

Accelerating Magnetic Resonance Imaging by Unifying Sparse Models and Multiple Receivers

by

Daniel S. Weller

B.S., Carnegie Mellon University (2006)

S.M., Massachusetts Institute of Technology (2008)

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

© 2012 Massachusetts Institute of Technology. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
March 30, 2012

Certified by.....
Vivek K. Goyal
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by.....
Professor Leslie A. Kolodziejwski
Chair, Committee on Graduate Students

Accelerating Magnetic Resonance Imaging by Unifying Sparse Models and Multiple Receivers

by

Daniel S. Weller

Submitted to the Department of Electrical Engineering and Computer Science
on March 30, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Abstract

Magnetic resonance imaging (MRI) is an increasingly versatile diagnostic tool for a variety of medical purposes. During a conventional MRI scan, samples are acquired along a trajectory in the spatial Fourier transform domain (called k-space) and the image is reconstructed using an inverse discrete Fourier transform. The affordability, availability, and applications of MRI remain limited by the time required to sample enough points of k-space for the desired field of view (FOV), resolution, and signal-to-noise ratio (SNR). GRAPPA, an accelerated parallel imaging method, and compressed sensing (CS) have been successfully employed to accelerate the acquisition process by reducing the number of k-space samples required. GRAPPA leverages the different spatial weightings of each receiver coil to undo the aliasing from the reduction in FOV induced by undersampling k-space. However, accelerated parallel imaging reconstruction methods like GRAPPA amplify the noise present in the data, reducing the SNR by a factor greater than that due to only the level of undersampling. Completely separate from accelerated parallel imaging, which capitalizes on observing data with multiple receivers, CS leverages the sparsity of the object along with incoherent sampling and nonlinear reconstruction algorithms to recover the image from fewer samples. In contrast to parallel imaging, CS actually denoises the result, because noise typically is not sparse. When reconstructing brain images, the discrete wavelet transform and finite differences are effective in producing an approximately sparse representation of the image. Because parallel imaging utilizes the multiple receiver coils and CS takes advantage of the sparsity of the image itself, these methods are complementary, and a combination of these methods would be expected to enable further acceleration beyond what is achievable using parallel imaging or CS alone.

This thesis investigates three approaches to leveraging both multiple receiver coils and image sparsity. The first approach involves an optimization framework for jointly optimizing the fidelity to the GRAPPA result and the sparsity of the image. This technique operates in the nullspace of the data observation matrix, preserving the acquired data without resorting to techniques for constrained optimization. While this framework is presented generally, the effectiveness of the implementation depends on

the choice of sparsifying transform, sparsity penalty function, and undersampling pattern. The second approach involves modifying the kernel estimation step of GRAPPA to promote sparsity in the reconstructed image and mitigate the noise amplification typically encountered with parallel imaging. The third approach involves imposing a sparsity prior on the coil images and estimating the full k-space from the observations using Bayesian techniques. This third method is extended to jointly estimate the GRAPPA kernel weights and the full k-space together. These approaches represent different frameworks for accelerating MRI imaging beyond current methods. The results presented suggest that these practical reconstruction and post-processing methods allow for greater acceleration with conventional Cartesian acquisitions.

Thesis Supervisor: Vivek K. Goyal

Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

My deepest thanks go to my parents, Jay and Leslie, and my brother, Brian, for being so supportive of me. From an early age, my parents encouraged me to be inquisitive and nurtured my passion for math and science. Good-natured competition with my brother helped prepare me for a demanding academic career. It is to my family that I dedicate this thesis.

There are so many here at MIT that made my experience here fulfilling. Vivek and the members of the STIR group, past and present, provided countless hours of advice, support, and encouragement. Julius, who finished his PhD in the group around the same time that I arrived in 2006, proved on my earliest days of graduate school that one can succeed through hard work. I am grateful to my fellow students, Adam, Andrea, Ha, John, Joong, Kirmani, Lav, Mike, Vahid, and Vinith, and the group's many visitors, including Aniruddha, Aycan, BJ, Götz, José, Pier Luigi, Ulugbek, and Woohyuk, for all the stimulating discussions and meetings over the last six years and just generally sticking together and making research so much fun!

Similarly, Elfar's MRI group, including Audrey, Berkin, Borjan, Div, Filiz, Jessica, Joonsung, Lohith, Obaidah, Shao Ying, and Trina, and my collaborators at MGH, including Fa-Hsuan, Larry, Jon, Kawin, Kyoko, and Thomas, were wonderful colleagues and friends who made available their vast knowledge to help me realize this thesis. Additionally, I express my sincerest gratitude to Leo at Siemens Corporate Research, for enthusiastically supporting my research, despite the hundreds of miles that separated us.

Many student groups influenced me during my time here. My colleagues and friends in the EECS GSA and in the Sidney Pacific House Council provided endless fun and enjoyment throughout my time here. I also enjoyed my involvement in the Graduate Student Council, and I thank Gerald, Leeland, and Alex for their encouragement. I cherish my friendships with Dennis, Tom, and others in the DSP Group, and playing tennis and games with Da, Ligong, and others from the sixth floor has been an excellent diversion.

I thank Professor Dahleh and Professor Oppenheim for their advice and support through all these years. Professor Dahleh helped guide me through my studies here at MIT and ensured that I did not overlook any of the requirements. Professor Oppenheim organized a graduate mentoring seminar for new students in my first year, including me, and helped cultivate a collegial atmosphere that made MIT a much warmer and less intimidating place. I also appreciated teaching with Professor Oppenheim, and I greatly enjoyed the experience.

Of course, I also want to thank my friends around MIT and beyond for keeping me grounded and sane. If not for playing board games or tennis, attending Celtics, Bruins, or Red Sox games, or just going out and having fun together, life beyond research would have been quite dull.

Last, but not least, I extend my utmost gratitude to my thesis committee, including my supervisor Vivek, and my readers Elfar, Pablo, Larry, and Leo. While not an official member of my thesis committee, Jon also helped extensively to supervise

my research. My experience would not have been nearly so satisfying without your input, comments, and mentoring. The administrative staff, namely Eric, Kathryn, and Gabrielle in STIR, Arlene in the MRI group, and the technologists and support staff at MGH, also played critical roles throughout the research process. Thank you for your assistance, support, and friendship.

On a less personal note, this research was supported by NSF CAREER Award CCF-0643836, Siemens Healthcare, NIH R01 EB007942 and EB006847, NIH NCCR P41 RR014075, and NDSEG and NSF Graduate Research fellowships.

Daniel S. Weller
Cambridge, MA, USA

Contents

1	Introduction	17
1.1	Outline	20
1.2	Bibliographical Notes	23
2	Magnetic Resonance Imaging	25
2.1	MR Physics	25
2.2	Cartesian MR Imaging	29
2.3	Accelerated MR Imaging	33
2.4	Parallel MR Imaging	34
2.5	Accelerated Parallel Imaging Reconstruction	38
3	Sparsity and Compressed Sensing	47
3.1	Measures of Sparsity	48
3.2	Sparsity-Based Denoising	52
3.3	Compressed Sensing Reconstruction	55
3.4	Compressed Sensing for MRI	59
4	Denoising GRAPPA with Sparsity	63
4.1	Theory	64
4.2	Simulations and Results	69
4.3	Discussion	87
5	GRAPPA Kernel Calibration with Sparsity	91
5.1	Theory	92

5.2	Simulations and Results	97
5.3	Discussion	105
6	Estimation Using GRAPPA and Sparsity	109
6.1	Theory	110
6.2	Simulations and Results	118
6.3	Discussion	121
7	Conclusion	125
A	Optimization Methods	131
A.1	Least-Squares Problems	131
A.2	Compressed Sensing Problems	136

List of Figures

2.1	While the magnetization \mathbf{M} is at an angle to the magnetic field $B_0\hat{\mathbf{k}}$, the derivative of the magnetization $d\mathbf{M}/dt$ is perpendicular to \mathbf{M} , causing the magnetization vector to precess around the magnetic field.	27
2.2	RF slice-selective excitation (in z -direction) followed by Cartesian sampling of k -space in k_xk_y -plane using x - and y -gradients. The acquisition is repeated for different magnitudes of G_y , resulting in the sampling of different phase encode scan lines, shown in the k -space plot on the right. The samples are taken during the frequency encoding x -gradient and are marked with the ADC.	31
2.3	The sample spacing Δk_x in k -space and the extent $k_{x,\max}$ relate to the FOV FOV_x and voxel size Δx , respectively, of the reconstructed image in the x -direction. Similarly, Δk_y and $k_{y,\max}$ are connected to FOV_y and Δy	32
2.4	Uniform undersampling results in coherent aliasing, while non-uniform or random sampling results in incoherent aliasing. Images shown for k -space undersampled by a factor of 4, with zero-filling reconstruction.	34
2.5	This 96-channel head array coil prototype has many small coils (metal rings) around the head enclosure. Each coil has its own data acquisition hardware, so all the channels can be acquired simultaneously. Commercially available array coils enable parallel imaging to be used for many MRI applications.	35

2.6	Block diagram of the observation model for an MRI acquisition with a P -channel parallel receive array coil. The sensed magnetizations $M_1(\mathbf{r}), \dots, M_P(\mathbf{r})$ all derive from the object magnetization $M(\mathbf{r})$. The k-space observations $y_1[\mathbf{k}], \dots, y_P[\mathbf{k}]$ are generated simultaneously from these sensed magnetizations.	36
2.7	Magnitude coil sensitivities for 32-channel head coil array computed from acquired data using 32-channel array and single-channel body coils.	37
2.8	Real and imaginary parts of SMASH approximations (top) to complex exponentials (bottom) using a least-squares fit with empirical sensitivities of a 32-channel head coil receive array.	41
3.1	Unit balls are shown in two dimensions for the ℓ_0, ℓ_1 , and ℓ_p measures. Note that the two lines that form the two-dimensional unit ball for the ℓ_0 “norm” actually extend to $\pm\infty$ and exclude the origin.	49
3.2	The ℓ_1 norm is plotted with ℓ_p^p “norms” for different values of p ($0 < p < 1$). The ℓ_0 “norm” is included for comparison.	49
3.3	The Cauchy penalty function is plotted for different values of α . The ℓ_0 “norm” is included for comparison.	50
3.4	K-space undersampling patterns and their point spread functions. . .	60
4.1	GRAPPA reconstructions for 2-D uniformly spaced (a) 3×3 , (b) 4×4 , (c) 5×5 , and (d) 6×6 nominal undersampling with increasing total acceleration factor R and a 32-channel coil at 3 T.	63
4.2	Ground truth magnitude images of real ((a)-(c)) and synthetic ((d)-(f)) datasets. Inset regions (white rectangles) are used later to show detail.	70
4.3	Sparsity model cdf’s for the transform coefficient magnitudes of a Shepp-Logan phantom in the finite differences domain.	73
4.4	Sparsity model cdf’s for the transform coefficient magnitudes of a T_1 -weighted real image in the four-level ‘9-7’ DWT domain.	73

4.5	Continuation scheme parameters studied for DESIGN denoising of T_1 -weighted image #1 (4×4 nominal undersampling: $R = 10.5$) using the Cauchy penalty function with four-level ‘9-7’ DWT.	74
4.6	DESIGN denoising of Shepp-Logan phantom (4×4 nominal undersampling: $R = 8.7$) using ℓ_1 , ℓ_p^p , and Cauchy penalty functions with the finite differences representation.	75
4.7	DESIGN denoising of T_1 -weighted image (4×4 nominal undersampling: $R = 10.5$) using ℓ_1 , ℓ_p^p , and Cauchy penalty functions with the four-level ‘9-7’ DWT.	75
4.8	Trend in the optimal choice of λ as determined by coarse-then-fine parameter sweeps for the ℓ_1 norm and Cauchy penalty functions, for the T_1 -weighted image #1 with four-level ‘9-7’ DWT sparsifying transform. The nominal undersampling increases from 3×3 ($R = 7.0$) to 6×6 ($R = 16.3$).	76
4.9	Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L_1 SPIRiT for $R = 10.5$ undersampled T_1 -weighted image #1 with four level ‘9-7’ DWT sparsifying transform.	78
4.10	Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L_1 SPIRiT for $R = 10.5$ undersampled T_1 -weighted image #2 with four level ‘9-7’ DWT sparsifying transform.	79

4.11	Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L ₁ SPIRiT for $R = 14.6$ undersampled T_2 -weighted image with four level ‘9-7’ DWT sparsifying transform.	79
4.12	Estimated g-factors (in dB) are plotted for T_1 -weighted image #1 with $R = 10.5$ acceleration factor (4×4 nominal undersampling).	82
4.13	Synthetic contrast phantom reconstructions using GRAPPA and denoising methods (4×4 nominal undersampling: $R = 12.1$) with a four-level ‘9-7’ DWT sparsifying transform.	83
4.14	Synthetic resolution phantom reconstruction comparisons of effective spatial resolutions using GRAPPA and denoising methods (4×4 nominal undersampling: $R = 12.1$) with a four-level ‘9-7’ DWT sparsifying transform.	84
4.15	GRAPPA and DESIGN denoising methods compared for uniform and random undersampled (4×4 nominal undersampling: $R = 8.7$) Shepp-Logan phantom with finite differences sparsifying transform.	86
4.16	GRAPPA and DESIGN denoising methods compared for uniform and Poisson disc random undersampled (5×5 nominal undersampling: $R = 13.7$) T_1 -weighted MPRAGE dataset with four-level ‘9-7’ DWT sparsifying transform.	88
5.1	GRAPPA reconstructions with low-quality kernel calibrations demonstrating noise amplification (left) and residual aliasing (right) with 4×4 nominal undersampling. The GRAPPA reconstruction on the left was calibrated with 36×36 ACS lines and no regularization, and the reconstruction on the right was calibrated with 20×20 ACS lines (underdetermined) with Tikhonov regularization.	92

5.2	GRAPPA reconstructions of T_1 -weighted image #1 (4×4 nominal undersampling: $R = 10.5$) with high-quality kernel calibrations with no regularization, Tikhonov regularization, and sparsity-promoting regularization.	99
5.3	GRAPPA reconstructions of T_1 -weighted images #1 and #2 (4×4 nominal undersampling: both $R = 10.5$) with low-quality kernel calibrations with no regularization, Tikhonov regularization, and sparsity-promoting regularization.	100
5.4	GRAPPA reconstructions of 4×4 nominally undersampled T_1 -weighted image #1 ($R = 13.7$) and image #2 ($R = 12.9$) with underdetermined kernel calibrations with Tikhonov regularization and sparsity-promoting regularization.	101
5.5	Trade-off curves for un-regularized, Tikhonov-regularized, and sparsity-promoting GRAPPA kernel calibration depicting the relationship between reconstructed image PSNR and total acceleration R as the number of ACS lines is varied. Nominal undersampling is held fixed at 4×4 .	103
5.6	Trade-off curves for un-regularized, Tikhonov-regularized, and sparsity-promoting GRAPPA kernel calibration depicting the relationship between reconstructed image PSNR and total acceleration R as the total acceleration is varied (by varying both nominal undersampling and the number of ACS lines).	104
5.7	GRAPPA and DESIGN-denoised reconstructions of T_1 -weighted image #2 (4×4 nominal undersampling: $R = 12.9$) with underdetermined sparsity-promoting kernel calibration.	105
5.8	GRAPPA with un-regularized and sparsity-promoting calibration and DESIGN-denoised GRAPPA with sparsity-promoting calibration of T_1 -weighted image #2 (4×4 nominal undersampling: $R = 10.5$) with 36×36 ACS lines.	106

6.1	Reconstructed and difference images using conventional GRAPPA and Bayesian full k-space estimation for T_1 -weighted image #2 for several acceleration factors (nominal undersampling increases from 4×3 to 5×5).	120
6.2	Reconstructed and difference images using conventional GRAPPA and Bayesian joint estimation of the kernel and full k-space for T_1 -weighted image #2 at higher acceleration factors (nominal undersampling 5×4 and 5×5 for the top and bottom rows, respectively).	121
6.3	Reconstructed and difference images using conventional GRAPPA and Bayesian joint estimation of the kernel and full k-space for T_1 -weighted image #2 with a larger GRAPPA kernel (nominal undersampling is 5×4).	122
A.1	Steepest descent (dotted line) and conjugate gradient (solid line) methods compared for $\mathbf{A} = [\frac{1}{4}, 0; 0, 1]$ and $\mathbf{b} = [0; 0]$	135

List of Tables

4.1	Penalty functions and associated sparsity priors on transform coefficient magnitudes.	72
4.2	PSNRs (in dB) of reconstruction methods at different acceleration factors for T_1 -weighted image #1.	80
4.3	PSNRs (in dB) of reconstruction methods at different acceleration factors for T_1 -weighted image #2.	80
4.4	PSNRs (in dB) of reconstruction methods at different acceleration factors for T_2 -weighted image.	80
A.1	Weights for half-quadratic minimization.	138

Chapter 1

Introduction

Since its development in the 1970s, magnetic resonance imaging (MRI) has steadily gained in importance to clinicians and researchers for its ability to produce high quality images non-invasively without the side effects of ionizing X-ray radiation. MRI is used extensively to image soft tissue throughout the whole body [22]. Moreover, magnetic resonance imaging can be used to distinguish gray and white matter in the brain, observe blood flow, and measure diagnostically valuable quantities such as cortical thickness [13, 47, 32]. Because of its great versatility, MRI has myriad applications in both medical research and diagnostic and perioperative clinical imaging. However, magnetic resonance imaging remains limited by the time required to generate these images. A typical MRI of a brain can take between five and ten minutes, during which the subject must remain perfectly still. This requirement is a hardship for certain populations like young children, the elderly, and patients experiencing chronic or acute pain. Since many MRI bores are narrow enclosed spaces, subjects may experience claustrophobia, making remaining motionless more difficult. Because multiple scans are typical for many applications, sessions commonly extend beyond one hour in duration, increasing costs and reducing availability of the scanner. In addition, compromises in image quality such as resolution reduction are necessary for time-critical applications like functional MRI [71, 6].

MRI acquisition speed is limited by physiological constraints connected to the effects of spatially varying magnetic fields on the body. A spatially-varying applied

magnetic field can induce currents in the nervous system; at high enough rates, these currents can stimulate the nerves, irritating or distressing the subject [40]. As the fields used to encode the spatial information for Fourier coefficients are spatially varying, this constraint essentially limits the rate we can collect MRI data. Past efforts in accelerating MRI have centered upon adjusting the sampling pattern or acquiring multiple samples simultaneously. All these methods have their advantages and disadvantages. Adjusting the sampling pattern often means reducing resolution, losing phase information, or requiring more complicated reconstruction methods [9, 70]. Fast MRI acquisition techniques also can use multiple echoes to reduce imaging time while reducing contrast or increasing susceptibility to magnetic field inhomogeneity [62, 41, 30].

A different approach for accelerating MRI uses multiple receivers in parallel and post-processing to recover complete images from fewer samples. Parallel imaging had already been used effectively to mitigate noise, and now, accelerated parallel imaging methods also enable faster acquisitions [82, 87, 76, 38]. Whereas conventional receiver coils have a single channel with spatially uniform sensitivity to magnetization, parallel receiver coils have multiple channels with different non-uniform magnetic sensitivities [82]. Accelerated parallel imaging reconstruction methods use the different sensitivities of the coil channels to resolve the ambiguity due to undersampling [76]. Such methods already are popular in commercial scanners, enabling modest levels of acceleration for many types of imaging, but these methods alone are insufficient for the high acceleration levels we would like to attain.

Another technique for reconstructing images from undersampled data called compressed sensing (CS) emerged in the signal processing community [18, 16, 20, 27]. Compressed sensing takes advantage of the sparsity or compressibility of an appropriate representation or transform of the desired image. While not specific to MRI, MRI is a widely suitable candidate for CS due to the approximate transform sparsity of many MR images and the ability to use nearly arbitrary (random) sampling patterns [57]. For instance, many MR images have few edges or have simple textures representable using a small number of wavelet coefficients. CS has enabled successful

reconstructions of modestly accelerated MRI data [57].

By combining the sparsity models with the accelerated parallel imaging reconstruction methods already developed, we expect high quality reconstructions from data collected with even greater undersampling. Linear system inversion techniques for accelerated parallel imaging reconstruction like SENSE [76] and SPIRiT [59] can be directly combined with the compressed sensing reconstruction framework. Methods like SparseSENSE [52] and L_1 SPIRiT [56] follow this approach, yielding a sparsity-promoting regularized reconstruction method that can recover high quality images from moderate accelerations with random undersampling.

With conventional uniform undersampling, we aim to improve the auto-calibrating kernel-based interpolation method GRAPPA [38]. As a direct method (not an inversion), this reconstruction approach cannot be directly incorporated into a compressed sensing framework. Further complicating the combination with sparse models is the two-step formulation of the GRAPPA method: both the calibration and interpolation steps influence the reconstruction quality and can introduce noise or artifacts. Also, while theoretical results concerning compressed sensing rely on a random or pseudo-random observation matrix, the observations are uniformly spaced, yielding coherent aliasing that cannot be distinguished based on sparsity alone. In this work, we study three different approaches to improving GRAPPA using sparsity models: denoising the reconstructed image, regularizing the calibration step, and estimating the channel images using Bayesian sparsity models.

We demonstrate that all these approaches successfully extend the GRAPPA accelerated parallel imaging method to higher accelerations by having either greater spacing between samples, or less calibration data, and yielding high quality images. The denoising method reduces the noise amplification from both undersampling the data and the GRAPPA reconstruction process. The improved calibration method reduces the amount of calibration data needed to produce a quality GRAPPA reconstruction, mitigating the aliasing and noise that would otherwise result. The joint estimation method combines these ideas to reconstruct both the GRAPPA kernel needed for interpolation and the denoised full channel images from the undersampled

data, enabling reconstructions from highly undersampled data with less calibration data. Results using real MRI data are presented that portray the effectiveness of these methods relative to conventional accelerated parallel imaging at high levels of acceleration. We conclude from these results that significant gains in both image quality and total acceleration can be made using all three of these methods, enabling much faster MRI scans with currently employed image acquisition paradigms.

1.1 Outline

Effective combination of accelerated parallel imaging methods with compressed sensing requires an in-depth understanding of the advantages and drawbacks of each. Keeping in mind the strengths and weaknesses of these methods, we propose three distinct approaches to combining GRAPPA, a widely-used accelerated parallel imaging method, and image sparsity. We introduce a denoising method that mitigates noise amplification at moderate levels of acceleration. We also propose a sparsity-promoting auto-calibration method for GRAPPA that enables significantly greater acceleration by reducing the amount of calibration data needed. Finally, we consider a Bayesian estimation-theoretic framework for jointly calibrating the GRAPPA reconstruction method and reconstructing denoised full images from undersampled data. We conclude this thesis with a discussion of the merits and drawbacks of the proposed methods and their respective places in practical accelerated imaging.

Background on magnetic resonance imaging is presented in Chapter 2. We begin with a basic discussion of MR physics, emphasizing the classical aspects leading up to the signal equation, which describes the connection between the magnetic moments, fields, and the measured received signal. Connections between the sampling of k -space and the spatial resolution and field of view of the image are drawn. Methods for accelerating MRI acquisition, including partial Fourier imaging and fast pulse echo sequences are described, ending with an introduction to accelerated parallel imaging. Techniques for combining coil images and measuring coil sensitivities are presented, and important pre-existing accelerated parallel imaging reconstruction techniques are

described in detail. The nullspace formulation of SPIRiT is presented as an example of this constrained optimization method that proves useful later.

In Chapter 3, sparsity models and the compressed sensing framework are described in detail. The notions of sparsity, transform sparsity, and compressibility are developed, and measures of sparsity, including the ℓ_0 , ℓ_1 , and ℓ_p^p measures, are presented. An introduction to joint and group sparsity and appropriate hybrid measures follows. Linear and nonlinear methods for sparsity-based denoising are introduced and extended to the joint sparsity case. Once these preliminaries are complete, the compressed sensing framework is developed, and key theoretical concepts like the restricted isometry property and mutual coherence are explained. Compressed sensing is then applied to the problem of reconstructing MRI images from undersampled data, and major results from the literature are described. Additional time is spent depicting sampling patterns for compressed sensing MRI used in the literature. This chapter concludes with a discussion of the literature combining compressed sensing with existing accelerated parallel imaging reconstruction methods and how these methods differ from the contributions in this thesis.

As mentioned earlier, three approaches for improving GRAPPA accelerated parallel imaging using sparsity models are proposed. The first approach, denoising the GRAPPA result using sparsity, is described in Chapter 4. Motivating this development is the preponderance of noise present in GRAPPA reconstructed images at high accelerations. The proposed method aims to reduce the noise to a more acceptable level by adjusting the interpolated (missing) k-space frequencies to promote the joint transform sparsity of the coil images. A few innovations are made: the nullspace method is applied to preserve the acquired data while denoising the coil images; the GRAPPA result is used directly, saving on computation; and the method is developed with the explicit goal of denoising, not requiring any deviation from conventional uniform undersampling. The complete method also considers the contribution of each voxel in each coil channel to the final combined image, allowing for greater deviation from the GRAPPA reconstruction in those voxels deemed too noisy or too insignificant in the combined image.

A variety of studies are performed on real and simulated data using this denoising method. Interpreting the choice of sparsity-promoting regularization penalty as imposing a prior distribution on the sparse transform coefficients, the empirical cumulative distribution function (cdf) of the combined reference image transform coefficients is compared to the distributions for a variety of penalty functions, and denoising is performed using all these penalties to visualize the effects of imposing an appropriate prior on the denoised image. First performed for the Shepp-Logan phantom, this experiment is repeated for real MRI data.

Additional experiments depict the impact of continuation scheme parameters and the tuning parameter on the denoising quality. A series of comparisons are performed to portray differences in image quality, noise suppression, and contrast/resolution degradation among the proposed method and existing reconstruction and denoising methods. The chapter concludes with a depiction of denoising adapted to different sampling patterns and a discussion of the advantages and disadvantages of the proposed method that can be inferred from these experiments.

A second approach utilizes sparsity to regularize the GRAPPA kernel calibration step. In Chapter 5, this improved calibration method is derived and compared to un-regularized and conventionally regularized kernel calibration. Using real MRI reference images, reconstructions are performed using different numbers of ACS calibration data, and the impact of different kinds of regularization is portrayed in these experiments. Since varying the number of ACS lines can be interpreted as trading image quality for greater total acceleration, the trade-off curves for these different calibration methods are plotted, and the improvement in the achievable trade-off region is significant. This chapter ends with experiments depicting the additional improvement from post-processing the regularized GRAPPA method with the denoising method proposed in the previous chapter. From the improvement visible in these last experiments, we speculate that additional gains are possible from combining the calibration and reconstruction/denoising steps when regularizing with sparsity.

We investigate this combination of calibrating the GRAPPA kernel and reconstructing the full k-space in Chapter 6. We begin by formulating a Bayesian esti-

mation problem using both the acquired data and the GRAPPA reconstructions as observations (with different noise models) and treating the joint transform sparsity as a prior distribution on the full k-space across all the coil channels. After deriving the posterior-maximizing estimator for this problem, we consider how the estimation problem changes when the GRAPPA kernel is a variable. The transformed problem enables joint estimation of both the kernel and the full coil-by-coil k-space from the acquired data (including ACS lines). This problem is solved by adapting the iterative algorithms used in previous chapters to compute the denoised full k-space and regularized GRAPPA kernel. Experiments on real data depict significant improvements in image quality at very high accelerations, even when using relatively little calibration data. From these experiments, we conclude that this joint estimation method, by combining the effects of sparsity models on the calibrated kernel and on the full k-space, enables high quality imaging from even less data than before.

In Chapter 7, we conclude by summarizing the conclusions and contributions made in these chapters, and we follow this summary by a discussion of the impact on the field and what future directions may enable even greater improvements in accelerated MR imaging.

1.2 Bibliographical Notes

Parts of Chapter 4 appear in papers:

- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Denoising sparse images from GRAPPA using the nullspace method (DESIGN). *Magn. Reson. Med.*, to appear (available online, DOI: 10.1002/mrm.24116). PubMed Central PMID: 22213069.
- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Combined compressed sensing and parallel MRI compared for uniform and random Cartesian undersampling of k-space. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, pages 553-6, May 2011.

- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. SpRING: Sparse reconstruction of images using the nullspace method and GRAPPA. In *Proc. ISMRM 19th Scientific Meeting*, page 2861, May 2011.
- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Evaluating sparsity penalty functions for combined compressed sensing and parallel MRI. In *Proc. IEEE Int. Symp. on Biomedical Imaging*, pages 1589-92, March-April 2011.
- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Combining nonconvex compressed sensing and GRAPPA using the nullspace method. In *Proc. ISMRM 18th Scientific Meeting*, page 4880, May 2010.

Parts of Chapter 5 appear in papers:

- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Greater Acceleration through Sparsity-Promoting GRAPPA Kernel Calibration. In *Proc. ISMRM 20th Scientific Meeting*, to appear.
- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Regularizing grappa using simultaneous sparsity to recover denoised images. In *Proc. SPIE Wavelets and Sparsity XIV*, volume 8138, pages 81381M-1-9, Aug. 2011.

Parts of Chapter 6 appear in the manuscript:

- D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Accelerated Parallel Magnetic Resonance Imaging Reconstruction Using Joint Estimation with a Sparse Signal Model. In preparation.

Chapter 2

Magnetic Resonance Imaging

A thorough understanding of magnetic resonance imaging (MRI) begins with MR physics, namely the interactions between bulk material and magnetic fields, and the resulting signal received by nearby coils. These concepts can be employed to acquire 2-D or 3-D images that depict the spatial distribution of magnetically susceptible materials, including biological tissue. An acquisition executes a specific spatial frequency domain sampling pattern, the properties of which are connected to the voxel size and field of view of the reconstructed image. Several approaches for accelerating MRI within this framework also are described here. Conventional imaging is extended to parallel imaging using multiple receiver coils; existing techniques for reconstructing images from accelerated parallel imaging data are explained and compared, including the GRAPPA method, which is used extensively throughout this thesis.

2.1 MR Physics

The basic classical theory underlying MRI derives from the physics governing the interaction between particles in a bulk material and an externally applied magnetic field. These physical laws also govern detection of the magnetization of these particles and allow us to reconstruct an image of the magnetic properties of the bulk material. A concise, thorough treatment of these concepts is given in [68]. A summary of pertinent information from this reference is provided here.

2.1.1 Magnetic Moments

At a high level, MRI involves exciting particles in the test subject using a combination of several external magnetic fields and measuring in a nearby receiver coil the resulting signal generated by those particles. Atoms with an odd number of protons or neutrons have “spin,” which can be affected by an applied magnetic field. The most prevalent such particle in the human body is the single-proton hydrogen (^1H) atom found in both water and hydrocarbons, especially lipids. This abundance is fortunate as the (^1H) atom is highly sensitive to applied magnetic fields, so it produces a strong signal that is relatively easily detected. While it is convenient to think of individual atoms in isolation, the structure of the molecule or compound containing these magnetically susceptible atoms impacts the received signal. Since different tissue types contain different densities of different hydrogen-containing molecules, these signal differences create contrast between tissue types useful for generating useful images depicting anatomy or structure.

To consider the effect of a magnetic field on a susceptible particle, it is helpful to consider the “spin” as a vector quantity, called the magnetic moment. In a bulk material such as biological tissue, this vector is often expressed in terms of magnetization \mathbf{M} , the net magnetic moment per unit volume. The effect of a magnetic field \mathbf{B} on the magnetization is described by the differential equation

$$\frac{d\mathbf{M}}{dt} = \mathbf{M} \times \gamma\mathbf{B}, \quad (2.1)$$

where γ is the gyromagnetic ratio of the particle ($\gamma = 2\pi \cdot 4.2576 \cdot 10^7$ rad/s/T for the hydrogen atom [7]). In the presence of a sufficiently strong static magnetic field, such as the main field generated by a permanent or superconducting magnet in an MRI machine, these spins in equilibrium tend to be oriented in the direction of that magnetic field. In keeping with convention, we consider the main field to point in the z -direction of our right-handed 3-D coordinate system. In the context of human MRI, the main field is oriented parallel to the bore of the MRI system.

When the magnetization is at an angle to the main field, the magnetization vector

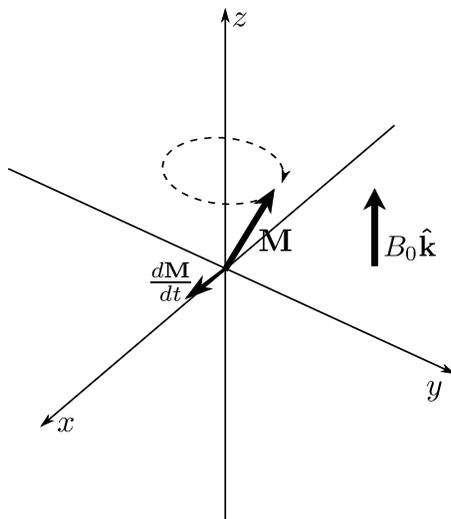


Figure 2.1: While the magnetization \mathbf{M} is at an angle to the magnetic field $B_0 \hat{\mathbf{k}}$, the derivative of the magnetization $d\mathbf{M}/dt$ is perpendicular to \mathbf{M} , causing the magnetization vector to precess around the magnetic field.

will precess in the plane normal to that field (called the transverse plane or axial plane) at the frequency $\omega = \gamma B$; when \mathbf{B} is the main field $B_0 \hat{\mathbf{k}}$, this frequency $\omega_0 = \gamma B_0$ is called the Larmor frequency. For a main field strength of $B_0 = 3$ T, the Larmor frequency of a hydrogen atom is $\omega_0 = 127.73$ MHz. This precession behavior is depicted in Figure 2.1.

To simplify later calculations, we often express Equation (2.1) in the reference frame rotating at the Larmor frequency in the transverse plane. In the rotating reference frame,

$$\frac{d\mathbf{M}}{dt} = \mathbf{M} \times (\gamma \mathbf{B} - \omega_0 \hat{\mathbf{k}}). \quad (2.2)$$

When $\mathbf{B} = B_0 \hat{\mathbf{k}}$, we have $\frac{d\mathbf{M}}{dt} = \mathbf{0}$. Thus, in the rotating reference frame, we can ignore the contribution of the main field when considering the effects of other external magnetic fields on the magnetization.

While the spins are precessing, they induce an electromotive force (emf) in a nearby receiver coil. The observed signal from all the spins can be approximated by integrating the transverse magnetization over the entire volume. For convenience, we write the transverse magnetization M_{xy} as a complex number with the real part representing the component in the x -direction, and the imaginary part representing

the component in the y -direction. So $M_{xy} = M_x + jM_y$. Thus, although the magnetization vector is real-valued, our measurements will be complex-valued to capture both components of the transverse magnetization conveniently. The phase of this complex-valued quantity contains information about the actual precession frequencies of the spins, which can be used to study chemical structure or composition, main field inhomogeneity, and (as we will use later) spatial location of the spins.

2.1.2 Relaxation and Excitation

While precessing, these moment vectors also move towards equilibrium in a process called relaxation. This relaxation occurs in two ways: by losing magnitude in the transverse plane, which is called transverse or T_2 relaxation, and by gaining magnitude in the main field direction, which is termed longitudinal or T_1 relaxation. Both relaxation processes are modeled by exponential decay with time constants T_1 and T_2 . Let $M_{xy}(t)$ be the magnitude of \mathbf{M} projected onto the transverse plane at time t ; this magnitude component decays as $M_{xy}(t) = M_{xy}(0)e^{-t/T_2}$. The component of \mathbf{M} in the z -direction $M_z(t)$ decays as $M_z(t) = M_\infty - (M_\infty - M_z(0))e^{-t/T_1}$. The magnitude M_∞ describes the equilibrium magnetization magnitude. Note that the relaxation time constants T_1 and T_2 are often quite different; for gray matter and white matter in the brain, $T_1 \gg T_2$, so the magnetization will appear to disappear in the transverse plane long before it reappears again in the longitudinal direction. Taking relaxation into account, we modify Equation (2.1):

$$\frac{d\mathbf{M}}{dt} = \mathbf{M} \times \gamma\mathbf{B} - \frac{M_x}{T_2}\hat{\mathbf{i}} - \frac{M_y}{T_2}\hat{\mathbf{j}} - \frac{(M_z - M_\infty)}{T_1}\hat{\mathbf{k}}. \quad (2.3)$$

The differential system in Equation (2.3) is termed the Bloch equations.

Without some way of perturbing the magnetic moments, the magnetizations would all decay to and remain at equilibrium, and imaging would not be possible. Fortunately, the cross product in the Bloch equations tells us that at equilibrium, the spins can be “tipped” into the transverse plane by applying a short radiofrequency (RF) pulse perpendicular to the main field with frequency equal to ω_0 . This process of

exciting the spins requires only a short-duration pulse of much smaller magnitude than the main field; however, unlike the main field, this pulse deposits energy into the susceptible particles, heating the tissue. Therefore, care must be taken during excitation to ensure the rate of heating does not exceed the specific absorption rate (SAR) limit of the subject. Fortunately, most subjects are capable of dissipating this heat to avoid tissue damage under normal conditions. Special care must be taken on subjects with metal present, or with homeostasis imbalance, as these conditions can increase the dangers of RF heating on the body. We will only use the same RF excitation pulse that is used in conventional MRI, so heating will not be affected by the reconstruction methods proposed in this thesis.

2.2 Cartesian MR Imaging

Cartesian MRI refers to acquiring a Cartesian grid of samples of the spatial Fourier transform of the bulk magnetization. This methodology is very common for acquiring images and volumes displaying local tissue contrast, and many important pulse sequences implement Cartesian imaging. The key element to Cartesian and other Fourier sampling methods is the use of spatial gradient fields during the relaxation of excited spins. As we describe below, the spacing and extent of the Cartesian grid of samples both affect the acquisition time and the field of view and voxel size of the resulting image.

2.2.1 Gradient Fields

While the Bloch equations describe how spins can be excited to allow the bulk magnetization to be measured, we need to introduce spatial selectivity to identify how that magnetization is spatially distributed and construct an image (or volume). The approaches discussed here utilize gradient magnetic fields that vary linearly in amplitude over space and are parallel (or anti-parallel) to the main field. We parameterize these fields using the spatial gradient $\mathbf{G}(t) = [G_x(t), G_y(t), G_z(t)] = \nabla_{x,y,z} B_z(t)$. With this gradient, the component of the net applied field in the z -direction at position

$\mathbf{r} = [x, y, z]$ is $B_z(\mathbf{r}, t) = (B_0 + \int_0^t \mathbf{G}(\tau) \cdot \mathbf{r} d\tau)$. The received signal is

$$y(t) = \int_{\mathbf{r}} M_{xy}(\mathbf{r}, t) e^{-j2\pi[\mathbf{k}(t) \cdot \mathbf{r} + \frac{\gamma}{2\pi} B_0 t]} d\mathbf{r}, \quad (2.4)$$

where

$$\mathbf{k}(t) = \frac{\gamma}{2\pi} \int_0^t \mathbf{G}(\tau) d\tau. \quad (2.5)$$

Limiting ourselves to time scales $t \ll T_1, T_2$, relaxation is insignificant, and the main dynamic in M_{xy} is due to precession, so demodulating by $e^{j\omega_0 t}$ yields a constant (in time) $M_{xy}(\mathbf{r})$:

$$y(t) = \int_{\mathbf{r}} M_{xy}(\mathbf{r}) e^{-j2\pi\mathbf{k}(t) \cdot \mathbf{r}} d\mathbf{r}. \quad (2.6)$$

We observe that Equation (2.6) describes the spatial Fourier transform of $M_{xy}(\mathbf{r})$ at spatial frequency $\mathbf{k}(t)$. The spatial Fourier transform domain (either 2-D or 3-D) measured this way is called k-space, and the path traced by $\mathbf{k}(t)$ is called the k-space trajectory. By carefully choosing $\mathbf{G}(t)$ and our sample times, we can sample the 2-D or 3-D spatial Fourier transform at uniformly spaced intervals. These samples are the discrete Fourier transform (DFT) of the discrete image we seek to acquire.

Note: In addition to exciting spins, the RF excitation pulse can be designed to select a particular 2-D slice of our image by applying a sinc-like pulse in combination with a linear gradient. Designing such pulses is a separate topic (see [7]), but it suffices for our discussion that we can design slice-selective excitation pulses for 2-D slices of almost any thickness, position, and orientation. The original MRI design actually used this slice selection approach in all three directions, requiring a separate excitation and relaxation for every voxel in the acquired volume; however, the speed of this approach is fundamentally limited by the excitation and relaxation times and is rarely used.

Conventional 2-D Cartesian MRI consists of repeatedly selecting and exciting a 2-D slice and sampling lines of k-space while applying gradient fields during relaxation. Suppose we wish to acquire a slice parallel to the transverse plane. We apply an RF pulse with a gradient varying linearly in the z -direction to select the slice. Then, we

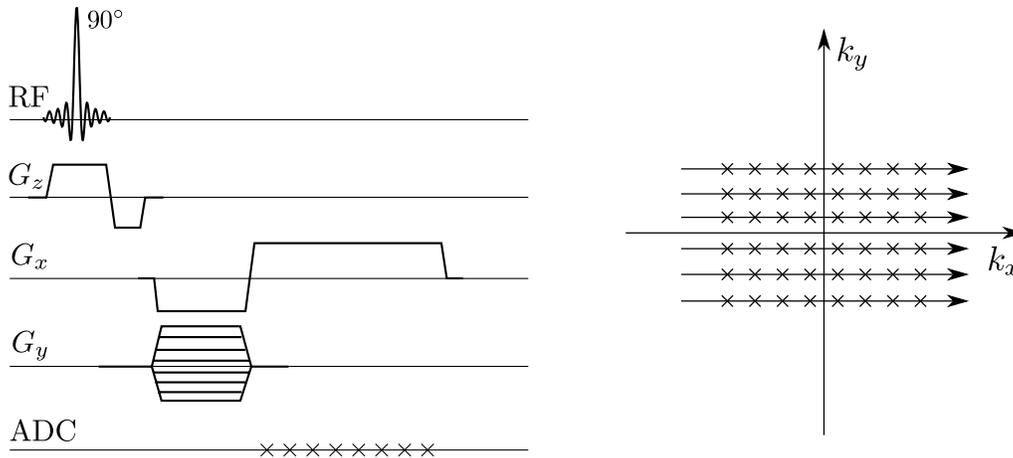


Figure 2.2: RF slice-selective excitation (in z -direction) followed by Cartesian sampling of k -space in $k_x k_y$ -plane using x - and y -gradients. The acquisition is repeated for different magnitudes of G_y , resulting in the sampling of different phase encode scan lines, shown in the k -space plot on the right. The samples are taken during the frequency encoding x -gradient and are marked with the ADC.

apply a gradient varying in the y -direction to shift the k -space trajectory in the k_y -direction. Finally, we demodulate and sample $y(t)$ while applying a gradient linearly varying in the x -direction. We repeat, changing the amplitude of the gradient varying in the y -direction to select different lines of k -space. Essentially, we are raster-scanning k -space. We denote the x -direction the readout or frequency encode direction because we sample while tracing the k -space trajectory in that direction. We denote the y -direction the (primary) phase encode direction, and we denote the z -direction the slice encode direction. This acquisition is depicted in Figure 2.2. This approach can be extended to 3-D Cartesian imaging by avoiding slice selection and instead exciting the entire volume and using the gradients varying in both the y - and z -directions to select the scan line in 3-D k -space. Then, we have two phase-encode directions.

2.2.2 Field of View and Spatial Resolution

In designing a Cartesian MRI acquisition, we need to determine appropriate choices of the field of view (FOV) and the spatial resolution (i.e. voxel size). To avoid spatial aliasing artifacts in the image, the FOV should be larger than what we are imaging. The voxel size should be sufficiently small to resolve the smallest features we are

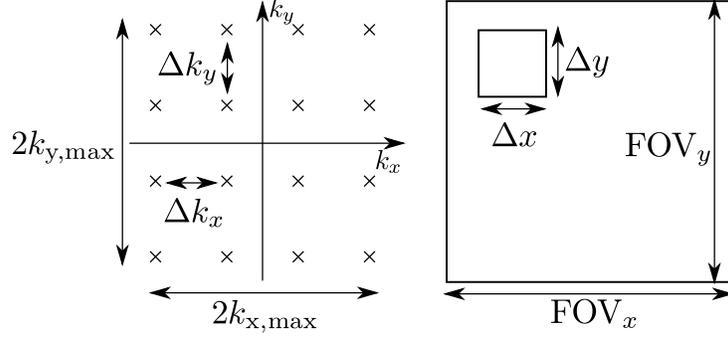


Figure 2.3: The sample spacing Δk_x in k-space and the extent $k_{x,\max}$ relate to the FOV FOV_x and voxel size Δx , respectively, of the reconstructed image in the x -direction. Similarly, Δk_y and $k_{y,\max}$ are connected to FOV_y and Δy .

interested in observing. The frequency spacing Δk between samples in k-space (in units of inverse distance) is equal to the reciprocal of the FOV: $\text{FOV} = 1/\Delta k$. The voxel size Δ is equal to the reciprocal of the extent of k-space ($-k_{\max}$ to k_{\max}) that is sampled: $\Delta = 1/(2k_{\max})$. These parameters are depicted physically in Figure 2.3.

For safety, we cannot arbitrarily increase the magnitude of the gradient fields, so the total acquisition time for Cartesian imaging is proportional to the number and length of the scan lines acquired. A larger FOV or smaller voxels in the phase encode direction necessitates more scan lines, and smaller voxels in the frequency encode direction increases the length of those scan lines. Assuming that we are not limited by the sampling rate of our analog-to-digital converter (ADC), we note that the acquisition time is unaffected by the spacing between samples within a scan line, so we can achieve arbitrarily large FOV in the frequency encode direction for free. Thus, we typically choose the frequency encode direction to point in the longest dimension of our volume to minimize the acquisition time, and we oversample k-space in that direction to avoid any possibility of aliasing.

Image quality is also affected by signal-to-noise ratio (SNR); when SNR is too low, tissue contrast and anomalous regions are difficult to distinguish from observation noise. SNR is roughly proportional to acquisition time, so reducing the acquisition time is accompanied by a similar reduction in SNR. The degradation in SNR will be a major concern in reconstructing quality images from accelerated MRI data.

2.3 Accelerated MR Imaging

Several approaches have become popular for accelerating Cartesian MRI. Keyhole and partial Fourier imaging reduce the extent of k-space that is sampled and use side information to recover the missing regions. Keyhole imaging [93] is a time-series imaging technique used primarily for contrast-enhanced imaging or cardiac imaging, where multiple volumes are collected, and changes of interest are primarily in the low spatial frequencies. A full volume is collected initially, but only low-resolution volumes are collected in subsequent frames, greatly reducing the acquisition time for each frame and increasing the temporal resolution of the technique. The high frequency data in these accelerated frames are substituted from the initial frame, as discussed in [9]. However, keyhole imaging cannot accelerate single-frame imaging like anatomical MRI without substantially reducing spatial resolution.

On the other hand, partial Fourier imaging can be applied to pretty much any acquisition. In partial Fourier imaging, only one half of k-space and a small part of the other half is fully sampled, and complex conjugation or a more sophisticated technique like homodyne processing [70] is utilized to fill in the remaining frequencies. However, partial Fourier techniques cause signal loss when the real-valuedness assumption of the image is violated, which can be caused by a number of factors including chemical shift, field inhomogeneity, blood flow, and the presence of air (e.g. in the sinuses or oral and nasal cavities), iron (e.g. in blood), or other materials with substantially different magnetic susceptibilities. All these variations can introduce valuable phase information that would be lost by assuming the image is real-valued.

A variety of echo train pulse sequences can be used to yield very fast acquisitions. Echo planar imaging (EPI) [62] and its 3-D analogue echo volumar imaging (EVI) [63] utilize a train of gradient echoes to acquire a complete slice using only a single excitation. Other echo train pulse sequences combining gradient and/or spin echoes like GRASE [30] and RARE [41] also achieve fast imaging, although not nearly as fast as techniques based on gradient echoes alone, since RF spin echoes require more time. All these techniques function on the basic premise of acquiring multiple

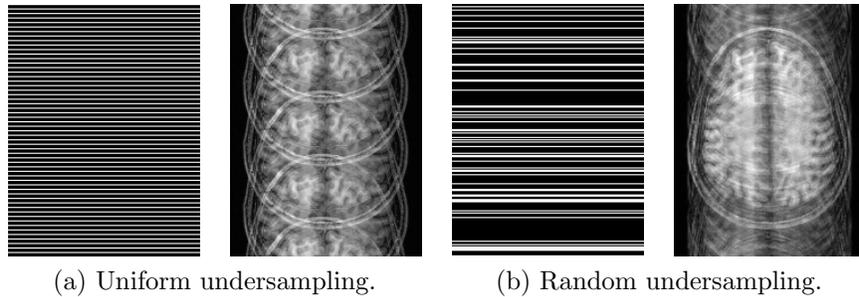


Figure 2.4: Uniform undersampling results in coherent aliasing, while non-uniform or random sampling results in incoherent aliasing. Images shown for k-space undersampled by a factor of 4, with zero-filling reconstruction.

scan lines during the same relaxation period. However, the artifacts and distortions that result from these imaging techniques can make echo train pulse sequences not an ideal approach for accelerated imaging. Although the acceleration is limited, these methods are particularly prevalent in several prominent MRI applications, including functional MRI and clinical and surgical neuroimaging.

Finally, we can increase the spacing between scan lines while maintaining the same k-space extent. This accelerated imaging approach maintains spatial resolution while reducing the FOV. When the object is larger than the reduced FOV, aliasing results, which may make the image unusable. As portrayed in Figure 2.4, uniformly undersampling k-space (keeping the spacing between scan lines equal) yields strongly coherent artifacts in the image domain, while non-uniformly (or randomly) undersampling k-space yields incoherent artifacts that appear lower in magnitude but more smeared throughout the image. As is discussed later, parallel imaging can be employed to undo coherent aliasing, and compressed sensing can resolve incoherent aliasing.

2.4 Parallel MR Imaging

Parallel MRI [82] was conceived to use multiple receiver coils to improve image quality by increasing SNR. Averaging P measurements of k-space improves SNR by a factor of \sqrt{P} , assuming equal noise variances and ignoring correlations across the coils. Instead of a single large receiver coil surrounding the entire FOV, an array of small coils is used, and ideally, each of these coils senses the magnetizations independently

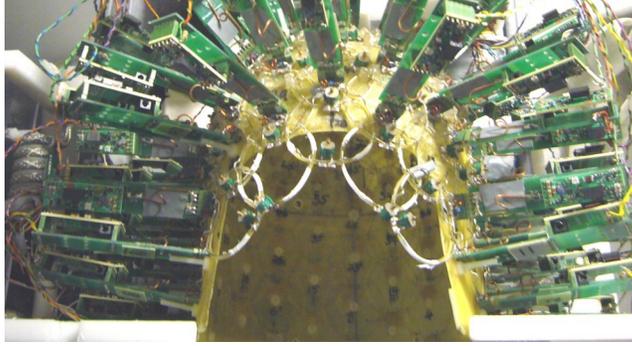


Figure 2.5: This 96-channel head array coil prototype has many small coils (metal rings) around the head enclosure. Each coil has its own data acquisition hardware, so all the channels can be acquired simultaneously. Commercially available array coils enable parallel imaging to be used for many MRI applications.

of the others. In reality, these coils are coupled due to the shared effects of the induced fields of one coil affecting the others, and as an end result, the noise is correlated to a small degree. Multi-channel receive coil arrays such as the 96-channel head coil [100] shown in Figure 2.5 are now widely available for a multitude of imaging applications.

To understand how parallel imaging can be useful for accelerated imaging with reduced-FOV data, we return to the signal equation (Equation (2.6)). With a single coil, we assume the coil senses the entire field of view uniformly, so that the “receive field” B_1^- is constant. With parallel imaging, the arrays are designed so that the individual coils have highly non-uniform spatial sensitivities. Denote by $S_p(\mathbf{r})$ the transverse components of the receive field of the p th coil (that coil’s sensitivity); as is done with the transverse magnetization, the two components are combined into a single complex-valued number. Then, the signal $y_p(t)$ observed by the p th coil is

$$y_p(t) = \int_{\mathbf{r}} M_{xy}(\mathbf{r}) S_p(\mathbf{r}) e^{-j2\pi\mathbf{k}(t)\cdot\mathbf{r}} d\mathbf{r}. \quad (2.7)$$

For convenience, we drop the transverse magnetization subscript and write $M(\mathbf{r}) = M_{xy}(\mathbf{r})$. Also, when referring to the magnetization sensed by the p th coil, we use the notation $M_p(\mathbf{r}) = M(\mathbf{r})S_p(\mathbf{r})$.

Each receive channel in the coil array is sampled at the same time, yielding estimates at the same k-space frequencies \mathbf{k} according to Equation (2.7). The sample

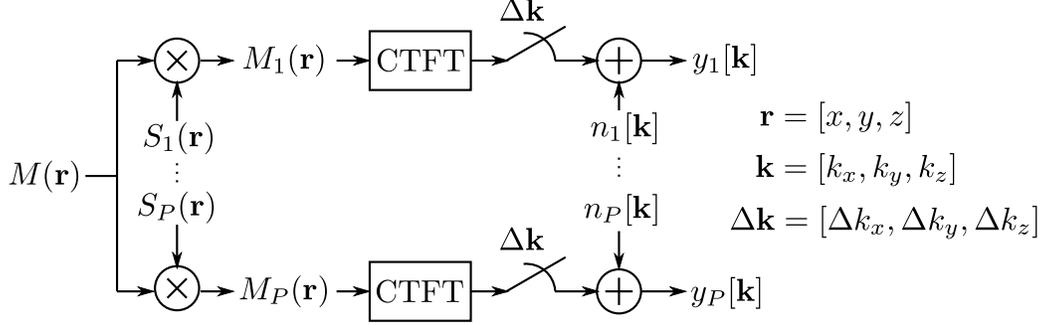


Figure 2.6: Block diagram of the observation model for an MRI acquisition with a P -channel parallel receive array coil. The sensed magnetizations $M_1(\mathbf{r}), \dots, M_P(\mathbf{r})$ all derive from the object magnetization $M(\mathbf{r})$. The k-space observations $y_1[\mathbf{k}], \dots, y_P[\mathbf{k}]$ are generated simultaneously from these sensed magnetizations.

$y_p[\mathbf{k}]$ represents the value of k-space at frequency \mathbf{k} ; these samples are corrupted by correlated observation noise, which is modeled by additive complex Gaussian noise $n_p[\mathbf{k}]$. The noise vector $[n_1[\mathbf{k}], \dots, n_P[\mathbf{k}]]^T$ has covariance $\mathbf{\Lambda}$, independent of \mathbf{k} . The observation model for parallel imaging with a P -channel coil is illustrated in in Figure 2.6. Given sufficiently many samples, the inverse DFT can be used to recover the noisy discretized samples $M_p[\mathbf{r}]$ of the sensed magnetization from $y_p[\mathbf{k}]$, and any number of coil combination methods can be used to form a combined image $M[\mathbf{r}]$.

2.4.1 Coil Sensitivity Estimation

In the far field (the distance is much greater than the radius of the coil), the magnitude of the sensitivity is inversely proportional to the square of the distance. In the near field, the coil can be treated as a series of finite elements, and the Biot-Savart law can be used to simulate the spatial sensitivity of the coil parameterized by curve C :

$$\mathbf{B}(\mathbf{r}) \propto \int_C \frac{d\boldsymbol{\ell} \times \mathbf{r}}{\|\mathbf{r}\|_2^3}, \quad (2.8)$$

where $\boldsymbol{\ell}$ is the tangent vector on the curve, in the direction of the current flow, and \mathbf{r} is the vector from the coil element to the spatial point in question. A simulator for estimating sensitivities of arbitrary coil array geometries can be downloaded from [53]. Note that neither the Biot-Savart law nor this simulator account for the

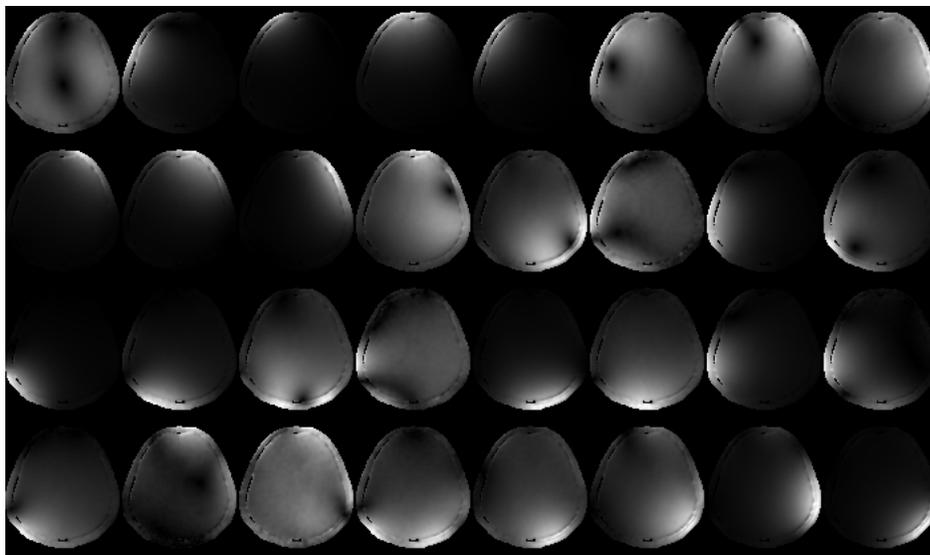


Figure 2.7: Magnitude coil sensitivities for 32-channel head coil array computed from acquired data using 32-channel array and single-channel body coils.

dynamic loading and cross-talk that occurs across coils. In addition, the sensitivities are themselves affected by the subject’s magnetization. Thus, sensitivities are best determined empirically, with the subject in the magnet.

Multiple approaches exist for empirical measurements. One such measurement divides each coil image by the sum-of-squares combined image to yield an estimate of the sensitivities without any additional data; this measurement is derived from comparing the sum-of-squares combination to the SNR-optimal formula (see Equations (2.9) and (2.10) below). However, this approach assumes that all the phase information belongs to the coils, so the result is not suitable for reconstructing complex-valued images. If a single-coil acquisition is available, the coil sensitivities can be derived by dividing the coil images by that single-coil image. Since coil sensitivities are usually slowly-varying at main field strengths up to 3 T, low-resolution acquisitions and polynomial fitting can be used to improve the robustness of the sensitivity estimation to noise in the single-coil image and regions with low signal (e.g. near the periphery of the FOV). An example of high-quality magnitude sensitivity maps estimated from empirical 32-channel and single-channel coil data is shown in Figure 2.7. These sensitivities depict smooth far field decay and higher-order effects like coil loading.

2.4.2 Combining Coil Images

One may ask, what is the optimal way to combine data across coils to produce a single image? The sum-of-squares method [82] is nearly optimal for a magnitude image:

$$M[\mathbf{r}] = \sqrt{\mathbf{M}[\mathbf{r}]^H \mathbf{\Lambda}^{-1} \mathbf{M}[\mathbf{r}]}, \quad (2.9)$$

where $M[\mathbf{r}]$ is the combined image magnetization for the voxel at \mathbf{r} , $\mathbf{M}[\mathbf{r}]$ is the column vector of voxel values $[M_1[\mathbf{r}], \dots, M_P[\mathbf{r}]]^T$ at position \mathbf{r} across the P coils, and $\mathbf{\Lambda}$ is the noise covariance matrix (the identity matrix may be used if a noise covariance matrix has not been measured using a noise-only pre-scan acquisition). If samples $S_p[\mathbf{r}]$ of the coil sensitivities are known, the unity-gain SNR-optimal coil combination [82] is

$$M[\mathbf{r}] = (\mathbf{S}[\mathbf{r}]^H \mathbf{\Lambda}^{-1} \mathbf{S}[\mathbf{r}])^{-1} \mathbf{S}[\mathbf{r}]^H \mathbf{\Lambda}^{-1} \mathbf{M}[\mathbf{r}], \quad (2.10)$$

where $\mathbf{S}[\mathbf{r}]$ is the vector of coil sensitivities $[S_1[\mathbf{r}], \dots, S_P[\mathbf{r}]]^T$ at voxel \mathbf{r} . The coil combination in Equation (2.10) preserves the signal gain. The linear combination that normalizes the noise is

$$M[\mathbf{r}] = (\mathbf{S}[\mathbf{r}]^H \mathbf{\Lambda}^{-1} \mathbf{S}[\mathbf{r}])^{-1/2} \mathbf{S}[\mathbf{r}]^H \mathbf{\Lambda}^{-1} \mathbf{M}[\mathbf{r}]. \quad (2.11)$$

The noise in this linear combination $M[\mathbf{r}]$ has unit variance for all \mathbf{r} .

2.5 Accelerated Parallel Imaging Reconstruction

Reducing the FOV causes image voxels to alias on top of each other. Parallel imaging introduces different non-uniform spatial weightings to the aliased voxels in each coil channel, enabling the aliased voxels to be separated during reconstruction. Accelerated parallel imaging methods such as SENSE [76], SMASH [87], GRAPPA [38], and SPIRiT [59] leverage the fact that each coil weights the positions that are aliased together differently to separate the component voxels and un-alias the image. These parallel imaging methods are described in the following subsections.

2.5.1 SENSE

Sensitivity encoding (SENSE) constructs and solves a linear system relating the desired un-aliased voxel magnetizations, the coil sensitivities, and the aliased magnetizations observed by each coil. For simplicity, consider a one-dimensional example (SENSE easily generalizes to higher dimensions) where we undersample an object $M[n]$, for $n = 0, 1, \dots, N - 1$, by a factor of R (assume R divides N), and suppose that the number of coils P is at least as large as R . The reduced-FOV image resulting from taking the inverse DFT of the k-space samples from coil channel p contains aliased pixels $M_p[n]$, for $n = 0, \dots, N/R - 1$, each of which is a weighted superposition of $M[n_r]$, where $n_r = (n - rN/R) \bmod N$ for periodic replicates $r = 0, \dots, R - 1$. SENSE assumes we have coil sensitivities available, so we can construct the $P \times R$ matrix \mathbf{S}_n of sensitivities for all the coils at all the replicate positions n_r . For convenience, collect the aliased voxels $M_p[n]$, for $p = 1, \dots, P$, into vector $\mathbf{M}_n^{(a)}$, and collect un-aliased voxels $M[n_r]$ for $r = 0, \dots, R - 1$, into vector $\mathbf{M}_n^{(na)}$. The resulting linear system for a single value of $n \in \{0, 1, \dots, N/R - 1\}$ is

$$\underbrace{\begin{bmatrix} M_1[n] \\ \vdots \\ M_P[n] \end{bmatrix}}_{\mathbf{M}_n^{(a)}} = \underbrace{\begin{bmatrix} S_1[n_0] & \cdots & S_1[n_{R-1}] \\ \vdots & & \vdots \\ S_P[n_0] & \cdots & S_P[n_{R-1}] \end{bmatrix}}_{\mathbf{S}_n} \underbrace{\begin{bmatrix} M[n_0] \\ \vdots \\ M[n_{R-1}] \end{bmatrix}}_{\mathbf{M}_n^{(na)}}. \quad (2.12)$$

SENSE also accounts for the noise in the observations: suppose we have measured the noise covariance matrix across all the coils $\mathbf{\Lambda}$ using a fast noise-only pre-scan acquisition (with no RF excitation). Then, SENSE estimates the SNR-optimal un-aliased voxel magnetizations $\hat{\mathbf{M}}_n^{(na)}$ in a single combined image by computing

$$\hat{\mathbf{M}}_n^{(na)} = (\mathbf{S}_n^H \mathbf{\Lambda}^{-1} \mathbf{S}_n)^{-1} \mathbf{S}_n^H \mathbf{\Lambda}^{-1} \mathbf{M}_n^{(a)}, \quad (2.13)$$

for each $n = 0, \dots, N/R - 1$. Note that if the coil sensitivities were not sufficiently non-uniform and distinct, the matrix \mathbf{S}_n would be rank-deficient, and SENSE would not be able to resolve aliasing using these sensitivities alone. Errors in the coil sen-

sitivity estimates will cause SENSE to improperly un-alias the reduced-FOV images, yielding a combined image with visible aliasing artifacts. To reduce the effect of coil sensitivity errors on the SENSE reconstruction, JSENSE [103] jointly estimates and refines the sensitivities and the image, using a low-degree polynomial basis for the sensitivities. In addition, the noise in the result may be amplified in the combined un-aliased image; this noise amplification due to the parallel imaging method (as opposed to undersampling, which reduces SNR by a factor of \sqrt{R}) is called the g-factor. Analytically, the g-factor g is defined as

$$g = \frac{\text{SNR}_{\text{full}}}{\sqrt{R} \cdot \text{SNR}_{\text{redu}}}, \quad (2.14)$$

where SNR_{full} is the SNR of the combined fully-sampled image, and SNR_{redu} is the SNR of the combined reconstructed image. Due to the spatial inhomogeneity induced by the coil sensitivities, the noise amplification is expected to vary spatially, so g-factors are usually computed for each voxel in the full field of view. To reduce noise amplification and aliasing artifacts, the SENSE method can be regularized using Tikhonov regularization [90, 54], a sparsity-promoting ℓ_1 norm, or a low rank matrix prior using the nuclear norm [60].

2.5.2 SMASH

SMASH, the SiMultaneous Acquisition of Spatial Harmonics, is an early method for accelerated parallel imaging that attempts to interpolate the missing frequencies in reduced-FOV k-space by approximating k-space frequency shifts by linear combinations of acquired data across coils.

Consider a one-dimensional example where a signal $M[n]$ of length N with k-space $y[k]$ ($k = 0, \dots, N - 1$) is undersampled by a factor of R . Denote the full-FOV k-space sample spacing Δk , so the undersampled k-space $y_p[k]$ ($k = 0, \dots, N/R - 1$) has spacing $R\Delta k$. Then, basic Fourier transform properties tell us that a shift in the frequency domain is equivalent to multiplying the signal by a complex exponential in the image domain: $e^{-j2\pi rn/N} M[n] \xrightarrow{\text{DFT}} y[k + r]$. As with SENSE, SMASH also

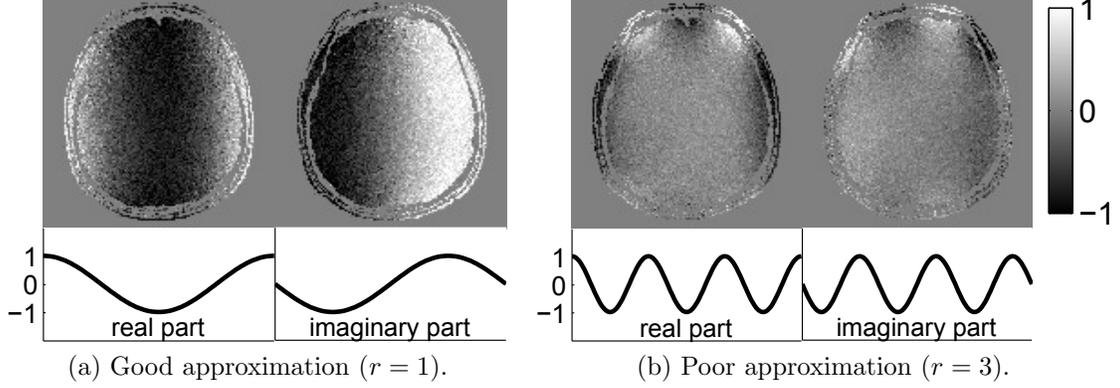


Figure 2.8: Real and imaginary parts of SMASH approximations (top) to complex exponentials (bottom) using a least-squares fit with empirical sensitivities of a 32-channel head coil receive array.

requires prior knowledge or estimates of the coil sensitivities $S_1[n], \dots, S_P[n]$. The SMASH method consists of finding weights $g_{p,r}$ such that

$$\sum_{p=1}^P g_{p,r} S_p[n] \approx e^{-j2\pi rn/N}, \quad (2.15)$$

for $r = 0, \dots, R - 1$. Suitable weights can be found using a least-squares fit. Then,

$$\sum_{p=1}^P g_{p,r} y_p[k] = \sum_{n=0}^{N-1} \sum_{p=1}^P g_{p,r} M[n] S_p[n] e^{-j2\pi Rkn/N} \quad (2.16)$$

$$\approx \sum_{n=0}^{N-1} M[n] e^{-j2\pi rn/N} e^{-j2\pi Rkn/N} \quad (2.17)$$

$$= \sum_{n=0}^{N-1} M[n] e^{-j2\pi(r+Rk)n/N} \quad (2.18)$$

$$= y[Rk + r]. \quad (2.19)$$

Then, the combined image estimate $\hat{M}[n]$ is the inverse DFT of the full interpolated k-space. Advantages of SMASH include robustness to errors in the coil sensitivity estimates and computational efficiency; SMASH only requires R linear system inversions, while SENSE requires N/R linear system inversions. However, SMASH is limited by the quality of the approximation in Equation (2.15). At high accelerations (large R), the slowly varying coil sensitivities are a poor match for high-

frequency complex exponentials, so the error in the SMASH reconstruction would be large. This limitation in approximating sensitivities is depicted in Figure 2.8. The good approximation (a) portrays an approximation that varies from -1 (black) to +1 (white) in a fashion mimicking the slowly varying sinusoid plotted immediately below the image. The poor approximation (b) is mostly gray in the center of the image, with limited variation between -1 and +1 only in the periphery, behavior that clearly contrasts with the higher frequency sinusoid plotted below it. Additionally, the poor approximation is highly non-uniform in the vertical direction.

2.5.3 GRAPPA

The GRAPPA method is designed to be a more robust extension to the SMASH method, with several fundamental differences. Instead of relying on coil sensitivity estimates explicitly, the GRAPPA method relies on a small set of calibration data called ACS lines that have frequency spacing corresponding to the full field of view; these lines are used to fit the interpolation weights as is done in the SMASH variant called AUTO-SMASH [44]. The ACS lines are usually chosen to be at the center of k-space, or in a region known to have high SNR, to minimize the effect of observation noise on the fit. Rather than form a single combined image k-space, the GRAPPA method interpolates the missing frequencies in each coil, yielding P full FOV coil images that can be combined using any of the parallel imaging coil combination methods described earlier (see Equations (2.9) and (2.10)). Finally, the GRAPPA method utilizes multiple neighboring acquired k-space data (source) points to interpolate missing (target) points, yielding an improved reconstruction. Let B represent the number of source points per coil; the set of source points is usually chosen to be centered around the block of target points (although this is not required). Mathematically, the interpolation step is

$$y_p[k+r] = \sum_{q=1}^P \sum_{b=0}^{B-1} g_{p,q,r}[b] y_q[k + (b - \lceil B/2 \rceil + 1)R]. \quad (2.20)$$

We call the vectors $g_{p,q,r}[b]$ GRAPPA kernels. When the undersampled data is uniformly spaced, the GRAPPA reconstruction can be implemented efficiently using convolution. Substituting the known values from the ACS lines $y_p^{\text{ACS}}[k]$ yields the calibration equations used for computing a least-squares fit of the kernel weights:

$$y_p^{\text{ACS}}[k+r] = \sum_{q=1}^P \sum_{b=0}^{B-1} g_{p,q,r}[b] y_q^{\text{ACS}}[k + (b - \lceil B/2 \rceil + 1)R]. \quad (2.21)$$

In the event that the calibration data is noisy, regularization of the least-squares fit may be required. Regularized calibration will be discussed in detail in Chapter 5. GRAPPA can be extended to two- or three-dimensional subsampling in several ways [11]. In this work, we leverage the fact that the 3-D acquisition is only subsampled in two phase-encode dimensions, so we inverse Fourier transform the acquisition in the frequency-encoded direction and process the resulting two-dimensional slices individually. The two-dimensionally subsampled (by R_y in the y -direction and by R_z in the z -direction) GRAPPA reconstruction equation is

$$y_p[k_y + r_y, k_z + r_z] = \sum_{q=1}^P \sum_{b_y=0}^{B_y-1} \sum_{b_z=0}^{B_z-1} g_{p,q,r_y,r_z}[b_y, b_z] \cdot y_q[k_y + (b_y - \lceil B_y/2 \rceil + 1)R_y, k_z + (b_z - \lceil B_z/2 \rceil + 1)R_z]. \quad (2.22)$$

The two-dimensional $B_y \times B_z$ GRAPPA kernels $g_{p,q,r_y,r_z}[b_y, b_z]$ are calibrated for each slice separately, using the ACS line fits

$$y_p^{\text{ACS}}[k_y + r_y, k_z + r_z] = \sum_{q=1}^P \sum_{b_y=0}^{B_y-1} \sum_{b_z=0}^{B_z-1} g_{p,q,r_y,r_z}[b_y, b_z] \cdot y_q^{\text{ACS}}[k_y + (b_y - \lceil B_y/2 \rceil + 1)R_y, k_z + (b_z - \lceil B_z/2 \rceil + 1)R_z]. \quad (2.23)$$

Due to the larger number of source points in two dimensions, calibration becomes even more important, and more ACS lines may be required to achieve a reconstruction of sufficient quality.

When discussing auto-calibrating methods like GRAPPA, both nominal undersampling and total (effective) acceleration are important. Nominal undersampling considers only the spacing between regular acquired k-space lines. The total acceleration also includes the calibration data and may be much less than the nominal undersampling factor. The total acceleration describes the reduction in scan time.

Conventional GRAPPA interpolates missing k-space from uniformly-spaced Cartesian accelerated parallel imaging data. GRAPPA can be extended to non-uniform or non-Cartesian data using multiple kernels or iterative reconstructions. When interpolating missing k-space from non-uniformly undersampled data, the patterns of source and target points vary from block to block, and different interpolation kernels can be calibrated for each pattern encountered. Alternatively, an iterative approach to non-uniform GRAPPA computes the least squares solution that is consistent with the GRAPPA reconstruction from the “source” points of the full k-space result with the “target” points of the result and is consistent with the acquired data [58]. GRAPPA can also be extended to non-Cartesian k-space trajectories using re-gridding, followed by conventional or non-uniform GRAPPA, or using the hybrid GROG approach [83].

As with other accelerated parallel imaging reconstruction methods, the noise amplification can be significant. GRAPPA g-factors can be computed analytically by considering interpolation as multiplication in the image domain [12]. When an analytical expression for the g-factors does not exist or is computationally expensive, the multiple replica method, which consists of taking multiple full- and reduced-FOV acquisitions and averaging the observed noise in the reconstructions, can be used to estimate g-factors in vivo. However, as tens or hundreds of repetitions may be necessary to yield statistically significant g-factors, the multiple replica method also may be impractical for human subjects. Alternatively, g-factors can be estimated using the pseudo multiple replica method, which uses only one set of reduced-FOV data and performs Monte Carlo simulations with synthetic additive noise to approximate the multiple replica estimate [81]. Suppose $\hat{\mathbf{Y}}$ is the accelerated parallel imaging reconstructed k-space from the acquired data \mathbf{D} . Then, each Monte Carlo repetition consists of adding complex Gaussian noise \mathbf{N} with covariance $\mathbf{\Lambda}$ to the acquired data

to form $\mathbf{D}' = \mathbf{D} + \mathbf{N}$, performing the accelerated parallel imaging reconstruction on \mathbf{D}' to form $\hat{\mathbf{Y}}'$, and finding the difference $\hat{\mathbf{Y}}' - \hat{\mathbf{Y}}$. When the reconstruction method is linear, this difference in the reconstructed k-space is due solely to the added noise, and the ratio of the standard deviations of the error for each combined image voxel observed over all the Monte Carlo trials to the standard deviation of the additive noise measures the total noise amplification. Dividing the noise amplification by \sqrt{R} yields the g-factor estimates for each voxel.

2.5.4 SPIRiT

SPIRiT is an accelerated parallel imaging reconstruction method akin to GRAPPA. Where the calibrated GRAPPA kernel interpolates missing frequencies in GRAPPA, SPIRiT uses the calibrated kernel to enforce consistency between every k-space point (acquired or missing) and all of its neighbors (again, acquired or missing). Rather than directly interpolate the missing k-space, SPIRiT forms a linear system and finds the missing k-space that results in a least-squares fit of that system. To describe SPIRiT mathematically, begin with an arbitrary k-space frequency k in the full-FOV k-space, and let ξ_k represent the set of its neighboring frequencies; B is the size of this set. Then, the SPIRiT kernels $g_{p,q}[b]$ fit the linear system

$$y_p[k] = \sum_{q=1}^P \sum_{b=0}^{B-1} g_{p,q}[b] y_q[\xi_k[b]], \quad (2.24)$$

for all k , $p = 0, \dots, P - 1$. Notice that unlike GRAPPA, the SPIRiT kernel does not depend on and makes no assumption about the undersampling pattern. In fact, SPIRiT has the advantage of being readily applied to arbitrary subsampling of Cartesian k-space, and with gridding, non-Cartesian k-space as well. Collecting the full-FOV k-space (known and unknown) for all coil channels into columns of the matrix \mathbf{Y} (each column represents all the k-space for a single coil channel), and the kernels into a block convolution matrix \mathbf{G} , SPIRiT attempts to solve the least-squares problem

$$\underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{I} - \mathbf{G}\| \text{vec}(\mathbf{Y})\|_2^2, \quad (2.25)$$

subject to some constraint related to the observations \mathbf{D} . The shorthand $\text{vec}(\cdot)$ stacks the columns of a matrix into a single vector. When we wish to denoise the data, we can solve either the constrained problem

$$\underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{I} - \mathbf{G}\| \text{vec}(\mathbf{Y})\|_2^2 \quad \text{s.t.} \quad \|\mathbf{D} - \mathbf{K}_a \mathbf{Y}\|_F^2 \leq \varepsilon, \quad (2.26)$$

or the unconstrained problem

$$\underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{I} - \mathbf{G}\| \text{vec}(\mathbf{Y})\|_2^2 + \mu \|\mathbf{D} - \mathbf{K}_a \mathbf{Y}\|_F^2. \quad (2.27)$$

The matrix \mathbf{K}_a is a simple subsampling matrix that extracts the acquired data from all of k-space. The Frobenius norm $\|\cdot\|_F$ of a matrix is equivalent to the ℓ_2 norm of the vector of all the elements of that matrix. The parameters ε and μ are determined according to the variance of the observation noise. Alternatively, we can preserve the acquired data exactly by constraining the observed k-space with equality:

$$\underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{I} - \mathbf{G}\| \text{vec}(\mathbf{Y})\|_2^2 \quad \text{s.t.} \quad \mathbf{D} = \mathbf{K}_a \mathbf{Y}. \quad (2.28)$$

We can formulate an equivalent unconstrained problem by operating in the nullspace of the full k-space. Let \mathbf{X} be the missing k-space for all the coils, and \mathbf{K}_{na} the subsampling matrix extracting missing k-space from the full k-space. Then,

$$\underset{\mathbf{X}}{\text{minimize}} \|\mathbf{I} - \mathbf{G}\| \text{vec}(\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D})\|_2^2, \quad (2.29)$$

and $\mathbf{Y} = \mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D}$. Although closed-form solutions of all these least-squares problems exist, we typically resort to iterative methods because the system is too large for direct inversion. Typically, we avoid storing \mathbf{G} in memory and use convolution to implement the necessary matrix-vector multiplications.

Chapter 3

Sparsity and Compressed Sensing

An alternative approach to improving image quality is to incorporate prior information about the image into the reconstruction. A popular such prior is rooted in the notion of “sparsity” – that a length N signal can be represented in a basis with only K nonzero coefficients, where $K \ll N$. However, the support, or indexes of those nonzero coefficients, is not known *a priori*. An intuitive example of a sparse signal (in the canonical basis) is the samples of a series of pulses; the sampled signal would have nonzero values at only a few time indexes. Another common example of sparsity is a piecewise constant signal, which has only a few nonzero coefficients when represented with the finite-differences transform basis.

Many signals, including natural images, are not truly sparse, but have only a few significant coefficients; the others are either zero or of much smaller magnitude. These signals, called approximately sparse, or compressible, can be represented to a high degree of accuracy using a small fraction of the signal coefficients, but the representation is not exact.

In this chapter, we discuss mathematical measures of sparsity used throughout the literature and provide an overview of both sparsity-regularized denoising and compressed sensing (CS) and relate these frameworks to accelerated MRI and accelerated parallel imaging reconstruction. Existing methods for combining accelerated parallel imaging and sparsity are described.

3.1 Measures of Sparsity

Several measures (we use the term measure in the imprecise non-mathematical way of comparing a property of two vectors, as opposed to the mathematical notion used in integration and measure theory) of the sparsity of a signal are possible. The ℓ_0 “norm” counts the number of nonzero values of a signal:

$$\|\mathbf{w}\|_0 = \sum_{n=0}^{N-1} \mathbf{1}_{\{w[n] \neq 0\}}. \quad (3.1)$$

This measure does not satisfy the scalability and triangle inequality properties of a norm, and it is not a convex function of \mathbf{w} . Furthermore, it has a jump-discontinuity at the origin, which means it is not differentiable. Another interesting property of this measure is it is constant for $w[n] \neq 0$; a large nonzero coefficient is just as significant as a small nonzero coefficient.

Several relaxations of the ℓ_0 “norm” exist in the literature. The most common relaxation is the ℓ_1 norm

$$\|\mathbf{w}\|_1 = \sum_{n=0}^{N-1} |w[n]|, \quad (3.2)$$

which satisfies all the properties of a norm, and is therefore convex. This convex relaxation is popular due to the wide variety of efficient techniques for solving ℓ_1 -regularized problems, and the global convergence guarantees provided by convexity [10]. A wide variety of nonconvex relaxations exist as well. The ℓ_p^p ($0 < p < 1$) “norms” [23] are popular:

$$\|\mathbf{w}\|_p^p = \sum_{n=0}^{N-1} |w[n]|^p. \quad (3.3)$$

The unit balls for the ℓ_0 , ℓ_1 , and ℓ_p^p penalty functions are depicted in two dimensions in Figure 3.1. The ℓ_0 and ℓ_p^p unit balls are nonconvex, while the ℓ_1 unit ball is convex. Thus, if we were to place a line (or in higher dimensions, a hyperplane) tangent to the unit ball, the line may intersect the ℓ_0 or ℓ_p^p unit balls in multiple locations, while the line may intersect the ℓ_1 norm unit ball in only one place (or along a single side of the ball, but the intersection forms a single connected set). The

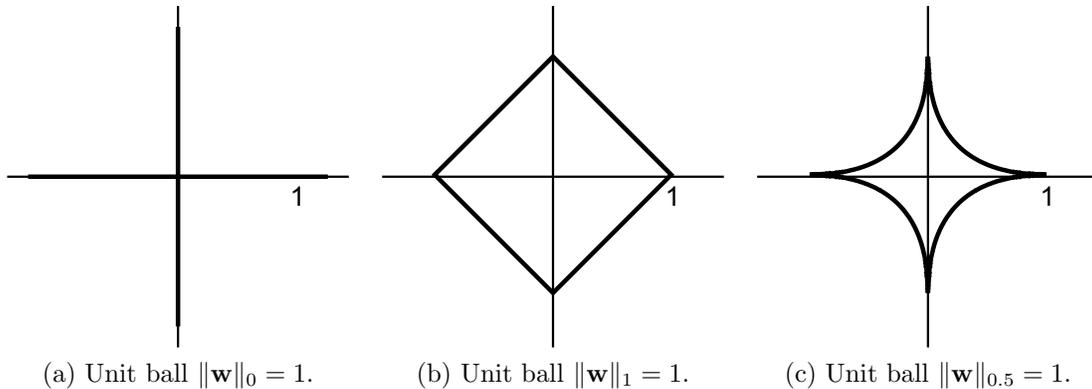


Figure 3.1: Unit balls are shown in two dimensions for the ℓ_0 , ℓ_1 , and ℓ_p measures. Note that the two lines that form the two-dimensional unit ball for the ℓ_0 “norm” actually extend to $\pm\infty$ and exclude the origin.

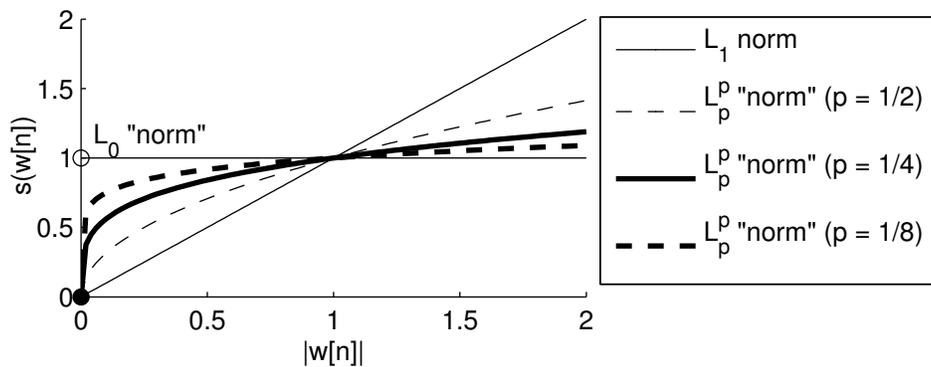


Figure 3.2: The ℓ_1 norm is plotted with ℓ_p^p “norms” for different values of p ($0 < p < 1$). The ℓ_0 “norm” is included for comparison.

ℓ_p^p penalty functions converge to the ℓ_0 “norm” as $p \rightarrow 0$. Graphically, as p decreases, the measure increases more rapidly to the $s(w[n]) = 1$ line and increases more slowly once it has reached that line. Although the ℓ_p^p measures of sparsity are not convex, they are monotonic and concave on \mathbb{R}^+ , and convergence guarantees for such functions are possible under certain conditions [91]. The graphs of ℓ_p^p penalty functions are compared against that of the ℓ_1 norm in Figure 3.2.

The Laplace penalty function $\|\mathbf{w}\|_{L(\alpha)}$, the Welsch penalty function $\|\mathbf{w}\|_{W(\alpha)}$, and

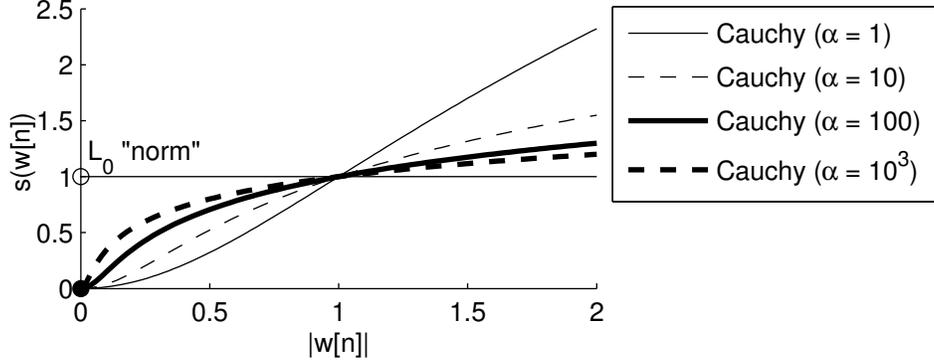


Figure 3.3: The Cauchy penalty function is plotted for different values of α . The ℓ_0 “norm” is included for comparison.

the Cauchy penalty function $\|\mathbf{w}\|_{C(\alpha)}$ also promote sparsity [91, 37, 85, 96, 99]:

$$\|\mathbf{w}\|_{L(\alpha)} = \sum_{n=0}^{N-1} 1 - e^{-\alpha|w[n]|}. \quad (3.4)$$

$$\|\mathbf{w}\|_{W(\alpha)} = \sum_{n=0}^{N-1} 1 - e^{-\alpha|w[n]|^2}. \quad (3.5)$$

$$\|\mathbf{w}\|_{C(\alpha)} = \frac{1}{\log(1 + \alpha)} \sum_{n=0}^{N-1} \log(1 + \alpha|w[n]|^2). \quad (3.6)$$

These nonconvex penalty functions all converge to the ℓ_0 “norm” as $\alpha \rightarrow \infty$. The choice of α balances the fidelity to the ℓ_0 “norm” with the numerical properties of the penalty function. In Figure 3.3, the Cauchy penalty function is plotted as a function of $|w[n]|$ for different values of α . As α increases, the convex region around $|w[n]| = 0$ shrinks, the function increases more rapidly to the $s(w[n]) = 1$ line, and the overshoot decreases.

When we are discussing a general measure of sparsity, we use the notation $\|\mathbf{w}\|_S$. Note that all these measures of sparsity are separable; i.e.

$$\|\mathbf{w}\|_S = \sum_{n=0}^{N-1} s(w[n]) \quad (3.7)$$

for some scalar function $s(\cdot)$.

The concept of sparsity can be extended from real to complex numbers in sev-

eral ways. One generalization treats the sparsity of the real and imaginary parts independently:

$$\|\mathbf{w}\|_S \triangleq \sum_{n=0}^{N-1} s(\Re\{w[n]\}) + s(\Im\{w[n]\}). \quad (3.8)$$

Another approach measures the sparsity of the magnitude of the complex number:

$$\|\mathbf{w}\|_S \triangleq \sum_{n=0}^{N-1} s(|w[n]|). \quad (3.9)$$

This second interpretation of the sparsity of complex numbers implies that both the real and imaginary parts share the same sparse support to a great degree. These approaches can be extended to consider sparsity among multiple (real or complex) vectors; this notion is called group sparsity, joint sparsity, or simultaneous sparsity. If we treat these P vectors as the columns of a matrix \mathbf{W} , we can define a joint sparsity measure analogous to the first as

$$\|\mathbf{W}\|_{S,1} \triangleq \sum_{n=0}^{N-1} s(\|[W_1[n], \dots, W_P[n]]\|_1), \quad (3.10)$$

and another measure analogous to the second as

$$\|\mathbf{W}\|_{S,2} \triangleq \sum_{n=0}^{N-1} s(\|[W_1[n], \dots, W_P[n]]\|_2). \quad (3.11)$$

These joint sparsity penalty functions can be generalized to a generic measure

$$\|\mathbf{W}\|_{S,q} \triangleq \sum_{n=0}^{N-1} s(\|[W_1[n], \dots, W_P[n]]\|_q). \quad (3.12)$$

Another common such measure uses $q = \infty$, which forces strict shared sparsity support across all the vectors. In this work, we mainly focus on the mixed norm with $q = 2$.

3.2 Sparsity-Based Denoising

One valuable use of signal models like sparsity is estimation of the signal from noisy observations of that signal. When the model is appropriate for the signal, estimation using this information can improve the quality of the estimates substantially.

Suppose N observations \mathbf{d} are made of a sparse signal \mathbf{y} of length N using the full-rank observation matrix \mathbf{K}_a , and these observations are corrupted with additive independent identically distributed (iid) zero-mean complex Gaussian noise \mathbf{n} with variance σ^2 . Mathematically,

$$\mathbf{d} = \mathbf{K}_a \mathbf{y} + \mathbf{n}. \quad (3.13)$$

Without additional information about the signal, the minimum mean squared error (MMSE) optimal linear estimator for \mathbf{y} is the maximum likelihood (ML) estimator [46], which is also the least-squares solution of Equation (3.13),

$$\hat{\mathbf{y}} = (\mathbf{K}_a^T \mathbf{K}_a)^{-1} \mathbf{K}_a^T \mathbf{d}. \quad (3.14)$$

Now, suppose the signal \mathbf{y} is known to be zero-mean, approximately sparse, and uncorrelated with the noise. In particular, let the sparse transform coefficients $\mathbf{w} = \mathbf{\Psi} \mathbf{y}$. One sparsity-promoting choice of prior is for \mathbf{w} to be independent and identically distributed with $p(w[n]) \propto e^{-\lambda|w[n]|}$ for some choice of λ . When $w[n]$ is real,

$$p(w[n]) = \frac{\lambda}{2} e^{-\lambda|w[n]|}, \quad (3.15)$$

and when $w[n]$ is complex,

$$p(w[n]) = \frac{\lambda^2}{2\pi} e^{-\lambda|w[n]|}. \quad (3.16)$$

The linear MMSE estimator is

$$\hat{\mathbf{y}} = (\mathbf{\Lambda}_y \mathbf{K}_a^T) (\mathbf{K}_a \mathbf{\Lambda}_y \mathbf{K}_a^T + \sigma^2 \mathbf{I})^{-1} \mathbf{d}, \quad (3.17)$$

where $\mathbf{\Lambda}_y$ is the signal covariance matrix. Suppose the forward transform is invertible,

so $\mathbf{\Lambda}_y = \sigma_w^2 \mathbf{\Psi}^{-1} \mathbf{\Psi}^{-H}$ (recall the $w[n]$'s are iid). For the real case, the variance is $\sigma_w^2 = 2/\lambda^2$, and for the complex case, the variance is $\sigma_w^2 = 6/\lambda^2$.

3.2.1 Multiple Measurement Model

This framework can be extended to the multiple measurement vector case

$$\mathbf{D} = \mathbf{K}_a \mathbf{Y} + \mathbf{N}, \quad (3.18)$$

where \mathbf{D} is the $N \times P$ matrix of observations, the columns of \mathbf{Y} are jointly sparse vectors with respect to the sparsifying transform $\mathbf{\Psi}$, and \mathbf{N} is a matrix of zero-mean complex Gaussian additive noise, with independent rows, and each row has covariance matrix $\mathbf{\Lambda}$. To cast this problem in the single vector framework, we form column vectors by stacking the columns of the matrices \mathbf{D} , \mathbf{Y} , and \mathbf{N} :

$$\text{vec}(\mathbf{D}) = (\mathbf{I}_{P \times P} \otimes \mathbf{K}_a) \text{vec}(\mathbf{Y}) + \text{vec}(\mathbf{N}), \quad (3.19)$$

where $\text{vec}(\cdot)$ is the column vectorization of a matrix, and \otimes is the Kronecker product.

The matrix of sparse coefficient vectors $\mathbf{W} = \mathbf{\Psi} \mathbf{Y}$ has independent rows, with each row distributed according to $p(W_1[n], \dots, W_P[n]) \propto e^{-\lambda \| [W_1[n], \dots, W_P[n]] \|_2}$. When \mathbf{W} is real,

$$p(W_1[n], \dots, W_P[n]) = \frac{\Gamma(P/2 + 1) \lambda^P}{P! \pi^{P/2}} e^{-\lambda \| [W_1[n], \dots, W_P[n]] \|_2}, \quad (3.20)$$

where $\Gamma(\cdot)$ is the Gamma function, and when \mathbf{W} is complex,

$$p(W_1[n], \dots, W_P[n]) = \frac{P! \lambda^{2P}}{(2P)! \pi^P} e^{-\lambda \| [W_1[n], \dots, W_P[n]] \|_2}. \quad (3.21)$$

For a single row, it is easy to show that the sparse coefficients are uncorrelated and have identical variances $\sigma_w^2 = (P + 1)/\lambda^2$ (real case) or $\sigma_w^2 = 2(2P + 1)/\lambda^2$ (complex case).

Plugging the observation matrix from Equation (3.19) and the variances into

Equation (3.17) yields the linear estimator

$$\text{vec}(\hat{\mathbf{Y}}) = \mathbf{\Lambda}_{\mathbf{Y},\mathbf{D}} \mathbf{\Lambda}_{\mathbf{D}}^{-1} \text{vec}(\mathbf{D}), \quad (3.22)$$

where

$$\mathbf{\Lambda}_{\mathbf{Y},\mathbf{D}} = \sigma_w^2 \mathbf{I}_{P \times P} \otimes \mathbf{\Psi}^{-1} \mathbf{\Psi}^{-H} \mathbf{K}_a^T, \quad (3.23)$$

and

$$\mathbf{\Lambda}_{\mathbf{D}} = \sigma_w^2 \mathbf{I}_{P \times P} \otimes \mathbf{K}_a \mathbf{\Psi}^{-1} \mathbf{\Psi}^{-H} \mathbf{K}_a^T + \mathbf{\Lambda} \otimes \mathbf{I}_{N \times N}. \quad (3.24)$$

Especially in the multiple measurement vector case, the matrix $\mathbf{\Lambda}_{\mathbf{D}}$ may be too large to store or invert directly, but numerical inversion is possible using methods that will be discussed later.

3.2.2 Regularization with Sparsity

While linear MMSE estimation is possible, the linear estimate typically is not (approximately) sparse. To yield an (approximately) sparse estimate, we turn to the maximum *a posteriori* (MAP) estimator, which maximizes $p(\mathbf{y} \mid \mathbf{d})$. Since \mathbf{d} is observed, MAP estimation is equivalent to maximizing the joint distribution

$$p(\mathbf{y}, \mathbf{d}) = p(\mathbf{d} \mid \mathbf{y}) p(\mathbf{y}) \quad (3.25)$$

$$\propto \exp\left\{-\frac{\|\mathbf{d} - \mathbf{K}_a \mathbf{y}\|_2^2}{2\sigma^2}\right\} \exp\{-\lambda \|\mathbf{\Psi} \mathbf{y}\|_1\}. \quad (3.26)$$

This problem is most efficiently solved in the log domain:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{minimize}} \frac{1}{2\sigma^2} \|\mathbf{d} - \mathbf{K}_a \mathbf{y}\|_2^2 + \lambda \|\mathbf{\Psi} \mathbf{y}\|_1. \quad (3.27)$$

The above formulation is not unlike basis pursuit denoising (BPDN) or the Lasso [89], and can be solved using any of a wide variety of iterative techniques developed for either framework [61, 92, 101, 36]. The extension to multiple jointly sparse measure-

ment vectors is similar:

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{minimize}} \frac{1}{2} \|\text{vec}(\mathbf{D} - \mathbf{K}_a \mathbf{Y})\|_{\mathbf{\Lambda} \otimes \mathbf{I}_{N \times N}}^2 + \lambda \|\Psi \mathbf{Y}\|_{1,2}. \quad (3.28)$$

Methods for approximately solving this type of problem include iteratively reweighted least squares (IRLS), half-quadratic minimization, and interior point methods for semi-definite programming [43, 34, 104].

3.3 Compressed Sensing Reconstruction

The denoising problem is presented for a complete set of observations; however, accelerated imaging provides an incomplete set of M observations (or an $M \times P$ matrix for parallel imaging). One approach to overcoming this limitation is to use some standard reconstruction technique (like GRAPPA for accelerated parallel imaging) to form an augmented dataset and denoise the result; this approach is investigated in Chapter 4. Alternatively, sparsity can be used as prior information to recover the complete signal. The popular framework for doing this is called compressed sensing (CS) [18, 16, 20, 27]. The key result of CS is that it is possible to obtain robust reconstructions of a length- N signal from $M < N$ observations, where M is on the order of K , the sparsity of the signal. The restricted isometry property and mutual coherence bound provide sufficient conditions for reconstruction of a sparse signal from a number of measurements on the order of $K \log N$, drastically reducing the number of observations required for sparse signals like many MRI images.

3.3.1 The Compressed Sensing Framework

Consider the observation model from Equation (3.13), with the observation vector \mathbf{d} now having length $M < N$. The matrix \mathbf{K}_a is now “fat” and does not have a left inverse. Thus, the linear MMSE estimator in Equation (3.17) no longer applies. However, we can still form the MAP estimate, using Equation (3.27) and iterative solution techniques for BPDN or the Lasso. This MAP estimate yields an approxi-

mately sparse solution even when the rank of \mathbf{K}_a is less than N , and when this matrix is chosen to satisfy certain properties, this method can be exceptionally effective even when $M \ll N$.

The CS framework has three basic components: the sparsifying transform, the sampling pattern, and nonlinear reconstruction. We have already presented the reconstruction method – it is the MAP estimator described before. Now, we turn to the importance of the two other components. The sparsifying transform Ψ defines a representation of the signal \mathbf{y} such that this representation of the desired signal is approximately sparse. The sparsity of the signal in this domain is connected to the minimum number of observations needed to reconstruct the signal. If the sparsity support were known *a priori*, it is obvious that we only need as many linearly independent measurements as there are nonzero sparse coefficients; \mathbf{K}_a multiplied by the appropriate columns of Ψ would be a square full-rank matrix, and the signal \mathbf{y} can be found directly. When we do not know the support, as many as N measurements would be needed if sparsity were not assumed. However, the use of random observations with sufficient incoherence means that we only need on the order of $K \log N$ measurements, provided certain conditions on \mathbf{K}_a are met [18].

Key to achieving a quality reconstruction from undersampled data is the observation matrix \mathbf{K}_a . Precise conditions are given in the literature [27], but the essential message is that the choice of matrix should mitigate the possibility that the sparse signal lies in the nullspace of the matrix. Common examples of observation matrices for CS include a random matrix with iid Gaussian entries and a randomly chosen subset of a unitary matrix like a DFT matrix. Conditions like the restricted isometry property (RIP) [19] and mutual coherence bounds [17] on the observation matrix can be used to assess the suitability of the matrix for CS as well as its effect on the quality of the estimated signal. Given a matrix \mathbf{K}_a and sparsity level $K \leq N$, the RIP defines a constant δ_K that satisfies

$$(1 - \delta_K) \|\mathbf{w}\|^2 \leq \|\mathbf{K}_a \Psi^{-1} \mathbf{w}\|^2 \leq (1 + \delta_K) \|\mathbf{w}\|^2, \quad (3.29)$$

for all K -sparse coefficient vectors \mathbf{w} . The RIP constants measure how well the ℓ_2 norm is preserved by the observation matrix for the class of K -sparse vectors. Of course, a true isometry is not possible with $M < N$, but we could consider an observation matrix with small RIP constants to be nearly unitary in a sense.

While determining the RIP constants for a matrix is challenging, the mutual coherence $\mu(\mathbf{K}_a \Psi^{-1})$ is straightforward to determine:

$$\mu(\mathbf{K}_a \Psi^{-1}) = \sqrt{N} \max_{i,j} |\langle k_i, \tilde{\psi}_j \rangle|, \quad (3.30)$$

for rows k_i of \mathbf{K}_a and columns $\tilde{\psi}_j$ of Ψ^{-1} ($\langle \cdot, \cdot \rangle$ is the inner product of two vectors). Note that these columns $\tilde{\psi}_j$ form a sparsifying basis for a signal \mathbf{y} : $\mathbf{y} = \sum_j w[j] \tilde{\psi}_j$. The number of measurements sufficient for CS to succeed is proportional to $\mu^2(\mathbf{K}_a \Psi^{-1}) K \log N$ [17]. A different approach to performance analysis for CS that can be extended to a larger class of regularized least-squares estimators relies on the Replica method to simplify a complex optimization problem into a scalar estimation problem, enabling precise error and support recovery analysis in a computationally tractable way [79]. While these theoretical results are useful when the sparsity level is known, empirical methods like cross validation may be useful to ascertain reconstruction quality when the image sparsity is unknown [94].

Now, we go back to the optimization problem used for reconstruction. The CS framework is more general than the MAP estimator in Equation (3.27). If the observations are exact (no additive noise), we can solve the basis pursuit problem

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{minimize}} \|\Psi \mathbf{y}\|_1 \quad \text{s.t.} \quad \mathbf{d} = \mathbf{K}_a \mathbf{y}. \quad (3.31)$$

When the observations are noisy, we can solve either the unconstrained problem corresponding to MAP estimation discussed earlier or the constrained optimizations

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{minimize}} \|\Psi \mathbf{y}\|_1 \quad \text{s.t.} \quad \|\mathbf{d} - \mathbf{K}_a \mathbf{y}\|_2^2 \leq \varepsilon; \quad (3.32)$$

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{minimize}} \|\mathbf{d} - \mathbf{K}_a \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\Psi \mathbf{y}\|_1 \leq K. \quad (3.33)$$

The problem in Equation (3.33) is known as the Lasso [89]. Both of these denoising CS variants are equivalent to the MAP estimation problem with appropriate choices of λ . A related problem is the Dantzig selector [21]:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{minimize}} \|\Psi\mathbf{y}\|_1 \quad \text{s.t.} \quad \|\mathbf{K}_a^T(\mathbf{d} - \mathbf{K}_a\mathbf{y})\|_\infty \leq (1 + t^{-1})\sigma\sqrt{2\log N}, \quad (3.34)$$

where $t > 0$ is a scalar parameter connected to the RIP constants of \mathbf{K}_a .

3.3.2 Compressed Sensing with Joint Sparsity

Compressed sensing can be extended to capture dependencies between multiple measurement vectors. Joint sparsity, described previously to capture shared support across all the measurement vectors, has an analogue in CS: distributed compressed sensing [3]. Using a joint sparsity model and an ℓ_1 norm-based reconstruction, results demonstrated a reduction in the number of observations necessary to reach the same distortion level for multiple measurement vectors with common sparse support. Alternatively, the reduce-and-boost method can be applied to transform the joint sparsity CS reconstruction problem into a conventional CS reconstruction problem [64].

Similar to joint sparsity, structured group sparsity accounts for knowledge of shared support across particular subsets of the set of multiple measurement vectors [45]. These subsets may overlap, in which case the component vectors of each subset have to be weighted carefully. This approach can be employed to leverage information like persistence across scales in the wavelet transform representation of natural images [2].

A looser interpretation of joint sparsity, where sparse supports do not necessarily overlap significantly but have some common degree of sparsity, Bayesian multi-task compressed sensing has been applied to the joint reconstruction of several different types of MRI images (e.g. T_1 - and T_2 -weighted data) [8]. This hierarchical Bayesian model shares information across images via a single hyperparameter common to the sparsity-promoting prior distributions of all the images.

3.4 Compressed Sensing for MRI

Compressed sensing has been applied previously to magnetic resonance imaging to enable the reconstruction of high quality images from accelerated acquisitions [57]. Many MR images are compressible in wavelet or finite differences transform domains, and observations in k-space are incoherent (have low mutual coherence) with respect to both of these bases. Additionally, random undersampling of k-space is practical in MRI due to the precise control over the gradient magnetic fields available during acquisition.

In addition to the commonly used discrete wavelet transform (DWT) and finite difference transform as sparse domains for natural images, several other sparse transform domains have been found to be favorable to MR images. Overcomplete transforms like the curvelet, contourlet, and shearlet extend the multiresolution idea of wavelet transforms to incorporate directional information [15, 26, 39]. CS can also accommodate adaptive and learned dictionaries, learning the sparsifying transform using training data prior to reconstruction or during reconstruction from the acquired data [24, 80]. A training set of MR images can be decomposed into small patches, and a sparse dictionary can be learned using a method like K-SVD [1]. A learned dictionary captures details particular to MR images more effectively than generic transforms and preserves those details in the reconstructed image. However, use of learned dictionaries may be found unacceptable in clinical MRI due to the bias towards features found in the training set, which may not include the abnormalities present in clinical scans.

A wide variety of k-space subsampling patterns have been investigated in the context of accelerated MRI. Early work eschewed uniformly random undersampling for variable-density sampling that samples the center of k-space, where the signal is concentrated, more densely than the outside [57]. Since mutual incoherence is strongly influenced by the k-space sampling pattern, it is desirable to choose a sampling pattern that minimizes the coherence of the sampling pattern. This coherence bound can be approximated by the sidelobe-to-peak ratio (SPR) of the point spread function (PSF)

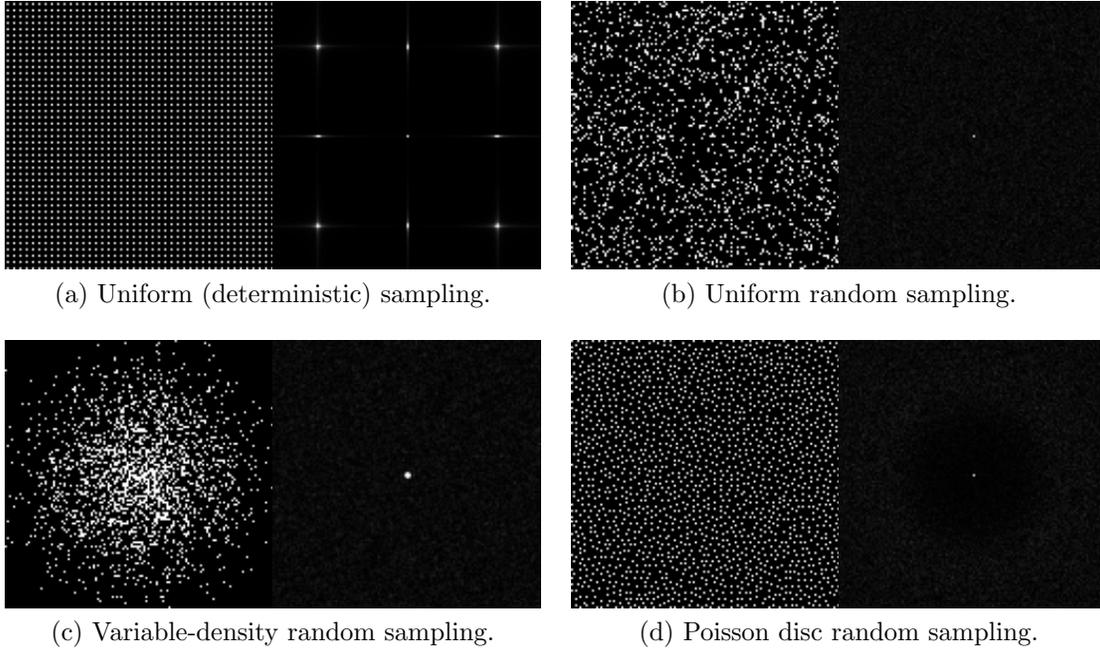


Figure 3.4: K-space undersampling patterns and their point spread functions.

in the image or sparse transform domain. One downside of random sampling patterns for GRAPPA-like methods is the large number of interpolation kernels that need to be calibrated, since the patterns of source and target points in each block of k-space are so variable. An approach to reducing the number of kernels is to tile the sampling pattern with a limited set of small randomly generated patches, so there are only a few different types of blocks that need interpolation kernels [49].

More recent work investigates Poisson disc sampling, which incorporates randomness while guaranteeing that samples are not clustered too close and gaps between samples are not too large, in accelerated parallel MR imaging [56, 77, 95]. Avoiding large gaps is especially useful for parallel imaging reconstruction methods like GRAPPA, since GRAPPA-like methods have difficulty approximating large frequency shifts with linear combinations of coil data with slowly varying sensitivities. Poisson disc sampling guarantees an upper bound on the gap size while producing a PSF very similar to uniformly distributed random sampling. Combinations of Poisson disc, variable density sampling, and tiled sampling are also possible [50]. Examples of these 2-D undersampling patterns, with their image domain PSFs, are shown in Figure 3.4.

3.4.1 Compressed Sensing and Parallel Imaging

Recent developments in accelerated MR image reconstruction include various combinations of accelerated parallel imaging methods and sparsity or compressed sensing. SparseSENSE [55] and CS-SENSE [52] combine CS with the SENSE parallel imaging method. SparseSENSE is a direct extension of the SparseMRI [57] framework to SENSE reconstruction:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \|\Psi\mathbf{x}\|_1 + \alpha\|\mathbf{x}\|_{\text{TV}} \quad \text{s.t.} \quad \|\mathbf{D} - \mathbf{K}_a\mathbf{F}[\mathbf{S}_1 \cdot \mathbf{x}, \dots, \mathbf{S}_P \cdot \mathbf{x}]\|_F^2 \leq \varepsilon, \quad (3.35)$$

where $\|\cdot\|_{\text{TV}}$ is a discrete approximation of the total variation measure using finite differences

$$\|\mathbf{x}\|_{\text{TV}} = \sum_{n=0}^{N-1} \sqrt{|\partial_x x[n]|^2 + |\partial_y x[n]|^2}, \quad (3.36)$$

$\partial_x x[n]$ is approximated by the first-order backward difference in the x -direction, $\partial_y x[n]$ is approximated by the first-order backward difference in the y -direction, and α and ε are tuning and noise parameters. CS-SENSE sequentially applies CS to clean up all the aliased coil images and SENSE to combine the sparsified coil images and generate a single un-aliased result. The downside of these approaches is that they still rely on accurate measurements of the coil sensitivities to optimally un-alias the image. Distributed CS can be used to improve the coil sensitivities' estimates and SparseSENSE or CS-SENSE to perform the reconstruction [75].

CS-GRAPPA [31] alternates nonconvex coil-by-coil CS and GRAPPA reconstruction steps on radially acquired data, re-inserting the gridded acquired data between each step, and iterating until convergence. This method is similar to the iterative GRAPPA [58] approach for reconstructing undersampled arbitrary (non-Cartesian) k-space, with CS incorporated between the steps, and uses GROG to re-grid the radially sampled data onto Cartesian k-space. Another sequential combination of CS and accelerated parallel imaging uses a GRAPPA-like reconstruction method to interpolate uniformly undersampled parts of k-space and CS to fill in the remaining k-space [5]. This iterative combination of GRAPPA and compressed sensing may not

be as effective as a joint combination of GRAPPA and compressed sensing.

The L_1 SPIRiT [56] method regularizes the SPIRiT method for accelerated parallel imaging reconstruction with joint sparsity across the coil images. The ℓ_1 norm-regularized version of data-preserving SPIRiT is

$$\underset{\mathbf{Y}}{\text{minimize}} \ \|(\mathbf{I} - \mathbf{G}) \text{vec}(\mathbf{Y})\|_2^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{1,2} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{K}_a \mathbf{Y}. \quad (3.37)$$

L_1 SPIRiT is implemented with Poisson disc (gridded) undersampled k-space and compares favorably to both SPIRiT and compressed sensing alone in preserving details and un-aliasing images at high accelerations. A weakness of L_1 SPIRiT is the reliance on non-uniform sampling and compressed sensing to undo aliasing; conventionally acquired accelerated parallel imaging data with uniformly spaced k-space samples reconstructed using L_1 SPIRiT may have residual aliasing.

The success of previous work in unifying compressed sensing and parallel imaging does not obviate the goals of this dissertation; instead, successful methods like CS-SENSE and L_1 SPIRiT lend credence to the idea that similar success is possible combining sparsity with GRAPPA. Past developments rely heavily on the non-uniform undersampling of k-space to achieve greater acceleration; this thesis focuses on the reconstruction and post-processing of images from conventionally acquired uniformly spaced undersampled data. By attacking the failure modes of GRAPPA at high accelerations, namely noise amplification and poor kernel calibration, we aim to utilize sparsity in entirely different ways, enabling reconstructions on par with or superior to existing methods without modifying the acquisition process.

Chapter 4

Denoising GRAPPA with Sparsity

Reducing the acquisition time by decreasing the field of view diminishes the SNR by a factor of \sqrt{R} , and accelerated parallel imaging methods like GRAPPA typically introduce additional noise amplification. Taken together, the SNR loss from accelerated parallel imaging can be significant enough to visibly degrade the resulting combined image and hide important tissue contrast. An example dataset is undersampled by different acceleration factors and reconstructed using GRAPPA in Figure 4.1. The reconstructed image quality degrades significantly as the acceleration factor increases, to the point of being totally dominated by noise.

To denoise such reconstructed images, we leverage the transform sparsity of the desired image (in the DWT or finite differences domain) and the unstructured nature

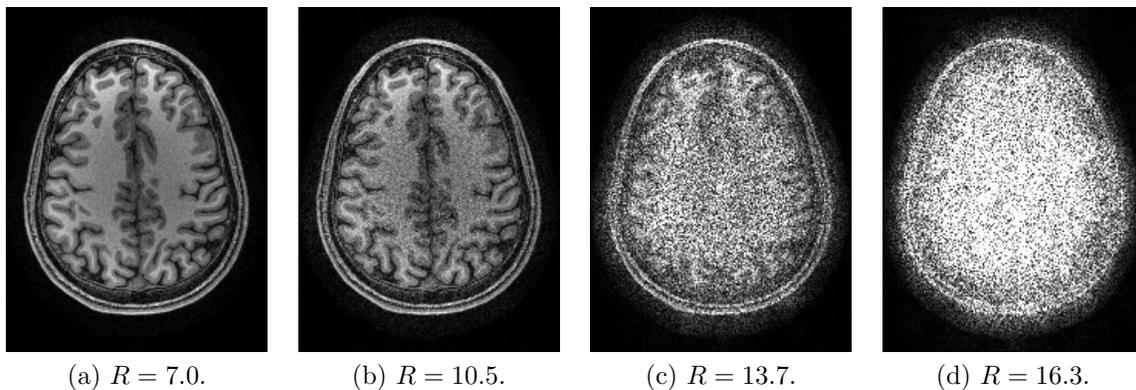


Figure 4.1: GRAPPA reconstructions for 2-D uniformly spaced (a) 3×3 , (b) 4×4 , (c) 5×5 , and (d) 6×6 nominal undersampling with increasing total acceleration factor R and a 32-channel coil at 3 T.

of the reconstructed noise. Taking the appropriate sparsifying transform concentrates the signal into a small number of transform coefficients, but leaves the noise spread out among all the coefficients. Thus, sparsifying the image preserves most of the signal energy while eliminating most of the noise energy in the reconstruction, effectively denoising the signal.

The Denoising Sparse Images from GRAPPA using the Nullspace method (DESIGN) effectively denoises the GRAPPA-reconstructed multi-channel k-space while preserving the acquired data, to minimize the bias on the reconstruction due to imposing a sparsity prior. This method regularizes the GRAPPA reconstructed k-space using the joint sparsity across all the DWTs of the coil images and operates in the nullspace of the observation matrix that subsamples the full k-space.

In this chapter, we first derive the proposed method and extend the framework to non-uniform undersampling. The simulations comparing the performance of the algorithm to that of other parallel imaging reconstruction and denoising techniques are presented and discussed, and conclusions about the proposed DESIGN method are discussed.

4.1 Theory

To denoise GRAPPA with sparsity, the objective function consists of two parts: (i) a least squares term to favor fidelity to the GRAPPA k-space result, and (ii) a measure $\|\cdot\|_{S,2}$ (see Equation (3.11)) that promotes joint sparsity across the coil images in the sparse transform domain. Not all k-space points in the GRAPPA result are equally good representations of the coil images, so we employ the coil combination weights in Equation (2.10) to weight the reconstruction fidelity in the image domain: we construct the matrix \mathbf{C} such that each column represents the combination weights for a single coil. Ignoring the effects of accelerated imaging, voxels in a coil with larger coil combination weights typically have higher SNR, so this approximation penalizes deviation from the parallel imaging reconstruction more greatly in those locations with greater SNR. While accelerated parallel imaging amplifies and colors the noise

in a spatially varying manner, this approximation is useful for denoising coils with different sensitivity profiles differently. Denote a tuning parameter λ that trades off these two objectives. We have

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{C} \cdot (\mathbf{F}^{-1}(\mathbf{Y} - \mathbf{G}(\mathbf{D})))\|_F^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{S,2} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{K}_a \mathbf{Y}. \quad (4.1)$$

Here, $\mathbf{G}(\mathbf{D})$ represents the GRAPPA-reconstructed full k-space using the acquired data \mathbf{D} . While the ideal GRAPPA kernel size may depend on the nominal under-sampling factor, we fix the kernel to be 3×3 blocks in size. Several approaches exist for solving constrained optimization problems [10]. Newton's method for solving unconstrained problems can be extended to constrained optimization problems directly. Alternatively, one can transform the constrained problem into an unconstrained problem by eliminating the equality constraint, after which any unconstrained optimization method can be used. Augmented or penalized Lagrangian methods produce a series of unconstrained problems that asymptotically approach the constrained problem [69]. A similar method, Bregman iteration, has been proposed for compressed sensing and related problems [102]. In this work, we take advantage of the simple form of the observation matrix \mathbf{K}_a to re-express the optimization as an unconstrained problem operating in the nullspace of \mathbf{K}_a . Since \mathbf{K}_a is a simple subsampling matrix, we utilize the nullspace decomposition $\mathbf{Y} = \mathbf{K}_a^T \mathbf{D} + \mathbf{K}_{na}^T \mathbf{X}$ with missing k-space \mathbf{X} and subsampling matrix \mathbf{K}_{na} that extracts the missing frequencies from the full set of k-space. Then, we rewrite Equation (4.1) as

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\text{minimize}} \|\mathbf{C} \cdot (\mathbf{F}^{-1} \mathbf{K}_{na}^T (\mathbf{X} - \mathbf{K}_{na} \mathbf{G}(\mathbf{D})))\|_F^2 + \lambda \|\Psi \mathbf{F}^{-1} (\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D})\|_{S,2}. \quad (4.2)$$

The full k-space solution is $\hat{\mathbf{Y}} = \mathbf{K}_{na}^T \hat{\mathbf{X}} + \mathbf{K}_a^T \mathbf{D}$. This unconstrained problem can be solved in several ways depending on the choice of penalty function $\|\cdot\|_{S,2}$. The general approach used here is IRLS, with the reweighting matrix $\Delta^{(i)}$ determined according to half-quadratic minimization. The least-squares problem to be solved in the i th

Algorithm 1 Algorithm for solving Equation (4.2) using IRLS.

Require: $\hat{\mathbf{X}}^{(0)}$, λ , ε , I , tol

- 1: Set $\hat{\mathbf{Y}}^{(0)} \leftarrow \mathbf{K}_{na}^T \hat{\mathbf{X}}^{(0)} + \mathbf{K}_a^T \mathbf{D}$ and compute initial objective value $f^{(0)}$ by plugging in $\hat{\mathbf{X}}^{(0)}$ into the objective in Equation (4.2).
 - 2: **for** $i = 1 : I$ **do**
 - 3: Compute $\mathbf{W}^{(i-1)} \leftarrow \Psi \mathbf{F}^{-1} \hat{\mathbf{Y}}^{(i-1)}$ and set $\Delta_{n,n}^{(i-1)} = \frac{1}{w[n]} \frac{ds(w)}{dw} \big|_{w=w[n]}$ for $w[n] = \|[W_{n,1}, \dots, W_{n,P}, \varepsilon]\|_2$, for all $n = 0, \dots, N - 1$.
 - 4: Solve for $\hat{\mathbf{X}}^{(i)}$ using Equation (4.3).
 - 5: Fill in the full k-space: $\hat{\mathbf{Y}}^{(i)} \leftarrow \mathbf{K}_{na}^T \hat{\mathbf{X}}^{(i)} + \mathbf{K}_a^T \mathbf{D}$.
 - 6: Compute $f^{(i)}$ by plugging in $\hat{\mathbf{X}}^{(i)}$ into the objective in Equation (4.2).
 - 7: **if** $f^{(i-1)} - f^{(i)} \leq \text{tol} \cdot f^{(i-1)}$ **then**
 - 8: **break**
 - 9: **end if**
 - 10: **end for**
 - 11: **return** Full k-space result $\hat{\mathbf{Y}}^{(i)}$.
-

iteration is

$$\hat{\mathbf{X}}^{(i)} \leftarrow \underset{\mathbf{X}}{\text{minimize}} \|\mathbf{C} \cdot (\mathbf{F}^{-1} \mathbf{K}_{na}^T (\mathbf{X} - \mathbf{K}_{na} \mathbf{G}(\mathbf{D})))\|_F^2 + \frac{\lambda}{2} \|(\Delta^{(i-1)})^{1/2} \Psi \mathbf{F}^{-1} (\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D})\|_F^2. \quad (4.3)$$

The IRLS algorithm for solving Equation (4.2) is depicted in Algorithm 1. Other approaches not investigated here include gradient descent/line search methods and interior point methods [10]. More details about IRLS and related methods can be found in Appendix A.

The drawback of IRLS and related approaches is the difficulty of solving Equation (4.3). A typical slice of an MRI volume may have as many as 256×256 voxels, meaning that \mathbf{Y} has 2^{16} rows. For a 32-channel coil, \mathbf{Y} has 2^{21} entries, over a million. For high accelerations, \mathbf{X} has almost as many entries as \mathbf{Y} , so we are often solving large-scale optimization problems with over a million variables. Direct inversion of a general least-squares problem with n variables is $O(n^3)$ complexity, so iterative methods are used to solve this problem instead. Differentiating the least squares problem yields the normal equations:

$$\mathbf{A}^H \mathbf{A} \mathbf{x} = \mathbf{A}^H \mathbf{b}. \quad (4.4)$$

For this problem, we have $\mathbf{x} = \text{vec}(\mathbf{X})$,

$$\mathbf{A} = \begin{bmatrix} \text{diag}(\text{vec}(\mathbf{C})) \\ \sqrt{\lambda}(\mathbf{I}_{P \times P} \otimes (\mathbf{\Delta}^{(i-1)})^{1/2} \mathbf{\Psi}) \end{bmatrix} (\mathbf{I}_{P \times P} \otimes \mathbf{F}^{-1} \mathbf{K}_{na}^T), \quad (4.5)$$

and

$$\mathbf{b} = \begin{bmatrix} \text{vec}(\mathbf{C} \cdot \mathbf{F}^{-1} \mathbf{K}_{na}^T \mathbf{K}_{na} \mathbf{G}(\mathbf{D})) \\ -\sqrt{\lambda} \text{vec}((\mathbf{\Delta}^{(i-1)})^{1/2} \mathbf{\Psi} \mathbf{F}^{-1} \mathbf{K}_a^T \mathbf{D}) \end{bmatrix}. \quad (4.6)$$

The conjugate transpose of \mathbf{A} is

$$\mathbf{A}^H = (\mathbf{I}_{P \times P} \otimes \frac{1}{N} \mathbf{K}_{na} \mathbf{F}) \left[\text{diag}(\text{vec}(\mathbf{C}^*)) \quad \sqrt{\lambda}(\mathbf{I}_{P \times P} \otimes \mathbf{\Psi}^H (\mathbf{\Delta}^{(i-1)})^{1/2}) \right]. \quad (4.7)$$

Since the system of normal equations is always Hermitian symmetric and positive semidefinite, conjugate gradient (CG) methods are popular for approximately solving Equation (4.4) using only a small number of iterations (much less than a million). Since the normal equations also solve the least squares problem $\mathbf{A}\mathbf{x} = \mathbf{b}$, the LSQR method [74, 73] can also be used. However, since these iterative methods are terminated early, the LSMR method [33] is used instead because it guarantees a monotonic decrease in both the normal residual $\|\mathbf{A}^H(\mathbf{b} - \mathbf{A}\mathbf{x})\|_2$, like the CG method, and the least-squares residual $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$, like LSQR. These iterative least-squares solvers are described and compared in Appendix A. These iterative approaches all require only that we can perform matrix-vector multiplies with \mathbf{A} and \mathbf{A}^H quickly; fortunately, these matrices are composed of DFTs, DWTs, and diagonal and subsampling matrices, which all can be implemented efficiently (complexity of $O(n \log n)$ or $O(n)$), and are even parallelizable for implementation on a GPU or multi-threaded CPU.

When greater sparsity is desired, we can combine the implementation defined above with homotopy continuation, starting with a sparsity-promoting penalty function $\|\cdot\|_{S,2}$ that is nearly convex and repeat with a sequence of nonconvex penalty functions that converge to the ℓ_0 “norm” that describes exact sparsity, initializing each successive problem with the solution to the previous iteration. One must take care as to not transform the penalty function too quickly, so the solution to the previ-

Algorithm 2 Algorithm for using homotopy continuation with the DESIGN denoising method.

Require: $\hat{\mathbf{X}}^{(0)}$, α_0 , α_{\max} , $\beta > 1$, I

- 1: $\alpha \leftarrow \alpha_0$.
 - 2: **for** $i = 1 : I$ **do**
 - 3: Run Algorithm 1 with $s_\alpha(\cdot)$ and initial guess $\hat{\mathbf{X}}^{(i-1)}$.
 - 4: $\hat{\mathbf{X}}^{(i)} \leftarrow \mathbf{K}_{na} \hat{\mathbf{Y}}^{(i)}$.
 - 5: $\alpha \leftarrow \alpha \cdot \beta$.
 - 6: **if** $\alpha > \alpha_{\max}$ **then**
 - 7: **break**
 - 8: **end if**
 - 9: **end for**
 - 10: **return** Full k-space result $\hat{\mathbf{Y}}^{(i)}$.
-

ous problem is sufficiently close to the solution to the next problem as to avoid local convergence issues caused by nonconvexity. The use of concave penalty functions like the ℓ_p^p ($p < 1$) “norm” or the Welsch penalty function can also cause dynamic range issues due to the sharp slope near zero or the nearly flat behavior away from the origin. The Cauchy penalty function is a good choice to avoid such issues because its slope near the origin increases linearly with α , and the function increases logarithmically fast away from the origin. The homotopy continuation algorithm is listed in Algorithm 2.

The DESIGN denoising method can be adapted to non-uniform undersampling of Cartesian k-space in two primary ways. The direct approach requires adapting the GRAPPA reconstruction to the non-uniform sampling pattern in question. Since the GRAPPA reconstruction would only be used once, either direct or iterative GRAPPA methods for non-uniform sampling can be used. Alternatively, we can introduce the subsampling matrices \mathbf{K}_s and \mathbf{K}_t that extract the normal source and target points used for GRAPPA reconstruction, respectively, from the full k-space estimate \mathbf{Y} and substitute $\mathbf{K}_s \mathbf{Y}$ for \mathbf{D} inside the GRAPPA fidelity term of Equation (4.1):

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{minimize}} \|\mathbf{C} \cdot (\mathbf{F}^{-1}(\mathbf{Y} - \mathbf{G}(\mathbf{K}_s \mathbf{Y})))\|_F^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{S,2} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{K}_a \mathbf{Y}. \quad (4.8)$$

The nullspace formulation for this problem is

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\text{minimize}} \|\mathbf{C} \cdot (\mathbf{F}^{-1}(\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D} - \mathbf{G}(\mathbf{K}_s(\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D}))))\|_F^2 + \lambda \|\Psi \mathbf{F}^{-1}(\mathbf{K}_{na}^T \mathbf{X} + \mathbf{K}_a^T \mathbf{D})\|_{S,2}, \quad (4.9)$$

followed by $\hat{\mathbf{Y}} = \mathbf{K}_{na}^T \hat{\mathbf{X}} + \mathbf{K}_a^T \mathbf{D}$, as before. This formulation is very similar to L_1 SPIRiT, with GRAPPA reconstruction fidelity used in place of the SPIRiT consistency equations. Like L_1 SPIRiT, GRAPPA must be applied in each iteration, and while implementable efficiently using fast Fourier transforms (FFTs), the additional computational burden is not insignificant. On a positive note, using non-uniform undersampling has the additional benefit of creating incoherent aliasing instead of the coherent aliasing observed when uniformly spaced undersampling is used, meaning that sparsity can be used here not only to denoise the GRAPPA result but also to augment parallel imaging in resolving and separating the aliased copies in the result. An acknowledged limitation of DESIGN applied to uniformly undersampled k-space is the total reliance on GRAPPA and parallel imaging to un-alias the coil images; sparsity does not really help here.

4.2 Simulations and Results

Both simulated and real acquired data are used in evaluation of the DESIGN denoising method. Simulated data consists of the Shepp-Logan phantom (available through the MATLAB `phantom` function) and contrast and resolution phantoms based on a compressed sensing phantom in the literature [86]. Multi-channel simulated data is synthesized from these datasets using the Biot-Savart Law-based B_1 simulator available online at [53]. The real data presented was previously acquired using a Siemens Tim Trio 3 T systems with a Siemens 32-channel head coil array. Two T_1 -weighted volumes were acquired using magnetization-prepared rapid gradient echo (MPRAGE) sequences ($256 \times 256 \times 176$ voxels, 1.0 mm isotropic resolution), and a T_2 -weighted image was acquired by a turbo spin echo (TSE) sequence ($264 \times 256 \times 23$

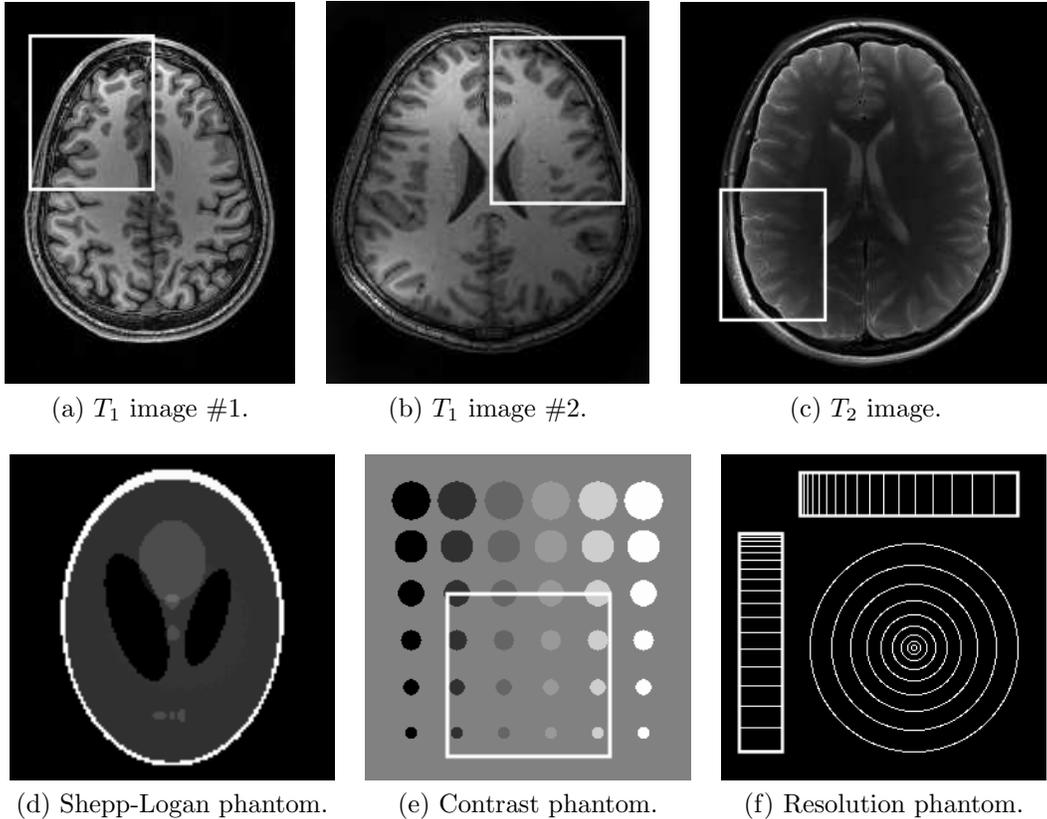


Figure 4.2: Ground truth magnitude images of real ((a)-(c)) and synthetic ((d)-(f)) datasets. Inset regions (white rectangles) are used later to show detail.

voxels, $0.75 \times 0.78 \times 5$ mm resolution). These sequences require 4 – 8 minutes to acquire the complete volumes. In addition, coil noise covariance measurements were computed from noise-only (no RF excitation) pre-scan acquisitions taken with the subject in the magnet.

Single slices to be used as examples are extracted from these datasets and cropped and normalized; combined magnitude images from these slices are retained as ground truth for the purpose of calculating quantitative and qualitative error metrics. The ground truth images for all the datasets are shown in Figure 4.2. Undersampled data is generated in MATLAB from these slices, including blocks of ACS lines to be used for calibration and in the reconstruction. As with GRAPPA, the reported total acceleration R includes all the acquired data, even the ACS lines.

Reconstructions are compared both qualitatively and quantitatively. Qualitative comparisons are made using the magnitude images and difference images between

the magnitude images of the reconstruction and ground truth. These comparisons depict relative noise levels, aliasing artifacts, and spatial resolution and blurring. Quantitative comparisons are made using peak signal-to-noise ratio (PSNR), which is computed in dB using the formula

$$\text{PSNR} = 20 \log_{10} \frac{\max_n |\text{ground truth}|}{\text{RMSE}}, \quad (4.10)$$

where the root mean squared error (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_n ||\text{recon. image}| - |\text{ground truth}||^2}. \quad (4.11)$$

While these quantitative measures are useful for comparing noise levels, PSNR or RMSE are not necessarily representative of image quality, especially where local distortions are concerned. For instance, a reconstructed image may have high PSNR, but a local abnormality a couple of pixels wide may be distorted or missing, eliminating critical information that may be present in another reconstruction. Thus, qualitative visual comparisons are emphasized throughout.

4.2.1 Penalty Functions and Continuation

We begin by exploring the choice of penalty function, and optionally, continuation, on the performance of the DESIGN denoising method [96]. Interpreting solving the DESIGN optimization problem as a form of MAP estimation, the choice of sparsity penalty function imposes a sort of prior distribution on the magnitude of transform coefficients of the signal. The ℓ_1 norm corresponds to an exponential distribution on the transform coefficient magnitudes, while the ℓ_p^p penalty corresponds to an exponential power distribution on the magnitudes. The Cauchy penalty function imposes a one-sided student-T distribution on the transform coefficient magnitudes (the one-sided Cauchy distribution is a special case for $\lambda = \log(1 + \alpha)$). These penalty functions and their associated distributions are listed in Table 4.1. The Cauchy penalty function for smaller λ , the ℓ_0 “norm”, and the Welsch penalty function all have density

Table 4.1: Penalty functions and associated sparsity priors on transform coefficient magnitudes.

$\ \mathbf{w}\ _S$	$s_\alpha(w[n])$	$p(w ; \lambda)$
ℓ_1	$ w[n] $	$\lambda e^{-\lambda w }$
$\ell_p^p, p < 1$	$ w[n] ^p$	$\frac{\lambda^{1/p}}{\Gamma(1+\alpha)} e^{-\lambda w ^p}$
Cauchy, $\frac{\lambda}{\log(1+\alpha)} > \frac{1}{2}$	$\frac{\log(1+\alpha w[n] ^2)}{\log(1+\alpha)}$	$\frac{2\sqrt{\alpha}\Gamma(\lambda/\log(1+\alpha))}{\sqrt{\pi}\Gamma(\lambda/\log(1+\alpha)-\frac{1}{2})} (1 + \alpha w ^2)^{-\lambda/\log(1+\alpha)}$

functions that are not integrable; their priors are called improper.

These sparsity priors are compared against the empirical distribution of the transform coefficient magnitudes observed for different datasets. The empirical cumulative distribution function (cdf) is defined as

$$F(w) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{1}_{\{|w[n]| \leq w\}}, w \geq 0. \quad (4.12)$$

We compare the empirical cdf to the sparsity prior cdf $F_s(w)$ using the Kolmogorov-Smirnov (K-S) test, which measures the maximum difference between the two cdf's:

$$\text{KS} = \max_{w \geq 0} |F(w) - F_s(w)|. \quad (4.13)$$

Because of the discrete nature of empirical distribution functions, it is enough to find the maximum over all $|w[n]|$ in the empirical dataset transform coefficient vector \mathbf{w} :

$$\text{KS} = \max_{n \in \{0, \dots, N-1\}} |F(|w[n]|) - F_s(|w[n]|)|. \quad (4.14)$$

Note that the endpoints are inconsequential since the cdf's are always equal at 0 and ∞ by definition. A comparison of cdf's for the ℓ_1 , ℓ_p^p ($p < 1$), and Cauchy penalty functions to the sparsity model of the Shepp-Logan phantom is portrayed in Figure 4.3. Another comparison for real data, T_1 -weighted image #1 is shown in Figure 4.4.

From the visual comparisons of sparsity models, the Cauchy penalty appears to be most appropriate for the simulated Shepp-Logan phantom, and the Cauchy and ℓ_p^p penalties are closest to the real T_1 -weighted data. To test how these penalty functions

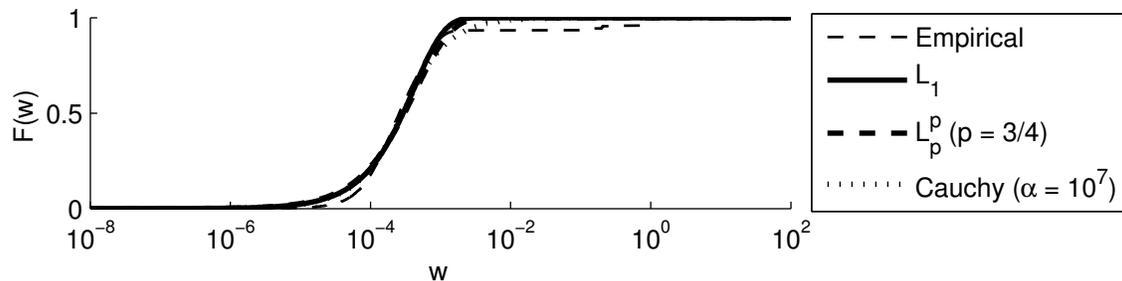


Figure 4.3: Sparsity model cdf's for the transform coefficient magnitudes of a Shepp-Logan phantom in the finite differences domain.

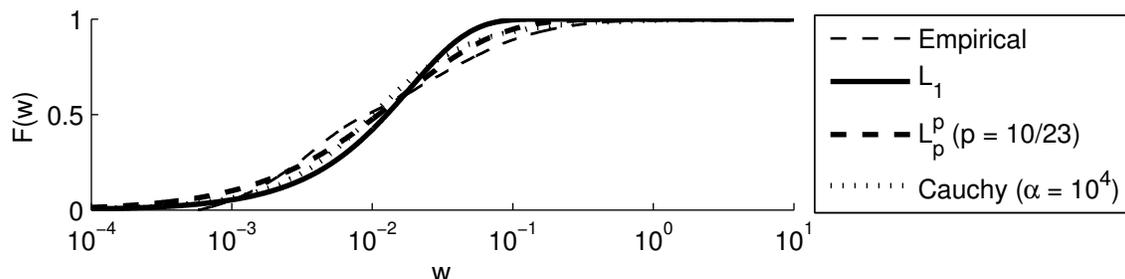


Figure 4.4: Sparsity model cdf's for the transform coefficient magnitudes of a T_1 -weighted real image in the four-level '9-7' DWT domain.

affect the denoising process, the DESIGN denoising method is applied to uniformly undersampled slices from both datasets using the three ℓ_1 , ℓ_p^p , and Cauchy penalty functions. Since the ℓ_p^p and Cauchy penalty functions are strongly nonconvex for small p /large α , homotopy continuation is used to avoid numerical and local convergence issues.

To ensure proper operation of the simulations involving homotopy continuation that follow, the effects of the continuation scheme parameters on the reconstruction quality and number of iterations is studied for the first T_1 -weighted real dataset. In particular, the continuation parameter multiplier β and the IRLS convergence tolerance tol are increased and the effects measured in terms of image quality and computation time. In this experiment, $\alpha_0 = 1$ and $\alpha_{\max} = 10^4$, and the computational cost is measured by the total number of LSMR iterations.

The effects of varying the continuation parameters are shown in Figure 4.5. The first reconstruction uses $\beta = 2$ to slowly increase the continuation parameter α between iterations and $\text{tol} = 0.01$ to ensure IRLS convergence between increasing the

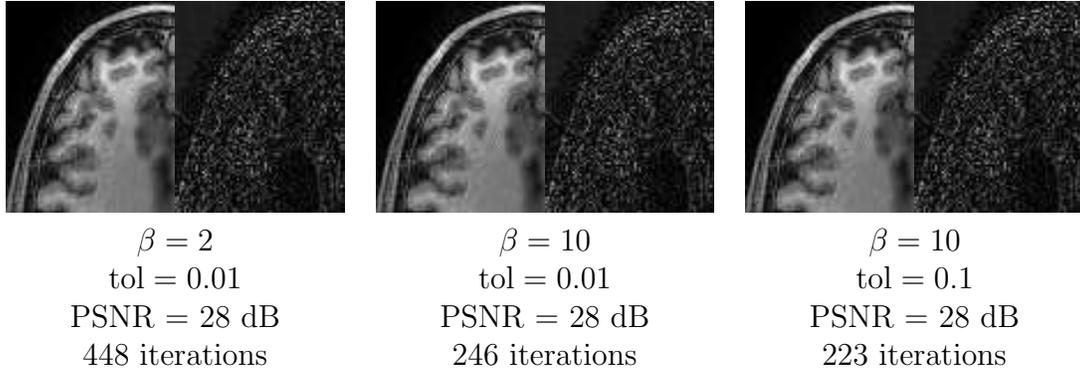
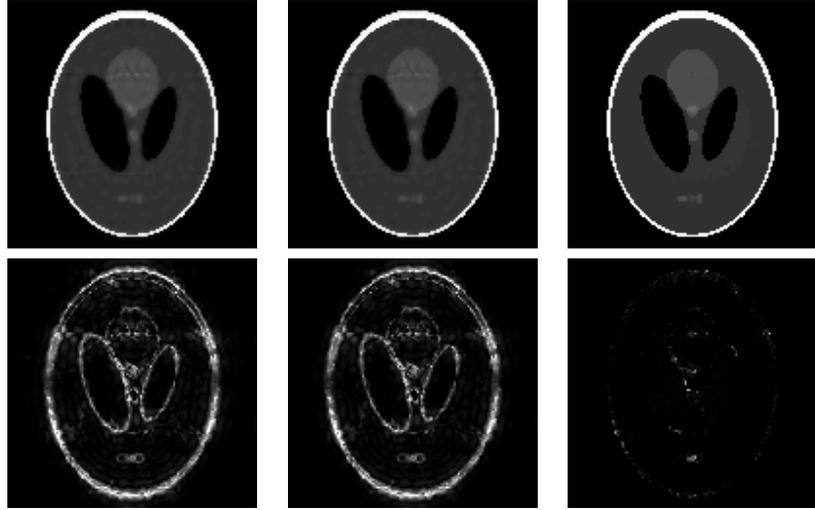


Figure 4.5: Continuation scheme parameters studied for DESIGN denoising of T_1 -weighted image #1 (4×4 nominal undersampling: $R = 10.5$) using the Cauchy penalty function with four-level ‘9-7’ DWT.

parameter. Increasing β to 10 or tol to 0.1 have negligible effects on image quality, and increasing β significantly decreases the number of LSMR iterations. Thus, we use $\beta = 10$ and tol = 0.01 for the Cauchy penalty in the following experiments.

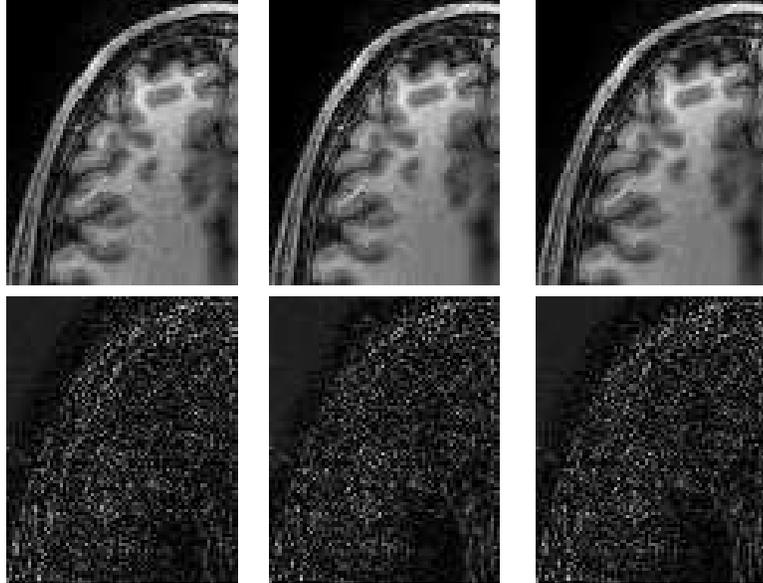
The DESIGN denoising reconstructions of the Shepp-Logan phantom depicted in Figure 4.6 suggest the potential for improvement from using nonconvex penalty functions. However, the differences in K-S test statistics among these models are too insignificant to conclude the existence of a trend between the model fit and image reconstruction quality. While noise is insignificant in all three reconstructions, the Cauchy penalty function does a much better job of mitigating aliasing of small features and blurring of edges in the phantom. However, the Shepp-Logan phantom is admittedly a poor approximation of real MR images, as it is very sparse, has high contrast, and well-defined edges. The improvement in PSNR is significant, so we ask if similar gains in reconstructed image quality can be realized for real MRI data.

The reconstructions portrayed in Figure 4.7 suggest that this improvement does not necessarily carry over to real data. The reconstructed images have approximately the same noise level, and the PSNR values are nearly identical. The ℓ_1 norm does a better job than the other sparsity penalty functions of preserving the sharpness of the edges, especially near the center of the image, where contrast is more limited. The two reconstructions using nonconvex penalty functions appear to be significantly oversmoothed, suggesting that these nonconvex penalty functions are not appropriate



Penalty function:	ℓ_1	ℓ_p^p ($p = 3/4$)	Cauchy ($\alpha = 10^7$)
Parameter λ :	$10^{1.6}$	$10^{1.6}$	$10^{1.4}$
PSNR:	35 dB	35 dB	51 dB
KS test statistic:	0.066	0.066	0.065

Figure 4.6: DESIGN denoising of Shepp-Logan phantom (4×4 nominal undersampling: $R = 8.7$) using ℓ_1 , ℓ_p^p , and Cauchy penalty functions with the finite differences representation.



Penalty function:	ℓ_1	ℓ_p^p ($p = 10/23$)	Cauchy ($\alpha = 10^4$)
Parameter λ :	$10^{0.4}$	$10^{0.6}$	1
PSNR:	28 dB	28 dB	28 dB
KS test statistic:	0.14	0.065	0.088

Figure 4.7: DESIGN denoising of T_1 -weighted image (4×4 nominal undersampling: $R = 10.5$) using ℓ_1 , ℓ_p^p , and Cauchy penalty functions with the four-level ‘9-7’ DWT.

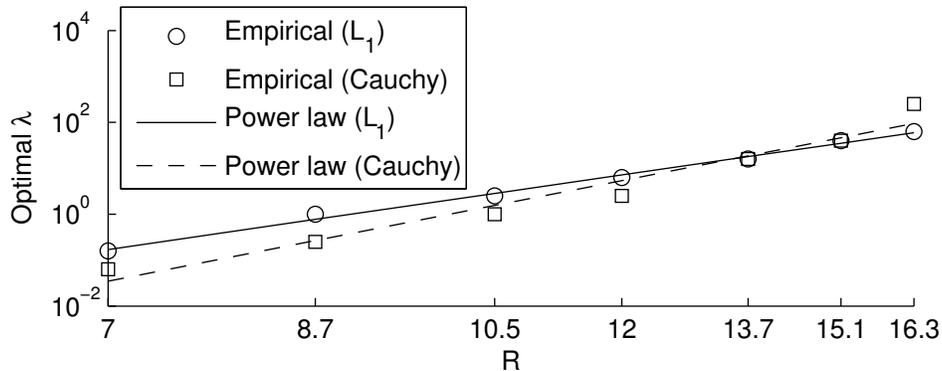


Figure 4.8: Trend in the optimal choice of λ as determined by coarse-then-fine parameter sweeps for the ℓ_1 norm and Cauchy penalty functions, for the T_1 -weighted image #1 with four-level ‘9-7’ DWT sparsifying transform. The nominal undersampling increases from 3×3 ($R = 7.0$) to 6×6 ($R = 16.3$).

for reconstructing data that is not truly sparse. The model fit as quantified by the K-S test statistic does not predict this oversmoothing and degradation in image quality. Whether this holds true for reconstructions using a learned or adaptive dictionary tailored to this dataset rather than a generic sparsifying transform like the DWT is not known, but the literature suggests that adaptive dictionary learning with small image patches is effective at reducing oversmoothing and loss of edge-definition due to sparsification [80], so nonconvex penalty functions may be more effective when combined with a dictionary.

4.2.2 Tuning Parameter Selection

The tuning parameter λ balances the fidelity to the GRAPPA solution and the sparsity of the result in Equation (4.2). A larger value of λ favors sparsity, while a smaller value of λ favors the GRAPPA solution. To determine the optimal value of λ , a two-level parameter sweep is used, a coarse sweep followed by a fine sweep around the optimal coarse sweep value. The trend in λ as a function of effective acceleration R is shown in Figure 4.8 for both the ℓ_1 and the Cauchy penalty functions. Both trends are approximated by power-law regressions $\lambda = 2.4 \times 10^{-7} R^{6.9}$ with $R^2 = 0.99$ and $\lambda = 4.3 \times 10^{-10} R^{9.4}$ with $R^2 = 0.96$ for the ℓ_1 norm and Cauchy penalty functions, respectively.

Algorithm 3 Multi-channel locally adaptive Wiener filter-based denoising algorithm.

Require: \mathbf{Y}, \mathbf{S}

- 1: Generate coil images $\mathbf{X} \leftarrow \mathbf{F}^{-1}\mathbf{Y}$.
 - 2: Form combined image \mathbf{x} using Equation (2.9) or Equation (2.10).
 - 3: Estimate global noise variance $\sigma^2 \leftarrow \text{median}(|\mathbf{w}_{\text{Hi}}|)/0.6745$, where \mathbf{w}_{Hi} are the high-pass coefficients of the DWT coefficients $\mathbf{w} = \mathbf{\Psi}\mathbf{x}$.
 - 4: **for** Each voxel $[x, y, z]$ **do**
 - 5: Get set of voxels ξ in $3 \times 3 \times 3$ neighborhood.
 - 6: Compute mean $\mu_x \leftarrow \frac{1}{|\xi|} \sum_{n=0}^{|\xi|-1} \xi[n]$ and variance $\sigma_x^2 \leftarrow \frac{1}{|\xi|} \sum_{n=0}^{|\xi|-1} |\xi[n]|^2 - |\mu_x|^2$.
 - 7: Mean $\boldsymbol{\mu}_x \leftarrow \mu_x \mathbf{S}[x, y, z]$.
 - 8: Signal covariance $\boldsymbol{\Lambda}_x \leftarrow \max\{0, \sigma_x^2 - \sigma^2\} \mathbf{S}[x, y, z] \mathbf{S}[x, y, z]^H$.
 - 9: Noise covariance $\boldsymbol{\Lambda} \leftarrow \sigma^2 \mathbf{S}[x, y, z] \mathbf{S}[x, y, z]^H$.
 - 10: Denoised voxel across coils is $\hat{\mathbf{x}}[x, y, z] \leftarrow \boldsymbol{\Lambda}_x (\boldsymbol{\Lambda}_x + \boldsymbol{\Lambda})^{-1} (\mathbf{x}[x, y, z] - \boldsymbol{\mu}_x) + \boldsymbol{\mu}_x$.
 - 11: **end for**
 - 12: **return** Full k-space result $\hat{\mathbf{Y}} \leftarrow \mathbf{F}\hat{\mathbf{X}}$.
-

The optimal choice of λ is thought to vary with the observation noise level and the sparsity of the image to be reconstructed, so the generality of this optimal choice of λ to other datasets is limited. Therefore, in the experiments conducted with multiple datasets, the parameter sweeps for λ are repeated for each dataset.

4.2.3 Performance Comparisons

The performance of the DESIGN denoising method is compared against a conventional multi-channel Wiener filter-based denoising method, multi-channel compressed sensing [37], and L_1 SPIRiT [98, 99]. Because the signal model statistics (mean and variance) are not known exactly, an adaptive Wiener filter-based approach that forms local estimates of the signal mean and variance is used [51]. The global noise variance is estimated using the median absolute deviation method with the same four-level DWT used as a sparsifying transform [28]. The signal and noise statistics are measured from the combined image, and multi-channel statistics are formed using low-resolution coil sensitivity estimates \mathbf{S} formed from the ACS lines apodized with a Blackman window [72]. The multi-channel Wiener filter-based denoising method is described in Algorithm 3.

Other compared methods include CS and L_1 SPIRiT. The nullspace approach

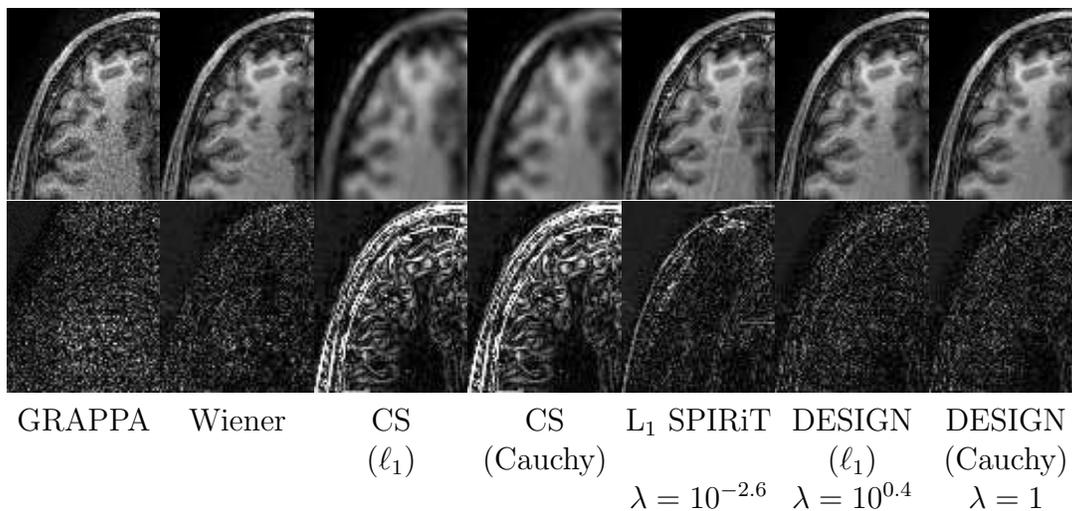


Figure 4.9: Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L_1 SPIRiT for $R = 10.5$ undersampled T_1 -weighted image #1 with four level ‘9-7’ DWT sparsifying transform.

and IRLS are used together to solve the joint sparsity version of basis pursuit. The parallelized implementation of L_1 SPIRiT [66] used for performance comparisons is available online from [65]. After running preliminary simulations, a 7×7 SPIRiT kernel size is chosen (the SPIRiT kernel size refers to the number of points in full-FOV k-space). The regularization parameter for the ℓ_1 term is determined via coarse-then-fine parameter sweeps, as is done for the proposed DESIGN denoising method, for each dataset.

Examples depicting the relative performance of the DESIGN denoising method are shown for three real datasets in Figures 4.9-4.11. The PSNRs of these reconstructions are listed for different effective accelerations R (nominal undersampling increases from 3×3 to 6×6) in Tables 4.2-4.4 (highest PSNRs in bold). From the reconstructed images and difference images, it is evident that when the tuning parameter λ is chosen correctly, both variants of the DESIGN method effectively denoise the GRAPPA reconstruction, better than the multi-channel Wiener filter-based denoising method. With uniformly spaced undersampling of k-space, the CS methods and L_1 SPIRiT are less effective than they could be with Poisson disc or random

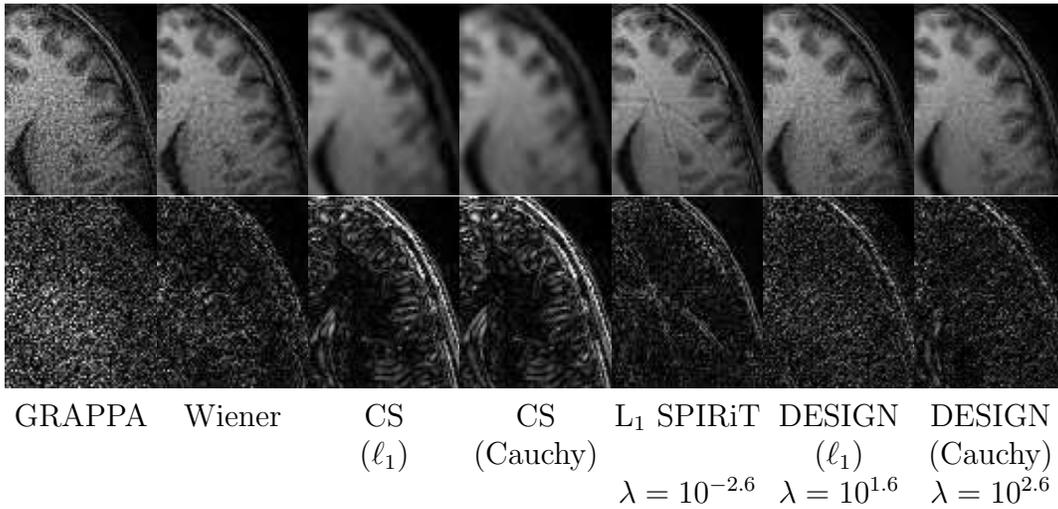


Figure 4.10: Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L₁ SPIRiT for $R = 10.5$ undersampled T_1 -weighted image #2 with four level ‘9-7’ DWT sparsifying transform.

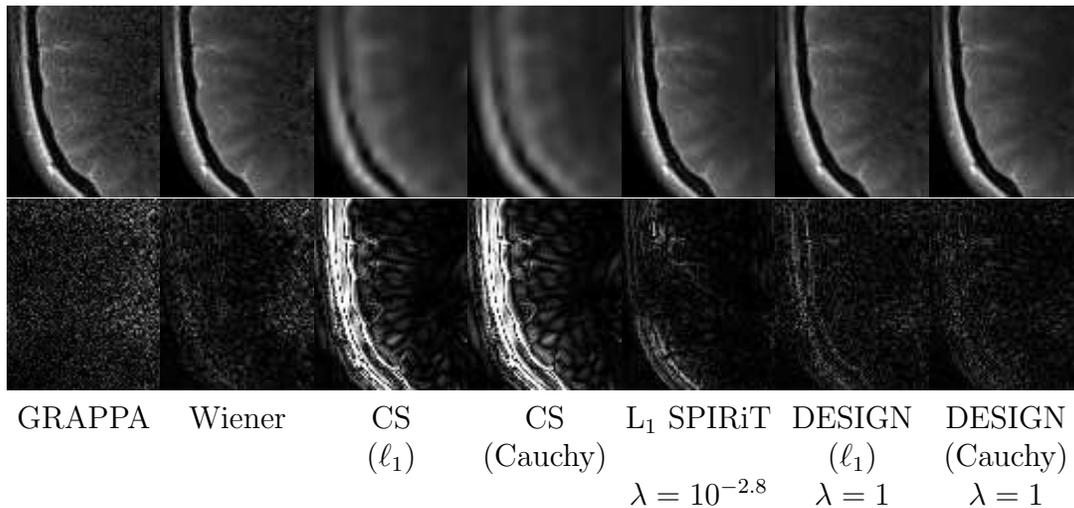


Figure 4.11: Reconstructed images (top row) and difference images (bottom row) for DESIGN denoising with ℓ_1 and Cauchy penalty functions compared to GRAPPA, GRAPPA with Wiener filter-based denoising, CS with joint sparsity and both ℓ_1 and Cauchy penalty functions, and L₁ SPIRiT for $R = 14.6$ undersampled T_2 -weighted image with four level ‘9-7’ DWT sparsifying transform.

Table 4.2: PSNRs (in dB) of reconstruction methods at different acceleration factors for T_1 -weighted image #1.

Method	R						
	7	8.7	10.5	12	13.7	15.1	16.3
GRAPPA	31.9	30.1	25.9	20.5	13.5	8.9	4.9
Wiener	31.1	30.2	27.9	23.4	15.7	10.9	7.1
CS (ℓ_1)	21.9	21.5	21.3	21.2	21.1	20.9	20.9
CS (Cauchy)	21.6	21.4	21.2	21.0	20.9	20.9	20.8
L_1 SPIRiT	31.3	30.1	28.3	26.9	25.5	24.6	23.9
DESIGN (ℓ_1)	32.1	30.7	28.2	25.5	22.9	21.9	21.2
DESIGN (Cauchy)	32.1	30.8	28.4	25.7	22.7	21.6	21.0

Table 4.3: PSNRs (in dB) of reconstruction methods at different acceleration factors for T_1 -weighted image #2.

Method	R						
	7.0	8.6	10.5	12.1	13.6	14.8	16.1
GRAPPA	33.1	29.5	24.5	18.9	12.5	8.1	4.4
Wiener	34.2	32.2	28.6	22.6	15.2	10.5	6.7
CS (ℓ_1)	25.4	25.1	24.9	24.8	24.8	24.7	24.6
CS (Cauchy)	24.8	24.6	24.4	24.3	24.3	24.2	24.2
L_1 SPIRiT	34.5	32.5	30.7	29.5	28.4	27.6	27.0
DESIGN (ℓ_1)	33.5	30.7	27.6	25.5	25.2	24.9	24.7
DESIGN (Cauchy)	34.2	31.9	29.4	27.0	25.3	24.8	24.6

Table 4.4: PSNRs (in dB) of reconstruction methods at different acceleration factors for T_2 -weighted image.

Method	R						
	7.8	9.9	12.4	14.6	17.1	19.3	21.4
GRAPPA	44.6	40.9	36.9	30.4	25.1	18.9	15.4
Wiener	43.6	41.8	39.5	33.3	27.5	20.7	17.2
CS (ℓ_1)	26.1	25.7	25.5	25.3	25.1	24.9	24.8
CS (Cauchy)	25.4	25.0	24.8	24.6	24.5	24.4	24.3
L_1 SPIRiT	44.3	40.7	37.2	34.2	32.3	30.6	29.5
DESIGN (ℓ_1)	45.0	41.9	38.8	34.5	31.3	28.3	27.1
DESIGN (Cauchy)	45.0	42.1	39.5	35.5	32.5	29.1	27.6

sampling patterns that have become the norm for CS-based approaches, and the L_1 SPIRiT results in Figures 4.10 and 4.11 contain residual aliasing not noticeable in the other reconstructions. Comparing the ℓ_1 and Cauchy penalty function variants of DESIGN, we observe that the ℓ_1 norm version introduces less oversmoothing into the reconstructed image, preserving edges and contrast better, especially near the center of the image. Thus, even though the Cauchy version of DESIGN has generally better PSNR, the ℓ_1 version has better image quality.

4.2.4 G-Factors

The reconstructions presented in the performance comparisons in the previous section display various types of errors. Noise amplification is a significant problem in highly accelerated parallel imaging, amplifying random noise to the point of masking low-contrast or small features. Residual aliasing artifacts may also be prevalent at high accelerations. Finally, denoising methods can introduce smoothing or blurring artifacts, reducing contrast and edge-definition and smoothing over small features. To measure the noise amplification due to the parallel imaging reconstruction method, we estimate the g-factors for each method using the pseudo multiple replica method. In the following experiment, we add synthetic complex Gaussian noise with covariance equal to the measured noise covariance $\mathbf{\Lambda}$ and examine the difference in reconstructed images before and after noise is added. We average the g-factors over 400 trials.

In Figure 4.12, the g-factors are plotted for each method for the undersampled image reconstructed in Figure 4.9. These g-factors demonstrate that both DESIGN denoising methods are fairly effective at mitigating noise amplification, on average yielding images only a few dB worse noise amplification than aliased images produced directly from reduced-FOV data. The L_1 SPIRiT method is also very effective at denoising, with an average g-factor of nearly one. The multi-channel Wiener filter denoising method is not as effective on average as any of these methods, but the average noise amplification is reduced by almost 4 dB relative to GRAPPA alone.

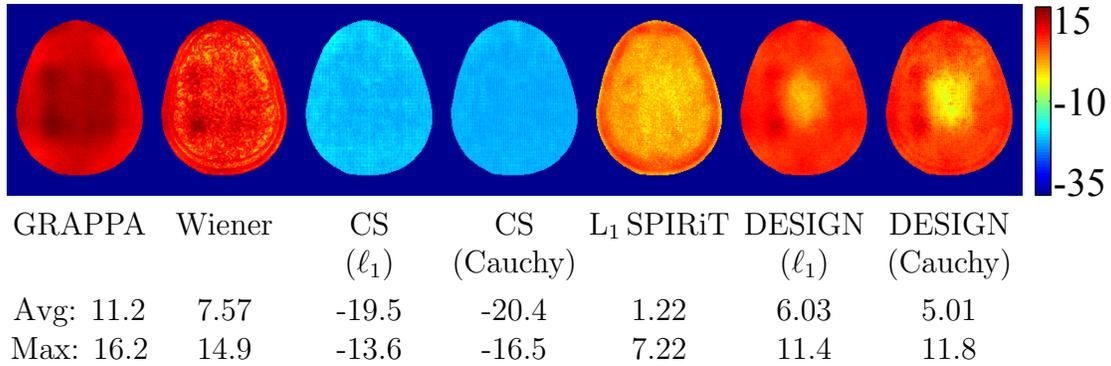
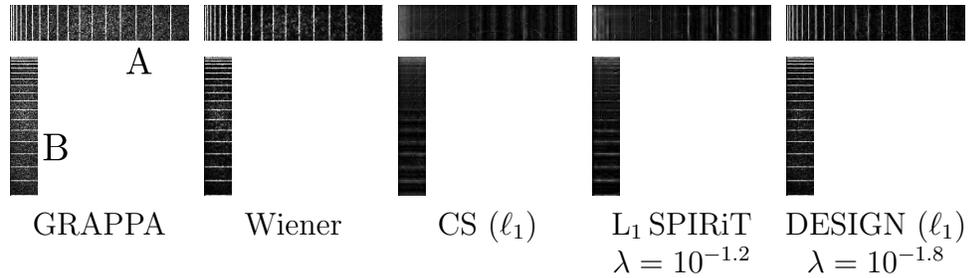


Figure 4.12: Estimated g-factors (in dB) are plotted for T_1 -weighted image #1 with $R = 10.5$ acceleration factor (4×4 nominal undersampling).

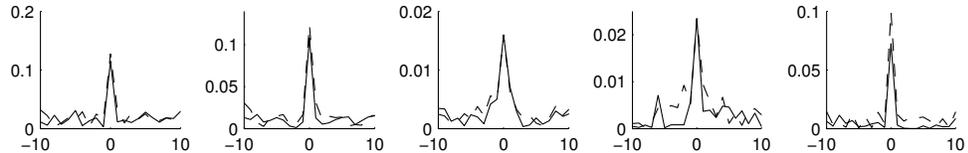
4.2.5 Oversmoothing Effects

Denoising methods are known to reduce the amount of detail in an image via oversmoothing. This reduction can be manifested as a loss of spatial resolution, of edge-definition, or of contrast in low-contrast areas. While these effects may be inconsequential in many image or video denoising applications, the loss of resolution or contrast may affect the utility of such medical images to practitioners. To study the prevalence of such effects in images processed using DESIGN denoising, we turn to contrast and resolution phantoms with features designed specifically to observe these oversmoothing effects and quantify contrast or resolution loss.

We use the synthetic contrast phantom in Figure 4.2 to quantify contrast loss in each of these denoising/reconstruction methods. This phantom has six rows of differently sized discs, with contrast between the circles and the background varying linearly across columns. Enough complex Gaussian noise is added to the reduced-FOV data to yield a GRAPPA reconstructed image with noise masking the lowest contrast circles almost completely. Wiener filter-based denoising, ℓ_1 norm-based CS, L_1 SPIRiT, and ℓ_1 norm-based DESIGN denoising are performed, and their results in Figure 4.13 indicate which contrast circles were eliminated from the result, and which were preserved. The Wiener filter was particularly poor at preserving the lowest contrast circles, while the CS, L_1 SPIRiT, and DESIGN denoising methods preserved all the circles, even the smallest ones, with minimal contrast loss ($\approx 2\%$, where the



Horizontal (Solid) and Vertical (Dashed) PSF Curves



Horz. Res.:	1.1 mm	1.1 mm	1.6 mm	1.3 mm	1.0 mm
Vert. Res.:	1.3 mm	1.3 mm	1.5 mm	1.4 mm	1.1 mm

Figure 4.14: Synthetic resolution phantom reconstruction comparisons of effective spatial resolutions using GRAPPA and denoising methods (4×4 nominal undersampling: $R = 12.1$) with a four-level ‘9-7’ DWT sparsifying transform.

For the resolution phantoms examined here, we use different image regions for measuring horizontal resolution and vertical resolution. Thus, we window and apodize the original and reconstructed images prior to computing PSFs.

Once the PSFs are computed, we locate the peak and measure the full width at half maximum (FWHM) around that peak to determine the resolution. Since we are dealing with two-dimensional PSFs, a single resolution number would correspond to the maximum FWHM in all directions around the peak. Since we measure spatial resolution specifically in horizontal and vertical directions, it suffices to measure the FWHM in the horizontal direction (for the horizontal resolution) and in the vertical direction (for the vertical resolution). Since we only have samples of the PSF, and the resolution is expected to be on the order of a voxel, we linearly interpolate the PSF between points to refine our estimate.

The experiment in Figure 4.14 demonstrates the effects of denoising on the effective spatial resolution. The selected regions in the synthetic phantom are designed to illustrate the importance of resolution, as the ℓ_1 norm-based CS method blurs the lines together as to make them indistinguishable as separate lines. Regions A and

B are used to measure horizontal and vertical resolution, respectively. This effect would be expected to become more noticeable as the noise level increases, as greater sparsification would be needed to reproduce a denoised image. From this synthetic example, the DESIGN denoising method appears to have no worse spatial resolution than GRAPPA alone, and performs better than the other denoising methods evaluated here.

4.2.6 Sampling Patterns

One of the requirements for conventional compressed sensing is that the observation matrix is random in some fashion to minimize the probability of a significant component of the signal falling into the nullspace of the observation matrix. The earlier experiments all handicap the CS and L_1 SPIRiT methods by constraining the sampling pattern to be uniformly-spaced, resulting in coherent aliasing. The DESIGN method as presented is designed for uniformly-spaced subsampling of k-space to accommodate conventional GRAPPA accelerated parallel imaging reconstruction. However, as shown earlier, we can extend GRAPPA, and hence DESIGN denoising, to non-uniform sampling patterns that would be more appropriate for CS. In the experiments that follow, we investigate the performance of GRAPPA and DESIGN using random and Poisson disc sampling patterns. For computational efficiency, we use a direct GRAPPA implementation for non-uniform subsampled Cartesian k-space, and use this GRAPPA result without otherwise modifying the DESIGN denoising algorithm.

Comparing the GRAPPA and DESIGN denoising reconstructions of the uniformly and randomly undersampled Shepp-Logan phantom data displayed in Figure 4.15, the GRAPPA method appears significantly adversely affected by random sampling, but combining random sampling with DESIGN denoising yields improved reconstructions for both ℓ_1 and Cauchy penalty functions. The GRAPPA difference image for uniform undersampling shows coherent aliasing, while the GRAPPA difference image for random undersampling shows noise-like artifacts. Since CS can eliminate the noise-like artifacts rather effectively for such a sparse image as the Shepp-Logan phantom,

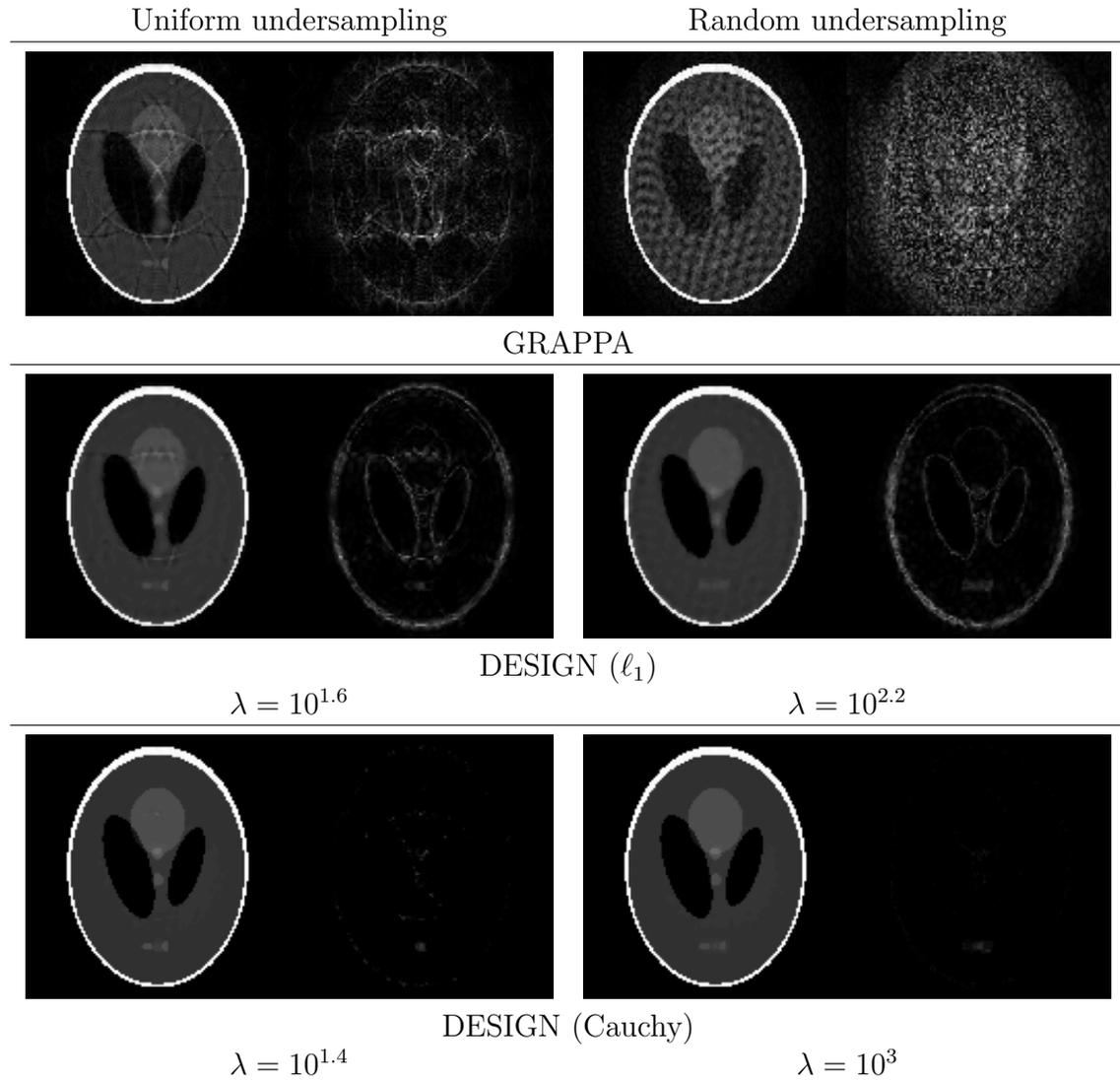


Figure 4.15: GRAPPA and DESIGN denoising methods compared for uniform and random undersampled (4×4 nominal undersampling: $R = 8.7$) Shepp-Logan phantom with finite differences sparsifying transform.

both DESIGN reconstructions can eliminate the random sampling artifacts. The ℓ_1 norm-based DESIGN denoising cannot totally eliminate the coherent aliasing from uniform undersampling, although DESIGN denoising with the Cauchy penalty function and uniform undersampling can eliminate all but the smallest of these artifacts through additional sparsification.

To determine how the choice of sampling pattern affects real data, we use the T_1 -weighted image #1 example slice from Figure 4.2. To elicit strong aliasing artifacts in the uniformly undersampled data (more significant than the amplified noise), we more aggressively crop the data, use only 10 out of 32 coil channels, and increase the undersampling factor (to $R = 13.7$). The resulting GRAPPA and DESIGN denoising reconstructions with uniform and Poisson disc undersampling are shown in Figure 4.16. While the resulting image quality is not great for any of the reconstructions shown, we call attention to the elimination of coherent aliasing artifacts in the DESIGN denoised reconstructions with Poisson disc sampling. The ℓ_1 norm-based DESIGN reconstruction with Poisson disc sampling does a reasonable job of denoising and mitigating aliasing without significantly blurring the image; most of the error appears to be on the strong outer edges, which as we know from previous experiments, get smoothed. Thus, as Poisson disc sampling is prescribed for L_1 SPIRiT reconstruction, Poisson disc sampling also may be appropriate for DESIGN denoising when GRAPPA alone is insufficient to eliminate coherent aliasing.

4.3 Discussion

Based on the simulations presented in the previous section, the DESIGN denoising method is an effective denoising method for accelerated parallel images reconstructed with GRAPPA, with several caveats. First, the DESIGN denoising method tends to smooth edges in the reconstructed image. While such smoothing does not appear to result in loss of spatial resolution or significant contrast reduction, both of which would be major concerns for clinical application of this method, smoothed edges may impact MRI applications that rely on edge detection, like brain segmentation and cor-

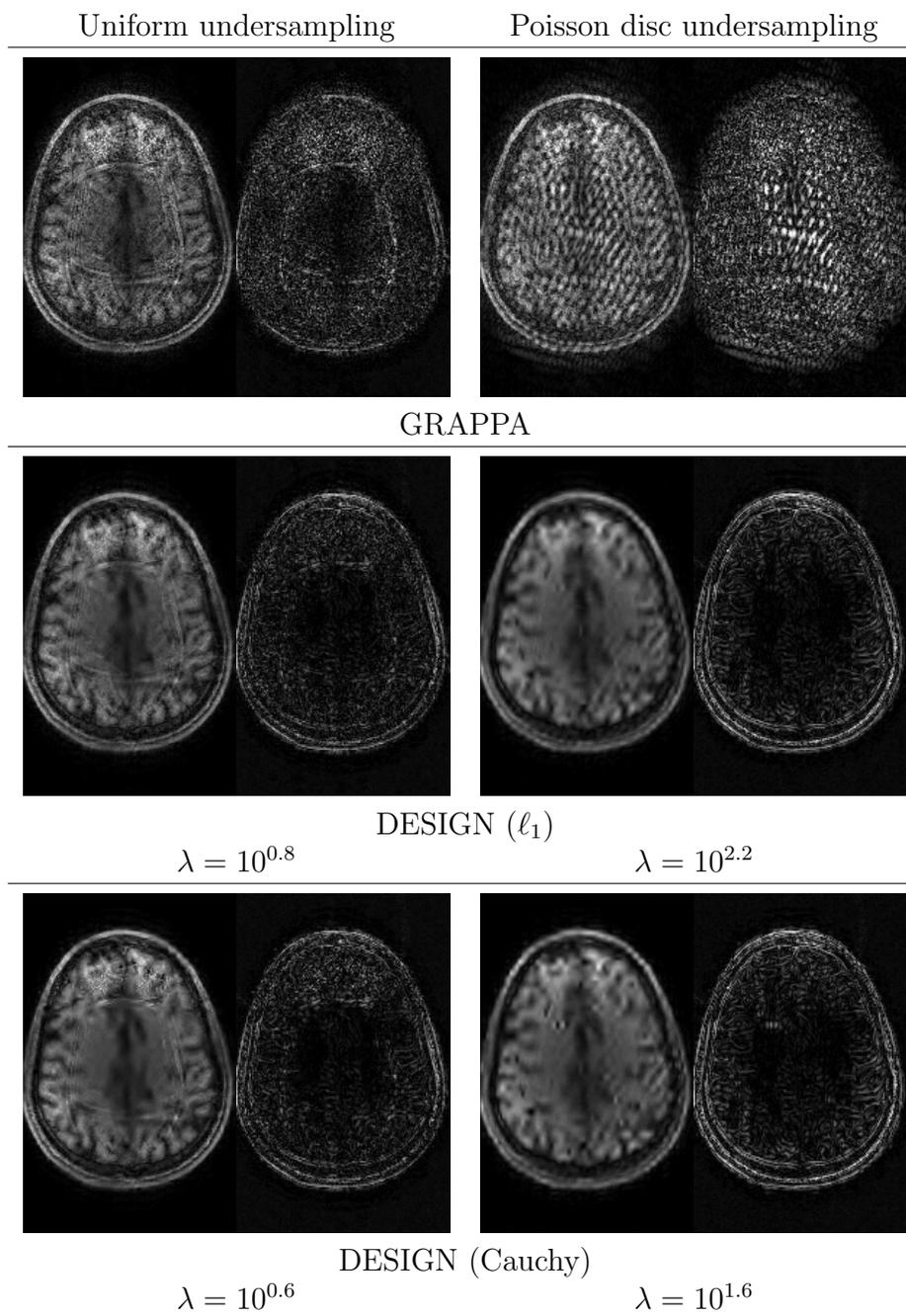


Figure 4.16: GRAPPA and DESIGN denoising methods compared for uniform and Poisson disc random undersampled (5×5 nominal undersampling: $R = 13.7$) T_1 -weighted MPRAGE dataset with four-level ‘9-7’ DWT sparsifying transform.

tical thickness measurement [32]. The experiments portrayed here all involve healthy subjects. Due to the great variety of features and abnormalities that can be encountered in human MRI, a more thorough analysis of the effects of DESIGN denoising on images with lesions, resections, and other visible abnormalities. In addition, as depicted in Figure 4.16, DESIGN denoising is relatively ineffective at mitigating coherent aliasing left behind by GRAPPA. In addition, as the undersampling factor continues to increase, the amount of sparsification performed by DESIGN denoising to mitigate noise amplification can significantly oversmooth the image, limiting the usefulness of DESIGN denoising at extremely high accelerations.

Compared to multi-channel Wiener filter-based denoising, the proposed method is more effective at mitigating noise amplification, as measured by the g-factors in Figure 4.12. Furthermore, DESIGN denoising is not plagued by the residual coherent aliasing visible in the L_1 SPIRiT results in Figures 4.9-4.11. The SNR of the GRAPPA reconstruction appears to degrade more rapidly than SPIRiT as the acceleration factor R increases, so DESIGN denoising may outperform L_1 SPIRiT at high acceleration factors if the SPIRiT reconstruction is used in place of the GRAPPA reconstruction in Equation (4.2). Another advantage of the DESIGN denoising method is that it is effective with conventional uniformly-spaced undersampling of Cartesian k-space, which is by far the most common k-space sampling approach in practice. Therefore, DESIGN denoising can be applied retroactively to denoise previously acquired GRAPPA or other accelerated parallel imaging reconstructions.

To be effective clinically, DESIGN denoising needs to be performed during the scan session. To accelerate the implementation of DESIGN denoising, we can leverage multi-threaded CPU or GPU computation to reduce the computational cost of the dominant FFT and DWT operations via parallelization. Another future extension is to combine DESIGN denoising with adaptive dictionary learning to improve image quality. Because DESIGN denoising is limited by the quality of the image to denoise, it is also useful to improve the quality of the accelerated parallel imaging reconstruction method itself. The next chapter focuses on using sparsity to improve GRAPPA kernel calibration.

Chapter 5

GRAPPA Kernel Calibration with Sparsity

The reconstructed image quality of auto-calibrating methods like GRAPPA can be impacted by the quality of the calibration. When the calibration data is noisy, or the quantity of calibration data is insufficient to yield a quality fit, the calibrated GRAPPA kernel can both amplify noise unnecessarily and cause coherent aliasing artifacts to remain in the reconstruction. Examples of GRAPPA reconstructions when the fit to the ACS lines is either of poor quality or underdetermined are shown in Figure 5.1. The DESIGN denoising method in the previous section is a post-processing approach for improving the GRAPPA reconstructed image quality, but that method, as its name suggests, is primarily useful for denoising. In this chapter, we present a novel method that aims to improve the calibration step of GRAPPA to reduce noise amplification and minimize un-resolved aliasing. Motivated by the use of sparsity as regularization, we regularize the least squares fit normally used for kernel calibration using the sparsity of the full-FOV coil images that would be reconstructed from the reduced-FOV data using the calibrated kernel. This approach would be followed by conventional GRAPPA to yield an improved reconstruction.

In this chapter, we begin with a brief discussion of regularization techniques like Tikhonov regularization, which leads to the exposition of the proposed method. Simulations comparing the proposed calibration regularization method to other forms

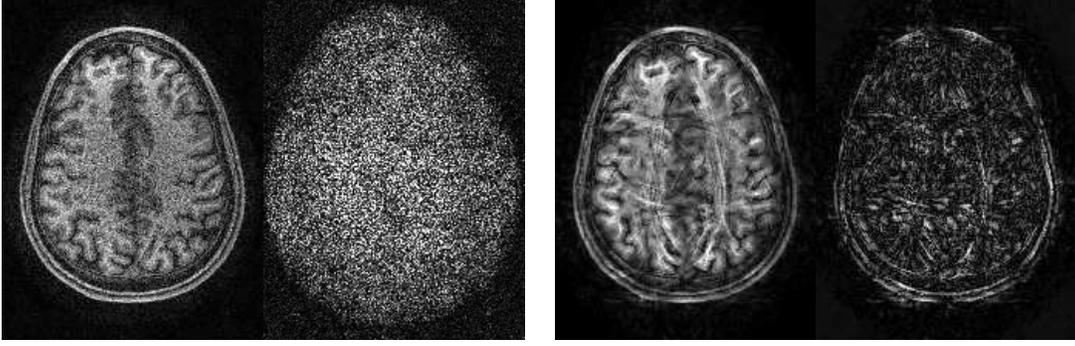


Figure 5.1: GRAPPA reconstructions with low-quality kernel calibrations demonstrating noise amplification (left) and residual aliasing (right) with 4×4 nominal undersampling. The GRAPPA reconstruction on the left was calibrated with 36×36 ACS lines and no regularization, and the reconstruction on the right was calibrated with 20×20 ACS lines (underdetermined) with Tikhonov regularization.

of GRAPPA kernel calibration are presented, and the trade-off between total acceleration R and image quality is investigated for these kernel calibration techniques. We conclude the chapter with a discussion of the impact of the calibration method and potential extensions including combination with post-processing methods like DESIGN denoising.

5.1 Theory

Consider the two-dimensional ACS fit equation used for GRAPPA kernel calibration in Equation (2.23). Collecting all these fit equations into a single linear system yields the least squares problem

$$\hat{\mathbf{G}} = \underset{\mathbf{G}}{\text{minimize}} \|\mathbf{Y}_t^{\text{ACS}} - \mathbf{Y}_s^{\text{ACS}} \mathbf{G}\|_F^2, \quad (5.1)$$

where the matrices $\mathbf{Y}_s^{\text{ACS}}$ and $\mathbf{Y}_t^{\text{ACS}}$ contain the source and target points, respectively, from the ACS lines, and \mathbf{G} contains all the GRAPPA kernel weights. Suppose the desired GRAPPA kernel measures $B_y \times B_z$ blocks of size $R_y \times R_z$, and we have $N_{\text{ACS},y} \times N_{\text{ACS},z}$ ACS lines. Then, the number of ACS fits is equal to

$$N_{\text{fit}} = (N_{\text{ACS},y} - B_y \max\{R_y - 1, 1\})(N_{\text{ACS},z} - B_z \max\{R_z - 1, 1\}), \quad (5.2)$$

and there are $B_y B_z P$ total source points and $(R_y R_z - 1)P$ total target points. Thus, $\mathbf{Y}_s^{\text{ACS}}$ is an $N_{\text{fit}} \times (B_y B_z P)$ matrix, whose rows represent the source points from the ACS lines for the fit equations, and $\mathbf{Y}_t^{\text{ACS}}$ is an $N_{\text{fit}} \times ((R_y R_z - 1)P)$ matrix with each row containing the target points from the ACS lines for each fit. The matrix \mathbf{G} has size $(B_y B_z P) \times ((R_y R_z - 1)P)$, with each column a vector of weights that are used to compute a single target point within each block. For this least squares system to have a single solution, the matrix $\mathbf{Y}_s^{\text{ACS}}$ must have full column rank, and it is necessary that $N_{\text{fit}} \geq B_y B_z P$. When the solution to Equation (5.1) is unique, it is equal to

$$\hat{\mathbf{G}} = (\mathbf{Y}_s^{\text{ACS}H} \mathbf{Y}_s^{\text{ACS}})^{-1} \mathbf{Y}_s^{\text{ACS}H} \mathbf{Y}_t^{\text{ACS}}. \quad (5.3)$$

In real acquisitions, the source and target point values from the ACS lines have noise. The noise in the target points is amplified in the kernel by the gain of the matrix $(\mathbf{Y}_s^{\text{ACS}H} \mathbf{Y}_s^{\text{ACS}})^{-1} \mathbf{Y}_s^{\text{ACS}H}$, and the noise in the source points perturbs this matrix if the noise level exceeds the minimum singular value of the matrix. The matrix $\mathbf{Y}_s^{\text{ACS}}$ has a singular value decomposition (SVD)

$$\mathbf{Y}_s^{\text{ACS}} = \mathbf{U} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{\min} \end{bmatrix}}_{\Sigma} \mathbf{V}^H, \quad (5.4)$$

where rectangular matrices \mathbf{U} and \mathbf{V} have unitary columns, and $\sigma_1 \geq \dots \geq \sigma_{\min}$ are the singular values of the matrix in decreasing order collected in diagonal matrix Σ . For rectangular $\mathbf{Y}_s^{\text{ACS}}$, this is the economic decomposition; the traditional SVD always has square unitary \mathbf{U} and \mathbf{V} and rectangular Σ padded with zeros. Then, if $\mathbf{Y}_s^{\text{ACS}}$ has full column rank, $(\mathbf{Y}_s^{\text{ACS}H} \mathbf{Y}_s^{\text{ACS}})^{-1} \mathbf{Y}_s^{\text{ACS}H}$ has SVD (with singular values in reverse order)

$$\mathbf{V} \underbrace{\begin{bmatrix} \sigma_1^{-1} & & \\ & \ddots & \\ & & \sigma_{\min}^{-1} \end{bmatrix}}_{\Sigma^+} \mathbf{U}^H. \quad (5.5)$$

One measure of gain of a matrix is the induced matrix 2-norm, defined as

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x}} \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2} = \max_{\{\mathbf{x}:\|\mathbf{x}\|_2=1\}} \|\mathbf{Ax}\|_2, \quad (5.6)$$

which is equal to the maximum singular value of the matrix. The condition number κ is equal to the ratio of maximum to minimum singular values. The condition number σ_1/σ_{\min} of $\mathbf{Y}_s^{\text{ACS}}$ is equal to the condition number of the pseudoinverse of this matrix. We examine the condition number of the matrix to determine the perturbation susceptibility of the kernel calibration to noise in the ACS lines.

When the matrix $\mathbf{Y}_s^{\text{ACS}}$ does not have full column rank, or it is poorly conditioned, we can introduce regularization into Equation (5.1) to yield a single well-behaved solution that is robust to noise in the ACS lines and numerical precision issues. Several regularization techniques exist. Tikhonov regularization [90] is commonly applied to SENSE reconstruction [54] and can also be applied to GRAPPA kernel calibration

$$\hat{\mathbf{G}} = \underset{\mathbf{G}}{\text{minimize}} \|\mathbf{Y}_t^{\text{ACS}} - \mathbf{Y}_s^{\text{ACS}}\mathbf{G}\|_F^2 + \|\alpha\mathbf{G}\|_F^2, \quad (5.7)$$

where $\alpha > 0$ is the regularization parameter. Then, the regularized solution is

$$\hat{\mathbf{G}} = (\mathbf{Y}_s^{\text{ACS}H}\mathbf{Y}_s^{\text{ACS}} + \alpha^2\mathbf{I})^{-1}\mathbf{Y}_s^{\text{ACS}H}\mathbf{Y}_t^{\text{ACS}}. \quad (5.8)$$

For $\alpha > 0$, $(\mathbf{Y}_s^{\text{ACS}H}\mathbf{Y}_s^{\text{ACS}} + \alpha^2\mathbf{I}) \succ 0$ since $\mathbf{Y}_s^{\text{ACS}H}\mathbf{Y}_s^{\text{ACS}} \succeq 0$, so the matrix inverse in Equation (5.8) is guaranteed to exist. This type of regularization attempts to find the minimum energy solution to the un-regularized problem. Tikhonov regularization can also be applied to an affine transform of the kernel $\mathbf{\Gamma}(\mathbf{G} - \mathbf{G}_0)$ to elicit a solution satisfying some other criterion, like smoothness, in which case, $\mathbf{\Gamma}$ is a finite differences transform, and $\mathbf{G}_0 = \mathbf{0}$.

An alternative approach to regularizing GRAPPA kernel calibration is to utilize the frequency-shift operator interpretation of the GRAPPA kernel [14]. For the GRAPPA kernel \mathbf{G} that performs a frequency shift by one over the full field of view, applying the kernel R_y or R_z times should yield the original data shifted in the y -

or z -direction. This insight essentially constrains the R th power of the GRAPPA kernel operator to approximate a permutation matrix, yielding a nonlinear system of equations that can be used to regularize the GRAPPA kernel calibration. This nonlinear regularization can be applied to multi-dimensional accelerated parallel imaging, although the complexity of the nonlinear system increases significantly.

The regularization approach proposed here takes advantage of the joint sparsity of the ideal reconstructed coil images in the sparse transform domain to improve the calibration of the kernel. Using the $\ell_{1,2}$ mixed norm to promote joint sparsity of the transform domain coefficients across the coils, we regularize Equation (5.1) to form

$$\hat{\mathbf{G}} = \underset{\mathbf{G}}{\text{minimize}} \|\mathbf{Y}_t^{\text{ACS}} - \mathbf{Y}_s^{\text{ACS}} \mathbf{G}\|_F^2 + \lambda \|\Psi \mathbf{F}^{-1} \text{GRAPPA}(\mathbf{G}, \mathbf{D})\|_{1,2}, \quad (5.9)$$

where $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$ is the GRAPPA reconstruction operation using kernel \mathbf{G} on acquired data \mathbf{D} (including ACS lines that are treated as data). Typically, conventional GRAPPA reconstruction is represented as a block convolution operation on the acquired data. Since convolution is commutative, we can also represent GRAPPA as a block convolution matrix of the data with the GRAPPA kernel, with an additive term for the data that gets passed through to the output (since GRAPPA does not alter the acquired k-space data). Therefore, $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$ is an affine function of \mathbf{G} , and the adjoint operator $\text{GRAPPA}^*(\mathbf{G}, \mathbf{D})$ is linear. In fact, the adjoint operation of convolution is convolution by the complex conjugate of the reversed kernel (this identity holds for one- and multi-dimensional convolution). Since convolution can be implemented efficiently using the FFT, the presence of the GRAPPA reconstruction operation in Equation (5.9) should not be too problematic from a computational point of view.

The same approaches used for solving the unconstrained optimization problem version of DESIGN denoising in Equation (4.2) can be employed here to solve Equation (5.9). We again turn to IRLS, using weights determined according to half-quadratic minimization, to solve the problem. The least-squares approximation of

Equation (5.9) is

$$\hat{\mathbf{G}}^{(i)} = \underset{\mathbf{G}}{\text{minimize}} \|\mathbf{Y}_t^{\text{ACS}} - \mathbf{Y}_s^{\text{ACS}} \mathbf{G}\|_F^2 + \frac{\lambda}{2} \|(\Delta^{(i-1)})^{1/2} \Psi \mathbf{F}^{-1} \text{GRAPPA}(\mathbf{G}, \mathbf{D})\|_F^2, \quad (5.10)$$

where the reweighting matrix $\Delta^{(i-1)}$ is a diagonal matrix with entries $\Delta_{n,n}^{(i-1)} = 1/\|[W_{n,1}, \dots, W_{n,P}]\|_2$ computed from the sparse transform of the GRAPPA reconstructed coil images $\mathbf{W} = \Psi \mathbf{F}^{-1} \text{GRAPPA}(\hat{\mathbf{G}}^{(i-1)}, \mathbf{D})$ using the previous kernel estimate $\hat{\mathbf{G}}^{(i-1)}$. To ensure that the objective function in Equation (5.10) is differentiable, we use a smoothed approximation of the ℓ_2 norm in the re-weighting matrix calculation:

$$\Delta_{n,n}^{(i-1)} = 1/\|[W_{n,1}, \dots, W_{n,P}, \varepsilon]\|_2, \quad (5.11)$$

for fixed $\varepsilon > 0$. The derivative of the objective function in Equation (5.10) is set to zero:

$$2\mathbf{Y}_s^{\text{ACS}H} (\mathbf{Y}_s^{\text{ACS}} \mathbf{G} - \mathbf{Y}_t^{\text{ACS}}) + \lambda \text{GRAPPA}^*(\mathbf{F}^{-H} \Psi^H \Delta^{(i-1)} \Psi \mathbf{F}^{-1} \text{GRAPPA}(\mathbf{G}, \mathbf{D}), \mathbf{D}) = \mathbf{0}. \quad (5.12)$$

To solve this linear system, we again employ LSMR, solving for \mathbf{g} , the vectorized version of \mathbf{G} , using $\text{LSMR}(\mathbf{A}, \mathbf{b})$ with

$$\mathbf{A} = \begin{bmatrix} \sqrt{2}(\mathbf{I}_{P \times P} \otimes \mathbf{Y}_s^{\text{ACS}}) \\ \sqrt{\lambda}(\mathbf{I}_{P \times P} \otimes ((\Delta^{(i-1)})^{1/2} \Psi \mathbf{F}^{-1} \mathbf{K}_{na}^T \mathbf{D}_{\text{conv}})) \end{bmatrix}, \quad (5.13)$$

$$\mathbf{A}^H = \begin{bmatrix} \sqrt{2}(\mathbf{I}_{P \times P} \otimes \mathbf{Y}_s^{\text{ACS}H}) & \frac{\sqrt{\lambda}}{N}(\mathbf{I}_{P \times P} \otimes (\mathbf{D}_{\text{conv}}^H \mathbf{K}_{na} \mathbf{F} \Psi^H (\Delta^{(i-1)})^{1/2})) \end{bmatrix}, \quad (5.14)$$

and

$$\mathbf{b} = \begin{bmatrix} \sqrt{2} \text{vec}(\mathbf{I}_{P \times P} \otimes \mathbf{Y}_t^{\text{ACS}}) \\ -\sqrt{\lambda} \text{vec}(\Delta^{(i-1)})^{1/2} \Psi \mathbf{F}^{-1} \mathbf{K}_a^T \mathbf{D} \end{bmatrix}. \quad (5.15)$$

In the above equations, we decompose $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$ into a block convolution

Algorithm 4 Algorithm for sparsity-promoting GRAPPA kernel calibration.

Require: $\hat{\mathbf{G}}^{(0)}$, λ , ε , I , tol

- 1: Compute initial $f^{(0)}$ by plugging into the objective in Equation (5.9).
 - 2: **for** $i = 1 : I$ **do**
 - 3: Compute $\mathbf{W} \leftarrow \Psi \mathbf{F}^{-1} \text{GRAPPA}(\hat{\mathbf{G}}^{(0)}, \mathbf{D})$ and set $\Delta_{n,n}^{(i-1)} = \frac{1}{w[n]}$ for $w[n] = \|[W_{n,1}, \dots, W_{n,P}, \varepsilon]\|_2$, for all $n = 0, \dots, N - 1$.
 - 4: Prepare matrices \mathbf{A} and \mathbf{b} with new reweighting matrix $\Delta^{(i-1)}$ and run LSMR: $\hat{\mathbf{G}}^{(i)} \leftarrow \text{LSMR}(\mathbf{A}, \mathbf{b})$.
 - 5: Compute $f^{(i)}$ by plugging in $\hat{\mathbf{G}}^{(i)}$ into the objective in Equation (5.9).
 - 6: **if** $f^{(i-1)} - f^{(i)} \leq \text{tol} \cdot f^{(i-1)}$ **then**
 - 7: **break**
 - 8: **end if**
 - 9: **end for**
 - 10: **return** Calibrated GRAPPA kernel $\hat{\mathbf{G}}^{(i)}$.
-

operation $\mathbf{D}_{\text{conv}} \mathbf{G}$ and an affine term \mathbf{D} and use $\text{GRAPPA}(\mathbf{G}, \mathbf{D}) = \mathbf{K}_{na}^T \mathbf{D}_{\text{conv}} \mathbf{G} + \mathbf{K}_a^T \mathbf{D}$. The complete sparsity-promoting GRAPPA kernel calibration algorithm is described in Algorithm 4.

5.2 Simulations and Results

The simulations in this chapter use the same T_1 -weighted real datasets acquired for testing DESIGN denoising. Example slices are extracted from each dataset, cropped and normalized, and magnitude combined images of these slices are used as ground truth for generating difference images and calculating PSNR. As before, the four-level ‘9-7’ DWT is chosen as sparsifying transform for these images.

We begin by investigating the effects of regularization in three regimes marked by the relative number of ACS fit equations to source points (across all coils) [97]. When the number of fits N_{fit} is much greater than the number of source points $B_y B_z P$, we expect regularization not to have a significant effect on image quality, as the calibration is already fairly robust to noise by virtue of the large number of ACS lines. When the number of fits is approximately equal to the number of source points, regularization could be important for mitigating the effects of noise on the calibration, and the effects on image quality can be significant. Finally, when the

number of fits is less than the number of source points, un-regularized calibration is not possible, as the kernel is not unique, and regularization is required to determine an appropriate choice of kernel that appropriately un-aliases the coil images.

We follow these experiments with a study of the effects of regularization on the trade-off between number of ACS lines/total acceleration and the reconstructed image quality. Our hypothesis is that quality of the calibration is a major driver of reconstructed image quality and effective regularization can enable greater overall acceleration while maintaining image quality.

5.2.1 Regularized Calibration Performance Comparisons

For $R_y = R_z = 4$ two-dimensional k-space undersampling with 36×36 ACS lines, the kernel size is first chosen to be $B_y = B_z = 3$, as in the DESIGN denoising experiments in Chapter 4. The GRAPPA kernel to be calibrated weights $B_y B_z P = 288$ source points to generate $(R_y R_z - 1)P = 480$ target points, and from Equation (5.2), the number of fit equations for each target point is 784. The matrix of source points from the ACS lines $\mathbf{Y}_s^{\text{ACS}}$ has size 784×288 , and it has full column rank, meaning it has a pseudoinverse. GRAPPA reconstructions using kernels calibrated using no regularization, Tikhonov regularization, and sparsity-promoting regularization are shown in Figure 5.2. In all these experiments, the best choices (found via parameter sweep) of α and λ are used for Tikhonov and sparsity-promoting regularization, respectively.

The quality of the GRAPPA-reconstructed image in Figure 5.2 with un-regularized calibration is relatively high, plagued mainly by some noise amplification, with no visible aliasing in either the magnitude or difference images. Both regularized kernels yield GRAPPA reconstructions with somewhat improved noise characteristics, with little observable difference between the two.

Using such a small kernel may be introducing some unexpected dependencies in the interpolated k-space. Since the number of target points exceeds the number of source points, some target points are going to be linear combinations of the other target points, and this redundancy would be expected to reduce the amount of frequency information in the reconstructed coil images. Increasing the size of the kernel so that

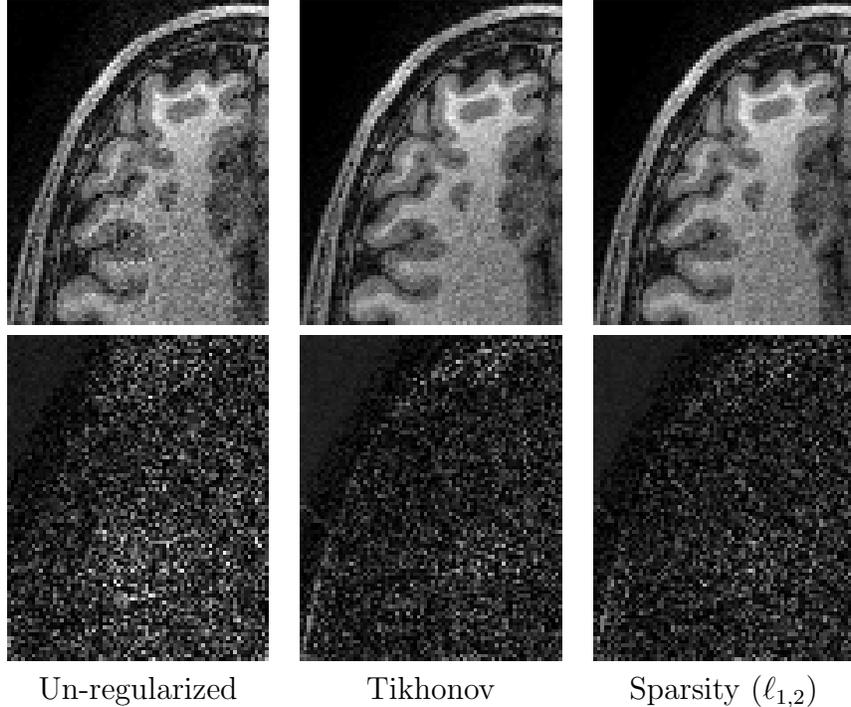


Figure 5.2: GRAPPA reconstructions of T_1 -weighted image #1 (4×4 nominal undersampling: $R = 10.5$) with high-quality kernel calibrations with no regularization, Tikhonov regularization, and sparsity-promoting regularization.

$B_y B_z$ is at least $R_y R_z - 1$ will decrease the likelihood that this redundancy exists, but it requires more ACS fits since there are more source points, and it reduces the number of fits available from a fixed set of ACS lines. When we increase B_y and B_z to 4, the number of source points increases to 512, and the number of fits from the 36×36 ACS block decreases to 576. In this regime, the number of fits is comparable to the number of source points, so we would expect the calibration to be more susceptible to noise.

The reconstructed images in Figure 5.3 demonstrate the effect of calibration on image quality. The GRAPPA reconstructions with kernels calibrated without regularization have substantially amplified noise, and either regularization technique is effective at reducing the noise level substantially (PSNR improves by between 6 and 9 dB). In addition, there is no perceptible degradation in image quality due to blurring, smoothing, or edge definition typically associated with denoising.

At high accelerations, the ACS lines can take almost as much time as the rest of

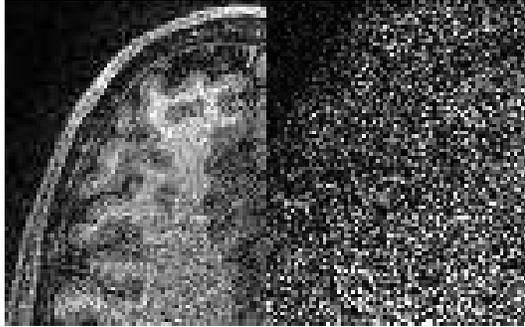


Image #1: Un-regularized

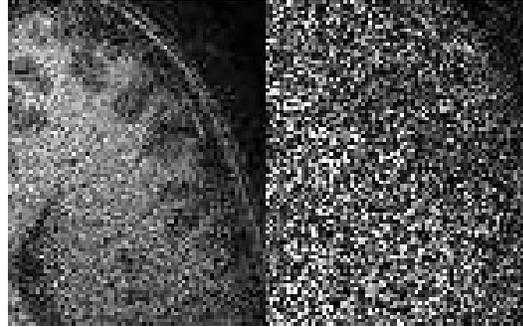


Image #2: Un-regularized

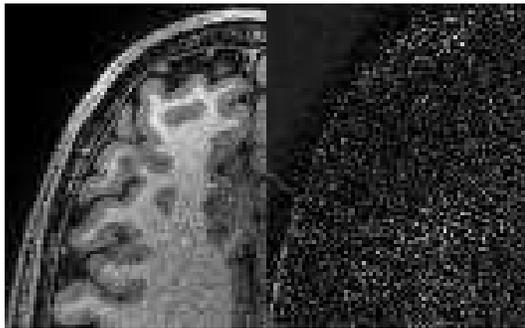


Image #1: Tikhonov

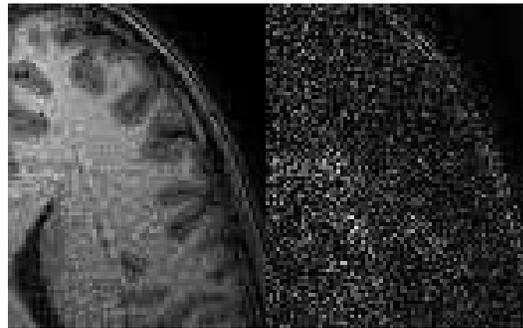


Image #2: Tikhonov

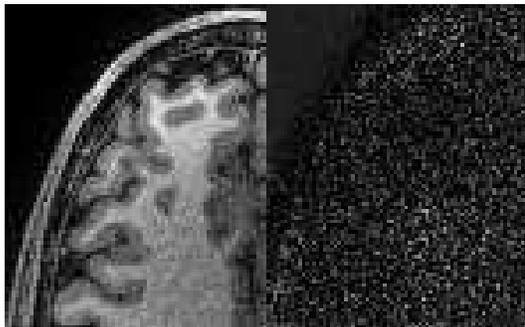


Image #1: Sparsity ($\ell_{1,2}$)



Image #2: Sparsity ($\ell_{1,2}$)

Figure 5.3: GRAPPA reconstructions of T_1 -weighted images #1 and #2 (4×4 nominal undersampling: both $R = 10.5$) with low-quality kernel calibrations with no regularization, Tikhonov regularization, and sparsity-promoting regularization.

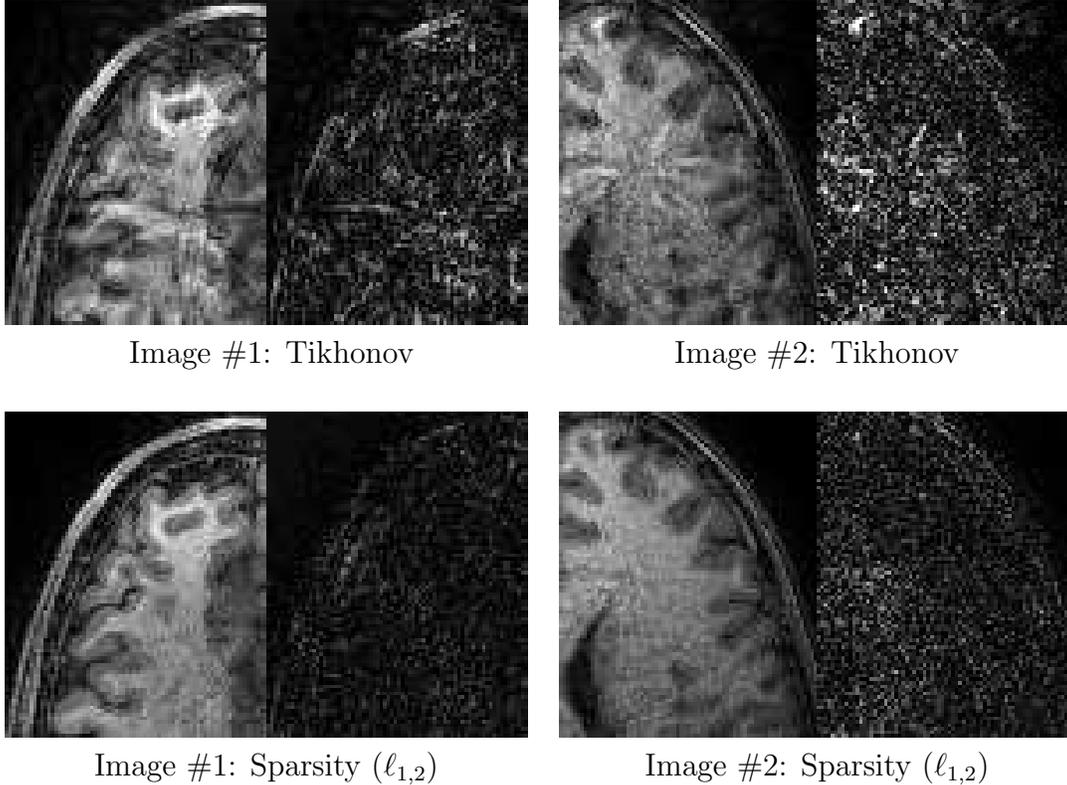


Figure 5.4: GRAPPA reconstructions of 4×4 nominally undersampled T_1 -weighted image #1 ($R = 13.7$) and image #2 ($R = 12.9$) with underdetermined kernel calibrations with Tikhonov regularization and sparsity-promoting regularization.

the data to acquire; for $R_y = R_z = 4$, a 36×36 set of ACS lines takes half as much time as the rest of the data. Reducing the number of ACS lines significantly would cause the number of fits N_{fit} to fall below the number of source points, transforming the least squares problem in Equation (5.1) into an underdetermined problem. To solve this problem, regularization is necessary to identify a single solution; in the simulations below, we compare Tikhonov regularization to the proposed sparsity-promoting regularized calibration.

The regularization techniques are compared for underdetermined GRAPPA kernel calibration for T_1 -weighted datasets #1 and 2 in Figure 5.4. In both cases, the Tikhonov regularized kernels do not successfully undo aliasing in the coil images, yielding magnitude images that are practically unusable. The images reconstructed using GRAPPA kernels calibrated with sparsity-promoting regularization show far less aliasing. While not quite the same quality as the regularized reconstructions

in the high-quality calibration experiments, the image quality is comparable to un-regularized reconstructions with far more ACS lines. This observation suggests that such regularized calibration can enable high quality reconstructions from far fewer ACS lines than with un-regularized calibration, yielding higher total accelerations.

5.2.2 Trade-Off between Acceleration and Image Quality

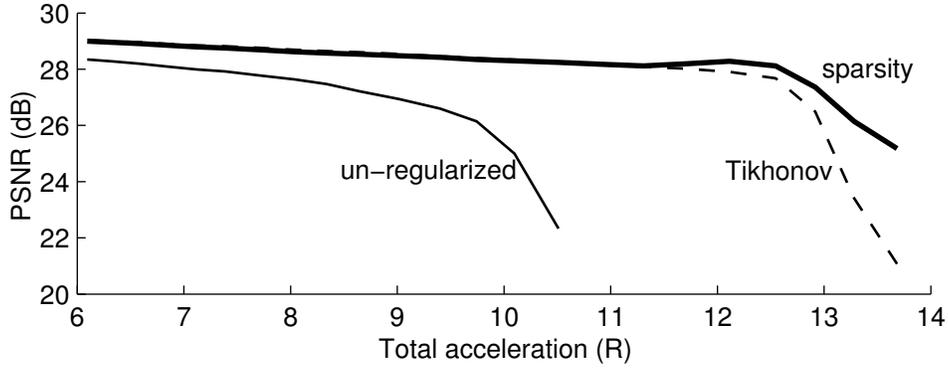
As we increase the number of ACS lines, we trade total acceleration R for improved image quality. For two-dimensional undersampling by factors of $R_y \times R_z$ of $N_y \times N_z$ k-space, the effective acceleration including $N_{\text{ACS},y} \times N_{\text{ACS},z}$ ACS lines is approximately

$$R = \frac{R_y R_z}{1 + (R_y R_z - 1)^{\frac{N_{\text{ACS},y} N_{\text{ACS},z}}{N_y N_z}}}. \quad (5.16)$$

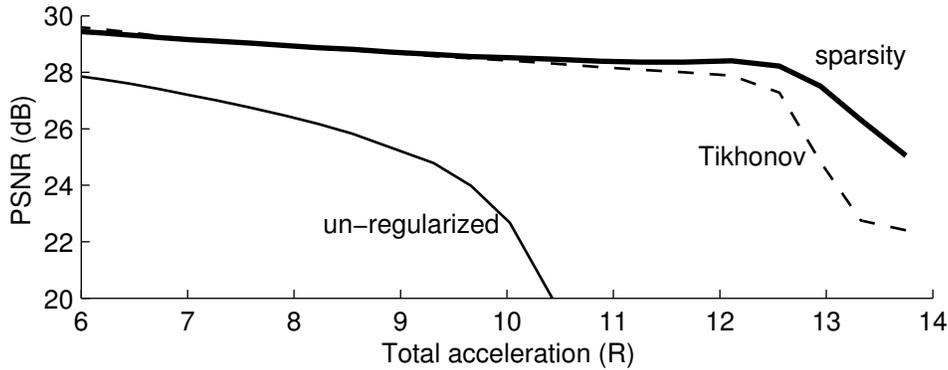
If we need to ensure the number of fit equations per target point N_{fit} is at least $B_y B_z P$, and set $B_y = R_y$, and $B_z = R_z$, then $N_{\text{fit}} \geq R_y R_z P$. For a square kernel and square ACS region, $N_{\text{ACS},y} = N_{\text{ACS},z} \geq R_y (\sqrt{P} + R_y - 1)$, which means that the number of ACS lines is quadratic in the undersampling factor, and for significant undersampling, the increase in number of ACS lines can diminish the benefit of or even overcome the acceleration due to undersampling the rest of k-space. Thus, it behooves users of GRAPPA to be able to decrease the number of ACS lines needed to obtain a quality image.

In this experiment, we examine the effect on image quality as approximated by PSNR as we vary the number of ACS lines, and hence, the total acceleration R . We repeat the experiment for GRAPPA with both un-regularized calibration, and with the Tikhonov- and sparsity-regularized calibration techniques described earlier. The trade-off curves are plotted for T_1 -weighted images #1 and #2 in Figure 5.5.

For both images, the trade-off curves for un-regularized GRAPPA kernel calibration portray a rapid drop-off in PSNR as total acceleration increases. In contrast, all the regularization curves fall off much more slowly until total accelerations of about $R = 12$. At higher accelerations, the Tikhonov-regularized calibration yields reconstructed images with rapidly diminishing PSNR, while the drop-off in the recon-



(a) T_1 -weighted Image #1.



(b) T_1 -weighted Image #2.

Figure 5.5: Trade-off curves for un-regularized, Tikhonov-regularized, and sparsity-promoting GRAPPA kernel calibration depicting the relationship between reconstructed image PSNR and total acceleration R as the number of ACS lines is varied. Nominal undersampling is held fixed at 4×4 .

constructed image quality with sparsity-promoting calibration does not decrease nearly as quickly. The significantly shifted trade-off curves suggest that much greater total acceleration is possible by using regularized GRAPPA kernel calibration, especially the sparsity-promoting calibration proposed here.

Since total acceleration depends on both the nominal undersampling and number of ACS lines, we repeat this experiment for smaller and larger nominal undersampling factors to gain a more complete picture of the optimal trade-off between PSNR and total acceleration. To optimize PSNR for a given total acceleration, we construct the upper enveloping curve of these curves for different nominal undersampling factors. The optimal trade-off curves for un-regularized, Tikhonov-regularized, and sparsity-promoting GRAPPA kernel calibration are depicted in Figure 5.6 using 4×3 , 4×4 ,

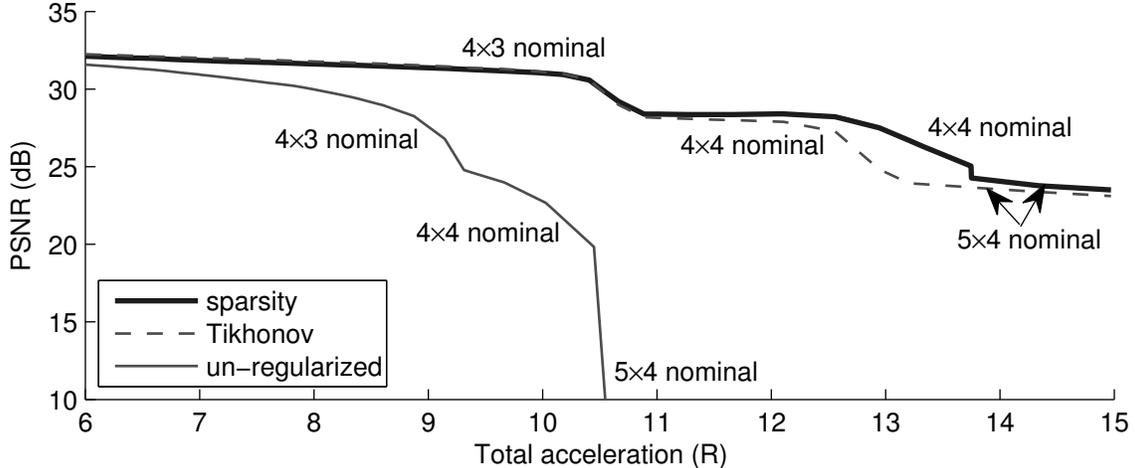


Figure 5.6: Trade-off curves for un-regularized, Tikhonov-regularized, and sparsity-promoting GRAPPA kernel calibration depicting the relationship between reconstructed image PSNR and total acceleration R as the total acceleration is varied (by varying both nominal undersampling and the number of ACS lines).

and 5×4 nominal 2-D undersampling. As we increase the desired total acceleration, the optimal choice of nominal undersampling increases. Also, the most significant improvement in PSNR due to sparsity-promoting calibration over Tikhonov-regularized calibration is evident at the 4×4 nominal undersampling level. At higher levels of undersampling ($R \geq 14$), visual inspection of the reconstructed images suggest that the total acceleration may be beyond the range where any one of these methods can produce a high quality image for this dataset.

5.2.3 Post-Processing with DESIGN Denoising

The sparsity-promoting GRAPPA kernel calibration technique yields a GRAPPA kernel that produces images via GRAPPA reconstruction, and these reconstructed images can be post-processed by DESIGN denoising like conventional GRAPPA accelerated parallel imaging. Such post-processing would be expected to mitigate residual noise amplification and yield a smoother image than with sparsity-promoting calibration alone. Examples of reconstructions using sparsity-promoting calibration and denoising are carried out for the second T_1 -weighted dataset with a 4×4 block GRAPPA kernel and 24 (underdetermined) and 36 (low-quality) ACS lines.

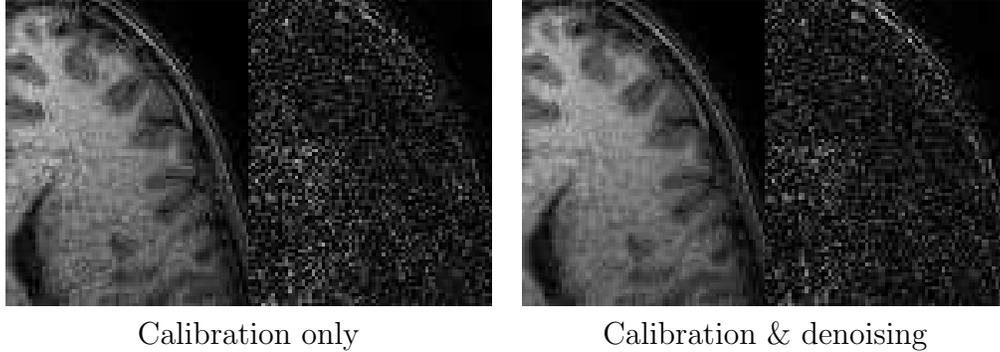


Figure 5.7: GRAPPA and DESIGN-denoised reconstructions of T_1 -weighted image #2 (4×4 nominal undersampling: $R = 12.9$) with underdetermined sparsity-promoting kernel calibration.

Figure 5.7 depicts the effects of sparsity-promoting calibration and denoising in the underdetermined GRAPPA kernel calibration regime. The denoising method effectively mitigates residual noise visible in the image, at the cost of losing some edge definition and gray-white matter contrast. The images in Figure 5.8 tell a similar story for the low-quality ACS fit calibration regime. The sparsity-promoting calibration (middle) greatly reduces noise amplification found in the conventional GRAPPA reconstruction (left), and the DESIGN denoising result (right) has the residual noise removed. As observed in experiments in Chapter 4, the denoised image loses some edge-definition, but small features remain discernible. Clearly, in combination, the sparsity-promoting calibration has the greater effect on image quality, but the post-processing using DESIGN can yield visible improvements in the noise level.

5.3 Discussion

The proposed sparsity-promoting calibration method improves the image quality of GRAPPA reconstructions by reducing the effect of noise in the ACS data on the kernel weights. The images reconstructed using this novel regularization method exhibit both reduced noise amplification and better un-aliasing than GRAPPA with either un-regularized or Tikhonov-regularized calibration. The impact of regularization appears significant at high accelerations, where a larger GRAPPA kernel is used, and where acquiring a lot of ACS lines is undesirable. In fact, regularized calibration can

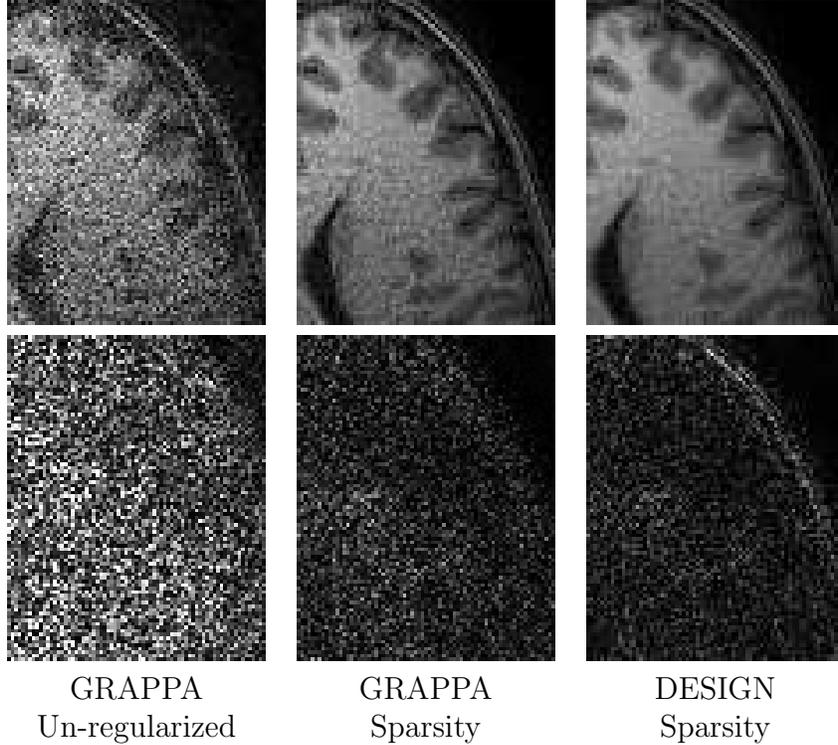


Figure 5.8: GRAPPA with un-regularized and sparsity-promoting calibration and DESIGN-denoised GRAPPA with sparsity-promoting calibration of T_1 -weighted image #2 (4×4 nominal undersampling: $R = 10.5$) with 36×36 ACS lines.

yield GRAPPA reconstructions from fewer ACS lines with the same image quality as un-regularized calibration with more ACS lines. The shifted trade-off curve of image PSNR versus total acceleration demonstrates the value of the proposed method for highly accelerated parallel imaging. The sparsity-promoting regularization even appears to mitigate aliasing when the number of ACS lines normally produces an underdetermined ACS fitting problem.

The sparsity-promoting GRAPPA kernel calibration method can be combined with DESIGN denoising to further mitigate noise amplification. This combined approach also elucidates a way to extend the calibration method to non-uniformly under-sampled Cartesian k-space. Direct or iterative computation of the GRAPPA reconstruction $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$ for non-uniformly subsampled k-space is far slower than for uniformly-spaced reduced-FOV k-space. We can alternatively replace \mathbf{D} with an estimate of uniformly-spaced k-space formed using a combination of the GRAPPA reconstruction and sparsity, similar to L_1 SPIRiT with the GRAPPA interpolation

kernel in place of the SPIRiT consistency kernel, and iterate updating the GRAPPA kernel and estimating the uniformly-spaced data.

Sparsity-promoting calibration significantly improves the quality of GRAPPA-reconstructed images when the ACS lines are few in number. However, the image quality is still degraded at very high accelerations, and we suspect that further improvement may be possible. One approach to be investigated is the joint estimation of the GRAPPA kernel and the full k-space, using an estimation-theoretic framework with a sparsity-promoting prior on the full k-space and a likelihood model on the observations. This approach will be the focus of the next chapter.

Chapter 6

Estimation Using GRAPPA and Sparsity

The image reconstruction problem can be framed as an estimation problem, where the full k-space for all the coil images is estimated from a set of observations. Naturally, these observations are corrupted by noise, and since the MRI signal equation suggests that the observations are orthogonal weighted summations of a large number of spin vectors, the central limit theorem suggests that the observation noise can be approximated by iid Gaussian noise vectors. Using the complex representation of the transverse magnetization, we have an observation model with complex Gaussian noise.

From just this observation model, likelihood estimates of the acquired k-space can be formed, but the missing k-space cannot be estimated due to the assumption of statistical independence across frequencies of the observations. However, parallel imaging and transform sparsity can be used to link the missing data to these observations. In this chapter, we utilize a combination of GRAPPA and sparsity to form estimates of the full k-space for coil images. By imposing a sparsity prior, we are imposing a Bayesian estimation framework on the reconstruction problem, which enables optimal estimation based on the posterior distribution of the full k-space given the data, as is done for sparsity in Chapter 3. We derive a full k-space observation model using the GRAPPA reconstructed k-space and tie the observation model to-

gether with the sparsity prior to form a MAP estimation problem. Motivated by the efforts in Chapter 5 to improve calibration of the GRAPPA kernel weights, we expand the estimation problem to jointly estimate the full k-space and these weights and pose an optimization problem that can be solved for situations where the conventionally calibrated kernel may be inaccurate. Simulations depicting the performance of the proposed Bayesian combination of GRAPPA and sparsity are followed by performance of the joint estimation method for GRAPPA calibrated with a reduced number of ACS lines.

6.1 Theory

First, we assume we have calibrated a GRAPPA kernel \mathbf{G} using either conventional kernel calibration using just the ACS lines or any of the regularized calibration methods discussed in Chapter 5. Using this calibrated GRAPPA kernel and the acquired data \mathbf{D} , we form a GRAPPA reconstruction $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$, which includes “observations” of the un-acquired k-space. To estimate the true values of the full k-space \mathbf{Y} from these observations, we formulate an observation model for both the acquired and GRAPPA-interpolated data, paying careful attention to the noise amplification due to the GRAPPA kernel. We then add a joint sparsity-promoting signal model for \mathbf{Y} and discuss the MMSE and MAP estimators for this Bayesian inference problem. While the MMSE estimator is not easy to compute, the MAP estimator has a similar form to the optimization problems for DESIGN denoising or L_1 SPIRiT and can be computed using the same familiar tools.

Allowing the kernel \mathbf{G} to vary, we extend our approach to estimate jointly the most likely kernel and the full k-space. We formulate an observation model for the target points $\mathbf{Y}_t^{\text{ACS}}$ of the ACS lines’ fit equations; this observation model is a function of the GRAPPA kernel, so finding the most likely choice of the target k-space and kernel corresponds to maximizing simultaneously the likelihood of \mathbf{G} and the posterior density of $\mathbf{Y}_t^{\text{ACS}}$, based on our observations of the source points $\mathbf{D}_s^{\text{ACS}}$ of the ACS lines’ fit equations. Similarly, the GRAPPA-interpolated observations of the un-acquired

k-space are also functions of the kernel \mathbf{G} , so finding the most likely choice of the un-acquired k-space and the kernel is equivalent to maximizing jointly the likelihood of \mathbf{G} and the posterior density of $\mathbf{K}_{na}\mathbf{Y}$. The resulting joint optimization problem combining all these models is similar to the optimization problem encountered earlier. However, since the noise covariances for the target points of the ACS lines' fit equations and the un-acquired k-space depend on the unknown kernel, a modification to the IRLS method is necessary. We propose holding the covariance matrices fixed while solving the re-weighted least-squares problems and updating the covariance matrices along with the diagonal re-weighting matrix between IRLS iterations. The effectiveness of both estimation techniques is studied later in this chapter.

6.1.1 Estimating the Full K-Space

Consider the full multi-channel coil k-space \mathbf{Y} and a pre-calibrated GRAPPA kernel \mathbf{G} . We subsample this k-space using the matrix \mathbf{K}_a ; denote by $\mathbf{K}_a\mathbf{Y}$ the subsampled multi-channel k-space and by \mathbf{D} the acquired samples. Each row of \mathbf{D} is a vector of the multi-channel coil samples of a single k-space point \mathbf{k} . Our observation model for the acquired data is simple: the true k-space values in $\mathbf{K}_a\mathbf{Y}$ are corrupted by additive zero-mean complex Gaussian noise \mathbf{N}_a . Each row of \mathbf{N}_a has covariance $\mathbf{\Lambda}$, with uncorrelated real and imaginary parts, and each row (k-space frequency) is independent of every other row. Therefore, we write

$$\mathbf{D} = \mathbf{K}_a\mathbf{Y} + \mathbf{N}_a, \quad (6.1)$$

where $\text{vec}(\mathbf{N}) \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Lambda} \otimes \mathbf{I}_{N \times N})$, a complex normal distribution with zero mean, uncorrelated real and imaginary parts, and covariance $\mathbf{\Lambda} \otimes \mathbf{I}_{N \times N}$. When we estimate the full k-space, our model treats the acquired data as noisy, so instead of preserving the acquired data as is done in Chapters 4 and 5, the acquired data is denoised as well. However, if preserving the acquired k-space data is desired, the methods presented in this chapter can be adjusted easily using the nullspace formulation of the problem, as is done with DESIGN denoising.

The GRAPPA reconstruction $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$ includes estimates \mathbf{X} for the missing k-space $\mathbf{K}_{na}\mathbf{Y}$, which we treat as observations by making the assumption that the GRAPPA reconstruction operation would exactly reproduce the true values of the missing k-space $\mathbf{K}_{na}\mathbf{Y}$ given the GRAPPA kernel \mathbf{G} and the true values of the acquired k-space $\mathbf{K}_a\mathbf{Y}$ as inputs. Taking advantage of the bilinearity of the GRAPPA reconstruction yields

$$\begin{aligned}\mathbf{X} &= \mathbf{K}_{na} \text{GRAPPA}(\mathbf{G}, \mathbf{D}) \\ &= \mathbf{K}_{na} \text{GRAPPA}(\mathbf{G}, \mathbf{K}_a\mathbf{Y}) + \mathbf{K}_{na} \text{GRAPPA}(\mathbf{G}, \mathbf{N}_a) = \mathbf{K}_{na}\mathbf{Y} + \mathbf{N}_{na},\end{aligned}\quad (6.2)$$

where $\mathbf{N}_{na} \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Lambda}_{\text{GRAPPA}})$ is complex Gaussian noise amplified by convolution with the GRAPPA kernel. The GRAPPA reconstruction operator introduces correlation across all the points in a k-space block as well as between the target points in a block and the source points and target points in adjacent blocks covered by overlapping kernels. From the 2-D GRAPPA reconstruction description in Equation (2.22), the correlation between a target point's noise $n_{p1}[k_y + r_{y1}, k_z + r_{z1}]$ in channel $p1$ and a source point's noise $n_{p2}[k_y + b_y R_y, k_z + b_z R_z]$ in channel $p2$ used by GRAPPA to interpolate that target point is

$$\begin{aligned}\mathbb{E}\{n_{p1}[k_y + r_{y1}, k_z + r_{z1}]n_{p2}^*[k_y + b_y R_y, k_z + b_z R_z]\} \\ = \sum_{q1=1}^P g_{p1, q1, r_{y1}, r_{z1}} [b_y + \lceil B_y/2 \rceil - 1, b_z + \lceil B_z/2 \rceil - 1] \Lambda_{q1, p2}.\end{aligned}\quad (6.3)$$

Similarly, the correlation between the noise in two target points in the same block is

$$\begin{aligned}\mathbb{E}\{n_{p1}[k_y + r_{y1}, k_z + r_{z1}]n_{p2}^*[k_y + r_{y2}, k_z + r_{z2}]\} \\ = \sum_{q1, q2=1}^P \Lambda_{q1, q2} \sum_{b_y=0}^{B_y-1} \sum_{b_z=0}^{B_z-1} g_{p1, q1, r_{y1}, r_{z1}} [b_y, b_z] g_{p2, q2, r_{y2}, r_{z2}}^*[b_y, b_z].\end{aligned}\quad (6.4)$$

Finally, the correlation between noise in two target points in blocks that are reconstructed from some of the same source points is (assume $0 \leq b_{y2} < B_y$ and

$0 \leq b_{z2} < B_z$; otherwise, reverse summations):

$$\begin{aligned} & \mathbb{E}\{n_{p1}[k_y + r_{y1}, k_z + r_{z1}]n_{p2}^*[k_y + b_{y2}R_y + r_{y2}, k_z + b_{z2}R_z + r_{z2}]\} \\ &= \sum_{q1, q2=1}^P \Lambda_{q1, q2} \sum_{b_{y1}=b_{y2}}^{B_y-1} \sum_{b_{z1}=b_{z2}}^{B_z-1} g_{p1, q1, r_{y1}, r_{z1}}[b_{y1}, b_{z1}]g_{p2, q2, r_{y2}, r_{z2}}^*[b_{y1} - b_{y2}, b_{z1} - b_{z2}]. \quad (6.5) \end{aligned}$$

Each source point is correlated with up to $B_y B_z (R_y R_z - 1)P$ target points, and each target point is correlated with $B_y B_z (R_y R_z - 1)P - 1$ other target points and $B_y B_z P$ source points, so the complete GRAPPA noise covariance matrix has approximately $B_y B_z R_y R_z NP^2$ nonzero entries; this matrix can become prohibitively large for realistic choices of N , P , acceleration factor, and kernel size, so when implementing an algorithm, we need to approximate the matrix by a simpler matrix that is preferably easy to invert. We choose to ignore correlations between k-space frequencies, reducing to NP^2 the number of nonzero entries in the covariance matrix Λ_{GRAPPA} . Alternatively, one could ignore only the correlations across blocks of k-space frequencies, which would reduce the number of nonzero entries to $R_y R_z NP^2$.

If cross terms are retained in the GRAPPA noise covariance matrix Λ_{GRAPPA} , and the number of GRAPPA kernel source points $B_y B_z$ is less than the number of target points $R_y R_z$, the noise covariance matrix will be rank deficient. Thus, when the acceleration factor is high enough so that $R_y R_z > B_y B_z$, the approximation of Λ_{GRAPPA} ignoring correlations across k-space frequencies should be used to avoid attempting to invert a singular matrix.

From the observation models for the acquired and GRAPPA-interpolated data, we can form maximum likelihood estimate of the full k-space \mathbf{Y} , which is simply $\hat{\mathbf{Y}} = \text{GRAPPA}(\mathbf{G}, \mathbf{D})$. However, this estimate is not very satisfying, as it is simply the GRAPPA reconstruction. To denoise the GRAPPA reconstruction, we turn to the joint sparsity-promoting signal model for the full multi-channel k-space. Sparsity-promoting priors take the form of

$$p(\mathbf{Y}) \propto \exp\{-\lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{s,q}\}, \quad (6.6)$$

where $\|\cdot\|_{S,q}$ is defined in Equation (3.12). Priors resulting from a Cauchy or ℓ_0 penalty function can be improper (see Table 4.1). For this distribution to be well-defined, it is sufficient for the mapping from \mathbf{Y} to $\mathbf{W} = \boldsymbol{\Psi}\mathbf{F}^{-1}\mathbf{Y}$ to be a bijection and the prior on \mathbf{W} to be proper (normalizable/integrable). When the sparsity measure is separable across transform coefficients, the above distribution becomes proportional to

$$p(\mathbf{Y}) \propto \exp \left\{ -\lambda \sum_{n=0}^{N-1} s(\|\mathbf{w}[n]\|_q) \right\}, \quad (6.7)$$

where $\mathbf{w}[n] = [W[n, 1], \dots, W[n, P]]$ for $\mathbf{W} = \boldsymbol{\Psi}\mathbf{F}^{-1}\mathbf{Y}$. In this chapter, we use the $\ell_{1,2}$ mixed norm as a joint sparsity measure, so the prior becomes

$$p(\mathbf{Y}) \propto \prod_{n=0}^{N-1} \exp\{-\lambda\|\mathbf{w}[n]\|_2\}. \quad (6.8)$$

This prior distribution is exactly the complex multiple measurement vector prior listed in Equation (3.21). The tuning parameter λ can either be manually tuned during estimation or treated as a parameter with hyperprior distribution $p(\lambda)$, such as a Gamma distribution. In this chapter, we choose λ using the same coarse-then-fine parameter sweeps used in previous chapters. However, as λ is connected only to the prior distribution, we would expect the value of λ to vary only with the sparsity of the dataset, not with the acceleration factor or noise amplification as in previous chapters.

We now have a Bayesian estimation problem with an observation model based on GRAPPA interpolation and a signal model based on promoting sparsity. The MMSE-optimal estimator is the posterior mean $\mathbb{E}\{\mathbf{Y} \mid \mathbf{D}, \mathbf{X}\}$. This estimator can be computed using Bayes' rule:

$$\mathbb{E}\{\mathbf{Y} \mid \mathbf{D}, \mathbf{X}\} = \frac{\int_{\mathbb{C}^{N \times P}} \mathbf{Y} p(\mathbf{D} \mid \mathbf{K}_a \mathbf{Y}) p(\mathbf{X} \mid \mathbf{K}_{na} \mathbf{Y}) p(\mathbf{Y}) d\mathbf{Y}}{\int_{\mathbb{C}^{N \times P}} p(\mathbf{D} \mid \mathbf{K}_a \mathbf{Y}) p(\mathbf{X} \mid \mathbf{K}_{na} \mathbf{Y}) p(\mathbf{Y}) d\mathbf{Y}}. \quad (6.9)$$

The numerator computes the expectation of \mathbf{Y} with respect to the joint distribution, and the denominator computes the probability density function (pdf) of the observations \mathbf{D}, \mathbf{X} by marginalizing by the full k-space \mathbf{Y} . Neither integral has a closed

form, so it may be tempting to employ numeric integration techniques like quadrature [48]. Unfortunately, the variables in \mathbf{Y} are tightly coupled for both integrals by the sparse transform and correlations, and quadrature is not helpful due to the curse of dimensionality (i.e. the computational complexity is exponential in the number of variables).

Alternatively, we compute the MAP estimator, which is equivalent to

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{maximize}} p(\mathbf{D} | \mathbf{K}_a \mathbf{Y}) p(\mathbf{X} | \mathbf{K}_{na} \mathbf{Y}) p(\mathbf{Y}). \quad (6.10)$$

This optimization problem is very similar to DESIGN denoising or L_1 SPIRiT:

$$\begin{aligned} \hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{minimize}} & \frac{1}{2} \|\text{vec}(\mathbf{D} - \mathbf{K}_a \mathbf{Y})\|_{\mathbf{\Lambda} \otimes \mathbf{I}_{M \times M}}^2 \\ & + \frac{1}{2} \|\text{vec}(\mathbf{X} - \mathbf{K}_{na} \mathbf{Y})\|_{\mathbf{\Lambda}_{\text{GRAPPA}}}^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{1,2}. \end{aligned} \quad (6.11)$$

The notation $\|\mathbf{y}\|_{\mathbf{\Lambda}}^2$ is short for $\|\mathbf{\Lambda}^{-1/2} \mathbf{y}\|_2^2$, a weighted least-squares term. This optimization problem can be solved using IRLS or any of the other tools mentioned for solving the unconstrained optimization formulation of DESIGN denoising. When preserving the acquired data $\mathbf{K}_a \mathbf{Y}$, the first term of the objective in Equation (6.11) vanishes, and the optimization problem becomes constrained:

$$\begin{aligned} \hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{minimize}} & \frac{1}{2} \|\text{vec}(\mathbf{X} - \mathbf{K}_{na} \mathbf{Y})\|_{\mathbf{\Lambda}_{\text{GRAPPA}}}^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{1,2} \\ \text{s.t. } & \mathbf{D} = \mathbf{K}_a \mathbf{Y}. \end{aligned} \quad (6.12)$$

The nullspace method used for DESIGN denoising can be used to optimize over the missing data $\mathbf{K}_{na} \mathbf{Y}$, an unconstrained problem. In this chapter, we focus mainly on denoising the full k-space.

6.1.2 Joint Estimation of the Kernel and Full K-Space

So far, we have treated the GRAPPA kernel as a known parameter in the observation model for the un-acquired k-space data. In this estimation-theoretic framework, we

can estimate the GRAPPA kernel jointly with the full k-space. Conventional calibration uses the least squares problem in Equation (5.1). Since the ACS lines are noisy, we can transform the calibration equation into an observation model by treating the GRAPPA ACS target matrix as random variables $\mathbf{Y}_t^{\text{ACS}}$ and replacing the unknown true GRAPPA ACS source matrix $\mathbf{Y}_s^{\text{ACS}}$ with the observed GRAPPA source matrix $\mathbf{D}_s^{\text{ACS}}$:

$$\mathbf{D}_s^{\text{ACS}}\mathbf{G} = \mathbf{Y}_t^{\text{ACS}} + \mathbf{N}. \quad (6.13)$$

The complex Gaussian noise \mathbf{N} is modeled as the appropriate subset of the GRAPPA noise \mathbf{N} from Equation (6.2).

In the observation models for the ACS fits and the un-acquired k-space, the GRAPPA kernel appears as an unknown parameter to be estimated. The MAP estimation problem in Equation (6.10) becomes a joint estimation of both the GRAPPA kernel \mathbf{G} and the full k-space \mathbf{Y} , which includes the ACS fit target matrix $\mathbf{Y}_t^{\text{ACS}}$:

$$\{\hat{\mathbf{G}}, \hat{\mathbf{Y}}\} = \underset{\mathbf{G}, \mathbf{Y}}{\text{maximize}} p(\mathbf{D} | \mathbf{K}_a \mathbf{Y}) p(\mathbf{X} | \mathbf{K}_{na} \mathbf{Y}; \mathbf{G}) p(\mathbf{D}_s^{\text{ACS}} | \mathbf{Y}_t^{\text{ACS}}; \mathbf{G}) p(\mathbf{Y}). \quad (6.14)$$

Plugging in the least-squares terms and sparsity term like before yields

$$\begin{aligned} \{\hat{\mathbf{G}}, \hat{\mathbf{Y}}\} = \underset{\mathbf{G}, \mathbf{Y}}{\text{minimize}} & \frac{1}{2} \|\text{vec}(\mathbf{D}_s^{\text{ACS}}\mathbf{G} - \mathbf{Y}_t^{\text{ACS}})\|_{\Lambda_{\text{GRAPPA}}^{\text{ACS}}}^2 + \frac{1}{2} \|\text{vec}(\mathbf{D} - \mathbf{K}_a \mathbf{Y})\|_{\Lambda_{\mathbb{I}M \times M}}^2 \\ & + \frac{1}{2} \|\text{vec}(\mathbf{D}_{\text{conv}}\mathbf{G} - \mathbf{K}_{na} \mathbf{Y})\|_{\Lambda_{\text{GRAPPA}}}^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{1,2}. \end{aligned} \quad (6.15)$$

We may be tempted to apply the same tools to this problem as we applied to previous problems that had objectives composed of least-squares terms and a joint sparsity term. However, the noise covariance matrices Λ_{GRAPPA} and $\Lambda_{\text{GRAPPA}}^{\text{ACS}}$ depend on the GRAPPA kernel \mathbf{G} , so this problem is not quite of the same form as Equation (6.11). However, given an initial estimate of the kernel, we can fix the noise covariance matrix and recompute it after each problem iteration like we would the IRLS re-weighting matrix. Ignoring the dependency on \mathbf{G} of the noise covariance matrices, the joint estimation problem can be solved for \mathbf{G} and \mathbf{Y} simultaneously, since there are no

mixed terms containing both \mathbf{G} and \mathbf{Y} . To see this, we rewrite Equation (6.15) as

$$\{\hat{\mathbf{G}}, \hat{\mathbf{Y}}\} = \underset{\mathbf{G}, \mathbf{Y}}{\text{minimize}} \frac{1}{2} \left\| \Lambda_{\text{full}}^{-1/2} \text{vec} \left(\begin{bmatrix} \mathbf{D}_s^{\text{ACS}} & -\mathbf{K}_t^{\text{ACS}} \\ \mathbf{0} & \mathbf{K}_a \\ \mathbf{D}_{\text{conv}} & -\mathbf{K}_{na} \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \mathbf{Y} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{D} \\ \mathbf{0} \end{bmatrix} \right) \right\|_2^2 + \lambda \|\Psi \mathbf{F}^{-1} \mathbf{Y}\|_{1,2}, \quad (6.16)$$

with block covariance matrix

$$\Lambda_{\text{full}} = \begin{bmatrix} \Lambda_{\text{GRAPPA}}^{\text{ACS}} & & \\ & \Lambda \otimes \mathbf{I}_{M \times M} & \\ & & \Lambda_{\text{GRAPPA}} \end{bmatrix}. \quad (6.17)$$

When vectorizing the matrix in the first term of Equation (6.16), we stack the columns of the first of the three rows, then the columns of the second row, and finish with the columns of the third row. Holding the covariance matrix as fixed, this problem can be solved using the same machinery as is used for solving the unconstrained optimization problems for DESIGN or L₁ SPIRiT. The matrix \mathbf{A} and vector \mathbf{b} for the least squares problem $\text{minimize}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ we solve in each iteration are

$$\mathbf{A} = \begin{bmatrix} \left[\begin{array}{c} \Lambda_{\text{full}} \\ \Delta^{1/2} \end{array} \right]^{-1/2} \\ \left[\begin{array}{cc} \mathbf{I}_{P \times P} \otimes \mathbf{D}_s^{\text{ACS}} & -\mathbf{I}_{P \times P} \otimes \mathbf{K}_t^{\text{ACS}} \\ \mathbf{0} & \mathbf{I}_{P \times P} \otimes \mathbf{K}_a \\ \mathbf{I}_{P \times P} \otimes \mathbf{D}_{\text{conv}} & -\mathbf{I}_{P \times P} \otimes \mathbf{K}_{na} \\ \mathbf{0} & \sqrt{2\lambda} \mathbf{I}_{P \times P} \otimes (\Psi \mathbf{F}^{-1}) \end{array} \right] \end{bmatrix}, \quad (6.18)$$

and

$$\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \Lambda \otimes \mathbf{I}_{M \times M} \text{vec}(\mathbf{D}) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \quad (6.19)$$

The complete method is described in Algorithm 5.

Algorithm 5 Algorithm for Bayesian joint estimation of GRAPPA kernel and full k-space.

Require: $\hat{\mathbf{G}}^{(0)}$, $\hat{\mathbf{Y}}^{(0)}$, I , tol

- 1: Compute noise covariance matrix $\mathbf{\Lambda}_{\text{full}}$ and objective value $f^{(0)}$ by plugging in $\hat{\mathbf{G}}^{(0)}$ and $\hat{\mathbf{Y}}^{(0)}$ into Equation (6.16).
 - 2: **for** $i = 1 : I$ **do**
 - 3: Compute $\mathbf{W} \leftarrow \mathbf{\Psi}\mathbf{F}^{-1}\hat{\mathbf{Y}}^{(i-1)}$.
 - 4: Set $\Delta_{n,n} \leftarrow \frac{1}{w[n]}$ for $w[n] \leftarrow \|[W_{n,1}, \dots, W_{n,P}, \varepsilon]\|_2$, for all $n = 0, \dots, N - 1$.
 - 5: Solve for $\hat{\mathbf{G}}^{(i)}$ and $\hat{\mathbf{Y}}^{(i)}$ using LSMR(\mathbf{A} , \mathbf{b}) with \mathbf{A} in Equation (6.18) and \mathbf{b} in Equation (6.19).
 - 6: Compute noise covariance matrix $\mathbf{\Lambda}_{\text{full}}$ from new kernel and objective value $f^{(i)}$ from $\hat{\mathbf{G}}^{(i)}$ and $\hat{\mathbf{Y}}^{(i)}$.
 - 7: **if** $f^{(i-1)} - f^{(i)} \leq \text{tol} \cdot f^{(i-1)}$ **then**
 - 8: **break**
 - 9: **end if**
 - 10: **end for**
 - 11: **return** GRAPPA kernel $\hat{\mathbf{G}}^{(i)}$ and full k-space result $\hat{\mathbf{Y}}^{(i)}$.
-

As with the Bayesian estimation of the full k-space described earlier, holding the GRAPPA kernel fixed, this joint estimation problem can also be adapted to preserve the acquired data exactly. In this case, we also hold the target ACS points in the first term of the objective in Equation (6.15) fixed.

6.2 Simulations and Results

The 32-channel T_1 -weighted image #2 acquired on a Siemens 3 T scanner as described in Chapter 4 is used again here, with an example axial slice extracted, cropped, and undersampled with uniform spacing and a 36×36 block of ACS lines. The difference images and PSNR values are computed from magnitude images of the combined reconstructed and reference data, as is done in previous chapters. The four level ‘9-7’ DWT is used as a sparsifying transform for the sparsity prior, and the tuning parameter λ for both algorithms is determined for each simulation via the two-level coarse-then-fine parameter sweep used previously.

The first set of simulations evaluates the full k-space estimation method as compared to conventional GRAPPA for different acceleration factors. The performance

comparisons are repeated for various acceleration factors, from 3×3 uniformly spaced undersampling ($R = 7.0$) up to 6×6 uniform undersampling ($R = 16.1$). In Figure 6.1, comparisons are depicted for several levels of acceleration. For the acceleration levels not shown, the estimation method increases PSNR over conventional GRAPPA from 33.1 dB to 34.6 dB for $R = 7.0$, from 8.1 dB to 26.8 dB for $R = 14.8$, and from 4.4 dB to 26.0 dB for $R = 16.1$. For all these acceleration factors, the PSNR-optimal choice of λ is the equal to the same value of 10^5 .

The GRAPPA reconstructions display rapidly increasing noise amplification with greater acceleration, from minimal noise having little effect at $R = 8.6$ to significant noise masking most features at $R = 13.6$. The Bayesian MAP estimates of the full k-space have far less noise amplification, with a slowly increasing noise level through accelerations up to $R = 13.6$. However, through the denoising, some residual aliasing is visible in the Bayesian estimated images at high accelerations. Since we uniformly space samples in k-space, we primarily rely on the GRAPPA reconstruction to undo this aliasing, as with DESIGN denoising. Thus, we would not expect the Bayesian estimation method to resolve this residual aliasing.

The next set of simulations focus on joint estimation of the kernel weights and the full k-space. The method described in Algorithm 5 is applied to the T_1 -weighted image #2 for the same range of undersampling as in the first set of experiments. In Figure 6.2, the joint estimation results for high accelerations are shown alongside conventional GRAPPA. For all these simulations, we have used a 3×3 GRAPPA kernel without regard to the acceleration factor. However, for high accelerations, such a kernel does not contain enough source points to interpolate all the missing frequencies while maintaining linear independence across target points. Increasing the number of source points to match the acceleration reduces the number of ACS fit equations, producing the type of ill-conditioned or underdetermined fits that should be regularized as discussed in Chapter 5. Rather than regularizing the calibration, we jointly estimate the kernel and full k-space here, increasing the number of fit equations by fitting the kernel to the full k-space instead of just the ACS lines. The result is shown for a 5×4 GRAPPA kernel and $R = 12.1$ acceleration (5×4 uniform

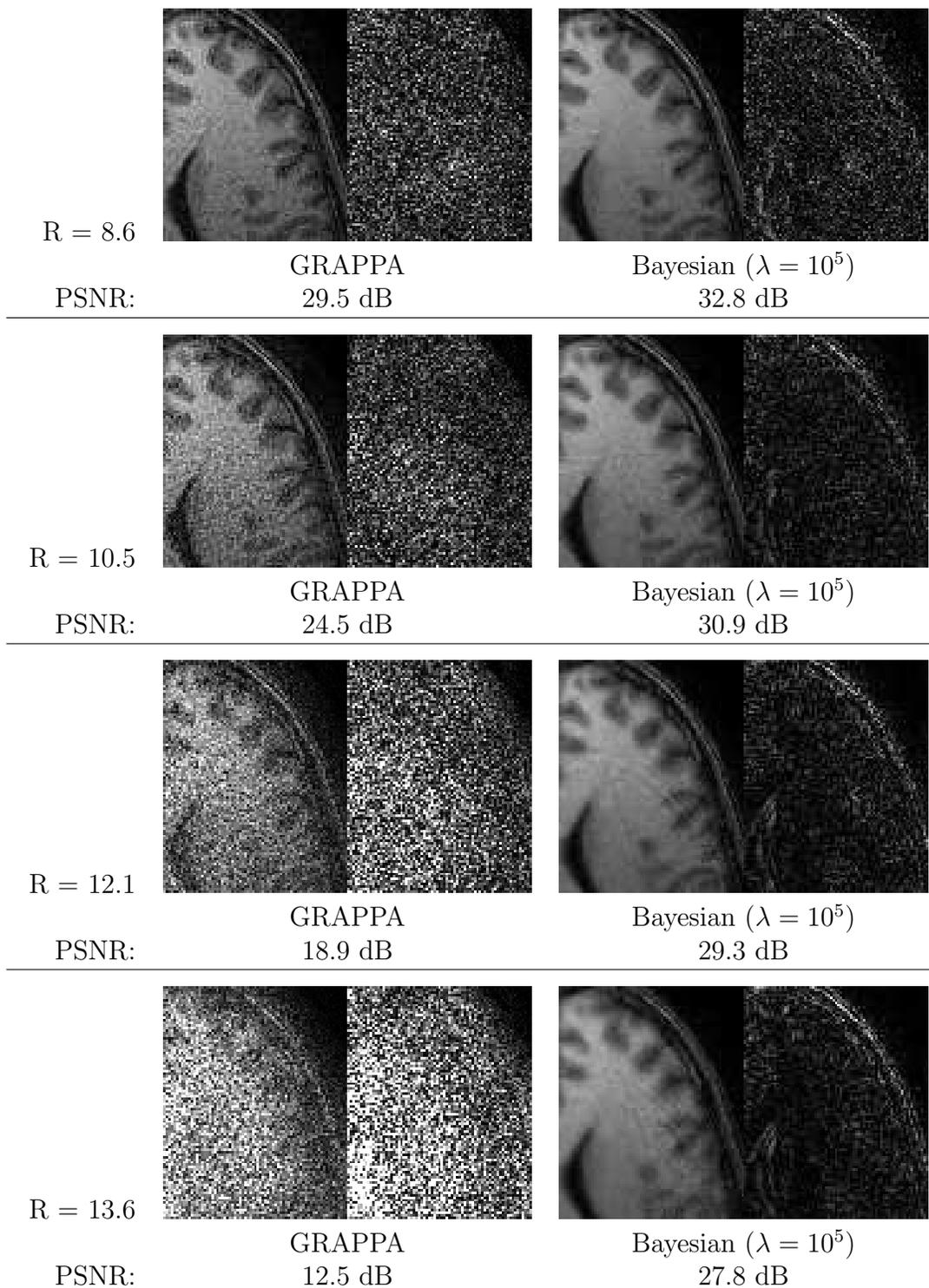


Figure 6.1: Reconstructed and difference images using conventional GRAPPA and Bayesian full k-space estimation for T_1 -weighted image #2 for several acceleration factors (nominal undersampling increases from 4×3 to 5×5).

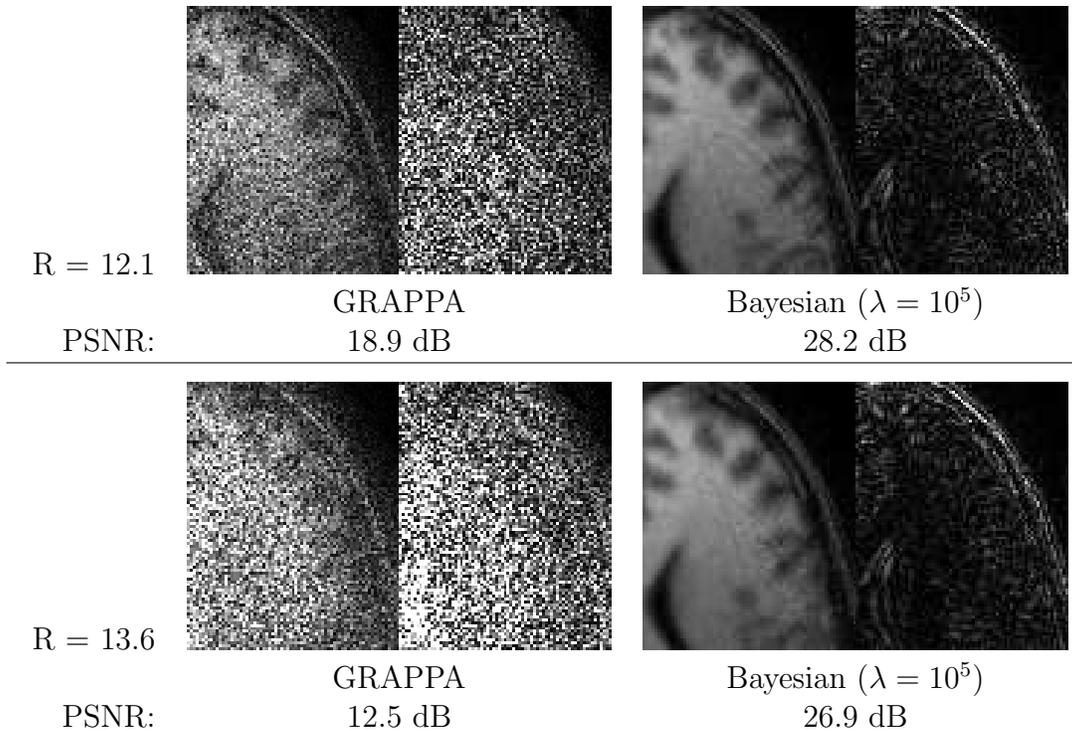


Figure 6.2: Reconstructed and difference images using conventional GRAPPA and Bayesian joint estimation of the kernel and full k-space for T_1 -weighted image #2 at higher acceleration factors (nominal undersampling 5×4 and 5×5 for the top and bottom rows, respectively).

undersampling) in Figure 6.3.

These joint estimation results are similar to the full k-space estimation results shown in Figure 6.1, with a bit lower PSNR. The jointly estimated result has similar un-aliasing to the full k-space estimate using the original GRAPPA kernel. Using a larger kernel yields a slight improvement in PSNR, although image quality does not change appreciably.

6.3 Discussion

This estimation-theoretic approach successfully extends denoising to higher accelerations than what is achievable with DESIGN denoising in Chapter 4. This improvement at higher accelerations is likely due to the noise model for the GRAPPA-reconstructed k-space. The least-squares term in Equation (6.11) accounts for the

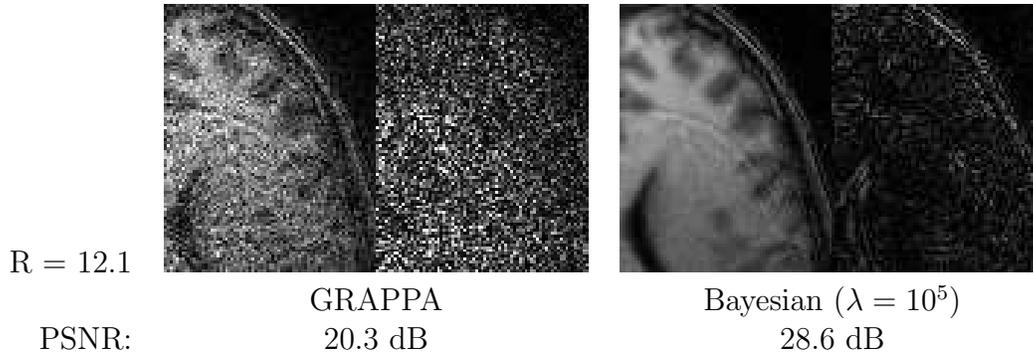


Figure 6.3: Reconstructed and difference images using conventional GRAPPA and Bayesian joint estimation of the kernel and full k-space for T_1 -weighted image #2 with a larger GRAPPA kernel (nominal undersampling is 5×4).

SNR loss of using highly accelerated GRAPPA, while the DESIGN denoising least-squares term in Equation (4.2) uses coil combination weights, which are more appropriate for un-accelerated data. In addition, the proposed estimation technique has an advantage over DESIGN denoising in that the tuning parameter is insensitive to the undersampling pattern or acceleration factor; rather, the tuning parameter is tied to the sparsity of the image to be reconstructed. Since similar images can be expected to have similar sparsity, a suitable choice of tuning parameter may be established and re-used for similar images, even across subjects. Additional simulations are necessary to validate this approach, and if the optimal choice of tuning parameter is shown to translate across acquisitions or across subjects, the benefits for practical usage could not be more significant.

While successful at denoising the GRAPPA reconstruction, the joint estimation method fails to improve the kernel calibration in the simulations presented. The main innovation in improving the kernel calibration lies in adding fit equations based on the full k-space, so that the calibration is performed on a larger set of data. However, these k-space points outside the ACS region typically possess less signal, diminishing the value of these fit equations. This joint estimation method can be combined with a regularized calibration technique like the sparsity-promoting approach described in Chapter 5; such a combination may be more effective at improving the kernel quality better than introducing more fits near the periphery of k-space at high accelerations.

Just like the sparsity-promoting GRAPPA calibration technique, the joint estimation method is limited to uniform undersampling by the computational efficiency of the GRAPPA reconstruction term $\text{GRAPPA}(\mathbf{G}, \mathbf{D})$. Using direct computation for GRAPPA reconstruction with non-uniform Cartesian subsampling preserves the structure of the optimization problem, but direct computation typically requires many kernels and cannot be implemented efficiently via convolution or the DFT. Iterative computation requires a single kernel, but the joint optimization problem would include mixed terms since the kernel would be multiplying un-acquired k-space. This approach is problematic because we are forming linear combinations of estimates with nontrivial noise covariances rather than of acquired data with fixed noise covariances, and the GRAPPA noise covariance is no longer straightforward to compute.

Chapter 7

Conclusion

Magnetic resonance imaging is a versatile and powerful tool for clinical imaging and medical research. Despite all the benefits a non-invasive, non-ionizing, whole-body imaging tool provides, the acquisition time remains a significant bottleneck to the success of MRI. Recent developments using multiple channel receiver coils and sparsity models to reconstruct MR images from undersampled data motivate combining these two complementary approaches to produce high quality images with even greater undersampling. Recent methods mainly focus on combining compressed sensing with linear inversion methods for accelerated parallel imaging reconstruction. These approaches are geared to non-uniformly undersampled or non-Cartesian sampling patterns and inverse problem formulations of parallel imaging reconstruction. Instead, we focus on uniting sparsity models with direct parallel imaging methods like GRAPPA that are suitable for conventional uniform undersampling.

In Chapters 2 and 3, we provide background on magnetic resonance imaging and sparsity. A brief description of classical MR physics and the image acquisition process is followed by a discussion of methods for accelerated imaging, including accelerated parallel imaging methods like GRAPPA. Measures of sparsity and joint sparsity are described, and uses of sparsity models including denoising and compressed sensing are explored. The chapter concludes with a survey of compressed sensing methods for MRI and previous work combining compressed sensing and accelerated parallel imaging.

In Chapters 4 through 6, we present three different approaches for augmenting the GRAPPA reconstruction method for accelerated parallel imaging using sparsity. The first method, DESIGN denoising, addresses the noise amplification problem common to all accelerated parallel imaging methods by post-processing the reconstructed image, optimizing a balance between fidelity to the GRAPPA solution and joint sparsity of the sparse transform (e.g. DWT) of the coil images, while preserving the under-sampled acquired k-space data. This denoising method appears effective at reducing the noise level present at moderately high accelerations ($R \approx 10$), while introducing minimal loss of contrast and spatial resolution. Both variants of DESIGN denoising, using the ℓ_1 norm and using the Cauchy penalty with homotopy continuation, reduce the noise amplification, as measured by g-factors, by greater than 5 dB over the GRAPPA reconstruction at $R = 10.5$ acceleration. However, the Cauchy penalty-based denoising method is both more computationally intensive and introduces more significant oversmoothing artifacts in the reconstructed image. From these results, the ℓ_1 norm is more appropriate for denoising approximately sparse images.

We observe that the DESIGN denoising method does not mitigate residual aliasing remaining from GRAPPA reconstruction in approximately sparse MR images. Based on the hypothesis that adjusting the GRAPPA kernel can improve un-aliasing, we turn our attention to regularizing the calibration step using joint sparsity. From the simulation results, regularizing the calibration to promote joint sparsity in the coil images reconstructed using the calibrated kernel both reduces noise amplification when the fit to the ACS lines is low-quality and improves un-aliasing when the fit to the ACS lines is underdetermined. By improving calibration, the proposed sparsity-promoting method shifts the trade-off between image quality and the number of ACS lines, increasing the total acceleration significantly for even moderate accelerations. Since this method applies in the underdetermined ACS fit case, we can also accommodate greater undersampling spacing with the same number of ACS lines. In combination with DESIGN denoising, this sparsity-promoting calibration method both reduces noise amplification and mitigates residual aliasing using fewer ACS lines.

Although the methods presented for denoising and calibrating the GRAPPA accelerated parallel imaging reconstruction are fairly successful at moderate accelerations, their performance degrades significantly as the sample spacing in k-space increases further. To extend the combination of GRAPPA and sparsity to higher accelerations, we turn to a Bayesian estimation framework of the full k-space reconstruction problem. We treat the acquired data and the GRAPPA reconstruction together as observations of the full k-space, with noise variances for the GRAPPA-reconstructed k-space amplified according to the calibrated GRAPPA kernel. The image sparsity yields a prior distribution on the full k-space; the signal and observation models are combined to form a MAP estimator of the full k-space. From the simulations, we infer that this MAP estimator effectively denoises GRAPPA even at high accelerations. In addition, we propose a joint estimation framework for both the full k-space and the GRAPPA kernel and use the joint estimation framework to produce high quality images from fewer ACS lines, as is done with the sparsity-promoting calibration in Chapter 5. This joint estimation method effectively combines GRAPPA and sparsity to yield a novel reconstruction technique for accelerated parallel imaging.

All these methods successfully improve GRAPPA reconstruction using sparsity in different ways. These methods are mostly limited to conventional uniform undersampling to leverage the efficient implementation of GRAPPA reconstruction as convolution using the FFT, and extensions to non-uniform Cartesian undersampling are problematic. The DESIGN denoising method is perhaps the most easily adaptable to non-uniform or non-Cartesian undersampling, since we can simply substitute another more suitable reconstruction technique for the GRAPPA reconstructed k-space in the optimization problem. The k-space (not joint) estimation technique in Chapter 6 also can be adapted easily if noise amplification is readily quantifiable; direct methods for non-uniform GRAPPA are more suitable for this purpose. Both the sparsity-promoting calibration and joint estimation methods become more computationally complex with non-uniform sampling since the GRAPPA reconstruction operator and adjoint operator must both be applied in every iteration of the algorithm. These limitations with non-uniform sampling limit the synergy achievable from

combining GRAPPA and compressed sensing in that CS typically uses non-uniform or random sampling patterns; with uniform sampling, these methods rely more on the GRAPPA piece to mitigate aliasing, which may not be sufficient at very high accelerations. However, considering the ability to improve reconstruction quality using uniform undersampling patterns means that we can both retrospectively reconstruct conventionally acquired accelerated parallel imaging data and acquire data for reconstruction with these algorithms using existing accelerated imaging pulse sequences.

Thus, we have tackled the problem of improving reconstruction of accelerated MRI using a combination of GRAPPA accelerated parallel imaging and coil image joint sparsity models. Novel methods were derived to denoise the result, improve the kernel calibration, and jointly estimate the full k-space and GRAPPA kernel. Simulations portraying the improved reconstruction quality achieved by and the drawbacks and limitations of these algorithms were performed, and the results discussed. We conclude with a discussion of future directions of this research that can further benefit the field.

Further development is required to maximize the potential of the methods described in this thesis. Fast implementations of GRAPPA exist only for uniformly undersampled k-space, but GRAPPA can be extended to non-uniform and non-Cartesian acquisitions. While DESIGN denoising readily adapts to other direct parallel imaging methods, the sparsity-promoting calibration and joint estimation methods do not generalize so easily. Direct adaptations of GRAPPA involve many more kernel weights and much slower reconstructions, while a noise model for k-space reconstructed using iterative GRAPPA methods would appear much more difficult to derive. However, a novel approach to extending the calibration or estimation methods may be able to overcome these difficulties and enable faster acquisitions with non-uniform or non-Cartesian sampling patterns.

Many variants of MRI may benefit from the proposed reconstruction or denoising techniques. Dynamic and functional imaging typically both have narrow or localized differences between successive frames. Time frame subtraction can produce significantly greater sparsity than structural imaging without a time dimension. These

methods can be expected to yield similar levels of improvement in highly undersampled k-t acquisitions. Structured noise and other artifacts can be prevalent in these and other imaging variants; DESIGN denoising, sparsity-promoting calibration, and joint Bayesian estimation may need to be combined with other post-processing methods to produce high quality images.

In addition, improved performance may be attained using more sophisticated sparse models or adaptive or learned dictionary representations. Overcomplete sparsifying transforms like contourlets, curvelets, or shearlets may outperform wavelet-based sparse representations due to the directional nature of edges in MR images. Combinations of sparsity-promoting regularizers with different sparsifying representations, like wavelets and total variation, also may reduce the artifacts present in denoised images. In addition, to be suitable for clinical applications, these methods must be applied to a variety of representative images, both normal and with assorted pathologies, and the results should be examined and graded against accepted imaging methods by trained clinicians.

Having discussed potential directions for future advancements, we conclude by remarking that the methods proposed in this thesis succeed in enabling even greater acceleration of MR imaging, several times greater than what can be accomplished using accelerated parallel imaging alone. However, as described, these contributions do not mark the end of research in advancing the science of accelerating magnetic resonance imaging. Other methods for accelerated imaging like partial Fourier and multi-shot sequences may be combined with accelerated parallel imaging and sparsity-promoting reconstruction. Using the research described in this thesis, MRI can be performed faster, and higher quality images can be acquired, lowering costs, improving subject comfort, and expanding the utility of MRI in both the clinic and research.

Appendix A

Optimization Methods

In this appendix, we present some background on the different optimization problems that appear in this thesis and common techniques for solving these problems. Not all these problems have closed-form solutions, and even when closed-form solutions exist, computing those solutions may be computationally impractical. Fortunately, efficient iterative methods exist in the literature for all these problems, ensuring that the methods proposed are feasible computationally.

A.1 Least-Squares Problems

The canonical least-squares optimization problem is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax} - \mathbf{b}\|_2^2. \quad (\text{A.1})$$

The least-squares problem finds the solution to the overdetermined linear system $\mathbf{Ax} = \mathbf{b}$ that minimizes the residual $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$. Such a solution is unique when \mathbf{A} has full column rank, and this solution satisfies the “normal equations” $\mathbf{A}^H \mathbf{Ax} = \mathbf{A}^H \mathbf{b}$. When \mathbf{A} has full column rank, $\mathbf{A}^H \mathbf{A}$ is a Hermitian symmetric positive definite matrix, and we can simply compute $\mathbf{x} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{b}$ using direct inversion, Gauss elimination, or similar exact solution method.

When the linear system is underdetermined, it may be desirable to find the

minimum-energy solution, which corresponds to minimizing the ℓ_2 norm of the solution \mathbf{x} to the underdetermined system. Using Lagrange multipliers, this constrained optimization problem is transformed into an unconstrained problem, and assuming \mathbf{A} has full row rank, the unique minimum-energy solution is $\mathbf{x} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b}$. Related to minimizing the solution energy, Tikhonov regularization also can be used to find a solution to the underdetermined system:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \|\alpha\mathbf{x}\|_2^2, \quad (\text{A.2})$$

where $\alpha > 0$ controls the amount of regularization. The least-squares parts of the objective combine to yield

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \left\| \begin{bmatrix} \mathbf{A} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2, \quad (\text{A.3})$$

which yields the solution $\mathbf{x} = (\mathbf{A}^H\mathbf{A} + \alpha^2\mathbf{I})^{-1}\mathbf{A}^H\mathbf{b}$. Note that for $\alpha > 0$, the matrix $\mathbf{A}^H\mathbf{A} + \alpha^2\mathbf{I}$ is positive definite regardless of the shape or rank of \mathbf{A} .

Although the solution of least-squares problems is straightforward to calculate, it is not always feasible to compute exact solutions when \mathbf{A} is large (many variables, many equations, or both). To approximate solutions to least-squares problems in these situations, a wide array of iterative methods have been developed, including gradient descent, conjugate gradients (CG), and more sophisticated methods like LSQR and LSMR.

A.1.1 Descent Methods: Steepest Descent and CG

The family of descent methods is broadly applicable to a wide variety of unconstrained optimization problems. As with all these iterative methods, we initialize the problem with a choice of \mathbf{x} . Each iteration of a descent method consists of roughly two parts: choosing a suitable descent direction, and minimizing the objective following a line in that direction. Of particular interest are the steepest descent and conjugate gradients

algorithms, which mainly differ in the choice of descent direction.

The steepest descent method (also called gradient descent), computes the gradient vector of the objective $f(\mathbf{x})$ we are trying to minimize using the current guess for \mathbf{x} and searches along that direction for the new \mathbf{x} that minimizes f . In the context of a least-squares problem, the gradient vector is $\nabla_{\mathbf{x}}f(\mathbf{x}) = \mathbf{A}^H \mathbf{r}$, where the residual is $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ for the current guess of \mathbf{x} . From multivariate calculus, $-\nabla_{\mathbf{x}}f(\mathbf{x})$ is the direction of steepest descent of the objective f . Then, we can perform a simple line search to find the new value of \mathbf{x} in this direction from the current \mathbf{x} that minimizes f :

$$\alpha^* = \underset{\alpha}{\text{minimize}} \|\mathbf{A}\nabla f(\mathbf{x})\alpha - \mathbf{r}\|_2^2, \quad (\text{A.4})$$

with residual \mathbf{r} . The optimal α is $((\mathbf{A}\nabla f)^H(\mathbf{A}\nabla f))^{-1}(\mathbf{A}\nabla f)^H \mathbf{r}$, and the new choice of \mathbf{x} is $\mathbf{x} - \alpha\nabla f(\mathbf{x})$. This iterative method can be repeated until the objective or the choice of \mathbf{x} has converged.

Unfortunately, the steepest descent method may take an infinite number of iterations to converge to the exact value of \mathbf{x} that solves Equation (A.1), and convergence may be exceedingly slow. This is due to the local nature of the gradient vector; while the gradient may point in the direction of steepest descent around the current point, the minimum of the function may not lie in that direction. Furthermore, the gradient at the next point is not related to the previous gradient, and the steepest descent algorithm will typically move in a zigzag pattern. As an alternative, the conjugate gradient method enforces “conjugacy” between descent vectors.

Technically, the conjugate gradient method [42, 84] describes an iterative approach to solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ for a square, symmetric matrix \mathbf{A} with full rank; fortunately, the normal equations $\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b}$ satisfy these conditions for least-squares problems with \mathbf{A} having full column rank. The CG method operates over the real field, so to extend CG to complex-valued least-squares problems, we use

$$\mathbf{A} = \begin{bmatrix} \Re\{\mathbf{A}^H \mathbf{A}\} & \Im\{\mathbf{A}^H \mathbf{A}\}^T \\ \Im\{\mathbf{A}^H \mathbf{A}\} & \Re\{\mathbf{A}^H \mathbf{A}\} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \Re\{\mathbf{b}\} \\ \Im\{\mathbf{b}\} \end{bmatrix}. \quad (\text{A.5})$$

To be unambiguous, we use the matrix ($\mathbf{A}^H \mathbf{A}$) in the conjugate gradient algorithm and assume everything is real. Two vectors \mathbf{x} and \mathbf{y} are conjugate with respect to this system if $\mathbf{x}^H \mathbf{A}^H \mathbf{A} \mathbf{y} = 0$. Let \mathbf{A} be $M \times N$; if we consider a set of N conjugate vectors $\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$, these vectors form a basis, and we can find coefficients $\alpha_1, \dots, \alpha_N$ such that $\hat{\mathbf{x}} = \sum_{n=1}^N \alpha_n \mathbf{y}_n$ solves the normal equations. Pre-multiplying both sides of the normal equations by \mathbf{y}_n^H yields $\mathbf{y}_n^H \mathbf{A}^H \mathbf{b} = \mathbf{y}_n^H \mathbf{A}^H \mathbf{A} \hat{\mathbf{x}}$. Employing conjugacy, $\mathbf{y}_n^H \mathbf{A}^H \mathbf{A} \mathbf{y}_m = 0$ for $m \neq n$, and

$$\mathbf{y}_n^H \mathbf{A}^H \mathbf{b} = \alpha_n \mathbf{y}_n^H \mathbf{A}^H \mathbf{A} \mathbf{y}_n. \quad (\text{A.6})$$

Thus, the optimal value of α_n is

$$\alpha_n = \frac{\mathbf{y}_n^H \mathbf{A}^H \mathbf{b}}{\mathbf{y}_n^H \mathbf{A}^H \mathbf{A} \mathbf{y}_n}. \quad (\text{A.7})$$

To find a suitable set of conjugate vectors, we can use an orthogonalization technique like Gram-Schmidt, initializing using the gradient at the initial guess of \mathbf{x} . In particular, to compute the next conjugate vector \mathbf{y}_{n+1} , for $n \geq 2$, the conjugate gradient method needs only the previous two normal equation residuals and the current conjugate vector \mathbf{y}_n :

$$\mathbf{y}_{n+1} = \mathbf{r}_n + \frac{\mathbf{r}_n^H \mathbf{r}_n}{\mathbf{r}_{n-1}^H \mathbf{r}_{n-1}} \mathbf{y}_n, \quad (\text{A.8})$$

where the residuals are computed as $\mathbf{r}_n = \mathbf{A}^H (\mathbf{b} - \mathbf{A} \mathbf{x}_n)$. Note that if not for the second term, the conjugate gradient method would be equivalent to steepest descent. However, that second term enforces conjugacy with the previous direction. Because the N conjugate vectors form a basis, we are guaranteed to converge to an exact solution within N iterations of the CG method (although realistically, we terminate the algorithm long before exact convergence).

An example of slow convergence of the steepest descent method and fast convergence of CG is shown in Figure A.1. In the example, the gradient descent method has not converged after 20 iterations ($\|\mathbf{r}\|_2 = 0.014$, $\|\mathbf{A} \mathbf{r}\|_2 = 0.0075$), while the conjugate gradient method converges in just two iterations. Unfortunately, the con-

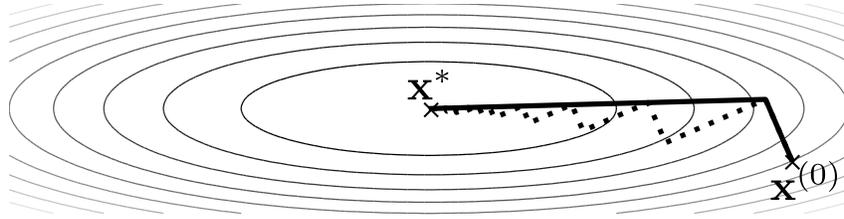


Figure A.1: Steepest descent (dotted line) and conjugate gradient (solid line) methods compared for $\mathbf{A} = [\frac{1}{4}, 0; 0, 1]$ and $\mathbf{b} = [0; 0]$.

vergence of the conjugate gradient method is rapid only for early iterations, and the method slows exponentially. More modern methods like LSQR and LSMR attempt to improve on the convergence rate of CG, although neither method has guaranteed finite convergence.

A.1.2 LSQR and LSMR

The conjugate gradient method requires that \mathbf{A} is square, symmetric, and positive definite. Since we are solving a least-squares problem, we apply CG to the normal equations $\mathbf{A}^H \mathbf{A} \mathbf{x} = \mathbf{A}^H \mathbf{b}$. When \mathbf{A} is ill-conditioned, with condition number κ , the CG method solves the normal equations with condition number κ^2 . Thus, CG may converge slowly when applied to ill-conditioned problems. The LSQR method [74] both improves convergence when \mathbf{A} is ill-conditioned and extends this problem to rank-deficient \mathbf{A} .

The LSQR method uses Golub-Kahan bidiagonalization on $[\mathbf{b} \ \mathbf{A}]$ to form unitary \mathbf{U} , \mathbf{V} , and upper bidiagonal \mathbf{B} such that $\mathbf{UB} = [\mathbf{b} \ \mathbf{A}]\mathbf{V}$. Then, if we define \mathbf{y} such that $\mathbf{x} = \mathbf{V}\mathbf{y}$, we get $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{V}\mathbf{y} = \mathbf{U}(\|\mathbf{b}\|_2[1, 0, 0, \dots]^T - \mathbf{B}\mathbf{y})$, and minimizing \mathbf{r} can be accomplished by finding the vector \mathbf{y} that minimizes $\|\|\mathbf{b}\|_2[1, 0, 0, \dots]^T - \mathbf{B}\mathbf{y}\|_2^2$ and computing $\mathbf{x} = \mathbf{V}\mathbf{y}$.

The LSQR method is effective at minimizing $\|\mathbf{r}\|_2$; however, the norm of the residual of the normal equations $\|\mathbf{A}^H \mathbf{r}\|_2$ is not nearly so well behaved, and does not necessarily decrease monotonically with each iteration. The more recent LSMR method [33] was developed to address this shortcoming by applying the bidiagonalization transformation to the problem of minimizing $\|\mathbf{A}^H \mathbf{r}\|_2$. The resulting algorithm

demonstrates monotonic behavior for both $\|\mathbf{r}\|_2$ and $\|\mathbf{A}^H \mathbf{r}\|_2$, yielding a better behaved and more robust iterative least-squares solver, even for rank-deficient \mathbf{A} . One notable downside of LSMR is that in certain rare cases, the algorithm can converge much more slowly than LSQR. Of course, one may trade the robustness of LSMR for the rapid convergence of LSQR in these situations.

A.2 Compressed Sensing Problems

The ℓ_1 norm and other penalty functions present in compressed sensing and related optimization problems motivate a great variety of iterative methods. The (noiseless) basis pursuit framework in Equation (3.31) with the signal \mathbf{y} and data \mathbf{d} belonging to the real field can be construed as a linear program:

$$\begin{aligned} \hat{\mathbf{y}} &= \underset{\mathbf{w}^+, \mathbf{w}^-, \mathbf{y}}{\text{minimize}} \mathbf{1}^T (\mathbf{w}^+ + \mathbf{w}^-) \\ \text{s.t. } &\mathbf{w}^+, \mathbf{w}^- \geq 0, \\ &\mathbf{w}^+ - \mathbf{w}^- = \Psi \mathbf{y}, \\ &\mathbf{d} = \mathbf{K}_a \mathbf{y}. \end{aligned} \tag{A.9}$$

We observe that for each index n , either \mathbf{w}_n^+ or \mathbf{w}_n^- equals zero. Otherwise, we could make both smaller and reduce the objective while still satisfying the constraints. Thus, this basis pursuit problem can be solved efficiently using the simplex algorithm or a similar method for linear programming.

When the signal is complex, and we desire to promote magnitude sparsity, or we use the $\ell_{1,2}$ joint sparsity penalty on multiple signals, the basis pursuit optimization problem is no longer expressible as a linear program; the problem is now a second-order cone program (a subtype of convex or semidefinite program):

$$\begin{aligned} \hat{\mathbf{Y}} &= \underset{\mathbf{k}, \mathbf{Y}}{\text{minimize}} \mathbf{1}^T \mathbf{k} \\ \text{s.t. } &\|\Psi_{n,:} \mathbf{Y}\|_2 \leq k_n, \quad n = 1, \dots, N \\ &\mathbf{D} = \mathbf{K}_a \mathbf{Y}. \end{aligned} \tag{A.10}$$

Here we employ the shorthand notation $\Psi_{n,:}\mathbf{Y}$ for generating the n th coefficient of the sparse vectors for each column of \mathbf{Y} . A variety of convex iterative solvers can be applied to second-order cone programs, including interior point methods like SeDuMi [88].

Another family of optimization problems arises when the data is noisy, or there is a least-squares term in the objective. The basis pursuit denoising problem with either the ℓ_1 norm for sparsity or the $\ell_{1,2}$ norm for joint sparsity in Equation (3.32) is an unconstrained convex optimization problem, which can be solved using a number of iterative methods. In general, we consider solving the regularized least-squares problem

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{minimize}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \mathcal{R}(\mathbf{C}\mathbf{x} + \mathbf{d}), \quad (\text{A.11})$$

with regularizer $\mathcal{R}(\mathbf{C}\mathbf{x} + \mathbf{d}) \geq 0$. Examples of convex regularizers include $\|\Psi\mathbf{x}\|_1$ and $\|\Psi\mathbf{x}\|_{1,2}$; other nonconvex regularizers of interest include the ℓ_p ($0 < p < 1$) and Cauchy penalty functions in Equations (3.3) and (3.6), respectively.

One broadly applicable iterative method for solving this problem for all these regularizers is called IRLS [43]. The IRLS method iteratively approximates Equation (A.11) with a weighted least-squares approximation of the regularizer, like a trust-region method:

$$\hat{\mathbf{x}}^{(i)} = \underset{\mathbf{x}}{\text{minimize}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda}{2} \|\Delta^{(i-1)}(\mathbf{C}\mathbf{x} + \mathbf{d})\|_2^2. \quad (\text{A.12})$$

The reweighting matrix $\Delta^{(i-1)}$ calibrates the second least-squares term to approximate the shape of the regularizer function around the previous iteration's estimate of \mathbf{x} . The IRLS method exhibits linear and superlinear (approaching quadratic as $p \rightarrow 0$) convergence behavior for ℓ_1 - and ℓ_p -regularized least-squares problems, respectively [25].

The closely-related approach of half-quadratic minimization [34, 35] actually refers to two different methods for approximating with quadratic functions separable regularizers of the form $\lambda \sum_n \phi(x_n)$. The core idea of the first method, which is used extensively in this thesis, is that the regularizer can be approximated by a family

Table A.1: Weights for half-quadratic minimization.

$\phi(x_n)$	$\phi'(x_n)/x_n$	$\phi''(0^+)$	$cx_n - \phi'(x_n)$
$ x_n \approx \sqrt{x_n^2 + \varepsilon^2}$	$\frac{1}{\sqrt{x_n^2 + \varepsilon^2}}$	$\frac{1}{\varepsilon}$	$\frac{x_n}{\varepsilon} - \frac{x_n}{\sqrt{x_n^2 + \varepsilon^2}}$
$ x_n ^p \approx (x_n^2 + \varepsilon^2)^{p/2}$	$\frac{p}{(x_n^2 + \varepsilon^2)^{1-p/2}}$	$\frac{p}{\varepsilon^{2-p}}$	$\frac{px_n}{\varepsilon^{2-p}} - \frac{px_n}{(x_n^2 + \varepsilon^2)^{1-p/2}}$
$\log(1 + \alpha x_n^2)$	$\frac{2\alpha}{1 + \alpha x_n^2}$	2α	$2\alpha x_n - \frac{2\alpha x_n}{1 + \alpha x_n^2}$

of quadratic functions centered at zero and multiplicatively weighted. The critical assumptions of this approach are that $\phi(\sqrt{x_n})$ is concave, $\phi''(0^+) > 0$, and $\lim_{x_n \rightarrow \infty} \phi(x_n)/x_n^2 = 0$. Using these facts, a dual function $\psi(y)$ exists such that $\phi(x_n) = \inf_y \{\frac{1}{2}yx_n^2 + \psi(y)\}$. This expression is minimized using a function $y(x_n) = \phi'(x_n)/x_n$, for $x_n \neq 0$, and $y(0) = \phi''(0^+)$. These minimizers $y(x_n)$ fill the diagonal reweighting matrix Δ in Equation (A.12). Table A.1 lists weighting formulas for a few regularizers used in this thesis. Note that this approximation requires $\phi(x_n)$ is twice differentiable, so smoothed approximations are used for both the ℓ_1 and ℓ_p^p measures. The second approach uses an additive shift instead of a multiplicative rescaling of the quadratic to approximate the regularizer. This approximation replaces $\phi(x_n)$ with $(x_n - s)^2$, where $s(x_n) = cx_n - \phi'(x_n)$, and the optimal choice of c is $\sup_{x_n} \phi''(x_n)$ [67]. The optimal value of c is equal to $\phi''(0^+)$ for the regularizers in Table A.1.

In Equation (A.12), the reweighting matrix $\Delta^{(i-1)}$ can be specified according to half-quadratic minimization. Let $\mathbf{w} = \mathbf{C}\mathbf{x}^{(i-1)} + \mathbf{d}$. Then, for $n = 1, \dots, N$, $\Delta_{n,n}^{(i-1)} = \phi'(w_n)/w_n$, using the expressions specified in Table A.1. For a row-separable matrix penalty $\mathcal{R}(\mathbf{C}\mathbf{X} + \mathbf{D})$ of the form

$$\mathcal{R}(\mathbf{W}) = \sum_{n=1}^N \phi(s([W_1[n], \dots, W_P[n]])), \quad (\text{A.13})$$

we instead compute $w_n = s([W_1[n], \dots, W_P[n]])$, for $\mathbf{W} = \mathbf{C}\mathbf{X}^{(i-1)} + \mathbf{D}$, and set the reweighting matrix to $\Delta_{n,n}^{(i-1)} = \phi'(w_n)/w_n$. This extension of the half-quadratic method also applies to vector penalties on the magnitudes of the elements of a complex-valued vector. Matrix norms such as $\ell_{1,2}$ and $\ell_{1,\infty}$ can be approximated using this approach.

Split-Bregman iteration [102] is a valuable method for solving problems of the form of Equation (A.11). We begin by introducing an auxiliary variable $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{d}$, and transform Equation (A.11) into a constrained optimization over both \mathbf{x} and \mathbf{y} :

$$\{\hat{\mathbf{x}}, \hat{\mathbf{y}}\} = \underset{\mathbf{x}, \mathbf{y}}{\text{minimize}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \mathcal{R}(\mathbf{y}), \text{ s.t. } \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{d}. \quad (\text{A.14})$$

Bregman iterative regularization is then used to solve this problem through a series of iterations of the form

$$\begin{aligned} \{\mathbf{x}^{(i+1)}, \mathbf{y}^{(i+1)}\} = \underset{\mathbf{x}, \mathbf{y}}{\text{minimize}} & D_J^{\mathbf{p}^{(i)}}(\{\mathbf{x}, \mathbf{y}\}, \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}) \\ & + \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{1}{2} \|\mathbf{y} - \mathbf{C}\mathbf{x} - \mathbf{d}\|_2^2, \end{aligned} \quad (\text{A.15})$$

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} - \begin{bmatrix} \mathbf{A}^H & -\mathbf{C}^H \\ 0 & \mathbf{I} \end{bmatrix} \left(\begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(i+1)} \\ \mathbf{y}^{(i+1)} \end{bmatrix} - \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix} \right), \quad (\text{A.16})$$

where $D_J^{\mathbf{p}}(\{\mathbf{x}, \mathbf{y}\}, \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\})$ is the Bregman “distance”

$$D_J^{\mathbf{p}}(\{\mathbf{x}, \mathbf{y}\}, \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}) = J(\{\mathbf{x}, \mathbf{y}\}) - J(\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}) - \langle \mathbf{p}, \{\mathbf{x} - \mathbf{x}^{(i)}, \mathbf{y} - \mathbf{y}^{(i)}\} \rangle, \quad (\text{A.17})$$

for regularizer $J(\{\mathbf{x}, \mathbf{y}\}) = \lambda \mathcal{R}(\mathbf{y})$. The first step can be solved for \mathbf{x} and \mathbf{y} separately. Solving a least-squares problem yields \mathbf{x} , and each element of \mathbf{y} can be computed efficiently using the soft-thresholding operator. The second step consists of matrix-vector multiplications and is straightforward to implement, too. This Bregman iteration approach can be applied to basis pursuit problems like Equation (3.31) as well, skipping the variable splitting step. These Bregman iterations have been shown to be equivalent to the augmented Lagrangian method [69], so results concerning convergence and numerical stability for this well-known method carry over to split-Bregman iteration.

A final set of methods for solving compressed sensing-type problems utilizes a Bayesian interpretation of the compressed sensing framework and uses belief propagation over the graph of the problem to converge to a solution rather efficiently [4, 29, 78]. These methods rely on the large scale limiting behavior of the associated

graphical models to reduce the CS reconstruction problem to iterating simple scalar estimators. Thus, they scale well to large problems with many variables. While not used explicitly in this thesis, modifications of these methods may increase the computational speed of the proposed sparsity-promoting algorithms greatly. Thus, developing such modifications is of great interest.

Bibliography

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, Nov. 2006.
- [2] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. Inform. Theory*, 56(4):1982–2001, Apr. 2010.
- [3] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk. An information-theoretic approach to distributed compressed sensing. In *Proc. 43rd Allerton Conf. Commun., Cont., Comput.*, Sept. 2005.
- [4] D. Baron, S. Sarvotham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Trans. Signal Process.*, 58(1):269–280, Jan. 2010.
- [5] P. J. Beatty, K. F. King, L. Marinelli, C. J. Hardy, and M. Lustig. Sequential application of parallel imaging and compressed sensing. In *Proc. ISMRM 17th Scientific Meeting*, page 2824, Apr. 2009.
- [6] J. W. Belliveau, D. N. Kennedy, R. C. McKinstry, B. R. Buchbinder, R. M. Weisskoff, M. S. Cohen, J. M. Vevea, T. J. Brady, and B. R. Rosen. Functional mapping of the human visual cortex by magnetic resonance imaging. *Science*, 254(5032):716–719, Nov. 1991.
- [7] M. A. Bernstein, K. F. King, and X. J. Zhou. *Handbook of MRI Pulse Sequences*. Elsevier Academic Press, Burlington, MA, 2004.
- [8] B. Bilgic, V. K. Goyal, and E. Adalsteinsson. Multi-contrast reconstruction with Bayesian compressed sensing. *Magn. Reson. Med.*, 66(6):1601–1615, Dec. 2011.
- [9] J. E. Bishop, G. E. Santyr, F. Kelcz, and D. B. Plewes. Limitations of the Keyhole technique for quantitative dynamic contrast-enhanced breast MRI. *J. Magn. Reson. Imag.*, 7(4):716–723, July/August 1997.
- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge UP, New York, 2004.

- [11] A. C. Brau, P. J. Beatty, S. Skare, and R. Bammer. Comparison of reconstruction accuracy and efficiency among autocalibrating data-driven parallel imaging methods. *Magn. Reson. Med.*, 59(2):382–95, 2008.
- [12] F. A. Breuer, S. A. Kannengiesser, M. Blaimer, N. Seiberlich, P. M. Jakob, and M. A. Griswold. General formulation for quantitative G-factor calculation in GRAPPA reconstructions. *Magn. Reson. Med.*, 62(3):739–746, Sept. 2009.
- [13] G. M. Bydder, R. E. Steiner, I. R. Young, A. S. Hall, D. J. Thomas, J. Marshall, C. A. Pallis, and N. J. Legg. Clinical NMR imaging of the brain: 140 cases. *Am. J. Roentgenol.*, 139(2):215–236, 1982.
- [14] M. Bydder and Y. Jung. A nonlinear regularization strategy for GRAPPA calibration. *Magn. Reson. Imaging*, 27(1):137–41, 2009.
- [15] E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Model. Simul.*, 5(3):861–899, Jan. 2006.
- [16] E. J. Candès and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6(2):227–54, 2006.
- [17] E. J. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Prob.*, 23(3):969–986, June 2007.
- [18] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, Feb. 2006.
- [19] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, Dec. 2005.
- [20] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inform. Theory*, 52(12):5406–25, Dec. 2006.
- [21] E. J. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n . *Ann. Statist.*, 35(6):2313–2351, Dec. 2007.
- [22] A. E. Chang, Y. L. Matory, A. J. Dwyer, S. C. Hill, M. E. Girton, S. M. Steinberg, R. H. Knop, J. A. Frank, D. Hyams, and J. L. Doppman. Magnetic resonance imaging versus computed tomography in the evaluation of soft tissue tumors of the extremities. *Ann. Surg.*, 205(4):340–348, Apr. 1987.
- [23] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14(10):707–710, Oct. 2007.
- [24] Y. Chen, X. Ye, and F. Huang. A novel method and fast algorithm for MR image reconstruction with significantly under-sampled data. *AIMS Inverse Probl. Imag.*, 4(2):223–240, May 2010.

- [25] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math.*, 63(1):1–38, Jan. 2010.
- [26] M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Process.*, 14(12):2091–2106, Dec. 2005.
- [27] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, Apr. 2006.
- [28] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, Aug. 1994.
- [29] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proc. Natl. Acad. Sci. USA*, 106(45):18914–18919, Nov. 2009.
- [30] D. A. Feinberg and K. Oshio. GRASE (gradient- and spin-echo) MR imaging: a new fast clinical imaging technique. *Radiology*, 181:597–602, Nov. 1991.
- [31] A. Fischer, N. Seiberlich, M. Blaimer, P. Jakob, F. Breuer, and M. Griswold. A combination of nonconvex compressed sensing and GRAPPA (CS-GRAPPA). In *Proc. ISMRM 17th Scientific Meeting*, page 2813, Apr. 2009.
- [32] B. Fischl and A. M. Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proc. Natl. Acad. Sci. USA*, 97(20):11050–11055, Sept. 2000.
- [33] D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011.
- [34] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(3):367–83, 1992.
- [35] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Process.*, 4(7):932–46, 1995.
- [36] P. R. Gill, A. Wang, and A. Molnar. The in-crowd algorithm for fast basis pursuit denoising. *IEEE Trans. Signal Process.*, 59(10):4595–4605, Oct. 2011.
- [37] L. J. Grady and J. R. Polimeni. Nullspace compressed sensing for accelerated imaging. In *Proc. ISMRM 17th Scientific Meeting*, page 2820, Apr. 2009.
- [38] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn. Reson. Med.*, 47(6):1202–10, 2002.

- [39] K. Guo and D. Labate. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.*, 39(1):298–318, 2007.
- [40] C. L. G. Ham, J. M. L. Engels, G. T. van de Wiel, and A. Machielsen. Peripheral nerve stimulation during MRI: Effects of high gradient amplitudes and switching rates. *J. Magn. Reson. Imag.*, 7(5):933–937, September/October 1997.
- [41] J. Hennig, A. Nauerth, and H. Friedburg. RARE imaging: a fast imaging method for clinical MR. *Magn. Reson. Med.*, 3(6):823–833, Dec. 1986.
- [42] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49(6):409–436, Dec. 1952.
- [43] P. W. Holland and R. E. Welsch. Robust regression using iteratively re-weighted least-squares. *Commun. Statist. A: Theory Meth.*, 6(9):813–827, 1977.
- [44] P. M. Jakob, M. A. Griswold, R. R. Edelman, and D. K. Sodickson. AUTO-SMASH: a self-calibrating technique for SMASH imaging. simultaneous acquisition of spatial harmonics. *MAGMA*, 7(1):42–54, 1998.
- [45] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *J. Machine Learning Research*, 12:2777–2824, Oct. 2011.
- [46] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall Signal Processing Series. Prentice Hall, Upper Saddle River, NJ, 1993.
- [47] S.-G. Kim and J. J. H. Ackerman. Quantification of regional blood flow by monitoring of exogenous tracer via nuclear magnetic resonance spectroscopy. *Magn. Reson. Med.*, 14(2):266–282, May 1990.
- [48] P. K. Kythe and M. R. Schäferkötter. *Handbook of Computational Methods for Integration*. Chapman and Hall/CRC, Boca Raton, FL, 2005.
- [49] P. Lai, M. Lustig, P. Beatty, A. C. Brau, S. S. Vasanawala, and M. Alley. Faster L1SPIRiT reconstruction with parallel imaging initialization. In *Proc. ISMRM 3rd Int. Workshop Parallel MRI*, Oct. 2009.
- [50] P. Lai, T. Zhang, M. Lustig, S. S. Vasanawala, and A. C. Brau. Improving compressed sensing parallel imaging using autocalibrating parallel imaging initialization with variable density tiled random k-space sampling. In *Proc. ISMRM 19th Scientific Meeting*, page 68, May 2011.
- [51] J.-S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-2(2):165–168, Mar. 1980.
- [52] D. Liang, B. Liu, J. Wang, and L. Ying. Accelerating SENSE using compressed sensing. *Magn. Reson. Med.*, 62(6):1574–1584, Dec. 2009.

- [53] F. H. Lin. B1 simulator : Simulate B1 field for MRI RF coils. http://www.nmr.mgh.harvard.edu/~fhlin/tool_b1.htm, July 2005.
- [54] F. H. Lin, K. K. Kwong, J. W. Belliveau, and L. L. Wald. Parallel imaging reconstruction using automatic regularization. *Magn. Reson. Med.*, 51(3):559–67, 2004.
- [55] B. Liu, F. M. Sebert, Y. Zou, and L. Ying. SparseSENSE: Randomly-sampled parallel imaging using compressed sensing. In *Proc. ISMRM 16th Scientific Meeting*, page 3154, Apr. 2008.
- [56] M. Lustig, M. Alley, S. Vasanawala, D. L. Donoho, and J. M. Pauly. L₁ SPIRiT: Autocalibrating parallel imaging compressed sensing. In *Proc. ISMRM 17th Scientific Meeting*, page 379, Apr. 2009.
- [57] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58(6):1182–95, 2007.
- [58] M. Lustig and J. M. Pauly. Iterative GRAPPA: a general solution for the GRAPPA reconstruction from arbitrary k-space sampling. In *Proc. ISMRM 15th Scientific Meeting*, page 333, May 2007.
- [59] M. Lustig and J. M. Pauly. SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magn. Reson. Med.*, 64(2):457–471, Aug. 2010.
- [60] A. Majumdar and R. K. Ward. Nuclear norm regularized SENSE reconstruction. *Magn. Reson. Imag.*, 30(1):9–18, 2012.
- [61] D. M. Malioutov, M. Çetin, and A. S. Willsky. Homotopy continuation for sparse signal representation. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, volume 5, pages v/733–v/736, Mar. 2005.
- [62] P. Mansfield. Multi-planar image formation using NMR spin echoes. *J. Phys. C: Solid State Phys.*, 10(3):L55–L58, Feb. 1977.
- [63] P. Mansfield, R. Coxon, and J. Hykin. Echo-volumar imaging (EVI) of the brain at 3.0 T: First normal volunteer and functional imaging results. *J. Comput. Assist. Tomogr.*, 19(6):847–852, Nov/Dec 1995.
- [64] M. Mishali and Y. C. Eldar. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. *IEEE Trans. Signal Process.*, 56(10):4692–4702, Oct. 2008.
- [65] M. Murphy. l1-SPIRiT MRI reconstruction. <http://www.cs.berkeley.edu/~mjmurphy/l1spirit.html>, Apr. 2010.

- [66] M. Murphy, K. Keutzer, S. Vasanaawala, and M. Lustig. Clinically feasible reconstruction time for L1-SPIRiT parallel imaging and compressed sensing MRI. In *Proc. ISMRM 18th Scientific Meeting*, page 4854, May 2010.
- [67] M. Nikolova and M. K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM J. Sci. Comput.*, 27(3):937–966, Dec. 2005.
- [68] D. G. Nishimura. *Principles of Magnetic Resonance Imaging*. Stanford, Palo Alto, CA, Apr. 1996.
- [69] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, second edition, July 2006.
- [70] D. C. Noll, D. G. Nishimura, and A. Macovski. Homodyne detection in magnetic resonance imaging. *IEEE Trans. Med. Imag.*, 10(2):154–163, June 1991.
- [71] S. Ogawa, T. M. Lee, A. R. Kay, and D. W. Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proc. Natl. Acad. Sci. USA*, 87(24):9868–9872, Dec. 1990.
- [72] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Upper Saddle River, NJ, third edition, 2010.
- [73] C. C. Paige and M. A. Saunders. Algorithm-583 - LSQR - sparse linear-equations and least-squares problems. *ACM Trans. on Mathematical Software*, 8(2):195–209, 1982.
- [74] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, Mar. 1982.
- [75] C. Prieto, B. R. Knowles, M. Usman, P. G. Batchelor, F. Odille, D. Atkinson, and T. Schaeffter. Autocalibrated approach for the combination of compressed sensing and SENSE. In *Proc. ISMRM 18th Scientific Meeting*, page 4862, May 2010.
- [76] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magn. Reson. Med.*, 42(5):952–62, 1999.
- [77] S. Ramani and J. A. Fessler. Parallel MR image reconstruction using augmented Lagrangian methods. *IEEE Trans. Med. Imag.*, 30(3):694–706, Mar. 2011.
- [78] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 2168–2172, Aug. 2011.
- [79] S. Rangan, A. K. Fletcher, and V. K. Goyal. Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing. *IEEE Trans. Inform. Theory*, 58(3):1902–1923, Mar. 2012.

- [80] S. Ravishankar and Y. Bresler. MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Trans. Med. Imag.*, 30(5):1028–1041, May 2011.
- [81] P. M. Robson, A. K. Grant, A. J. Madhuranthakam, R. Lattanzi, D. K. Sodickson, and C. A. McKenzie. Comprehensive quantification of signal-to-noise ratio and g-factor for image-based and k-space-based parallel imaging reconstructions. *Magn. Reson. Med.*, 60(4):895–907, Oct. 2008.
- [82] P. B. Roemer, W. A. Edelstein, C. E. Hayes, S. P. Souza, and O. M. Mueller. The NMR phased array. *Magn. Reson. Med.*, 16(2):192–225, 1990.
- [83] N. Seiberlich, F. A. Breuer, M. Blaimer, K. Barkauskas, P. M. Jakob, and M. A. Griswold. Non-cartesian data reconstruction using GRAPPA operator gridding (GROG). *Magn. Reson. Med.*, 58(6):1257–1265, Dec. 2007.
- [84] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. <http://www.cs.berkeley.edu/~jrs/jrspapers.html#cg>, Aug. 1994.
- [85] C. A. Sing-Long, C. A. Tejos, and P. Irarrazaval. Evaluation of continuous approximation functions for the l0-norm for compressed sensing. In *Proc. ISMRM 17th Scientific Meeting*, page 4585, Apr. 2009.
- [86] D. S. Smith and E. B. Welch. Non-sparse phantom for compressed sensing MRI reconstruction. In *Proc. ISMRM 19th Scientific Meeting*, page 2845, May 2011.
- [87] D. K. Sodickson and W. J. Manning. Simultaneous acquisition of spatial harmonics (SMASH): Fast imaging with radiofrequency coil arrays. *Magn. Reson. Med.*, 38(4):591–603, 1997.
- [88] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.21 available from <http://sedumi.ie.lehigh.edu/>.
- [89] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc., Ser. B (Method.)*, 58(1):267–288, 1996.
- [90] A. N. Tikhonov and V. I. A. Arsenin. *Solutions of Ill-Posed Problems*. Winston; distributed solely by Halsted Press, Washington/New York, 1977.
- [91] J. Trzasko and A. Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0 -minimization. *IEEE Trans. Med. Imag.*, 28(1):106–121, Jan. 2009.
- [92] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, Nov. 2008.

- [93] J. J. Van Vaals, M. E. Brummer, W. T. Dixon, H. H. Tuithof, H. Engels, R. C. Nelson, B. M. Gerety, J. L. Chezmar, and J. A. Den Boer. “Keyhole” method for accelerating imaging of contrast agent uptake. *J. Magn. Reson. Imag.*, 3(4):671–675, July/August 1993.
- [94] R. Ward. Compressed sensing with cross validation. *IEEE Trans. Inf. Theory*, 55(12):5773–5782, Dec. 2009.
- [95] D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Combined compressed sensing and parallel MRI compared for uniform and random cartesian undersampling of k-space. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, pages 553–6, May 2011.
- [96] D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Evaluating sparsity penalty functions for combined compressed sensing and parallel MRI. In *Proc. IEEE Int. Symp. on Biomedical Imaging*, pages 1589–92, March–April 2011.
- [97] D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Regularizing grappa using simultaneous sparsity to recover denoised images. In *Proc. SPIE Wavelets and Sparsity XIV*, volume 8138, pages 81381M–1–9, Aug. 2011.
- [98] D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. SpRING: Sparse reconstruction of images using the nullspace method and GRAPPA. In *Proc. ISMRM 19th Scientific Meeting*, page 2861, May 2011.
- [99] D. S. Weller, J. R. Polimeni, L. Grady, L. L. Wald, E. Adalsteinsson, and V. K. Goyal. Denoising sparse images from GRAPPA using the nullspace method (DESIGN). *Magn. Reson. Med.*, to appear.
- [100] G. C. Wiggins, J. R. Polimeni, A. Potthast, M. Schmitt, V. Alagappan, and L. L. Wald. 96-Channel receive-only head coil for 3 Tesla: Design optimization and evaluation. *Magn. Reson. Med.*, 62(3):754–762, Sept. 2009.
- [101] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, July 2009.
- [102] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing. *SIAM J. Imag. Sciences*, 1(1):143–168, 2008.
- [103] L. Ying and J. Sheng. Joint image reconstruction and sensitivity estimation in SENSE (JSENSE). *Magn. Reson. Med.*, 57(6):1196–1202, June 2007.
- [104] A. C. Zelinski, V. K. Goyal, and E. Adalsteinsson. Simultaneously sparse solutions to linear inverse problems with multiple system matrices and a single observation vector. *SIAM J. Sci. Comput.*, 31(6):4533–4579, Jan. 2010.