Acceleration Methods for MRI

by

Matthew J. Muckley

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biomedical Engineering)
in the University of Michigan
2016

Doctoral Committee:

Professor Jeffrey A. Fessler, Co-Chair
Professor Douglas C. Noll, Co-Chair
Assistant Professor Laura K. Balzano
Research Associate Professor Luis Hernandez-Garcia
Associate Research Scientist Scott J. Peltier

2016

To my mother and father, who raised me and taught me how to think.

ACKNOWLEDGMENTS

First of all, I'd like to thank my advisers, Doug Noll and Jeff Fessler. I'm not sure how I convinced them to let me be their student. When I first came to Michigan, I did not know what a norm was and I could barely write out a matrix multiplication. This dissertation contains ample amounts of both. Doug has always been an excellent guide for keeping my focus on the application at hand. As I've become more technical in my work, I often find that Doug's questions are the hardest to answer as they push me from thinking about *how* to do something to *why* I'm doing it. Jeff has been an incredible mentor for me as I developed my mathematical skills. He is always able to give intuitive explanations for very difficult concepts, and his enthusiasm and energy in every meeting was a constant inspiration for me in my career. I could not have become a successful graduate student without the mentorship of my senior lab mates. Even today, much of my intuition and understanding of MRI comes from conversations I had with Daehyun Yoon and Feng Zhao during my first two years. They were always patient and enthusiastic with me when I knew so little. The work of Sathish Ramani, Donghwan Kim, Hung Nien, and Madison McGaffin is readily apparent throughout this dissertation, and I can happily say that much of my research was spent trying to understand if their ideas for X-Ray CT could be extended to MRI. I'd also like to thank my other lab mates, Gopal Nataraj, Kathleen Ropella, Alan Chu, Hao Sun, Yash Shah, Tianrui Luo, Sydney Williams, Mai Le, Paige Castle, Steve Schmitt, Dan Weller, and Brian Gonzalez for the numerous research discussions (and good times!) that we've had over the years. I cannot overstate how important my friends in Ann Arbor are to me. Nidhi, Sriram, Sydney, Gopal, Janki, John, David, and Adrienne were a constant source of joy for me over the last five years, and I'm going to miss you all greatly. I'd also like to thank my girlfriend, Kara. Her presence, love, and support over the last few months have made writing this dissertation one of the funnest parts of my graduate career. Lastly, I'd like to thank my family. My brother Brandon has both encouraged me with kind words and inspired me with his own incredible achievements. My mom and dad are everything to me. I love you both, and I wouldn't have gotten half this far without you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Acceleration Methods for MRI

by

Matthew J. Muckley

Co-Chairs: Jeffrey A. Fessler and Douglas C. Noll

Acceleration methods are a critical area of research for MRI. Two of the most important acceleration techniques involve parallel imaging and compressed sensing. These advanced signal processing techniques have the potential to drastically reduce scan times and provide radiologists with new information for diagnosing disease. However, many of these new techniques require solving difficult optimization problems, which motivates the development of more advanced algorithms to solve them. In addition, acceleration methods have not reached maturity in some applications, which motivates the development of new models tailored to these applications. This dissertation makes advances in three different areas of accelerations. The first is the development of a new algorithm (called B1-Based, Adaptive Restart, Iterative Soft Thresholding Algorithm or BARISTA), that solves a parallel MRI optimization problem with compressed sensing assumptions. BARISTA is shown to be 2-3 times faster and more robust to parameter selection than current state-of-the-art variable splitting methods. The second contribution is the extension of BARISTA ideas to non-Cartesian trajectories that also leads to a 2-3 times acceleration over previous methods. The third contribution is the development of a new model for functional MRI that enables a 3-4 factor of acceleration of effective temporal resolution in functional MRI scans. Several variations of the new model are proposed, with an ROC curve analysis showing that a combination low-rank/sparsity model giving the best performance in identifying the resting-state motor network.

CHAPTER 1

Introduction

Acceleration methods are currently one of the foremost topics of research in MRI. This dissertation discusses two forms of acceleration. One form of acceleration involves acquiring less data and reconstructing an image of the same resolution. Another acceleration involves reconstructing the images faster than existing methods for undersampled data once the data is acquired. These two forms of acceleration are often intertwined. Acceleration methods based on parallel imaging [1] or compressed sensing [2] can necessitate solving complicated optimization problems. In some clinical settings such as DCE MRI [3], these problems can be solved offline, but for many applications the images need to be available as soon as possible while the patient is at the clinic. These optimization problems can involve large system matrices and nondifferentiable functions, which precludes the use of many standard optimization algorithms. When the cost function is quadratic, then the conjugate gradient algorithm is often an excellent choice, but when compressed sensing is used the optimization problem is no longer quadratic. In some cases, complicated system matrices can make even a quadratic optimization problem difficult to solve efficiently [4]. These considerations all motivate the development of advanced optimization algorithms.

The current state-of-the-art optimization algorithms for subsampled MRI are based on variable splitting. These algorithms can be developed from the augmented Lagrangian [5] or split-Bregman [6] perspective for compressed sensing problems, and they have been extended to the MRI setting [7]. Variable-splitting methods work by decomposing a complicated optimization problem into simple subproblems. The subproblems often have closed-form solutions that are simple to compute. However, a drawback in creating this decomposition is the need to select penalty parameters that connect the subproblems. These penalty parameters can greatly affect the convergence speed of variable-splitting methods, but are difficult to tune for the fastest convergence rate. The best penalty parameters can change depending on the data set, regularizer, or regularization parameter. As such, it can be difficult to assess the robustness of variable-splitting methods for clinical imaging.

This work explores the development of momentum-accelerated majorize-minimize algorithms as an alternative to variable splitting algorithms. The difficulty in designing majorize-minimize algorithms is the derivation of a majorizer, a function whose minimization guarantees a decrease in the function of the primary optimization problem. Procedures for developing majorizers can vary greatly depending on the optimization problem. In this dissertation we cover two cases specific to MRI: parallel MRI with Cartesian sampling [8] and parallel MRI with non-Cartesian sampling. Both of these settings require careful consideration of the MRI system matrix and the regularizer. In the Cartesian sampling setting, we use coupling effects between the receive sensitivity coils and the spatial localization property of the wavelet transform to get space-variant step sizes in a majorize-minimize algorithm. In the non-Cartesian setting, we use coupling effects between the sampling density function of the k-space trajectory and the frequency localization property of the wavelet transform to get frequency-dependent step sizes. Both of these approaches require mathematical techniques for upper bounding positive semidefinite matrices, which are covered in chapters 3 and 4. A parallel development is the extension of adaptive momentum restarting techniques [9] and optimized momentum [10] to MRI [8, 11].

A further advance to acceleration methods is the development of new modeling techniques to facilitate increases in temporal resolution for dynamic imaging. Low-rank models have been proposed for a number of MRI applications such as cardiac MRI [12] and dynamic contrast-enhanced (DCE) MRI [3]. These models have also been extended to functional MRI (fMRI) as well, although low rank methods in fMRI thus far have only included strictly low rank regularizers [13]. Since fMRI data are not strictly low rank, large ranks are necessary to represent the richness of functional activations [13]. Here we propose an alternative hybrid model that blends low-rank methods with other regularizers that allow more rich, full-rank reconstructions in chapter 5.

CHAPTER 2

Background on MR Image Reconstruction

Here we give a basic overview on MRI physics and the corresponding reconstruction methods. We choose to derive these methods from a statistical signal processing perspective. Although many of the described techniques were developed from perspectives other than that of statistical signal processing, the benefit of this approach is that it is very general and allows a streamlined treatment of a wide class of techniques used in MRI from the same basic framework.

Throughout all of this, the source of our signal will be single proton nuclei in water molecules. These are the most abundant nuclei that are detectable by nuclear magnetic resonance in the human body, and as such the signal from them far outweighs that of all other atomic species in MRI applications.

2.1 The MRI Signal Equation

We will assume there is a magnetic field, B_0, that is aligned in the z-direction. When placed in this magnetic field, the nuclear magnetic moments of the hydrogen nuclei (so-called *spins*) precess around the z-axis at the Larmor frequency:

$$\omega_0 = \gamma B_0, \tag{2.1}$$

where γ is the gyromagnetic ratio, a constant that changes depending on the atomic species.

The steady-state condition is for the spins to be aligned with the z-axis. To obtain a signal, a radiofrequency (RF) pulse must be applied to tip the spins into the transverse plane. This pulse effectively applies a torque to the net magnetization vector that can be used to move it into the transverse plane. The degree to which the net magnetization vector is rotated depends on the duration of the RF pulse.

When the spins are in the transverse plane, relaxation occurs. The return of the magnetization vector to its equilibrium state is governed by two time constants: T_1 and T_2. T_1

governs the recovery of the magnetization vector along the z-axis, i.e.

$$M_z(t) = M_{z0}\left(1 - e^{-t/T_1}\right) \tag{2.2}$$

for a 90-degree flip. T_2 correspondingly governs the decay of the net magnetization vector in the xy-plane, i.e.

$$M_{xy}(t) = M_{z0}e^{-t/T_2} \tag{2.3}$$

for a 90-degree flip. Different tissues have different T_1 and T_2 values. As such, variations in signal intensities in MR images result from differences in T_1 and T_2 in different tissues, and these variations are the source of contrast in MRI. Functional MRI has an extra parameter of interest: T_2^*, which includes the effects of small variations in the magnetic field due to the blood oxygenation level dependent (BOLD) effect. The BOLD effect creates contrast between activated and deactivated brain regions. The BOLD effect is a vascular phenomenon that occurs at frequencies slower than 0.1 Hz. A standard BOLD response is shown in Figure 2.1. Due to the low-frequency aspect of the BOLD response, fMRI acqui-

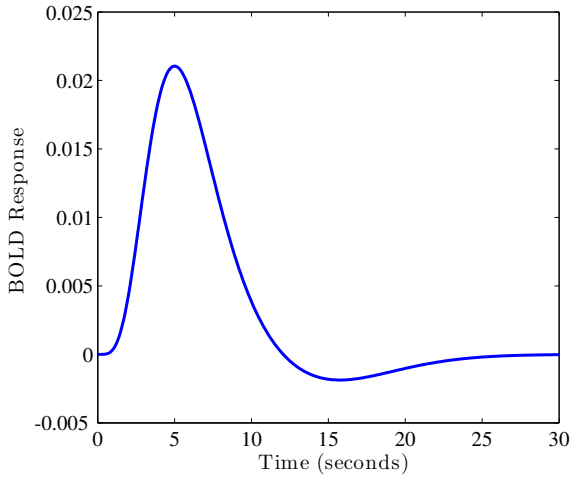


Figure 2.1: The canonical bold hemodynamic response function.

sitions with a repetition time as slow as 2 seconds are used in fMRI experiments. Typically, the data acquisition window in MRI is short enough that these time constants (i.e., T_1, T_2, and T_2^*) do not drastically change the image contrast during the readout. As such, during the data acquisition window we can typically model the magnetization in the scanner as being in a static state.

After the spins are tipped into the transverse plane, the signal received in MRI comes from the entire imaging volume. To achieve spatial localization, we apply gradients to locally vary the magnetic field. Most MRI scanners are equipped with secondary magnets,

called gradients, that linearly vary the field in the x, y, and z-directions, i.e.

$$B = B_0 + G_x x + G_y y + G_z z = B_0 + \boldsymbol{G} \cdot \boldsymbol{x}, \tag{2.4}$$

which then gives a location-dependent frequency of precession:

$$\omega = \gamma(B_0 + \boldsymbol{G} \cdot \boldsymbol{x}). \tag{2.5}$$

This location-dependent frequency of precession can then be exploited for imaging. Recalling that the spins in the magnetic field can be thought of as a number of localized oscillators, each oscillator can be thought of as possessing a magnitude, $m(\boldsymbol{x}, t)$, and a phase term, $\phi(\boldsymbol{x}, t)$. If the signal receiver is uniform over the imaging volume, then the received signal, $y(t)$, is the integration of all the oscillators in the imaging volume:

$$y(t) = \int m(\boldsymbol{x}, t) e^{-i\phi(\boldsymbol{x},t)} d\boldsymbol{x}. \tag{2.6}$$

Frequency is equal to the time rate of change in phase, so we can write

$$\frac{d}{dt}\phi(\boldsymbol{x}, t) = \omega(\boldsymbol{x}, t) = \gamma(B_0 + \boldsymbol{G} \cdot \boldsymbol{x}). \tag{2.7}$$

This implies that

$$\begin{aligned} \phi(\boldsymbol{x}, t) &= \int_0^t \omega(\boldsymbol{x}, \tau) d\tau = \gamma \int_0^t B(\boldsymbol{x}, \tau) d\tau \\ &= \omega_0 t + \boldsymbol{k}(t) \cdot \boldsymbol{x}, \end{aligned} \tag{2.8}$$

where

$$\boldsymbol{k}(t) = \begin{bmatrix} 2\pi\gamma \int_0^t G_x(\tau) d\tau \\ 2\pi\gamma \int_0^t G_y(\tau) d\tau \\ 2\pi\gamma \int_0^t G_z(\tau) d\tau \end{bmatrix}. \tag{2.9}$$

We call $\boldsymbol{k}(t)$ the k-space trajectory. Plugging these expressions into (2.6) and demodulating by the constant $\omega_0 t$ term shows that the MRI signal can be written as

$$y(t) = \int m(\boldsymbol{x}) e^{-i\boldsymbol{k}(t)\cdot\boldsymbol{x}} d\boldsymbol{x}. \tag{2.10}$$

(2.10) is a Fourier-transform like operation on the magnetization, $m(\boldsymbol{x})$. It acquires harmonics of $m(\boldsymbol{x})$, where the specific harmonic acquired is determined by $\boldsymbol{k}(t)$. (The time-dependency in $m(\boldsymbol{x})$ has been removed due to the demodulation.)

2.2 MRI System Matrix Models

(2.10) cannot be used on a computer since it is not discretized. For the purpose of discretization, we will assume that samples are acquired at $p \in [1, ..., P]$ time points. We will also approximate $m(\boldsymbol{x})$ as being composed of a train of N evenly-spaced Dirac impulses:

$$m(\boldsymbol{x}) = \sum_{n=1}^{N} x_n \delta(\boldsymbol{x} - \boldsymbol{x}_n). \tag{2.11}$$

Thus, we could encode the magnetization in a vector, $\mathbf{x} \in \mathbb{C}^{N \times 1}$, where the nth element of $\mathbf{x}$ is x_n. Lastly, due to thermal noise both from the induction coil used for detecting the signal and the body of the subject, the received signal is corrupted by complex Gaussian noise, $\boldsymbol{\epsilon}$. Thus, we collect P data points in a vector, $\mathbf{y} \in \mathbb{C}^{P \times 1}$ where

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \tag{2.12}$$

and the pth, nth element of the $\mathbf{A}$ is $e^{-i\boldsymbol{k}(t_p)\cdot\boldsymbol{x}_n}$. When the k-space trajectory, $\boldsymbol{k}(t)$, gives samples on a Cartesian grid, we compute $\mathbf{A}\mathbf{x}$ using a discrete Fourier transform (DFT) implemented as a Fast Fourier Transform (FFT).

2.2.1 Beyond Cartesian Fourier Encoding

The DFT representation of $\mathbf{A}$ is sufficient when the k-space samples lie on a Cartesian grid. However, many k-space sampling patterns use non-Cartesian trajectories that allow greater coverage of k-space within the hardware constraints of MRI. This means the simple FFT can no longer be used as an operator for $\mathbf{A}$. To retain the efficiency of the FFT, non-uniform fast Fourier transform (NUFFT) methods have been developed [14]. These methods apply an FFT to $\mathbf{x}$, yielding frequency samples that lie on an evenly-spaced grid. Then, they interpolate to the off-grid locations specified by $\boldsymbol{k}(t)$.

Another aspect of MRI that can disrupt standard Fourier encoding is off-resonance. Off-resonance occurs when the main magnetic field, B_0, is not uniform throughout the object. This causes the frequency demodulation used in receiving the signal to be non-ideal in certain areas of the object. Since the magnetic field is related to frequency via the Larmor equation in (2.1), we typically think of this off-resonance in terms of frequencies. The frequency map of off-resonance, $\omega(\boldsymbol{x})$, can be measured by acquiring images at different echo times and then applying a complex division. Methods have been developed for doing this in a noise-robust manner [15]. When the signal is received in the presence of field

inhomogeneities, the signal equation becomes

$$y(t) = \int m(\boldsymbol{x}) e^{i\omega(\boldsymbol{x})t} e^{-i\boldsymbol{k}(t)\cdot\boldsymbol{x}} d\boldsymbol{x}. \tag{2.13}$$

The discretization of the system matrix, $\mathbf{A}$, is now more complicated. At first sight, (2.13) would seem to imply that we now need to apply a new Fourier transform for each data point that we acquire since the magnetization has been perturbed by $\omega(\boldsymbol{x})t$ in a different way. Fast methods for approximating this process have been developed [16, 17]. These methods apply a Fourier operator on a small subset of the P time points and then fill in the the other time points via interpolation. The approximation error in these techniques is usually quite low when a sufficient number of terms are used in the approximation.

2.2.2 Parallel Imaging

When only one receive coil is used that has a uniform sensitivity profile, Fourier encoding methods with appropriate NUFFT and field inhomogeneity approximations are all that is necessary to specify $\mathbf{A}$. However, when the response profile of the receive coil is non-uniform or when there are multiple receive coils, further modifications to the signal equation are necessary. A straightforward modification for C receive coils is sensitivity encoding (SENSE) [1], which yields the following signal equation:

$$y_c(t) = \int m(\boldsymbol{x}) s_c(\boldsymbol{x}) e^{-i\boldsymbol{k}(t)\cdot\boldsymbol{x}} d\boldsymbol{x}. \tag{2.14}$$

$y_c(t)$ now specifies the received signal in coil c where $c \in [1, ..., C]$. Based on 2.14, the sampled measurement $\mathbf{y}$ is now a vector of size $PC \times 1$ and $\mathbf{A}$ can be formulated as

$$\mathbf{A} = \begin{bmatrix} \mathbf{F}_1 & & \\ & \ddots & \\ & & \mathbf{F}_1 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_C \end{bmatrix}, \tag{2.15}$$

where each of the $\mathbf{S}_c$ matrices has elements of $s_c(\boldsymbol{x}_n)$ on its diagonal. Each of the $\mathbf{F}_1$ are an identical Fourier encoding operator based on $\boldsymbol{k}(t)$, which is common to all the coils. One difficulty in using sensitivity encoding is calculation of the senstivity maps in $\mathbf{S}$. These maps can change from patient to patient. The standard method is to divide low-resolution images from each sensitivity coil by another image of uniform sensitivity. The uniform-sensitivity image can be obtained from the square-root sum-of-squares of all the coil images or a separate body coil images. Since there is a division, this process can involve noise

amplification. Methods for mitigating this noise amplification have been developed [4].

2.3 MRI Reconstruction

As mentioned previously, the noise in MRI is additive complex Gaussian:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}. \tag{2.16}$$

From the Gauss-Markov Theorem, the best estimate of $\mathbf{x}$ is

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) \\ f(\mathbf{x}) &= \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2 \\ \hat{\mathbf{x}} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{y}. \end{aligned} \tag{2.17}$$

When the noise is low and there is no subsampling, (2.17) provides an excellent estimate of $\mathbf{x}$. When $\mathbf{A}$ is a simple DFT matrix (as is the case in many MRI applications), $\hat{\mathbf{x}}$ is the inverse DFT of $\mathbf{y}$. When $\mathbf{A}$ has full, Cartesian sampling of k-space and multiple coils in a SENSE setting (i.e., the formulation from Section 2.2.2), $\hat{\mathbf{x}}$ is sum of the coil images from each coil with a phase compensation factor and a normalization by the sum-of-squares of the sensitivity coil patterns.

However, in cases with low SNR or high k-space undersampling the inversion of $\mathbf{A}^H\mathbf{A}$ can amplify the noise. In the case of subsampling, $\mathbf{A}^H\mathbf{A}$ might not even be invertible. (This can clearly be shown when SENSE is not used.) Large noise levels may be present in certain types of imaging, such as MR spectroscopy. Subsampling is a topic of significant interest since it allows reductions in MRI scan times, so we will explore alternative estimators of $\mathbf{x}$ for these settings.

A general approach for improving on the estimator in (2.17) is by incorporating prior knowledge through regularization. These methods reduce variance typically at the price of increased bias. This can be done by modifying the optimization problem in (2.17) to

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}}\, \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2 + \beta R(\mathbf{x}), \tag{2.18}$$

where $R(\mathbf{x})$ is a regularizing function that can be chosen based on prior information. A few forms for $R(\mathbf{x})$ are shown in the next sections.

2.3.1 Quadratic Regularization

One way to incorporate regularization is through quadratic regularization, i.e.

$$R(\mathbf{x}) = \|\mathbf{R}\mathbf{x}\|_2^2 , \tag{2.19}$$

where $\mathbf{R}$ is a matrix chosen by the user. One choice is $\mathbf{R} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. A more common choice is where $\mathbf{R}$ is a first-order finite differencing matrix. Both of these approaches have drawbacks. $\mathbf{R} = \mathbf{I}$ undesirably lowers the average value in the signal estimate $\hat{\mathbf{x}}$ below that of the true value. When $\mathbf{R}$ is a finite differencing matrix, the nature of the quadratic penalty causes edges to be blurred out and smeared, also an undesirable property in high-resolution images. Due to these factors, β is often kept to a small value so as to not overregularize the reconstructed image. This limits the ability of quadratic regularization to remove noise or mitigate subsampling effects. Generally, using $\mathbf{R}$ as a finite-differencing matrix is superior to $\mathbf{R} = \mathbf{I}$.

2.3.2 Sparsity-Promoting Regularization

A drawback of quadratic regularization is that it more heavily penalizes large coefficients of $\mathbf{R}\mathbf{x}$ than small ones. This is an unnatural feature for many MRI images when $\mathbf{R}$ is a finite-differencing matrix. We expect the output of a finite-differencing matrix to be mostly near zero in the flat regions of the image and large near edges. As such, a small number of large coefficients of $\mathbf{R}\mathbf{x}$ are expected, which is not well-captured by the quadratic regularizer. Under this assumption, we might consider the ℓ_0 norm over the quadratic norm, i.e.

$$R(\mathbf{x}) = \|\mathbf{R}\mathbf{x}\|_0 . \tag{2.20}$$

However, this form of $R(\mathbf{x})$ is non-convex and makes the problem in (2.18) NP-hard. Non-convex regularizers are often undesirable for imaging since the estimated image can change drastically based on the measurements. Instead, what is more often used is the convex relaxation of the ℓ_0 norm: the ℓ_1 norm:

$$R(\mathbf{x}) = \|\mathbf{R}\mathbf{x}\|_1 . \tag{2.21}$$

Using the ℓ_1 norm leaves the cost function in (2.18) convex while preserving desirable sparsity-promoting properties of the ℓ_0 norm. The ℓ_1 norm penalizes large coefficients far less than the ℓ_2 norm-squared. Use of the ℓ_1 norm in MR image reconstruction is a frequently-studied topic.

There are a number of regularizing matrices used with the sparsity-promoting regularizer. The finite-differencing matrix is one of them, as it was in the quadratic case. Sparsity-promoting regularization with finite-finite differencing matrices is often called *anisotropic total variation* due to its relation as the discrete version of the continuous-form total variation integral. Another class of matrices used with sparsity-promoting regularization includes discrete wavelet transforms. These transforms decompose the image into a large number of localized coefficients that encode the high and low frequency information in various patches of the image. Both orthogonal and overcomplete/undecimated discrete wavelet transforms are used for imaging. Orthogonal transforms are useful since they are mathematically simpler to work with, although overcomplete/undecimated wavelet transforms are typically thought to provide better images.

2.3.3 Dynamic MRI and Low Rank Penalties

In dynamic MRI settings it is more natural to think of $\mathbf{X}$, the object to be reconstructed, as an $N \times T$ matrix where T indicates the number of time points rather than an N-length vector. Thinking of $\mathbf{X}$ as a matrix allows the use of low-rank models, which are a natural way to constrain dynamic image reconstruction. In dynamic MRI, it is natural to think of the image series as being composed of a mostly-static background and a small dynamic portion of the image. Mathematically, this implies that the time-evolution of magnetization can be written as

$$m(\boldsymbol{x}, t) = \sum_{r=1}^{R} u_r(\boldsymbol{x}) v_r(t), \tag{2.22}$$

where R is the rank of the model. Discretizing (2.22) gives

$$\mathbf{X} = \mathbf{U}\mathbf{V}, \tag{2.23}$$

where $\mathbf{U}$ is $N \times R$ and $\mathbf{V}$ is $R \times T$. Typically $R \ll N$ and $R \ll T$, so the number of free parameters with this model is far less than NT. When this model is assumed, it can be encouraged via the nuclear norm penalty, i.e.

$$R(\mathbf{X}) = \|\mathbf{X}\|_* . \tag{2.24}$$

An alternative way of promoting a low-rank model is by estimating $\mathbf{U}$ and $\mathbf{V}$ directly via a two-step process.

2.4 MM Algorithms for MR Image Reconstruction

The choice of regularizer significantly affects the choice of algorithm for obtaining the estimator in (2.18). In the case where quadratic regularizers are used, the preconditioned conjugate gradient algorithm is an excellent choice of algorithm. This algorithm is well-studied and has been used extensively in MRI applications. When sparsity-promoting or low rank regularizers are used, the algorithm must be more carefully considered due to the nondifferentiability of these regularizers.

Here we briefly review a general derivation of a classic algorithm: the majorize-minimize algorithm. This method forms quadratic surrogates for the cost function in (2.18) with desirable mathematical properties. Majorize-minimize algorithms iteratively form surrogates for the cost function and decrease the surrogate. We illustrate the approach here for a simple quadratic. Extensions of this derivation to include ℓ_1 regularizers are a major topic of the rest of this dissertation. We begin by recalling the form of a quadratic optimization problem:

$$f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2 \tag{2.25}$$

If the surrogate at the kth iteration, $\phi_k(\mathbf{x})$, satisfies the conditions,

$$f(\mathbf{x}^{(k)}) = \phi_k(\mathbf{x}^{(k)}), \tag{2.26}$$

$$f(\mathbf{x}) \leq \phi_k(\mathbf{x}), \tag{2.27}$$

then iteratively decreasing $\phi_k(\mathbf{x})$ is guaranteed to decrease $f(\mathbf{x})$. We form such a surrogate by first rewriting $f(\mathbf{x})$ around a current estimate, $\mathbf{x}^{(k)}$, as

$$\begin{aligned} f(\mathbf{x}) = f(\mathbf{x}^{(k)}) &+ \Re\left\{(\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H(\mathbf{x} - \mathbf{x}^{(k)})\right\} \\ &+ \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^H\mathbf{A}^H\mathbf{A}(\mathbf{x} - \mathbf{x}^{(k)}), \end{aligned} \tag{2.28}$$

where $\Re\{\cdot\}$ returns the real part of its argument and $\mathbf{A}^H$ is the Hermitian transpose of $\mathbf{A}$. If we have $\mathbf{A}^H\mathbf{A} \preceq \mathbf{M}_f \in \mathbb{R}^{N\times N}$ for some matrix, $\mathbf{M}_f$ (where $\mathbf{M} \succeq \mathbf{0}$ implies that $\mathbf{M}$ is

positive semidefinite), we can write

$$\begin{aligned} f(\mathbf{x}) \leq \phi_k(\mathbf{x}) &:= f(\mathbf{x}^{(k)}) \\ &\quad + \Re\left\{(\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H(\mathbf{x} - \mathbf{x}^{(k)})\right\} \\ &\quad + \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_{\mathbf{M}_f}^2 \\ &= \psi(\mathbf{x}, \mathbf{x}^{(k)}) + \zeta, \\ \psi(\mathbf{x}, \mathbf{x}^{(k)}) &:= \frac{1}{2}\left\|\mathbf{x} - (\mathbf{x}^{(k)} - \mathbf{M}_f^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))\right\|_{\mathbf{M}_f}^2, \end{aligned} \tag{2.29}$$

where ζ is a constant that arises from completing the square and is independent of $\mathbf{x}$. Decreasing $\psi(\mathbf{x}, \mathbf{x}^{(k)})$ causes $\phi_k(\mathbf{x})$ to decrease by the same amount. Standard majorize-minimize procedures use $\mathbf{M}_f = L\mathbf{I}$, where L is the maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$. When an MM method is used on a convex cost function, then Nesterov momentum can be used for acceleration [18]. These methods can be further accelerated with adaptive momentum restarting [9, 8].

The key step in designing a fast MM method is the calculation of $\mathbf{M}_f$. There are two desirable traits for $\mathbf{M}_f$. The first is that $\mathbf{M}_f^{-1}$ must have an efficient implementation since it must be applied at every iteration. The second is that $\mathbf{M}_f$ be as small as possible such that $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$. This makes the MM bound tighter and accelerates convergence. Methods for designing $\mathbf{M}_f$ for MRI are a key contribution in this dissertation.

CHAPTER 3

Fast Parallel MR Image Reconstruction Based on Sensitivity Maps

3.1 Introduction

This chapter is based on work published in [8, 19]. Magnetic resonance imaging (MRI) is an imaging modality where improving the resolution requires increased acquisition time. As a result, the cost of MRI also increases with higher resolution since the cost is directly proportional to the scan time. In addition to reducing the cost of high-resolution MRI, scanning time reductions can also help accommodate pediatric and elderly patients that have difficulty remaining motionless during long scans. Such scan time reductions are facilitated by combining undersampling and advanced signal processing methods to remove the associated aliasing artifacts. SENSitivity Encoding (SENSE) is an MRI technique that undoes aliasing effects caused by undersampling by exploiting variations in the sensitivity profiles (i.e., B1 maps) of multiple coils placed around the patient [1]. When the image can reasonably be assumed to be sparse in some transform domain, compressed sensing techniques can be applied to facilitate further accelerations [2].

Image estimation that leverages SENSE MRI and compressed sensing assumptions can be mathematically formulated as an ℓ_1-regularized optimization problem [20]. Since the ℓ_1 term is non-differentiable, these problems are difficult to minimize using standard gradient-based methods. Some methods convert such problems into a different form where fast minimization techniques can be applied. One such class are variable splitting algorithms, where one forms a constrained optimization problem and then proceeds within the augmented Lagrangian formalism to find the solution to the original ℓ_1-regularized problem [6, 21, 22, 7]. A difficulty with applying these methods is the tuning of a constraint penalty parameter that heavily affects convergence speed. Sufficient conditions for optimally choosing these parameters are unknown, so current practice is to resort to heuristics for setting these param-

eters [7]. Since the optimal parameter can change from problem to problem (i.e., patient to patient), robust performance of these methods can be difficult to ascertain.

Alternatives to variable splitting methods are majorize-minimize methods such as fast iterative soft thresholding (FISTA) [18]. FISTA methods converge at a rate that depends on the *Lipschitz constant*, a constant that upper bounds the eigenvalues of the Hessian of the data fit term. This constant is on the order of the maximum of the sum of squared absolute values of the sensitivity coils, which means that this constant can be very loose for low signal regions that occur at the center of the object in SENSE MRI. As a result, majorize-minimize methods such as FISTA have performed poorly relative to their variable splitting counterparts in MRI applications [7].

We address the looseness of the Lipschitz bound by formulating tighter bounds that vary spatially based on the sensitivity coil profiles. The approach requires finding a diagonal majorizer in the range of the regularizing matrix. In this paper we show that for several regularizers of interest (including orthogonal wavelets, anisotropic total variation, and undecimated Haar wavelets), such diagonal upper bounds are simple to compute and give large accelerations relative to FISTA with the Lipschitz constant. When combined with adaptive momentum restarting [9], these methods outperform variable splitting methods in all of these cases. The proposed methods also use parameters in the form of convergence tolerances, but in numerical experiments we found that once a reasonable choice was made for these parameters, further optimization was not necessary.

3.2 Problem Formulation and General Approach

From compressed sensing theory, one can recover a sparse signal by minimizing a convex cost function with ℓ_1 regularization [20]. Let C denote the number of sensitivity coils, D denote the number of data points, and N denote the number of pixels to be estimated. The ℓ_1 minimization procedure for parallel MR image reconstruction can be mathematically formulated as

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}\in\mathcal{M}}{\operatorname{argmin}} \left\{ f(\mathbf{x}) + \beta R(\mathbf{x}) \right\}, \\ f(\mathbf{x}) = \frac{1}{2} \left\| \mathbf{y} - \mathbf{A}\mathbf{x} \right\|_2^2, \; R(\mathbf{x}) &= \left\| \mathbf{R}\mathbf{x} \right\|_1, \; \mathbf{A} = \mathbf{F}\mathbf{S}, \end{aligned} \tag{3.1}$$

where $\mathbf{F} \in \mathbb{C}^{D\times CN}$ is a block diagonal matrix with each block having the same down-sampled DFT operator and $\mathbf{S} \in \mathbb{C}^{CN\times N}$ is a block column matrix with diagonal blocks. We include a masking set, $\mathcal{M}$, that constrains elements outside the mask to be zero. We call $f(\mathbf{x})$ the data fit term and $R(\mathbf{x})$ the regularizer. Weighted quadratic data fit terms used for noise correlations between coils can be converted to this unweighted form [7].

The parameter β must be selected by the user to balance trade-offs between the data fit term and the regularizer (Monte Carlo techniques have been developed for estimating these parameters that perform well under mean-squared error metrics [23]). Defining $\mathbf{A} = \mathbf{FS}$, we note that $\mathbf{S}$ gives $\mathbf{A}$ a highly shift-variant nature, a property that we will consider in our algorithm design. $\mathbf{R}$ is a sparsifying transform. If $\mathbf{R}$ is left-invertible (i.e. $\mathbf{NR} = \mathbf{I}$ for some matrix, $\mathbf{N}$), we say that (3.1) is a *synthesis* reconstruction problem since we can define $\mathbf{u} = \mathbf{Rx}$ and rewrite (3.1) as an optimization problem over $\mathbf{u}$. We assume that $\mathbf{R} \in \mathbb{C}^{M \times N}$. If $\mathbf{R}$ is not left-invertible, then we call (3.1) an *analysis* reconstruction problem and assume that $\mathbf{R} \in \mathbb{R}^{M \times N}$. This restriction of $\mathbf{R}$ to be real-valued includes important classes of analysis regularizers such as total variation [2] and undecimated Haar wavelets [7]. Each of these regularization forms necessitates different algorithm considerations.

Although solving (3.1) allows one to obtain high-quality estimates of $\mathbf{x}$ with less data, (3.1) is typically difficult to minimize. Most methods instead minimize a different problem related to (3.1). The related problem should be easy to minimize relative to (3.1), but still offer information relevant to the solution of (3.1). Two procedures for defining and minimizing related problems are *majorize-minimize* procedures and *variable splitting* procedures. For completeness we note that "corner rounding" has also been proposed for dealing with the non-differentiability of the ℓ_1 regularizer [2], but this has been found to yield algorithms slower than those of the variable splitting class [7]. Our method is of the majorize-minimize class, but is different from previous majorize-minimize methods in that it carefully considers any coupling of the structures of $\mathbf{A}$ and $\mathbf{R}$. We outline the general approach in the following section.

3.2.1 Separable quadratic surrogates

Majorize-minimize methods work by forming a surrogate cost function (i.e., a majorizer, $\phi_k(\mathbf{x})$) and then minimizing the surrogate each iteration to find the minimizer of the original cost function. Any quadratic of the form $\frac{1}{2}\|\mathbf{y} - \mathbf{Ax}\|_2^2$ can be majorized with a *separable quadratic surrogate* (SQS), a procedure that we briefly review [24, 25]. If a surrogate, $\phi_k(\mathbf{x})$, satisfies the following two conditions, then decreasing the surrogate will decrease the original cost function [26]:

$$f(\mathbf{x}^{(k)}) = \phi_k(\mathbf{x}^{(k)}), \tag{3.2}$$

$$f(\mathbf{x}) \leq \phi_k(\mathbf{x}). \tag{3.3}$$

We allow the surrogate to be indexed by k since it can vary with iteration. We form such a surrogate for SENSE MRI by first rewriting $f(\mathbf{x})$ around a current estimate, $\mathbf{x}^{(k)}$, as

$$\begin{aligned} f(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \Re\left\{(\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H(\mathbf{x} - \mathbf{x}^{(k)})\right\} \\ + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})^H\mathbf{A}^H\mathbf{A}(\mathbf{x} - \mathbf{x}^{(k)}), \end{aligned} \tag{3.4}$$

where $\Re\{\cdot\}$ returns the real part of its argument and $\mathbf{A}^H$ is the Hermitian transpose of $\mathbf{A}$. If we have $\mathbf{A}^H\mathbf{A} \preceq \mathbf{D}_f \in \mathbb{R}^{N\times N}$ for some diagonal matrix, $\mathbf{D}_f$ (where $\mathbf{M} \succeq \mathbf{0}$ implies that $\mathbf{M}$ is positive semidefinite), we can write

$$\begin{aligned} f(\mathbf{x}) \leq \phi_k(\mathbf{x}) &= f(\mathbf{x}^{(k)}) \\ &\quad + \Re\left\{(\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))^H(\mathbf{x} - \mathbf{x}^{(k)})\right\} \\ &\quad + \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_{\mathbf{D}_f}^2 \\ &= \psi(\mathbf{x}, \mathbf{x}^{(k)}) + \zeta, \\ \psi(\mathbf{x}, \mathbf{x}^{(k)}) &:= \frac{1}{2}\left\|\mathbf{x} - (\mathbf{x}^{(k)} - \mathbf{D}_f^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))\right\|_{\mathbf{D}_f}^2, \end{aligned} \tag{3.5}$$

where ζ is a constant that arises from completing the square and is independent of $\mathbf{x}$. Decreasing $\psi(\mathbf{x}, \mathbf{x}^{(k)})$ causes $\phi_k(\mathbf{x})$ to decrease by the same amount. Standard majorize-minimize procedures use $\mathbf{D}_f = L\mathbf{I}$, where L is the maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$. We instead use a more general $\mathbf{D}_f$ that is a tighter bound for $\mathbf{A}^H\mathbf{A}$. In the case where $\mathbf{A} = \mathbf{F}\mathbf{S}$, we have $\mathbf{A}^H\mathbf{A} = \mathbf{S}^H\mathbf{F}^H\mathbf{F}\mathbf{S}$. In general $\mathbf{F}^H\mathbf{F} \preceq F\mathbf{I}$, where F is the maximum eigenvalue of $\mathbf{F}^H\mathbf{F}$. In the case of Cartesian MRI with unitary DFT matrices, $F = 1$. One can estimate F offline in the non-Cartesian case via power iteration since it does not depend on the object. Noting this we have

$$\mathbf{D}_f := F\mathbf{S}^H\mathbf{S} \succeq \mathbf{A}^H\mathbf{A}, \tag{3.6}$$

where $\mathbf{S}^H\mathbf{S}$ is a diagonal matrix with the sum of the squared absolute values of the sensitivity coils along its diagonal. We could use $\mathbf{D}_f$ to upper bound any SENSE-type quadratic data fit term with a separable quadratic surrogate. We will use this property in the following sections. Furthermore, $\mathbf{D}_f$ is easy to compute once one has determined the coil sensitivities, and with the recent development of fast algorithms for SENSE map estimation it is quickly available in online settings [4].

3.2.2 Proposed minimization algorithm

We note through the majorization conditions that solving the following problem will decrease the cost function in (3.1):

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}\in\mathcal{M}}{\operatorname{argmin}} \left\{\eta(\mathbf{x}, \mathbf{x}^{(k)}) := \psi(\mathbf{x}, \mathbf{x}^{(k)}) + \beta R(\mathbf{x})\right\}. \tag{3.7}$$

The minimization problem in (3.7) is where synthesis and analysis regularizers differ. In the synthesis case, $\eta(\mathbf{x}, \mathbf{x}^{(k)})$ is either fully separable or it can be converted into a fully separable form. When $\eta(\mathbf{x}, \mathbf{x}^{(k)})$ is fully separable, closed-form solutions exist via shrinkage functions. In the analysis case, closed form solutions do not exist and we have to run a few steps of an iterative algorithm to decrease $\eta(\mathbf{x}, \mathbf{x}^{(k)})$. We discuss the synthesis case in detail in Section 3.3 and the analysis case in Section 3.4.

Iteratively applying (3.7) qualifies as a majorize-minimize procedure, and as such it can be accelerated with momentum techniques [18]. Momentum accelerations can be enhanced with adaptive momentum restarting [9]. This gives a general algorithm, which we call B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithm, or BARISTA, since it has step sizes that depend on the sensitivity or B1 maps. Fig. 3.1 shows the overall algorithm. Variants of this general form are shown in Fig. 3.2 for the synthesis case and Fig. 3.3 for

Figure 3.1: BARISTA: B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithm

initialize $k = 0, \mathbf{z}^{(0)} = \mathbf{x}^{(0)}, \mathbf{D}_f, \alpha$
while $k < K$ **do**
  $\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$
  $\mathbf{x}^{(k+1)} = \underset{\mathbf{x}\in\mathcal{M}}{\operatorname{argmin}}\, \eta(\mathbf{x}, \mathbf{z}^{(k)})$
  $\kappa = \left\|\mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}\right\|_2 \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2$
  if $\Re\left\{\left\langle \mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\rangle\right\} > \alpha\kappa$ **then**
    $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)}$
    $\tau^{(k+1)} = 1$
  else
    $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$
  end if
  $k = k + 1$
end while
$\hat{\mathbf{x}} = \mathbf{x}^{(K)}$

the analysis case. The tracking of the momentum is provided by the $\tau^{(k)}$ parameter and an auxiliary variable, $\mathbf{z}^{(k)}$. If the algorithm takes a step in a certain direction, then $\mathbf{z}^{(k+1)}$ takes a larger step in the same direction where the size is determined by $\tau^{(k)}$. The restart

is shown with the "if" statement at the end of an algorithm step. If the cosine of the angle between $\mathbf{x}^{(k+1)} - \mathbf{z}^{(k)}$ and $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is greater than α, then the momentum is wiped away. This helps prevent the generalized gradient and the momentum term from taking the algorithm in opposite directions. In our numerical experiments we found that good values for α are negative and lie near 0; we used $\alpha = -\cos(4\pi/9)$. As stated previously, one challenge is in the minimization of $\eta(\mathbf{x}, \mathbf{z}^{(k)})$. Another associated challenge is the design of matrices similar to $\mathbf{D}_f$, but in the range of the regularizer for both synthesis and analysis regularization. The following sections discuss these topics.

3.3 Synthesis Regularization

We use the term *synthesis regularization* when $\mathbf{R}$ is left-invertible, which allows rewriting the minimization problem in the basis of the regularizer. For notational simplicity in this section we discuss $\mathbf{R}$ that forms a *tight frame*, i.e. $\mathbf{R}^H\mathbf{R} = \mathbf{I}$. Orthogonal wavelet transforms for SENSE MRI and DFT/DCT regularizers for dynamic MRI are examples of unitary matrices that might be used in synthesis problems in MRI. Defining $\mathbf{u} = \mathbf{R}\mathbf{x}$, we can rewrite (3.1) as

$$\begin{aligned} \hat{\mathbf{u}} &= \underset{\mathbf{u}\in\mathcal{M}_{\text{synth}}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{A}\mathbf{R}^H\mathbf{u} \right\|_2^2 + \beta \left\| \mathbf{u} \right\|_1 \right\}, \\ \hat{\mathbf{x}} &= \mathbf{R}^H\hat{\mathbf{u}} \end{aligned} \tag{3.8}$$

where $\mathcal{M}_{\text{synth}}$ is a synthesis mask that restricts a subset of the synthesis coefficients to be zero. It is less natural to use a mask for the synthesis approach than for analysis, so if masking is desired we recommend using the algorithm outlined in Section 3.4. Now if we find a diagonal matrix, $\mathbf{D}_R$, such that

$$\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H \tag{3.9}$$

and defining $\mathbf{B} = \mathbf{A}\mathbf{R}^H$, the surrogate in (3.7) is

$$\begin{aligned} &\eta(\mathbf{u}, \mathbf{u}^{(k)}) \\ &= \frac{1}{2} \left\| \mathbf{u} - (\mathbf{u}^{(k)} - \mathbf{D}_R^{-1}\mathbf{B}^H(\mathbf{B}\mathbf{u}^{(k)} - \mathbf{y})) \right\|_{\mathbf{D}_R}^2 + \beta \left\| \mathbf{u} \right\|_1, \end{aligned} \tag{3.10}$$

the constrained minimum of which is a shrinkage solution:

$$\begin{aligned}\mathbf{u}^{(k+1)} &= \underset{\mathbf{u}\in\mathcal{M}_{\text{synth}}}{\text{argmin}}\ \eta(\mathbf{u},\mathbf{u}^{(k)}) \\ &= P_{\mathcal{M}_{\text{synth}}}(\text{shrink}(\mathbf{b}^{(k)},\beta\mathbf{d}_R^{-1})),\end{aligned} \tag{3.11}$$

where $\mathbf{b}^{(k)} = \mathbf{u}^{(k)} - \mathbf{D}_R^{-1}\mathbf{B}^H(\mathbf{B}\mathbf{u}^{(k)} - \mathbf{y})$. $\mathbf{d}_R^{-1}$ is a vector composed of the diagonal elements of $\mathbf{D}_R^{-1}$ and the shrinkage function is defined as $\text{shrink}(\mathbf{b},\beta) = \text{diag}\{\text{sign}(b_i)\}(|\mathbf{b}| - \beta\mathbf{1})_+$. In this case $|\cdot|$ denotes the absolute value function, $(\cdot)_+$ sets values less than zero to zero, $\text{diag}\{\cdot\}$ takes the input elements and arranges them as a diagonal matrix, and $\text{sign}(\cdot)$ returns the complex sign of its argument. The $P_{\mathcal{M}_{\text{synth}}}$ operator projects its argument on to the set, $\mathcal{M}_{\text{synth}}$, which in this case corresponds to setting all elements outside the mask to zero. Fig. 3.2 shows the synthesis version of BARISTA. Our goal now is to design $\mathbf{D}_R$,

Figure 3.2: BARISTA for synthesis

1: initialize $k = 0, \mathbf{u}^{(0)} = \mathbf{R}\mathbf{x}^{(0)}, \mathbf{z}^{(0)} = \mathbf{u}^{(0)}, \mathbf{D}_f, \alpha$
2: calculate $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$ according to Theorem 1
3: **while** $k < K$ **do**
4: $\quad\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$
5: $\quad\mathbf{b}^{(k)} = \mathbf{z}^{(k)} - \mathbf{D}_R^{-1}\mathbf{B}^H(\mathbf{B}\mathbf{z}^{(k)} - \mathbf{y})$
6: $\quad\mathbf{u}^{(k+1)} = P_{\mathcal{M}_{\text{synth}}}(\text{shrink}(\mathbf{b}^{(k)},\beta\mathbf{d}_R^{-1}))$
7: $\quad\kappa = \left\|\mathbf{z}^{(k)} - \mathbf{u}^{(k+1)}\right\|_2 \left\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\right\|_2$
8: $\quad$**if** $\Re\left\{\left\langle \mathbf{z}^{(k)} - \mathbf{u}^{(k+1)}, \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\right\rangle\right\} > \alpha\kappa$ **then**
9: $\quad\quad\mathbf{z}^{(k+1)} = \mathbf{u}^{(k+1)}$
10: $\quad\quad\tau^{(k+1)} = 1$
11: $\quad$**else**
12: $\quad\quad\mathbf{z}^{(k+1)} = \mathbf{u}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)})$
13: $\quad$**end if**
14: $\quad k = k + 1$
15: **end while**
16: $\hat{\mathbf{x}} = \mathbf{R}^H\mathbf{u}^{(K)}$

which will allow us to take larger step sizes in step 5 of Fig. 3.2 and apply more aggressive shrinkage in step 6.

3.3.1 Diagonal upper bounds in unitary bases

The challenge in designing $\mathbf{D}_R$ is that it must be constructed in the basis of the regularizer, while $\mathbf{D}_f$ is in the basis of the image. For this purpose we will use Theorem 1, which gives a means of constructing $\mathbf{D}_R$. Theorem 1 can be applied for any $\mathbf{R}$, although it is most

useful for unitary regularizing matrices with compact support. We will use it here since orthogonal wavelets fall into this class.

Theorem 1. *Let $\mathbf{R} \in \mathbb{C}^{M\times N}$ be any matrix and let $\mathbf{D}_f \in \mathbb{R}^{N\times N}$ be diagonal with diagonal elements $d_{n,f}$. Let $\mathbf{r}_m$ be the mth row of $\mathbf{R}$ and let $\mathcal{S}_m \subset \{1, ..., N\}$ be the support set for $\mathbf{r}_m$. Define $t_m = max_{n\in\mathcal{S}_m}(d_{n,f})$ and $d_{m,R} = \sum_{l=1}^{M} min(t_m, t_l)|\langle \mathbf{r}_m, \mathbf{r}_l\rangle|$. Let $\mathbf{D}_R$ be a diagonal matrix with diagonal elements $d_{m,R}$, then $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$.*

To prove Theorem 1 we first define $\mathbf{R}^H = [\mathbf{r}_1, ..., \mathbf{r}_M]$, where $\mathbf{r}_m$ denotes the mth column of $\mathbf{R}^H$. We then recognize that the inner product of two compactly supported vectors can be computed over either vector's support, i.e.

$$\langle \mathbf{r}_m, \mathbf{r}_l\rangle = \sum_{n\in\mathcal{S}_m} r_{m,n}r^*_{l,n} = \sum_{n\in\mathcal{S}_l} r_{m,n}r^*_{l,n}. \tag{3.12}$$

The entries in $\mathbf{V} := \mathbf{R}\mathbf{D}_f\mathbf{R}^H$ are weighted inner products of the form

$$v_{m,l} = \langle \mathbf{r}_m, \mathbf{r}_l\rangle_{\mathbf{D}_f}, \tag{3.13}$$

where $v_{m,l}$ is the mth, lth entry of $\mathbf{V}$. We recall that if $\mathbf{D}_1 \succeq \mathbf{D}_f$, then $\mathbf{R}\mathbf{D}_1\mathbf{R}^H \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$. One such $\mathbf{D}_1$ is a diagonal matrix where the diagonal entries are defined as

$$d_{1,n} = \begin{cases} t_1, & \text{if } n \in \mathcal{S}_1 \\ d_{n,f}, & \text{otherwise} \end{cases} \tag{3.14}$$

We also note that finding a $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f\mathbf{R}^H$ is equivalent to finding a $\mathbf{D}_R$ such that $\mathbf{w}^H\mathbf{D}_R\mathbf{w} \geq \mathbf{w}^H\mathbf{R}\mathbf{D}_f\mathbf{R}^H\mathbf{w}$ for any vector $\mathbf{w}$. To accomplish this, we make the partition $\mathbf{R}^H = [\mathbf{r}_1, \mathbf{R}_1^H]$ where $\mathbf{R}_1^H = [\mathbf{r}_2, ..., \mathbf{r}_M]$. We also make the partition $\mathbf{w} = [w_1, \mathbf{w}_2]$. We now have

$$\begin{aligned} \mathbf{w}^H\mathbf{R}\mathbf{D}_1\mathbf{R}^H\mathbf{w} &= t_1|w_1|^2\|\mathbf{r}_1\|_2^2 \\ &\quad + t_1\sum_{m=2}^{M} 2\Re\{w_1w_m^*\langle \mathbf{r}_1, \mathbf{r}_m\rangle\} + \mathbf{w}_2^H\mathbf{R}_1\mathbf{D}_1\mathbf{R}_1^H\mathbf{w}_2 \\ &\leq t_1|w_1|^2\|\mathbf{r}_1\|_2^2 \\ &\quad + t_1\sum_{m=2}^{M}(|w_1|^2 + |w_m|^2)|\langle \mathbf{r}_1, \mathbf{r}_m\rangle| + \mathbf{w}_2^H\mathbf{R}_1\mathbf{D}_1\mathbf{R}_1^H\mathbf{w}_2, \end{aligned} \tag{3.15}$$

which comes from applying (3.12) and (3.13). This implies that

$$\mathbf{R}\mathbf{D}_f\mathbf{R}^H \preceq \begin{bmatrix} t_1 \sum_{m=1}^{M} |\langle \mathbf{r}_1, \mathbf{r}_m \rangle| & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1\mathbf{D}_1\mathbf{R}_1^H \end{bmatrix} + \begin{bmatrix} 0 & & & \\ & t_1|\langle \mathbf{r}_1, \mathbf{r}_2 \rangle| & & \\ & & \ddots & \\ & & & t_1|\langle \mathbf{r}_1, \mathbf{r}_M \rangle| \end{bmatrix}. \tag{3.16}$$

Without loss of generality, we can assume that $t_1 \leq t_2 \leq ... \leq t_M$. If this is not satisfied, then the appropriate permutation can be applied to $\mathbf{V}$ to make it so. The procedure can again be applied to $\mathbf{R}_1\mathbf{D}_1\mathbf{R}_1^H$, and then again recursively. Applying this procedure recursively through M gives Theorem 1.

Theorem 1 states that any matrix of the form $\mathbf{R}\mathbf{D}_f\mathbf{R}^H$ can be upper bounded with a diagonal matrix by taking maximums over patches of $\mathbf{D}_f$ and scaling those maximums by sums of inner products. These inner product sums increase as $\mathbf{R}$ becomes less unitary, but in our synthesis case we assume unitary $\mathbf{R}$ so $d_{m,R} = t_m$. We have found that this is an effective majorizing matrix for unitary regularizing matrices, and we used Theorem 1 to design $\mathbf{D}_R$ for orthogonal Haar and Daubechies D4 wavelets in our numerical experiments where we ran the algorithm in Fig. 3.2.

3.4 Analysis Regularization

In the analysis setting $\mathbf{R}$ is not left-invertible and we can no longer define $\mathbf{u} = \mathbf{R}\mathbf{x}$ and rewrite (3.1) as an optimization problem over $\mathbf{u}$. As such, we must leave (3.1) in its original form. The forms of $\mathbf{R}$ of this type that are of interest for SENSE MRI include anisotropic total variation regularizers [2] and undecimated wavelets where the approximation coefficients are unregularized [27, 7]. However, we can still form a quadratic surrogate for the data fit term. This gives the *analysis denoising problem*. Since we do not have a closed-form expression for the solution of this problem, we run a few iterations of a denoising procedure. Fig. 3.3 shows the overall analysis algorithm, while the denoising procedure is shown in Fig. 3.4.

We must decide on a stopping criterion for the iterative algorithm used for the denoising step. Previous methods have used a fixed iteration count for this step [7, 28], but we instead use an $\epsilon^{(k)}$ criterion. When large steps are being taken in the outer iterations, the denoising step only needs to provide an approximate solution to progress the algorithm, whereas very

Figure 3.3: BARISTA for analysis

initialize $k = 0, \mathbf{z}^{(0)} = \mathbf{x}^{(0)}, \mathbf{D}_f, \alpha, \epsilon^{(0)}$
calculate $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$ as outlined in Section 3.4.2
while $k < K$ **do**
    $\tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$
    Run Fig. 3.4 algorithm to $\epsilon^{(k)}$ convergence to get $\mathbf{x}^{(k+1)}$
    $\kappa = \left\|\mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}\right\|_2 \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2$
    if $\Re\left\{\left\langle \mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\rangle\right\} > \alpha\kappa$ **then**
        $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)}$
        $\tau^{(k+1)} = 1$
    else
        $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$
    end if
    $\epsilon^{(k+1)} = \max(\min(\epsilon_{\text{diff}} \frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{x}^{(k)}\right\|_2}, \epsilon^{(k)}), \epsilon_{\text{min}})$
    $k = k + 1$
end while
$\hat{\mathbf{x}} = \mathbf{x}^{(K)}$

accurate solutions are beneficial for later iterations where the outer steps are small.

Fig. 3.3 shows a strategy for choosing $\epsilon^{(k)}$ that was effective in our numerical experiments. $\epsilon^{(k)}$ is chosen to be ϵ_{diff} times the norm difference of the previous iteration, restricted between the upper and lower bounds of $\epsilon^{(k-1)}$ and ϵ_{min}. We choose ϵ_{diff} to balance the cost of solving the denoising problem and progressing the outer iterations. In all experiments we used $\epsilon_{\text{diff}} = 10^{-1}$. We choose ϵ_{min} based on the precision level of the machine that runs the algorithm; its primary purpose is to prevent the algorithm from stalling as a result of numerical precision. For double precision, we observed that $\epsilon_{\text{min}} = 10^{-12}$ gave agreeable convergence in later iterations. We set $\epsilon^{(0)} = 10^{-1}$. We decrease $\epsilon^{(k)}$ monotonically, so that the denoising subproblem is solved more accurately as the algorithm progresses toward the solution.

3.4.1 Analysis denoising subroutine

We now discuss the so-called *analysis denoising problem* that needs to be solved in step 6 of the algorithm in Fig. 3.3, which is formulated as follows:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \underset{\mathbf{x}\in\mathcal{M}}{\operatorname{argmin}}\, \eta(\mathbf{x}, \mathbf{x}^{(k)}), \\ \eta(\mathbf{x}, \mathbf{x}^{(k)}) &= \frac{1}{2}\left\|\mathbf{b}^{(k)} - \mathbf{x}\right\|_{\mathbf{D}_f}^2 + \beta \left\|\mathbf{R}\mathbf{x}\right\|_1, \\ \mathbf{b}^{(k)} &= \mathbf{x}^{(k)} - \mathbf{D}_f^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}). \end{aligned} \tag{3.17}$$

This is equivalent to solving step 4 of the algorithm in Fig. 3.1. There are many potential approaches to solving this step, including nonlinear CG [2] and split Bregman schemes [6]. As mentioned in the previous section, our goal is to minimize the cost function in (3.17) to some pre-specified numerical precision. As a result, whatever procedure is chosen should perform well under all numerical precision environments, a property not satisfied by nonlinear CG due to its corner-rounding parameter or split Bregman due to its constraint penalty parameter. Instead, we choose to extend the results in [28] to general ℓ_1 regularizers, adapt it to complex numbers, and reintroduce our diagonal majorizing matrix in the range of the regularizer. This approach meets the numerical precision requirements and gave agreeable convergence speed in numerical experiments. Our derivation requires real-valued $\mathbf{R}$, which includes interesting classes of anisotropic total variation and undecimated Haar wavelet regularizers. Fig. 3.4 shows the algorithm that arises from extending the results in [28].

For the extension of the results in [28], we assume $\mathbf{R} \in \mathbb{R}^{M\times N}$, which includes the classes of total variation and undecimated wavelet regularizers that are of interest to us. The difficulty in minimizing (3.17) is the fact that $\mathbf{R}$ mixes different elements of $\mathbf{x}$. To decouple the mixing effects, we will introduce dual variables. Let $\gamma \in \mathbb{R}$ and $\nu \in \mathbb{R}$ be two variables and define $\mathcal{P} = \{(\gamma, \nu) \in \mathbb{R}^2 : \gamma^2 + \nu^2 \leq 1\}$. Then for any $c \in \mathbb{C}$ we have

$$|c| = \max_{(\gamma,\nu)\in\mathcal{P}} \left\{\gamma \Re\{c\} + \nu \Im\{c\}\right\}, \tag{3.18}$$

where $\Im\{\cdot\}$ returns the imaginary part of its argument. Noting this, we now have

$$\left\|\mathbf{R}\mathbf{x}\right\|_1 = \max_{(\boldsymbol{\gamma},\boldsymbol{\nu})\in\mathcal{P}^M} \left\{\boldsymbol{\gamma}^T\mathbf{R}\Re\{\mathbf{x}\} + \boldsymbol{\nu}^T\mathbf{R}\Im\{\mathbf{x}\}\right\}, \tag{3.19}$$

where $\mathcal{P}^M$ is a Cartesian product of M sets of the form of $\mathcal{P}$. Note that $\mathcal{P}^M$ is the ℓ_∞ unit

ball in $\mathbb{C}^M$. We also have

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \underset{\mathbf{x}\in\mathcal{M}}{\operatorname{argmin}} \max_{(\boldsymbol{\gamma},\boldsymbol{\nu})\in\mathcal{P}^M} \theta_k((\boldsymbol{\gamma},\boldsymbol{\nu}),\mathbf{x}),\\ \theta_k((\boldsymbol{\gamma},\boldsymbol{\nu}),\mathbf{x}) &\\ &= \frac{1}{2}\left\|\mathbf{b}^{(k)}-\mathbf{x}\right\|_{\mathbf{D}_f}^2 + \beta(\boldsymbol{\gamma}^T\mathbf{R}\Re\{\mathbf{x}\}+\boldsymbol{\nu}^T\mathbf{R}\Im\{\mathbf{x}\}).\end{aligned} \tag{3.20}$$

To simplify notation, we will now drop the "arg" and implicitly take $\mathbf{x}$ from wherever the critical point of the cost function is. Since $\mathcal{M}$ is a convex set, $\mathcal{P}^M$ is a compact, convex set, and (3.20) is convex in $\mathbf{x}$ and concave in ($\boldsymbol{\gamma}$, $\boldsymbol{\nu}$), we apply Sion's Theorem [29] to exchange the order of maximization and minimization, which gives

$$\max_{(\boldsymbol{\gamma},\boldsymbol{\nu})\in\mathcal{P}^M} \min_{\mathbf{x}\in\mathcal{M}} \theta_k((\boldsymbol{\gamma},\boldsymbol{\nu}),\mathbf{x}). \tag{3.21}$$

Now we use the fact that since $\mathbf{D}_f \in \mathbb{R}^{N\times N}$ and is diagonal, the weighted 2-norm squared is separable into its real and imaginary parts, i.e.

$$\begin{aligned}&\left\|\mathbf{b}^{(k)}-\mathbf{x}\right\|_{\mathbf{D}_f}^2\\ &\quad= \left\|\Re\{\mathbf{b}^{(k)}\}-\Re\{\mathbf{x}\}\right\|_{\mathbf{D}_f}^2 + \left\|\Im\{\mathbf{b}^{(k)}\}-\Im\{\mathbf{x}\}\right\|_{\mathbf{D}_f}^2.\end{aligned} \tag{3.22}$$

Defining $\mathbf{q}=\boldsymbol{\gamma}+i\boldsymbol{\nu}$, the inner minimization in (3.21) has a solution where $\mathbf{x}=P_{\mathcal{M}}(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})$. As stated previously, the $P_{\mathcal{M}}(\cdot)$ operator simply sets elements outside the mask to zero. Plugging this back into (3.21) reveals a new maximization problem:

$$\begin{aligned}&\max_{\mathbf{q}\in\mathcal{P}^M}\left\{-\frac{1}{2}\left\|\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q}\right\|_{\mathbf{D}_f}^2\right.\\ &\left.+\frac{1}{2}\left\|P_{\mathcal{M}}(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})-(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})\right\|_{\mathbf{D}_f}^2\right\}\\ &=\min_{\mathbf{q}\in\mathcal{P}^M}\left\{\frac{1}{2}\left\|\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q}\right\|_{\mathbf{D}_f}^2\right.\\ &\left.-\frac{1}{2}\left\|P_{\mathcal{M}}(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})-(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})\right\|_{\mathbf{D}_f}^2\right\}\\ &=\min_{\mathbf{q}\in\mathcal{P}^M}\left\{\frac{1}{2}\left\|P_{\mathcal{M}}(\mathbf{b}^{(k)}-\beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q})\right\|_{\mathbf{D}_f}^2\right\}.\end{aligned} \tag{3.23}$$

Since the target cost function is now a constrained minimization over a quadratic, we can once again apply the separable quadratic surrogates techniques outlined in Section 3.2. We choose to do this instead of developing other quadratic minimization routines due to the presence of the constraint. Applying this procedure gives the minimization problem over a

surrogate:

$$\begin{aligned}\mathbf{q}^{(j+1)} =& \underset{\mathbf{q}\in\mathcal{P}^M}{\text{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{q} - (\mathbf{q}^{(j)} - \beta^{-1}\mathbf{D}_R^{-1}\mathbf{R}\mathbf{x}^{(k,j+1)}) \right\|_{\mathbf{D}_R}^2 \right\} \\ \mathbf{x}^{(k,j+1)} :=& P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q}^{(j)})\end{aligned} \tag{3.24}$$

for $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. This is obtained by recognizing that the $P_{\mathcal{M}}(\cdot)$ operator where $\mathcal{M}$ is a masking set can be formulated as a projection matrix, $\mathbf{M}$, where $P_{\mathcal{M}}(\cdot) = \mathbf{M}(\cdot)$. The Hessian in (3.23) arising from the inclusion of this linear projection matrix is $\beta^2\mathbf{R}\mathbf{D}_f^{-1}\mathbf{M}^T\mathbf{D}_f\mathbf{M}\mathbf{D}_f^{-1}\mathbf{R}^T$, which is upper bounded by $\beta^2\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. The majorize-minimize algorithm arising from using this surrogate with momentum acceleration and adaptive momentum restart is shown in Fig. 3.4.

Figure 3.4: Analysis denoising algorithm

initialize $j = 0, \mathbf{q}^{(0)}, \mathbf{v}^{(0)} = \mathbf{v}^{(0)}, \mathbf{D}_f, \mathbf{D}_R, \epsilon, \alpha$
repeat
  $\tau^{(j+1)} = (1 + \sqrt{1 + 4(\tau^{(j)})^2})/2$
  $\mathbf{x}^{(k,j+1)} = P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{v}^{(j)})$
  $\mathbf{q}^{(j+1)} = P_{\mathcal{P}^M}(\mathbf{v}^{(j)} - \beta^{-1}\mathbf{D}_R^{-1}\mathbf{R}\mathbf{x}^{(k,j+1)})$
  $\kappa = \left\|\mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}\right\|_2 \left\|\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)}\right\|_2$
  if $\Re\left\{\left\langle \mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}, \mathbf{q}^{(k+1)} - \mathbf{q}^{(k)} \right\rangle\right\} > \alpha\kappa$ **then**
    $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)}$
    $\tau^{(k+1)} = 1$
  else
    $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)})$
  end if
  $j = j + 1$
until $\frac{\left\|\mathbf{x}^{(k,j)} - \mathbf{x}^{(k,j-1)}\right\|_2}{\left\|\mathbf{x}^{(k,j-1)}\right\|_2} \leq \epsilon$
$\mathbf{x}^{(k+1)} = P_{\mathcal{M}}(\mathbf{b}^{(k)} - \beta\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{q}^{(j)})$

The $P_{\mathcal{P}^M}(\cdot)$ operator in Fig. 3.4 projects its argument on to the set, $\mathcal{P}^M$, the ℓ_∞ unit ball. This set arises from our dual formulation discussed in the Appendix. In this case this means that $P_{\mathcal{P}^M}(\cdot)$ examines each element in its input vector and normalizes any elements with an absolute value greater than 1 to an absolute value of 1, preserving the complex sign. For this inner denoising step we include an ϵ stopping criterion, the choice of which as discussed in Section 3.4 depends on the step sizes of the outer iterations of the algorithm in Fig. 3.3. Although not noted in Fig. 3.4, we also included a maximum iteration number to prevent the algorithm from stalling. We did not observe that this was necessary in our numerical experiments, but we wanted to ensure stable convergence in a variety of circumstances. We measure the convergence based on $\mathbf{x}^{(k,j+1)}$, which is calculated from the

momentum variable $\mathbf{v}^{(j)}$, although the actual convergence would depend on the dual variable, $\mathbf{q}^{(j)}$. This simplification avoids making extra computations each iteration that would be required to estimate $\mathbf{x}$ from $\mathbf{q}^{(j)}$, and with the adaptive restart we expect $\mathbf{v}^{(j)}$ to be a good approximation for $\mathbf{q}^{(j)}$ near the solution. Lastly, we note that we initialized the analysis denoising algorithm with the last value for $\mathbf{q}$ from the previous run of the algorithm. This warm start greatly helps the convergence speed of the analysis denoising subroutine.

The algorithm in Fig. 3.4 requires computing a $\mathbf{D}_R$ that satisfies the analysis majorizer condition:

$$\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T, \tag{3.25}$$

where $\mathbf{R}^T$ is the transpose of $\mathbf{R}$. We discuss $\mathbf{D}_R$ for the cases of anisotropic total variation and undecimated Haar wavelets.

3.4.2 Diagonal majorizers for analysis regularizers

One could use Theorem 1 to upper bound any matrix, including $\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. However, in practice we have found that bound somewhat loose for the analysis regularizing matrices of anisotropic total variation and undecimated Haar wavelets. We discuss tighter bounds for these two cases. For the case of anisotropic total variation, we choose

$$\mathbf{D}_R = \text{diag}\{\text{abs}(\mathbf{R})\mathbf{D}_f^{-1}\text{abs}(\mathbf{R}^T)\mathbf{1}\}, \tag{3.26}$$

where $\text{abs}(\cdot)$ returns a matrix that has entries that are the absolute value of the input matrix. This is guaranteed to be a majorizer as it is a modification of the techniques of De Pierro [30], and we have found it to be very tight for anisotropic total variation. Its calculation is also simple.

For the case of undecimated Haar wavelets, we present a different approach that builds on Theorem 1 via the polarization identity. The idea is to split up a non-orthogonal $\mathbf{R}$ into orthogonal pieces for which Theorem 1 will provide tight diagonal majorizers. Consider $\mathbf{R}$ of the form,

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_Q \end{bmatrix}, \tag{3.27}$$

where Q is the number of submatrices of $\mathbf{R}$. Defining $\mathbf{c}_i = \mathbf{D}_f^{-1/2}\mathbf{R}_i^T\mathbf{w}_i$ for any arbitrary

vector $\mathbf{w}^H = [\mathbf{w}_1^H, ..., \mathbf{w}_Q^H]$ (possibly different sized $\mathbf{w}_i$), we then have

$$\begin{aligned}\mathbf{w}^H\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T\mathbf{w} &= \sum_{i=1}^{Q}\|\mathbf{c}_i\|_2^2 + \sum_{i=1}^{Q}\sum_{j=i+1}^{Q} 2\Re\left\{\langle\mathbf{c}_i,\mathbf{c}_j\rangle\right\} \\ &\leq \sum_{i=1}^{Q}\|\mathbf{c}_i\|_2^2 + (Q-1)\sum_{i=1}^{Q}\|\mathbf{c}_i\|_2^2 \\ &= \sum_{i=1}^{Q} Q\mathbf{w}_i\mathbf{R}_i\mathbf{D}_f^{-1}\mathbf{R}_i^T\mathbf{w}_i,\end{aligned} \tag{3.28}$$

where one proceeds from the first to the second step by applying the polarization identity,

$$2\left|\Re\left\{\langle\mathbf{c}_i,\mathbf{c}_j\rangle\right\}\right| \leq \|\mathbf{c}_i\|_2^2 + \|\mathbf{c}_j\|_2^2, \tag{3.29}$$

and collecting all inner product pairs. Thus, finding a majorizer for each $\mathbf{R}_i\mathbf{D}_f^{-1}\mathbf{R}_i$ submatrix will provide a majorizer for $\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$. Such a structure applies to the 2-level undecimated Haar wavelet case since a 2-level undecimated Haar wavelet can be written as a cascade:

$$\mathbf{R} = \mathbf{R}_B\mathbf{R}_A. \tag{3.30}$$

In this case the first step of the cascade, $\mathbf{R}_A$, can be broken up into pieces:

$$\mathbf{R}_A = \begin{bmatrix}\mathbf{R}_{A,1}\\ \mathbf{R}_{A,2}\\ \mathbf{R}_{A,3}\\ \mathbf{R}_{A,4}\end{bmatrix}, \tag{3.31}$$

where each of the $\mathbf{R}_{A,i}$ is an orthogonal 1-level Haar wavelet transform. We apply the inequality in (3.28) to majorize $\mathbf{R}_A\mathbf{D}_f^{-1}\mathbf{R}_A^T$ while using Theorem 1 to majorize each $\mathbf{R}_{A,i}\mathbf{D}_f^{-1}\mathbf{R}_{A,i}^T$ term. We applied the procedure recursively to $\mathbf{R}_B$ since it has a similar structure. In the undecimated Haar wavelet case each of the $\mathbf{R}_{A,1}, ...\mathbf{R}_{A,4}$ is a similar operation, so we expect that each of the $\mathbf{c}_i$ will be approximately linearly dependent and this inequality approach will be fairly tight.

3.5 Experiments

3.5.1 Experimental setup

In the interest of reproducible research, MATLAB code for implementing these methods will be uploaded to the Image Reconstruction Toolbox at

`web.eecs.umich.edu/~fessler/`

All experiments were run on a machine with an Intel Xeon E31230 Processor that had four cores with each core running at 3.2 GHz. The machine had 16 GB of memory. All experiments used $\alpha = -\cos(4\pi/9)$.

We compared the convergence speed of BARISTA to state-of-the-art variable splitting methods in several experiments on four data sets. We present *in vivo* brain results in the main paper body and include results for a numerical brain phantom, a breast phantom, and an American College of Radiology phantom in the supplementary material. The variable splitting methods were each of the AL-P1 or split Bregman type [7]. The AL-P2 method in [7] uses condition number heuristics to tune AL penalty parameters, but we found that these condition number heuristics could change between different regularizers. Tuning AL-P2 for each regularizer would have required setting multiple condition number parameters. AL-P1 has only one constraint penalty parameter, μ, and it had comparable speed to AL-P2, so we used AL-P1 with careful manual tuning of μ as a representative of AL-based methods. We also investigated dynamically updating the μ parameter using update rules proposed by Boyd (Section 3.4.1 of [31]), which helps mitigate tuning difficulties. We initialized such AL methods with dynamic μ updates with one of the manually tuned μ values. In the plots this method with dynamic μ updates is denoted as "AL, dynamic μ." We also introduced a diagonal preconditioner for the conjugate gradient (CG) subroutine in step 4 of AL-P1. We used $\mathbf{P} = (\mathbf{S}^H\mathbf{S}+\mu\mathbf{I})^{-1}$ for all wavelet regularizers and $\mathbf{P} = (\mathbf{S}^H\mathbf{S})^{-1}$ for the total variation regularizer. These preconditioners were not mentioned in [7], but we observed that they accelerated AL-P1 on the order of 50% time to reach the same point of convergence. The AL-P1 methods all used 5 preconditioned CG (PCG) iterations for step 4 of AL-P1.

To track convergence, we computed the following normalized residual as a function of iteration:

$$\xi(k) = 20 \log_{10} \left(\frac{\left\| \mathbf{x}^{(k)} - \mathbf{x}^{(\infty)} \right\|_2}{\left\| \mathbf{x}^{(\infty)} \right\|_2} \right), \tag{3.32}$$

where $\mathbf{x}^{(\infty)}$ is a "converged" solution obtained by running many thousands of iterations of

AL-P1. Note that even though $\mathbf{R}$ is not full column rank in the total variation case, the AL-P1 method is still convergent [32]. In our convergence plot comparisons to AL-based methods we set the lower bound for $\xi(k)$ at -140 dB. We chose to do this for two reasons: 1) our raw MRI data were less precise than single precision and 2) BARISTA vastly outperformed all other methods in reaching double precision, so these parts of the plots were less interesting. We also stored the time at which the kth estimate was computed and in our figures and we plot $\xi(\cdot)$ as a function of elapsed CPU time instead of iteration. We choose to do this since iterations of the proposed majorize-minimize methods and the variable splitting methods have drastically different compute times due to the PCG subroutine and the analysis denoising step in the proposed methods. All methods used identical subroutines for matrix multiplications.

We selected regularization parameter, β, to give visually appealing solutions for each regularizer. In practice the regularization parameter could be estimated via Monte Carlo SURE methods [23].

In our plots we only show BARISTA from the classes of majorize-minimize algorithms as opposed to other methods such as FISTA with $\mathbf{D}_f = L\mathbf{I}$ since BARISTA was always the fastest majorize-minimize method. Fig. 3.5 shows an example of the relative convergence speed of majorize-minimize methods in the case of orthogonal Haar wavelet regularization. In this case, BARISTA was twice as fast as RFISTA (restart FISTA), three times as fast as

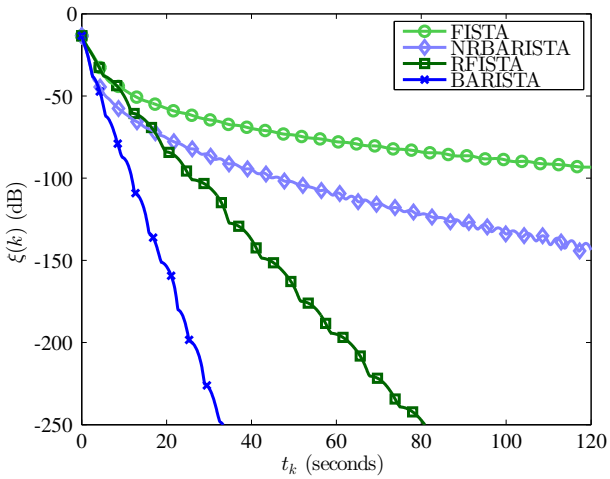
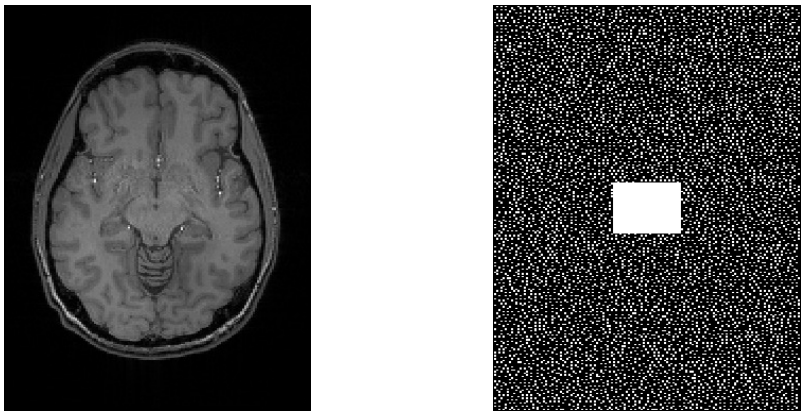


Figure 3.5: Comparison of different majorize-minimize methods with orthogonal Haar wavelet regularization. Markers are placed at 50 iteration intervals. FISTA used $\mathbf{D}_f = \mathbf{D}_R = L\mathbf{I}$ while BARISTA and NRBARISTA (non-restart BARISTA) used the proposed $\mathbf{D}_f$ and $\mathbf{D}_R$. BARISTA is the fastest method; this was also observed for the other experiments with varying degrees of acceleration. Both restart methods exhibit a stair step pattern, where new "steps" arise when the momentum is restarted.

NRBARISTA, (non-restart BARISTA) and over five times as fast as FISTA in reaching -120 dB. Although RFISTA converges rapidly to double precision, in early iterations it is not competitive with BARISTA or variable splitting methods. In a practical setting, the algorithms may not even be run to convergence, so early-iteration convergence speed is critical for general adoption of the proposed methods. Furthermore, negligible time is required to use the majorizing matrices discussed in this paper, so the factor of two speed-up over RFISTA more or less comes for free. We also observed the factor of two speed-up or greater with the orthogonal Daubechies D4 regularizer and the undecimated Haar wavelet regularizer. Speed-up of BARISTA vs. RFISTA in the anisotropic total variation case was negligible. We are unsure why this occurred, but it may be that a shift-variant majorizer makes the analysis denoising problem more difficult to solve in the total variation case.

For the *in vivo* experiment, a 3D data set was acquired on a GE 3T scanner with an 8-channel head coil with acquisition parameters $T_R = 25$ ms, $T_E = 5.172$ ms, and voxel size 1 mm $\times$ 1.35 mm $\times$ 1 mm. The data matrix size was $256 \times 144 \times 128$ uniformly spaced samples. Sensitivity maps were estimated using a quadratic regularized least squares routine [4]. The data were retrospectively undersampled in the Fourier domain using a Poisson disk sampling scheme [33] with a fully sampled center (32-by-32 block), which has been demonstrated to be useful in compressed sensing MRI applications [34]. This sampling pattern simulates one slice of a 3D MRI experiment where the sampling pattern in Fig. 3.6b is in the phase encode plane [34, 7] (this sampling pattern would be impractical for 2D MRI). Only 20% of the full DFT sampling was used for reconstruction. Fig. 3.6a shows



(a) (b)

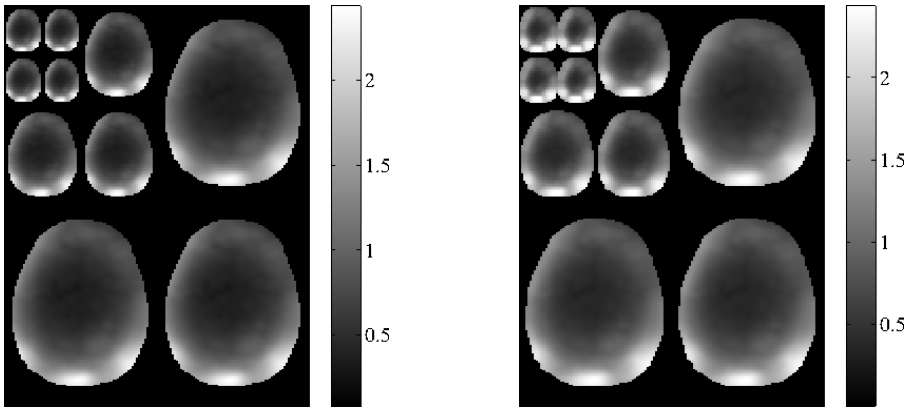
Figure 3.6: Images corresponding to the *in vivo* experiments. (a) $\mathbf{x}$ estimated from fully sampled data. Some residual noise is present at the center. (b) Sampling pattern for the *in vivo* experiments with a densely sampled 32×32 center.

$\mathbf{x}$ estimated from fully sampled data, while Fig. 3.6b shows the Poisson-disc sampling pat-

tern with a densely-sampled center used in all the *in vivo* experiments.

3.5.2 Synthesis regularizer results

As stated earlier, we performed numerical experiments with orthogonal Haar and Daubechies D4 wavelet regularizers to examine the convergence speed of the proposed method in the synthesis setting. We set the regularization parameter to zero for the approximation coefficients since a sparse model does not fit these coefficients as well as the detail coefficients [27]. Fig. 3.7a shows an example of the diagonal majorizing elements in the Haar wavelet basis. Fig. 3.7b shows the majorizer for the Daubechies D4 wavelets. The majorizer for



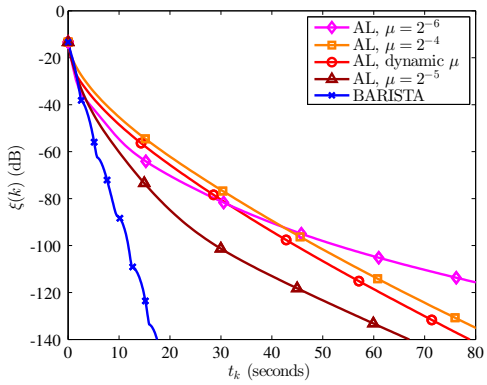
(a) (b)

Figure 3.7: Examples of diagonal elements of $\mathbf{D}_R$ for synthesis regularizers, rearranged into an image. (a) Elements of the diagonal of $\mathbf{D}_R$ in the Haar wavelet basis. Areas outside the brain have been masked for presentation. (b) Elements of $\mathbf{D}_R$ for the Daubechies D4 basis. Since the rows of a Daubechies D4 matrix have larger support than those of the Haar, the majorizer is smoother. For both cases color bars are shown to give a sense of the variation across the image caused by the sensitivity coils. The Lipschitz constant was 2.98, while the maximum value of the squared absolute values of the sensitivity coils was 3.36. Many of the entries in $\mathbf{D}_R$ are smaller than the Lipschitz constant, which gives larger step sizes in step 5 of the algorithm in Fig. 3.2

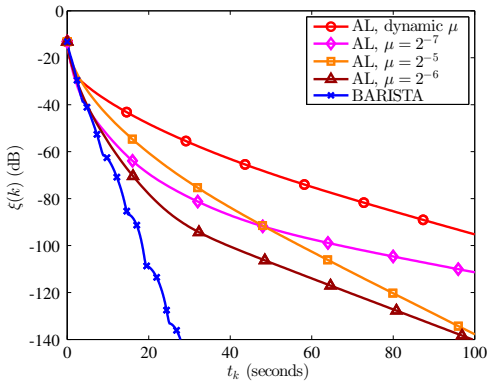
.

the Daubechies D4 wavelet case is smoother than the Haar case since it requires taking maximums over larger patches.

Fig. 3.8a and Fig. 3.8b show the convergence results for Haar and Daubechies D4 wavelets, respectively. BARISTA converges faster than the other methods. The time positions when BARISTA undergoes restart are visible in the stair step pattern in the convergence plots. Several AL parameters are shown to demonstrate the range of speeds of AL-based methods, although we can make no theoretical guarantees on the optimal speed



(a) Orthogonal Haar Wavelet Regularization



(b) Orthogonal Daubechies D4 Wavelet Regularization

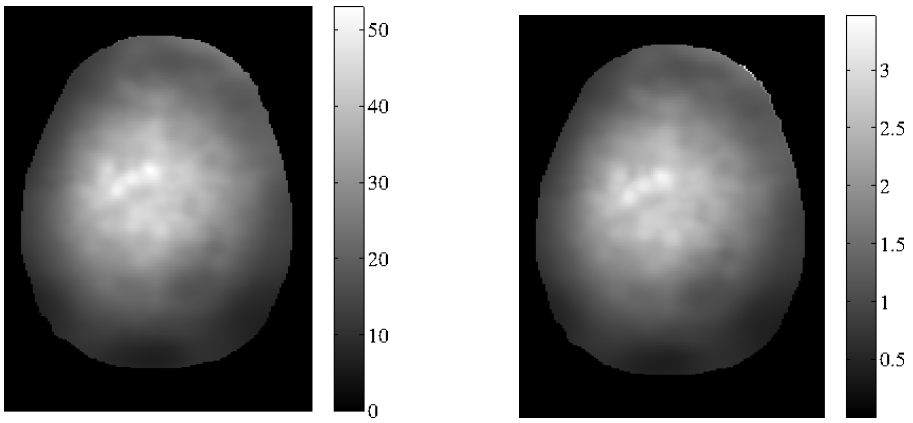
Figure 3.8: Summary of convergence results for two different synthesis regularizers. Markers are placed at 30 iteration intervals for all algorithms. (a) Convergence plot comparing the proposed method to variable splitting methods for orthogonal Haar wavelets. The proposed method with momentum restarting is faster than the other methods. (b) Another convergence plot with orthogonal Daubechies D4 wavelets.

of AL-based methods since we do not know any theoretically optimal way to tune the penalty parameter.

3.5.3 Analysis regularizer results

We performed numerical experiments with total variation and 2-level undecimated Haar wavelet regularization to examine the convergence speed of the proposed methods in the analysis setting. Our anisotropic total variation implementation took differences in vertical, horizontal, and diagonal directions. We did not regularize the approximation coefficients of the 2-level undecimated Haar wavelet transform [27]. Fig. 3.9 shows examples of elements from $\mathbf{D}_R$ for the analysis cases. Since for the analysis case we design $\mathbf{D}_R \succeq \mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$, the sensitivity elements are now inverted relative to the synthesis case. Our analysis algorithm formulation required setting the ϵ_{diff}, ϵ_{min}, and $\epsilon^{(0)}$ parameters. We chose $\epsilon_{\text{min}} = 10^{-12}$, $\epsilon_{\text{diff}} = 10^{-1}$, and $\epsilon^{(0)} = 10^{-1}$. We note that although these convergence criteria parameters require some tuning, we were able to use the same convergence criteria for all regularizers in all experiments. Conversely, we had to tune the constraint penalty parameters for the AL-P1 method each time when changing regularizers or data sets.

Fig. 3.10 shows results for the analysis regularizers. BARISTA matches the other methods in early iterations and outperforms all other methods in later iterations. As previously, the time steps at which the algorithm restarts are shown in the stair step pattern in the



(a) (b)

Figure 3.9: Examples of diagonal elements of $\mathbf{D}_R$ for analysis regularizers rearranged into an image. (a) A subset of the elements of $\mathbf{D}_R$ for the total variation case with areas outside the brain masked for presentation. Since this matrix must upper bound $\mathbf{R}\mathbf{D}_f^{-1}\mathbf{R}^T$, the sensitivity elements have been inverted. (b) A subset of the elements of $\mathbf{D}_R$ for the overcomplete Haar basis case.

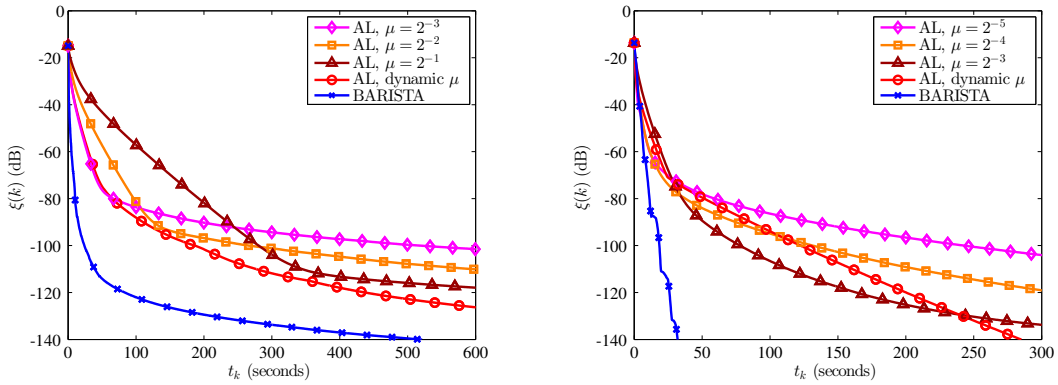
convergence plots. We also observed that all algorithms converged slower with the total variation regularizer than the other regularizers. Results with analysis regularizers with an image domain mask were similar and are shown in the supplementary material. Notably, BARISTA converged about twice as fast when using a mask than without a mask.

3.6 Discussion

3.6.1 Convergence speed of BARISTA vs. AL methods

BARISTA was observed to converge faster than the AL-based methods in both early and late iterations. The early iteration speed of BARISTA is due to its tight approximation of the Hessian of the cost function via the diagonal majorizers developed in this paper and the use of Nesterov momentum acceleration. Nesterov momentum has been added to AL algorithms in some cases [35], although these algorithms require an estimate of the Lipschitz constant, so the diagonal majorizers presented here may be useful for these methods.

The late-iteration speed of BARISTA is due to the use of adaptive restart. We are unaware of a means to apply adaptive restart to AL-based methods. We attempted to recover some of the benefits of adaptive restart through the use of dynamic AL parameter updates, but this did not give the same large convergence speed boost as adaptive restart.



(a) Total Variation Regularization

(b) Undecimated Haar Wavelet Regularization

Figure 3.10: Summary of convergence results for two different analysis regularizers. (a) Convergence plot comparing the proposed method to variable splitting methods for total variation regularization. Markers are placed at 100 iteration intervals. The proposed method with momentum restarting is faster than the other methods. (b) Convergence plot comparing BARISTA to variable splitting methods with undecimated Haar wavelet regularization. Markers are placed at 30 iteration intervals. The proposed method with momentum restarting is faster than the other methods, especially in later iterations.

3.6.2 Selection of AL penalty parameters

We attempted to manually optimize the AL penalty parameters for speed. For some cases, such as total variation and 2-level undecimated Haar wavelet regularizers, we observed a trade-off between early and late iteration convergence speed, with smaller parameters favoring early iteration speed and larger parameters favoring late iteration speed. In our tests we chose the small parameters that gave reasonable convergence to -120 dB; however, this behavior suggests that changing the penalty parameter in a dynamic fashion may improve the convergence speed of AL-based methods. The dynamic tuning method from [31] helped in some cases, but not consistently, and we still observed faster convergence with BARISTA in both early and late iterations.

From a theoretical point of view, analysis of AL-based methods considers static penalty parameters [5]. The fact that AL theory considers static penalty parameters is considered one of the primary motivations for using AL methods instead of penalty methods in the first place [5, 7]; adaptively changing the parameter removes this advantage. Conversely, our use of the $\epsilon^{(k)}$ parameter falls within the MFISTA theory provided monotonicity checks are used, and although [28] does not cover the case of adaptive restart we observed stable convergence of BARISTA in all numerical experiments.

3.6.3 Surrogate tightness and sensitivity coil smoothness

In our data set, the sum of squares of absolute values of the sensitivity coils exhibited high variability across the object. As the sensitivity coils become more uniform, our proposed $\mathbf{D}_f$ and $\mathbf{D}_R$ matrices will more closely approximate their Lipschitz counterparts (i.e., $\mathbf{D}_f = L\mathbf{I}$). In these cases, the advantages of BARISTA will diminish relative to that of RFISTA. However, we typically expect RFISTA to be a lower bound for the speed of BARISTA. Furthermore, as the sensitivity coils become smoother, the proposed surrogate functions actually become better approximations to the original cost function in (3.1), so we expect the speed of the proposed methods to be superior with smooth sensitivity coil profiles than our case with large sensitivity coil variability.

3.6.4 Tuning the restart criterion

The restart criterion in [9] used $\alpha = 0$ in all of their experiments. We found that this choice led to too infrequent restarts because using $\alpha = 0$ allows the momentum and generalized gradient to begin to point in different directions before restarting. Instead, in our experiments we used $\alpha = -\cos(4\pi/9)$. We found that this choice gave very good early iteration convergence in 24 numerical experiments with four different regularizers (see supplementary material).

3.6.5 Near monotonicity of BARISTA

In [28] it is stated that when an iterative procedure is applied to minimize the surrogate cost function, one should apply a monotonicity check to ensure stable convergence of the algorithm in the analysis total variation setting. The primary cause of non-monotonicity with FISTA algorithms is when the momentum takes the algorithm in a bad direction near the solution [9]. In our numerical experiments we observed that the combination of the $\epsilon^{(k)}$ parameter and adaptive restart made the monotonicity checks in [28] unnecessary and the proposed method performed as a monotone algorithm. Nonetheless, the monotonicity checks of [28] could be included in a practical setting if monotonicity is still deemed to be an issue.

3.6.6 Relations to Proximal Newton Methods

The methods outlined in this paper have some relations to Proximal Newton methods (e.g. [36]), which use alternative methods to approximate the Hessian. One issue with such methods is that the memory storage requirements can be undesirably large for medical

imaging problems. Low memory versions of these methods also exist (e.g. L-BFGS, see [36, 37]). BARISTA can also be thought of as having a low memory approximation to the Hessian due to its diagonal structure, which may be more accurate if the SENSE maps dominate the behavior of the Hessian. Comparisons between our proposed method and these more general Proximal Newton methods are an avenue for future investigation. One could even potentially modify BARISTA to use an L-BFGS Hessian approximation update, although Proximal Newton methods are often developed for real numbers and may require adaptations for the complex numbers in MRI reconstruction.

3.6.7 Comparisons to Gradient-Based Methods

A further possible comparison could be to gradient-based methods where the ℓ_1 regularizer is modified slightly to make it differentiable. These methods were found previously to be slower than methods from the AL class [7], possibly due to interaction effects between the sensitivity maps and the regularizer. In principle, one could extend BARISTA to these differentiable settings since a shrinkage operation can be calculated for these regularizers as well [38]. We plan to investigate such approaches in future work.

3.7 Conclusion

We have introduced generalizations of the FISTA algorithm, which we call BARISTA, for SENSE-type MR imaging with compressed sensing regularizers that compensate for the shift-variant aspects of the sensitivity coils. The methods gave superior convergence speed relative to state-of-the-art variable splitting methods in numerical experiments. Furthermore, the proposed methods avoid the penalty parameter tuning associated with variable splitting methods, instead relying on unitless convergence tolerance parameters. We have provided heuristics for selecting these parameters and found that the same values worked well across 24 numerical experiments conducted with four different regularizers on four different data sets. We expect that the proposed methods will give fast, high-quality reconstructions across a wide variety of data sets and will aid in the adoption of compressed sensing methods in a clinical setting.

CHAPTER 4

Majorize-Minimize Algorithms for Non-Cartesian MRI

4.1 Introduction

This chapter is based on work published in [39, 40]. Magnetic resonance imaging (MRI) is a useful imaging modality for diagnosing disease. In MRI, increases in spatial resolution require increases in acquisition time. Since the cost of MRI is proportional to scan time, high resolution MRI images are expensive to obtain. Scanning time reductions can reduce the cost of MRI as well as potentially increasing the temporal resolution in dynamic MRI settings. There are a number of means for facilitating scan time reductions. Non-Cartesian sampling patterns traverse the Fourier space in a way that is more efficient given the hardware constraints of the scanner. These methods typically sample the center of k-space more than the edges since most of the energy in k-space is located at the center. Another way to accelerate MRI is with parallel imaging. SENSitivity Encoding (SENSE) is a parallel MRI technique that undoes aliasing effects caused by undersampling by exploiting variations in the sensitivity profiles (i.e., B1 maps) of multiple coils placed around the patient [1]. When the image can reasonably be assumed to be sparse in some transform domain, compressed sensing regularizers can be used to facilitate further acceleration [2].

When non-Cartesian trajectories, SENSE MRI, and compressed sensing are used, one can estimate the image by solving an ℓ_1-regularized optimization problem [20] where the system matrix incorporates the effects of the multiple sensitivity coils and the non-Cartesian trajectory. The ℓ_1 optimization problem is difficult to solve since the ℓ_1 regularizer is non-differentiable. The nondifferentiability precludes the use of gradient-based optimization algorithms. Nonetheless, there are many methods for minimizing objective functions with ℓ_1 terms. One class of methods is majorize-minimize methods with momentum acceleration [18, 41, 8]. These methods generally work quite well, but they require the calculation of a bound on the Hessian of the data fit term. The standard FISTA method for calculating

this bound [18], which yields the Lipschitz constant that is equal to the maximum eigenvalue of the Hessian, is large due to the large density of samples near the center of k-space for most non-Cartesian trajectories. As such, the Lipschitz constant is a loose bound for the high frequencies of k-space, and these components of the image converge slowly with such algorithms. This issue can be mitigated by more carefully formulating majorizers based on the k-space trajectory and the regularizer [41]; however, previous methods only considered orthogonal wavelet regularizers in the context of single-coil MRI with an iteration-dependent cost function. As such, it remains an open problem to extend these ideas to the full parallel MRI setting with analysis regularizers, which are thought to provide higher image quality [27].

Here we build on the ideas in [41] to build a new algorithm (called CIRCMAJ for "circulant majorizer") tailored to non-Cartesian MRI. Our methods are very similar to those used previously in Chapter 3 [8], the main difference being that in this case we are considering coupling between the density of samples in k-space and the frequency localization property of the wavelet transform rather than coupling between the sum of squares of the sensitivity coils and the spatial localization property of the wavelet transform. Our new algorithm is specifically designed for non-Cartesian imaging to better-capture the variation in k-space sampling density. In numerical experiments we observe that this leads to faster convergence of high-spatial frequency features (such as edges) in the images, as well as convergence speed measured by the 2-norm.

4.2 Theory

Non-Cartesian MR image reconstruction with sparsity-promoting regularization can be formulated as the following optimization problem:

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) + \beta R(\mathbf{x}) \\ f(\mathbf{x}) &= \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2, R(\mathbf{x}) = \left\|\mathbf{R}\mathbf{x}\right\|_1 \end{aligned}. \tag{4.1}$$

We call $f(\mathbf{x})$ the data fit term and $R(\mathbf{x})$ the regularizer. As discussed in Chapter 2, we assume that we are in the SENSE setting, so we have

$$\mathbf{A} = \mathbf{F}\mathbf{S}, \tag{4.2}$$

where $\mathbf{F} \in \mathbb{C}^{NC \times NC}$ is a Fourier encoding operator and $\mathbf{S} \in \mathbb{C}^{NC \times N}$ is a matrix with sensitivity maps. N is the number of voxels in the image and C is the number of sensitivity

coils. $\mathbf{F}$ and $\mathbf{S}$ have the following block structures:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & & \\ & \ddots & \\ & & \mathbf{F}_1 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_C \end{bmatrix}, \tag{4.3}$$

where each of the $\mathbf{F}_1$ is a non-Cartesian Fourier encoding operator and each of the $\mathbf{S}_c$ is a distinct diagonal matrix with the sensitivity map of coil c along its diagonal. A similar formulation with Cartesian sampling was studied in [8].

There are a number of potential approaches for solving (4.1). One would be to use variable-splitting methods via the augmented Lagrangian (AL) [7, 42]. A drawback of using variable-splitting methods is that they require tuning penalty parameters. The main way of setting the penalty parameters is to use heuristics such as looking at conditioning of various subproblems of the AL algorithm. However, these heuristics are not guaranteed to give optimal parameters, and in Chapter 3 we showed that momentum-accelerated algorithms can be faster than AL algorithms even when the penalty parameter is manually tuned [8]. Furthermore, majorize-minimize methods are currently used in a number of non-Cartesian MRI applications [3, 43]. For these reasons, we instead choose to develop a momentum-accelerated algorithm by first developing a majorize-minimize method for minimizing the cost function in (4.1) and then adding Nesterov momentum [18]. The standard momentum-accelerated majorize-minimize algorithm is FISTA [18], but FISTA is slow in this setting because the associated Lipschitz constant depends primarily on the density samples near the center of k-space for most non-Cartesian trajectories. As a result, this Lipschitz bound is loose for the samples far away from the center of k-space. We address this issue with FISTA by carefully considering the structures of $\mathbf{A}$ and $\mathbf{R}$. We develop an MM algorithm for solving (4.1) in two steps similar to those used previously [28, 8]. In the first step we develop a tight quadratic surrogate for $f(\mathbf{x})$, which requires majorizing $\mathbf{A}^H\mathbf{A}$ with a circulant matrix, $\mathbf{M}_f$. In the second step we consider $R(\mathbf{x})$ via a dual gradient ascent algorithm. This requires developing a second bound involving careful consideration of the structure of $\mathbf{M}_f$ and $\mathbf{R}$.

4.2.1 Bounding the Non-Cartesian System Matrix

We first recall from Chapter 2 that minimizing the following quadratic surrogate for $f(\mathbf{x})$ is guaranteed to decrease $f(\mathbf{x})$ when $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$ [26, 8]:

$$\psi(\mathbf{x}, \mathbf{x}^{(k)}) := \frac{1}{2}\left\|\mathbf{x} - (\mathbf{x}^{(k)} - \mathbf{M}_f^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}))\right\|_{\mathbf{M}_f}^2 \tag{4.4}$$

This surrogate performs better when $\mathbf{M}_f$ is a tight bound for the Hessian, $\mathbf{A}^H\mathbf{A}$. The Hessian has two major components: the non-Cartesian Fourier encoder and the sensitivity maps. In the non-Cartesian setting, the effects of $\mathbf{F}$ become more dominant due to the wide variability of k-space sampling density in most non-Cartesian trajectories. As such, the behavior of $\mathbf{A}^H\mathbf{A}$ can be loosely approximated by that of $\mathbf{F}^H\mathbf{F}$. $\mathbf{F}^H\mathbf{F}$ is a matrix that amplifies the low spatial frequencies of its input. We capture these effects by designing $\mathbf{M}_f$ to be a circulant matrix. The frequency response of this circulant matrix will amplify low frequencies proportional to the density of k-space samples.

If we constrain $\mathbf{M}_f$ to be circulant, then we can write

$$\mathbf{M}_f = \mathbf{Q}^{-1}\mathbf{\Lambda}_f\mathbf{Q}, \tag{4.5}$$

where $\mathbf{Q}$ is a DFT matrix and $\mathbf{\Lambda}_f$ is the frequency response of $\mathbf{M}_f$. Since $\mathbf{Q}$ is fixed, our design for $\mathbf{M}_f$ only requires calculation of $\mathbf{\Lambda}_f$. We calculate $\mathbf{\Lambda}_f$ in a three-step process based on $\mathbf{F}_1$, the non-Cartesian k-space encoding matrix that is similar to one previously used for non-Cartesian MRI [42].

Step 1: Find a close circulant approximation to $\mathbf{F}_1^H\mathbf{F}_1$ by taking the DFT of an impulse response. Let $\mathbf{e}_0 \in \mathbb{C}^{N\times 1}$ be a vector with a 1 at the center of the image. Then, $\mathbf{\Lambda}_{f,1}$, the first step of calculating $\mathbf{\Lambda}_f$ is

$$\mathbf{\Lambda}_{f,1} = \Re\left\{\text{diag}(\mathbf{Q}\mathbf{F}_1^H\mathbf{F}_1\mathbf{e}_0)\right\}, \tag{4.6}$$

which yields a close circulant approximation to $\mathbf{F}_1^H\mathbf{F}_1$.

Step 2: Increase the minimum entry in $\mathbf{\Lambda}_{f,1}$ by κ_2 where κ_2 is the minimum eigenvalue of $\mathbf{Q}^{-1}\mathbf{\Lambda}_{f,1}\mathbf{Q} - \mathbf{F}_1^H\mathbf{F}_1$. This gives $\mathbf{\Lambda}_{f,2}$.

Step 3: Calculate the final $\mathbf{\Lambda}_f$ as

$$\mathbf{\Lambda}_f = \mathbf{\Lambda}_{f,2} + \kappa_3\mathbf{I}, \tag{4.7}$$

where κ_3 is the minimum eigenvalue of $\mathbf{Q}^{-1}\mathbf{\Lambda}_{f,2}\mathbf{Q} - \mathbf{F}_1^H\mathbf{F}_1$ (calculated via power iteration).

Step 1 finds a good circulant approximation to $\mathbf{F}_1^H\mathbf{F}_1$. Step 3 ensures that $\mathbf{M}_f \succeq \mathbf{F}_1^H\mathbf{F}_1$.

Step 2 is inserted between Steps 1 and 3 to ensure that the maximum entry in $\mathbf{\Lambda}_{f,1}$, corresponding to the density of samples at the center of k-space, is not increased too much in the majorizer calculation. We found empirically that $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$ in numerical experiments when the sensitivity maps were calculated with a sum-of-squares normalization. Calculation of $\mathbf{M}_f$ in this fashion empirically gave a far tighter for $\mathbf{A}^H\mathbf{A}$ than the Lipschitz constant. Inserting this surrogate into the cost function in (4.1) gives the following surrogate for the entire optimization problem:

$$\eta\left(\mathbf{x}, \mathbf{x}^{(k)}\right) = \psi\left(\mathbf{x}, \mathbf{x}^{(k)}\right) + \beta R(\mathbf{x}). \tag{4.8}$$

Iteratively minimizing this surrogate would yield a majorize-minimize algorithm if $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$. When a majorize-minimize algorithm is used, one can add Nesterov momentum to speed the algorithm [18]. Here we choose to add a form of momentum related to Nesterov momentum that has been proven to be faster for smooth, convex cost functions and verified to yield performance improvements in MRI [10, 11]. We also adapted this new brand of momentum to use adaptive momentum restarting [9]. The addition of momentum to a

Figure 4.1: CIRCMAJ for Analysis

1: initialize $k = 0, \mathbf{z}^{(0)} = \mathbf{x}^{(0)}, \mathbf{M}_f, \alpha, \epsilon^{(0)}$
2: calculate $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$ as outlined in Section 4.2.1
3: calculate $\mathbf{D}_R \succeq \mathbf{R}\mathbf{M}_f^{-1}\mathbf{R}^T$ as outlined in Section 4.2.2
4: **while** $k < K$ **do**
5: $\quad \tau^{(k+1)} = (1 + \sqrt{1 + 4(\tau^{(k)})^2})/2$
6: $\quad \mathbf{b}^{(k)} = \mathbf{z}^{(k)} - \mathbf{M}_f^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{z}^{(k)} - \mathbf{y})$
7: $\quad$ Run Fig. 4.2 algorithm to $\epsilon^{(k)}$ convergence to get $\mathbf{x}^{(k+1)}$
8: $\quad \kappa = \left\|\mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}\right\|_2 \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2$
9: $\quad$ **if** $\Re\left\{\left\langle \mathbf{z}^{(k)} - \mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\rangle\right\} > \alpha\kappa$ **then**
10: $\quad\quad \mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)}$
11: $\quad\quad \tau^{(k+1)} = 1$
12: $\quad$ **else**
13: $\quad\quad \mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + \frac{\tau^{(k)}}{\tau^{(k+1)}}(\mathbf{x}^{(k+1)} - \mathbf{z}^{(k)})$
14: $\quad$ **end if**
15: $\quad \epsilon^{(k+1)} = \max(\min(\epsilon_{\text{diff}} \frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{x}^{(k)}\right\|_2}, \epsilon^{(k)}), \epsilon_{\text{min}})$
16: $\quad k = k + 1$
17: **end while**
18: $\hat{\mathbf{x}} = \mathbf{x}^{(K)}$

standard majorize-minimize algorithm can be seen in the use of an additional $\mathbf{z}^{(k)}$ variable and the calculation of the gradient in Step 6 of the the algorithm in Fig. 4.1. The use of adaptive momentum restarting is seen in the sequence of if/else statements at the end of

each iteration. The adaptive momentum restarting resets the momentum when it is taking the iterates in a direction away from that of the generalized gradient, which improves convergence speed. The algorithm in Fig. 4.1 requires a method for decreasing the surrogate in (4.8). There are a number of ways for doing this. We discuss a dual gradient ascent algorithm as well as an additional matrix bound below.

4.2.2 Consideration of the Regularizer

Section 4.2.1 gives a means for calculating $\mathbf{M}_f$, where we observe empirically that $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$, which enables us to use $\psi\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$ as a surrogate for $f(\mathbf{x})$. We now need to develop a method for decreasing the surrogate in (4.8):

$$\eta\left(\mathbf{x}, \mathbf{x}^{(k)}\right) = \psi\left(\mathbf{x}, \mathbf{x}^{(k)}\right) + \beta R(\mathbf{x}). \tag{4.8}$$

This surrogate can be decreased with a shrinkage step if we use a synthesis model or via a dual gradient ascent algorithm if we use an analysis model [8]. Both of these updates require the calculation of a diagonal matrix, $\mathbf{D}_R$. For the shrinkage step, we must have $\mathbf{D}_R \succeq \mathbf{R}\mathbf{M}_f\mathbf{R}^T$. For the dual gradient ascent algorithm, we must have $\mathbf{D}_R \succeq \mathbf{R}\mathbf{M}_f^{-1}\mathbf{R}^T$. Since $\mathbf{M}_f = \mathbf{Q}^{-1}\mathbf{\Lambda}_f\mathbf{Q}$ and $\mathbf{M}_f^{-1} = \mathbf{Q}^{-1}\mathbf{\Lambda}_f^{-1}\mathbf{Q}$, mathematical techniques used to develop $\mathbf{D}_R$ for either case are interchangeable. For this reason, we focus primarily on the analysis case where $\mathbf{D}_R \succeq \mathbf{R}\mathbf{M}_f^{-1}\mathbf{R}^T$.

To calculate $\mathbf{D}_R$ we will assume that $\mathbf{R}$ can be written as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_B \end{bmatrix}, \tag{4.9}$$

where $\mathbf{R}_b = \mathbf{Q}^{-1}\mathbf{\Lambda}_b\mathbf{Q}$. This implies that $\mathbf{R}$ can be decomposed into the output of B filters with periodic boundary conditions where the bth filter has frequency response $\mathbf{\Lambda}_b$. This includes many popular choices of $\mathbf{R}$ such as undecimated wavelets and finite differences. When $\mathbf{R}$ takes on this form, we can write

$$\mathbf{R}\mathbf{M}_f^{-1}\mathbf{R}^T = \begin{bmatrix} \mathbf{Q}^{-1} & & \\ & \ddots & \\ & & \mathbf{Q}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1\mathbf{\Lambda}_f\mathbf{\Lambda}_1^* & \cdots & \mathbf{\Lambda}_1\mathbf{\Lambda}_f\mathbf{\Lambda}_B^* \\ \vdots & \ddots & \vdots \\ \mathbf{\Lambda}_B\mathbf{\Lambda}_f\mathbf{\Lambda}_1^* & \cdots & \mathbf{\Lambda}_B\mathbf{\Lambda}_f\mathbf{\Lambda}_B^* \end{bmatrix} \begin{bmatrix} \mathbf{Q} & & \\ & \ddots & \\ & & \mathbf{Q} \end{bmatrix}, \tag{4.10}$$

due to the interaction effects of the DFT matrices in $\mathbf{R}$ and $\mathbf{M}_f^{-1}$. Applying Geršgorin's

Theorem to the central matrix gives

$$\mathbf{R}\mathbf{M}_f^{-1}\mathbf{R}^T \preceq \mathbf{D}_R := \begin{bmatrix} d_{R,1}\mathbf{I} & & \\ & \ddots & \\ & & d_{R,B}\mathbf{I} \end{bmatrix}, \tag{4.11}$$

where $d_{R,b} = \max_n \left(\sum_{p=1}^{B} |\lambda_{n,b}\lambda_{n,b}^* / d_{n,f}| \right)$. Note: in the synthesis setting where we use $\mathbf{M}_f$ we have $d_{R,b} = \max_n \left(\sum_{p=1}^{B} |\lambda_{n,b}\lambda_{n,b}^* d_{n,f}| \right)$. This matrix can then be used in a shrinkage step or a dual gradient ascent algorithm in the fashion described in Chapter 3 and [8]. This gives the algorithm in Fig. 4.2.

Figure 4.2: Analysis Denoising Algorithm for CIRCMAJ

initialize $j = 0, \mathbf{q}^{(0)}, \mathbf{v}^{(0)} = \mathbf{v}^{(0)}, \mathbf{M}_f, \mathbf{D}_R, \epsilon, \alpha$
repeat
  $\tau^{(j+1)} = (1 + \sqrt{1 + 4(\tau^{(j)})^2})/2$
  $\mathbf{x}^{(k,j+1)} = \mathbf{b}^{(k)} - \beta \mathbf{M}_f^{-1}\mathbf{R}^T\mathbf{v}^{(j)}$
  $\mathbf{q}^{(j+1)} = P_{\mathcal{P}^M}(\mathbf{v}^{(j)} - \beta^{-1}\mathbf{D}_R^{-1}\mathbf{R}\mathbf{x}^{(k,j+1)})$
  $\kappa = \left\|\mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}\right\|_2 \left\|\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)}\right\|_2$
  if $\Re\left\{\left\langle \mathbf{v}^{(k)} - \mathbf{q}^{(k+1)}, \mathbf{q}^{(k+1)} - \mathbf{q}^{(k)} \right\rangle\right\} > \alpha\kappa$ **then**
    $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)}$
    $\tau^{(k+1)} = 1$
  else
    $\mathbf{v}^{(k+1)} = \mathbf{q}^{(k+1)} + \frac{\tau^{(k)}-1}{\tau^{(k+1)}}(\mathbf{q}^{(k+1)} - \mathbf{q}^{(k)}) + \frac{\tau^{(k)}}{\tau^{(k+1)}}(\mathbf{q}^{(k+1)} - \mathbf{v}^{(k)})$
  end if
  $j = j + 1$
until $\frac{\left\|\mathbf{x}^{(k,j)} - \mathbf{x}^{(k,j-1)}\right\|_2}{\left\|\mathbf{x}^{(k,j-1)}\right\|_2} \leq \epsilon$
$\mathbf{x}^{(k+1)} = \mathbf{b}^{(k)} - \beta \mathbf{M}_f^{-1}\mathbf{R}^T\mathbf{q}^{(j)}$

$P_{\mathcal{P}^M}(\cdot)$ is a projection operator that projects on to the set, $\mathcal{P}^M$, which is the ℓ_∞ unit ball [8]. The algorithm in Fig. 4.2 can be derived as a majorize-minimize algorithm if $\mathbf{M}_f \succeq \mathbf{A}^H\mathbf{A}$ [8]. For this reason we have included momentum [18, 44] and adaptive momentum restarting [9] here as well. The momentum can be seen in the use of an extra $\mathbf{v}^{(j)}$ variable and the adaptive momentum restarting can be seen in the sequence of if/else statements at the end of each iteration.

4.3 Experiments

We compared the convergence speed of the proposed algorithm, CIRCMAJ, to that of FISTA in numerical experiments. Instead of using the standard FISTA method [18], we included all standard FISTA algorithm accelerations used with CIRCMAJ such as adaptive momentum restarting [9] and optimal momentum calculation [44, 11]. As such, the FISTA methods are equivalent to CIRCMAJ with $\mathbf{M}_f = L\mathbf{I}$, where L is the Lipschitz constant or maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$. Both methods used adaptive momentum restarting rules previously developed for MRI [8].

To track convergence, we computed the normalized residual as a function of iteration:

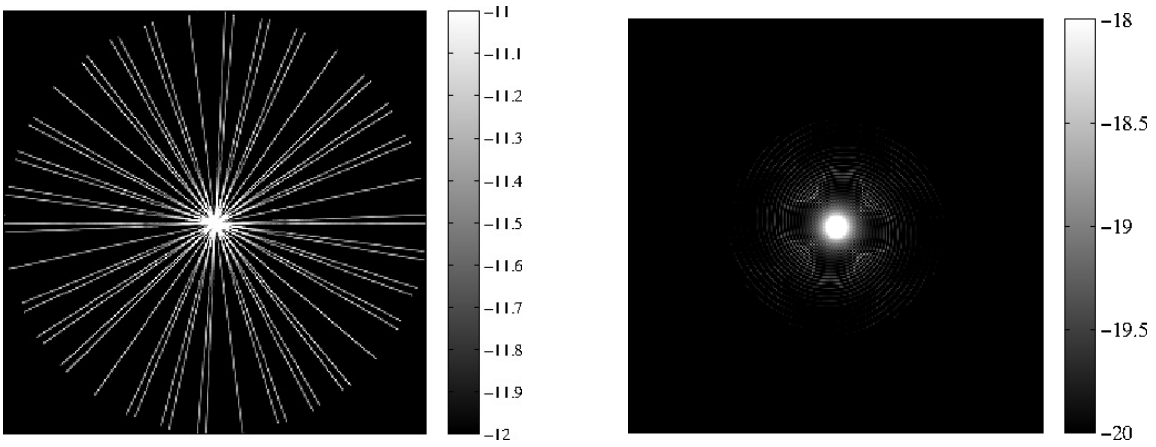
$$\xi(k) = \frac{\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(\infty)}\right\|}{\left\|\mathbf{x}^{(\infty)}\right\|}, \tag{4.12}$$

where $\mathbf{x}^{(\infty)}$ is a "converged solution" obtained by running many thousands of iterations of FISTA. We also stored the time at which the kth estimate was computed and in our figures we plot $\xi(k)$ as a function of time t_k. We choose to do this since the iteration cost of FISTA and CIRCMAJ are slightly different due to the DFTs in the circulant majorizer. The regularization parameter, β, was selected to give visually appealing images for each data set. Numerical experiments on synthetic data were run with both radial and spiral trajectories. All numerical experiments used NUFFT routines with a min-max interpolator and Kaiser-Bessel scaling coefficients [14, 17] for the Fourier encoder in each $\mathbf{F}_c$.

4.3.1 Structure of the Majorizers

The result of the bounds in Section 4.2 depend on the structure of $\mathbf{A}^H\mathbf{A}$. Since $\mathbf{M}_f$ is designed based on $\mathbf{F}_1$ (which corresponds to a non-Cartesian Fourier encoder based on the k-space trajectory), its frequency response is generally much larger where there are more k-space samples. This can be seen in Fig. 4.3. Fig. 4.3a shows the density of samples in a 30-spoke radial trajectory (7.5% of fully-sampled data) while Fig. 4.3b shows the density of samples in a 10-leaf spiral trajectory (21% of fully-sampled data). The magnitude of the brightest spot in these images is on the order of the standard Lipschitz constant for non-Cartesian trajectories. Fig. 4.4 shows the variation in the coefficients of $\mathbf{\Lambda}_f$ on a dB scale normalized to the maximum value. The tighter circulant $\mathbf{M}_f$ bounds $\mathbf{A}^H\mathbf{A}$ much tighter at higher frequencies than the Lipschitz constant. As such, we can expect the edges of the image to converge faster with the non-Cartesian majorizer than with standard FISTA.

The structure of the $\mathbf{D}_R$ matrices depend on the couplings between the frequency responses of different subbands of $\mathbf{R}$. In our numerical experiments we chose to make $\mathbf{R}$ and



(a) (b)

Figure 4.3: Examples of $\mathbf{\Lambda}_f$ for different trajectories in dB relative to the maximum entry. (a) shows $\mathbf{\Lambda}_f$ for a 30-spoke radial trajectory. (b) shows $\mathbf{\Lambda}_f$ for a subsampled spiral trajectory.

undecimated Haar wavelet transform, so each $\mathbf{R}_b$ is a simple 2-by-2 high-pass filter. The Geršgorin bound would be tighter if we used other wavelets where the frequency response of each highpass filter overlaps less. Fig. 4.5 shows that the overlap from these various high-pass filters is modest. Higher order Daubechies wavelets could be expected to have even less overlap in their frequency responses, so the Haar wavelet example in Fig. 4.5 can be thought of more as a worst-case example.

4.3.2 Radial Experiments

A BrainWeb phantom of size 256×256 was used to generate a synthetic data set from a 30-spoke radial k-space trajectory. Eight sensitivity coils were simulated for the data generation using a previous method [45] and Gaussian noise was added to the k-space data. The sensitivity coils for reconstruction were calculated from this data set using a sum-of-squares normalization and a model-based sensitivity map estimation scheme [4]. The image was then reconstructed using the cost function in (4.1) with both FISTA and the proposed CIRCMAJ method to compare convergence speed. The regularizer, $\mathbf{R}$, was chosen to be an undecimated Haar wavelet transform and β was chosen to give visually pleasing images by manually tuning over a log-2 scale.

Fig. 4.6 shows images from various stages of the reconstruction algorithms. Fig. 4.6a shows $\mathbf{x}^{(\infty)}$, the converged image from the cost function in (4.1) with the manually-chosen β. Fig. 4.6b shows $\mathbf{x}^{(5)}$ from FISTA, which is blurry due to the looseness of the Lipschitz bound for high spatial frequencies. Fig. 4.6c shows $\mathbf{x}^{(5)}$ from CIRCMAJ, which has better-

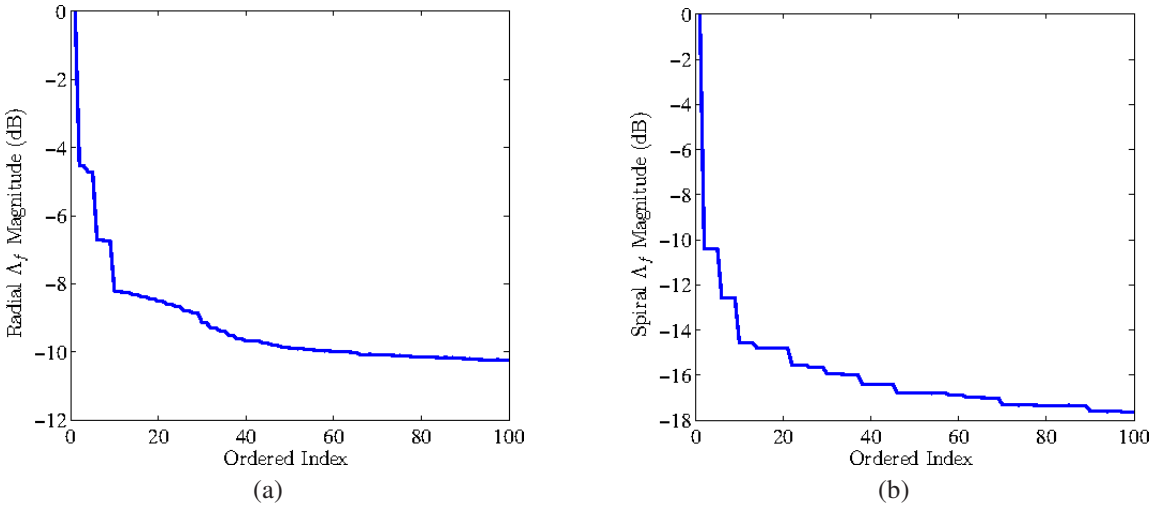


Figure 4.4: Sorted plot of magnitude of coefficients in $\mathbf{\Lambda}_f$ for radial and spiral trajectories. (a) shows the magnitude of coefficients from $\mathbf{\Lambda}_f$ from a 30-spoke radial trajectory. (b) shows the magnitude of coefficients from $\mathbf{\Lambda}_f$ from a 10-leaf spiral trajectory.

defined edges due to the circulant majorizer. Although Fig. 4.6b and 4.6c look different at iteration 5, both algorithms converge to the same $\mathbf{x}^{(\infty)}$ in Fig. 4.6a. Fig. 4.6 indicates that the edges take longer to converge with FISTA.

Fig. 4.7 compares the convergence speed of CIRCMAJ to that of FISTA. CIRCMAJ is about 2-3 times faster in 2-norm convergence. We note that the 2-norm is dominated by the lower spatial frequencies, where the improvement of CIRCMAJ over FISTA is not as dramatic. Nonetheless, Fig. 4.7 indicates that CIRCMAJ still provides a noticeable improvement in 2-norm convergence speed. The stair-step pattern in the convergence plots are where momentum restarting is taking place.

4.3.3 Spiral Experiments

The spiral numerical experiments were identical to the radial numerical experiments with the exception that a 10-leaf spiral trajectory (undersampling factor of 21%) was used instead of a radial trajectory. As such, the experiments also involved eight sensitivity coils that simulated and then estimated from simulated data via a model-based routine [4]. Gaussian noise was added to the simulated data. Fig. 4.8 shows the reconstructed image from various stages of running the reconstruction algorithms. Fig. 4.8a shows $\mathbf{x}^{(\infty)}$, the "converged" image after running many thousands of iterations of FISTA. Fig. 4.8b shows $\mathbf{x}^{(5)}$ after running FISTA for 5 iterations. Fig. 4.8c shows $\mathbf{x}^{(5)}$ for CIRCMAJ. As in the radial case, CIRCMAJ has better resolved the edges relative to FISTA in early iterations. There

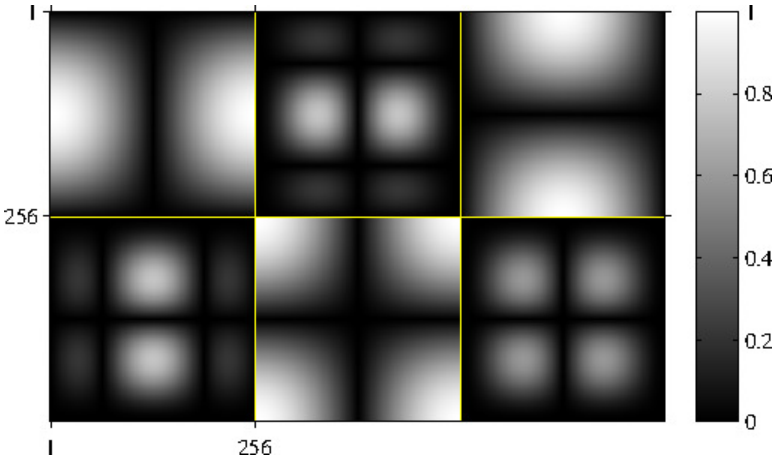
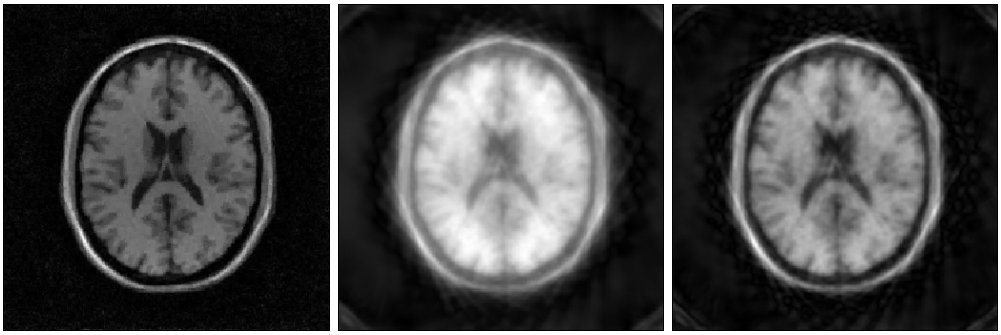


Figure 4.5: Image of 2D frequency responses for various high-pass filters in a Haar wavelet transform.



(a) (b) (c)

Figure 4.6: Images from various stages of the reconstruction algorithms. (a) shows $\mathbf{x}^{(\infty)}$, the converged image after many thousands of iterations of FISTA. (b) shows $\mathbf{x}^{(5)}$ after five iterations of FISTA. (c) shows $\mathbf{x}^{(5)}$ after five iterations of CIRCMAJ.

is more substantial aliasing in the spiral images than the radial images at early iterations. This could possibly be due to the center of k-space being less densely-sampled for these trajectories, as there are only 10 spiral lines near the center as opposed to 30 radial lines for the radial trajectory. We observed that these aliasing artifacts resolved more quickly with CIRCMAJ than FISTA. We also compared the algorithms in 2-norm convergence speed in Fig. 4.9. As in the case with a radial k-space trajectory, CIRCMAJ provides a modest boost in 2-norm convergence speed over that of FISTA.

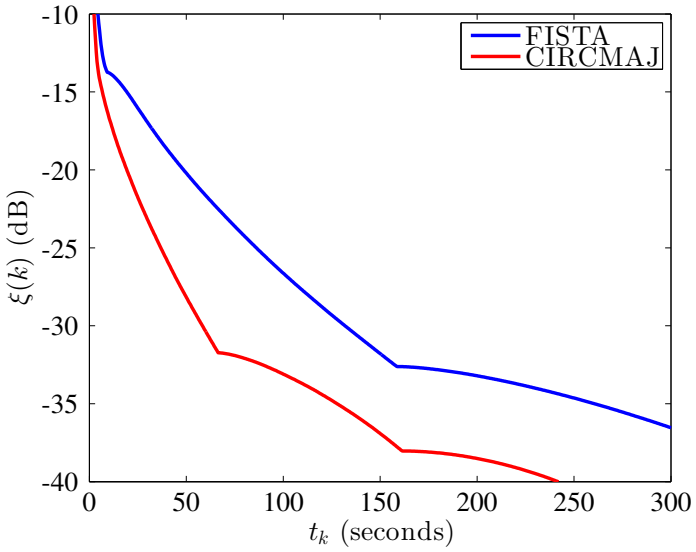


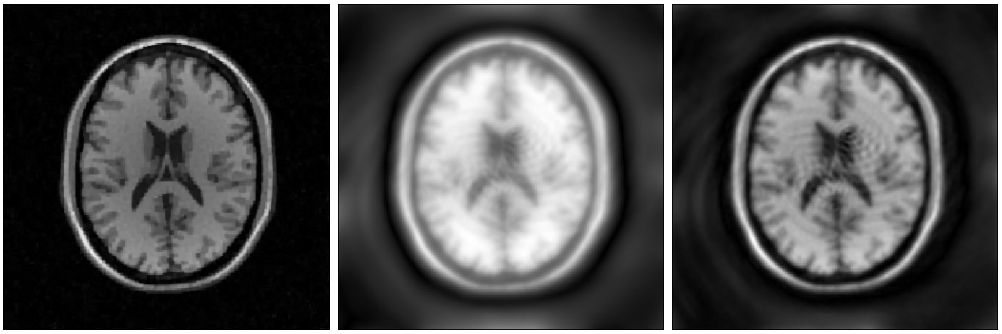
Figure 4.7: Convergence plot comparing FISTA to CIRCMAJ for a radial trajectory.

4.4 Discussion

4.4.1 Bounding the Sensitivity Maps

Here we derived $\mathbf{M}_f$ only from offline calculations based on $\mathbf{F}_1^H\mathbf{F}_1$. This guarantees that $\mathbf{M}_f$ is an upper bound for $\mathbf{F}_1^H\mathbf{F}_1$, but it does not guarantee that $\mathbf{M}_f$ is an upper bound for $\mathbf{A}^H\mathbf{A}$. We found empirically that $\mathbf{M}_f$ is an upper bound for $\mathbf{A}^H\mathbf{A}$ when the sensitivity maps are calculated using a sum-of-squares normalization. This essentially requires that the sensitivity map squares add up to 1 at every location in the image, so the fact that we observed that $\mathbf{M}_f$ is an upper bound for $\mathbf{A}^H\mathbf{A}$ makes sense from the perspective that $\mathbf{A}^H\mathbf{A}$ can be thought of as a weighted combination of $\mathbf{F}_1^H\mathbf{F}_1$ matrices. Nonetheless, we were unable to prove this fact analytically, and further experimental validation may be necessary before CIRCMAJ is used in a clinical setting.

Although we were unable to analytically bound $\mathbf{A}^H\mathbf{A}$, this issue could be compensated for by backtracking [18]. FISTA with backtracking in this setting would essentially give the MFISTA algorithm, which still has a well-developed convergence theory [28]. The incorporation of backtracking would add some computational cost, so determination of whether this approach is useful in a practical setting would require further investigation.



(a) (b) (c)

Figure 4.8: Images from various stages of the reconstruction algorithms with the spiral trajectory. (a) shows $\mathbf{x}^{(\infty)}$, the converged image after many thousands of iterations of FISTA. (b) shows $\mathbf{x}^{(5)}$ after five iterations of FISTA. (c) shows $\mathbf{x}^{(5)}$ after five iterations of CIRCMAJ.

4.4.2 Convergence Speed of CIRCMAJ

The experiments on spiral and radial trajectories led to a modest acceleration in 2-norm convergence as observed in Fig. 4.7 and Fig. 4.9. However, we note that the 2-norm is dominated by the low frequencies. Our new formulation in CIRCMAJ is not substantially different for low frequencies relative to FISTA; most of the benefit of our new approach can be seen in Fig. 4.6 and Fig. 4.8, where the edges are more sharply resolved earlier in the algorithm. Since the human visual system is better-tuned to interpret the edges in the image, this may indicate that the 2-norm convergence plots underestimate the improvements in early-iteration image quality given by the algorithm; however, confirmation of this improvement would require analysis of the images by expert radiologists.

4.4.3 Use of a Mask

Previously in our analysis algorithm development we included a mask. The resulting algorithm was easier to analyze since it resulted in a separable surrogate with box constraints (the box constraints being a support mask). With a circulant majorizer, the resulting surrogate is no longer separable and the algorithm becomes more complicated. As such, we did not include a mask in our numerical experiments. Nonetheless, use of a mask may not impact the algorithms drastically since most of the region outside the brain is close to zero. The resulting algorithm would be similar to projected gradient descent, which has been demonstrated to be useful in other applications. In the future we plan to run further experiments to determine the effects of masking with a circulant, non-separable majorizing function.

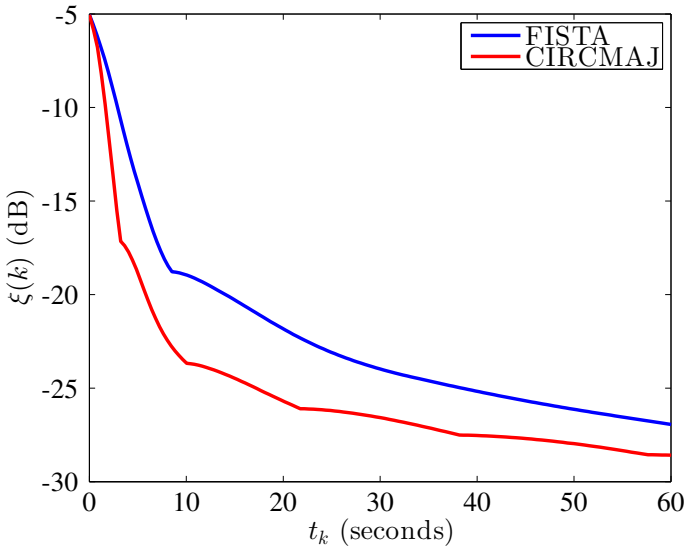


Figure 4.9: Convergence plot comparing FISTA to CIRCMAJ for a spiral trajectory.

4.4.4 Extension to Dynamic Imaging

Here we considered static imaging, although the methods could possibly be extended to dynamic imaging settings where the regularizer is space-time separable. An example of this would be dynamic imaging settings that use a total variation penalty on the spatial aspect of the reconstructed time series. For this case, one could majorize the data fit term with the same circulant majorizer applied to each time point. Then, the step-sizes in the dual gradient ascent algorithm could be decomposed to a form that resembles the static reconstruction problem discussed extensively in this work and the corresponding techniques could be applied. We plan to investigate extensions to dynamic imaging in future work.

4.5 Conclusion

We have introduced a number of key improvements to previous methods [46]. The first of these is an extension that allows the use of analysis regularizers (e.g., undecimated Haar wavelets) in place of an iteration-dependent cost function used in [46]. This extension allows a more streamlined theoretical treatment of the properties of the images that our algorithm yields. Secondly, we demonstrated that the trajectory-based majorizers that we developed could be used for multicoil reconstruction, whereas previously only disjoint single-coil reconstruction had been considered [46]. Finally, we have incorporated the advances of optimized momentum [10] and adaptive momentum restarting [9], which yield further accelerations to the algorithm. In the future we plan to run more experiments to em-

pirically verify that our matrix bound also applies when the sensitivity maps are calculated with a sum-of-squares normalization. We also plan to incorporate further modifications to CIRCMAJ, such as backtracking, to account for cases where our bound is insufficient. Finally, in the future we plan to explore whether the use of an object support mask is possible with our circulant majorizer.

CHAPTER 5

Acceleration of fMRI Scans

5.1 Introduction

Functional MRI (fMRI) is a powerful technique for studying brain function that relies on changes in blood oxygenation levels that result from neuronal activity [47, 48, 49]. fMRI has traditionally been used for research into brain function, and more recent advances have proposed using fMRI as a biomarker for neurological disorders [50, 51]. One major fMRI paradigm for use as a biomarker is resting-state fMRI; however, certain types of analyses of resting-state data are hindered by the sampling rates of traditional fMRI acquisition schemes. For example, temporal independent components analysis (ICA) is unstable with small numbers of time samples [52]. However, temporal ICA can be a powerful way to identify overlapping brain networks, so its use is desirable in resting-state experiments. Others have suggested that there may be components of the blood oxygenation level dependent (BOLD) present at frequencies higher than 0.1 Hz [53, 54]. These studies suggest that accelerations in fMRI sampling rate beyond the standard 2-second repetition time (TR) are necessary for the types of advanced analyses that neuroscientists want to perform.

The most common current acceleration method is multiband imaging [55, 56]. In this method, multiple slices of the brain are excited simultaneously. The slices are then unaliased using the receive coil sensitivity patterns. Multiband acceleration factors are typically limited to around 8 before sensitivity coils are no longer able to undo the associated aliasing. Another limiting factor of multiband sequences is that they increase the power requirements of the RF pulse and the heat energy deposited in the subject [55]. Also, using receive coils to undo the aliasing patterns can lead to false correlations between voxels and their aliases [57].

Advanced regularization and reconstruction techniques could potentially facilitate accelerations complementary to those from multiband imaging. Low-rank models have been applied extensively in other dynamic MRI applications, such as dynamic contrast-enhanced

(DCE) and cardiac MRI [58, 12, 3]. These methods can be applied under normal scanner operating conditions and do not suffer from hardware limitations such as gradient or slew rates. Low-rank methods with strict low-rank constraints have been extended to fMRI [13, 59]. However, these strictly low-rank methods require very high ranks (on the order of 128) to successfully reconstruct fMRI time series. These high ranks potentially run close to the upper limit of those theoretically recoverable [60] and suggest that fMRI data are not truly low rank. Instead, we opt for a hybrid model closer to that of the robust PCA class [60, 3] where the data are allowed to be an additive combination of a low-rank component and a full rank component that uses a different regularity condition. The low-rank component fits many of the high-temporal frequency components in fMRI data sets such as the B_0 fluctuation due to the respiratory rhythm. We estimate this part first. Then, we estimate the full-rank component by solving a regularized optimization problem that fits the residual data. This hybrid model is able to better adapt to the many features of fMRI data sets that are more approximately full rank while still allowing for compressed-sensing style accelerations independent of the sensitivity coils.

5.2 Methods

We begin by outlining our model and reconstruction approach. Then, we describe two sets of experiments to test the efficacy of our hybrid low-rank model. The first experiment was a simulation experiment where a high temporal resolution echo-planar imaging (EPI) scan was retrospectively undersampled. We then applied the hybrid low-rank model and compared the statistical parameters of interest to those from the original full data. The second experiment consisted of a prospectively undersampled 3D sequence with a finger tapping task with a hybrid low-rank model compared to a standard multislice acquisition.

5.2.1 General Reconstruction Model

We begin by recalling that a function, $m(\boldsymbol{x}, t)$ is *partially-separable* if it can be written as

$$m(\boldsymbol{x}, t) = \sum_{r=1}^{R} u_r(\boldsymbol{x}) v_r(t) \tag{5.1}$$

where the collection of functions $u_r(\boldsymbol{x})$ and $v_r(t)$ are called *basis functions*. If we consider $m(\boldsymbol{x}, t)$ as the magnetization fluctuations in the brain that arise to cardiac and respiratory effects, we could consider an approximation with $R = 2$, where $v_1(t)$ would contain the respiratory time course and $v_2(t)$ would contain the cardiac time course. $u_1(\boldsymbol{x})$ would a

be a spatial map encompassing the entire brain (since the respiratory fluctuations affect the brain globally [61]), while $u_2(\boldsymbol{x})$ would outline the arteries and sinuses of the brain. Due to delays in blood flow reaching the periphy of the brain and higher-order respiratory effects, in reality one would need $R > 2$ to accurately model these physiological processes. If we also consider fMRI BOLD activations, then a larger R would be necessary, but since in many cases the different voxels within a brain region activate concurrently, we could still assume a small R. The sum of these effects small effects mean that although fMRI is approximately low rank, it is not actually low rank.

Due to the structure of the functional signals and physiological noise in fMRI, we opt for a hybrid model that combines low rank and full rank assumptions:

$$\mathbf{X} = \mathbf{L} + \mathbf{F}, \tag{5.2}$$

where $\mathbf{X} \in \mathbb{C}^{M \times N}$ is the fully-sampled fMRI time series, $\mathbf{L}$ is the low rank part, and $\mathbf{F}$ is the full rank part. To make (5.2) a feasible model for reconstruction, further constraints are necessary for $\mathbf{F}$. Previous dynamic MRI techniques have used sparsity assumptions [3], but it is not clear that this can be extended to the fMRI setting since fMRI data are not strictly sparse [62]. Instead, we assume that $\mathbf{F}$ has only low temporal frequencies since the BOLD response itself is slow. We implicitly assume that the physiological effects of the respiratory and cardiac rhythms are captured in $\mathbf{L}$. Since the physiological rhythms are typically large in amplitude and cover the whole brain, they should be the first components recovered with low-rank approximations. Meanwhile, the remaining functional MRI signals are better approximated as low-frequency fluctuations rather than low-rank approximations, since they exhibit a number of shifts and drifts across the brain that are not well-approximated by a strictly low-rank model.

5.2.2 Sampling Pattern

We consider sampling patterns that have one degree of freedom in the k_z dimension. Within this one degree of freedom, we densely sample the center of k-space at every frame to acquire a navigator data set. This navigator data is used to estimate the temporal basis, $\mathbf{V}$, in a fashion similar to that done in cardiac MRI [12], which is then used to reconstruct $\mathbf{L}$. Between acquisitions of the navigator data, points are acquired randomly away from the k-space center using a Poisson-disk sampling density. Fig. 5.1 shows an example of this sampling pattern. We chose this pattern since it prevents clumping of samples, which we

[1] Note that our use of $\mathbf{F}$ here denotes a full-rank matrix rather than a Fourier encoder as in Chapters 3 and 4.

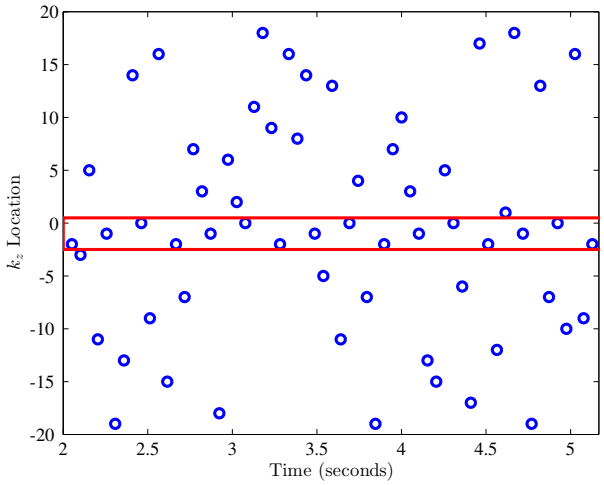


Figure 5.1: Plot of which k_z samples are acquired in the suggested sampling pattern.

believe is desirable both from the standpoint of reconstructing with multiple receive coils (which are smooth and thus only interpolate over small distances in k-space) and from the standpoint of evenly covering k-space across time frames.

5.2.3 Reconstruction Approach

Based on the model in (5.2), we formulate a two-stage reconstruction approach that blends some of the ideas from previous low-rank methods. First, we estimate a temporal basis, $\mathbf{V}$, from data sampled at a high temporal frequency at the center of k-space. Then, we use this temporal basis to estimate a spatial basis, $\mathbf{U}$, and set $\mathbf{L} = \mathbf{UV}$. We use the residual of this process to reconstruct $\mathbf{F}$ using a different constraint. Defining $\mathbf{u} = \text{vec}(\mathbf{U})$, $\mathbf{f} = \text{vec}(\mathbf{F})$ and $\mathbf{l} = \text{vec}(\mathbf{L})$, this process can be mathematically formulated as:

1. Estimate R-component $\mathbf{V}$ using densely-sampled center of k-space.
2. $\hat{\mathbf{u}} = \underset{\mathbf{u}}{\text{argmin}} \, \frac{1}{2} \left\| \mathbf{y} - \mathbf{A}_{\mathbf{V}} \mathbf{u} \right\|_2^2$
3. $\hat{\mathbf{L}} = \hat{\mathbf{U}}\hat{\mathbf{V}}$
4. $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{A}_{\mathbf{V}} \mathbf{u}$
5. Calculate $\hat{\mathbf{f}}$ from $\tilde{\mathbf{y}}$ using a different constraint.
6. $\hat{\mathbf{X}} = \hat{\mathbf{L}} + \hat{\mathbf{F}}$

where $\mathbf{y}$ is the sampled k-space data and $\mathbf{A_V}$ is a system matrix incorporating the effects of field inhomogeneity [16, 17], multiple sensitivity coils [1], and the partially-separable model with temporal basis, $\mathbf{V}$ [12].

Since most of the high-frequency components are naturally high-energy and low-rank, they should be captured in the low rank estimate in Step 2. As such, the residual signal should be dominated by noise and low-frequency BOLD effects. We examined three different methods for the alternative constraint in Step 5. The first (SLIDING) was a simple sliding window reconstruction from $\tilde{\mathbf{y}}$. The second (HYBRID) solved the following optimization problem:

$$\hat{\mathbf{f}} = \frac{1}{2}\left\|\bar{\mathbf{y}} - \tilde{\mathbf{A}}\right\|_2^2 + \frac{\beta}{2}\left\|\mathbf{C}_{\text{hybrid}}\mathbf{f}\right\|_2^2, \tag{5.3}$$

where $\tilde{\mathbf{A}}$ is a system matrix with the effects of field inhomogeneity and sensitivity coils with no partially-separable model and $\mathbf{C}_{\text{hybrid}}$ is a temporal finite-differencing operator. The third (SPARSE) solved the following optimization problem:

$$\hat{\mathbf{f}} = \frac{1}{2}\left\|\bar{\mathbf{y}} - \tilde{\mathbf{A}}\right\|_2^2 + \beta\left\|\mathbf{C}_{\text{sparse}}\mathbf{f}\right\|_1, \tag{5.4}$$

where $\mathbf{C}_{\text{sparse}}$ is a temporal DFT.

The first step of this procedure can be completed by first compressing the data from all parallel receive coils to one virtual coil using an SVD-based compression method [63] and then applying a standard SVD procedure to get the R-component $\mathbf{V}$. The remaining steps are least-squares problems that can be solved via routines such as conjugate gradient descent. The algorithm for step 2 was initialized with a rank-1 $\mathbf{U}$, where the first component was an image volume calculated from data that were averaged across time. Step 5 was initialized with a sliding window, gridding-based reconstruction that involved a density-compensation function [64] and a time-segmented approximation for field inhomogeneity [16, 17].

5.2.4 Simulation Experimental Methods

A high temporal resolution EPI scan with a 200 ms repetition time and a $64 \times 64 \times 3$ data matrix size was acquired with an eight-channel receive coil and a resting-state paradigm for 10 minutes. The imaging planes were selected to encompass the motor cortex. A synthetic data set was generated by taking taking 7 of the total 64 k_y k-space lines from each time frame in a fashion similar to that in Fig. 5.3. For each k_y location that we specify in the sampling pattern, k_x and k_z samples associated with that k_y location are also acquired. For reconstruction, 3 of these 7-line acquisitions were grouped together, so the temporal

sampling period of the reconstructed data was 600 ms with a 21/64 undersampling rate. The sampling pattern was designed to acquire 4 of these 21 lines near the center of k-space for every reconstruction frame. This allows us to accurately estimate the high-frequency respiratory signal (about 0.3-0.6 Hz).

After reconstruction, we applied standard fMRI processing methods. We applied the magnitude operation to all images and detrended by regressing out the first five bases of the DCT transform. Then, we applied a low-pass FIR filter (Parks-McClellan, 0.08-0.15 Hz transition band) and truncated the beginning and the end of the time series of images, removing the same number of time points as the length of the FIR filter. The final preprocessing step was convolving with a 5 mm $\times$ 5 mm Gaussian blur kernel. The average of eight voxels in the right motor cortex was used to form a "seed time course." We calculated correlation coefficients between this seed time course and the time courses of all other pixels in the imaging area, generating a map of correlation coefficients that corresponds to the motor network [65]. The resting-state motor network estimate using the proposed low-rank method applied on retrospectively undersampled data was compared to that from the original high temporal resolution data. All correlation maps used a correlation coefficient threshold of 0.6 for display.

We performed an ROC curve analysis between the three methods for the constrained reconstruction of $\mathbf{F}$ by assuming that an activation map from fully-sampled data was the ground truth. This enabled us to exactly calculate true and false positive rates as a function of activation threshold. Although we've performed an ROC curve analysis, we still chose to keep our correlation threshold at 0.6 since in prospectively undersampled settings we do not know the true false positive rate.

5.2.5 In Vivo Experimental Methods

To validate the methods on prospectively undersampled data, one subject was recruited and trained in a one-handed finger tapping task (20 seconds on, then 20 seconds off, total of 6 on/off repetitions) that is expected to activate the primary motor cortex, the sensorimotor cortex, and the supplementary motor area. For all functional scans, we used a 3D stack-of-spirals sequence. An example of a 3-platter stack-of-spirals sequence is shown Fig. 5.2. The experimental 3D sequences had 39 platters. First, we acquired calibration data for sensitivity map estimation using a 5 ms T_E spiral-out version of the 3D spiral sequence. Then, we acquired two calibration volumes for field map estimation using a spiral-in sequence with T_Es of 28 and 30 ms. The actual functional scan used a 30 ms T_E 3D spiral-in sequence with randomly-acquired k_z platters.

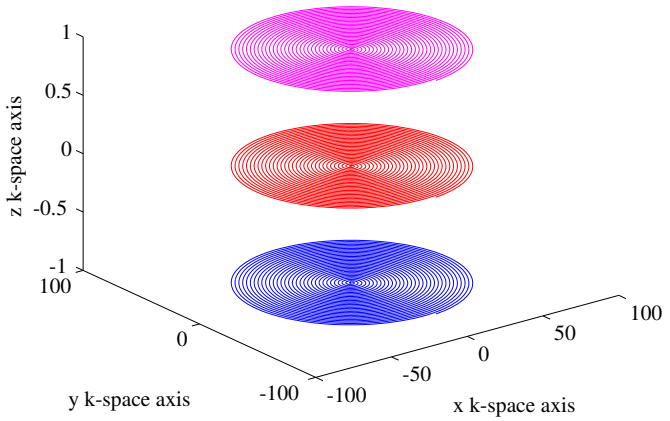


Figure 5.2: Example of a 3-platter stack-of-spirals sequence used in the *in vivo* experiments.

The sensitivity maps were estimated using a model-based method that had previously been developed [4]. The original coil array had 32 channels with 4 rings of 8 coils. To preserve the z variation, we compressed each ring of 8 coils to 2 coils using a coil compression method [63], for a final array of 8 virtual coils. The field map was estimated from the field map calibration data in a two-stage process. First, we applied a phase subtraction with the image volumes and then convolved with a $5 \times 5 \times 5$ rect function to get an initial estimate of the field map. Then, the 28 and 30 ms T_E images were reconstructed with this initial field map and a second field map was estimated from the new image volumes via phase subtraction and convolution with the same rect function. This second field map was the one used for final reconstruction of the functional time series.

During the functional scan, one of the 39 spiral k-space platters was selected and sampled at random after each excitation (as opposed to a standard 3D spiral sequence where would would coherently sample a platter). 12 platters were combined for each reconstruction frame. The sampling pattern was designed such that the three central platters were sampled for each reconstruction frame to estimate the temporal basis, $\mathbf{V}$ [12]. Fig. 5.3 shows the division of frames among the spiral platters while Fig. 5.1 outlines the data used to reconstruct $\mathbf{V}$. This reconstruction procedure resulted in an effective temporal resolution of 0.615 seconds. We reconstructed the images using the SLIDING low-rank method described in the Methods section. Although this was not the best method based on simulation experiments, it did give similar activations patterns to HYBRID and required far less computation time. In addition to the random 3D acquisition, we also acquired a 39-slice image volume with a standard 2D sequence with the same finger tapping paradigm

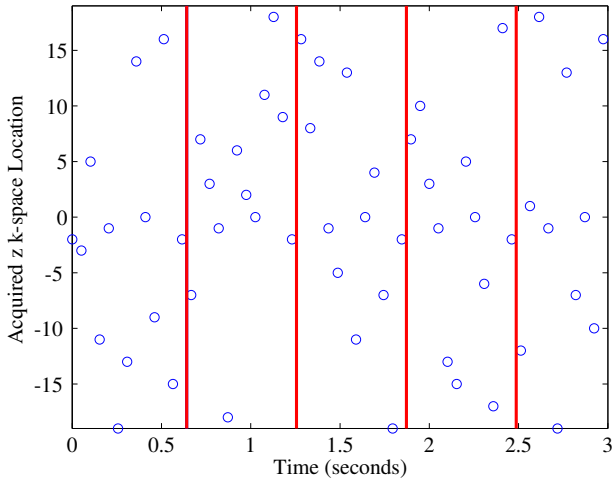


Figure 5.3: Several frames showing which k_z platters are selected in the random 3D stack-of-spirals sampling pattern. Each red line denotes a reconstruction frame boundary.

for comparison.

We processed the time series of magnitude images from both the standard 2D acquisition and the random 3D acquisition in SPM software for statistical analysis. The processing pipeline consisted of motion correction, coregistration with an anatomical acquisition, and smoothing with a $5 \times 5 \times 5$ mm Gaussian blur. Then, we applied GLM analysis and compared the resulting activation maps.

5.3 Results

5.3.1 Simulation Results

Our acquired data size for reconstruction was 64 k_x points $\times$ 21 k_y locations per frame $\times$ 3 k_z points $\times$ 1000 frames $\times$ 8 coils. The number of coefficients required to estimate $\mathbf{U}$ and $\mathbf{V}$ was $64 \times 64 \times 3 \times R$ for $\mathbf{U}$ and $R \times 1000$ for $\mathbf{V}$, where R was 30 for HYBRID, 20 for SPARSE, and 50 for SLIDING. Thus, the low-rank reconstruction was about 1.2% of the size of the original data for HYBRID, 0.8% the size of the original data for SPARSE, and 2.1% of the size of the original data for SLIDING. The most natural high-frequency signal for the low-rank model is the respiratory signal, which affects the brain globally. We use the phase of the center of k-space as a proxy for the respiration signal and plot the phase from the original, high-temporal resolution simulation data and the three reconstruction methods (HYBRID, SPARSE, and SLIDING) in Fig. 5.4. Fig. 5.4 shows that the period of the respiration signal is accurately reconstructed with all three methods. As such, we expect

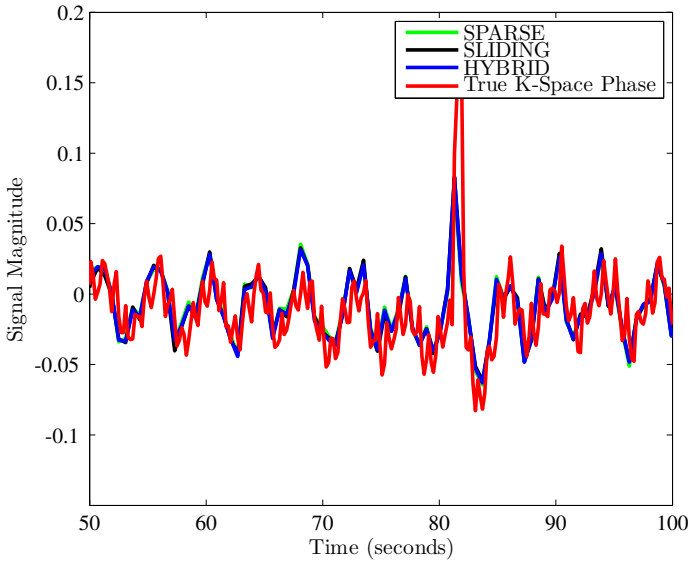


Figure 5.4: Respiration signals from original and reconstructed data.

that features that arise from the respiratory signal in the reconstruction can be removed via simple low pass filtering in a subsequent preprocessing step before functional analysis.

The results from the simulations show that all methods reasonably reproduced the activations of interest while SPARSE and SLIDING introduced more false positives into the data sets. We found in numerical experiments that the different methods required different ranks for optimal reconstructions. In our figures below we only present results from each method after optimizing the rank for that method. This resulted in HYBRID using a rank $R = 30$, SLIDING using a rank $R = 50$, and SPARSE using a rank $R = 20$. The SLIDING method required a particularly high rank since the sliding-window reconstruction of $\mathbf{F}$ was inherently less stable relative to the regularizers used in SPARSE or HYBRID. Fig. 5.5 shows the time courses from the seed region for all three methods compared to the fully-sampled case. All three methods generally recreate most of the features of the seed time course, although SPARSE and SLIDING seem to overshoot peaks and troughs more than HYBRID.

Fig 5.6 compares the resting-state motor network from the three reconstruction methods with the seed region indicated in green. All of the methods recreate the primary activations of interest, although SLIDING and SPARSE also introduce false activations outside the region of interest. HYBRID yields a more conservative activation map than SLIDING due to the temporal smoothness regularizer included in Step 5 of the proposed method. We compared the consistency of the operators in an ROC plot in Fig. 5.7. Contrasting with the results in Fig. 5.6, SPARSE was the best method according to ROC analysis. The areas

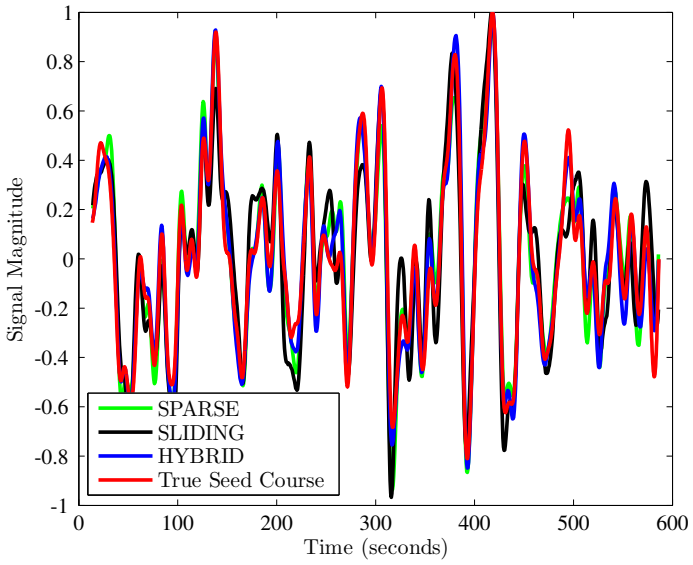
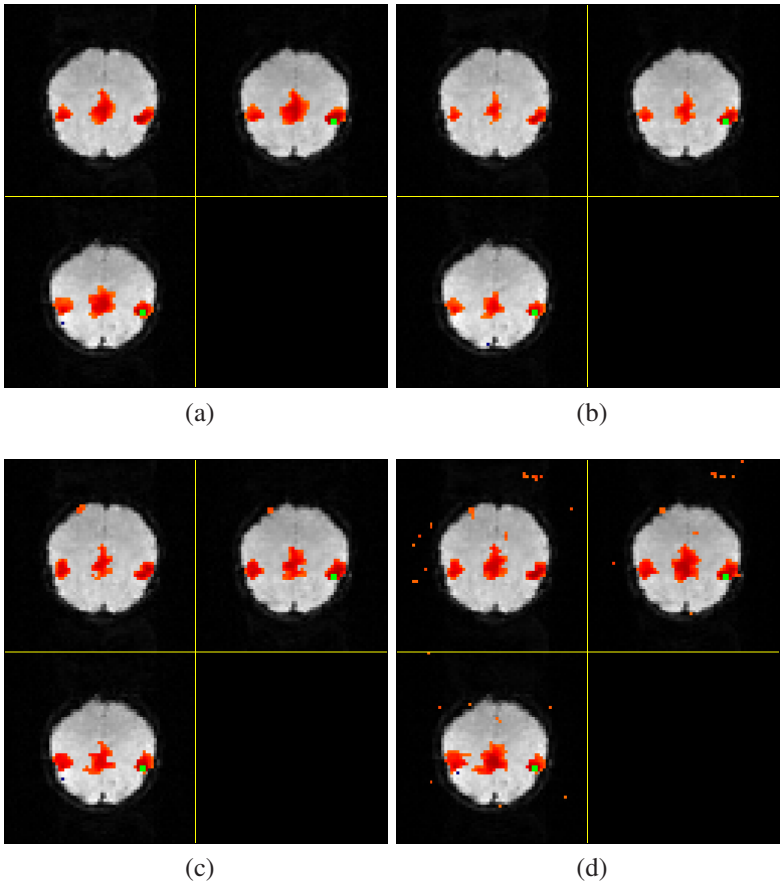


Figure 5.5: Seed time courses from the proposed reconstruction methods.

under the curve for the three methods was 0.9650 for HYBRID, 0.9672 for SLIDING, and 0.9681 for SPARSE. We chose an identical correlation threshold for the activation maps in Fig. 5.6, but the ROC curve analysis indicates that different thresholds may be necessary for the different methods.

5.3.2 In Vivo Results

Our *in vivo* experiments collected a random 3D stack-of-spirals sequence where the k_z spiral platters were selected according to the sampling pattern shown in Fig. 5.1. Our acquired data size for reconstruction was 8092 spiral points $\times$ 12 spiral platters per frame $\times$ 386 frames $\times$ 8 virtual coils. The number of coefficients required to estimate a 50-rank $\mathbf{U}$ and $\mathbf{V}$ was $64 \times 64 \times 39 \times 50$ for $\mathbf{U}$ and 50×386 for $\mathbf{V}$. Thus, the low-rank reconstruction was about 2.7% the size of the original data. As surrogate for comparing the respiratory signal, we compare the original center of k-space phase to the reconstructed in Fig. 5.8. The SLIDING method is able accurately reconstruct the center of k-space respiratory time course, which is in agreement with the results from Fig. 5.4. Fig. 5.9 compares the activation maps between the proposed 3D method and a standard multislice acquisition. The activation threshold was set to the same value (0.05 with familywise error rate) with both methods, although we note that T-scores were much higher with the standard multislice acquisition. The activation map from the SLIDING method was cleaner with fewer false activations. However, SLIDING appeared to have a different sensitivity level,



(a) (b)

(c) (d)

Figure 5.6: Examples of correlation maps from a resting-state task with different reconstructions. The seed region is indicated in green. (a) correlation map calculated from the original, high-temporal resolution data. (b) HYBRID, rank $R = 30$. (c) SLIDING, rank $R = 50$. (d) SPARSE, rank $R = 20$.

as we also observed that cerebellum activation was not significant with SLIDING even though it was significant with a standard acquisition. This can be seen in Fig. 5.10. We note that the two data sets came from different runs of the same task.

5.4 Discussion

5.4.1 Model Comparisons for Estimating the Full-Rank Component

We compared three different methods for reconstruction $\mathbf{F}$ on simulation data: one with a temporal roughness penalty, one with a temporal Fourier sparsity penalty, and that used a simple sliding window reconstruction. All three of the methods generally performed well for estimating the activations of interest. A major reason for the success of the methods is

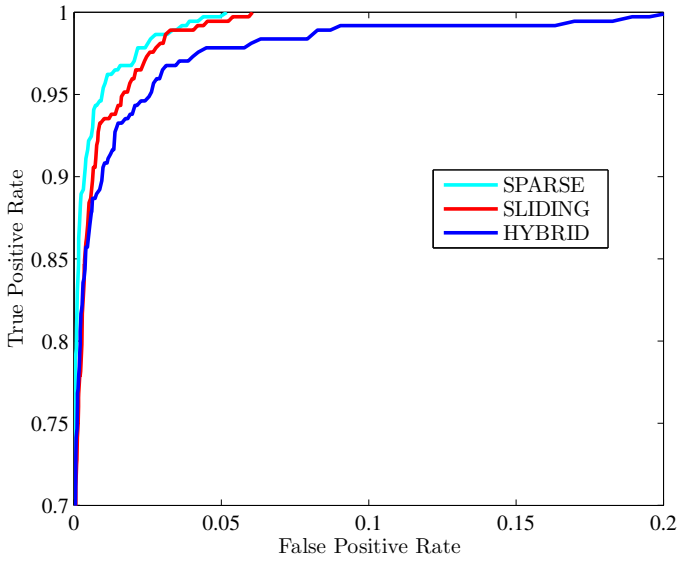


Figure 5.7: ROC Curve Comparing Reconstruction Methods.

the slowness of the BOLD hemodynamic response function (HRF). Since global physiological effects like the respiratory rhythm are reconstructed in the estimate for $\mathbf{L}$, the residual from this initial low-rank procedure should only have low frequency signals present in it.

Conversely, the introduction of false activations by the SPARSE method indicates that fMRI data are not exactly sparse. The low-rank plus sparse model appears to introduce too much structure to the data for standard resting-state analysis. On the other hand, the sliding window method may not be sophisticated enough to model all frame-to-frame temporal features, which would be a reason for the larger number of false activations in SLIDING vs. those of HYBRID.

The final result is that it is currently inconclusive what the best method is for constraining the full-rank component. It appears that SPARSE is the most sensitive method, although it more quickly inserts false activations. This later seemed to be apparent in the *in vivo* data where the reconstructions did not identify cerebellar activations and T-scores were lower in general. We are not sure why we observed losses in the cerebellum, but it could be that much of the information remaining the residual is noise and the SLIDING method primarily decreases structure by adding noisy signal to the reconstruction. This would serve to suppress activation. In the future we plan to further investigate other means for constraining the full-rank part of the reconstruction that more carefully consider the residual noise.

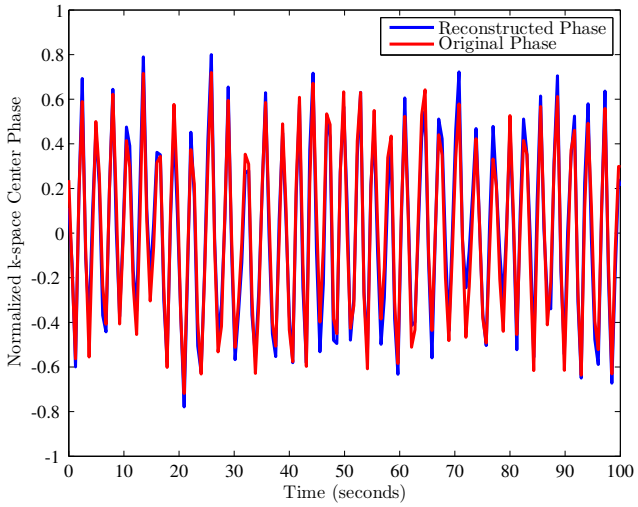


Figure 5.8: Comparison of the normalized k-space center phase between the original and reconstructed data as a surrogate for reconstructing the respiratory signal.

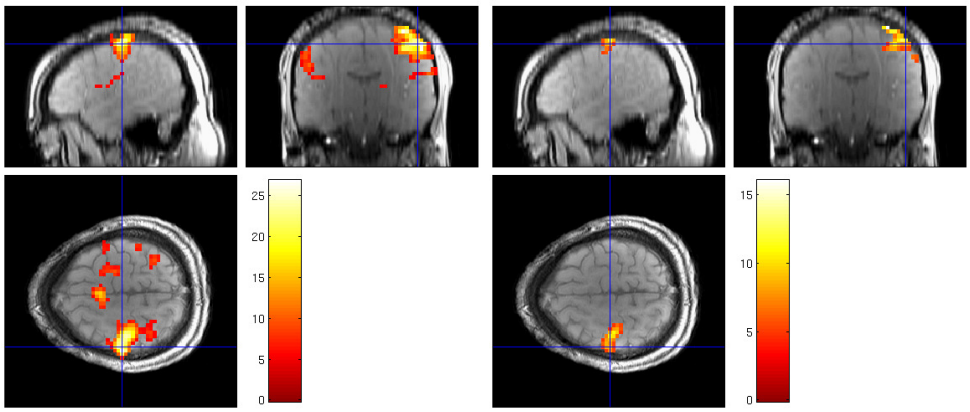
5.4.2 Development of New Statistical Analysis Methods

The use of a low-rank model effectively changes the number of degrees of freedom of the data in a nonlinear fashion. This means that standard statistical assumptions (i.e., iid Gaussian processes) are no longer satisfied. In this setting, it can be difficult to accurately assess the degrees of freedom. As Figure 5.6 shows, the false positive rate can change for different regularizers. In the future we plan to investigate the development of new statistical analysis techniques that synthesize estimates of false positive rates from the data itself. Nonparametric methods have previously been developed for fMRI [66], and we plan to build on these previous methods for our own reconstruction technique.

5.4.3 Estimating the Cardiac Signal

Here we used a strictly low-rank assumption to reconstruct the high temporal frequency effects in the fMRI signal. This assumption is very natural for the respiratory signal as shown in Fig. 5.4. It is less clear that this model is effective for the cardiac signal. The cardiac signal reaches vessels in the superior part of the brain later than those in the inferior. This timing delay effectively requires a higher rank. Although the ranks we used were well-above the simple $R = 1$ that would be in principle necessary for respiratory, it may be beneficial to develop a more rich model that explicitly handles cardiac.

Since the cardiac heartbeats are at fairly regular frequencies, one potential approach would be to use a strictly Fourier sparse assumption that assumes the cardiac signal is



(a) (b)

Figure 5.9: Examples of motor cortex activation patterns from a finger-tapping task. The cursor is centered on the primary motor cortex. (a) shows activation from a standard multislice acquisition, while (b) shows the activation from a SLIDING reconstruction.

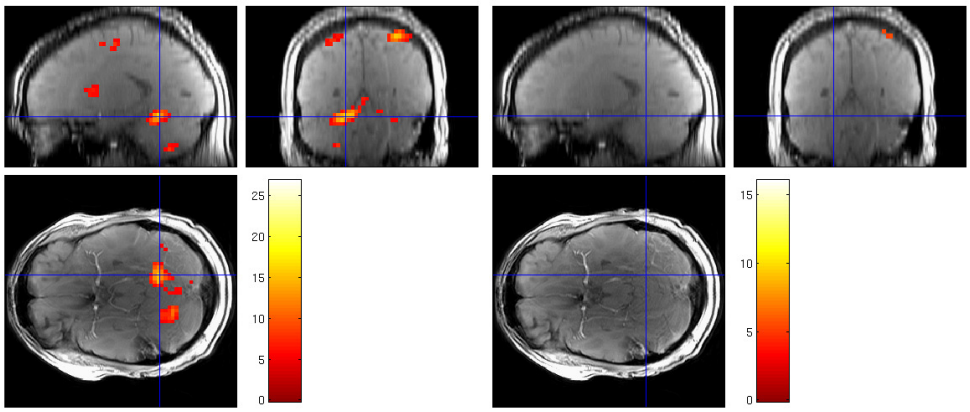
strictly localized to a small number of temporal frequencies. These frequencies could then be estimated using a greedy algorithm such as orthogonal matching pursuit (OMP) [67]. Greedy algorithms have been explored loosely in fMRI previously [68, 69] and a summary of greedy methods for MRI is given in Appendix A.

The basic idea behind a greedy algorithm is to perform a sequence of locally optimal steps. In this case, the algorithm would begin with a set of candidate frequencies. Then, the algorithm finds which frequency optimally fits the data in a least-squares sense, and this fit to the data is subtracted from the data to create the residual. The procedure is then repeated on the residual in a recursive manner until a stopping criterion is reached. For estimating the cardiac signal, one could subtract the low-rank signal from the data first and then run a greedy algorithm with candidate frequencies chosen to be common heart rate frequencies. Since cardiac frequencies rarely overlap with those for respiratory or the BOLD activations of interest, this procedure would be able to specifically target the cardiac signal.

5.4.4 Reconstruction Algorithms

Step 5 of the reconstruction process was typically the bottleneck for reconstruction speed since it requires and a forward and a backward NUFFT operation for every time point, coil, and iteration. In our reconstruction problem this requires 3072 forward and backward NUFFT operators at each iteration, a prohibitive cost. This motivates the development of more sophisticated reconstruction algorithms for this step.

The conjugate gradient algorithm currently used for step 5 is generally very good for



(a) (b)

Figure 5.10: Examples of cerebellar activation patterns from a finger-tapping task. The cursor is centered on the primary motor cortex. (a) shows activation from a standard multislice acquisition, while (b) shows the activation from a SLIDING reconstruction.

quadratic problems; however, the computation of the gradient itself is prohibitively expensive. One potential alternative is to use incremental gradient methods (also called ordered subsets for PET and X-Ray CT imaging). These methods work by using a random subset of the data to compute the gradient at each step of the algorithm. If the gradient is incoherent across the data divisions, then the gradient computed from the subset of the data will be in the same direction as the gradient from the entire data set. This approach was previously explored for splitting up MRI data by coil in parallel imaging settings [70], but in this case a more natural approach would be to take subsets of time points for the reconstruction.

5.4.5 Further Applications

Although we have presented acceleration of resting-state fMRI scans as the primary motivation for the low-rank-plus-sparse model, other applications such as real-time fMRI [71] may benefit from low-rank accelerations. These applications typically use fMRI data to present the subject or patient with information on their brain state. The subject or patient is then asked to modulate their brain state in response to this information. Low-rank accelerations would improve these techniques since they would improve the contrast-to-noise ratio. However, the low-rank methods outlined in this document typically had long reconstruction times (from 10 minutes for the simulation results to over a day for the *in vivo* results). As such, an extension to real-time fMRI applications may require alterations of current online low-rank algorithms (e.g., GRASTA [72]). These online low-rank algorithms can give performance on par with those of offline algorithms, but require far less computation

[73]. They typically require taking an alternative manifold-based approach to low-rank algorithms that allows a more direct approach to low-rank modeling. We plan to investigate such approaches in the future.

5.5 Conclusion

We have shown the efficacy of a hybrid low-rank model for use in fMRI. The hybrid model estimates the effects of high-frequency signals, such as the respiratory signal, in addition to accurately reproducing the functional BOLD signals of interest in fMRI. A tradeoff is that the more sophisticated model may be less sensitive to certain activations. In the future we plan to investigate further model advances, such as more advanced design of the high pass filter used in Step 5 of the proposed reconstruction procedure, further comparison of sampling patterns, and the incorporation of further modeling assumptions to remove the cardiac signal.

CHAPTER 6

Discussion and Future Work

This dissertation made a number of advances in the area of accelerating MRI scans. Chapter 3 showed that when careful consideration of the sensitivity maps can lead to very fast majorize-minimize algorithms for the ℓ_1-regularized problems that are present in MRI with compressed sensing. The resulting majorize-minimize algorithm takes step sizes that depend on the sum of squares of the sensitivity maps. This means that BARISTA is best-used in cases where the sensitivity maps are calculated by using a body coil normalization, since the sum of squares will be relatively flat when normalizing by a coil image sum. Normalization by a body coil may be a better way to calculate sensitivity maps in the compressed sensing setting, since the optimization problem will naturally regularize more in low-signal reasons. Conversely, coil image normalization would lead to an implicit assumption of higher signal near the center of the brain, which would decrease regularization to a level below what might be needed.

BARISTA only considered static majorizing functions based on the sensitivity maps, but it may be possible to gain further accelerations by using information from each step of the algorithm to update the majorizer. One possible way to do this would be by using line search methods [74]. Another method previously discussed was to examine whether proximal Newton methods (e.g. [36]) could also be used to speed convergence.

An alternative to the sensitivity map-based approach of BARISTA is to consider the properties of the k-space sampling trajectory. Chapter 4 gave a mathematical framework for designing a circulant majorizer based on the k-space trajectory and then designing an algorithm that considered coupling between this circulant majorizer and the frequency localization property of the wavelet transform. This gave a 2-3 factor of acceleration in 2-norm convergence in the non-Cartesian setting, although we note that the main benefit of this new approach was to speed convergence of the edges in the image. The 2-norm mostly depends on the low frequency content of the image, so the benefits of CIRCMAJ are underestimated by this metric and better-shown in Figs. 4.6 and 4.8.

One difficulty is mathematically guaranteeing that the circulant majorizer routine outlined in Chapter 4 guarantees a majorizer when also considering the sensitivity maps. For this setting, we considered sensitivity maps calculated with a sum of squares normalization. This gave system matrices that could be thought of weighted averages of non-Cartesian Fourier operators, so it seems natural that a majorizer designed solely based on the k-space trajectory would remain valid with sensitivity maps. Nonetheless, further test cases are necessary to fully-confirm this, and in the future we may investigate the use of backtracking [18] or monotonocity checks [28] to ensure stable performance of the algorithm in a wide array of applications. Another application is to standard quadratic reconstruction. Non-Cartesian SENSE reconstructions with subsampled data can be slow with the standard conjugate gradient algorithm. CIRCMAJ represents a potential acceleration here, although one would need to ensure that the conjugate gradient algorithm is properly preconditioned when making comparisons.

In Chapter 5 we developed a new model for the purpose of accelerating fMRI scans. The new model allows a richer class of signals than previous strictly low-rank methods [13]. The *in vivo* experiments also showed that this method reproduced the activations of interest in a motor task with suppressed noisy activations, although the sensitivity of the method appeared to be less than that of a standard multislice method.

Our hybrid low-rank model required very long reconstruction times (on the order of a few hours) in order to produce images. One potential way to speed the algorithm would be to use incremental gradient techniques that have been explored already in PET [75], X-Ray CT [24], and machine learning [76]. Convergence theory has been developed for accelerating these methods with Nestov momentum [76] and applied in X-Ray CT [77]. In the future we would explore taking subsets of the time points for the reconstruction of the full-rank component and applying these stochastic descent methods. Other extensions to the fMRI model would be a more explicit modeling of the cardiac effects as outlined in Section 5.4.3 and the addition of motion correction. Motion correction methods could potentially be built into the forward model and then used in a conjugate gradient algorithm. This has previously been applied in X-Ray CT with a similar cost function [78]. The motion parameters themselves could be estimated using our center-of-k-space navigator acquisition [79].

Another extension could include modification of the acquisition scheme. Currently, multiband imaging techniques excite a sequence of slices and then apply a coherent 3D readout to unalias the slices. This coherent 3D readout could be replaced with an incoherent readout as described by us for our k_z spiral samplign scheme. We could then apply the low-rank reconstruction to each collection of simulatneously-excited slices. In our lab we have

observed artifacts for multiband factors of 8 or more; this approach could potentially be used to mitigate these artifacts rather than as an explicit acceleration scheme.

APPENDIX A

Mathematical Tools For Partially-Separable Models

A.1 Partially separable models and MRI

This appendix includes material from [80, 81]. This section begins with an overview of partially separable models and how they define the structure of the system matrices $\mathbf{A}_{\mathrm{U}}$ and $\mathbf{A}_{\mathrm{V}}$ when used in the context of the MR signal equation. Most of the formalism described here is inspired by the papers of Liang [82] and Haldar [83].

A.1.1 The signal equation in the absence of multiple coils or field inhomogeneity

We begin by recalling the partially separable model for the time evolution magnetization in an MRI experiment. We consider magnetization evolutions that are piecewise constant in time to simplify notation (specifically we do this so we can use t to denote time from excitation at a later point). This assumption is implicit in other dynamic imaging studies [82, 84], although we note that magnetization evolutions over non-piecewise constant time bases are a potential avenue for future investigation. Assuming piecewise constant magnetization evolution, the partially separable model gives

$$m(\boldsymbol{x}, n) = \sum_{r=1}^{R} u_r(\boldsymbol{x}) v_r(n), \tag{A.1}$$

where n is a discrete time variable, $\{u_r(\boldsymbol{x})\}_{r=1}^{R}$ is a set of spatial basis functions, and $\{v_r(n)\}_{r=1}^{R}$ is a set of temporal basis functions. Ignoring the effects of multiple coils and

field inhomogeneity, the signal equation becomes

$$\begin{aligned} s(t,n) &= \int m(\boldsymbol{x},n)e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}}d\boldsymbol{x} \\ &= \int \sum_{r=1}^{R} u_r(\boldsymbol{x})v_r(n)e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}}d\boldsymbol{x} \\ &= \sum_{r=1}^{R} v_r(n)\int u_r(\boldsymbol{x})e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}}d\boldsymbol{x} \\ &= \sum_{r=1}^{R} v_r(n)\tilde{u}_r(\boldsymbol{k}_n(t)), \end{aligned}$$

where $\tilde{u}_r(\boldsymbol{k}_n(t))$ is the spatial Fourier transform of $u_r(\boldsymbol{x})$. The set of functions $\{\boldsymbol{k}_n(t)\}_{n=1}^{N}$ specify which Fourier samples (k-space locations) are sampled at each time point. Suppose there are M k-space locations of interest, then the samples of the signal ignoring noise can be written as

$$s(\boldsymbol{k}_m,n) = \sum_{r=1}^{R} v_r(n)\tilde{u}_r(\boldsymbol{k}_m). \tag{A.2}$$

This is possible since in the absence of field inhomogeneity and multiple coils, the partial separability is maintained between the image domain and k-space domain since the signal equation applies the spatial Fourier transform. Consider $\tilde{\mathbf{C}} \in \mathbb{C}^{M\times N}$, the matrix formed by from the elements of $s(\boldsymbol{k}_m,n)$:

$$\tilde{\mathbf{C}} = \begin{bmatrix} s(\boldsymbol{k}_1,1) & \dots & s(\boldsymbol{k}_1,N) \\ \vdots & \ddots & \vdots \\ s(\boldsymbol{k}_M,1) & \dots & s(\boldsymbol{k}_M,N) \end{bmatrix}. \tag{A.3}$$

From the partially separable model, $\text{rank}(\tilde{\mathbf{C}}) \leq R$. We use the cumbersome notation, $\tilde{\mathbf{C}}$, to differentiate from the analogous matrix, $\mathbf{C}$ that can be formed in the image domain:

$$\mathbf{C} = \begin{bmatrix} m(\boldsymbol{x}_1,1) & \dots & m(\boldsymbol{x}_1,N) \\ \vdots & \ddots & \vdots \\ m(\boldsymbol{x}_M,1) & \dots & m(\boldsymbol{x}_M,N) \end{bmatrix}. \tag{A.4}$$

In reality we cannot sample all M Fourier coefficients at every time point, so we must recover $\tilde{\mathbf{C}}$ with only a subset of the data points. This is the so-called matrix completion problem. In the absence of multiple coils or field inhomogeneity, recovery of $\tilde{\mathbf{C}}$ or $\mathbf{C}$ are equivalent problems; one can be converted into the other by a simple 2D or 3D Fast Fourier

Transform (FFT) of the columns. Note that this is only true given images that are made up of Dirac delta functions; it may be useful in the future to consider more realistic images. However, the presence of multiple coils and field inhomogeneity typically necessitates the recovery of $\mathbf{C}$ due to the fact that it is our ultimate object of interest. In the presence of field inhomogeneity and multiple coils we have the so-called "matrix sensing problem." System models for this case are presented, although full development of associated algorithms will take place over the proposed time period of the project. Since our ultimate object of interest is $\mathbf{C}$, we also have a matrix sensing problem in the absence of field inhomgeneity and multiple coils, but to simplify computation we will just consider matrix completion for $\tilde{\mathbf{C}}$ for these cases.

A.1.2 Forward and adjoint operations under the partially separable model

We will now consider the structures of the system matrices $\mathbf{A}_{\tilde{\mathbf{U}}}$ and $\mathbf{A}_{\mathbf{V}}$. We begin with the following definitions:

$$\begin{aligned}
\tilde{\mathbf{U}} &\in \mathbb{C}^{M\times R}, \tilde{\mathbf{U}}_{m,r} = \tilde{u}_r(\boldsymbol{x}_m) \\
\tilde{\mathbf{u}} &\in \mathbb{C}^{MR\times 1}, \tilde{\mathbf{u}} = \text{vec}(\tilde{\mathbf{U}}) \\
\mathbf{V} &\in \mathbb{C}^{R\times N}, \mathbf{V}_{r,n} = v_r(n) \\
\mathbf{v} &\in \mathbb{C}^{NR\times 1}, \mathbf{v} = \text{vec}(\mathbf{V}) \\
\mathbf{Y} &\in \mathbb{C}^{P\times N}, \mathbf{Y}_{p,n} = \sum_{r=1}^{R} u_r(\boldsymbol{k}_{b_{p,n}}) v_r(n) \\
\mathbf{y} &\in \mathbb{C}^{D\times 1}, \mathbf{y} = \text{vec}(\mathbf{Y}), D = PN
\end{aligned}$$

In the above definitions, $b_{p,n} \mapsto m \in [1, M]$ is an index variable that indexes the k-space samples of choice according to the set of functions $\{\boldsymbol{k}_n(t)\}_{n=1}^{N}$. Noting that in practical settings our data vector, $\mathbf{y}$ will be samples from $\tilde{\mathbf{C}}$, we define the sampling operator, $\mathcal{S}(\cdot)$, such that $\mathbf{y} = \mathcal{S}(\tilde{\mathbf{U}}\mathbf{V})$. Alternatively, $\mathbf{y}$ can be written as matrix operations on $\tilde{\mathbf{u}}$ and $\mathbf{v}$. We define $\mathbf{A}_{\mathbf{V}}$ such that $\mathbf{A}_{\mathbf{V}}\tilde{\mathbf{u}} = \mathbf{y}$ and $\mathbf{A}_{\tilde{\mathbf{U}}}\mathbf{v} = \mathbf{y}$. In practice $\mathbf{A}_{\mathbf{V}}$ and $\mathbf{A}_{\tilde{\mathbf{U}}}$ are too large to store on many computers. In addition, these matrices are sparse, so it is inefficient to implement them as actual matrices in a programming environment. As such, it is desirable to describe the forward operation and the adjoint of each one of these matrices for practical implementation purposes.

First, we will consider the matrix, $\mathbf{A}_{\mathbf{V}}$. Consider the forward operation, $\mathbf{y} = \mathbf{A}_{\mathbf{V}}\tilde{\mathbf{u}}$.

Under the definitions above, we have

$$\mathbf{Y}_{p,n} = \sum_{r=1}^{R} \tilde{u}_r(\boldsymbol{k}_{b_{p,n}}) v_r(n). \tag{A.5}$$

The adjoint, $\mathbf{A}_{\mathbf{V}}^H$, satisfies

$$\langle \mathbf{A}_{\mathbf{V}} \tilde{\mathbf{u}}, \mathbf{y} \rangle = \left\langle \tilde{\mathbf{u}}, \mathbf{A}_{\mathbf{V}}^H \mathbf{y} \right\rangle. \tag{A.6}$$

Using the relations in (A.5) and (A.6), we can write

$$\begin{aligned}
\langle \mathbf{A}_{\mathbf{V}} \tilde{\mathbf{u}}, \mathbf{y} \rangle &= \sum_{n=1}^{N} \sum_{p=1}^{P} (\sum_{r=1}^{R} \tilde{u}(\boldsymbol{k}_{b_{p,n}}) v_r(n)) (\mathbf{Y}_{p,n})^* \\
&= \sum_{n=1}^{N} \sum_{p=1}^{P} (\sum_{r=1}^{R} v_r(n) (\sum_{m=1}^{M} \tilde{u}_r(\boldsymbol{k}_m) \mathbb{1}_{\{b_{p,n}=m\}})) (\mathbf{Y}_{p,n})^* \\
&= \sum_{m=1}^{M} \sum_{r=1}^{R} \tilde{u}_r(\boldsymbol{k}_m) (\sum_{n=1}^{N} \sum_{p=1}^{P} v_r^*(n) \mathbb{1}_{\{b_{p,n}=m\}} \mathbf{Y}_{p,n})^* \\
&= \left\langle \tilde{\mathbf{u}}, \mathbf{A}_{\mathbf{V}}^H \mathbf{y} \right\rangle,
\end{aligned}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function and $(\cdot)^*$ denotes the complex conjugate operation. So the adjoint operation applied to $\mathbf{y}$, $\mathbf{A}_{\mathbf{V}}^H \mathbf{y}$, can be computed as $\sum_{n=1}^{N} \sum_{p=1}^{P} v_r^*(n) \mathbb{1}_{\{b_{p,n}=m\}} \mathbf{Y}_{p,n}$, which is essentially what MATLAB's `accumarray` command does. This is useful for the implementation of iterative algorithms.

An analogous relation exists for $\mathbf{A}_{\tilde{\mathbf{U}}}$. Consider the forward operation, $\mathbf{y} = \mathbf{A}_{\tilde{\mathbf{U}}} \mathbf{v}$. This can also be written as

$$\mathbf{Y}_{p,n} = \sum_{r=1}^{R} \tilde{u}_r(\mathbf{k}_{b_{p,n}}) v_r(n). \tag{A.7}$$

The adjoint operation, $\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{y}$ satisfies

$$\langle \mathbf{A}_{\tilde{\mathbf{U}}} \mathbf{v}, \mathbf{y} \rangle = \left\langle \mathbf{v}, \mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{y} \right\rangle. \tag{A.8}$$

Using the relations in (A.7) and (A.8), we can write

$$\begin{aligned}\langle \mathbf{A}_{\tilde{\mathbf{U}}}, \mathbf{y} \rangle &= \sum_{n=1}^{N} \sum_{p=1}^{P} (\sum_{r=1}^{R} \tilde{u}_r(\mathbf{k}_{b_{p,n}}) v_r(n)) (\mathbf{Y}_{p,n})^* \\ &= \sum_{n=1}^{N} \sum_{r=1}^{R} v_r(n) (\sum_{p=1}^{P} \tilde{u}_r^*(\mathbf{k}_{b_{p,n}}) \mathbf{Y}_{p,n})^* \\ &= \left\langle \mathbf{v}, \mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{y} \right\rangle,\end{aligned}$$

so the adjoint operation applied to $\mathbf{y}$, $\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{y}$, can be computed as $\sum_{p=1}^{P} \tilde{u}_r^*(\mathbf{k}_{b_{p,n}}) \mathbf{Y}_{p,n}$.

A.1.3 Matrix structure in least squares problems under the partially separable model

We will now consider least squares problems involving the matrices $\mathbf{A}_{\tilde{\mathbf{U}}}$ and $\mathbf{A}_{\mathbf{V}}$. We will begin with the following minimization problem:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{\tilde{\mathbf{U}}} \mathbf{v}\|_2^2 . \tag{A.9}$$

From the normal equations, (A.9) has the solution,

$$\hat{\mathbf{v}} = (\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{A}_{\tilde{\mathbf{U}}})^{-1} \mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{y}, \tag{A.10}$$

assuming that $\mathbf{A}_{\tilde{\mathbf{U}}}$ has full column rank such that $\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{A}_{\tilde{\mathbf{U}}}$ is invertible. Under the definitions in Section A.1.2, $\mathbf{A}_{\tilde{\mathbf{U}}}$ has a block diagonal structure with N blocks of size $P \times R$. This structure provides a useful method for solving the minimization problem in (A.9). Since $\mathbf{A}_{\tilde{\mathbf{U}}}$ is block diagonal, $\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{A}_{\tilde{\mathbf{U}}}$ is also block diagonal with N blocks of size $R \times R$. Thus, the inverse of $\mathbf{A}_{\tilde{\mathbf{U}}}^H \mathbf{A}_{\tilde{\mathbf{U}}}$ can be computed by finding the inverse of each individual block. This technique is amenable to parallelization and leads to fast, accurate solutions of the minimization problem in (A.9).

An analogous case exists for the matrix, $\mathbf{A}_{\mathbf{V}}$, although it requires a permutation and a definition of a new matrix. This is described in [85]. Let G_m be the number of times k-space location $\boldsymbol{k}_m$ is sampled such that $G_m = \sum_{n=1}^{N} \sum_{p=1}^{P} \mathbb{1}_{\{b_{p,n}=m\}}$. Define $\mathbf{W}_m \in \mathbb{N}^{G_m \times N}$ as a matrix with 1s and 0 such that $\mathbf{W}_m$ has a 1 if the k-space location, $\boldsymbol{k}_m$ is sampled at time n and a 0 otherwise. We avoid more formally defining this $\mathbf{W}_m$ since it would require even more notation for low marginal benefit. Now consider the permutation of $\mathbf{A}_{\mathbf{V}}$ that arises when we re-define $\tilde{\mathbf{u}} = \text{vec}(\tilde{\mathbf{U}}^T)$, that is, we "group" all R components corresponding to

k-space location m together in the $\tilde{\mathbf{u}}$ vector. Consider a similar permutation on $\mathbf{y}$, where the entries in $\mathbf{y}$ are ordered first by k-space location and then by time. Lastly, let $\mathbf{v}_r \in \mathbb{C}^{N\times 1}$ correspond to the vector of samples obtained from $v_r(n)$. Then we have

$$\mathbf{A}_{\mathbf{V}} = \begin{bmatrix} \mathbf{W}_1\mathbf{v}_1 & \dots & \mathbf{W}_1\mathbf{v}_R & & & & & 0 \\ & & & \mathbf{W}_2\mathbf{v}_1 & \dots & \mathbf{W}_2\mathbf{v}_R & & \\ & & & & & & \ddots & \\ 0 & & & & & & \mathbf{W}_M\mathbf{v}_1 & \dots & \mathbf{W}_M\mathbf{v}_R \end{bmatrix}. \tag{A.11}$$

The matrix, $\mathbf{A}_{\mathbf{V}}^H\mathbf{A}_{\mathbf{V}}$, now has a block diagonal structure with M blocks. The mth block has the form,

$$[\mathbf{A}_{\mathbf{V}}^H\mathbf{A}_{\mathbf{V}}]_{\text{block } m} = \begin{bmatrix} \mathbf{v}_1^H\mathbf{W}_m^H\mathbf{W}_m\mathbf{v}_1 & \dots & \mathbf{v}_1^H\mathbf{W}_m^H\mathbf{W}_m\mathbf{v}_R \\ \vdots & \ddots & \vdots \\ \mathbf{v}_R^H\mathbf{W}_m^H\mathbf{W}_m\mathbf{v}_1 & \dots & \mathbf{v}_R^H\mathbf{W}_m^H\mathbf{W}_m\mathbf{v}_R \end{bmatrix}. \tag{A.12}$$

So the least squares minimization problem,

$$\hat{\tilde{\mathbf{u}}} = \underset{\tilde{\mathbf{u}}}{\text{argmin}}\, \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}_{\mathbf{V}}\tilde{\mathbf{u}}\right\|_2^2, \tag{A.13}$$

can be solved via block inversion as was the case previously with $\mathbf{A}_{\tilde{\mathbf{U}}}$.

A.1.4 Partially separable models in the presence of field inhomogeneity and multiple coils

When field inhomogeneity and multiple coils are considered, the signal equation at the lth coil at time t from the nth excitation becomes

$$s_l(t,n) = \int h_l(\boldsymbol{x})m(\boldsymbol{x},n)e^{-i\omega(\boldsymbol{x})t}e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}}d\boldsymbol{x}, \tag{A.14}$$

where $h_l(\boldsymbol{x})$ is the sensitivity map for the lth coil and $\omega(\boldsymbol{x})$ is the off-resonance field map. Under the partially separable model, we have

$$\begin{aligned} s_l(t,n) &= \int h_l(\boldsymbol{x}) m(\boldsymbol{x},n) e^{-i\omega(\boldsymbol{x})t} e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}} d\boldsymbol{x} \\ &= \int h_l(\boldsymbol{x}) \sum_{r=1}^{R} u_r(\boldsymbol{x}) v_r(n) e^{-i\omega(\boldsymbol{x})t} e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}} d\boldsymbol{x} \\ &= \sum_{r=1}^{R} v_r(n) \int h_l(\boldsymbol{x}) u_r(\boldsymbol{x}) e^{-i\omega(\boldsymbol{x})t} e^{-i\boldsymbol{k}_n(t)\cdot\boldsymbol{x}} d\boldsymbol{x} \\ &= \sum_{r=1}^{R} v_r(n) \tilde{u}_r(\boldsymbol{k}_n(t),t) \end{aligned}$$

where the set of basis functions $\{\tilde{u}_r(\boldsymbol{k}_n(t),t)\}_{r=1}^{R}$ are no longer simply the Fourier transforms of the basis functions, $\{u_r(\boldsymbol{x})\}$, so in general partial separability is not preserved between the image domain and the domain of the signal in these settings. Under certain conditions, partial separability can be maintained. One such condition would be when the same k-space location is not sampled at two different post excitation time points throughout the experiment. This condition also requires an assumption that the field map, $\omega(\boldsymbol{x})$, does not change during the experiment. To otherwise take advantage of partial separability, image reconstruction problems in these settings must be formulated in the image domain.

A.2 Greedy algorithms with partially separable models

In this section we will consider the greedy pursuit algorithm presented in this work. We will begin with the simple cases without multiple coils or field inhomogeneity outlined in Section A.1.1 and then note possible extensions and relevant literature for formulating these extensions.

A.2.1 DCS-SOMP introduction

We assume that the temporal basis vectors $\{v_r(n)\}_{r=1}^{R}$ are elements from a dictionary, Φ. In another way, it only requires R elements from Φ to completely describe the time evolution of all fMRI images. Given this assumption, the reconstruction problem can be formulated as a minimization problem. Let Λ be an index set that indexes elements from Φ and let the $|\cdot|$ operator denote the cardinality of a set. Then, we would like to solve the following

problem:

$$\hat{\Lambda} = \underset{\Lambda \subset \Omega}{\operatorname{argmin}} \, |\Lambda| \;\; \text{such that} \;\; \left\| \mathbf{y} - \mathcal{S}(\hat{\tilde{\mathbf{C}}}_{\Lambda}) \right\|_2^2 < \epsilon \;\; . \tag{A.15}$$

At the end of solving this problem, we would form $\hat{\tilde{\mathbf{C}}}_{\hat{\Lambda}}$, the optimal least squares approximation to $\tilde{\mathbf{C}}$ using the elements from Φ selected by $\hat{\Lambda}$. The problem described by (A.15) is generally an NP-hard problem, so we resort to greedy methods. The greedy algorithm described here is based on orthogonal matching pursuit (OMP) [86, 67]. Many OMP algorithms are designed for a single measurement vector. Under the conditions of multiple measurement vectors, if these vectors are assumed to be jointly sparse then simultaneous orthogonal matching pursuit can be used [87]. If the dictionary changes for each measurement vector, DCS-SOMP can be used [88]. Algorithm A.1 describes a DCS-SOMP algorithm for dynamic MR image reconstruction. In Algorithm A.1, $\mathbf{R}_j$ denotes the resid-

Figure A.1: DCS-SOMP

1: initialize $\Lambda_0 = \emptyset, \hat{\tilde{\mathbf{C}}}_0 = \mathbf{0}, \mathbf{S} = \mathbf{0}$
2: **while** $\|\mathbf{R}_j\|_F^2 > \epsilon$ **do**
3: $\quad \lambda_j = \underset{\lambda \in \Omega, \lambda \notin \Lambda_j}{\operatorname{argmax}} \sum_{m=1}^{M} \frac{|\langle \mathbf{R}_{j-1}\mathbf{e}_m, \phi_\lambda \rangle_{\mathbf{W}_m}|^2}{\langle \phi_\lambda, \phi_\lambda \rangle_{\mathbf{W}_m^H \mathbf{W}_m}}$
4: $\quad \Lambda_j = \Lambda_{j-1} \cup \{\lambda_j\}$
5: $\quad \mathbf{V}_j = \Phi_{\Lambda_j}$
6: $\quad \tilde{\mathbf{u}}_j = \underset{\tilde{\mathbf{u}}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{y} - \mathbf{A}_{\mathbf{V}_j} \tilde{\mathbf{u}} \right\|_2^2$
7: $\quad \hat{\tilde{\mathbf{C}}}_j = \tilde{\mathbf{U}}_j \mathbf{V}_j$
8: $\quad \mathbf{R}_j = \mathbf{Y} - \mathcal{S}(\hat{\tilde{\mathbf{C}}}_j)$
9: $\quad j = j + 1$
10: **end while**

ual at iteration j, λ denotes a single element from an index set, Ω is the set of all possible indices for elements from Φ, ϕ_λ denotes a single element from Φ indexed by λ, and Φ_Λ returns all the elements from Φ indexed by Λ. Intuitively, at each iteration DCS-SOMP finds the element from Φ that best spans the residual, $\mathbf{R}_j$. "Best" in this sense means "minimizes the mean squared error of approximation," and in this case is accomplished by taking a maximum over the inner products in Step 3. The residual is updated in Step 6, which can be solved using the fast methods described in Section A.1.3. A proof that the maximization over the inner products finds the optimal element from Φ is given in the next section.

A.2.2 Finding the Optimal Element From the Dictionary

More notation is necessary for this section. Let $\mathbf{r}_j = \text{vec}(\mathbf{R}_j)$, and we will assume that it is ordered as described in the second part of Section A.1.3. At the jth step, we would like to solve the following problem:

$$\lambda_j = \arg \min_{\lambda \in \Omega, \lambda \notin \Lambda_j} \min_{\tilde{\mathbf{u}}} \frac{1}{2} \|\mathbf{r}_j - \mathbf{A}_{\mathbf{V}_\lambda} \tilde{\mathbf{u}}\|_2^2 \tag{A.16}$$

We use the normal equations to find the solution to the inner minimization problem such that $\tilde{\mathbf{u}} = (\mathbf{A}_{\mathbf{V}}^H \mathbf{A}_{\mathbf{V}})^{-1} \mathbf{A}_{\mathbf{V}}^H \mathbf{r}_j$. Substituting for $\tilde{\mathbf{u}}$ and expanding for the second minimization gives

$$\begin{aligned} \lambda_j = \underset{\lambda \in \Omega, \lambda \notin \Lambda_j}{\text{argmin}} \; & \frac{1}{2} \mathbf{r}_j^H \mathbf{r}_j - \mathbf{r}_j^H \mathbf{A}_{\mathbf{V}_\lambda} (\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda})^{-1} \mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{r}_j \\ & + \frac{1}{2} \mathbf{r}_j^H \mathbf{A}_{\mathbf{V}_\lambda} (\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda})^{-1} \mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda} (\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda})^{-1} \mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{r}_j. \end{aligned} \tag{A.17}$$

By recognizing that the minimization does not depend on $\mathbf{s}_j^H \mathbf{s}_j$ and applying the appropriate simplifications, we can convert the problem to

$$\lambda_j = \underset{\lambda \in \Omega, \lambda \notin \Lambda_j}{\text{argmax}} \; \mathbf{r}_j^H \mathbf{A}_{\mathbf{V}_\lambda} (\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda})^{-1} \mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{r}_j. \tag{A.18}$$

Since λ selects a single element from Ω, the matrices, $\{\mathbf{A}_{\mathbf{V}_\lambda}\}_{\lambda \in \Omega, \lambda \notin \Lambda_j}$ have a special form. Specifically, they have the form described in the second part of Section A.1.3 for the case where $R = 1$ and $\mathbf{v}_1 = \phi_\lambda$. This means that the term, $\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{A}_{\mathbf{V}_\lambda}$, is diagonal and the term $\mathbf{A}_{\mathbf{V}_\lambda}^H \mathbf{r}_j$ consists of inner products between parts of the residual and the element of the dictionary selected by λ. Applying these definitions gives the final form for this optimization step:

$$\lambda_j = \underset{\lambda \in \Omega, \lambda \notin \Lambda_j}{\text{argmax}} \sum_{m=1}^{M} \frac{|\langle \mathbf{R}_{j-1} \mathbf{e}_m, \phi_\lambda \rangle_{\mathbf{W}_m}|^2}{\langle \phi_\lambda, \phi_\lambda \rangle_{\mathbf{W}_m^H \mathbf{W}_m}}. \tag{A.19}$$

This formulation is useful particularly for the DFT dictionary, since the inner products can be computed using FFTs and the normalization factor is simply the number of times k-space location m is sampled during the experiment.

A.2.3 DCS-SOMP with field inhomogeneity and multiple coils

In the presence of field inhomogeneity the signal equation takes on the form described in Section A.1.4. However, as shown in Section A.1.4 the field inhomogeneity and multiple coils do not change the fact that the k-space samples still lie in the span of a small set

of complex exponentials. This would suggest that we can leave Step 3 of the DCS-SOMP algorithm unchanged; however, this step was derived assuming some sort of "grouping" being applied to the k-space samples of $\{\tilde{u}_r(\boldsymbol{k}_n(t))\}_{r=1}^{R}$ that can be done when partial separability is preserved between the image and signal domains, but in general $\boldsymbol{k}_n(t)$ can change with each time point we cannot assume preservation of partial separability for the proof in Section A.2.2 to show that the optimization over the inner products yields the optimal complex exponential at iteration j. However, if the alterations made by coil sensitivities and field inhomogeneity are small, then the inner product optimization may still yield the correct set of complex exponentials. In addition, the approximation routine in Step 6 of the DCS-SOMP algorithm can be done while considering field inhomogeneity and multiple coils. For now, we will ignore field inhomogeneity in this step while accounting for the multiple coils, with the possibility of considering more sophisticated methods of selecting the basis in the future.

Considering the assumptions outlined above, the only new consideration is the approximation routine in Step 6. We will rely on the results of Pruessman [1] to modify the system matrix and solve this step. Essentially, the presence of multiple coils necessitates a new system matrix, $\mathbf{A}$. This matrix contains elements of coil sensitivities and the partially separable model. The final approximation would also include field inhomogeneity; we will rely on the results of Sutton [16] for solving this step. As shown in Section A.1.4, expansions along each of these variables can be permuted in different orders. When using iterative algorithms such as conjugate gradient to solve Step 6, the system matrix should be designed such that the order of operations is optimized for speed. Alternatively, since reconstructions considering field inhomogeneity, partial separability, and coil sensitivities separately are computationally more simple, a variable splitting approach may yield significant speed-up time for this step [7]. We will explore using both conjugate gradient and variable splitting algorithms for optimizing the speed of this step.

APPENDIX B

Supplementary Results for BARISTA

This appendix includes material from [8]. This section shows convergence results from a numerical brain phantom, a real breast phantom, and an American College of Radiology (ACR) phantom to supplement those shown in "Fast Parallel MR Image Reconstruction via B1-Based, Adaptive Restart, Iterative Soft Thresholding Algorithms (BARISTA)," *IEEE Transactions on Medical Imaging*.

B.1 Experiment with synthetic data

The synthetic data set was generated by downloading an image from the BrainWeb data base [89] and simulating eight receive sensitivity coils using the methods from [45]. The data were sampled using a Poisson-disc sampling pattern with a densely sampled 32-by-32 center similar to Fig. 6b in the main paper body. Sensitivity coil profiles were estimated from noisy, simulated data using a central 32-by-32 block of kspace via the methods in [4]. About 20% of the fully-sampled data were used for reconstruction. Reconstructions were then run using orthogonal Haar wavelets, orthogonal Daubechies D4 wavelets, anisotropic total variation, and 2-level undecimated Haar wavelets as regularizers. Regularization parameters were chosen to give visually appealing images. Fig. B.1 shows the true simulation image and reconstructions using orthogonal Haar (Fig. B.1c), orthogonal Daubechies D4 (Fig. B.1d), total variation (Fig. B.1e), and 2-level undecimated Haar wavelet (Fig. B.1f) regularizers. Fig. B.1a shows the sum of squared absolute values of the sensitivity coils. We show the sum of squares instead of the square root sum of squares since the sum of squares is more relevant to the behavior of $\mathbf{A}^H\mathbf{A}$. Fig. B.2 shows convergence results for the four regularizers on this synthetic data set. We plot the normalized residual, $\xi(k)$, defined as

$$\xi(k) = 20\log_{10}\left(\frac{\left\|\mathbf{x}^{(k)}-\mathbf{x}^{(\infty)}\right\|_2}{\left\|\mathbf{x}^{(\infty)}\right\|_2}\right), \tag{B.1}$$

where $\mathbf{x}^{(\infty)}$ is a "converged solution" obtained by running many thousands of iterations of the AL-based methods.

Figures B.2a and B.2b are similar to those in Figures 2c and 2d in [19]. Figures B.2c and B.2d show some acceleration of AL methods with optimal tuning in early iterations with anisotropic total variation and 2-level undecimated Haar wavelet regularizers, but BARISTA reaches single precision faster than the AL-based methods. BARISTA used the exact same parameters as in the *in vivo* experiments in [8], whereas extensive time was spent re-tuning the AL-based methods to achieve the results shown. Of particular interest is the AL method with the dynamic μ updates with parameters from [31], which stalls in early iterations. The AL method with dynamic μ updates was initialized with one of the good, manually-tuned parameters, but the μ update heuristics quickly increased the magnitude of μ. This data set required rather small μ parameters compared to the other data sets, which may be due to the fact that the true image and SENSE maps were orders of magnitude different in intensity. Such issues could be mitigated by always ensuring appropriate normalizations of the SENSE maps and data, but the convergence speed results here also show that BARISTA is robust when such normalizations are not performed.

B.2 Experiment with breast phantom data

We also obtained a SENSE breast phantom imaging data set for numerical experiments that had been used for a previous paper [4]. The breast phantom consisted of two containers plastered with vegetable shortening and filled with "Super Stuff" bolus material (Radiation Products Design Inc. Albertville, MN). Four surface coils were used in the experiment. The data were sampled using a Poisson-disc sampling pattern similar to that in Fig. 6b in [8] with a densely-sampled 32-by-32 center in the orthogonal wavelets case and a 32-by-16 center in the anisotropic total variation and undecimated Haar wavelets case. The sensitivity coil profiles were estimated using the data from the densely-sampled center using the methods described in [4]. About 20% of the fully-sampled data were used for reconstruction. Fig. B.3 shows reconstructed images of real breast phantom data for each of the four regularizers of interest (orthogonal Haar, orthogonal Daubechies D4, anisotropic total variation, and 2-level undecimated Haar wavelets). Figure B.4 shows the convergence speed results for the breast phantom imaging data set. Figures B.4a and B.4b are similar to those in Figures 2c and 2d in [19]. BARISTA was the fastest method for all four of the regularizers of interest for this data set.

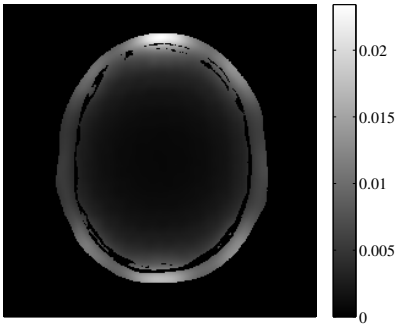
B.3 Experiment with ACR phantom data

We also obtained a new SENSE ACR phantom data set for numerical experiments. The phantom was scanned with a GE MR 750 scanner with a 32-channel receive array (Nova Medical Group). The phantom was scanned with a 256-by-256 grid and then retrospectively downsampled with a Poisson-disc sampling pattern similar to that in Fig. 6b in [8] with a densely-sampled 32-by-32 center. The sensitivity coil profiles were estimated using the data from the densely-sampled center using the methods described in [4]. About 20% of the fully-sampled data were used for reconstruction. Since our previous experiments indicated that BARISTA and AL methods were converging to the same solution, in this case we calculated $\mathbf{x}^{(\infty)}$ with BARISTA to minimize the time to run the numerical experiments. The SENSE maps from this experiment and the reconstructed $\mathbf{x}^{(\infty)}$ for each regularizer are shown in Fig. B.5.

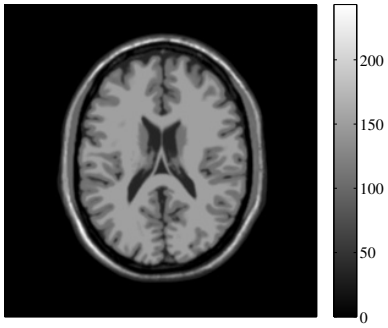
Fig. B.6 shows convergence speed results for the ACR phantom data set. Figures B.6a and B.6b are similar to those in Figures 2c and 2d in [19]. BARISTA was the fastest method for all four of the regularizers of interest for this data set.

B.4 Experiments with masking

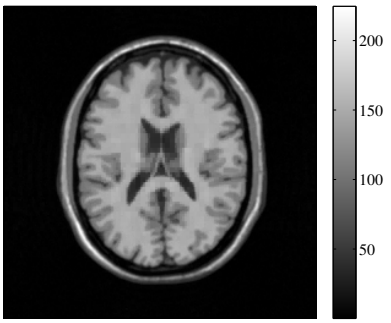
We also compared convergence speed between BARISTA and the AL-based methods with image-domain masks for total variation and 2-level undecimated Haar wavelet regularizers for each data set. "Converged" images were calculated using BARISTA. All other experimental parameters were the same as run previously. Masks were calculated by first thresholding the $\mathbf{x}^{(\infty)}$ image magnitude at 10% of the maximum magnitude, then performing a 10-by-10 dilation of the mask (20-by-20 dilation for the ACR phantom). Our goal is not to propose a method for doing masked SENSE, but to simply examine algorithm performance with a mask. Examples of image-domain masks for each data set are shown in Fig. B.7. Fig. B.8 shows convergence results using an image domain mask for each of the four data sets. Figs. B.8a and B.8b are similar to Fig. 10 in the main paper body. Figs. B.8c and B.8d are similar to B.2c and B.2d, respectively. Figs. B.8e and B.8f are similar to B.4c and B.4d, respectively. Figs. B.8g and B.8h are similar to B.6c and B.6d, respectively. Also of note is that BARISTA converged at least twice as fast when using an image domain mask, so masking appears to be desirable both from an image quality and a convergence speed perspective.



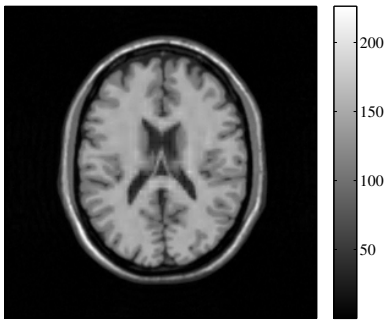
(a) SENSE Map Sum of Squares



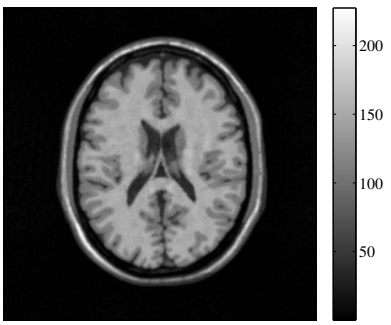
(b) True Image



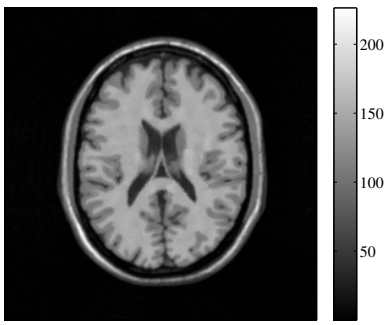
(c) Orthogonal Haar Regularizer



(d) Orthogonal D4 Regularizer

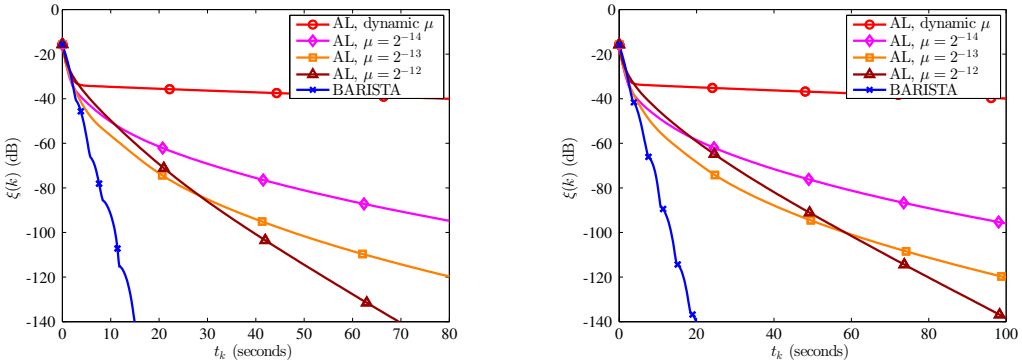


(e) Total Variation Regularizer



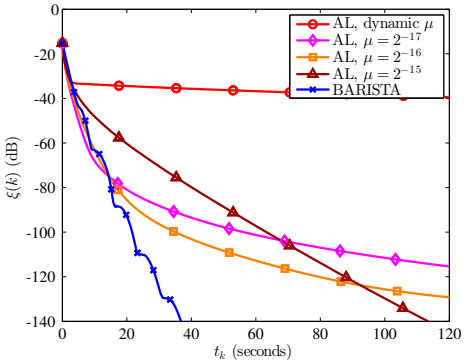
(f) Undecimated Haar Regularizer

Figure B.1: Examples of true and reconstructed images from the simulated data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the numerical phantom experiment with areas outside the brain masked for presentation. (b) The ground truth $\mathbf{x}$ used for the numerical simulations. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization, NRMSE of 2.6%. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization, NRMSE of 2.6%. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization, NRMSE of 2.9%. (f) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization, NRMSE of 2.7%.

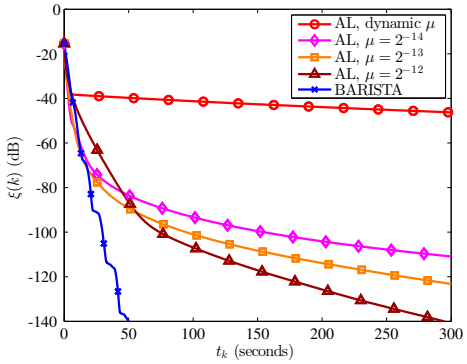


(a) Orthogonal Haar Regularizer

(b) Orthogonal D4 Regularizer

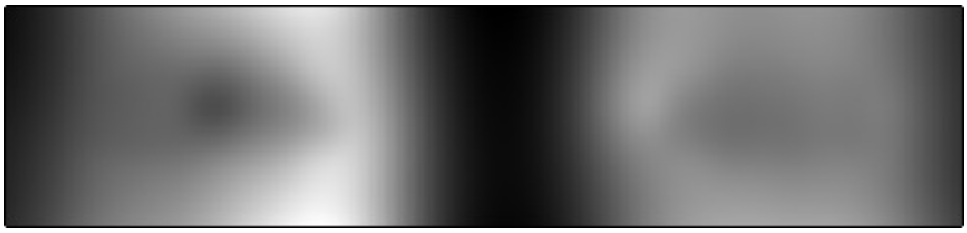


(c) Total Variation Regularizer



(d) Undecimated Haar Regularizer

Figure B.2: Convergence speed results from fully synthetic data. Markers are placed at 30-iteration intervals for all algorithms. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.



(a) SENSE Map Sum of Squares

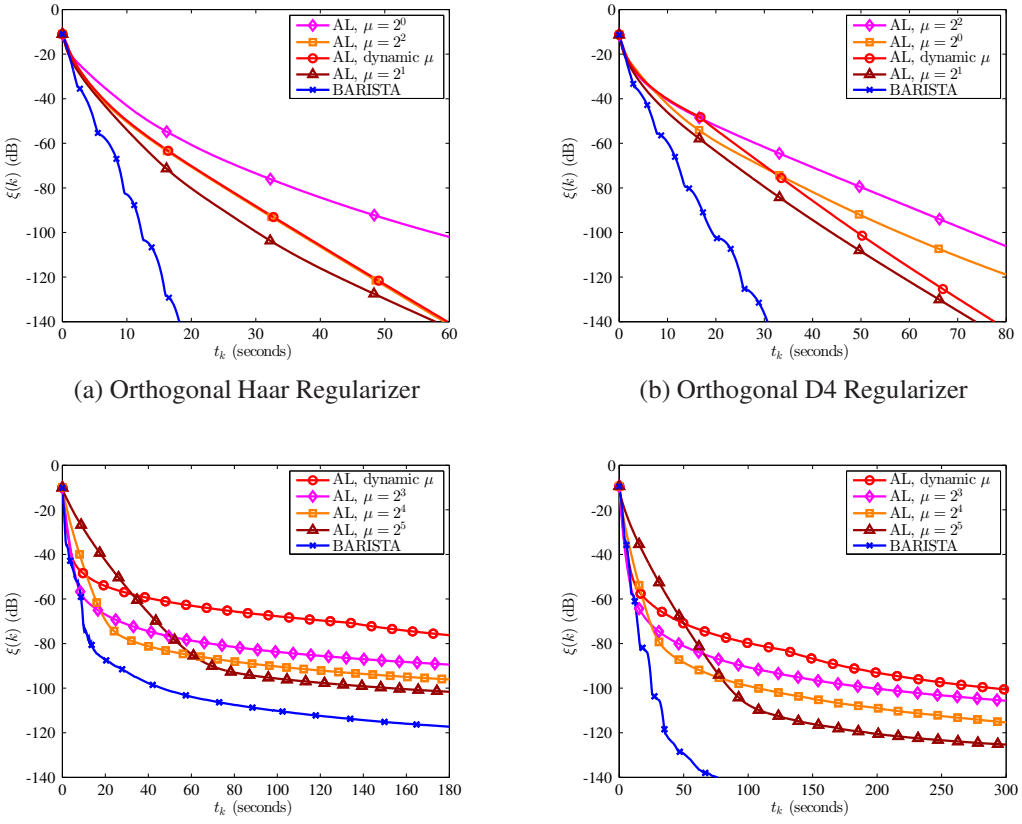
(b) Orthogonal Haar Regularizer

(c) Orthogonal D4 Regularizer

(d) Total Variation Regularizer

(e) Undecimated Haar Regularizer

Figure B.3: Examples of reconstructed images from the breast phantom imaging data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the real breast phantom. (b) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization.



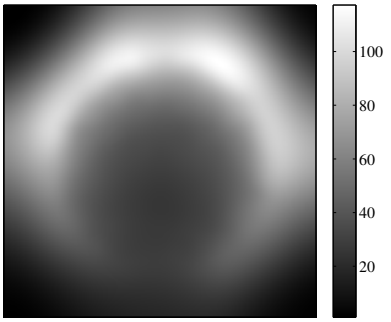
(a) Orthogonal Haar Regularizer

(b) Orthogonal D4 Regularizer

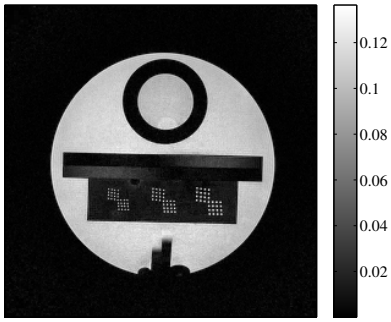
(c) Total Variation Regularizer

(d) Undecimated Haar Regularizer

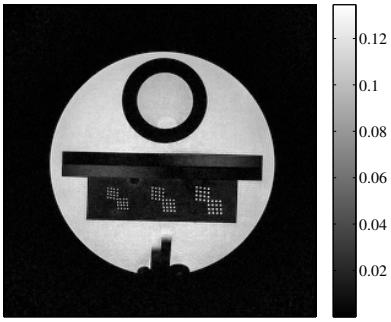
Figure B.4: Convergence speed results from breast phantom imaging data. Markers are placed at 50-iteration intervals. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.



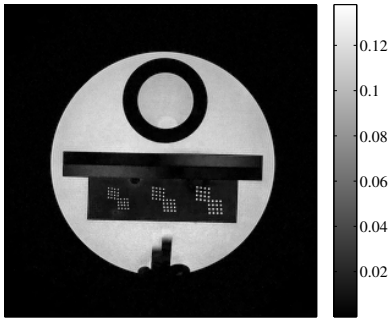
(a) SENSE Map Sum of Squares



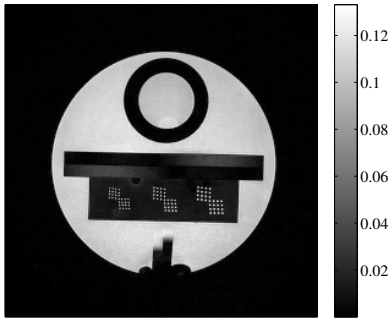
(b) Orthogonal Haar Regularizer



(c) Orthogonal D4 Regularizer

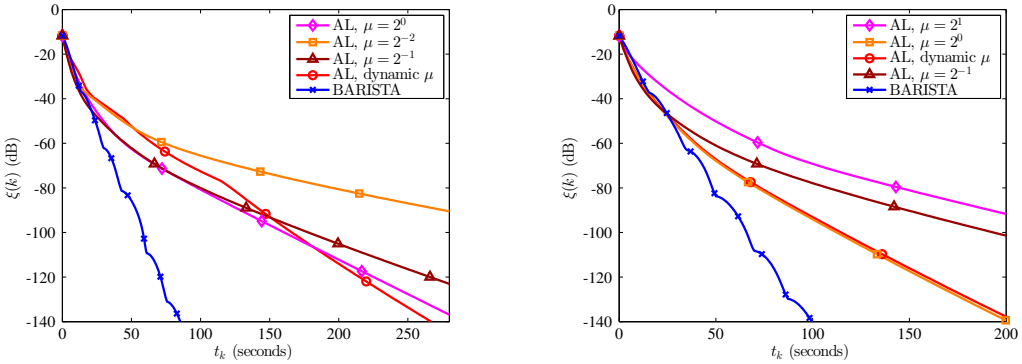


(d) Total Variation Regularizer



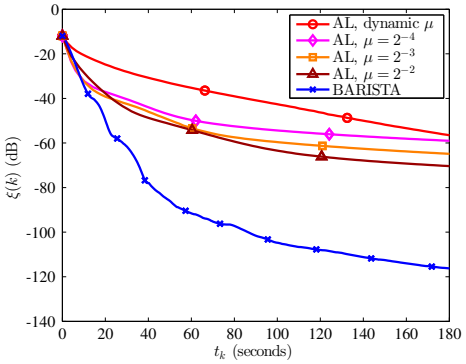
(e) Undecimated Haar Regularizer

Figure B.5: Examples of sum of SENSE maps and reconstructed images from the ACR phantom data set. (a) Sum of squared absolute values of the estimated sensitivity coil profiles for the ACR phantom data set. (c) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Haar wavelet regularization. (d) The reconstructed $\mathbf{x}^{(\infty)}$ using orthogonal Daubechies D4 wavelet regularization. (e) The reconstructed $\mathbf{x}^{(\infty)}$ using total variation regularization. (f) The reconstructed $\mathbf{x}^{(\infty)}$ using 2-level undecimated Haar wavelet regularization.

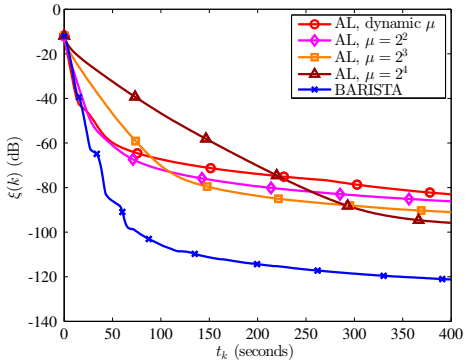


(a) Orthogonal Haar Regularizer

(b) Orthogonal D4 Regularizer

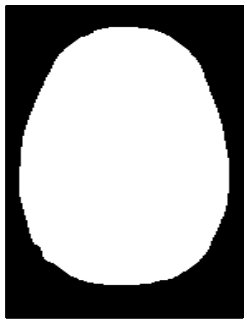


(c) Total Variation Regularizer

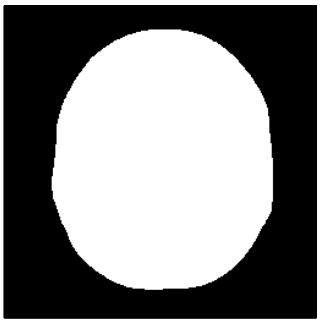


(d) Undecimated Haar Regularizer

Figure B.6: Convergence speed results from ACR phantom imaging data. Markers are placed at 30-iteration intervals. (a) Orthogonal Haar wavelet regularizer. (b) Orthogonal Daubechies D4 wavelet regularizer. (c) Anisotropic total variation regularizer. (d) Undecimated Haar wavelet regularizer.

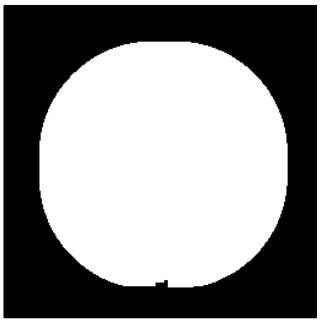


(a) *In vivo* brain mask



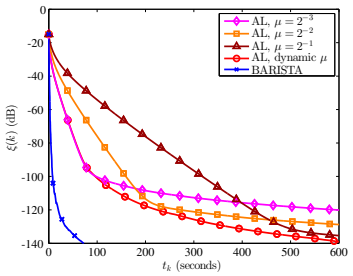
(b) BrainWeb numerical phantom mask

(c) Breast phantom mask

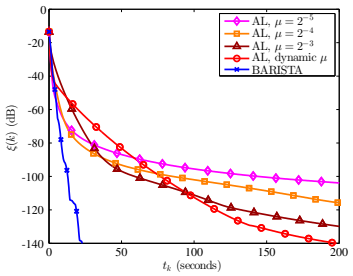


(d) ACR phantom mask

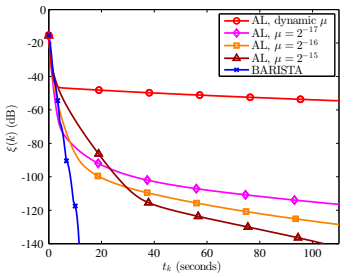
Figure B.7: Examples of image-domain masks for each of the four data sets. (a) Mask for the *in vivo* brain data set. (b) Mask for the BrainWeb numerical phantom. (c) Mask for the breast phantom. (d) Mask for the ACR phantom.



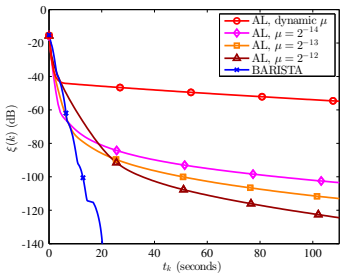
(a) *In vivo* brain, total variation



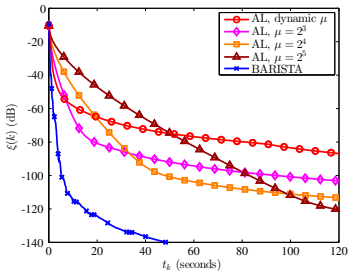
(b) *In vivo* brain, undecimated wavelets



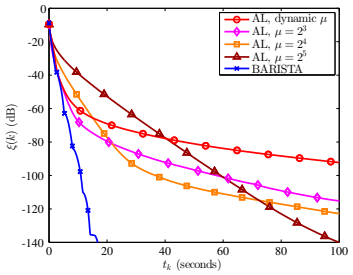
(c) BrainWeb phantom, total variation



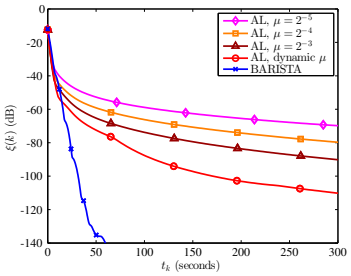
(d) BrainWeb phantom, undecimated wavelets



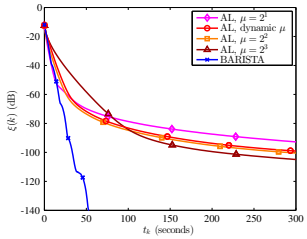
(e) Breast phantom, total variation



(f) Breast phantom, undecimated wavelets



(g) ACR phantom, undecimated wavelets



(h) ACR phantom, undecimated wavelets

Figure B.8: Convergence speed results from the algorithms using a mask. Markers are placed at 30-iteration intervals, except for the *in vivo* brain data set where they are placed at 100-iteration intervals.

BIBLIOGRAPHY

[1] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger, "SENSE: sensitivity encoding for fast MRI," *Mag. Res. Med.*, vol. 42, pp. 952–62, Nov. 1999.

[2] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Mag. Res. Med.*, vol. 58, pp. 1182–95, Dec. 2007.

[3] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Mag. Res. Med.*, vol. 73, pp. 1125–36, Mar. 2015.

[4] M. J. Allison, S. Ramani, and J. A. Fessler, "Accelerated regularized estimation of MR coil sensitivities using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 32, pp. 556–64, Mar. 2013.

[5] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, Apr. 1992.

[6] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–43, 2009.

[7] S. Ramani and J. A. Fessler, "Parallel MR image reconstruction using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 30, pp. 694–706, Mar. 2011.

[8] M. J. Muckley, D. C. Noll, and J. A. Fessler, "Fast parallel MR image reconstruction via B1-based, adaptive restart, iterative soft thresholding algorithms (BARISTA)," *IEEE Trans. Med. Imag.*, vol. 34, pp. 578–88, Feb. 2015.

[9] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. Comp. Math.*, vol. 15, pp. 715–32, June 2015.

[10] D. Kim and J. A. Fessler, "Optimized first-order methods for smooth convex minimization," *Mathematical Programming*, 2016.

[11] M. J. Muckley, D. C. Noll, and J. A. Fessler, "Momentum optimization for iterative shrinkage algorithms in parallel MRI with sparsity-promoting regularization," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 3413, 2015.

[12] B. Zhao, J. P. Haldar, A. G. Christodoulou, and Z.-P. Liang, "Image reconstruction from highly undersampled (k, t)-space data with joint partial separability and sparsity constraints," *IEEE Trans. Med. Imag.*, vol. 31, pp. 1809–20, Sept. 2012.

[13] M. Chiew, S. M. Smith, P. J. Koopmans, N. N. Graedel, T. Blumensath, and K. L. Miller, "k-t FASTER: Acceleration of functional MRI data acquisition using low rank constraints," *Mag. Res. Med.*, vol. 74, pp. 353–64, Aug. 2015.

[14] J. A. Fessler and B. P. Sutton, "Nonuniform fast Fourier transforms using min-max interpolation," *IEEE Trans. Sig. Proc.*, vol. 51, pp. 560–74, Feb. 2003.

[15] A. K. Funai, J. A. Fessler, D. T. B. Yeo, V. T. Olafsson, and D. C. Noll, "Regularized field map estimation in MRI," *IEEE Trans. Med. Imag.*, vol. 27, pp. 1484–94, Oct. 2008.

[16] B. P. Sutton, D. C. Noll, and J. A. Fessler, "Fast, iterative image reconstruction for MRI in the presence of field inhomogeneities," *IEEE Trans. Med. Imag.*, vol. 22, pp. 178–88, Feb. 2003.

[17] J. A. Fessler, S. Lee, V. T. Olafsson, H. R. Shi, and D. C. Noll, "Toeplitz-based iterative image reconstruction for MRI with correction for magnetic field inhomogeneity," *IEEE Trans. Sig. Proc.*, vol. 53, pp. 3393–402, Sept. 2005.

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

[19] M. J. Muckley and J. A. Fessler, "Fast MR image reconstruction with orthogonal wavelet regularization via shift-variant shrinkage," in *Proc. IEEE Intl. Conf. on Image Processing*, pp. 3651–5, 2014.

[20] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–23, 2006.

[21] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data," *IEEE J. Sel. Top. Sig. Proc.*, vol. 4, pp. 288–97, Apr. 2010.

[22] M. V. Afonso, José. M. Bioucas-Dias, and Mário. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Im. Proc.*, vol. 19, pp. 2345–56, Sept. 2010.

[23] S. Ramani, Z. Liu, J. Rosen, J.-F. Nielsen, and J. A. Fessler, "Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods," *IEEE Trans. Im. Proc.*, vol. 21, pp. 3659–72, Aug. 2012.

[24] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, pp. 2835–51, Nov. 1999.

[25] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *American Statistician*, vol. 58, pp. 30–7, Feb. 2004.

[26] M. W. Jacobson and J. A. Fessler, "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms," *IEEE Trans. Im. Proc.*, vol. 16, pp. 2411–22, Oct. 2007.

[27] I. W. Selesnick and M. A. T. Figueiredo, "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors," in *Proc. SPIE 7446 Wavelets XIII*, p. 74460D, 2009. Wavelets XIII.

[28] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Im. Proc.*, vol. 18, pp. 2419–34, Nov. 2009.

[29] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–6, 1958.

[30] A. R. De Pierro, "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography," *IEEE Trans. Med. Imag.*, vol. 14, pp. 132–7, Mar. 1995.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. & Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[32] H. Nien and J. A. Fessler, "A convergence proof of the split Bregman method for regularized least-squares problems," 2014. arxiv 1402.4371.

[33] D. Dunbar and G. Humphreys, "A spatial data structure for fast Poisson-disk sample generation," *ACM Trans. on Graphics*, vol. 25, pp. 503–8, July 2006. SIGGRAPH.

[34] M. Lustig, M. Alley, S. Vasanawala, D. L. Donoho, and J. M. Pauly, "L1 SPIR-iT: Autocalibrating parallel imaging compressed sensing," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 334, 2009.

[35] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao, "An accelerated linearized alternating direction method of multipliers," *SIAM J. Imaging Sci.*, vol. 8, no. 1, pp. 644–81, 2015.

[36] J. Lee, Y. Sun, and M. Saunders, "Proximal Newton-type methods for convex optimization," in *NIPS*, vol. 25, pp. 827–35, 2012.

[37] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Software*, vol. 23, pp. 550–60, Dec. 1997.

[38] S. Ramani and J. A. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. Med. Imag.*, vol. 31, pp. 677–88, Mar. 2012.

[39] M. J. Muckley, D. C. Noll, and J. A. Fessler, "Majorizer design for non-Cartesian MRI with sparsity-promoting regularization," in *ISMRM Workshop on Data Sampling & Image Reconstruction*, 2016.

[40] M. J. Muckley, D. C. Noll, and J. A. Fessler, "Fast, iterative subsampled spiral reconstruction via circulant majorizers," in *Proc. Int. Soc. Magn. Res. Med.*, 2016. To appear.

[41] M. Guerquin-Kern, D. Van De Ville, C. Vonesch, J.-C. Baritaux, K. P. Pruessmann, and M. Unser, "Wavelet-regularized reconstruction for rapid MRI," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 193–6, 2009.

[42] S. Ramani and J. A. Fessler, "Accelerated nonCartesian SENSE reconstruction using a majorize-minimize algorithm combining variable-splitting," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 704–7, 2013.

[43] Q. Wang, B. Mailhe, R. Grimm, M. D. Nickel, K. T. Block, H. Chandarana, and M. S. Nadar, "Novel sparse model and reconstruction for dynamic contrast-enhanced MRI," in *Proc. Int. Soc. Magn. Res. Med.*, p. 3791, 2015.

[44] D. Kim and J. A. Fessler, "Optimized gradient methods for smooth convex minimization," in *Intl. Symp. Math. Prog.*, 2015.

[45] M. I. Grivich and D. P. Jackson, "The magnetic field of current-carrying polygons: An application of vector field rotations," *Amer. J. Phys.*, vol. 68, pp. 469–74, May 2000.

[46] M. Guerquin-Kern, M. Haberlin, K. P. Pruessmann, and M. Unser, "A fast wavelet-based reconstruction method for magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 30, pp. 1649–60, Sept. 2011.

[47] P. A. Bandettini, E. C. Wong, R. S. Hinks, R. S. Tikovsky, and J. S. Hyde, "Time course EPI of human brain function during task activation," *Mag. Res. Med.*, vol. 25, pp. 390–7, June 1992.

[48] S. Ogawa, D. W. Tank, R. Menon, J. M. Ellermann, S.-G. Kim, H. Merkle, and K. Ugurbil, "Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging," *Proc. Natl. Acad. Sci.*, vol. 89, pp. 5951–5, July 1992.

[49] K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weisskoff, B. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, R. Turner, H. Cheng, T. J. Brady, and B. R. Rosen, "Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation," *Proc. Natl. Acad. Sci.*, vol. 89, pp. 5675–9, June 1992.

[50] S.-J. Li, Z. Li, G. Wu, M.-J. Zhang, M. Franczak, and P. G. Antuono, "Alzheimer disease: Evaluation of a functional MR imaging index as a marker," *Radiology*, vol. 225, no. 1, pp. 253–9, 2002.

[51] C. S. Monk, S. J. Peltier, J. L. Wiggins, S.-J. Weng, M. Carrasco, S. Risi, and C. Lord, "Abnormalities of intrinsic functional connectivity in autism spectrum disorders," *NeuroImage*, vol. 47, no. 2, pp. 764–72, 2009.

[52] S. M. Smith, K. L. Miller, S. Moeller, J. Xu, E. J. Auerbach, M. W. Woolrich, C. F. Beckmann, M. Jenkinson, J. Andersson, M. F. Glasser, D. C. V. Essen, D. A. Feinberg, E. S. Yacoub, and K. Ugurbil, "Temporally-independent functional modes of spontaneous brain activity," *Proc. Nat. Acad. Sci.*, vol. 109, no. 8, pp. 3131–36, 2012.

[53] J. E. Chen and G. H. Glover, "BOLD fractional contribution to resting-state functional connectivity above 0.1 hz," *NeuroImage*, vol. 107, pp. 207–218, 2015.

[54] A. P. R. Smith-Collins, K. Luyt, A. Heep, and R. A. Kauppinen, "High frequency functional brain networks in neonates revealed by rapid acquisition resting state fmri," *Human Brain Mapping*, 2015.

[55] S. Moeller, E. Yacoub, C. A. Olman, E. Auerbach, J. Strupp, N. Harel, and K. Uğurbil, "Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri," *Mag. Res. Med.*, vol. 63, no. 5, pp. 1144–1153, 2010.

[56] S. M. Smith, C. F. Beckmann, J. Andersson, E. J. Auerbach, J. Bijsterbosch, G. Douaud, E. Duff, D. A. Feinberg, L. Griffanti, M. P. Harms, M. Kelly, T. Laumann, K. L. Miller, S. Moeller, S. Petersen, J. Power, G. Salimi-Khorshidi, A. Z. Snyder, A. T. Vu, M. W. Woolrich, J. Xu, E. Yacoub, K. Uğurbil, D. C. V. Essen, and M. F. Glasser, "Resting-state fmri in the human connectome project," *NeuroImage*, vol. 80, pp. 144–168, 2013.

[57] I. P. Bruce and D. B. Rowe, "Quantifying the statistical impact of GRAPPA in fcMRI data with a real-valued isomorphism," *IEEE Trans. Med. Imag.*, vol. 33, no. 2, pp. 495–503, 2014.

[58] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, "Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR," *IEEE Trans. Med. Imag.*, vol. 30, pp. 1042–54, May 2011.

[59] M. Chiew, N. N. Graedel, J. A. McNab, S. M. Smith, and K. L. Miller, "Accelerating functional MRI using fixed-rank approximations and radial-Cartesian sampling," *Mag. Res. Med.*, 2016.

[60] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. Assoc. Comput. Mach.*, vol. 58, pp. 1–37, May 2011.

[61] D. C. Noll and W. Schneider, "Theory, simulation, and compensation of physiological motion artifacts in functional mri," in *Proc. IEEE Intl. Conf. on Image Processing*, pp. 40–4, 1994.

[62] C. Guan and M. Chiew, "Comparison of strict sparsity and low-rank constraints for accelerated FMRI data reconstruction," in *ISMRM Workshop on Data Sampling and Image Reconstruction*, 2016.

[63] M. Buehrer, K. P. Pruessmann, P. Boesiger, and S. Kozerke, "Array compression for MRI with large coil arrays," *Mag. Res. Med.*, vol. 57, pp. 1131–9, June 2007.

[64] C. H. Meyer, B. S. Hu, D. G. Nishimura, and A. Macovski, "Fast spiral coronary artery imaging," *Mag. Res. Med.*, vol. 28, pp. 202–13, Dec. 1992.

[65] B. Biswal, Y. F. Zerrin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar mri," *Magn. Res. Med.*, vol. 34, no. 4, pp. 537–41, 1995.

[66] T. E. Nichols and A. P. Holmes, "Nonparametric permutation tests for functional neuroimaging: A primer with examples," *Hum. Brain Map.*, vol. 15, no. 1, pp. 1–25, 2002.

[67] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Info Theory*, vol. 53, no. 12, pp. 4655–66, 2007.

[68] D. Yoon, J. A. Fessler, A. C. Gilbert, and D. C. Noll, "Fast joint design method for parallel excitation RF pulse and gradient waveforms considering off-resonance," *Mag. Res. Med.*, vol. 68, pp. 278–85, July 2012.

[69] M. J. Muckley, S. J. Peltier, J. A. Fessler, and D. C. Noll, "Group sparsity reconstruction for physiological noise correction in functional MRI," in *ISMRM Workshop on Data Sampling & Image Reconstruction*, 2013.

[70] M. Muckley, D. C. Noll, and J. A. Fessler, "Accelerating SENSE-type MR image reconstruction algorithms with incremental gradients," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 4400, 2014.

[71] J. Sulzer, S. Haller, F. Scharnowski, N. Weiskopf, N. Birbaumer, M. L. Blefari, A. B. Bruehl, L. G. Cohen, R. Gassert, R. Goebel, U. Herwig, S. LaConte, D. Linden, A. Luft, E. Seifritz, and R. Sitaram, "Real-time fMRI neurofeedback: progress and challenges," *NeuroImage*, vol. 76, pp. 386–99, 2013.

[72] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 1568–75, 2012.

[73] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," 2011. arxiv 1006.4046.

[74] M. V. W. Zibetti, E. S. Helou, E. X. Miqueles, and A. R. D. Pierro, "Accelerating the over-relaxed iterative shrinkage-thresholding algorithms with fast and exact line search for high resolution tomographic image reconstruction," in *Proc. IEEE Intl. Conf. on Image Processing*, pp. 2305–8, 2015.

[75] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, pp. 601–9, Dec. 1994.

[76] O. Devolder, "Stochastic first order methods in smooth convex optimization," 2011.

[77] D. Kim, D. Pal, J.-B. Thibault, and J. A. Fessler, "Accelerating ordered subsets image reconstruction for X-ray CT using spatially non-uniform optimization transfer," *IEEE Trans. Med. Imag.*, vol. 32, pp. 1965–78, Nov. 2013.

[78] R. Zeng, J. A. Fessler, and J. M. Balter, "Estimating 3-D respiratory motion from orbiting views by tomographic image registration," *IEEE Trans. Med. Imag.*, vol. 26, pp. 153–63, Feb. 2007.

[79] E. B. Welch, A. Manduca, R. C. Grimm, H. A. Ward, and C. R. J. Jr, "Spherical navigator echoes for full 3D rigid body motion measurement in MRI," *Mag. Res. Med.*, vol. 47, pp. 32–41, Jan. 2002.

[80] M. J. Muckley, S. J. Peltier, D. C. Noll, and J. A. Fessler, "Model-based reconstruction for physiological noise correction in functional MRI," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 2623, 2013.

[81] M. J. Muckley, S. J. Peltier, J. A. Fessler, and D. C. Noll, "Group sparsity reconstruction for physiological noise correction in functional MRI," in *ISMRM Workshop on Data Sampling and Image Reconstruction*, 2013.

[82] Z.-P. Liang, "Spatiotemporal imaging with partially separable functions," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 988–91, 2007.

[83] J. P. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using PowerFactorization," *IEEE Signal Proc. Letters*, vol. 16, pp. 584–7, July 2009.

[84] Y. Hu, S. Lingala, and M. Jacob, "A fast majorize minimize algorithm for the recovery of sparse and low-rank matrices," *IEEE Trans. Im. Proc.*, vol. 21, pp. 742–53, Feb. 2012.

[85] B. Zhao, J. Haldar, C. Brinegar, and Z.-P. Liang, "Low rank matrix recovery for real-time cardiac MRI," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 996–9, 2010.

[86] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc., IEEE Asilomar Conf. on Signals, Systems, and Comp.*, pp. 40–4, 1993.

[87] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing*, vol. 86, pp. 572–88, Mar. 2006.

[88] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," 2009. arxiv 0901.3403.

[89] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Trans. Med. Imag.*, vol. 17, pp. 463–8, June 1998.