A NOVEL FRAMEWORK FOR COMPRESSED SENSING BRAIN MRI EDGE DETECTION AND DE-NOISING

by

Rouzbeh Zamyadi

Bachelor of Engineering, Ryerson, 2008

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2011

©Rouzbeh Zamyadi 2011

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

A Novel Framework for Compressed Sensing Brain MRI Edge Detection and De-noising

Master of Applied Science 2011

Rouzbeh Zamyadi

Electrical and Computer Engineering

Ryerson University

In this thesis a novel edge detection technique is developed that employs compressed sensing image reconstruction techniques. The ability of compressed sensing noise reduction is combined with wavelet transforms, acting both as an sparsifying transform as well as an edge detection media. The proposed design was implemented and simulated on a brain phantom. The simulation results were provided for a variety of different sets of variables, and the differences were explained. The results obtained are compared with other edge detection techniques already in use. One important comparison criteria is the visual quality of images; according to which the proposed technique presents improved noise reduction and edge preservation. In addition to qualitative evaluation a method of quantitative measurement based on structural content is also utilized. It is found that the values for such a measure of the proposed method is 1.0755, 1.0174, and 0.5590 for Gaussian, Speckle, and Salt & Pepper noise types respectively. These results indicate that this novel method also improves edge preservation, while the visual quality inspection indicates how much noise has been suppressed.

Acknowledgements

First and foremost I would like to extend my gratitude to my thesis supervisor, Dr. Kaamran Raahemifar. His support and timely encouragements along with him giving me an opportunity to pursue my graduate studies under his supervision were all instrumental to the completion of this thesis.

I owe a great debt to my parents, for their continued love and everything they have given me not just during my graduate studies but all my life; to my sister and brother Mojdeh and Mehran for their support and love; and to Shahrzad for reminding me of the important things in life, and for her unconditional encouragements and support.

Finally, I would like to thank my thesis defense chair, Dr. Vadim Geurkov; and my defense committee members, Dr. Farah Mohammadi, Dr. Xavier Fernando, and Dr. Alagan Anpalagan for taking the time to read my thesis and for providing insightful comments from their individual perspectives that have improved this thesis.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Literature Review	2
1.3	Objective & Contributions	3
1.4	Organization	3
2	Compressed Sensing & MRI	5
2.1	Overview	5
2.2	Magnetic Resonance Imaging	5
2.2.1	Physics of Nuclear Magnetization	5
2.2.2	NMR signal detection	8
2.2.3	Spatial Localization & k-space	10
2.3	Compressed Sensing	14
2.3.1	Sparsity	15
2.3.2	Incoherence and Random Sampling	15
2.3.3	Recovery of Under-Sampled Signal	17
2.4	Compressed Sensing MRI	25
2.4.1	Magnetic Resonance Image Sparsity	25
2.4.2	Incoherent Sampling	25
2.4.3	Compressed Sensing Image Recovery	29
2.5	Summary	30
3	From Fourier to Wavelets	31
3.1	Overview	31
3.2	Fourier Transform	31
3.3	Short-Time Fourier Transform	32
3.4	Wavelet Transform	35
3.4.1	Discrete Wavelet Transform	38
3.4.2	Multiresolution Analysis	39
3.4.3	Orthogonal Wavelet Bases	41

3.5	Summary	48
4	Image Edge Detection	49
4.1	Overview	49
4.2	Edge Detection	49
4.2.1	Introduction to Edges	49
4.2.2	Conventional Edge Detection Methods	50
4.3	Wavelet Edge Detection	53
4.4	Brain MRI Edge Detection	57
4.5	Summary	57
5	Methodology	59
5.1	Overview	59
5.2	Design Requirements	59
5.3	Image Acquisition	60
5.3.1	Sampling Trajectory	60
5.3.2	Non-uniform Sampling	61
5.3.3	Image Collection	62
5.4	Edge Image Acquisition	64
5.4.1	CS Reconstruction	64
5.4.2	Sparsifying transform	68
5.4.3	De-noising	71
5.5	Evaluation	73
5.6	Summary	74
6	Results	75
6.1	Overview	75
6.2	Regularization Parameter	75
6.3	Wavelet Assessment	80
6.4	Pseudo-random Sampling	83
6.5	Noise Suppression	84
6.6	Summary	91
7	Conclusions & Future Work	93
7.1	Conclusion	93
7.2	Future Work	94
	References	107

List of Tables

5.1	Pseudo-code for a sample gradient descent algorithm.	67
5.2	Algorithm for backtracking line search.	67
5.3	Algorithm for solving of a convex optimization problem using conjugate gradient technique. . . .	68
6.1	SNR values for different wavelet members of different wavelet families.	80
6.2	SNR values for different ratio of samples collected using pseudo-random Cartesian sampling (i.e. 50% indicates that only half the samples required by Nyquist criterion are collected).	83
6.3	SNR values for different wavelet members of different wavelet families.	89

List of Figures

2.1	The precession motion of magnetization vector about an external static magnetic field.	7
2.2	The magnetization vector and its components under influence of $\vec{B}(t)$	7
2.3	(a) A pulse sequence for sample frequency encoding scanning, and (b) its corresponding Fourier trajectory on the frequency plane. G_{ss} is slice selection gradient (G_z in this case) and T_{aq} is the acquisition time for the Analog to Digital Converter (ADC).	11
2.4	(a) A pulse sequence for sample phase encoding scanning, and (b) its corresponding Fourier trajectory on the frequency plane. T_{ph} is the duration of the phase encoding pulse.	12
2.5	(a) A pulse sequence for a series of samples. Each pair of gradients moves to its corresponding Fourier trajectory on (b) the k-space. If all possible combinations of all 5 pulses are utilized, the resulting sampling trajectory would be a Cartesian grid of acquisitions.	13
2.6	MRI Image formation by taking the two dimensional Inverse Fourier Transform (IFT) of the k-space image.	14
2.7	A two dimensional Euclidean space with two arbitrary $\vec{A}$ and $\vec{B}$ and their corresponding displacement vector $\vec{r}$	20
2.8	(a) Visual representation of an ℓ_2 norm and (b) visual representation of an ℓ_1 norm. Dashed lines represent the unit circles.	21
2.9	(a) Visual representation of an ℓ_2 norm minimization; for ℓ_2 norm the sparse solution (s') to the null space is not found (b) visual representation of an ℓ_1 norm minimization; since ℓ_1 circle has fewer solutions (solid edges), this method has a high chance of finding the sparse solution.	24
2.10	A generalization of frequency domain of a signal, $F(t)$ represents the forward Fourier transform. (a): Original signal (time domain) sampled at Nyquist rate, the Fourier domain information contains all significant components and reconstruction can recover the original signal. (b): Samples collected at equally spaced intervals but at sub-Nyquist rate, the resulting frequency domain plot includes the combination of shifted frequency components; reconstruction of these components would not result in the original signal being recovered (the resulting signal would be a superposition of aliased signal copies). (c): Original signal sampled at sub-Nyquist rate with randomly (incoherent) spaced intervals between samples. This time the aliasing components in the frequency domain behave as noise; and the non-sparse coefficients can be recovered by thresholding above the noise ceiling.	26

2.11	After the first round of thresholding is performed (top-left), the stronger non-zero components are recovered. Consequently, it is possible to calculate their contributions to the original interference. After this is done (bottom-right), the resulting “noise” level will be lower, revealing the smaller non-sparse components; which can be recovered by setting another (lower) threshold (top-right). Threshold levels are also designed with respect to the existing interference level.	27
2.12	Results of different sampling patterns	29
3.1	A visual representation of the windowing process of a signal $f(t)$. Left: A window function $g_{u,\omega}$ of width T is centred at time u . The signal energy in this window is represented on a time-frequency plane (Right). The window is then moved and process repeated until the signal is finished. Right: Each time a segment of the signal is <i>windowed</i> it is represented on a time-frequency plane, via a Heisenberg box. The accumulation of these boxes for the whole signal makes up the spectrogram (note that all boxes are equal in terms of area).	33
3.2	An illustration of Heisenberg boxes on a time-frequency plane at different time and frequency selections. The widths of boxes are σ_t and σ_ω for time width and frequency width, respectively; where $\sigma_t\sigma_\omega = C$ is constant. Note that the energy peaks at the centre of the box and diminishes as it spreads out.	34
3.3	An illustration of Heisenberg boxes on a time-frequency plane for three different wavelet functions ψ_1, ψ_2 , and ψ_3 ; each with scaling and translation factors of u_1, u_2, u_3 , and s_1, s_2, s_3 , respectively. Note that $s_1 < s_2 < s_3$. Once again $\sigma_t\sigma_\omega = C$ is constant. Hence, as scale s decreases, so does the time support, but since the area of the box is constant, the frequency support is increased and moves toward higher frequency. On the other hand, when scale is increased, the time support also increases at the expense of frequency support at lower frequencies. In this manner the frequency response of the wavelet can change by adjusting the scale factors, this is crucial in filtering analysis of wavelets.	37
3.4	An illustration of the dyadic wavelet function windowing in the time frequency domain. Note that the frequency value is fixed and scaled down by factors of 2. The dyadic wavelet illustrates flexibility in terms of scale and frequency resolutions by being able to analyze a signal at different scale resolutions while selecting a certain frequency resolution.	40
3.5	(a) Analysis step of a signal $x[n]$; convolved with a high-pass filter $\bar{g}$ (retaining high frequency signal components) as well as a low pass filter $\bar{h}$ (decomposing low frequency signal components). After the filtering is performed the outputs are down-samples by a factor of 2; which results in retaining only the even entries (in the sequence). (b) Reconstruction step of the signal taking in the approximation and detail signals a and d , respectively, up-sampling them by a factor of two (see 3.54), and then filtering them with g and h ; summing the output of the filters in a piecewise manner results in the reconstruction of the original signal, $x[n]$	44
3.6	Coefficient pyramid for a 2D wavelet analysis by way of separable bases; detail images are denoted by d and the average image is denoted by a . there are 3 octaves (levels) in total with $j - 1$ as the first level. Each level contains 2^{2k} coefficients; where k is the number of the level.	46

3.7	Coefficient pyramid for a 2D wavelet analysis by way of separable bases; detail images are denoted by d and the average image is denoted by a . there are 3 octaves (levels) in total with $j - 1$ as the first level. Each level contains 2^{2k} coefficients; where k is the number of the level.	47
4.1	Different edge detection techniques.	56
5.1	Illustration of the difference between (a) a random Cartesian trajectory and (b) a variable density, non-uniform Cartesian trajectory. Note that in variable density pattern, most of the samples are collected from high energy components of the k-space, resulting in better reconstruction.	62
5.2	(a) An image of a Shepp-Logan phantom.	63
5.3	The Fourier transform (k-space) of a Shepp-Logan phantom. The Fourier transform produces (a) symmetric frequency plane with low frequency components located at the corners of the plane, the image needs to be (b) shifted in order to acquire the proper k-space as an MRI machine would sample.	63
5.4	Illustration of PDF functions for different sampling density (left) and their corresponding non-uniform Cartesian sampling trajectory (right): (a) a PDF function with sampling density of 30% of total number of samples, (b) the sampling trajectory to collect 30% of samples; (c) PDF for sampling density of 40% and (d) corresponding Cartesian trajectory; (e) PDF for 50% sampling and (f) variable density Cartesian trajectory for 50% sample collection.	65
5.5	The Daubechies wavelet function family from (a)(db4) to (f)(db14) the vanishing moments increase by one from figure to figure; where dbN is a Daubechies wavelet function with $N/2$ vanishing moments. Note how the support of the wavelet function is increased as the number of vanishing points increase. Only the first three wavelets in the family (i.e. db4, db6, and db8) are used in the results.	70
5.6	Illustration of the Symmlet family of wavelets with increasing vanishing moments from (a) to (d); where $symN$ has N vanishing moments. The number of vanishing points of a Symmlet wavelet is half its support. Note the symmetry of the wavelets, and its energy distribution; toward the middle of the support. Only the first three (i.e. sym4, sym6, and sym8 will be used in the design evaluation).	71
5.7	Coiflet family of wavelets, with (a) through (c) representing coif1, coif2, and coif4, respectively; where $coifN$ has $2 \times N$ vanishing moments. Note the energy distribution of the Coiflet family is concentrated toward the beginning of the support, much like the Daubechies family.	72
6.1	Graphs of SNR vs. λ_{TV} values, with a Canny edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).	76
6.2	Graphs of SNR vs. λ_{ψ} values, with a Canny edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).	76
6.3	Graphs of SNR vs. λ_{TV} values, with a multi-scale edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).	77

6.4	Graphs of SNR vs. λ_Ψ values, with a multi-scale edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).	77
6.5	Edge detected phantom images utilizing (a) Canny edge detection and (b) multi-scale edge detection.	78
6.6	Edge detected image using the proposed methods with λ_Ψ set to zero and λ_{TV} set to (a) 0.05 (b) 0.1 (c) 0.15, and (j) 0.50; images are from lower (coarse) resolution scale.	79
6.7	A fine-tuned edge image with λ_Ψ and λ_{TV} set to 0.05 and 0.15, respectively.	80
6.8	Edge detected image using the proposed methods with λ_{TV} set to zero and λ_Ψ set to (a) 0.05 (b) 0.1 (c) 0.15, and (d); images are from lower (coarse) resolution scale.	81
6.9	The control signal used for SNR evaluation henceforth.	82
6.10	Edge detection images from the proposed method for wavelet choices of (a) <i>Daubechies4</i> , (b) <i>Daubechies6</i> , and (c) <i>Daubechies8</i> , respectively.	82
6.11	Edge detection images from the proposed method for wavelet choices of (a) <i>Coiflet1</i> , (b) <i>Symmet4</i> , and (c) <i>Daubechies4</i> , respectively.	83
6.12	Edge detection images from the proposed method for different number of samples collected: (a) 50% of samples collected, (b) 40% of samples collected, and (c) 30% of samples collected.	84
6.13	Edge detection images from the proposed method for a Gaussian noise contaminated image. Different images with different regularization parameters.	85
6.14	Edge detection images from the proposed method for a Speckle noise contaminated image. Different images with different regularization parameters.	87
6.15	Edge detection images from the proposed method for a salt and pepper noise contaminated image. Different images with different regularization parameters.	88
6.16	A high resolution edge detected image of 3 rd octave for $\lambda_{TV} = 0.15$ and $\lambda_\Psi = 0.05$	90
A.1	Edge detection images from the proposed method for wavelets with different vanishing moments of the Coiflet family.	97
A.2	Edge detection images from the proposed method for wavelets with different vanishing moments of the Symmetlet family.	97
A.3	Edge detection images from the proposed method for wavelets with different vanishing moments of the Daubechies family.	98
A.4	The progression of edge detection images from $\lambda_{TV} = 0.05$ to $\lambda_{TV} = 0.60$ with $\lambda_\Psi = 0$	99
A.5	The progression of edge detection images from $\lambda_\Psi = 0.05$ to $\lambda_\Psi = 0.60$ with $\lambda_{TV} = 0$	100
A.6	A sample of the final iteration of the wavelet transform illustrating different detail images. The top left corner image is the low resolution approximation image.	101

List of Appendices

A Additional Processed Images

97

Chapter 1

Introduction

1.1 Motivation

The Magnetic Resonance Imaging has become a popular medical imaging tool in recent decades, partly because of its non-invasive, non-ionizing image collection technique. Moreover, MRI provides excellent contrast in images, especially between different tissues; since its contrast mechanism is based on tissue property. Hence, MRI is an exceptional tool for anatomical imaging and diagnostics. MRI is particularly effective, in comparison with other imaging techniques, in brain imaging. Its ability to distinguish between different tissue types of brain via contrast differential makes it a popular tool for imaging brain and nerve tissue.

Unfortunately, MRI image acquisition is a very slow process, compared to other imaging modalities. An MRI machine take one sample at a time from the frequency plane of the slice of body being examined. In order to reconstruct an image from the samples, the sampling frequency should be at the Nyquist rate; which means a long time to acquire all samples. In addition, MRI images are very sensitive to discrepancies, and system imperfections. In particular brain MRI images are extremely sensitive to magnetic field inhomogeneity (a know system imperfection in MRI).

Compressed Sensing has been proven [40, 39] to improve the speed of image acquisition of MRI by taking fewer samples and being able to provide good quality images. Further, CS has the added benefit of interference cancellation, especially when it is used in conjunction with iterative optimization reconstruction mechanisms. Hence, it is a promising field that has potential to improve imaging speed and quality; shortening waiting times that are required to perform an MRI imaging, especially in countries with universal health care system like Canada.

Many pathological disorders, such as brain tumors or Multiple Sclerosis (MS) lesions require good visualization after post processing is performed. To do this, most post processing algorithms employ a method of border/edge detection to start a process of region detection (via other processing techniques such as segmentation). This is coupled with the desire to develop automatic feature recognition and extraction tools for MRI (particularly brain for reasons aforementioned); that make the existence of an acceptable edge detection technique all the more interesting. Hence, MRI brain edge detection methods is a topic of extensive research that has much room left for improvements.

The development of wavelets in recent times has been growing, with constant findings on their abilities as a processing tool in different areas of science and engineering. One of such applications is in the field of edge detection. Much research has been performed on the topic of edge detection using wavelets, and many aimed at brain MRI edge detection and noise reduction. However, most of methods developed require the image to be reconstructed first (using Nyquist sampling), and de-noised using smoothing filters; which does not address the speed limitations of MRI imaging. Hence, there is still much work to be done to develop and/or improve techniques of brain MRI edge detection.

1.2 Literature Review

Although the first signs of what CS is capable of were observed in the research community in the 1970s and 1980s; it was not until much later, in 2006, that a comprehensive mathematical explanation of the phenomenon was offered by [13], [25], and [14] on different occasions. In addition to the explanation of the reconstruction of an under-sample signal using convex optimization norm minimization techniques; some of these works also provided evidence of the effect of different sampling techniques (random techniques) on the quality of the results.

Since 2006, the body of scientific literature available on Compressed Sensing and its application has grown steadily and rapidly in only a few years. One area of interest for an application of CS has become in Magnetic Resonance Imaging (MRI). The prospect of CS as an image reconstruction tool for MRI is a strong one as it has the potential to decrease the imaging time required; which in turn reduce costs, as well as the fact that MRI image encoding is a perfect fit for Compressed Sensing.

By the year 2008 the application of CS in MRI was introduced in [40]. The authors use a technique of Compressed Sensing for MRI image acquisition and examine its outcome in a variety of conditions. One area of MRI imaging that has harbored much attention is the dynamic MRI imaging. One technique specifically designed for dynamic MRI (heart) is called *k-t SPARSE* [41]; which takes advantage of the both time and spatial sparsity of images that constitute a dynamic image. Later on, improved dynamic CS imaging techniques were also provided in [33]. Another important aspect of CS MRI introduced in [40] was that of brain imaging; by helping reduce the imaging time of the most clinically popular use of MRI.

After its introduction as Compressed Sensing MRI in 2008, the MRI community has constantly attempted to expand the process to other areas of MRI or improve upon the original idea by studying smaller details of the CS process that might affect the final image quality. An instance of such attempts can be found in [64]; which studies (and improves) the effect of two dimensional Cartesian sampling over the traditional one dimensional method. In addition, the application of CS has been extended to other types of MRI imaging such as parallel imaging in [32]. Some other papers provide a variety of studies on different optimization techniques that can be used in conjunction with CS in order to reconstruct the MRI image; such as the interior point method [36].

Most of these works are directed towards the concept of CS MRI image reconstruction itself. In this thesis, the goal is to create an application of edge detection for CS MRI; which utilizes the advantages of CS in order to reduce noise and is proposed in Chapter 5. In order to evaluate the technique in the work presented here, it is significant to compare it with existing techniques; which can be utilized as an automatic process.

One of these techniques, called the Canny edge detection[16] is considered to be the best available (con-

ventional) edge detection method. A wavelet method produced by Mallat[58] is based on the wavelet transform and models smoothing functions using wavelet coefficients, and is better at noise suppression compared to some other techniques available. A brief study of different edge detection methods and their respective advantages and disadvantages are presented in [57].

A number of edge detection techniques have been created specifically for brain MRI images[1],[27], that utilize some format of already existing methods or attempt to improve upon it. This is what we strive to do in this work with the inclusion of Compressed Sensing. As such the framework presented here is not based on any previous or existing works; but rather utilizes different theories and methods such as the Compressed Sensing MRI image reconstruction, and wavelet transforms.

Hence, it is deemed necessary to provide background information on such aspects, while reviewing some of the work that has been done in each specific area. This is what is attempted in this thesis; Chapters 2 through 4 explain, in details, how these principles apply to what is being done in this work. Any work that has been done in one particular area is referenced throughout the text. As the explanations provided are summaries of comprehensive applications, the reader is encouraged to consult the references provided for further readings and to obtain a complete picture of the state of the art research that is performed (especially in the case of Compressed Sensing).

1.3 Objective & Contributions

In this thesis the objective is to provide a brain MRI edge detection process that provides reduces noise sensitivity while addressing time consumption problem of acquiring MRI images. This work attempts to reach this goal by providing the following contributions:

- Designing a Compressed Sensing based method of edge detection that incorporates wavelets as an edge detector tool. A wavelet transform also acts as an sparsifying transform, becoming an integral part of CS reconstruction.
- Providing an outline for optimal regularization parameter selection that optimizes noise suppression while providing maximum edge detection, accurately. Creating a framework for optimal wavelet choice criteria.
- Implementation of a CS edge detection process. The process reduces the sampling time required to acquire an image, effectively cutting down imaging time. Further, the edge detection is performed using this sample without having to reconstruct the original image; while providing the option of reconstructing the original image at the same time by changing two variables.

1.4 Organization

This thesis presents a method of edge detection for brain MRI images using Compressed Sensing reconstruction algorithm for under-sampled signals. The material presented will be organized as follows:

Chapter 2 (Compressed Sensing MRI)

This chapter provides essential background information and introduction to principles of Magnetic Resonance Imaging and Compressed Sensing. It moves one by introducing three significant components a signal processing problem must possess in order to qualify for CS reconstruction. Finally, this chapter indicates the reasons that make an MRI image construction befitting as a CS problem, and finalizes by introducing a solution to solve the under-sampled reconstruction.

Chapter 3 (From Fourier to Wavelets)

This chapter outlines the evolution of frequency domain harmonic analysis from Fourier Transform to Short Time Fourier Transform and Wavelet Transform. It provides a short summary of key concepts for each analysis with a focus on wavelets; specifically Multi-Resolution Analysis (MRA) and Filter Banks. The chapter extends the topic of Discrete Wavelet Transform into two dimensional DWT and orthogonal wavelets bases, and their application to image processing. This is a crucial chapter as wavelets play a key role to both edge detection and CS reconstruction in this thesis.

Chapter 4 (Image Edge Detection)

Chapter 4 focuses on the subject of image edge detection and its role in image processing and machine vision. It provides an introduction to basic elements required in understanding the importance of edge detection in image processing, such as mathematically defining edges in an image. Further, this part of the thesis examines some of the more widely used classical edge detection methods such as the Sobel and Canny edge detection techniques. Then it moves onto wavelet edge detection methods that utilize filtering properties of wavelets to obtain edges. This chapter is finalized by outlining the importance of brain MRI edge detection in the field of medical imaging.

Chapter 5 (Methodology)

This chapter summarizes key concepts of the *proposed method* of brain MRI image edge detection. It provides details as to how the two concepts of Compressed Sensing and wavelet transform edge detection can be combined together to create an interference canceling technique for edge detection of brain images (especially noisy images). It approaches the proposed design in a step by step advancement while explaining the significance of every step in reaching the goal. In the end this chapter provides some evaluation techniques that can be used to assess the proposed method.

Chapter 6 (Results)

Finally, this chapter provides the results obtained by performing the CS based edge detection technique on an image. The results are divided into different sections; each of which examine the effect of a different variant in the final image acquired. These variants are regularization parameters, wavelet families used, and the non-uniform sampling trajectories used. In the end, this chapter examines the performance of the proposed edge detection technique in presence of three different types of noise.

Chapter 2

Compressed Sensing & MRI

2.1 Overview

In recent years Compressed Sampling/Sensing (CS) has attracted extensive attention from the scientific community, more specifically from the signal processing community; with a wide range of applications from Information Theory to Medical Image Processing. This chapter provides a look at some of the mathematical foundation on which Compressed Sensing relies as well as some insight on Magnetic Resonance Imaging modality. As the discussion here is limited to examination of concepts most related to this work, the reader is advised to consult references provided for a more comprehensive understanding of the underlying concepts.

2.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging is a non-ionizing imaging modality that revolves around the physics of Nuclear Magnetic Resonance (NMR). After about 40 years [51] the idea of NMR came to life with the introduction of the first MRI system in the 1970s. Magnetic Resonance Imaging provides high levels of detail with high resolution and excellent contrast, while it lacks the imaging speed of some other imaging modalities. Subsequent sections will introduce some of key concepts and physics of MRI [51, 38, 10, 48].

2.2.1 Physics of Nuclear Magnetization

An NMR signal is acquired by utilizing certain properties of the nuclei of atoms that are found in abundance in the body; most notably the Hydrogen atoms. The nucleus of a Hydrogen atom possesses an angular momentum $\vec{\Phi}$ resulting from it spinning. The angular momentum and the charge on the atom creates a *microscopic magnetic field* defined with a vector $\vec{\mu}$ as:

$$\vec{\mu} = \gamma \vec{\Phi}, \quad (2.1)$$

where γ ; the *gyro-magnetic ratio*, has units of angular frequency per strength of the magnetic field (Tesla), and is a constant related to the sample tissue property.

Together, a collection of a number of nuclei is called a *spin system*. Spin systems can be magnetized *macroscopically* using an external magnetic field $\vec{B}$; the magnetization of a spin system is called bulk magnetization $\vec{M}$. Intuitively, one could interpret bulk magnetization as the sum of every single $\vec{\mu}$ of every single nucleus in (2.1), given by:

$$\vec{M} = \sum_{n=1}^{N_s} \vec{\mu}_n, \quad (2.2)$$

where N_s is the spin number for each individual nucleus¹. From now on bulk magnetization will simply be referred to as magnetization. After the external magnetic field reaches equilibrium, it creates a magnetization vector $\vec{M}_0$ in the sample which has the same direction² as the external magnetic field and a magnitude given by:

$$M_0 = \left(\frac{B_0 \gamma^2 \hbar^2}{4kT} \right) P_D, \quad (2.3)$$

where T is the temperature, k is called the Boltzmann's constant, and P_D is the sample proton density. One can observe from (2.3) that the magnitude of the magnetization vector at equilibrium, $\vec{M}_0$ is dependent on the magnitude of the external magnetic field. It is important to distinguish between the magnetization vector at equilibrium $\vec{M}_0$ and the time varying magnetization vector $\vec{M}(t)$; with the former dependent on spatial position $\vec{r}$ while the latter is dependent on both $\vec{r}$ and time t . This is a significant realization as the external magnetic field is generally a time varying component, $\vec{B}(t)$ in the MRI system in contrast to the external *static* magnetic field $\vec{B}_0$ where:

$$\vec{B}_0 = B_0 \hat{r}. \quad (2.4)$$

In (2.4), B_0 is the magnitude of the static magnetic field pointing in the direction $\hat{r}$. Here-after, it will be assumed that the external magnetic field is direct in the z direction of the laboratory frame.

The presence of a time varying magnetic field (in addition to $\vec{B}_0$) causes the magnetization vector to move away from its original orientation (direction), and *precess*³ about $\vec{B}_0$ at a frequency that is proportional to the magnitude of the static magnetic field (Fig 2.1), given by:

$$\omega_0 = \gamma B_0, \quad (2.5)$$

where ω_0 is called the *Larmor Frequency*, and has units of radians per second. The precession motion causes an angular momentum in relation to the magnetization vector, called $\vec{J}$. The relationship between the external magnetic field and the angular momentum of the magnetization vector can be described as

$$\frac{d\vec{J}(t)}{dt} = \vec{M}(t) \times \vec{B}(t), \quad (2.6)$$

¹Different atoms have different pre-measured spin numbers

²It is of importance for the reader to recognize that the spin direction of each single nucleus is *not* the same as $\vec{B}$, rather the net direction of the collective spins; the bulk, is the same.

³Precession can best be described as the motion of a dreidel where the top point is fixed on an imaginary axis while the point touching the surface is not fixed and changing orientation in a circular motion

Figure 2.1: The precession motion of magnetization vector about an external static magnetic field.

where

$$\vec{M} = \gamma \vec{J}. \quad (2.7)$$

Combining equations (2.6) and (2.7), one can derive:

$$\frac{d\vec{M}(t)}{dt} = \gamma \vec{M}(t) \times \vec{B}(t). \quad (2.8)$$

One can calculate[38] components of the magnetization vector for $\vec{M}(t) = (M_x(t), M_y(t), M_z(t))$ from (2.8) for any given point in time, t .

Figure 2.2: The magnetization vector and its components under influence of $\vec{B}(t)$.

$M_z(t)$ is called the *longitudinal magnetization* component, while together $M_x(t)$ and $M_y(t)$ form the *transverse magnetization* component as:

$$M_{xy}(t) = M_x(t) + jM_y(t), \quad (2.9)$$

where the phase of $M_{xy}(t)$ is:

$$\varphi = \arctan \frac{M_y}{M_x}. \quad (2.10)$$

The precession motion of the magnetization vector and its dependent on an external, *controllable* magnetic field is what makes the existence of NMR signals and ultimately MRI possible. Detection and localization of NMR signals will be explored in more detail in the next section.

2.2.2 NMR signal detection

The motion of the magnetization vector, more specifically, the transverse magnetization component creates a radio frequency (RF) signal in the sample, decided by the magnitude of the external magnetic field. This signal is detectable via an external coil system according to *Faraday's law of induction*. As the basis of electromagnetism, Faraday's law states that when a circuit (or a series of coils) cuts through a time varying magnetic field, the magnetic flux through the circuit induces a current in the coil that produces an *electromotive force* (EMF); which in its own right creates a time varying voltage.

In the case of an NMR signal, the time varying magnetic field is $\vec{M}(\vec{r}, t)$ at point $\vec{r}$ in space. There is an additional magnetic field created by the external magnet, $\vec{B}_r(\vec{r})$ at the same point in space. By putting together these magnetic fields, the flux through the detection coil can be written as:

$$\Phi_B(t) = \int_{sample} \vec{M}(\vec{r}, t) \cdot \vec{B}_r(\vec{r}) d\vec{r}, \quad (2.11)$$

where $\cdot$ is the inner product operator.

The potential difference in the circuit is the rate of change of magnetic flux, and is given by:

$$\hat{v}(t) = -\frac{d}{dt} \int_{sample} \vec{M}(\vec{r}, t) \cdot \vec{B}_r(\vec{r}) d\vec{r}. \quad (2.12)$$

In order to get the NMR signal a number of assumptions need to be made in order to simplify the relationship in (2.12). To begin with, the position dependency of the magnetic fields is cumbersome. This can be dealt with by assuming that the magnetic field is uniform throughout the sample (i.e. the sample is homogeneous); which means that $\vec{M}(\vec{r}, t) = \vec{M}(t)$. In addition, the rate of change of the magnetization vector in the z direction is very slow and can be neglected, making the NMR signal mostly dependent on the transverse magnetization. Using these assumptions, one can solve the expression in (2.12) to be:

$$\hat{v}(t) = -V_s \left[\frac{dM_x(t)}{dt} B_r + \frac{dM_y(t)}{dt} B_r \right], \quad (2.13)$$

where V_s is the sample volume, and $\hat{v}$ is the NRM signal. Without going into much detail, it can be shown [51] that the NMR signal strength is proportional to B_0^2 . The signal can also be maximized by pushing the transverse

magnetization component further away from the z axis at an angle α , called the *tip angle*. This can be done by introducing an external RF excitation using another, smaller, external magnetic field; $\vec{B}_1(t)$.

The RF excitation is in the form of a pulse, usually a rectangular pulse, with duration τ_p . The RF field produced by the pulse pushes the transverse magnetization component to the point where it lies on the $x - y$ plane and starts precessing after the RF pulse is turned off. As time passes by the RF field starts to decay forming a decaying signal called *Free Induction Decay* (FID), unlike what is predicted in (2.8). The envelope of this signal is B_1^e . The tip angle is proportional to the amplitude of the signal envelope over the course of the duration of the pulse, given by:

$$\alpha = \gamma \int_0^{\tau_p} B_1^e(t) dt, \quad (2.14)$$

which can be solved for a rectangular pulse to be:

$$\alpha = \gamma B_1 \tau_p. \quad (2.15)$$

The tip angle is one variable controlled by the user via B_1 to acquire a desired NMR signal.

After the magnetic field $\vec{B}_1(t)$ is turned off, the magnetization vector $\vec{M}(t)$ precesses due to the existence of $\vec{B}_0(t)$ and *relaxes* until there is no more NMR signal to be detected. In order to detect another NMR signal another RF excitation is needed creating another precession and relaxation cycle. This cycle is repeated as many times as desired to receive adequate NMR signals. Since relaxation is a characteristic of the magnetization vector, it also has two components: *transverse* and *longitudinal* relaxation.

Transverse relaxation occurs first, when the transverse magnetization vector $M_{xy}(t)$ precesses back from the $x - y$ plane to the z -axis; which causes the received signal to decay. Transverse relaxation can be modeled by a negative exponential as $(e^{-\frac{t}{T_2}})$. Where T_2 is the time constant relative to tissue characteristics.

Longitudinal relaxation takes place when the longitudinal magnetization component $M_z(t)$ relaxes back to its equilibrium magnitude M_0 . Although the value of this component is increasing, since it contributes to a signal loss it is treated as a negative exponential model as $(e^{-\frac{t}{T_1}})$. Here, T_1 is also a time constant dependent on tissue property. Together, the two relaxation time constants T_1 and T_2 are responsible for creating MRI images with different contrasts.

It can be observed that the relaxation of the magnetization vector plays a significant role in the magnitude of the detected NMR signal and needs to be accounted for. The relaxation behaviour of magnetization can be described as:

$$R[\vec{M}(t) - \vec{M}_0], \quad (2.16)$$

where $\vec{M}_0$ only has one component in the z direction, and R is the relaxation matrix:

$$R = \begin{pmatrix} \frac{1}{T_2} & 0 & 0 \\ 0 & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{pmatrix}. \quad (2.17)$$

Combining (2.16) with the relationship in (2.8) will result in a more complete picture of what the magnetiza-

tion vector looks like.

$$\frac{d\vec{M}(t)}{dt} = \gamma\vec{M}(t) \times \vec{B}(t) - R[\vec{M}(t) - \vec{M}_0], \quad (2.18)$$

where it is understood that the relaxation matrix reduces the overall signal detected; hence the "−" sign, and

$$\vec{B}(t) = \vec{B}_0 + \vec{B}_1(t). \quad (2.19)$$

The equation in (2.18) can be expanded and expressed in terms of components of the magnetization vector; these equations are called the *Bloch equations*. Bloch equations are the governing set of relationships that define the behaviour of the magnetization vectors and ultimately explain how a certain NMR signal can be achieved. All that remains now is to devise a method of signal localization that can assist in recognizing which sample, or voxel¹, is the NMR signal coming from.

2.2.3 Spatial Localization & k-space

Spatial Localization

In order to encode signal information spatially an MRI system uses the so called *gradient fields*. These are smaller magnetic fields produced by gradient coils, in addition to the strong static magnetic field² Modern day MRI systems produce gradient fields in three directions, and are denoted by G_x , G_y , and G_z consistent with their direction. Gradient fields enable slice selective imaging that makes MRI a tomographic imaging technique.

The addition of gradient fields changes (2.4) to:

$$\vec{B} = (B_0 + \vec{G} \cdot \vec{r})\hat{z}, \quad (2.20)$$

where $\vec{G} = (G_x, G_y, G_z)$, $\vec{r} = (x, y, z)$ is the vector pointing to spatial position r in space, and B_0 is in the direction of z -axis; that is:

$$\vec{B} = (B_0 + G_x x + G_y y + G_z z)\hat{z}. \quad (2.21)$$

Incorporating (2.21) in (2.5) results in a new relationship for Larmor frequency which now accounts for the presence of gradient fields:

$$\omega(\vec{r}) = \gamma(B_0 + \vec{G} \cdot \vec{r}). \quad (2.22)$$

One can observe from (2.21) that by adding or changing the gradient field at a spatial position in space ($\vec{r}$), the frequency of the received signal from that point is changed; essentially encoding the signal spatially. The concept of spatially encoding samples using gradients as a means of changing frequency is referred to as *frequency encoding* (Fig 2.3). It is important to observe that the application of the frequency encoding gradient coincides with the acquisition period T_{aq} . This is because (in most cases) one is interested in acquiring many samples in one row of an image.

In current imaging techniques, one gradient (normally G_z) is used for slice selection¹ while the other two are

¹A 3-Dimensional pixel

²The gradient fields can work to add and subtract from the static magnetic field.

¹Here z direction is assumed to provide an axial image with the z -axis pointing from the head to toe of a subject.

utilized to encode the position of a spin system in that slice. The received signal from the (x, y) coordinate is a linear combination of the NMR signal (M_{xy}) emitted from each single spin system located at that exact coordinate; this signal is denoted as $m(x, y)$. The signal is modulated at a certain frequency indicated by (2.5); which is the baseband frequency of the received signal. By taking into account the effect of each gradient on the Larmor frequency in (2.22), the signal received from spatial position (x, y) is:

$$s(t) = e^{-i2\pi f_0 t} \int_{-\infty}^{\infty} m(x, y) e^{-i2\pi \frac{\gamma}{2\pi} (G_x x + G_y y) t} dx dy, \quad (2.23)$$

where f_0 is the Larmor frequency in (2.5) stated in cyclic terms rather than radial units, and the time limits of the integrals are dictated by the duration of the pulses. Equation (2.23) is the 2D Fourier transform of signal $m(x, y)$ with Fourier frequencies defined as:

$$k_x = \frac{\gamma}{2\pi} G_x t, \quad (2.24)$$

$$k_y = \frac{\gamma}{2\pi} G_y t, \quad (2.25)$$

where G_x and G_y are amplitudes of the gradient pulses produced, with the vector pointing to point (k_x, k_y) in the frequency domain has the phase:

$$\theta = \arctan \frac{k_y}{k_x}. \quad (2.26)$$

In frequency domain u is in the Cartesian x direction while v is in the y direction. Using frequency encoding only allows for one to move back and forth on the u -axis, called frequency encoding axis. In order to be able to move up and down on the v -axis, and encode spatial data with respect to v one can use a method called *phase encoding* (Fig 2.4). Phase encoding involves the use of one gradient (conventionally G_y) to change the phase of spin systems located in a slice, with respect to each other, in one dimension of the specified slice. One can observe from (2.23) that the phase *accumulated* during T_{ph} is given by:

$$\phi_y = -\gamma G_y y T_{ph}, \quad (2.27)$$

This is achieved by producing a short duration pulse right before the frequency encoding gradient is activated and a sample is acquired. According to (2.27), the location of interest on v is controlled by the phase encoding gradient pulse duration, T_{ph} .

Together, frequency and phase encoding mechanisms provide better flexibility in image acquisition and can speed up the scan process since they provide two degrees of freedom instead of one. Furthermore, the frequency encoding gradient is controlled by the amplitude of the pulse while the phase encoding gradient is dependent on the pulse duration¹.

¹This is the most widely used technique for controlling the gradients; however, for different imaging trajectories there other approaches are implemented.

Figure 2.3: (a) A pulse sequence for sample frequency encoding scanning, and (b) its corresponding Fourier trajectory on the frequency plane. G_{ss} is slice selection gradient (G_z in this case) and T_{aq} is the acquisition time for the Analog to Digital Converter (ADC).

Figure 2.4: (a) A pulse sequence for sample phase encoding scanning, and (b) its corresponding Fourier trajectory on the frequency plane. T_{ph} is the duration of the phase encoding pulse.

K-space

The spatial encoding gradients introduced provide a set of tools for navigating the frequency domain of the slice being imaged. These gradients are manipulated using different RF pulses in pre-determined sequences called *pulse sequences*. One pulse, identified by RF, controls the tip angle as described in (2.15); another pulse is used for slice selection, denoted as G_{ss} . Two more pulses representing G_x (frequency encoding gradient) and G_y (phase encoding gradient) control the gradient fields depending on type of encoding used (frequency and/or phase encoding).

Figure 2.5: (a) A pulse sequence for a series of samples. Each pair of gradients moves to its corresponding Fourier trajectory on (b) the k-space. If all possible combinations of all 5 pulses are utilized, the resulting sampling trajectory would be a Cartesian grid of acquisitions.

Every time G_x and G_y pulses are activated, the Fourier trajectory moves to a single point in the spatial frequency domain; consequently that point is acquired by the MRI system. The process of acquiring of one sample is called an *acquisition*. Subsequently, another set of gradient pulses are created that force the trajectory to move to another point for sampling. This acquisition procedure is repeated with a combination of different pulse sequences until *enough* samples are acquired. When put together, these samples create a visual representation of a frequency domain image called the *k-space*. In simple words an MRI system samples (Fig 2.5) the k-space of an image; the image itself can then be reconstructed by applying the inverse Fourier transform.

Since sampling trajectories in MRI are handled by pulse sequences, the user has immense flexibility when it comes to designing acquisition patterns. Many such trajectories are already existent; Cartesian, radial, and spiral to name a few; with each possessing its own unique set of advantages and disadvantages. Freedom in designing different sampling patterns for k-space, and ability to change the sample density from one point in k-space to another are crucial in applying CS reconstruction in MR image reconstruction.

Figure 2.6: MRI Image formation by taking the two dimensional Inverse Fourier Transform (IFT) of the k-space image.

2.3 Compressed Sensing

Traditional signal processing paradigm relies on the Shannon-Nyquist Sampling criteria; which states that in order for a Continuous Time (CT) to be reconstructible signal needs to be sampled at a rate twice its bandwidth:

$$f_s = 2B, \quad (2.28)$$

where f_s is the sampling frequency and B is the bandwidth. In the case of an image or other Discrete Time (DT) signals this rate is changed to meet the temporal or spatial resolution required. The theorem also predicts that no information is lost in the reconstruction process while the original signal can be obtained using linear methods in most cases.

Although the Shannon-Nyquist criteria performs well as the backbone of signal processing paradigm, it creates some problems from a practical standpoint. With the increase in the number of signals that need to be processed everyday and the amount of information they contain, traditional sampling methods pose two main problems: high clock rates, and large storage requirements.

In order for a CT signal to be sampled at a sampling frequency f_s , a signal processing system requires a processing unit that is capable of producing clock rates equal or larger that of the sampling frequency. Additionally, in many applications, sampling rates are much higher than that required by the Shannon-Nyquist criteria, and can approach rates as high as eight times the bandwidth or more. This, coupled with the fact that there is a large pool of signals that need to be sampled means higher hardware complexity. Furthermore, many signal processing applications, such as medical imaging, involve some sort of data storage mechanism. The storage capacity requirement is proportional to the number of samples acquired, and increases as the number of samples increase. Expanding storage can be realized at the expense of higher hardware cost. Both of these issues call for a more efficient method of sampling that reduces cost and complexity.

Compressive Sampling [15] is a *non-adaptive* signal sampling framework that collects samples from *sparse* signals at rates much lower than that indicated in (2.28); which can then be used in conjunction with a non-linear optimization procedure to reconstruct the original signal.

In order to perform a successful CS procedure, from sampling stage to reconstruction, there are three condi-

tions that need to be met and resolved; which are examined in depth in the following sections.

2.3.1 Sparsity

Most practical signals and images are *sparse* in any one domain; including but not limited to time, frequency, pixel, or spatial frequency domain. Simply put, a sparse signal is one that contains elements that do not carry any significant information. Let us assume that a discrete, finite length signal is represented as $F[n]$, where for simplicity $F[n]$ is a one dimensional vector of size $N \times 1$ in space $\mathbb{R}^N$ and $n = 1, 2, \dots, N$. The same assumptions can be made for a two dimensional signal or image. Since $\mathbb{R}^N$ is a linear vector space, signal $F[n]$ can be represented as a linear combination of the orthonormal basis of the vector space. The linear combination is [4]:

$$F[n] = s_1\lambda_1 + s_2\lambda_2 + \dots + s_n\lambda_n = \sum_{n=1}^N s_n\lambda_n, \quad (2.29)$$

where s_n is a $N \times 1$ weighting coefficient matrix, and $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_N]$ is the N dimensional basis matrix that spans the vector space.

If only K weighting coefficients with $K \subset s_n$, are required for a successful signal representation, then the signal is said to be K – *sparse* [13]; that is to say the other $(N - K)$ coefficients are insignificant with only K non-zero elements in the matrix. In the case where $K \ll N$, the signal $F[n]$ can be *compressed*, by collecting only the non-zero terms and discarding the insignificant coefficients. As an example one can assume that the signal $F[n]$ is in Fourier domain with the Discrete Fourier Transform of:

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-i2\pi\Omega kn/N}, \quad (2.30)$$

where Ω is the set of all frequencies available in the signal, $k = 0, 1, \dots, N - 1$. Selecting only ω_n of these frequencies results in compressed sampling.

Indeed this is the foundation of many signal and image compression techniques such as JPEG and MPEG. These compression techniques often sample signals at the Nyquist rate, then sift through and select K relevant components while the $(N - K)$ remaining samples are wasted. This type of signal compression technique (used in *transform coding*) raises a fundamental question about the choice of sampling rate: why should so much effort be put toward acquiring samples when most of them will be discarded? It would seem logical to collect only those samples that are significant in reconstruction of the signal; during the sample collection stage. Hence, any sparse signal has the potential to undergo compressed sampling¹.

2.3.2 Incoherence and Random Sampling

Sparsity is the first step in any CS process; though, it does not guarantee it to be a success. The problem now becomes the manner in which the significant components of the signal are acquired. Since in most practical cases there is no information available as to where each valuable element is located in a vector space, one must revert

¹Compressed Sensing and Compressed Sampling terms are used interchangeably and refer to the same concept.

to other approaches when *sensing* a signal. A matrix Φ used to collect samples from a sparse signal is called a sensing or a measurement matrix. Designing a proper and stable sensing matrix is of utter importance when solving a CS problem.

Compressed Sensing is a non-adaptive signal sampling mechanism. This means that the sample collection process should be signal independent, as opposed to an adaptive method where sampling signals vary from one signal to another. This suggests that the sensing matrix should be set up to collect samples randomly without any knowledge of the signal. In fact, a good sensing matrix should be select samples randomly and incoherently. It should be understood that randomness and incoherence are two different attributes and are not to used interchangeably as they generally don't refer to the same concept. While randomness indicates that samples must be chosen without any deterministic preconditions, and might provide incoherence in some cases, incoherent sampling does not necessarily follow the same guideline.

Coherence, by definition, is a measure of connections between elements of a set; which is calculated in terms of correlation of the set. Incoherence sampling, then, would attempt to try an minimize the correlation between the elements of a set. In a sensing problem this translates into collecting samples without any deterministic or initial conditions. Although random sampling does not have any initial conditions, there are other criteria that must be taken into consideration when designing an incoherent sensing matrix.

As mentioned, random sampling is independent of any initial conditions and selects samples at random. However, a random sensing matrix might not always be incoherent; this is the main difference between random sampling and incoherent sampling. A reader familiar with the notion of orthogonality of a matrix can refer to incoherent matrix as a *partially* orthogonal matrix.

An orthogonal matrix O must satisfy the following criteria:

$$O^{-1} = O^T, \quad (2.31)$$

where O^T is the transpose of O , and

$$O^T O = O O^T = I, \quad (2.32)$$

where I is the identity matrix. Traditional signal sensing matrices are orthogonal; assuring non-singularity, and hence, a solution (signal recovery).

The concept of orthogonality can be extended to indicate the orthogonality between two different matrices. This is particularly significant because the sensing matrix and signal basis, or the sparsity basis (to promote sparsity). To clarify, one can assume that a signal, $f(t)$ is being sampled in a certain sparse basis Ψ , by sensing matrix Φ ; where $(\Psi, \Phi) \in \mathbb{R}_N$, and also $\Psi \Psi^T = I$, $\Phi \Phi^T = I$, indicating they are both orthogonal. The measure of orthogonality or coherence between these two matrices is called *mutual coherence*, μ , and is[12]

$$\mu(\Phi, \Psi) = \max_{k,j} |\langle \phi_k, \psi_j \rangle|. \quad (2.33)$$

Equation (2.33) can be normalized to be

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{k,j} |\langle \phi_k, \psi_j \rangle|, \quad (2.34)$$

where n is the index defining the largest element in the union of the two matrices, and the index for the smallest entry is 1. Hence, one can see that the result of (2.34) is a value ranging from 1 to $\sqrt{n}$, with 1 indicating maximum incoherence (or least correlation between elements of the two sets) and $\sqrt{n}$ being the lowest incoherence (or maximum correlation between the elements of the two sets). Therefore, the goal in designing a *good* sampling matrix is to get μ as close as possible to 1.^{1, 2}

A final note on the sampling process should be provided on the number of samples acquired. As described before, a signal f is said to be K -sparse if there are only K significant components out of the possible N with $f \in \mathbb{R}^N$. Then it would seem logical to collect only those K components; however, there is a small issue to be addressed. Compressed sensing employs random sample selection, and collecting only K samples may not result in all the significant components being gathered. In order for a good reconstruction to take place one needs to collect more than just K samples; the number of samples required, M , for a good reconstruction could be calculated according to results in [13],[25]; where $K < M \ll N$. This number could vary in practice; results provided by Lustig et al. [39], [11], and [61] demonstrate that $M \equiv (2 \sim 5)K$.

In CS samples are collected by a simple linear projections onto the signal (in the signal basis), and are used in a *non-linear* recovery method to reconstruct the original signal.

2.3.3 Recovery of Under-Sampled Signal

Up until this point all the focus has been concentrated on identifying the type of signals that are suitable for compressive sampling, and how best to (under-)sample these signals. However, probably the most complex process in developing an effective CS solution is the reconstruction of the signal from its sparse coefficients. In order to develop a solution for the reconstruction algorithm, a more simplistic approach is utilized.

Let us assume that the signal measurement problem can simply be stated as

$$y = Ax,^1 \quad (2.35)$$

where x is a $N \times 1$ signal in a row vector format, A is a $M \times N$ measurement matrix, and y is the $M \times 1$ matrix of samples. Following (2.35), one can recover the original signal by performing an inverse problem such that

$$x = Ay', \quad (2.36)$$

where y' is the solution to equation (2.36). Now let's consider the case for a CS recovery with the forward problem stated as

$$y = \Phi x, \quad (2.37)$$

where x is the signal being processed, and Φ is the sensing matrix. Assuming that the signal has a specifying vector Ψ in the proper basis with $K < M \ll N$, and only M measurements made, the relationship in (2.37)

¹Since partial sampling is applied, it is practically impossible to get $\mu = 1$; the goal should be minimizing coherence, while collecting as few samples as possible.

²It should be noted here that excellent reconstruction results [13],[25] have come from purely random sample collections; however, for practical purposes in the case of CS imaging and certainly in MR imaging, incoherent sampling is the method of choice.

³For simplicity, from here onward, vector notations for vectors are removed; since it is generally understood that in dealing with signals all parameters define either vectors or matrices.

becomes:

$$y_s = \Phi \Psi s = \Theta s, \quad (2.38)$$

with s representing the set of measured coefficients with size $M \times 1$ where $K < M \ll N$ for a K -sparse signal x . In order to recover the original $N \times 1$ signal from y_s measurements, one could provide an inverse problem following (2.36)

$$x = \Theta s'. \quad (2.39)$$

Given that x is $N \times 1$ while s' is only $M \times 1$ with $M \ll N$, one can observe that the inverse problem for a CS recover is an *under-determined* ill posed problem; for which there is an abundant number of solutions. Therefore, to solve the inverse CS problem optimization techniques need to be employed that help *approximate* the best possible solution. More specifically, *convex optimization* methods are used to recover the original signal from an incomplete set of samples.

Before entering any further discussions on the topic of CS recovery, it is imperative to be familiarized with some of the basic concepts of convex optimization; as well as concepts such as norms and penalty functions. As optimization is a broad field; the following discussion is limited to methods related to the topic of this work; for a more in depth study of approaches presented henceforth, as well as other topics in the field of optimization, the reader is encouraged to consult many excellent texts available[8],[54].

Convex Optimization

Optimization, in general, is the process of minimizing or maximizing a function of a set of functions according to some preset constraints. Often times one might be interested to find the best possible solution for a variable or a set of variables that define a function. Mathematically, a general, constrained optimization problem can be formulated[8] in the following format.

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{such that} && f_i(x) \leq b_i, \end{aligned} \quad (2.40)$$

where $i = 1, 2, \dots, m$, vector $x = (x_1, x_2, \dots, x_n)$ is a set of optimization variables, while the function $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, and functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. The optimization problem in (2.40) is a *multi variable*, f_0 is the *objective* function, and f_i are called the *constraint* functions of inequalities. The solution to the optimization problem stated above is x^* ; which is called the *optimal* solution. Therefore, x^* is the vector that results in the smallest possible value for the objective function which also satisfies the constraints.

If the constraint and objective functions are linear; that is to say they satisfy

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y), \quad (2.41)$$

for $x, y \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$; the optimization problem is called a *linear program*. Linear optimization programming is a subclass of a more general optimization problem that requires less constraint and satisfies

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \quad (2.42)$$

with the additional conditions that $\alpha + \beta = 1$ and $\alpha, \beta \geq 0$. Convex optimization; hence, is less a restrictive parallel of an optimization problem.

In order to solve a convex optimization problem (or any optimization problem), one must first be able to identify the problem, its objective functions, and/or its set of constraint functions, if any¹. Convex optimization problems can be categorized based on the properties of their functions. For instance, certain forms of objective or constraint functions could categorize one group of optimization problems; the number of variables or structure of the set of variables (or objective function) are other characteristics of an optimization problem that could help in determining the class of the problem. An example of a special structure is *sparsity*; which is the topic of interest in the field of compressed sensing.

After formulating the problem, the task is to decide which method is best suited for solving the problem. There are no analytical solutions² to a convex optimization problem; however, there are many different *algorithms* previously developed that suite certain classes of convex optimization problems.

The goal of any solution for a given optimization problem is to solve the problem formulation with high accuracy and reliability, time after time for the same problem, with the hope that after a solution is found the original problem will be solved (*i.e.* finding a solution in case of an overdetermined system).

Convex optimization has many applications in different areas of science and engineering (specially in signal processing); many fitting and approximation algorithms rely heavily on norm approximations and minimizations to solve for several reasons that will become clear.

Norms and Norm Approximations

The norm of a vector in any Euclidean space $\mathbb{R}^n$, is the Euclidean *distance* or length from the tail of the vector to its tip. For simplicity, let's consider a case with $n = 2$, for a two dimensional Euclidean space (Fig 2.7).

If there are two arbitrarily positioned vectors $\vec{A} = (a_x, a_y)$ and $\vec{B} = (b_x, b_y)$, then the distance between the tips of the two vectors is also a vector called the displacement vector, represented by $\vec{r} = (a_x - b_x, a_y - b_y) = \vec{A} - \vec{B}$. Then, according to the Pythagorean theorem, the magnitude of $\vec{r}$ can be calculated as

$$\|\vec{r}\|^2 = \|\vec{A}\|^2 - \|\vec{B}\|^2 - 2\vec{A} \cdot \vec{B}, \quad (2.43)$$

which can be stated as

$$\|\vec{r}\| = \sqrt{\|\vec{A}\|^2 - \|\vec{B}\|^2 - 2\vec{A} \cdot \vec{B}}, \quad (2.44)$$

or

$$\|\vec{r}\| = \sqrt{(\vec{A} - \vec{B}) \cdot (\vec{A} - \vec{B})}, \quad (2.45)$$

where the term under the square root is the dot product of the same vector. If the relationship in (2.45) is used to find the distance between the tail and tip of one vector (magnitude), it can be rewritten as

$$\|\vec{p}\|_2 = \sqrt{\vec{p} \cdot \vec{p}} = \sqrt{p_x^2 + p_y^2}, \quad (2.46)$$

¹It is possible that an optimization problem has no constraints as it will be seen later on.

²There are some exceptions in cases of well established and widely used algorithms such as least squares.

Figure 2.7: A two dimensional Euclidean space with two arbitrary $\vec{A}$ and $\vec{B}$ and their corresponding displacement vector $\vec{r}$.

where $\vec{p}$ is an arbitrary vector from origin of space to point (p_x, p_y) . The equation in (2.46) is called the Euclidean norm, or the ℓ_2 norm for a two dimensional vector space; a vector space with defined norms is called a *normed space*.

The concept of norms can be easily extended for higher or lower dimensions¹. As a generalization, one can define a parametric equation for the norm of vector $\vec{x}$, and $p \geq 1$ (an ℓ_p space) to be

$$\|\vec{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}. \quad (2.47)$$

An important property of an ℓ_p space is the triangle inequality stating that for any $x, y \in \mathbb{R}^n$

$$\|x + y\| \leq \|x\| + \|y\|. \quad (2.48)$$

One significant issue to be addressed about norms is the case when $0 < p < 1$. A norm for this special case can be defined as in equation (2.47); however, it can be seen that such annotation violates the triangle rule in (2.48). For these special cases normed space is not convex (as opposed to $p \geq 1$ where it is convex) and are not desirable for practical situations; they do, however, have uses in fields of information theory.

So far ℓ_p norms have been examined except for one case: $p = 0$. Kalton et al.[34] has shown that the ℓ_0 normed space includes all other normed spaces. Similar to the case of $0 < p < 1$ this norm also has applications

¹It is important not to confuse the dimension of the vector space and the rank of the norm as there could be lower rank norms in a higher dimension space. For instance, a two dimensional vector space could have ℓ_2 and ℓ_1 defined.

Figure 2.8: (a) Visual representation of an ℓ_2 norm and (b) visual representation of an ℓ_1 norm. Dashed lines represent the unit circles.

in certain theoretical areas. Here, the interest is in another definition of an ℓ_0 "norm" which has the form

$$\|\vec{x}\|_0 = \lim_{p \rightarrow 0} \left(\sum_{i=1}^n |x_i|^p \right). \quad (2.49)$$

The norm in (2.49) is not a proper norm; however, this particular definition of norm can be applied to locally bounded sets (whereas the general p-norm applies to all infinite sequences of real and/or complex numbers). The zero "norm" introduced above simply finds the number of zero elements[20] in a set, or in case of a signal, the non significant components of data. Hence, one can appreciate its role in dealing with sparse signals, where it might be of interest to find the number of sparse coefficients. From this point onward, wherever the notation ℓ_0 or $\|\cdot\|_0$ is used, it refers to this notion of zero-"norm" and the quotations are dropped.

Graphically, norms can be illustrated as a surface encompassing a space, and the volume inside the surface is the set of all possible norms. This concept can become more clear with an example. In a Euclidean $\mathbb{R}^2$ space, an ℓ_2 norm can be realized as a two dimensional circle on a plane centred at the origin of the coordinate system (see Fig 2.8). The space encompassing set of all vectors with norm equal to 1 is called the *unit circle*. An ℓ_1 norm space can be visually conceived as a square. Two important properties of all visual representation of norms to consider are that they are symmetric about the centre; and also convex, as we move toward the centre of the circle we get closer zero (a minima). These graphs are useful in aiding with the explanation of certain norm operations such as norm minimization and norm approximation.

The importance of norms becomes apparent in their power as tools of solving systems of linear equations (homogeneous and non homogeneous). In linear algebra, the *null space* or *kernel* [50] is the set of all vectors $\vec{x}$ for which

$$Ax = 0. \quad (2.50)$$

If the above equation is a system of linear equations, the null space would be the solution of the (homogeneous)

system. However, the null space is not only limited to homogeneous systems of equations and can be utilized in solving non homogeneous systems of linear equations of the form

$$Ax = b \quad (Ax = y), \quad (2.51)$$

where the null space is the difference between (two) possible solutions of the system. In addition, the null space of a set is not affected by basic operations (addition, subtraction); hence, one can find a basis for the null space by performing elementary operations. Since the base of any vector space is a set of linearly independent vectors that span the set; once the base of the null space is acquired, all possible solutions can be generated. Therefore, the basis of a null space is the *optimal* solution to a system of equations.

The process of finding the basis of a set is generally referred to as *basis pursuit*, and can be formulated as a convex optimization problem. There are several different, well-established basis pursuit algorithms in practice; which involve norm approximation (or minimization). A generic representation of a norm approximation problem is described henceforth.

The general form of a system of linear equations in (2.51) can be rearranged as

$$Ax - b = 0, \quad (2.52)$$

where $A \in \mathbb{R}^{M \times N}$ is a transform matrix that maps elements of vector of variables (or solutions), $x \in \mathbb{R}^N$ to $b \in \mathbb{R}^M$; vector of observations or measurements. If Ax is expressed in terms of linear combination of form in equation (2.29) as

$$Ax = a_1x_1 + a_2x_2 + \dots + a_nx_n, \quad (2.53)$$

with $a_1, a_2, \dots, a_n$ as columns of transform matrix A ; then one can formulate the norm approximation problem for (2.52) as

$$\text{minimize} \quad \|Ax - b\|. \quad (2.54)$$

The objective of solving the relationship in (2.54), is to find a solution vector for x such that b (the measurements vector) can be approximated as best as possible. The difference between the actual b and its approximation is referred to as the *residual*

$$r = Ax - b, \quad (2.55)$$

and is calculated as a function of distance (norm). If the vector of variables is *translated* into another set where a solution can be found, it is said that the solution is found in the *translated null space* of A .

An extension of norm approximation problems is the so called *least norm problem* (LNP) of form

$$\begin{aligned} &\text{minimize} \quad \|x\| \\ &\text{such that} \quad Ax = b, \end{aligned} \quad (2.56)$$

where all matrices and vectors are the same as before; and the solution, $\hat{x}$, to the problem is called *least norm solution*. The least norm problem is a convex optimization problem that can be utilized to solve under-determined systems of linear equations; where $M < N$ for $A \in \mathbb{R}^{M \times N}$, and columns of A are independent. As one can observe

from (2.56), LNP problems are constrained, with constraint function $Ax = b$; which implies data consistency: solution vector x that solves the constraint is consistent with data. As an approximation problem, the LNP formulation can be thought of as a means to estimate (or approximate) the vector of variables x , given only $M < N$ linear measurements in b .

As a solution to under-determined systems, least norm problems are excellent candidates to solve problems in signal processing where fewer measurements are acquired than the number of columns of the transform matrix. By looking back at equation (2.39), it can be observed that the inverse compressed sensing problem may be formulated and solved as a least norm problem. It is; therefore, of great interest to examine possible solutions to the inverse CS problem by considering common least norm problems, specifically ℓ_2 and ℓ_1 norms; a discussion for ℓ_0 norm will also be included.

Inverse CS

To examine least norm recovery of a sparse signal undergoing compressive sensing, equation (2.39) should first be formulated as a LNP convex optimization program with a general p – norm:

$$\begin{aligned} & \text{minimize} \quad \|s'\|_p \\ & \text{such that} \quad \Theta s' = y. \end{aligned} \quad (2.57)$$

The most common approach to solving an under-determined system with a least norm solution is the *least square solution* of the form

$$\begin{aligned} \hat{s}' &= \min \|s'\|_2^2 \\ \text{such that} \quad & \Theta s' = y. \end{aligned} \quad (2.58)$$

The relationship in (2.58) resembles the equation for energy of a signal. Indeed, the least square solution attempts to find the solution with the least amount of energy in a signal. The least square solution is so well known that an analytical solution has been developed to specifically deal with such problems; with a convenient and fast algorithm that can be implemented as

$$\hat{s}' = \Theta^\dagger y, \quad (2.59)$$

where $A^\dagger$ is the pseudo-inverse matrix of A and can be written as $A^\dagger = A^T(AA^T)^{-1}$. As a compressed sensing reconstruction algorithm, the least square approach fails to provide the correct solution as illustrated by Candes et al [13]. The ℓ_2 reconstruction measures and minimizes signal energy but does not promote sparsity; which is one condition of required for a CS reconstruction. This is because the solution $\hat{s}'$ is almost never sparse (specially in the case of a practical signal with sudden changes in signal energy), and the components not selected in the sampling process are set to zero; this causes the recovered signal to contain many artifacts.

It was previously stated that the zero-norm introduced in (2.49) counts the number of zeros in a given set of elements; it is particularly useful in dealing with sparse signals.

$$\hat{s}' = \min \|s'\|_0$$

Figure 2.9: (a) Visual representation of an ℓ_2 norm minimization; for ℓ_2 norm the sparse solution (s') to the null space is not found (b) visual representation of an ℓ_1 norm minimization; since ℓ_1 circle has fewer solutions (solid edges), this method has a high chance of finding the sparse solution.

$$\text{such that } \Theta s' = y. \quad (2.60)$$

In fact, ℓ_0 minimization is guaranteed to find the sparsest solution among all possible answers. Hence, this solution can recover the signal of interest accurately. There are; however, practical limitations to utilizing the zero-norm. To recover a K -sparse signal ($K \ll N$), one must search through $C(n, r)$ [5] possibilities as a set of solutions; this is a highly time consuming task that is practically not feasible to implement.

While ℓ_0 and ℓ_2 norm minimizations fail in providing a solution for the inverse problem in (2.39), there is one least norm solution that can solve this problem: the ℓ_1 minimization. The ℓ_1 minimization was first used in the 1970s in a few seismology experiments [18], [56]; when it was observed that the sparse seismic data (signals with abruptly changing amplitudes) could be reconstructed using the ℓ_1 minimization. Graphically, solutions to ℓ_1 and ℓ_2 can be visualized (Fig 2.9) on norm circles (norm spheres in three dimensional space), where the point of contact between (translated) null space ($y = \Theta s'$) and the circles is the solution $\hat{s}'$.

Although it was used in practice, the mathematical rationalization of why the ℓ_1 minimization performs well in recovery of sparse signals did not come to be until recently [13].

The ℓ_1 minimization objective function with a linear equality constraint can be solved as a linear program through a convex optimization algorithm. The solution to this problem generates results with a large number of components close to zero; by doing this, ℓ_1 minimization imitates ℓ_0 recovery; while it is drastically more efficient. Nonetheless, ℓ_1 optimization method requires slightly higher number of samples for an *exact* reconstruction. This is a viable trade-off considering the disadvantages of both ℓ_0 and ℓ_2 minimization processes; and is somewhat compensated for with more efficient algorithms.

The ℓ_1 minimization technique is now developed as the main¹ formulated solution to recover an under-sampled sparse signal. Along with a fundamental understanding of magnetic resonance imaging, compressed sensing; and

¹Other methods have also been proposed and experimented with; such as greedy search methods.

with the three pillars of an effective CS process in place, it is now time to explore how CS can be applied to MRI as an effective tool.

2.4 Compressed Sensing MRI

Magnetic Resonance Imaging is an inherently slow image acquisition technique; attracting immense research efforts for methods to reduce image acquisition time. Meanwhile, Compressed Sensing is a recent development in signal processing with technical elements that allow a (sparse) signal to be compressed, and recovered with fewer collected than that required by the Nyquist Theorem. One element that makes the application of CS to MRI in particular, is the fact that compressed sensing proposes to reduce the number of samples during sample acquisition.

In MR imaging, image acquisition time is directly influenced by the number of samples required; the more samples collected the longer the acquisition time would become. Hence, the prospect of reducing the samples collected via CS could have an immense impact on scan time reduction. There are; however, a few requirements to qualify MRI as a candidate for compressed sensing; namely sparsity, incoherence sampling, and the existence of a reconstruction solution.

2.4.1 Magnetic Resonance Image Sparsity

Most natural and medical image are sparse, and MR images are no exception. MR images are sparse in the pixel domain; but they are encoded during the acquisition process. Samples for MR image are gathered in the frequency domain; these images (the k-space) are also sparse (see Fig. 2.5(b)), rendering the application of sparsifying transform needless.

Although the Fourier domain representation of an MRI image is very sparse, different images can be made even sparser in other domains. This can be shown by applying different sparsifying transforms to a conventional, pixel domain, MR image and reconstructing the original image with a subset (significant components) of the total number of coefficients. Results from experimental reconstructions [39],[40] demonstrate that, certain transforms perform better for a class of images. For instance, angiograms depict better quality when they undergo finite differences sparsifying transform, while brain images perform better with wavelet transforms; while using only about 5% to 10% of samples. In addition to single sparsifying transforms, some research works [53] have shown improved results by combining two or more transforms (each having a different weight factor).

2.4.2 Incoherent Sampling

The concept of incoherent sampling was introduced in section 2.3.2; where it was explained how incoherence works along with its relevance to designing a good sampling matrix. Nonetheless, practical implications of choosing an incoherent sensing matrix in a CS problem were not analyzed. Therefore, the goal is to explain the significance of incoherence sampling in compressive sensing reconstruction, in a practical context.

The importance of incoherent sampling is best explained with a simple example. The example provided here is taken directly from [40] and [31]. Let us consider one of the most basic and common signal processing prob-

Figure 2.10: A generalization of frequency domain of a signal, $F(t)$ represents the forward Fourier transform. (a): Original signal (time domain) sampled at Nyquist rate, the Fourier domain information contains all significant components and reconstruction can recover the original signal. (b): Samples collected at equally spaced intervals but at sub-Nyquist rate, the resulting frequency domain plot includes the combination of shifted frequency components; reconstruction of these components would not result in the original signal being recovered (the resulting signal would be a superposition of aliased signal copies). (c): Original signal sampled at sub-Nyquist rate with randomly (incoherent) spaced intervals between samples. This time the aliasing components in the frequency domain behave as noise; and the non-sparse coefficients can be recovered by thresholding above the noise ceiling.

Figure 2.11: After the first round of thresholding is performed (top-left), the stronger non-zero components are recovered. Consequently, it is possible to calculate their contributions to the original interference. After this is done (bottom-right), the resulting “noise” level will be lower, revealing the smaller non-sparse components; which can be recovered by setting another (lower) threshold (top-right). Threshold levels are also designed with respect to the existing interference level.

lem; reconstructing a signal from its frequency information. Here, we assume that the signal of interest is sparse in its frequency domain. Let us also assume that we would like to use CS to compress and recover the signal (since the signal is sparse), meaning that the signal is not sampled at Nyquist rate. If the samples are taken at equally spaced intervals, the reconstruction of the original signal results in ambiguity. The ambiguity is caused by aliasing artifacts since the resulting frequency domain is a combination of overlapping copies of significant coefficients (aliasing artifact). In contrast, if samples are acquired in randomly spaced intervals (at sub-Nyquist rate), the resulting aliasing components in the frequency domain would reduce to *noise like*¹ components. A simple thresholding mechanism would be enough to recover the significant components of the frequency information (see Fig. 2.10). In some cases it is possible that one or more of the non-sparse coefficients are not recovered by first attempt to threshold the noise out. This is mainly due to the fact that the noise level may sometimes be as high as these components. To recover such components, iterative thresholding is performed. After the first thresholding recovers some non-zero components, the role they play in the original aliasing interference is calculated (since the energy of these components is known) and deducted from the Fourier domain results in Fig. (b). Doing so results in another reduction in the interference level that enables the detection of lower energy significant components. Thresholding the frequency domain once more results in all non-sparse components to be recovered. This dis-

¹These artifacts are not actually noise, rather a reduction in energy of significant frequency components due to aliasing; non significant components of the signal may also see their energy increase, but this increase causes no harm as we are not interested in these components. Furthermore, they will be eliminated through thresholding.

cussion gives rise to the implementation of an iterative thresholding approach in reconstruction of under-sampled signals in CS, whereby an exact recovery is possible [26]. In addition, this methods gives CS reconstruction the property of being a noise reducing technique, since low frequency noise components or artifacts naturally available in a signal can also be reduced significantly with thresholding.

In performing a compressed sensing acquisition for MRI, the goal is to sparsely and incoherently sample the k-space. One has control over the trajectory of sampling pattern in k-space through pulse sequences that control frequency and phase encoding gradients. Hence, a sensing matrix can be designed by using the appropriate pulse sequence; and considering the fact that the MRI hardware has limitations in terms of how fast and in which direction the gradients can move. There are other issues that must be addressed when designing a sampling trajectory for MR image acquisition that are discussed below.

1. Observing from Figure 2.5(b), one can immediately detect that most of information of an image is concentrated at the centre of the k-space. The reason behind this phenomenon is that most of the energy of the signal is contained in the lower frequency components (centre of k-space). This; however, does not indicate that collecting samples from only the centre of k-space is enough. The higher frequency components of an image (outer bounds of k-space) provide high resolution components that are required for construction of a good quality image. In order to be able to reconstruct a good quality image with one should collect samples from both low and high frequency elements of k-space. Nevertheless, the density of samples amassed need not be evenly distribute throughout the k-space, as the frequency components themselves are not evenly spread through the spectrum. Generating *variable density* [60] sampling trajectories for compressed sensing MR image acquisition; in which the centre of k-space is sampled more heavily than anywhere else, provides both incoherence and randomness while reducing aliasing artifacts and noise.
2. Performing a completely random, two dimensional, k-space sampling demands that x (frequency encoding) and y (phase encoding) gradients need to be changing randomly, and simultaneously. Given that every time a gradient has to change, the gradient coils inside the MRI machine have to move (and randomly not in a sequential manner), and the fact that the they are restricted by slew rate and amplitudes; doing so substantially increases image acquisition time beyond any practical measure (it is ‘almost’ impossible to implement such a trajectory).

These two issues together necessitate design of sampling trajectories that emulate randomness, while preserving incoherence; in addition to providing variable density sample collection. These intentionally designed patterns are called *pseudo-random* since they are not designed to be incoherent. Creating a sampling trajectory with such arrangements is an active field of research in MRI and compressed sensing MRI; that has given rise to many different trajectories already in use. Some of these patterns available are variable density Cartesian sampling; uniform, variable density, and variable density perturbed spirals, and radial trajectories. A few papers have proposed other trajectories such as the two dimensional variable density Cartesian sampling [64]. In this work the focus is on variable density Cartesian sampling, as it is the most commonly used acquisition method in clinical setups. Furthermore, other variable density trajectories such as radial and spiral techniques are mostly suited for rapid and dynamic MRI; which are not subjects of interest in this thesis.

Figure 2.12: Results of different sampling patterns

2.4.3 Compressed Sensing Image Recovery

It has been shown that for a sparse signal, the best way to recover the signal while preserving sparsity and achieving an exact recovery is to use the ℓ_1 minimization convex optimization of the form:

$$\begin{aligned} \min_x \quad & \|\Psi x\|_1 \\ \text{such that} \quad & y = \Phi x, \end{aligned} \quad (2.61)$$

where Ψ is the sparsifying transform, Φ is the encoding matrix, x is the signal to be recovered, and y is the sampled measurements.

In the case of MRI signal recovery, the signal x is a 2D image that is to be recovered from y , sparse measurements of its k-space. The k-space is sampled using one of aforementioned sampling trajectories; which selects Fourier coefficients from the k-space. The under-sampled sensing matrix is represented by $\mathbb{F}_s$; the Fourier transform of the sampling trajectory. The for MRI CS recovery, equation (2.61) becomes

$$\begin{aligned} \min_x \quad & \|\Psi x\|_1 \\ \text{such that} \quad & y = \mathbb{F}_s x. \end{aligned} \quad (2.62)$$

The choice of sparsifying transform could have considerable implications on the quality of the reconstructed image, much like the sampling trajectory. One could choose the Fourier transform, readily available k-space, to be the sparse transform. This is indeed an intuitive choice, as it requires no more transformations performed on the image. However, Chen et al. [17] suggests that promoting sparsity in a signal improves results acquired from a basis pursuit algorithm in the form of ℓ_1 minimization. Additionally, no matter how incoherent the sampling pattern utilized is, a purely Fourier reconstruction is bound to contain artifacts, as there is a finite number of Fourier harmonics in a practical signal. Therefore, there is need for a transform that reduces these artifacts while promotes sparsity; beyond that resulted from Fourier reconstruction.

The original paper [13] on CS introduces a sparse transform, that is especially useful for images, called the *total variation* of an image, which is an objective function for the finite differences evaluation of an image. Another sparsifying transform, among others, is the wavelet transform; which will be explained in the next chapter.

2.5 Summary

This chapter began by providing an introduction to the physical principles surrounding the operation of an MRI machine. It then explained how images are acquired by MRI and how one can control how images are acquired using pulse sequences that define which frequency components are selected in a frequency plane, called k-space. Afterwards, the concept of compressed sensing was introduced, from a signal processing point of view. The three main requirements for a successful CS implementation were also outlined. With the fundamentals of both MRI and CS discussed; the final section was allocated to an explanation of the integration of CS into MRI image acquisition. It was pointed out that an MRI image reconstruction can be performed using compressed sensing because

- MRI image are sparse in a coded transform domain (i.e. wavelet)
- incoherent sampling is possible in the process of image acquisition in MRI
- non-linear optimization reconstruction techniques are available for image reconstruction purposes.

Hence, CS is a suitable reconstruction technique that can be used for MRI image reconstruction with potential to reduce acquisition time.

Chapter 3

From Fourier to Wavelets

3.1 Overview

3.2 Fourier Transform

In the world of signal processing, it is often desired to look at a signal domains other than its original time domain (spatial domain in case of an image). The frequency domain of a signal, for instance, provides excellent information that can be used to analyze a signal and extract useful features; by means of filtering. In order to obtain the frequency spectrum of a linear time-invariant (LTI) signal, the Fourier Transform (FT) must be applied to the signal. Frequency domain signal analysis is so significant that the Fourier transform has become a staple to almost all signal processing applications.

Fourier analysis, a form of harmonic analysis, suggests that a finite signal (or any energy function) $f(t)$ can be represented as the sum of sinusoidal waves, also called *basic waves* as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} d\omega, \quad (3.1)$$

where $\omega = 2\pi f$, for a frequency f cycles per second, and $\hat{f}$ is the amplitude of each sinusoid. The complex exponential along with the amplitude of the signal is called the Fourier coefficient and is a complex number. Ideally, an infinite number of basic waves can represent the signal perfectly; however, in practical signal processing, the number of harmonics is finite, creating less than perfect representations. The Fourier transform is the function representing the amplitude of each sinusoidal wave as

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt. \quad (3.2)$$

The equation in (3.2) is the forward Fourier transform problem, and (3.1) is regarded as the inverse Fourier

transform. For a discrete signal equations (3.2) and (3.1) can be re-written as

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-i2\pi\omega kn/N}, \quad (3.3)$$

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i2\pi kn/N}, \quad (3.4)$$

where (3.3) is the forward discrete Fourier transform and (3.4) is the inverse. By decomposing the signal into various sinusoids, one can determine how much each frequency contributes to the overall signal. This can be visually depicted (recall the k-space of an MRI image) in the frequency domain of the signal; when a certain base signal at a certain frequency ω_1 is integrated and has a large value $|\hat{f}(\omega_1)|$, then it contributes more, while another base signal with frequency ω_2 may equate to a lower value in $|\hat{f}(\omega_2)|$; hence contributing less to the overall signal.

As mentioned in the beginning of this section, Fourier transform can be applied to LTI signals. Since the Fourier analysis uses sinusoids as basic harmonics, it can be used to represent a periodic signal. However, Fourier analysis is not limited to periodic signals and can be expanded to accommodate aperiodic signals. This is done by limiting the integration interval in (3.2) as required; indicating that Fourier analysis has a *compact support*, meaning that it can represent signals of finite interval lengths. Hence, the Fourier analysis is suited for applications to a wide variety of practical signals.

Despite being a powerful tool in signal processing, Fourier transform does have its disadvantages. By close inspection of equation (3.2), one can observe that $\hat{f}\omega$ is dependent on values of signal for the complete duration of that signal, requiring all of the signal for analysis (or an interval). Nevertheless, if the time interval of interest is very small or if there are abrupt changes in the signal at any time, it becomes very difficult to estimate the signal using Fourier coefficients. In general, the smoother the signal is the easier it is to approximate the signal; while sudden changes require an increased number of coefficients to, if at all possible, be able to estimate the signal. Hence, the Fourier transform is said to provide global information, and is unable to give a clear picture about any local event in a signal (loss of time information).

3.3 Short-Time Fourier Transform

Many real life signals are non-stationary¹ and may include many sudden changes; which is not desirable when performing Fourier transform. In order to address this shortcoming of FT, modifications need to be made to the original FT in (3.2). A window function can be added to the equation that results in

$$STFT\{f(u, \omega)\} = \int_{-\infty}^{+\infty} f(t)g(t-u)e^{-i\omega t} dt, \quad (3.5)$$

where

$$g_{u,\omega} = g(t-u)e^{i\omega t}. \quad (3.6)$$

¹ A stationary signal is a signal whose time domain properties do not vary significantly.

In (3.6), $g_{u,\omega}$ is the *symmetric* (i.e. $g(t) = g(-t)$) window function and u is the translation variable for the window function. This relationship is analogous to the well known convolution integral and is called the Short-Time Fourier Transform (STFT) or a windowed Fourier transform, first introduced by Dennis Gabor in 1946 [28]. As it can be observed from (3.5), the window function is centred around $t = u$, essentially localizing the FT about a certain point in the time span of signal. The process of performing a STFT operation starts by choosing a

Figure 3.1: A visual representation of the windowing process of a signal $f(t)$. Left: A window function $g_{u,\omega}$ of width T is centred at time u . The signal energy in this window is represented on a time-frequency plane (Right). The window is then moved and process repeated until the signal is finished. Right: Each time a segment of the signal is *windowed* it is represented on a time-frequency plane, via a Heisenberg box. The accumulation of these boxes for the whole signal makes up the spectrogram (note that all boxes are equal in terms of area).

window function, $g_{u,\omega}$, with the appropriate *time width* (also called the support of the window); different window functions can also have different weights. The window is then centred at the start of the signal $f(t)$, overlapping a portion of the wave. The overlapping segment of the signal is multiplied by the windows function, and the extracted event is the new signal, $f_1(t)$, under study. Thereafter, the FT of $f_1(t)$ is calculated; which provides the frequency information for a portion of the original signal. Then, the window is shifted to another location in time, and the same evaluations are made. This process is repeated for the entire length of the signal. Hence, it can be observed that by performing this process, the STFT provides localized frequency information about an event at a certain point in time. The time and frequency information obtained from each windowed transform can be represented visually as the energy spread of signal f about time u and frequency ω for the window function $g_{u,\omega}$. This information is expressed in a time-frequency plane using a *Heisenberg box*, whose widths (σ_t as time spread width and σ_ω as frequency spread width) indicate the time and frequency energy spread of the window function. Together these boxes for all windows of the define an energy density spectrum called the *spectrogram* of the signal (Fig 3.1); which can be mathematically stated [44] as

$$P_s\{f(u, \omega)\} = |STFT\{f(u, \omega)\}|^2 = \left| \int_{-\infty}^{+\infty} f(t)g(t-u)e^{-i\omega t} dt \right|^2. \quad (3.7)$$

There are several already established windowing functions one can choose from, such as Rectangular, Ham-

ming, Gaussian, and Hanning windows; each of which have their own advantages and disadvantages and are appropriate for different types of signals. All of these window functions are even and symmetric, as they are required to be. However, whatever the choice of the window may be, one thing remains constant: the width of the windows are independent of the frequency of the signal, ω , and the time, u , at which the signal is being examined. This means that once the window function is chosen, its width is constant for the duration of the process. Let us also re-state that a windows function is both symmetric and even, such that:

$$g_{u,\omega}(t) = g_{u,\omega}(-t), \quad (3.8)$$

where $g_{u,\omega}$ is centred around u in time. In addition, since the choice of u and ω do not change widths of the

Figure 3.2: An illustration of Heisenberg boxes on a time-frequency plane at different time and frequency selections. The widths of boxes are σ_t and σ_ω for time width and frequency width, respectively; where $\sigma_t \sigma_\omega = C$ is constant. Note that the energy peaks at the centre of the box and diminishes as it spreads out.

window,

$$g_{u,\omega}(t) = g(t), \quad (3.9)$$

and $f(t)$ in equation (3.7) becomes $t - u = t$; since the concern here is the window itself not the signal under study. By combining all these facts and assumptions, one can calculate the time width of the window as:

$$\sigma_t^2 = \int_{-\infty}^{+\infty} t^2 |g(t)|^2 dt, \quad (3.10)$$

which is illustrated in Fig 3.2. By asserting that the Fourier transform of $g_{u,\omega}$ is (by (3.2))

$$\hat{g}_{u,\omega_1} = \hat{g}(\omega - \omega_1) e^{[-iu(\omega - \omega_1)]}, \quad (3.11)$$

and observing that equation (3.7) is in time domain, the frequency energy spread around ω_1 can be written as

$$\sigma_\omega^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \omega^2 |\hat{g}(\omega)|^2 d\omega. \quad (3.12)$$

Hereafter, σ_t is referred to as window length, and σ_ω is called the frequency bandwidth of the windowing function. Since, the energy spread around time and frequency are independent of u , and ω , the area of a Heisenberg box corresponding to a window function of a STFT is always constant: $\sigma_t \sigma_\omega = C$. Coupled with the fact that the window length is also constant during the process signal transformation; the constant area of the box means that the time and frequency resolutions will remain the same. Although it is possible to alter resolutions by scaling the window function, it is impossible to acquire high time and frequency resolutions simultaneously, as the two are reciprocally related.

The Short-Time Fourier Transform, addresses some of the issues of traditional FT, by incorporating windowing functions to transform and extract information from a signal (which is non-stationary on a global time frame) one small segment at a time. However, the STFT assumes that the signal is stationary within the chosen window. Possible problems arise when this is not the case: when a signal is non-stationary within a window. This problem can be resolved by decreasing the length of the window (compact support); which results in poor frequency resolution, as length of the window is the same everywhere. In addition, in general, since the signal is windowed before it is transformed into frequency domain information, the spectral information acquired by performing a FT on the signal is restricted by the support of the window.

Despite being more flexible than the original FT, the STFT fails to provide sufficient versatility required to process signals with a large number of discontinuities and rapid changes; such as images. The desire to allocate a transform for such signals has led to the development of the *Wavelet Transform* (WT); which provides a solution for constant time-frequency resolution of STFT. The following is an introduction to the world of wavelets, explaining their strength in dealing with fast varying signals.

3.4 Wavelet Transform

A wavelet, as the name suggests, is a little wave; a wave whose energy is finite in time and/or space (in contrast to a regular wave or signal). The word wavelet was a direct consequence of the work of a geophysicist named Morlet. He later collaborated with a theoretical physicist by the name of Grossman [29]; which, in 1984, marked the beginning of the process of formalization of the Wavelet Transform (WT). Morlet worked in reflection seismology; which involves transmission of a high frequency waveform and gathering its return to surface in an attempt to distinguish between layers of geological layers. However, traditional waveforms used for this reason were of periodic nature, and possessed high frequencies; making it difficult to separate returns off closely spaced features. The solution to this problem was found in using smaller duration pulses of high frequency. These smaller pulses can be created by performing scaling and translation operations on the original waveform, ψ ; called a mother wavelet. A class of smaller waveforms, the children, are created from the mother wavelet with different dilation and scaling factors is called a wavelet dictionary. A mother wavelet, ψ , described in (3.13), has zero average over

the time span of the corresponding wavelet.

$$\hat{\psi}(0) = \int_{-\infty}^{+\infty} \psi(t) dt = 0. \quad (3.13)$$

The mother wavelet can be translated in time by u and scaled by a scaling factor of s , such that

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right), \quad (3.14)$$

where $u \in \mathbb{R}$ and s is a positive number. Before Morlet and Grossman provided a formal explanation of wavelets, Alfred Haar[30] laid the groundwork for wavelets when he proposed a piecewise constant function with the form

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

which can form the orthonormal basis of a $\mathcal{L}^2(\mathbb{R})^1$ when it is scaled and translated according such that

$$\psi_{u,j}(t) = \left\{ \frac{1}{\sqrt{2^j}} \psi\left(\frac{t-2^j u}{2^j}\right) \right\}_{(j,u) \in \mathbb{Z}_2 \times [0,1]}. \quad (3.17)$$

The function defined in (3.15) is called the *Haar* mother wavelet. by comparing the two relationships in (3.14) and (3.15), it can be recognized that the Haar wavelet is a more specific case of the generalized wavelet function introduced by Morlet and Grossman. Hence, the equation for a mother wavelet in (3.14) can be said to form an orthonormal basis for all signals $f(t)$, where $f(t) \in \mathcal{L}^2(\mathbb{R})$. Any function that possesses this property can be represented as a set of *wavelet coefficients* defined by

$$WT\{f(t)\} = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt, \quad (3.18)$$

where $\langle . \rangle$ is the inner product operator. The relationship in (3.18) is the *Continuous Wavelet Transform (CWT)* with respect to mother wavelet ψ ; of the signal being examined. By translating and scaling the mother wavelet, a dictionary of *daughter* wavelets are created, each of which modulate the signal. Then the original signal can be reconstructed by adding each segment of signal modulated by a daughter wavelet; this can be shown

¹ $\mathcal{L}^p$ spaces, where $p \in \mathbb{N}^+$, are extensions of normed vector spaces introduced in chapter 2. The space $\mathcal{L}^p(X)$ is a function space that includes all finite energy, integrable functions of the form

$$\left\{ f: X \rightarrow \mathbb{R} \mid \int_X |f(x)|^p dx < \infty \right\} \quad (3.16)$$

An orthonormal basis of this space (analogous to orthonormal basis of a vector space) is a function or a set of functions, that spans the space. Hence, $\mathcal{L}^2$ is the space of all finite energy, square integrable signal functions.

Figure 3.3: An illustration of Heisenberg boxes on a time-frequency plane for three different wavelet functions ψ_1 , ψ_2 , and ψ_3 ; each with scaling and translation factors of u_1 , u_2 , u_3 , and s_1 , s_2 , s_3 , respectively. Note that $s_1 < s_2 < s_3$. Once again $\sigma_t \sigma_\omega = C$ is constant. Hence, as scale s decreases, so does the time support, but since the area of the box is constant, the frequency support is increased and moves toward higher frequency. On the other hand, when scale is increased, the time support also increases at the expense of frequency support at lower frequencies. In this manner the frequency response of the wavelet can change by adjusting the scale factors, this is crucial in filtering analysis of wavelets.

mathematically as

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{s^2} WT\{f(t)\}_{u,s} \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du ds. \quad (3.19)$$

The reconstruction can only be applied if and only if

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty. \quad (3.20)$$

The relationship in (3.19) is the *inverse continuous wavelet transform* (ICWT) for the signal $f(t)$. However, the inverse operation can only be carried out successfully, if the wavelet function meets the condition in (3.20) for constant C_ψ ; called the *admissibility* condition. The admissibility condition restricts the choice of wavelet functions, so that the original signal can be recovered completely.

The wavelet transform in the form stated above is capable of transferring signal features of different scales onto a time-frequency plane. Moreover, certain wavelets can also separate amplitude and phase properties of a signal, and much the same as STFT, analyze transient features of a signal. These wavelets are called *analytical* or complex wavelets. Another category of wavelets; which are capable of inspecting abrupt transitions in the signal is called *real* wavelets. Henceforth, when the term wavelet is used, it implies a *real* wavelet. Regardless of the

type of wavelet, in contrast to the STFT, wavelet transforms are capable of analyzing signal structure of varying window sizes; this is demonstrated in Figure 3.3 Wavelet Transforms ability to analyze features based on scales is instrumental to its rapid expansion into the world of signal processing, as it provides flexibility in terms of time and frequency resolutions.

Continuous wavelet transforms are well suited to represent continuous time signals. However, the subject of this work is on edge detection of images; whereby images are taken as original signals. Since images are discrete signals, CWT is not the appropriate basis for their description. Hence, *discrete wavelet transforms* (DWT) need to be developed to be utilized in conjunction with discrete signals and images. Discrete wavelet transforms also provide the groundwork for numerical analysis of signals as well as fast computational algorithms. Nevertheless, before offering an introduction to DWT, there is one more concept related to wavelets that is worth mentioning: *wavelet scaling function*.

Every mother wavelet function has a corresponding scaling function, also called the *father wavelet* that is embedded in a mother wavelet. The scaling function, ϕ has the form

$$\phi_{s_0}(t) = \frac{1}{\sqrt{s_0}} \phi\left(\frac{t}{s_0}\right), \quad (3.21)$$

where the scaling factor of the father wavelet enables the mother wavelet to reproduce daughter wavelets that are not within the frequency range of the mother wavelet itself. To clarify, let us go back to the original wavelet equation in (3.18); where now it is assumed that the mother wavelet is not translated, only scaled. Then the wavelet function can be re-written as

$$\tilde{\psi}_s(t) = \frac{1}{\sqrt{s}} \psi^*\left(\frac{-t}{s}\right), \quad (3.22)$$

with its Fourier transform stated as

$$\widehat{\tilde{\psi}}_s(\omega) = \sqrt{s} \hat{\psi}^*(s\omega). \quad (3.23)$$

Considering that the wavelet function has zero average (see 3.13), the frequency domain information (3.23) of the mother wavelet indicates that it is a band pass filter. With the help of the scaling function ϕ , the mother wavelet ψ can produce band pass filters with different bandwidths. Combining the two wavelets, the wavelet transform equation in (3.18) can be written as

$$WT\{f(t)\} = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \phi\left(\frac{t-u}{s}\right) dt. \quad (3.24)$$

With the added knowledge of the effect of the father wavelet, there is now sufficient information to understand the basics of the DWT and how its applications to image processing.

3.4.1 Discrete Wavelet Transform

The continuous wavelet function that belongs to the family of $\mathcal{L}^2(\mathbb{R})$ functions, and is able to represent a continuous signal. In order to represent a discrete signal, $f[n]$ using a subset of values for s and u that forms the basis of $\mathcal{L}^2(\mathbb{R})$ that is able to duplicate localization properties of a continuous wavelet function, while representing the

signal. In essence, a discrete wavelet function is the direct result of discretization of a continuous wavelet function. Hence, the task is divided into two stages: discretization of scale the parameter, and discretization of the translation parameter.

Let us assume that signal $f[n]$ is sampled at uniformly spaced intervals of $\frac{1}{N}$. Then, the discretization of the wavelet transform is possible by sampling the continuous wavelet function. For the scaling factor s this can be done by setting its value to a particular value; whereby $s = a^j$, where $j \in \mathbb{Z}$. The time translation parameter of the wavelet can also be set to a set value k ; where $u = k$, and the fixed value can be changed multiplication to a constant $b \in \mathbb{Z}$. Hence, the discrete wavelet function $\psi_{k,j}$ is

$$\psi_{k,j} = \frac{1}{\sqrt{a^j}} \psi\left(\frac{n - ba^j k}{a^j}\right). \quad (3.25)$$

A special case of the discrete wavelet function is called a *dyadic* wavelet, for $a = 2$ and $b = 1$ which has the form, whereby the relationship in (3.25) can be stated as

$$\psi_{k,j} = \frac{1}{\sqrt{2^j}} \psi\left(\frac{n}{2^j} - k\right). \quad (3.26)$$

The dyadic wavelet is the basis of $\mathcal{L}^2(\mathbb{R})$; space of all finite energy signals (i.e. images). The wavelet coefficients of a discrete signal, $f[n]$ with $f \in \mathcal{L}^2(\mathbb{R})$ are calculated by a fast wavelet transform algorithm as

$$DWT f[n] = \sum_{m=0}^{N-1} f[m] \psi_{j,k}^*[m - n] = f[n] \otimes \bar{\psi}_{j,k}[n], \quad (3.27)$$

where the size of the signal is N , $\bar{\psi}_{j,k} = \psi_{j,k}^*[-n]$, and $\otimes$ represents the circular convolution operator; and wavelets are treated as periodic signals of period N . Figure 3.4 assists with the understanding of how the dyadic wavelet function transforms and scales a signal. Since the scaling parameter of the dyadic wavelets is s^j , its *scale* resolution is increased by factors of 2 while its corresponding frequency resolution ($T = \frac{1}{\text{frequency}}$) is decreased by a factor of two. The frequency resolution of $\frac{1}{2^j}$ is referred to as the resolution of the wavelet, while its inverse 2^j is called the scaling parameter (corresponding to the n axis in the figure). This is the underlying principle of what is called *Multiresolution Analysis* (MRA) introduced by Stephane Mallat [42] and explained in more detail in [44].

3.4.2 Multiresolution Analysis

The idea behind the MRA is to provide a set of conditions according to which a wavelet function can be constructed. The wavelet construction by means of MRA analysis is not unique. It is in the context of MRA that the scaling function (introduced before) becomes significant. The discrete scaling function $\phi_{k,j}$ can be stated as

$$\phi_{k,j} = \frac{1}{\sqrt{2^j}} \phi\left(\frac{n}{2^j} - k\right). \quad (3.28)$$

Together the scaling wavelet and the dyadic (mother) wavelet can be used to develop a family of wavelets with different frequency properties. In a nutshell, the mother wavelet constitutes the wavelet shape while the father

Figure 3.4: An illustration of the dyadic wavelet function windowing in the time frequency domain. Note that the frequency value is fixed and scaled down by factors of 2. The dyadic wavelet illustrates flexibility in terms of scale and frequency resolutions by being able to analyze a signal at different scale resolutions while selecting a certain frequency resolution.

wavelet determines the wavelet scale.

An MRA is a set of closed subspaces $V_j | j \in \mathbb{Z}$ of $\mathcal{L}^2(\mathbb{R})$, such that

$$V_j \subset V_{j-1} \subset \mathcal{L}^2(\mathbb{R}), \quad (3.29)$$

which implies that smaller values of j means a larger subspace. The statement in (3.29) is called the *nesting* property of MRA. Moreover, there are two conditions for when $j \rightarrow +\infty$ and $j \rightarrow -\infty$:

$$\lim_{j \rightarrow +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = 0, \quad (3.30)$$

$$\lim_{j \rightarrow -\infty} V_j = \text{Closure} \left(\bigcup_{j=-\infty}^{+\infty} V_j \right) = \mathcal{L}^2(\mathbb{R}). \quad (3.31)$$

In addition, the subspaces in MRA are related together with a *scaling* property whereby

$$V_j = D_2(V_{j-1}), \quad (3.32)$$

where D is a scaling operator and $D_2 f(t) = f(\frac{t}{2})$. This property is central to Multiresolution; implying that all subspaces are related to the original space V_0 by a scaling operator. The MRA also has a *translation invariance* property asserting that

$$\text{if } f(t) \in V_j \Rightarrow f(t - 2^j) \in V_j, \quad (3.33)$$

which indicates that V_j is translation invariant as long as the translation factor is proportional to 2^j , keeping in mind that scale 2^j is the inverse of resolution 2^{-j} . One more important property of MRA, condition of existence of scaling function, states that $\exists \phi \in V_0 | \phi_{0,n}$ is an *orthonormal* basis of V_0 where $n \in \mathbb{Z}$ and $\phi_{k,j}$ is in the format of equation (3.28). This indicates that, since V_j is a subset of (scaled version of) V_0 , and $\phi_{0,n}$ an orthonormal basis of V_0 , then $\phi_{k,j}$ is an orthonormal basis of V_j . In addition, if all of these conditions hold valid for a subspace V_j (i.e. V_j is an MRA of $\mathcal{L}_2(\mathbb{R})$) then there exists a wavelet function that has the format indicated by equation (3.26), and is an orthonormal basis of $\mathcal{L}_2(\mathbb{R})$ space (capable of representing functions in that space). Hence, one can now construct wavelet and scaling function (that are orthonormal bases of V_j and $\mathcal{L}_2(\mathbb{R})$, respectively), given that the MRA requirements are met. More importantly, once the orthonormal basis of any subspace V_j is found, it is possible to find the *orthogonal* basis for that space. This concept is explained next.

3.4.3 Orthogonal Wavelet Bases

In the previous section it was indicated that the scaling function $\phi_{k,j}$ is an orthonormal basis for the subspace $V_j | j \in \mathbb{Z}$. Then, it follows that $\phi_{k,-1}$ is an orthonormal basis for V_{-1} and $\phi_{0,0}$ is orthonormal basis of V_0 subspace; where $V_0 \subset V_{-1}$ according to the nesting property in (3.29). Since both of the scaling functions are orthonormal, they can be related together via an inner product operator as

$$h_k = \langle \phi, \phi_{k,-1} \rangle, \quad (3.34)$$

where $\phi_{0,0} = \phi$ and the norm of their inner product is

$$\sum_{k \in \mathbb{Z}} \|h_k\|^2 = 1, \quad (3.35)$$

indicating that $\langle \phi_{0,0}, \phi_{k,-1} \rangle$. In addition

$$\phi_{k,-1} = \sqrt{2}\phi[2x - k]. \quad (3.36)$$

Combining the relationships in (3.36), (3.34), and acknowledging (3.35) one can come to the conclusion [50] that

$$\phi = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi[2x - k]. \quad (3.37)$$

Here, h_k is defined as a filter that essentially down-samples a signal by a factor of two. Therefore, according to (3.37) any scaling function can be represented by h_k . The only thing remaining is representing the wavelet function.

Let us now introduce a subspace of W_j that is an orthogonal complement¹ of V_j , where also $V_j \subset V_{j-1}$. Then $V_{j-1} = V_j \oplus W_j$. The subspace W_j has the scaling property stated in (3.32); therefore, the wavelet function, $\psi_{k,0}$, is an orthonormal basis for W_0 iff $\psi_{k,j}$ is and orthonormal basis of W_j . The orthogonal projection of elements of

¹If $x \in V$ and $y \in W$, then W is said to be an orthogonal complement of V if and only if $\langle x, y \rangle = 0$. Further, if $z \in V_i$ where $V \subset V_i$ then $z = x + y$.

W_0 onto W_j is given by an inner product operator such that

$$g_k = \langle \psi, \phi_{k,-1} \rangle = (-1)^k h_{-k+1}, \quad (3.38)$$

yielding the orthonormal basis for W_0 as

$$\psi = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \phi[2x - k] = \sum_{k \in \mathbb{Z}} g_k \phi_{k,-1}, \quad (3.39)$$

where g_k is also a filter that complements h_k , and mirrors its operation (inclusion of the -1 factor), together h_k and g_k are used in the development of the *Filter Bank* theory. The filter bank theory provides guidelines for construction of filters ($\bar{h}$ and $\bar{g}$) that are able to decompose and reconstruct a signal. Prior to the development of filter banks, one needs to understand how the two bases established in (3.37) and (3.39) are utilized to transform a signal.

In order to approximate a function, $f \in \mathcal{L}^2(\mathbb{R})$, by a (sub)space V_j , one should project the signal onto that set, denoted as $P_{V_j}f$. Recall equation (3.30) suggesting that for a resolution 2^{-j} , as j increases, the resolution degrades and information on the signal is lost. In contrast equation (3.31) indicates that as j increases so does the resolution and the approximation of the signal (P_{V_j}) is closer to the original signal; where the approximation error can be defined as

$$\|f - P_{V_j}\|. \quad (3.40)$$

The approximation error approaches zero as $j \rightarrow -\infty$. Furthermore, the nesting property of MRA in (3.29) guarantees that if approximation is made at resolution 2^{-j} , it contains all necessary information to project an approximation at a lower resolution of 2^{-j-1} ; while the scaling property in (3.32) scaling a function in V_j (of resolution 2^{-j}) by 2 amplifies the details by 2 in V_j while providing enough detail for an approximation in V_{j-1} .

Recall that $V_{j-1} = V_j \oplus W_j$, then

$$P_{V_{j-1}} = P_{V_j} + P_{W_j}, \quad (3.41)$$

where the orthogonal projection of the function onto W_j is

$$P_{W_j} = P_{V_{j-1}} - P_{V_j} \Rightarrow D_{W_j} = P_{V_{j-1}} - P_{V_j}, \quad (3.42)$$

and P_{W_j} is denoted as D_{W_j} . Hereafter, P_{V_j} is denoted by P_j and D_{W_j} is replaced with D_j for simplicity of notation. Equation (3.41) simply states that the orthogonal projection of a set (signal) on V_{j-1} is just the sum of orthogonal projections of the function onto its subspaces V_j and W_j . It is also true that the orthogonal projection of a function (or vector), f onto an orthonormal basis of an N dimensional subspace V is defined as

$$P_V(f) = \sum_{i=1}^N \langle f, v_i \rangle v_i, \quad (3.43)$$

where $\{v_1, v_2, \dots, v_N\}$ is the orthonormal basis of V . Recalling that the orthonormal basis of the *detail space* W_j

is defined by $\psi_{k,j}$; resulting in the orthogonal projection on W_j , D_j as

$$D_j f = \sum_{k \in \mathbb{Z}} \langle f, \psi_{k,j} \rangle \psi_{k,j}, \quad (3.44)$$

which is the *partial* expansion of signal into the detail space. Hence, the signal can be approximated as an expansion of the wavelet coefficients, $d_j[k] = \langle f, \psi_{k,j} \rangle$, at all scales as

$$f = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_j[k] \psi_{k,j}. \quad (3.45)$$

Let us now re-write (3.41), in terms of the signal, and with simplified notations as

$$P_{j-1}f = P_j f + D_j f, \quad (3.46)$$

which illustrates the *decomposition* of the signal into a *coarse* approximation (P_j) and a *detailed* representation (D_j). Henceforth, $P_j f$ is denoted as f_j while $D_j f$ is described by d_j . Once again, by referring to the scaling and nesting properties of the MRA, it can be understood that the detail space of f_{j-1} contains signal information that are not present in f_j ¹.

Now that the tools required to express a signal (digital signal and images) in a wavelet basis, it is time to see how this process can be implemented in an efficient manner. Let us begin by recalling (3.26) to represent $\psi_{k,j}$; which forms an orthonormal basis for W_j . Also recall (3.39) for ψ as a general basis (non-unique with respect to the translation parameter); while performing a change of variable whereby $k \Rightarrow n$. Combining these two equations yields:

$$\begin{aligned} \psi(x)_{k,j} &= 2^{-\frac{j}{2}} \sum_{n \in \mathbb{Z}} g_n \phi_{n-1}(2^{-j}x - k) \\ &= 2^{-\frac{j}{2}} \sum_{n \in \mathbb{Z}} g_n 2^{\frac{1}{2}} \phi(2^{-(j-1)}x - 2k - n) \\ &= \sum_{n \in \mathbb{Z}} g_n 2^{-\frac{(j-1)}{2}} \phi(2^{-(j-1)}x - (2k + n)) \\ &= \sum_{n \in \mathbb{Z}} g_n \phi_{j-1, 2k+n}(x) \\ \psi(x)_{k,j} &= \sum_{n \in \mathbb{Z}} g_{n-2k} \phi_{j-1,n}(n), \end{aligned} \quad (3.47)$$

where, between the last two lines a change of variable takes place such that $n \Rightarrow (n - 2k)$. Hence, the relationship in (3.44) can be stated in terms of the filter g_n as

$$\langle f, \psi_{k,j} \rangle = \sum_{n \in \mathbb{Z}} \bar{g}_{n-2k} \langle f, \phi_{n,j-1} \rangle. \quad (3.48)$$

By virtue of the same analysis the low resolution part of the function can be decomposed using the following relationship:

$$\langle f, \phi_{k,j} \rangle = \sum_{n \in \mathbb{Z}} \bar{h}_{n-2k} \langle f, \phi_{n,j-1} \rangle, \quad (3.49)$$

¹Information lost, scaling from a coarser resolution 2^{-j-1} to a higher resolution in 2^{-j}

where

$$\phi_{j,k} = \sum_n h_{n-2k} \phi_{n,j-1}(x). \quad (3.50)$$

The two operations in (3.48) and (3.49) emulate the convolution operation of a filter where $\bar{h}$ and $\bar{g}$ are the filters;

Figure 3.5: (a) Analysis step of a signal $x[n]$; convolved with a high-pass filter $\bar{g}$ (retaining high frequency signal components) as well as a low pass filter $\bar{h}$ (decomposing low frequency signal components). After the filtering is performed the outputs are down-samples by a factor of 2; which results in retaining only the even entries (in the sequence). (b) Reconstruction step of the signal taking in the approximation and detail signals a and d , respectively, up-sampling them by a factor of two (see 3.54), and then filtering them with g and h ; summing the output of the filters in a piecewise manner results in the reconstruction of the original signal, $x[n]$.

this process is followed by a dilation of a factor of two. Previously, $\langle f, \psi_{k,j} \rangle$ was denoted by $d_j[k]$ (*detailsignal*); similarly, $\langle f, \phi_{k,j} \rangle$ shall be defined by $a_j[k]$ (*average signal*); where the first sampling of function, f (relative to V_0) is a_0 , and everything else is derived from a_0 using (3.48) and (3.49). In keeping with the notations allocated to $a_j[k]$ and $d_j[k]$, a general sequence x_n has the following properties:

$$x[n] = x_n, \quad (3.51)$$

$$\bar{x}[n] = x[-n] = x_{-n}, \quad (3.52)$$

$$y[n] = x[2n] = x_{2n}, \quad (3.53)$$

$$\tilde{x}[n] = x\left[\frac{n}{2}\right], \quad (3.54)$$

$$y * x[n] = \sum_{k \in \mathbb{Z}} y[n - k]x[k], \quad (3.55)$$

where equations (3.53) and (3.54) denote down-sampling and up-sampling procedures, and $\tilde{x}[n] = 0$ for $x[\frac{n-1}{2}]$ in (3.54). The statement in (3.55) defines a discrete convolution process. Utilizing these notations (3.49) and (3.48) become

$$a_j[k] = \sum_{n \in \mathbb{Z}} \bar{h}[n - 2k]a_{j-1}[k], \quad (3.56)$$

and

$$d_j[k] = \sum_{n \in \mathbb{Z}} \bar{g}[n - 2k]d_{j-1}[k], \quad (3.57)$$

respectively. These two equations decompose a signal, without any loss, into the average and detail signals; the process can continue recursively into smaller subspaces, as mentioned beforehand. The original signal can be reconstructed via a recombination step where

$$a_{j-1}[k] = \sum_n h[k - 2n]a_j[n] + \sum_n g[k - 2n]d_j[n]. \quad (3.58)$$

Together h , g , $\bar{h}$, and $\bar{g}$ are called *Quadrature Mirror Filters* (QMF); which are symmetric about a certain frequency. The pair of equations in (3.56) and (3.57) are part of a *analysis* step, while (3.58) is involved during a *synthesis* step. This form of decomposition and recombination (Fig. 3.5) in the format of a *pyramidal* filter bank was first introduced by Mallat [43], and is what generally is referred to as the DWT. The algorithm can also be extended to a case of *bi-orthogonal* wavelet bases [63]. Bi-orthogonal wavelet bases add more redundancy and degrees of freedom. An orthogonal filter bank uses a set of QMF filters for decomposition and reconstruction purposes; where the filters are symmetric. Doing so has a disadvantage; the output of the synthesis step is not the same as the input signal. In fact if the output of reconstruction is denoted by $\tilde{x}[n]$, then $x[n] \neq \tilde{x}[n]$. It has been shown [47], [59], that perfect reconstruction can be achieved by using dual band low and high pass, finite impulse response (FIR), filters, $\check{h}$ and $\check{g}$ (satisfying certain conditions) called *Conjugate Mirror Filters* (CMF), in the reconstruction process; whereby $\tilde{x}[n] = x[n]$. In general, bi-orthogonal wavelets can be constructed using filters that could be symmetric; they are inverted but not necessarily orthogonal. The discussion of bi-orthogonal wavelet bases is not in the scope of this text and will not be discussed any further. It suffices to state that bi-orthogonal wavelet bases are more versatile in terms of filter design, and can also be used to attain perfect signal synthesis.

Seperable MRA

Thus far, the discussions provided have revolved around signals in the $\mathcal{L}^2(\mathbb{R})$ which can be represented as one dimensional sequences. The challenge, with respect to the objective of this thesis, is to find out how two dimensional signals in $\mathcal{L}^2(\mathbb{R}^2)$ space can be stated in terms of their wavelet coefficients. Similar to MRA analysis for a one dimensional signal, one needs to create an MRA for two dimensional signals.

If a signal can be represented by an orthonormal basis $\psi_{k,j}[x]$ with $k, j \in \mathbb{Z}^2$ for $\mathcal{L}^2(\mathbb{R})$, a straightforward

Figure 3.6: Coefficient pyramid for a 2D wavelet analysis by way of separable bases; detail images are denoted by d and the average image is denoted by a . there are 3 octaves (levels) in total with $j - 1$ as the first level. Each level contains 2^{2k} coefficients; where k is the number of the level.

approach is to represent the orthonormal basis for $\mathcal{L}^2(\mathbb{R}^2)$ with $\psi_{k,j}[x], \psi_{l,m}[y]$ where $j, k, l, m \in \mathbb{Z}^4$. The problem with this process is that it has no sense of directionality with respect to the basis functions, that is to say that the information about x and y are mixed and not separated. To overcome this it can be shown[44],[23] that an MRA can be constructed for $\mathcal{L}^2(\mathbb{R}^2)$.

The orthogonal complement of V_j, W_j can be divided into three subspaces such that:

$$W_j = W_j^h \oplus W_j^v \oplus W_j^d, \quad (3.59)$$

where h, v , and d stand for *horizontal, vertical, and diagonal*, respectively. Analogous to equations (3.57) and (3.56), high-pass (g) and low-pass (h) filters can be described and utilized to calculate wavelet coefficients a_j, d_j^h, d_j^v , and d_j^d , as shown in Figure 3.6. It should be noted from the illustration that each consecutive level contains fewer coefficients. The application of DWT takes place in one direction first and then in the other (see Fig. 3.7). The synthesis step of two dimensional DWT in Figure 3.7 is similar to that shown in Figure 3.5(b); with similar fast algorithm procedure.

The final step in performing wavelet analysis is to design a suitable wavelet for the task at hand. However, the process of designing a wavelet is beyond the confines of this work and it shall not be discussed here (this thesis uses already available wavelets). Nonetheless, an inspection of criteria that a wavelet must meet in order to be chosen for a certain application (i.e. edge detection and sparsity in the case of this work). Most of these applications rely on the wavelet transform to be able to represent a signal with only a few wavelet coefficients. This mostly depends on three properties of a wavelet: its regularity, the number of its vanishing moments, and its support. A concise survey of such criteria are provided hereafter.

Figure 3.7: Coefficient pyramid for a 2D wavelet analysis by way of separable bases; detail images are denoted by d and the average image is denoted by a . there are 3 octaves (levels) in total with $j - 1$ as the first level. Each level contains 2^{2k} coefficients; where k is the number of the level.

Regularity

The regularity of a wavelet is a measure of how well it reacts to singularities in a signal. Singularities could be localized or distributed and non-isolated in irregular signals. Wavelets (as compared to Fourier analysis) are able to compute the distribution of such irregularities and detect them. The regularity of the wavelet itself can alter how the coefficients are depicted (in a two dimensional wavelet coefficient pyramid, for instance); in general the more regular the wavelet the better. No more discussion on regularity is provided here, for more in depth analysis the reader is encouraged to consult [50].

Vanishing moments

A wavelet ψ is said to have p vanishing moments if

$$\int_{-\infty}^{+\infty} t^k \psi(t) dt = 0, \quad (3.60)$$

for $0 \leq k < p$. The relationship in (3.60) means that ψ is orthogonal to any polynomial signal of degree $p - 1$. If the signal under examination is a $p - 1$ degree polynomial and is regular enough, then the wavelet coefficients are small at higher scales. This means that, if the wavelet coefficients are stated in terms of z-transforms, then there are p zeros at π (in the unit circle). In genera, the higher the number of vanishing moments, the better the wavelet is at approximating signals more accurately (and more complex signals). Hence, the desirable condition is to have as many vanishing moments as possible. However, the number of vanishing moments in a wavelet also affect its *support*.

Compact support

If a wavelet basis is infinite, it poses inconvenience in representing infinite signals; in which case the wavelet has to be adjusted in order to be able to express the signal¹

A compact support wavelet suggests that values outside the wavelet support are zero. The scaling function of an MRA set has compact support if and only if its corresponding low-pass filter h has compact support. If the compact support of h lies on interval $[a, b]$, then the compact support of the scaling function ϕ is also $[a, b]$, while the support of the wavelet function ψ is $[\frac{a-b+1}{2}, \frac{b-a+1}{2}]$. The support of wavelets has a direct relation with the number of vanishing moments. The scaling function of a wavelet basis with p vanishing moments has a support that is greater than or equal to $2p - 1$ on an interval $[0, n \geq 2p - 1]$; where n is the outer bound support. One can see that as the number of vanishing moments increase, so does the support of a wavelet (as a whole).

If the scaling function support is exactly $[0, 2p - 1]$, then its wavelet has a support of $[-p + 1, p]$; this family of wavelets are called the *Daubechies* family. The Daubechies family has a minimum support size for a given number of vanishing moments compared to other families of wavelets; which is an ideal case (highest number of vanishing moments possible with minimal support). The Daubechies family and other families of wavelets will be explored in more detail in the following chapters.

Hence, considering all three conditions, one can see that designing a wavelet family requires trade-offs; however, the trade-offs are application dependent, as different wavelet families are suitable for different applications.

3.5 Summary

The goal of this chapter was provide fundamental information on wavelets, necessary in understanding the role of wavelets in this work. It started by introducing continuous mother wavelet and scaling functions and their relationship to DWT. A brief description of MRA sets, with conditions central to creation of wavelet families, QMF and CMF filters with the purpose of decomposition and reconstruction of signals was given. After the groundwork was established, the concept was expanded to cover two dimensional signals, such as images, as images are the topic of interest in this research. Further, some conditions for choosing the suitable wavelet, for a given application, was provided. Decomposition via filtering in wavelets (pyramidal decomposition) is central to edge detection and sparsity promotion; both of which are essential for successful results in compressed sensing edge detection. Inspection of wavelet edge detection methods will further help grasp the role of wavelets in singularity analysis.

¹A number of solutions are available to deal with such wavelets, such as constructing periodic wavelets; along with other methods. However, all these methods create difficulties that are specially not desired in fast algorithms.

Chapter 4

Image Edge Detection

4.1 Overview

4.2 Edge Detection

4.2.1 Introduction to Edges

In an image, edges could be defined as a contour where sudden changes in the intensity take place. These changes correspond to discontinuities in a function which can be identified as singular values; where the first derivative of the function approaches infinity. For images and other discrete signals, these singular values are finite and often have large values. Therefore, these abrupt changes in intensity are located by finding the local maxima of the gradient of the function.

The human eye is capable of recognizing the boundaries of an image without much trouble, and accurately process edge information that are used in interacting with objects. However, with and increasing interest in automation, it is required for machines to be able to perform tasks that humans take on, many of which require *seeing* objects and the ability to distinguish between different objects. Therefore, edge detection becomes a significant undertaking in the world of image processing.

Image processing operations frequently involve image segmentation, image registration, pattern recognition, scene analysis, and image visualization. Whatever the case may be, all these processes have a common first step, an initial piece of information to conduct any of these (or some other processes not mentioned here) operations: they all require the image edge information. Consequently, the existence of an accurate edge detection technique is crucial; making it one of the most widely studied and researched topics in the field of image processing; which covers a vast range of applications from machine vision to medical image processing (and post-processing).

Edges in an image are sudden changes in intensity that are spread out over just a few pixels. Let us assume that the distance between two pixels is Δx (spatial domain); with the corresponding (spatial) frequency for the two pixels written as $\frac{1}{\Delta x}$. Then, it is clear that the larger the distance between two pixels, the lower the frequency, and the smaller the distance, the larger the frequency. Therefore, edges in an image can be considered to be high frequency features; which can be selected using a high pass filter. The problem of image edge detection,

nevertheless, can be stated as a high pass filtering problem.

Since edge detection operators utilize some manner of high pass filtering, they are highly susceptible to noise. Noise in an image are mostly high frequency components and can not be surpassed during the filtering process. In order to reduce noise and be able to distinguish between noise and edges in an image, many operators employ techniques that manipulate the noisy image. One disadvantage of such techniques is the fact that by manipulating the noisy image, the quality of the final edge information is often degraded; hence the trade-off between noise cancellation and edge detection quality is apparent in these techniques. Therefore, the development of a noise resilient, robust edge detection method capable of producing high quality, well detailed images is highly desired, and an active field of research.

4.2.2 Conventional Edge Detection Methods

The majority of current edge detection techniques can be divided into two main categories: conventional techniques, and wavelet based techniques. Conventional techniques use the intensity plot of an image in conjunction with a gradient (first partial derivative), Laplacian (second partial derivative), Gaussian, or a zero crossing operator to filter high frequencies in an image. These methods lack the ability to surpass impulsive noise, and are incapable of recognizing slow varying intensities. Wavelet methods [45], [58], [9], [62], [2], on the other hand, utilize the natural behaviour of wavelets to divide a signal into different frequency bands: low frequency (approximation), and high frequency (in 3 different orientations). In addition, wavelets are highly localized and well capable of recognizing discontinuities in a signal, and respond better to impulsive noise components in a signal. Conventional edge detectors can further be divided [57] into different categories. A brief explanation of some of the most widely used of these techniques follows.

Sobel Edge Detector

Sobel edge detector is a gradient edge detection operator that employs the first directional derivative of the image, and recognize edges as local maxima. The sobel operator convolves (similar to other gradient methods) a directional derivative mask with the image to classify edges. Two gradient masks are used for the vertical (y-direction) and horizontal (x-direction), named f_x and f_y , respectively. These gradients can be applied either separately, to find the gradients in each direction, or combined to find the absolute value of the gradients as

$$|\nabla f| = \sqrt{f_x^2 + f_y^2}, \quad (4.1)$$

with the orientation of the edge calculated as

$$\theta = \arctan\left(\frac{f_y}{f_x}\right). \quad (4.2)$$

The sobel operator uses 3×3 gradient masks¹; which are predefined. Other gradient edge detector operators include Perwitt's operator (a simpler mask compared to Sobel), and Robert's cross operator. The main difference between the Robert's operator and the other two methods mentioned, is the fact that Robert's technique utilizes 2×2

¹These masks are discrete approximations of the gradients into the appropriate dimension

masks whose gradient responds to edges at a 45 degree angle rather than the conventional x and y directions. The orientation angle in (4.2) is rotated by 135 degrees (or $\frac{3\pi}{4}$) to adjust for the image orientation.

In general, gradient edge detectors are mathematically very simple, and are able to classify edges based on strength and orientation. However, these algorithms tend to be computationally costly since the image has to be convolved with two masks in order to compute the edges orientations. In addition, these techniques are highly inaccurate in detection of noisy images. Another disadvantage of these methods is the fact that their operators (gradient masks) are fixed and lack flexibility; which sometimes results in very inaccurate edge information.

Laplacian Edge Detector

In order to improve noise sensitivity of gradient edge detection, and perform more accurate extraction, Marr and Hildreth [46] proposed a method based on image filtering (for noise reduction) before the process of edge detection. This is the most common approach to noise reduction (in conventional edge detectors); using either a filter or a smoothing function on the image before processing it. This method is called the Laplacian of Gaussian (LoG).

The Laplacian operator is a second order partial differential operator; such operators detect any zero crossings in the second order signal of the image¹. The Laplacian of an image (intensity values) $I(x, y)$ is

$$\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}. \quad (4.3)$$

The Laplacian filter, similar to gradient edge detectors, is approximated by 3×3 masks that are convolved with the image. However, before the Laplacian is applied to the image, for noise reduction purposes, the image being processed is filtered by a Gaussian smoothing filter. A typical two dimensional Gaussian function has the form

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left\{\frac{x^2+y^2}{2\sigma^2}\right\}}, \quad (4.4)$$

where σ is the standard deviation of the Gaussian distribution. The smoothing Gaussian filter produces a somewhat blurred version of the original image which is less sensitive to noise. This is especially significant in the case of Laplacian operators as the second derivatives are most responsive to noise elements in the image. The Laplacian and Gaussian operators can first be combined together (convolution associativity rule); called the Laplacian of Gaussian (LoG) operator, and the resulting equation can then be applied to the image. Doing so will also reduce the computation time of the edges. The LoG operator is derived by differentiation the relationship in (4.4) twice such that [49]

$$LoG = \frac{1}{\sigma^2} \left(\frac{(x^2 + y^2)}{\sigma^2} - 2 \right) e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \quad (4.5)$$

The relationship in (4.5) can be utilized to produce an edge detection template that is used on the image of interest. The frequency response of the LoG operator can be shown [49] to be similar to a band pass filter, with σ controlling the bandwidth of the filter.

¹Second order differentiation produces zeros at local maxima of the signal (peak of first order derivative) since the rate of change at the peak is constant

One advantage of the LoG operator is the fact that it is more flexible; the bandwidth of the filter can be controlled via σ , which can be changed by the user. In addition, the LoG operator also detects edges based on scale; that is to say that if a scale is employed by the template smaller regions are detected (more detail), while a larger scale means that the smaller regions can be combined into a single feature (less detail). This technique is also less arithmetically complex, in part due to the fact that the operator is *isotropic* (performs the same in both directions; non-directional); which can also be one of the drawbacks of the LoG scheme. Since the operator is not able to detect the orientation of the edges, it could have difficulties detecting certain curved edges.

Other second order derivative operators and zero crossing techniques also exist which are not covered in this work. These techniques (with the exception of Marr-Hildreth LoG operator) are generally very sensitive to noise. For more information on such approaches the user is encouraged to consults relative material [7].

Canny Edge Detection

Another edge detection approach that utilizes the Gaussian operator (4.4), is the widely used Canny edge detector, named after John Canny (developed in 1986) [16]; its developer. Canny edge detection is a combination of different processes performed on the image in a sequential order, starting with the Gaussian filtering. Before any explanation is given on how the Canny operator executes, a few requirements established by John Canny need to be highlighted. Canny imposed three conditions, which would create an *optimal* edge detector, if ratified. These three benchmarks are:

1. optimal edge detection, identifying as many edges of the image as possible; with no false edges detected;
2. acceptable localization of the edges with the goal that the marked edges are as close as possible to the real edges in the image;
3. a single response per edge, and elimination of multiple responses.

In order to meet these conditions, the Canny edge detection is performed in multiple stages; where each stage satisfies one condition. The first requirement above has to do with reducing noise sensitivity: an optimal edge detector detects only true edges and aims to suppress noise elements. In Canny edge detectors, noise reduction is achieved with the application of a Gaussian smoothing filter. According to Canny, and his vision of optimal edge detection, a Gaussian filtering process is optimal for images. Such filters are in the form of pre-defined discrete representations of the two dimensional (isotropic) function in (4.4); whose distribution is governed by the standard variation. In general, an increase in the standard deviation creates more blurring and decreases noise sensitivity at the mercy of fewer detected edges (fewer true edges), and lower edge image quality. After smoothing is performed, edges are detected using a gradient operator; Canny edge detector uses four different operators to detect in four different direction. Up until this point the process of edge detection is similar to other techniques indicated before.

The second condition enforces detection accuracy by attempting to select the edges in their true position, such that the difference between the edge image and the edges in the original image is minimal. This is achieved through a process called *non-maximum suppression*. Let us assume that the intensity of an edge can be modeled by a bell shaped continuous graph; with intensity increasing gradually to a *maximum* intensity and then decreasing

gently. The non-maximum suppression technique keeps the point with the highest intensity on top of the graph while suppressing the other points. This process results in the output edge image to consist of thin edges; which enhances the accuracy of the output.

In addition to improving precision, non-maximum suppression also assists with the achievement of results that satisfies the third constraint. When gradient edge detection (second stage) is performed, due to the spread of intensity and change of brightness on the outline of an edge, often more than one edge might be detected per one true edge. This goal is accomplished by thresholding by *hysteresis*. The process of thresholding involves choosing two thresholds: high and low. One applies the high threshold to ensure that the intensity values selected include edges. By moving along the gradient directions; and assuming that the edges are along a continuous path, one can arrive at lower intensity edges while that qualify as lower threshold boundaries. The result is a discrete representation of the continuous edge map provided by gradient operators; which localizes edges.

Canny edge detection, typically, performs better than any other conventional techniques; whereby many other methods are compared with the Canny method to assess their performance. However, it is more computationally complex than a simple gradient technique of the LoG process, and takes more time to process.

4.3 Wavelet Edge Detection Methods

Conventional edge detection methods utilize either gradients of Laplacians (and both in case of LoG). For noise reduction, these methods rely on smoothing filters that are applied to the image prior to processing. Conventional edge detection methods are often computationally expensive and inflexible; because they are often estimated with fixed gradient and Laplacian matrices. It would be logical question to ask if there is a method that is faster and more flexible.

Since their introduction into signal processing, wavelets have increasingly been used in a variety of applications, such as signal and image compression (JPEG 2000). Their ability to represent a signal with a few coefficients as well as their capability to provide a signal representations in terms of scale and frequency, is part of their success. With respect to image processing, one attribute of wavelets is intriguing; multiresolution analysis allows wavelets to divide an image into its low and high frequency components.

Hence, wavelets could be utilized as functions that provide both high frequency and low frequency filtering of an image at the same time; though the scale of the output images are lower (Fig. 3.6). Since high frequency components of a signal characterize its edges (along with other sudden variations such as noise), it appears befitting to use wavelet analysis for edge detection.

In 1992, Mallat and Zhong [58] provided a framework that resembles Canny edge detection. In their work, Mallat and Zhong characterized edges by their *Lipschitz* regularity. A function is said to be regularly Lipschitz α over interval $[a, b]$, if and only if

$$|f(x) - f(x_0)| \leq K|x - x_0|^\alpha, \quad (4.6)$$

where x and x_0 are in the selected interval, and K is a constant indicating the slope of the function (on a graph). The Lipschitz criteria can be expanded to two dimensions, for images, whereby

$$|f(x, y) - f(x_0, y_0)| \leq K(|x - x_0|^2 + |y - y_0|^2)^{\frac{\alpha}{2}}. \quad (4.7)$$

The regularity criteria can be related to a wavelet transform. Let us now denote a smoothing function that is the scaling function of a wavelet as $\theta(x)$, where

$$\theta_{2^j}(x) = 2^{-j}\theta(2^{-j}x), \quad (4.8)$$

and its conjugate is

$$\bar{\theta}_{2^j}(x) = \theta_{2^j}(-x). \quad (4.9)$$

A mother wavelet can be described as the partial derivative of its scaling function such that

$$\begin{aligned} \psi(x_1) = \psi^1 &= -\frac{\partial\theta}{\partial x_1}, \\ \psi x_2 = \psi^2 &= -\frac{\partial\theta}{\partial x_2}, \end{aligned} \quad (4.10)$$

where $x = (x_1, x_2)$. Here, the scale changes with a dyadic wavelet of scale $2^j | j \in \mathbb{Z}$, then

$$\begin{aligned} \psi_{2^j}^1(x_1) = \psi^1 &= \frac{1}{2^j}\psi^1\left(\frac{x_1}{2^j}\right), \\ \psi_{2^j}^2 x_2 = \psi^2 &= \frac{1}{2^j}\psi^1\left(\frac{x_2}{2^j}\right), \end{aligned} \quad (4.11)$$

and

$$\bar{\psi}_{2^j}^{1,2}(x) = \psi_{2^j}^{1,2}(-x). \quad (4.12)$$

Then, the dyadic wavelet transform of a point $u = (u_1, u_2)$ is

$$W^{1,2}f(u, 2^j) = f * \bar{\psi}_{2^j}^{1,2}(u). \quad (4.13)$$

One can then scale the wavelets (smooth the wavelet function) by $\bar{\theta}(x)$ to form the wavelet family

$$\begin{aligned} \bar{\psi}_{2^j}^1 &= 2^j \frac{\partial \theta_{2^j}}{\partial x_1}, \\ \bar{\psi}_{2^j}^2 &= 2^j \frac{\partial \theta_{2^j}}{\partial x_2}, \end{aligned} \quad (4.14)$$

which can be combined with equation (4.13) to show that wavelet transform components can be stated in terms of gradient vectors of image f smoothed by smoothing function $\bar{\theta}_{2^j}$:

$$2^j \nabla (f * \bar{\theta}_{2^j})(u) = \begin{pmatrix} W^1 f(u, 2^j) \\ W^2 f(u, 2^j) \end{pmatrix}, \quad (4.15)$$

where $W_{2^j}^n f(u)$ is analogous to $f(x_m)$ in (4.6). The magnitude of the gradient vector (that will be compared to a threshold to determine whether a coefficient is an edge or not) is determined by

$$|f(u, 2^j)|^2 = \left(|W^1 f(u, 2^j)|^2 + |W^2 f(u, 2^j)|^2 \right)^{\frac{1}{2}}. \quad (4.16)$$

Then angle of the edge detected can also be calculated as

$$\varphi(u) = \arctan\left(\frac{W^2 f(u, 2^j)}{W^2 f(u, 2^j)}\right). \quad (4.17)$$

The equation in (4.16) referred to as modulus (magnitude) of the gradients; where a point v is a local maxima, where $u = v$. The local maxima of a wavelet are together called the wavelet *modulus maxima*, where W^1 find the modulus maxima in the horizontal direction and W^2 predicts where the vertical modulus maxima are located. The combination edge image can be produce using (4.16) and (4.17). The process of using a wavelet in this manner is called *multiscale edge detection*.

In multiscale edge detection the choice of the wavelet and the scaling function adjusts the gradient function. Some works of research [37] attempt to model gradient¹ functions of conventional edge detectors using wavelets; while others [62] attempt to model different smoothing filters. IN both of these methods, the approach generally follows the multiscale edge detection method introduced by Mallat in (4.16).

Brannock and Weeks[9] suggest a different view on wavelet edge detection methods. Their scheme uses two dimensional wavelet transforms to filter an image into low and high pass components (see Fig. 3.7). After one octave is processed, the approximation image is taken and the two dimensional DWT applied to the approximation image. The recursive process is continued for a number levels, or octaves (6 in the experiment), until a desired resolution is achieved. Then the three detail images; HL, LH, and HH, are combined together resulting in one image of the feature outlines of the original picture. After the application of 2D-DWT, pixels are altered, in that they assigned a value base on their standard deviation from the mean of overall pixel values.

The method explained above is objectively different from multiscale edge detection, in that it does not apply a gradient or attempt to emulate other conventional edge detectors; rather, this technique applies the 2D-DWT directly to the image and filters the image, recursively, into low and high pass sub-images, where edges are filtered onto the high frequency sub images. This technique, like the ones mentioned before, does not provide any type of *active* de-noising method, and is sensitive to noise (although noise sensitivity could be application dependent).

Understanding the use of wavelets to filter images is fundamental to the proposed method of edge detection here. Before providing any explanation this technique; however, one should explore the significance of brain MRI edge detection.

Figure 4.1 summarizes the result of different edge detection methods; from top to bottom: (a) a normal phantom image, (b) a phantom contaminated with Gaussian noise, (c) phantom with *salt* and *pepper* noise, and (d) phantom with speckle noise; from the second row to the last row, images demonstrate edge images calculated using Sobel, Canny, LoG, and Multiescale edge detection, respectively (from left to right for normal, Gaussian noise, salt and pepper noise, and speckle noise, respectively).

¹By discretizing continuous wavelet transform gradients.

Figure 4.1: Different edge detection techniques.

4.4 Brain MRI Edge Detection

As it was previously stated, image edge detection is essential to image processing; it provides crucial information that are used to gather vital information from an image, by way of other processes. Along with the development of many medical imaging tools, image processing has become an integral part of diagnostic imaging. Whether the goal is to provide information for image segmentation (providing candidate images) [52] or for post-processing to extract more information, edge detection plays an important role in medical image processing.

In medical imaging, and MRI in particular, recognition of pathological disorders is essential to a successful diagnosis. Unfortunately, MRI images suffer from variety of artifacts that can disturb the image and make feature recognition difficult. This problem is magnified in brain MRI images due to the fact that brain images contain many different types of biological tissues each possessing its own contrast properties. Hence, the problem of noise and artifact defection proposes considerable challenge in the area of image feature extraction and edge detection. In addition, with the expanding knowledge of brain anatomy, much effort is allocated to connecting mental disorders with brain abnormalities. This, along with artifact reduction, is inspired by the prospect of implementing an automatic feature detection technique; that is able to distinguish different tissue types as well as different structures (such as tumor or legions).

A number of techniques have been developed [3], [35], [27], [67], [1] to provide brain MRI edge detection and/or segmentation. Some of there techniques rely on conventional edge detectors [3], [1], [27]; while attempting to improve upon conventional algorithms to promote robustness. Other procedures [67], [35] employ wavelet transforms. In [67] the author attempts to incorporate *stationary wavelet transform* operators; which limit the translation invariance property of regular DWT, and is shown to perform better.

Another work that provides good noise reduction results is [1]; where an alternative, modified Gaussian filter is approximation is created instead of the one used in conventional Canny operators. Then two edge images are created, one with the modified kernel and the other with the conventional Canny kernel. By fusing the two images produced together, the resulting output will have all the edges in the original input, and nothing would be missed (any missing components through one kernel would be picked up by the other kernel). The process results in more edges being detected correctly and with less noise.

Regardless of the approach and the outcome of these methods, they all have one thing in common: they all acquire samples at full density (Nyquist rate) and process the image first, and perform edge detection only after on the brain image.

4.5 Summary

In keeping with the goal of devising a robust edge detection method for brain MRI images, this chapter provided some fundamental issues surrounding image edge detection in general. It started by defining edges in an image and how they can be mathematically explained. After the introduction of edges, a few classical edge detection techniques based on gradient and/or Laplacian operators were presented; considering their respective strengths as well as shortcomings. The discussion was extended to wavelet dependent edge detection methods, namely multi-scale edge detection; which models gradient operators using wavelets and scaling functions, and other methods that rely on multi-resolution filtering of images using wavelets. The latter plays a central role to the proposed

method in this paper.

Chapter 5

Methodology

5.1 Overview

This chapter proposes a novel method intended for brain MRI image edge detection with minimal noise sensitivity. The presentation of the technique combines artifact/noise cancellation property of compressed sensing with wavelets as an sparsifying transform while taking advantage of their ability to filter images into different frequency bands (namely, low and high frequency). The development of process, from MRI image acquisition to implementation to output edge image, is discussed in detail; and preliminary results are provided.

5.2 Design Requirements

In general, image edge detection revolves around one central idea: detect as many *true* edges as possible, with highest accuracy possible, while suppressing as much of the image noise as possible. These points are vital for any edge information to be of any practical use. Brain MRI images are prone to noise and artifacts due to magnetic field inhomogeneity and movement, along with other imperfections cause either by the MRI device itself or the patient. In addition, brain images have many different features and tissue types that make developing a robust edge detector all the more challenging, and essential at the same time. By recognizing this fact, and keeping in mind the challenge at hand, a brain MRI edge detection technique is developed here and explained in the following sections. The proposed method is built upon three principle pillars:

1. The first step in performing an edge detection of any image is to acquire the image. In MRI, as explained in Chapter 2, the image is constructed from Fourier encoded samples. It was shown in Section 2.3.2 that incoherent sampling, and pseudo-random sampling trajectories enable the construction process to reject unwanted noise and artifacts when combined with an iterative thresholding process. However, it is not enough to acquire samples in an incoherent manner (while also under-sampling Fourier plane), the proper construction technique is required. In fact, pseudo-random sampling is closely tied to the compressed sensing reconstruction step described next.

2. Compressed sensing (Section 2.3) provides the groundwork for a signal sampling and reconstruction cycle that allows for samples to be collected at much lower rates compared to the Nyquist criteria. Reconstruction algorithms for an under-sampled image work by promoting sparsity; while incoherent, non-uniform under-sampling provides artifact suppression. This is an inherent characteristic of CS acquisition, that will be exploited in designing an edge detector. In addition, CS MRI reduces image acquisition time and decreasing the number of samples required to rebuild an image.
3. On route to reconstructing an under-sampled image, CS utilizes a convex optimization program whose object (objective or cost function) is to promote sparsity. This task is done by first taking the image through a sparsifying transform that is sparsest domain for the image. Then, ℓ_1 minimization is performed with the goal of finding the basis of the solution space, while checking to see if the basis is consistent with the image data (data consistency). The wavelet transform is known to be a sparsifying transform for images, as it can represent an image with a few large coefficients (a main motivation behind the development of CS) in comparison with other transforms such as the Fourier transform. By taking advantage of this fact, wavelet transform is utilized in the proposed method whilst serving as an edge detection filter as well.

By combining these techniques a novel brain MRI edge detection method is presented hereafter. A discussion involving the development of a reconstruction algorithm as well as design and implementation of suitable wavelets will be presented; as well as explanations regarding the choices made for different variables. The presented method employs only Cartesian sampling trajectories (non-uniform density), and ℓ_1 norm minimization for reconstruction; the choice of the sparsifying transform will be clarified. Although the implementation developed with the rules of universality in mind, any assumptions (if existing) will also be stated and explained.

5.3 Image Acquisition

5.3.1 Sampling Trajectory

The process of MRI image acquisition involves acquiring a number of Fourier encoded coefficients (k-space), by manipulating directional gradient magnetic fields. These gradient fields are controlled by a series of RF pulses designed and inputted into a computer that controls the motion of the magnets. For the sake of practicality and speed, these pulses generally create smoothly changing trajectories with no jumps or sudden changes. A number of different trajectories have been designed and implemented over the years that are suitable for different applications. These trajectories can be divided into two broad categories: Cartesian and non-Cartesian.

Cartesian trajectories are concerned with selection of samples on a straight path from one side of the k-space to the other. Cartesian sampling is by far the simplest technique to implement; as it involves the manipulation of only one gradient pulse (most of the time). Cartesian sampling trajectory is also one of the most robust schemes; hence becoming one of the most widely (clinically) implemented trajectories. Non-Cartesian schemes, on the other hand, are highly sensitive to system imperfections and are more complicated to implement, with more complex pulse arrangements. However, these trajectories are not without their share of advantages; they are often faster in terms of sample collections and some, such as spiral sampling, are better suited for dynamic MRI imaging.

As the goal here to provide a practical, clinically applicable, tool for brain image edge detection; and since brain imaging (unlike dynamic imaging) does not require accelerated sampling, the proposed method is implemented using Cartesian scheme. In addition, utilizing Cartesian sampling trajectory in real life has the added benefit lower artifact sensitivity.

5.3.2 Non-uniform Sampling

In keeping with the notion of incoherence; and its implicit affinity for imperfection suppression in a CS reconstruction platform, possible sampling trajectories need to be investigated. By looking at Figure 2.10 in subsection 2.4.2 it is easy to see the importance of random sampling as a backbone of CS: random sampling adds the capability for better recognition of major signal components, while diminishing the effect of noise. However, randomly selecting samples is not a practical endeavor in MRI image acquisition. Hence, a solution needs to be implemented to provide the benefits of random sampling, while considering a few other factors.

In Cartesian sampling, random selection can be performed by selecting lines at random, with at different intervals. Using this scheme, while more practical than random point sampling, does not provide optimal quality. Furthermore, Cartesian under-sampling at random¹ also results in degraded image reconstruction. The reason can become obvious by studying the *point spread function* (PSF) of this trajectory. The PSF function can be used to measure the coherence of a sampling trajectory matrix. It has been shown[40] that the PSF of under-sampled of a Cartesian trajectory suffers from energy leakage. This means that during reconstruction the energy of one pixel is *leaked* to other pixels, introducing aliasing artifacts.

Another issue with randomly selected Cartesian sampling lines is the fact that it is not localized; it ignores the energy distribution of pixels of the image in the k-space. By looking at the k-space image in Figure 2.6 in Section 2.2, it can be recognized that most of the energy of an image is concentrated in the central region; at the lower frequency of the spectrum. This is because high resolution (detail) features of an image do not contribute to the energy of an image as much as the lower resolution (low frequency) particles do. Hence, it seems irrational to select equal number of samples from the lower and higher frequency ends of the spectrum. Non-uniform sampling is a way to deal with this issue. By varying the density of samples from region to region, one is able to choose more samples from the higher energy particles and fewer samples from the high frequency components.

Hence, a more suitable trajectory design, and the one used in this proposed technique, is variable density (non-uniform) Cartesian sampling design. Incorporating variable density (Cartesian) sampling trajectory has other advantages as well. By focusing the sampling lines at the centre of the k-space, where more energy is concentrated, the effect of energy leakage can be reduced (in relative terms). Since more samples with higher energies are selected, the average energy of the sampled components increase with respect to the energy leakage of each individual pixel; creating a framework whereby more artifacts (aliasing components) can be suppressed. This is an addition to the fact that these trajectories also imitate incoherence characteristics of completely normal Cartesian sampling.

¹It is re-iterated here that equi-spaced Cartesian under-sampling is not optimal as stated in previous chapters.

Figure 5.1: Illustration of the difference between (a) a random Cartesian trajectory and (b) a variable density, non-uniform Cartesian trajectory. Note that in variable density pattern, most of the samples are collected from high energy components of the k-space, resulting in better reconstruction.

5.3.3 Image Collection

With the appropriate sampling trajectory in place, the next step to acquire images in order to test the edge detection algorithm. Since the acquisition is two dimensional, Cartesian sampling can be implemented with a series of parallel lines along the phase encode axis (y-axis in the k-space). In order to materialize the aforementioned trajectory the corresponding pulse sequence needs to be designed.

The pulse sequence for phase encode sample acquisition is similar to that in Figure 2.4(a); where the amplitude G_x (gradient for x-direction movement) is responsible for moving choosing the location of the line along the x-axis of k-space, while G_y (gradient responsible for motion along y-direction) moves the trajectory up and down along the chosen line, while samples are selected at $T_a q$ intervals. By choosing the right combination of amplitudes the two gradients, one can control where the density of sampling lines is higher and where it should be lower. This is the pulse sequence that needs to be implemented for to achieve the desired variable density Cartesian sampling trajectory for this design.

Unfortunately, due to inaccessibility to clinical MRI units and/or real patient consented k-space database, genuine brain images could not be collected for testing purposes; and another course of action needs to be taken.

In many medical image processing and MRI image processing, particularly those related with brain, a few phantoms have been created that replicate features of a brain. One of these phantoms is the well known *Shepp-Logan* phantom (Fig. 5.2(a)). The Shepp-Logan phantom is an excellent tool in analyzing and testing an algorithm as it contains many features of different intensities and shapes, at different distances from each other. However, this image can not be used without modifications. The MRI provides k-space data; which is sparse and can be randomly sampled. Thus, in order to simulate a real life MRI acquisition, the images in database need to be converted into their k-space counterparts. This can be achieved by taking the two dimensional DFT of the original image. Since the DFT is symmetric, providing frequency components at each half of the plane that mirror each other, with the higher frequency components on the edge of the frequency domain, the data needs to be shifted to

Figure 5.2: (a) An image of a Shepp-Logan phantom.

the centre of the plane; where lower frequency components need to be (Fig. 5.3).

Figure 5.3: The Fourier transform (k-space) of a Shepp-Logan phantom. The Fourier transform produces (a) symmetric frequency plane with low frequency components located at the corners of the plane, the image needs to be (b) shifted in order to acquire the proper k-space as an MRI machine would sample.

After calculating the k-space for each image, variable density Cartesian trajectories need to be applied in order to sample the k-space. A variable density sampling trajectory can be created using a Probability Density Function (PDF) of the form

$$PDF(x) = \begin{cases} 1 & x \leq R \\ (1-x)^p & x > R \end{cases}, \quad (5.1)$$

where $0 < R < 1$ and R is a real number, x is the x direction co-ordinate on the plane, and p is the power of the polynomial that controls how fast $PDF(x) \rightarrow 0$ as $R \rightarrow 1$ ¹. The first line in equation (5.1) dictates how widely

¹In general, the higher p is the steeper the decline of the polynomial outside the selected region for x (i.e. $PDF(x) = 1$ for $x \leq R$).

the centre of k-space is sampled; combined with the relationship on the second line one can control percentage of sampling lines used; which is in effect the under-sampling factor. In order to test the quality of edge detection with respect to the percentage of samples acquired, three different sampling densities (50, 40, and 30 percent of total number of samples) are to be investigated that are depicted in Figure 5.4.

Each mask will be applied to the k-space of each image in order to obtain the samples for reconstruction.

5.4 Edge Image Acquisition

5.4.1 CS Reconstruction

The process of k-space sample acquisition is just the first step in providing a meaningful edge image, though very significant for reasons mentioned before. To be able to output an image with sub-Nyquist sampling, an optimization reconstruction method is required, similar to one stated in (2.62). A more in depth analysis of this equation follows. It has been shown[17] that to approximate a signal with an error ϵ (error could be noise and other artifacts) a *basis pursuit de-noising algorithm* of form

$$\hat{a} = \min_a \|a\|_1 \text{ such that } \|\Psi^* a - f\|_2 \leq \epsilon, \quad (5.2)$$

where $\hat{a}$ are signal coefficients (approximated) in a sparse domain Ψ , and the signal can then be reconstructed by

$$\hat{f} = \Psi^* \hat{a}, \quad (5.3)$$

where $\hat{f}$ is the reconstructed signal. This algorithm can also be used for approximation of sparsely sampled signals (as they include imperfections). Let the image be denoted by a complex matrix x and y be the set of samples acquired. In addition, the sparsifying basis for the image is Ψ and the Fourier representation of the sampling trajectory is F_u (an under-sampled subset of all possible Fourier coefficients). By replacing these parameters into equation (5.2), and re-arranging the equation into its *Lagrangian* format as

$$\arg \min_x \|F_u x - y\|_2^2 + \lambda \|\Psi x\|_1, \quad (5.4)$$

where λ is a *regularization* parameter, that is responsible for certain aspects of signal reconstruction; more insight on the responsibility of λ in reconstruction will be provided later. The relationship in (5.4) is an *unconstrained* form of equation (5.2). Solving the equation as an unconstrained problem in this manner has benefits; since the solution space is large it is more efficient to solve an unconstrained problem rather than a cost function that is constrained. The unconstrained problem in (5.4) can be solved using iterative techniques (among other approaches) that perform thresholding from one iteration to another; which also assist with de-noising and interference cancellation. These approaches use thresholding [19], [24] to decrease the ℓ_1 norm of the signal coefficients (in this case x) and a *gradient descent* to decrease the overall value of the least square term.

Figure 5.4: Illustration of PDF functions for different sampling density (left) and their corresponding non-uniform Cartesian sampling trajectory (right): (a) a PDF function with sampling density of 30% of total number of samples, (b) the sampling trajectory to collect 30% of samples; (c) PDF for sampling density of 40% and (d) corresponding Cartesian trajectory; (e) PDF for 50% sampling and (f) variable density Cartesian trajectory for 50% sample collection.

Gradient descent

Gradient descent method of optimization is part of a group of techniques for solving unconstrained optimization problems. A descent method attempts to minimize a sequence of form

$$x_{k+1} = x_k + t_k \Delta x_k, \quad (5.5)$$

where $k = 1, 2, \dots$, and the process of minimization continues in an iterative fashion until the optimal x_k is achieved; whereby $x_{k+1} = x_k$ ($t_k > 0$ unless x_k is optimal). In (5.5) Δx is called a step or search direction and t_k is the step size at the k 'th iteration. A descent method is characterized by the fact that $f(x_{k+1}) < f(x_k)$ ¹. Convex optimization problems (such as the one in (5.4)) are solved by finding the rate of change of a function with respect to a parameter; by finding the fastest rate of change (ascent or descent, depending on the direction required), one is able to close in on a solution (finding the best solution among others). The rate of change of a function with respect to its parameters x_n can be determined using its gradient vector of form

$$\nabla f(x_n) = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right\}^T, \quad (5.6)$$

where T stands for a transposition operation. The gradients in the vector represent the *direction* of the steepest ascent and the negative values of the vector components represent the direction of the steepest descent. Now the goal is to find the rate of change of function f along the direction Δx_k away from a point x_{k+1} (this is how the maxima or minima is identified); with respect to the step size, t_k (the parameter). In other words

$$\frac{df}{dt_{k+1}} = \frac{\partial f}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial t_k}, \quad (5.7)$$

where according to equation (5.5)

$$\frac{\partial x_{k+1}}{\partial t_k} = \frac{\partial}{\partial t_k} (x_k + t_k \Delta x_k) = \Delta x_k. \quad (5.8)$$

Combining (5.7) and (5.8) results in

$$\frac{df}{dt_k} = \nabla f(x_k)^T \Delta x_k, \quad (5.9)$$

where the objective is to find a value for t_k^* that minimizes f so that $\nabla f(x_k)^T \Delta x_k = 0$. One way to get the optimal value for the step size is to use the *steepest descent* method. However, it is shown[54] that the steepest descent method results can be improved by using a *gradient descent* method with an algorithm described as follows

Gradient Descent Algorithm

- 1: **Given** An arbitrary initial point x_k
- 2: **Iterate**
- 3: Determine the descent direction as $\Delta x_k = -\nabla f(x_k)$
- 4: Search for optimal step size t_k^* : $t_k^* > 0$

¹There are many *ascent* methods as well; whereby $f(x_{k+1}) > f(x_k)$.

```

5:      If
6:           $x_{k+1} = x_k - t_k^* \nabla f(x_k)$ 
7:      Stop
8:      Else
9:           $k = k + 1$ 
10:     End Iterate

```

Table 5.1: Pseudo-code for a sample gradient descent algorithm.

The process of searching for step size on line (4) of algorithm in Table 5.1 is called a *line search*. One method of performing this search is a *backtracking line search* which is less computationally costly than some other methods; namely, exact line search². The backtracking line search reduces the step size by a factor β until a certain criteria is met. This line search is computed as

Backtracking Line Search

```

1: Given  $\Delta x_k$  with  $\alpha \in (0, 0.5)$  and  $\beta \in (0, 1)$ ; initialize  $t = 1$ 
2:      while
3:           $f(x_k + t\Delta x_k) > f(x_k) + \alpha t \nabla f(x_k)^T \Delta x_k$ 
4:          Upgrade  $t = \beta t$ 

```

Table 5.2: Algorithm for backtracking line search.

When the backtracking line search is incorporated in the gradient descent method provides a *conjugate gradient* method that can be solved using the following algorithm.

Conjugate Gradient Algorithm

```

1: Given An arbitrary initial point  $x_k$ ; assume  $k = 0$ .
2:       $\Delta x_k = -\nabla f(x_k) = -\nabla f_k$ 
3:       $t = 1$ 
4:      Iterate
5:      while

```

²Exact line search is only used when the cost of calculating the step size is lower than the cost of computing the search direction itself.

```

6:       $f(x_k + t\Delta x_k) > f(x_k) + \alpha t \nabla f(x_k)^T \Delta x_k$ 
7:      Upgrade  $t = \beta t$ 
8:       $\Delta x_{k+1} = -\nabla f_{k+1} + \frac{|\nabla f_{k+1}|^2}{|\nabla f_k|^2} \Delta x_k$ 
9:      If
10:          $x_{k+1} = x_k - t_k^* \nabla f(x_k)$ 
11:      Stop
12:      Else
13:          $k = k + 1$ 
14:      End Iterate

```

Table 5.3: Algorithm for solving of a convex optimization problem using conjugate gradient technique.

The above algorithm can be used in conjunction with two stopping criteria as outlined by [40]; one using the maximum number of iterations (set to 100) set to ensure convergence, and the other a minimum magnitude for the gradient $\nabla f(x_k)$ (set at 10^{-4}). The two constants α and β can be set to 0.05 and 1, respectively.

The algorithm in Table 5.3 provides a solution to equation (5.4); using which, the image can be reconstructed. In order to do so, the gradient of the relationship in (5.4) needs to be calculated such that

$$\nabla f(x) = 2F_u^*(F_u x - y) + \lambda_\Psi \nabla \|\Psi x\|_1, \quad (5.10)$$

where λ_{TV} is omitted for simplicity.

Although in this thesis the main concern is not the quality of the final image, the iterative nature of the reconstruction algorithm is vital to achieve the desired results. The iterative nature of the algorithm (lines 4-14) act as a soft thresholding method that reduces noise; coupled with a noise reducing TV operator, the algorithm reduces noise at every iteration while preserving important features. By nesting the iteration loop in the algorithm presented in Table 5.3 into another iteration loop, it is possible to decompose the image after each thresholding process is completed. Then the reconstructed image (with fewer noise components and artifacts) is processed with the conjugate gradient technique once again, and then decomposed. This process is repeated for a number of times until the image is completely noise free.

Aside from the algorithm and the TV operator, there is another parameter with essential implications to the results; the choice of sparsifying transform.

5.4.2 Sparsifying transform

One condition of a successful CS process is to have an sparse signal; the more sparse the signal the better. Hence, choosing a optimal sparsifying transform is of extreme importance for good quality reconstruction.

Although images in their k-space are sparse, there maybe other more sparse domain for such signals. In Chapter 3 it was shown that wavelets are capable of representing images (signals in general) with just a few large coefficients; much fewer than the Fourier domain representation. In addition, wavelets have the ability to filter

images into approximation and detail variations; with the detail signals being filtered in three different directions. This makes wavelets a perfect choice for both edge detection and for compressed sensing reconstruction of brain MRI images.

It has been shown that wavelets are the optimal choice of sparsifying transform for both brain MRI CS reconstruction as well as edge detection. However, there are many different classes of wavelets to choose from; each having a certain set of properties that can affect the outcome of the edge extraction and its quality. In order to examine the effect of different wavelets, a variety of wavelet families are implemented and utilized.

Daubechies wavelet

The Daubechies wavelet family has been explained to have minimum support for a p number of vanishing moments. These wavelets can be calculated with a conjugate mirror filter h , where the frequency response of the filter, $\hat{h}$, has p zeroes at $\omega = \pi$. Then the wavelet (the conjugate mirror filter h) is shown [23] to have $2p$ non-zero coefficients. Hence, the number of wavelet coefficients increases two folds as the number of vanishing moments increase. Consequently as the number of vanishing moments increase so does the regularity of the wavelet; since there are more coefficients available to model the wavelet (see Fig. 5.5).

On the other hand, it was mentioned earlier that an increase in the number of vanishing moments is responsible for growth of the support of the wavelet (not a desired situation). The Daubechies wavelets are able to minimize the effect of high number of vanishing moments as they provide minimum support for a number of vanishing moments compared to other wavelet families. As it can be detected from Figure 5.5, Daubechies wavelets are highly asymmetrical¹. This means that most of the energy of the wavelet is accumulated toward the start of its support; while asymmetric filters are also more difficult to construct. Hence, it might be useful to examine a more symmetric family of wavelets.

Symmlet wavelet

Symmlet wavelets are designed to provide compact support while optimizing the the conjugate mirror filter h for a more symmetric design. Symmlets have a support of $[-p + 1, p]$ for a number of vanishing moments p (compared to $[0, 2p - 1]$ for Daubechies). It can be seen from the support that the energy is more evenly distributed around the centre of the support; as it is illustrated in Figure 5.6. The Symmlet family of wavelets, since symmetric, provide redundancy in producing the wavelet coefficients. Although this is not a desirable condition from a compressed sensing point of view (remembering that in CS the objective is to represent a signal with as few coefficients as possible), it might lead to a better quality edge detection.

Coiflet wavelet

Another symmetric (Fig. 5.7) family of wavelets is called Coiflets, designed by Daubechies at the request of Ronald Coifman [22]; whose wavelet and scaling function both have the same number of vanishing moments. This wavelet family is particularly interesting because it provides good approximation of signal samples at finer scales. The Coiflet wavelet family provides a support of $3p - 1$ compared to Daubechies' $2p - 1$.

¹This is because they are designed to provide minimum support and follow a minimum phase design.

Figure 5.5: The Daubechies wavelet function family from (a)(db4) to (f)(db14) the vanishing moments increase by one from figure to figure; where dbN is a Daubechies wavelet function with $N/2$ vanishing moments. Note how the support of the wavelet function is increased as the number of vanishing points increase. Only the first three wavelets in the family (i.e. db4, db6, and db8) are used in the results.

Figure 5.6: Illustration of the Symmlet family of wavelets with increasing vanishing moments from (a) to (d); where $symN$ has N vanishing moments. The number of vanishing points of a Symmlet wavelet is half its support. Note the symmetry of the wavelets, and its energy distribution; toward the middle of the support. Only the first three (i.e. sym4, sym6, and sym8 will be used in the design evaluation).

5.4.3 De-noising

Now that the choice of sparsifying transform is clarified, with several wavelet families to be examined; it is time to see how a de-noised edge image can be constructed.

In Section 5.4.1 it was shown how iterative reconstruction approaches utilize thresholding process as a tool for noise reduction and interference cancellation. However, some approaches [53] indicate that utilizing other transforms could provide better noise suppression. One of such transforms; which was also used in the original CS paper by Candes et al. [13], is the *Total Variation* (TV) norm³. Total Variation objective is a well known function used in many imaging applications for de-noising purposes [55].

The finite difference of an image is essentially the difference between the intensities of two pixels at two different positions. Total Variation is a smoothing operator that attempts to reconstruct an image from its finite differences transform by *smoothing* the difference between intensities of different pixels. In effect TV applies a measure of continuity to the image, by *filling in* the differences with approximated intensities. The TV can be

³TV norm is the objective function of optimization where the finite differences is used as a sparsifying transform.

Figure 5.7: Coiflet family of wavelets, with (a) through (c) representing *coif1*, *coif2*, and *coif4*, respectively; where *coifN* has $2 \times N$ vanishing moments. Note the energy distribution of the Coiflet family is concentrated toward the beginning of the support, much like the Daubechies family.

stated as a smoothing function as

$$\varphi_{TV}(\hat{m}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i| = \|TV(\hat{x})\|_1, \quad (5.11)$$

where $x \in \mathbb{R}^n$ and $\|TV(\hat{x})\|_1$ is called the TV of x . The total variation penalty in an optimization problem works by assigning large penalties to abruptly changing values of $\hat{x}$. Hence, TV penalty adds more noise reduction benefits; while preserving edges, making it an optimal regularization parameter for noise reduction of edge images. There is; however, the potential for TV to attenuate overall image intensity as well as edge intensity; which is not a desired situation. Hence, the optimal choice for the TV penalty is a challenge in acquiring a high quality edge image. With the addition of the TV penalty, equation (5.4) becomes

$$\arg \min_x \|F_u x - y\|_2^2 + \lambda_\Psi \|\Psi x\|_1 + \lambda_{TV} TV(x), \quad (5.12)$$

where λ_Ψ and λ_{TV} regularization parameters that trade Ψ (wavelet) sparsity and TV sparsity between them; while they also balance between sparsity and data consistency. In general, higher regularization values results in more noise cancellation, while lower values result in poor quality reconstruction and higher noise. In fact with respect

to equation (5.2), the values for λ can be chosen so that the reconstruction results in

$$\|F_u x - y\|_2^2 \approx \epsilon. \quad (5.13)$$

However, this does not indicate that one could increase the values of regularization parameters in order to get noise-less reconstruction. Extremely high values of these parameters have been shown to cause resolution loss, while also causing over-smoothness (especially in the case of λ_{TV}), and loss of signal features.

Thus, finding the right balance between the value of regularization parameters is crucial to obtaining acceptable results; while keeping in mind that the primary objective in this work is to suppress noise in edge images, rather than obtaining high quality (original) image reconstruction. Therefore, over-emphasis of noise reduction and sparsity, compared to normal reconstruction values, is required in this work.

In order to obtain optimal values for λ_ψ and λ_{TV} , no selection criteria is at hand; hence, a more experimental approach is taken⁴. The approach is performed by keeping one parameter constant, while the other one is increased, then the two are reversed. This process is performed for a number of different values; whereby a graph is obtained that compares these values against the quality of image (the difference between the reconstructed image and the original image). The graph can be utilized to find the optimal values for the regularization parameters.

However, the objective here is not to arrive at a high quality image reconstruction, but a good edge image. Hence, instead of a graph that compares regularization parameters against quality of the reconstructed image, a method is devised that ranks the SNR(Signal-to-Noise Ratio) of the reconstructed edge image against corresponding regularization parameters. After a graph is created for each parameter, the best values with the highest SNR are picked, and then fine-tuned for better performance.

5.5 Evaluation

In order to evaluate the performance of the proposed method of edge detection, it is necessary to define certain benchmarks to compare the method with. The proposed design is optimized for robustness, while its objective is to provide de-noised, artifact free, edge images of a brain MRI image.

For any signal/image de-noising problem the goal is to calculate and approximation signal $\hat{x}$ of the image that is, ideally, the same as a noise free image x , under a certain set of criteria. Therefore, similar to many approximation problems, the goal here is to reduce the *error* of the output image compared to the ideal image. A widely used criteria to quantitatively measure this error is the *Mean Square Error* (MSE) that can be calculated as

$$MSE = \frac{1}{N} \|x - \hat{x}\|^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2. \quad (5.14)$$

Whenever a signal processing problem centred around the topic of noise is involved (as is often the case), another

⁴There are optimal *L_Curve*[21] selection methods for other reconstruction techniques such as Tikhonov regularization available; however, no mathematical algorithm is yet found for parameter selection of the reconstruction method presented here.

criteria is always measured, called the *Signal-to-Noise Ratio* (SNR) of the signal; which can be stated as

$$SNR = 10 \log_1 0 \frac{\|\vec{x}\|^2}{\|\vec{x} - \hat{\vec{x}}\|^2} = 10 \log_1 0 \frac{\frac{\|\vec{x}\|^2}{N}}{MSE}, \quad (5.15)$$

in terms of the MSE; where the SNR is in *dB*. Another evaluation measure is the *Peak Signal-to-Noise Ratio* (PSNR); which calculates the ratio of the maximum pixel value in the image and the highest possible noise power in the image. For a gray-scale image the highest intensity level of any pixel (for an 8 bit image) is 255 and the noise component could be stated as the *MSE*. Then the PSNR can be calculated as

$$PSNR = 10 \log_1 0 \frac{\frac{255^2}{N}}{MSE}. \quad (5.16)$$

All of these performance measure could be used in estimating the capability of a noise reduction processing technique to approximate the original signal. However, especially in images, these measurement criteria do not fully express the performance of the process (more information also in [66]).

For images, their *visual quality* is significant, and can not be simply stated as a numeric measurement of difference between two set of matrices (images). Visual quality reflects how human eyes would interpret an image; for instance, humans maybe more tolerable in defining some noise particles in an image. In this manner, visual quality is very dependent on the application [6] and is said to be subjective; meaning that in some applications certain artifacts may result in the incorrect data interpretations while in other it they may not. Thus, quantitative assessment of output images is not adequate on its alone; in fact, visual inspection of these of edge images may provide better understanding of how it performs.

In addition, another problem with quantitative performance measurement using SNR and MSE indicators, is the fact that these equations, in essence, compare the output image with a control (standard), uncorrupted image. Nevertheless, such a control image does not exist for most images in the database. In order to construct a control image, the edge image that is the result of the designed technique can only be compared to another edge image resulting from another edge detection method that is considered an standard (Canny and multi-scale edge detection algorithms are the two used here); which makes the comparison only relative to the control image. Moreover, even the best control image may contain noise particles and/or may not contain the same set of edges of the test image. This will become more clear in the following section, where results are discussed.

Hence, both quantitative (where applicable) and visual inspection (and comparison) of the test images are provided for validation purposes.

5.6 Summary

This chapter described the key concepts that shape the implementation of the proposed edge detection technique based on compressed sensing. The chapter starts by providing some key requirements that need to be addressed during the design process; after which, it discusses the image acquisition method. Then, the core concepts behind the actual edge detection algorithm are explored. The chapter finalizes by describing some of the existing evaluation techniques that can be used to measure the success of the proposed technique.

Chapter 6

Results

6.1 Overview

In this chapter the performance of the proposed technique will be evaluated. The chapter begins by explaining the role of the two regularization parameters λ_{TV} , and λ_Ψ . Further, the discussion is expanded onto how these parameters can be chosen for optimized performance. From there, the discussion will move onto the effect of different wavelet families as well as the effect of different vanishing points of wavelets. Thereafter, the response of the algorithm will be tested in accordance to different sampling trajectories; and finally, a discussion of noise handling of the algorithm will be provided.

6.2 Regularization Parameter

The edge detection technique described in Section 5.4 is implemented using the algorithm described in Table 5.3, in order to acquire edge images vis compressed sensing.

The first step in performing edge detection using the designed method is to adjust λ_{TV} and λ_Ψ for optimal image quality. This is done by developing SNR graphs as explained in Section 5.4.1. The values of regularization parameters are incremented in steps of 0.05 from zero to 0.7 and the resulting images are compared base on their SNR values. The original image is a Shepp-Logan phantom (noise-free), and the control images are Canny edge detected phantom and and edge image achieved by performing multi-scale edge detection. Figure 6.1, illustrates these graphs for CS edge detection technique with low and high resolution wavelet scales; Figures 6.2, 6.3, and 6.4 illustrate the same results for λ_Ψ , λ_{TV} , and λ_Ψ variations with respect to a Canny edge detected, and Multi-scale edge detected (the latter two) images, respectively.

By inspecting these graphs and comparing the results it can be concluded that, compared to a Canny edge detected image, the SNR of a lower resolution CS edge detected image increases as the value of λ_{TV} increase (see Fig. (a)), while for the higher resolution images, the SNR approximately stays constant (Fig. (b)). Thus it would seem sensible to choose a value for λ_{TV} that is as high as possible. However, by having a second look at Figure (a) it seems that for values higher than 0.2 does not have much effect on the SNR of the image. This, coupled with

Figure 6.1: Graphs of SNR vs. λ_{TV} values, with a Canny edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).

Figure 6.2: Graphs of SNR vs. λ_{Ψ} values, with a Canny edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).

the fact that (as previously mentioned) a higher λ_{TV} works against edge preservation, makes the choice a value in the range of 0.05 – 0.2 the ideal choice.

In addition, the SNR of the CS edge detected image decreases on both low and high resolution scales (Fig. fig:xfmSNRcanny) when compared to a Canny edge image. Hence, a low λ_{Ψ} value seems to be the logical choice. However, a higher λ_{Ψ} increases sparsity; which is an optimal situation for edge detection. The exact value of this parameter will be fine tuned with respect to the final results (more explanation on the choice of parameters and the SNR will be provided later in the results). In addition, it is observed that the SNR of the edge image is higher at higher resolution scales; which indicates that with every progressive octave in the wavelet,

more edges are preserved. Observation of graphs that compare CS edge detected image with multi-scale edge

Figure 6.3: Graphs of SNR vs. λ_{TV} values, with a multi-scale edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).

Figure 6.4: Graphs of SNR vs. λ_Ψ values, with a multi-scale edge detected image as the control image; (a) compares images achieved at lower resolution wavelet scales, while (b) compares images resulting from higher resolution wavelet scales. The SNR values are in Decibels (dB).

images; results in the same conclusion; the ideal choice for λ_{TV} lies on a range of 0.05 – 0.2, while the choice for λ_Ψ should be fine tuned.

Inspection of SNR graphs; nevertheless, reveals other details; it is discovered that the SNR values are surprisingly low; unacceptable for most image processing applications, despite the fact that the images are normalized. However, the reason behind this fact does not lie within the edge detection technique. Further analysis of the actual images retrieved may provide some insight.

Figure 6.5: Edge detected phantom images utilizing (a) Canny edge detection and (b) multi-scale edge detection.

Figure 6.6 provides a set of edge detected images using the proposed methods at different (increasing) values of λ_{TV} . When these images are compared to those in Figure 6.5, it is apparent that the proposed method detects more edges than the multi-scale edge detection; while matching the Canny edge detection algorithm in terms of the number of edges detected.

Since the for SNR calculations a signal is compared to a control signal that is considered to be noise free¹. However, in the case of the multi-scale edge detected image of the phantom, the image has fewer edges marked. This means that the extra edges in the image from the proposed method will be interpreted as noise during an SNR calculation; decreasing the signal-to-noise ratio significantly.

In the case of analysis with respect to the Canny edge detected image, it can be argued that the Canny method provides thinner edges (because to non-maximum suppression step), while the edges marked in the image outputted by CS edge detection technique are thicker. Similar to the aforementioned scenario, these edges are also treated as noise in the process of SNR computation.

Hence, the SNR, as mentioned in Section 5.5 is not capable of correctly and thoroughly evaluating the performance of proposed technique. Still, this, by no means, implies that the resulting edge image from the CS edge detection technique is free of any distortion. In fact, the images suggest that improvements can be made. It can be seen from the images in Figure 6.6, that by increasing λ_{TV} the area around the edges of the image becomes more bulky, and are drawn out to the background pixels. This is due to the fact that TV tends to spread energy from higher energy pixels toward lower energy (intensity) pixels; this is how it applies continuity to the image.

On the contrary, results indicate that the edges acquired become lighter as λ_ψ is increased, to a point that there are only sporadic pixels on the outline of the edge (Fig. 6.8(d)); which also reduces the SNR. This is because a higher λ_ψ promotes sparsity; which means fewer coefficients represent the signal. Hence, it is an integral part of noise reduction when coupled with λ_{TV} ; making them both an integral part of producing a high quality edge image. The complete dictionary of images for different variations of λ_ψ and λ_{TV} are available in Appendix A.

Nonetheless, the values for the two regularization parameters are chosen from the aforementioned ranges.

¹ Although all input images are noise free, CS reconstruction is subjected to under-sampling artifacts; promoting the notion of SNR.

Figure 6.6: Edge detected image using the proposed methods with λ_Ψ set to zero and λ_{TV} set to (a) 0.05 (b) 0.1 (c) 0.15, and (j) 0.50; images are from lower (coarse) resolution scale.

The process of edge detection is performed for a number of different combinations of parameters; fine tuned every time to render better results. After a series of fine tuning steps it was observed (from visual quality and perception stand point) that the best for λ_Ψ and λ_{TV} are 0.05 and 0.15, respectively (see Fig. 6.7). The process of obtaining an edge image includes performing a number of iterations of the aforementioned conjugate gradient technique. The conjugate gradient is modified to perform 10 iterations; after every 10 iterations, a new image x is acquired, then the process is repeated for again. This second iteration is also performed for 8 iterations. Hence, the total number of iterations is 80; which is about what is needed for these images to be evaluated without any artifacts remaining. The edge image acquired is in the form of that stated in Figure 3.7, where each sub image (detail) is approximated in one direction (vertical, horizontal, and diagonal). A sample of such an image is provided in Appendix A. These three sub-images are fused together (combined) via a absolute maximum fusion

Figure 6.7: A fine-tuned edge image with λ_Ψ and λ_{TV} set to 0.05 and 0.15, respectively.

Wavelet	SNR
<i>Coif1</i>	0.4331
<i>Coif2</i>	-0.4428
<i>Coif4</i>	1.2180
<i>Sym4</i>	0.8137
<i>Sym6</i>	-0.4702
<i>Sym8</i>	1.2156
<i>Db4</i>	8.0978
<i>Db6</i>	6.3269
<i>Db8</i>	5.1035

Table 6.1: SNR values for different wavelet members of different wavelet families.

rule, meaning that the maximum intensity pixels are selected between the three images and combined together. This is the edge image that is used as the final result.

But if the control images used are not suited for SNR evaluation, what are other options? This is a valid question as it was promised that quantitative evaluation (though not an absolute measure) will be provided. The control image could be computed by constructing an edge image using the proposed method while full sampling is performed, making sure there are no artifacts. Indeed this control signal, illustrated in Figure 6.9, is what will be used from this point forward as a measure of quantitative SNR evaluation.

6.3 Wavelet Assessment

With the parameters adjusted for optimal performance, it is time to assess the effect different wavelet families (and different support size within each family) have on the quality of edge images acquired. The wavelet families and the specific wavelet to be used was indicated in Section 5.4.2; these wavelets are *db4*, *db6*, *db8*, *sym4*, *sym6*, *sym8*, *coif1*, *coif2*, and *coif4*. The edge detection process is performed with each wavelet and the

Figure 6.8: Edge detected image using the proposed methods with λ_{TV} set to zero and λ_{Ψ} set to (a) 0.05 (b) 0.1 (c) 0.15, and (d); images are from lower (coarse) resolution scale.

results are compared based on SNR as well as image quality. Every time a wavelet transform is performed it is taken to the 6^{th} scale. Since these wavelets have a scale factor of 2^j representing each octave, these means that the wavelets are iterated to 3 octaves. This means that the structure of the decomposition pyramid is similar to that in Figure 3.7. Since each level has 2^{2k} coefficients (where k is the level number), it is implied that the final level (higher resolution) estimates the signal with more wavelet coefficients than the level before. This, in turn, suggests that the previous levels are more sparse; while, at the same time, levels at higher resolution (higher level numbers) are expected to provide better information on the detail of an image, namely the edges. However, more coefficients, and in particular, more high resolution coefficients, means that wavelets at higher octaves are possible also more susceptible to noise particles, even in the presence of a noise canceling algorithm. In addition, larger features of blurred features are better defined at higher octaves, while sharper features can be observed at lower

Figure 6.9: The control signal used for SNR evaluation henceforth.

Figure 6.10: Edge detection images from the proposed method for wavelet choices of (a) *Daubechies4*, (b) *Daubechies6*, and (c) *Daubechies8*, respectively.

levels.

The SNR values for all families of wavelets used are summarized in Table 6.1; which suggests that the best wavelet choice is the *Daubechies4* wavelet, and the best wavelet family is the *Daubechies* wavelet as well. Figure 6.10 illustrates the edge images created using three wavelets from the *Daubechies* family. Visual inspection of the quality of edge images confirms that the best choice of wavelet is also *Daubechies4*. Figure 6.11 provides images for the best quality edge detection corresponding to the best wavelet choice of each family. For a complete illustration of edge images for all wavelets mentioned above, please see Appendix A.

The difference between the quality of edge detection images obtained from difference wavelet transforms can be tracked to the number of vanishing moments of each wavelet. As it was mentioned earlier, vanishing moments decide how smooth a wavelet function is (regularity), but also extent its support. The number of vanishing moments of a wavelet function is decisive in how well it is able to represent polynomial behaviour of a signal. Each vanishing moment can be seen as a representation of a polynomial coefficient; indicating that a wavelet function

Figure 6.11: Edge detection images from the proposed method for wavelet choices of (a) *Coiflet1*, (b) *Symmlet4*, and (c) *Daubechies4*, respectively.

NO. of Samples	SNR
50	5.0894
40	3.6357
30	2.0977

Table 6.2: SNR values for different ratio of samples collected using pseudo-random Cartesian sampling (i.e. 50% indicates that only half the samples required by Nyquist criterion are collected).

with one vanishing moment is capable of estimating polynomials of with one coefficient over its support, while a wavelet function with two vanishing moments can represent polynomials with two coefficients. However, a larger support size is prone to *leakage* problems caused by overlapping end of translated wavelet and scaling functions (this is analogous to aliasing effect in signal reconstruction). Hence, although higher number of vanishing moments might be desirable, it creates problems of its own. For edge detection purposes, higher number of vanishing moments is not ideal as coefficient approximation is inaccurate and can visually be detected as stretched out edges, or detection of untrue edges.

6.4 Pseudo-random Sampling

Another important factor in the quality of edge images is the number of samples collected. In this thesis the sampling method, or rather, the under-sampling method is that of pseudo random Cartesian sampling. The algorithm is arranged to use the *Daubechies4* wavelet; since it is known to be the best choice of wavelet and produce the best results. Then edge detection is performed for each trajectory, and the SNR values are compared (Table 6.2). The results, as expected, indicate that the sampling pattern collecting half of the samples produces better edge images.

Although a 50% reduction in the number of samples is significant, visual inspection testifies that the difference between 50% and 40% sample acquisition rate is not significant (no visual loss). Hence, one may choose to perform edge detection with the lower sampling rate (i.e. 40%) for the sake of time reduction, and reduced patient

Figure 6.12: Edge detection images from the proposed method for different number of samples collected: (a) 50% of samples collected, (b) 40% of samples collected, and (c) 30% of samples collected.

exposure, and wait time; all without sacrificing quality of the image.

In addition, the performance of the edge detection technique could further be optimized by selecting an optimized non-uniform sampling trajectory. One way to do this is by designing a Cartesian sampling pattern that is created to obtain more samples from the centre of the k-space. For instance, if in a 50% sampling density Cartesian trajectory, half of the samples are collected from the centre and the other half from anywhere else in the k-space, the ratio could be changed so that 70% of the samples are collected from the centre while only 30% are selected from other parts. Then if this ratio is decreased to only 40% or even 30% of the total number of samples being acquired; the effect of artifacts can be limited.

Moreover, other sampling trajectories, namely non-Cartesian patterns can also be examined. These approaches may provide better edge preservation and/or noise cancellation (although in general they are more susceptible to system imperfection). This along with the above hypothesis are described later as part of the Future Work section.

6.5 Noise Suppression

Arguably, one of the most defining features of any practical edge detection method, is its capacity to suppress noise. The importance of such ability is only amplified in the world of medical imaging, as images are often contaminated with different types of noises and artifacts. The problem of noise is amplified in an edge detection process since noise particles are often views as high frequency components, similar to edges. Hence, simply filtering a noisy image with a high pass filter is not sufficient for interference cancellation.

As it was described in Chapter 4 most conventional edge detection techniques utilize smoothing functions in order to reduce noise in an image, and then perform the edge detection on the de-noised image. The performance of these edge detection algorithms is assessed based on their ability to suppress noises (among other criteria). To provide better noise cancellation, multi-scale edge detection, based on wavelet transforms, models smoothing functions and gradients using wavelets. This method is considered to be more successful than the classical approaches, as it is better at de-noising an image.

As it can be seen, interference cancellation is central to rating the success of an edge detection technique.

Hence, it is essential that such evaluations are performed on the proposed design, and compared to other techniques. In order to do this, the original image of a phantom is contaminated with different types of noise particles. These noise types are Gaussian, salt and pepper, and speckle noise. Each of these noise types (especially speckle and Gaussian) can represent a real life noise in an MRI brain image. Thereafter, the k-space of the noisy image is calculated the same as other images, using the 2D Fourier transform. Then the edge detection algorithm is performed on the images with 40% sampling rate.

Figure 6.13: Edge detection images from the proposed method for a Gaussian noise contaminated image. Different images with different regularization parameters.

The first noise type to be examined is the Gaussian noise; which represents an additive white noise in an image with a normally distributed probability density function. The results of edge detection are collected after 100 iterations, and the images are examined. These images are illustrated in Figure 6.13 for different values of regularization parameters. Figure 6.13(b), represents the edge detected signal using the original values for λ_{TV}

and λ_Ψ derived in Section 6.2.

In order to examine the effect of regularization parameters on noise rejection, these values are varied with respect to the original ones and the results are assessed. It is clear, by inspecting the quality of the edge images, that the original set of values for the parameters result in the best edge detected provides the best quality. However, it is invaluable to discuss findings from that are outcomes of changing these variables; as they explain different scenarios and trade-offs that could be implemented in a real world application of the proposed method.

Observation of the difference between images 6.13(a) and 6.13(b) indicates that the increase in λ_{TV} reduces the intensity of some of the edges; more specifically, smaller edges of lower contrast². This, coincides with the fact that a higher λ_{TV} value also rejects more of the additive noise, by applying a higher penalty to faster changing particles. Hence, not surprisingly, a good balance between the amount of noise suppressed and the intensity of the edges detected, is the trade-off that can be optimized by λ_{TV} depending on the exact situation.

In addition, the value of λ_Ψ also provides flexibility in terms of noise cancellation. It is observed that at lower λ_{TV} values, and increase in λ_Ψ provides the necessary noise reduction by means of sparsity promotion. This means that fewer significant coefficients are selected for signal reconstruction, and hence fewer high frequency components are selected. By doing so, λ_Ψ provides the means of noise cancellation by rejecting more high frequency components. Figure 6.13(d) indicates that higher values of both parameters provide better results compared to the case where λ_{TV} is set to 0.05. In addition to quality inspection, the signal-to-noise ratio of the best edge image available out of the four images is calculate; for the image with $\lambda_{TV} = 0.15$, and $\lambda_\Psi = 0.05$; which is evaluated to be 1.77 compared to the control image in Figure 6.9. In comparison to the SNR values in Table 6.1 and Table 6.2, this value is quite low. The main reason for this phenomenon is that there are background noise particles available in the background of the edge image, while the overall intensity level of the edge image may also be lower than the control signal. However, as stated time and again before, the SNR is a relative measure and does not indicate the absolute quality of the image produced. In addition, level of background noise compared to the edge particles is very low; which does not contribute to wrong visual perception of an edge.

Another type of noise to be examined is the so called Speckle noise. Speckle noise another type of noise that might be available in many medical imaging techniques; such as ultrasound, as well as MRI images. Existence of Speckle noise in an image causes resolution degradation; which becomes a problem since edges are considered to be high resolution particles, in general. Hence, it is of extreme significance, in MRI image processing and edge detection, to be able to reduce speckle noise wise preserving edges and the resolution of a signal.

In order to evaluate the performance of the CS based edge detection method; Similar to the treatment of Gaussian noise in an image, Speckle noise is applied to the phantom image. After the k-space of the phantom image with noise is determined, incoherent sampling is performed and the process of edge detection is undertaken, and continued for 100 iterations. Images for different values of λ_{TV} , and λ_Ψ are also provided in Figure 6.14.

The images indicate that the method is, again, able to detect all edge in the image. However, the amount of noise present in the image varies for different sets of regularization parameters. Again, a higher value of λ_{TV} performs better in terms of interference cancellation at the expense of image edge intensity. The noticeable difference between the results from the two different noise types (Gaussian and Speckle) is when λ_Ψ is increased in place of λ_{TV} (Fig. 6.13(c) and Fig. 6.14(c)). By comparing the two images, it is clear that in terms of the

²This was further confirmed by producing an image where λ_{TV} was set to 0.30; which indicated that the intensity of the inner feature edges is reduced even more

Figure 6.14: Edge detection images from the proposed method for a Speckle noise contaminated image. Different images with different regularization parameters.

speckle noise, the technique performs better in terms of preserving the edges. The reason could be explained by looking into the iterative thresholding mechanism of the conjugate gradient. In the case of the Speckle noise, the difference (contrast) between the intensities of the noise particles and the actual image is more than that of the Gaussian noise. Hence, when thresholding is performed on the significant components, the difference between them and the noise components is more evident. Hence, more significant components survive the threshold than noise particles, compared to the case of a Gaussian noise.

Hence, to suppress Speckle noise in an image for the purpose of edge detection, increasing the weight factor of the sparsifying transform may be desired in some cases where noise rejection may be more important than edge preservation. In addition, discontinuities in edges, as a result of higher λ_{ψ} value could be altered using edge linking or chaining methods; which connect discontinuities in edges to provide a continuous, smoothly varying

Figure 6.15: Edge detection images from the proposed method for a salt and pepper noise contaminated image. Different images with different regularization parameters.

edge. Finally, the SNR of the Speckle noise edge detected image is calculated (for Fig. 6.14(c)) to be 1.64; which is slightly lower than what the results obtained from Gaussian noise rejection.

The final noise type to be assessed is the salt and pepper noise. This type of noise is simulated by a set of random occurrences of white and black pixels in an image; which resemble the effects of faulty readings at a certain pixel. This type of noise is the least common noise in an MRI image; nonetheless, it simulates the existence of noise particles in an MRI image that are developed due to non-uniformity of the RF signals collected. The phantom image is noise contaminated and processed for edges; with the results compared in Figure 6.15.

In general, the images predict that the salt and pepper noise is the worst handled type of noise by the proposed method. Despite this, it is clear that the presence of noise does not interfere with the detection and marking of true edges in the image. This is a positive indication, as in many edge detection algorithms, noise hinders and

Method	Noise Type	SC
Proposed	Gaussian	1.0755
Canny	Gaussian	0.1322
Sobel	Gaussian	0.7687
Multi-scale	Gaussian	0.8566
Proposed	Speckle	1.0174
Canny	Speckle	0.2638
Sobel	Speckle	0.7687
Multi-scale	Speckle	0.9050
Proposed	Salt & Pepper	0.5590
Canny	Salt & Pepper	0.2182
Sobel	Salt & Pepper	0.2347
Multi-scale	Salt & Pepper	0.9973

Table 6.3: SNR values for different wavelet members of different wavelet families.

degrades the quality of the edge detection. Nevertheless, there is still noise present in the image. In addition, Figure 6.15(d) indicates that unlike in the previous cases, an increase in both λ_{TV} (0.15), and λ_Ψ (0.15) provides the best noise cancellation effect. In addition, the intensity of the edges produced using these set of values is higher than those in the other cases. Hence, in a practical setting these values could be adjusted accordingly, with the expected amount of noise available in the signal. The SNR of the noisy image in Figure 6.15(c) is calculated as 0.94; which indicates that the salt and pepper noise is the least rejected noise among the three.

However, as it was mentioned before, the concept of SNR and PSNR (Peak Signal-to-Noise Ratio) do not provide a good performance measurement criteria due to the fact that image perception is very subjective. In addition, we are also interested in the amount of edge that is preserved, whereby, the difference in the amount of edge present in the image can greatly affect the value of the SNR. This has made the finding of a *good* quantitative measurement tool a much sought after research area[65]. One criteria in particular is useful, as it provides some measure of how much edge is retained in the edge detected image; called the *structural content*. The structural content of two images is essentially the ratio of the amount of information that is available in the *reference image*, with respect to the *distorted image*. In the case of the work done here, the reference image is an edge image for a non-noisy phantom, while the distorted image is considered to be the edge images obtained from noise contaminated phantoms.

$$SC = \frac{\sum_{i,j=1}^{N,M} (fo_{i,j})}{\sum_{i,j=1}^{N,M} (fn_{i,j})} \quad (6.1)$$

In the above equation fo is the edge image of the original image, and fn is the edge image of the noise contaminated image. Hence, it can be stated that the closer the structural content (SC) is to 1 there is less noise in the noisy edge image (the actual amount of preserved edges can be verified via visual perception detection), where as more noise in the denominator creates a lower structural content value. The structural content parameter was calculated for different techniques and different noise types. The results of the calculation can be found in Table 6.3.

The results clearly indicate that the proposed technique is much superior to other techniques in the amount of edge preserved while rejecting the most amount of noise. A closer inspection of the values suggest that in case of the salt & pepper noise, the proposed technique falls short of the performance that is possible with the multi-scale edge detection process. However, by observing the corresponding image in Figure 4.1(s), it is clear that the reason the value of the structural content is higher than that of the proposed method is the fact that the multi-scale edge detection is incapable of obtaining smaller edges that are located inside the phantom. This example is proof that qualitative analysis of edge images are as important (if not more so) as quantitative evaluation of the edge detection technique.

In general, the values of the regularization parameters dictate how much noise is canceled and how much of the intensity of the edges in an image is preserved. Considering all three possible noise types explained here, the choice of these two parameters provides excellent flexibility in interference cancellation using the proposed technique. These values can be adjusted in a real life setting to respond to varying amount of different noises in an image; whereas in this experiment, the amount of these noise types are constantly distributed throughout an image.

It is also significant to discuss the edge images of higher wavelet resolution (i.e. the 3rd octave). These images are smaller in terms of scale (size) of the final image, but provide valuable information, nevertheless. An image of a third octave, detail approximation is provided in Figure 6.16.

Figure 6.16: A high resolution edge detected image of 3rd octave for $\lambda_{TV} = 0.15$ and $\lambda_{\Psi} = 0.05$.

The ensuing discussion can be generalized to all of the three types of noises explained before. As it is expected, images in higher resolution spectrum of the wavelet coefficients, carry more of the high frequency components (coefficients). Thus, the edges acquired in these images possess better quality in comparison to their lower resolution counterparts. Nevertheless, since noise particles are also higher frequency particles, there are also more noise components in high resolution scales. This is evident in the image, where with the same set of parameters and the same type of noise as in Figure 6.13(b), the higher level image contains more noise. The ability of higher resolutions to preserve more edges more accurately, may be beneficial under some conditions where more accurate edge preservation maybe the goal, and existence of a few noise particles does not result in the wrong interpretation of an image.

6.6 Summary

This chapter provided the results of the effect of different variables in the process of edge detection using the proposed technique. It provided insight on how regularization parameters are chosen, and what their effect is on the final edge detected image. Then, it provided the details on what the effects of different wavelet families could be on the final results. Further, the role of the number of samples acquired, as well as the role of incoherent sampling was demonstrated. Finally, the success of the proposed method in the area of noise cancellation of an edge image was evaluated in this chapter.

Chapter 7

Conclusions & Future Work

7.1 Conclusion

The research performed towards the completion of this thesis was aimed at designing and implementing a brain MRI edge detection method based on a rapidly developing compressed sensing theory that is utilized as an image reconstruction technique. The method combines the de-noising capability of compressed sensing algorithm, based on convex optimization approaches, with the ability of wavelets to perform edge detections via high pass filtering.

Before exploring the novel method developed in this thesis, an explanation of fundamental theories behind the design and implementation of the technique were provided to utilize the reader with the required understanding of these topics. Some of these topics included the theory of Magnetic Resonance Imaging and image acquisition using this medical imaging modality. In addition, the topic of compressed sensing and its core requirements were presented from a signal processing perspective. A basic theory of wavelets, as related to this work, was presented, as well as surveys of existing CS based image and signal processing methods, along with existing edge detection techniques already in use.

Using these principles, a method of edge detection based on compressed sensing was designed and evaluated. The method was evaluated and the effect of different variables in the design were observed and described in Chapter 6. Furthermore, the performance and success of the CS base edge detection technique was evaluated in presence of noise components; a realistic happening in many real life images. The results were comparable to those provided by both classical and new, wavelet based edge detection approaches explained in Chapter 4, and illustrated in Figure 4.1. The robustness of the designed method was examined by performing evaluations on images with three different types of noise: Gaussian, Speckle, and Salt & Pepper noise.

Simulation results were evaluated using qualitative and quantitative measures. An important aspect of the evaluation process is the visual quality or the perception quality of the detected edges. However, there is a need to be able to also quantitatively measure and evaluate the technique. For this reason a number of different method such as the SNR and PSNR evaluation were utilized. However, as expected, these measurements are only relative and do not provide and optimal means of evaluation. Thus, the *Structural Content* of the edge images (noisy) were calculated; which demonstrate how much of the edge image is preserved and how much of the noise suppressed.

For the proposed technique these values were 1.0755, 1.0174, and 0.5590 for Gaussian, Speckle, and Salt & Pepper noise, respectively. For the Canny edge detection technique, these values (in the same order) are 0.1322, 0.2638, and 0.2182; while for Sobel edge detection they are 0.7687, 0.7689, and 0.2347. By comparing the results, one can see (ideally the number should be 1) that the proposed technique preserves more edges; by comparing these results and combining it with the visual quality of images, it is possible to *see* how much noise is reduced. One exception to these results is in the case of the Multi-scale edge detection technique, where the structural content values are 0.8566, 0.9050, and 0.9973 for Gaussian, Speckle, and Salt & Pepper noise types; whereby, it is observed that this technique (according to these values) performs better than the proposed technique in the case of the Salt & Pepper noise. However, by comparing the qualitative results of the two methods, it is seen that the reference image of the Multi-scale technique (as well as the noisy edge image) actually contains fewer of the edges than the proposed technique (thus the higher value of structural content). Hence, it is concluded that quantitative evaluation alone does not provide useful information and is completely subjective and relative to the process being performed; amplifying the need for qualitative comparison during the process.

In addition to establishing better noise reduction (and better edge preservation), the proposed method provides the following benefits as well:

- It uses compressed sensing that utilizes under-sampling of MRI image k-space; which reduces imaging times and patient exposure, with the potential to reduce wait times for MRI imaging.
- It performs an edge detection algorithm on the acquired samples without having to reconstruct the image first; whereas other techniques, construct the image, perform noise reduction algorithms and then apply edge detection, all in different sequences; limiting the potential of implementation for a real-time, automatic feature extraction tool.
- finally, the proposed techniques offers excellent flexibility by providing a number of different variables; namely (and most importantly), the regularization parameters, different wavelet families, and a variety of sampling trajectories; all of which can have a role in the quality of both noise reduction, and edge preservation.

All in all, the proposed technique in this thesis promises potential to be utilized as a tool for real-time MRI edge detection; which in turn can be used in a variety of MRI image processing algorithms.

7.2 Future Work

This thesis provides a novel framework for brain MRI edge detection based on a combination of compressed sensing, and multi-resolution properties of wavelets. As such, similar to many new techniques, there remains a gap between the concept of the idea and its practical implementations as part of an MRI image processing tool. A number of steps can be taken to improve and/or modify the existing framework; an explanation of some of such tasks is provided.

As mentioned in Section 5.3.3, at the time of performing the evaluation of the proposed work, real MRI data was not available. Although, most image processing concept designs test their results on phantom images, it is

essential to evaluate the existing algorithm on real brain MRI images, and make any modifications accordingly. In addition, the work provided here concentrates on non-uniform Cartesian (under)sampling for sampling purposes; while providing reasons behind the choice. Although, reasoning behind the choice of Cartesian trajectory is solid, for a complete evaluation of the technique, other trajectories can also be tested; such as spiral, radial, and other non-uniform (incoherent) sampling trajectories; these improvements could also be done in terms of design of a better non-uniform Cartesian pattern. By doing so, one can also extend the concept provided here beyond the application on brain MRI images. Since this research is reliant on a compressed sensing approach for sample acquisition and image reconstruction; the prospect of extension of this work into other areas of MRI imaging could significant implications. By combining a fast method of sample collection with a one step edge detection, this work could be applied to dynamic MRI imaging and three dimensional MRI imaging as well.

It has also been shown that the two most important variables in getting a good quality edge image are the two regularization parameters, λ_{TV} and λ_{Ψ} ; which control data consistency and fidelity. Hence, development a technique to better indicate which values should be used is one of the major challenges at hand; as the current method is an exhaustive one and based on trial based tuning. Moreover, improvement of the evaluation technique is also desirable. Despite the fact that qualitative inspection is an accepted method of evaluation for edge detection algorithms; the current numerical evaluation of the images with SNR does not provide an absolute measure of how good the resulting images are, and is only a relative quantitative approach.

Finally, code optimization for performance (not a topic of interest in this work) is essential to any real-time fast signal or image processing technique. Hence, optimization and implementation of an efficient code is also vital to the performance of this method in a practical setting. With the benefit of improvements, modifications, and extensions, a robust edge detection technique could improve and assist both other image post-processing requirements, and provide better diagnostic techniques; while paving the way for more robust automatic feature extraction approaches.

Appendix A

Additional Processed Images

Figure A.1: Edge detection images from the proposed method for wavelets with different vanishing moments of the Coiflet family.

Figure A.2: Edge detection images from the proposed method for wavelets with different vanishing moments of the Symmlet family.

(a) *daub4* (b) *daub6* (c) *daub8*

Figure A.3: Edge detection images from the proposed method for wavelets with different vanishing moments of the Daubechies family.

99
Figure A.4: The progression of edge detection images from $\lambda_{TV} = 0.05$ to $\lambda_{TV} = 0.60$ with $\lambda_{\Psi} = 0$.

Figure A.5: The progression of edge detection images from $\lambda_{\Psi} = 0.05$ to $\lambda_{\Psi} = 0.60$ with $\lambda_{TV} = 0$.

Figure A.6: A sample of the final iteration of the wavelet transform illustrating different detail images. The top left corner image is the low resolution approximation image.

References

- [1] S. Agaian and A. Almuntashri. Noise-resilient edge detection algorithm for brain mri images. *31st Annual International Conference of the IEEE EMBS*, pages 3689–3692, September 2009.
- [2] P. Akhtar, T. J. Ali, M. I. Bhatti, and M. A. Muqet. A framework for edge detection and linking using wavelets and image fusion. In *Proceedings of the 2008 Congress on Image and Signal Processing*, volume 1, pages 273–277, Washington, DC, USA, 2008. IEEE Computer Society.
- [3] E. F. Badran, E. G. Mahmoud, and N. Hamdy. An algorithm for detecting brain tumors in mri images. *IEEE*, pages 368–373, 2010.
- [4] R. G. Baranuik. Compressive sensing. *IEEE Signal Processing Magazine*, pages 118–124, July 2007.
- [5] D. Baron, M. B. Wakin, M. Duarte, S. Sarvotham, and R. G. Baraniuk. Distributed compressed sensing. Online, 2005.
- [6] P. G. J. Barten. Contrast sensitivity of the human eye and its effects on image quality. *SPIE Press*, 1999.
- [7] A. C. Bovik. *Handbook of Image and Video Processing*. Elsevier, 2005.
- [8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, The Edinburgh Building, Cambridge, CB2 8RU, UK, 2004.
- [9] E. Brannock and M. Weeks. Edge detection using wavelets. In *Proceedings of the 44th annual ACM South-east Regional Conference*, page 649654, March 2006.
- [10] S. C. Bushong. *Magnetic Resonance Imaging: Physical and Biological Principles*. Mosby, St. Louis, Missouri, 2003.
- [11] E. Candes and J. Romberg. Signal recovery from random projections. In *Computational Imaging III*, page 5674, San Jose, 2005. SPIE.
- [12] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling, November 2006.
- [13] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, August 2005.

- [14] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, December 2006.
- [15] E. J. Candes. Compressive sampling. In *Mathematics Subject Classification*, Madrid, Spain, 2006. International Congress of Mathematics, European Mathematical Society.
- [16] J. Canny. A computational approach to edge detection. *IEEE Transactions, Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.
- [17] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *Statistical and Scientific Computing*, pages 33–61, 1999.
- [18] I. F. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics Magazine*, 38(5):826–844, October 1973.
- [19] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.
- [20] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). In *VLDB*, pages 335–345, 2002.
- [21] C.R.Vogel. Non-convergence of the l-curve regularization parameter selection method. *Inverse Problems*, 12:535–547, 1996.
- [22] I. Daubechies. Orthonormal bases for compactly supported wavelets. *Communications on Pure and Applied Mathematics*, pages 909–996, 1988.
- [23] I. Daubechies. *Ten Lectures on Wavelets*. SIAM Press, Philadelphia, PA, 1992.
- [24] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1541, 2004.
- [25] D. Donoho. Compressed sensing. *IEEE Transactions. Information Theory*, 52(4):1289–1306, April 2006.
- [26] D. Donoho, Y. Tsaig, I. Drori, and J. L. Starck. Sparse solution for under-determined linear equations by stagewise orthogonal matching pursuit. *Technical Report*, 2006.
- [27] A. Fabijanska and D. Sankowski. Edge detection in brain images. *MEMSTECH*, pages 60–62, May 2008.
- [28] D. Gabor. Theory of communication. *IEEE*, (93):429–457, November 1946.
- [29] A. Grossman and J. Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Numerical Analysis*, 15(4):723–736, July 1984.
- [30] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, (69):331–371, 1910.
- [31] G. Hennenfent and F. J. Herrmann. Simply denoise: Wavefield reconstruction via jittered under-sampling. *Geophysics*, 73(3), June 2008.

- [32] J. X. Ji, C. Zhao, and T. Lang. Compressed sensing parallel magnetic resonance imaging. *IEEE*, pages 1671–1674, 2008.
- [33] H. Jung, J. C. Ye, and E. Y. Kim. Improved k-t blask and k-t sense using focuss. *Physics in Medicine and Biology*, 52:3201–3226, 2007.
- [34] N. J. Kalton, N. T. Peck, and J. W. James. An f-space sampler. In *London Mathematical Society Lecture Note Series*, 89. Cambridge University Press, 1984.
- [35] D. A. Karras and B. G. Mertzios. Efficient segmentation in mri applying discrete wavelet transform and topology preserving neural networks. In *IEEE IST*, pages 113–118. International Workshop on Imaging Systems and Techniques, May 2005.
- [36] S. J. Kim, K. Koh, M. Lustig, and S. Boyd. An efficient method for compressed sensing. *IEEE*, pages 117–120, 2007.
- [37] J. Li. A wavelet approach to edge detection. Master’s thesis, Sam Houston State University, Huntsville, Texas, August 2003.
- [38] Z. Liang and P. C. Lauterbur. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. IEEE Press, New York, New York, 2000.
- [39] M. Lustig, D. Donoho, and J. M. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine*.
- [40] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, pages 72–82, March 2008.
- [41] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly. k-t sparse: High frame rate dynamic mri exploiting spatio-temporal sparsity. *ISMRM*, May 2006.
- [42] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $\mathcal{L}^2(\mathbb{R})$. *Transaction of American Mathematical Society*, 315(1):69–87, September 1989.
- [43] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674693, July 1989.
- [44] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier Inc, 30 Corporate Drive, Suite 400, Burlington, Massachusetts 01803, third edition, 2009.
- [45] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions Information Theory*, 38(2):617–643, 1992.
- [46] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society of London, B, Biological Sciences*, pages 187–217. The Royal Society, February 1980.

- [47] F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):626–630, 1985.
- [48] D. G. Mitchell and M. S. Cohen. *MRI Principles*. Elsevier, Philadelphia, Pennsylvania, 2004.
- [49] M. S. Nixon and A. S. Aguado. *Feature Extraction and Image Processing*. Newnes, 225 Wildwood Avenue, Woburn, MA 01801-2041, first edition, 2002.
- [50] L. parsad and S. S. Iyengar. *Wavele Analysis with Applications to Image Processing*. CRC Press, 1997.
- [51] J. L. Prince and J. M. Links. *Medical Imaging Signals and Systems*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2006.
- [52] Z. Y. Qian, G. W. Hua, C. Z. Cheng, T. J. Tian, and L. L. Yun. Medical images edge detection based on mathematical morphology. In *Proceedings of the 2005 IEEE, Engineering in Medicine and Biology*, Shanghai, China, September 2005.
- [53] X. Qu, X. Cao, D. Guo, C. Hu, and Z. Chen. Combined sparsifying transforms for compressed sensing mri. 46(2), January 2010. Electronic Letters.
- [54] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Inc, Hoboken, New Jersey, fourth edition, 2009.
- [55] L. Rudin, S. Osher, and E. Fatemi. Non-linear total variation noise removal algorithm. *Physics D: Applied Physics*, 60:259–268, November 1992.
- [56] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *Scientific and Statistical Computing*, 7(4):1307–1330, 1986.
- [57] M. Sharifi, M. Fathy, and M. T. Mahmoudi. A classified and comparative study of edge detection algorithms. In *Proceedings of the International Conference on Information Technology: Coding and Computing*. IEEE, 2002.
- [58] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 14(17):710–732, 1992.
- [59] M. J. Smith and T. P. Barnwell III. A procedure for designing exact reconstruction filter banks for tree structured sub-band coders. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1984.
- [60] C. M. Tsai and D. Nishimura. Reduced aliasing artifacts using variable density k-space sampling trajectories. *Magnetic Resonance in Medicine*, 43(3):452–458, 2000.
- [61] Y. Tsaig and D. Donoho. Extensions of compressive sensing. 2004.
- [62] S. Udomhunsakul and P. Wongsita. Feature extraction in medical mri images. pages 340–344. *Cybernetics and Intelligent Systems*, IEEE, 2004.

-
- [63] M. Vetterli. Splitting a signal into subsampled channels allowing perfect reconstruction. In *Proceedings of the IASTED Conference on Applications of Signal Processing and Digital Filtering*, June 1985.
- [64] H. Wang, D. Liang, and L. Ying. Pseudo 2d random sampling for compressed sensing mri. pages 2672–2675, Minneapolis, Minnesota, USA, September 2009. 31st Annual International Conference of the IEEE EMBS, IEEE.
- [65] S. Wang, F. Ge, and T. Liu. Evaluating edge detection through boundary detection. *EURASIP Journal on Applied Signal Processing*, pages 1–15, 2006.
- [66] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it?: A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, pages 98–117, January 2009.
- [67] Y. Zhang, Z. Dong, L. Wu, S. Wang, and Z. Zhou. Feature extraction of brain mri by stationary wavelet transform. 2010.