitude of the image.

$$\begin{aligned}\sum |Wx| &= \sum |W|(x_r + ix_i)| \\ \nabla \sum |Wx| &= \sum \partial \frac{|W|(x_r + ix_i)|}{\partial x_r} + \partial \frac{|W|(x_r + ix_i)|}{\partial x_i} \\ \nabla \sum |Wx| &= \sum \frac{x_r}{|(x_r + ix_i)|} W^T \text{sign}(W|x_r + ix_i|) + \frac{x_i}{|(x_r + ix_i)|} W^T \text{sign}(W|x_r + ix_i|)\end{aligned}$$

3. Apply the transform to the real parts and imaginary parts separately.

$$\begin{aligned}\sum |Wx| &= \sum (|Wx_r| + |Wx_i|) \\ \nabla \sum |Wx| &= \sum \partial \frac{|Wx_r|}{\partial x_r} + \partial \frac{|Wx_i|}{\partial x_i} \\ \nabla \sum |Wx| &= \sum W^T \text{sign}(Wx_r) + W^T \text{sign}(Wx_i)\end{aligned}$$

where T is the transpose. According to the sum of error measurements, the third approach performs the best with wavelets for regularization. These derivations can be applied with other sparsifying transforms; however, if the transform is not orthogonal, the transpose may not be equivalent to the inverse. For TV, the three approaches can also be used; however, the derivations will be different because a transform is not used.

Appendix B

MRI Dataset Parameters

1. Single Channel T1-weighted Brain, 3.0 T, axial, 2D T1-weighted spin echo, single channel quadrature transmit/receive head coil, TE = min full, TR = 700, 25 kHz, 22 cm FOV, 256 × 256, 5 mm, skip 0, 10 slices, Fully sampled
2. Eight Channel T1-Weighted Brain, 3.0 T, axial, 2D T1-weighted spin echo, 8-channel receive only head coil TE = min full, TR = 700, 25 kHz, 22 cm FOV, 256 × 256, 5 mm, skip 0, 10 slices Fully sampled
3. Eight Channel 3D Gradient Echo TOF head, 3.0 T, axial, 3D gradient echo, 8-channel receive only head coil, Flow comp, flip = 25, min TE, TR = 38, 16 kHz, 22 cm FOV, 384 × 224, 1. 4 mm, 32 slices / slab, 3 slabs, fully sampled

Appendix C

Software References for Different Transforms Used

1. FastTestCS

- <http://people.uwm.edu/gxu4uwm/fasttestcs>.

2. Contourlet Toolbox

- <http://www.ifp.uiuc.edu/~minhdo/software/>

3. CRISP Contourlet toolbox

- http://www.uam.es/personal_pdi/ciencias/mtc/software.htm

4. Wavelet Toolbox

- <http://www-stat.stanford.edu/~wavelab/WaveLab701.html>
- <http://www.mathworks.com/products/wavelet/>

5. Curvelet Toolbox

- <http://www.curvelet.org/software.html>

6. Discrete Cosine Transform (DCT)

- <http://www.mathworks.com/access/helpdesk/help/toolbox/signal/dct.html>

7. K-SVD, OMP, and over-complete DCT

- <http://www.cs.technion.ac.il/~elad/software/>
- <http://www.cs.technion.ac.il/~ronrubin/software.html>

8. ℓ_1 -Magic

- <http://statweb.stanford.edu/~candes/l1magic/>

9. GSL - GNU Scientific Library

- <http://www.gnu.org/software/gsl/>

CURRICULUM VITÆ

Christopher Adams Baker
Place of birth: Black River Falls, WI

Education B.S. Valparaiso University, May 2000
Major: Electrical Engineering

M.S. University of Wisconsin - Milwaukee, May 2005
Major: Electrical Engineering
Thesis: "The Design and Construction of a High Speed Near-Infrared Imaging System for Functional Noninvasive Bioimaging"

Dissertation Investigation of Sparsifying Transforms in Compressed Sensing for Magnetic Resonance Imaging with FastTestCS

Publications Baker, Christopher A. "Comparison of MRI Under-Sampling Techniques for Compressed Sensing with Translation Invariant Wavelets Using Fast Test CS: A Flexible Simulation Tool." Accepted: *Journal of Signal and Information Processing*, 2016.

Baker, Christopher A. "Sparse Representation by Frames with Signal Analysis." *Journal of Signal and Information Processing* 7, no. 01, 2016: 39.

Baker, Christopher A. "A Note on Sparsification by Frames." arXiv: 1308.5249v1 [cs.IT], 23 Aug. 2013.

Baker, Christopher A., Kevin King, Dong Liang, and Leslie Ying. "Translational Invariant Dictionaries for Compressed Sensing in Magnetic Resonance Imaging." Presentation, *In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1602-1605. IEEE, 2011.

Baker, Christopher A., Kevin King, Dong Liang, and Leslie Ying. "Investigation of Sparsifying transforms for Compressed Sensing in MRI Reconstruction." Poster, *ISMRM*, vol. p.4583, 2009.

Baker, Christopher A. and Kevin King. "Investigation of Sparsifying Transforms for Compressed Sensing in MRI." *General Electric Healthcare MR ASL Technical Note* 08-02, 2008.

Baker, Christopher A. "A High-Speed Near-Infrared Imaging System for Functional Noninvasive Bioimaging." 29th Annual Great Lakes Biomedical Conference, Racine, WI, 1st place student poster, 2005.

Research Internships

2006 & 2007 summer, internship at GE Healthcare CT ASL. Topics: Noise uniformity and filtering, image registration between slabs, patent proposal.

2008 summer and fall, internship at GE MRI ASL. Topics: compressed sensing sparsifying transform investigation with contourlets, wavelets and K-SVD.

2009, UWM research assistant compressed sensing MRI. Topics: Investigated sparse transforms of CS with K-SVD, DCT and wavelets. Created FastTestCS from Matlab in C++ to improve efficiency.

Employment

2011 - present, General Electric Healthcare, MRI Reconstruction Algorithm Software Engineer, Ownership: non-cartesian, cardiac and test automation.