Reconstruction Methods for Free-Breathing Dynamic Contrast-Enhanced MRI

by

Mai T. Le

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2017

Doctoral Committee:

Professor Jeffrey A. Fessler, Chair
Professor James M. Balter
Assistant Professor Laura Balzano
Associate Professor Marina Epelman

Mai T. Le

mtle@umich.edu

ORCID iD: 0000-0003-0972-5481

For my Ong Noi, lost to liver cancer, and my Ba Ngoai, survivor of breast cancer.

Acknowledgements

There are so many people I would like to thank who helped made this dissertation possible. First, I would like to thank my PhD advisor, Jeffrey Fessler for serving as my support, guidance, and inspiration over the years. I am grateful for all of the "green bananas" we have worked on together. I could not have asked for a more compassionate or more capable advisor.

I would also like to thank each of my committee members for their guidance and service. I thank James Balter for educating me about the role of MR in radiation therapy, collecting data on the Siemens scanner with me, and teaching me basic abdominal anatomy. I thank Laura Balzano for an unforgettable experience as a graduate student instructor and for all of her encouraging career advice and helpful networking introductions. I also thank Professor Marina Epelman for her insights and suggestions on my work.

I would also like to thank Sathish Ramani for his encouragement, advice, and technical help. Thanks also to Yue Cao, Adam Johannson, and Antonis Matakos for providing feedback on my work and helping me collect and use scanner data. I want to thank each post-doc and student in the lab for their camaraderie and fun memories.

More generally, I would like to thank my friends and classmates who have helped me survive and enjoy graduate school. Thanks to my fellow gals in EECS: Yelin Kim, Yi-Chin Wu, Azadeh Ansari, Jean Kwon, and particularly Parinaz Naghizadeh. Thanks also to Emily Wong and Airan Liu for an enduring friendship. I'd like to extend a big thanks also to Sara Rimer, Liz Dreyer, Katherine Avery, Darlene Yao, Sahithya Reddivari, Liz Getto, Stacy Ramcharan, and Ha Nguyen. I feel so thankful to have taught clubs and workshops together, celebrated and potlucked together, and run 5Ks and marched together.

Last but not least, I would like to thank my family and Byron. Bo and Ma, your love and never-ending support have made all of my accomplishments possible. Every

day I draw inspiration from your persistence and dedication. To my sisters, Lan and Kim, thank you for your support and love. Lastly, a deep, heartfelt thank you to Byron for being my rock throughout everything. Your support and encouragement have kept me going at every step of the way.

Table of Contents

List of Figures

Figure

List of Tables

Table

Abstract

Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI) is a valuable diagnostic tool due to the combination of anatomical and physiological information it provides. However, the sequential sampling of MRI presents an inherent tradeoff between spatial and temporal resolution. Compressed Sensing (CS) methods have been applied to undersampled MRI to reconstruct full-resolution images at sub-Nyquist sampling rates. In exchange for shorter data acquisition times, CS-MRI requires more computationally intensive iterative reconstruction methods.

We present several model-based image reconstruction (MBIR) methods to improve the spatial and temporal resolution of MR images and/or the computational time for multi-coil MRI reconstruction. We propose efficient variable splitting (VS) methods for support-constrained MRI reconstruction, image reconstruction and denoising with non-circulant boundary conditions, and improved temporal regularization for breast DCE-MRI. These proposed VS algorithms decouple the system model and sparsity terms of the convex optimization problem. By leveraging matrix structures in the system model and sparsifying operator, we perform alternating minimization over a list of auxiliary variables, each of which can be performed efficiently. We demonstrate the computational benefits of our proposed VS algorithms compared to similar proposed methods. We also demonstrate convergence guarantees for two

proposed methods, ADMM-tridiag and ADMM-FP-tridiag. With simulation experiments, we demonstrate lower error in spatial and temporal dimensions for these VS methods compared to other object models.

We also propose a method for indirect motion compensation in 5D liver DCE-MRI. 5D MRI separates temporal changes due to contrast from anatomical changes due to respiratory motion into two distinct dimensions. This work applies a pre-computed motion model to perform motion-compensated regularization across the respiratory dimension and improve the conditioning of this highly sparse 5D reconstruction problem. We demonstrate a proof of concept using a digital phantom with contrast and respiratory changes, and we show preliminary results for motion model-informed regularization on *in vivo* patient data.

Chapter 1

Introduction

1.1 Motivation: Radiation Therapy for Cancer

Cancer is now the second largest cause of death of Americans, behind only heart disease [3]. As the global population grows wealthier and older, rates of cancer are likely to increase, as will the demand for treatments such as radiation therapy. To help physicians monitor radiation therapy treatments, accurate parameterization of organ pharmacokinetics is increasingly valuable.

To measure tumor position and metabolic behavior, patients are injected with a contrast agent that perfuses normally into healthy tissues and more dramatically into tumorous lesions. Malignant lesions exhibit angiogenesis, the creation of additional blood vessels that feed their abnormal growth. By measuring the rate of contrast uptake, we can quantify lesion aggressiveness. Both Magnetic Resonance Imaging (MRI) and X-ray Computed Tomography (X-ray CT) can capture a series of dynamic contrast-enhanced (DCE) images, and the temporal changes measured from

those dynamic images determine pharmacokinetic parameter values. Pharmacokinetic parameters describe the bloodflow relationships between different tissues and blood vessels, characterizing processes such as absorption and distribution. These pharmacokinetic values serve as important indicators for diagnosis, treatment design and assessment, and prognosis.

Though X-ray CT acquisitions are fast and can easily yield high resolution images, they also feature a significant drawback– ionizing radiation. Unlike carefully targeted radiation therapy, X-ray CT radiation doses raise the concern of possible long-term health effects, particularly cancer [4]. As patients are imaged repeatedly for radiation therapy assessment, the risk from accumulated radiation doses multiplies. For this reason, MRI, with its safe, non-ionizing radiation, is a more compelling modality for the future of dynamic contrast enhanced imaging.

The use of MRI for DCE imaging does not come without its own challenges. Due to the nature of MRI, measurements must be made sequentially, so hardware limits the number of samples that can be acquired over a specific amount of time. To increase spatial resolution, more samples must be acquired, and more time elapses between temporal frames of the DCE images. The primary goal of DCE-MRI is to overcome this inherent tradeoff between spatial resolution and temporal resolution. For accurate pharmacokinetic parameter maps, DCE-MRI must have both.

The challenge of preserving spatial resolution while increasing temporal resolution is further complicated by patient motion. Patients cannot control involuntary motion, such as cardiac motion or intestinal peristalsis. Though patients may be asked to

hold their breath for some types of abdominal imaging, this request becomes unreasonable for the duration of a typical DCE-MRI scan, e.g., two minutes. The growing popularity of free-breathing DCE-MRI [5] [6] [7] means greater patient comfort but also presents a bigger challenge for image reconstruction.

1.2 About this Thesis

This thesis focuses on model-based image reconstruction (MBIR) methods to improve spatial resolution, temporal resolution, and pharmacokinetic parameter accuracy for anatomical and DCE-MRI. With each proposed model, we also present tailored optimization strategies to lessen the burden of computation that MBIR presents.

To move beyond the spatial resolution-temporal resolution tradeoff, we exploit spatial and temporal redundacies in DCE images to dramatically undersample data compared to conventional techniques. For example, the smoothness of contrast agent concentration time curves in tissue allows us to create an object model that lies in a smoothly varying temporal subspace consisting of B-splines. By modeling the temporal evolution between frames using these basis functions, we can estimate contrast agent dynamics with higher temporal resolution. Our minds use this principle often, such as when viewing time lapse nature photography. Because the changes being recorded occur so slowly, fewer photographs are needed to depict the overall story. The viewer assumes smooth changes between the moments captured.

Similarly, we wish to achieve a nonlinear "dimensionality reduction" on dynamic images, describing them by a small set of pharmacokinetic parameters rather than

values at each time point. The full set of samples over time is redundant when it can be adequately described with five pharmacokinetic parameters, as in the liver.

Perhaps the biggest opportunity for exploiting redundancy that we investigated is the redundancy across respiratory states. Neighboring respiratory states are likely to have many common voxel values, and we expect the similarities to increase after compensating for estimated motion. Just as camera stabilization makes for a smoother viewing experience, we explored computing a motion model from pre-contrast injection free-breathing dynamic MRI data and used it to help isolate the truly interesting signal changes from the interference of motion.

Ultimately we hope that the combination of the strategies proposed in this thesis leads to more accurate and reproducible pharmacokinetic parameter maps that can aid in patient treatment, particularly radiation therapy. The proposed optimization methods that accompany the proposed object models are intended to make this work accessible, fast, and attractive enough for further use.

1.3 Organization and Contributions of the Thesis

This thesis is constructed as follows:

Chapter 2 reviews the derivation of MRI's discrete, linear system model from MR physics. We also describe popular approaches for accelerating MRI and variable splitting optimization methods that we utilize in our proposed work. Lastly, we describe challenges specific to breast and liver DCE-MRI, including nonlinear

pharmacokinetic parameter estimation and motion compensation.

Chapter 3 presents a novel variable splitting method for efficient computation of a support-constrained image reconstruction problem. The proposed method is compared against similar variable splitting methods for speed. This chapter is based on [8].

Chapter 4 presents a variable splitting method for spatially regularized image reconstruction, designed specifically for parallelizable variable updates. This algorithm is shown to have convergence guarantees through equivalence with the Alternating Direction Method of Multipliers. This chapter is based on [9].

Chapter 5 presents a temporal spline interpolation method for breast DCE-MRI that increases apparent temporal resolution of contrast enhancement curves. In simulation, the proposed method outperformed traditional models in pharmacokinetic parameter accuracy, and the accompanying variable splitting method provided greater computational speed. This chapter is based on [10].

Chapter 6 proposes a method for indirect motion compensation in 5D MRI. By exploiting redundancy in respiratory states as well as redundancy across adjacent temporal frames, 5D motion model-informed reconstruction demonstrates some ability to reduce motion artifacts in simulation.

Chapter 7 summarizes the thesis and outlines several proposed future projects on liver DCE-MRI. We suggest several methods for improving 5D motion model-informed reconstruction for patient data. Lastly, we propose methods that combine the typical 2-step approach of estimating pharmacokinetic paramters

into a single direct estimation approach, while also exploiting respiratory redundancies.

Chapter 2

Background

2.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a method of medical imaging that uses non-ionizing electromagnetic fields to excite magnetic spins in the body. It offers excellent soft tissue contrast and has many specialized applications, including functional MRI to measure brain function and diffusion MRI for diagnosing neurological conditions and disorders. This thesis will focus on reconstruction techniques for anatomical imaging and Dynamic Contrast-Enhanced (DCE) MRI. We begin by describing an overview of signal generation and acquisition common to all applications of MRI.

2.1.1 Spins and Magnetic Fields

Atomic nuclei that possess both a magnetic moment and angular momentum possess the nuclear magnetic resonance property and are referred to as "spins". Since

the human body consists largely of water, ^{1}H is very abundant in the body. It is these protons that are primarily imaged in MR. In the absence of a strong external magnetic field, the spins inside a material are oriented randomly, resulting in a weak net magnetization in an arbitrary direction. However, an external magnetic field induces spins to align parallel or anti-parallel to the field. For MR signal generation, this large, external magnetic field is referred to as the main field, $\vec{B}_0$ [Tesla or Gauss]. By convention, this direction is usually chosen as the z-axis or longitudinal direction:

$$\vec{B}_0 = B_{z0}\vec{k}, \tag{2.1}$$

where $\vec{k}$ is the unit vector in the z-direction. This field is generated by superconducting coils or large permanent magnets and remains static. Since the anti-parallel state has a higher energy than the parallel state (with energy difference ΔE), nuclei are more likely to align parallel to $\vec{B}_0$, along the z-direction, creating a net magnetization vector, denoted $\vec{M}(r,t)$ [Am/m].

When a magnetic moment, $\vec{\mu}$, is placed in a external magnetic field, $\vec{B}_0$, it will precess like a top at a frequency proportional to the applied magnetic field, known as the Larmor frequency.

The following relationship is fundamental to nuclear magnetic resonance (NMR) and MRI:

$$f_0 = \overline{\gamma} B_{z0} \quad \text{or} \quad \omega_0 = \gamma B_{z0} \tag{2.2}$$

where f_0 is the angular frequency of the spin precession [Hz], ω_0 is the resonant frequency, B_{z0} is the strength of the applied magnetic field, and $\overline{\gamma} = \frac{\gamma}{2\pi}$ is the gyromagnetic ratio of the material, [MHz T^{-1}]. For ^{1}H, $\overline{\gamma} = 42.28$ MHz/Tesla.

2.1.2 Excitation with an RF pulse

In the excitation phase of a scan, a very short time-varying radiofrequency (RF) field, $\vec{B}_1(t)$ is applied perpendicular to the main field, $\vec{B}_0$. This is an amplitude-modulated pulse transmitted through quadrature RF coils and is often called the "excitation field". This pulse is tuned to the resonant frequency of the material. This causes the magnetization to precess, and for ease of notation, the precession is usually analyzed in a rotating frame of reference. In the rotating frame of reference, $\vec{M}(\vec{r},t)$ can be decomposed into a transverse component, $M_{XY}(\vec{r},t)$ and the longitudinal component, $M_z(\vec{r},t)\vec{k}$. We express the transverse magnetization as a complex quantity for convenience:

$$M_{XY}(\vec{r},t) = M_X(\vec{r},t) + iM_Y(\vec{r},t). \tag{2.3}$$

The RF pulse tips the magnetization vector away from the main field so that a portion of it lies in the x-y plane, with some energy perpendicular to $\vec{k}$. After the RF pulse, the magnetization vector will relax to its equilibrium state, parallel to $\vec{k}$. Magnetization behavior during and after the RF pulse is described by the Bloch

equation [11]:

$$\frac{d\vec{M}}{dt} = \vec{M} \times \gamma\vec{B} - \frac{M_X\vec{i} + M_Y\vec{j}}{T_2} - \frac{(M_Z - M_{z0})\vec{k}}{T_1} \tag{2.4}$$

where M_{z0} is the largest possible magnitude, $M_z(\vec{r}, 0^-)$, or the value of M_z an instant before the RF pulse is applied.

The Bloch Equation has the following solutions after the RF pulse is turned off. The longitudinal component recovers according to Equation (2.5), whereas the magnitude of the transverse component decays according to Equation (2.6), as follows:

$$M_z(\vec{r}, t) = M_{z0}(\vec{r})\left(1 - e^{-t/T_1(\vec{r})}\right) + M_z(\vec{r}, 0)e^{-t/T_1(\vec{r})} \tag{2.5}$$

$$|M_{xy}(\vec{r}, t)| = |M_{xy}(\vec{r}, 0)|e^{-t/T_2(\vec{r})}. \tag{2.6}$$

T_1 and T_2 values vary depending on the material and thus provide anatomical contrast. T_1 is the spin-lattice constant, and T_2 is the spin-spin time constant. Typical values for T_1 are 100-1000 msec, and typical values for T_2 are 10-100 msec. MRI acquisition sequences can be designed to produce T_1- or T_2- weighted images.

As the magnetization precesses, its transverse component, $M_{xy}(\vec{r}, t)$, is detected by the receiving coil via Faraday's Law of Induction. By measuring the rate at which $M_z(\vec{r}, t)$ recovers and at which $|M_{xy}(\vec{r}, t)|$ decays, it is possible to measure the T_1 and T_2 constants for the object at hand. However, this does not provide any localization

of signal, so images cannot be produced from this information alone.

2.1.3 Localization with Field Gradients

From Equation (2.2), we know that the angular frequency of the precession is directly proportional to the overall magnetic field. By varying the magnetic field along the x and y directions, we can encode location information into the frequencies of precession. More specifically, we can design $\vec{B}(\vec{r})$ as follows:

$$\vec{B}(\vec{r}) = (B_{z0} + \underbrace{xG_x + yG_y + zG_z}_{\vec{r}\cdot\vec{G}})\vec{k} \tag{2.7}$$

where G_x, G_y, G_z indicate linear field gradients, with typical values less than 1 Gauss/cm. Application of these field gradients causes spins at different locations to precess at different frequencies, thus encoding location information into the precession frequency. For this pioneering work, Paul Lauterbur and Peter Mansfield earned a Nobel Prize in 2003.

2.1.4 Signal Equation

The signal detected by the RF receiver coils can be approximated as:

$$s(t) \approx -i\omega_0 B_{1xy} e^{-i\omega_0 t} \int\int\int M_{xy}(\vec{r}, 0) e^{-t/T_2(\vec{r})} e^{-i\phi(\vec{r},t)} d\vec{r}, \tag{2.8}$$

where the phase term $\phi(\vec{r}, t)$ depends on the design of the field gradients over

time:

$$\phi(\vec{r},t) = \gamma \int_0^t \vec{r} \cdot \vec{G}(\tau) d\tau. \tag{2.9}$$

After I/Q demodulation, the baseband signal is then

$$s(t) \triangleq \int\int\int M_{xy}(\vec{r},0) e^{-t/T_2(\vec{r})} e^{-i\phi(\vec{r},t)} d\vec{r}. \tag{2.10}$$

Introducing the notion of k-space, the Fourier domain of image space, the following useful formulation arises:

$$s(t) \approx \int\int\int M(\vec{r},0) e^{-t/T_2(\vec{r})} e^{-i2\pi\left(\vec{r}\cdot\vec{k}(t)\right)} d\vec{r}, \tag{2.11}$$

where k-space coordinates $\vec{k}(t)$ is defined as follows:

$$\vec{k}(t) = \overline{\gamma} \int_0^t \vec{G}(\tau) d\tau. \tag{2.12}$$

Further approximation of the relaxation and encoding terms allows us to define an image of the transverse magnetization over a selected rectangular slice of thickness Δz centered at z_0. We define the image $m(\vec{r})$ as:

$$m(\vec{r}) \triangleq \int_{z_0-\frac{\Delta z}{2}}^{z_0+\frac{\Delta z}{2}} M(\vec{r},0)e^{-t/T_2(\vec{r})}dz. \tag{2.13}$$

This generates a familiar relationship between $s(t)$ and $m(\vec{r})$:

$$s(t) = \int\int m(\vec{r})e^{-i2\pi\left(\vec{r}\cdot\vec{k}(t)\right)}d\vec{r} = \mathcal{M}(\vec{k}(t)). \tag{2.14}$$

We can recognize $\mathcal{M}(\vec{k}(t)) = \mathcal{F}\{m(\vec{r})\}$, the Fourier transform of the image $m(\vec{r})$ [12]. K-space can be interpreted as spatial frequencies, the inverse of image space.

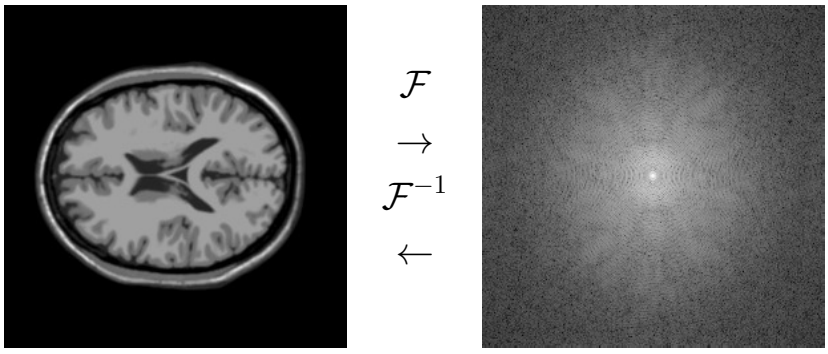


Figure 2.1: A T_1-weighted simulated brain image (via Brainweb) and its k-space representation are related via the Fourier Transform. (a) $m(\vec{r})$ in the image domain (b) log scale of magnitude of $\mathcal{M}(\vec{k})$ in k-space

2.1.5 Imaging

Since the signal measured by the RF coils directly samples the 2D Fourier Transform of the image, the field gradients can be designed so that the k-space trajectories

adequately sample $\mathcal{M}(\vec{k}(t))$, from which $m(\vec{r})$ can be reconstructed via a simple inverse FFT.

This is traditionally done by sampling k-space in a Cartesian pattern with sampling rate that satisfies the Nyquist criterion. To prevent spatial aliasing, the sampling rate in k-space, Δ_k, is chosen depending on the field of view (FOV) of the object: $\Delta_k \leq \frac{1}{FOV}$. The data acquired after RF excitation, during the readout portion of the pulse sequence, corresponds to a single line of samples in Cartesian imaging, often referred to as a "readout". For 3D Cartesian imaging, these lines are also called "phase encodes".

Non-Cartesian sampling trajectories, such as radial, spiral and rosette sampling, are popular as fast imaging methods. We will discuss radial sampling in more detail in Section 2.5.3. Cartesian sampling itself is quite broad and can be accomplished with a variety of acquisition sequences, such as Echo Planar Imaging (EPI).

Examples of Cartesian and radial sampling trajectories are shown in Figure 2.2.

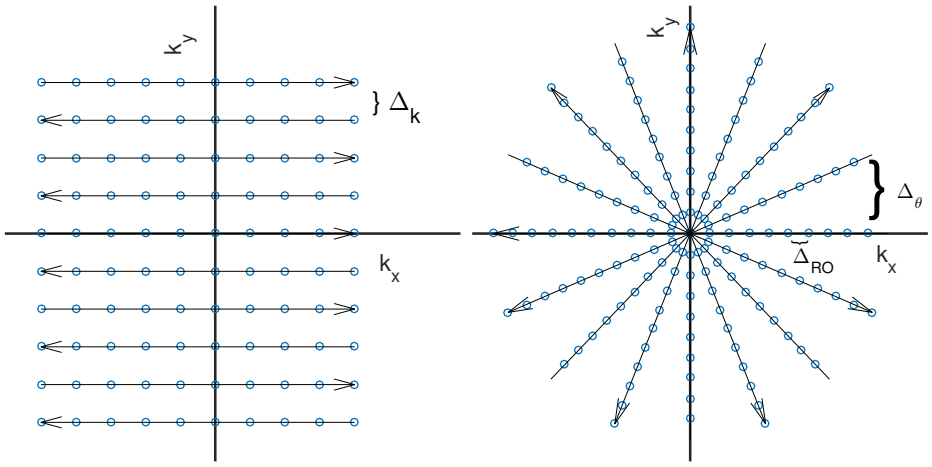


Figure 2.2: (a) Cartesian trajectory with 10 readout lines and sampling rate Δ_k; (b) radial trajectory with 8 spokes (i.e., readouts), angular spacing Δ_θ, and sampling rate of Δ_{RO} in a given readout.

.

2.1.6 Discrete Model

The measurements made by the receiving coil are sampled by an A/D converter, leading to noisy discrete measurements:

$$y_i = s(t_i) + \varepsilon_i \tag{2.15}$$

where y_i is the ith element of the discretized signal y, and ε_i represents complex Gaussian noise. The discrete signal $s(t_i)$ can be approximated by discretizing the continuous image $m(\vec{r})$:

$$m(\vec{r}) \approx \sum_{j=1}^{N_r} m_j b(\vec{r} - \vec{r}_j), \tag{2.16}$$

where $b(\vec{r})$ denotes a spatial basis function, usually rectangular pixels, $b(\vec{r}) = \text{rect}(x, y, z)$. The number of voxels in the image is N_r. This yields the following approximation for $s(t_i)$:

$$s(t_i) \approx \sum_{j=1}^{N_r} m_j \int\int b(\vec{r} - \vec{r}_j) e^{-i2\pi[\vec{r}\cdot\vec{k}(t_i)]} d\vec{r} \tag{2.17}$$

$$= \sum_{j=1}^{N_r} a_{i,j} m_j, \tag{2.18}$$

$$\text{where } a_{i,j} = \int\int b(\vec{r} - \vec{r}_j) e^{-i2\pi[\vec{r}\cdot\vec{k}(t_i)]} d\vec{r}. \tag{2.19}$$

Finally, we arrive at a convenient matrix model for the imaging system:

$$y = \mathbf{A}m + \varepsilon \tag{2.20}$$

where $\mathbf{A}$ is the system matrix containing the $a_{i,j}$ entries, and m is the vector of unknown pixel values. For the remainder of this report, m will be renamed x to follow convention. We can recognize $\mathbf{A}$ as being the Discrete Fourier Transform.

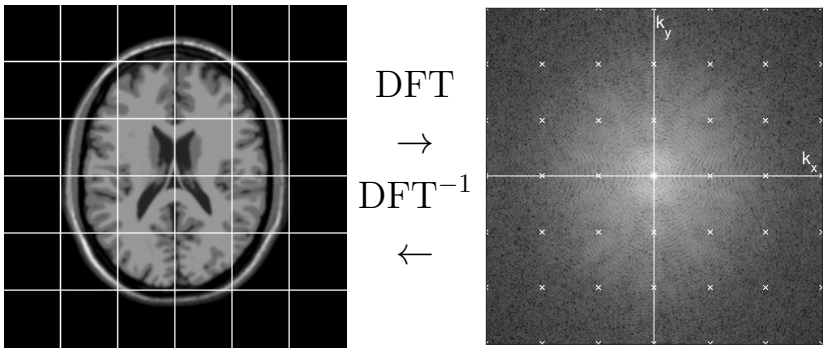


Figure 2.3: Discrete k-space measurements y relate to the Discrete Fourier Transform of the voxels of the discretized object, m.

.

2.1.7 Non-Iterative Reconstruction

For Cartesian sampling, the desired image x can be efficiently recovered from the data y by the inverse Fast Fourier Transform. This section describes other common non-iterative methods for MRI reconstruction.

Many non-Cartesian sampling trajectories result in non-uniform sampling density. For example, the radial trajectory pictured in Figure 2.2 has a higher density of samples toward the center of k-space, with 8 repeated measurements of the origin itself. Towards the periphery, the spacing between samples grows. Because most of the information in k-space is located at or near the center of k-space, many trajectories have been proposed with a higher density of samples in the center, such as spirals [13] and rosettes [14].

To adjust for the non-uniformity of samples, a density compensation function is usually applied to the samples before application of the inverse Fourier transform.

The density compensation function weights samples proportionally to the sampling density at that k-space location. For regular sampling patterns, such as radial and spiral trajectories, this density compensation function can be determined analytically. For more complicated non-Cartesian trajectories, the density compensation function is often computed according to the area of Voronoi cells of that trajectory [15]. Voronoi cells encapsulate the region of k-space points that are closest to a given sample. Figure 2.4 depicts a Voronoi diagram for a radial trajectory.

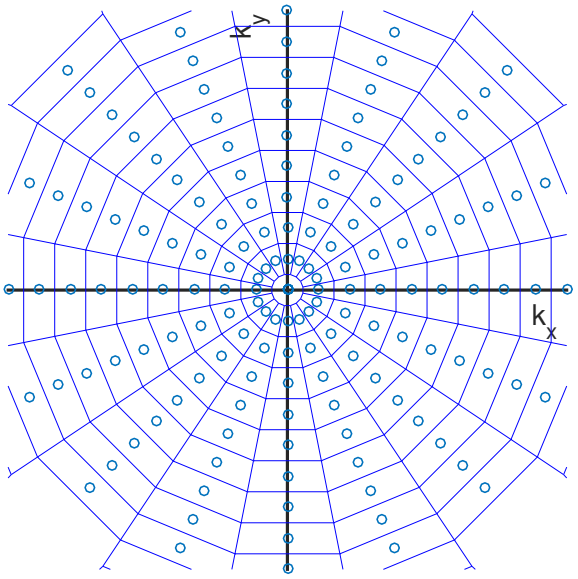


Figure 2.4: Voronoi diagram for a radial sampling trajectory with 8 spokes. Samples are weighted proportionally to the area in its Voronoi cell to account for non-uniform sampling density.

.

To efficiently perform the Fourier transform for non-Cartesian samples of k-space, the samples must be interpolated to the Cartesian grid, in a process called "gridding" [16]. This is often done practically with a Kaiser-Bessel interpolation kernel

[17].

2.2 Accelerated MRI

2.2.1 Motivation for MRI Acceleration

In clinical practice, a typical suite of MRI scans can take up to 90 minutes to perform. Lengthy scan times are disadvantageous for a number of reasons. Firstly, the longer a patient is in the MRI scanner, the more costly the procedure is. The capital spent on a new MRI scanner is distributed over the number of patients that it services, so improving patient throughput reduces cost per patient. MRI acceleration may also make MRI a more attractive choice of imaging modality for physicians. MRI is slow compared to X-Ray CT, but it does not harm the patient with ionizing radiation. Though MRI cannot compete against ultrasound on compactness or real-time imaging, MRI is able to provide higher resolution images and able to penetrate more anatomical areas. More prevalent use of MRI by physicians could result in better outcomes for some patients.

Additionally, the longer the duration of acquisition, the more likely motion artifacts will degrade image quality. Motion artifacts arise from conscious movement, as well as from essential body functions such as breathing, cardiac movement, and blood flow. Just as a longer exposure for a film camera makes photographs more susceptible to blur, patient movement similarly degrades organ and lesion contrast in MR images.

Acceleration is also particularly useful for DCE-MRI, since accurate estimation of contrast agent concentration time series requires both high spatial resolution and high temporal resolution. Unfortunately, there exists a fundamental tradeoff between spatial resolution and temporal resolution. Increasing the number of k-space locations sampled in a given frame also increases the time between frames (and temporal sampling rate).

Because MRI samples are acquired sequentially, the primary method of acceleration is to undersample k-space, i.e., sample fewer points than required by the Nyquist sampling theorem. However, undersampling causes Equation (2.20) to be underdetermined, which necessitates the introduction of additional information. One class of methods, parallel imaging, acquires spatially encoded information on multiple receive coils simultaneously. Another method is to introduce assumptions about image smoothness or sparsity in particular domains.

2.2.2 Sensitivity Encoding (SENSE)

Parallel imaging methods utilize additional RF surface coils placed adjacent to the body. Parallel imaging accelerates MRI acquisition by collecting redundant information on multiple receive coils in parallel, thus permitting undersampling. Each coil is able to detect RF signals close to it with a spatially-varying strength, described by its sensitivity map. This is in contrast to traditional MRI that uses a single body coil designed to have uniform sensitivity throughout.

In this report, we focus on one parallel imaging paradigm, Sensitivity Encoding

(SENSE). In SENSE, the sensitivity maps are explicitly estimated and applied in the image domain. The estimation of sensitivity maps is an active research field [18]. Other reconstruction methods for parallel imaging include Simultaneous Acquisition of Spatial Harmonics (SMASH) [19] and Generalized Autocalibrating Partially Parallel Acquistions (GRAPPA) [20].

Though SENSE was initially developed as a method for increasing local SNR, the use of multiple receive coils also permits undersampling of k-space. In its original formulation, SENSE was combined with a structured undersampling scheme: entire readout lines are omitted such that only 1 out of R readout lines are acquired, where the reduction (or undersampling) factor R is less than or equal to N_c, the number of sensitivity coils. The undersampled k-space data produces a series of N_c aliased coil images, that can be unmixed to reconstruct the original image [21].

For a general (unstructured) undersampling scheme, SENSE can be formulated as a least squares problem:

$$x = \underset{x}{\operatorname{argmin}} \frac{1}{2} \left\| y_{\mathrm{MC}} - \mathbf{A}_{MC}\mathbf{S}x \right\|^2 \tag{2.21}$$

$$y_{\mathrm{MC}} = \begin{bmatrix} y_1 \\ \vdots \\ y_{N_c} \end{bmatrix}; \quad \mathbf{A}_{MC} = \mathbf{I}_{N_c} \otimes \mathbf{A}; \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_{N_c} \end{bmatrix}. \tag{2.22}$$

In this multicoil setup, N_{samp} is the number of positions sampled in k-space, $y_{\mathrm{MC}} \in \mathbb{C}^{N_c N_{\mathrm{samp}}}$ stacks each coil's data, y_i, $\mathbf{A}_{MC}$ applies the undersampled Fourier encoding operator $\mathbf{A}$ to each coil image, and $\mathbf{S}$ is a stack of the diagonalized sensitivity maps for each of the N_c coils. An undersampled Fourier encoding operator simply omits the outputs that are not sampled and can be pictured as a fat matrix. Figure 2.5 shows simulated coil images that correspond to $\mathbf{S}x$.

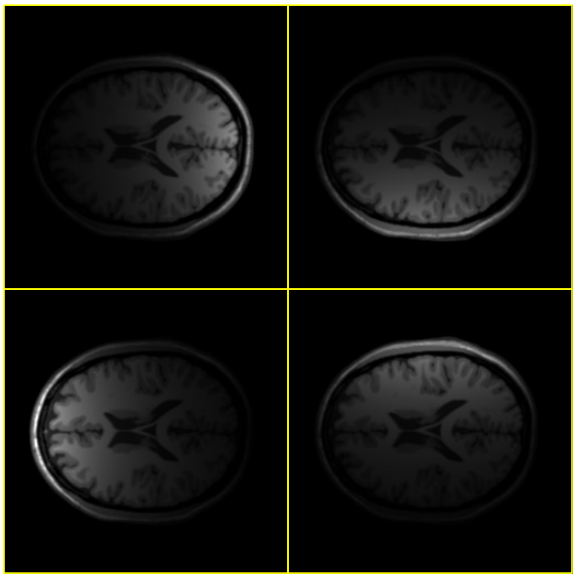


Figure 2.5: Simulated brain image with 4 simulated sensitivity coil maps applied. In the generalized SENSE framework, multi-coil samples y_{MC} are modeled as k-space samples of these images.

.

Depending on the desired spatial and temporal resolution, coil geometry, and anatomical challenges, this problem can still be ill-posed despite the additional information from the N_c receive coils. For example, if the object field of view (FOV) is large, the sensitivity of the receive coils might be weak at the center of the object, resulting in poor SNR. This can be remedied by introducing regularization.

2.2.3 Regularization

Regularization is a method of introducing prior information to an ill-conditioned inverse problem. In the case of undersampled MRI, there exist more unknowns than data. Thus, a multitude of solutions (reconstructed images) fit the limited acquired data. By assuming the image has certain properties, e.g., smoothly varying or piece-wise constant, it is possible to penalize solutions that do not fit the a priori assumptions and improve the condition number of the reconstruction problem.

It has been demonstrated that regularization reduces noise and aliasing artifacts in SENSE reconstruction [22][23], and various types of regularizers have been proposed, including Tikhonov regularization [22], nuclear norm regularization [24], and Bregman iteration regularization [25].

Regularization involves the choice of the potential function as well as the domain of regularization. A general form for a regularization term is $R(x) = \lambda \sum_{k=1}^{K} \psi_k \left([\mathbf{C}x]_k \right)$, where $\mathbf{C}$ transforms x to the domain over which we evaluate the potential function, ψ. For many of the following experiments, we chose $\mathbf{C}$ to be first-order finite-differences, measuring the difference between neighboring pixels in space or time. Wavelets are

another popular choice for $\mathbf{C}$. A crucial design component of regularization is the regularization parameter, here denoted λ, that controls the balance between the datafit term and the regularization term in the cost function. In many of the following methods, the potential function ψ was chosen to be the ℓ_1 norm to promote sparsity of Cx. Sparsity is a desirable property for applications of compressed sensing.

2.2.4 Compressed Sensing MRI

In recent years, the field of Compressed Sensing (CS) has been applied to MRI reconstruction with great success [26]. MRI is a natural fit for CS for a number of reasons. Firstly, MR images are compressible. Angiograms, images of blood vessels, are sparse in the image domain, and anatomical images are often approximately piece-wise constant and thus approximately sparse after finite-differencing. Dynamic images are often further compressible than static images, because large portions of the image remain constant or undergo limited movement. Thus it is often possible to find transforms under which MR images become sparse.

Secondly, MRI samples are acquired in k-space, and k-space is often incoherent with respect to the sparsifying transform domain. This is important because randomized or irregular sampling in k-space causes incoherent aliasing in the image domain. In other words, the aliasing, or leakage of energy from a given pixel to surrounding pixels, does not combine in a constructive or destructive manner as overlapping replicas but rather appears as noise. The recovery of the sparse coefficients can be performed with a nonlinear iterative thresholding algorithm. Non-iterative methods,

such as those described in Section 2.2.6 can be used to initialize these iterative reconstruction methods.

Though many of the theoretical results for CS hinge on the assumption of truly random sampling, it is simply a sufficient, not necessary, condition for applying CS theory [27]. This random undersampling condition is particularly troublesome for MRI, as energy is not uniformly distributed in k-space. For this reason, sampling schemes used in CS-MRI are an active field of research.

2.2.5 Undersampling Methods

Various sampling methods have been proposed that balance randomness and ensuring a particular density (or uniformity) of samples. One such method is Poisson Disk Sampling.

A Poisson sampling distribution is generated by selecting coordinates of samples independently from a uniform distribution. Because the number of samples in a given area is Poisson distributed, this is called the Poisson distribution. However, this can lead to "clumping" of samples in some areas and large voids in others. To remedy this, the Poisson disk sampling method generates samples in the same manner, with the additional restriction that points must be a minimum distance from each other [28]. This method can be used to select a subset of Cartesian grid points or lattice grid points [29] for acquisition.

Undersampling in k-space presents particular challenges. The majority of the energy of k-space, and thus most of the image information, resides at or near DC. For this

reason, it is desirable to have a higher density of samples in the region close to the center, or DC. One method of doing this is to use a variable density sampling pattern, such as in [30]. Another method is to fully sample central k-space and undersample only the higher frequencies. Because this method has demonstrated satisfactory image quality [26], this approach was chosen for many of the following Cartesian simulations and experiments.

Cartesian sampling patterns have also been adapted to mimic the motion robustness of radial sampling [31], and several experiments in 5 were implemented using this sampling scheme.

For liver DCE-MRI experiments in Section 7, data was acquired in a radial sampling pattern. More specifically, the 3D volume was sampled in a stack-of-stars configuration, in which the sampling is Cartesian along z but radial in each axial slice. This trajectory is pictured in Figure 2.6. Radial sampling also provides a higher density of samples near the center of k-space.

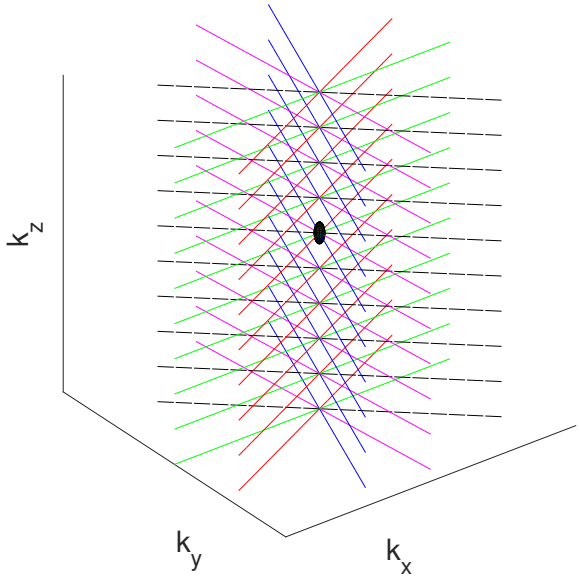


Figure 2.6: Golden-angle stack-of-stars trajectory with 5 golden-angle spokes and 10 slices. The center-of-k-space is denoted with a large black dot.

.

2.2.6 Datasharing Methods for Dynamic MRI

Several popular methods have been proposed for simply reconstructing dynamic MR images from undersampled data. These methods rely on temporal interpolation of specially designed undersampling patterns to achieve a balance between spatial and temporal resolution in the resulting dynamic MRI image. In other words, data is shared from one frame to another to fill in missing samples. After datasharing is applied, standard non-iterative reconstruction methods, such as the FFT or NUFFT can be used for efficient reconstruction.

In the keyhole method [32] [33], a fully sampled reference image is measured before

or well after contrast injection, when dynamic changes are minimal. During the injection of contrast, only the low-frequency k-space locations are sampled. This results in fewer sampling locations in each frame and higher temporal resolution. Rather than reconstructing the undersampled DCE data alone, which would result in poor spatial resolution, the contrast data is combined with the higher-frequency data of the reference image, and a higher spatial resolution frame is reconstructed from the combination of the two datasets. This can be viewed as an sampling pattern in which the center of k-space is fully sampled in each frame, and the outer regions are completely unsampled, with the exception of a fully sampled initial frame. The intuition is that higher frequency data describe details such as edges, and these edges are not expected to change much during a breath hold scan.

Datasharing is a generalization of the keyhole method, in which the undersampling pattern is not restricted the keyhole design. Outer regions of k-space can be sampled during the course of a dynamic imaging experiment, resulting in less severe temporal interpolation in outer regions of k-space.

Radial trajectories acquire high spatial frequency samples at little extra cost, so keyhole and traditional datasharing methods are less efficient. Instead, radial datasharing methods combine portions of spokes (i.e., readouts) from neighboring frames to the full spokes of the pertinent frame. K-space Weighted Image Contrast (KWIC) divides k-space into annular regions, including more neighboring frames' spoke segments in outer annuli [34]. KWIC is also particularly useful for sampling trajectories that don't ensure repeated sampling of k-space locations, such as the golden angle radial sampling scheme, discussed in Section 2.5.3.

For these datasharing methods, the notion of temporal resolution is no longer straightforward. In traditional dynamic image reconstruction, only data acquired during a specific frame is used to reconstruct that frame. Datasharing uses data from many other frames to reconstruct a given frame, with a temporal "footprint" that is often much larger than the reported temporal resolution. For datasharing methods, temporal resolution is often reported as the full duration of the experiment divided by the number of reconstructed frames, which does not account for the temporal footprint of datasharing.

Figure 2.7 shows the radial datasharing method as applied to a toy example with 24 golden angle radial spokes apportioned over 3 frames. Each spoke readout consists of 20 samples and has a gradient delay, resulting in a constant in-spoke offset of 0.75 samples in k-space. In the top left, we show the full and partial spokes included in frame 2. For reference, the annuli based on Nyquist limits are drawn, and pairs of points that violate the maximum distance allowed by Nyquist are adorned with circles. Beyond the largest annuli, we have included all possible spokes and cannot satisfy Nyquist.

2.3 Optimization Methods for Image Reconstruction

Many cost functions designed for MR image reconstruction have structures that can be expoited for faster computation. General purpose methods for unconstrained minimization of differentiable cost functions include gradient methods such as gradient descent and preconditioned conjugate gradient. Here we discuss computation-

ally efficient variable splitting alternatives for solving proposed image reconstruction problems.

2.3.1 Variable Splitting

The simplest variable splitting method for use on regularized cost functions is the Split Bregman method [35]. Here we explore this variable splitting in additional detail to illustrate the intuition behind this and more complex variable splitting schemes. The following discussion will focus on the application of Generalized Split Bregman to an ℓ_1-regularized problem.

Consider the following optimization problem, with convex functions $|\Phi(x)|_1$ and $H(x)$:

$$\min_x H(x) + |\Phi(x)|_1. \tag{2.23}$$

Furthermore, assume $\Phi(\cdot)$ to be differentiable. Equation (2.23) is equivalent to the following constrained optimization problem:

$$\min_{x,u} H(x) + |u|_1 \text{ s.t. } u = \Phi(x). \tag{2.24}$$

Constrained problems can be converted into unconstrained problems using a quadratic penalty function:

$$\min_{x,u} H(x) + |u|_1 + \frac{\mu}{2} \|u - \Phi(x)\|_2^2 . \tag{2.25}$$

One method to enforce equivalence with the problem in (2.24) is to make $\mu \to \infty$,

called continuation. Instead, we apply a Bregman iteration, iteratively minimizing original variable x and auxiliary variable u:

$$\left(x^{(n+1)}, u^{(n+1)}\right) = \min_{x,u} H(x) + |u|_1 + \frac{\mu}{2}||u - \Phi(x) - \eta^{(n)}||_2^2 \tag{2.26}$$

$$\eta^{(n+1)} = \eta^{(n)} + \left(\Phi(x^{(n+1)}) - u^{(n+1)}\right). \tag{2.27}$$

We update the Bregman parameter η with the constraint error. Rather than minimizing x and u jointly, we apply alternating minimization:

$$x^{(n+1)} = \min_{x} H(x) + |u^{(n)}|_1 + \frac{\mu}{2}||u^{(n)} - \Phi(x) - \eta^{(n)}||_2^2 \tag{2.28}$$

$$u^{(n+1)} = \min_{u} H(x^{(n+1)}) + |u|_1 + \frac{\mu}{2}||u - \Phi(x^{(n+1)}) - \eta^{(n)}||_2^2. \tag{2.29}$$

By now, we have decoupled the ℓ_1 and ℓ_2 portions of the cost function into two simpler problems. The x-update is now differentiable, and can be solved through a variety of methods, such as conjugate gradient. The u-update has a closed form solution that consists of shrinkage. For the ℓ_1 norm, the shrinkage function is a very fast operation, soft thresholding:

$$u^{(n+1)} = \text{soft}\left(\Phi(x^{(n+1)}) + \eta^{(n)}, \frac{1}{\mu}\right) \tag{2.30}$$

$$\text{soft}(x, \gamma) = \text{sign}(x)\max\left(|x| - \gamma, 0\right) \tag{2.31}$$

Split Bregman

Input $y, \mathbf{F}, \mathbf{S}, \mathbf{M}, \mathbf{C}_s$
Select μ.
Initialize x
Initialize $u = \Phi(x)$ and $\eta = 0$.
for N iterations **do**
    for M iterations **do**
        $x^{(n+1)} = \min_x H(x) + |u^{(n)}|_1 + \frac{\mu}{2}||u^{(n)} - \Phi(x) - \eta^{(n)}||_2^2$.
        $u^{(n+1)} = \text{soft}\left(\Phi(x^{(n+1)}) + \eta^{(n)}, \frac{1}{\mu}\right)$.
    end for
    $\eta^{(n+1)} = \eta^{(n)} + \left(\Phi(x^{(n+1)}) - u^{(n+1)}\right)$.
    $j = j + 1$
end for

In practice, only $M = 1$ inner iteration is often used for updating x and u. This is because the Split Bregman method is robust to numerical imprecision and the Bregman parameter update often obviates the need for precision in Equation (2.26) [35].

Split Bregman has been shown to have faster convergence for cost functions with ℓ_1-regularization compared to continuation methods. Smart choice of parameter μ can reduce the condition number for the subproblem (2.28) and avoid numerical instabilities that arise with continuation schemes. The speed gains from Split Bregman

result from decoupling the ℓ_1 denoising problem from the ℓ_2 data consistency term, each of which can be independently solved faster via alternating minimization than a joint minimization.

2.3.2 Alternating Direction Method of Multipliers

Additional variable splits can be added to decouple exploitable structures in the cost function. Speed gains usually result as each inner problem becomes easier to solve, making outer iterations faster. However, the addition of many auxiliary variables can have diminishing speed returns, complicate parameter tuning, occupy lots of memory, and obfuscate convergence guarantees.

Consider a more general optimization problem:

$$\min f(x) + g(z) \tag{2.32}$$

$$\text{s.t. } \mathbf{M}x = z \tag{2.33}$$

with closed, proper convex functions $f : \mathcal{R}^n \mapsto (-\infty, +\infty]$ and $g : \mathcal{R}^s \mapsto (-\infty, +\infty]$ and matrix $\mathbf{M}$ with size $s \times n$. Then the Alternating Direction Method of Multipliers (ADMM) solution consists of the following recursive steps:

$$x^{(n+1)} = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} \{ f(x) + \langle p^{(n)}, Mx \rangle + \frac{\lambda}{2} \left\| \mathbf{M}x - z^{(n)} \right\|^2 \tag{2.34}$$

$$z^{(n+2)} = \underset{z \in \mathcal{R}^s}{\operatorname{argmin}} \{ g(z) - \langle p^{(n)}, z \rangle + \frac{\lambda}{2} \left\| \mathbf{M}x^{(n+1)} - z \right\|^2 \tag{2.35}$$

$$p^{(n+1)} = p^{(n)} + \lambda \left(\mathbf{M}x^{(n+1)} - z^{(n+1)} \right) \tag{2.36}$$

for $\lambda > 0$ [36] [37]. The objective is guaranteed to approach the optimal value, and the iterates approach feasibility, i.e., constraint satisfaction.

Variable splitting approaches for image reconstruction can be designed to satisfy equivalence with ADMM, thus ensuring convergence. In particular, if the matrix describing the ensemble variable splitting scheme has full rank [38], we can rely on its equivalence with ADMM.

2.4 Dynamic Contrast Enhanced MRI

Dynamic Contrast Enhanced (DCE) MRI is a method of MRI in which a series of images are acquired in succession to characterize changes in contrast agent uptake over time. During the scan, the patient is intravenously injected with a contrast agent that shortens T_1 in the surrounding tissues. Because the amount of T_1 shortening is proportional to the concentration of contrast agent, it is possible to infer the levels of contrast uptake for a particular region. For breast and liver DCE-MRI studies, gadolinium diethylenetriaminepentaacetic acid (Gd-DTPA) is commonly used as the

contrast agent. By acquiring a series of images over time, it is possible to visualize rates at which areas of the body uptake the contrast agent and quantitative changes in contrast agent concentration change over time. These contrast changes are then used to measure pharmacokinetic parameters, for use in diagnosis and radiation therapy assessment. DCE-MRI is a useful tool because of its ability to depict both physiology and morphology.

2.4.1 Effect of Contrast Agent on Image Signal

For a given voxel, the concentration of contrast agent in the tissue, $C_{\mathrm{t}}(t)$, affects the $T_1(t)$ value as follows:

$$\frac{1}{T_1(t)} = \frac{1}{T_{10}} + r_1 C_{\mathrm{t}}(t). \tag{2.37}$$

Relaxivity parameter r_1 is particular to the contrast agent (e.g., 4.50 mM $^{-1}$ s^{-1} at 3T for Gd-DTPA [39]). T_{10} is the inherent spin-lattice constant of the tissue (e.g., 812 ms for liver at 3T, 1194 ms for kidney at 3T [40]), and $T_1(t)$ is the dynamically changing spin-lattice constant. As the tissue contrast agent concentration increases, the T_1 value of the tissue decreases.

For steady state Spoiled Gradient Recalled (SPGR) sequences, the MR signal equation is:

$$m_0 = \rho_0 \sin(\theta) \left(\frac{1 - E_1}{1 - E_1 \cos(\theta)} \right) e^{-\frac{T_E}{T_2^*}}; \qquad E_1 \triangleq e^{-\frac{T_R}{T_1(t)}}. \tag{2.38}$$

The signal intensity, m_0, is a function of the proton density, ρ_0, the flip angle, θ, the dynamically changing spin-lattice constant, $T_1(t)$, repetition time, T_R, echo time, T_E, and effective T_2 (which includes effects of field inhomogeneities). The presence of contrast agent lowers T_1 values, and subsequently E_1 values, increasing the signal m_0, making the tissue appear brighter. We group together the product of several unknowns in the signal equation as $\kappa_6(\vec{r}) = \rho_0(\vec{r})e^{-\frac{T_E}{T_2^*}}$. From initial pre-enhancement image estimates and T_1 maps, we can precompute this quantity. An alternative approach is to estimate this quantity from post-contrast data.

For DCE imaging, a T_1 mapping scan with multiple flip angles is done prior to contrast injection. This scan is usually short in duration, approximately 10 seconds for each flip angle, so patients are trained to hold their breath for the scan. This allows for isolation of E_1 term from measured m_0 in Equation (2.38).

2.4.2 Breast Pharmacokinetic Model

For the breast, a popular pharmacokinetic model is the single-compartment Tofts model [41]. We apply this model voxel-by-voxel. This model describes three pharmacokinetic parameters: the volume transfer constant, K^{trans}, the flux rate constant, k_{ep}, and the volume of extravascular extracellular space per unit volume of tissue, v_e. These parameters are interrelated: $k_{ep} = K^{trans}/v_e$, so only two need to be calculated. These parameters describe the relationships between various time series. The primary time series of interest is the concentration of contrast agent in the breast tissue over time, $C_t(t)$. This quantity is related to the contrast agent concentration

in the blood plasma near the breast, $C_p(t)$ [2]. $C_p(t)$ is closely related to the arterial input function (AIF) through h, the hematocrit (or volume percentage of cells in blood), as follows:

$$\mathrm{AIF} = (1 - h)\, C_{\mathrm{p}}(t). \tag{2.39}$$

A value of $h = 0.45$ has used in previous breast DCE-MRI studies [42].

To estimate the values for K^{trans}, k_{ep}, and v_e, it is necessary to measure or estimate two time series: $C_p(t)$ and $C_t(t)$. K^{trans}, k_{ep}, and v_e describe how $C_p(t)$ and $C_t(t)$ are related:

$$C_{\mathrm{t}}(t) = K^{\mathrm{trans}} C_{\mathrm{p}}(t) * (e^{-k_{\mathrm{ep}}t} u(t)). \tag{2.40}$$

Here, $*$ denotes convolution and $u(t)$ is the unit step function. The unknown time delay of the bolus arrival (first arrival of concentration agent in the breast) is hidden within $C_p(t)$. Tissue concentration can be measured from reconstructed DCE-MRI images by manual selection of the tissue voxels. Several different approaches are used to estimate the AIF, and consequently $C_p(t)$.

The AIF can be directly measured from a manually chosen region of interest (ROI) within internal mammary arteries or axillary arteries, but these small vessels can be affected by motion. Alternatively, the AIF can be derived from measurements in

other vessels, such as the descending or ascending aorta, which can be included in the FOV [43]. Yet another approach is to avoid estimating patient-specific AIFs and use population-averaged values [44].

2.4.2.1 Method of Moments Approach to Tissue Concentration

The noisy measurements of DCE imaging, y, depend on the unknown DCE image, m_0, through system $\mathbf{A}$ in a setting with complex Gaussian noise:

$$y = \mathbf{A}m_0 + \varepsilon, \quad \varepsilon \sim \mathcal{CN}. \tag{2.41}$$

Overall system matrix $\mathbf{A}$ can consist of many components, such as $\mathbf{A}_{MC}$ and $\mathbf{S}$ as described for multi-coil imaging in (2.21). As described in Section 2.2.3, the penalized likelihood estimator for m_0 is:

$$\hat{m}_0 = \underset{m_0}{\operatorname{argmin}} \|y - \mathbf{A}m_0\|^2 + \lambda R(m_0). \tag{2.42}$$

From the estimate for m_0, we seek to estimate the unknown pharmacokinetic parameters through a method of moments approach to estimating tissue concentration, $C_{\mathrm{t}}(t)$. With a precomputed $\kappa_6(\vec{r})$, we isolate $C_{\mathrm{t}}(t)$ by combining (2.37) and (2.38) as follows:

$$\hat{E}_1(\vec{r},t) = \frac{\left(\frac{\hat{m}_0(\vec{r},t)}{\kappa_6(\vec{r})\sin(\theta(\vec{r}))}\right) - 1}{\left(\frac{\hat{m}_0(\vec{r},t)\cos(\theta(\vec{r}))}{\kappa_6(\vec{r})sin(\theta(\vec{r}))}\right) - 1} \tag{2.43}$$

$$\hat{C}_t(\vec{r},t) = -\frac{1}{T_R r_1}\ln\left(\frac{\hat{E}_1(\vec{r},t)}{e^{-\frac{T_R}{T_{10}(\vec{r})}}}\right). \tag{2.44}$$

From estimated tissue contrast agent concentration, $\hat{C}_t$, we can estimate the remaining pharmacokinetic parameters through the variable projection method described below.

2.4.2.2 Introduction to Variable Projection

Variable projection is a method which separates the linear variables c_j from the nonlinear variables $\mathbf{d}$ within the following problem [45]:

$$\min_{\mathbf{c},\mathbf{d}} \left\| a_i - \sum_{j=1}^{n} c_j b_j(\mathbf{d};t_i) \right\|^2. \tag{2.45}$$

Each measurement a_i corresponds to independent variables t_i through a linear combination of n nonlinear functions, $b_j(\mathbf{d};t_i)$. Collecting the nonlinear contributions $b_j(\mathbf{d};t_i)$ into a matrix $\mathbf{B}(\mathbf{d})$, we can describe the problem as:

$$\hat{\mathbf{d}} = \underset{\mathbf{d}}{\operatorname{argmin}} \left\| \left(\mathbf{I} - \mathbf{B}(\mathbf{d})\mathbf{B}(\mathbf{d})^{+} \right) \mathbf{a} \right\|^{2} \tag{2.46}$$

where $\mathbf{B}(\mathbf{d})^{+}$ is the generalized Moore-Penrose pseudo-inverse of $\mathbf{B}(\mathbf{d})$. Equation (2.46) minimizes the variable projection of $\mathbf{d}$ into the orthogonal complement of $\mathbf{B}(\mathbf{d})$.

The linear variable estimates are computed based on the optimal nonlinear variable values:

$$\hat{\mathbf{c}} = \mathbf{B}(\hat{\mathbf{d}})^{+}\mathbf{a}. \tag{2.47}$$

Variable projection is a powerful method because the optimization for the nonlinear variables is performed in a reduced space. Furthermore, the minima for the reduced problem are better defined than the full nonlinear estimation problem, often resulting in faster convergence [45].

2.4.2.3 Variable Projection for Breast Pharmacokinetic Parameter Estimation

For variable projection, we will estimate the pharmacokinetic parameters voxel-by-voxel, omitting argument $\vec{r}$. Recall Equation (2.40):

$$C_{\mathrm{t}}(t) = K^{\mathrm{trans}} b(k_{\mathrm{ep}}; t) \quad \text{where,}$$
$$b(k_{\mathrm{ep}}; t_i) = C_{\mathrm{p}}(t) * (e^{-k_{\mathrm{ep}} t} u(t))\Big|_{t=t_i}. \tag{2.48}$$

Recognizing k_{ep} as the single nonlinear variable and K^{trans} as the single linear variable, we can estimate the two quantities using variable projection as follows:

$$\hat{k_{\mathrm{ep}}} = \underset{k_{\mathrm{ep}}}{\operatorname{argmin}} \left\| \left(\mathbf{I} - \mathbf{B}(k_{\mathrm{ep}}) \mathbf{B}(k_{\mathrm{ep}})^{+} \right) \hat{\mathbf{C}}_{\mathbf{t}} \right\|^2 \tag{2.49}$$

$$\hat{K^{\mathrm{trans}}} = \mathbf{B}(\hat{k_{\mathrm{ep}}})^{+} \hat{\mathbf{C}}_{\mathbf{t}}. \tag{2.50}$$

For convenience, we arrange values of $\hat{C_{\mathrm{t}}}(t)$ as a vector: $\hat{\mathbf{C}}_{\mathbf{t}} = \left[\hat{C}_t(t_1) \cdots \hat{C}_t(t_{N_t}) \right]^T$. Because $\mathbf{B}(k_{\mathrm{ep}})$ is a column vector, the solution for $\hat{k_{\mathrm{ep}}}$ can be simplified to maximization of an inner product as follows:

$$\hat{k_{\text{ep}}} = \underset{k_{\text{ep}}}{\text{argmin}} \left\| \left(\mathbf{I} - \mathbf{B}(k_{\text{ep}}) \frac{\mathbf{B}(k_{\text{ep}})^T}{\mathbf{B}(k_{\text{ep}})^T \mathbf{B}(k_{\text{ep}})} \right) \mathbf{C_t} \right\|^2 \tag{2.51}$$

$$= \underset{k_{\text{ep}}}{\text{argmin}} \left\| \mathbf{C_t} \right\|^2 - \frac{\left(\mathbf{B}(k_{\text{ep}})^T \mathbf{C_t} \right)^2}{\mathbf{B}(k_{\text{ep}})^T \mathbf{B}(k_{\text{ep}})} \tag{2.52}$$

$$= \underset{k_{\text{ep}}}{\text{argmax}} \frac{\left(\mathbf{B}(k_{\text{ep}})^T \mathbf{C_t} \right)^2}{\mathbf{B}(k_{\text{ep}})^T \mathbf{B}(k_{\text{ep}})} \tag{2.53}$$

$$= \underset{k_{\text{ep}}}{\text{argmax}} \left| \left(\frac{\mathbf{B}(k_{\text{ep}})}{\|\mathbf{B}(k_{\text{ep}})\|} \right)^T \mathbf{C_t} \right|^2 \tag{2.54}$$

Since this is a one-dimensional optimization problem, we choose to solve it with a simple global search over the feasible space for k_{ep}, defined by values found in literature. With the estimate for k_{ep}, we compute the estimate for K^{trans} using (2.50). This method was outlined by [2].

2.4.3 Liver Pharmacokinetic Model

In contrast to the breast, the liver is commonly modeled as a dual-input, single-compartment system [46]. We apply this model voxel-by-voxel. Blood enters the liver through two main blood supplies, the aorta and the portal vein. Aortic blood supply reaches the liver first through the hepatic artery, and the portal vein carries blood from the gastrointestinal tract and spleen to the capillary beds in the liver for detoxification. We model liver in its entirety, including the tissue, capillaries, and extracellular extravascular space (EES), as one compartment. From this single

compartment, outflow or leakage occurs. This relationship can be modeled with the following differential equation:

$$\frac{dC_L(t)}{dt} = k_{1a}C_a(t) + k_{1p}C_p(t) - k_2C_L(t). \tag{2.55}$$

$C_a(t)$ and $C_p(t)$ represent the contrast agent concentration in the aorta and portal vein. $C_L(t)$ represents the concentration agent enhancement curve for a particular voxel in the liver. Pharmacokinetic parameters k_{1a} and k_{1p} are the aortic and portal venous inflow rate constants, and k_2 is the outflow rate constant. Solving for $C_L(t)$ and adding two parameters, τ_a and τ_p, to describe the transit time along each of the blood input routes, we obtain:

$$\begin{aligned} C_L(t) &= \int_0^t \left[k_{1a}C_a(t'-\tau_a) + k_{1p}C_p(t'-\tau_p)\right] e^{-k_2(t-t')}dt' \\ &= \left[k_{1a}(\vec{r})C_a(t-\tau_a(\vec{r})) + k_{1p}(\vec{r})C_p(t-\tau_p(\vec{r}))\right] * e^{-k_2(\vec{r})t} \end{aligned} \tag{2.56}$$

This yields five (N_{PK}) unknown pharmacokinetic parameters at each liver voxel: k_{1a}, k_{1p}, k_2, τ_a, τ_p. Note that portal vein input always occurs after the aortic input, so τ_p is always larger than τ_a.

To measure liver pharmacokinetic parameters from a dynamic image, we first estimate the contrast enhancement curves $C_a(t)$ and $C_p(t)$. Then we can use a combination of a method of moments approach described in Section 2.4.2.1 to estimate $C_L(t)$ and variable projection to estimate the five pharmacokinetic parameters from $\hat{C}_L(t)$, $C_a(t)$, and $C_p(t)$. The method of moments approach to estimating $C_L(t)$ is the same

as for breast, described in Section 2.4.2.1. The variable projection method for liver pharmacokinetics is described below.

2.4.3.1 Variable Projection for Liver Pharmacokinetic Parameter Estimation

From $C_L(t)$, we can use variable projection [45] to estimate the five unknown pharmacokinetic parameters separately for each voxel. For the liver pharmacokinetic estimation problem, $n = 2$, and the following quantities correspond to the variable projection framework described in Equation (2.45):

$$\mathbf{c} = \begin{bmatrix} k_{1a} \\ \\ k_{1p} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \tau_a \\ \\ \tau_p \\ \\ k_2 \end{bmatrix} \tag{2.57}$$

We construct vector $\mathbf{C_L}$ similar to $\mathbf{C_t}$ for the breast. We also define two new vectors for ease of notation:

$$v_a(\mathbf{d}; t_i) \triangleq C_a(t - \tau_a) * e^{-k_2 t}\Big|_{t=t_i}, \qquad v_p(\mathbf{d}; t_i) \triangleq C_p(t - \tau_p) * e^{-k_2 t}\Big|_{t=t_i}. \tag{2.58}$$

In matrix form, we have the following solution for the linear parameters $\mathbf{c}$:

$$\mathbf{B}(\mathbf{d}) = \begin{bmatrix} | & | \\ \mathbf{v}_a(\mathbf{d}) & \mathbf{v}_p(\mathbf{d}) \\ | & | \end{bmatrix} \tag{2.59}$$

$$\hat{\mathbf{c}} = \mathbf{B}(\mathbf{d})^{+}\hat{\mathbf{C}}_{\mathbf{L}}. \tag{2.60}$$

The generalized Moore-Penrose pseudo-inverse of $\mathbf{B}(\mathbf{d}) \in \mathbb{R}^{N_t \times 2}$, $\mathbf{B}(\mathbf{d})^{+} \in \mathbb{R}^{2 \times N_t}$ is:

$$\mathbf{B}(\mathbf{d})^{+} = \left(\mathbf{B}(\mathbf{d})^T\mathbf{B}(\mathbf{d})\right)^{-1}\mathbf{B}(\mathbf{d})^T \tag{2.61}$$

$$= \left(\begin{bmatrix} \mathbf{v}_a(\mathbf{d})^T\mathbf{v}_a(\mathbf{d}) & \mathbf{v}_a(\mathbf{d})^T\mathbf{v}_p(\mathbf{d}) \\ \mathbf{v}_p(\mathbf{d})^T\mathbf{v}_a(\mathbf{d}) & \mathbf{v}_p(\mathbf{d})^T\mathbf{v}_p(\mathbf{d}) \end{bmatrix}\right)^{-1} \begin{bmatrix} -\mathbf{v}_a(\mathbf{d})^T- \\ -\mathbf{v}_p(\mathbf{d})^T- \end{bmatrix} \tag{2.62}$$

To solve for nonlinear parameters, $\mathbf{d}$, we minimize the following cost functions:

$$\mathbf{d} = \underset{\mathbf{d}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{C}_{\mathbf{L}} - \mathbf{B}(\mathbf{d})\mathbf{B}(\mathbf{d})^{+}\mathbf{C}_{\mathbf{L}}\right\|^2 \tag{2.63}$$

$$= \underset{\mathbf{d}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{C}_{\mathbf{L}} - \begin{bmatrix} | & | \\ \mathbf{v}_a(\mathbf{d}) & \mathbf{v}_p(\mathbf{d}) \\ | & | \end{bmatrix} \left(\begin{bmatrix} \mathbf{v}_a(\mathbf{d})^T\mathbf{v}_a(\mathbf{d}) & \mathbf{v}_a(\mathbf{d})^T\mathbf{v}_p(\mathbf{d}) \\ \mathbf{v}_p(\mathbf{d})^T\mathbf{v}_a(\mathbf{d}) & \mathbf{v}_p(\mathbf{d})^T\mathbf{v}_p(\mathbf{d}) \end{bmatrix}\right)^{-1} \begin{bmatrix} -\mathbf{v}_a(\mathbf{d})^T- \\ -\mathbf{v}_p(\mathbf{d})^T- \end{bmatrix} \mathbf{C}_{\mathbf{L}}\right\|^2 \tag{2.64}$$

A sweep over the parameters would be cumbersome for 3D $\mathbf{d}$ in Equation (2.57). Other methods, such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm for unconstrained nonlinear problems may prove useful for this step. Because $\tau_a < \tau_p$, we can reduce the feasible set by one half. Other constraints on the feasible set of $\mathbf{d}$, such as using bounds based on values found in literature, may also prove useful.

2.4.4 Motion Compensation for Abdominal Imaging

Liver imaging presents a greater challenge than breast imaging due to the greater effect of respiration. Beyond the usual degradatory effects of motion, respiration is complicated by hysteresis. The exertion of the diaphragm creates differences in the relationship between lung volume and air pressure during inhalation and exhalation. This also results in a different anatomical trajectories [47]. Ignoring irregularities, breathing can be modeled as smoothly cycling between a small number of discrete respiratory phases [48].

2.5 Previous Approaches to Motion Compensated Dynamic Image Reconstruction

Here we review several existing methods for motion compensation in image reconstruction. These methods are presented for completeness and as a theoretical basis for comparison.

2.5.1 Navigators and Respiratory Gating

One of the simplest approaches to dealing with motion is to categorize the data according to respiratory state and only use data correspoding to the desired respiratory state. This process, known as "gating", circumvents the need to compensate for motion at all, assuming that little motion occurs among the gated data. Though it is simple to implement, this can complicate pharmacokinetic parameter estimation, particularly if crucial temporal changes occur during an undesired respiratory state.

Respiratory gating requires information about the patient's underlying respiratory state. This information is described with a navigator signal. In the past, this information was measured by means of an external device, such as a belt or bellows, or additional RF-pulses. The experimental data presented in Chapter 5 uses a navigator measured during adapted pre-winder gradients for phase encodes [49], pictured in Figure 2.8.

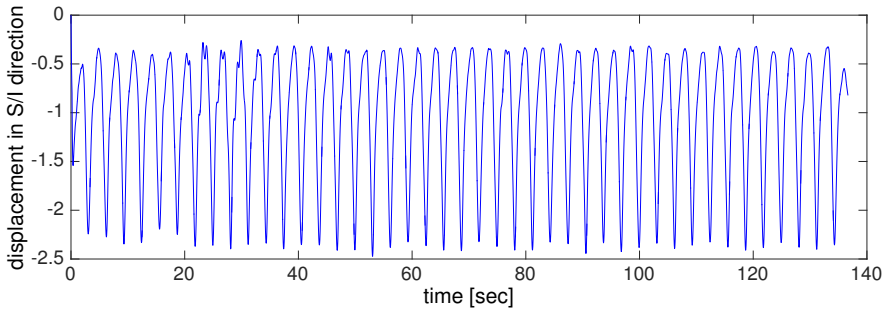


Figure 2.8: A Butterfly navigator from a pediatric abdominal imaging dataset.

Experiments proposed in Chapter 7 use self-navigation, in which the respiratory

state is estimated without additional measurements. For trajectories that repeatedly sample the center of k-space, the DC value works as a good proxy for respiratory state. As organs move in and out of the FOV, the total magnetization (DC value) changes accordingly.

2.5.2 Joint Motion Estimation and Image Reconstruction

One sophisticated approach to compensating for motion in image reconstruction is to estimate the underlying dynamic image and the motion corrupting the image of interest simultaneously. This approach may improve motion and image estimates but is also very challenging [50] [51].

2.5.3 Golden Angle Radial Sparse Parallel Imaging (GRASP)

GRASP, sometimes called iGRASP (iterative Golden angle RAdial Sparse Parallel imaging), [52] is a popular method for dynamic image reconstruction that has been applied to various MR applications, such as prostate [53], liver [54], and breast[52]. The two core ideas of GRASP are the use of golden-angle radial sampling and a penalty on temporal differences.

Golden angle sampling is radial sampling in which the angular increment of the spokes is the golden angle, $\pi(3 - \sqrt{5})$ radians. The use of golden-angle radial sampling in a free-breathing paradigm results in sampling patterns that are robust to motion and permit compressed-sensing reconstruction methods. More specifically, for a Fibonacci number of spokes, the resulting sampling patterns are somewhat

angularly uniform and have increasing density toward the center of k-space. The randomness of the undersampling reduces the effect of respiratory gating on the gated sampling pattern. Furthermore, the formula for generating spoke angles is simple and easily reproduceable. In the absence of gradient-induced eddy currents, which cause shifts in actual sample locations, radial spokes repeatedly sample the center-of-k-space, allowing for easy self-navigation. This golden-angle stack-of-stars trajectory is pictured in Figure 2.6.

The main contribution of GRASP is the application of temporal regularization to the golden angle radial sampling framework. GRASP reconstructs dynamic image x by minimizing the following cost function:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\| y - \mathbf{FS}x \right\|^2 + \lambda \left\| \mathbf{C}_t x \right\|_1 , \tag{2.65}$$

where $\mathbf{C}_t$ computes temporal finite differences, $\mathbf{S}$ applies sensitivity encoding identically over each temporal frame, and $\mathbf{F}$ is the NUFFT operator defined on the sampling pattern assigned to each frame of x. Feng et al. initially minimized this cost fuction using nonlinear conjugate gradient in Matlab, but later implemented it via L-BFGS in C++ combined with coil compression and slice-by-slice parallelization for clinical use.

GRASP performed better than methods without temporal considerations on physician rated image quality and temporal fidelity metrics [52]. As expected, results demonstrated a reduction of streaking artifacts as well. However, GRASP does not account for motion between neighboring frames and its results can exhibit motion

artifacts.

2.5.4 Extra-Dimensional GRASP (XD-GRASP)

Recently, the GRASP method was extended to include respiratory and cardiac dimensions in eXtra-Dimensional GRASP (XD-GRASP) [55]. Parameterization of the unknown object along dimensions of motion (rather than time) reduced motion artifacts, increased organ edge sharpness, and improved vessel clarity. In further work [56] [57], Feng et al. parameterized a liver DCE-MRI image with a combination of time and respiratory state, also showing improved image quality. This method is also called 5D MRI due to the five dimensions of the estimated object: x, y, and z in space, time, and respiratory state.

We illustrate this concept with a 2D Shepp-Logan phantom, modified to include contrast dynamics, shown in Figure 2.9. Tiles progress in time from left to right, and occupy different motion states from top to bottom. In a single row, highlighted for convenience in yellow, the motion state is fixed but time varies. Among this set of image volumes, only contrast changes occur. Along a single column, also highlighted in yellow, the temporal frame is fixed, but motion state varies, resulting in only motion changes between the selected volumes.

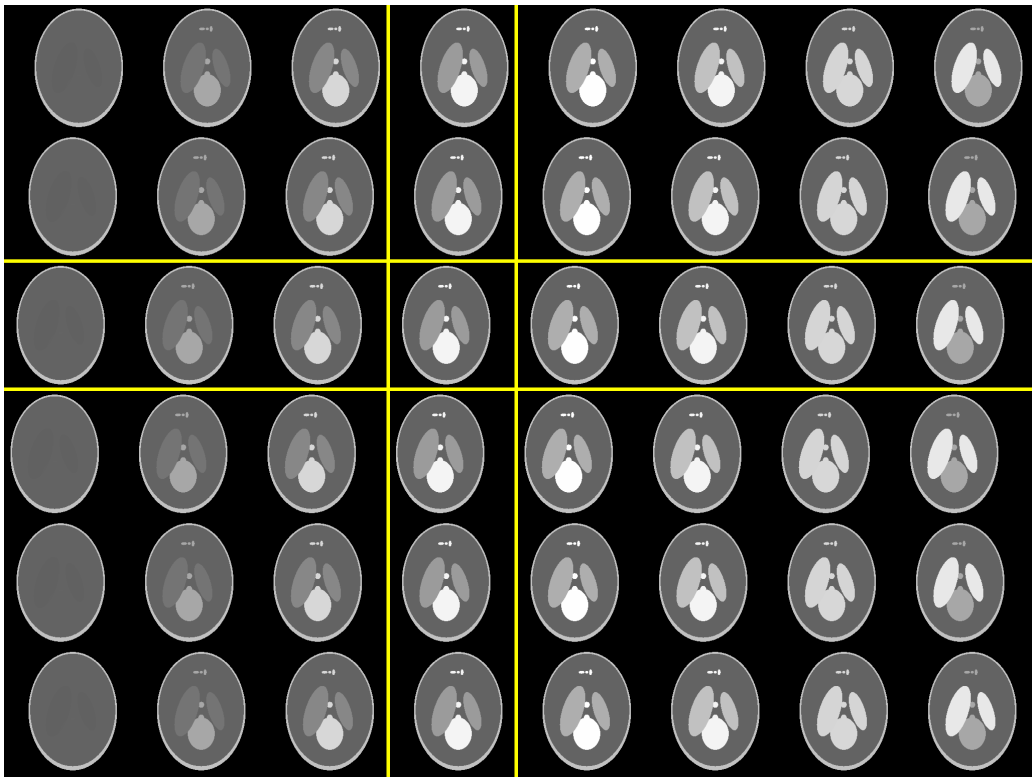


Figure 2.9: Illustration of 5D MRI, separating respiratory motion from contrast changes across time. The x-axis represents discrete time values. The y-axis represents discrete motion state values.

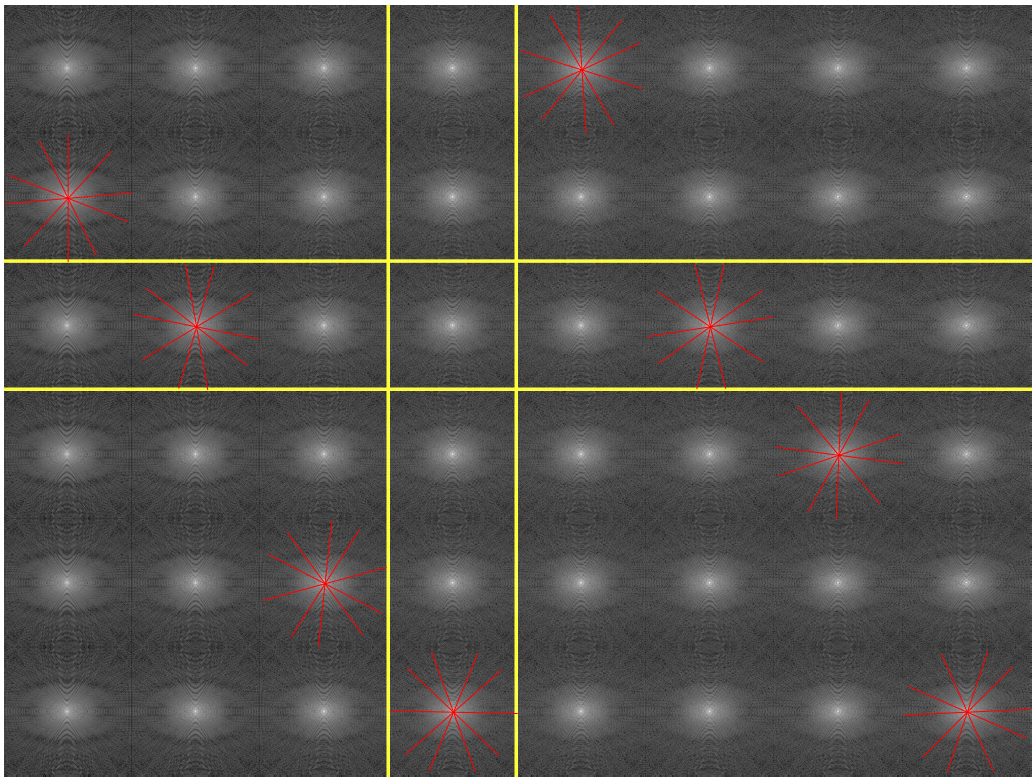


Figure 2.10: Illustration of 5D MRI, showing the samples that can be realistically acquired from k-space corresponding to Figure 2.9.

Because only one respiratory phase can be sampled at any given time, the estimation problem for a 5D respiratory-DCE object is quite ill-conditioned. Figure 2.10 emphasizes the degree of undersampling, showing an example Golden Angle radial sampling pattern for the k-space corresponding to the 5D object in Figure 2.9. Because sampling is limited to certain respiratory states at certain time values, the remaining motion-time state pairs must be estimated using regularization. The cost function for DCE XD-GRASP penalizes changes along the additional respiratory

dimension as well as time:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\| y - \mathbf{FS}x \right\|^2 + \lambda_1 \left\| \mathbf{C}_t x \right\|_1 + \lambda_2 \left\| \mathbf{C}_r x \right\|_1 . \tag{2.66}$$

$\mathbf{C}_r$ computes finite differences across neighboring respiratory states. For N_{resp} desired respiratory states, the unknown x has size $[N_x \times N_y \times N_z \times N_t \times N_{\text{resp}}]$, N_{resp} larger than the unknown DCE object in GRASP without additional data. Feng et al. did not report great computational burden for few temporal frames, $N_t \leq 40$. For accurate pharmacokinetic parameter estimation, finer temporal resolution (i.e., $N_t > 40$ for a 2 minute scan) may prove necessary, which in turn may require additional regularization.

2.5.5 Direct Pharmacokinetic Parameter Estimation

Rather than measuring pharmacokinetic parameters from reconstructed dynamic images, several works have proposed directly estimating pharmacokinetic values from raw k-space data. This has been demonstrated in the brain for two pharmacokinetic parameters [58] and prostate for four pharmacokinetic parameters [59].

For the direct estimation of brain pharmacokinetic parameters, [58] minimized:

$$(\hat{K_t}, \hat{v_p}) = \underset{K_t, v_p}{\operatorname{argmin}} \left\| y - \mathbf{FS}P(K_t, v_p) \right\|^2 + \lambda_1 \left\| \mathbf{W}K_t \right\|_1 + \lambda_2 \left\| \mathbf{W}v_p \right\|_1 , \tag{2.67}$$

with a sparsifying wavelet transform $\mathbf{W}$ and nonlinear operator P representing the brain pharmacokinetic model and MR physics. This optimization was performed

using limited-memory BFGS.

This work leaves open the opportunity for more sophisticated regularization and tailored optimization. Additionally, DCE brain images are not as susceptible to motion as other organs, because head motion can be minimized due to restraints. For this reason, [58] does not compensate for motion. Additionally, the brain pharmacokinetic model was used for the entire FOV, a strategy that may not translate well to abdominal imaging, where many organs with different pharmacokinetic behavior are in the FOV.

For prostate direct pharmacokinetic estimation, [59] used a Bayesian inference method that incorporated a low rank prior for the dynamic signal and a Gaussian prior for the pharmacokinetic parameters. This method was initialized with global population-based pharmacokinetic parameter values, and required tuning of various threshold parameterss. The Bayesian inference was implemented as a Metroplis-Hastings algorithm and demonstrated significant computational cost, with some experiment calculations lasting 17 hours.

This method did not explicitly accomodate for motion, but showed increased robustness in the face of simulated abdominal motion compared to indirect pharmacokinetic estimation methods. Furthermore, respiratory motion in the prostate is less problematic than respiratory motion in abdominal imaging: 2.7 ± 1.9 mm for prostate in the superior/inferior (S/I) direction [60] as opposed to 17.9 ± 5.1 mm for liver [61]. This work did not distinguish between voxels that adhere to the given pharmacokinetic model and those that do not: the prostate pharmacokinetic model is used over

the entire pelvic region.

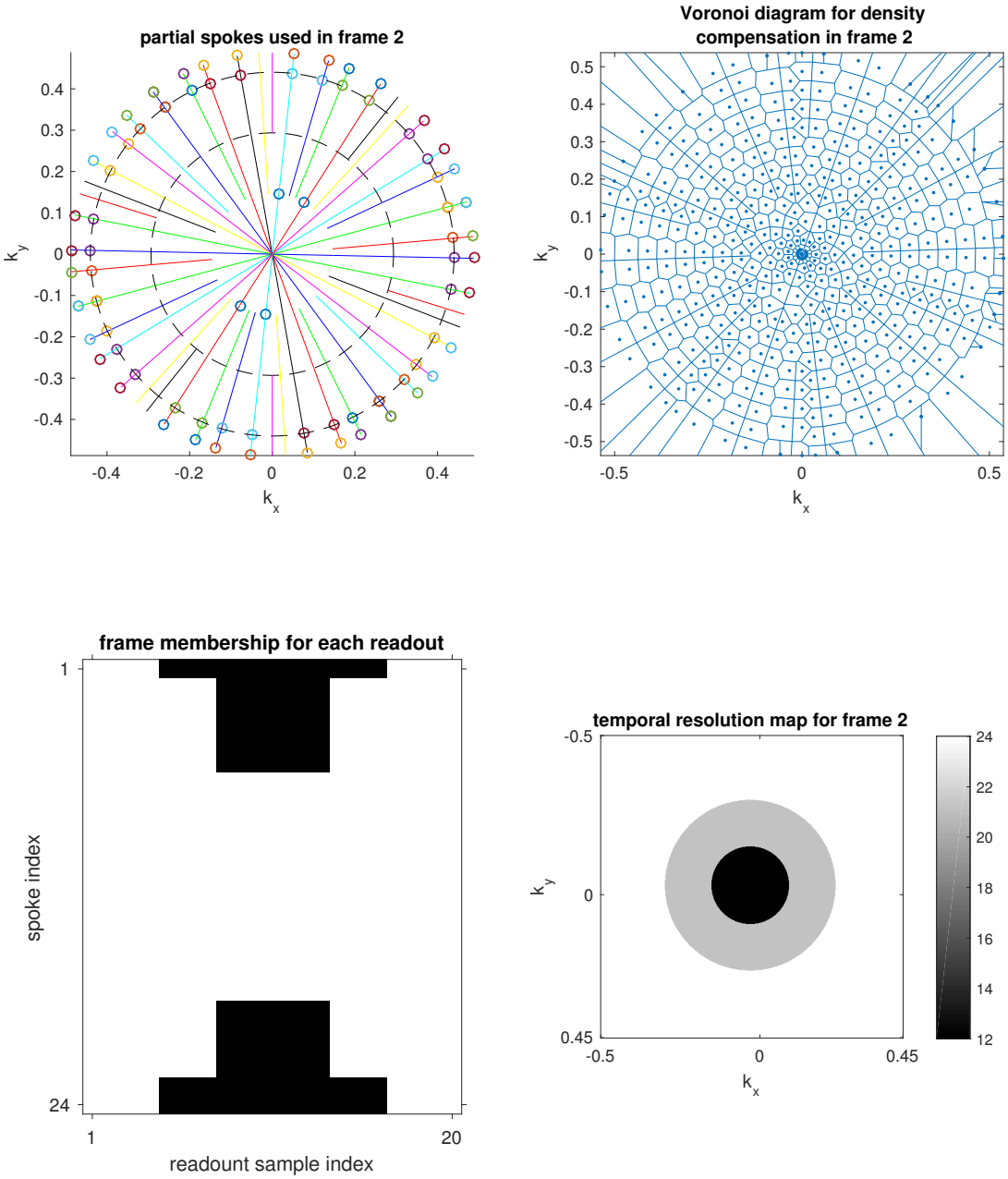


Figure 2.7: Radial datasharing for a toy example. (a) full and partial pokes included in datashared frame 2 of 3; (b) Voronoi map used for density compensated gridding reconstruction; (c); frame membership map for readout indices; the sideways hourglass shape describes the growing inclusion of k-space data in high-frequency k-space; (d) map indicating number of contributing spokes in different annuli.

Chapter 3

Efficient Support-Constrained Image Reconstruction

3.1 Theoretical Motivations

[1] The conventional method for storing image data is in a two-dimensional array that describes the values of all $N_x \times N_y$ pixels in the field of view. However the region of interest in MRI images is defined by edges of the human body, a object that does not hold a rectangular shape. MRI pulse sequences are designed to select a FOV that encapsulates the entire bodily cross-section of interest. Traditional image reconstruction methods are designed to reconstruct the entirety of this FOV, without regard to which of these pixels are located within the body and are of interest for diagnosis and to which of the pixels lie outside the body and can be safely assumed to be air. Ideally, we would like to estimate a reduced number of pixels within the image, spending computation only on the unknown pixel values inside the body. However, it is still necessary to display these pixel values in a rectangular image

[1] This chapter is based on [8].

and to preserve two-dimensional spatial relationships for multiplying by the system matrix described in (2.21). To do so, we define a logical array, called a "mask" over the rectangular matrix that indicates which pixels will be estimated and which will be assumed to be air.

Though it is not possible to know exactly which pixels lie in and outside the body, we can select a mask from a fast approximate image. A fast approximate image can obtained from the sum-of-squares (SoS) of the N_c images reconstructed from zero-filling k-space and performing an IFFT.

3.2 Static MRI Variable Splitting Algorithm

To introduce the support constraint into SENSE reconstruction, we construct the following system model:

$$y = \mathbf{FSM}x + \varepsilon \tag{3.1}$$

N_{mask} denotes the number of pixels in the support constraint.

N_p denotes the number of pixels in the rectangular image to be reconstructed.

N_{samp} denotes the number of samples collected, i.e., the number of non-zero elements in the sampling pattern.

N_c is the number of receiving coils.

y is a $N_{\text{samp}}N_c$ length vector containing the data from the N_c sensitivity coils.

$\mathbf{F}$ is an $N_{\text{samp}}N_c \times N_pN_c$ undersampled Fourier encoding matrix. In one dimension,

this would be equivalent to omitting rows from the standard DFT matrix. $\mathbf{F} = \mathbf{I}_{N_c} \otimes \tilde{\mathbf{F}}$, where $\tilde{\mathbf{F}}$ is the $N_{\text{samp}} \times N_p$ undersampled Fourier encoding matrix, where $N_{\text{samp}} < N_p$.

$\mathbf{S}$ is a stack of diagonal matrices, $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_{N_c}$, each of which contains the sensitivity data of each of the coils. Each $\mathbf{S}_i$ is $N_p \times N_p$, and $\mathbf{S}$ is $N_p N_c \times N_p$.

$\mathbf{M}$ is the $N_p \times N_{\text{mask}}$ masking matrix that embeds a short vector x (of length N_{mask}) into a longer vector corresponding to the 2D indexing of the image (of length N_p).

ε is complex Gaussian noise from thermal fluctuations in the patient.

Note that this matrix $\mathbf{M}$ is a column-reduced identity matrix such that $\mathbf{M}'\mathbf{M} = \mathbf{I}_{N_{\text{mask}}}$ and $\mathbf{M}\mathbf{M}' = \texttt{diag(mask(:))}$, where N_{mask} is the number of pixels in the mask. Note also that for $N_c > \frac{N_p}{N_{\text{samp}}}$, this optimization problem is over-determined. To improve the conditioning of this problem as motivated in Section 2.2.3, we introduce an ℓ_1 regularization term, which yields the following optimization problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \, ||y - \mathbf{FSM}x||_2^2 + \lambda ||\mathbf{C}_s \mathbf{M}x||_1 \tag{3.2}$$

$\mathbf{C}_s$ is the regularization operator, chosen such that $\mathbf{C}_s'\mathbf{C}_s$ is circulant to facilitate fast computation.

λ is the regularization parameter that controls the balance between the fitting the data and the a priori assumptions.

This cost function is challenging to minimize directly for several reasons. The Hessian for the datafit term, $\mathbf{M}'\mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S}\mathbf{M}$, is highly shift-variant and lacking any easily exploitable structure (e.g., circulant, diagonal). Furthermore, the ℓ_1 -norm regularization term is non-differentiable.

Many proposed methods, such as [62], [63], for ℓ_1 regularized problems are computationally intensive and slow.

3.2.1 Variable Splitting

To make Equation (3.2) more tractable, we introduce auxiliary variables related to x such that when those relations are enforced, the optimization problem is equivalent to that in Equation (3.2). This allows us to decouple the system matrix, $\mathbf{FSM}$, into Hessian matrices with amenable structures. The following constrained optimization problem is equivalent to the unconstrained optimization problem in Equation (3.2):

$$\hat{x} = \underset{x,u,v,z}{\operatorname{argmin}} ||y - \mathbf{F}u||_2^2 + ||z||_1 \quad s.t.\ u = \mathbf{S}\mathbf{M}x,\ v = \mathbf{M}x,\ z = \mathbf{C}_s v \tag{3.3}$$

In other words, Equations (3.2) and (3.3) have the same minimizer, $x^{(*)}$, and the same optimal objective function value. We designed the number of splittings and the nesting structure of the constraints to yield favorable optimization problems for each of the auxiliary variables, as discussed in Section 3.2.3.

3.2.2 Introduction of the Augmented Lagrangian

Through the framework established in [64], and further developed and analyzed in [65] and [66] and applied to Parallel MRI in [67], ((3.3)) yields the following Augmented Lagrangian (AL) function:

$$\mathcal{L}(x,u,v,z,\eta_u,\eta_v,\eta_z;\mu_u,\mu_v,\mu_z) = \frac{1}{2}||y-\mathbf{F}u||_2^2 + \lambda||z||_1 + \frac{\mu_u}{2}||u-\mathbf{SM}x-\eta_u||_2^2 + \frac{\mu_v}{2}||v-\mathbf{M}x-\eta_v||_2^2 + \frac{\mu_z}{2}||z-\mathbf{C}_s v-\eta_z||_2^2 \tag{3.4}$$

The additional variables η_u, η_v, η_z are Augmented Lagrangian-like variables that indicate how closely the auxiliary variables reflect the original constraints in ((3.3)). The parameters μ_u, μ_v, μ_z control the speed of convergence but do not affect the solution.

The AL framework yields the following algorithm:

1: Input $y, \mathbf{F}, \mathbf{S}, \mathbf{M}, \mathbf{C}_s$.
2: Select μ_u, μ_v, μ_z.
3: Initialize x.
4: Initialize $u = \mathbf{SM}x$, $v = \mathbf{M}x$, $z = \mathbf{C}_s v$, and η_u, η_v, $\eta_z = 0$.
5: **for** n iterations **do**
6: $\quad \{x^{(n+1)}, u^{(n+1)}, v^{(n+1)}, z^{(n+1)}\} = \text{argmin}_{x,u,v,z}\, \mathcal{L}(x^{(n)}, u^{(n)}, v^{(n)}, z^{(n)}, \eta_u^{(n)}, \eta_v^{(n)}, \eta_z^{(n)})$.
7: $\quad \eta_u^{(n+1)} = \eta_u^{(n)} - (u^{(n+1)} - \mathbf{SM}x^{(n+1)})$.
8: $\quad \eta_v^{(n+1)} = \eta_v^{(n)} - (v^{(n+1)} - \mathbf{M}x^{(n+1)})$.
9: $\quad \eta_z^{(n+1)} = \eta_z^{(n)} - (z^{(n+1)} - \mathbf{C}_s v^{(n+1)})$.
10: $\quad j = j + 1$
11: **end for**

3.2.3 Alternating Minimization

Rather than jointly minimizing x, u, v, and z, as required in line 5 of the AL framework, we choose instead to minimize each variable in turn within a particular iteration. To derive the x-update equation for the $(j+1)$th iteration, we need only minimize over the two quadratic terms of Equation (3.4) containing x. Recalling that $\mathbf{M}'\mathbf{M} = \mathbf{I}_{N_{\text{mask}}}$, this yields the following update rule:

$$x^{(n+1)} = \left(\mathbf{M}'\mathbf{S}'\mathbf{SM} + \frac{\mu_v}{\mu_u}\mathbf{I}\right)^{-1}\left[\mathbf{M}'\mathbf{S}'\left(u^{(n)} - \eta_u^{(n)}\right) + \frac{\mu_v}{\mu_u}\mathbf{M}'\left(v^{(n)} - \eta_v^{(n)}\right)\right]. \quad (3.5)$$

Just as $\mathbf{S}'\mathbf{S}$ is diagonal, so too is $\mathbf{M}'\mathbf{S}'\mathbf{SM}$, albeit with only N_{mask} elements. The diagonal structure allows for easy inversion of the Hessian matrix $\mathbf{H}_{\text{x}} = \mathbf{M}'\mathbf{S}'\mathbf{SM} + \frac{\mu_v}{\mu_u}\mathbf{I}$ and a direct solution for (3.5).

We derive the update for auxiliary variable u in a similar way. If each of the DFT blocks of $\mathbf{F}$ is denoted $\mathbf{F}_i$, for $i = 1, \ldots, N_c$, then each $\mathbf{F}_i'\mathbf{F}_i$ is block circulant. Thus, the Hessian is circulant and can be diagonalized as follows: $\mathbf{H}_{\mathrm{u}} = \mathbf{F}'\mathbf{F} + \mu_u\mathbf{I} = \mathbf{Q}N_c'(\mathbf{\Lambda}_{FF} + \mu_u\mathbf{I})\mathbf{Q}_{N_c}$. Note that since u is the concatenation of the estimated image under each sensitivity coil, $\mathbf{Q}_{N_c}$ denotes a block DFT matrix with N_c blocks:

$$\mathbf{Q}_{N_c} = \begin{bmatrix} \mathbf{Q}_{(1)} & & \\ & \ddots & \\ & & \mathbf{Q}_{(N_c)} \end{bmatrix}; \quad \mathbf{Q}_{(i)} = \text{2-D } N_x \times N_y \text{ DFT} \tag{3.6}$$

$$u^{(n+1)} = (\mathbf{F}'\mathbf{F} + \mu_u\mathbf{I})^{-1}(\mathbf{F}'y + \mu_u(\mathbf{SM}x^{(n+1)} + \eta_u^{(n)})) \tag{3.7}$$

$$u^{(n+1)} = \mathbf{Q}'(\mathbf{\Lambda}_{FF} + \mu_u\mathbf{I})^{-1}\mathbf{Q}(\mathbf{F}'y + \mu_u(\mathbf{SM}x^{(n+1)} + \eta_u^{(n)})). \tag{3.8}$$

By using the precomputed eigenvalues of $\mathbf{H}_{\mathrm{u}}$, we update u efficiently and exactly. The choice of $u = \mathbf{SM}x$ in the variable splitting scheme was motivated by the separation of the circulant behavior of $\mathbf{F}'\mathbf{F}$ from the diagonal structure of $\mathbf{M}'\mathbf{S}'\mathbf{SM}$. Though minimization of the datafit term in Equation (3.2) now requires updating two variables instead of one, performing both of these updates is simpler and faster than inverting $\mathbf{M}'\mathbf{S}'\mathbf{F}'\mathbf{FSM}$.

The resulting Hessian for the minimization of v is $\mathbf{H}_{\mathrm{v}} = \mathbf{C}_s'\mathbf{C}_s + \frac{\mu_v}{\mu_z}\mathbf{I}$. Since we have chosen $\mathbf{C}_s$ such that $\mathbf{C}_s'\mathbf{C}_s$ is circulant, $\mathbf{H}_{\mathrm{v}}$ is also circulant and can be diagonalized as $\mathbf{H}_{\mathrm{v}} = \mathbf{Q}\left(\mathbf{\Lambda}_{CC} + \frac{\mu_v}{\mu_z}\mathbf{I}\right)\mathbf{Q}'$, where $\mathbf{Q}$ is the 2-D $N_x \times N_y$ DFT matrix. The choice

of $v = \mathbf{M}x$ and $z = \mathbf{C}_s v$ in the variable splitting was motivated by isolation of $\mathbf{C}_s'\mathbf{C}_s$ from the influence of the masking matrix $\mathbf{M}$. If this nested auxiliary variable split were omitted, then the Hessian would have the form $\mathbf{M}'\mathbf{C}_s'\mathbf{C}_s\mathbf{M} + c\mathbf{I}$. This Hessian is neither diagonal nor circulant, and this update would have required an iterative method to solve. The update for v is:

$$v^{(n+1)} = (\mathbf{C}_s'\mathbf{C}_s + \frac{\mu_v}{\mu_z}\mathbf{I})^{-1}\left((\mathbf{C}_s'z^{(n)} - \mathbf{C}_s'\eta_z) + \frac{\mu_v}{\mu_z}(\mathbf{M}x^{(n)} + \eta_v)\right) \tag{3.9}$$

$$v^{(n+1)} = \mathbf{Q}'\left(\mathbf{\Lambda}_{CC} + \frac{\mu_v}{\mu_z}\mathbf{I}\right)^{-1}\mathbf{Q}\left((\mathbf{C}_s'z^{(n)} - \mathbf{C}_s'\eta_z) + \frac{\mu_v}{\mu_z}(\mathbf{M}x^{(n)} + \eta_v)\right) \tag{3.10}$$

The auxiliary variable z was designed to encompass the entire ℓ_1 term in Equation (3.2) so that we are able to employ soft-thresholding in the following manner to exactly minimize z:

$$z^{(n+1)} = \text{shrink}\left(\mathbf{C}_s v + \eta_z, \frac{\lambda}{\mu_z}\right), \tag{3.11}$$

where the shrinkage operator is defined as:

$$\text{shrink}(x, \gamma) = \text{sign}(x)\max\{|x| - \gamma, 0\} \tag{3.12}$$

This yields the following algorithm:

Algorithm 1 Masked AL Static SENSE algorithm

Input $y, \mathbf{F}, \mathbf{S}, \mathbf{M}, \mathbf{C}_s$
Select μ_u, μ_v, μ_z.
Initialize x
Initialize $u = \mathbf{SM}x$, $v = \mathbf{M}x$, $z = \mathbf{C}_s v$, and η_u, η_v, $\eta_z = 0$.
for n iterations **do**
    $x^{(n+1)} = \left(\mathbf{M}'\mathbf{S}'\mathbf{SM} + \frac{\mu_v}{\mu_u}\mathbf{I}\right)^{-1}\left[\mathbf{M}'\mathbf{S}'\left(u^{(n)} - \eta_u^{(n)}\right) + \frac{\mu_v}{\mu_u}\mathbf{M}'\left(v^{(n)} - \eta_v^{(n)}\right)\right]$.
    $u^{(n+1)} = \mathbf{Q}'_{N_c}(\mathbf{\Lambda}_{FF} + \mu_u\mathbf{I})^{-1}\mathbf{Q}_{N_c}(\mathbf{F}'y + \mu_u(\mathbf{SM}x^{(n+1)} + \eta_u^{(n)}))$.
    $v^{(n+1)} = \mathbf{Q}'\left(\mathbf{\Lambda}_{\mathbf{C}_s\mathbf{C}_s} + \frac{\mu_v}{\mu_z}\mathbf{I}\right)^{-1}\mathbf{Q}\left(\mathbf{C}'_s(z^{(n)} - \eta_z^{(n)}) + \frac{\mu_v}{\mu_z}(\mathbf{M}x^{(n+1)} + \eta_v^{(n)})\right)$.
    $z^{(n+1)} = \text{shrink}\left(\mathbf{C}_s v^{(n+1)} + \eta_z^{(n+1)}, \frac{\lambda}{\mu_z}\right)$.
    $\eta_u^{(n+1)} = \eta_u^{(n)} - (u^{(n+1)} - \mathbf{SM}x^{(n+1)})$.
    $\eta_v^{(n+1)} = \eta_v^{(n)} - (v^{(n+1)} - \mathbf{M}x^{(n+1)})$.
    $\eta_z^{(n+1)} = \eta_z^{(n)} - (z^{(n+1)} - \mathbf{C}_s v^{(n+1)})$.
    $j = j + 1$
end for

3.3 Comparison with Other Variable Splitting Algorithms

3.3.1 Comparison with AL-P2

A recently proposed variable splitting algorithm was proposed for SENSE reconstruction by Ramani et al. in [67]. This algorithm, AL-P2, was not designed for the model in Equation (3.1), but its variable splitting scheme can be extrapolated to match the model, resulting in the following algorithm.

$$x^{(*)} = \underset{x,u,v,z}{\text{argmin}} \, ||y - \mathbf{F}u||_2^2 + ||z||_1 \quad s.t.\ u = \mathbf{SM}x,\ v = x,\ z = \mathbf{C}_s\mathbf{M}v \tag{3.13}$$

1: Input $y, \mathbf{S}, \mathbf{M}$
2: Initialize x
3: **for** n iterations **do**
4: $\quad x^{(n+1)} = \left(\mathbf{M}'\mathbf{S}'\mathbf{S}\mathbf{M} + \frac{\mu_v}{\mu_u}\mathbf{I}\right)^{-1}\left[\mathbf{M}'\mathbf{S}'\left(u^{(n)} - \eta_u^{(n)}\right) + \frac{\mu_v}{\mu_u}\left(v^{(n)} - \eta_v^{(n)}\right)\right]$.
5: $\quad u^{(n+1)} = (\mathbf{F}'\mathbf{F} + \mu_u\mathbf{I})^{-1}(\mathbf{F}'y + \mu_u(\mathbf{S}\mathbf{M}x^{(n)} + \eta_u^{(n)}))$.
6: $\quad v^{(n+1)} = (\mathbf{M}'\mathbf{C}_s'\mathbf{C}_s\mathbf{M} + \frac{\mu_v}{\mu_z}\mathbf{I})^{-1}\left((\mathbf{M}'\mathbf{C}_s'z^{(n)} - \mathbf{M}'\mathbf{C}_s'\eta_z^{(n)}) + \frac{\mu_v}{\mu_z}(x^{(n)} + \eta_v^{(n)})\right)$.
7: $\quad z^{(n+1)} = \text{shrink}\left(\mathbf{C}_s\mathbf{M}v + \eta_z^{(n)}, \frac{\lambda}{\mu_z}\right)$.
8: $\quad \eta_u^{(n+1)} = \eta_u^{(n)} - (u^{(n+1)} - \mathbf{S}\mathbf{M}x^{(n+1)})$.
9: $\quad \eta_v^{(n+1)} = \eta_v^{(n)} - (v^{(n+1)} - x^{(n+1)})$.
10: $\quad \eta_z^{(n+1)} = \eta_z^{(n)} - (z^{(n+1)} - \mathbf{C}_s\mathbf{M}v^{(n+1)})$.
11: $\quad j = j + 1$
12: **end for**

The most significant difference between the proposed method and AL-P2 is the v update, line 6. For AL-P2, the Hessian is $\mathbf{H}_{\text{v}} = \mathbf{M}'\mathbf{C}_s'\mathbf{C}_s\mathbf{M} + \frac{\mu_v}{\mu_z}\mathbf{I}$, which is neither circulant or diagonalizable. Thus, the v update itself would require an iterative method to solve, such as Conjugate Gradient (CG).

3.3.2 Comparison with Chambolle-Pock's Primal Dual Algorithm

Chambolle-Pock's Primal Dual Algorithm solves a saddlepoint problem that is the primal-dual problem of the original minimization problem [68] . As such, it does not rely on variable splitting or the Augmented Lagrangian formulation, though Chambolle et al. demonstrate that it is equivalent to the preconditioned version of the Alternating Direction Method of Multipliers (ADMM). ADMM is also known as the "decomposable method of multipliers", because it alternatingly minimizes over sets of variables in a separable objective function [37]. Though the proposed variable

splitting method does not satisfy the conditions for ADMM convergence guarantees, it too is based on alternating minimization followed by a Augmented-Langrangian-like vector update.

3.3.3 Comparison with Split Bregman

The Split Bregman method was developed by Goldstein et al. for ℓ_1 regularized problems in [35]. We discuss the Split Bregman method in more detail in Section 2.3.1. Like the proposed method, the Split Bregman algorithm decouples the ℓ_1 and ℓ_2 terms in the objective function by introducing a single auxiliary variable. The method then converts a constrained optimization problem into an unconstrained optimization problem by introduction of a quadratic penalty term followed by alternating minimization over the original and auxiliary variable within each iteration. The Split Bregman method departs from the proposed method in its single variable splitting, which yields more complicated inner optimization problems. In particular, the Hessian for the variable update of x does not demonstrate any circulant or diagonal structure to exploit during inversion, so this step must be performed with an iterative method. For the purpose of comparison, we chose to perform this step with the Conjugate Gradient (CG) method.

3.4 Results

3.4.1 Simulation

We used a T_1-weighted 258×258 image from BrainWeb as ground truth for simulation. We also simulated sensitivity maps for an 8-channel coil based on [69]. The mask was an ellipse hand-placed based on the fast sum-of-squares (SoS) estimate, as shown in Figure 3.1. Figure 3.2 shows the sampling scheme used— a Poisson disk sampling scheme with the central 9×9 central phase encodes added, as motivated by Section 2.2.5. This sampling scheme used code available from [70]. We chose R to be anisotropic total variation (TV), measuring first-order differences with all eight cardinal and diagonal neighbors. We added complex i.i.d. white Gaussian noise to the simulated sensitivity encoded k-space for an SNR of 40 dB. We implemented all algorithms in Matlab, utilizing the Fessler Image Reconstruction Toolbox [71] and executed on a 3.30 GHz CPU. Figure 3.3 shows the objective function value from Equation (3.2) as a function of iteration.

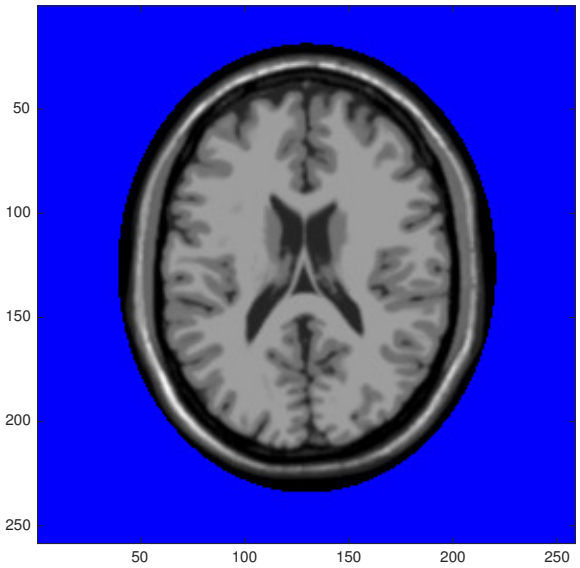


Figure 3.1: User selected elliptical mask overlaid on top of ground truth image.

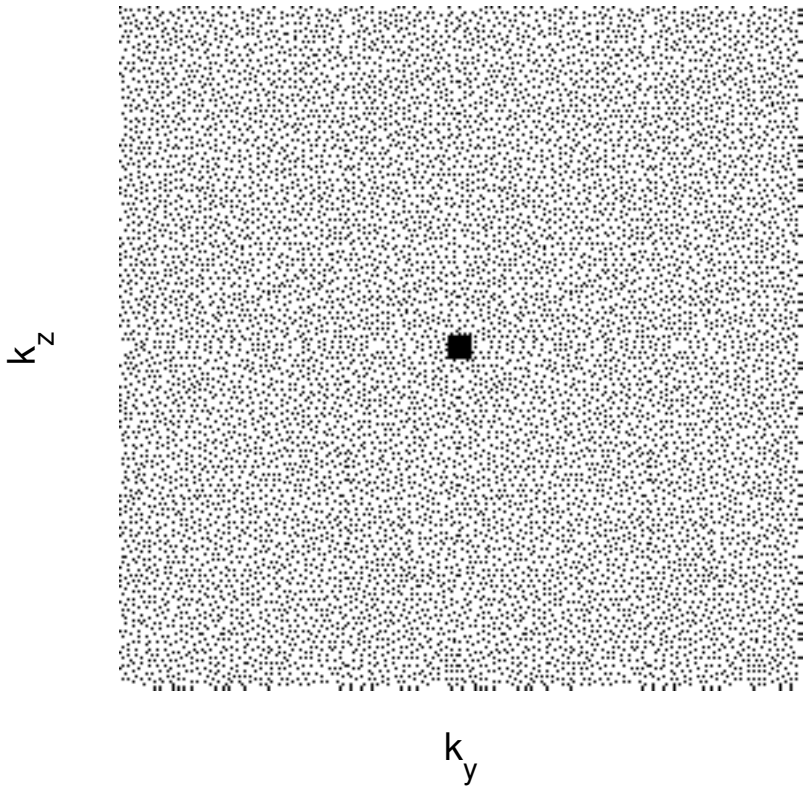


Figure 3.2: Sampling scheme used for simulation— union of a Poisson disk sampling pattern and the central 9×9 phase encodes, total reduction factor of ≈ 5.6

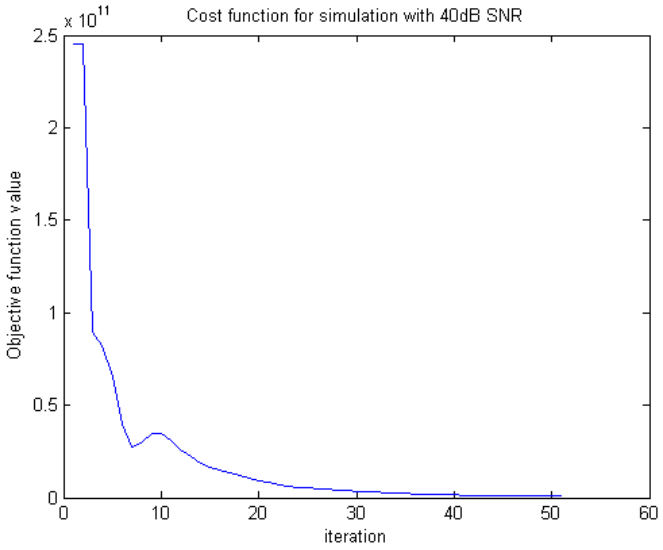


Figure 3.3: Original cost function as a function of iteration for a simulation with 40 dB SNR and undersampling of 5.6.

To further justify the use of a support constraint in the model proposed in Equation (3.1), we performed experiments varying reduction factors and sizes of the support constraint. Figure 3.5 shows the Normalized Root Mean Squared Error (NRMSE) of the reconstructed image after 800 iterations as a function of acceleration factor. Using the masked model and the mask depicted in Figure 3.1, the proposed method achieved lower NRMSE using the same model without a support constraint. The difference in quality becomes more dramatic at higher acceleration factors.

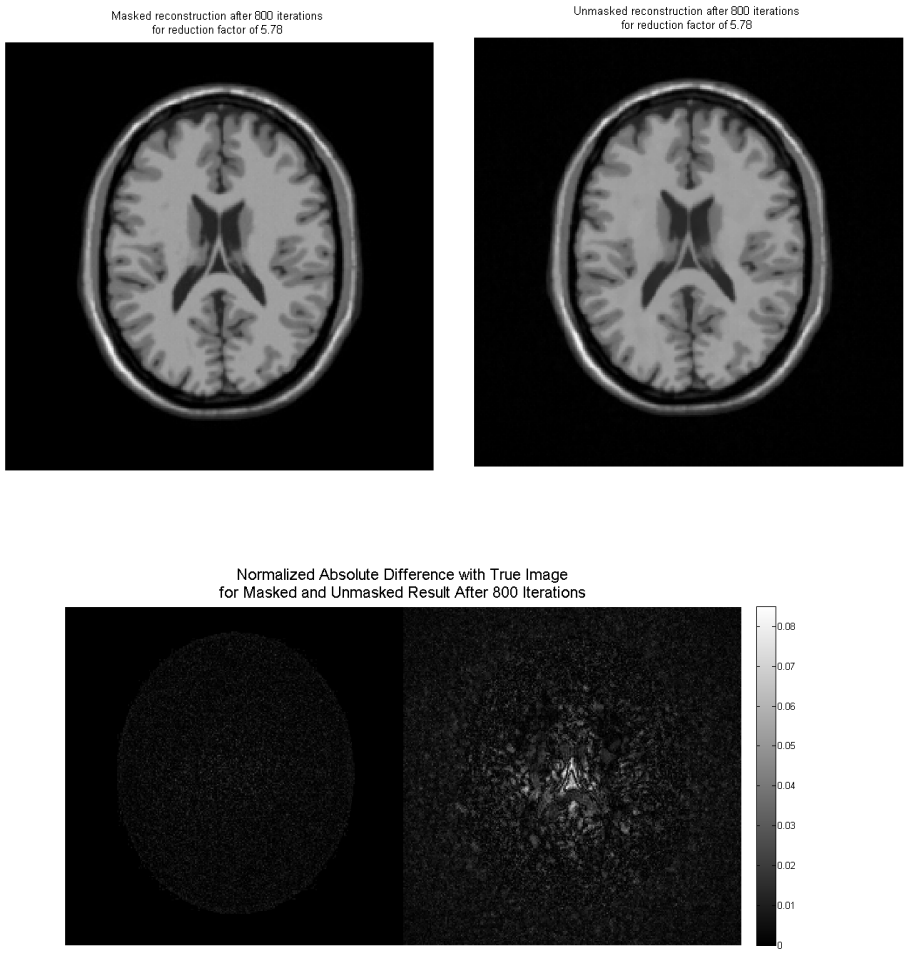


Figure 3.4: (a) Reconstructed image using tight mask after 800 iterations; (b) Reconstructed image using no mask after 800 iterations; (c) Normalized absolute difference with true image for masked model (left) and unmasked model (right).

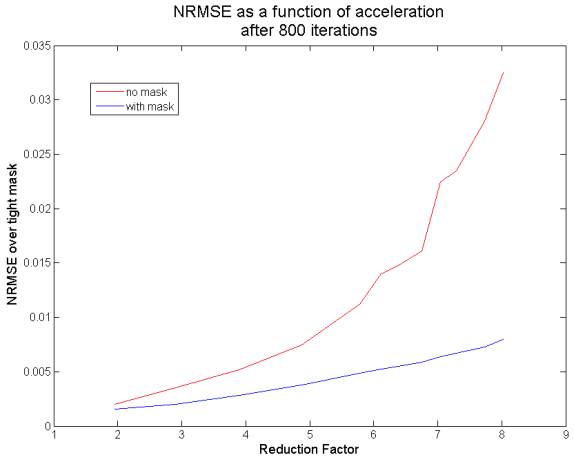


Figure 3.5: Normalized root mean squared error (NRMSE) after 800 iterations as a function of reduction factor and masking. NRMSE was calculated only over the tightly masked region for a fair comparison.

Figure 3.6 shows how NRMSE decreases as the mask fits more tightly around the object. The NRMSE for both Figures 3.5 and 3.6 were computed after 800 iterations and only over the tightest defined mask.

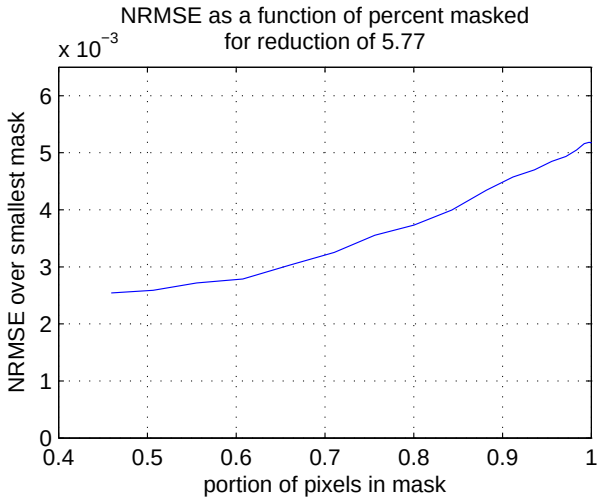


Figure 3.6: Normalized root mean squared error after 800 iterations as a function of the size of the mask for a reduction factor of 5.77. NRMSE was calculated only over the tightly masked region for a fair comparison.

3.4.2 In-vivo experiment

We used a 3D in-vivo human data set acquired with a GE 3T scanner with $T_R = 25$ ms, $T_E = 5.172$ ms, an 8-channel coil, and a voxel size of $1 \times 1.35 \times 1\ mm^3$. The slices were 256×144 in size with 128 samples in the readout direction and were acquired with a 3D spoiled gradient echo routine.

For the sampling pattern, we used a Poisson-disk-based undersampling pattern with its central 32×18 phase-encodes fully sampled. We retrospectively undersampled the 3D dataset using the pattern pictured in Figure 3.7 in the k_y-k_z plane, while fully sampling along k_x. Thus, the black pixels indicate the location of the quasi-randomly located readout lines along k_x.

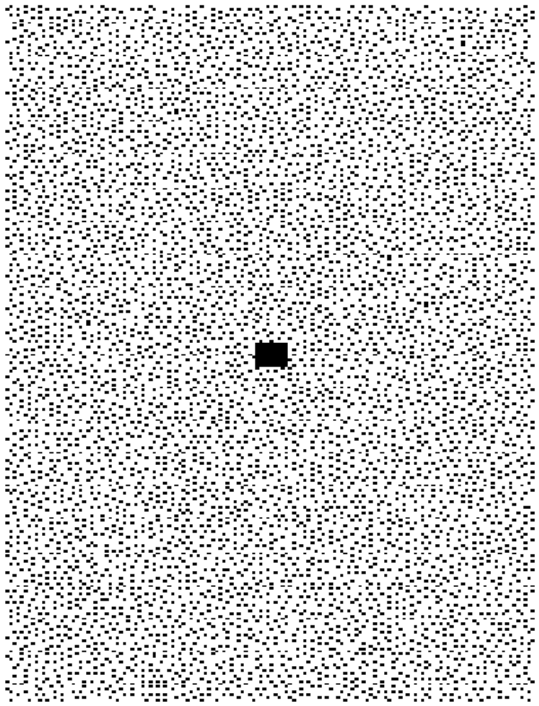


Figure 3.7: Poisson-disk-based sampling including fully sampled central 32×18 phase-encodes, with overall reduction factor of ≈ 5.65

We used the fully sampled center of k-space to generate a low-resolution image that we then used for estimation of the sensitivity maps. We estimated the sensitivity maps using [1], and they are shown in Figure 3.8. We used the sum of squares (SoS) of the zero-filled iFFT-reconstructed coil images, depicted in Figure 3.9 as a guide for the user selection of the spatial mask and as the initial estimate in each of the algorithms below.

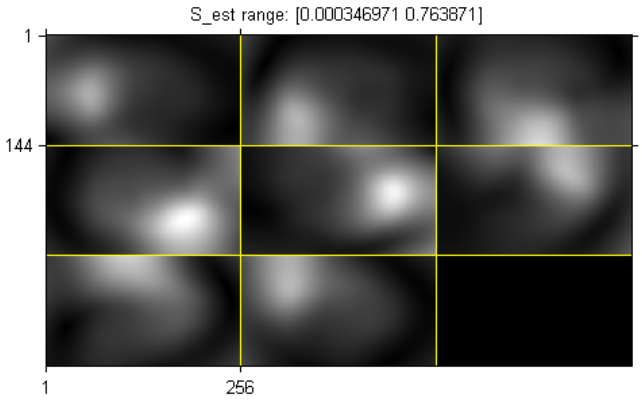


Figure 3.8: Sensitivity maps for 8-channel coil as estimated by [1] based on low-resolution images.

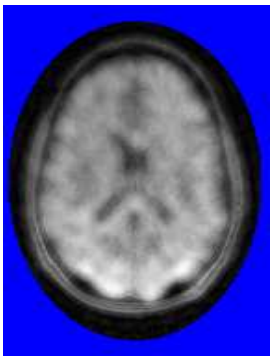


Figure 3.9: User selected elliptical mask overlaid on sum of squares (SoS) of iFFT zero-filled central k-space data.

Table 3.1 lists the three algorithms used in the convergence speed comparison in Figure 3.10 and describes the method used in the inner iteration step (if applica-

ble).

Table 3.1: Algorithms used in speed comparison

Algorithm Name	Abbreviation	Number of inner iterations	Method for inner iterations
Split Bregman [35]	SB	2	CG
Ramani AL Method [67]	AL-P2	2	PCG
Chambolle-Pock Primal Dual Algorithm [68]	CP-PDA	N/A	N/A
Proposed Method	AL-Mask	N/A	N/A

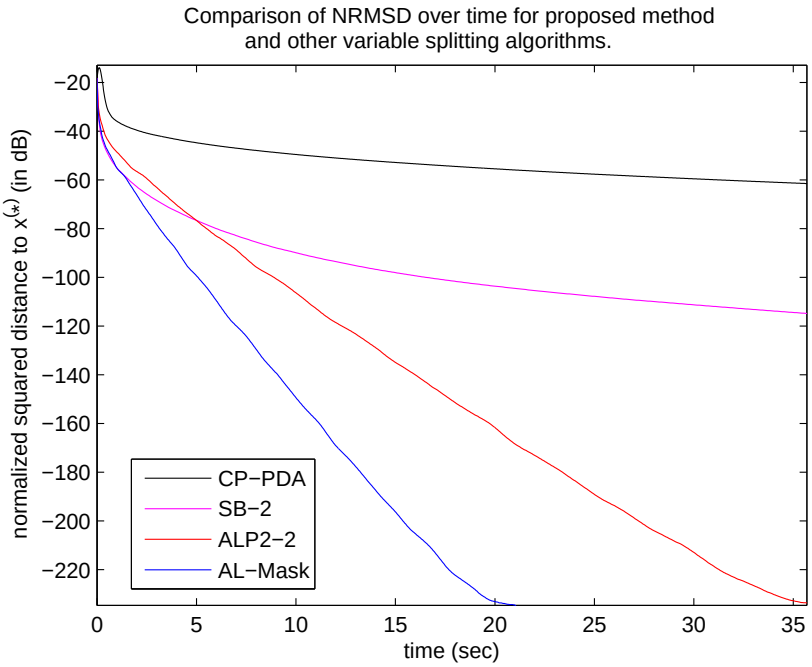
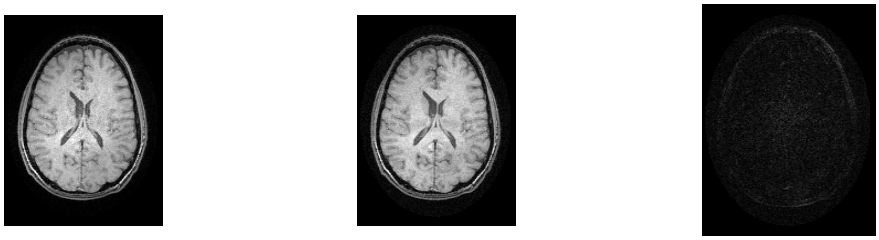


Figure 3.10: Comparison of convergence rates over computation time for proposed method, AL-Mask, with the Chambolle-Pock Primal Dual Algorithm, the Split Bregman method, and AL-P2. Convergence was measured as normalized root mean square distance (NRMSD) to the true solution, $x^{(*)}$.

To compare the convergence speed of the algorithms, we compared the normalized root mean squared distance (NRMSD) from the current iterate to the optimum $x^{(*)}$.

We approximated $x^{(*)}$ by running SB with 10 inner CG iterations for 2000 outer iterations. Since the μ_u, μ_v, μ_z parameters affect the convergence rate of AL-Mask (as corresponding convergence parameters affect the other algorithms), we chose the parameters of each algorithm based on the thresholding step that results from the ℓ_1 norm of the regularization term. After setting the thresholding value to be the same for all the algorithms, the remaining parameter values were chosen to yield Hessian matrices with condition numbers in the range of $[10, 36]$ as described in [67]. The regularization parameter λ was chosen manually to yield an image of acceptable quality.

To assess image quality, we also compared the result of the proposed reconstruction algorithm at 1200 iterations (Figure 3.11a) with the body coil image (Figure 3.11b), reconstructed from data received by the full body coil, as opposed to the 8-channel receiving sensitivity coils. The absolute difference image is shown in Figure 3.11c on the same colorscale as the reconstructed and bodycoil image.



(a) (b) (c)

Figure 3.11: (a) The reconstructed image after 1200 iterations of AL-Mask; (b) A body coil image for comparison; (c) Absolute differences image between (a) and (b)

3.5 Conclusions

Though AL-mask demonstrated modest image quality and speed improvements over unmasked competitors, the overall potential gain in implementing the mask was not dramatic. The opportunity to crop out large air regions is also reduced in many applications, such as abdominal imaging. Furthermore, a rectangular choice of mask, which may be preferable for certain slice orientations and MRI applications, erases the benefits of the AL-Mask method compared to AL-P2.

The principles of AL-mask will manifest in some of the following proposed methods as an enforcement of a logical mask over the operators that act directly on dynamic images, such as $\mathbf{S}$ and $\mathbf{C}_s$, and by estimating only the unknown N_{mask} pixels.

Chapter 4

Variable Splitting with an Efficient Tridiagonal Solver

4.1 Introduction

[1] Accelerated scan time benefits many applications of Magnetic Resonance Imaging (MRI), reducing cost and motion blurring. A popular strategy for reducing scan time is to acquire fewer k-space samples. To compensate for the reduced sampling, parallel MRI and Compressed Sensing (CS) are often used. SENSitivity Encoded (SENSE) MRI is a popular framework for parallel MRI [21]. By simultaneously acquiring data from multiple receive coils with spatially varying sensitivities, more data can be collected without additional scan time. SENSE can be combined with CS-inspired techniques to further reduce acquisition time [26]. Irregular undersampling patterns enable higher acceleration but require iterative model-based image

[1] This chapter is based on [9].

reconstruction.

CS MRI reconstructs images assuming image sparsity in some transform domain. To balance adherence of the estimated image to the noisy, undersampled data with the prior assumption of sparsity, we seek to minimize a cost function that describes both. These methods employ regularization with ℓ_1 norms that promote sparsity but also present computational challenges.

Variable splitting (VS) [72] [67] [73] is a versatile optimization approach for these cost functions. VS decouples a costly nonlinear optimization into simpler problems via the augmented Lagrangian (AL) framework. VS converts the original cost function into a constrained cost function involving additional auxiliary variables updated with alternating minimization. The AL-P2 algorithm proposed in [67] demonstrated that VS combined with AL can yield significant speed gains over conjugate gradients (CG) and monotone fast iterative shrinkage-thresholding algorithm (MFISTA) [74] for CS-SENSE-MRI. However, AL-P2 lacks a convergence guarantee and uses periodic boundary conditions when applying sparsifying transforms. This assumption leads to a circulant Hessian and an $\mathcal{O}(n \log n)$ FFT-based solution for one of the inner problems, where n is the number of pixels.

The use of periodic boundary conditions is not reasonable for reconstructing 2D axial slices of the brain, when it is surrounded by air at all boundaries. However, when reconstructing an entire 3D volume, it could be undesirable to impose periodic boundary conditions across the top and bottom slices. Non-periodic boundary conditions are also useful in dynamic imaging with acyclic temporal behavior. For

example, Dynamic Contrast Enhanced (DCE) images differ significantly in the final frame compared to the initial frame.

Here we present two related variable splitting methods for CS-SENSE-MRI: ADMM-tridiag and AL-tridiag. ADMM-tridiag leverages [75] to ensure convergence. Both proposed algorithms use a regularizer with non-periodic boundary conditions to more accurately reflect the reality of the unknown image. Both algorithms incorporate parallelizable tridiagonal solvers that efficiently handle the non-periodic boundary conditions with $\mathcal{O}(n)$ operations. We present numerical results with real *in vivo* data to demonstrate the efficacy of the proposed methods.

We also apply the principles to describe novel VS algorithms for the simpler "special case" of image inpainting using regularization based on a combination of wavelets and anisotropic total variation (TV) with non-periodic boundary conditions.

4.2 Problem Formulation

The analysis formulation often used for SENSE MRI reconstruction estimates the unknown image, $\hat{x}$, by seeking the minimizer of a cost function consisting of a datafit term plus a regularizer. Regularization is particularly important for undersampled problems.

Many regularizers have been used for SENSE MRI reconstruction. For CS-SENSE-MRI, sparsifying transforms such as wavelets and first-order finite differences are often used with an ℓ_1 norm to promote sparsity [26]. This chapter focuses on Carte-

sian SENSE reconstruction with regularizers having non-periodic boundary conditions.

Let N_c denote the number of sensitivity coils, N_s the number of samples received from each coil, and $N_r = N_x N_y$ the number of pixels in the latent image x. We formulate regularized SENSE reconstruction as the following optimization problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \|y - \mathbf{F}\mathbf{S}x\|_2^2 + \lambda \|\mathbf{C}_{\mathrm{H}}x\|_1 + \lambda \|\mathbf{C}_{\mathrm{V}}x\|_1 \tag{4.1}$$

where $y \in \mathbb{C}^{N_c N_s}$ is the undersampled k-space data from all coils, $\mathbf{F} \in \mathbb{C}^{N_c N_s \times N_c N_r}$ is a block diagonal matrix consisting of undersampled DFT matrices, $\mathbf{S} \in \mathbb{C}^{N_c N_r \times N_r}$ is a stack of diagonal matrices containing the sensitivity maps, $\mathbf{C}_{\mathrm{H}} \in \mathbb{R}^{N_r \times N_r}$ and $\mathbf{C}_{\mathrm{V}} \in \mathbb{R}^{N_r \times N_r}$ denote finite differences in the horizontal and vertical directions (equivalent to anisotropic TV), and $\hat{x}$ is the reconstructed image. The regularization parameter $\lambda > 0$ balances adherence to noisy data and prior assumptions that inform choice of regularizers. Section 4.3.5 extends (4.1) to include discrete wavelet transforms. Although we focus on 2D imaging for notational simplicity, the methods generalize readily to 3D problems by adding another term to (4.1).

We choose $\mathbf{C}_{\mathrm{H}}$ and $\mathbf{C}_{\mathrm{V}}$ to have non-periodic boundary conditions. This is in contrast to many proposed algorithms that use finite differences with periodic boundary conditions, such as AL-P2 [67], RecPF [76], and recMRI [77]. Periodic boundary conditions have been used for computational convenience, despite being physically unnatural. Non-periodic boundary conditions are preferable in most applications [78, 79, 18]. Differences across the boundaries of medical images do not provide use-

ful information for reconstruction. Furthermore, penalizing differences across spatial boundaries may degrade image quality when the region of interest extends to the boundary. For example, a coronal abdominal image depicts different anatomy at the top and bottom boundaries. Thus this chapter focuses on developing methods that are fast yet suitable for non-periodic boundary conditions.

4.3 Variable Splitting Methods

The non-differentiable ℓ_1 norms make (4.1) a challenging optimization problem. Variable splitting methods like [80, 81, 67] are useful for such problems. Here, we reformulate (4.1) in an equivalent constrained form using the following novel variable splitting scheme:

$$\begin{aligned} \underline{\hat{u}} &= \underset{\underline{u}}{\operatorname{argmin}}\, f\left(\underline{u}\right) \\ f\left(\underline{u}\right) &= \frac{1}{2}\left\|y - \mathbf{F}u_2\right\|^2 + \lambda\left\|u_0\right\|_1 + \lambda\left\|u_1\right\|_1 \end{aligned} \tag{4.2}$$

$$\begin{aligned} \text{s.t. } u_0 &= \mathbf{C}_{\mathrm{H}}x, & u_1 &= \mathbf{C}_{\mathrm{V}}u_3, \\ u_2 &= \frac{1}{2}\mathbf{S}u_3 + \frac{1}{2}\mathbf{S}x, & u_3 &= x. \end{aligned} \tag{4.3}$$

For convenience, we group x and the auxiliary variables $u_0, \ldots, u_3$ into one column vector: $\underline{u} \triangleq (u_0, u_1, u_2, u_3, x)$.

This variable splitting scheme intentionally separates the horizontal and vertical finite

differences operators, applying them to different auxiliary variables. This separates the tridiagonal structures in the Hessians resulting from AL, permitting decoupled, computationally efficient variable updates further detailed in Section 4.3.4. If the finite difference matrices were combined in the same auxiliary variable as in AL-P2 [67], the resulting Hessians would have a block-tridiagonal with tridiagonal blocks (BTTB) structure that cannot take advantage of an $\mathcal{O}(N_r)$ tridiagonal solver.

4.3.1 Direct AL Approach: AL-tridiag

Here we detail the algorithm resulting from directly applying AL with alternating minimization to (4.2). The resulting algorithm, AL-tridiag, does not satisfy the sufficient conditions for convergence in [75], so currently it lacks convergence guarantees. However, it has worked well in all of our experiments, so it is possible that future generalized convergence proofs could be applicable.

We rewrite constrained cost function (4.2) with a matrix constraint as follows:

$$\min_{\underline{u}} f(\underline{u}) \text{ s.t. } \mathbf{P}\underline{u} = 0 \tag{4.4}$$

$$\mathbf{P} \triangleq \begin{bmatrix} -\mathbf{I}_{N_r} & 0 & 0 & 0 & \mathbf{C}_{\mathrm{H}} \\ 0 & -\mathbf{I}_{N_r} & 0 & \mathbf{C}_{\mathrm{V}} & 0 \\ 0 & 0 & -\mathbf{I}_{N_r N_c} & \frac{1}{2}\mathbf{S} & \frac{1}{2}\mathbf{S} \\ 0 & 0 & 0 & -\mathbf{I}_{N_r} & \mathbf{I}_{N_r} \end{bmatrix}.$$

The constraint $\mathbf{P}\underline{u} = 0$ enforces (4.3). The augmented Lagrangian corresponding to (4.4) is:

$$\mathcal{L}\left(\underline{u}, \underline{\eta}; \mathbf{M}\right) = f\left(\underline{u}\right) + \frac{1}{2}\left\|\mathbf{P}\underline{u} - \underline{\eta}\right\|_{\mathbf{M}}^{2}. \tag{4.5}$$

This formulation introduces dual variables, stacked in a vector: $\underline{\eta} \triangleq (\eta_0, \ldots, \eta_3) \in \mathbb{C}^{(3+N_c)N_r}$. Matrix $\mathbf{M}$ is a positive definite diagonal matrix, consisting of user-selected AL penalty parameters. The diagonal block corresponding to the ith segment of $\underline{u}$ is denoted $\mathbf{M}_i$. The choice of $\mathbf{M}$ does not affect the final solution of (4.5), but it can affect the convergence rate of the resulting algorithm. For many multiplier methods, using positive penalty parameters guarantees convergence to the solution of the original problem that does not involve the penalty parameters [65]. Section 4.4.4 discusses heuristics for selecting $\mathbf{M}$.

Ideally, an AL method would update block variables $\underline{u}$ and $\underline{\eta}$ at iteration $n+1$ as follows:

$$\underline{u}^{(n+1)} = \underset{\underline{u}}{\operatorname{argmin}}\, \mathcal{L}\left(\underline{u}, \underline{\eta}^{(n)}\right) \tag{4.6}$$

$$\underline{\eta}^{(n+1)} = \underline{\eta}^{(n)} - \mathbf{P}\underline{u}^{(n+1)}. \tag{4.7}$$

Our proposed AL-tridiag algorithm uses alternating minimization across $u_0, \ldots, u_3, x$ to descend the AL term in (4.6). Section 4.3.4 and the supplement describe the $\underline{u}$ variable updates in more detail. Due to the variable splitting design of (4.3), each variable update has a direct, closed-form solution with an efficient implementation,

e.g., by FFTs or a parallelizable tridiagonal solver.

4.3.2 Variable updates for AL-tridiag

This section describes the block-variable updates for alternating minimization of (4.6). These variable updates are very similar for both proposed methods, AL-tridiag and ADMM-tridiag, differing only in tuning parameter indices and use of $\underline{v}$ and $\underline{\eta}$ on the right-hand side of these updates.

$$u_0^{(n+1)} = \text{soft}\left(\mathbf{C}_{\text{H}} x^{(n)} - \eta_0^{(n)}, \frac{\lambda}{\mu_0}\right) \tag{4.8}$$

$$u_1^{(n+1)} = \text{soft}\left(\mathbf{C}_{\text{V}} u_3^{(n)} - \eta_1^{(n)}, \frac{\lambda}{\mu_1}\right) \tag{4.9}$$

$$u_2^{(n+1)} = \mathbf{H}_2^{-1}\left(\mathbf{F}' y + \mu_2\left(\frac{1}{2}\mathbf{S} u_3^{(n)} + \frac{1}{2}\mathbf{S} x^{(n)} - \eta_2^{(n)}\right)\right) \tag{4.10}$$

$$u_3^{(n+1)} = \mathbf{H}_3^{-1}\left(\mu_1 \mathbf{C}_{\text{V}}'\left(u_1^{(n+1)} + \eta_1^{(n)}\right) + \frac{\mu_2}{2}\mathbf{S}'\left(u_2^{(n+1)} - \frac{1}{2}\mathbf{S} x^{(n)} + \eta_2^{(n)}\right) + \mathbf{M}_3\left(x^{(n)} - \eta_3^{(n)}\right)\right) \tag{4.11}$$

$$x^{(n+1)} = \mathbf{H}_{\text{x}}^{-1}\left(\mu_0 \mathbf{C}_{\text{H}}'\left(u_0^{(n)} + \eta_0^{(n)}\right) + \frac{\mu_2}{2}\mathbf{S}'\left(u_2^{(n+1)} - \frac{1}{2}\mathbf{S} u_3^{(n+1)}) + \eta_2^{(n)}\right) + \mathbf{M}_3\left(u_3^{(n)} + \eta_3^{(n)}\right)\right), \tag{4.12}$$

where the Hessians $\mathbf{H}_2$ and $\mathbf{H}_3$ are the same as for ADMM-tridiag, defined in Equations (4.29)-(4.30). In AL-tridiag, $\mathbf{H}_{\text{x}} \triangleq \mu_0 \mathbf{C}_{\text{H}}'\mathbf{C}_{\text{H}} + \frac{\mu_2}{4}\mathbf{S}'\mathbf{S} + \mathbf{M}_3$. The updates for u_0, u_1, and u_2, (4.8)-(4.10) are identical to those in ADMM-tridiag. The updates for u_3 and x differ only in the rightmost term containing η_3 and in the Hessian $\mathbf{H}_{\text{x}}$.

In practice, the implementation differences between AL-tridiag and ADMM-tridiag are further diminished by choosing the same $\mathbf{H}_{\mathrm{x}}$ for both algorithms. Due to the parameter selection technique for ADMM-tridiag described in (4.4.4), both algorithms use the same Hessians.

4.3.3 ADMM Equivalence for ADMM-tridiag

To design a minimization algorithm with convergence guarantees, we reformulate (4.2) as an instance of the generalized Alternating Direction Method of Multipliers (ADMM) [75] [38]. This formulation allows us to invoke the convergence proof in [75] for our second proposed algorithm, ADMM-tridiag.

We first express the matrix constraint $\mathbf{P}$ in (4.4) as a product of two matrices, $\mathbf{P} = \mathbf{BA}$, and incorporate the left matrix $\mathbf{B}$ into the following convex cost function:

$$\min_{\underline{u},\underline{v}} \; f(\underline{u}) + g(\underline{v}) \text{ s.t. } \mathbf{A}\underline{u} = \underline{v} \tag{4.13}$$

$$g(\underline{v}) = \begin{cases} 0, & \mathbf{B}\underline{v} = 0 \\ +\infty, & \mathbf{B}\underline{v} \neq 0. \end{cases} \tag{4.14}$$

For (4.13) to satisfy the sufficient conditions for convergence of ADMM in [75], $\mathbf{A}$ must have full rank. For our proposed algorithm, ADMM-tridiag, we design $\mathbf{B}$ and

$\mathbf{A}$ as follows:

$$\mathbf{B}\mathbf{A}\underline{u} = 0 \tag{4.15}$$

$$\mathbf{B} \triangleq \begin{bmatrix} \mathbf{I}_{N_r} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I}_{N_r} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I}_{N_r N_c} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_{N_r} & \mathbf{I}_{N_r} \end{bmatrix}$$

$$\mathbf{A} \triangleq \begin{bmatrix} -\mathbf{I}_{N_r} & 0 & 0 & 0 & \mathbf{C}_{\mathrm{H}} \\ 0 & -\mathbf{I}_{N_r} & 0 & \mathbf{C}_{\mathrm{V}} & 0 \\ 0 & 0 & -\mathbf{I}_{N_r N_c} & \frac{1}{2}\mathbf{S} & \frac{1}{2}\mathbf{S} \\ 0 & 0 & 0 & -\mathbf{I}_{N_r} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_{N_r} \end{bmatrix}.$$

For convenience, we describe $\underline{v}$ in terms of its block elements: $\underline{v} \triangleq (v_0, \cdots, v_4)$. The full rank of $\mathbf{A}$, combined with the following alternating minimization framework, satisfies the convergence conditions for ADMM [75]. Thus, (4.13)-(4.15) describe an instance of ADMM and guarantees convergence to a minimizer of (4.13). To handle the constraints of (4.13), ADMM-tridiag uses the following augmented Lagrangian, similar to Section 4.3.1:

$$\mathcal{L}\left(\underline{u}, \underline{v}, \underline{\eta}; \mathbf{M}\right) = f(\underline{u}) + g(\underline{v}) + \frac{1}{2}\left\|\mathbf{A}\underline{u} - \underline{v} - \underline{\eta}\right\|_{\mathbf{M}}^2. \tag{4.16}$$

The dual variables $\underline{\eta} \triangleq (\eta_0, \ldots, \eta_4) \in \mathbb{C}^{N_{\mathrm{r}}(4+N_{\mathrm{c}})}$ have an additional block-element

compared to the AL case. For convenience, we reuse the notation for the dual variables for both algorithms. The matrix $\mathbf{M}$ for ADMM-tridiag is also larger than in the AL case, consisting of blocks $\{\mathbf{M}_i\}_{i=0,\ldots,4}$.

Following [65] [75], ADMM-tridiag alternates between updating $\underline{u}$, $\underline{v}$, and the dual variables $\underline{\eta}$. Examining the $\underline{v}$ update and recalling (4.14), we see that the role of $\underline{v}$ simplifies greatly by considering the feasible set of $\underline{v}$, $\Omega \triangleq \{\underline{v} \in \mathbb{C}^{(4+N_c)N_r} : \mathbf{B}\underline{v} = 0\}$, resulting in the following alternating updates:

$$\underline{u}^{(n+1)} \underset{\epsilon_n}{\approx} \underset{\underline{u}}{\operatorname{argmin}} f(\underline{u}) + \frac{1}{2}\left\|\mathbf{A}\underline{u} - \underline{v}^{(n)} - \underline{\eta}^{(n)}\right\|_{\mathbf{M}}^2 \tag{4.17}$$

$$\underline{v}^{(n+1)} \underset{\zeta_n}{\approx} \underset{\underline{v}\in\Omega}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{A}\underline{u}^{(n+1)} - \underline{v} - \underline{\eta}^{(n)}\right\|_{\mathbf{M}}^2 \tag{4.18}$$

$$\underline{\eta}^{(n+1)} = \underline{\eta}^{(n)} - \left(\mathbf{A}\underline{u}^{(n+1)} - \underline{v}^{(n+1)}\right). \tag{4.19}$$

Here, we allow for some inexactness in the updates of $\underline{u}$ and $\underline{v}$ at each iteration, $\{\epsilon_n\}$ and $\{\zeta_n\}$, respectively. ADMM convergence holds if the inexactness sequences are summable [72, Theorem 8] [75, Theorem 2.1].

The update for $\underline{v}$ is simple and exact to machine precision. Due to the simple structure of $\mathbf{B}$, we have $v_0, v_1, v_2 = 0$ and $v_4 = -v_3$. The entire update for $\underline{v}$ reduces to a quadratic minimization problem for the auxiliary variable v_3 of size N_r with simple, closed-form solution:

$$v_3^{(n+1)} = (\mathbf{M}_3 + \mathbf{M}_4)^{-1}\left[\mathbf{M}_3\left(-u_3^{(n+1)} - \eta_3^{(n)}\right) \quad + \mathbf{M}_4\left(-x^{(n+1)} + \eta_4^{(n)}\right)\right]. \tag{4.20}$$

For the joint minimization of $\underline{u}$ in (4.17), ADMM-tridiag employs alternating minimization, as detailed in Section 4.3.4. If one round of alternating minimization sufficiently approximates the joint minimizer of (4.17), the iterates of ADMM-tridiag, $\{x^{(n)}\}$, converge to a minimizer of (4.1), per [75]. This alternating minimization approach is common in other ADMM methods, such as [78] [18].

Compared to AL-tridiag in Section 4.3.1, the direct AL approach to (4.3), ADMM-tridiag involves one additional variable split. An alternative way to understand the effect of (4.15) is to describe this ADMM algorithm as the result of applying AL to the following variable splitting scheme for (4.1):

$$\min_{\underline{u}} \frac{1}{2} \|y - \mathbf{F}u_2\|^2 + \lambda \|u_0\|_1 + \lambda \|u_1\|_1 \tag{4.21}$$

$$\text{s.t. } u_0 = \mathbf{C}_{\mathrm{H}} x, \quad u_2 = \frac{1}{2}\mathbf{S}u_3 + \frac{1}{2}\mathbf{S}x,$$

$$u_1 = \mathbf{C}_{\mathrm{V}} u_3, \quad u_3 = -v_3, \quad v_3 = -x \tag{4.22}$$

This formulation indirectly enforces $u_3 = x$ through additional auxiliary variable v_3. However, written in this form it is less clear that the full-rank condition of [75] is satisfied, whereas that is clear in (4.13)-(4.15).

4.3.4 Variable Updates with Parallelizable Tridiagonal Solvers

This section describes the block-variable updates for alternating minimization of (4.17). These variable updates are very similar for both proposed methods, AL-tridiag and ADMM-tridiag, differing only in tuning parameter indices and use of $\underline{v}$

and $\underline{\eta}$ on the right-hand side of these updates.

First, we consider the special case where some blocks of $\mathbf{M}$ are constant diagonal matrices. We leave $\mathbf{M}_3$ and $\mathbf{M}_4$ as general positive definite diagonal matrices for reasons explained in Section 4.4.4. Letting $\mathbf{M}_i \triangleq \mu_i \mathbf{I},\ \mu_i > 0$ for $i = 0, 1, 2$ and leveraging the constraint that $v_4 = -v_3$, (4.17) expands to:

$$\underline{u}^{(n+1)} = \underset{x,u_0,\ldots,u_3}{\operatorname{argmin}} \frac{1}{2} \|y - \mathbf{F}u_2\|^2 + \lambda \|u_0\|_1 + \lambda \|u_1\|_1 + \frac{\mu_0}{2} \left\| -u_0 + \mathbf{C}_{\mathrm{H}} x - \eta_0^{(n)} \right\|^2 + \frac{\mu_1}{2} \left\| -u_1 + \mathbf{C}_{\mathrm{V}} u_3 - \eta_1^{(n} \right.$$

(4.23)

ADMM-tridiag uses alternating minimization to update the blocks of $\underline{u}$. The resulting variable updates are:

$$u_0^{(n+1)} = \operatorname{soft}\left(\mathbf{C}_{\mathrm{H}} x^{(n)} - \eta_0^{(n)}, \frac{\lambda}{\mu_0} \right) \tag{4.24}$$

$$u_1^{(n+1)} = \operatorname{soft}\left(\mathbf{C}_{\mathrm{V}} u_3^{(n)} - \eta_1^{(n)}, \frac{\lambda}{\mu_1} \right) \tag{4.25}$$

$$u_2^{(n+1)} = \mathbf{H}_2^{-1} \left(\mathbf{F}' y \mu_2 \left(\frac{1}{2} \mathbf{S} u_3^{(n)} + \frac{1}{2} \mathbf{S} x^{(n)} - \eta_2^{(n)} \right) \right) \tag{4.26}$$

$$u_3^{(n+1)} = \mathbf{H}_3^{-1} \left(\mu_1 \mathbf{C}_{\mathrm{V}}' \left(u_1^{(n+1)} + \eta_1^{(n)} \right) + \frac{\mu_2}{2} \mathbf{S}' \left(u_2^{(n+1)} - \frac{1}{2} \mathbf{S} x^{(n)} + \eta_2^{(n)} \right) + \mathbf{M}_3 \left(-v_3^{(n)} - \eta_3^{(n)} \right) \right)$$

(4.27)

$$x^{(n+1)} = \mathbf{H}_{\mathrm{x}}^{-1} \left(\mu_0 \mathbf{C}_{\mathrm{H}}' \left(u_0^{(n+1)} + \eta_0^{(n)} \right) + \frac{\mu_2}{2} \mathbf{S}' \left(u_2^{(n+1)} - \frac{1}{2} \mathbf{S} u_3^{(n+1)} + \eta_2^{(n)} \right) + \mathbf{M}_4 \left(-v_3^{(n)} + \eta_4^{(n)} \right) \right). \tag{4.28}$$

The soft-thresholding operator performs element-wise shrinkage for the ℓ_1 norm using a given threshold
τ: $\mathrm{soft}(x, \tau) \triangleq \mathrm{sign}(x) \max(|x| - \tau, 0)$. Thus, (4.24) and (4.25) provide simple, direct solutions for updating u_0 and u_1. The bulk of the computation is "inverting" the following Hessians:

$$\mathbf{H}_2 \triangleq \mathbf{F}'\mathbf{F} + \mu_2 \mathbf{I} = \mathbf{Q} \left(\mathbf{\Lambda}_F + \mu_2 \mathbf{I} \right) \mathbf{Q}' \tag{4.29}$$

$$\mathbf{H}_3 \triangleq \mu_1 \mathbf{C}_{\mathrm{V}}' \mathbf{C}_{\mathrm{V}} + \frac{\mu_2}{4} \mathbf{S}'\mathbf{S} + \mathbf{M}_3 \tag{4.30}$$

$$\mathbf{H}_{\mathrm{x}} \triangleq \mu_0 \mathbf{C}_{\mathrm{H}}' \mathbf{C}_{\mathrm{H}} + \frac{\mu_2}{4} \mathbf{S}'\mathbf{S} + \mathbf{M}_4. \tag{4.31}$$

Due to Cartesian undersampling, $\mathbf{H}_2$ is diagonalizable via N_c FFTs, each of which operate efficiently in $\mathcal{O}(N_r \log N_r)$ time. The multi-coil FFT operator is denoted $\mathbf{Q}$. The x update in (4.28) uses $\mathbf{H}_{\mathrm{x}}$ in (4.31), a block diagonal matrix with tridiagonal blocks (BDTB) that can be "inverted" in $\mathcal{O}(N_r)$ time via Gaussian elimination. Because it is block diagonal, we parallelize this variable update over the N_y independent blocks. We reformulate the u_3 update as another instance of the same BDTB inverse problem through permutation and solve (4.27) using a tridiagonal solver parallelized over N_x blocks. Each variable update is exact and easy to implement.

We designed the proposed variable splitting in (4.2) to enable these efficient variable updates. The separation of horizontal and vertical finite differences into u_0 and u_1 allows for BDTB structures in $\mathbf{H}_3$ and $\mathbf{H}_\mathrm{x}$. If the finite difference matrices were combined in the same auxiliary variable as in AL-P2 [67], the resulting Hessian would have a block-tridiagonal with tridiagonal blocks (BTTB) structure and the associated variable update would require a more computationally costly solution [82] [83].

Algorithm 1 summarizes the overall procedure for ADMM-tridiag. All of the variable updates are done in place, so the memory requirements for storing x, $u_0, \ldots, u_3$, v_3, and $\underline{\eta}$ are $8N_\mathrm{r}(4 + N_\mathrm{c})$ bytes for $N_\mathrm{r}(4 + N_\mathrm{c})$ complex single-precision values. For AL-tridiag, the $\underline{u}$ updates are very similar.

Algorithm 2 ADMM-Tridiag

Initialize x to square root of sum-of-squares (SoS)
    of zero-filled iFFT images.
Initialize $u_0 = \mathbf{C}_\mathrm{H} x$, $u_1 = \mathbf{C}_\mathrm{V} u_3$, $u_3 = x$,
  $u_2 = \frac{1}{2}\mathbf{S}u_3 + \frac{1}{2}\mathbf{S}x$, $v_3 = -x$, $\underline{\eta} = \underline{0}$
for $n \leq$ total iterations **do**
    Compute $u_0^{(n+1)}$ via soft-thresholding (4.24)
    Compute $u_1^{(n+1)}$ via soft-thresholding (4.25)
    Compute $u_2^{(n+1)}$ via FFTs (4.26)
    Compute $u_3^{(n+1)}$ via tridiagonal solver (4.27)
    Compute $x^{(n+1)}$ via tridiagonal solver (4.28)
    Compute $v_3^{(n+1)}$ via (4.20)
    Compute $\underline{\eta}^{(n+1)}$ via (4.19)
end for

4.3.5 Regularization with Finite Differences and Wavelets

The variable splitting scheme in (4.22) readily generalizes to combinations of finite difference and orthonormal wavelet regularization. Let $\mathbf{W}$ be an orthonormal wavelet transform (e.g., Haar wavelets). Then the following combined TV / wavelet sparsity cost-function can be manipulated to resemble (4.1):

$$\begin{aligned}
&\frac{1}{2}\left\|y-\mathbf{FS}x\right\|_2^2+\lambda_1\left\|\mathbf{C}_{\mathrm{H}}x\right\|_1 \\
&\qquad\qquad +\lambda_1\left\|\mathbf{C}_{\mathrm{V}}x\right\|_1+\lambda_2\left\|\mathbf{W}x\right\|_1 &(4.32)\\
&=\frac{1}{2}\left\|y-\mathbf{FS}x\right\|_2^2+\lambda_1\left\|\tilde{\mathbf{C}}_{\mathrm{H}}x\right\|_1+\lambda_1\left\|\tilde{\mathbf{C}}_{\mathrm{V}}x\right\|_1 &(4.33)\\
&\tilde{\mathbf{C}}_{\mathrm{H}}\triangleq\begin{bmatrix}\mathbf{C}_{\mathrm{H}}\\ \frac{\alpha_w\lambda_2}{\lambda_1}\mathbf{W}\end{bmatrix};\quad \tilde{\mathbf{C}}_{\mathrm{V}}\triangleq\begin{bmatrix}\mathbf{C}_{\mathrm{V}}\\ \frac{(1-\alpha_w)\lambda_2}{\lambda_1}\mathbf{W}\end{bmatrix}. &(4.34)
\end{aligned}$$

An additional spatial regularization parameter, λ_2, controls the weight of the wavelet regularization. Due to the orthonormality of $\mathbf{W}$, we can use this regularizer for both AL-tridiag and ADMM-tridiag, with only minor changes to variable updates (4.24) - (4.28). For ADMM-tridiag, the only Hessians affected by introducing wavelets are:

$$\tilde{\mathbf{H}}_3\triangleq\mu_1\mathbf{C}_{\mathrm{V}}'\mathbf{C}_{\mathrm{V}}+\frac{\mu_2}{4}\mathbf{S}'\mathbf{S}+\tilde{\mathbf{M}}_3 \tag{4.35}$$

$$\tilde{\mathbf{H}}_{\mathrm{x}}\triangleq\mu_0\mathbf{C}_{\mathrm{H}}'\mathbf{C}_{\mathrm{H}}+\frac{\mu_2}{4}\mathbf{S}'\mathbf{S}+\tilde{\mathbf{M}}_4, \tag{4.36}$$

with positive definite matrices $\tilde{\mathbf{M}}_3=\mathbf{M}_3+\frac{(1-\alpha_w)^2\lambda_2^2}{\lambda_1^2}\mathbf{I}$ and $\tilde{\mathbf{M}}_4=\mathbf{M}_4+\frac{\alpha_w^2\lambda_2^2}{\lambda_1^2}\mathbf{I}$. Hes-

sians $\tilde{\mathbf{H}}_3$ and $\tilde{\mathbf{H}}_x$ are still BDTB, and they can be "inverted" efficiently with a parallelizable tridiagonal solver routine. This wavelet-inclusive variation is featured in experimental results in Section 4.6.3.

4.4 Fully Parallelized ADMM: ADMM-FP-Tridiag

In this supplementary material, we introduce another variation of ADMM-tridiag [9] that expands constraint matrix $\mathbf{P}$ in (4.4) even further. ADMM-tridiag leverages [75] to ensure convergence via equivalence with the Alternating Direction Method of Multipliers (ADMM). As discussed in (4.3.3), ADMM-tridiag relies on one cycle of alternating minimization to solve the joint minimization problem in (4.17). Though this is a common approach in many ADMM methods, we also investigate a fully parallelized alternative, called ADMM-FP-tridiag, which decomposes the constraint matrix in such a way that the resulting joint minimization problem and the alternating minimization approach are identical. This lessens the degree to which this update is inexact, but does not compensate for other sources of inexact computation, such as machine precision. This algorithm is inspired by the fully parallelized ADMM in [38].

To design a fully parallelized variable update scheme, we define constraint matrices $\mathbf{A}_{\text{FP}}$ and $\mathbf{B}_{\text{FP}}$ as follows:

$$\mathbf{B}_{\mathrm{FP}} \triangleq \begin{bmatrix} \mathbf{I} & \mathbf{I} & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & \mathbf{I} & \mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & \mathbf{I} & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} & \mathbf{I} \end{bmatrix}, \quad \mathbf{A}_{\mathrm{FP}} \triangleq \begin{bmatrix} -\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{C}_{\mathrm{H}} \\ 0 & -\mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_{\mathrm{V}} & 0 \\ 0 & 0 & -\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}\mathbf{S} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\mathbf{S} \\ 0 & 0 & 0 & -\mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix}. \quad (4.37)$$

This results in a joint minimization problem for $\underline{u}$ that is entirely decoupled for each block of $\underline{u}$. Tradeoffs include $\underline{v}_{\mathrm{FP}}$ ($\in \mathbb{C}^{7N_r+2N_cN_r}$) that is much larger than $\underline{v}$ in ADMM-tridiag and more non-trivial blocks of $\underline{v}_{\mathrm{FP}}$ to update. Likewise, $\underline{\eta}_{\mathrm{FP}}$ is also larger than in ADMM-tridiag.

Choosing $\mathbf{M}_i \triangleq \mu_i \mathbf{I},\ \mu_i \in \mathbb{R}$ for $i = 0, \ldots, 6$, the expansion of (4.17) is:

$$\begin{aligned} \underline{u}^{(n+1)} = \underset{x,u_0,u_1,u_2,3}{\operatorname{argmin}} & \frac{1}{2}\left\|y - \mathbf{F}u_2\right\|^2 + \lambda\left\|u_0\right\|_1 + \lambda\left\|u_1\right\|_1 + \frac{\mu_0}{2}\left\|-u_0^{(n)} - v_0^{(n)} - \eta_0^{(n)}\right\|^2 \\ & + \frac{\mu_1}{2}\left\|\mathbf{C}_{\mathrm{H}}x^{(n)} - v_1^{(n)} - \eta_1^{(n)}\right\|^2 + \frac{\mu_2}{2}\left\|-u_1^{(n)} - v_2^{(n)} - \eta_3^{(n)}\right\|^2 + \frac{\mu_3}{2}\left\|\mathbf{C}_{\mathrm{V}}u_3^{(n)} - v_3^{(n)} - \eta_3^{(n)}\right\|^2 \\ & + \frac{\mu_4}{2}\left\|-u_2^{(n)} - v_4^{(n)} - \eta_4^{(n)}\right\|^2 + \frac{\mu_5}{2}\left\|\frac{1}{2}\mathbf{S}u_3^{(n)} - v_5^{(n)} - \eta_5^{(n)}\right\|^2 + \frac{\mu_6}{2}\left\|\frac{1}{2}\mathbf{S}x^{(n)} - v_6^{(n)} - \eta_6^{(n)}\right\|^2 \\ & + \frac{1}{2}\left\|-u_3^{(n)} - v_7^{(n)} - \eta_7^{(n)}\right\|_{\mathbf{M}_7}^2 + \frac{1}{2}\left\|x^{(n)} - v_8^{(n)} - \eta_8^{(n)}\right\|_{\mathbf{M}_8}^2. \end{aligned} \quad (4.38)$$

The variable updates that result from alternating minimization are:

$$u_0^{(n+1)} = \text{soft}\left(v_0^{(n)} - \eta_0^{(n)}, \frac{\lambda}{\mu_0}\right) \tag{4.39}$$

$$u_1^{(n+1)} = \text{soft}\left(v_2^{(n)} - \eta_3^{(n)}, \frac{\lambda}{\mu_2}\right) \tag{4.40}$$

$$u_2^{(n+1)} = \mathbf{H}_{2,\text{FP}}^{-1}\left(\mathbf{F}'y + \mu_4\left(v_4^{(n)} - \eta_4^{(n)}\right)\right) \tag{4.41}$$

$$u_3^{(n+1)} = \mathbf{H}_{3,\text{FP}}^{-1}\left(\mu_3\mathbf{C}_\text{V}'\left(v_3^{(n)} + \eta_3^{(n)}\right) + \frac{\mu_5}{2}\mathbf{S}'\left(v_5^{(n)} + \eta_5^{(n)}\right) + \mu_7\left(-v_7^{(n)} - \eta_7^{(n)}\right)\right) \tag{4.42}$$

$$x^{(n+1)} = \mathbf{H}_{\text{x,FP}}^{-1}\left(\mu_1\mathbf{C}_\text{H}'\left(v_1^{(n)} + \eta_1^{(n)}\right) + \frac{\mu_6}{2}\mathbf{S}'\left(v_6^{(n)} + \eta_6^{(n)}\right) + \mu_8\left(v_8^{(n)} + \eta_8^{(n)}\right)\right). \tag{4.43}$$

The soft-thresholding operator performs element-wise shrinkage for the ℓ_1 norm using a given threshold τ: $\text{soft}(x,\tau) \triangleq \text{sign}(x)\max(x-\tau, 0)$. The bulk of the computational cost lies in inverting the following Hessians:

$$\mathbf{H}_{2,\text{FP}} \triangleq \mathbf{F}'\mathbf{F} + \mu_4\mathbf{I} = \mathbf{Q}\left(\mathbf{\Lambda}_F + \mu_4\mathbf{I}\right)\mathbf{Q}' \tag{4.44}$$

$$\mathbf{H}_{3,\text{FP}} \triangleq \mu_3\mathbf{C}_\text{V}'\mathbf{C}_\text{V} + \frac{\mu_5}{4}\mathbf{S}'\mathbf{S} + \mu_7\mathbf{I} \tag{4.45}$$

$$\mathbf{H}_{\text{x,FP}} \triangleq \mu_1\mathbf{C}_\text{H}'\mathbf{C}_\text{H} + \frac{\mu_6}{4}\mathbf{S}'\mathbf{S} + \mu_8\mathbf{I}. \tag{4.46}$$

As in AL-P2, $\mathbf{H}_{2,\text{FP}}$ is diagonalizable via FFTs, which operate efficiently in $\mathcal{O}(n\log n)$ time. The x update in (4.43) requires inverting $\mathbf{H}_{\text{x,FP}} \triangleq \mu_1\mathbf{C}_\text{H}'\mathbf{C}_\text{H} + \frac{1}{2}\frac{1}{2}\mu_6\mathbf{S}'\mathbf{S} + \mu_8\mathbf{I}$, a block diagonal matrix with tridiagonal blocks that can be inverted in $\mathcal{O}(n)$ time via Gaussian elimination. Because it is block diagonal, we parallelize this variable

update over the N_y blocks, whose computations are independent of one another. As in ADMM-tridiag, we reformulate the u_3 update as another instance of the same minimization problem through permutation. The remaining variable updates are exact and easy to implement.

The updates for $\underline{v}_{\text{FP}}$ all simplify to quadratic problems and can be implemented

directly as follows:

$$v_0^{(n+1)} = \frac{1}{\mu_0+\mu_1}\left(\mu_0\left(-u_0^{(n+1)}-\eta_0^{(n)}\right)+\mu_1\left(-\mathbf{C}_{\mathrm{H}}x^{(n+1)}+\eta_1^{(n)}\right)\right) \tag{4.47}$$

$$v_1^{(n+1)} = -\,v_0^{(n)} \tag{4.48}$$

$$v_2^{(n+1)} = \frac{1}{\mu_2+\mu_3}\left(\mu_2\left(-u_1^{(n+1)}-\eta_2^{(n)}\right)+\mu_3\left(-\mathbf{C}_{\mathrm{V}}u_3^{(n+1)}+\eta_3^{(n)}\right)\right) \tag{4.49}$$

$$v_3^{(n+1)} = -\,v_2^{(n)} \tag{4.50}$$

$$\begin{aligned} v_4^{(n+1)} = \frac{1}{\mu_4\mu_5+\mu_5\mu_6+\mu_4\mu_5}\Bigg[&(\mu_5+\mu_6)\left(\mu_4\left(-u_2^{(n+1)}+\eta_2^{(n)}\right)+\mu_6\left(\frac{1}{2}\mathbf{S}x^{(n+1)}-\eta_6^{(n)}\right)\right)\\ &-\mu_6\left(\mu_5\left(\frac{1}{2}\mathbf{S}u_3^{(n+1)}-\eta_5^{(n)}\right)-\mu_6\left(\frac{1}{2}\mathbf{S}x^{(n+1)}-\eta_6^{(n)}\right)\right)\Bigg] \end{aligned} \tag{4.51}$$

$$\begin{aligned} v_5^{(n+1)} = \frac{1}{\mu_4\mu_5+\mu_5\mu_6+\mu_4\mu_5}\Bigg[&-\mu_6\left(\mu_4\left(-u_2^{(n+1)}+\eta_2^{(n)}\right)+\mu_6\left(\frac{1}{2}\mathbf{S}x^{(n+1)}-\eta_6^{(n)}\right)\right)\\ &(\mu_4+\mu_6)\left(\mu_5\left(\frac{1}{2}\mathbf{S}u_3^{(n+1)}-\eta_5^{(n)}\right)+\mu_6\left(\frac{1}{2}\mathbf{S}x^{(n+1)}-\eta_6^{(n)}\right)\right)\Bigg] \end{aligned} \tag{4.52}$$

$$v_6^{(n+1)} = -\,v_4^{(n+1)} - v_5^{(n+1)} \tag{4.53}$$

$$v_7^{(n+1)} = \frac{1}{\mu_7+\mu_8}\left(\mu_7\left(-u_3^{(n+1)}-\eta_7^{(n)}\right)+\mu_8\left(-x^{(n+1)}+\eta_8^{(n)}\right)\right) \tag{4.54}$$

$$v_8^{(n+1)} = -\,v_7^{(n+1)}. \tag{4.55}$$

By enforcing constraints arising from the structure of $\mathbf{B}_{\mathrm{FP}}$, the full minimization of $\underline{v}_{\mathrm{FP}}$ reduces to the computation of just five of the nine block variables: v_0, v_2, v_4, v_5, and v_7. The extraneous four variables can be expressed in terms of the remaining four and omitted from the algorithm altogether.

Finally, the dual variables are updated as follows:

$$\underline{\eta}_{\mathrm{FP}}^{(n+1)} = \underline{\eta}_{\mathrm{FP}}^{(n)} - \left(\mathbf{A}_{FP}\underline{u}^{(n+1)} - \underline{v}_{\mathrm{FP}}^{(n+1)}\right). \tag{4.56}$$

ADMM-FP-tridiag is highly amenable to parallelization. Each block update for $\underline{u}$ can be done in parallel, as can each block update of $\underline{v}_{\mathrm{FP}}$, as described in 3. This opportunity for parallelization helps to offset some of the additional computational cost incurred from the greater number of variable updates per iteration.

In summary, ADMM-FP-tridiag eliminates alternating minimization across the blocks of $\underline{u}$ and $\underline{v}_{\mathrm{FP}}$ by decoupling each of the four original variable splitting constraints (4.2). This results in additional auxiliary variables in $\underline{v}_{\mathrm{FP}}$ and dual variables $\underline{\eta}_{\mathrm{FP}}$. However, each block-variable has blocks that can be updated in parallel with one another. This approach also provides exact variable updates (to within numerical precision) by avoiding alternating minimization. The procedure for ADMM-FP-tridiag is summarized in Algorithm 3.

Algorithm 3 ADMM-FP-tridiag

Initialize: $u_0, u_1, u_2, u_3, x, v_0, v_2, v_4, v_5, v_7, \underline{\eta}_{\text{FP}}$
for $n \leq$ total iterations **do**
  do in parallel
    Compute $u_0^{(n+1)}$ using soft-thresholding (4.39)
    Compute $u_1^{(n+1)}$ using soft-thresholding (4.40)
    Compute $u_2^{(n+1)}$ using FFTs (4.41)
    Compute $u_3^{(n+1)}$ using parallelized tridiagonal solver (4.42)
    Compute $x^{(n+1)}$ using parallelized tridiagonal solver (4.43)
  end parfor
  do in parallel
    Compute $v_0^{(n+1)}$ using (4.47)
    Compute $v_2^{(n+1)}$ using (4.49)
    Compute $v_4^{(n+1)}$ using (4.51)
    Compute $v_5^{(n+1)}$ using (4.52)
    Compute $v_7^{(n+1)}$ using (4.54)
  end parfor
  Compute $\underline{\eta}_{\text{FP}}^{(n+1)}$ using (4.56)
end for

4.4.1 Parameter Selection for ADMM-FP-tridiag

This proposed algorithm, ADMM-FP-tridiag, has a total of nine AL tuning parameters, substantially more than for ADMM-tridiag. Using the condition number heuristics from [67] on Hessians (4.45)-(4.46) leaves six remaining degrees of freedom. This is a result of the two tridiagonal Hessians that include entirely different tuning parameters, unlike the case with ADMM-tridiag:

$$\mathbf{H}_{3,\mathrm{FP}} \triangleq \mu_3 \mathbf{C}_{\mathrm{V}}' \mathbf{C}_{\mathrm{V}} + \frac{\mu_5}{4} \mathbf{S}'\mathbf{S} + \mathbf{M}_7$$
$$\mathbf{H}_{\mathrm{x,FP}} \triangleq \mu_1 \mathbf{C}_{\mathrm{H}}' \mathbf{C}_{\mathrm{H}} + \frac{\mu_6}{4} \mathbf{S}'\mathbf{S} + \mathbf{M}_8.$$

The difficulty of designing these parameters is one of the tradeoffs in choosing the fully parallelized alternative of ADMM-tridiag. Tuning parameters for the following speed comparisons were chosen to enforce the aforementioned Hessian values, leaving some degrees of freedom unexplored.

4.4.2 Special Case: Image Inpainting

To highlight the value of non-periodic boundary conditions, we examine a specific application of the proposed variable splitting scheme in (4.2), namely inpainting. Image inpainting fills in image data that is lost or corrupted. Many image inpainting, deblurring, and denoising methods use finite difference regularizers like anisotropic total variation (TV) [84] [79] [85].

Let $\mathbf{D}$ be a binary, diagonal matrix whose nonzero entries denote the set of indices in the inpainting domain. Setting $\mathbf{FS} = \mathbf{D}$ in the CS-SENSE-MRI cost function (4.1) leads to the following simpler image inpainting problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \|y - \mathbf{D}x\|_2^2 + \lambda \left\| \tilde{\mathbf{C}}_{\mathrm{H}} x \right\|_1 + \lambda \left\| \tilde{\mathbf{C}}_{\mathrm{V}} x \right\|_1, \tag{4.57}$$

where $\tilde{\mathbf{C}}_{\mathrm{H}}$ and $\tilde{\mathbf{C}}_{\mathrm{V}}$ are defined in (4.34).

For inpainting, we simplify the VS scheme developed for SENSE MRI in (4.2) to:

$$\begin{aligned} &\underline{\hat{u}} = \underset{\underline{u}}{\operatorname{argmin}} f(\underline{u}) \\ &f(\underline{u}) = \frac{1}{2} \left\| y - \mathbf{D} \left(\frac{1}{2} u_2 + \frac{1}{2} x \right) \right\|^2 + \lambda \|u_0\|_1 + \lambda \|u_1\|_1 \\ &\text{s.t. } u_0 = \tilde{\mathbf{C}}_{\mathrm{H}} x,\ u_1 = \tilde{\mathbf{C}}_{\mathrm{V}} u_2,\ u_2 = x. \end{aligned} \tag{4.58}$$

The resulting VS algorithm is a simplification of AL-tridiag, which we denote AL-tridiag-inpaint. Due to the entirely diagonal system matrix, the variable updates consist only of shrinkage and tridiagonal solver updates, eliminating the need for any FFT-based updates. Similarly, we can generalize ADMM-tridiag to the inpainting problem by applying an extra variable splitting, resulting in an additional quadratic minimization problem in each iteration. Section 4.6.3 illustrates the effect of non-periodic boundary conditions for noisy inpainting.

4.4.3 Comparison with AL-P2

We compare ADMM-tridiag with AL-P2 [67], a fast VS scheme designed for CS-SENSE-MRI. The original AL-P2 version in [67] used periodic boundary conditions, whereas here we modify it for the non-periodic conditions of (4.1) and call the modified algorithm AL-P2-NC. The suffix "NC" refers to the non-circulant Hessian we describe in this section. To define AL-P2-NC, we stack the finite difference matrices into a tall matrix, $\mathbf{R} \triangleq [\mathbf{C}_{\mathrm{H}}; \mathbf{C}_{\mathrm{V}}]$. Applying the AL-P2 variable splitting scheme to (4.1) yields the following constrained cost function:

$$\min_{x,u,v,z} \|y - \mathbf{F}u\|^2 + \lambda \|v\|_1 \tag{4.59}$$

$$\text{s.t. } u = \mathbf{S}x,\ v = \mathbf{R}z,\ z = x. \tag{4.60}$$

Applying the augmented Lagrangian to the constrained cost function (4.59) and using alternating minimization results in variable updates like in [67]:

$$x^{(n+1)} = \mathbf{H}_{\mathrm{x}}^{-1} \left(\begin{array}{c} \mu_u \mathbf{S}' \left(u^{(n)} - {\eta_u}^{(n)} \right) + \mu_z \left(z^{(n)} - {\eta_z}^{(n)} \right) \\ \end{array} \right) \tag{4.61}$$

$$u^{(n+1)} = \mathbf{H}_{\mathrm{u}}^{-1} \left(\mathbf{F}'y + \mu_u \left(\mathbf{S}x^{(n)} + {\eta_u}^{(n)} \right) \right) \tag{4.62}$$

$$v^{(n+1)} = \text{soft}\left(\mathbf{R}z^{(n)} + {\eta_v}^{(n)}, \frac{\lambda}{\mu_v}\right) \tag{4.63}$$

$$z^{(n+1)} = \mathbf{H}_{\text{z}}^{-1}\left(\begin{array}{l} \mu_v \mathbf{R}'\left(v^{(n+1)} - {\eta_v}^{(n)}\right) + \mu_z\left(x^{(n+1)} + {\eta_z}^{(n)}\right) \\ \end{array}\right). \tag{4.64}$$

The scalar AL penalty parameters, $\mu_u, \mu_v, \mu_z > 0$, do not affect the final solution but do affect convergence rate. The Hessians for x and u are simple to invert and are the same as in ADMM-tridiag:

$$\mathbf{H}_{\text{x}} = \mu_u \mathbf{S}'\mathbf{S} + \mu_z \mathbf{I} \tag{4.65}$$

$$\mathbf{H}_{\text{u}} = \mathbf{F}'\mathbf{F} + \mu_u \mathbf{I} = \mathbf{Q}\left(\Lambda_F + \mu_u \mathbf{I}\right)\mathbf{Q}'. \tag{4.66}$$

The Hessian $\mathbf{H}_{\text{z}}$ for the z update is BTTB (non-circulant) as follows:

$$\mathbf{H}_{\text{z}} = \mu_v \mathbf{R}'\mathbf{R} + \mu_z \mathbf{I}, \tag{4.67}$$

for which there is no $\mathcal{O}\left(N_{\text{r}}\right)$ solver. To "invert" $\mathbf{H}_{\text{z}}$ for (4.64) we applied one iteration of preconditioned gradient descent with a circulant preconditioner. The resulting computation per iteration is essentially identical to that of the original AL-P2 with periodic boundary conditions in [67].

4.4.4 Parameter Selection

For all of the experiments shown in Section 4.6 and in the supplement, we manually chose the spatial regularization parameter, λ, so that the converged image $x^{(\infty)}$

resembled the true image (for inpainting and CS-SENSE-MRI simulations in the supplement) or the fully-sampled body coil image (for *in vivo* data).

As with other AL algorithms, the tuning parameters $\mathbf{M}_0, \ldots, \mathbf{M}_4 \succeq 0$ do not affect the solution $\hat{x}$, but they can greatly affect the convergence rate. To facilitate comparison with AL-P2-NC, we chose the AL tuning parameters of AL-P2-NC based on the guidelines provided in [67, Eqns. (40)-(41)] as follows:

$$\begin{aligned} \kappa\left(\mathbf{H}_{\mathrm{u}}\right) = 24;\ \kappa\left(\mathbf{H}_{\mathrm{z}}\right) = 12; \\ \kappa\left(\mathbf{H}_{\mathrm{x}}\right) = 0.9\kappa(\mathbf{S}'\mathbf{S}). \end{aligned} \tag{4.68}$$

For consistency, we selected the AL tuning parameters of AL-tridiag and ADMM-tridiag with a similar heuristic strategy. For the Hessian of the multi-coil FFT step (4.26) we selected its associated parameter, μ_2, such that $\kappa\left(\mathbf{H}_2\right) = 24$, exactly as in AL-P2. Our proposed variable splitting (4.2) results in Hessians in which the regularizer is combined with the sensitivity encoding, so the remaining AL-P2 tuning rules are inapplicable. Instead, we selected the remaining tuning parameters (μ_0, μ_1, $\mathbf{M}_3$, and $\mathbf{M}_4$) to enforce $\kappa\left(\mathbf{H}_3\right) = \kappa\left(\mathbf{H}_{\mathrm{x}}\right) = 12$. Because our remaining Hessians $\mathbf{H}_3$ and $\mathbf{H}_{\mathrm{x}}$ partially consist of $\mathbf{C}_{\mathrm{V}}'\mathbf{C}_{\mathrm{V}}$ and $\mathbf{C}_{\mathrm{H}}'\mathbf{C}_{\mathrm{H}}$, respectively, we choose to enforce the condition number of 12 used for $\mathbf{H}_{\mathrm{z}}$ of AL-P2, which is characterized by the periodic boundary finite differences. We choose to apply this heuristic over the alternative guideline based on $\kappa(\mathbf{S}'\mathbf{S})$ because our choices of $\mathbf{M}_3$ and $\mathbf{M}_4$ in (4.74)-(4.75) below make $\mathbf{H}_3$ and $\mathbf{H}_{\mathrm{x}}$ approximately circulant, but far from diagonal.

First we chose the parameters that interact with the thresholding steps, μ_0 and μ_1,

based on the maximum value of the initial image, $x_{\max}$, and spatial regularization parameter, λ:

$$\mu_0 = \mu_1 = \frac{\lambda}{0.02\ x_{\max}}. \tag{4.69}$$

This sets the threshold of the shrinkage step in (4.24) and (4.25) at 2% of the maximum initial image value. This threshold worked well for the noise level of the following simulated and *in vivo* experiments. Recalling the BDTB structures of $\mathbf{H}_3$ (4.30) and $\mathbf{H}_{\mathrm{x}}$ (4.31), we designed $\mathbf{M}_3$ and $\mathbf{M}_4$ to enforce the following conditions for scalar $c_3, c_4 > 0$:

$$c_3\mathbf{I} = \frac{\mu_2}{4}\mathbf{S}'\mathbf{S} + \mathbf{M}_3 \tag{4.70}$$

$$c_4\mathbf{I} = \frac{\mu_2}{4}\mathbf{S}'\mathbf{S} + \mathbf{M}_4. \tag{4.71}$$

The constant diagonal term results from allowing a spatially varying $\mathbf{M}_3$ and $\mathbf{M}_4$. Therefore, $\mathbf{M}_3$ is higher in regions where the sum-of-squares (SoS) of the sensitivity maps is low and vice versa. Intuitively this results in stronger enforcement of the $u_3 = -v_3$ and $x = -v_3$ constraints in spatial regions where the $u_2 = \frac{1}{2}\mathbf{S}u_3 + \frac{1}{2}\mathbf{S}x$ constraint provides less information. We use [86] for an analytical solution for maximum and minimum eigenvalues of $\mathbf{C}'_{\mathrm{H}}\mathbf{C}_{\mathrm{H}}$ and $\mathbf{C}'_{\mathrm{V}}\mathbf{C}_{\mathrm{V}}$. Due to (4.70)-(4.71), $\mathbf{H}_3$ and $\mathbf{H}_{\mathrm{x}}$ are

approximately circulant, and eigenvalue analysis of $\mathbf{H}_3$ and $\mathbf{H}_{\mathrm{x}}$ becomes simple:

$$\kappa\left(\mathbf{H}_3\right) = \frac{\mu_1 \lambda_{\max}\left(\mathbf{C}_{\mathrm{V}}'\mathbf{C}_{\mathrm{V}}\right) + c_3}{\mu_1 \lambda_{\min}\left(\mathbf{C}_{\mathrm{V}}'\mathbf{C}_{\mathrm{V}}\right) + c_3} \tag{4.72}$$

$$\kappa\left(\mathbf{H}_{\mathrm{x}}\right) = \frac{\mu_0 \lambda_{\max}\left(\mathbf{C}_{\mathrm{H}}'\mathbf{C}_{\mathrm{H}}\right) + c_4}{\mu_0 \lambda_{\min}\left(\mathbf{C}_{\mathrm{H}}'\mathbf{C}_{\mathrm{H}}\right) + c_4}. \tag{4.73}$$

Let $\hat{c}_3$ and $\hat{c}_x$ be the respective solutions for (4.72) and (4.73) for $\kappa\left(\mathbf{H}_3\right) = \kappa\left(\mathbf{H}_{\mathrm{x}}\right) = 12$. Then the values for $\mathbf{M}_3$ and $\mathbf{M}_4$ are as follows:

$$\mathbf{M}_3 \triangleq \max\left(\hat{c}_3\mathbf{I} - \frac{\mu_2}{4}\mathbf{S}'\mathbf{S},\ 10^{-3}\right) \tag{4.74}$$

$$\mathbf{M}_4 \triangleq \max\left(\hat{c}_4\mathbf{I} - \frac{\mu_2}{4}\mathbf{S}'\mathbf{S},\ 10^{-3}\right). \tag{4.75}$$

This choice of $\mathbf{M}_3$ and $\mathbf{M}_4$ is informed by the thresholding levels through (4.69) but also enforces the positive-definite condition for $\mathbf{M}_3$ and $\mathbf{M}_4$. We selected AL-tridiag parameters using the same procedure.

4.5 Variable Splitting Balance Parameter

This section describes possible variations of the proposed algorithms in which the variable splitting scheme does not exhibit the symmetry in (4.3), namely,

$$\hat{\underline{u}} = \underset{\underline{u}}{\operatorname{argmin}}\, f\left(\underline{u}\right) \tag{4.76}$$

$$f\left(\underline{u}\right) = \frac{1}{2}\left\|y - \mathbf{F}u_2\right\|^2 + \lambda\left\|u_0\right\|_1 + \lambda\left\|u_1\right\|_1$$

$$\begin{aligned} \text{s.t. } u_0 &= \mathbf{C}_{\mathrm{H}} x, & u_1 &= \mathbf{C}_{\mathrm{V}} u_3, \\ u_2 &= (1-\alpha)\mathbf{S}u_3 + \alpha \mathbf{S}x, & u_3 &= x, \end{aligned} \tag{4.77}$$

in which the variable splitting balance parameter, $\alpha \in [0, 1]$. In principle, using different choices of α for AL-tridiag, such as in (4.76)-(4.77) could lead to different limits than when $\alpha = \frac{1}{2}$, due to the lack of convergence theory. However, we found empirically that varying $\alpha = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ in AL-tridiag-inpaint resulted in solutions identical to machine precision.

The convergence guarantees of ADMM-tridiag and ADMM-FP-tridiag ensure that the algorithm will converge to the same solution for any $\alpha \in [0, 1]$. However, the choice of α may affect the convergence rate and is closely intertwined with the AL parameters.

Unlike the balance parameter for orthonormal wavelets, α_w, described in Section 4.3.5, α has a negligible effect on computation time per iteration. Rather, α affects the convergence rate by controlling the connectivity between the auxiliary variables and the extent to which one round of alternating minimization solves (4.6).

To investigate the effect of α on convergence rate, we conducted timing experiments for AL-tridiag-inpaint for varying α. The simpler inpainting problem includes fewer additional parameters that may obfuscate the role of α.

For the inpainting problem (4.58), the resulting Hessians for AL-tridiag-inpaint

are:

$$\mathbf{H}_2 = (1-\alpha)^2\,\mathbf{D}'\mathbf{D} + \mu_1 \tilde{\mathbf{C}}_{\mathrm{V}}'\tilde{\mathbf{C}}_{\mathrm{V}} + \mathbf{M}_2 \tag{4.78}$$

$$\mathbf{H}_{\mathrm{x}} = \alpha^2\mathbf{D}'\mathbf{D} + \mu_0 \tilde{\mathbf{C}}_{\mathrm{H}}'\tilde{\mathbf{C}}_{\mathrm{H}} + \mathbf{M}_2. \tag{4.79}$$

Because we tune AL parameters μ_0, μ_1, and $\mathbf{M}_2$ to enforce $\kappa(\mathbf{H}_2) = \kappa(\mathbf{H}_{\mathrm{x}})$, this suggests that an even distribution of the influence of the inpainting operator $\mathbf{D}$ across u_2 and x also allows for even influence of $\tilde{\mathbf{C}}_{\mathrm{H}}$ and $\tilde{\mathbf{C}}_{\mathrm{V}}$.

Figure 4.1 shows computation time comparisons to the MFISTA $x^{(\infty)}$ for AL-tridiag-inpaint for varying values of α. The clear speed margin for $\alpha = 0.5$ reinforces the specific choice of α in [9].

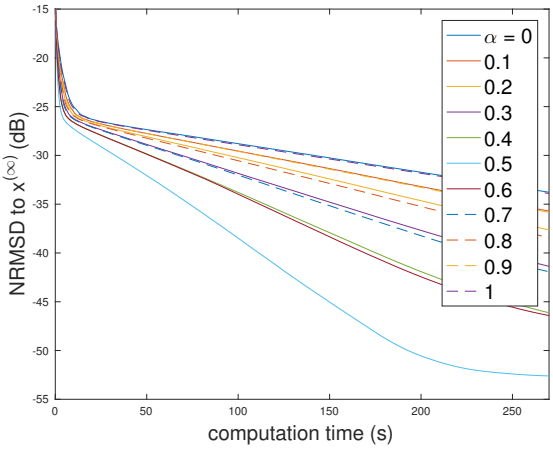


Figure 4.1: Convergence of $|x^{(n)} - x^{(\infty)}|$ in dB versus computation time for AL-tridiag-inpaint as a function of balance parameter α.

4.6 Results

4.6.1 Simulated Data

4.6.1.1 Axial Slice Reconstruction

We also conducted additional experiments on an undersampled digital phantom to measure reconstruction error. We simulated noisy multi-coil data from a T_1-weighted 240 $\times$ 200 BrainWeb image with linear phase. We generated sensitivity maps for a 6-channel head coil array and generated noisy k-space data with SNR of 40. A Poisson-disk based sampling pattern [70] containing the central 16×16 phase encodes and with an overall undersampling factor of 6 was used for undersampling, as shown in Figure 4.2.

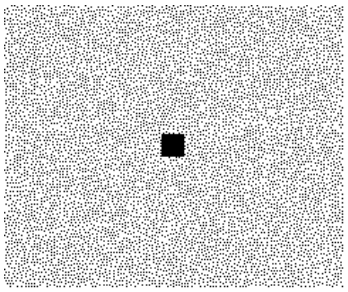


Figure 4.2: Poisson-disk-based undersampling pattern used for retrospective undersampling, with reduction factor of 6 and fully sampled central 16×16 phase-encodes.

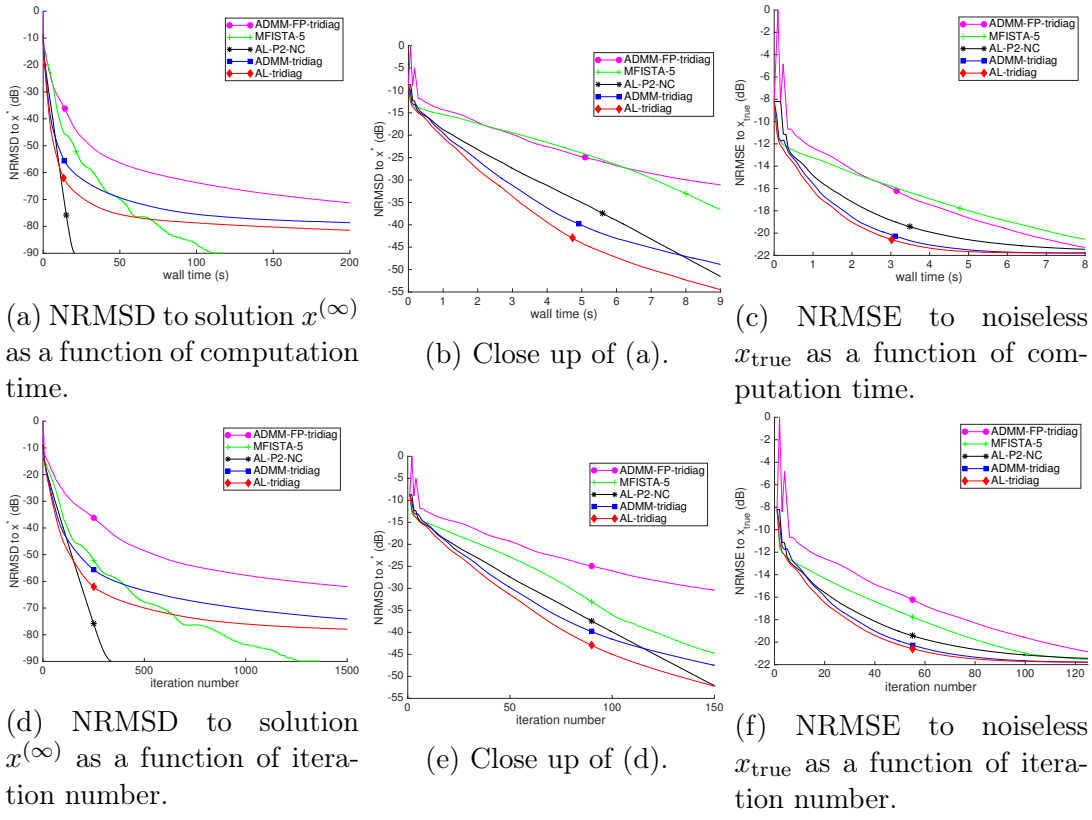
The central phase encodes were included to capture the rich information near the center of k-space. The Poisson disk sampling pattern reduces clustering of sample points in the outer regions of k-space. As with the *in vivo* experimental data, we chose to compare the proposed methods to AL-P2-NC [67] and MFISTA [74]. To

measure the convergence rate of these algorithms, we computed the normalized root mean squared distance (NRMSD) between a given iterate $x^{(n)}$ and the converged solution, $x^{(\infty)}$, in terms of decibels (4.80).

We generated the solution, $x^{(\infty)}$, by running MFISTA for 50000 outer iterations with 5 inner NCG iterations. We solved the inner iterative update for z in AL-P2-NC with preconditioned conjugate gradient using a circulant preconditioner and 1 inner iteration. We also computed the normalized root mean squared error (NRMSE) between a given iterate $x^{(n)}$ and x_{true}, the noiseless BrainWeb image used to generate the synthetic data. Convergence toward to the true image x_{true} provides useful context for a termination condition for these iterative algorithms.

As in the *in vivo* experiments, time spent computing the maximum eigenvalue of $\mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S}$ via power iteration, required for MFISTA, was not included in computation time. For these simulations, this computation took 60.4 seconds.

Figure 4.3 shows that the fully parallelized variant, ADMM-FP-tridiag has significant computational overhead due to the additional auxiliary variables updated in each iteration. In terms of NRMSD to the solution $x^{(\infty)}$, ADMM-tridiag and AL-tridiag outperform AL-P2-NC and other methods in the first hundred iterations, having already achieved -40 dB NRMSD or better. AL-P2-NC eventually overtakes the proposed methods around -50 dB difference to $x^{(\infty)}$. By comparing distance to the noiseless BrainWeb image used for x_{true}, AL-tridiag and ADMM-tridiag are very competitive with AL-P2-NC, reaching similar error levels at similar computation times and iteration numbers. The proposed algorithms show little improvement



(a) NRMSD to solution $x^{(\infty)}$ as a function of computation time.

(b) Close up of (a).

(c) NRMSE to noiseless x_{true} as a function of computation time.

(d) NRMSD to solution $x^{(\infty)}$ as a function of iteration number.

(e) Close up of (d).

(f) NRMSE to noiseless x_{true} as a function of iteration number.

Figure 4.3: Speed comparison of ADMM-tridiag, AL-tridiag, ADMM-FP-tridiag, MFISTA, and AL-P2-NC for simulated axial data. Top row shows speed as a function of computation time. Bottom row shows speed as a function of iteration. From left to right: NRMSD to $x^{(\infty)}$, a close-up of performance over the first 100 iterations, and NRMSE to x_{true}.

in NRMSE after 10 seconds of computation or roughly 120 iterations. Further progression of image estimates to $x^{(\infty)}$ do not translate to image quality improvements. In this simulation, ADMM-FP-tridiag compares more favorably in terms of NRMSE convergence than in the *in vivo* experiments.

The blur in the initial estimate is greatly reduced after 5000 iterations of ADMM-tridiag. The reconstructed image differs from the true, fully-sampled, noiseless image,

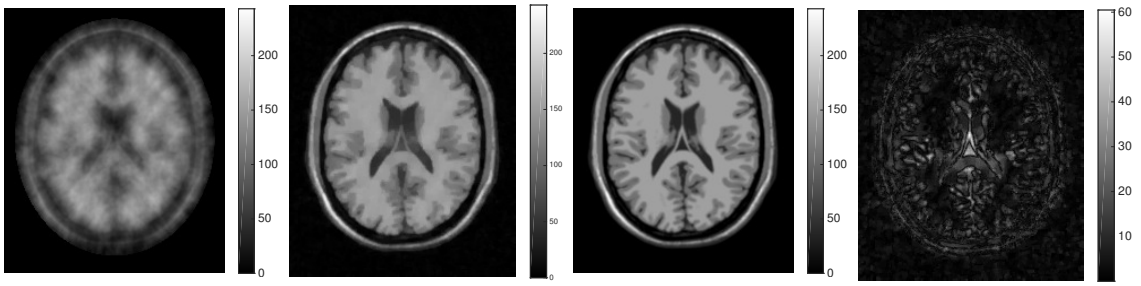


Figure 4.4: Left to right: (a) sum-of-squares of the zero-filled iFFT coil images, used as an initial estimate used for all algorithms; (b) ADMM-tridiag solution $\hat{x}$ after 5000 iterations; (c) x_{true} fully sampled, noiseless image; (d) difference between $\hat{x}$ and x_{true} after 5000 iterations.

x_{true}, primarily at anatomical edges and centrally located ventricles, as shown in Figure 4.4. The reconstructed image at 5000 iterations has an NRMSE of -21.6 dB.

4.6.1.2 Sagittal Slice Reconstruction

4.6.2 *in vivo* Experiment Setup

Following [67], we used a 3D in-vivo volunteer data set acquired from a GE 3T scanner ($T_R = 25$ ms, $T_E = 5.172$ ms, voxel size $= 1{\times}1.35{\times}1$ mm^3) with an 8-channel head coil. A corresponding body coil dataset was also acquired for sensitivity map estimation and image quality comparison. The fully-sampled data was 256×144 with 128 samples in the read-out direction along z. We performed the proposed reconstruction algorithms for two 2D axial slices from retrospectively undersampled data. To promote FFT efficiency, we resampled the data to correspond to an image size of 256×128. Fig. 4.5 shows the Poisson-disk-based undersampling pattern

(reduction factor 6) in the k_x-k_y phase-encode plane that included the central 16 $\times$ 16 phase-encodes, pictured in Figure 4.5. This sampling corresponds to one slice of a 3D acquisition with frequency encoding in k_z.

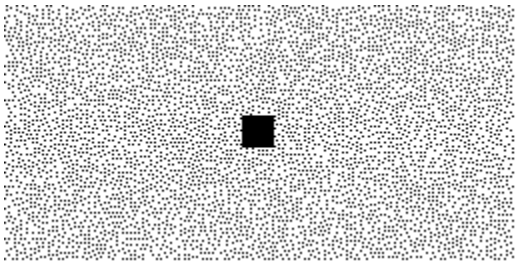
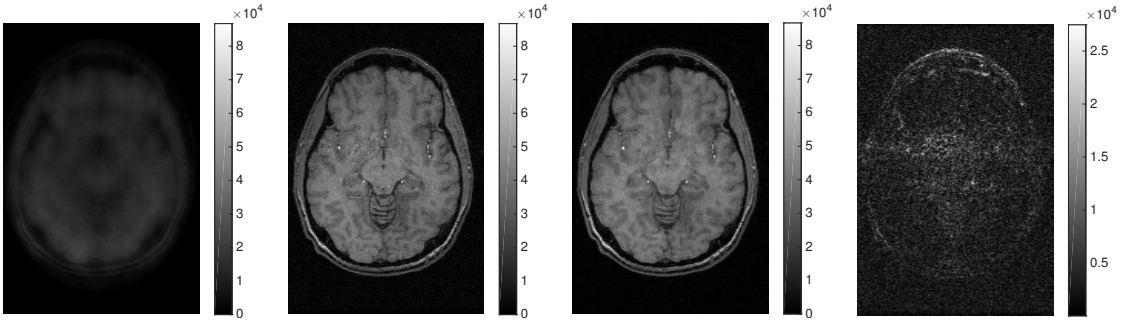


Figure 4.5: Retrospective Poisson-disk-based undersampling pattern used for *in vivo* experiments, with reduction factor of 6 and fully sampled central 16×16 phase-encodes.

We used the central 16 $\times$ 16 phase-encodes to generate low resolution images that were then used with the body coil image to estimate smooth sensitivity maps [18], shown in Figure 4.10 of the Supplement. The sensitivity values in the air regions were truncated in magnitude to control the maximum value in the sum-of-squares of the sensitivity maps and aid in tuning AL penalty parameters.

Computation was done on a Genuine Intel Xeon CPU E5-2680 with a 2.8 GHz 20 core machine with hyper-threading. The operating system was 64-bit Red Hat 6.7 running gcc version 4.4.7. All algorithms were implemented in Matlab version 8.6 using the image reconstruction toolbox [71], and all algorithms operated on single precision data. We performed parallelization of variable updates in (4.27) and (4.28) with a Pthreaded MEX function. The Pthreaded MEX function performed blockwise Gaussian elimination in parallel across each of the tridiagonal blocks of $\mathbf{H}_3$ and $\mathbf{H}_{\mathrm{x}}$. We allocated 20 Pthreads for these operations.



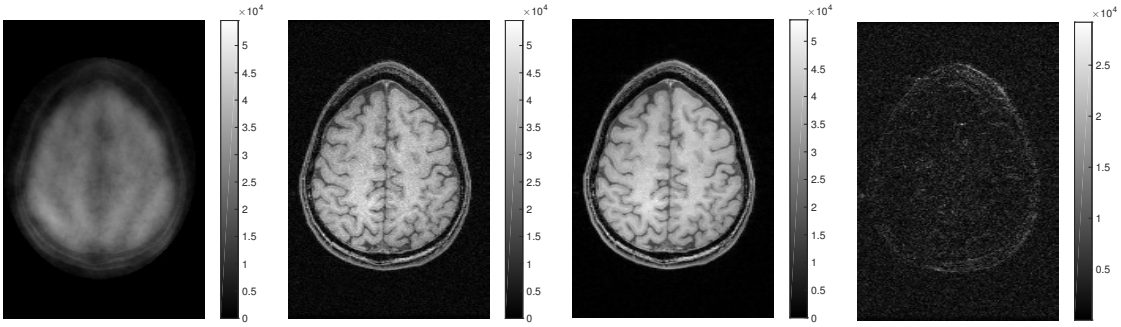
(a) iFFT (b) body coil (c) $x^{(\infty)}$ (d) difference

Figure 4.6: Axial slice 38 of volunteer data (a) square root of sum-of-squares of the zero-filled iFFT coil images, used as an initial estimate for all algorithms; (b) separately acquired body coil image; (c) $x^{(\infty)}$ calculated by MFISTA; (d) difference between body coil and $x^{(\infty)}$.

The initial estimate for axial slice 38 was the square-root of the sum-of-squares of the zero-filled iFFT coil images, shown in Figure 4.6. Figure 4.6 also shows the qualitative similarity between the separately acquired body coil image and the MFISTA solution of (4.1).

We repeated the experiment with axial slice 90 from the same *in vivo* dataset and using the same sampling pattern. The sensitivity maps estimated for this axial slice are shown in Figure 4.10 of the supplement. Figure 4.7 shows the initial sum-of-squares estimate, the separately acquired body coil image, and MFISTA solution, and the difference between the body coil and MFISTA solution. For both slice 38 and 90, the converged MFISTA solution shows lower noise than the body coil image.

For both slice 38 and 90, both proposed algorithms reached the solution $x^{(\infty)}$ (to within machine precision), shown in Figure 4.6c and 4.7c. For brevity, Figures 4.6c and 4.7c do not include $x^{(\infty)}$ for AL-tridiag or ADMM-tridiag, because they are



(a) iFFT (b) body coil (c) $x^{(\infty)}$ (d) difference

Figure 4.7: Axial slice 90 of volunteer data (a) square root of sum-of-squares of the zero-filled iFFT coil images, used as an initial estimate for all algorithms; (b) separately acquired body coil image; (c) $x^{(\infty)}$ calculated by MFISTA; (d) difference between body coil and $x^{(\infty)}$.

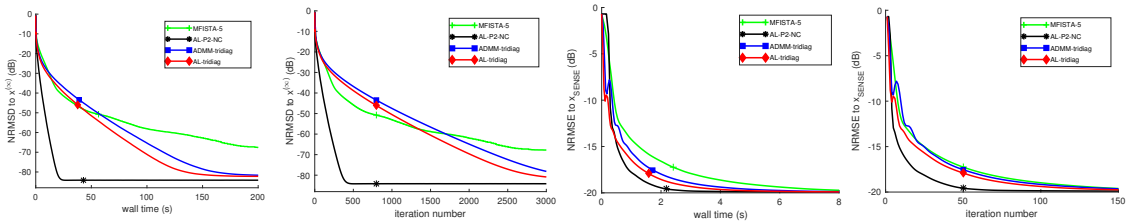
visually indistinguishable from the MFISTA solution. The solutions $x^{(\infty)}$ were also visually similar to the fully sampled SENSE reconstruction without regularization, $\hat{x}_{\text{SENSE}}$. Image comparisons are presented in the supplement.

4.6.2.1 Computation Speed Results for *in vivo* MRI data

We quantified the convergence rate of these algorithms using the normalized root mean squared distance (NRMSD) between a given iterate $x^{(n)}$ and the converged solution, $x^{(\infty)}$, in decibels:

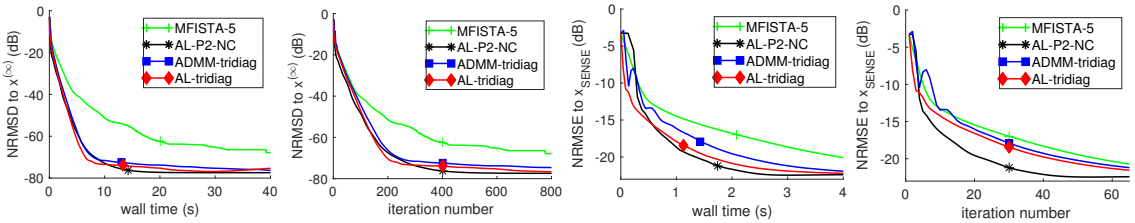
$$\text{NRMSD} \triangleq 20 \log_{10} \left(\frac{\left\| x^{(n)} - x^{(\infty)} \right\|_2}{\left\| x^{(\infty)} \right\|_2} \right) \tag{4.80}$$

To generate the solution, $x^{(\infty)}$, we ran MFISTA for 50000 outer iterations with 5 inner NCG iterations. We also calculated the normalized root mean squared error



(a) (b) (c) (d)

Figure 4.8: Axial slice 38: (a, b) NRMSD comparison of AL-tridiag, ADMM-tridiag, MFISTA, and AL-P2-NC to $x^{(\infty)}$; (c, d) NRMSE comparison of AL-tridiag, ADMM-tridiag, MFISTA, and AL-P2-NC to fully sampled SENSE reconstruction.



(a) (b) (c) (d)

Figure 4.9: Axial slice 90: (a, b) NRMSD comparison of AL-tridiag, ADMM-tridiag, MFISTA, and AL-P2-NC to $x^{(\infty)}$; (c, d) NRMSE comparison of AL-tridiag, ADMM-tridiag, MFISTA, and AL-P2-NC to fully sampled SENSE reconstruction.

(NRMSE) between a given iterate $x^{(n)}$ and the fully-sampled SENSE reconstruction, $\hat{x}_{\text{SENSE}}$, computed without any regularization.

For computation speed, we measured the wall time of each algorithm. For the AL and ADMM algorithms, we omitted time spent tuning AL penalty parameters and compiling Pthreaded MEX functions. The MFISTA method requires precomputation of the maximum eigenvalue of $\mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S}$ via power iteration, which took approximately 4.3 seconds for *in vivo* experiments, whereas all the VS methods avoid this overhead. Computation time excludes time spent computing this maximum eigenvalue. For all algorithms, we omitted the time spent computing the initial sum-of-squares

estimate.

As demonstrated in Figure 4.8, MFISTA is costlier per iteration than the proposed methods and AL-P2-NC. AL-tridiag converges slightly faster than ADMM-tridiag due to having fewer auxiliary and dual variables to update. AL-P2-NC converged the fastest for this slice.

Figure 4.9 shows that axial slice 90 presented a change in relative computation speed toward $x^{(\infty)}$: AL-tridiag and ADMM-tridiag converge faster than AL-P2-NC down to -65 dB NRMSD and up to 500 iterations. In the simulation results shown in the supplement, ADMM-tridiag also converged faster than AL-P2-NC in the early iterations. Overall, the speed of ADMM-tridiag is generally comparable to that of AL-P2-NC. For all AL/ADMM methods, the convergence rate depends on parameter selection; the heuristics used for parameter design in [67] may perform better under some conditions than others. One possible reason for the difference in relative convergence speeds in experiments for slice 38 and slice 90 is the smaller anatomical support in slice 90. Due to the head coil geometry, the smaller head circumference at slice 90 results in a lower signals from the surface coils, which may present a more difficult reconstruction problem.

For these axial slices, we also examined the NRMSE between iterates $x^{(n)}$ and the fully sampled SENSE reconstruction $\hat{x}_{\text{SENSE}}$ without regularization. AL-tridiag and ADMM-tridiag reach the minimum NRMSE after similar amounts of computation as AL-P2-NC, approximately 4 seconds and 60 iterations. By this metric, MFISTA performs slightly worse as a function of computation time and iterations, and would

be far worse when one accounts for the overhead of running the power iteration to find the maximum eigenvalue of $\mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S}$.

4.6.2.2 *in vivo* Experiment Results for ADMM-FP-tridiag

For the same 3D in-vivo volunteer data set used in [9], we tested the convergence speed of ADMM-FP-tridiag. We applied ADMM-FP-tridiag to the same undersampled data from axial slices 38 and 90 using the same estimated smooth sensitivity maps, shown in Figure 4.10 and undersampling patterns, shown in Figure 4.5 of [9]. We initialized ADMM-FP-tridiag with the same zero-filled iFFT image as the other algorithms, shown in Figures 4.6 and 4.7 of [9].

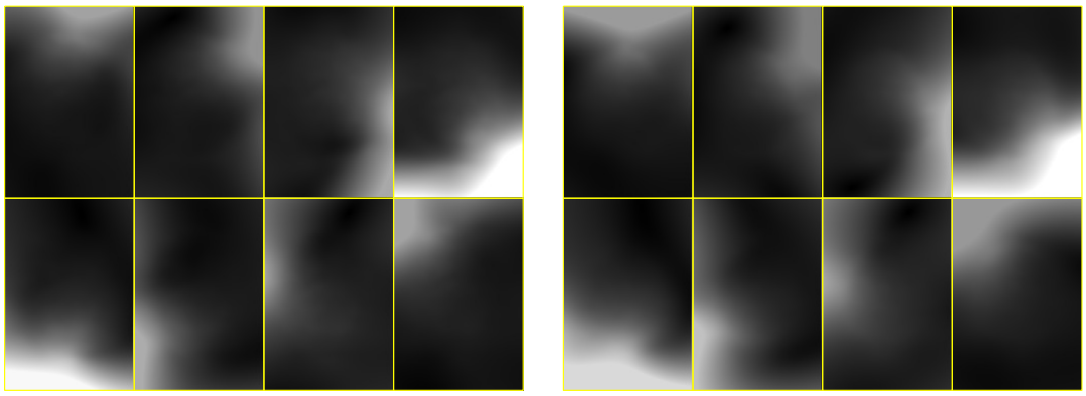


Figure 4.10: Magnitudes of the sensitivity maps estimated using central 16 $\times$16 phase encodes for axial slice 38 (left) and axial slice 90 (right).

The proposed ADMM-FP-tridiag algorithm was implemented with 20 Pthreads allocated to the tridiagonal updates in (4.42) and (4.43), but no parallelization was implemented across the blocks of $\underline{u}$ or $\underline{v}$. Here we present the same results as shown in Figures 4.8 and 4.9 in [9], with additional convergence speed measurements for

ADMM-FP-tridiag. As was the case for AL-tridiag and ADMM-tridiag, time spent tuning AL penalty parameters for ADMM-FP-tridiag was not included in computation time.

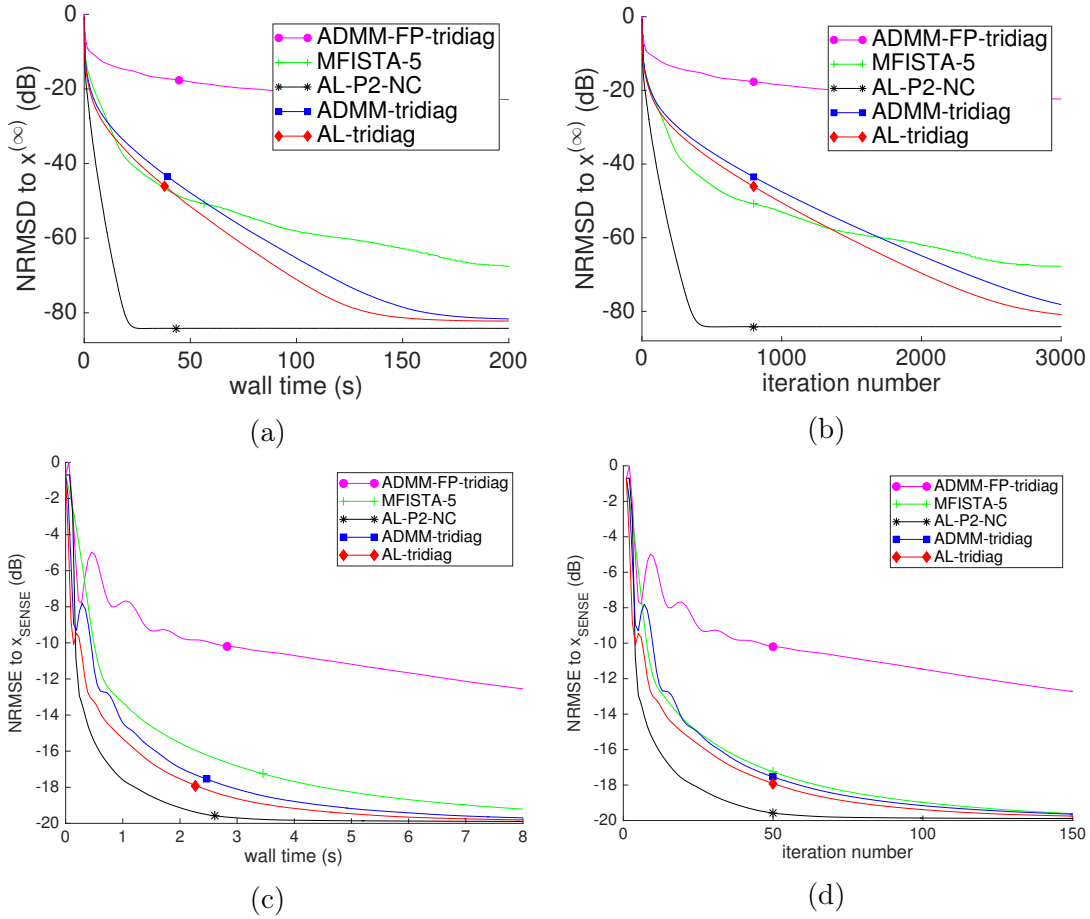


Figure 4.11: Axial slice 38: (a, b) NRMSD comparison of proposed algorithms, MFISTA, and AL-P2-NC to $x^{(\infty)}$; (c, d) NRMSE comparison of proposed algorithms, MFISTA, and AL-P2-NC to fully sampled SENSE reconstruction..

The fully parallelizable proposed method, ADMM-FP-tridiag, was not implemented with parallelization across the blocks of $\underline{u}$, $\underline{v}_{\mathrm{FP}}$, or $\underline{\eta}_{\mathrm{FP}}$. For this reason, as well as the substantially larger number of auxiliary and dual variables, ADMM-FP-tridiag

performs the slowest of the algorithms in this comparison. It is likely that finely tuning the AL penalty parameters of ADMM-FP-tridiag would result in improved speed, but these speed gains likely will not offset the additional bookkeeping required for $\underline{v}_{\text{FP}}$ and $\underline{\eta}_{\text{FP}}$. The use of additional auxiliary variables in ADMM-FP-tridiag may also require more iterations for information to propagate across block elements of $\underline{u}$ and $\underline{v}_{\text{FP}}$.

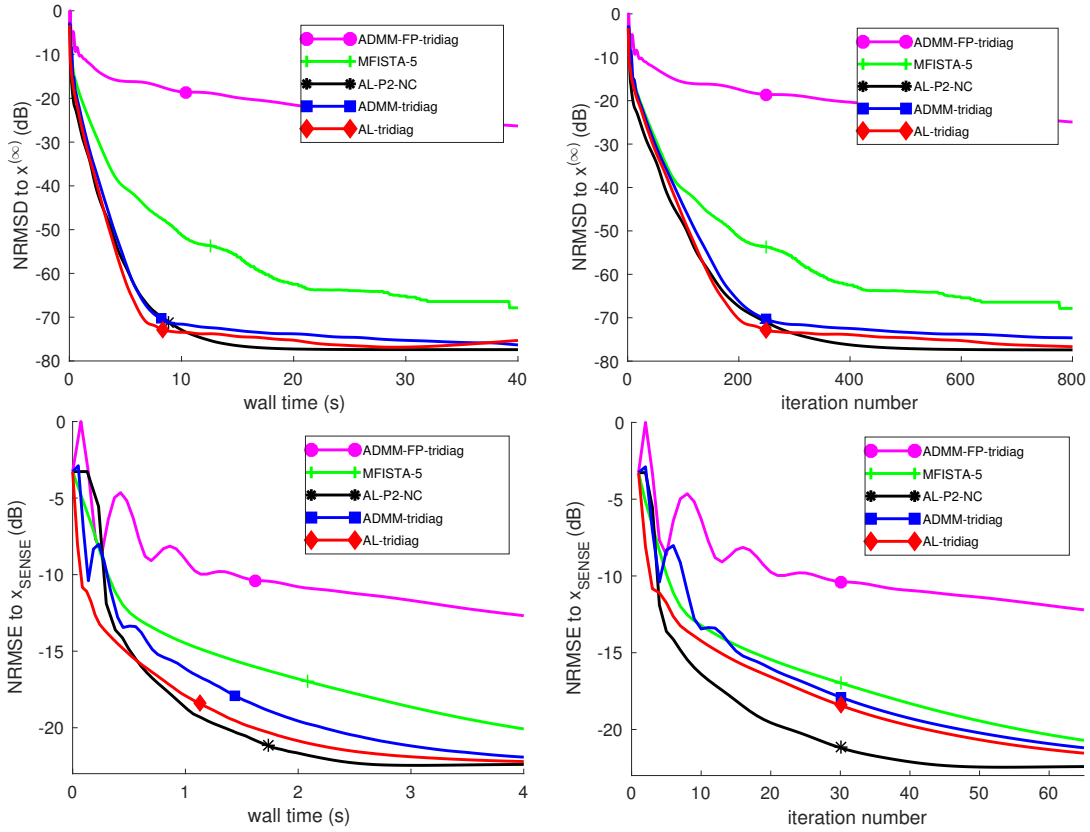


Figure 4.12: Axial slice 90: (a,b) NRMSD comparison of proposed algorithms, MFISTA, and AL-P2-NC to $x^{(\infty)}$; (c,d) NRMSE comparison of proposed algorithms, MFISTA, and AL-P2-NC to fully sampled SENSE reconstruction.

For axial slice 90, ADMM-FP-tridiag remains the slowest method. However, due

to enforcing equivalence with ADMM, the solution to ADMM-FP-tridiag is within machine precision of x^*, despite the much longer convergence time. For this reason, we omit images of the solution to ADMM-FP-tridiag for slices 38 and 90.

4.6.2.3 Image Quality Comparison for *in vivo* Data

In this section, we compare the image quality of the solution to (4.1) to the body coil image and the fully sampled SENSE reconstruction $\hat{x}_{\text{SENSE}}$. For both axial slices 38 and 90, all algorithms in the comparison eventually reach the same solution, $x^{(\infty)}$. The reconstructed image from ADMM-tridiag at 5000 iterations, $\hat{x}$ is visually similar to the bodycoil, shown in Figures 4.13 and 4.14. ADMM-tridiag is able to reconstruct many of the anatomical details missing in the initial zero-filled iFFT images. The estimated images for AL-tridiag at 5000 iterations is very similar to that of ADMM-tridiag and are not pictured.

Figures 4.13 and 4.14 also show the absolute difference of the body coil and reconstructed image, as well as the fully sampled SENSE reconstruction without regularization, $\hat{x}_{\text{SENSE}}$ and the difference between the reconstructed image and SENSE image. The difference image between $\hat{x}$ and $\hat{x}_{\text{SENSE}}$ show that the regions with the highest error are those in which the g-factor is low due to the head coil geometry. The reconstructed image more closely resembles the fully sampled SENSE image than the relatively noisy body coil image.

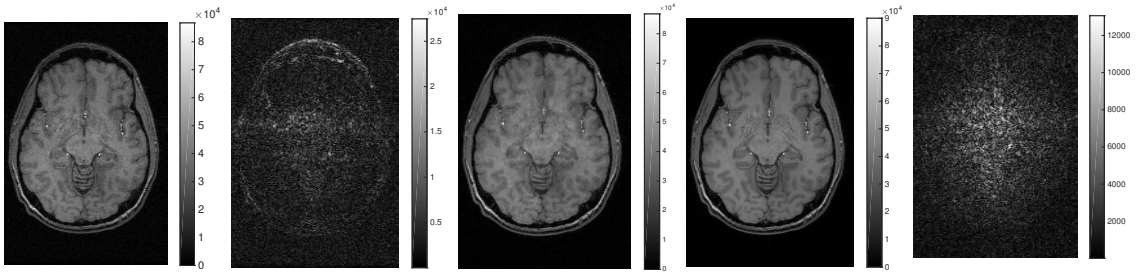


Figure 4.13: *in vivo* experiment for axial slice 38. Left to right: (a) body coil image; (b) differences between body coil and ADMM-tridiag; (c) ADMM-tridiag reconstruction; (d) fully sampled SENSE reconstruction; (e) differences between fully sampled SENSE reconstruction and ADMM-tridiag.

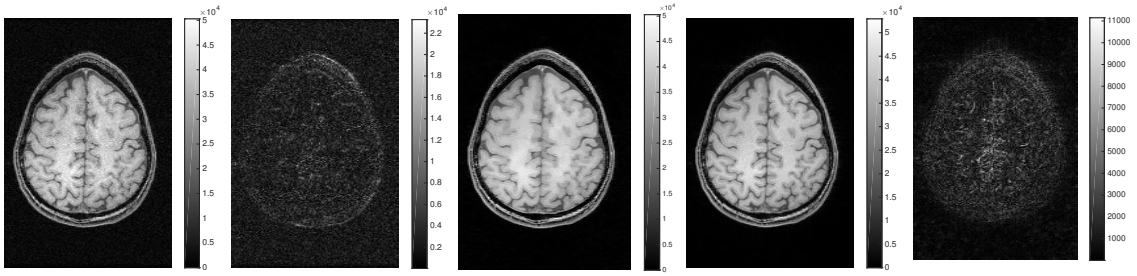


Figure 4.14: *in vivo* experiment for axial slice 90. Left to right: (a) body coil image; (b) differences between body coil and ADMM-tridiag; (c) ADMM-tridiag reconstruction; (d) fully sampled SENSE reconstruction; (e) differences between fully sampled SENSE reconstruction and ADMM-tridiag.

4.6.3 Image Inpainting

This section illustrates the benefits of non-periodic boundary conditions and the proposed variable splitting scheme for an inpainting problem. Unlike medical images that often have air at one or more boundaries, natural scenes typically contain useful, distinct information at the boundaries.

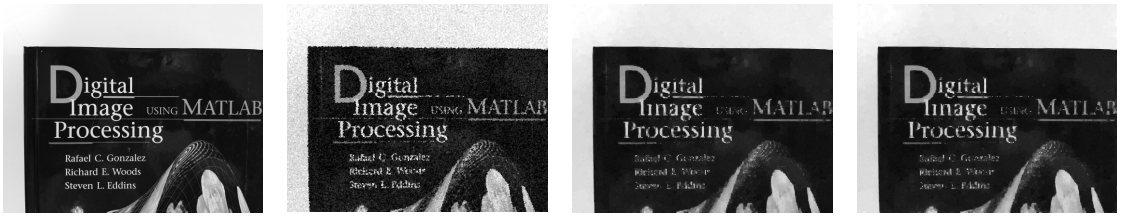
To test the effect of AL-tridiag-inpaint, we took a 432×540 digital photograph using a Samsung SM-G930V camera, randomly discarded 75% of the pixels, and added

white Gaussian noise corresponding to 20 dB SNR to the remaining pixels. We used 2D nearest neighbor interpolation to initialize the inpainting estimate, pictured in Fig. 4.15.

To demonstrate the ease with which the proposed formulation accommodates orthonormal wavelets, we estimated the inpainted image using AL-tridiag-inpaint with the modified regularization operators in (4.34). We selected regularization parameters λ_1 and λ_2 for good image reconstruction quality, and set $\alpha_w = 1$ to limit additional memory usage. We also applied the AL-P2 variable splitting scheme to the inpainting problem, using finite-differences with periodic boundary conditions and Haar wavelets. We show the inpainting images estimated using non-periodic boundary conditions in Fig. 4.15. All images are displayed on the same grayscale axis as the original image, unless otherwise noted.

Figure 4.16 shows the error between the inpainting estimates using non-periodic vs. periodic boundary conditions. The use of periodic boundary conditions results in higher error near the boundaries of the image.

We conducted the inpainting computational speed experiments on the machine described in Section 4.6.2, and we compute wall time using the same rules as in Section 4.4.4, excluding time spent tuning AL parameters. We measure NRMSD to the MFISTA solution, $x^{(\infty)}$, as a function of wall time. As in the CS-SENSE-MRI experiments, we compare AL-tridiag-inpaint to a variant of AL-P2 to better understand the effect of the proposed variable splitting scheme. We apply the AL-P2 variable splitting scheme to the inpainting problem with non-periodic boundary conditions



(a) x_{true} (b) $x^{(0)}$ (c) Non-Circ. (d) Circ.

Figure 4.15: Inpainting images (a) true image; (b) nearest neighbor interpolation of noisy, partial data (SNR = 20 dB, 75% discarded); (c) inpainting estimate using finite differences with non-periodic boundary conditions and Haar wavelet regularization; (d) inpainting estimate using finite differences with periodic boundary conditions and Haar wavelet regularization.

(a) (b) (c) (d)

Figure 4.16: Absolute difference image between true image and inpainting reconstruction ($\times$ 10) (a) using finite differences with non-periodic boundary conditions and Haar wavelets (NRMSE = 0.153); (b) using finite differences with periodic boundary conditions and Haar wavelets (NRMSE = 0.155); (c) corner detail of (a); (d) corner detail of (b).

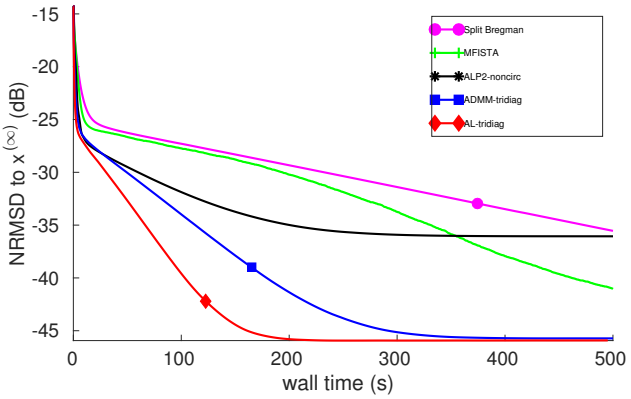


Figure 4.17: NRMSD to $x^{(\infty)}$ as a function of elapsed computational time for the proposed inpainting algorithms, AL-tridiag-inpaint and ADMM-tridiag-inpaint, compared with existing methods AL-P2-NC, Split Bregman, and MFISTA.

(4.57), and we call the resulting AL algorithm AL-P2-NC-inpaint. Similar to AL-P2-NC, it requires an inner iterative variable update due to the non-circulant Hessian. We solve this inner step using one iteration of preconditioned gradient descent with a circulant preconditioner.

Supposing that boundary artifacts are a secondary concern to computational speed, we also compare the speed of AL-tridiag-inpaint to AL-P2-inpaint. AL-P2-inpaint is distinct from AL-P2-NC-inpaint due to its cost function, which uses regularizers with periodic boundary conditions. AL-P2-inpaint is not handicapped by an inner iterative update, because the circulant Hessian can be diagonalized efficiently via FFTs. Though AL-tridiag-inpaint must complete two tridiagonal solver variable updates for each of AL-P2-inpaint's FFT-based variable updates, the $\mathcal{O}(n)$ runtime of the tridiagonal solver and the parallelized implementation result in comparably fast iterations for AL-tridiag-inpaint. The average computation time for each iteration was 0.0543 seconds for AL-P2-inpaint, 0.1193 seconds for AL-P2-NC-inpaint, 0.0583

seconds for AL-tridiag-inpaint, and 0.0745 seconds for ADMM-tridiag-inpaint. Both methods converge with less computation time than MFISTA and Split Bregman [35].

Figure 4.17 also demonstrates convergence benefits of the proposed variable splitting scheme. Both AL-tridiag-inpaint and ADMM-tridiag-inpaint were able to reach the same $x^{(\infty)}$ as MFISTA, unlike AL-P2-NC. AL-P2-NC reaches a different solution with the same objective value as the MFISTA $x^{(\infty)}$. An investigation into the convergence behavior of AL-P2-NC can be future work.

4.7 Discussion

The ADMM-tridiag algorithm provides a simple way to ensure convergence for variable splitting methods. By examining the constraint matrix and designing $\mathbf{B}$, we show equivalence between the variable splitting scheme in (4.13) and ADMM. The additional variable split and variable update led to parallelizability of two of the resulting variable updates. For applications as sensitive as medical diagnosis, an algorithm with convergence guarantees may be preferable to those having unknown convergence properties.

Unlike AL-P2 [67], the proposed algorithm, ADMM-tridiag, has a convergence guarantee and addresses non-periodic boundary conditions, while demonstrating comparable computational speed. Using heuristic parameter tuning based on condition numbers of variable update Hessians, we demonstrated that the speeds of AL-P2-NC and ADMM-tridiag are similar but can vary depending on experimental condi-

tions.

The proposed variable splitting scheme can be useful for a variety of image processing problems, because many natural scenes have non-zero values at the boundaries that do not relate periodically to opposite boundaries. As shown in Section 4.4.2, the proposed method AL-tridiag is readily adapted to image denoising and inpainting problems. Image deblurring is also a good candidate for this proposed variable splitting. Separation of horizontal and vertical differences has also been explored in image segmentation [87].

To fully benefit from the $\max(N_x, N_y)$ parallelizable tridiagonal updates of AL-tridiag and ADMM-tridiag, one should use a highly parallel computing platform1, such as a GPU.

The algorithms proposed in this work have several limitations. Though the proposed algorithms are designed to facilitate fast computation, the convergence speed is highly dependent on good penalty parameter choice. Though we present some useful heuristics for choosing the AL convergence parameters, the optimal procedure for designing these parameters is a difficult analysis problem and still unknown. (For some simpler ADMM methods, optimal parameter tuning has been analyzed [88].) Moreover, the complexity of convergence parameter design increases with the number of variable splits, and this work is built around an additional separation of horizontal and vertical differences into distinct auxiliary variables. The increased number of tuning parameters introduces another degree of freedom. Using non-scalar penalty parameter matrices $\mathbf{M}_3$ and $\mathbf{M}_4$ further increases the degrees of freedom compared

to the simple scalar choice used in most AL methods.

Though the proposed variable splitting scheme can be easily extended to 3D reconstruction problems, this would require introducing two additional variable splits, separating each of the three finite difference directions and introducing a second auxiliary variable proxy for x. Though the corresponding variable updates can be quickly computed with shrinkage and the parallelizable tridiagonal solver, this 3D variable splitting scheme could further complicate penalty parameter analysis.

Finally, the variable splitting scheme at the center of the proposed algorithms is applicable only for regularization with first-order finite differences. Though Section 4.3.5 shows that the formulation also accommodates orthonormal wavelet penalties combined with finite differences, this variable splitting scheme yields no benefit for other sparsity transforms, e.g., non-orthonormal wavelets and learned dictionaries.

4.8 Summary and Conclusion

This chapter proposed a variable splitting algorithm for SENSE MRI reconstruction, ADMM-tridiag. The proposed method offers convergence guarantees and efficient variable updates for non-periodic boundary conditions. ADMM-tridiag efficiently handles the non-periodic boundary conditions by separating the finite differences in the horizontal and vertical directions to create easily solvable and parallelizable tridiagonal problems. The method for inducing ADMM equivalence requires only one additional variable split and variable update.

We also presented a simpler variation of this algorithm: AL-tridiag. AL-tridiag was derived from the same variable splitting scheme as ADMM-tridiag, but has a simpler update procedure, resulting in a slight speed increase albeit without any convergence guarantees. We showed a simple relationship between AL-tridiag and ADMM-tridiag and compared their convergence speeds to that of AL-P2-NC and MFISTA. Convergence speed was evaluated in terms of distance to the solution of the proposed cost function (4.1) as well as to the fully sampled SENSE reconstruction. For retrospectively undersampled *in vivo* data, the proposed algorithms demonstrated comparable convergence speed and produced reconstructed images with good image quality. AL-tridiag was also applied to a noisy image inpainting problem, demonstrating faster convergence speed than AL-P2-NC, and improved image fidelity at the boundaries than AL-P2 with periodic boundary conditions.

All of the proposed algorithms require selection of penalty parameters. We use heuristics determined in [67] to select these AL penalty parameters, although we exploited a more general version of tuning parameters to enable computation of condition numbers for tridiagonal Hessians. Using methods that adapt the parameters as a function of iteration [89] might simplify and accelerate AL methods.

Chapter 5

Spline Temporal Basis for Improved Pharmacokinetic Parameter Estimation

5.1 Introduction

[1] Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI) is an application of MRI for visualizing perfusion in various organs. During the dynamic imaging, a series of k-space samples are acquired in succession to characterize MR changes in tissue over time. DCE-MRI is a useful tool because of its ability to depict both physiology and morphology. It is routinely used for assessing targeted radiation therapy procedures.

For breast DCE-MRI studies, a T_1-shortening contrast agent such as gadolinium diethylenetriaminepentaacetic acid (Gd-DTPA) is injected into the patient, passing from major blood vessels into the blood plasma and into organ tissue. Measuring the contrast agent enhancement curves in the lesions allows for classifying malignant and

[1] This chapter is based on [10].

benign tumors, which are distinguished by different levels of angiogenesis, the growth of new blood vessels. In particular, we wish to estimate several pharmacokinetic parameters: the volume transfer constant, K^{trans}, the flux rate constant, k_{ep}, and the volume of extravascular extracellular space per unit volume of tissue, v_{e}. These parameters describe the relationships between contrast agent concentrations in blood plasma and breast tissue, as shown in Fig. 5.1 and are interrelated: $k_{\text{ep}} = K^{\text{trans}}/v_{\text{e}}$. The primary time series of interest is the amount of contrast agent in the breast

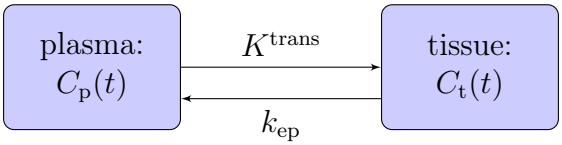


Figure 5.1: Two-compartment model for breast pharmacokinetics.

tissue over time, $C_{\text{t}}(t)$. This quantity is related to the contrast agent concentration in the blood plasma near the breast, $C_{\text{p}}(t)$. K^{trans}, k_{ep} , and v_{e} describe how $C_{\text{p}}(t)$ and $C_{\text{t}}(t)$ are related.

To estimate these pharmacokinetic parameters accurately, we desire high resolution images to delineate tumors from other tissues, as well as highly detailed time series to accurately estimate K^{trans} and k_{ep}. However, dynamic MRI presents a tradeoff between spatial and temporal resolution because of the sequential nature of k-space sampling. To overcome this tradeoff, we turn to MRI acceleration methods.

Numerous studies have shown that applying compressed sensing methods can compensate for randomly undersampling in k-space [26]. The power of compressed sensing is amplified for dynamic MRI data because dynamic images can be also be spar-

sified in time. Compressed sensing ideas can be combined with parallel imaging techniques, such as SENSitivity Encoded (SENSE) MRI [21]. Parallel imaging consists of simultaneously collecting spatially-encoded k-space data from multiple receiving coils.

In this work, we seek to further improve the condition of the inverse problem by restricting the dynamic image to a predefined, easily computable, spatio-temporal subspace of B-splines. This has the twofold goal of easing computation by reducing the number of unknown parameters and improving accuracy of the estimated enhancement curves by preventing spurious highly changing temporal components. Temporal spline functions have been used for dynamic MRI image reconstruction problems [90], and it has been investigated for PET [91].

5.2 Problem Formulation

Intuitively, injected contrast agent perfuses through the body in a slow and steady manner. This expectation of slowly-varying contrast agent concentrations is captured in the two-compartment model in Fig. 5.1. The contrast agent concentration of interest, $C_{\mathrm{t}}(t)$, is modeled as a convolution of $C_{\mathrm{p}}(t)$ with $K^{\mathrm{trans}}e^{-k_{\mathrm{ep}}t}$ to represent the flow back and forth from the blood plasma and breast tissue. These standard pharmacokinetic modeling equations further reinforce the idea that $C_{\mathrm{t}}(t)$ should be smooth and slowly varying.

Temporal regularization is one method for encouraging smooth and slowly varying temporal dynamics. This work uses this assumption more explicitly by enforcing

that the dynamic object, traditionally characterized by N_p pixels over N_f frames, be expressed with fewer than $N_p N_f$ parameters. Rather than representing an arbitrary time series, we choose to restrict the object to having time series representable by wide, overlapping, smoothly varying temporal basis functions.

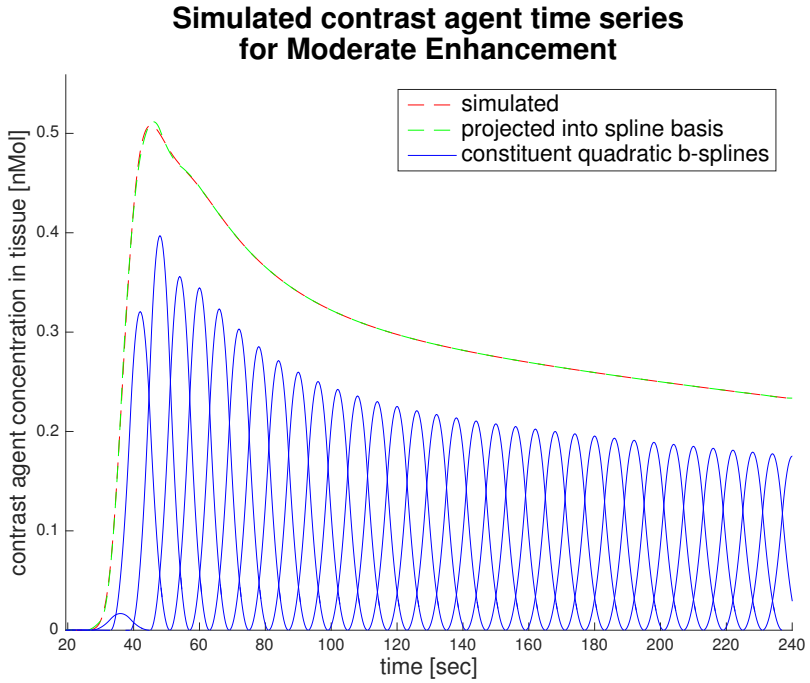


Figure 5.2: The red curve is a simulated tissue contrast concentration over time, using parameters from [2]. The blue curves are the quadratic B-splines whose sum best approximates the tissue contrast concentration curve. The green curve shows the sum of the B-splines, the approximation of the true time series.

5.2.1 Object Model

The object we wish to estimate is $f(r,t)$, the transverse magnetization at pixel r (including the effect of T_2^*) at time t near echo time. For ease of notation, this section will use one-dimensional notation for spatial location r, but this model can

be expanded to three-dimensional volumes. The most common object model in dynamic imaging studies implicitly assumes a non-overlapping rectangular temporal basis function. That is to say, the object is described over every discrete time point, implying that the measurement at time t_i is affected only by the object parameters associated with time t_i (though this is often balanced with a temporal regularizer to introduce temporal correlations). It is also common to assume a spatial rectangular basis function. Together, a rectangular spatial basis and a rectangular temporal basis yield a tensor product object model as follows:

$$f(r,t) = \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} \sum_{j_t=0}^{N_t-1} f_{j_r,j_t} \text{rect}(r - j_r\Delta_r)\text{rect}(t - j_t\Delta_t) \tag{5.1}$$

Pixel spacing is denoted by Δ_r, and frame duration is Δ_t. Discretizing $f(r,t)$ appropriately will result in the samples coinciding with the unknown f_{j_r,j_t} coefficients. Estimating these coefficients (often organized in a Casorati matrix) describes the unknown spatiotemporal object via Equation (5.1).

In this work, we consider a more flexible temporal basis, in which the basis functions overlap each other and are smoothly varying. In particular, we focus on splines for their finite support and high degree of smoothness. Assuming that the dynamic object lies reasonably close to the subspace spanned by the particular spatial basis and temporal splines, the goal is to express the dynamic object more succinctly with fewer spline coefficients. For now, we consider a general spatial basis, $b_r(r)$.

A generalized version of the object model in Equation (5.1) is:

$$f(r,t) = \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} \sum_{j_t=0}^{N_t-1} f_{j_r,j_t} b_r(r - j_r\Delta_r) b(t - j_t\Delta_t). \tag{5.2}$$

To make the problem more tractable computationally, we choose a temporal basis $b(t)$ that is constant over small intervals of length MT_R, an integer number of readouts. This duration can be considered a "subframe", in that it may be shorter in length than a typical MRI frame. Our goal is not to estimate the object at all subframes, just the image coefficients that coincide with particular subframes. The resulting object model is:

$$f(r,t) \approx \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} \sum_{j_t=0}^{N_t-1} f_{j_r,j_t} b_r(r - j_r\Delta_r) b\left(MT_R \left\lfloor \frac{t - j_t\Delta_t}{MT_R} \right\rfloor\right). \tag{5.3}$$

5.2.2 System Model

For parallel MRI, the signal received from the lth coil at time t is a function of the coil's sensitivity map, $C_l(r)$, the T_2^* value at pixel r, and the k-space trajectory $k_r(t)$ as follows:

$$s_l(t) = \int C_l(r) f(r,t) e^{-t/T_2^*(r)} e^{-i2\pi(rk_r(t))} dr, \qquad l = 1, \ldots, N_c. \tag{5.4}$$

Hereafter, we ignore the effect of T_2^* decay because readout duration is small com-

pared to T_E, so this term will not change much over the course of a readout. (For an example T_2^* value of 36 ms [92], readout of 1 ms, and T_E of 3 ms, the decay from e^{-t/T_2^*} would be approximately 3%).

By assuming a rectangular spatial basis for the sensitivity maps with coefficients c_{l,j_r} and substituting the proposed object model Eqn.(5.3), we can rewrite the signal Equation in terms of object coefficients f_{j_r,j_t}:

$$\begin{aligned} s_l(t) &\approx \sum_{j_t=0}^{N_t-1} b\left(MT_R \left\lfloor \frac{t-j_t\Delta_t}{MT_R} \right\rfloor\right) \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} c_{l,j_r} f_{j_r,j_t} \int b_r(r-j_r\Delta_r) e^{-i2\pi(r\cdot k_r(t))} dr \\ &\approx \sum_{j_t=0}^{N_t-1} b\left(MT_R \left\lfloor \frac{t-j_t\Delta_t}{MT_R} \right\rfloor\right) \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} a_{l,j_r}(t) f_{j_r,j_t}, \end{aligned} \tag{5.5}$$

where $a_{l,j_r}(t) \triangleq c_{l,j_r} e^{-i2\pi(j_r\Delta_r\cdot k_r(t))}$for choice of $b_r(r) = \delta(r)$.

We define the sequence $b[n]$ as the samples of the temporal spline basis function $b(t)$ of support W with a sampling period of $MT_R = \Delta_f$,

$$b[n] \triangleq b(nMT_R), \text{ for } n = -\left\lceil \frac{W}{2\Delta_f} \right\rceil, \ldots, \left\lceil \frac{W}{2\Delta_f} \right\rceil, \tag{5.6}$$

with Q samples between the centers of each spline ($\Delta_t = Q\Delta_f$), as shown in Fig. 5.3.

Using the samples of the temporal basis function $b(t)$, we can describe the signal at

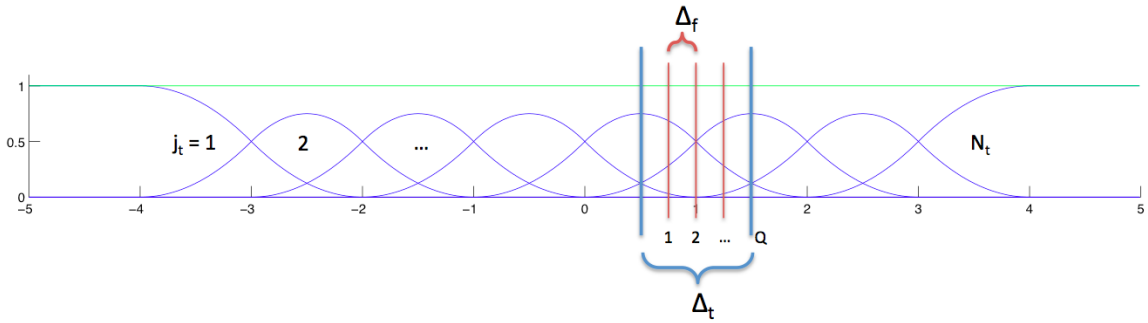


Figure 5.3: Diagram of temporal spline basis spacing parameters.

discrete time points $\{t_i\}_{i=1}^{MN_f}$ as

$$s_l(t_i) \approx s_{l,i} \triangleq \sum_{j_t=0}^{N_t-1} b_i[j_t] \sum_{j_r=-\frac{N_r}{2}}^{\frac{N_r}{2}-1} a_{l,i,j_r} f_{j_r,j_t}. \tag{5.7}$$

We define a_{l,i,j_r} and $b_i[j_t]$ as follows:

$$a_{l,i,j_r} \triangleq a_{l,j_r}(t_i) = c_{l,j_r} e^{-i2\pi j_r \Delta_r k_r(t_i)} \tag{5.8}$$

$$b_i[j_t] \triangleq b\left(MT_R \left\lfloor \frac{t_i - j_t\Delta_t}{MT_R} \right\rfloor\right) = b[n_i - j_t Q] \text{ for } n_i = \frac{t_i}{MT_R}. \tag{5.9}$$

By applying the sifting property in reverse and switching the order of summation, we can expand the signal approximation in Equation (5.7) into a useful formulation:

$$s_l(t_i) = \sum_{k=0}^{N_r-1} \sum_{n=0}^{N_f-1} \delta[k-k_i, n-n_i] \sum_{j_t=0}^{N_t-1} b[n-j_tQ] \sum_{j_r=0}^{N_r-1} \tilde{\mathbf{F}}_{k,j_r} c_{l,j_r} f_{j_r,j_t} \text{ with } k_i \triangleq k_r(t_i). \tag{5.10}$$

$\tilde{\mathbf{F}}_{k,j_r} \triangleq e^{-i2\pi k j_r/N_r}$ represents the contribution of the j_rth pixel to the kth k-space location, and $\tilde{\mathbf{F}} \in \mathbb{C}^{N_r \times N_r}$. In this chapter, we restrict our analysis to Cartesian sampling, so this Fourier transform can be performed with FFTs. To yield a useful and concise formulation, we arrange the other terms in Equation (5.10) into matrix form.

We define $\mathbf{S}_l \in \mathbb{C}^{N_r \times N_r}$ to be a diagonal matrix, where the ith diagonal element is the sensitivity of the lth coil for the ith pixel, $c_{l,i}$.

The operator $\tilde{\mathbf{T}} \in \mathbb{R}^{N_r Q N_t \times N_r N_t}$, so named for inTerpolation, is a Kronecker product $\tilde{\mathbf{T}} = \hat{\mathbf{T}} \otimes \mathbf{I}_{N_r}$, where $\hat{\mathbf{T}} \in \mathbb{R}^{Q N_t \times N_t}$, which has elements $\hat{\mathbf{T}}_{i,j} = b[n_i - jQ]$. $\tilde{\mathbf{T}}$ effectively takes a weighted linear combination of k-space for different temporal basis frames to interpolate the values for time samples that lie in between the centers of the temporal basis functions. We efficiently implemented this via upsampling and convolution with $b[n]$ as the filter.

Finally, $\tilde{\mathbf{P}} \in \{0,1\}^{N_s \times N_r Q N_t}$ is a sampling matrix that selects the appropriate k-space location and interpolation index, with elements $\tilde{\mathbf{P}}_{i,j} = \delta[l_i - k_j, m_i - n_j]$. l_i denotes the k-space location and m_i the subframe index associated with the ith sample, for $i = 0, \ldots, N_s - 1$. Pairs $\{k_j, n_j\}$ denote the k-space location and subframe index of the jth output element of $\tilde{\mathbf{T}}$, for $j = 0, \ldots, N_r Q N_t - 1$.

Vectorizing the signals of the lth coil as $\mathbf{s}_l \in \mathbb{C}^{N_s \times 1}$ and the object coefficients as $\mathbf{f} \in \mathbb{R}^{N_r N_t \times 1}$ leads to the following matrix-vector expression for the lth coil:

$$\mathbf{s}_l = \tilde{\mathbf{P}} \tilde{\mathbf{T}} \left(\mathbf{I}_{N_t} \otimes \tilde{\mathbf{F}} \right) \left(\mathbf{I}_{N_t} \otimes \mathbf{S}_l \right) \mathbf{f}, \qquad l = 1, \ldots, N_c. \tag{5.11}$$

Stacking these vectors and matrices over each coil yields the following system matrix for multi-coil SENSE MRI:

$$\mathbf{s} = \underbrace{\left(\mathbf{I}_{N_c} \otimes \tilde{\mathbf{P}}\right)}_{\mathbf{P}} \underbrace{\left(\mathbf{I}_{N_c} \otimes \tilde{\mathbf{T}}\right)}_{\mathbf{T}} \underbrace{\left(\mathbf{I}_{N_c N_t} \otimes \tilde{\mathbf{F}}\right)}_{\mathbf{F}} \underbrace{\left(\mathbf{I}_{N_t} \otimes \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_{N_c} \end{bmatrix}\right)}_{\mathbf{S}} \mathbf{f}. \tag{5.12}$$

By renaming the Kronecker products for simplicity, the signal model simplifies to:

$$\mathbf{s} = \mathbf{PTFSf}. \tag{5.13}$$

Note that for the conventional rectangular basis function, $\mathbf{T} = \mathbf{I}_{N_t}$.

5.2.3 B-Spline Considerations

Polynomial functions are useful for approximation because they are differentiable, integrable, and can be easily evaluated. However, polynomial interpolation is very sensitive to the choice of interpolation points. To avoid using high degree polynomials or very large number of interpolation points, it is useful to consider splines, piecewise polynomial functions that are smooth at the connection points, known as knots.

In particular, we choose to look at cardinal splines, whose knots are $\mathbb{Z}$. For cardinal splines, all the splines, $\{B_{i,k}(x)\}_{i=-\infty}^{\infty}$ are shifted versions of the original kth order

spline, $Q_k(x)$, which is defined over $[0, k]$ [93]:

$$B_{i,k}(x) = Q_k\left(\frac{x - ih}{h}\right), \qquad i = -\infty, \ldots, \infty \tag{5.14}$$

This shift-invariance allows us to use convolution to efficiently compute spline interpolation. Furthermore, splines exhibit a property useful for basis functions, partition of unity, i.e. $\sum_i B_i(x) = 1 \ \forall x$. However, the time series we wish to approximate is of finite duration, unlike the knot sequence. To accomodate for this, we modify the first and last spline functions in our temporal basis to have infinite duration and to preserve partition of unity.

$$B_{0,k} = \begin{cases} Q_k\left(\frac{x}{h}\right) + Q_k\left(\frac{x+h}{h}\right), & 1 \leq x \leq k \\ 1, & x < 1 \end{cases} \tag{5.15}$$

$$B_{N_t-1,k} = \begin{cases} Q_k\left(\frac{x-(N_t-i)h}{h}\right) + Q_k\left(\frac{x-N_t h}{h}\right), & N_t - 1 - k \leq x \leq N_t - 1 \\ 1, & x \geq N_t - 1 \end{cases} \tag{5.16}$$

5.3 Proposed Methods

Our measurement model assumes i.i.d. complex Gaussian noise:

$$\mathbf{y} = \mathbf{s} + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma I) \tag{5.17}$$

To perform image reconstruction using this system model, we propose the following

optimization problem:

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, \Psi(f), \qquad \Psi(f) = \frac{1}{2}||y - \mathbf{PTFS}f||_2^2 + \lambda_s R_s(f) + \lambda_t R_t(f) \tag{5.18}$$

$R_s(\cdot)$ is an edge-preserving spatial regularizer, and $R_t(\cdot)$ is a temporal regularizer.

5.3.1 Temporal Regularization with Non-Circulant Boundary Conditions

Revisiting Equation (5.18), we consider choosing quadratically penalized, temporal finite differences $R_t(f) = ||\mathbf{C}_t f||_2^2$ to encourage a smoothly varying contrast agent concentration in the estimated object [94]. In Chapter 3, spatial circulant boundary conditions were not problematic because the support of the object did not extend to the boundaries of the image. However, circulant boundary conditions for temporal regularization would cause unwanted temporal correlations between the first and final temporal basis frames. To continue to take advantage of the circulant Hessian structure resulting from circulant boundary conditions, we use a temporal truncation matrix, $\mathbf{B}_t$, as in [78], to construct regularization term $R_t(f) = ||\mathbf{B}_t\mathbf{C}_t f||_2^2$. $\mathbf{B}_t$ is a identity matrix with the first row omitted. With this design, the regularization term does not penalize differences between the first and last frame. For anisotropic TV in the spatial dimension we define, $\mathbf{C}_s \in \mathbb{R}^{4N_rN_tN_rNt}$ to have the following structure:

$$\mathbf{C}_s \triangleq [\mathbf{C}_{\text{south}}; \mathbf{C}_{\text{east}}; \mathbf{C}_{\text{northeast}}; \mathbf{C}_{\text{southeast}}]. \tag{5.19}$$

We also define a diagonal regularization parameter matrix, $\mathbf{\Lambda}$, whose diagonal consists only of λ_s and $\sqrt{\lambda_t}$, and index sets $\mathcal{S}_1$ and $\mathcal{S}_2$. These sets correspond to the indeces of the spatial differences and the time differences, respectively. The cost function is now:

$$\Psi(f) = \frac{1}{2}||y - \mathbf{PTFS}f||_2^2 + \lambda_s \psi\left(\mathbf{B}_s\mathbf{C}_s f\right) + \lambda_t||\mathbf{B}_t\mathbf{C}_t f||_2^2 \tag{5.20}$$

In this case, the truncation matrix $\mathbf{B}_s$ selects only the spatial non-boundary differences and $\mathbf{B}_t$ selects only the temporal non-boundary differences:

$$\mathbf{B}_s \triangleq \text{diag}\{\mathbf{B}_{\text{south}}, \mathbf{B}_{\text{east}}, \mathbf{B}_{\text{northeast}}, \mathbf{B}_{\text{southeast}}\}. \tag{5.21}$$

5.3.1.1 Variable Splitting

Minimizing Equation (5.20) is equivalent to the following constrained optimization problem:

$$\begin{aligned} \hat{f} = \underset{f,u_0,u_1,u_2,u_3}{\text{argmin}} \; & \Psi_u(u_{0a}, u_{0b}, u_{1a}, u_{1b}, u_2, u_3, f) \\ \text{where } \Psi_u = \frac{1}{2}||y - \mathbf{PT}u_3||_2^2 & + \lambda_s\psi\left(u_{1a}\right) + \lambda_t||u_{1b}||_2^2 \\ \text{s.t. } u_{0a} = \mathbf{C}_s u_2; \quad u_{0b} = \mathbf{C}_t u_2; & \quad u_{1a} = \mathbf{B}_s u_{0a}; \\ u_{1b} = \mathbf{B}_t u_{0b}; \quad u_2 = f; & \quad u_3 = \mathbf{FS}f. \end{aligned} \tag{5.22}$$

Applying the Augmented Lagrangian (AL) framework to this problem and complet-

ing the square yields the following AL cost function:

$$\begin{aligned}\mathcal{L}(f, u_{0a}, u_{0b}, u_{1a}, u_{1b}, u_2, u_3; \mu_0, \mu_1, \mu_2, \mu_3) = \frac{1}{2}||y - \mathbf{PT}u_3||_2^2 + \lambda_s \psi\left(u_{1a}\right) + \lambda_t ||u_{1b}||_2^2 \\ + \frac{\mu_0}{2}||u_{0a} - \mathbf{C}_s u_2 - \eta_{0a}||_2^2 + \frac{\mu_0}{2}||u_{0b} - \mathbf{C}_t u_2 - \eta_{0b}||_2^2 + \frac{\mu_1}{2}||u_{1a} - \mathbf{B}_s u_{0a} - \eta_{1a}||_2^2 \\ + \frac{\mu_1}{2}||u_{1b} - \mathbf{B}_t u_{0a} - \eta_{1b}||_2^2 + \frac{\mu_2}{2}||u_2 - f - \eta_2||_2^2 + \frac{\mu_3}{2}||u_3 - \mathbf{FS}f - \eta_3||_2^2 \quad (5.23)\end{aligned}$$

We update the original and auxiliary variables $f, u_0, \ldots, u_3$ one-by-one in an alternating minimization scheme:

$$\begin{aligned}u_{0a}^{(n+1)} =& \mathbf{H_{0a}}^{-1}\left(\mu_0(\mathbf{C}_s u_2^{(n)} + \eta_{0a}^{(n)}) + \mu_1 \mathbf{B}_s'(u_{1a}^{(n)} - \eta_{1a}^{(n)})\right) \\ u_{0b}^{(n+1)} =& \mathbf{H_{0b}}^{-1}\left(\mu_0(\mathbf{C}_t u_2^{(n)} + \eta_{0b}^{(n)}) + \mu_1 \mathbf{B}_t'(u_{1b}^{(n)} - \eta_{1b}^{(n)})\right) \quad (5.24) \\ u_{1a}^{(n+1)} =& \text{shrink}\left(\mathbf{B}_s u_{0a}^{(n+1)} + \eta_{1a}^{(n)}, \frac{\lambda_s}{\mu_1}\right) \quad (5.25) \\ u_{1b}^{(n+1)} =& \left(\mathbf{B}_t u_{0b}^{(n+1)} + \eta_{1b}^{(n)}\right) / \left(2\frac{\lambda_t}{\mu_1} + 1\right) \quad (5.26) \\ u_2^{(n+1)} =& \mathbf{H_2}^{-1}(\mu_0\left(\mathbf{C}_s'(u_{0a}^{(n+1)} - \eta_{0a}^{(n)}) + \mathbf{C}_t'(u_{0b}^{(n+1)} - \eta_{0b}^{(n)})\right) \\ & + \mu_2(f + \eta_2^{(n)})) \quad (5.27) \\ u_3^{(n+1)} =& \mathbf{H_3}^{-1}\left(\mathbf{T}'\mathbf{P}'y + \mu_3(\mathbf{FS}f^{(n)} + \eta_3^{(n)})\right) \quad (5.28) \\ f^{(n+1)} =& \mathbf{H_f}^{-1}\left(\mu_2(u_2^{(n+1)} - \eta_2^{(n)}) + \mu_3 \mathbf{S}'\mathbf{F}'(u_3^{(n+1)} - \eta_3^{(n)})\right) \quad (5.29)\end{aligned}$$

where the shrinkage operator depends on the potential function, $\phi(\cdot)$. For the ℓ_1

norm, it is soft-thresholding as follows:

$$\text{shrink}(x,t) = \text{sign}(x_i)(|x_i| - t)_+ \tag{5.30}$$

The Hessian matrices are defined as:

$$\mathbf{H_{0a}} \triangleq \mu_0 I + \mu_1 \mathbf{B}_s' \mathbf{B}_s \tag{5.31}$$

$$\mathbf{H_{0b}} \triangleq \mu_0 I + \mu_1 \mathbf{B}_t' \mathbf{B}_t \tag{5.32}$$

$$\mathbf{H_2} \triangleq \mu_2 \mathbf{I} + \mu_0 \left(\mathbf{C}_s' \mathbf{C}_s + \mathbf{C}_t' C t\right) \tag{5.33}$$

$$\mathbf{H_3} \triangleq \mathbf{T}'\mathbf{P}'\mathbf{P}\mathbf{T} + \mu_3 \mathbf{I} \tag{5.34}$$

$$\mathbf{H_f} \triangleq \mu_3 \mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S} + \mu_2 \mathbf{I}. \tag{5.35}$$

We update the dual variables $\eta_0, \ldots, \eta_3$ according to Alternating Direction Minimization as follows:

$$\eta_{0a}^{(n+1)} = \eta_{0a}^{(n)} - (u_{0a}^{(n+1)} - \mathbf{C}_s u_2^{(n+1)}) \tag{5.36}$$

$$\eta_{0b}^{(n+1)} = \eta_{0b}^{(n)} - (u_{0b}^{(n+1)} - \mathbf{C}_t u_2^{(n+1)}) \tag{5.37}$$

$$\eta_{1a}^{(n+1)} = \eta_{1a}^{(n)} - (u_{1a}^{(n+1)} - \mathbf{B}_s u_{0a}^{(n+1)}) \tag{5.38}$$

$$\eta_{1b}^{(n+1)} = \eta_{1}^{(n)} - (u_{1b}^{(n+1)} - \mathbf{B}_t u_{0b}^{(n+1)}) \tag{5.39}$$

$$\eta_{2}^{(n+1)} = \eta_{2}^{(n)} - (u_{2}^{(n+1)} - f^{(n+1)}) \tag{5.40}$$

$$\eta_{3}^{(n+1)} = \eta_{3}^{(n)} - (u_{3}^{(n+1)} - \mathbf{F}\mathbf{S} f^{(n+1)}) \tag{5.41}$$

5.3.1.2 Hessian Structures

$\mathbf{H_{0a}}$ and $\mathbf{H_{0b}}$ consist of masking and scaling and has an easily invertible diagonal structure. $\mathbf{H_f}$ is also diagonal, because the fully-sampled Fourier transform is unitary, and $\mathbf{S'S}$ is also diagonal. Because of the choice of circulant end conditions for finite-differencing matrix $\mathbf{C}$, $\mathbf{H_2}$ is circulant, making the u_2 easy to compute via diagonalization with the DFT matrix. Therefore, all of the variable updates except u_3 are straightforward and easy to compute.

For quadratic B-splines, $\mathbf{H_3}$ is block-diagonal with pentadiagonal blocks. This update can be solved with an efficient solver similar to, but more complex than the tridiagonal solver in Chapter 4. We can also exploit the N_r independent blocks of $\mathbf{H_3}$ to parallelize the variable update. For now, we compute the u_3 update with an inner iterative method, CG. Future work involves finding an efficient, parallelizable way to solve the u_3 update.

5.3.2 Spline Temporal Basis with ADMM

Here we present an alternate optimization method for minimizing Equation (5.20) that has guaranteed convergence properties. We do so by demonstrating equivalence with the Alternating Direction Method of Multipliers [75].

We manipulate the previous variable splitting to fit the conditions of the Alternating Direction Method of Multipliers (ADMM) by examining and factoring the constraint matrix, inspired by [38], and previously discussed in Section 4.3.3. The constraint

matrix for the previous splitting is constructed by consolidating the splitting constraints in matrix form as follows:

$$\underbrace{\begin{bmatrix} \mathbf{I} & -\begin{bmatrix}\mathbf{B}_s\\ \mathbf{B}_t\end{bmatrix} & & & \\ & \mathbf{I} & -\begin{bmatrix}\mathbf{C}_s\\ \mathbf{C}_t\end{bmatrix} & & \\ & & \mathbf{I} & -\mathbf{I} & \\ & & & \mathbf{I} & -\mathbf{FS} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \begin{bmatrix}u_{1a}\\ u_{1b}\end{bmatrix} \\ \begin{bmatrix}u_{0a}\\ u_{0b}\end{bmatrix} \\ u_2 \\ u_3 \\ f \end{bmatrix}}_{u} = 0. \tag{5.42}$$

We define the constant N_p as shorthand for $N_p = N_x N_y N_t$, the number of elements in f. For a convenient block upper triangular structure, we rearrange the first and second constraints, resulting in the following constraint matrix, $\mathbf{A}$. We also stack all the original and auxiliary variables into a vector, u.

To satisfy the conditions for ADMM with inexact updates, we desire to have a constraint matrix with full column rank. We can factor $\mathbf{A}$ and adapt our cost function, Ψ_u from (5.22) to accomodate the manipulation. We choose to factor $\mathbf{A}$ into square $\mathbf{A}_1$, which has separated the $u_2 = f$ requirement, and $\mathbf{A}_2$ (same size as

$\mathbf{A}$) which reinforces that constraint as follows:

$$\underbrace{\begin{bmatrix} \mathbf{I} & & & \\ & \mathbf{I} & & \\ & & \mathbf{I} & \mathbf{I} \\ & & & \mathbf{I} \end{bmatrix}}_{\mathbf{A}_2} \underbrace{\begin{bmatrix} \mathbf{I} & -\begin{bmatrix} \mathbf{B}_s \\ \mathbf{B}_t \end{bmatrix} & & & \\ & \mathbf{I} & -\begin{bmatrix} \mathbf{C}_s \\ \mathbf{C}_t \end{bmatrix} & & \\ & & \mathbf{I} & & \\ & & & \mathbf{I} & -\mathbf{FS} \\ & & & & -\mathbf{I} \end{bmatrix}}_{\mathbf{A}_1} \begin{bmatrix} \begin{bmatrix} u_{1a} \\ u_{1b} \end{bmatrix} \\ \begin{bmatrix} u_{0a} \\ u_{0b} \end{bmatrix} \\ u_2 \\ u_3 \\ f \end{bmatrix} = 0. \tag{5.43}$$

Since $\mathbf{A}_1$ is block upper triangular with $\mathbf{I}$ along the diagonal, it has full rank, and using $\mathbf{A}_1$ in lieu of $\mathbf{A}$ as the constraint matrix will satisfy ADMM conditions. To enforce equivalence with Eqn (5.22), we augment the original cost function over u, $\Psi_u(u)$, with $\Psi_v(v)$:

$$\min \ \Psi_u(u) + \Psi_v(v) \qquad \text{s.t.} \ \begin{bmatrix} A_1 - \mathbf{I} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0 \tag{5.44}$$

with the following two convex functions in the cost function:

$$\Psi_u(u) \triangleq \frac{1}{2}||y - \mathbf{PT}u_3||^2 + \lambda_s \psi\left(u_{1a}\right) + \lambda_t ||u_{1b}||_2^2 \tag{5.45}$$

$$\Psi_v(v) \triangleq \begin{cases} 0, & A_2 v = 0 \\ +\infty, & A_2 v \neq 0. \end{cases} \tag{5.46}$$

For notational ease, we divide the additional variable v into its constituent parts: $v_0, v_1 \in \mathbb{C}^{5N_p}$, $v_2 \in \mathbb{C}^{N_p}$, $v_3 \in \mathbb{C}^{N_pN_c}$, and $v_4 \in \mathbb{C}^{N_p}$. Calculations with v become simple because Equation (5.46) requires that only two v variables are non-trivial and have the relationship: $v_2 = -v_4$. The others must equal zero, $v_0, v_2, v_3 = 0$, and are not included in the following variable updates. These simplifications reduce the ADMM requirement into a simple additional variable update: $u_2 = v_2$ and $v_2 = f$ instead of a direct $u_2 = f$.

To enforce the constraints in Equation (5.44), we turn again to the Augmented Lagrangian, which looks very similar to the non-ADMM variable splitting. The differences are highlighted in red below:

$$\begin{aligned}\mathcal{L}(u, v; \mu_0, \ldots, \mu_4) &= \Psi_u(u) + \Psi_v(v) + \frac{\mu}{2}||\mathbf{A}_1 u - v - \eta||^2_{\frac{\mu}{2}} \\ &= \frac{1}{2}||y - \mathbf{PT}u_3||_2^2 + \lambda_s \psi\left(u_{1a}\right) + \lambda_t||u_{1b}||_2^2 + \frac{\mu_0}{2}||u_{0a} - \mathbf{C}_s u_2 - \eta_{0a}||_2^2 \\ &\quad + \frac{\mu_0}{2}||u_{0b} - \mathbf{C}_t u_2 - \eta_{0b}||_2^2 + \frac{\mu_1}{2}||u_{1a} - \mathbf{B}_s u_{0a} - \eta_{1a}||_2^2 + \frac{\mu_1}{2}||u_{1b} - \mathbf{B}_t u_{0a} - \eta_{1b}||_2^2 \\ &\quad + \frac{\mu_2}{2}||u_2 - \color{red}{v_2} - \eta_2||_2^2 + \frac{\mu_3}{2}||u_3 - \mathbf{FS}f - \eta_3||_2^2 + \color{red}{\frac{\mu_4}{2}||-f + v_2 - \eta_4||_2^2}.\end{aligned} \tag{5.47}$$

Here we use μ as shorthand to indicate a diagonal matrix with values $\mu_0, \ldots, \mu_4$ and η to be the stack of all the scaled Lagrange multipliers, $\eta_{0a}, \ldots, \eta_4$. The resulting alternating minimization variable updates are listed below, with the differences again

highlighted in red:

$$u_{0a}^{(n+1)} = \mathbf{H_{0a}}^{-1}\left(\mu_0(\mathbf{C}_s u_2^{(n)} + \eta_{0a}^{(n)}) + \mu_1 \mathbf{B}_s'(u_{1a}^{(n)} - \eta_{1a}^{(n)})\right) \tag{5.48}$$

$$u_{0b}^{(n+1)} = \mathbf{H_{0b}}^{-1}\left(\mu_0(\mathbf{C}_t u_2^{(n)} + \eta_{0b}^{(n)}) + \mu_1 \mathbf{B}_t'(u_{1b}^{(n)} - \eta_{1b}^{(n)})\right) \tag{5.49}$$

$$u_{1a}^{(n+1)} = \text{shrink}\left(\mathbf{B}_s u_{0a}^{(n+1)} + \eta_{1a}^{(n)}, \frac{\lambda_s}{\mu_1}\right) \tag{5.50}$$

$$u_{1b}^{(n+1)} = \frac{\mathbf{B}_t u_{0b}^{(n+1)} + \eta_{1b}^{(n)}}{2\frac{\lambda_t}{\mu_1} + 1} \tag{5.51}$$

$$u_2^{(n+1)} = \mathbf{H_2}^{-1}\mu_0\left(\mathbf{C}_s'(u_{0a}^{(n+1)} - \eta_{0a}^{(n)}) + \mathbf{C}_t'(u_{0b}^{(n+1)} - \eta_{0b}^{(n)})\right) + \mathbf{H_2}^{-1}\mu_2(v_2^{(n)} + \eta_2^{(n)}) \tag{5.52}$$

$$u_3^{(n+1)} = \mathbf{H_3}^{-1}\left(\mathbf{T}'\mathbf{P}'y + \mu_3(\mathbf{F}\mathbf{S}f^{(n)} + \eta_3^{(n)})\right) \tag{5.53}$$

$$f^{(n+1)} = \mathbf{H_f}^{-1}\left(\mu_4(v_2^{(n)} - \eta_4^{(n)}) + \mu_3\mathbf{S}'\mathbf{F}'(u_3^{(n+1)} - \eta_3^{(n)})\right) \tag{5.54}$$

$$v_2^{(n+1)} = \mathbf{H_v}^{-1}\left(\mu_4(f^{(n+1)} + \eta_4^{(n)}) + \mu_2(u_2^{(n+1)} - \eta_2^{(n)})\right). \tag{5.55}$$

The Hessian matrices are defined as before, except $\mathbf{H_f} = \mu_3\mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S} + \mu_4\mathbf{I}$ and $\mathbf{H_v} = \mu_2\mathbf{I} + \mu_4\mathbf{I}$. The updates for the scaled Lagrange multipliers are similar to the AL variable splitting, except that $\eta_2^{(n+1)} = \eta_2^{(n)} - u_2^{(n+1)} - v_2^{(n+1)}$, and there is an additional component to update: $\eta_4^{(n+1)} = \eta_4^{(n)} - (v_2^{(n+1)} - f^{(n+1)})$.

5.3.3 Spline Temporal Basis with fully expanded ADMM

The proposed ADMM method in subsection 5.3.2 relies on the assumption that one pass of alternating minimization for u and then one pass for v are sufficiently close to their true joint optimal values before performing the Lagrange multiplier updates.

To avoid this assumption (or having to perform multiple passes of variable updates within each ADMM iteration), we seek an alternate ADMM splitting which fully decouples the variable updates of u from that of v. To do so, we factor the original constraint matrix $\mathbf{A}$ more aggresively, placing each block matrix onto its own block line:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} & & & & & & \\ & & \mathbf{I} & \mathbf{I} & & & & \\ & & & & \mathbf{I} & \mathbf{I} & & \\ & & & & & & \mathbf{I} & \mathbf{I} \end{bmatrix}}_{\mathbf{A}_2} \underbrace{\begin{bmatrix} \mathbf{I} & & & & & & \\ & -\begin{bmatrix}\mathbf{B}_s \\ \mathbf{B}_t\end{bmatrix} & & & & & \\ & & \mathbf{I} & & & & \\ & & & -\begin{bmatrix}\mathbf{C}_s \\ \mathbf{C}_t\end{bmatrix} & & & \\ & & & & \mathbf{I} & & \\ & & & & & -\mathbf{I} & \\ & & & & & & \mathbf{I} \\ & & & & & & -\mathbf{FS} \end{bmatrix}}_{\mathbf{A}_1} \underbrace{\begin{bmatrix} \begin{bmatrix} u_{1a} \\ u_{1b} \end{bmatrix} \\ \begin{bmatrix} u_{0a} \\ u_{0b} \end{bmatrix} \\ u_2 \\ u_3 \\ f \end{bmatrix}}_{\mathbf{u}} = 0. \tag{5.56}$$

As in Equation (5.44), we choose to minimize $\Psi_u(u) + \Psi_v(v)$, which enforces that $\mathbf{A}_2 v = 0$. This relationship simplifes to four constraints: $v_0 = -v_1, v_2 = -v_3, v_4 = -v_5, v_6 = -v_7$. We simplify the resulting Augmented Lagrangian function by enforc-

ing those equivalences as follows:

$$\begin{aligned}
\mathcal{L}(u, v; \mu_0, \ldots, \mu_7) = \frac{1}{2}||y - \mathbf{PT}u_3||_2^2 + \lambda_s \psi\left(u_{1a}\right) + \lambda_t ||u_{1b}||_2^2 \\
+ \frac{\mu_0}{2}||u_{1a} - v_{0a} - \eta_{0a}||_2^2 + \frac{\mu_0}{2}||u_{1b} - v_{0b} - \eta_{0b}||_2^2 \\
+ \frac{\mu_1}{2}||-\mathbf{B}_s u_{1a} + v_{0a} - \eta_{1a}||_2^2 + \frac{\mu_1}{2}||-\mathbf{B}_t u_{1b} + v_{0b} - \eta_{1b}||_2^2 \\
+ \frac{\mu_2}{2}||u_{0a} - v_{2a} - \eta_{2a}||_2^2 + \frac{\mu_2}{2}||-u_{0b} - v_{2b} - \eta_{2b}||_2^2 \\
+ \frac{\mu_3}{2}||-\mathbf{C}_s u_{0a} + v_{2a} - \eta_{3a}||_2^2 + \frac{\mu_3}{2}||-\mathbf{C}_t u_{0b} + v_{2b} - \eta_{3b}||_2^2 \\
+ \frac{\mu_4}{2}||u_2 - v_4 - \eta_4||_2^2 + \frac{\mu_5}{2}||-f + v_4 - \eta_5||_2^2 \\
+ \frac{\mu_6}{2}u_3 - v_6 - \eta_6||_2^2 + \frac{\mu_7}{2}||-\mathbf{FS}f + v_6 - \eta_7||_2^2. \qquad (5.57)
\end{aligned}$$

Though the Augmented Lagrangian has gained more terms and we need to compute variable updates for four non-trivial v terms, the individual variable updates for the u terms are fully separated from each other, as are the v updates. This eliminates any ambiguity about the use of alternating minimization to execute ADMM. More

importantly, all of the following u updates can be computed in parallel:

$$u_{0a}^{(n+1)} = (\mu_1 \mathbf{B}_s' \mathbf{B}_s + \mu_2 \mathbf{I})^{-1} \left(\mu_1 \mathbf{B}_s' \left(v_{0a}^{(n)} - \eta_{1a}^{(n)}\right) + \mu_2 \left(v_{2a}^{(n)} + \eta_{2a}^{(n)}\right)\right) \tag{5.58}$$

$$u_{0b}^{(n+1)} = (\mu_1 \mathbf{B}_t' \mathbf{B}_t + \mu_2 \mathbf{I})^{-1} \left(\mu_1 \mathbf{B}_t' \left(v_{0b}^{(n)} - \eta_{1b}^{(n)}\right) + \mu_2 \left(v_{2b}^{(n)} + \eta_{2b}^{(n)}\right)\right) \tag{5.59}$$

$$u_{1a}^{(n+1)} = \text{shrink}\left(v_{0a}^{(n)} + \eta_{0a}^{(n)}, \frac{\lambda_s}{\mu_0}\right) \tag{5.60}$$

$$u_{1b}^{(n+1)} = \frac{v_{0b}^{(n)} + \eta_{0b}^{(n)}}{2\frac{\lambda_t}{\mu_0} + 1} \tag{5.61}$$

$$u_2^{(n+1)} = (\mu_4 \mathbf{I} + \mu_3(\mathbf{C}_s' \mathbf{C}_s + \mathbf{C}_t' \mathbf{C}_t))^{-1} (\mu_3 \left(\mathbf{C}_s'(v_{2a}^{(n)} - \eta_{3a}^{(n)}) + \mathbf{C}_t'(v_{2b}^{(n)} - \eta_{3b}^{(n)})\right) + \mu_4(v_4^{(n)} + \eta_4^{(n)}) \tag{5.62}$$

$$u_3^{(n+1)} = (\mathbf{T}'\mathbf{P}'\mathbf{P}\mathbf{T} + \mu_3 \mathbf{I})^{-1} \left(\mathbf{P}\mathbf{T}' y + \mu_6(v_6^{(n)} + \eta_6^{(n)})\right) \tag{5.63}$$

$$f^{(n+1)} = (\mu_7 \mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S} + \mu_5 \mathbf{I})^{-1} \left(\mu_7 \mathbf{S}'\mathbf{F}'(v_6^{(n)} - \eta_7^{(n)}) + \mu_5(v_4^{(n)} - \eta_5^{(n)})\right). \tag{5.64}$$

The v updates are also parallelizable with respect to each other:

$$v_{0a}^{(n+1)} = \frac{\mu_0(u_{1a}^{(n+1)} - \eta_{0a}^{(n)}) + \mu_1(\mathbf{B}_s u_{0a}^{(n+1)} + \eta_{1a}^{(n)})}{\mu_0 + \mu_1} \tag{5.65}$$

$$v_{0b}^{(n+1)} = \frac{\mu_0(u_{1b}^{(n+1)} - \eta_{0b}^{(n)}) + \mu_1(\mathbf{B}_t u_{0b}^{(n+1)} + \eta_{1b}^{(n)})}{\mu_0 + \mu_1} \tag{5.66}$$

$$v_{2a}^{(n+1)} = \frac{\mu_2(u_{0a}^{(n+1)} - \eta_{2a}^{(n)}) + \mu_3(\mathbf{C}_s u_2^{(n+1)} + \eta_{3a}^{(n)})}{\mu_2 + \mu_3} \tag{5.67}$$

$$v_{2b}^{(n+1)} = \frac{\mu_2(u_{0b}^{(n+1)} - \eta_{2b}^{(n)}) + \mu_3(\mathbf{C}_t u_2^{(n+1)} + \eta_{3b}^{(n)})}{\mu_2 + \mu_3} \tag{5.68}$$

$$v_4^{(n+1)} = \frac{\mu_4(u_2^{(n+1)} - \eta_4^{(n)}) + \mu_5(f^{(n+1)} + \eta_5^{(n)})}{\mu_4 + \mu_5} \tag{5.69}$$

$$v_6^{(n+1)} = \frac{\mu_6(u_3^{(n+1)} - \eta_6^{(n)}) + \mu_7(\mathbf{FS} f^{(n+1)} + \eta_7^{(n)})}{\mu_6 + \mu_7}. \tag{5.70}$$

The redundant v_1, v_3, v_5, and v_7 variables are omitted entirely.

Expanding $\eta = \eta - (\mathbf{A_1}u - v)$, we have the following scaled Lagrange multiplier

updates:

$$\eta_{0a}^{(n+1)} = \eta_{0a}^{(n)} - (u_{1a}^{(n+1)} - v - 0a^{(n+1)}) \tag{5.71}$$

$$\eta_{0b}^{(n+1)} = \eta_{0b}^{(n)} - (u_{1b}^{(n+1)} - v - 0b^{(n+1)}) \tag{5.72}$$

$$\eta_{1a}^{(n+1)} = \eta_{1a}^{(n)} - (-\mathbf{B}_s u_{0a}^{(n+1)} + v_{0a}^{(n+1)}) \tag{5.73}$$

$$\eta_{1b}^{(n+1)} = \eta_{1b}^{(n)} - (-\mathbf{B}_t u_{0b}^{(n+1)} + v_{0b}^{(n+1)}) \tag{5.74}$$

$$\eta_{2a}^{(n+1)} = \eta_{2a}^{(n)} - (u_{0a}^{(n+1)} - v_{2a}^{(n+1)}) \tag{5.75}$$

$$\eta_{2b}^{(n+1)} = \eta_{2b}^{(n)} - (u_{0b}^{(n+1)} - v_{2b}^{(n+1)}) \tag{5.76}$$

$$\eta_{3a}^{(n+1)} = \eta_{3a}^{(n)} - (-\mathbf{C}_s u_2^{(n+1)} + v_{2a}^{(n+1)}) \tag{5.77}$$

$$\eta_{3b}^{(n+1)} = \eta_{3b}^{(n)} - (-\mathbf{C}_t u_2^{(n+1)} + v_{2b}^{(n+1)}) \tag{5.78}$$

$$\eta_4^{(n+1)} = \eta_4^{(n)} - (u_2^{(n+1)} - v_4^{(n+1)}) \tag{5.79}$$

$$\eta_5^{(n+1)} = \eta_5^{(n)} - (-f^{(n+1)} + v_4^{(n+1)}) \tag{5.80}$$

$$\eta_6^{(n+1)} = \eta_6^{(n)} - (u_3^{(n+1)} - v_6^{(n+1)}) \tag{5.81}$$

$$\eta_7^{(n+1)} = \eta_7^{(n)} - (-\mathbf{F}\mathbf{S}^{(n+1)} + v_6^{(n+1)}). \tag{5.82}$$

5.4 Methods

We implemented the method in Section 5.3.1 in Matlab using the Image Reconstruction Toolbox [71] and conducted experiments on an Ubuntu machine with a 4-core 3.3 GHz Intel CPU.

We conducted simulations on a Brainweb T_1 image downsampled to dimensions $N_y = 336$ and $N_x = 336$ with an additional circular tumor placed manually in a uniform

region of the brain, using moderate enhancement values for K^{trans} and k_{ep}, 0.6 and 2.0 min^{-1}, respectively. We chose $T_R = 4.6$ msec and a flip angle of 10°.

We simulated the use of 8 sensitivity coils. We constructed the measurement data by generating physiological data for the total number of subframes, applying the sensitivity maps, applying the Fourier transform, and then adding complex, white Gaussian noise with an SNR of 20.

We chosen quadratic B-splines with spacing $Q = 3$ and $Q = 7$ for the temporal basis functions, resulting in 9- and 21-tap spline filters, and we designed the boundary conditions such that the first and last frame occured at spline peaks. We chose the spatial and temporal regularization parameters, λ_s and λ_t, based on an exhaustive search over the parameter space to minimize squared error with the known true image coefficients.

5.4.1 Sampling Scheme

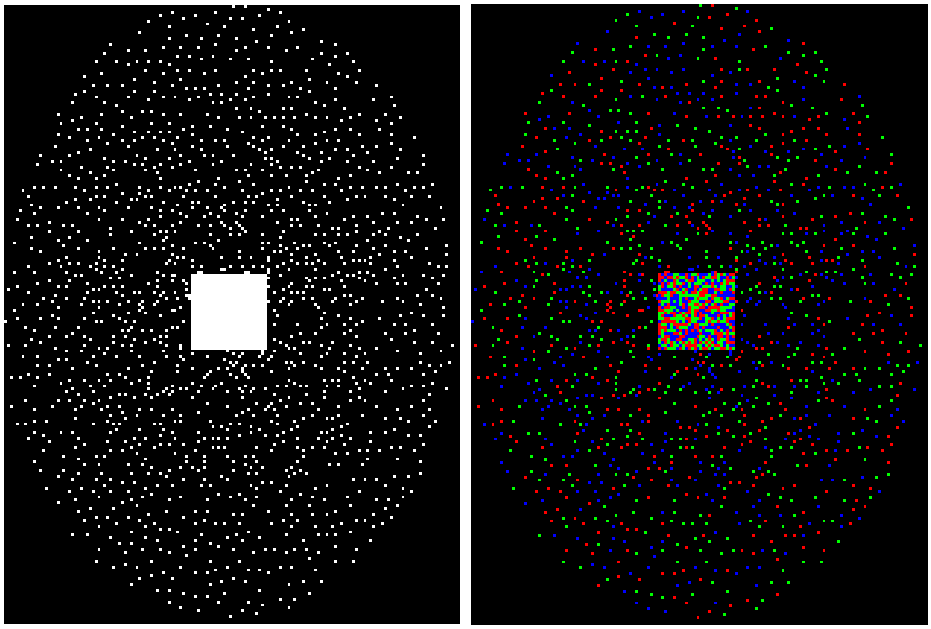
Because we approximated the object as stationary for M successive readouts, we group those readouts together into a "subframe", similar to the conventional notion of a frame in dynamic imaging. We wish to distinguish subframes, which have higher undersampling and higher temporal resolution, from the temporal basis frames, which coincide with temporal basis coefficients and number fewer by a factor of Q: $N_t = N_f/Q$.

This subsection describes the design choices for the sampling scheme at both the subframe level and the temporal basis frame level. For a wide rectangle temporal

basis, it is natural to consider the temporal basis frame sampling pattern to be the union of the sampling frames of the associated subframes. With this in mind, we modified the Poisson-disk sampling [70] pattern to enforce constraints within each subframe and across neighboring subframes at the temporal basis frame level.

The rectangular central region of k-space is fully sampled in each temporal basis frame because the majority of the information and energy in k-space lies at or near DC. We assigned the center k-space region to to each subframe randomly for an approximate sampling rate in the central subframe of $\frac{1}{Q}$.

Outside this region, the sampling scheme is a variable density Poisson disk sampling pattern, modified to enforce minimum distance constraints in each subframe as well as across its temporal basis subframe group. One desirable result was that no k-space location in this outer region is sampled more than once in each set of Q frames. The overall effect is a variable density undersampling pattern, with guaranteed sampling of central k-space every Q frames and specialized poisson disk sampling pattern at higher frequencies. This sampling scheme is pictured in Figure 5.4.



(a) (b)

Figure 5.4: Variable density Poisson-disk sampling pattern in k_y-k_z plane for $Q = 3$ consecutive subframes, or equivalently 1 temporal basis frame. (a) the aggregate sampling pattern for the temporal basis frame (b) colors indicate the three subframe sampling patterns that constitute the overall temporal basis frame sampling pattern.

5.4.2 Simulated Contrast Agent Dynamics

We simulated the effect of contrast agent on the T_1-weighted images by using population-based enhancement curves for the aorta concentration, $C_{\mathrm{p}}^{\mathrm{Aorta}}(t)$, which in turn models the concentration in the blood plasma, $C_{\mathrm{p}}(t)$, which determines the ultimate time series of interest, the tissue concentration, $C_{\mathrm{t}}(t)$.

These quantities are related through a transport function, $h(t)$ and a two-compartment model [95]. We designed the concentration in the Aorta, $C_{\mathrm{p}}^{\mathrm{Aorta}}(t)$ based on a popu-

lation average described by:

$$C_{\mathrm{p}}^{\mathrm{Aorta}}(t) = 7.5527e^{\frac{-(t-0.171)^2)}{0.00605}} + 1.003e^{\frac{-(t-0.364)^2}{0.035912}} + 1.064\frac{e^{-0.083t}}{1+e^{-37.772(t-0.482)}}. \tag{5.83}$$

The concentration of contrast agent in the blood plasma near the tissue can be described as a convolution of a transport function, $h(t)$, and $C_{\mathrm{p}}^{\mathrm{Aorta}}(t)$:

$$C_{\mathrm{p}}(t) = C_{\mathrm{p}}^{\mathrm{Aorta}}(t) * h(t). \tag{5.84}$$

Factors such as the bolus arrival time are included in $h(t)$. Lastly, the two-compartment model between the blood plasma and tissue describes the forward flow and back-wash of contrast agent through the following convolution, as described in Section 2.4.2:

$$C_{\mathrm{t}}(t) = K^{\mathrm{trans}}C_{\mathrm{p}}(t) * (e^{-k_{\mathrm{ep}}t}u(t)). \tag{5.85}$$

$u(t)$ is the unit step function. The concentration of contrast agent in the tissue shortens the T_1 constant of the tissue according to:

$$\frac{1}{T_1(t)} = \frac{1}{T_{10}} + r_1 C_{\mathrm{t}}(t) \tag{5.86}$$

where r_1 is the relaxivity parameter, T_{10} is the inherent spin-lattice constant of the tissue, and $T_1(t)$ is the new, dynamically changing spin-lattice constant.

Lastly, the ratio between the spin-lattice constant and T_R as well as the flip angle, θ, and echo time and T_2^* value determine the MR signal:

$$m_0 = \rho_0 \sin(\theta) \left(\frac{1 - E_1}{1 - E_1 cos(\theta)} \right) e^{-\frac{T_E}{T_2^*}}; \qquad E_1 \triangleq e^{-\frac{T_r}{T_1(t)}} \tag{5.87}$$

5.4.3 Choice of Spline Parameters

For a scan duration, t_{scan}, of 4 minutes and T_R of 4.6 msec, we have the undersampling factor, u, ratio of subframes to temporal basis frames, Q, and the number of temporal basis frames, N_t related as follows:

$$N_t = \frac{t_{\text{scan}}}{\Delta_t} = \frac{t_{\text{scan}}}{Q\Delta_s} = \frac{t_{\text{scan}} \cdot u}{QN_xN_yT_R}. \tag{5.88}$$

We chose the subframe to temporal basis ratio, Q, and the number of splines, N_t, to be $Q = 3$ and $N_t = 32$. We chose this spline basis for its ability to represent the true tumor voxel time series with good fidelity. With fixed values for Q and N_t, we shifted the spline basis with respect to the bolus arrival to allow for optimal representation of the tumor time series, resulting in a NRMSE of 0.06%.

For an image size of 156×212, this results in a subframe undersampling factor of 60, or alternatively a temporal basis frame undersampling factor of 20.

5.4.4 Choice of Convergence Parameters

We chose the AL convergence parameters, $\{\mu_i\}_{i=0}^{3}$ to upper bound the condition numbers of $\mathbf{H_0}$, $\mathbf{H_2}$, $\mathbf{H_3}$ and $\mathbf{H_f}$ (Equations (5.31) - (5.35)). We chose to set an upper bound of 10 for each of these condition numbers based on [67]. The condition number of $\mathbf{H_2}$ is known exactly, since we have predetermined the number of neighbors for $\mathbf{C}_s$ and $\mathbf{C}_t$, and the maximum eigenvalue for $\mathbf{B'\Lambda'\Lambda B}$ is simply the large of the two regularization parameters squared. The diagonal $\mathbf{S'F'FS}$ has a maximum eigenvalue that is N_R times the maximum sum of squares value of the estimated sensitivity maps.

Because $\mathbf{T'P'PT} \prec \mathbf{T'T}$, we bound the maximum eigenvalue of $\mathbf{T'P'PT}$ with that of $\mathbf{T'T}$. After permutation, $\mathbf{T'T}$ is a block diagonal matrix, consisting of identical $N_t \times N_t$ blocks which are approximately Toeplitz, except for the boundary conditions. From [96], we bound the eigenvalues of $\mathbf{T'T}$ by the maximum value of the Fourier series of the Toeplitz sequence.

5.5 Results

5.5.1 Simulation 1: Moderate Enhancement

Moderate enhancement for a lesion is characterized by rapid rise in contrast agent concentration and a steady, slow washout, with typical values of $K^{\text{trans}} \approx 0.6$ min^{-1} and $k_{\text{ep}} \approx 2.0$ min^{-1}. We compared the estimated voxel values, contrast agent

time series, and error in k_{ep} and K^{trans} for the following methods, outlined in Table 5.1.

temporal basis \ optimization method	N/A (non-iterative)	variable splitting and AL	corner rounding and CG
N_f subframe rects	datasharing	skinny rects AL	
N_t temporal basis frame rects		wide rects AL	
N_t quadratic splines		spline AL	spline CG

Table 5.1: Proposed spline AL method and related methods for comparison. Methods are classified by choice of temporal basis and optimization method.

Beyond the optimization method and temporal basis choice for each method, we describe the methods in more detail below.

datasharing : temporal nearest neighbor assignment, followed by inverse FFT, used to initialize all iterative methods

skinny rects AL : each of the $N_f = QN_t$ subframes interact only via temporal regularization, frame rate of 1.99 seconds, solved with AL method in Section 5.3.1 ($\mathbf{T} = \mathbf{I}$)

wide rects AL : each of the rectangular temporal basis frames interact only via temporal regularization, frame rate of 5.96 seconds for $Q = 3$ and undersampling of 60, solved with AL method in Section 5.3.1 ($\mathbf{T} = \mathbf{I}$)

spline AL : each of the spline temporal basis coefficients interact through inter-

polation of subframe values in the datafit term as well as through temporal regularization, solved with AL method in Section 5.3.1

spline CG : spline temporal basis coefficients interact through datafit term as well as through temporal regularization, solved with conjugate gradient with a Fair potential spatial regularization

Figure 5.5 shows the mean time series in the dynamically changing tumor region. We estimated the tissue concentrations, $C_{\mathrm{t}}(t)$, Fig. 5.6, from the estimated time series via the method of moments approach in Section 2.4.2.1. We used the variable projection approach in Section 2.4.2.3 to estimate the K^{trans} and k_{ep} parameters, listed in Table 5.2.

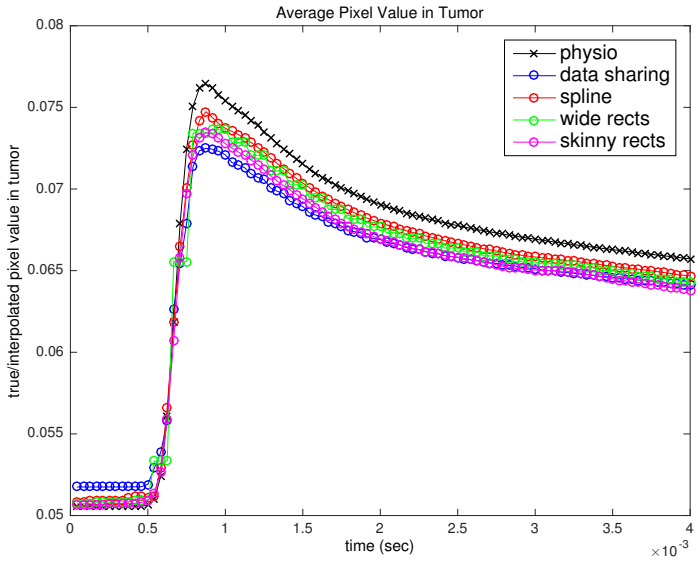


Figure 5.5: Estimated time-series for undersampling factor $u = 60$ of moderate contrast enhancement. The ground truth is marked in black and labeled "physio".

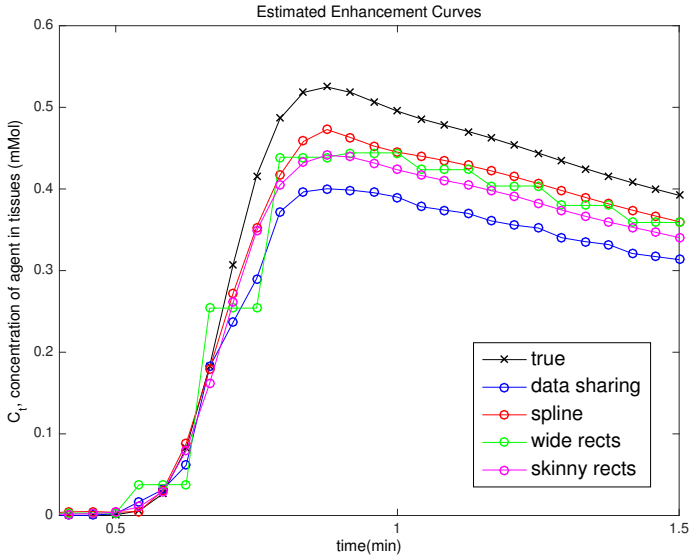


Figure 5.6: Estimated contrast agent concentrations $C_{\mathrm{t}}(t)$ for undersampling factor of 60 of moderate enhancement. Time window is shortened from full 4 minute experiment to 1 minute to emphasize bolus arrival.

	physio (ground truth)	data-sharing	spline	wide rect	skinny rect
k_{ep}	2.00	1.88	1.95	1.89	1.96
% error k_{ep}	-	6.0	**2.5**	5.5	**2.5**
K^{trans}	0.600	0.45	0.53	0.51	0.51
% error K^{trans}	-	25	**11.2**	15.6	15.6
compute time [sec]	-	**25**	**278**	1185	1812

Table 5.2: Estimated pharmacokinetic parameters for undersampling of 60 and moderate enhancement and compute time until k_{ep} and K^{trans} remained within 1% of final value.

Intuitively, the wide rectangular basis is unable to accurately model the sharp rise of the contrast agent concentration resulting in a poor estimate of both pharmacokinetic parameters. The spline AL method is able to reach a better estimate of K^{trans}

than either the wide rectangle AL method and the skinny rectangle AL method and performs comparably with skinny rectangle AL on k_{ep} estimation. However, the compute time for spline AL was significantly less than either wide rectangle or skinny rectangle AL. Compute time was defined as seconds until the iterative method remained within 1% of its final value for both k_{ep} and K^{trans}.

5.5.2 Simulation 2: Rapid Enhancement

Rapid enhancement of contrast agent concentration is typified by values of $K^{\text{trans}} \approx 3.0\ \text{min}^{-1}$ and $k_{\text{ep}} \approx 6.0\ \text{min}^{-1}$. We used the tissue concentrations, $C_{\text{t}}(t)$, Fig. 5.8 to estimate the K^{trans} and k_{ep} parameters, listed in Table 5.3.

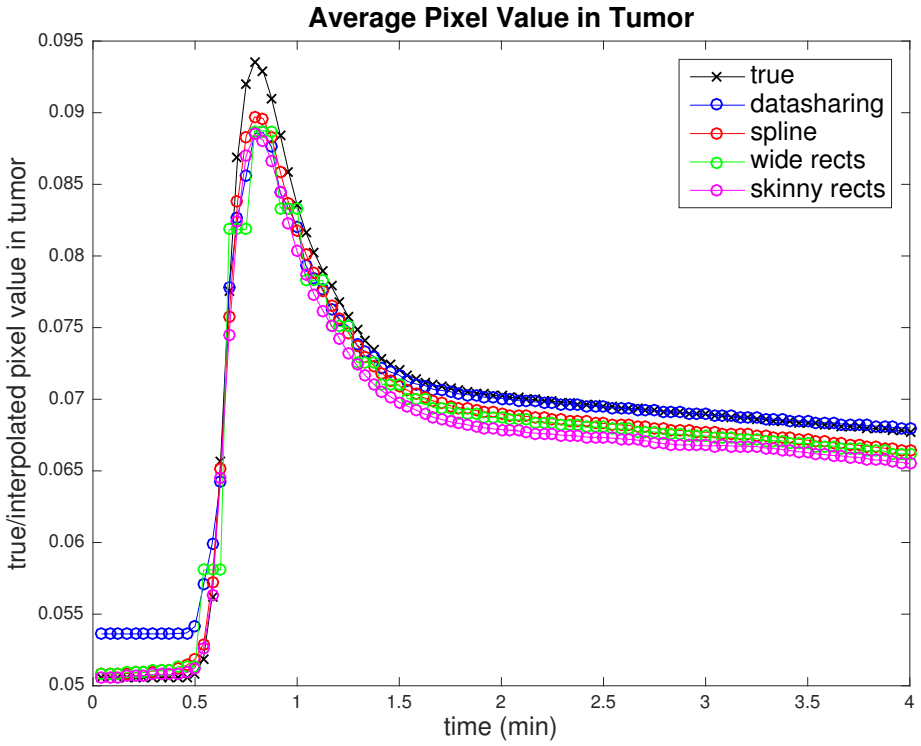


Figure 5.7: Estimated time-series for undersampling factor of 60 for rapid enhancement.

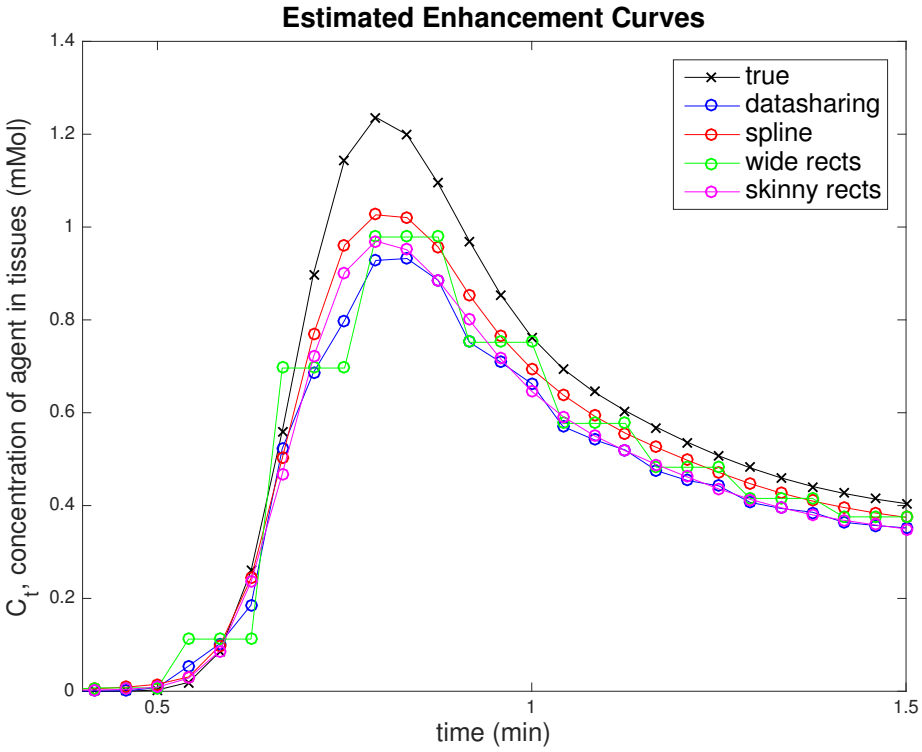


Figure 5.8: Estimated contrast agent concentrations $C_{\mathrm{t}}(t)$ for undersampling factor of 60 for rapid enhancement. Time window is shortened from full 4 minute experiment to 1 minute to emphasize bolus arrival.

	physio (ground truth)	data-sharing	spline	wide rect	skinny rect
k_{ep}	6.00	4.96	6.01	5.28	6.02
% error k_{ep}	-	17.3	**0.2**	12.0	**0.3**
K^{trans}	2.00	1.32	2.03	1.80	2.00
% error K^{trans}	-	34.0	**1.5**	10.0	**0.0**
compute time [sec]	-	**26.8**	330.0	219.3	505.1

Table 5.3: Estimation of pharmacokinetic parameters for undersampling rate of 60 and rapid enhancement.

For the rapid enhancement, the deficiencies of the wide rectangle basis are more

noticeable and the datasharing method also fails to capture the full rise of the bolus arrival. The improvement that the spline basis offers for pharmacokinetic parameter estimation is most evident in this case, as shown in Table 5.3.

5.5.3 Experiment: Free-breathing Pediatric Data

We also applied the proposed spline model to free-breathing abdominal DCE MRI data from a 6.5 year old male patient with an abdominal mass. The data was acquired on GE 3T MR750 scanner with a 3D modified SPGR sequence with Butterfly motion navigation [49] and a VD Rad sampling pattern [31], with T_R of 3.7 ms, flip angle 15°, voxel size of 1.1×1.1×0.9 mm. The 2D slice size was 156×100 with 18 frames of 7.6 seconds each, resulting in an undersampling factor of 7.6. We estimated sensitivity maps with [1] and we applied soft-gating [6] to the free-breathing data to reduce the effect of motion.

Figure 5.9: VDRad sampling pattern broken in subframes of 2.5 second durations ($Q = 3$). First three of 54 subframes are shown. The union of these three sampling patterns describes the sampling pattern for one temporal basis frame. The overall reduction factor is ≈ 22.8.

5.5.3.1 Soft Gating for Free-Breathing Data

The soft respiratory gating applies weights to k-space data corresponding to the respiratory cycle phase at which that datapoint was acquired. The respiratory cycle for a given readout line, d, is measured from Butterfly navigator information and the corresponding soft-gating weight is calculated as follows:

$$w = \begin{cases} e^{-\alpha(d-\tau)}, & d > \tau \\ 1, & otherwise \end{cases}$$

The threshold and exponential factor parameters were chosen to match the rule of thumb developed in the original soft-gating study. Let $d_{\max}$ = the maximum S/I motion. We set $\tau = 0.1 d_{\max}$ and $\alpha = \frac{3}{d_{\max}}$.

These weights were then used in the datafit term of the cost function by collecting the weights for each readout line in a real, positively valued, diagonal matrix, $\mathbf{W}$ and including $\mathbf{W}$ as follows:

$$\Psi(f) = \frac{1}{2}||\mathbf{W}\left(y - \mathbf{PTFS}f\right)||_2^2 + \lambda_s \psi\left(\mathbf{B}_s \mathbf{C}_s f\right) + \lambda_t ||\mathbf{B}_t \mathbf{C}_t f||_2^2 \tag{5.89}$$

$$\tag{5.90}$$

This can be viewed as a "soft" alternative to traditional respiratory gating, in which

data outside of the desired respiratory phase is discarded or not acquired. In this soft-gating formulation, the data acquired during other respiratory phases are still used in the reconstruction but required adherence to the forward model is reduced, effectively letting the regularization hold greater sway. Intuitively, this is a way to let data outside of the chosen respiratory phase contribute to the reconstruction, but only partially. This does not attempt to register the data or estimate the relative shift from one respiratory phase to another.

The weighting function is depicted side-by-side with a portion of the navigator signal to show the effect of soft-gating in Figure

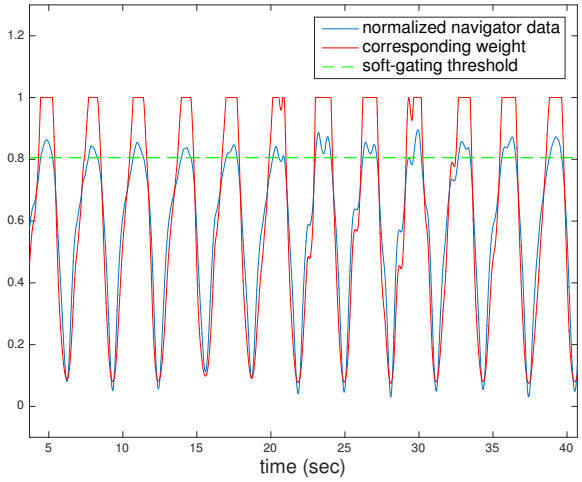


Figure 5.10: Segment of Butterfly navigator signal with normalized navigator data (blue), corresponding weight (red), and threshold τ (green). Soft-gating weighting function gives full weight to samples acquired in the "highest" respiratory state and exponentially decreasing weight otherwise.

5.5.3.2 Results

We solved Equation (5.89) using CG for one slice of the 3D dataset. Figure 5.11 shows the resulting reconstruction when no soft-gating is applied ($\mathbf{W} = \mathbf{I}$) and Figure 5.12 shows the effect of the soft-gating weights in the cost function. Note that the soft-gating slightly attenuates the image values and subdues the early enhancement of the aorta in frame 5.

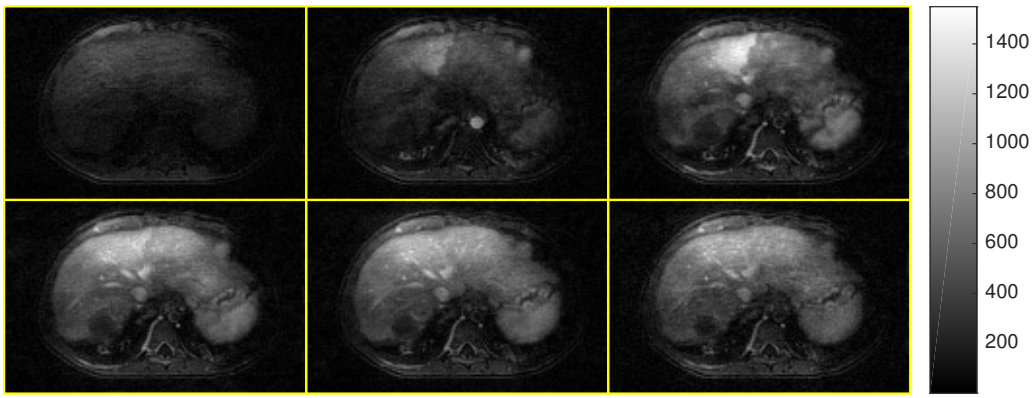


Figure 5.11: Selected temporal basis frames from a spline-based reconstruction of a 2D axial slice. Temporal basis frames shown are $t = 1, 5, 7, 10, 13, 18$ from an $N_t = 18$ reconstruction, corresponding to times $7.6, 38.0, 53.1, 75.0, 98.7, 136.7$ seconds in a 136.7 second experiment.

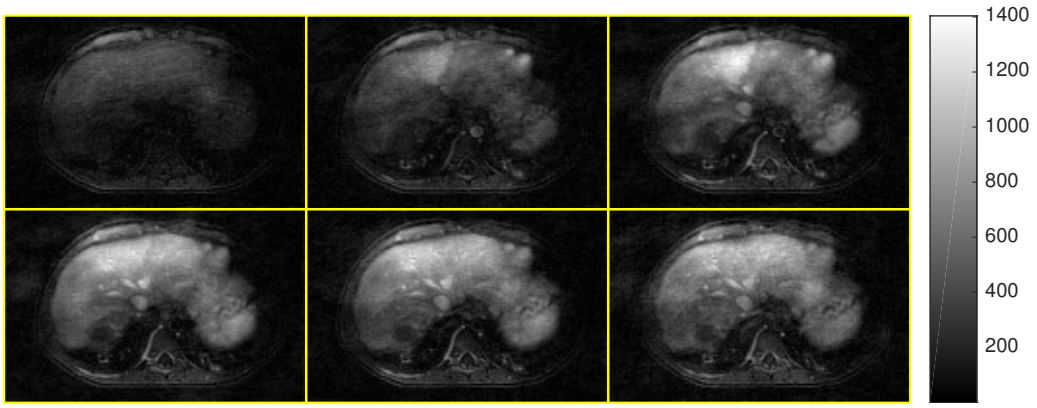
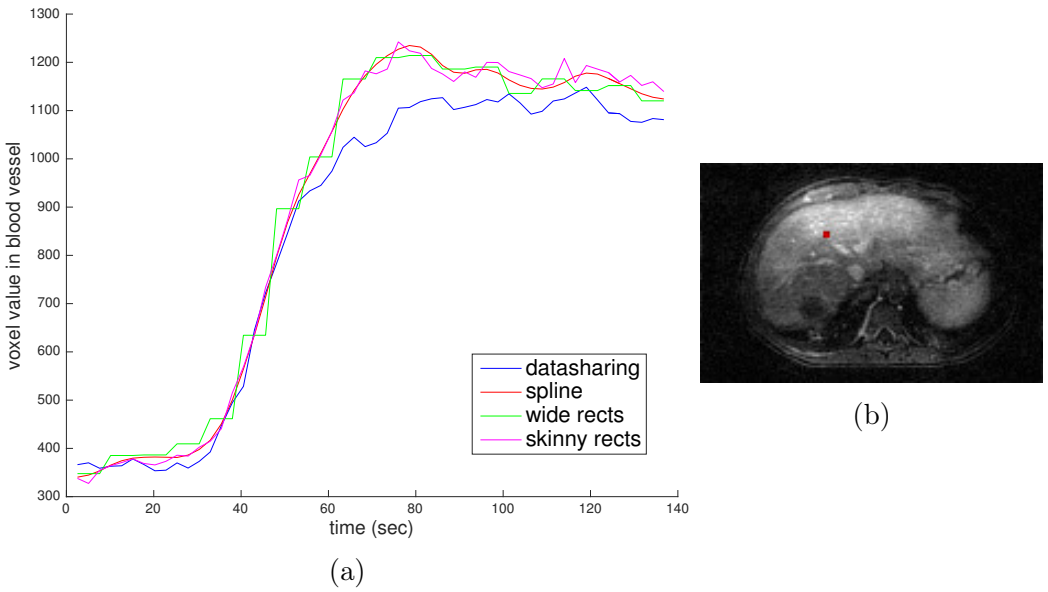


Figure 5.12: Selected temporal basis frames from a spline-based reconstruction of a 2D axial slice, using soft-gating to reduce motion artifacts. Temporal basis frames shown correspond to those in Figure 5.11.



(a)

(b)

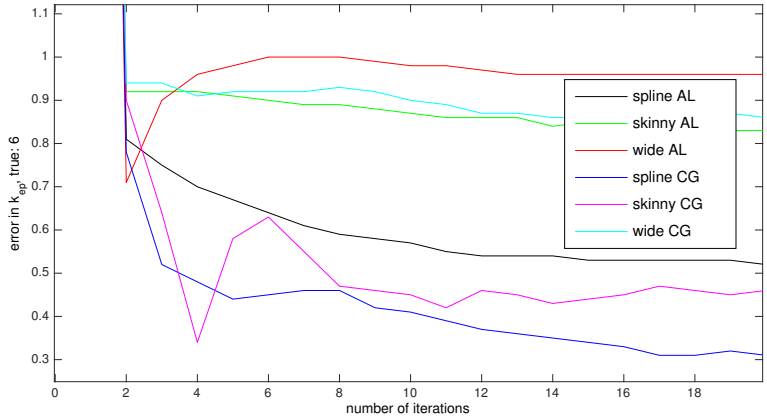
Figure 5.13: (a) Voxel enhancement estimation from different temporal basis models. (b) Voxel chosen belonged to a small blood vessel in the abdomen, marked in red.

As expected from simulation, datasharing is unable to capture the steep rise of the bolus arrival due to indiscriminate sharing across temporal frames. The spline basis

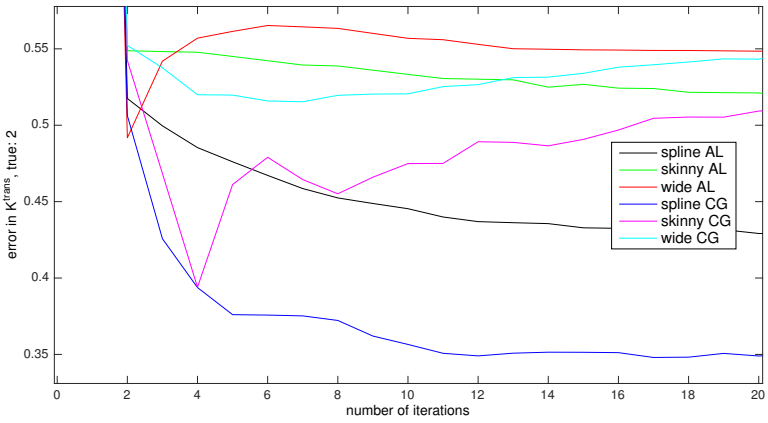
model estimate exhibits a smoother quality. Further experiments with AIF measurements will be needed to assess pharmacokinetic parameter estimation error.

5.5.4 Speed Performance of AL

The current implementation of Spline AL suffers from a poor implemenation of the u_3 update (Equation (5.34)), discussed in Section 5.3.1.2. Figure 5.14 compares the performance of CG to AL for moderate enhancement, measured as error in pharmacokinetic parameters for the rapid enhancement simulation. Note that for comparison, the ℓ_1 norm has been changed to a corner-rounded Fair potential.



(a)



(b)

Figure 5.14: Error in k_{ep} and K^{trans} as a function of iteration for the proposed AL method and CG.

The most problematic variable update step involves the u_3 update, which has a pentadiagonal Hessian. For ease of implementation, we solve the u_3 minimization problem with an inner conjugate gradient step, which takes approximately 94 % of the total computation time for a given outer iteration.

Another avenue of exploration involves investigating hybrid VS methods which retain the benefit of isolating the ℓ_1 spatial regularization term, while batching the complexities of other variable updates into an inner iterative step. This also has the potential advantage of easily incorporating a non-Cartesian trajectory, for which $\mathbf{F}'\mathbf{F}$ is not diagonal.

5.6 Conclusion

This chapter presented a system matrix for an object with a wider, smooth temporal basis, in particular B-splines. For this model, we presented a viable optimization method using variable splitting and the Augmented Lagrangian formulation. Each of the inner variable updates is exact and easy to perform.

Enforcement of an appropriate temporal spline basis improves pharmacokinetic parameter estimation compared to datasharing, and the spline-basis model reaches better estimates more quickly than a dynamic model for short time frames relying only on temporal regularization (skinny rectangles) and reaches better estimates than the dynamic model for longer time frames relying only on temporal regularization (wide rectangles).

5.7 Future Work

Future investigations could include methods for estimating the correct spline basis offset, best spline width and spline type.

The AL methods currently demonstrate slower convergence than the corner-rounded CG solution. Implementation and testing of hybrid VS and CG approaches may yield further insight. Alternatively, parallelism of variable updates may speed up the proposed AL method.

By dividing data into frames of different durations, the existing cardinal spline basis algorithm could be used to emulate irregular spline spacing. This could yield benefits in estimating the sharp bolus arrival phase in the contrast time series.

Another large area for exploration is motion compensation to accomodate breathing and other patient movement so that the initial assumption of smooth temporal changes is applicable. The soft-gating approach is simple and computationally easy but perhaps is less effective than parameterization of multiple respiratory states. These respiratory parameterization methods are explored in Chapter 7.

Chapter 6

Motion Compensation for Extra-Dimensional Abdominal MRI

6.1 Introduction

This chapter proposes a novel motion compensated image reconstruction (MCIR) method for free-breathing, abdominal DCE-MRI. We perform motion-compensation by modeling the reconstructed dynamic image with an extra respiratory dimension and by using motion model-informed regularization across this dimension.

Dynamic contrast-enhanced (DCE) CT and MRI provide useful information in the assessment and optimization of radiation therapy (RT). In addition to providing anatomical information, such as the location of tumors, DCE imaging provides contrast information over time, which characterizes the perfusion and pharmacokinetic behavior of lesions and healthy tissue. The pharmacokinetic information can be used to design an initial radiation therapy session or to assess the effect of a previous round of radiation therapy. Unlike chapter 5, this chapter focuses on liver DCE imaging,

which faces several challenges compared to other body sites.

Radiation-induced liver disease (RILD) is a limiting factor in radiation therapy for cancer, but individual patients exhibit a broad range of radiation sensitivity that is not accommodated by many existing RILD prediction models [97] [98]. Individualized radiation therapy can recommend higher doses to radiation resistant patients and lower doses for patients exhibiting higher sensitivity [99]. This variability elevates the need for precise and accurate DCE imaging in the liver; however, current methods have yet to demonstrate the reproducibility necessary for use in a clinical setting.

Respiratory motion is one of the chief technical challenges for liver DCE MRI. Unlike other organs such as the prostate or breast, the liver is highly affected by respiratory motion, due to its proximity to the diaphragm. This presents a challenge for liver DCE-MRI, because liver contrast changes occur over the course of several minutes, which is significantly longer than most patients can breathhold. Ignoring the effect of motion, which can typically range 20 mm in the z direction [61], results in large motion artifacts which reduce the diagnostic value of DCE-MRI.

One common approach to remove motion artifacts is to perform respiratory gating: the exclusion of data acquired outside of a desired respiratory state. Once can combine gating with breath-holding so that the majority of data is acquired during the desired respiratory state. However, this method presents a major drawback in that important pharmacokinetic changes may occur during the undesired respiratory phase, adversely affecting quantitative pharmacokinetic parameter estimation. An

increasingly popular alternative is to collect and use data at all respiratory states and allow the patient to breath freely and comfortably [100] [5].

Free-breathing approaches have several additional benefits over respiratory gating. By using data from multiple respiratory states, free-breathing methods are more information efficient, collecting more usable data with each MRI readout. Free-breathing data also allows for the modeling of patient-specific respiratory motion, which is of great interest in the radiation therapy community [101] [102].

DCE-MRI is a desirable alternative to DCE-CT due to its superior soft tissue contrast and non-ionizing radiation. However, DCE-MRI is still underutilized in RT planning due to its slower speed. To characterize fast contrast changes, free-breathing DCE-MRI must solve a highly undersampled image reconstruction problem in the presence of motion.

6.1.1 Previous Motion Compensation for Free-Breathing MRI

To move beyond breath-hold, respiratory-gated MRI, respiratory motion must be quantified and/or reduced. Rather than use all data acquired during free-breathing, some methods use data from other respiratory states to a limited degree. Like respiratory-gating methods, the goal of "soft-gating" is to reconstruct a dynamic volume at a single respiratory state, rather than multiple respiratory states. Data collected outside of the desired respiratory state is weighted in the iterative reconstruction method based on its closeness to the desired respiratory state [103].

Other methods seek to estimate the respiratory motion jointly with the dynamic

image volume. To reduce the degrees of freedom in the motion estimation problem, most methods use simplified motion models with several parameters. Rigid motion, including translation and rotation, has been modeled and estimated by [104]. For motion of other organs, [105] used a one-parameter pulsation motion model. Other joint image-motion estimation methods include [50] [51].

Other methods perform non-rigid motion estimation as a pre-processing step and apply the estimated motion in the datafit term of their iterative reconstruction method [106].

Feng et al. introduce an additional respiratory dimension to the dynamic volume to decouple the effects of contrast over time and respiratory motion in [107], which directly inspired this work.

6.1.2 Extra-Dimensional Dynamic MRI

We denote an unknown 5D respiratory-contrast object as $f_{5\mathrm{D}}(\vec{r}, t, \phi)$, where t corresponds to the scan time over the experiment, indexed from $1, \ldots, N_{\mathrm{t}}$. N_{t} is the number of frames, chosen to provide a sampling rate adequate for later pharmacokinetic parameter estimation. More specifically, each of the N_{t} frames encompasses many acquired lines of k-space, and the frame duration is much longer than repetition time, T_R. Voxel index $\vec{r} = 1, \ldots, N_{\mathrm{r}}$, and respiratory state index $\phi = 1, \ldots, N_{\mathrm{resp}}$. Cost functions for the 5D MRI problem have the form:

$$J(f_{5\mathrm{D}}) = \frac{1}{2} \left\| y - \mathbf{F}_{5\mathrm{D}} \mathbf{S}_{5\mathrm{D}} f_{5\mathrm{D}} \right\|^2 + R(f_{5\mathrm{D}}). \tag{6.1}$$

The data vector $y \in \mathbb{C}^{N_{\text{samp}} \times N_c}$ holds the noisy, under-sampled, free-breathing, multi-coil k-space measurements from each of N_{c} coils, f_{5D} is size $N_{\text{r}} \times Nt \times N_{\text{resp}}$, and $R(\cdot)$ represents regularization across some combination of spatial, respiratory, and temporal dimensions. Two challenges are defining an appropriate regularizer $R(\cdot)$ and computing the minimizer of the cost function $J(\cdot)$ to estimate f_{5D}.

Fourier encoding matrix $\mathbf{F}_{\text{5D}}$ includes subsampling for each respiratory phase. In this 5D case, the subsampling is far more severe than in a conventional dynamic image reconstruction due to the addition of the fifth respiratory dimension. More specifically,

$$\mathbf{F}_{\text{5D}} = \mathbf{I}_{N_{\text{c}} N_{\text{t}}} \otimes \begin{bmatrix} \tilde{\mathbf{F}}_1 & & \\ & \ddots & \\ & & \tilde{\mathbf{F}}_{N_{\text{resp}}} \end{bmatrix}, \tag{6.2}$$

for a total size of $N_{\text{samp}} N_{\text{c}} \times N_{\text{r}} N_{\text{resp}} N_{\text{c}} N_{\text{t}}$. Likewise, $\mathbf{S}_{\text{5D}}$ is also larger than $\mathbf{S}_{\text{4D}}$ by a factor of N_{t}: $\mathbf{S}_{\text{5D}} = \mathbf{I}_{N_{\text{resp}} N_{\text{t}}} \otimes \tilde{\mathbf{S}}$, because the sensitivity maps $\mathbf{S}_{\text{4D}}$ apply to all motion phases. We assume the sensitivity maps were estimated over a large enough support to accommodate respiratory motion; We used the regularized method in [18].

Since this is such an undersampled problem, regularization is essential to improve the conditioning of the problem. Previous works addressing 5D image reconstruction have used finite-difference regularization across spatial, temporal, and respiratory dimensions independently [107] [108], as follows:

$$R_s(f_{5\mathrm{D}}) = \lambda_s \sum_{t=1}^{N_\mathrm{t}} \sum_{\phi=1}^{N_\mathrm{resp}} \sum_{\vec{r}}^{N_r} \sum_{\vec{r}' \in \mathcal{N}_r} \psi\left(f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}(\vec{r} - \vec{r}', t, \phi)\right) \tag{6.3}$$

$$R_t(f_{5\mathrm{D}}) = \lambda_t \sum_{t=2}^{N_\mathrm{t}} \sum_{\phi=1}^{N_\mathrm{resp}} \sum_{\vec{r}}^{N_r} \psi\left(f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}(\vec{r}, t-1, \phi)\right) \tag{6.4}$$

$$R_r(f_{5\mathrm{D}}) = \lambda_r \sum_{t=1}^{N_\mathrm{t}} \sum_{\phi=1}^{N_\mathrm{resp}} \sum_{\vec{r}}^{N_r} \psi\left(f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}(\vec{r}, t, \phi-1)\right) \tag{6.5}$$

This work proposes an alternative regularizer for the respiratory dimension designed to increase sparsity and reduce motion artifacts. Rather than penalize differences across respiratory states, the motion model-informed regularizer penalizes the motion compensated differences between respiratory states of $f_{5\mathrm{D}}$.

6.2 Motion Model Informed Regularization

6.2.1 Motion Modeling from Pre-Contrast Data

To develop a motion model, we first estimate a respiratory object, $f_\mathrm{r}(\vec{r}, \phi)$, that describes the anatomical volume at each respiratory phase ϕ. For this recontruction we simply pool together pre-contrast data from each respiratory state and perform density-compensated gridding.

The set of spatial transformations that register the N_resp respiratory phases to the reference respiratory phase constitute the motion model. We use this motion model

to regularize post-contrast data in the 5D MRI problem. Let W_{ϕ_0,ϕ_1} describe the transformation of spatial coordinates from one respiratory phase, ϕ_1, to another, ϕ_0, as follows:

$$f_{\mathrm{r}}(W_{\phi_0,\phi_1}(\vec{r}),\phi_1) \approx f_{\mathrm{r}}(\vec{r},\phi_0). \tag{6.6}$$

More specifically, we calculated the transformations by pair-wise registration of neighboring respiratory phases of $\hat{f}_r$. In anticipation of motion models that may be one-to-one but not onto, thereby ignoring some voxel values, we also include the reverse direction warp in our regularization term:

$$f_{\mathrm{r}}(W_{\phi_1,\phi_0}(\vec{r}),\phi_0) \approx f_{\mathrm{r}}(\vec{r},\phi_1). \tag{6.7}$$

Including both directions can also improve motion model-informed regularization in cases when $f_{\mathrm{r}}(\vec{r},\phi_i)$ is difficult to approximate from $f_{\mathrm{r}}(W_{\phi_i,\phi_{i-1}}(\vec{r}),\phi_{i-1})$ but is better approximated by its other neighboring respiratory state $f_{\mathrm{r}}(W_{\phi_i,\phi_{i+1}}(\vec{r}),\phi_{i+1})$.

Once precomputed, we can apply these transformations efficiently to the larger 5D MRI problem.

This motion model can describe non-rigid motion and is more general than [105]. This motion model is designed for respiratory motion, which is largely along the S/I

direction, as opposed to the models proposed in [109]. Unlike other patient-specific respiratory motion models, this approach does not require additional data such as optical surface imaging [110]. However this work is limited in only representing S/I motion.

6.2.2 Motion Model-Informed Regularization for Dynamic Imaging

We investigated applying these warpings to the estimation of 5D object, $f_{5\mathrm{D}}(\vec{r}, t, \phi_i)$, via motion model-informed regularization as follows:

$$R_{\mathrm{mm}}(f_{5\mathrm{D}}) = \lambda_{\mathrm{mm}} \sum_{t=1}^{N_{\mathrm{t}}} \sum_{\vec{r}=\vec{0}}^{N_r} \sum_{\phi=1}^{N_{\mathrm{resp}}} \psi\left(f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}\left(W_{\phi,\phi-1}(\vec{r}), t, \phi - 1\right)\right) + \psi\left(f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}\left(W_{\phi,\phi+1}(\vec{r}), t, \phi + 1\right)\right). \quad (6.8)$$

Combining $R_{\mathrm{mm}}(f_{5\mathrm{D}})$ with conventional spatial and temporal regularization in (6.3)

and (6.4), we have:

$$
\begin{aligned}
J(f_{5\mathrm{D}}) = \frac{1}{2}\left\|y - \mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}P(f_{5\mathrm{D}})\right\|^2 \\
+ \lambda_s \sum_{t=1}^{N_\mathrm{t}} \sum_{\phi=1}^{N_\mathrm{resp}} \sum_{\vec{r}} \sum_{\vec{r}'\in\mathcal{N}_r} \psi_s\left(f_{5\mathrm{D}}(\vec{r},t,\phi) - f_{5\mathrm{D}}(\vec{r}-\vec{r}',t,\phi)\right) \\
+ \lambda_t \sum_{t=2}^{N_\mathrm{t}} \sum_{\phi=1}^{N_\mathrm{resp}} \sum_{\vec{r}} \psi_t\left(f_{5\mathrm{D}}(\vec{r},t,\phi) - f_{5\mathrm{D}}(\vec{r},t-1,\phi)\right) \\
+ \lambda_{\mathrm{mm}} \sum_{t=1}^{N_\mathrm{t}} \sum_{\vec{r}=\vec{0}}^{N_r} \sum_{\phi=1}^{N_\mathrm{resp}} \psi_m\left(f_{5\mathrm{D}}(\vec{r},t,\phi) - f_{5\mathrm{D}}\left(W_{\phi,\phi-1}(\vec{r}),t,\phi-1\right)\right) \qquad (6.9)
\end{aligned}
$$

$$
\hat{f_{5\mathrm{D}}} = \underset{f_{5\mathrm{D}}}{\operatorname{argmin}}\, J(f_{5\mathrm{D}}) \qquad (6.10)
$$

Compared to previous 5D MRI work, (6.10) is better conditioned. A priori knowledge of the motion model relates each of the respiratory phases to each other. In the extreme case of perfect motion compensation, the reconstruction problem reduces to estimation of a 4D dynamic volume, for a N_resp-fold reduction of unknowns.

The incorporation of motion compensation in the regularization rather than the datafit term is the chief novel contribution of this work. Many previous motion compensation methods use their motion models to correct the data or estimated image in the datafit term, which may present significant drawbacks for inaccurately modeled motion. By indirectly encouraging conformity to the motion model via regularization, the formulation in (6.10) may be more robust to poor motion models.

Another benefit of motion compensating only in the regularization term is that this approach may be more amenable to reliable pharmacokinetic parameter estimation

than other methods for accommodating non-rigid motion. Pharmacokinetic parameter estimation depends on the amount of signal measured at specific voxels, and irregularity of deformable fields causes concerns [111].

6.2.3 Discretized Warping

To compute motion model-informed regularization, we define a new linear operator $\tilde{\mathbf{W}} : \mathbb{C}^{N_r \times N_\mathrm{t} \times N_\mathrm{resp}} \mapsto \mathbb{C}^{N_r \times N_\mathrm{t} \times N_\mathrm{resp}}$, that incorporates the interpolation and coordinate transform from each respiratory phase to a reference respiratory phase. The cost function in (6.10) becomes:

$$J(f_{5\mathrm{D}}) = \frac{1}{2} \left\| y - \mathbf{F}_{5\mathrm{D}} \mathbf{S}_{5\mathrm{D}} f_{5\mathrm{D}} \right\|^2 + \psi \left(\mathbf{L} \underbrace{\begin{bmatrix} \mathbf{C}_s \\ \mathbf{C}_t \end{bmatrix}}_{\tilde{\mathbf{C}}} f_{5\mathrm{D}} \right) + \lambda_m \psi_m \left(\mathbf{C}_r \tilde{\mathbf{W}} f_{5\mathrm{D}} \right) \qquad (6.11)$$

$\tilde{\mathbf{W}} f_{5\mathrm{D}}$ computes differences between respiratory states *after motion compensation.* For this work, we chose the ℓ_1 norm for potential functions ψ and ψ_m.

6.2.4 Registration Across Respiratory States

In section 6.2.3 we define the warping $\tilde{\mathbf{W}}$ in terms of a precomputed registration between neighboring respiratory states. In this section we describe the simple, patch-based method for non-rigid registration used in the following simulation and *in vivo*

experiments.

Myriad registration methods exist for medical imaging that account for multiple imaging modalities/contrasts or biomechanical models. In this work, we model respiratory motion only between neighboring respiratory states in the same dynamic T_1-weighted MRI sequence.

Because the majority of respiratory motion occurs along the S/I axis, we represent respiratory motion as shifts along one dimension, z. For each voxel in a given respiratory state of f_{r}, we examine the surrounding patch and compare it to candidate patches in another respiratory state. Let $P_{\mathrm{x,y,z}}$ denote the extraction operator for patches centered at x, y, z. Then we choose ζ, the displacement in z from respiratory state ϕ to $\phi - 1$, via:

$$d_{\phi,\phi-1}(x,y,z) = \underset{\zeta}{\operatorname{argmin}}\, w(\zeta)\Psi\left(P_{\mathrm{x,y,z}} f_{\mathrm{r}}(x,y,z,\phi), P_{\mathrm{x,y,z+}\zeta} f_{\mathrm{r}}(x,y,z,\phi-1)\right) \quad (6.12)$$

Ψ is a general comparison metric and w is a weighting to encourage smaller displacements. We used cross-correlation in z for Ψ. We also use the minimum value achieved by d as a metric for confidence in the patch match. We can use this confidence value in a weighting matrix for the regularization term.

We use these measured displacements to construct linear warping operator $\tilde{\mathbf{W}}$. One limitation of this patch-based regularization method is that the resulting warping is one-to-one but not onto. To avoid leaving any elements of f_{5D} unregularized, we

also construct and penalize the ℓ_1 norm of $\tilde{\mathbf{W}}_{\mathrm{r}}$ from reverse measured displacement $d_{\phi-1,\phi}$.

Before using this warping in iterative reconstruction, we demonstrate that $\mathbf{C}_r\tilde{\mathbf{W}}f_{\mathrm{r}}$ is sparser than $\mathbf{C}_r f_{\mathrm{r}}$.

This method is simple to implement and apply, but it can result in unrealistic motion models. In particular, regions of sharp upward movement neighboring regions of sharp downward movement may represent sliding organs, but represent implausible motion in other anatomical areas. Again, a benefit of the regularization approach over modifying the datafit term is possible improved robustness to such motion model errors, particularly if confidence weighting were incorporated into the regularization term.

6.3 Simulations

To demonstrate the benefits of motion model-informed regularization, we first apply it to simulated free-breathing DCE-MRI data. We simulate data of the abdomen by using an adapted XCAT phantom. Originally developed for X-Ray CT, XCAT can model respiratory motion and contrast dynamics independently.[112] [113]. We reconstructed one coronal slice of a 3D volume to demonstrate the benefit of modeling simple S/I motion in the regularizer.

6.3.0.1 PINCAT anatomical phantom

To simulate MRI data, we changed organ and tissue attenuation values to mimic T1-contrast MR images following PINCAT [114]. We added smoothly varying phase, shown in Figure 6.3 to the magnitude-only PINCAT images. We also cropped the image to size 100×256 pixels to reduce computation.

Figure 6.1 shows the magnitude of the PINCAT image at the inhale respiratory state across four selected frames. The only change across the frames is the contrast value of the liver, seen in the bottom left corner of the image. The dynamic sequence spans 91 seconds total across 20 frames, for a temporal resolution of 4.5 seconds. Figure 6.2 shows the PINCAT image across four respiratory states at a given time in contrast dynamics. In-plane movement of the liver and kidneys is noticeable, particularly between the first two respiratory states. Through-plane motion is also visible among other organs and blood vessels in the liver.

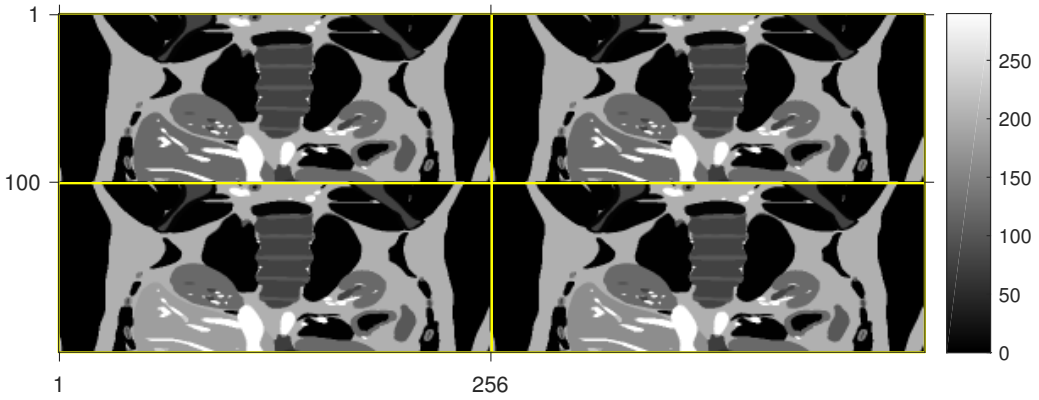


Figure 6.1: Selected frames of PINCAT at inhale state.

We simulated k-space data by applying simulated sensitivity maps and applying 40

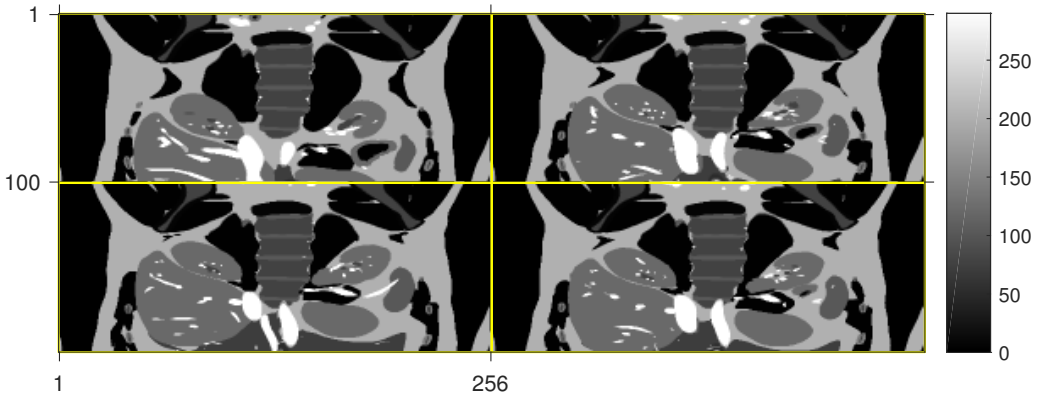


Figure 6.2: Four respiratory states of PINCAT at first temporal frame.

Figure 6.3: Smoothly-varying phase applied to each temporal frame and respiratory state of PINCAT.

dB SNR complex white Gaussian noise. The eight simulated sensitivity coils are arranged in two rings around the abdomen of the patient, similar to the *in vivo* experiment setup, and their sensitivity profiles are shown in Figure 6.4.

6.3.1 Variable Density Cartesian Sampling

We used a 1D variable density Cartesian sampling pattern, in which the fully-sampled readouts run in the S/I direction (akin to full k_z sampling in GRASP stack-of-stars), and 10% of k-space is sampled at each frame. The center eight readouts are sampled in each frame to ensure good image quality. Each of these undersampled slices should

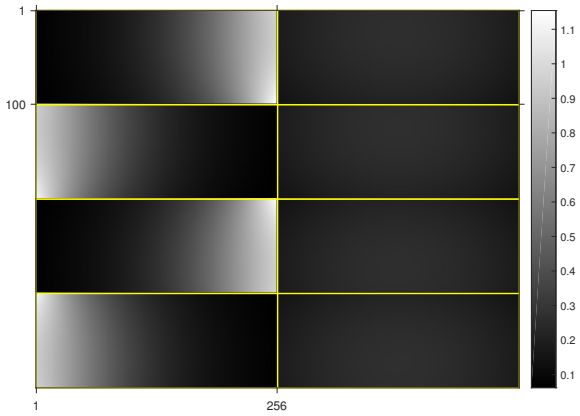


Figure 6.4: Simulated sensitivity maps for PINCAT experiment. The even indexed coils are located out of plane and show little in-plane variation.

be considered one coronal slice of a full 3D acquisition, in which 2D undersampling is applied within the $k_x - k_y$ plane with fully sampled readouts along z. We sampled and reconstructed one coronal slice for simplicity.

An example of a typical sampling pattern for one frame is shown in Figure 6.5. We assigned each of these frames to one of four respiratory states based on the Butterfly navigator for an *in vivo* abdominal DCE-MRI experiment [6]. The binning scheme for the samples is shown in Figure 6.6.

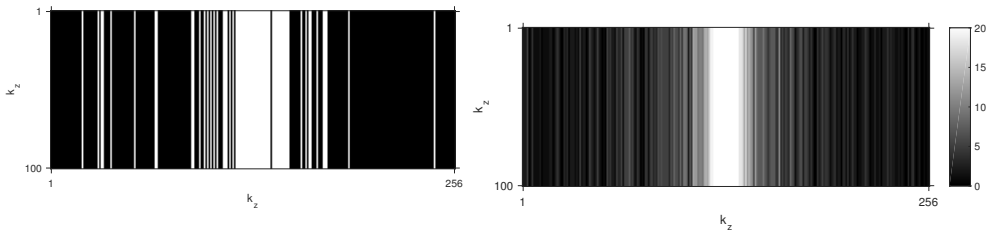


Figure 6.5: (a) 1D variable density sampling pattern for one frame (with central 8 readouts always collected); (b) sampling total for each readout, cumulative over all frames, showing a variable density structure.

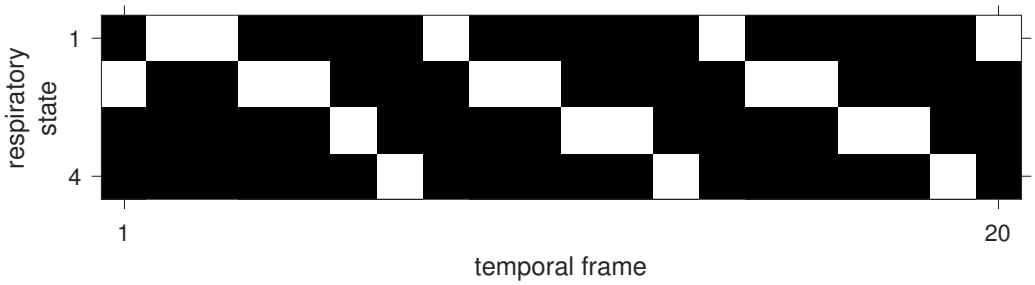


Figure 6.6: 1D variable density sampling pattern for one frame.

6.3.2 Motion Modeling from Patch-Based Regularization

Before performing the iterative motion model-informed reconstruction, we estimated respiratory motion from a oracle respiratory object using the patch-based regularization descirbed in Section 6.2.4. The estimated S/I motion between neighboring respiratory states is shown in Figure 6.7.

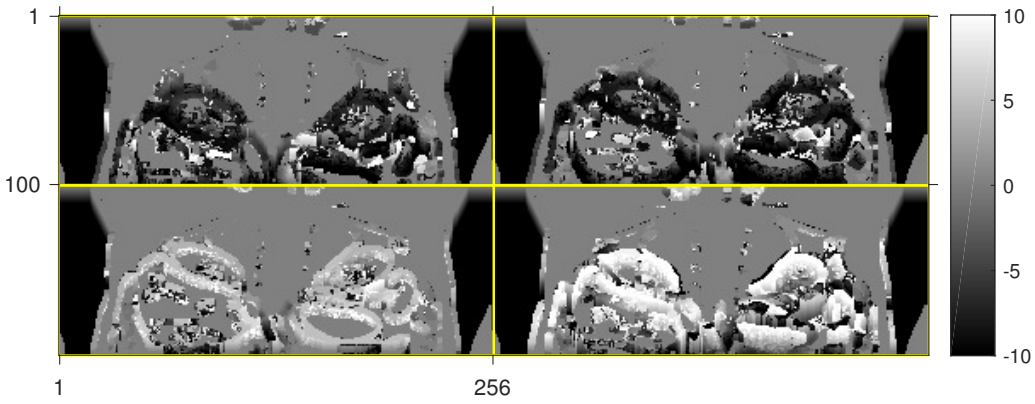


Figure 6.7: Motion model mapping each respiratory state to the following neighbor. Negative values indicate pixels from the next respiratory state are moved downward, and positive values indicate motion upwards.

We first examined the utility of this rudimentary motion model by applying it to one frame of the true 5D object. Figure 6.8 shows each warping applied to each

respiratory state of the frame, $f_{5\mathrm{D}}\left(W_{\phi,\phi+1}(\vec{r}), t, \phi+1\right)$.

The differences between this motion-compensated frame and the original frame, $f_{5\mathrm{D}}(\vec{r}, t, \phi) - f_{5\mathrm{D}}\left(W_{\phi,\phi+1}(\vec{r}), t, \phi+1\right)$ is also shown in Figure 6.8. These differences are much sparser than the direct differences across respiratory states.

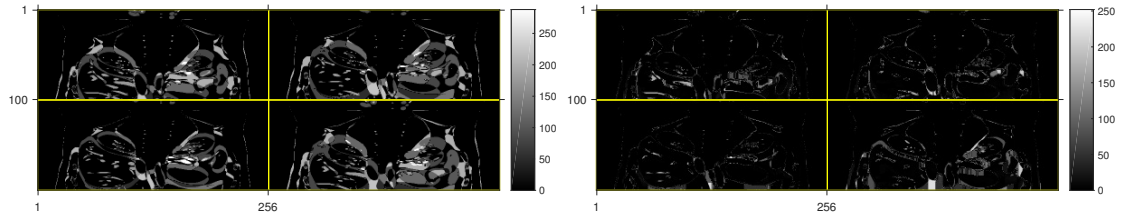


Figure 6.8: (a) differences across respiratory states of PINCAT object; (b) differences across respiratory states of PINCAT object *after applying motion compensation*, i.e., argument of $\psi(\cdot)$ in (6.8).

6.3.3 Reconstruction Results

To assess the benefit of motion model-informed regularization, we chose two error metrics: normalized root mean squared error (NRMSE) compared to the true 5D PINCAT phantom, and average NRMSE over the liver region compared to the true PINCAT phantom. We solved (6.10) using the motion model-informed regularization in (6.11) (x_{MMI}) and compared it to the direct finite differences across respiratory states used in XD-GRASP [107] (x_{XDG}). We solved 6.11 using a conjugate gradient optimization algorithm for 250 iterations. To do so, we applied corner rounding to the ℓ_1 norm in (6.11) using the Huber function with $\delta = 1$, compared to a maximum image value of 290. Each method was initialized with a dynamic image, constructed from a zero-filled IFFT) duplicated over four respiratory states.

We enforced the same regularization parameter across the three spatial dimensions, leaving three regularization parameters: λ_s, λ_t, and λ_r for each reconstruction problem. We tuned these parameters by selecting the combination from a grid search that yielded the lowest NRMSE for each reconstruction problem. As expected, XD-GRASP achieved lower error with stronger temporal regularization and weaker respiratory regularization than the proposed motion model-informed method.

Figure 6.9 shows the reconstructed 5D images at frame $N_{\mathrm{t}} = 1$. Their respective difference images compared to the true PINCAT phantom are in Figure 6.10. Due to the sampling pattern shown in Figure 6.6, k-space data is available for the second respiratory state $N_{\mathrm{resp}} = 2$ for $N_{\mathrm{t}} = 1$, which shows few motion artifacts. Similarly, the first respiratory state shows low error because the following frame $N_{\mathrm{t}} = 2$ was sampled during the first respiratory state. The proximity to data in Figure 6.9 correlates to high image quality for a temporal-respiratory coordinate. The other respiratory states rely more heavily on temporal and respiratory regularization to fill in the missing data and result in higher error. Figure 6.10 shows that the motion model-informed regularization improves error in respiratory states 3 and 4, most noticeably near the kidneys and left side of the patient. NRMSE over the entire 5D object for x_{MMI} is 0.076, a 31% decrease over an NRMSE of 0.11 for x_{XDG}.

We can also verify that the reconstructed f_{5D} objects feature an increase in sparsity with the proposed regularizer, compared to simple finite-differences, as shown for the phantom in Figure 6.8. Figure 6.11 shows the differences across respiratory states with and without warping for x_{MMI} and x_{XDG}.

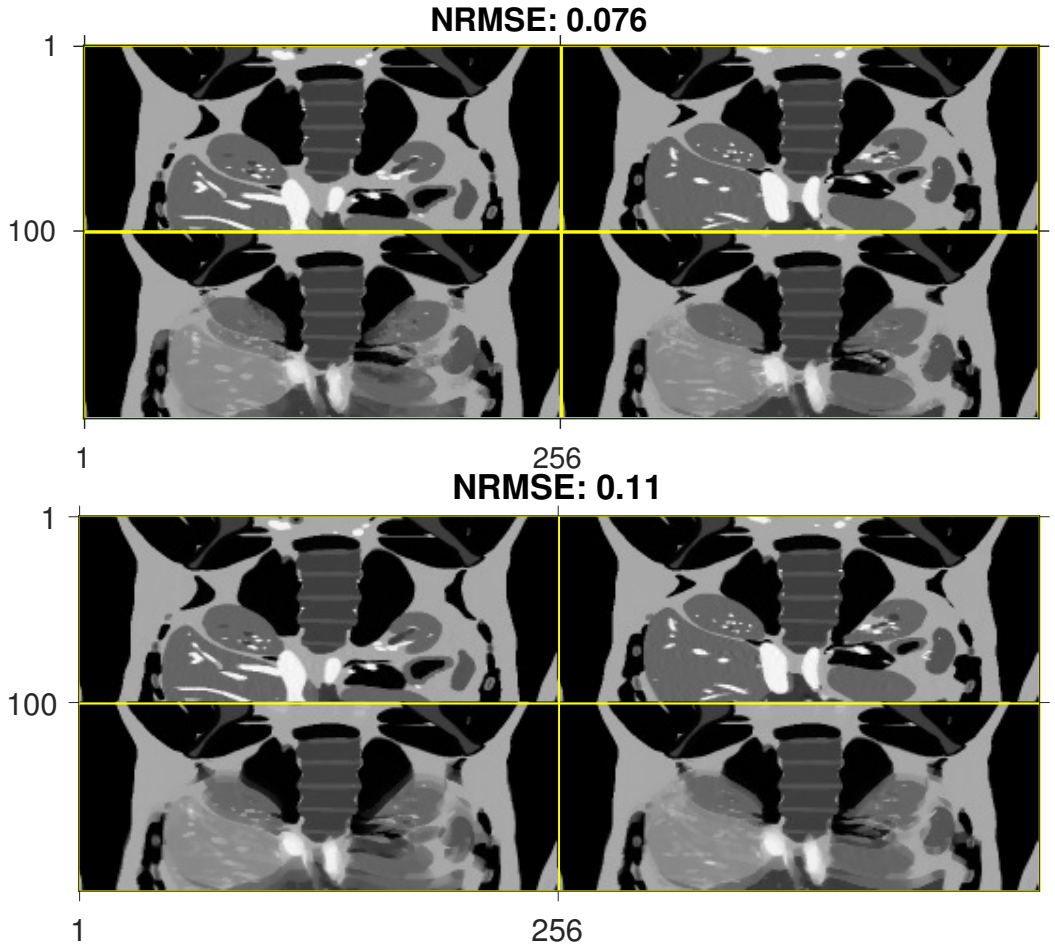


Figure 6.9: Selection of frame $N_\mathrm{t} = 1$ for (a) x_MMI with motion model-informed regularization (b) x_XDG constructed via XD-GRASP. The four modeled respiratory states $N_\mathrm{resp} = 1, \ldots, 4$ are shown in lexicographic order, starting from the top left.

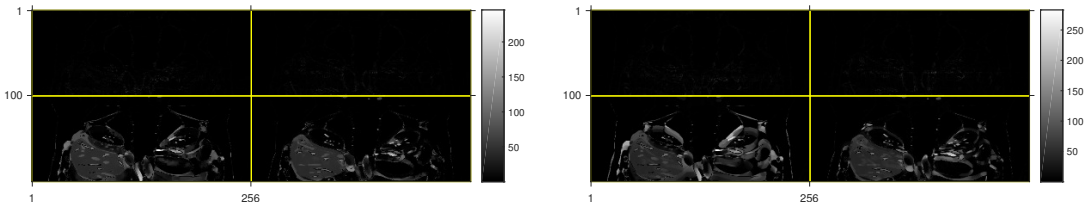


Figure 6.10: Error between f_5D reconstructions at $N_\mathrm{t} = 1$ and true PINCAT phantom for (a)x_MMI (b) x_XDG. The four modeled respiratory states $N_\mathrm{resp} = 1, \ldots, 4$ are shown in lexicographic order, starting from the top left.

x

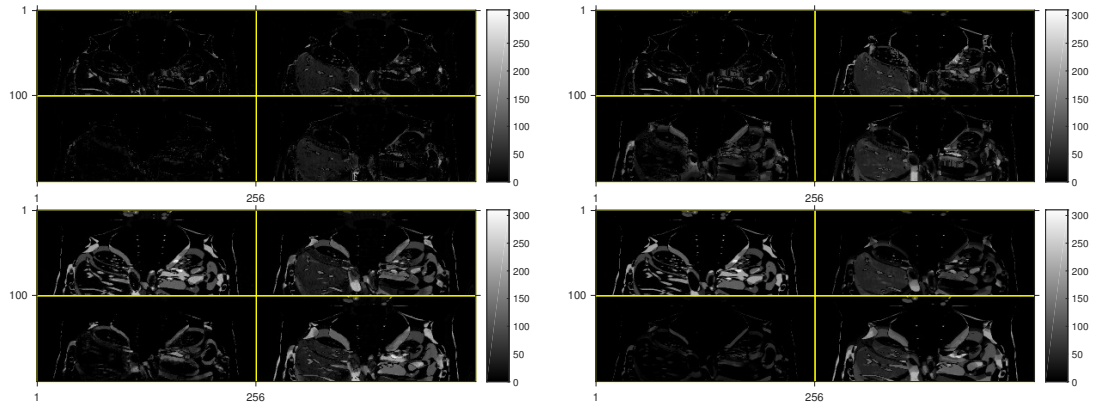
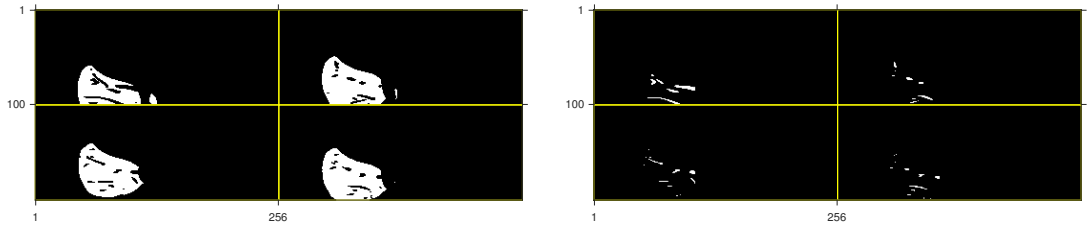


Figure 6.11: Differences across respiratory dimension for f_{5D} reconstructions at $N_t = 1$ for (a) x_{MMI} with warping, (b) x_{XDG} with warping, (c) x_{MMI}, without warping, (d) x_{XDG} without warping. The inclusion of warping results in sparser difference images for both reconstructions.

Figure 6.11 also demonstrates limitations of the simple patch-based regularization method described in Section 6.2.4. The motion model only captures in-plane motion for this coronal slice, but the anatomical PINCAT phantom includes motion and deformation through the coronal plane. This is visible at multiple locations in the slice, including the bright lobes at the bottom center of the image, the portions of blood vessels in the liver, and the left kidney. The patch-based registration method is most effective at capturing motion of the kidneys and ribs.

To assess the usefulness of motion model-informed registration for later pharmacokinetic parameter estimation, we also compared the average MR contrast in the liver over time for x_{MMI} and x_{XDG}. Furthermore, we also estimated the average MR contrast in blood vessels in the liver over time. Due to the very small size of the blood vessels, this quantity will be more susceptible to respiratory motion artifacts. To measure these quantities, we estimated from the PINCAT phantom a voxel mask



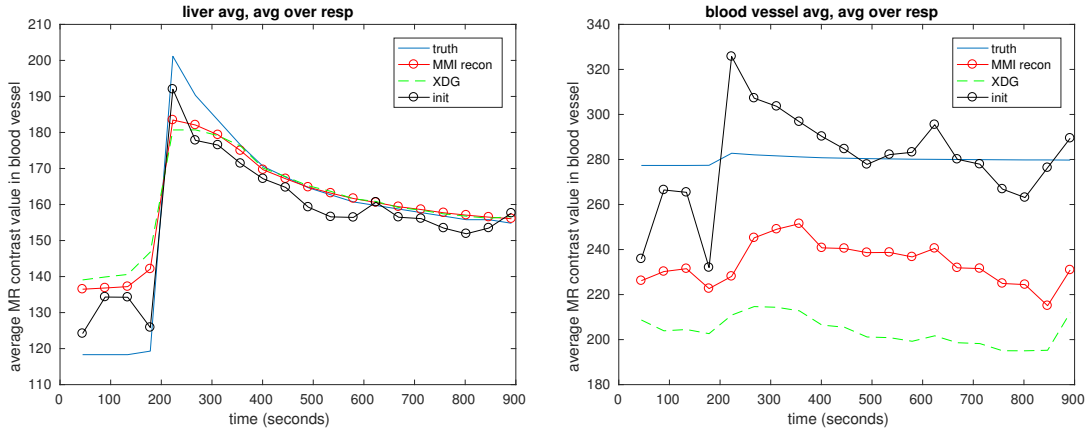
(a) Liver mask (b) Blood vessel mask

Figure 6.12: Masks used at each respiratory state to measure average ROI MR contrast.

corresponding to the liver tissue and the blood vessels in the liver, shown in Figure 6.12.

Figure 6.13 shows the effect of motion model-informed regularization on MR contrast time series for specific anatomical areas. x_{MMI} demonstrates a small improvement over x_{XDG} for the average liver contrast. This result is consistent with the finding that most of the error reduction in x_{MMI} compared to x_{XDG} was in other anatomical areas, as shown in Figure 6.10. The motion-sensitive blood vessel contrast values show a more marked improvement for the proposed method. Because the blood vessel contrast values are consistently higher than the background liver contrast values at every frame in the PINCAT phantom, respiratory motion artifacts manifest as low contrast values within the blood vessels, a result of blurred liver and blood vessel values. x_{MMI} mitigates some of the lowered blood vessel contrast values found in x_{XDG}. This is consistent with the frame shown in Figure 6.9, in which blood vessels are more defined in respiratory states 3 and 4 for x_{MMI} than for x_{XDG}.

These simulations demonstrate that motion model-informed regularization can improve image quality and reduce motion artifacts for 5D MRI reconstruction. To



(a) Average liver contrast over time for true PINCAT phantom, x_{MMI}, x_{XDG}, and initial x.

(b) Average blood vessel contrast over time for true PINCAT phantom, x_{MMI}, x_{XDG}, and initial x.

Figure 6.13

assess the use of motion model-informed regularization in a less controlled setting, we turn to *in vivo* patient data.

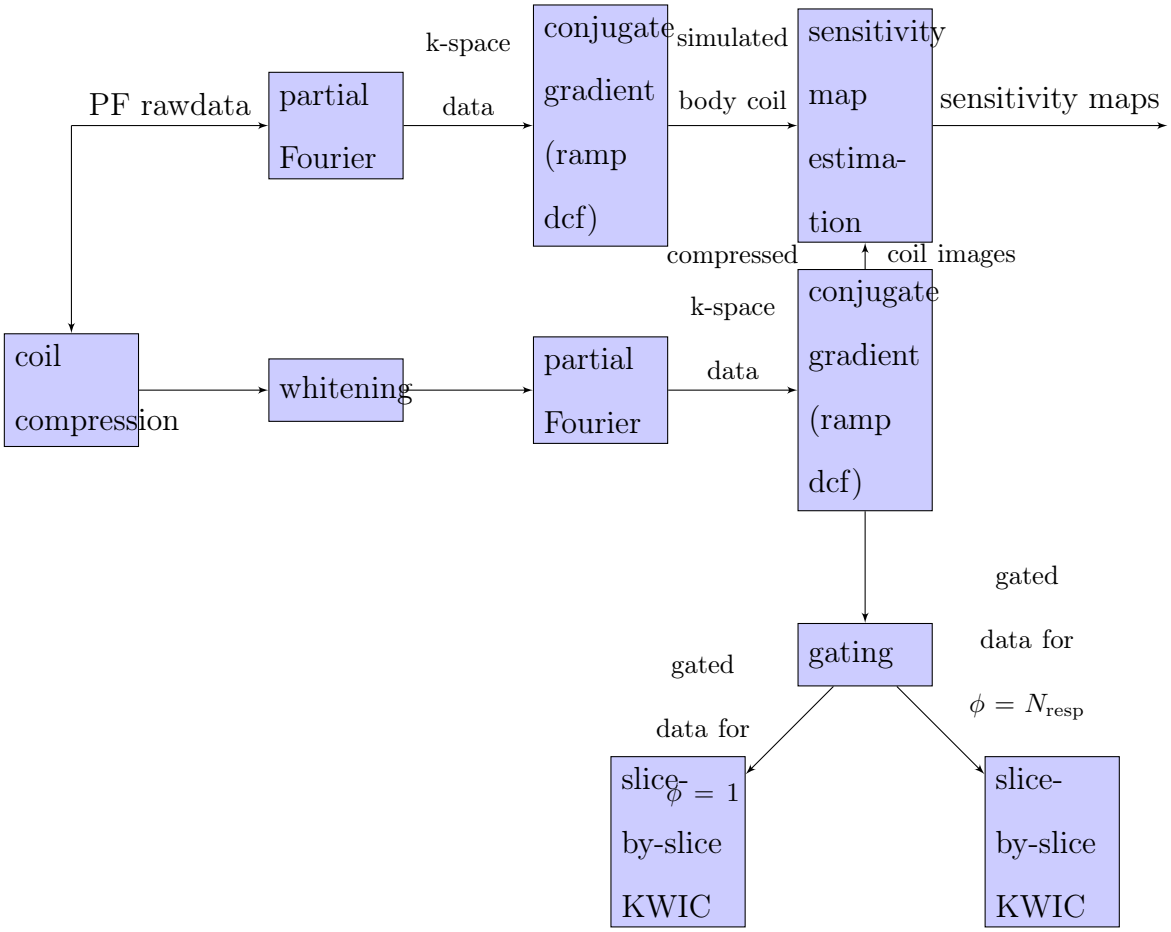
6.4 Patient Data Experiments

We also applied motion model-informed regularization to *in vivo* data. Under an IRB protocol, we acquired patient data on a 3T Siemens Skyra MRI. For dataset 1, the total scan time was 299 seconds. After a short period of free-breathing golden angle acquisition, the patient was injected with a gadolinium contrast agent for 30 seconds. In total, 2000 golden angle radial spokes across 46 Cartesian axial slices were acquired on 16 channels, with 384 samples and a T_R of 2.72 ms per readout. Of the 16 channels, 4 were embedded in the patient table, and 12 were placed across the chest of the patient.

The k-space data corresponds to a volume size of $192 \times 192 \times 64$ voxels of size $2.6 \times 2.6 \times 3.5$ mm. Note that the acquired k-space data describes only partial Fourier data along z.

Compared to the simulation, the patient experiments are affected by a number of experimental factors that make motion modeling and image reconstruction more challenging. These challenges fall into three categories: estimating accurate k-space sampling locations, estimating sensitivity maps, and estimating the respiratory navigator.

Figure 6.4 summarizes the pre-processing required for patient data reconstruction.



6.4.1 Preprocessing

6.4.1.1 Gradient Delay

These patient experiments used a golden angle radial sampling scheme as in [107], which has several desirable properties. The stack-of-stars sampling scheme provides information at regular intervals, around 8 center-of-k-space measurements per sec-

ond. The Cartesian sampling in z allows for decoupled slice-by-slice processing in the absence of S/I motion compensation. Golden angle radial sampling also demonstrates robustness to motion artifacts and pseudo-random angular sampling. However, radial acquisitions commonly suffer from gradient delays, causing a discrepancy between the desired and actual sampling location in k-space.

For a radial spoke at angle ϕ, the ideal k-space sampling locations (using complex plane coordinates) are:

$$S_\phi(k) = \Delta_k k e^{i\pi\phi}, \qquad k = -\frac{N_{ro}}{2}, \ldots, \frac{N_{ro}}{2} - 1, \tag{6.13}$$

where N_{ro} is the number of samples in each readout, and Δ_k is the spacing between each sample.

We choose to focus on estimating the discrepancy along the readout direction, rather than on inaccuracies in the spoke angle. This choice avoids the additional hassle of additional calibration scans, such as [115], allowing us to estimate gradient delays from the data itself. The additional offset parameter g_ϕ affects the true k-space sampling locations as follows:

$$S_\phi(k) = \Delta_k (k + g_\phi) e^{i\pi\phi}, \qquad k = -\frac{N_{ro}}{2}, \ldots, \frac{N_{ro}}{2} - 1, \tag{6.14}$$

We adapted the method presented in [116] for regular radial sampling to the non-repeating golden angle radial trajectory. This method considers two readouts in opposite directions to be simultaneous samples of k-space, and estimates a phase shift of the Fourier Transform of each readout. This phase shift translates into a spatial shift in k-space.

Because the golden angle is irrational, the trajectory never repeats the same angle ϕ nor its opposite, $\phi + \pi$. Instead, we compare approximately opposite spokes by comparing each spoke with its closest opposite. This approach has several drawbacks. If we choose to limit the spokes used for gradient delay calibration to the pre-contrast period, then we can avoid comparing two "opposite" spokes that exist in different phases of contrast. However, limiting the calibration to pre-contrast spokes means selecting from a smaller pool of candidate "opposite" spokes, and a greater discrepancy in the angle of the each pair of spokes. We use this method over all spokes for each channel and for each slice in z. However, our patient data shows that there is also a significant discrepancy in k-space not described by estimated g_k.

Figure 6.14 shows sorted, selected k-space for four selected patient datasets. We examine the square-root of the sum-of-squares across channels for raw k-space data for the center slice. We reduce the number of spokes shown for interpretability, selecting approximately one in four spokes across the entire scan, including before and after contrast injection. The figure shows k-space data across the readout dimension and across sorted spoke angle.

Bright bars indicate spokes acquired when center-slice contrast was particularly high,

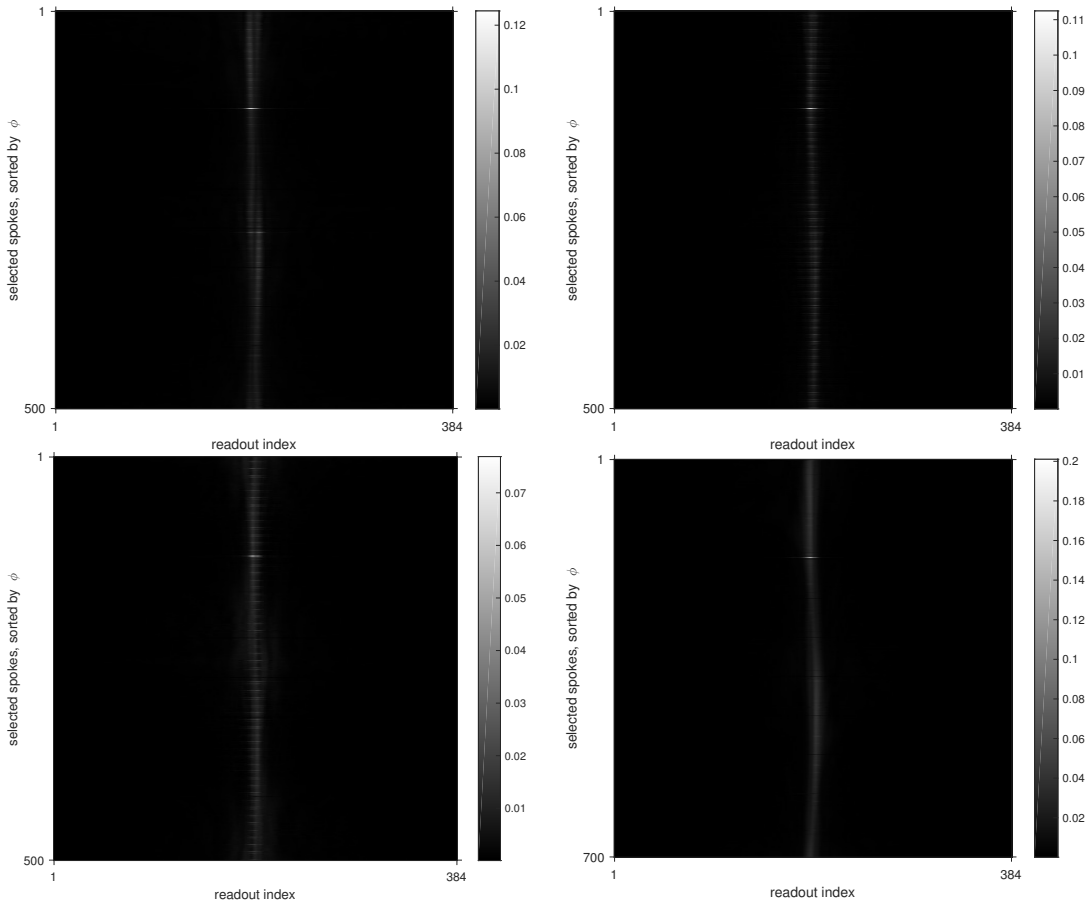


Figure 6.14: Gradient delay analysis of *in vivo* k-space data for four patient datasets, datasets 1-4. Dataset 1 is on the top left, and the results of the other datasets are not presented in this chapter. Only 25% of spokes are displayed for ease of interpretability.

coinciding with contrast agent bolus arrival. Variations in brightness across sorted spokes demonstrate the mixing of pre-contrast and post-contrast data. More importantly, the center of the bright region gives valuable information about the relationship between the ideal and actual location of each readout. For a simple gradient delay, in which g_ϕ is constant for all spoke angles ϕ, we expect to see a single bright line, located g_ϕ from sample 192. For gradient delay modeled as in (6.14), we would expect to see a bright line that deviates as a function of the spoke angle. Instead, we observe dual lines that vary in intensity as a function of spoke angle. The origin of this pattern remains an open question for future work.

6.4.1.2 Navigator Estimation and Respiratory Binning

To separate k-space data based on respiratory state, we estimated the respiratory state from the k-space data itself. This method of self-navigation contrasts with methods that require external devices (such as respiratory belts and bellows [117]) and methods that acquire additional k-space data for measuring respiratory state [49].

To implement self-navigation, we used the center-of-k-space as a proxy for respiratory state. This signal correlates to respiratory state because the total amount of magnetization detected by a coil changes as organs move in and out of the field-of-view. Since each coil measures a spatially-weighted amount of total magnetization, smart choice of channel center-of-k-space signals can provide a robust navigator.

In the absence of external respiratory data, it is not possible to test self-navigation

accuracy for patient data. Because of the scan parameter commonalities with [107], we used the same navigator estimation method. The process performs coil clustering on each coil center-of-k-space signal to detect which coil center-of-k-space signals correlate to respiratory motion. Then we perform PCA on the selected center-of-k-space signals and hand select the PCA element that best represents respiratory motion. Finally, we perform envelope removal to eliminate amplitude changes due to contrast injection.

We quantize respiratory navigator values into the number of respiratory states modeled in f_{5D}. Assuming that the navigator estimation process preserves the relationship between center-of-k-space data and anatomical respiratory position, we divide the navigator signal using amplitude thresholds. The two extreme respiratory states, inhalation and exhalation, are not affected by hysteresis. However, we use the Hilbert transform to differentiate between the direction of transitional respiratory states (those that exist between inhale and exhale). The results of this binning method are overlaid on the estimated navigator in Figure 6.15.

For comparison, we also implemented navigator estimation methods based on [118] [119]. Those results are not shown in this work due to their subjective inferior performance on our patient datasets.

6.4.1.3 Non-Iterative Initialization

After estimating the angle-dependent gradient delay, we use a variant of [34] to create respiratory-binned datasharing volumes. This non-iterative reconstruction

method uses the outer segments of spokes from neighboring frames to satisfy Nyquist sampling requirements. Each resulting frame consists of low frequency k-space data with high temporal resolution and high frequency k-space data with low temporal resolution. Further details are in Section 2.2.6.

We also use this initial image to define a support mask for the image. As motivated in Chapter 3, defining an anatomical mask reduces the number of unknowns, improving the conditioning of (6.11). This masking also yields a practical benefit of reducing memory requirements for representing f_{5D}.

6.4.1.4 Sensitivity Map Estimation

We estimated 3D sensitivity maps from datasharing volumes using [18] for each axial slice and performed low-pass filtering along z. In future work, the 3D sensitivity maps could be estimated directly using a 3D implementation of [18].

To allieviate memory costs associated with the data, we performed coil compression to reduce 16 physical channels to 8 virtual channels [120]. We estimated sensitivity maps from virtual coil images compared to a simulated bodycoil image constructed from full 16-channel data.

The middle three planes for the first four virtual coils are shown in Figure 6.16.

6.4.2 Motion Modeling from Patch-Based Registration

We create a 4D respiratorily-binned volume $f_{\mathrm{r}}(\vec{r}, \phi)$ for dataset 1 by binning each datapoint based on the estimated navigator. Each respiratorily-binned volume was reconstructed using density-compensated gridding for each set of coil data. Due to the high spatial resolution, we simply combine the coil encoded images using the square root of the sum of squares. Sagittal and coronal slices of the respiratory volume are shown in Figure 6.17. The region with the most visible changes across respiratory states is the diaphragm, indicated with a red arrow.

Applying the patch-based registration method in Section 6.2.4 to this respiratory volume yields three-dimensional motion maps linking each respiratory state to each of its two neighbors. As shown in Figure 6.18, these motion models seem inadequate in describing the motion between respiratory states. This may be due to model mismatch arising from estimating PINCAT motion (that occurs in three dimensions) with a motion model restricted to one dimension. The patch-based registration method is also prone to finding false positives in areas with low signal. Unlike the simulation, there are no large regions with similar displacement values, representing upward or downward respiratory motion.

6.4.3 Results

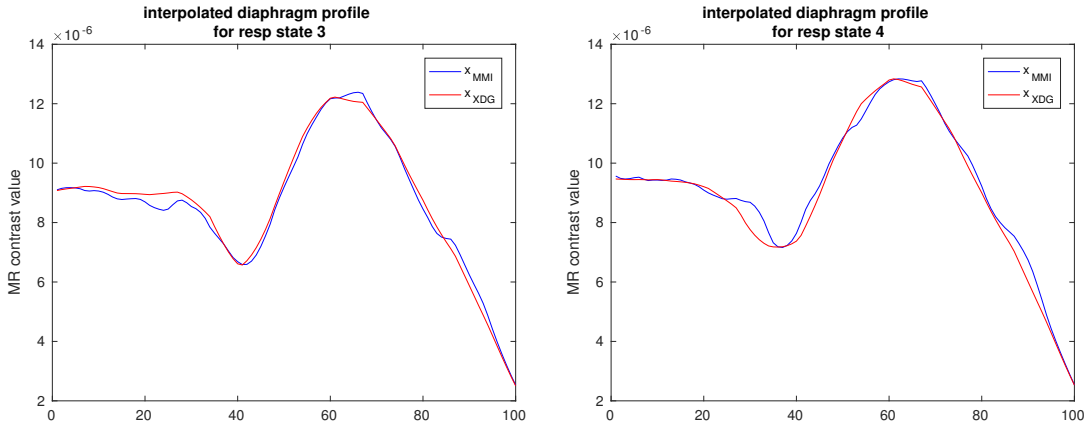
Using CG with corner rounding for 20 iterations, we computed the 5D MRI reconstructions for x_{MMI} and x_{XDG} across 24 temporal frames, for a temporal resolution

of 10.4 seconds/frame. Figure 6.19 shows sagittal slices at a selected temporal frame with high amounts of contrast agent. The corresponding coronal orientation is shown in Figure 6.20.

Compared to the respiratory object shown in Figure 6.17, these 5D reconstructions show poorer spatial resolution. This is likely due to the 24-fold sparsity increase. We also compared the estimated images with the Siemens DICOM dynamic images. Siemens creates these images using radial datasharing for 96 temporal frames, with each frame consisting of 21 spokes of data in the center of k-space and 89 spokes of data in the outermost regions. As a result, these frames have an ostensible temporal resolution of 2.6 seconds, but include information spanning across 11.1 seconds. These DICOM images are also processed for geometric distortion correction.

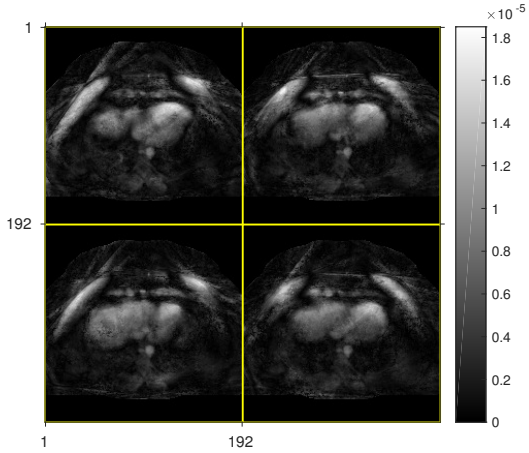
Respiratory motion near the diaphragm is distinguishable in both x_{XDG} and x_{MMI}, especially when compared to the Siemens DICOM image. To better resolve the differences between the two 5D MRI reconstructions, Figure 6.21 shows a profile that crosses the diaphragm, a region we expect to be highly affected by respiratory motion. The profile is shown overlaid onto sagittal and coronal slices of x_{MMI}.

To see the effects of motion model-informed on the diaphragm profile, 6.22 shows plots of the average diaphragm profile across all temporal frames for respiratory state 3, which does not demonstrate significant differences between x_{XDG} and x_{MMI}, and respiratory state 4, which shows a narrower low-contrast gap between the heart and liver.

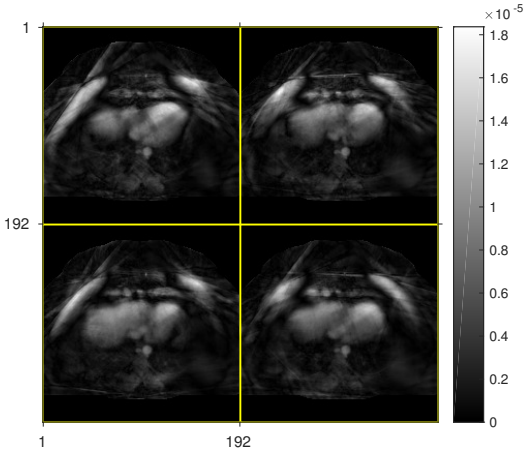


(a) (b)

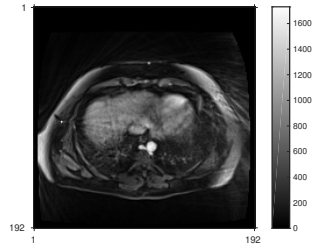
Figure 6.22: Comparison of diaphragm profiles for x_{XDG} and x_{MMI} at respiratory states 3 and 4, averaged across all temporal frames. The fourth respiratory state demonstrated the largest differences between x_{XDG} and x_{MMI}.



(a) Axial slice 15 of x_{MMI}



(b) Axial slice 15 of x_{XDG}



(c) Axial slice 15 of Siemens DICOM

Figure 6.23: Axial slice comparisons of 5D x_{MMI}, x_{XDG}, and Siemens DICOM.

A comparison of x_{XDG}, x_{MMI} and the Siemens DICOM image in the axial orientation, shown in Figure 6.23 reveals that both iterative reconstruction methods seem to emphasize information from the body coil over the information from the spine matrix coils embedded in the patient table. Both 5D MRI reconstructions show the dome of the liver moving through the plane across motion states, a distinction not available in the Siemens DICOM image.

6.5 Discussion

The results of the PINCAT experiments can be extrapolated to a 3D simulation and more sophisticated motion models that allow for multi-dimensional displacement. We expect simulations on digital phantoms with only in-plane motion to show better performance due to improved accuracy of the motion model.

Compared to the PINCAT simulations, *in vivo* patient experiments yielded ambiguous results. Several challenges exist in applying this method to patient data, including model mismatch in k-space sampling and respiratory binning. Due to image quality concerns for dataset 1, we examined the following components of our experiments for sources of error: estimated gradient delay, estimating 3D sensitivity maps from 2D axial sensitivity maps, spatial masking, self-navigation estimation, respiratory binning approaches, coil whitening, and coil compression with and without alignment across z. Further work is required to identify problematic components and produce diagnostic quality 5D MRI images.

This chapter presents *in vivo* data for only one dataset, because patient datasets displayed a large variation in respiratory models, estimated gradient delay, and ultimately quality of image reconstruction results. Many pre-processing steps require hand-tuning to produce favorable results, which is an obstacle to large-scale testing of motion model-informed regularization for multiple patient datasets.

The proposed motion model-informed regularization is agnostic to the registration method used to construct the motion model. We expect that more comprehensive

motion models that account for through-plane motion, realistic anatomical motion, and sub-pixel registration would yield further benefits. Because we model the motion as a pre-processing step, the registration only needs to be performed once, as opposed to joint motion estimation methods that require repeated motion modeling during the iterative optimization.

This chapter uses a standard optimization method, conjugate gradient (CG), to solve (6.11), but other optimization methods may significantly reduce computation time. The binary property of warping matrix $\tilde{\mathbf{W}}$ may be exploitable for faster convergence. Various variable splitting strategies for this problem are proposed in Section 7.2, however, we did not implement these due to computational constraints.

6.5.1 Computational Considerations

The choice to represent the unknown object in five dimensions as $f_{5\text{D}}(\vec{r}, t, \phi)$ has many benefits for modeling but comes with many implementation drawbacks. The large object size and system matrices require more system memory and long computation times. Depending on the amount of memory available, this may preclude memory-intensive iterative algorithms, such as variable splitting.

Cost function (6.10) includes many opportunities for parallelized computation. The system matrices $\mathbf{F}_{5\text{D}}, \mathbf{S}_{5\text{D}}$ can be represented as kronecker products of smaller oper-

ators as follows:

$$\mathbf{F}_{5\mathrm{D}} = \mathbf{I}_{N_\mathrm{z} N_\mathrm{c}} \otimes \mathbf{F}_{\mathrm{t},\phi} \tag{6.15}$$

$$\mathbf{F}_{\mathrm{t},\phi} : \mathbb{C}^{N_\mathrm{x} \times N_\mathrm{y}} \mapsto \mathbb{C}^{N_\mathrm{ro} \times N_\mathrm{spokes}(t,\phi)} \tag{6.16}$$

$$\mathbf{S}_{5\mathrm{D}} = \mathbf{I}_{N_\mathrm{t} N_\mathrm{resp}} \otimes \mathbf{S} \tag{6.17}$$

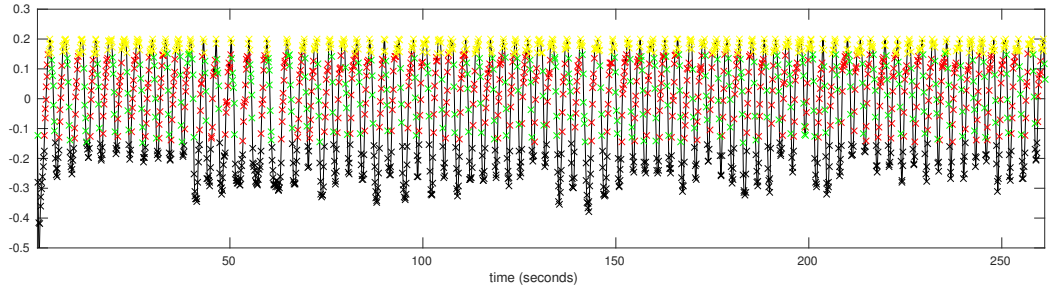
$$\mathbf{S} : \mathbb{C}^{N_\mathrm{x} \times N_\mathrm{y} \times N_\mathrm{z}} \mapsto \mathbb{C}^{N_\mathrm{x} \times N_\mathrm{y} \times N_\mathrm{z} \times N_\mathrm{c}} \tag{6.18}$$

To exploit these repeated operations, we parallelize the $\mathbf{F}_{5\mathrm{D}}, \mathbf{S}_{5\mathrm{D}}$ operations over $N_\mathrm{z} N_\mathrm{c}$ and $N_\mathrm{t} N_\mathrm{resp}$, respectively. Unfortunately, the different dimensions of parallelization are not amenable to a jointly parallelized operator. We use Matlab's built-in `parfor` routine because each parallelized job requires much more compute time than the overhead in starting and managing a parallel pool. Future implementations could benefit from a lower-level parallelization of $\mathbf{F}_{5\mathrm{D}}$ and $\mathbf{S}_{5\mathrm{D}}$. A drawback of `parfor` is that data is copied by value to each worker, effectively doubling the memory footprint of the k-space data, sensitivity maps, and 5D object.

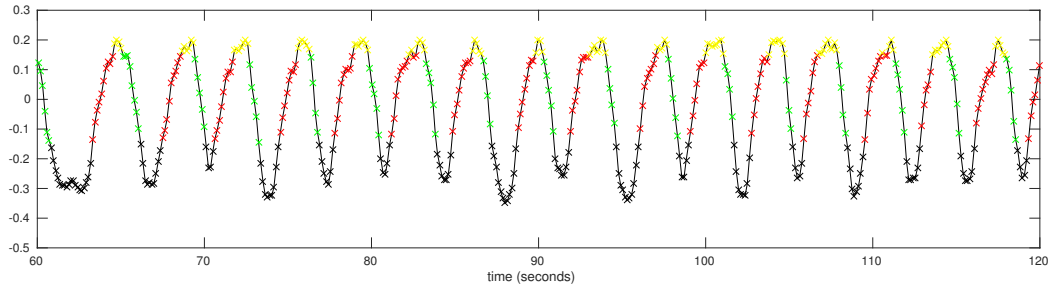
A typical patient dataset consisting of 2800 spokes, 380 readouts per spoke, 20 channels, and 58 slices in complex single-precision occupies 18.4 GB of memory. For $N_\mathrm{t} = 80$ and $N_\mathrm{resp} = 2$, $f_{5\mathrm{D}}$ requires 5.1 GB. For only five temporal frames and two respiratory states, we need 326 MB.

6.6 Conclusion

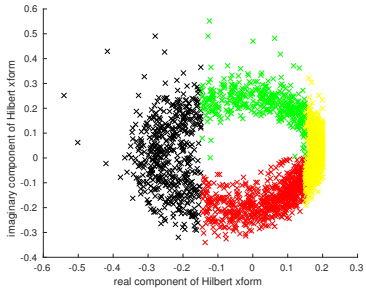
Preliminary work on motion model-informed regularization for simulated 5D MRI reconstruction demonstrates that even rudimentary motion modeling yields noticeable differences in image quality. Patient data experiments demonstrate some differences in reconstructed images, but further work is needed to clarify the image quality of 5D MRI reconstructions with and without motion model-informed regularization. Chapter 7 describes future work based on motion model-informed regularization.



(a)



(b)



(c)

Figure 6.15: (a) respiratory navigator signal for the entire scan, binned into 4 respiratory states; (b) 60 second excerpt of (a); (c) Hilbert transform of navigator signal, used for differentiating the transition from inhale to exhale from the reverse.

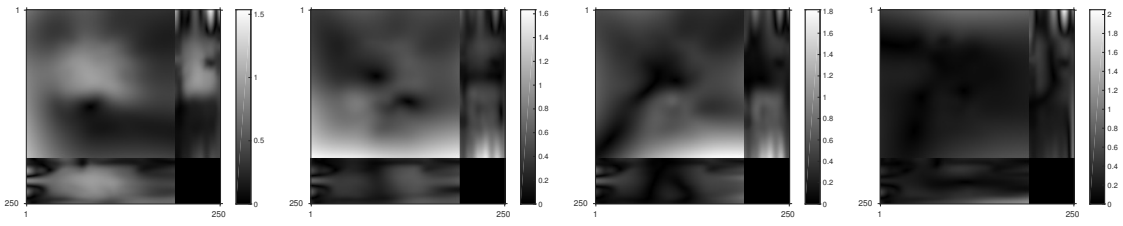
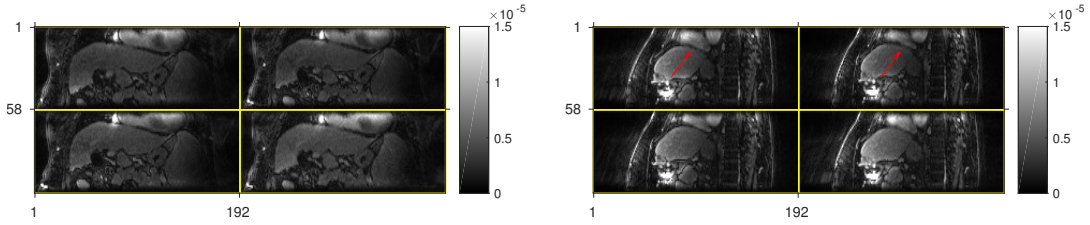


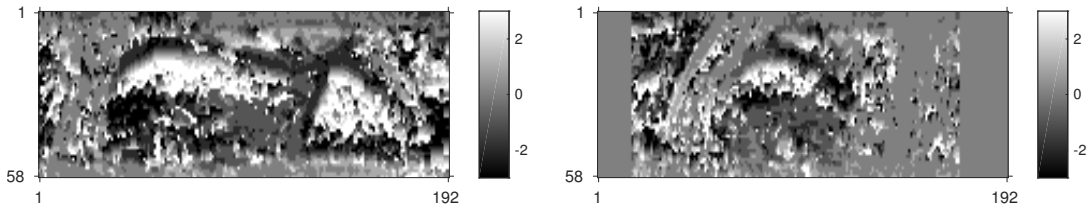
Figure 6.16: First four of eight virtual coils used for image reconstruction.



(a) Coronal slice of respiratory volume $f_{\mathrm{r}}(\vec{r}, \phi)$

(b) Sagittal slice of respiratory volume $f_{\mathrm{r}}(\vec{r}, \phi)$

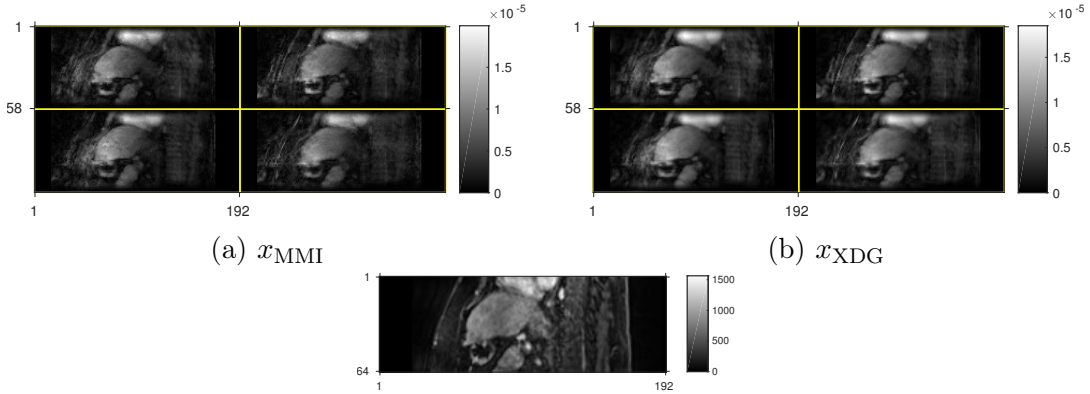
Figure 6.17: Respiratory volume $f_{\mathrm{r}}(\vec{r}, \phi)$, with 4 respiratory phases arranged lexicographically.



(a) Coronal slice of motion map.

(b) Sagittal slice of motion map.

Figure 6.18: Motion model relating respiratory state 3 to respiratory state 2. This volume of displacements was calculated from patch-based registration as described in Section 6.2.4.

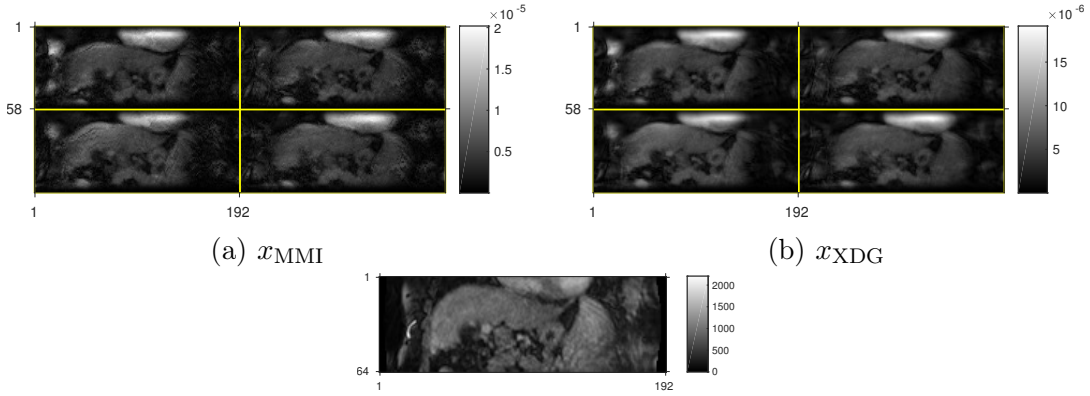


(a) x_{MMI}

(b) x_{XDG}

(c) Siemens DICOM

Figure 6.19: (a, b) Sagittal slices for four respiratory states at temporal frame 7 of 24, corresponding to 73 seconds into the scan; (c) corresponding Siemens DICOM slice.



(a) x_{MMI}

(b) x_{XDG}

(c) Siemens DICOM

Figure 6.20: (a, b) Coronal slices for four respiratory states at temporal frame 7 of 24, corresponding to 73 seconds into the scan; (c) corresponding Siemens DICOM slice.

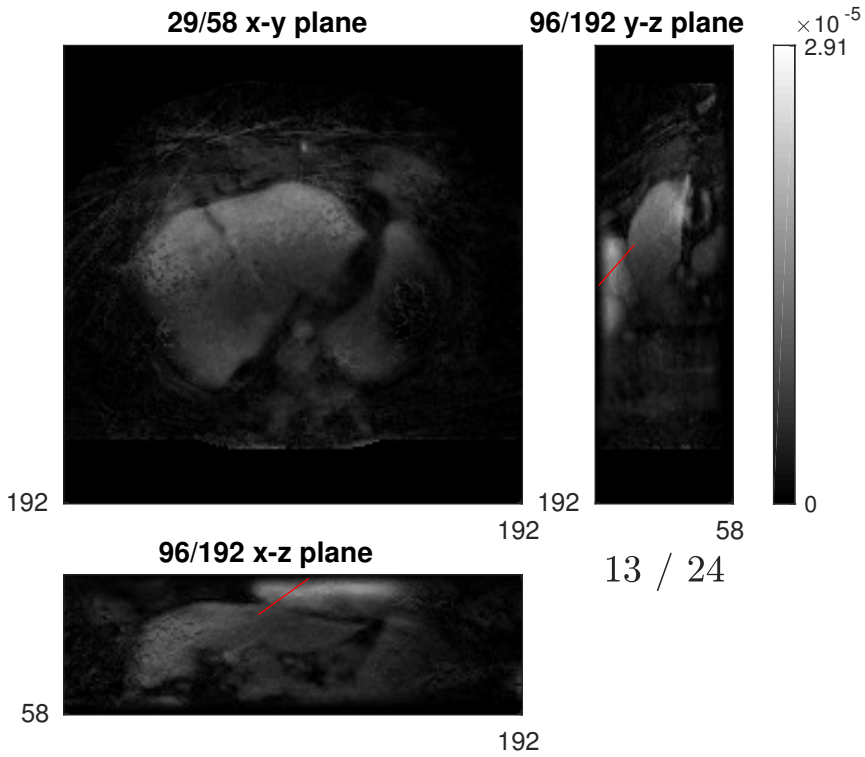


Figure 6.21: Selected profile crossing diaphragm, displayed on frame 13 of x_{MMI}.

Chapter 7

Conclusions and Future Work

7.1 Summary

This thesis presents a variety of MR image reconstruction models and algorithms. Chapters 4 - 6 present methods that focus on or allow for reconstructing accurate dynamics, a key goal in DCE-MRI. Models presented in Chapters 3 - 6 share a common theme: model-based image reconstruction methods should reflect known anatomical constraints and behavior. AL-P2-mask in Chapter 3 incorporates known anatomical support, AL-tridiag in Chapter 4 does not enforce spurious periodic boundary conditions across space or time, spline temporal bases in Chapter 5 model smooth contrast changes in low motion DCE-MRI, and motion model-informed regularization in Chapter 6 accounts for modeled motion in 5D MRI. Altogether, in this body of work, we seek to realize ultimate goal of improving free-breathing DCE-MRI.

In addition to presenting improved models for MR image reconstruction, this thesis also presents efficient algorithms for minimizing the resulting cost functions, with

the goal of reducing the computational burden of iterative reconstruction. The algorithms in this thesis rely on variable splitting and alternating minimization to decouple the optimization problem into sub-problems with exploitable Hessian structures. Chapters 4 and 5 also present parallelizable algorithms that satisfy equivalence with ADMM and thus are guaranteed to convergence to a minimizer.

The proposed variable splitting method AL-P2-mask demonstrates faster convergence compared to AL-P2 and similar variable splitting methods for *in vivo* data. AL-tridiag and its convergent and parallelizable variants demonstrate comparable convergence speed and improved image quality to AL-P2 for simulated and *in vivo* MR data and lower boundary errors for inpainting applications. Spline temporal basis modeling for DCE-MRI lowered error for pharmacokinetic parameter estimation for simulated DCE-MRI with a low compute time, compared to conventional rectangular temporal bases. Lastly, we show motion model-informed regularization results in some reduction of motion artifacts for PINCAT simulations, even with rudimentary motion modeling. We believe that improved motion models can amplify the benefits of motion model-informed regularization for 5D MRI.

7.2 Improved motion model-informed 5D MRI Reconstruction

7.2.1 Variable Splitting Methods for Motion-Informed Regularization

Here we outline a variable splitting method for efficient optimization of (6.10) incorporating R_{mm} as well as spatial and temporal finite differences:

$$J(f_{5\text{D}}) = \frac{1}{2}\left\|y - \mathbf{F}_{5\text{D}}\mathbf{S}_{5\text{D}}f_{5\text{D}}\right\|^2 + \lambda_s R_s(f_{5\text{D}}) + \lambda_t R_t(f_{5\text{D}}) + \lambda_m R_{\text{mm}}(f_{5\text{D}}) \tag{7.1}$$

$$\begin{aligned} J(f_{5\text{D}}) = \frac{1}{2}&\left\|y - \mathbf{F}_{5\text{D}}\mathbf{S}_{5\text{D}}P(f_{5\text{D}})\right\|^2 \\ &+ \lambda_s \sum_{t=1}^{N_t}\sum_{\phi=1}^{N_{\text{resp}}}\sum_{\vec{r}}^{N_r}\sum_{\vec{r'}\in\mathcal{N}_r} \psi_s\left(f_{5\text{D}}(\vec{r},t,\phi) - f_{5\text{D}}(\vec{r}-\vec{r}',t,\phi)\right) \\ &+ \lambda_t \sum_{t=2}^{N_t}\sum_{\phi=1}^{N_{\text{resp}}}\sum_{\vec{r}}^{N_r} \psi_t\left(f_{5\text{D}}(\vec{r},t,\phi) - f_{5\text{D}}(\vec{r},t-1,\phi)\right) \\ &+ \lambda_{\text{mm}} \sum_{t=1}^{N_t}\sum_{\vec{r}=\vec{0}}^{N_r}\sum_{\phi=1}^{N_{\text{resp}}} \psi_m\left(f_{5\text{D}}(\vec{r},t,\phi) - f_{5\text{D}}\left(W_{\phi,\phi-1}(\vec{r}),t,\phi-1\right)\right) \end{aligned} \tag{7.2}$$

$$J(f_{5\mathrm{D}}) = \frac{1}{2}\left\|y - \mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}f_{5\mathrm{D}}\right\|^2 + \psi\left(\mathbf{L}\underbrace{\begin{bmatrix}\mathbf{C}_s\\ \mathbf{C}_t\end{bmatrix}}_{\tilde{\mathbf{C}}} f_{5\mathrm{D}}\right) + \lambda_m\psi_m\left(\mathbf{C}_r\tilde{\mathbf{W}}f_{5\mathrm{D}}\right) \tag{7.3}$$

$$\mathbf{L} = \mathrm{diag}\{\lambda_s \otimes \mathbf{I}_{N_r N_t N_{\mathrm{resp}}|\mathcal{N}_\nabla|}, \lambda_t \otimes \mathbf{I}_{N_r(N_t-1)N_{\mathrm{resp}}}\} \tag{7.4}$$

$$\begin{aligned}\psi(x) = {} & \psi_s\left(x(1 : N_r N_t N_{\mathrm{resp}}|\mathcal{N}_r|)\right) \\ & + \psi_t\left(x(N_r N_t N_{\mathrm{resp}}|\mathcal{N}_r| + 1 : N_r N_t N_{\mathrm{resp}}|\mathcal{N}_r| + N_r(N_t-1)N_{\mathrm{resp}})\right)\end{aligned} \tag{7.5}$$

Here we introduce a new linear operator $\tilde{\mathbf{W}} : \mathbb{C}^{N_r \times N_{\mathrm{t}} \times N_{\mathrm{resp}}} \mapsto \mathbb{C}^{N_r \times N_{\mathrm{t}} \times N_{\mathrm{resp}}}$, that incorporates the interpolation and coordinate transform from each respiratory phase to a reference respiratory phase. Essentially, $\tilde{\mathbf{W}}f_{5\mathrm{D}}$ is the pharmacokinetic-respiratory object with all respiratory phases converted to $\phi = 1$ via the precomputed motion model. $\mathbf{L}$ is a diagonal matrix for conveniently applying the two distinct regularization parameters, λ_s and λ_t, and $\psi(x)$ is a notationally convenient combination of potential functions ψ_s and ψ_t, which operate on different portions of the input.

One variable splitting scheme for this problem is outlined as follows:

$$J(f_{5\mathrm{D}}, u, v, w) = \frac{1}{2}\left\|y - \mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}w\right\|^2 + \psi\left(u\right) + \lambda_m\psi_m\left(v\right) \tag{7.6}$$

$$\text{s.t. } u = \mathbf{L}\tilde{\mathbf{C}}f_{5\mathrm{D}};\ v = \mathbf{C}_r\tilde{\mathbf{W}}\left(f_{5\mathrm{D}}\right);\ w = f_{5\mathrm{D}}. \tag{7.7}$$

With the above choice of u and v, we have isolated the arguments of the regularization

terms, facilitating simple updates. The third split for w isolates the $\mathbf{S}'_{5\mathrm{D}}\mathbf{F}'_{5\mathrm{D}}\mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}$ Hessian from $\tilde{\mathbf{W}}\ \mathbf{C}'_r\mathbf{C}_r\tilde{\mathbf{W}}$.

The Augmented Lagrangian function is

$$\mathcal{L}(f_{5\mathrm{D}},u,v,w)=\frac{1}{2}\left\|y-\mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}w\right\|^2+\psi\left(u\right)+\lambda_m\psi\left(v\right)+\frac{\mu_u}{2}\left\|u-\mathbf{L}\tilde{\mathbf{C}}f_{5\mathrm{D}}-\eta_u\right\|^2 + \frac{\mu_v}{2}\left\|v-\mathbf{C}_r\tilde{\mathbf{W}}f_{5\mathrm{D}}-\eta_v\right\|^2+\frac{\mu_w}{2}\left\|w-f_{5\mathrm{D}}-\eta_w\right\|^2. \tag{7.8}$$

Applying alternating minimization, we arrive at the following variable updates:

$$f_{5\mathrm{D}}^{(n+1)}=H_x^{-1}\left(\mu_u\tilde{\mathbf{C}}'\mathbf{L}'\left(u-\eta_u\right)+\mu_v\tilde{\mathbf{W}}'\mathbf{C}'_r\left(v-\eta_v\right)+\mu_w\left(w-\eta_w\right)\right) \tag{7.9}$$

$$H_x=\mu_u\tilde{\mathbf{C}}'\mathbf{L}^{\mathbf{2}}\tilde{\mathbf{C}}+\mu_v\tilde{\mathbf{W}}'\mathbf{C}'_r\mathbf{C}_r\tilde{\mathbf{W}} \tag{7.10}$$

$$u^{(n+1)}=\mathrm{shrink}\left(\mu_u,\mathbf{L}\tilde{\mathbf{C}}f_{5\mathrm{D}}^{(n)}+\eta_u^{(n)}\right) \tag{7.11}$$

$$v^{(n+1)}=\mathrm{shrink}\left(\frac{\mu_v}{\lambda_m},\mathbf{C}_r\tilde{\mathbf{W}}f_{5\mathrm{D}}^{(n)}+\eta_v^{(n)}\right) \tag{7.12}$$

$$w^{(n+1)}=H_w^{-1}\left(\mathbf{S}'_{5\mathrm{D}}\mathbf{F}'_{5\mathrm{D}}y+\mu_w(f_{5\mathrm{D}}^{(n)}+\eta_w^{(n)})\right) \tag{7.13}$$

$$H_w=\mathbf{S}'_{5\mathrm{D}}\mathbf{F}'_{5\mathrm{D}}\mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}+\mu_w\mathbf{I}_{N_rN_tN_{\mathrm{resp}}} \tag{7.14}$$

The updates for $f_{5\mathrm{D}}$ and w could be computed with PCG, specifically with circulant preconditioners. The behavior of $\tilde{\mathbf{W}}$ could be approximated as a permutation matrix, whose orthogonality would reduce H_x to a circulant Hessian. In reality, $\tilde{\mathbf{W}}$ is a

sparse binary matrix, in the case of integer-pixel registration, or a sparse matrix with normalized, compact rows for non-integer pixel registration.

7.2.2 Variations to 5D MRI Problem

The following variations of the 5D MRI cost function, detailed in Sections 7.2.2.1 - 7.2.2.4, may improve image quality and warrant investigation.

7.2.2.1 Soft-gated datafit

Using a soft-gating function in the datafit allows measurement data to contribute to more than one respiratory phase, similar to the spline temporal interpolation in Chapter 5. For a single respiratory phase (i.e., gated reconstruction), the following cost function has been shown to improve image quality [6]:

$$J_{\text{soft}}(f_{\text{5D}}) = \sum_{\phi=1}^{N_{\text{resp}}} \frac{1}{2} \left\| W_\phi (y - \mathbf{F}_{\text{5D}} \mathbf{S}_{\text{5D}} f_{\text{5D}}) \right\|^2 + R(f_{\text{5D}}). \tag{7.15}$$

$$W_\phi(t) = \begin{cases} 1, & g(t) = \phi \\ e^{-\alpha(|g(t)-\phi|-\tau)}, & |g(t)-\phi| \leq \tau \\ 0, & |g(t)-\phi| > \tau \end{cases} \tag{7.16}$$

Data within a threshold, τ, of a given respiratory phase contributes to the datafit term with an influence factor of α. A smaller α allows neighboring respiratory phases to have greater influence, and as $\alpha \to \infty$, we approach the hard-gating, i.e., binning,

described in (6.10). This soft-gating idea could be combined with finer temporal resolution for manifold-style regularization [121].

7.2.2.2 Bolus-Sensitive Temporal Regularization

Another adaptation of temporal regularization that may better accomodate sharp changes in contrast is bolus-sensitive regularization:

$$R_{\text{bolus}}(f_{\text{5D}}) = \sum_{\phi=1}^{N_{\text{resp}}} \sum_{\vec{r}}^{N_r} \sum_{t=2}^{N_t} \beta_t \psi_t \left(f_{\text{5D}}(\vec{r}, t, \phi) - f_{\text{5D}}(\vec{r}, t-1, \phi) \right). \tag{7.17}$$

β_t is a weight that penalizes temporal varations during slowly-varying pre-contrast and washout frames more heavily than those near aortic and hepatic bolus arrival, when contrast changes are most dramatic. One can design β_t from the raw navigator signal, which describes the total magnetization present in the volume. Sharp changes in total magnetization mark the two bolus arrivals.

7.2.2.3 Translation Motion in System Model

The forward model could be expanded to include a simple estimate of the S/I respiratory motion. This simple addition to the system model might describe a significant portion of respiratory movement and reduce model mismatch, yielding a more accurate estimate for f_{5D}. The new datafit term would be

$$\frac{1}{2}\left\|y - \mathbf{D}\mathbf{F}_{5\mathrm{D}}\mathbf{S}_{5\mathrm{D}}f_{5\mathrm{D}}\right\|^2. \tag{7.18}$$

$\mathbf{D}$ is a diagonal matrix that accounts for z-translation in the Fourier domain by mapping all respiratory phases to a reference phase. By restricting datafit motion modelling to rigid, translational motion, $\mathbf{D}$ is diagonal and invertible. The translational shift of the respiratory phases could be estimated from coil-specific navigator data to better describe local motion.

7.2.2.4 Spatially Varying Temporal Regularization

An adaptation of spatial regularization that may better accomodate sharp contrast changes in certain regions is spatially varying temporal regularization:

$$R(f_{5\mathrm{D}}) = \sum_{t=2}^{N_t}\sum_{\phi=1}^{N_{\mathrm{resp}}}\sum_{\vec{r}}^{N_r}\lambda(\vec{r})|f_{5\mathrm{D}}(\vec{r},t,\phi) - f_{5\mathrm{D}}(\vec{r},t-1,\phi)|. \tag{7.19}$$

$$= \left\|\mathbf{L}\mathbf{C}_t f_{5\mathrm{D}}\right\|_1, \ \mathbf{L} \triangleq \mathrm{diag}\{\lambda(\vec{r})\} \otimes \mathbf{I}_{N_t} \tag{7.20}$$

From the initial datasharing estimates, one could design a set of spatially-varying weights for temporal regularization. Voxels with little temporal variation in the initial datasharing estimate (e.g., voxels of air, bone) would have higher weights than voxels with high temporal variation.

7.2.3 Memory-Efficient Reconstruction Algorithms

The variable splitting approaches for 5D MRI in Section 7.2.1 may not be practical depending on hardware limitations, due to large memory requirements in using auxiliary variables related to the already large 5D MRI object. One possible solution is to use distributed optimization as described in [122] to limit local memory demand. Another approach to alleviate high memory requirements would be to use online image reconstruction methods, such as [123].

7.3 Regularized Indirect Pharmacokinetic Parameter Estimation

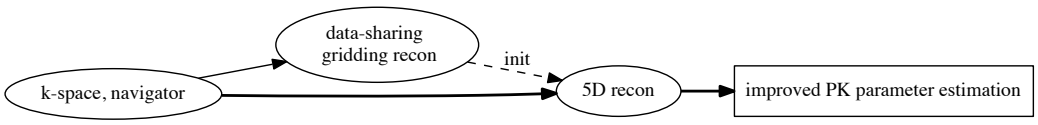


Figure 7.1: Pharmacokinetic parameter pipeline with improved pharmacokinetic parameter estimation from 5D dynamic images.

The ultimate goal of this research direction is to improve upon conventional pharmacokinetic parameter estimation from reconstructed dynamic images, the final stage in the pipeline outlined in Figure 7.1. One could use the estimated values of 5D MRI, $\hat{f_{5D}}$, to estimate regularized pharmacokinetic maps.

Consider indirect estimation of a pharmacokinetic map, one for each respiratory phase. Equivalently, one could consider N_{resp} separate reconstruction problems using gated data. By considering one respiratory phase (e.g., inspiration) at a time, one could examine the usefulness of pharmacokinetic parameter regularization unencumbered by the effects of respiratory motion.

Let the unknown pharmacokinetic maps for respiratory phase ϕ be $x_{\text{PK},\phi}(\vec{r},\kappa)$ where $\kappa = 1,\ldots,5$ for liver pharmacokinetic parameters $[k_{1a}, k_{1p}, k_2, \tau_a, \tau_p]$, and ϕ is the respiratory phase index. This model allows for regularization across pharmacokinetic parameters, as well as regularization within a given pharmacokinetic parameter index.

Since this approach estimates pharmacokinetic parameter values from dynamic images $\hat{f}_{5D}$, one could use those input dynamic images to manually segment the liver, and perform this reconstruction over the liver mask for N_{mask} voxels. Let the nonlinear relationship between the pharmacokinetic parameters and the resulting time series for each voxel be represented by the operator $P : \mathbb{R}^{N_r N_{\text{PK}}} \mapsto \mathbb{C}^{N_r N_t}$.

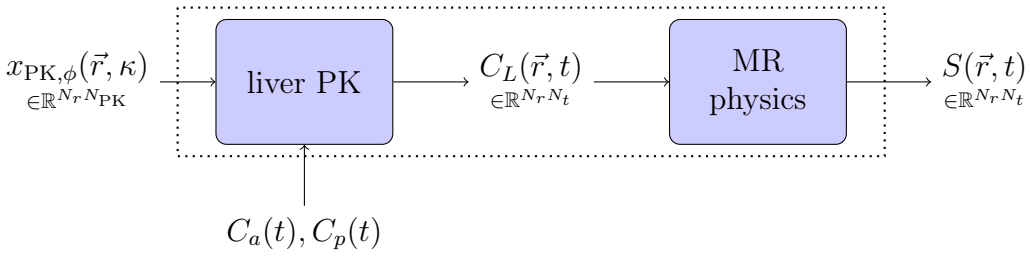


Figure 7.2: Block diagram of operator P. Note that aortal and portal vein inputs $C_a(t)$ and $C_p(t)$ are pre-computed.

For the liver PK step in Figure 7.2, one could apply the dual-input, single-compartment

model for each voxel, as in Equation (2.56), restated here:

$$C_L(\vec{r},t) = \int_0^t \left[k_{1a}(\vec{r})C_a(\vec{r},t'-\tau_a) + k_{1p}(\vec{r})C_p(\vec{r},t'-\tau_p)\right] e^{-k_2(\vec{r},t-t')}dt'.$$

The MR physics block of Figure 7.2 computes the dynamic image $S(t)$ through the SPGR signal equation, previously discussed in Equations (2.37) and (2.38), and restated here:

$$\begin{aligned}\frac{1}{T_1(\vec{r},t)} &= \frac{1}{T_{10}(\vec{r})} + r_1 C_{\mathrm{t}}(t)(\vec{r}) \\ m_0(\vec{r}) &= \rho_0(\vec{r})\sin\left(\theta(\vec{r})\right)\left(\frac{1-E_1(\vec{r})}{1-E_1(\vec{r})\cos\left(\theta(\vec{r})\right)}\right) e^{-\frac{T_E}{T_2^*(\vec{r})}}; \qquad E_1(\vec{r}) = e^{-\frac{T_R}{T_1(\vec{r},t)}}.\end{aligned}$$

Values T_R and θ are known. The remaining unknown product of proton density, $\rho_0(\vec{r})$, and the exponential term, $e^{-\frac{T_E}{T_2^*(\vec{r})}}$, can be treated as an additional pharmacokinetic parameter, to be estimated along with k_{1a}, k_{1p}, k_2, τ_a, and τ_p, extending N_{PK} to six: $\kappa_6(\vec{r}) = \rho_0(\vec{r})e^{-\frac{T_E}{T_2^*(\vec{r})}}$.

Then the cost function minimize is

$$J(x_{\mathrm{PK},\phi}) = \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}; C_a(t), C_p(t)\right)\right\|^2 + R(x_{\mathrm{PK},\phi}). \tag{7.21}$$

The next section discusses possible regularizers.

7.3.1 Pharmacokinetic Parameter Regularization

The general regularization term $R(x_{\mathrm{PK},\phi})$ in (7.21) can include penalties and constraints such as:

- independent spatial regularization for each pharmacokinetic parameter

$$R_s(x_{\mathrm{PK},\phi}) = \lambda_s \sum_{\kappa=1}^{N_{\mathrm{pk}}} \sum_{\vec{r}}^{N_{\mathrm{mask}}} \sum_{\vec{r}' \in \mathcal{N}_r} \psi\left(x_{\mathrm{PK},\phi}(\vec{r},\kappa) - x_{\mathrm{PK},\phi}(\vec{r}-\vec{r}',\kappa)\right) \tag{7.22}$$

- Total Variation across pharmacokinetic parameters

$$R_{\mathrm{PKTV}}(x_{\mathrm{PK},\phi}) = \lambda_{\mathrm{PKTV}} \sum_{\vec{r}}^{N_{\mathrm{mask}}} \sqrt{\sum_{\kappa=1}^{N_{\mathrm{pk}}} \left(\frac{\sum_{\vec{r}' \in \mathcal{N}_r} x_{\mathrm{PK},\phi}(\vec{r},\kappa) - x_{\mathrm{PK},\phi}(\vec{r}-\vec{r}',\kappa)}{p(\kappa)} \right)^2} \tag{7.23}$$

  where $p(\kappa)$ is some normalization for a given pharmacokinetic parameter. We expect pharmacokinetic parameter spatial boundaries to coincide, such as along tumor boundaries, and at organ/tissue interfaces [124].

- nonnegativity of all pharmacokinetic parameters

$$R_{\mathrm{neg}}(x_{\mathrm{PK},\phi}) = \sum_{\kappa=1}^{N_{\mathrm{pk}}} \sum_{\vec{r}}^{N_{\mathrm{mask}}} h(x_{\mathrm{PK},\phi}(\vec{r},\kappa)) \tag{7.24}$$

$$h(x) = \begin{cases} \infty, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{7.25}$$

- indirect enforcement of pharmacokinetic informed inequality constraints (i.e., membership in refined convex set)

$$R_\tau(x_{\mathrm{PK},\phi}) = \sum_{\vec{r}}^{N_{\mathrm{mask}}} h(x_{\mathrm{PK},\phi}(\vec{r},5) - x_{\mathrm{PK},\phi}(\vec{r},4)) \quad \text{effectively, } \tau_p > \tau_a \tag{7.26}$$

$$R_k(x_{\mathrm{PK},\phi}) = \sum_{\vec{r}}^{N_{\mathrm{mask}}} h(x_{\mathrm{PK},\phi}(\vec{r},2) - x_{\mathrm{PK},\phi}(\vec{r},1)) \quad \text{effectively, } k_{1p} > k_{1a} \tag{7.27}$$

  The portal vein input always occurs after the aortic input, and studies have shown that k_{1p} is usually 70 percent of $k_{1p} + k_{1a}$ [46].

Equation (7.23) describes a novel use of vectorial TV to encourage joint sparsity for pharmacokinetic parameter estimation. Vectorial TV has been used for color image denoising [125] and image deblurring [126]. Though joint sparsity constraints and regularization have been used for DCE image reconstruction [127], it has yet to be applied to pharmacokinetic parameter estimation.

One could also explore constraining the pharmacokinetic parameter values to lie within intervals determined from literature (i.e., applying box constraints):

$$R(x_{\mathrm{PK},\phi}) = \sum_{\kappa=1}^{N_{\mathrm{pk}}} \sum_{\vec{r}}^{N_{\mathrm{mask}}} h_\kappa(x_{\mathrm{PK},\phi}) \tag{7.28}$$

$$h_\kappa(x) = \begin{cases} \infty, & x \notin [a_\kappa,\ b_\kappa] \\ 0, & x \in [a_\kappa,\ b_\kappa] \end{cases}. \tag{7.29}$$

7.3.2 Minimization Algorithms and Implementation Details

7.3.2.1 Initialization for Iterative Methods

One could compute the initial estimate of $x_{\mathrm{PK},\phi}$ from $\hat{f_{\mathrm{5D}}}$ via conventional variable projection for each voxel and each respiratory state, without regularization as follows:

$$x_{\mathrm{PK},\phi}{}^{(0)} = \operatorname{argmin} \frac{1}{2} \left\| \hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}; C_a(t), C_p(t)\right) \right\|^2. \tag{7.30}$$

This can be computed as shown in Section 2.4.3.

7.3.2.2 Opportunities for Parallelization

When spatial regularization is included (e.g., (7.22)), one could use variable splitting as follows to create a spatially separable sub-problem:

$$J(x_{\mathrm{PK},\phi}) = \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}\right)\right\|^2 + \lambda_s \sum_{\kappa=1}^{N_{\mathrm{pk}}} \sum_{\vec{r}}^{N_{\mathrm{mask}}} \sum_{\vec{r}'\in\mathcal{N}_r} \psi\left(x_{\mathrm{PK},\phi}(\vec{r},\kappa) - x_{\mathrm{PK},\phi}(\vec{r}-\vec{r}',\kappa)\right) \tag{7.31}$$

$$= \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}\right)\right\|^2 + \lambda_s \psi(\mathbf{C}_s x_{\mathrm{PK},\phi}) \tag{7.32}$$

$$J(x_{\mathrm{PK},\phi}, u) = \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}\right)\right\|^2 + \lambda_s \psi(\mathbf{C}_s u) \tag{7.33}$$

$$\text{s.t. } u = x_{\mathrm{PK},\phi}. \tag{7.34}$$

This problem has the following Augmented Lagrangian:

$$\mathcal{L}(x_{\mathrm{PK},\phi}, u; \mu) = \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - P\left(x_{\mathrm{PK},\phi}\right)\right\|^2 + \lambda_s \psi(\mathbf{C}_s u) + \frac{\mu}{2}\left\|u - x_{\mathrm{PK},\phi} - \eta\right\|^2. \tag{7.35}$$

Alternating minimization yields a spatially separable update for $x_{\mathrm{PK},\phi}$. Defining $\kappa_\phi(\vec{r}) \triangleq x_{\mathrm{PK},\phi}(\vec{r},\cdot)$, the update is:

$$\begin{aligned}\hat{\kappa_\phi}^{(n+1)} &= \underset{\kappa_\phi \in \mathbb{R}^{N_{\mathrm{PK}}}}{\operatorname{argmin}} \frac{1}{2} \left\| \hat{f}_{5D}(\vec{r}, \cdot, \phi) - P\left(\kappa_\phi^{(n)}\right) \right\|^2 + \frac{\mu}{2} \left\| u(\vec{r}, \cdot)^{(n)} - \kappa_\phi^{(n)} - \eta(\vec{r}, \cdot)^{(n)} \right\|^2 \\ &= \underset{\kappa_\phi}{\operatorname{argmin}} \frac{1}{2} \left\| \begin{bmatrix} f_{5D}(\vec{r}, \cdot, \phi) \\ u(\vec{r}, \cdot)^{(n)} - \eta(\vec{r}, \cdot)^{(n)} \end{bmatrix} - \tilde{P}\left(\kappa_\phi^{(n)}\right) \right\|^2 \qquad \text{for } \vec{r} = 1, \ldots, N_r \end{aligned} \tag{7.36}$$

$$\text{with } \tilde{P}(x) = \begin{bmatrix} P(x) \\ x \end{bmatrix}. \tag{7.37}$$

This inner problem can be rewritten as a nonlinear least-squares problem, making it a candidate for variable projection methods. Since this inner update is now spatially separable, one could perform the minimization over each voxel in parallel.

The update for auxiliary variable u is a simple denoising problem:

$$\underset{u}{\operatorname{argmin}} \frac{\mu}{2} \left\| u - x_{\mathrm{PK},\phi} - \eta \right\|^2 + \lambda \psi(\mathbf{C}_s u), \tag{7.38}$$

which in turn, one could solve iteratively for a differentiable ψ as follows:

$$u^{(m+1)} = \left(\mathbf{I} + \frac{\lambda}{\mu}\mathbf{C}_s'\mathbf{D}(u^{(m)})\mathbf{C}_s\right)^{-1}(x_{\mathrm{PK},\phi} + \eta). \tag{7.39}$$

$$\mathbf{D}(u) \triangleq \mathrm{diag}\{\omega_\psi([\mathbf{C}_s u]_k)\} \tag{7.40}$$

$$\omega_\psi(t) \triangleq \frac{\dot{\psi}(t)}{t} \tag{7.41}$$

Using wavelets instead of finite differences in the regularizer would enable a synthesis formulation and a non-iterative u-update. For non-differentiable ψ, efficient GPU algorithms [128] and variable splitting approaches [78] can be used.

7.3.2.3 ADMM with a Nonlinear Constraint

Alternatively, Equation (7.21) can be restated as minimization of a convex function with a nonlinear constraint, as follows:

$$J(x_{\mathrm{PK},\phi}, u) = \frac{1}{2}\left\|\hat{f}_{5D}(\cdot,\cdot,\phi) - u\right\|^2 + R(x_{\mathrm{PK},\phi}) \text{ s.t. } u = P\left(x_{\mathrm{PK},\phi}; C_a(t), C_p(t)\right) \tag{7.42}$$

Then ADMM can be applied by using Taylor linearizations of $P(\cdot)$ in each of the updates for $x_{\mathrm{PK},\phi}$ and u, as described in [129]. In this approach, the u-update would also be amenable to voxel-wise parallelization.

7.4 Indirect Pharmacokinetic-Respiratory map Estimation

Here we introduce a related but philosophically distinct estimation problem. Rather than estimating N_{resp} distinct respiratory maps separately (i.e., $\{x_{\text{PK},\phi}\}_{\phi=1,\ldots,N_{\text{resp}}}$), one could jointly estimate all of the respiratory phases of the pharmacokinetic maps. This approach allows us to exploit redundant information across respiratory phases and assess the benefit of applying respiratory regularizers for estimation of pharmacokinetic maps. The new pharmacokinetic-respiratory map is related to $x_{\text{PK},\phi}$ as follows:

$$x_{\text{PK}}(\vec{r},\kappa,\phi) \triangleq \begin{bmatrix} x_{\text{PK},1}(\vec{r},\kappa) \\ \vdots \\ x_{\text{PK},\text{N}_{\text{resp}}}(\vec{r},\kappa) \end{bmatrix}, \tag{7.43}$$

which one could estimate with the following cost function:

$$J(x_{\text{PK}}) = \frac{1}{2}\left\| \hat{f_{\text{5D}}} - P\left(x_{\text{PK}}\right)\right\|^2 + R(x_{\text{PK}}). \tag{7.44}$$

The operator $P_r : \mathbb{R}^{N_r N_{\text{PK}} N_{\text{resp}}} \mapsto \mathbb{C}^{N_r N_t N_{\text{resp}}}$ now performs liver pharmacokinetic modeling and MR physics independently for each of the respiratory phases, $\phi = 1,\ldots,N_{\text{resp}}$. This is effectively applying $P(\cdot)$ to each respiratory phase: $P_r = \mathbf{I}_{N_{\text{resp}}} \otimes P$. This formulation allows for all of the regularizers described in Section

7.3, as well as motion model-informed respiratory regularization described in Section 6.2.2. We expect that exploiting redundancy across the respiratory dimension could improve the accuracy of the pharmacokinetic parameter estimates, and that this method could outperform both gated indirect pharmacokinetic parameter estimation of Section 7.3 as well as conventional indirect pharmacokinetic parameter estimation methods.

One could compare the three methods in simulation, using a digital respiratory liver pharmacokinetic phantom constructed from XCAT as well as from in vivo free-breathing patient data. Simulation experiments could be assessed based on pharmacokinetic parameter error, as well as computational speed to reach certain levels of error. Patient data results could be assessed qualitatively based on pharmacokinetic map quality and presence of motion artifacts.

7.5 Pharmacokinetic-Respiratory Map Estimation

Exploration of motion model-informed regularization in the 5D MRI problem in Chapter 6 and pharmacokinetic parameter regularization methods in Section 7.3 pave the way for a novel object model, a pharmacokinetic -respiratory map (PKRM) model. This proposed method builds upon the previous proposed algorithms in Sections 6.2.2 and 7.3 to directly estimate pharmacokinetic parameters from k-space data.

Let the unknown object be $x_{\text{PKR}}(\vec{r}, \kappa, \phi)$ where κ describes a set of pharmacokinetic parameters for each N_r voxel in each of $\phi = 1, \ldots, N_{\text{resp}}$ respiratory states. Like

x_{PK}, this model allows for regularization across respiratory phases as well as across pharmacokinetic parameters.

Compared to previous direct pharmacokinetic estimation methods [58] [59], this work is differentiated by parameterization of respiration. By expanding the unknown object along the respiratory dimension, we seek to detangle the data belonging to different respiratory states while allowing for some regularizing influence between respiratory states. This method is also distinct from the method proposed above in Section 7.4 by focusing on direct estimation from raw k-space data.

This concept is analogous to 5D MRI methods, such as XD-GRASP [56], with the time dimension replaced by a pharmacokinetic dimension. This could permit solving for fewer unknowns in a potentially better conditioned problem, because $N_{\mathrm{pk}} \ll N_f$, as well as leveraging the benefits of direct pharmacokinetic parameter estimation.

The direct estimation paradigm replaces the conventional two-stage problem, allowing one to encourage desirable pharmacokinetic parameter characteristics in the fitting of the data to dynamic images. Overall, the method may also be more efficient and parsimonious than the conventional two-stage method. One could investigate whether the direct estimation permits better liver pharmacokinetic quantification. Previous studies have shown qualitative benefits of direct pharmacokinetic parameter estimation for in vivo brain and prostate DCE studies [58] [59].

Let $P(\cdot)$ represent the extrapolation of pharmacokinetic parameters into time series, acting on each respiratory state. The cost function one could use to estimate the

pharmacokinetic-respiratory map is:

$$\hat{x}_{\text{PKR}} = \underset{x_{\text{PKR}}}{\text{argmin}} \frac{1}{2} \left\| y - \mathbf{FS}P(x_{\text{PKR}}) \right\|^2 + R(x_{\text{PKR}}). \tag{7.45}$$

In this case, $\mathbf{F} = \mathbf{I}_{N_c N_t} \otimes \begin{bmatrix} \mathbf{F}_1 & & \\ & \ddots & \\ & & \mathbf{F}_{N_{\text{resp}}} \end{bmatrix}$ and $\mathbf{S} = \mathbf{I}_{N_{\text{resp}} N_t} \otimes \tilde{\mathbf{S}}$. These are the same system matrices as $\mathbf{F}_{5\text{D}}$ and $\mathbf{F}_{5\text{D}}$ in Chapter 6 but are renamed here for simplicity. Regularization terms in $R(x_{\text{PKR}})$ can include extensions of the previously discussed regularization terms:

- spatial regularization for each respiratory phase and pharmacokinetic parameter

$$R_s(x_{\text{PKR}}) = \lambda_s \sum_{\kappa=1}^{N_{\text{pk}}} \sum_{\phi=1}^{N_{\text{resp}}} \sum_{\vec{r}}^{N_r} \sum_{\vec{r'} \in \mathcal{N}_r} \psi_s \left(x_{\text{PKR}}(\vec{r}, \kappa, \phi) - x_{\text{PKR}}(\vec{r} - \vec{r'}, \kappa, \phi) \right)$$

- regularization across pharmacokinetic parameter boundaries

$$R_{\text{PKTV}}(x_{\text{PKR}}) = \lambda_{\text{PKTV}} \sum_{\phi=1}^{N_{\text{resp}}} \sum_{\vec{r}}^{N_{\text{mask}}} \sqrt{\sum_{\kappa=1}^{N_{\text{pk}}} \left(\frac{\sum_{\vec{r'} \in \mathcal{N}_r} x_{\text{PKR}}(\vec{r}, \kappa, \phi) - x_{\text{PKR}}(\vec{r} - \vec{r'}, \kappa, \phi)}{p(\kappa)} \right)^2}$$

- motion model-informed respiratory regularization

$$R_{\text{mm}}(x_{\text{PKR}}) = \lambda_{\text{mm}} \sum_{\vec{r}=\vec{0}}^{N_r} \sum_{\kappa=1}^{N_{\text{pk}}} \sum_{\phi=1}^{N_{\text{resp}}} \psi_m \left(x_{\text{PKR}}(\vec{r}, \kappa, \phi) - x_{\text{PKR}} \left(W_{\phi,\phi-1}(\vec{r}), \kappa, \phi - 1 \right) \right)$$

7.5.1 Minimization Algorithms

One could apply many of the algorithms listed in Section 7.3.2 to the PKRM estimation problem. Here we discuss implementation details specifically relevant for the direct PKRM estimation problem.

7.5.1.1 Initialization of the PKRM Problem

Like the indirect estimation problem in Section 7.3, the initial estimate for the pharmacokinetic-respiratory map requires the estimation of pharmacokinetic parameters from gated datasharing images. The compute time could be reduced by leveraging the precomputed motion model to combine the respiratory dynamic images into a single roughly denoised, motion-compensated dynamic image. The computation time could be further reduced by doing the variable projection for each of the N_r voxels in parallel. Figure 7.3 illustrates the pipeline for this process.

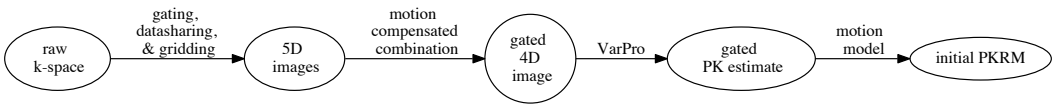


Figure 7.3: Initialization process for PKRM estimation.

7.5.1.2 Variable Splitting Approaches

To design spatially separable inner problems and leverage parallelization, additional variable splits are necessary for solving (7.45). Now we combine variable splitting methods from Sections 7.2.1 and 7.3.2 for a tractable approach for the direct PKRM problem. In this section we describe a cost function with fewer regularization terms to illustrate useful variable splitting schemes. The final PKRM estimation problem is likely to include additional regularization terms and variable splits. To facilitate variable splitting, we express the motion model-informed regularization as finite-differences across a motion-compensated pharmacokinetic map:

$$J(x_{\text{PKR}}) = \frac{1}{2}\left\|y - \mathbf{F}\mathbf{S}P(x_{\text{PKR}})\right\|^2 + \lambda_s R_{\text{spat}}(x_{\text{PKR}}) + \lambda_m R_{\text{mm}}(x_{\text{PKR}}) \tag{7.46}$$

$$\begin{aligned} J(x_{\text{PKR}}) = &\frac{1}{2}\left\|y - \mathbf{F}\mathbf{S}P(x_{\text{PKR}})\right\|^2 \\ &+ \lambda_s \sum_{\kappa=1}^{N_{\text{pk}}} \sum_{\phi=1}^{N_{\text{resp}}} \sum_{\vec{r}}^{N_r} \sum_{\vec{r'}\in\mathcal{N}_r} \left\| x_{\text{PKR}}(\vec{r},\kappa,\phi) - x_{\text{PKR}}(\vec{r}-\vec{r'},\kappa,\phi) \right\|_1 \\ &+ \lambda_m \sum_{\vec{r}=\vec{0}}^{N_r} \sum_{\kappa=1}^{N_{\text{pk}}} \sum_{\phi=1}^{N_{\text{resp}}} \psi\left(x_{\text{PKR}}(\vec{r},\kappa,\phi) - x_{\text{PKR}}\left(W_{\phi,\phi-1}(\vec{r}),\kappa,\phi-1\right)\right) \end{aligned} \tag{7.47}$$

$$J(x_{\text{PKR}}) = \frac{1}{2}\left\|y - \mathbf{F}\mathbf{S}P(x_{\text{PKR}})\right\|^2 + \lambda_s \left\|\mathbf{C}_s x_{\text{PKR}}\right\|_1 + \lambda_m \psi\left(\mathbf{C}_r \tilde{\mathbf{W}} x_{\text{PKR}}\right) \tag{7.48}$$

As in Section 6.2.2, $\tilde{\mathbf{W}}x_{\text{PKR}} \in \mathbb{R}^{N_r \times N_{\text{PK}} \times N_{\text{resp}}}$ is the pharmacokinetic-respiratory map with all respiratory phases converted to $\phi = 1$ via the precomputed motion model.

The following variable splitting scheme incorporates the useful variable splits of Sec-

tion 7.2.1 along with an additional split to isolate the nonlinear portion of the system model:

$$J(x_{\text{PKR}}, u, v, w, z) = \frac{1}{2}\left\|y - \mathbf{FS}w\right\|^2 + \lambda_s \left\|u\right\|_1 + \lambda_m \psi\left(v\right) \tag{7.49}$$

$$\text{s.t. } u = \mathbf{C}_s x_{\text{PKR}};\ v = \mathbf{C}_r \tilde{\mathbf{W}}\left(x_{\text{PKR}}\right);\ w = P(z);\ z = x_{\text{PKR}} \tag{7.50}$$

This leads to the following Augmented Lagrangian function:

$$\begin{aligned}\mathcal{L}(x_{\text{PKR}}, u, v, w, z) = {} & \frac{1}{2}\left\|y - \mathbf{FS}w\right\|^2 + \lambda_s \left\|u\right\|_1 + \lambda_m \psi\left(v\right) \\ & + \frac{\mu_u}{2}\left\|u - \mathbf{C}_s x_{\text{PKR}} - \eta_u\right\|^2 + \frac{\mu_v}{2}\left\|v - \mathbf{C}_r \tilde{\mathbf{W}}(x_{\text{PKR}}) - \eta_v\right\|^2 \\ & + \frac{\mu_w}{2}\left\|w - P(z) - \eta_w\right\|^2 + \frac{\mu_z}{2}\left\|z - x_{\text{PKR}} - \eta_z\right\|^2,\end{aligned} \tag{7.51}$$

and the following alternating variable updates:

$$x_{\text{PKR}}^{(n+1)} = H_x^{-1}\left(\mu_u \mathbf{C}_s'(u^{(n)} - \eta_u^{(n)}) + \mu_v \mathbf{C}_r' \tilde{\mathbf{W}}'(v^{(n)} - \eta_v^{(n)}) + \mu_z(z^{(n)} - \eta_z^{(n)})\right) \quad (7.52)$$

$$H_x = \mu_u \mathbf{C}_s'\mathbf{C}_s + \mu_v \tilde{\mathbf{W}}^{-1}\mathbf{C}_r'\mathbf{C}_r\tilde{\mathbf{W}} + \mu_z \mathbf{I}_{N_r N_{\text{PK}} N_{\text{resp}}} \quad (7.53)$$

$$u^{(n+1)} = \text{soft}\left(\frac{\mu_u}{\lambda_s}, \mathbf{C}_s x_{\text{PKR}}^{(n+1)} + \eta_u^{(n)}\right) \quad (7.54)$$

$$v^{(n+1)} = \text{shrink}\left(\frac{\mu_v}{\lambda_m}, \mathbf{C}_r\tilde{\mathbf{W}}(x_{\text{PKR}}^{(n+1)}) + \eta_v^{(n)}\right) \quad (7.55)$$

$$w^{(n+1)} = H_w^{-1}\left(\mathbf{S}'\mathbf{F}'y + \mu_w(P(z^{(n)}) + \eta_w^{(n)})\right) \quad (7.56)$$

$$H_w = \mathbf{S}'\mathbf{F}'\mathbf{F}\mathbf{S} + \mu_w \mathbf{I}_{N_r N_t N_{\text{resp}}} \quad (7.57)$$

$$z^{(n+1)} = \underset{z}{\text{argmin}} \frac{\mu_w}{2}\left\| \begin{bmatrix} w^{(n+1)} - \eta_w^{(n)} \\ \sqrt{\frac{\mu_z}{\mu_w}}\left(x_{\text{PKR}}^{(n+1)} + \eta_z^{(n)}\right) \end{bmatrix} - \begin{bmatrix} P(z^{(n)}) \\ \sqrt{\frac{\mu_z}{\mu_w}} z^{(n)} \end{bmatrix} \right\|^2 \quad (7.58)$$

One could solve the updates for x_{PKR} and w with PCG, and the update for z with variable projection pixel-by-pixel. Dual variables η_u, η_v, η_w, and η_z are updated in the usual fashion with the constraints' errors.

7.5.2 Segmented Direct Pharmacokinetic Parameter Estimation

One major challenge with direct pharmacokinetic parameter estimation for the liver is that the FOV for abdominal imaging includes many other organs and tissues that do not subscribe to the liver pharmacokinetic model. To work around this, one could partition the voxels into multiple classes and applying appropriate datafit and regularization terms to each class. This mask could be manually selected from initial

data-sharing estimates or automatically estimated from the raw k-space data.

Consider the more generalized reconstruction problem below, in which the embedding operator $\mathbf{M}_i$ handles the voxels belonging to each segmented region, $\mathcal{S}_i$, $i = 1, \ldots, N_{\text{class}}$. Segmented voxels, $\mathcal{S}_i$, partition all of the spatial coordinates $\vec{r} = 1, \ldots, N_r$. We introduce a generalization of x_{PKR}, a set of coefficients for each class, $c_i(\vec{r}, a, \phi)$. These coefficients are only defined over voxels segmented into the ith class, $\vec{r} \in \mathcal{S}_i$, but across all respiratory phases, $\phi = 1, \ldots, N_{\text{resp}}$. The second dimension, a represents the number of coefficients used to represent the dynamics: coefficients of a low rank basis or the number of nonlinear pharmacokinetic parameters in a liver voxel. The size of this dimension varies depending on the class.

Then the segmented PKRM cost function is:

$$\{\hat{c_i}\}_{i=1,\ldots,N_{\text{class}}} = \text{argmin} \frac{1}{2} \left\| y - \mathbf{FS} \sum_{i=1}^{N_{\text{class}}} \mathbf{M}_i c_i \right\|^2 + \sum_{i=1}^{N_{\text{class}}} R_i(c_i) \tag{7.59}$$

with one arbitrary class assigned to the liver pharmacokinetic model,

$$c_1 = P(x_{\text{PKR}}(\vec{r}, \cdot, \cdot)), \text{ for } \vec{r} \in \mathcal{S}_1. \tag{7.60}$$

Once could first implement this segmented PKRM problem segmenting only between liver and non-liver voxels. The liver voxels would be modeled using the dual-input, single-compartment model, and non-liver voxel dynamics would be estimated with a temporally low-rank object.

class index	anatomy	datafit stipulations	regularizing terms
1	liver	$c_1 = P(x_{\mathrm{PKR}}(\vec{r}, \cdot, \cdot))$	R_{spat}, R_{PKTV}, R_{mm}
2	non-liver	-	$\lambda_{\mathrm{LR}} \left\| c_2 \right\|_*$

Table 7.1: Description of two-class segmentation framework for PKRM estimation.

The low-rank prior has been previously applied globally and locally to DCE image reconstruction problems, with good image quality results [6]. Low-rank approximation also improves the conditioning of the estimation problem, resulting in denoising effects as well as computational speed gains. Here, one might seek to balance the benefits of low-rank approximation with caution in underfitting the non-liver regions.

The danger in underfitting the non-liver regions, by parameterizing them with too few temporal dimensions, lies with the spatial mixing of the Fourier encoding operator. Datafit residuals in one class of pixels become included in the model for other classes of other pixels. The accomodation of liver pharmacokinetic model residuals by non-liver voxels is not problematic since those voxels are not used for diagnosis or radiation therapy assessment. We also assume that the dual-input, single compartment liver model is a sufficiently detailed model. However, the converse situation, in which liver voxels are corrupted by non-liver underfitting should be carefully avoided.

7.5.2.1 Low-Rank Considerations

In the above segmented PKRM estimation problem, we penalized rank using the nuclear norm. Usually, the nuclear norm is applied to the Casorati matrix of a dynamic image. Here, one could choose between several options:

Sum of Nuclear Norm of Each Respiratory Phase

$$\|c_2\|_* = \sum_{\phi=1}^{N_{\text{resp}}} \|c_2(\cdot,\cdot,\phi)\|_* \tag{7.61}$$

Nuclear Norm of Reshaped Object

$$\|c_2\|_* = \left\| \left[c_2(\cdot,\cdot,1) \cdots c_2(\cdot,\cdot,N_{\text{resp}}) \right] \right\|_* \tag{7.62}$$

Additionally, one could describe a penalty of the nuclear norm rather than an explicit low-rank model. A valid alternative would stipulate the following structure of c_2 for a predetermined rank ρ:

$$c_2(\cdot,\cdot,\phi) = \underbrace{U(\cdot,\cdot,\phi)}_{\mathbb{R}^{N_r \times \rho}} \underbrace{V(\cdot,\cdot,\phi)^T}_{\mathbb{R}^{\rho \times N_t}} \text{ for } \phi = 1,\ldots,N_{\text{resp}}. \tag{7.63}$$

One benefit of this explicitly spatially-temporally separable model is that enforcing dimensionality reduction yields a smaller unknown component c_2 and possibly a simpler minimization problem.

7.5.2.2 Minimization Algorithms

This section describes a general minimization strategy for the two-class segmented PKRO problem outlined in Section 7.5.2, in which one could seek to estimate two sets

of coefficients, c_1(for liver) and c_2 (for non-liver). One could re-express the problem as:

$$(\hat{x_{\text{PKR}}}, \hat{c}_2) = \underset{x_{\text{PKR}}, c_2}{\operatorname{argmin}} \frac{1}{2} \left\| y - \mathbf{FS} \left(\mathbf{M}_1 P(x_{\text{PKR}}) + \mathbf{M}_2 c_2 \right) \right\|_2^2 + \lambda_s \psi_s(\mathbf{C}_s x_{\text{PKR}}) + \lambda_{\text{mm}} \psi_m(\mathbf{C}_r \tilde{\mathbf{W}} x_{\text{PKR}}) + \lambda_{\text{LR}} \left\| c_2 \right\|_* \tag{7.64}$$

Applying alternating minimization to the two unknowns yields:

$$x_{\text{PKR}}^{(n+1)} = \underset{x_{\text{PKR}}}{\operatorname{argmin}} \frac{1}{2} \left\| \underbrace{\left(y - \mathbf{FSM}_2 c_2^{(n)} \right)}_{\tilde{y}_1} - \mathbf{FSM}_1 P(x_{\text{PKR}}^{(n)}) \right\|_2^2 + \lambda_s \psi_s(\mathbf{C}_s x_{\text{PKR}}^{(n)}) + \lambda_{\text{mm}} \psi_m(\mathbf{C}_r \tilde{\mathbf{W}} x_{\text{PKR}}^{(n)}) \tag{7.65}$$

$$c_2^{(n+1)} = \underset{c_2}{\operatorname{argmin}} \frac{1}{2} \left\| \underbrace{\left(y - \mathbf{FSM}_1 P(x_{\text{PKR}}^{(n+1)} \right)}_{\tilde{y}_2} - \mathbf{FSM}_2 c_2^{(n)} \right\|_2^2 + \lambda_{\text{LR}} \left\| c_2^{(n)} \right\|_* \tag{7.66}$$

Equation (7.65) could be solved with variable splitting techniques described in Section 7.5.1.2.

Equation (7.66) could be tackled with an inner iterative method involving variable splitting and singular value thresholding (SVT) [130]. Consider the following equivalent problem:

$$\hat{c}_2, \hat{u} = \underset{c_2,u}{\operatorname{argmin}} \frac{1}{2} \left\| \tilde{y}_2 - \mathbf{FSM}_2 c_2 \right\|_2^2 + \lambda_{\mathrm{LR}} \left\| u \right\|_* \text{ s.t. } u = c_2 \tag{7.67}$$

Then the corresponding Augmented Lagrangian function is:

$$\mathcal{L}\left(c_2, u; \mu\right) = \frac{1}{2} \left\| \tilde{y}_2 - \mathbf{FSM}_2 c_2 \right\|_2^2 + \lambda_{\mathrm{LR}} \left\| u \right\|_* + \frac{\mu}{2} \left\| u - c_2 - \eta \right\|_2^2. \tag{7.68}$$

Alternating minimization yields the following inner updates:

$$c_2^{(m+1)} = \mathbf{H}_{c_2}^{-1} \left(\mathbf{M}_2' \mathbf{S}' \mathbf{F}' \tilde{y}_2 + \mu \left(u^{(m+1)} - \eta^{(m+1)} \right) \right) \tag{7.69}$$

$$\mathbf{H}_{c_2} = \mathbf{M}_2' \mathbf{S}' \mathbf{F}' \mathbf{F} \mathbf{S} \mathbf{M}_2 + \mu \mathbf{I} \tag{7.70}$$

$$u^{(m+1)} = \mathcal{D}_{\lambda_{\mathrm{LR}}} \left(c_2^{(m+1)} + \eta^{(m)} \right) \tag{7.71}$$

$$\eta^{(m+1)} = \eta^{(m)} - \left(u^{(m+1)} - c_2^{(m+1)} \right) \tag{7.72}$$

The SVT operator $\mathcal{D}_\tau$ operates on a matrix $\mathbf{X}$ with SVD: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}'$ as follows:

$$\mathcal{D}_\tau(\mathbf{X}) \triangleq \mathbf{U} \mathcal{D}_\tau(\mathbf{\Sigma}) \mathbf{V}^*, \qquad \mathcal{D}_\tau(\mathbf{\Sigma}) = \operatorname{diag}\left(\max(\sigma_i - \tau, 0) \right). \tag{7.73}$$

In other words, we perform soft-thresholding of the singular values of $\mathbf{X}$, $\sigma_1, \ldots, \sigma_\rho$. As discussed in Section 7.5.2.1, there are multiple ways one could define the low-rank nature of c_2, which would affect the exact implementation of (7.71). The inner c_2 update in (7.69) could be solved with PCG.

7.5.2.3 Multi-class Segmentation

If the two-class segmentation framework is insufficient for modeling abdominal dynamics, one could extend the number of classes. The embedding mask for class 1, M_1 enforces that the pixels in the liver ROI will be mapped to pharmacokinetic parameters within the liver. Other classes can map contrast-enhanced organs to the appropriate specific case of liver pharmacokinetics, and yet another class can represent zero-signal air. The air class is akin to masking of the 5D object, implemented in Section 6.2.2.

The segmentation system could be expanded to better model different classes of tissues, enumerated in Table 7.2.

class	**characteristics**	**anatomy**	**datafit stipulations**	**regularizing terms**
1	liver PK	liver	$c_1 = P(x_{\text{PKR}}(\vec{r}, \cdot, \cdot))$	R_{spat}, R_{PKTV}, R_{mm}
2	non-DCE	air, bone	-	$\lambda_{\text{LR}} \left\|c_2\right\|_*$, R_{spat}, R_t, R_{mm}
3	liver inputs	aorta, portal vein	$c_1 = P(x_{\text{PKR}}(\vec{r}, \cdot, \cdot))$	$k_{1a} = 1, k_{1p} = 0, k_2 = 0$
4	miscellaneous	heart, bowels, kidneys	-	$\lambda_{\text{LR}} \left\|c_2\right\|_*$

Table 7.2: Description of multi-class segmentation framework for PKRM estimation.

7.5.2.4 Automatic Segmentation Via Subspace Clustering

One could perform automatic segmentation on the datashared, motion-combined, dynamic image, one of the intermediate steps of the initialization process, described

in Figure 7.3. To classify the voxels of a single respiratory phase, one could apply existing subspace clustering methods, such as Sparse Subspace Clustering (SSC) [131] on the the dynamic image.

Preliminary results suggest one could also encourage spatial contiguity in the class memberships by including the spatial coordinates along with the temporal data for each voxel as an input to the subspace clustering methods.

In summary, 5D MRI presents many opportunities for improved modeling and accelerated reconstruction algorithms. Combining 5D MRI with pharmacokinetic parameter estimation, a direct formulation connecting k-space measurements to pharmacokinetic parameter maps also presents a rich area for future study.

Bibliography

Bibliography

[1] M. J. Allison, S. Ramani, and J. A. Fessler, "Regularized MR coil sensitivity estimation using augmented Lagrangian methods," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 394–7, 2012.

[2] K. A. Khalsa, *Temporal regularization use in dynamic contrast-enhanced MRI*. PhD thesis, Univ. of Michigan, Ann Arbor, MI, 48109-2122, Ann Arbor, MI, 2011.

[3] S. L. Murphy, J. Xu, and K. D. Kochanek, "Deaths: final data for 2010.," *National vital statistics reports: from the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System*, vol. 61, no. 4, pp. 1–117, 2013.

[4] D. J. Brenner, R. Doll, D. T. Goodhead, E. J. Hall, C. E. Land, J. B. Little, J. H. Lubin, D. L. Preston, R. Julian Preston, J. S. Puskin, E. Ron, R. K. Sachs, J. M. Samet, R. B. Setlow, and M. Zaide, "Cancer risks attributable to low doses of ionizing radiation: assessing what we really know," *Proc. Natl. Acad. Sci.*, vol. 100, pp. 13761–6, Nov. 2003.

[5] H. Chandarana, T. K. Block, A. B. Rosenkrantz, R. P. Lim, D. Kim, D. J. Mossa, J. S. Babb, B. Kiefer, and V. S. Lee, "Free-breathing radial 3D fat-suppressed T1-weighted gradient echo sequence: A viable alternative for contrast-enhanced liver imaging in patients unable to suspend respiration," *Invest. Radiol.*, vol. 46, pp. 648–53, Oct. 2011.

[6] J. Cheng, M. Uecker, M. Alley, S. Vasanawala, J. Pauly, and M. Lustig, "Free-breathing pediatric imaging with nonrigid motion correction and parallel imaging," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 312, 2013.

[7] H. Chandarana, L. Feng, T. K. Block, A. B. Rosenkrantz, R. P. Lim, J. S. Babb, D. K. Sodickson, and R. Otazo, "Free-breathing contrast-enhanced multiphase

MRI of the liver using a combination of compressed sensing, parallel imaging, and golden-angle radial sampling," *Invest. Radiol.*, vol. 48, pp. 10–6, Jan. 2013.

[8] M. Le, S. Ramani, and J. A. Fessler, "An efficient variable splitting based algorithm for regularized SENSE reconstruction with support constraint," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 2654, 2013.

[9] M. Le and J. A. Fessler, "Efficient, convergent SENSE MRI reconstruction for non-periodic boundary conditions via tridiagonal solvers," *IEEE Trans. Computational Imaging*, vol. 3, pp. 11–21, Mar. 2017.

[10] M. Le and J. A. Fessler, "Spline temporal basis for improved pharmacokinetic parameter estimation in SENSE DCE-MRI," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 3698, 2015.

[11] D. G. Nishimura, "Principles of magnetic resonance imaging," 1996. Unpublished textbook.

[12] Z.-P. Liang and P. C. Lauterber, *Principles of magnetic resonance imaging.* New York: IEEE, 2000.

[13] C. B. Ahn, J. H. Kim, and Z. H. Cho, "High-speed spiral-scan scho planar NMR imaging - I," *IEEE Trans. Med. Imag.*, vol. 5, pp. 2–7, Mar. 1986.

[14] R. S. Likes, "Moving gradient zeugmatography," Dec. 22 1981. US Patent 4,307,343.

[15] V. Rasche, R. Proksa, R. Sinkus, P. Börnert, and H. Eggers, "Resampling of data between arbitrary grids using convolution interpolation," *IEEE Trans. Med. Imag.*, vol. 18, pp. 385–92, May 1999.

[16] H. Schomberg and J. Timmer, "The gridding method for image reconstruction by Fourier transformation," *IEEE Trans. Med. Imag.*, vol. 14, pp. 596–607, Sept. 1995.

[17] J. A. Fessler, "On NUFFT-based gridding for non-Cartesian MRI," *J. Mag. Res.*, vol. 188, pp. 191–5, Oct. 2007.

[18] 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.

[19] D. K. Sodickson, M. A. Griswold, and P. M. Jakob, "SMASH imaging," *Magn Reson Imaging Clin N Am.*, vol. 7, pp. 237–54, May 1999.

[20] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Mag. Res. Med.*, vol. 47, pp. 1202–10, June 2002.

[21] 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.

[22] K. F. King and A. Angelos, "SENSE image quality improvement using matrix regularization," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 1771, 2001.

[23] L. Ying, B. Liu, M. Steckner, G. Wu, M. Wu, and S.-J. Li, "A statistical approach to SENSE regularization with arbitrary k-space trajectories," *Mag. Res. Med.*, vol. 60, pp. 414–21, Aug. 2008.

[24] A. Majumdar and R. K. Ward, "Nuclear norm-regularized SENSE reconstruction," *Mag. Res. Im.*, vol. 30, pp. 213–21, Feb. 2012.

[25] B. Liu, K. King, M. Steckner, J. Xie, J. Sheng, and L. Ying, "Regularized sensitivity encoding (SENSE) reconstruction using Bregman iterations," *Mag. Res. Med.*, vol. 61, pp. 145–52, Jan. 2009.

[26] 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.

[27] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[28] M. McCool and E. Fiume, "Hierarchical poisson disk sampling distributions," in *Proceedings of the conference on Graphics interface*, vol. 92, pp. 94–105, 1992.

[29] J. X. Ji, "Compressive sensing imaging with randomized lattice sampling: Applications to fast 3D MRI," in *Proc. Int'l. Conf. IEEE Engr. in Med. and Biol. Soc.*, pp. 3736–9, 2011.

[30] R. M. Lebel, J. Jones, J.-C. Ferre, M. Law, and K. S. Nayak, "Highly accelerated dynamic contrast enhanced imaging," *Mag. Res. Med.*, vol. 71, pp. 635–44, Feb. 2014.

[31] J. Y. Cheng, T. Zhang, M. T. Alley, M. Lustig, S. S. Vasanawala, and J. M. Pauly, "Variable-density radial view-ordering and sampling for time-optimized

3D Cartesian imaging," in *ISMRM Workshop on Data Sampling and Image Reconstruction*, 2013.

[32] J. J. Van Vaals, M. E. Brummer, W. T. Dixon, H. H. Tuithof, H. Engels, R. C. Nelson, B. M. Gerety, J. L. Chezmar, and J. A. Den Boer, "Keyhole method of accelerating imaging of contrast agent uptake," *J. Mag. Res. Im.*, vol. 3, pp. 671–5, July 1993.

[33] R. A. Jones, O. Haraldseth, T. B. Müller, P. A. Rinck, and A. N. Øksendal, "K-space substitution: A novel dynamic imaging technique," *Mag. Res. Med.*, vol. 29, pp. 830–4, June 1993.

[34] H. K. Song and L. Dougherty, "K-space weighted image contrast (KWIC) for contrast manipulation in projection reconstruction MRI," *Mag. Res. Med.*, vol. 44, pp. 825–32, Dec. 2000.

[35] 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.

[36] R. Glowinski and A. Marrocco, "Sur lapproximation par elements nis dordre un, et la resolution par penalisation-dualite dune classe de problemes de dirichlet nonlineaires, rev. francaise daut," *Inf. Rech. Oper.*, vol. R-2, pp. 41–76, 1975.

[37] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[38] S. Ramani and J. A. Fessler, "Regularized parallel MRI reconstruction using an alternating direction method of multipliers," in *Proc. IEEE Intl. Symp. Biomed. Imag.*, pp. 385–8, 2011.

[39] M. Sasaki, E. Shibata, Y. Kanbara, and S. Ehara, "Enhancement effects and relaxivities of gadolinium-DTPA at 1.5 versus 3 Tesla: A phantom study," *Magnetic Resonance in Medical Sciences*, vol. 4, no. 3, pp. 145–9, 2005.

[40] G. J. Stanisz, E. E. Odrobina, J. Pun, M. Escaravage, S. J. Graham, M. J. Bronskill, and R. M. Henkelman, "T_1, T_2 relaxation and magnetization transfer in tissue at 3T," *Mag. Res. Med.*, vol. 54, pp. 507–12, Sept. 2005.

[41] P. S. Tofts, G. Brix, D. L. Buckley, J. L. Evelhoch, E. Henderson, M. V. Knopp, H. B. Larsson, T. Y. Lee, N. A. Mayr, G. J. Parker, R. E. Port, J. Taylor, and R. M. Weisskoff, "Estimating kinetic parameters from dynamic contrast-

enhanced T(1)-weighted MRI of a diffusable tracer: standardized quantities and symbols," *J. Mag. Res. Im.*, vol. 10, pp. 223–32, Sept. 1999.

[42] M. C. Schabel, G. R. Morrell, K. Y. Oh, C. A. Walczak, R. B. Barlow, and L. A. Neumayer, "Pharmacokinetic mapping for lesion classification in dynamic breast MRI," *Journal of Magnetic Resonance Imaging*, vol. 31, no. 6, pp. 1371–1378, 2010.

[43] V. V. Chebrolu, D. D. Shanbhag, S. Thiruvenkadam, S. Kaushik, U. Patil, P. Hervo, S. N. Gupta, and R. Mullick, "Automated arterial input function detection in ascending aorta for breast DCE-MRI," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 0414, 2014.

[44] G. J. Parker, A. Macdonald, S. Cheung, D. L. Buckley, A. Jackson, G. Jayson, and C. Roberts, "An experimentally-derived functional form for a population-averaged high temporal resolution arterial input function," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 2100, 2005.

[45] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Prob.*, vol. 19, pp. R1–26, Apr. 2003.

[46] R. Materne, A. M. Smith, F. Peeters, J. P. Dehoux, A. Keyeux, Y. Horsmans, and B. E. V. Beers, "Assessment of hepatic perfusion parameters with dynamic MRI," *Mag. Res. Med.*, vol. 47, pp. 135–42, Jan. 2002.

[47] V. Boldea, G. C. Sharp, S. B. Jiang, and D. Sarrut, "4D-CT lung motion estimation with deformable registration: Quantification of motion nonlinearity and hysteresis," *Med. Phys.*, vol. 35, pp. 1008–18, Mar. 2008.

[48] R. Zeng, J. A. Fessler, J. M. Balter, and P. A. Balter, "Iterative sorting for four-dimensional CT images based on internal anatomy motion," *Med. Phys.*, vol. 35, pp. 917–26, Mar. 2008.

[49] J. Y. Cheng, M. T. Alley, C. H. Cunningham, S. S. Vasanawala, J. M. Pauly, and M. Lustig, "Nonrigid motion correction in 3D using autofocusing with localized linear translations," *Mag. Res. Med.*, vol. 68, no. 6, pp. 1785–97, 2012.

[50] M. W. Jacobson and J. A. Fessler, "Joint estimation of image and deformation parameters in motion-corrected PET," in *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, vol. 5, pp. 3290–4, 2003.

[51] S. Y. Chun and J. A. Fessler, "Joint image reconstruction and nonrigid motion estimation with a simple penalty that encourages local invertibility," in *Proc. SPIE 7258 Medical Imaging 2009: Phys. Med. Im.*, p. 72580U, 2009.

[52] L. Feng, R. Grimm, K. T. Block, H. Chandarana, S. Kim, J. Xu, L. Axel, D. K. Sodickson, and R. Otazo, "Golden-angle radial sparse parallel MRI: Combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI," *Mag. Res. Med.*, vol. 72, pp. 707–17, Sept. 2014.

[53] A. B. Rosenkrantz, C. Geppert, R. Grimm, T. K. Block, C. Glielmi, L. Feng, R. Otazo, J. M. Ream, M. M. Romolo, S. S. Taneja, D. K. Sodickson, and H. Chandarana, "Dynamic contrast-enhanced MRI of the prostate with high spatiotemporal resolution using compressed sensing, parallel imaging, and continuous golden-angle radial sampling: Preliminary experience," *J. Mag. Res. Im.*, vol. 41, pp. 1365–73, May 2015.

[54] H. Chandarana, L. Feng, T. K. Block, A. B. Rosenkrantz, R. P. Lim, D. Chu, D. K. Sodickson, and R. Otazo, "Free-breathing dynamic contrast-enhanced MRI of the liver with radial golden-angle sampling scheme and advanced compressed-sensing reconstruction," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 4206, 2012.

[55] L. Feng, S. Coppo, D. Piccini, R. P. Lim, M. Stuber, D. K. Sodickson, and R. Otazo, "Five-dimensional cardiac and respiratory motion-resolved whole-heart MRI," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 0027, 2015.

[56] L. Feng, H. Chandarana, D. Piccini, J. Ream, D. K. Sodickson, and R. Otazo, "Rapid free-breathing dynamic contrast-enhanced mri using motion-resolved compressed sensing," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 0568, 2015.

[57] H. Chandarana, L. Feng, J. Ream, A. Wang, J. S. Babb, K. T. Block, D. K. Sodickson, and R. Otazo, "Respiratory motion-resolved compressed sensing reconstruction of free-breathing radial acquisition for dynamic liver magnetic resonance imaging," *Invest. Radiol.*, vol. 50, pp. 749–56, Nov. 2015.

[58] Y. Guo, Y. Zhu, S. G. Lingala, R. M. Lebel, and K. Nayak, "Highly accelerated brain DCE MRI with direct estimation of pharmacokinetic parameter maps," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 0573, 2015.

[59] N. Dikaios, S. Arridge, V. Hamy, S. Punwani, and D. Atkinson, "Direct parametric reconstruction from undersampled (k,t)-space data in dynamic contrast enhanced MRI," *Med. Im. Anal.*, vol. 18, pp. 989–1001, Oct. 2014.

[60] J. Dinkel, C. Thieke, C. Plathow, P. Zamecnik, H. Prüm, P. E. Huber, H.-U. Kauczor, H.-P. Schlemmer, and C. M. Zechmann, "Respiratory-induced prostate motion: characterization and quantification in dynamic MRI," *Strahlentherapie und Onkologie*, vol. 187, no. 7, pp. 426–432, 2011.

[61] H. Abbas, B. Chang, and Z. J. Chen, "Motion management in gastrointestinal cancers," *Journal of gastrointestinal oncology*, vol. 5, no. 3, p. 223, 2014.

[62] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithm," *Physica D*, vol. 60, pp. 259–68, Nov. 1992.

[63] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Comp.*, vol. 17, pp. 227–38, Jan. 1996.

[64] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, pp. 303–20, Nov. 1969.

[65] D. P. Bertsekas, "Multiplier methods: A survey," *Automatica*, vol. 12, pp. 133–45, Mar. 1976.

[66] R. Glowinski and P. L. Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. Soc. Indust. Appl. Math., 1989.

[67] 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.

[68] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Im. Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[69] 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.

[70] 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.

[71] J. A. Fessler, "Matlab tomography toolbox," 2004. Available from `http://web.eecs.umich.edu/~fessler/code/`.

[72] 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.

[73] Y. Chen, W. Hager, F. Huang, D. Phan, X. Ye, and W. Yin, "Fast algorithms for image reconstruction with application to partially parallel MR imaging," *SIAM J. Imaging Sci.*, vol. 5, no. 1, pp. 90–118, 2012.

[74] 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.

[75] J. Eckstein, "Parallel alternating direction multiplier decomposition of convex programs," *J. Optim. Theory Appl.*, vol. 80, pp. 39–62, Jan. 1994.

[76] 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.

[77] Y. Chen, X. Ye, and F. Huang, "A novel method and fast algorithm for MR image reconstruction with significantly undersampled data," *Inverse Prob. and Imaging*, vol. 4, pp. 223–40, May 2010.

[78] A. Matakos, S. Ramani, and J. A. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. Im. Proc.*, vol. 22, pp. 2019–29, May 2013.

[79] M. S. C. Almeida and M. A. T. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. Im. Proc.*, vol. 22, pp. 3074–86, Aug. 2013.

[80] 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.

[81] X. Ye, Y. Chen, and F. Huang, "Computational acceleration for MR image reconstruction in partially parallel imaging," *IEEE Trans. Med. Imag.*, vol. 30, pp. 1055–63, May 2011.

[82] B. Fischer and J. Modersitzki, "Fast inversion of matrices arising in image processing," *Numer. Algorithms*, vol. 22, no. 1, pp. 1–11, 1999.

[83] P. J. Davis, *Circulant matrices*. New York: Wiley, 1983.

[84] R. Chan, J. Yang, and X. Yuan, "Alternating direction method for image inpainting in wavelet domains," *SIAM J. Imaging Sci.*, vol. 4, no. 3, pp. 807–26, 2011.

[85] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algorithms*, vol. 51, pp. 67–92, Jan. 2010.

[86] G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, no. 1, pp. 135–47, 1999.

[87] M. Storath and A. Weinmann, "Fast partitioning of vector-valued images," *SIAM J. Imaging Sci.*, vol. 7, no. 3, pp. 1826–52, 2014.

[88] E. Ghadimi, André. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," 2013. Optimization Online.

[89] 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.

[90] M. Filipovic, P.-A. Vuissoz, A. Codreanu, M. Claudon, and J. Felblinger, "Motion compensated generalized reconstruction for free-breathing dynamic contrast-enhanced MRI," *Mag. Res. Med.*, vol. 65, pp. 812–22, Mar. 2011.

[91] T. E. Nichols, J. Qi, E. Asma, and R. M. Leahy, "Spatiotemporal reconstruction of list-mode PET data," *IEEE Trans. Med. Imag.*, vol. 21, pp. 396–404, Apr. 2002.

[92] J. K. Ryu, J.-H. Oh, H.-G. Kim, S. J. Rhee, M. Seo, and G.-H. Jahng, "Estimation of T2* relaxation times for the glandular tissue and fat of breast at 3T MRI system," *Journal of the Korean Society of Magnetic Resonance in Medicine*, vol. 18, no. 1, pp. 1–6, 2014.

[93] C. de Boor, *A practical guide to splines*. New York: Springer Verlag, 1978.

[94] G. Adluru and E. DiBella, "A comparison of L1 and L2 norms as temporal constraints for reconstruction of undersampled dynamic contrast enhanced cardiac scans with respiratory motion," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 340, 2008.

[95] C. Yang, G. S. Karczmar, M. Medved, and W. M. Stadler, "Multiple reference tissue method for contrast agent arterial input function estimation," *Mag. Res. Med.*, vol. 58, pp. 1266–75, Dec. 2007.

[96] R. M. Gray, "Toeplitz and circulant matrices: a review," 2006.

[97] Y. Cao, "The promise of dynamic contrast-enhanced imaging in radiation therapy," in *Seminars in radiation oncology*, vol. 21, pp. 147–156, Elsevier, 2011.

[98] Y. Cao, C. Pan, J. M. Balter, J. F. Platt, I. R. Francis, J. A. Knol, D. Normolle, E. Ben-Josef, R. K. Ten Haken, and T. S. Lawrence, "Liver function after irradiation based on computed tomographic portal vein perfusion imaging," *International Journal of Radiation Oncology* Biology* Physics*, vol. 70, no. 1, pp. 154–160, 2008.

[99] V. W. Wu, M. A. Epelman, H. Wang, H. E. Romeijn, M. Feng, Y. Cao, R. K. Ten Haken, and M. M. Matuszak, "Optimizing global liver function in radiation therapy treatment planning," *Physics in Medicine and Biology*, vol. 61, no. 17, p. 6465, 2016.

[100] A. C. Larson, P. Kellman, A. Arai, G. A. Hirsch, E. McVeigh, D. Li, and O. P. Simonetti, "Preliminary investigation of respiratory self-gating for free-breathing segmented cine MRI," *Mag. Res. Med.*, vol. 53, pp. 159–68, Jan. 2005.

[101] S. Vedam, P. Keall, A. Docef, D. Todor, V. Kini, and R. Mohan, "Predicting respiratory motion for four-dimensional radiotherapy," *Medical physics*, vol. 31, no. 8, pp. 2274–2283, 2004.

[102] P. J. Keall, V. R. Kini, S. S. Vedam, and R. Mohan, "Motion adaptive x-ray therapy: a feasibility study," *Phys. Med. Biol.*, vol. 46, pp. 1–10, Jan. 2001.

[103] T. Zhang, J. Cheng, M. Alley, M. Uecker, M. Lustig, J. Pauly, and S. Vasanawala, "Fast 3D free-breathing abdominal dynamic contrast enhanced MRI with high spatiotemporal resolution," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 0332, 2014.

[104] J. G. Pipe, "Motion correction with PROPELLER MRI: Application to head motion and free-breathing cardiac imaging," *Mag. Res. Med.*, vol. 42, pp. 963–9, Nov. 1999.

[105] P. G. Batchelor, D. Atkinson, P. Irarrazaval, D. L. G. Hill, J. Hajnal, and D. Larkman, "Matrix description of general motion correction applied to multishot images," *Mag. Res. Med.*, vol. 54, pp. 1273–80, Nov. 2005.

[106] M. Usman, D. Atkinson, F. Odille, C. Kolbitsch, G. Vaillant, T. Schaeffter, P. G. Batchelor, and C. Prieto, "Motion corrected compressed sensing for free-breathing dynamic cardiac MRI," *Mag. Res. Med.*, vol. 70, pp. 504–16, Aug. 2013.

[107] L. Feng, L. Axel, H. Chandarana, K. T. Block, D. K. Sodickson, and R. Otazo, "XD-GRASP: Golden-angle radial MRI with reconstruction of extra motion-state dimensions using compressed sensing," *Mag. Res. Med.*, vol. 75, pp. 775–88, Feb. 2016.

[108] J. Liu, X. Zhang, X. Zhang, H. Zhao, Y. Gao, D. Thomas, D. A. Low, and H. Gao, "5D respiratory motion model based image reconstruction algorithm for 4D cone-beam computed tomography," *Inverse Prob.*, vol. 31, p. 115007, Nov. 2015.

[109] D. Atkinson, D. L. G. Hill, P. N. R. Stoyle, P. E. Summers, S. Clare, R. Bowtell, and S. F. Keevil, "Automatic compensation of motion artifacts in MRI," *Magnetic Resonance in Medicine*, vol. 41, no. 1, pp. 163–170, 1999.

[110] A. Fassi, J. Schaerer, M. Fernandes, M. Riboldi, D. Sarrut, and G. Baroni, "Tumor tracking method based on a deformable 4d ct breathing motion model driven by an external surface surrogate," *International Journal of Radiation Oncology* Biology* Physics*, vol. 88, no. 1, pp. 182–188, 2014.

[111] J. M. Blackall, S. Ahmad, M. E. Miquel, J. R. McClelland, D. B. Landau, and D. J. Hawkes, "MRI-based measurements of respiratory motion variability and assessment of imaging strategies for radiotherapy planning," *Phys. Med. Biol.*, vol. 51, pp. 4147–70, Sept. 2006.

[112] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui, "Realistic CT simulation using the 4D XCAT phantom," *Med. Phys.*, vol. 35, pp. 3800–8, Aug. 2008.

[113] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Med. Phys.*, vol. 37, pp. 4902–15, Sept. 2010.

[114] B. Sharif and Y. Bresler, "Physiologically improved ncat phantom (pincat) enables in-silico study of the effects of beat-to-beat variability on cardiac mr," in *Proceedings of the Annual Meeting of ISMRM, Berlin*, vol. 3418, 2007.

[115] D. C. Peters, J. A. Derbyshire, and E. R. McVeigh, "Centering the projection reconstruction trajectory: Reducing gradient delay errors," *Mag. Res. Med.*, vol. 50, pp. 1–6, July 2003.

[116] K. T. Block and M. Uecker, "Simple method for adaptive gradient-delay compensation in radial MRI," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 2816, 2011.

[117] B. Stemkens, R. H. N. Tijssen, B. D. . Senneville, H. D. Heerkens, M. . Vulpen, J. J. W. Lagendijk, and C. A. T. . . Berg, "Optimizing 4-dimensional magnetic resonance imaging data sampling for respiratory motion analysis of pancreatic tumors," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 91, pp. 571–8, Mar. 2015.

[118] J. Liu, P. Spincemaille, N. C. F. Codella, T. D. Nguyen, M. R. Prince, and Y. Wang, "Respiratory and cardiac self-gated free-breathing cardiac CINE imaging with multiecho 3D hybrid radial SSFP acquisition," *Mag. Res. Med.*, vol. 63, pp. 1230–7, May 2010.

[119] P. Spincemaille, J. Liu, T. Nguyen, M. R. Prince, and Y. Wang, "Z intensity-weighted position self-respiratory gating method for free-breathing 3D cardiac CINE imaging," *Mag. Res. Im.*, vol. 29, pp. 861–8, July 2011.

[120] T. Zhang, J. M. Pauly, S. S. Vasanawala, and M. Lustig, "Coil compression for accelerated imaging with Cartesian sampling," *Mag. Res. Med.*, vol. 69, pp. 571–82, Feb. 2013.

[121] S. Poddar, J. D. Newell, and M. Hacob, "Free breathing CINE with low rank aided manifold smoothness regularization," in *Proc. Intl. Soc. Mag. Res. Med.*, p. 3807, 2015.

[122] F. Niu, B. Recht, C. Re, and S. J. Wright, "HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent," in *Neural Info. Proc. Sys.*, pp. 693–701, 2011.

[123] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging," 2016. arxiv 1611.04069.

[124] X. He, J. A. Fessler, L. Cheng, and E. C. Frey, "Regularized image reconstruction algorithms for dual-isotope myocardial perfusion SPECT (MPS) imaging

using a cross-tracer edge-preserving prior," *IEEE Trans. Med. Imag.*, vol. 30, pp. 1169–83, June 2011.

[125] P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Trans. Im. Proc.*, vol. 7, pp. 304–9, Mar. 1998.

[126] K. Chen, E. L. Piccolomini, and F. Zama, "Iterative constrained minimization for vectorial TV image deblurring," *Journal of Mathematical Imaging and Vision*, pp. 1–16, 2015.

[127] B. Zhao, W. Lu, T. K. Hitchens, F. Lam, C. Ho, and Z.-P. Liang, "Accelerated MR parameter mapping with low-rank and sparsity constraints," *Mag. Res. Med.*, vol. 74, pp. 489–98, Aug. 2015.

[128] M. McGaffin and J. A. Fessler, "Edge-preserving image denoising via group coordinate descent on the GPU," *IEEE Trans. Im. Proc.*, vol. 24, pp. 1273–81, Apr. 2015.

[129] M. Benning, F. Knoll, C.-B. Schönlieb, and T. Valkonen, "Preconditioned ADMM with nonlinear operator constraint," 2015. arxiv 1511.00425.

[130] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–82, 2010.

[131] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Trans. Patt. Anal. Mach. Int.*, vol. 35, pp. 2765–81, Nov. 2013.