A Variational Approach to MR Bias Correction

by

Ayres C. Fan

B.S. Electrical Engineering
University of Minnesota, 2000

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology
[February 2003]
January 2003

Signature of Author: _____
Department of Electrical Engineering and Computer Science
January 31, 2003

Certified by: _____
William M. Wells III
Research Scientist, MIT AI Lab
Associate Professor of Radiology,
Harvard Medical School and Brigham and Women's Hospital
Thesis Supervisor

Accepted by: _____
Arthur C. Smith
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

A Variational Approach to MR Bias Correction

by Ayres C. Fan

Submitted to the Department of Electrical Engineering and Computer Science on January 31, 2003 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering and Computer Science

Abstract

Magnetic resonance (MR) imaging has opened up new avenues of diagnosis and treatment that were not previously available. There are a number of artifacts which can arise in the MR imaging process and make subsequent analysis more challenging. Probably the most drastic visual effect is the intensity inhomogeneity caused by spatially-varying signal response in the electrical coil that receives the MR signal. This coil inhomogeneity results in a multiplicative gain field that biases the observed signal from the true underlying signal. A number of techniques exist that attempt to correct this bias field, but none are wholly satisfying. We present an algorithm derived from statistical principles that is based on our knowledge of the physical imaging model. Our algorithm is a variational method that produces nonlinear estimates of the bias field and true image. We regularize our solutions using ℓ_2 norms to ensure smoothness in the bias field and ℓ_p norms to enforce piecewise smoothness in the true image. The latter has an effect similar to an anisotropic filter that reduces the noise and preserves edges. We deal with the nonlinearity in our equations by first using coordinate descent to convert the difficult overall problem into simpler subproblems, and then using fixed-point iterative methods to linearize our equations. This allows us to employ the large body of existing work on fast iterative linear solvers. We also use multiresolution techniques to increase our solver speed. This results in an algorithm that is non-parametric, fast, and robust. We show how to extend our algorithm into a more general framework which allows us to seamlessly handle multiple receiving coils and multiple image pulse sequences. We demonstrate the utility of our algorithm on real prostate, heart, and brain data and a synthetic brain phantom that allows us to quantitatively assess the performance of our algorithm.

Thesis Supervisor: William M. Wells III
Title: Research Scientist, MIT AI Lab
Associate Professor of Radiology,
Harvard Medical School and Brigham and Women's Hospital

Acknowledgements

There are many people without whom this thesis would not have been possible. I would like to thank my thesis supervisor Sandy Wells for his support and assistance, and for his ability to come up with crazy ideas that always seem to work out. His ability to immediately grasp the big picture never ceases to amaze me. I would also like to thank the many people in the Stochastic Systems Group: Alan Willsky for his incredible insight and knowledge of the field, for his boundless energy, and for always making sure we stay focused and grounded in reality; Dr.^{2} Andy "PK" Tsai for helping me get up to speed when I first arrived and for making me afraid to ever get treated in a hospital again; John Fisher for showing me that all of the world's problems can be solved with just two tools and four letters: EM and MI; Müjdat Çetin for always lending a hand no matter how busy he may be; Martin Wainwright for sharing his love of techno music; Andrew Kim for voting for me; Dewart Tucker for always being able to put Andy in his place; Ron Dror for always having lots to talk about in grouplet; Junmo Kim for the interesting discussions about Korean culture; Alex Ihler for helping out whenever I crash the lab computers; Erik Sudderth for showing that not finishing your Master's in 2 years isn't the end of the world; Jason Johnson for almost singlehandedly completing our 6.252 project; Dmitry Malioutov for always being able to sniff out free food; Lei Chen for the Chinese lessons (though I still think that he's missing a character from his name); Walter Sun for talking football and showing how small the world really is; and, of course, Taylore Kelly without whom the lab would self destruct (and who can always ruin my day no matter how good things are going).

I have also benefitted greatly from my interactions with other people here at MIT as well as researchers from Harvard Medical School. The Prostate Interest Group Seminar (PIGS) was important when we were in the brainstorming stages. It was in these discussions where I learned about the importance of MPUs. Many people at Brigham and Women's Hospital were important for contributing ideas and/or helping to acquire data. Included among these are Clare Tempany, Steve Haker (who followed me here from Minnesota), Bob Mulkern, Steven Thibodeau, and Walid Kyriakos.

I would also like to thank many people at the University of Minnesota who helped put me where I am today. Specifically I would like to thank Allen Tannenbaum and

Tryphon Georgiou. It was a privilege to work for Allen. He really inspired my interest in image processing and research in general, and it's difficult to imagine what I would be doing right now if he hadn't taken a poor confused freshman under his wing. And Tryphon is one of the kindest people I know, and his willingness to go easy on his overworked students was very much appreciated.

I thank the Department of Defense and the American Society for Engineering Education for their support through the National Defense Science and Engineering Graduate Fellowship.

I would like to thank my friends and extended family scattered throughout the world. Your friendship and support throughout the years has been greatly valued. I would like to thank my sister Eva for being the best big sister I've ever had, though she loses points for moving to the wrong coast. Finally, I would like to thank my parents, to whom this thesis is dedicated. Thank you for always being there for me.

Contents

List of Figures

List of Tables

Notational Conventions

Symbol	Definition
General Notation	
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the space of n-dimensional vectors
$\mathbb{Z}$	the set of integers
$\mathbb{Z}^+$	the set of counting numbers (positive integers)
$\mathcal{A} \setminus \mathcal{B}$	the set $\mathcal{A}$ minus the set $\mathcal{B}$
$\boldsymbol{x}[i]$	the ith component of the vector $\boldsymbol{x}$
$[\boldsymbol{A}]_{i,j}$	element in the ith row and jth column of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{\mathrm{T}}$	the transpose of matrix $\boldsymbol{A}$
$\lvert\cdot\rvert$	absolute value
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	the inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{W}}$	the inner product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ weighted by $\boldsymbol{W}$
$\lVert\cdot\rVert$	ℓ_2 norm, unless indicated otherwise
$\lVert\cdot\rVert_p$	ℓ_p norm
$\lVert\cdot\rVert_{\boldsymbol{W}}$	ℓ_2 norm weighted by the matrix $\boldsymbol{W}$
$*$	the convolution operator (discrete or continuous)
$\circ$	the Hadamard (entrywise) product operator between two vectors or two matrices
∇	the gradient operator
$\nabla\cdot$	the divergence operator
∇^2	the Laplacian operator
∇^4	the biharmonic operator
$\mathcal{H}$	the Hessian operator
$\dot{f}$	the derivative of f
$f'(\boldsymbol{x}; \boldsymbol{d})$	the derivative of f with respect to $\boldsymbol{x}$ in the direction $\boldsymbol{d}$
$\mathcal{O}(\cdot)$	computational complexity (number of operations) is asymptotically bounded by a constant times the argument

Symbol	Definition
$\mathrm{E}[\mathbf{f}]$	expected value of $\mathbf{f}$
$\mathrm{E}[\mathbf{f}\|\mathbf{g}]$	expected value of $\mathbf{f}$ given $\mathbf{g}$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	represents a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\boldsymbol{x} \sim \mathcal{P}$	the observation vector $\boldsymbol{x}$ is drawn from distribution $\mathcal{P}$
$\mathrm{p}_{\mathsf{x}}(\boldsymbol{x})$	probability for the random vector $\boldsymbol{x}$
$\mathrm{p}_{\mathsf{x}\|\boldsymbol{y}}(\boldsymbol{x}\|\boldsymbol{y})$	probability of $\boldsymbol{x}$ given $\boldsymbol{y}$
$\mathrm{p}_{\mathsf{x}}(\boldsymbol{x}; \boldsymbol{z})$	probability of $\boldsymbol{x}$ parameterized by $\boldsymbol{z}$
$\mathrm{p}_{\mathsf{x},\mathsf{y}}(\boldsymbol{x}, \boldsymbol{y})$	joint probability for the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$\hat{\boldsymbol{x}}$	estimate of $\boldsymbol{x}$
$\hat{\boldsymbol{x}}^{(k)}$	estimate of $\boldsymbol{x}$ from the kth iteration
$\hat{\boldsymbol{x}}^{[s]}$	estimate of $\boldsymbol{x}$ at scale s
$\arg\min f(\boldsymbol{x})$	the choice of $\boldsymbol{x}$ that minimizes $f(\boldsymbol{x})$

Chapter 1

Introduction

NON-INVASIVE imaging technology has changed the face of medicine in the 20th Century. Medical imaging began with Roentgen's discovery of the X-ray in 1895 and continued with various three-dimensional (3D) imaging technologies such as computed tomography (CT), magnetic resonance (MR), ultrasound (US), positron emission tomography (PET), single photon emission computed tomography (SPECT), and a whole host of other acronyms [18]. Imaging technology has progressed dramatically from simple X-rays which only provide a two-dimensional (2D) projection of tissue density. Modern imaging techniques allow full 3D reconstructions at high resolutions and can show information beyond just tissue boundaries [18].

X-ray is still the most widely used medical imaging technique. The equipment is ubiquitous and cheap and does a good job of showing many ailments. It is limited in application due to the fact that it is a 2D projection and can only show how well the tissue absorbs X-rays (it essentially shows tissue density). X-ray is commonly applied as an initial onsite precautionary test as well as a diagnostic for many problems including broken bones, oral cavities, respiratory problems, and breast cancer (mammography).

CT is simply X-ray technology extended to 3D. An X-ray image is the projection of the 3D density map onto the imaging plane. By acquiring multiple X-ray images in different planes and solving the inverse problem (called tomography), we can find the density map that caused our observed images. With modern implementations, CT is fast and has very good spatial resolution. It does an excellent job of displaying bone and contrasting among hard tissue, soft tissue, and air. CT is widely used for many applications such as cardiac analysis, virtual colonoscopy, and vascular analysis. It is also the technology being used in the bomb-scanning machines being installed in airports. CT also retains many of the disadvantages of X-ray including the usage of radiation and the inability to display significant differentiation among soft tissues.

Another technique that is nearly as ubiquitous as X-ray is ultrasound. Ultrasound uses sonar-based echo location techniques to construct images. Probably the most well-known use of ultrasound is in viewing the fetus in pregnant women. Other uses include checking for the presence of gall stones and investigating growths in the kidney or liver. Ultrasound is very good with soft tissue but is not as good with bone and air cavities. Because ultrasound is constructed using sound waves instead of photons, the spatial resolution tends to be quite limited, and overall image quality is much inferior to that obtained from MR and CT.

MR is different from most other medical imaging techniques because it does not use tomography to obtain a 3D image. MR has revolutionized our ability to diagnose ailments, especially those involving soft tissue where its ability to provide contrast is unparalleled. MR imaging is based on the phenomenon known as nuclear magnetic resonance (NMR), and MR images are aggregate measurements of tissue composition at the molecular level. The extent that this molecular structure stays constant within tissue and varies among different tissues determines the effectiveness of MR imaging. MR is by far the most flexible imaging technique because there are an enormous number of parameters that can be controlled during the imaging process. It can measure things as simple as proton density to things as complex as brain activation maps (functional MRI) and blood flow (angiography). The main difficulty with MR is cost. The machines cost millions of dollars, the operating expenses are high, and only trained radiologists can interpret the results. MR is widely used for cancer detection, ligament and joint evaluation, and various brain diagnoses.

■ 1.1 Medical image processing

Along with the advances in generating medical imagery, our ability to process the images has also increased dramatically. A lot of the tasks required of radiologists can be repetitive and boring (*e.g.*, looking for tumors on mammographies, delineating organ boundaries). The number of medical scans being taken has increased dramatically while the number of radiologists trained to interpret the results has not. In the image processing and artificial intelligence (AI) communities, there has been a great deal of active research to shift the burden of the more repetitive tasks from humans to computers.

The image processing techniques required for medical imagery can range from the

classical such as denoising and contrast enhancement to more modern techniques such as segmentation and registration. The former can be viewed as preprocessing for either radiologist evaluation or computer processing. The latter can be viewed as fundamental building blocks for more complex image analysis tasks such as automated computer diagnosis. Registration is the process of aligning two or more images. This is a task that would be nearly impossible without the assistance of computers (especially registrations which allow non-rigid deformation). Registration is useful in a wide variety of contexts including multiple-modality alignment (*e.g.*, MR and CT) and comparison of patients with anatomical atlases. Segmentation is the division of an image into regions. The criteria for what constitutes a coherent region can vary from application to application. Segmentation is critical for many medical image analysis tasks including heart efficiency calculations, schizophrenia diagnosis, cancer detection, and image guided surgery.

The original impetus for the work in this thesis was the work being done at Brigham and Women's Hospital (BWH) involving prostate cancer treatment [72, 88]. Prostate cancer is a very serious disease that affects an enormous segment of the population. Among American men, it is the most common cancer besides skin cancers, and one in six men will be diagnosed with prostate cancer during his lifetime [2]. It is also a very serious illness—prostate cancer is the 2nd leading cause of cancer death in men.

The most common method to treat this disease is called radical prostatectomy. This procedure basically surgically extracts the prostate and perhaps some surrounding tissue such as the lymph nodes. A thorough description of the exact surgical details can be found in [72]. This technique has been used for quite some time and has advanced to the point that nearly 100% of cancers that remain confined to the prostate can be cured. A number of side effects make radical prostatectomy less than ideal. Impotence results in approximately 25% to 30% of the patients. Around 10% of patients experience incontinence months after the surgery, and another 3% to 12% experience bladder neck contracture.

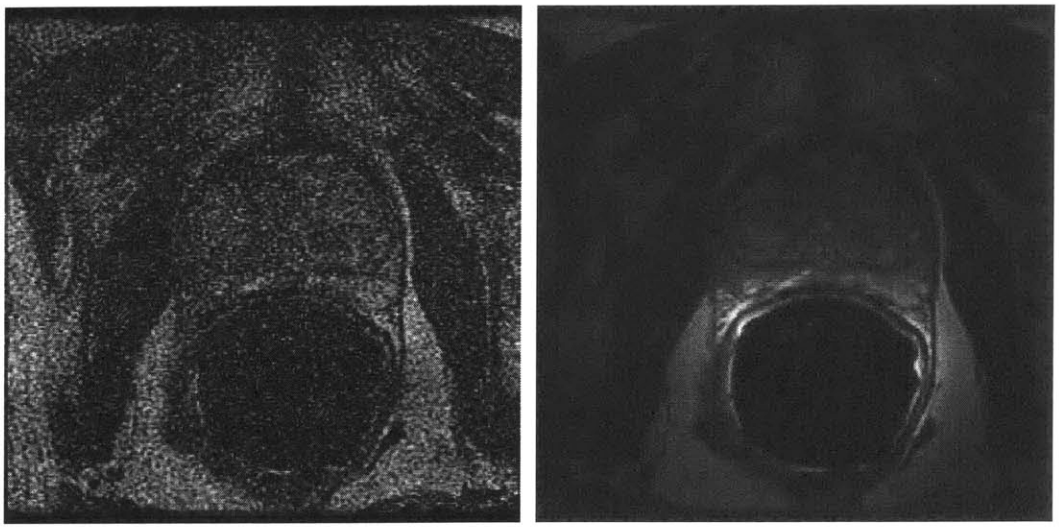
Several alternative treatments are being developed to address these concerns. Cryosurgery (also known as cryotherapy and cryoablation) treats localized cancers by freezing cells with a metal probe. Hormone therapy can slow the spread of cancer. Radiation therapy uses high-energy electromagnetic waves to kill cells. A newer type of radiation therapy known as brachytherapy involves surgically placing radioactive seeds at the cancerous locations. This is a minimally invasive procedure that allows clinicians to target the cancerous cells with high doses of radiation without severely impacting the

surrounding healthy tissue.

There is a wide variety of image processing problems that need to be tackled for the treatment of prostate cancer using brachytherapy. At the earliest stage, work is being done to investigate the usage of specialized MR scanning techniques such as diffusion tensor imaging for initial tumor detection and localization [15]. Diffusion tensor imaging gives a view of the diffusion of water at the cell level. One of the differentiating characteristics of tumors is their increased oxygen consumption which leads to an increased level of diffusion. MR can also be used in cancer staging. An imaging protocol called T_1-*weighted* is useful for determining the boundaries of the prostate. Expansion of the prostate is a common symptom of prostate cancer. Another imaging protocol called T_2-*weighted* allows us to see internal structure in the prostate as well as differentiate cancerous tissue from healthy tissue.

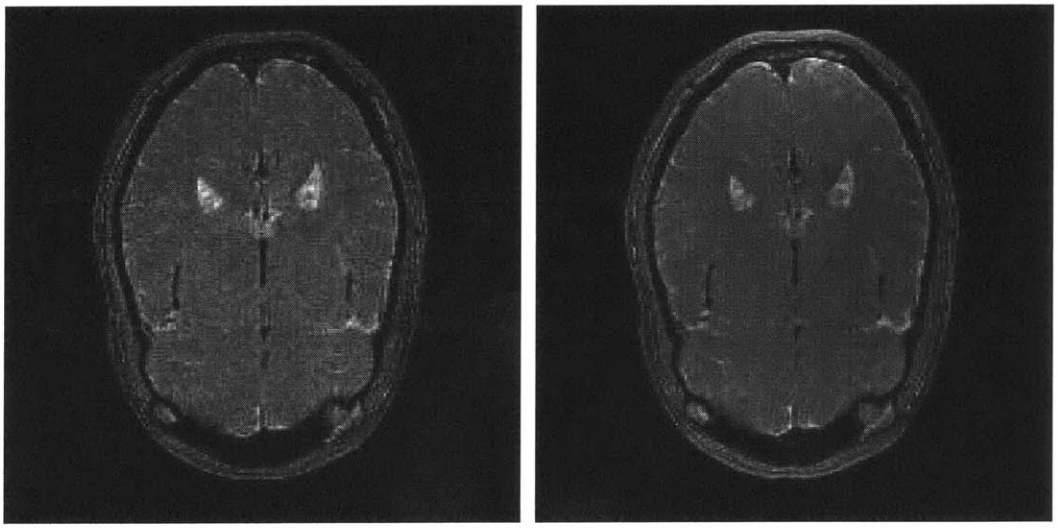
In brachytherapy the radioactive seeds must be precisely placed, so accurate knowledge about the shape and location of the patient's prostate is required. The MR images captured in conventional scanners can also be used for surgical planning, but real-time imaging information is necessary during the surgery to ensure full dosage of the cancerous regions. The open-configuration MR machine allows the surgeon access to the patient while still obtaining detailed imaging information of internal structures [88]. The surgical procedure requires both registration and segmentation steps. We need to be able to fuse the current location of the anatomy with the surgical plan. The conventional MR scans will be of higher quality than the real-time data, so we would also like to be able to incorporate information from the pre-operative scans into the intra-operative scans. Once the intra-operative data are acquired, segmentation is needed for precision needle targeting and to indicate to the surgeon areas to be avoided (such as the rectum wall and the seminal vesicles) [74,75]. Computation of the radiation dosage as a function of space is also needed. This is to ensure that the cancerous regions are sufficiently radiated. This calculation can be made easier by getting precise needle location data from the real-time images.

One thing that all of these image processing procedures do is that they operate on the intensity values obtained in MR images. This behavior is not restricted to prostate image processing. Image processing in general relies on the intensity values and can be significantly impaired by imperfections in the image collection process. MR images are constructed from electromagnetic responses and are captured using coils of wire. These intensities are corrupted both by random noise as well as systematic electromagnetic



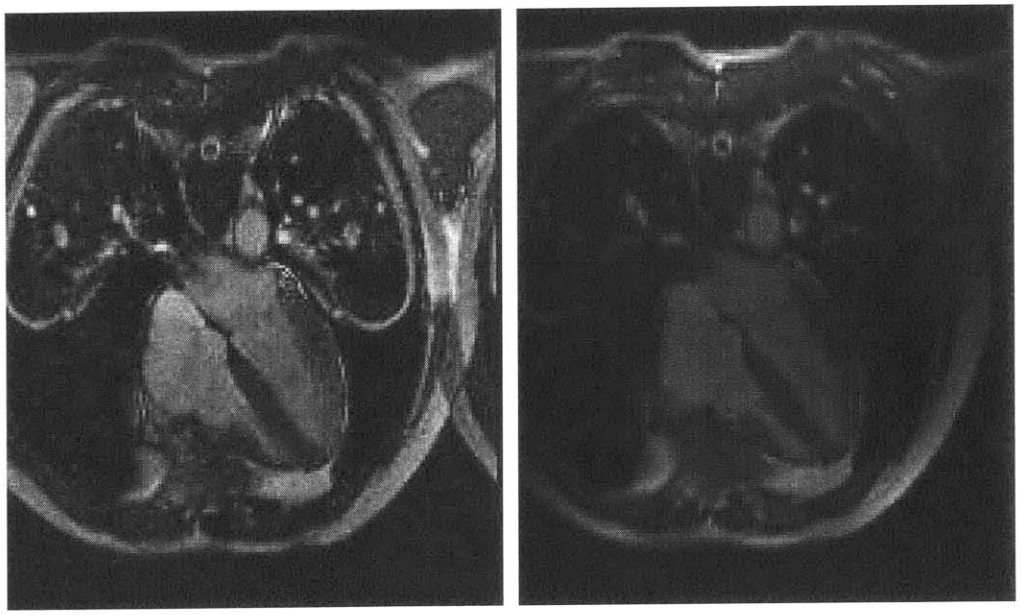
(a) (b)

Figure 1.1. Axial mid-gland T_2-weighted prostate images acquired using (a) a body coil and (b) an endorectal coil and pelvic phased-array coil combination. FOV is 12 cm x 12 cm and slice thickness is 3 mm.



(a) (b)

Figure 1.2. Gradient echo axial brain images acquired using (a) a body coil and (b) a four-coil phased-array.



(a) (b)

Figure 1.3. Fast-spin echo heart images acquired using (a) a body coil and (b) a four-coil phased-array. FOV is 32 cm x 32 cm and slice thickness is 8mm.

effects. The latter are collectively known as bias fields or intensity inhomogeneities. The bias in this case is a multiplicative bias rather than an additive bias that is more common. We use the term bias because the intensity inhomogeneity is a systematic effect and not a random effect. Both the noise and the bias can confuse automated image processing algorithms, and we would like to minimize both as much as possible. The bias fields can be minimized by using a coil such as a body coil that has a very homogeneous spatial sensitivity profile. This spatial homogeneity comes at the expense of signal response which leads to a decrease in the signal-to-noise ratio (SNR). In most MR imaging applications, we are only interested in a small region of the body. We can exploit this fact by using surface coils which are placed close to the region of interest (ROI) [4, 23]. These coils have high signal response close to the coil and the response diminishes with distance. This variable signal response allows much better visualization of the object of interest but is also the main cause of the bias field.

The endorectal coil is a surface coil used in imaging the prostate and probably has the most pronounced bias field among widely used surface coils. The reason for this is the extremely small size of the ROI which requires significantly higher signal response to maintain adequate image quality. Figure 1.1(a) depicts a prostate image captured using the homogeneous body coil, and Figure 1.1(b) shows a prostate image of the exact same location captured with an endorectal coil and pelvic phased-array coil combination. The surface coil image may not appear to be more usable than the body coil image, but radiologists can use simple correction techniques such as window/level to make the image more palatable locally. In Figure 1.2 and Figure 1.3, we show two more imaging applications where body coil and surface coil images are acquired. The body coil images for both the brain and the heart display much better SNR than for the prostate. This is because the objects being imaged are much larger and can use spatial averaging to decrease the noise. The lower noise means that the maximum coil response of the surface coils does not need to be extremely large, and the larger region to be imaged means that the response of the coil cannot diminish too rapidly. This means that the bias field in these images is not very pronounced. Even though both the brain and heart surface coil images do not display severe intensity inhomogeneities, we would still like to have the better SNR of the surface coil images and the improved homogeneity of the body coil images.

The bias correction problem is currently an open one and is very widely studied. One reason for the number of different approaches is that many groups begin working on an

application involving MR images and quickly find that much of their progress is stymied by the corrupting effects of the bias field. The importance of this problem will increase as MR magnets increase in strength and electromagnetic effects become more and more pronounced. There currently are not any wholly satisfying bias correction methods in the literature. We introduce a very general framework for bias correction that can provide superior results for a wide variety of applications with minor adjustments. Even better results can be obtained by adjusting the implementation to reflect the special properties of the imaging application being considered.

1.2 Main contributions

The main contribution of this thesis is a fully-automatic non-parametric approach to MR bias correction that provides qualitatively and quantitatively appealing results. In addition, we present a unified approach that simultaneously debiases and denoises the image. All existing bias correction techniques have trade-offs that affect their utility or their usability, and our method is no exception. Some methods are fast but provide limited bias correction capabilities. Others provide excellent results but require significant operator training and supervision to provide optimal results. Our method imposes additional requirements at the image acquisition step: we require a body coil image to be captured in addition to the surface coil image which we wish to correct. The need to separately acquire the body coil image is probably unavoidable due to coil coupling issues [62]. Otherwise, once parameters are chosen for a particular scanning protocol, our method requires no user input. We produce our corrected images by iteratively solving a variational problem which we have found to have nice convergence properties and to be resistant to poor initialization. In fact, initialization with random noise usually converges to the correct result. In addition, with multiresolution approaches, our algorithm is fast and computationally stable. We feel that the quality of our results makes this additional scanning requirement worthwhile, especially for certain applications such as the prostate which have severe intensity inhomogeneities.

We derive our algorithm using a statistical approach with some simplifying assumptions. This leads to an energy functional which we seek to minimize. At a simplest level, this provides us with intuition about what the resultant bias correction should do. We can then use this energy functional to optimally select parameters for other traditional bias correction techniques such as linear filters [12, 33]. Directly minimizing

the functional leads to an intuitive method for combining our multiple observation images once we have reliable bias field estimates (regardless of how those estimates were generated). The resultant image has noise characteristics superior than those available from either observed image.

We can also take a non-parametric approach and solve the variational problem of fully minimizing the energy functional. The bias field and the true MR image combine in a multiplicative manner. Many techniques available in the literature make this an additive relationship by taking the log and neglecting the noise term. We solve the problem directly in the original multiplicative form. This results in a cleaner formalism but imposes the need to do nonlinear estimation. We can perform this estimation using a fixed-point iterative method which leads to a set of simpler estimation problems. These smaller estimation problems can be viewed as a partial differential equation (PDE) in the continuous domain or a large linear system in the discrete domain. This results in bias and image estimates that are optimal from a mean squared error standpoint. By choosing appropriate priors, our algorithm can impose constraints on our estimated bias field and intrinsic image that make our estimates conform to our physical model. Specifically, we can make our bias field smooth and our intrinsic image piecewise constant. The former is necessary to make our algorithm give meaningful results. The latter is not necessary to generate adequate results, but it is similar to putting an anisotropic edge-preserving filter into our method and consequently produces results with less noise. By fully integrating the denoising operation into our algorithm, the resulting image and bias field estimates are better than what could be obtained by performing both operations sequentially because knowledge of one estimate improves our ability to obtain the other estimate.

■ 1.3 Outline

Chapter 2 contains background information on a variety of subjects needed to understand this work. We begin with mathematical and statistical preliminaries such as stochastic estimation, regularization of ill-posed problems, the calculus of variations, and nonlinear optimization. Knowledge of calculus and linear algebra is assumed, though the main features of the latter that we will use in this thesis are included in Appendix A. We then discuss the physics involved in MR image formation. This knowledge gives us insight into the cause of the bias field which allows us to accu-

rately model the problem. We also survey some of the copious number of existing bias correction techniques published in the literature.

Chapter 3 introduces our imaging model. We discuss the simplifications we make in order to make our problem formulation tractable. We construct a variational problem which we solve to provide estimates of the bias field and the corrected image. We discuss the statistical roots of the energy functional we construct. Analytical solutions are not possible, so we present iterative techniques to quickly obtain approximate results. A number of techniques are used to rapidly increase the convergence speed of our solvers. Generalizations to our model are presented to allow us to solve problems using general ℓ_p norms, operate on 3D volumes, process multiple-modality, multiple-coil scans, and apply multiresolution techniques to increase convergence speed.

We then present our results in Chapter 4. We begin by demonstrating our bias correction techniques on a variety of MR imaging applications: the prostate/rectum, the heart, and the brain. We demonstrate the robustness of the technique to very different imaging modalities and surface coil configurations. We show the result of using different parameter choices and different initial estimates as well as a unified approach that simultaneously performs bias correction and denoising. We then apply our correction to phantom brain images acquired from the Montreal Neurological Institute [19, 40]. These brain images are constructed using a very precise physical model of the brain. This allows us to test our method with very different surface coil profiles, and it gives us ground truth to compare our reconstruction results to.

Chapter 5 summarizes the thesis and presents some future research directions.

■ 1.4 Notation

Most of the notation that we will use in this thesis is fairly standard for this field. We italicize mathematical variables in a serif font, *e.g.*, x. When it is important to differentiate between a random variable and an observation, we will denote the random variable using an upright sans serif font, *e.g.*, x. Vectors are lower case and bold, *e.g.*, $\boldsymbol{x}$ while random vectors are lower case, bold, sans serif, and upright, *e.g.*, $\boldsymbol{\mathsf{x}}$. Matrices are upper case and bold, *e.g.*, $\boldsymbol{A}$. Any deviation from this convention is explicitly noted in the text.

We use a few mathematical operations that may perhaps be a bit obscure. The Hadamard product [29] (or Schür product or entrywise product) is an operator for

two vectors or two matrices of identical size. The elements of the output are simply the elements of the input multiplied entrywise. For example, let $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c} \in \mathbb{R}^n$. Then $\boldsymbol{a} = \boldsymbol{b} \circ \boldsymbol{c}$ results in

$$\boldsymbol{a}[k] = \boldsymbol{b}[k]\boldsymbol{c}[k] \ \forall \ k \in \{1, 2, \ldots, n\} \ . \tag{1.1}$$

We also use generalized ℓ_p norms quite a bit in this work. We define a ℓ_p norm as

$$\|\boldsymbol{x}\|_p = \left(\sum_i |\boldsymbol{x}[i]|^p \right)^{1/p} \ . \tag{1.2}$$

We describe a few special cases. The ℓ_0 norm counts the number of non-zero elements in $\boldsymbol{x}$. The ℓ_1 norm is simply the sum of the absolute value of the elements. The standard Euclidean norm is the ℓ_2 norm. The ℓ_∞ norm selects the absolute value of the element of $\boldsymbol{x}$ farthest from zero.

Chapter 2

Background Information

THIS chapter covers background material that is essential to understand the main contributions of this thesis. We begin with mathematical preliminaries: statistical estimation, regularization, variational methods, and nonlinear optimization techniques. The first two we use to develop our problem formulation, and the last two we use to solve the problem. The reader is assumed to be familiar with calculus and linear algebra. A brief synopsis of the main results of linear algebra that we use appears in Appendix A. The second section talks about the MR image formation process. We will later use these concepts in creating our image formation models. Finally we review previous work in MR bias correction. We will discuss the strengths and limitations of various approaches.

2.1 Mathematical Preliminaries

2.1.1 Statistical Estimation

The basic setup for most statistical problems (estimation or detection) is that there is some underlying process (random or non-random) $\mathbf{x} \in \mathbb{R}^n$ which we cannot directly observe. What we do observe is $\mathbf{y} \in \mathbb{R}^m$ which is related to $\mathbf{x}$ through a function $h : \mathbb{R}^n \to \mathbb{R}^m$ and a noise process $\mathbf{n}$:

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{n} \ . \tag{2.1}$$

If $h(\cdot)$ is a linear function (*i.e.*, $\mathbf{y} = \boldsymbol{H}\mathbf{x} + \mathbf{n}$, $\boldsymbol{H}$ is a $m \times n$ matrix), then this is termed a linear estimation problem.

We typically characterize $\mathbf{y}$ (and $\mathbf{x}$ if it is random) through their probability density functions (PDFs). See [57] for a comprehensive treatment on probability and random variables. We begin with examining a scalar x. PDFs must integrate to one and always

be non-negative. It is a probability density (and not a probability function) in the following sense:

$$\mathrm{P}[\alpha < \mathsf{x} < \beta] = \int_{\alpha}^{\beta} \mathrm{p}_{\mathsf{x}}(x)\mathrm{d}x \ . \tag{2.2}$$

The value of $\mathrm{p}_{\mathsf{x}}(x)$ for any particular x can be much greater than one.

We can define the expected value of a function over x in the following manner:

$$\mathrm{E}_{\mathrm{p}_{\mathsf{x}}(x)}\left[f(\mathsf{x})\right] \equiv \int f(x)\mathrm{p}_{\mathsf{x}}(x)\mathrm{d}x \ . \tag{2.3}$$

We often drop the subscript on the expected value if it is clear what density we are using. Essentially this is a weighted average of $f(\mathsf{x})$ with more weight given to higher probability outcomes.

We can then define some common statistics that are expected values of various functions of x. The moments of x are defined as

$$\mu_m \equiv \mathrm{E}\left[\mathsf{x}^m\right] = \int x^m \mathrm{p}_{\mathsf{x}}(x)\mathrm{d}x \ . \tag{2.4}$$

The first moment is simply the expected value of x and is referred to as the mean. This is not necessarily the most common value of x. It is simply the average value we would observe of x over a large number of trials.

The variance σ^2 is another commonly-used statistic. It is a second order statistic because it depends on the second moment:

$$\sigma^2 \equiv \mathrm{E}\left[(\mathsf{x} - \mu_{\mathsf{x}})^2\right] = \mathrm{E}\left[\mathsf{x}^2\right] - \mu_{\mathsf{x}}^2 \ . \tag{2.5}$$

The variance tells us how far away we can expect to find x from its mean. The square root of the variance is often referred to as the standard deviation.

We can also define probabilities over pairs of random variables. The function $\mathrm{p}_{\mathsf{x},\mathsf{y}}(x,y)$ is known as the joint PDF of x and y and is a density in the sense of (2.2). From $\mathrm{p}_{\mathsf{x},\mathsf{y}}(x,y)$ we can obtain the density of just x (which is termed the marginal density):

$$\mathrm{p}_{\mathsf{x}}(x) = \int \mathrm{p}_{\mathsf{x},\mathsf{y}}(x,y)\mathrm{d}y \ . \tag{2.6}$$

A similar expression holds for $\mathrm{p}_{\mathsf{y}}(y)$. The random variables x and y are called statistically independent if the value of one variable has no effect on the value of the other. This is true if and only if

$$\mathrm{p}_{\mathsf{x},\mathsf{y}}(x,y) = \mathrm{p}_{\mathsf{x}}(x)\mathrm{p}_{\mathsf{y}}(y) \ . \tag{2.7}$$

The covariance of two random variables is a statistic that measures how they vary with each other:

$$\lambda_{\mathsf{xy}} = \mathrm{E}\left[(\mathsf{x} - \mu_{\mathsf{x}})(\mathsf{y} - \mu_{\mathsf{y}})\right] = \mathrm{E}\left[\mathsf{xy}\right] - \mu_{\mathsf{x}}\mu_{\mathsf{y}} \ . \tag{2.8}$$

If x and y are independent, then $\mathrm{E}\left[\mathsf{xy}\right] = \mathrm{E}\left[\mathsf{x}\right]\mathrm{E}\left[\mathsf{y}\right]$ and $\lambda_{\mathsf{xy}} = 0$. If $\lambda_{\mathsf{xy}} = 0$, x and y are said to be uncorrelated. Note that independent random variables are always uncorrelated, but the converse is not always true.

We can easily replace a scalar x with a vector $\mathbf{x}$ in the preceding discussion. The PDF $\mathrm{p}_{\mathbf{x}}(\boldsymbol{x})$ must be defined in a similar manner to the multivariate PDF described above. When we compute the moments, we must compute them between every element combination in order to get all of the mth-order statistics. We can store the first-order statistics in a vector and the second order statistics in a matrix. A general m^{th}-order moment requires a m-dimensional matrix. Hence third and higher moments become quite unwieldy, and we do not generally use them. The first moment is largely unchanged except we take the expectation elementwise:

$$\boldsymbol{\mu}_{\mathbf{x}} = \mathrm{E}\left[\mathbf{x}\right] \ . \tag{2.9}$$

The variance becomes a vector outer product (and is now known as a covariance matrix):

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \mathrm{E}\left[(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})^{\mathrm{T}}\right] = \mathrm{E}\left[\mathbf{x}\mathbf{x}^{\mathrm{T}}\right] - \boldsymbol{\mu}_{\mathbf{x}}\boldsymbol{\mu}_{\mathbf{x}}^{\mathrm{T}} \ . \tag{2.10}$$

We define a conditional probability as the probability of observing $\mathbf{y}$ if we know that $\mathbf{x}$ occurred:

$$\mathrm{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{\mathrm{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y})}{\mathrm{p}_{\mathbf{x}}(\boldsymbol{x})} \ . \tag{2.11}$$

Note that if $\mathbf{x}$ and $\mathbf{y}$ are independent, $\mathrm{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{p}_{\mathbf{y}}(\boldsymbol{y})$. In the context of the model introduced in (2.1), $\mathrm{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x})$ is the known as the measurement model. It relates the observed values to the underlying process. We refer to $\mathrm{p}_{\mathbf{x}}(\boldsymbol{x})$ as the *a priori* or prior probability density and $\mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})$ as the *a posteriori* or posterior probability density. The former is what we know about $\mathbf{x}$ before the experiment is performed, and the latter is what we know about $\mathbf{x}$ after the experiment is performed (*i.e.*, after we observe $\mathbf{y}$).

We can compute the conditional expectation of $\mathbf{x}$ given $\mathbf{y}$:

$$\mathrm{E}_{\mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})}\left[\mathbf{x}|\mathbf{y}\right] = \int \mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})\boldsymbol{x}\mathrm{d}\boldsymbol{x} \ . \tag{2.12}$$

Note that previous expected values that we defined yielded numerical results while this conditional expectation yields a random variable (because $\mathbf{y}$ is a random variable).

We see that (2.11) can be easily rearranged as

$$\mathrm{p}_{\mathsf{x},\mathsf{y}}(\boldsymbol{x},\boldsymbol{y}) = \mathrm{p}_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x})\mathrm{p}_{\mathsf{x}}(\boldsymbol{x}) = \mathrm{p}_{\mathsf{x}|\mathsf{y}}(\boldsymbol{x}|\boldsymbol{y})\mathrm{p}_{\mathsf{y}}(\boldsymbol{y}) \ . \tag{2.13}$$

This observation leads us to Bayes' Rule which relates the two conditional probabilities:

$$\mathrm{p}_{\mathsf{x}|\mathsf{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\mathrm{p}_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x})\mathrm{p}_{\mathsf{x}}(\boldsymbol{x})}{\mathrm{p}_{\mathsf{y}}(\boldsymbol{y})} \ . \tag{2.14}$$

The significance of Bayes' Rule is that it allows us to relate the measurement model and our *a priori* density to the *a posteriori* density.

There are two main ways to approach statistical estimation: Bayesian and non-random [87]. Bayesian estimation imposes a prior on x and non-random does not. If we only have the measurement model available to us, x may be random or non-random, but it does not really matter because we have no distribution for x. Hence we can simply treat it as non-random. Given an observation $\boldsymbol{y}$, it makes sense to estimate $\boldsymbol{x}$ in the following manner:

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \mathrm{p}_{\mathsf{y}}(\boldsymbol{y};\boldsymbol{x}) \ . \tag{2.15}$$

The function $\mathrm{p}_{\mathsf{y}}(\boldsymbol{y};x)$ is also known as the likelihood function of the data. We say that the PDF for y is parameterized by $\boldsymbol{x}$. It does not give a probability of $\boldsymbol{x}$, but it gives some sense of how likely a certain $\boldsymbol{x}$ occurred given that we observed $\boldsymbol{y}$. This method is usually called maximum likelihood (ML) estimation. This method is also applicable in situations where x is random, but we do not know the prior probability:

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \mathrm{p}_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x}) \ . \tag{2.16}$$

Note that when we maximize the likelihood, we are not choosing the $\hat{\boldsymbol{x}}$ that has the highest probability of occurring given our observations. We are choosing the $\hat{\boldsymbol{x}}$ that maximizes the probability of us observing the data. This is a subtle distinction that will be made clearer later in this section.

The log function is strictly monotonic, so finding the maximum of the log probability is equivalent to finding the maximum of the probability. Hence we can rewrite (2.15) as

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \log \mathrm{p}_{\mathsf{y}}(\boldsymbol{y};\boldsymbol{x}) \ . \tag{2.17}$$

A large number of problems are easier to handle using log probabilities.

ML estimation is popular for a number of reasons. Statistical estimators are characterized by two main properties: the bias and the variance. The bias is the expected error, and the variance is the variance of the error estimate. The Cramer-Rao bound [87] gives a lower bound on the variance of any unbiased estimator. If an estimator satisfies the Cramer-Rao bound with equality, this estimator is said to be efficient. If an efficient estimator can be found, then that estimator must be the ML estimator. The converse is not true in general. Additionally, ML estimators tend to be very nice to use in practice. Even if closed-form expressions cannot be found, a wide variety of optimization techniques can be used to compute ML estimates.

In Bayesian estimation, the goal is usually to choose an estimate that minimizes the expected value of some cost function. A common choice of cost function would be the ℓ_p error (with $p = 0, 1, 2$ being the most prevalent choices):

$$\mathcal{E} = \mathrm{E}\left[\|\mathbf{x} - \boldsymbol{x}\|_p^p\right] \ . \tag{2.18}$$

Using a ℓ_1 norm is often referred to as minimum absolute error (MAE) estimation, and it can be easily shown that the proper estimate is the median of the posterior density $\mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})$. When we use a ℓ_2 norm, this becomes a least-squares optimization problem. If the full joint probability is known, we can do Bayes least-squares (BLS) estimation:

$$\hat{\boldsymbol{x}}_{\mathrm{BLS}}(\boldsymbol{y}) = \mathrm{E}\left[\mathbf{x}|\boldsymbol{y}\right] \ . \tag{2.19}$$

This, of course, is the mean of the posterior density.

We term an estimate as being linear when it is composed solely of a linear combination of the observations along with a constant offset[1]:

$$\hat{\boldsymbol{x}}(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{b} \ . \tag{2.20}$$

We can find the linear estimator that minimizes the expected ℓ_2 error. This is called linear least-squares (LLS) estimation:

$$\hat{\boldsymbol{x}}_{\mathrm{LLS}}(\boldsymbol{y}) = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\Lambda}_{\mathbf{xy}}\boldsymbol{\Lambda}_{\mathbf{y}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_{\mathbf{y}}) \ . \tag{2.21}$$

$\boldsymbol{\Lambda}_{\mathbf{xy}}$ is the covariance matrix of $\mathbf{x}$ and $\mathbf{y}$, and $\boldsymbol{\Lambda}_{\mathbf{y}}$ is the covariance matrix of $\mathbf{y}$. A common framework to develop this equation involves the use of abstract vector spaces

[1]This should be more properly be known as affine, but the usage of the term linear is fairly entrenched.

and projections [37]. LLS estimation is much simpler than BLS estimation because it only requires knowledge of the first and second order statistics.

When we use a ℓ_0 norm as a cost function, we end up with what is known as maximum *a posteriori* (MAP) estimation:

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \mathrm{p}_{\mathsf{x}|\mathsf{y}}(\boldsymbol{x}|\boldsymbol{y}) \ . \tag{2.22}$$

This is the mode of the posterior distribution. This is closely related to ML estimation. We can use the fact that log is monotonic along with Bayes' rule to obtain the following:

$$\begin{aligned} \hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) &= \arg\max_{\boldsymbol{x}} \log \mathrm{p}_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x}) + \log \mathrm{p}_{\mathsf{x}}(\boldsymbol{x}) - \log \mathrm{p}_{\mathsf{y}}(\boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{x}} \log \mathrm{p}_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x}) + \log \mathrm{p}_{\mathsf{x}}(\boldsymbol{x}) \ . \end{aligned} \tag{2.23}$$

Note that we can drop the term involving the probability of y because it does not vary with $\boldsymbol{x}$ at all. To perform the MAP estimate, we need the full distributions of both the measurement model and the prior probability. We can see that if the prior distribution of x is uniform, the MAP estimate reduces to the ML estimate. Otherwise the MAP estimate finds the $\hat{\boldsymbol{x}}$ that has the highest probability of occurring given the data. MAP is popular for reasons similar to that of ML: computational tractability. BLS estimation is difficult to approach as a variational problem, and LLS estimation is limited to only producing linear estimates.

There are a number of interesting things that happen when we examine the case where x and y are jointly Gaussian[2]. In the non-random case, the estimation becomes quite simple. Let $\mathsf{n} \sim \mathcal{N}(0, \boldsymbol{\Lambda})$. Then $\mathsf{y} \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{\Lambda})$. We write the log likelihood as

$$\log \mathrm{p}_{\mathsf{y}}(\boldsymbol{y}; \boldsymbol{x}) = -\log\left((2\pi)^{m/2}|\boldsymbol{\Lambda}|^{1/2}\right) - \tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) \ . \tag{2.24}$$

Differentiating with respect to $\boldsymbol{x}$ yields

$$\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}_{\mathsf{y}}(\boldsymbol{y}; \boldsymbol{x}) = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} - \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}\boldsymbol{x} = 0 \ . \tag{2.25}$$

Thus the ML estimate is[3]

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{y}) = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \ . \tag{2.26}$$

[2]Usually two random variables are jointly Gaussian only when each component is Gaussian, and the two random vectors are related through a linear observation model with additive Gaussian noise.

[3]The maximum is unique because $\boldsymbol{\Lambda}$ is positive definite and hence the likelihood function is strictly concave.

This is the projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{H}$ with the distance defined as a ℓ_2 norm weighted by $\boldsymbol{\Lambda}^{-1}$.

Now we can assign a prior probability distribution to $\mathbf{x}$. Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Lambda}_{\mathbf{x}})$ and $\mathbf{n} \sim \mathcal{N}(0, \boldsymbol{\Lambda}_{\mathbf{n}})$. This leads to $\mathbf{y} \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Lambda}_{\mathbf{y}})$ where $\boldsymbol{\Lambda}_{\mathbf{y}} = \boldsymbol{H}\boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\Lambda}_{\mathbf{n}}$ if $\mathbf{x}$ and $\mathbf{n}$ are independent. This leads to the following criterion for the MAP estimate:

$$\frac{\partial}{\partial \boldsymbol{x}} \log \mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}_{\mathbf{n}}^{-1}\boldsymbol{y} - \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}_{\mathbf{n}}^{-1}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{\Lambda}_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}} - \boldsymbol{\Lambda}_{\mathbf{x}}^{-1}\boldsymbol{x} = 0 \ . \tag{2.27}$$

This results in the following estimation equation:

$$\begin{aligned} \hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) &= (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}_{\mathbf{n}}^{-1}\boldsymbol{H} + \boldsymbol{\Lambda}_{\mathbf{x}}^{-1})^{-1}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\Lambda}_{\mathbf{n}}^{-1}\boldsymbol{y} + \boldsymbol{\Lambda}_{\mathbf{x}}^{-1}\boldsymbol{\mu}_{\mathbf{x}}) & (2.28) \\ &= \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{\Lambda}_{\mathbf{n}})^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}_{\mathbf{x}}) & (2.29) \end{aligned}$$

where the transition from the first line to the second line is accomplished using the matrix inversion lemma [34]. In the jointly Gaussian case, $\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \hat{\boldsymbol{x}}_{\mathrm{BLS}} = \hat{\boldsymbol{x}}_{\mathrm{LLS}}$. The first equality holds for any unimodal symmetric posterior distribution. The latter equality is only true in the jointly Gaussian case.

■ 2.1.2 Regularization

Many optimization problems are ill-posed. This can mean that the solution does not exist, is not meaningful, or is unstable. One example of this is deblurring. For any given blurred image, there are many corrected images that can produce the input image. Regularization is a technique that can make the problem well-posed at the expense of biasing the solution[4].

There are many different approaches to regularization. We will focus on the so-called Tikhonov framework [38, 73]. With Tikhonov regularization, a term is added to the energy functional to enforce the effect that we want. These are generally referred to as penalty functions because they enforce the constraint in a soft manner. If the data indicate something strongly enough, the penalty can be overcome. Energy functionals in this framework break down into two parts: a data fidelity term and a regularization term. The data fidelity term ensures that our estimate is consistent with our observations while the regularization term ensures that our estimate is consistent with our prior knowledge about the signal. For a continuous formulation, this can be characterized as

$$E(f) = \mathcal{D}_y(f) + \alpha\mathcal{R}(f) \tag{2.30}$$

[4]If the regularizing term ends up exactly describing the solution, then no bias will be introduced. The chances of this happening, especially for real data, are very slim.

where $\mathcal{D}_y$ is the data fidelity term (for observation y), $\mathcal{R}$ is the regularization term, and α is a parameter that controls the tradeoff between our data and our prior.

The regularization term can be any continuous function that operates on functions. As we will detail in Section 3.3.3, when $\mathcal{R}$ involves derivative operations, it has some attractive solution properties. Choosing $\mathcal{R}$ to incorporate derivatives means that derivative energy is penalized. This can be interpreted as penalizing higher frequency components if we examine the Taylor series expansion of f. Common choices for the regularization are

$$\mathcal{R}(f) = \int \|\nabla f(\boldsymbol{x})\|^2 \mathrm{d}\boldsymbol{x} \tag{2.31}$$

$$\mathcal{R}(f) = \int |\nabla^2 f(\boldsymbol{x})|^2 \mathrm{d}\boldsymbol{x} \tag{2.32}$$

For a discrete formulation, the complete energy functional becomes:

$$E(\boldsymbol{f}) = \mathcal{D}_{\boldsymbol{y}}(\boldsymbol{f}) + \alpha\mathcal{R}(\boldsymbol{f}) \ . \tag{2.33}$$

A common choice for $\mathcal{R}$ is the ℓ_p norm of a linear operator applied to $\boldsymbol{f}$:

$$\mathcal{R}(\boldsymbol{f}) = \|\boldsymbol{L}\boldsymbol{f}\|_p^p \ . \tag{2.34}$$

The linear operator can be a series of linear operators (*e.g.*, a discrete Fourier transform and a frequency-selective filter). Using a ℓ_2 norm is convenient because then the regularizer becomes a linear term when we compute the gradient of the energy functional. Generally we want $\boldsymbol{L}$ to be high-pass in nature to ensure that high frequencies are penalized and low frequencies are not.

Rudin *et al.* [63] were the main drivers for the usage of total variation (TV) regularization. In continuous form, the TV norm is

$$\mathcal{R}(f) = \int \|\nabla f(\boldsymbol{x})\|_1^1 \mathrm{d}\boldsymbol{x} \ . \tag{2.35}$$

In discrete form, TV is simply (2.34) with $p = 1$ and $\boldsymbol{L}$ representing a gradient operator.

One reason for using an $\mathcal{L}_1$ penalty as opposed to an $\mathcal{L}_2$ penalty in the continuous domain is that step edges are permitted under the former but not the latter. We will illustrate this with a 1D example. Let our model and data be such that without regularization, our signal estimate would be the unit step function $u(t)$. We define our

$\mathcal{L}_1$ and $\mathcal{L}_2$ penalties using a first-order differential operator $\frac{\mathrm{d}}{\mathrm{d}t}$:

$$\mathcal{R}_1(f) = \int |\tfrac{\mathrm{d}}{\mathrm{d}t} f(t)| \mathrm{d}t \tag{2.36a}$$

$$\mathcal{R}_2(f) = \int |\tfrac{\mathrm{d}}{\mathrm{d}t} f(t)|^2 \mathrm{d}t \ . \tag{2.36b}$$

Then $\mathcal{R}_1(f) = \int \delta(t)\mathrm{d}t = 1$ while $\mathcal{R}_2(f) = \int \delta^2(t)\mathrm{d}t$ is unbounded. Hence for any choice of $\alpha > 0$, a step edge is impossible using a $\mathcal{L}_2$ penalty while it is possible using a $\mathcal{L}_1$ penalty. This idea does not carry over perfectly into the discrete domain (with discrete functions, all changes in value cause step discontinuities), but the fact that TV is more permissive of edges continues to be true.

TV gets its name because it seeks to minimize the sum of the absolute variation over the image. It does not try to remove large gradient values and is willing to accept a large gradient at one place if it makes sense according to the data. For instance, let the true signal be

$$f(t) = \begin{cases} 0 & : \quad t \leq 2.5 \\ 1 & : \quad t > 2.5 \end{cases} \ . \tag{2.37}$$

We observe a function

$$g(t) = h(t) * f(t) + n(t) \tag{2.38}$$

where h is a Gaussian filter kernel and n is white Gaussian noise. We observe g at six times: $\{0, 1, 2, 3, 4, 5\}$. We observe a sample path of (2.38) as

$$\boldsymbol{g} = [0.01, 0.05, 0.18, 0.76, 1.02, 0.97]^{\mathrm{T}} \tag{2.39}$$

We examine two guesses for the true signal:

$$\boldsymbol{f}_1 = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]^{\mathrm{T}} \tag{2.40a}$$

$$\boldsymbol{f}_2 = [0, 0, 0, 1, 1, 1]^{\mathrm{T}} \ . \tag{2.40b}$$

Both have the same TV value but very different ℓ_2 values (0.2 vs 1). So if we were to attempt to denoise $\boldsymbol{f}$ using a ℓ_2 data fidelity term, we would prefer $\boldsymbol{f}_1$ if we used the ℓ_2 penalty while we would prefer $\boldsymbol{f}_2$ if we used a ℓ_1 penalty (with appropriately balanced regularization parameter). Thus we see that using a ℓ_2 penalty punishes large gradient values with extreme prejudice. Unfortunately, large gradient values are desirable if we are trying to reconstruct data with edge features.

2.1.3 Variational Techniques

Variational techniques are methods to find stationary points of energy functionals where we are optimizing over sets of functions. They are commonly used in image processing by representing the true image as a continuous-domain function. The calculus of variations is the counterpart to ordinary calculus, and it describes the conditions required for stationary points. We will only discuss the machinery that we will use in deriving our algorithms. For a more comprehensive treatment, see [69] or [83].

We can define a functional E as

$$E(u) = \int J(\boldsymbol{x}, u, \dot{u}) \mathrm{d}\boldsymbol{x} \ . \tag{2.41}$$

The variational problem is to find u^* such that all other choices of u result in a higher energy:

$$u^* = \arg\min_u E(u) \ . \tag{2.42}$$

Much as ordinary vector calculus computes derivatives using infinitesimal disturbances in $\mathbb{R}^n$, the calculus of variations computes variations using infinitesimal disturbances in the function space. Let v be a small function[5]. Then we define the first variation (with respect to u) as

$$\frac{\delta E}{\delta u} = \int \left(v \frac{\partial J}{\partial u} + \dot{v} \frac{\partial J}{\partial \dot{u}} \right) \mathrm{d}\boldsymbol{x} \ . \tag{2.43}$$

This is known as the Gateaux derivative. It is similar in concept to a directional derivative: it is the derivative of E at u in the direction v. The weak form of the condition for a stationary point is that the first variation be zero:

$$\left. \frac{\delta E}{\delta u} \right|_{u^*} = \int \left(v \frac{\partial J}{\partial u} + \dot{v} \frac{\partial J}{\partial \dot{u}} \right) \mathrm{d}\boldsymbol{x} = 0 \ \forall v \ . \tag{2.44}$$

This is not very useful because we must evaluate this condition over every v. If we apply integration by parts (ignoring the boundary conditions) and the divergence theorem to (2.44), we obtain the strong form:

$$\frac{\partial J}{\partial u} - \nabla \cdot \left(\frac{\partial J}{\partial \dot{u}} \right) = 0 \ . \tag{2.45}$$

This is the Euler-Lagrange differential equation. The left-hand side of the equation is often referred to as the Euler-Lagrange derivative of J.

[5] By small, we mean that some norm of the function (*e.g.*, the $\mathcal{L}_2$ norm) is small.

The Euler-Lagrange differential equation can be solved in multiple ways. First, an analytical solution may exist. Otherwise we can simply discretize the PDE using finite-difference approximations and solve the resulting linear system. This is sufficient for cases when (2.45) results in a linear PDE for u. To handle more general PDEs, we can introduce a dummy time variable t. We define t in such a way that the derivative of β with respect to t equals the negative of the Euler-Lagrange derivative of J:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{\partial J}{\partial u} + \nabla \cdot \left(\frac{\partial J}{\partial \dot{u}}\right) \quad . \tag{2.46}$$

This ensures that $\frac{\mathrm{d}u}{\mathrm{d}t}$ is always pointing in a descent direction of E. The right hand side of the equation is the Euler-Lagrange operator applied to J, so at a stationary point it is equal to zero. So when we find a stationary point, u stops changing with respect to time which is the desired behavior. To obtain an estimate of u^*, we can simply integrate both sides of (2.46) with respect to t:

$$\hat{u}(\boldsymbol{x}) = u_0(\boldsymbol{x}) + \int_0^\infty \frac{\mathrm{d}u(t,\boldsymbol{x})}{\mathrm{d}t}\mathrm{d}t \tag{2.47}$$

where $u_0(\boldsymbol{x})$ represents our initial conditions. This integral can be evaluated as a discrete sum.

■ 2.1.4 Nonlinear Optimization

In this thesis, all of our discrete algorithms solve some sort of optimization problem of the form

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} E(\boldsymbol{x}) \tag{2.48}$$

where $E : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function of $\boldsymbol{x}$. This is commonly referred to as unconstrained nonlinear optimization, and there is an extremely large body of work dealing with iteratively solving these sorts of problems. All methods can be broadly classified into two categories: derivative and non-derivative. Non-derivative techniques include methods such as Powell's level set technique, downhill simplex, and Gauss-Seidel. These techniques perform iterative updates using only past and present values of the variables and the energy functional. Derivative techniques can access this information as well as knowledge of the first and possibly higher-order derivatives. Needless to say, gradient methods generally outperform non-gradient methods. Oftentimes non-derivative methods are more appropriate when analytical expressions for the derivatives cannot be found, or the derivative does not exist. For the class of problems we encounter here

this is not the case, so we will only discuss derivative-based methods. All results and theorems in this section are adapted from [8].

Our objective in solving an optimization problem is to find $\boldsymbol{x}^*$ such that for all $\boldsymbol{x} \in \mathbb{R}^n$, $E(\boldsymbol{x}^*) \leq E(\boldsymbol{x})$. This is saying that we want to find a global minimum. A global minimum may be difficult to find, it may not be unique, and it may not even exist. For instance, if $E(x) = x$, then a minimum does not exist. Even if we deal only with functions that are bounded below, a minimum may not exist. Let $E(x) = \frac{1}{1+|x|}$. Then the infimum is 0, but no choice of x can ever achieve that value.

In any case, necessary and sufficient conditions for a local minimum of E are

$$\nabla E(\boldsymbol{x}^*) = 0 \tag{2.49}$$

$$\mathcal{H}\{E(\boldsymbol{x}^*)\} > 0 \ . \tag{2.50}$$

Note that the Hessian must be strictly positive definite. If it is zero, the critical point may not be a maximum or a minimum. For instance, if $E(x) = x^3$, the derivative of E is zero at $x = 0$. The second derivative is also zero there. But $x = 0$ is a saddle point of E.

The local minimum is guaranteed to be related to the global minimum only when E has some convexity properties. We include a theorem from Bertsekas without proof.

Theorem 1. *Let $E : X \to \mathbb{R}$ be a convex function and X be a convex set. Then if $\boldsymbol{x}^*$ is a local minimum for E, $\boldsymbol{x}^*$ is also a global minimum over X. If E is also strictly convex, then there is at most one global minimum over X.*

We employ iterative solvers to find minimums of our objective function. With an iterative solver, we begin with an initial guess $\boldsymbol{x}^{(0)}$, and we wish to iteratively produce a succession of revised estimates $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$ such that $\lim_{n\to\infty} \boldsymbol{x}^{(n)} = \boldsymbol{x}^*$. Without a convex objective function, we cannot guarantee that the last relation holds true. For non-convex problems, the best these types of algorithms can guarantee is convergence to a local minimum. Which local minimum gets chosen depends largely on what initial conditions are chosen. One method of finding a better local minimum is by trying several $\boldsymbol{x}^{(0)}$ and choosing the local minimum that has the lowest energy value.

The class of iterative gradient solvers we will study here will all have updates of the form

$$\hat{\boldsymbol{x}}^{(k+1)} = \hat{\boldsymbol{x}}^{(k)} + \eta^{(k)} \boldsymbol{d}^{(k)} \forall k \geq 1 \tag{2.51}$$

with $\hat{\boldsymbol{x}}^{(0)} = \hat{\boldsymbol{x}}_0$ as the initial guess and $\boldsymbol{d}^{(k)}$ as the descent direction at iteration k. So these methods all provide incremental updates to our estimate using the estimate from the previous iteration. Generally $\boldsymbol{d}^{(k)}$ is related to the gradient of the energy functional at $\hat{\boldsymbol{x}}^{(k)}$. The only requirement that we place on E is that it be everywhere differentiable. This necessarily implies that E is continuous.

Gradient Descent

Gradient descent is the simplest of the gradient-based solvers. It is also commonly referred to as steepest descent. For (2.51), we define

$$\boldsymbol{d}^{(k)} = -\boldsymbol{g}^{(k)} \tag{2.52}$$

where $\boldsymbol{g}^{(k)} = \nabla E(\hat{\boldsymbol{x}}^{(k)})$. With this method, we descend in the direction where $E(\boldsymbol{x})$ is locally decreasing the fastest. Each iteration is guaranteed to be a descent iteration as long as $\eta^{(k)}$ is small enough (we discuss later in this section how to choose $\eta^{(k)}$). The step size $\eta^{(k)}$ can be chosen in a variety of manners, but it is important to ensure that $E(\hat{\boldsymbol{x}}^{(k+1)}) < E(\hat{\boldsymbol{x}}^{(k)})$ (otherwise we are moving away from the minimum). We see that the linearization of E around $\hat{\boldsymbol{x}}^k$ is

$$E(\boldsymbol{x}) \approx E(\hat{\boldsymbol{x}}^{(k)}) + (\boldsymbol{x} - \hat{\boldsymbol{x}}^{(k)})^{\mathrm{T}} \boldsymbol{g}^{(k)} \ . \tag{2.53}$$

This approximation holds with some tolerance ϵ within some region $\|\boldsymbol{x} - \hat{\boldsymbol{x}}^{(k)}\| < \delta_\epsilon$. So for $\eta^{(k)}$ very small (chosen such that $\hat{\boldsymbol{x}}^{(k+1)}$ is within the hypersphere of radius δ_ϵ), the linearization provides a good approximation to the full function, and we observe the following change in the energy functional:

$$\begin{aligned} E(\hat{\boldsymbol{x}}^{(k+1)}) - E(\hat{\boldsymbol{x}}^{(k)}) &\approx \eta^{(k)} \langle \boldsymbol{d}^{(k)}, \boldsymbol{g}^{(k)} \rangle \\ &= -\eta^{(k)} \|\boldsymbol{g}^{(k)}\|^2 \ . \end{aligned} \tag{2.54}$$

This value is always negative except at a local minimum. We can see that the general condition for a direction $\boldsymbol{d}^{(k)}$ to be a descent direction is $\langle \boldsymbol{d}^{(k)}, \boldsymbol{g}^{(k)} \rangle < 0$. This is equivalent to saying that the angle between the gradient and the descent vector must be between $\frac{\pi}{2}$ and $\frac{3\pi}{2}$.

When the stepsize $\eta^{(k)}$ is too small, convergence to the minimum will be slow. For $\eta^{(k)}$ too large, our gradient descent step may turn out to increase the energy. For every iteration, there is some stepsize that minimizes the energy along the descent direction:

$$\eta^{(k)} = \arg\min_{\eta} E(\hat{\boldsymbol{x}}^{(k)} + \eta \boldsymbol{d}^{(k)}) \ . \tag{2.55}$$

Solving this minimization problem is referred to as a line search. Sometimes it may be possible to easily compute $\eta^{(k)}$ in closed form and perform an exact line search. Other times we can use iterative techniques to perform approximate line searches.

If even performing an approximate line search is too difficult, we can use a heuristic method. We begin by setting $\eta^{(0)} = \eta_0$ where η_0 is some predetermined constant. We let $\eta^{(k)} = \eta_0$ until iteration k is no longer a descent step. Then we set $\eta^{(k)} = \frac{1}{2}\eta_0$ and continue with that value until we no longer decrease the energy. We continue in this manner until we reached the desired convergence tolerance. This method is known as successive stepsize reduction, and it works well it practice. But it is not guaranteed to converge to a local minimum for all cases. Armijo's Rule [3] is a modification of this technique to alleviate this theoretical difficulty. We fix β, s, and σ such that $0 < \beta < 1$ and $0 < \sigma < 1$. Then we choose $\eta^{(k)} = s\beta^{m^{(k)}}$ where $m^{(k)}$ is the smallest $m \in \mathbb{Z}^+$ such that

$$E(\hat{\boldsymbol{x}}^{(k)}) - E(\hat{\boldsymbol{x}}^{(k)} + s\beta^m \boldsymbol{d}^{(k)}) \geq -\sigma s \beta^m (\nabla E(\hat{\boldsymbol{x}}^{(k)}))^{\mathrm{T}} \boldsymbol{d}^{(k)} \ . \tag{2.56}$$

For gradient descent it can be shown that if $\eta^{(k)}$ is chosen using line minimization or Armijo's Rule, then the limit point of the iterative solver will be a stationary point. Similar statements can be made when the step size is constant or decreasing[6] and certain technical assumptions (such as the gradient of E is Lipschitz continuous) are made.

A common technique to describe convergence rates is to compare the error at each iteration with an infinite sequence. We can define the error as, *e.g.*,

$$e^{(k)} = \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| \tag{2.57}$$

or

$$e^{(k)} = |E(\boldsymbol{x}^{(k)}) - E(\boldsymbol{x}^*)| \ . \tag{2.58}$$

Then we characterize the convergence rate of an algorithm through a comparison test:

$$e^{(i)} \leq a s_i \ \forall \ i \in \mathbb{Z}^+ \tag{2.59}$$

where $a \in \mathbb{R}^+$. When there exists an a such that the above expression is true for all i and

$$s_i = r^i \tag{2.60}$$

[6]The sum $\sum_{k=1}^{\infty} \eta^{(k)}$ needs to be unbounded otherwise the iteration sequence can converge to a non-stationary point.

for some $r \in [0,1)$, then the error is said to have linear (or geometric) convergence. When there exists an a such that (2.59) is true for all i and

$$s_i = r^{[p^i]} \tag{2.61}$$

for $r \in [0,1)$ and $p > 1$, then the algorithm has superlinear convergence. When $p = 2$ the error has quadratic convergence. To give an idea of the difference between linear and quadratic convergence, linear convergence adds a digit of accuracy every $-1/\log_{10}(r)$ iterations. Quadratic convergence doubles the number of digits of precision every iteration.

Gradient descent generally has linear convergence. This property can be easily shown on quadratic optimization problems. We define a general quadratic optimization problem as

$$E(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + c \tag{2.62}$$

with $\boldsymbol{Q} > 0$ and $c = \frac{1}{2}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{a}$. The minimum value of the energy is then 0. The worst case convergence rate of gradient descent for quadratic problems is related to the condition number of $\boldsymbol{Q}$. Let M be the largest eigenvalue of $\boldsymbol{Q}$ and m be the smallest. Then the condition number is the ratio M/m. It can be shown that the cost decrease (when the step size is chosen through line minimization) takes the following form:

$$\frac{E(\hat{\boldsymbol{x}}^{(k+1)})}{E(\hat{\boldsymbol{x}}^{(k)})} \leq \left(\frac{M/m - 1}{M/m + 1}\right)^2 . \tag{2.63}$$

Thus gradient descent displays linear convergence with the rate determined by the condition number. When the condition number is large, $\boldsymbol{Q}$ is termed ill-conditioned. In these instances, there is often a steep "valley" in the energy functional. Even for convex energy functionals, the gradient (which can only measure local rate of change) is nearly orthogonal to the vector that points from the current iterate $\hat{\boldsymbol{x}}^{(k)}$ to the global minimum. This convergence rate is of course simply an upper bound, but in practice the average rate is fairly close to the worst case.

Newton's Method

Newton's method (also known as Newton-Raphson) is a technique that incorporates the Hessian into the updates. We define the descent direction for (2.51) as

$$\boldsymbol{d}^{(k)} = -[\mathcal{H}\{E(\boldsymbol{x}^{(k)})\}]^{-1}\boldsymbol{g}^{(k)} . \tag{2.64}$$

Newton's method was originally developed to find the zeros of equations:

$$f(\boldsymbol{x}) = 0 \tag{2.65}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$. Let $\boldsymbol{x}_0$ be a zero of $f(\boldsymbol{x})$. If we have an initial estimate $\hat{\boldsymbol{x}}^{(0)}$, we can build a linear approximation to $f(\boldsymbol{x})$ around $\hat{\boldsymbol{x}}^{(0)}$:

$$\tilde{f}(\boldsymbol{x}) \approx f(\hat{\boldsymbol{x}}^{(0)}) + [\tfrac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} f(\hat{\boldsymbol{x}}^{(0)})](\boldsymbol{x} - \hat{\boldsymbol{x}}^{(0)}) \ . \tag{2.66}$$

We then pick our next iterate $\hat{\boldsymbol{x}}^{(1)}$ as the zero of this approximate function:

$$\hat{\boldsymbol{x}}^{(1)} = \hat{\boldsymbol{x}}^{(0)} + [\tfrac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} f(\hat{\boldsymbol{x}}^{(0)})]^{-1} f(\hat{\boldsymbol{x}}^{(0)}) \ . \tag{2.67}$$

We choose $\hat{\boldsymbol{x}}^{(2)}, \hat{\boldsymbol{x}}^{(3)}, \ldots$ in a similar manner. We can apply Newton's method to find local minima/maxima by having it solve for the condition that the gradient is zero:

$$f(\boldsymbol{x}) = \nabla E(\boldsymbol{x}) = 0 \ . \tag{2.68}$$

This leads naturally to (2.64).

If the Hessian is positive semi-definite, then the direction we take is a descent direction. Otherwise there are no guarantees. If we are dealing with non-positive Hessians, we generally regularize the Hessian to make it positive definite:

$$\boldsymbol{d}^{(k)} = -[\mathcal{H}\{E(\boldsymbol{x}^{(k)})\} + \zeta \boldsymbol{I}]^{-1} \boldsymbol{g}^{(k)} \ . \tag{2.69}$$

The value of ζ needs to be chosen sufficiently large so that $\mathcal{H}\{E(\boldsymbol{x}^{(k)})\} + \zeta \boldsymbol{I} \geq 0$. This means that ζ should be at least as large as the absolute value of the most negative eigenvalue of the Hessian.

The main advantage of Newton's method is that it is extremely fast. For conventional Newton's method, the stepsize $\eta^{(k)}$ is fixed at 1. For a quadratic problem, this will guarantee that the minimum is found in exactly one iteration. The error converges superlinearly when Newton's method is used with Armijo's Rule.

Besides the possibility of a non-positive Hessian, there are also some other drawbacks with using Newton's method. The main problem is poor capture range. When our initial estimate is far from the true solution, Newton's method often cannot find the true solution, even for convex problems. This is compounded by the fact that Newton's method only finds points where the gradient is zero. This can occur at a maximum, a minimum, or a saddle point. So Newton's method is as attracted to local maxima

as it is to local minima. In large problems, the inversion of the Hessian can be very computationally consuming. In some problems, the gradient may be available in closed form, but the Hessian may be difficult to compute[7]. In practice, the algorithm often begins with a method such as steepest descent to get an estimate that is close to a local minimum. Then this result is used as the initial estimate for Newton's method. This allows us to obtain the quadratic convergence of Newton's method close to the solution and the wide capture range of gradient descent far from the solution. Regardless of the large number of these difficulties, Newton's method is widely applied in practice due to the extremely fast convergence rate.

Conjugate Gradient

We can characterize the two previous techniques as being memoryless. When generating $\hat{\boldsymbol{x}}^{(k+1)}$, the only thing that matters is the current state at k. All of the previous iterations do not matter (except for how they lead to us arriving at $\hat{\boldsymbol{x}}^{(k)}$). In contrast, with conjugate gradient, *every* previous iteration affects the next iteration. So running conjugate gradient for 10 iterations is not the same thing as running conjugate gradient twice for 5 iterations each time. Even though conjugate gradient only uses first derivative information in generating descent directions, it is able to achieve superlinear convergence.

Conjugate gradient is best suited to solving unconstrained quadratic optimization problems. Let the dimensionality of our problem be n. Then conjugate gradient will find the exact minimum of this functional in n iterations.

We say two vectors are $\boldsymbol{Q}$-conjugate when $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{\boldsymbol{Q}} = 0$. Conjugate gradient begins by performing a gradient descent step for the first iteration. Then subsequent descent directions are chosen so that the direction is $\boldsymbol{Q}$-conjugate with all previous descent directions. Let $\boldsymbol{g}^{(k)}$ be the gradient of the energy functional at iteration k and $\mathcal{G}^{(k)} = \{\boldsymbol{x} | \boldsymbol{x} = \sum_{i=0}^{k} \alpha_i \boldsymbol{g}^{(i)} \, \forall \, \alpha_i \in \mathbb{R}\}$ be the subspace spanned by the gradient vectors from iterations 0 through k. To generate $\hat{\boldsymbol{x}}^{(k+1)}$, we want to choose an orthogonal basis for $\mathcal{G}^{(k)}$. We have an orthogonal basis from iteration k for $\mathcal{G}^{(k-1)}$, and $\mathcal{G}^{(k-1)} \subset \mathcal{G}^{(k)}$ with the dimensionality of the latter being 1 greater than that of the former. Hence we can simply add $\boldsymbol{g}^{(k)}$ to $\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}\}$ to obtain a basis for $\mathcal{G}^{(k)}$, and we can apply Gram-Schmidt orthogonalization [70] to choose $\boldsymbol{d}^{(k)}$. Doing this leads to the following

[7]This can be partially alleviated using so-called quasi-Newton methods which slowly build up approximations to the Hessian as we progress through the iterations.

relation:

$$\boldsymbol{d}^{(k)} = -\boldsymbol{g}^{(k)} + \beta^{(k)}\boldsymbol{d}^{(k-1)} \tag{2.70}$$

$$\beta^{(k)} = \frac{\|\boldsymbol{g}^{(k)}\|^2}{\|\boldsymbol{g}^{(k-1)}\|^2} \ . \tag{2.71}$$

Note that this is true only if all of the stepsizes $\eta^{(k)}$ for (2.51) are chosen through exact line minimization. If approximate line minimization techniques or other methods such as Armijo's Rule are used, we slowly lose $\boldsymbol{Q}$-conjugacy, and the solver needs to be reset every so often with a gradient step.

Conjugate gradient can also be adapted to general nonlinear functionals. One method to accomplish this would be to construct a quadratic local approximation of the energy functional and apply conjugate gradient to that. This has been found to not work as effectively as more heuristic methods of choosing $\beta^{(k)}$ such as

$$\beta^{(k)} = \frac{(\boldsymbol{g}^{(k)} - \boldsymbol{g}^{(k-1)})^{\mathrm{T}}\boldsymbol{g}^{(k)}}{\|\boldsymbol{g}^{(k-1)}\|^2} \ . \tag{2.72}$$

The general nonlinear conjugate gradient needs to be reset with a gradient descent step every once in a while because conjugacy gets lost, even with exact line searches.

To see why conjugate gradient is much more effective than gradient descent, we note two main results for quadratic optimization problems with convex energy functionals (*i.e.*, $\boldsymbol{Q} > 0$). We define a manifold $\mathcal{M}^{(k)} = \{\boldsymbol{x} | \boldsymbol{x} = \hat{\boldsymbol{x}}^{(0)} + \boldsymbol{z} \, \forall \, \boldsymbol{z} \in \mathcal{G}^{(k)}\}$. We note that $\mathcal{G}^{(k)}$ is spanned by both $\{\boldsymbol{g}^{(0)}, \ldots, \boldsymbol{g}^{(k)}\}$ and $\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k)}\}$.

Theorem 2 (Expanding Manifold Theorem). *Let $\hat{\boldsymbol{x}}^{(0)}$ be any point in $\mathbb{R}^n$ and $\{\hat{\boldsymbol{x}}^{(1)}, \ldots, \hat{\boldsymbol{x}}^{(m)}\}$ be generated by conjugate gradient with exact line search. Then*

$$\hat{\boldsymbol{x}}^{(k)} = \arg \min_{\boldsymbol{x} \in \mathcal{M}^{(k)}} E(\boldsymbol{x}) \ . \tag{2.73}$$

This theorem guarantees that after k iterations, we have found the optimum (when $\boldsymbol{Q} \geq 0$) over the manifold $\mathcal{M}^{(k-1)}$. The Expanding Manifold Theorem tells us why we could make the claim that conjugate gradient solves convex quadratic optimization problems of dimension n in exactly n iterations or less. If we do n conjugate gradient steps, then we have an iterate $\hat{\boldsymbol{x}}^{(n)}$ that satisfies the following property:

$$\hat{\boldsymbol{x}}^{(n)} = \arg \min_{\boldsymbol{x} \in \mathcal{M}^{(n-1)}} E(\boldsymbol{x}) \ . \tag{2.74}$$

We then note that $\mathcal{M}^{(n-1)} = \mathbb{R}^n$, and we have thus found the global minimum. An even stronger statement can be made. As noted by Bertsekas, conjugate gradient is optimal for solvers with update equations of the form:

$$\hat{\boldsymbol{x}}^{(k+1)} = \hat{\boldsymbol{x}}^{(k)} + \gamma_0^{(k)}\boldsymbol{g}^{(0)} + \ldots + \gamma_k^{(k)}\boldsymbol{g}^{(k)} \ . \tag{2.75}$$

Optimal in this sense means that conjugate gradient finds the coefficients $\gamma_i^{(k)}$ that minimize $E(\hat{\boldsymbol{x}}^{(k)} + \gamma_0^{(k)}\boldsymbol{g}^{(0)} + \ldots + \gamma_k^{(k)}\boldsymbol{g}^{(k)})$. Gradient descent falls in this framework with $\gamma_i^{(k)} = 0$ for $i = 0, \ldots, k-1$.

Theorem 3. *Let $\boldsymbol{Q}$ have $n-k$ eigenvalues in the interval $[a, b]$ and k eigenvalues greater than b. Additionally let $\hat{\boldsymbol{x}}^{(k+1)}$ be the vector resulting from $k+1$ steps of conjugate gradient with initial vector $\hat{\boldsymbol{x}}^{(0)}$ any vector in $\mathbb{R}^n$. Then*

$$\frac{E(\hat{\boldsymbol{x}}^{(k+1)})}{E(\hat{\boldsymbol{x}}^{(0)})} \leq \left(\frac{b-a}{b+a}\right)^2 \ . \tag{2.76}$$

Theorem 3 is valid for the functional $E(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$. It shows that each conjugate gradient iteration eliminates the effect of the largest remaining eigenvalue of $\boldsymbol{Q}$. This property is true for general quadratic functionals as well but is more cumbersome notationally. Many problems tend to have a few really large eigenvalues, and once those are eliminated, convergence is very fast. This is observed in practice with conjugate gradient having the tendency to take a couple of ineffective steps to begin with, and then suddenly take a number of really fast steps.

Preconditioners

As discussed above, the convergence rate of techniques such as gradient descent and conjugate gradient tend to be characterized by the condition number of the $\boldsymbol{Q}$ matrix. For a non-invertible matrix, the linear system has no solution, and the condition number is infinite. A matrix with a condition number of one is well-conditioned. The level sets of the energy functional are hyperspheres and both gradient descent and conjugate gradient will find the minimum in exactly one iteration (using exact line search).

The basic idea of using preconditioners is to replace the system $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{a}$ with another system $\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{a}}$. We design the second linear system so that the answer it produces can be used to get to the solution of the first system while also minimizing the condition number of $\tilde{\boldsymbol{Q}}$. This can be viewed as incorporating some knowledge of the curvature of the problem (because $\boldsymbol{Q}$ is the Hessian of a quadratic problem).

In order to do this, we transform the original linear system into a new set of coordinates:

$$\tilde{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{x} \tag{2.77}$$

$\boldsymbol{S}$ should be symmetric and invertible. Then in order for both linear systems to be equivalent subject to (2.77), the following need to be true:

$$\tilde{\boldsymbol{Q}} = \boldsymbol{S}^{-1}\boldsymbol{Q}\boldsymbol{S}^{-1} \tag{2.78}$$

$$\tilde{\boldsymbol{a}} = \boldsymbol{S}^{-1}\boldsymbol{a} \ . \tag{2.79}$$

If $\boldsymbol{S} = \boldsymbol{Q}^{1/2}$ ($\boldsymbol{Q}$ is assumed to be positive definite so a square root always exists), then $\tilde{\boldsymbol{Q}} = \boldsymbol{I}$, and the system is well-conditioned. The goal of using preconditioners is to make $\boldsymbol{S}$ approximate $\boldsymbol{Q}^{1/2}$ while also making the operation computationally efficient.

Multigrid

The error for iterative solvers does not diminish in a uniform manner with respect to frequency. The high-frequency errors diminish much more rapidly than the low-frequency errors. The reason for this is that high-frequency errors are reflected in local interactions while low-frequency errors result in more global interactions. Multigrid attempts to address this issue [13, 86] .

Multigrid is generally associated with non-gradient solvers such as Gauss-Seidel or successive over-relaxation (SOR). There are many variations, but the essential idea is that the low-frequency components of the solution converge faster when the problem is sampled on a coarser grid, and, as an added bonus, iterations require less time to compute. The classical multigrid techniques involve so-called V-cycles. One iteration of the iterative solver is performed at the finest level. This is termed a presmoothing step. The result of that iteration is resampled onto a coarser grid, and another step of the iterative solver is performed. This is repeated until we reach the coarsest level desired. Once that level is reached, the last iteration is interpolated onto a finer grid. Then a step of the iterative solver is executed. This is called a postsmoothing step. This is repeated until the finest level is reached. This entire process is called a V-cycle because we begin at the finest level, descend to the coarsest, and rise back up to the finest. The graph of this procedure looks like a "V". Multiple V-cycles are run until the desired convergence level is achieved. Multigrid can be proven to have $\mathcal{O}(n)$ complexity properties. Many multigrid-like implementations do not go through multiple V-cycles

because the transitions between levels can be costly in terms of computation. We will use multigrid to refer to the class of algorithms that use a multiresolution representation of the image and use the coarse representations to obtain the low-frequency components of the final reconstruction.

Half-Quadratic Optimization

Half-quadratic optimization is a technique popularized by Geman and Reynolds [26] and Geman and Yang [27]. Half-quadratic techniques can optimize problems of this form:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \alpha \sum_{i}^{N_l} \phi(\boldsymbol{l}_i^{\mathrm{T}} \boldsymbol{x}) \ . \tag{2.80}$$

We define a matrix $\boldsymbol{L}$ such that its ith row is $\boldsymbol{l}_i^{\mathrm{T}}$. We choose $\phi(t) = |t|^p$ to make this into a ℓ_p regularization problem. Half-quadratic methods are more general than this, but we are only interested in this special case. Note that gradient descent can also be used to perform this optimization, but half-quadratic optimization has proven to be faster empirically [78].

Let E be the energy functional we are trying to minimize in (2.80). We can define an augmented cost function

$$\mathcal{E}(\boldsymbol{x}, \boldsymbol{w}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \alpha \sum_{i}^{N_l} \left(Q(\boldsymbol{l}_i^{\mathrm{T}} \boldsymbol{x}, w_i) + \psi(w_i) \right) \tag{2.81}$$

where w_i is the ith element of $\boldsymbol{w}$, $Q(t, w)$ is a function quadratic in terms of t, and $\psi(\cdot)$ is called the dual function. We choose $Q(t, w) = t^2 w$ to obtain the so-called multiplicative form [54] of half-quadratic optimization. This makes the energy functional

$$\mathcal{E}(\boldsymbol{x}, \boldsymbol{w}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \alpha (\boldsymbol{L}\boldsymbol{x})^{\mathrm{T}} \boldsymbol{W}^{(k)} (\boldsymbol{L}\boldsymbol{x}) + \alpha \sum_{i}^{N_l} \psi(w_i) \ . \tag{2.82}$$

The function $\psi(\cdot)$ must be chosen to make the two energy functionals equivalent:

$$\phi(t) = |t|^p = \inf_{w} \{ t^2 w + \psi(w) \} \ . \tag{2.83}$$

Then the minimum of (2.81) occurs for the same argument as for (2.80). The equivalent constraint written for $\psi(\cdot)$ is

$$\psi(w) = \sup_{t} \{ |t|^p - t^2 w \} \ . \tag{2.84}$$

If $p \geq 1$, the problem is convex, and the limit exists. Hence we can replace inf and sup in the preceding equations to min and max.

We can minimize (2.81) using coordinate descent. This is an iterative technique that alternatively minimizes the energy with respect to $\boldsymbol{x}$ and $\boldsymbol{w}$. Define a diagonal matrix $\boldsymbol{W}$ such that the entries of $\boldsymbol{w}$ are along the diagonal. Then

$$\hat{\boldsymbol{w}}^{(k)} = \arg\min_{\boldsymbol{w}} \mathcal{E}(\boldsymbol{x}^{(k-1)}, \boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \sum_{i}^{N_l} w_i (\boldsymbol{l}_i^{\mathrm{T}} \boldsymbol{x}^{(k-1)})^2 + \sum_{i}^{N_l} \psi(w_i) \quad (2.85)$$

$$\hat{\boldsymbol{x}}^{(k)} = \arg\min_{\boldsymbol{x}} \mathcal{E}(\boldsymbol{x}, \boldsymbol{w}^{(k)}) = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + (\boldsymbol{L}\boldsymbol{x})^{\mathrm{T}} \boldsymbol{W}^{(k)} (\boldsymbol{L}\boldsymbol{x}) \ . \quad (2.86)$$

It turns out that the first optimization can be done element-by-element:

$$\hat{w}_i^{(k)} = \frac{\phi'(\boldsymbol{l}_i^{\mathrm{T}} \boldsymbol{x}^{(k-1)})}{2\boldsymbol{l}_i^{\mathrm{T}} \boldsymbol{x}^{(k-1)}} \ . \quad (2.87)$$

The second optimization can be written in terms of a matrix inversion:

$$\hat{\boldsymbol{x}}^{(k)} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \alpha \boldsymbol{L}^{\mathrm{T}} \boldsymbol{W}^{(k)} \boldsymbol{L})^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{y} \ . \quad (2.88)$$

■ 2.2 Magnetic Resonance Imaging

In this section we will discuss the physics of MR imaging. We begin by covering the NMR effect and how the MR imaging process exploits it to provide data about the tissue at the molecular level. We then discuss the sources of noise in the resulting MR image and accurate statistical characterizations of the noise. We introduce the bias field problem by explaining many of the systematic errors introduced in the MR imaging process and the effects they have on the final MR images.

■ 2.2.1 MR Physics

The nuclear magnetic resonance (NMR) effect was first observed separately by Bloch *et al.* and Purcell *et al.* in 1946 [10,60]. They observed that nuclei of different atoms absorbed electromagnetic waves of different frequencies. MR imaging was developed in the 1970s with Damadian *et al.* capturing the first whole-body image in 1977. It is extremely popular in a variety of clinical settings for its excellent soft-tissue contrast and imaging flexibility. One of the major deficiencies of computed tomography (CT) is that all it can measure is tissue density. All details from this section are adapted from [82] and [51].

MR imaging generally deals with NMR of the hydrogen atom. The human body is composed of 70% water of which each molecule contains two hydrogen atoms. In its most common form, the hydrogen atom contains one proton and one electron. Quantum mechanics tells us that the angular momentum or spin of the nucleus can take on one of two values in the presence of an applied magnetic field $\boldsymbol{B_0}$[8]: $\pm\hbar/2$ where $\hbar = h/2\pi$ and h is Planck's constant. The strength of the applied field sets the distribution of positive and negative spins. In practice, the excess number of protons with positive spin is very small[9]. Because the nucleus is positively charged, the spin causes a magnetic moment which points in the same direction as the spin. The two properties are related by the scale factor γ which is referred to as the magnetogyric ratio. γ varies depending on the atom being targeted. For hydrogen, $\gamma = 42.57$ MHz/T[10].

Without loss of generality, we will assume that $\boldsymbol{B_0}$ points in the z-direction. The angular momentum of the nucleus does not point in the exact direction of $\boldsymbol{B_0}$, but instead the spin wobbles around that direction like a spinning top. This wobble is called precession. The Larmour equation tells us that the frequency of precession (known as the Larmour frequency) is:

$$\omega_0 = -\gamma \|\boldsymbol{B_0}\| \ . \tag{2.89}$$

This means that the frequency is proportional to the strength of the applied magnetic field. This provides the key to MR imaging. Because we deal only with hydrogen nuclei, γ is constant. If we vary the intensity of $\boldsymbol{B_0}$ in space, the Larmour frequency will then vary in space. This means that the response of different locations in space will be encoded at different frequencies.

Once the large $\boldsymbol{B_0}$ field is in place, we apply a series of radio frequency (RF) pulses to the system at the Larmour frequency. These RF pulses tip the net magnetization vector into a coherent direction in the xy-plane. Once the pulses are applied, the steady-state tendency of the system is to have all spins realigned with $\boldsymbol{B_0}$. While that occurs, the magnetization vector in the xy-plane will also destabilize. Both can be

[8]We deviate from our standard notation of using lower-case letters for vectors and upper-case letters for matrices. The usage of $\boldsymbol{B_0}$ is so standard in the literature that it would simply be confusing trying to use something else.

[9]The excess number of protons with positive spin is mainly a function of the strength of $\boldsymbol{B_0}$. This is why researchers continually attempt to increase the strength of the $\boldsymbol{B_0}$ field. The larger the excess in positive-spin protons, the larger the observed NMR effect.

[10]T represents the Tesla which is a unit of magnetic flux density. MR magnet strengths are often specified in terms of the Tesla.

approximated as exponentially decaying processes. The time constant associated with the recovery of the longitudinal z-magnetization vector is referred to as T_1. This is known as spin-lattice relaxation. The time constant associated with the decay of the transverse xy-magnetization is called T_2. This is known as spin-spin relaxation. These values vary depending on tissue type.

In most commercial MR systems the $\boldsymbol{B_0}$ field is generated using a superconducting magnet. We can transmit and receive using the same coil, but generally separate coils are used in order to maximize SNR. The transmitting coil begins sending its pulses at time 0. At a time referred to as the time of echo (T_E), we measure the voltage in the receiving coil (Faraday's Law tells us that a voltage is induced by the changing magnetic moment). The RF pulse sequence is usually repeated at an interval T_R. Spatial coherence is obtained through frequency encoding using gradients on $\boldsymbol{B_0}$. This variation in space is typically referred to as the $\boldsymbol{B}_1$ field. Because the spatially distributed data are now located in the frequency domain (often called the k-space), we can use the Fourier transform to recover the image data. Due to slight phase errors, we typically obtain a complex-valued image. The absolute value of this image is used as the final image.

The image intensity that we observe at a specific voxel can be described in terms of two main imaging parameters (T_E and T_R) and three main intrinsic properties of the tissue (T_1, T_2, and the proton density ρ):

$$\varphi(\boldsymbol{x}) = \rho(\boldsymbol{x})e^{-T_E/T_2(\boldsymbol{x})}(1 - e^{-T_R/T_1(\boldsymbol{x})}) \ . \tag{2.90}$$

This equation holds for spin-echo sequences. Other pulse sequences will produce different dependencies on the imaging parameters.

By choosing a small T_E and a large T_R, the images we observe have little dependence on T_1 and T_2 and are termed PD-*weighted*. By choosing a small T_E and a moderate T_R, there is little dependence on T_2 and the images are termed T_1-*weighted*. By choosing a moderate T_E and a large T_R, we observe few effects from T_1 and the images are called T_2-*weighted*. Thus through appropriate manipulation of T_E and T_R, we can capture a variety of images that will hopefully allow us to differentiate tissue types.

For our purposes the important thing is not absolute measure of these intrinsic tissue properties. This is in contrast to, *e.g.*, NMR spectroscopy which tries to examine the fundamental properties of molecular structures. We just want our imaging process to produce contrast among different tissue types. X-ray based CT is very good at depicting

contrast between bone and soft tissue. However, most soft tissues have similar densities. Properties such as T_1 and T_2 get more at the molecular structure of the organs and hence can show good contrast between different tissue types, indicate the liquid vs. solid content, depict some internal structure within organs, and even highlight tumors.

2.2.2 Noise

The radio frequency signals that are observed at the receiver are corrupted by thermal noise which is accurately modeled as Gaussian-distributed and white. The data is then Fourier transformed to convert from the k-space representation to spatial coordinates. Because the Fourier transform is a linear transformation, the noise remains Gaussian at the output. As mentioned earlier, due to phase encoding errors, the reconstructed image will be complex. Thus there is white Gaussian noise on both the real and imaginary components of the signal. When we take the absolute value of this, the noise becomes Rician [61, 66].

The Rician PDF is generated by taking the absolute value of the sum of two Gaussian random variables with non-zero mean. It is often used in wireless communications to model channel fading. Say that we observe two deterministic measurements corrupted by additive Gaussian noise:

$$\mathsf{x}_1 = \mu_1 + \mathsf{n}_1 \tag{2.91a}$$

$$\mathsf{x}_2 = \mu_2 + \mathsf{n}_2 \tag{2.91b}$$

where $\mathsf{n}_1 \sim \mathcal{N}(0, \sigma^2)$ and $\mathsf{n}_2 \sim \mathcal{N}(0, \sigma^2)$. Then x_1 and x_2 are both Gaussian random variables with $\mathsf{x}_1 \sim \mathcal{N}(\mu_1, \sigma^2)$ and $\mathsf{x}_2 \sim \mathcal{N}(\mu_2, \sigma^2)$. We define our Rician random variable as $\mathsf{r} = \sqrt{\mathsf{x}_1^2 + \mathsf{x}_2^2}$. This has a PDF of

$$\mathrm{p}_\mathsf{r}(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) \mathrm{I}_0\left(\frac{Ar}{\sigma^2}\right) \tag{2.92}$$

for $r \geq 0$. I_0 is the zeroth-order modified Bessel function of the first kind:

$$\mathrm{I}_0(x) = \frac{1}{2\pi} \int_0^{2\pi} \exp(x \cos(\theta)) \mathrm{d}\theta \ ,$$

and $A^2 = \mu_1^2 + \mu_2^2$.

The total power of the signal is

$$P = E[\mathsf{r}^2] = E[\mathsf{x}_1^2 + \mathsf{x}_2^2] = \mu_1^2 + \sigma^2 + \mu_2^2 + \sigma^2 = A^2 + 2\sigma^2 \ . \tag{2.93}$$

The signal power is A^2, and the noise power is $2\sigma^2$. It is common to reparameterize the Rician distribution in terms of the SNR. This is referred to as the k factor: $k = A^2/2\sigma^2$. We can now summarize the equations that govern the relationships between k, P, A, and σ:

$$k = \frac{A^2}{2\sigma^2} \tag{2.94a}$$

$$P = A^2 + 2\sigma^2 \tag{2.94b}$$

$$\sigma^2 = \frac{P}{2(k+1)} \tag{2.94c}$$

$$A^2 = \frac{kP}{k+1} \ . \tag{2.94d}$$

This allows us to rewrite our PDF in terms of k and P:

$$\mathrm{p_r}(r) = \frac{2(k+1)\frac{r}{\sqrt{P}}}{\sqrt{P}} \exp\left(-(k+1)\left(\frac{r}{\sqrt{P}}\right)^2 - k\right) \mathrm{I}_0\left(2\sqrt{k(k+1)}\left(\frac{r}{\sqrt{P}}\right)\right) \ . \tag{2.95}$$

Even moments of r are easy to calculate, but computing odd moments is difficult due to the square root. The general equation for the moments can be characterized using the confluent hypergeometric function ${}_1F_1$ and the gamma function Γ:

$$\mu_n = \mathrm{E}\left[\mathrm{r}^n\right] = \left(\tfrac{P}{k+1}\right)^{n/2}\Gamma(1+\tfrac{n}{2})\,{}_1F_1\left(-\tfrac{n}{2};1;-k\right) \ . \tag{2.96}$$

The confluent hypergeometric function is defined as

$${}_1F_1(a,b;z) \equiv \frac{\Gamma(b)}{\Gamma(b-a)\Gamma(a)} \int_0^1 e^{zt} t^{a-1}(1-t)^{b-a-1}\mathrm{d}t \tag{2.97}$$

and the gamma function as

$$\Gamma(x) \equiv \int_0^\infty t^{x-1}e^{-t}\mathrm{d}t \ . \tag{2.98}$$

Numerical aspects of these functions can be found in [1].

When we are evaluating an even moment $n = 2m$, then all of the $n/2$ terms become integers:

$$\mu_{2m} = \mathrm{E}\left[\mathrm{r}^{2m}\right] = \left(\tfrac{P}{k+1}\right)^m \Gamma(m+1)\,{}_1F_1\left(-m;1;-k\right) \ . \tag{2.99}$$

For this special case, the confluent hypergeometric function can be described in terms of the Laguerre polynomials:

$${}_1F_1\left(-m;1;-k\right) = L_m(-k) = \sum_{i=0}^{m} \binom{m}{m-i} \frac{k^i}{i!} \ . \tag{2.100}$$

The gamma function can be viewed as a generalization of the factorial operation to the entire real line. It has the following special property:

$$\Gamma(x+1) = x! \text{ for } x \in \mathbb{Z}^+ \ . \tag{2.101}$$

All other values of x must be evaluated using (2.98). Substituting (2.100) and (2.101) into (2.99) shows that even moments are easily described using polynomial functions of k and P (or alternatively A and σ).

Regardless of whether the noise is Gaussian or Rician, high noise levels can significantly decrease the usability of medical imagery. The noise in the complex image is thermal in nature and can be viewed as continuous white Gaussian noise in four dimensions (three spatial and time). We may maximize SNR by increasing acquisition time, increasing voxel size, increasing coil sensitivity, and applying appropriate edge-preserving filtering techniques. All of these methods have corresponding limitations.

As noted earlier on Page 52, the pulse sequence is usually repeated a few times with the time between repetitions denoted as T_R. The signals measured for each repetition are then averaged to form the final k-space image. This results in the noise being averaged over time, and the noise level can be viewed as being inversely proportional to signal acquisition time. The main issue is increasing the number of excitations increases the overall imaging time. This makes the scanning procedure more uncomfortable for the patient and also proportionately increases image acquisition costs (time is money). Additionally, longer imaging times increase the likelihood that the patient will shift during the scanning procedure. For prostate imaging, the voxels may have dimensions of about $0.5 \times 0.5 \times 3.0$ mm. So even imperceptible movements by the patient can result in significant blurring between voxels. In reality large patient movements are fairly common and can result in fairly serious motion artifacts.

The digital MR image is sampled from the continuous signal in 3D voxels. Thus the noise observed in each voxel is inversely proportional to the volume of the voxel (relating to both intra-slice dimensions as well as slice thickness). In choosing voxel size, there is a trade-off between noise and resolution. Larger voxels are desirable due to noise considerations. Smaller voxels are good because they let users discern finer-scaled structure. Larger voxels also suffer more from partial volume effects where voxels on tissue boundaries contain more than one tissue type. This results in blurring of intensities on tissue boundaries. For large objects such as the brain, the relevant structures tend to be fairly large so the voxels can be commensurately large. Imaging

small objects is more problematic. For a small object (*e.g.*, the prostate) in order to attain the same SNR possible for larger objects, a much wider FOV (relative to the size of the object of interest) must be used. This can limit fine-scaled detail.

An immense number of filter techniques have been published in the literature, including a method by Nowak [55] specifically designed for MR images. These filtering techniques can take advantage of the locally-smooth nature of our data to reduce the noise. Unfortunately there is a tradeoff between SNR and edge fidelity. Linear low-pass filters are commonly employed to take advantage of the fact that white noise has a uniform frequency spectrum whereas most of the energy in real images tends to be in low-frequency bands. A linear filter can eliminate the noise in frequency bands where the image and noise do not overlap but can only partially suppress the noise in regions where they do overlap. Nonlinear filters such as the anisotropic diffusion method of Perona and Malik [58] provide improved edge-preservation performance. Nevertheless without explicit foreknowledge of the edge locations, it is impossible to completely prevent blurring of the edges.

Finally, rather than reduce the noise level, we can also increase the signal level to boost SNR. The signal strength is proportional to the coil sensitivity which is a function of space. Unfortunately, it is only possible to increase the sensitivity in a small region of space while reducing the sensitivity in other regions [17]. A higher desired maximum coil response results in more pronounced inhomogeneities. This spatially-varying sensitivity is the primal cause of MR intensity inhomogeneities. We will discuss this further in the next section.

■ 2.2.3 Intensity Inhomogeneities

The MR bias problem occurs in all MR imaging applications. The extent with which it occurs and the amount of correction that is needed varies from application to application. Intensity inhomogeneities occur due to a large number of reasons [39]. Most tissue has some magnetic susceptibility which can distort the magnetic field. The main magnetic field $\boldsymbol{B_0}$ is not truly uniform in space, and the slope of the gradient-encoding field $\boldsymbol{B_1}$ is not truly linear. The response of the transmitting coil (usually the body coil) is not completely uniform. And, of course, the response of the receiving coil is also not uniform, often very much so. The variations in $\boldsymbol{B_0}$ and $\boldsymbol{B_1}$ actually produce spatial location errors rather than intensity errors. The other effects can be grouped into one

spatially-varying function β:

$$\tilde{\varphi}(\boldsymbol{x}) = \beta(\boldsymbol{x})\varphi(\boldsymbol{x}) \ . \tag{2.102}$$

In this instance, φ is the function defined in (2.90) on page 52. We will refer to φ as the intrinsic image (using a term from computer vision [84]) or the true image. Bias field, intensity inhomogeneity, and coil reception profile will be used interchangeably to describe β. We will concentrate on the effects caused by the inhomogeneity in the receiving coil. In surface coil imaging, this effect dwarfs the other effects. The bias differs from noise because it is a systematic error. Under identical scanning conditions, the bias will remain unchanged while the noise will randomly fluctuate.

In some applications the body coil is used to receive the MR signal. In others a surface coil is used. The problem with using the same coil to transmit and receive is that different and largely incompatible design characteristics are desired for the two tasks [51, 82]. The transmitting coil needs to have as homogeneous a response as possible. Inhomogeneities in that response have a highly nonlinear effect on the resulting image without a commensurate gain in SNR[11]. The transmitting coil should also have a fairly decent Q-factor [56]. The Q-factor describes how peaky the response is in the frequency domain. High sensitivity near the Larmour frequency and low sensitivity for other frequencies is desired, but the Q-factor should not be too high otherwise ringing will occur in the pulse sequence. For the receiving coil, the main design consideration is sensitivity in the region to be imaged. High Q-factors are desirable, and ringing is less of a concern. Inhomogeneities are still unwanted, but maximizing SNR is the main goal.

Surface coils are coils that are placed very close to the object of interest [4]. They exhibit a strong response close to the coil, and the response rapidly decreases with distance. This maximizes SNR in the region of interest at the expense of intensity homogeneity. Surface coils are widely used in a variety of applications such as the knee, pelvis, and spine. Studies have shown that radiologists are more adept at ignoring the bias field (in effect performing a mental bias correction) than they are at ignoring noise. This leads to more accurate diagnosis [64]. There are a third class of receiving coils typified by the head coil. The more general class is termed quadrature birdcage coils, and they are a compromise between the higher signal levels of surface coils and the homogeneity of body coils.

[11]There are also federal limitations in place that regulate the level of the transmitted MR signal to prevent excessive tissue heating. These signal levels can be achieved using the body coil so higher gains are unnecessary.

Designing surface coils for most applications is relatively straightforward. Most surface coils are just a loop of wire placed near the object of interest. To design a surface coil for the prostate, however, some rather ingenious measures needed to be taken [49, 65]. The prostate is embedded in the middle of the body with no direct exposure to the outer surface. The coil used for the prostate is an endorectal coil which is copper wire taped to the inside of a concave balloon. The balloon is inserted into the rectum and inflated with the concave side facing towards the prostate. The concavity allows the coil to tightly nest against the prostate.

Bias fields tend to be spatially smooth in nature. The strength of the bias field is proportional to the norm of the magnetic field that would be induced by a constant current running through the coil. This magnetic field can be described by the solution to a vector Poisson's equation [17]. With a constant current, the solution is infinitely differentiable (*i.e.*, a member of C^∞). This means that there cannot be any discontinuities in the field and imposes a certain amount of smoothness as well. When the norm of the vector field is taken (which is an analytic operation), these properties are preserved.

■ 2.3 Intensity Correction Techniques

The fundamental problem with most approaches in the literature is that the bias field and intrinsic image are simply not separable without more information. We know *a priori* that the bias field is low frequency and smooth, and the intrinsic image is piecewise smooth. This does not sufficiently restrict the space of possible solutions. We can broadly classify existing bias correction techniques as retrospective or non-retrospective. The non-retrospective techniques simplify the problem by altering the scanning protocol to acquire more information. The various retrospective approaches build in mechanisms that restrict the solution space to make the problem tractable. This restriction can transform an ill-posed problem into a well-posed problem, but the true solution is most likely not within the solution space anymore (an answer very close to the true solution may remain). Some methods do linear filtering which ignores the fact that the bias field and the intrinsic image are not separable in the frequency domain. Others build bias field estimates in a parametric manner. Some recognize that segmentation and bias correction are related problems and attempt to perform simultaneous bias correction and segmentation.

We add noise to (2.102) to generate the typical observation model:

$$\psi(\boldsymbol{x}) = \beta^*(\boldsymbol{x})\varphi^*(\boldsymbol{x}) + n(\boldsymbol{x}) \ . \tag{2.103}$$

We use β^* and φ^* to refer to the true bias field and intrinsic image respectively. The objective of bias correction algorithms is to estimate φ^*. This can be done through directly estimating φ^* or by estimating β^* to get φ^*. Many of these methods operate on the log transform of these quantities. We will use the following notation:

$$\tilde{\psi}(\boldsymbol{x}) = \log \psi(\boldsymbol{x}) \tag{2.104a}$$

$$\tilde{\beta}^* = \log \beta^*(\boldsymbol{x}) \tag{2.104b}$$

$$\tilde{\varphi}^* = \log \varphi^*(\boldsymbol{x}) \tag{2.104c}$$

■ 2.3.1 Classical Techniques

The traditional method for radiologists to manually compensate for dynamic range issues in medical imagery is referred to as *window/level*. The *window* and *level* define a function that maps values generated by the imaging process to grayscale intensities to be displayed on a screen. In practice the window and level are used to specify an interval of input values that are mapped onto the full range of grayscale values. Anything falling below the interval is assigned to the lowest grayscale intensity, and anything higher than the interval receives the highest grayscale intensity. There is a linear map for any intensities inside the interval. The window value specifies the width of the interval, and the level value specifies the center of the interval. It is essentially a two-parameter gamma correction. This produces results that can cut off the worst effects of the bias field and thus make the images accessible for human qualitative analysis.

The earliest computer-aided bias correction techniques relied on phantoms [5]. An object with known intrinsic image (*e.g.*, a water or oil phantom which have uniform intensity in MR images) is placed in the MR machine with the surface coils appropriately mounted. Scans of the phantom and the patient are taken, both received with the surface coil. Because the intrinsic image for the phantom is known, we can get a good estimate of the bias field. We can then use this bias field estimate to correct the scan from the patient. There are a number of issues with this approach. Probably the most serious concern is registration. The bias fields in the two images need to be at the same location in order to provide adequate correction. This method also ignores loading effects which can alter the bias field.

The multiplicative effect of the bias field inspired many people to try homomorphic filtering [56] solutions. If the noise is small compared to the signal level, after we take the log we observe that

$$\tilde{\psi}(\boldsymbol{x}) \approx \tilde{\beta}^*(\boldsymbol{x}) + \tilde{\varphi}^*(\boldsymbol{x}) + \tilde{n}(\boldsymbol{x}) \ . \tag{2.105}$$

The noise $\tilde{n}$ in the log image is related in a complicated manner to β^*, φ^*, and n, the noise in the observed image. We know that β^* is mainly comprised of low-frequency components, and φ has some high-frequency components due to edges. Let h be a kernel for a low-pass filter. Then we can estimate $\log \beta^*$ as

$$\log \hat{\beta}(\boldsymbol{x}) = h(\boldsymbol{x}) * \tilde{\psi}(\boldsymbol{x}) \ . \tag{2.106}$$

The corrected image is then

$$\hat{\varphi}(\boldsymbol{x}) = \frac{\psi(\boldsymbol{x})}{\hat{\beta}(\boldsymbol{x})} = \frac{\psi(\boldsymbol{x})}{\exp\left(h(\boldsymbol{x}) * \tilde{\psi}(\boldsymbol{x})\right)} \ . \tag{2.107}$$

This filtering method can be seen as doing unsharp mask filtering in the log domain and is commonly termed homomorphic unsharp mask (HUM) filtering. If φ^* and β^* are separated in frequency, this method should be fairly effective. Unfortunately this tends to be too optimistic of an assumption for real data, and the results from this method tend to be fairly mediocre. As a slight variation on this, some methods will apply the low-pass filter directly on the observed image:

$$\hat{\varphi}(\boldsymbol{x}) = \frac{\psi(\boldsymbol{x})}{\hat{\beta}(\boldsymbol{x})} = \frac{\psi(\boldsymbol{x})}{h(\boldsymbol{x}) * \psi(\boldsymbol{x})} \ . \tag{2.108}$$

The results from both techniques tend to be similar.

Haselgrove and Prammer [33] were the first to apply these ideas using (2.108). Lufkin *et al.* [48] applied homomorphic filtering using (2.107). Axel *et al.* [5] apply homomorphic filtering and compare the results to those obtained using phantom-based correction. They imaged a bag of saline for the phantom image and a wrist for the test image. They observe that phantom correction provides much better results than homomorphic filtering but is also much more difficult to use in a clinical setting. Gelber *et al.* [25] apply homomorphic filtering to spine images. They have radiologists evaluate the quality of the results, and, while filtering does not produce optimal results, the radiologists thought it was better than no correction at all. Brinkmann *et al.* [14] assert

that homomorphic unsharp filtering is the most prevalent method of bias correction and link to a number of references. They construct brain phantoms by hand segmenting real data and compute RMS error for a number of different choices for h. They find that mean filters with wide kernels is most effective.

One of the many issues that homomorphic filtering has is that it tends to underestimate the bias field at tissue/air boundaries. There is no meaningful information in ψ about the bias field in those regions, and homomorphic filtering uses information in local neighborhoods to make bias estimates. Wald *et al.* [80] attempt to correct this using an edge-completed filter. They use a rudimentary thresholding scheme to discern tissue/air boundaries and fill-in the air regions using nearby tissue intensities. The bias field is then estimated on this modified observation image. Guillemaud [30] uses normalized convolution to estimate the bias field. Let $\tau(\boldsymbol{x})$ be 1 in tissue regions, 0 in air regions (roughly estimated using a threshold). Then Guillemaud estimates β^* as

$$\hat{\beta}(\boldsymbol{x}) = \exp\left(\frac{h(\boldsymbol{x}) * \tilde{\psi}(\boldsymbol{x})}{h(\boldsymbol{x}) * \tau(\boldsymbol{x})}\right) \quad . \tag{2.109}$$

In regions far away from air boundaries, this behaves exactly like regular homomorphic filtering. In regions near air boundaries, the pixels that contain air are ignored and only the pixels in tissue regions are used construct the bias field estimate.

■ 2.3.2 Parametric Methods

A number of techniques attempt to deal with the overwhelming number of degrees of freedom by using parametric estimation techniques. In these methods the bias field is represented as a sum of basis functions, and the bias correction algorithm only modifies the parameters that control the sum. Generally an energy functional is constructed and minimized to determine which parameters provide the best fit. A number of choices are possible. Maintaining fidelity to the data with a ℓ_2 norm is a common choice due to the computational aspects. In general, we would classify parametric techniques as computationally efficient and stable but limited.

Dawant *et al.* [21] propose using a least-squares method to fit basis functions to the bias field. They use thin-plate splines as example basis functions. The final bias estimate is a linear combination of the basis functions with the weights chosen to minimize the squared error at N reference points within one tissue class. The main assumption is that intensity within that tissue class would be constant except for noise and bias.

They detail manual and semi-automatic methods to select the fitted points. The latter method has the user select a few initial points. Then the algorithm uses a neural network classifier to select other points that it believes are in the same class. The parameters are chosen to minimized the ℓ_2 error over the chosen control points.

Styner *et al.* [71] and Brechbühler *et al.* [11] use orthogonal Legendre polynomials [1] as a basis. They operate on the log transform of the image and assume that the true image is piecewise constant. The user must specify the mean and noise variance of each tissue class. They construct an energy functional that punishes at each pixel minimum deviation from any of the class means. The error norm that they use can be viewed as a smoothed ℓ_0 norm in that it equally penalizes all errors above a small threshold. They then use an evolutionary search algorithm to minimize the energy functional over the polynomial weights.

Likar *et al.* [44, 45] and Viola [77] postulate that adding the bias field to the underlying true image increases the entropy (distributions that are "peaky" tend to have lower entropy than smoother PDFs because the observations of this distribution are more constrained). They parameterize the bias field using polynomials and find the parameterization that minimizes the entropy of the reconstructed image.

A review of various correction techniques has been performed by Velthuizen *et al.* [76]. Brain images captured using a head coil were processed using phantom corrections, homomorphic unsharp filtering (with filtering being performed in both the image and log domains), and Dawant's basis function method. The goal was to automatically measure brain tumor volume. Segmentation was performed using a k-nearest neighbor approach (kNN) and results were compared with an expert's segmentations. Surprisingly, the method that produced the highest degree of correlation with the expert results was that of not doing any correction at all. Dawant's technique trailed by a large margin, and the other three techniques produced results similar to that of no correction. The authors conclude that this may not be meaningful for general segmentation problems because tumors tend to be small so the effect of slowly-varying bias fields may not be significant. It is difficult to put much stock in these findings because without ground truth, much of this analysis is subjective.

■ 2.3.3 Non-Parametric Methods

More recently, a number of non-parametric techniques have been proposed which hope to alleviate the difficulties in using parameterized representations of the bias field. Wang

et al. [81] address the issue of inter-scan inhomogeneities. MR images tend to be very sensitive to environmental factors, and the actual intensities observed in an image can vary from day to day even when using the same coil and scanner. They correct for this by aligning the histograms from different scans.

Sled *et al.* [68] apply a log transform to the image which results in the bias becoming additive. They observe that if $\tilde{\varphi}^*(\boldsymbol{x})$ and $\tilde{\beta}^*(\boldsymbol{x})$ are modeled as independent stationary random processes, then (ignoring the noise) the PDF of $\tilde{\psi}(\boldsymbol{x})$ is equal to the convolution of the PDFs of $\tilde{\varphi}^*(\boldsymbol{x})$ and $\tilde{\beta}^*(\boldsymbol{x})$:

$$\mathrm{p}_{\tilde{\psi}}(\tilde{\psi}) = \mathrm{p}_{\tilde{\beta}^*}(\tilde{\beta}^*) * \mathrm{p}_{\tilde{\varphi}^*}(\tilde{\varphi}^*) \ . \tag{2.110}$$

They approximate $\mathrm{p}_{\tilde{\beta}^*}(\tilde{\beta}^*)$ as Gaussian. They then take an iterative approach where they alternately deconvolve the PDF of the corrected image with a small Gaussian and then use that PDF to do Bayes least-squares estimation of $\tilde{\varphi}^*(\boldsymbol{x})$.

Lai and Fang [42] operate on the log transform of the observed image and model the intrinsic image as piecewise constant. Then, except at discontinuities in φ^*, they observe that

$$\nabla\tilde{\psi}(\boldsymbol{x}) \approx \nabla\tilde{\beta}^*(\boldsymbol{x}) \ . \tag{2.111}$$

In 2D, this leads to two linear equations for each pixel not on an edge, and the resulting linear system can be solved to estimate $\tilde{\beta}^*$. Regularization is needed to minimize the effect of the noise as well as to interpolate across edge boundaries and into regions where little signal information is available (*e.g.*, air-filled regions).

Vokurka *et al.* [79] have techniques to deal with inter-slice and intra-slice inhomogeneities. The inter-slice correction uses quadratic functions to parameterize the bias. The intra-slice correction is a technique similar to Lai and Fang. The gradient of the bias field is estimated from the gradient of the image in a more complex manner (*i.e.*, not strict equality), and then the bias field estimate is constructed by integrating.

■ 2.3.4 Simultaneous Segmentation and Bias Correction

Many approaches recognize the duality behind the segmentation and bias correction problems. If perfect bias correction is available, segmentation becomes much easier. If perfect tissue segmentation is available, then bias correction is simple. The methods in this section alternate back and forth between bias correction and segmentation steps with the result from one step helping to improve the results from the other.

Meyer *et al.* [50] discuss a technique that uses the Liou-Chiu-Jain (LCJ) [47] segmentation algorithm. The LCJ algorithm is edge-based and uses gradient values to determine boundaries. An initial coarse segmentation is performed with relatively high thresholds. The algorithm then attempts to fit a polynomial to each region in a least-squares manner. If the error residual is too high, it lowers the threshold in that region and subdivides it. Otherwise that region becomes part of the final segmentation. This method produces a segmentation, and the polynomial error functions can be considered as the bias field estimate. This approach has a number of shortcomings. If the true image is not piecewise constant, the bias field estimate will include variation from the tissue texture. Also, it seems that the bias field will have discontinuities at tissue boundaries because the algorithm constructs several local bias field estimates rather than one global estimate.

Lee and Vannier [43] also use an existing algorithm that allows for piecewise smooth segmented regions. They use an adaptive fuzzy k-means statistical classifier to segment T_1-*weighted* brain images (looking for regions of either gray or white matter). Traditional k-means clustering finds a mean value for each cluster. The authors make it adaptive by defining local means that vary across a segment. The information from the local means are propagated throughout the image with a low-pass filtering operation.

Wells *et al.* [85] use a statistical approach to estimate the bias field. They use the expectation-maximization (EM) algorithm [22] to alternately optimize the segmentation map and the bias field estimate on brain images. The bias field is modeled as a Gaussian random vector, and the true image is modeled as piecewise-constant plus Gaussian noise. This method requires initial probability density functions on all of the tissue classes in the image as well as an initial bias field estimate. The E-step calculates the posterior class probabilities, and the M-step computes the MAP estimate of the bias field for a given set of tissue probabilities.

A number of techniques have been implemented to overcome shortcomings of the method of Wells *et al.* Guillemaud and Brady [31] propose some extensions to better deal with outliers and to provide better parameter initializations. In addition to a class for each tissue, they introduce an "other" class that is uniformly distributed. This prevents structure that is not explicitly modeled from corrupting the tissue class probabilities. They also use an initialization that is similar to that of Likar *et al.* in that they minimize the entropy for a parameterized representation of the bias field. Zhang *et al.* [89] improve on the Gaussian assumption of Wells *et al.* They model the observed

image as a Hidden Markov Model (HMM) with an underlying Markov Random Field (MRF) that spatially couples the pixel probabilities.

The main issue with the EM-based segmenters is that a statistical classifier must be able to segment the data in question. This is not practical for many applications such as prostate segmentation where the intensity distributions inside and outside of the object do not differ very much. Initialization and parameter selection is also an issue. All the methods require relatively accurate prior models on the tissue classes. Despite these difficulties, the key idea to take away from this section is that segmentation and bias correction need not be independent tasks.

■ 2.3.5 Non-Retrospective Techniques

A number of techniques have been proposed that we will term non-retrospective. These methods require modifications to the actual imaging procedure and cannot be applied to previously acquired MR images. A few techniques attempt to modify the imaging coil. Foo *et al.* [24] propose using a special dielectric in the coil to reduce the bias effect. Singh and NessAiver [67] use a special endorectal coil that contains an embedded tube filled with oil. The oil shows up as strong intensities in the observed image, and these marker points can then be used to pinpoint the exact location of the coil. This method is not always applicable because the coil may not be in the FOV. Once the coil location is known, the sensitivity profile can be computed using the Biot-Savart Law [17]. Moyher *et al.* [52, 53] apply a similar technique for brain imaging.

Other researchers capture additional scans beyond those required by the imaging protocol. Liney *et al.* [46] propose a technique specific to the prostate. They claim that capturing a proton density scan will give a good estimate of the bias field in the pelvic region. Everything near the rectum is just soft tissue, so the hope is that the density will be approximately homogeneous. Then the proton density image will behave just like a phantom image to correct the bias.

Brey and Narayana [12] suggest capturing a low SNR image using the body coil and capturing a high SNR image using the surface coil. The surface coil image ψ_S has low noise near the coil but has a very pronounced coil artifact. The body coil image ψ_B is essentially bias free[12] but has a large amount of noise. We can minimize the effect of

[12]We will discuss this assumption more in depth in Section 3.1.1.

the noise by low-pass filtering both of the images by a kernel h:

$$h(\boldsymbol{x}) * \psi_{\mathrm{S}}(\boldsymbol{x}) \approx h(\boldsymbol{x}) * (\beta^*(\boldsymbol{x})\varphi^*(\boldsymbol{x})) \tag{2.112}$$

$$h(\boldsymbol{x}) * \psi_{\mathrm{B}}(\boldsymbol{x}) \approx h(\boldsymbol{x}) * \varphi^*(\boldsymbol{x}) \ . \tag{2.113}$$

The bias field is then estimated as the ratio of the filtered surface coil image to the filtered body coil image:

$$\hat{\beta}(\boldsymbol{x}) = \frac{h(\boldsymbol{x}) * \psi_{\mathrm{S}}(\boldsymbol{x})}{h(\boldsymbol{x}) * \psi_{\mathrm{B}}(\boldsymbol{x})} \approx \frac{h(\boldsymbol{x}) * (\beta^*(\boldsymbol{x})\varphi^*(\boldsymbol{x}))}{h(\boldsymbol{x}) * \varphi^*(\boldsymbol{x})} \ . \tag{2.114}$$

Thus we see that if filtering by $h(\boldsymbol{x})$ does not disturb $\beta^*(\boldsymbol{x})\varphi^*(\boldsymbol{x})$ and $\varphi^*(\boldsymbol{x})$ very much, $\hat{\beta}(\boldsymbol{x})$ will be a very good approximation to $\beta^*(\boldsymbol{x})$.

Lai and Fang [41] take a more sophisticated approach. They take the ratio of the surface coil image to the body coil image and select a number of control points which they believe accurately represent the bias field. They then fit a thin membrane model to those points to interpolate over the rest of the image. Pruessmann *et al.* [59] take a similar approach except they fit a local polynomial at every point in the image. Both of these methods have the effect of producing smoother and more accurate sensitivity profiles, but, as we shall see later, the method of Brey and Narayana actually produces very respectable results.

Chapter 3

Bias Correction

THE near-field effect causes MR signal strength to be greatly increased close to the surface coil. This results in a multiplicative bias field which can impair image analysis. Most traditional bias correction techniques attempt to estimate the bias field directly from the MR surface coil image without any other information. In general, the approaches in the literature have major shortcomings. It is difficult to separate the bias field from the underlying structure without significant operator involvement. Our approach is to use side information to help us correct for the bias in a fully automated fashion.

Our method corrects the intensity inhomogeneities present in MR surface coil images by using simultaneous or near-simultaneous capture of body and surface coil images. We start from the basic imaging framework used by Brey and Narayana [12]. We have access to two images that are functions of the true underlying signal. One is corrupted by a significant bias field but has little noise, and the other is corrupted by heavy noise but has no bias field. We wish to recover estimates $\hat{\varphi}$ and $\hat{\beta}$ of the intrinsic image φ^* and the bias field β^* such that $\hat{\varphi}$ is compatible with the observed intensities in the body coil image; the product of $\hat{\beta}$ and $\hat{\varphi}$ is compatible with the observed surface coil image; and $\hat{\varphi}$ and $\hat{\beta}$ satisfy predetermined smoothness constraints. Our approach differs from Brey and Narayana in that we attack the problem from a first-principles variational approach. This allows us to not only obtain reasonable-looking estimates, but also allows us to understand the nature of the errors we make in the estimation process. Our framework also allows us to generalize to problems that cannot be easily handled using Brey and Narayana's technique.

3.1 Observation Model

In this section, we will formulate a model for the imaging process that we observe. Our imaging processing at the moment employs near-simultaneous image acquisition. Both acquisitions use the body coil to transmit the RF pulse sequence. Then we use the body coil to receive the signal in one image, and we use the surface coil to capture the other image.

It is possible to simultaneously capture images using both the body coil and the surface coil, but we would not get the results that we desire. When multiple coils are active, the magnetic fields that they receive are determined not just by the self inductance of the coil, but also the mutual inductance between all of the coils [17]. There has been work done to receive multiple surface coil images simultaneously, but these coil arrays need to be carefully crafted to minimize mutual induction between the coils [62]. It is not possible to uncouple the body coil from the other coils (which are fully contained within the body coil). Hence if we tried to simultaneously acquire data from the body coil and the surface coils, the image received by the body coil will be partially determined by the surface coil and vice versa. This is clearly not useful because we never have access to an image that is not influenced by the surface coil.

This need to capture the images sequentially poses a few problems. The reconstructed image will be more prone to motion artifacts because we are basing the image off of a time interval that is twice as long as would be the case if we only had a surface coil capture. It may also be possible for the patient to shift slightly between scans. This could cause misregistration between the two images. Also the longer imaging time increases patient discomfort and imaging costs.

In both images, the same imaging pulse sequence is used—the only difference is which coils are activated to receive. It is possible to vary the number of excitations (NEX) in the different scans. Thus we can capture a normal surface coil image and a body coil image with fewer excitations. This will make the body coil image noisier but will reduce the image acquisition time. The coil that is inactive must have its receiving circuit be open so as to not interfere with the other coil. The patient is not moved nor is anything else in the imaging FOV. This means that the surface coils are still present when we capture the body coil images. This is very important in applications such as prostate imaging. The endorectal coil that is used in prostate imaging is mounted inside a balloon that is inflated and inserted into the rectum. The balloon will actually

change the shape of the rectum and surrounding tissue. In order for the body coil image and surface coil image to be properly registered, it is important to leave the coil in the body.

3.1.1 Signal model

We recall that the intrinsic MR signal can be written in terms of operator-determined parameters (T_E,T_R) and tissue-dependent parameters (ρ, T_1, T_2) (see (2.90) on Page 52):

$$\varphi^*(\boldsymbol{x}) = \rho(\boldsymbol{x})e^{-T_E/T_2(\boldsymbol{x})}(1 - e^{-T_R/T_1(\boldsymbol{x})}) \ . \tag{3.1}$$

We will refer to this signal as the true signal or, to borrow a term from the computer vision literature, the intrinsic signal. The observed signal is then the product of the intrinsic signal with the receiving coil's sensitivity profile plus additive noise.

We assume that the sensitivity profile for the body coil is largely homogeneous in the ROI. Because the body coil is also the transmitting coil, its most important characteristic is uniform sensitivity. For something like the prostate where the ROI is on the order of centimeters, this uniformity assumption is probably true. For larger objects such as the brain or spine, this assumption is more questionable. We will show later in this section that not accounting for this effect will simply result in an error equal to the body coil inhomogeneity.

We observe a body coil image ψ_{B} and a surface coil image ψ_{S} and wish to obtain estimates of the true image φ^* and the bias field β^*. We can either view ψ_{S} and ψ_{B} as continuous random processes or discrete random vectors. From a continuous perspective, ψ_{S} and ψ_{B} are both maps from $\Omega \rightarrow \mathbb{R}$ (where $\Omega \subset \mathbb{R}^d$ and d is the dimensionality of the image):

$$\psi_{\mathrm{B}}(\boldsymbol{x}) = k\varphi^*(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) \tag{3.2a}$$

$$\psi_{\mathrm{S}}(\boldsymbol{x}) = \beta^*(\boldsymbol{x})\varphi^*(\boldsymbol{x}) + n_{\mathrm{S}}(\boldsymbol{x}) \tag{3.2b}$$

$$\boldsymbol{x} \in \Omega$$

where k is an arbitrary scale factor and n_{B} and n_{S} are white noise. We then wish to estimate the continuous functions φ^* and β^*. For a 2D image, Ω is generally a square but can be rectangular. We will term its dimensions as $N \times M$ (width and height). For a 3D volume, we will refer to the dimensions as $N \times M \times O$.

Note that if we wished to accurately model our observations, we should have the following relationship for the body coil image:

$$\psi_{\mathrm{B}}(\boldsymbol{x}) = \beta_{\mathrm{B}}(\boldsymbol{x})\varphi^*(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) \tag{3.3}$$

where β_B is the sensitivity profile of the body coil. When the sensitivity is perfectly homogeneous, this simply reduces to (3.2a). Otherwise we can make the following transformation to see the result of neglecting this effect:

$$\begin{aligned}
\tilde{\varphi}(\boldsymbol{x}) &= \beta_{\mathrm{B}}(\boldsymbol{x})\varphi^*(\boldsymbol{x}) && (3.4\mathrm{a})\\
\tilde{\beta}(\boldsymbol{x}) &= \beta^*(\boldsymbol{x})/\beta_{\mathrm{B}}(\boldsymbol{x}) && (3.4\mathrm{b})\\
\psi_{\mathrm{B}}(\boldsymbol{x}) &= \tilde{\varphi}(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) && (3.4\mathrm{c})\\
\psi_{\mathrm{S}}(\boldsymbol{x}) &= \tilde{\beta}(\boldsymbol{x})\tilde{\varphi}(\boldsymbol{x}) + n_{\mathrm{S}}(\boldsymbol{x}) \ . && (3.4\mathrm{d})
\end{aligned}$$

So we replace the estimation of φ^* and β^* with the estimation of $\tilde{\varphi}$ and $\tilde{\beta}$. The inhomogeneity from the body coil remains in our image estimate. But this inhomogeneity is much less pronounced than the bias field from the surface coil, so at worst we replace the large inhomogeneity with a much smaller one. Note that because both β_{B} and β^* are smooth, $\tilde{\beta}$ is also smooth (for instance, if both β_{B} and β^* are differentiable, then $\tilde{\beta}$ is also differentiable at all points where $\beta_{\mathrm{B}} \neq 0$).

From a discrete perspective, we can sample our continuous observations on a grid with spacing 1. This results in a matrix size $M \times N$. Generally it is easier to stack the columns of the matrix into a vector. To do this, we form sampling vectors $\boldsymbol{i}$ and $\boldsymbol{j}$:

$$\begin{aligned}
\boldsymbol{i}[n] &= \left\lfloor \frac{n}{N} \right\rfloor && (3.5\mathrm{a})\\
\boldsymbol{j}[n] &= n \bmod N && (3.5\mathrm{b})\\
& \quad n \in \{0, 1, \ldots, MN-1\} \ .
\end{aligned}$$

We then form observation vectors $\boldsymbol{y}_{\mathrm{S}}$ and $\boldsymbol{y}_{\mathrm{B}}$ and parameter vectors $\boldsymbol{b}^*$ and $\boldsymbol{f}^*$:

$$\begin{aligned}
\boldsymbol{y}_{\mathrm{B}}[n] &= \psi_{\mathrm{B}}(\boldsymbol{i}[n], \boldsymbol{j}[n]) && (3.6\mathrm{a})\\
\boldsymbol{y}_{\mathrm{S}}[n] &= \psi_{\mathrm{S}}(\boldsymbol{i}[n], \boldsymbol{j}[n]) && (3.6\mathrm{b})\\
\boldsymbol{b}[n] &= \beta(\boldsymbol{i}[n], \boldsymbol{j}[n]) && (3.6\mathrm{c})\\
\boldsymbol{f}[n] &= \varphi(\boldsymbol{i}[n], \boldsymbol{j}[n]) \ . && (3.6\mathrm{d})
\end{aligned}$$

We introduce two diagonal matrices $\boldsymbol{B}^*$ and $\boldsymbol{F}^*$ which have $\boldsymbol{b}^*$ and $\boldsymbol{f}^*$ respectively as their diagonal entries. We can then formulate our observation model as

$$\boldsymbol{y}_{\mathrm{B}} = k\boldsymbol{f}^* + \boldsymbol{n}_{\mathrm{B}} \tag{3.7a}$$

$$\boldsymbol{y}_{\mathrm{S}} = \boldsymbol{B}^*\boldsymbol{f}^* + \boldsymbol{n}_{\mathrm{S}} = \boldsymbol{F}^*\boldsymbol{b}^* + \boldsymbol{n}_{\mathrm{S}} = \boldsymbol{b}^* \circ \boldsymbol{f}^* + \boldsymbol{n}_{\mathrm{S}} \tag{3.7b}$$

where $\circ$ is the Hadamard entrywise product.

3.1.2 Noise modeling

As mentioned in Section 2.2.2, MR images are the absolute value of a signal corrupted by complex Gaussian noise. This results in the signal in the final image being Rician. The signal and the noise cannot be readily separated here in a linear manner, but we will establish some terminology here to allow some differentiation. Let us have a vector x of n non-zero mean Gaussian random variables with the same variance σ^2. Then $\mathsf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. We can then define a Rician random variable $r = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$. We refer to $A^2 = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}$ as the signal power, $n\sigma^2$ as the noise power, and $P = A^2 + n\sigma^2$ as the total power. We can separate r into its deterministic and random components:

$$\mathsf{r} = A + \tilde{\mathsf{r}} \ . \tag{3.8}$$

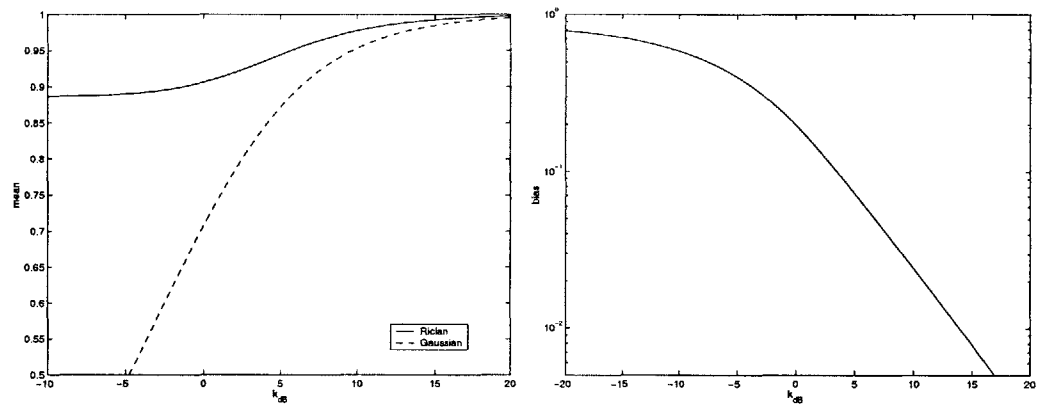
The PDF of $\tilde{r}$ is just the PDF of r shifted by A. The noise terms that we defined earlier (n_B and n_S) correspond to $\tilde{\mathsf{r}}$. We will refer to the mean of $\tilde{r}$ as the noise bias and the variance as the noise variance.

Rician noise has a few aspects that makes it troublesome to deal with. The bias is always positive, and the bias and variance depend upon the SNR at each voxel. This makes it difficult to create unbiased estimators. The Rician PDF is unwieldy to work with which makes it difficult to generate analytical results. For these reasons we treat the noise as Gaussian and zero-mean in our algorithm. In high-SNR regions, Rician noise is well approximated by a Gaussian random variable and non-zero bias, and the bias asymptotically approaches zero. We examine two special limiting cases. When $A^2 = 0$ then r is Rayleigh distributed with the following statistics:

$$\mathrm{E}\left[\mathsf{r}\right] = \sigma\sqrt{\frac{\pi}{2}} \tag{3.9}$$

$$\mathrm{E}\left[\mathsf{r}^2\right] = 2\sigma^2 \tag{3.10}$$

$$\sigma_{\mathsf{r}}^2 = (2 - \tfrac{\pi}{2})\sigma^2 \tag{3.11}$$



(a) (b)

Figure 3.1. Plots versus k value of the (a) means of Gaussian and Rician random variables for a fixed power level of $P = 1$ and (b) the bias (difference in the two means). The x-axis in both plots is in dB, so (b) is a log-log plot.

Normalized bias	10%	3%	1%	0.3%
k_{dB} value	4.4 dB	9.4 dB	14.2 dB	19.4 dB

Table 3.1. k_{dB} value for half-decade intervals of normalized bias values

When $A \gg \sigma$, then r is approximately Gaussian with the following statistics:

$$\mathrm{E}[\mathsf{r}] \approx A + \frac{\sigma^2}{2A} \tag{3.12}$$

$$\mathrm{E}[\mathsf{r}^2] = A^2 + 2\sigma^2 \tag{3.13}$$

$$\sigma_{\mathsf{r}}^2 \approx \sigma^2 \tag{3.14}$$

Thus we see at one extreme when there is no signal, the bias is very large. At the other extreme when the SNR is high, then the bias asymptotically approaches 0 while the variance approaches σ^2.

In Figure 3.1(a) we depict how $\mathrm{E}[\mathsf{r}]$ and A vary with k for a fixed total power level. We can see that as the signal level approaches zero, the mean of the Rician observation plateaus at a very high level. Figure 3.1(b) plots the Rician bias which is the difference between those two curves. At the right end of the curve (which is a log-log plot), we see the bias decreasing almost linearly which reflects the fact that the bias at high SNR is approximately $1/4k$. Table 3.1 lists a number of SNR levels needed in order

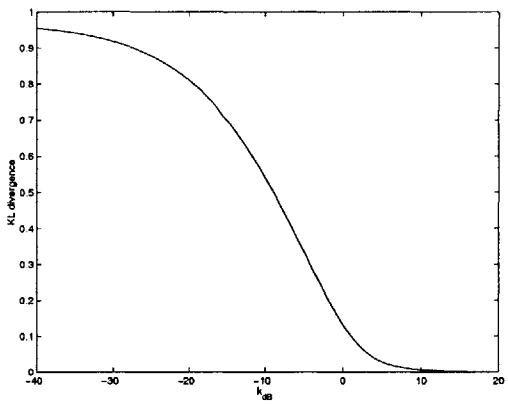


Figure 3.2. Plot versus k value of the Kullback-Leibler divergence between the Rician PDF and the underlying Gaussian PDF for a fixed power level of $P = 1$.

to guarantee normalized bias values below a certain threshold. We define normalized bias as the bias divided by the true signal level A. So even for reasonable SNR levels, a moderate upward bias is introduced. This behavior can be irritating but is not a huge concern in many applications. Often the actual value of the intrinsic image is not of importance. The various tissues just need to appear different enough so we can readily discern tissue boundaries. This upwards bias from the noise does not affect this behavior.

The Kullback-Leibler (KL) divergence [20] can be viewed as a quasi-distance[1] measure between two PDFs:

$$\mathrm{D}(\mathrm{p}_1||\mathrm{p}_2) = \mathrm{E}_{\mathrm{p}_1}\left[\log \frac{\mathrm{p}_1}{\mathrm{p}_2}\right] \quad . \tag{3.15}$$

In Figure 3.2, we vary k and plot the KL divergence between a Rician PDF and its underlying Gaussian PDF ($\mathcal{N}(A, \sigma^2)$) for a fixed $P = 1$. This distance measure is affected both by the bias and the change in the shape of the PDF. We can see that the KL divergence becomes quite small once we achieve SNR levels in the 10 dB range.

For surface coil images, the signal level near the coil is very high so the observed noise tends to behave similarly to our Gaussian noise assumption. It is convenient that the region near the coil is usually the part of greatest interest to us. It is difficult to state any certainties about the behavior of the noise for body coil images because so

[1]We say quasi-distance because the KL divergence is not symmetric nor does it satisfy the triangle inequality. Nonetheless, it does tend to behave as a monotonic map of a true distance function.

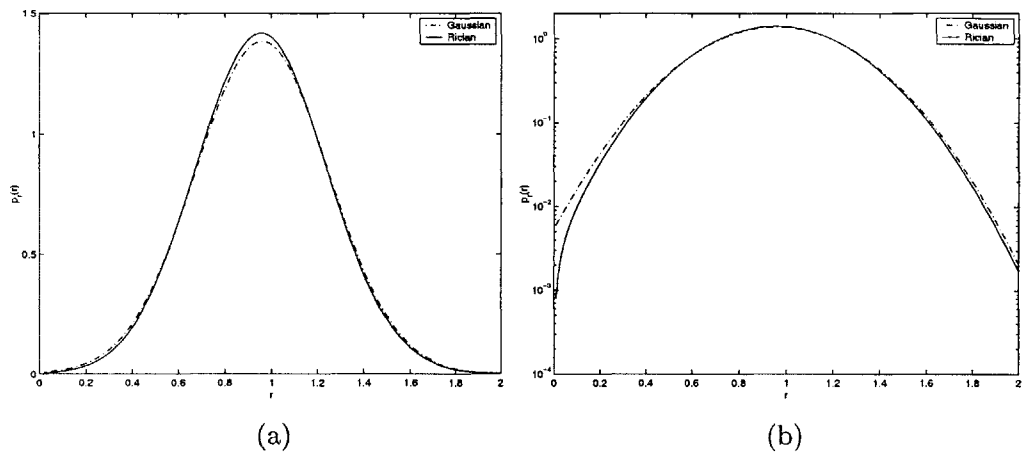


Figure 3.3. (a) Plot of Rician PDF with $k_{\mathrm{dB}} = 7$ and Gaussian PDF with same mean and variance$=\frac{1}{2(k+1)}$. (b) Plot of log probabilities.

much can change from application to application. For the prostate, we have a SNR of approximately 7 dB inside the gland for T_2-*weighted* images. At this level, the noise is fairly close to a Gaussian in shape, but there is a definite bias. For the heart and brain images we present, the body coil images have SNR of about 20 dB. The bias at these levels is quite small, and the noise is well approximated by a Gaussian.

In Figure 3.3(a) we plot a Rician PDF with parameter $k = 7$dB. Note that this is a normalized Rician ($P = 1$) so the x-axis really represents $r/\sqrt{P}$. We simultaneously plot a Gaussian with the same mean[2] and variance equal to $1/2(k+1)$. This is the variance of the underlying Gaussian process that generated the Rician process. We can see that even at 7 dB, the two behave similarly except the Gaussian puts a slightly higher weight in the middle and the Rician has heavier tails. Figure 3.3(b) plots the log probabilities of the two processes to emphasize the differences at the tails. These plots do not include the bias introduced by the Rician random variable. For $k = 7$dB, the signal level is $A = 0.9131$. The mean of the Rician random variable is 0.9600 which results in a normalized bias of 5.1%. We can conclude that at 7 dB, using a squared-error penalty (which a Gaussian noise assumption induces) is reasonable, but our estimates will have a fairly significant upward bias.

[2]We omit the bias in this graph to show how close the overall shape of the Rician PDF and a Gaussian PDF are at this SNR level.

The most pronounced effect from the wrong noise model will occur in extremely low SNR regions such as air-filled regions. This is where the KL divergence between our assumed noise distribution and our actual noise distribution is at a maximum. In air-filled regions, ignoring other signal distortions, the true value of the MR signal should be uniformly zero. It is almost impossible for our imaging model to produce proper estimates in this regime. Luckily, in most medical imaging applications, we do not really care about the intensity values in air-filled regions. It is important that we estimate low signal values there, but the actual number is unimportant.

3.2 Problem Formulation

Minimizing the noise and eliminating the bias field are two interrelated problems. If noise was not an issue, we would simply use the body coil to capture images, and there would not be a bias field. So in some sense, there is a trade-off we must make between intensity homogeneity and SNR. Looking at equivalent body coil and surface coil images (equivalent in the sense of imaging parameters), it is clear that once rudimentary bias correction techniques are applied (such as window/level), surface coil images are much more informative than body coil images in regions in close proximity to the coil. Due to the many different types of surface coil configurations, image quality away from the coil can range from terrible to moderate.

With our algorithm, we want to be able to get the best of both worlds: high SNR and low intensity inhomogeneity. We use the body coil image to obtain intensity homogeneity, and we combine both the body coil and the surface coil images along with non-linear filtering techniques to provide superior noise properties. Given our data observation model derived in the previous section, we can now formulate a method to estimate the intrinsic image and the bias field using a statistical approach. With appropriate regularization, this results in a bias correction algorithm that is non-parametric, fully automatic, and robust. We begin by presenting our algorithm and discussing its main features. We then use our observation model to show the statistical basis for our variational formulation.

3.2.1 Variational Formulation

Our method is based on the following observations:

1. The body coil images are generally bias-free but noisy.

2. The surface coil images have high SNR in the ROI but have strong intensity inhomogeneities.

3. The bias field is a relatively slowly-varying function of space.

4. The bias field is only a function of the magnetic field profile of the surface coil and is thus independent of tissue class.

Let φ^* be the intrinsic image we wish to recover and β^* be the actual intensity profile of the surface coil. We use $\boldsymbol{f}^*$ and $\boldsymbol{b}^*$ to indicate the discrete counterparts. We pose the estimation of φ^* and β^* as an optimization problem. We define an energy functional:

$$E(\varphi, \beta) = E_{\mathrm{B}}(\varphi) + \lambda E_{\mathrm{S}}(\varphi, \beta) + \alpha \mathcal{R}_{\beta}(\beta) + \gamma \mathcal{R}_{\varphi}(\varphi) \tag{3.16}$$

where λ, γ, and α are positive weights. In a statistical framework, the optimal choice of λ is related to the noise variances of ψ_{S} and ψ_{B}. E_{B} and E_{S} are data fidelity terms for ψ_{B} and ψ_{S} respectively. $\mathcal{R}_{\varphi}$ and $\mathcal{R}_{\beta}$ are regularization terms designed to impose smoothness or piecewise-smoothness on $\hat{\varphi}$ and $\hat{\beta}$ respectively.

We choose our optimal $\hat{\varphi}$ and $\hat{\beta}$ as the functions that minimize $E(\varphi, \beta)$:

$$\hat{\varphi}, \hat{\beta} = \arg\min_{\varphi, \beta} E(\varphi, \beta) \ . \tag{3.17}$$

If our energy functional is specified correctly, then $\hat{\varphi} \approx \varphi^*$ and $\hat{\beta} \approx \beta^*$. This unconstrained optimization can be viewed as the penalty version of a constrained optimization problem:

$$\begin{aligned} \hat{\varphi}, \hat{\beta} \ &= \ \arg\min_{\varphi, \beta} E_{\mathrm{B}}(\varphi) + \lambda E_{\mathrm{S}}(\varphi, \beta) \\ &\quad \text{subject to } \mathcal{R}_{\beta}(\beta) \leq c_1 \\ &\quad \text{and } \mathcal{R}_{\varphi}(\varphi) \leq c_2 \ . \end{aligned} \tag{3.18}$$

The regularization parameters α and γ are then the corresponding Lagrange multipliers that make the constraints true.

We define our data fidelity terms as the $\mathcal{L}_2$ estimation errors:

$$E_{\mathrm{B}}(\varphi) = \int_{\Omega} (\psi_{\mathrm{B}}(\boldsymbol{x}) - k\varphi(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x} \tag{3.19}$$

$$E_{\mathrm{S}}(\varphi, \beta) = \int_{\Omega} (\psi_{\mathrm{S}}(\boldsymbol{x}) - \beta(\boldsymbol{x})\varphi(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x} \ . \tag{3.20}$$

Without the regularizers, the minimization is a well-posed problem (in the sense the solution is attainable and unique), but it produces a trivial result. Without any constraints on $\hat{\varphi}$ or $\hat{\beta}$, we see that the minimum of E is 0 and is achieved for $\hat{\varphi} = \psi_{\mathrm{B}}/k$ and $\hat{\beta} = k\psi_{\mathrm{S}}/\psi_{\mathrm{B}}$.

As discussed in Section 2.1.2, there is a great deal of prior work on regularization for ill-posed problems. We will use Tikhonov regularization (or, more generally, $\mathcal{L}_p$-norms):

$$\mathcal{R}_\varphi(\varphi) = \int_\Omega \|L_\varphi(\varphi(\boldsymbol{x}))\|_p^p \mathrm{d}\boldsymbol{x} \tag{3.21}$$

$$\mathcal{R}_\beta(\beta) = \int_\Omega \|L_\beta(\beta(\boldsymbol{x}))\|_q^q \mathrm{d}\boldsymbol{x} \tag{3.22}$$

where $L_\varphi(\cdot)$ and $L_\beta(\cdot)$ are linear operators, and p and q are positive numbers chosen to enforce desired properties in φ^* and β^*. Derivative operators are common choices to enforce smoothness. Nonlinear operators can be used, but they make the optimization much more difficult.

These energy functionals can also be discretized for $\boldsymbol{f}$, $\boldsymbol{b}$, $\boldsymbol{y}_{\mathrm{S}}$, and $\boldsymbol{y}_{\mathrm{B}}$ by using ℓ_p norms:

$$E_{\mathrm{B}}(\boldsymbol{f}) = \sum_n (\boldsymbol{y}_{\mathrm{B}}[n] - k\boldsymbol{f}[n])^2 \tag{3.23a}$$

$$E_{\mathrm{S}}(\boldsymbol{f}, \boldsymbol{b}) = \sum_n (\boldsymbol{y}_{\mathrm{S}}[n] - \boldsymbol{b}[n]\boldsymbol{f}[n])^2 \tag{3.23b}$$

$$\mathcal{R}_f(\boldsymbol{f}) = \sum_n |\boldsymbol{L}_f(\boldsymbol{f})[n]|^p \tag{3.23c}$$

$$\mathcal{R}_b(\boldsymbol{b}) = \sum_n |\boldsymbol{L}_b(\boldsymbol{b})[n]|^q \ . \tag{3.23d}$$

These can be written equivalently in vector notation:

$$E_{\mathrm{B}}(\boldsymbol{f}) = \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 \tag{3.24a}$$

$$E_{\mathrm{S}}(\boldsymbol{f}, \boldsymbol{b}) = \|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{b} \circ \boldsymbol{f}\|^2 \tag{3.24b}$$

$$\mathcal{R}_f(\boldsymbol{f}) = \|\boldsymbol{L}_f \boldsymbol{f}\|_p^p \tag{3.24c}$$

$$\mathcal{R}_b(\boldsymbol{b}) = \|\boldsymbol{L}_b \boldsymbol{b}\|_q^q \ . \tag{3.24d}$$

For the most part we will work with the discrete formulation. All of our observations are discrete to begin with, so even a continuous solution must be implemented numerically. It is conceptually easier to deal with finite vectors than infinite function spaces. We will also see how our discrete formulation allows us to apply some powerful computational tools.

■ 3.2.2 ML Estimation

For simplicity, we will formulate the results in this section from a discrete perspective. We can rewrite our discrete observation model as

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_{\mathrm{B}} \\ \boldsymbol{y}_{\mathrm{S}} \end{pmatrix} = \boldsymbol{\mu} + \boldsymbol{n}$$

where $\boldsymbol{\mu}$ is the mean vector defined as

$$\boldsymbol{\mu} = \begin{pmatrix} k\boldsymbol{f} \\ \boldsymbol{b} \circ \boldsymbol{f} \end{pmatrix}$$

and $\boldsymbol{n}$ is modeled as zero-mean Gaussian independent and identically distributed (IID) white noise with covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{\mathrm{B}}^2 \boldsymbol{I} & 0 \\ 0 & \sigma_{\mathrm{S}}^2 \boldsymbol{I} \end{pmatrix} .$$

Then the PDF for $\boldsymbol{y}$ is:

$$\mathrm{p}_{\mathbf{y}}(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{b}) = \left((2\pi)^{2MN} |\boldsymbol{\Sigma}|\right)^{-1/2} \exp\left(-\tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu})\right) ,$$

and, ignoring the constant terms, we write the log likelihood as

$$\log \mathrm{p}_{\mathbf{y}}(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{b}) = -\tfrac{1}{2\sigma_{\mathrm{B}}^2} \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 - \tfrac{1}{2\sigma_{s}^2} \|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{b} \circ \boldsymbol{f}\|^2 . \tag{3.25}$$

For ML estimation, we choose our parameter estimates as to maximize the log likelihood:

$$\begin{aligned} \hat{\boldsymbol{f}}, \hat{\boldsymbol{b}} &= \arg\max_{\boldsymbol{f}, \boldsymbol{b}} \log \mathrm{p}_{\mathbf{y}}(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{b}) \\ &= \arg\min_{\boldsymbol{f}, \boldsymbol{b}} \tfrac{1}{\sigma_{\mathrm{B}}^2} \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 + \tfrac{1}{\sigma_{\mathrm{S}}^2} \|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{b} \circ \boldsymbol{f}\|^2 \\ &= \arg\min_{\boldsymbol{f}, \boldsymbol{b}} \sum_n (\boldsymbol{y}_{\mathrm{B}}[n] - k\boldsymbol{f}[n])^2 + \tfrac{\sigma_{\mathrm{B}}^2}{\sigma_{\mathrm{S}}^2} \sum_n (\boldsymbol{y}_{\mathrm{S}}[n] - \boldsymbol{b}[n]\boldsymbol{f}[n])^2 . \end{aligned} \tag{3.26}$$

So we can see minimizing (3.16) with no regularization is identical to performing ML estimation with a Gaussian noise assumption and $\lambda = \sigma_{\mathrm{B}}^2 / \sigma_{\mathrm{S}}^2$.

■ 3.2.3 MAP Estimation

The above formulation did not take advantage of any prior knowledge we have about $\boldsymbol{f}^*$ and $\boldsymbol{b}^*$. For instance, we know that $\boldsymbol{b}^*$ is smooth. We can add spatial priors to $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{b}}$

by assigning them Gaussian probability densities. Let $\boldsymbol{f} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_f)$ and $\boldsymbol{b} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_b)$. The $\boldsymbol{\Sigma}$ matrices are chosen so that the entries appropriately model the local interactions of $\boldsymbol{f}$ and $\boldsymbol{b}$. Note that it does not really make sense to view $\boldsymbol{b}$ and $\boldsymbol{f}$ as Gaussian since both are always non-negative. We will address this concern later in this section.

We apply Bayes' Rule and maximize the posterior probability for $\boldsymbol{f}$ and $\boldsymbol{b}$ conditioned on observing $\boldsymbol{y}$:

$$\begin{aligned}
\hat{\boldsymbol{f}}, \hat{\boldsymbol{b}} &= \arg\max_{\boldsymbol{f},\boldsymbol{b}} \log \mathrm{p}_{\mathbf{f},\mathbf{b}}(\boldsymbol{f}, \boldsymbol{b}|\boldsymbol{y}) \\
&= \arg\max_{\boldsymbol{f},\boldsymbol{b}} \log \left[\mathrm{p}_{\mathbf{y}|\mathbf{f},\mathbf{b}}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{b})\mathrm{p}_{\mathbf{f},\mathbf{b}}(\boldsymbol{f}, \boldsymbol{b})\right] \\
&= \arg\max_{\boldsymbol{f},\boldsymbol{b}} \log \mathrm{p}_{\mathbf{y}|\mathbf{f},\mathbf{b}}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{b}) + \log \mathrm{p}_{\mathbf{f}}(\boldsymbol{f}) + \log \mathrm{p}_{\mathbf{b}}(\boldsymbol{b}) \\
&= \arg\min_{\boldsymbol{f},\boldsymbol{b}} \tfrac{1}{\sigma_{\mathrm{B}}^2}\|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 + \tfrac{1}{\sigma_{\mathrm{S}}^2}\|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{b} \circ \boldsymbol{f}\|^2 + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Sigma}_f^{-1}\boldsymbol{f} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{\Sigma}_b^{-1}\boldsymbol{b} \quad (3.27)
\end{aligned}$$

Note that this is in the same form as (3.24) with $p = q = 2$, $\boldsymbol{L}_b$ and $\boldsymbol{L}_f$ as linear operators (*i.e.*, matrices), $\boldsymbol{\Sigma}_b^{-1} = \boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b$, and $\boldsymbol{\Sigma}_f^{-1} = \boldsymbol{L}_f^{\mathrm{T}}\boldsymbol{L}_f$.

This factorization exists for any valid covariance matrix (see Appendix A). Because the covariance matrices are symmetric and positive definite, their inverses have a Cholesky factorization $\boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ (where $\boldsymbol{L}$ is lower triangular). They can also be factored into the product of more arbitrary matrices by using an orthogonal matrix $\boldsymbol{Q}$ to form $\boldsymbol{L} = \boldsymbol{H}\boldsymbol{Q}$. Then $\boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} = (\boldsymbol{H}\boldsymbol{Q})(\boldsymbol{H}\boldsymbol{Q})^{\mathrm{T}} = \boldsymbol{H}\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$. For an arbitrary covariance matrix, it may be difficult or impossible to find a factorization with an appropriate sparsity structure so that the Cholesky factorization corresponds to a pleasant linear operator (for computational purposes). Thus we generally specify a sparse linear operator that produces an approximation to the behavior of the true covariance matrix.

An alternative view when the regularization terms are ℓ_2 norms is that (3.27) models a distribution, not of $\boldsymbol{b}$, but of a linear function applied to $\boldsymbol{b}$. Then we can view this as $\tilde{\boldsymbol{b}} = \boldsymbol{H}_b\boldsymbol{b}$, and thus we can see that we are assuming that $\tilde{\boldsymbol{b}}$ is Gaussian and IID with a mean of zero and variance of $1/\alpha$. A similar argument can be applied for $\boldsymbol{f}$. This viewpoint can be useful in computing optimal choices for α and γ if we have ground truth.

The ℓ_2 penalty that we apply to $\boldsymbol{f}$ is not ideal due to the presence of strongly defined edges at tissue boundaries. These edges tend to create large derivative values. A typical empirical distribution of some derivative of $\boldsymbol{f}$ would have tails that would be much heavier than would be the case for a Gaussian distribution. Thus, using a Gaussian

distribution overpenalizes large derivative values and has the effect of oversmoothing. This is why it is better to use ℓ_p norms with $p \leq 1$. A ℓ_1 norm applied to $\boldsymbol{L}_f \boldsymbol{f}$ carries an implicit assumption that $\boldsymbol{L}_f \boldsymbol{f}$ is Laplacian distributed and IID with a mean of zero and variance $2/\gamma^2$. This is in some ways related to the work in understanding statistics of natural images [35]. Here Huang and Mumford note that the best ℓ_p norm to use in conjunction with natural image derivatives is $p = 0.55$. We will use $p = 1$ because it preserves convexity in the f-step.

■ 3.2.4 Regularization

We briefly talked about a number of regularization techniques in Section 2.1.2. For our continuous formulation, we use a Tikhonov approach with derivative operators because there are a number of techniques that can be used to obtain stationary points [38, 73]. The gradient operator penalizes any deviation from a constant field. This type of regularization is sometimes referred to as the thin membrane model. Deviation from a constant field is allowed but only when the evidence from the observed images is sufficiently strong. We will show that for our problem, Euler-Lagrange tells us that applying a $\mathcal{L}_2$ gradient regularizer is equivalent to solving a damped Poisson's equation. The Laplacian operator penalizes curvature. It allows linear functions with zero energy penalty. This type of regularization is associated with thin plate splines. Applying a $\mathcal{L}_2$ Laplacian regularizer to our problem ends up being the same as solving a damped inhomogeneous biharmonic equation.

For the discrete formulation, we can approximate derivative operators as finite differences. This means that we can write derivatives as linear combinations of a local neighborhood of each point. Because the linear combination is space invariant, the derivative operator can be expressed as a linear filter and can be implemented using convolution. An equivalent method would be to express it as a matrix (all linear operators can be expressed as a matrix). The exact form of the matrix will vary depending on how we convert our image into an observation vector as well as the nature of our boundary conditions (*e.g.*, periodic, reflected, zero-padded). If we do it in the manner discussed in Section 3.2.1 and simply stack the columns, the resulting matrix will be sparse and banded. For the 1D case, the matrix is Toeplitz; for the 2D and higher case, the matrix is block Toeplitz. Tikhonov regularization allows us to use our MAP interpretation from the previous section. Regularization with ℓ_2 norms are the easiest to handle computationally because they turn into linear terms when we take the gradient

$-\frac{1}{2}$	0	$\frac{1}{2}$

(a)

-1	1	0

(b)

0	-1	1

(c)

Table 3.2. Kernels for the x-derivative operator (dx) for the (a) double-sided case, (b) left-sided case, and (c) right-sided case. The zero offset occurs in the middle of the kernel.

of the energy functional.

The gradient operator is actually composed of two operators (dx and dy) stacked as a vector. In Table 3.2, we list possible kernels to implement an x-derivative approximation. There are three reasonable choices we can make: double-sided, left-sided, and right-sided. None of them are ideal. The double-sided approximation is nice because it is symmetric with respect to the origin. Unfortunately it is wider than it needs to be and does not incorporate the value of the current point in computing the derivative. The even-indexed pixels have derivatives that are independent of values from the even-indexed pixels and similarly for the odd-indexed pixels. So we get the strange situation where we are imposing a smoothness constraint for the odd-indexed pixels and a separate one of the even-indexed pixels, and never the twain shall meet. Hence when using a two-sided approximation, if this is the only spatial coupling that is used, we are really solving two separate optimization problems over the even-indexed and odd-indexed pixels. The single-sided derivatives avoid this problem, but the derivatives are always off by half a pixel due to their asymmetry with respect to zero. For instance, let $f(x) = x^2$. We sample this function at unit intervals. The derivative at zero should be zero. This is the case when using the double-sided approximation. But when using the left-sided approximation, we compute that the derivative is -1, and when using the right-sided approximation, we get a derivative of $+1$. The same issues occur for y-derivatives. The kernels for those operators are the same as the x-derivative case, except they are oriented vertically.

Table 3.3 contains the kernel for a Laplacian operator. Because the width of a second derivative operator can be three, the symmetry issues that we experienced in the first derivative case no longer exist. The biharmonic operator in Table 3.4 also has a nice symmetry property. Our discrete formulation allows arbitrary linear convolutional kernels to be used instead of the derivative operators we have discussed here. In fact the linear operator does not need to be convolutional. There are no computational

0	1	0
1	-4	1
0	1	0

Table 3.3. Kernel for a 2D Laplacian (∇^2) operator. The origin is located in the middle of the kernel.

0	0	1	0	0
0	2	-8	2	0
1	-8	20	-8	1
0	2	-8	2	0
0	0	1	0	0

Table 3.4. Kernel for a 2D biharmonic (∇^4) operator. The origin is located in the middle of the kernel.

constraints on our choice of linear operators, but to have it enforce smoothness, the frequency response should be high-pass in nature.

By using these regularization techniques, we will usually end up biasing our answer from the true answer. This is in general impossible to avoid because we will not know *a priori* the correct model for either the bias field or the intrinsic image. All we can really hope to do is to pick a model that is computationally tractable and approximates the true distributions as closely as possible.

■ 3.3 Solutions

This section details the solution to the optimization problem defined in Section 3.2.1. As detailed earlier, the solution to the energy functional with no regularization term can be easily found in closed form, but it is not useful. If we formulate the problem as continuous estimation, we must use the calculus of variations to find a stationary point. If we formulate the problem as discrete estimation, we find that we can write the necessary conditions for the existence of a minimum in terms of a pair of coupled sparse linear systems. A closed-form solution for simultaneously obtaining both $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{b}}$ is not possible, so we are forced to turn to an iterative scheme to find a solution.

In the algorithms we will present, we always minimize our energy functional using coordinate descent. This is a technique that can be used to optimize a function of

several variables. It is useful in problems where computing solutions over all of the variables is difficult, but computing solutions over a subset of the variables is easy. The idea is to successively iterate over different subsets of the variables (making sure that each variable is in at least one subset) and hope that the result converges. It is easy to create special cases where this technique will break, but in practice this is a well-behaved method. In our problem, solving for either $\hat{\varphi}$ or $\hat{\beta}$ individually is easy. Finding the minimum simultaneously over both is not as clean. We will refer to a sub-iteration where we update $\hat{\varphi}$ as an f-step and a sub-iteration where we update $\hat{\beta}$ as a b-step.

For our discrete iterative methods, we use $\hat{\boldsymbol{b}}^{(k)}$ to indicate our estimate of $\boldsymbol{b}^*$ at iteration k. We indicate our initialization using $\hat{\boldsymbol{b}}^{(0)}$. Similar notation holds for $\hat{\boldsymbol{f}}^{(k)}$ and other parameters that arise in the iterations. For the continuous iterative methods, we use the notation $\hat{\beta}^{(t)}$ to indicate our estimate of β at time t. Time begins at $t = 0$, and we set $\hat{\beta}^{(0)}$ to our initialization (initial conditions for the PDE).

■ 3.3.1 Parameter Estimation

There are a large number of parameters that need to be set in (3.16): λ, α, γ, p, q, and k. Selection of the proper ℓ_p and ℓ_q norms is probably the most fundamental decision because they directly affect the type of features we are looking for, not just how strongly we look for them. The choice of p and q should be a conscious decision in the modeling process.

Through experimentation, we have found that our solutions are relatively resistant to incorrect selection of the weighting parameters. Being off by a factor of 2 in one of the parameters may result in a suboptimal image, but the method does not have convergence issues and the result still looks reasonable. With real data, there is no such thing as a "right" answer, so we can tune the parameters to highlight features that we want in the resulting output. Incorrect specification of λ will result in higher noise variance in the estimated image. Small errors in λ generally will not affect $\hat{\beta}$ because the regularization term allows us to smooth over the extra noise. Incorrect specification of α or γ are fairly decoupled. Changing α will have a fairly dramatic effect on $\hat{\beta}$ but does not impact $\hat{\varphi}$ much and vice versa.

As shown in Section 3.2.2, if we wish to formulate things from a statistical viewpoint, we should set the weighting parameter between the two data fidelity terms as $\lambda = \sigma_{\mathrm{B}}^2/\sigma_{\mathrm{S}}^2$. We can see that this choice penalizes errors in high variance measurements less than in low variance measurements. In Section 3.3.4 we show how this term becomes a

weighting factor when combining $\boldsymbol{y}_{\mathrm{B}}$ and $\boldsymbol{y}_{\mathrm{S}}$ into $\hat{\boldsymbol{f}}$.

In order to properly estimate σ_{B} and σ_{S}, we use the observation from Nowak [55] that the true signal should be uniformly zero in air-filled regions. Thus the variance of the underlying complex Gaussian process can be estimated from the second moment of the observed Rayleigh noise in air-filled regions. Because second-order statistics of Rician random variables are easy to calculate in general, this technique is not necessarily restricted to air-filled regions. But it is difficult to find other large regions that we can guarantee to have homogenous intensity values.

Air-filled regions are present in nearly all imaging applications. Most scans (*e.g.*, brain, breast, and spine) include the body part centered in the frame with air surrounding the outside of the body. Others (*e.g.*, prostate and rectum) have an air cavity inside the body adjacent to the object of interest. In fact, if no air-filled regions were present in the image, there would not be any place for the surface coil to go. In practice, these air-filled regions are not homogeneous due to ghosting that results from patient movement and other artifacts. This adversely affects our noise estimates but in a manner that is difficult to quantify. If many scans are taken using the exact same imaging protocol, the noise parameters should be the identical across the data sets. We can then obtain more reliable estimates of the noise parameters by averaging our estimates from each separate data set.

We recall that the second moment of a Rician random variable r was derived in Section 2.2.2 (and repeated here for convenience):

$$\mathrm{E}\left[\mathrm{r}^2\right] = A^2 + 2\sigma^2 \ . \tag{3.28}$$

A^2 is the power of the noiseless signal, and σ^2 is the variance of each Gaussian noise component. We define a subset $\mathcal{A}$=$\{i|i$ an air-filled location in space$\}$. $A^2 = 0$ in air-filled regions, and we get the following estimate of σ_{B}^2 for body coil images:

$$\sigma_{\mathrm{B}}^2 = \frac{1}{2|\mathcal{A}|} \sum_{i \in \mathcal{A}} \boldsymbol{y}_{\mathrm{B}}[i]^2 \ . \tag{3.29}$$

Note that because we do this in a region with zero intensity, the bias field has no effect, and we can also perform this technique on surface coil images.

The proper choice of k directly affects the proper choice of α and γ. Generally this k value will arise in situations where the overall ψ_{B} and ψ_{S} images have been scaled by different factors. The actual scale factors do not matter because we do not care if the

final image is scaled by some unknown number. All that we are concerned with are the relative scale factors. We can thus arbitrarily fix the scale factor for ψ_{S} at 1 and call k the scale factor for ψ_{B}.

Let ψ_{B}^* be the body coil image if $k = 1$ and ψ_{B} be the body coil image we actually observe. Then we get the following equation:

$$\psi_{\mathrm{B}}(\boldsymbol{x}) = k\psi_{\mathrm{B}}^*(\boldsymbol{x}) = k(\varphi^*(\boldsymbol{x}) + n_{\mathrm{B}}^*(\boldsymbol{x})) = k\varphi^*(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) \ . \tag{3.30}$$

We see that we should scale the noise variance σ_{B}^2 by k^2 to obtain the variance of n_{B}. If we estimate that noise variance directly from the observed image ψ_{B}, then the scale factor will be built into our estimate, and we only need to concern ourselves with the effect of k as in our original model formulation.

Say that we choose the wrong value for k, $\tilde{k} = \tau k$. Then we see ψ_{B} but think we see

$$\tilde{I}_B(\boldsymbol{x}) = \tilde{k}\varphi(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) = k\tilde{\varphi}(\boldsymbol{x}) + n_{\mathrm{B}}(\boldsymbol{x}) \tag{3.31}$$

where $\tilde{\varphi} = \tau\varphi$. So we actually estimate $\tilde{\varphi}$ instead of φ.

At first glance, this would not appear to be a problem. We already stated that scaling the final result by an arbitrary (and unknown) scale factor does not matter. An issue arises in the choices of α and γ. Imagine a scenario where we have no noise. Then we would produce the following estimates for φ and β:

$$\hat{\varphi}(\boldsymbol{x}) = \frac{\psi_{\mathrm{B}}(\boldsymbol{x})}{\tilde{k}} = \frac{\varphi(\boldsymbol{x})}{\tau} \tag{3.32a}$$

$$\hat{\beta}(\boldsymbol{x}) = \frac{\tilde{k}\psi_{\mathrm{S}}(\boldsymbol{x})}{\psi_{\mathrm{B}}(\boldsymbol{x})} = \tau\beta(\boldsymbol{x}) \ . \tag{3.32b}$$

If we choose α and γ assuming $\tilde{k}$ but actually having k, we end up having α being off by a factor of τ^q and γ being off by a factor of $1/\tau^p$. Say $\tau > 1$. Then by choosing the wrong k, we end up over-regularizing $\hat{\beta}$ and under-regularizing $\hat{\varphi}$. The opposite is true for $\tau < 1$. Thus the sensitivity to error in k is solely determined by sensitivity to error in α and γ.

As we discussed in Section 3.2.3, one way to view the regularization parameter for the discrete formulation is as an IID statistical prior on derivatives of our fields $\boldsymbol{f}$ and $\boldsymbol{b}$. A ℓ_1 norm imposes a zero-mean Laplacian prior on the derivative while a ℓ_2 norm imposes a zero-mean Gaussian prior. The regularization parameters α and γ can then be interpreted from a statistical viewpoint as having some relationship to the variance we expect to see in our fields.

We will only address the choice of α here. Arguments for the choice of γ can be made in the same manner. Let $\boldsymbol{L}$ represent our derivative operator (or, more generally, linear operator). For a ℓ_2 penalty on $\hat{\boldsymbol{b}}$, $\alpha = \sigma_{\mathrm{B}}^2/\sigma^2$ where σ^2 is the variance of $\boldsymbol{L}\boldsymbol{b}^*$. Ordinarily the weight would be $1/2\sigma^2$ from the Gaussian PDF. We only use three weight parameters for the four terms in our energy functional (3.16) by normalizing by $1/2\sigma_{\mathrm{B}}^2$ which would otherwise appear as the weight for the body coil data fidelity term. The inverse relationship to the variance of $\boldsymbol{L}\boldsymbol{b}^*$ properly reflects the fact that if we know that there are many high derivative values, then we should penalize large derivative values less.

Perhaps the best way to compute α would be to have some training data consisting of known bias profiles of the surface coil we are employing. We could then apply our derivative operators to these bias fields and calculate the sample statistics and use these values in our algorithm. In the absence of training data, there is not really a good way to estimate these parameters from the data. If we attempt to estimate α from an initial estimate of $\boldsymbol{b}^*$, we are then forcing our final bias field estimate to have similar smoothness properties as this rough estimate.

There is a technique in the literature for regularization parameter selection called the L-curve [7, 32]. The method is based on the following observation: close to the optimal value, to one side the regularization energy is very sensitive to the regularization penalty while the data fidelity term is relatively insensitive; the opposite is true on the other side. Hence we can find the optimum choice by looking for the point where this change occurs. This method is not theoretically justified and its asymptotic behavior is proven to give incorrect results. Nonetheless, it seems to provide good results in practice. Another common *ad hoc* method is manually adjusting the regularization penalty until desirable results are achieved. This can be useful when automatic techniques fail. This may still be incorporated into a fully automatic algorithm if training data are available, and the data produce a consistent choice for the regularization penalty (or a consistent relationship can be easily calculated).

■ 3.3.2 Initial Values

All of the techniques that we will describe require initial guesses to begin the iterative solver. The performance and behavior of the solvers can be greatly impacted by our choice of initializations. The closer we start to the correct answer, the less time it will take to converge to the global minimum and the less likely it will be to get caught in

local minima. Here we will just briefly mention a few techniques that can be used to initialize our algorithm.

If we look at our measurement model, the obvious choice for $\hat{\varphi}^{(0)}$ is to take ψ_B, put it through a low-pass filter, and divide by k. This will help average out the noise at the expense of blurring across regions and losing some edge information. We can also use ψ_S to generate $\hat{\varphi}^{(0)}$ by dividing by $\hat{\beta}^{(0)}$, but it may be better at first to begin with an estimate that is explicitly bias free. If we were confident that $\hat{\beta}^{(0)}$ does a good job correcting ψ_S, then the problem would be solved. Additional problems can arise in areas that are far away from the surface coil. At these points, the sensitivity of the surface coil is very low and the noise will overwhelm the signal.

Our choice for $\hat{\varphi}^{(0)}$ was simple because we have access to a signal that is just φ^* corrupted by additive noise. It is not quite as simple for the bias field because we do not have access to β^* isolated from φ^*. Generating a good $\hat{\beta}^{(0)}$ is equivalent to generating a good estimate of β^*. So we can apply other techniques already in the literature to create $\hat{\beta}^{(0)}$. We have already discussed a variety of methods in Section 2.3. Most techniques in the literature require either significant computation or user involvement. We desire for the initialization calculations to be much less than the computation required for the remainder of the algorithm, and one of the goals of our algorithm is to have zero user input.

The simplest of the techniques to apply is homomorphic unsharp mask filtering. Using this method, we set $\hat{\beta}^{(0)}$ as the result of a low-pass linear filter applied to either ψ_S or $\log(\psi_S)$. This method ignores the extra data that we have available in ψ_B. Homomorphic filtering will hopefully minimize the noise and give us the low frequency part of the product of β^* and φ^*. This results in our initial bias estimate still having a lot of tissue dependence.

A more effective technique is to apply the method of Brey and Narayana. This method utilizes both the surface coil image and the body coil image to generate an estimate of the bias field. This method actually produces very good results and is only slightly more complicated than homomorphic filtering. To generate $\hat{\beta}^{(0)}$ we need to apply low-pass filters to both ψ_S and ψ_B and divide the results.

One thing we notice using these initializers is that we do not get a good estimate of β^* in regions where φ^* is near zero—we are unable to observe the effects of the bias field due to the lack of signal intensity. So when we run our solver on this, it takes a while for the bias to propagate into the air-filled regions. This is especially evident

when using a large α which imposes a lot of spatial smoothness. One solution to this is to estimate $\hat{\beta}^{(0)}$ from the images only in regions where ψ_{B} is large. In the regions where ψ_{B} is small, we interpolate the bias field from the part where we have meaningful observations. Essentially we fill in the missing data such that the regularization term is minimized. This interpolation does not affect the rest of the energy functional. The data fidelity term for ψ_{B} does not involve β. Assuming we have a reasonable estimate for φ^*, the data fidelity term for ψ_{S} should also be unaffected. $\hat{\varphi}$ should be small in these regions which mitigates the effects of errors in $\hat{\beta}$. For a Laplacian penalty, this initialization is equivalent to solving Laplace's equation with boundary conditions. For a gradient penalty, it is impossible to get zero energy from air-filled regions unless all of the values on the boundary are exactly the same. So we seek to minimize $\|\boldsymbol{L}_\beta \beta\|_q^q$ by altering the values within the air-filled region. When $q = 2$, we can minimize the energy by again solving Laplace's equation.

There is also the question of whether we wish to begin on a f-step or a b-step. Generally we will want to begin on the step that corresponds to whichever initialization we have the least confidence in (*i.e.*, if we trust $\hat{\varphi}^{(0)}$ more than we trust $\hat{\beta}^{(0)}$, then we should keep $\hat{\varphi}^{(0)}$ and begin on a b-step). This is because we use $\hat{\beta}$ to compute $\hat{\varphi}$ and vice versa. If we are not very confident in $\hat{\beta}^{(0)}$, then there is little point in starting on a f-step and propagating our uncertainty in $\hat{\beta}^{(0)}$ into $\hat{\varphi}$. In general, the choice of which step to begin with does not affect whether the algorithm will converge. But it can affect how quickly we converge.

The λ in (3.16) should reflect the total uncertainty we have in our estimates of ψ_{B} and ψ_{S}. When $\hat{\varphi}$ and $\hat{\beta}$ are close to φ^* and β^*, most of the uncertainty will be noise. In this case we should primarily determine λ from the noise variances. We should generally determine λ based on the total estimation error, not just that due to noise. This is relevant in our algorithm because we minimize our energy functional using coordinate descent. In this method, we alternately iterate over $\hat{\beta}$ and $\hat{\varphi}$ while fixing the other variable. So when we minimize (3.16) over β assuming $\hat{\varphi}$, we should account both for the uncertainty in our measurement of ψ_{S} and the uncertainty in our estimate $\hat{\varphi}$. It is difficult to do this quantitatively because a lot of the uncertainty in $\hat{\varphi}$ comes from the noise in ψ_{B} and ψ_{S}. But the idea we should take from this is that when estimating φ^* at the beginning of the algorithm, we are better off putting more emphasis on the data fidelity with ψ_{B} than with ψ_{S} because the latter can be heavily influenced by an incorrect $\hat{\beta}$. Generally this is not as big of an issue with $\hat{\beta}$ because we know we can get

reliable estimates of φ^* from ψ_{B}.

3.3.3 Continuous variational solution

The calculus of variations was described in Section 2.1.3. We can use it to find functions which minimize an energy functional. A stationary point can only occur at the solution of the Euler-Lagrange equation. So we will construct the appropriate Euler-Lagrange equations and solve the resulting PDEs. We will derive the appropriate equations for gradient and Laplacian regularizers with $\mathcal{L}_2$ norms on $\hat{\beta}$ and $\hat{\varphi}$. In addition, we will generate methods to solve minimization problems when using the TV-norm (gradient penalty and a $\mathcal{L}_1$ norm) on $\hat{\varphi}$.

No regularization

We begin with a discussion of the solutions that occur for the no regularization case. The unregularized solutions for both $\hat{\varphi}$ and $\hat{\beta}$ later become components of the full solution using regularization. We often implement the energy functional with no regularization on $\hat{\varphi}$. The reason for this is that the problem is sufficiently constrained by our body coil observations—$\hat{\varphi}$ will not deviate significantly from ψ_{B}. The implementation of regularization on $\hat{\varphi}$ will improve the results by reducing the noise, but it is not necessary and slows down computational speed.

To minimize (3.16) with respect to φ given $\hat{\beta}$ and $\gamma = 0$, we see that we only need to minimize the following functional:

$$\begin{aligned} E(\varphi) &= E_{\mathrm{B}}(\varphi) + \lambda E_{\mathrm{S}}(\varphi, \beta) && (3.33) \\ &= \int_{\Omega} [(\psi_{\mathrm{B}}(\boldsymbol{x}) - k\varphi(\boldsymbol{x}))^2 + \lambda(\psi_{\mathrm{S}}(\boldsymbol{x}) - \hat{\beta}(\boldsymbol{x})\varphi(\boldsymbol{x}))^2] \mathrm{d}\boldsymbol{x} \ . && (3.34) \end{aligned}$$

We can clearly see that given any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, altering the value of $\varphi(\boldsymbol{x}_1)$ does not affect the choice of $\varphi(\boldsymbol{x}_2)$. There is no spatial coupling. Hence we can simply optimize E on a pointwise basis:

$$\hat{\varphi}(\boldsymbol{x}) = \arg\min_{\varphi(\boldsymbol{x})} (\psi_{\mathrm{B}}(\boldsymbol{x}) - k\varphi(\boldsymbol{x}))^2 + \lambda \left(\psi_{\mathrm{S}}(\boldsymbol{x}) - \hat{\beta}(\boldsymbol{x})\varphi(\boldsymbol{x})\right)^2 \forall \boldsymbol{x} \in \Omega \ . \quad (3.35)$$

Necessary and sufficient conditions for the minimum are that the first derivative is zero and the second derivative is positive:

$$\begin{aligned} \frac{\partial}{\partial \varphi} \mathrm{d}E &= -2k(\psi_{\mathrm{B}}(\boldsymbol{x}) - k\varphi(\boldsymbol{x})) - 2\lambda\hat{\beta}(\boldsymbol{x})(\psi_{\mathrm{S}}(\boldsymbol{x}) - \hat{\beta}(\boldsymbol{x})\varphi(\boldsymbol{x})) = 0 && (3.36) \\ \frac{\partial^2}{\partial \varphi^2} \mathrm{d}E &= 2k^2 + 2\lambda\hat{\beta}^2(\boldsymbol{x}) \ . && (3.37) \end{aligned}$$

Clearly the condition on the second derivative is always satisfied (for positive λ), and the condition on the first derivative is satisfied when

$$\hat{\varphi}(\boldsymbol{x}) = \frac{k\psi_{\mathrm{B}}(\boldsymbol{x}) + \lambda\hat{\beta}(\boldsymbol{x})\psi_{\mathrm{S}}(\boldsymbol{x})}{k^2 + \lambda\hat{\beta}^2(\boldsymbol{x})} \ . \tag{3.38}$$

Note that when $\hat{\beta}(\boldsymbol{x}) \gg k$, $\hat{\varphi}(\boldsymbol{x}) \approx \psi_{\mathrm{S}}(\boldsymbol{x})/\hat{\beta}(\boldsymbol{x})$. When $\hat{\beta}(\boldsymbol{x}) \ll k$, $\hat{\varphi}(\boldsymbol{x}) \approx \psi_{\mathrm{B}}(\boldsymbol{x})/k$. An interesting observation we can make is that $\hat{\varphi}$ is the noise-weighted convex combination of ψ_{B}/k and $\psi_{\mathrm{S}}/\hat{\beta}$ where the weighting factor in the combination is spatially varying. The noise variance of ψ_{B}/k is $\sigma_{\mathrm{B}}^2/k^2$, and the noise variance of $\psi_{\mathrm{S}}(\boldsymbol{x})/\hat{\beta}(\boldsymbol{x})$ is $\sigma_{\mathrm{S}}^2/\hat{\beta}^2(\boldsymbol{x})$. So we can rewrite (3.38) as

$$\begin{aligned} \hat{\varphi}(\boldsymbol{x}) &= \eta(\boldsymbol{x})\frac{\psi_{\mathrm{B}}(\boldsymbol{x})}{k} + (1-\eta(\boldsymbol{x}))\frac{\psi_{\mathrm{S}}(\boldsymbol{x})}{\hat{\beta}(\boldsymbol{x})} \\ \eta(\boldsymbol{x}) &= \frac{\frac{k^2}{\sigma_{\mathrm{B}}^2}}{\frac{k^2}{\sigma_{\mathrm{B}}^2} + \frac{\hat{\beta}^2(\boldsymbol{x})}{\sigma_{\mathrm{S}}^2}} = \frac{k^2}{k^2 + \lambda\hat{\beta}^2(\boldsymbol{x})} \ . \end{aligned} \tag{3.39}$$

This provides a natural way to combine both of our input images into our resulting output image. The results that we obtain are similar to those obtained by Roemer *et al.* [62]. We will discuss these connections in more detail in Section 3.4.3.

This reconstruction from both observation images is different than the method of Brey and Narayana [12] which only uses the data from ψ_{S} to construct $\hat{\varphi}$. Assuming we know the optimal solution, then $\hat{\beta} \equiv \beta^*$ and the error variance of $\hat{\varphi}_{\mathrm{BN}}$ is

$$E[(\hat{\varphi}_{\mathrm{BN}}(\boldsymbol{x}) - \varphi(\boldsymbol{x}))^2] = \sigma_{\mathrm{S}}^2/\beta^{*2}(\boldsymbol{x}) \ . \tag{3.40}$$

When β^* is large, the error variance is small. When β^* is small, the error variance is large. For our reconstruction using both images, the error variance is

$$\sigma_{\mathrm{err}}^2(\boldsymbol{x}) = E[(\hat{\varphi}(\boldsymbol{x}) - \varphi^*(\boldsymbol{x}))^2] = \frac{\sigma_{\mathrm{B}}^2/k^2}{1 + \lambda\beta^2(\boldsymbol{x})/k^2} = \frac{\sigma_{\mathrm{S}}^2/\beta^{*2}(\boldsymbol{x})}{1 + \frac{1}{\lambda}k^2/\beta^{*2}(\boldsymbol{x})} \ . \tag{3.41}$$

When $\lambda \geq k^2/\beta^{*2}(\boldsymbol{x})$,

$$\frac{\sigma_{\mathrm{B}}^2}{2k^2} \leq \sigma_{\mathrm{err}}^2(\boldsymbol{x}) \leq \frac{\sigma_{\mathrm{B}}^2}{k^2} \ .$$

When $\lambda \leq k^2/\beta^{*2}(\boldsymbol{x})$,

$$\frac{\sigma_{\mathrm{S}}^2}{2\beta^{*2}(\boldsymbol{x})} \leq \sigma_{\mathrm{err}}^2(\boldsymbol{x}) \leq \frac{\sigma_{\mathrm{S}}^2}{\beta^{*2}(\boldsymbol{x})} \ .$$

Let $\sigma_{\mathrm{min}}^2 = \min(\sigma_{\mathrm{S}}^2/\beta^{*2}, \sigma_{\mathrm{B}}^2/k^2)$. Then we see that

$$\tfrac{1}{2}\sigma_{\mathrm{min}}^2 \leq \sigma_{\mathrm{err}}^2 \leq \sigma_{\mathrm{min}}^2 \ . \tag{3.42}$$

Thus we see that in the worst case—which occurs when one of the observation signals is zero—we achieve variance equal to that of the best corrected image. In the best case—which occurs when both corrected images have the same variance—we achieve variance that is half that of the best corrected image. This boosts the SNR from 0 to 3 dB above that attainable by just using the best corrected image. While this may not be a dramatic increase, it can be many dB better than the error achieved using the Brey and Narayana method. In a location far from the surface coil, the variance of ψ_S/β^* is much greater than the variance of ψ_B/k. Brey and Narayana will achieve error variance of $\sigma_S^2/\beta^{*2}(\boldsymbol{x})$ while our method will have error variance approximately equal to σ_B^2/k^2. This can be a tremendous gain as we will demonstrate in Chapter 4

We can do a similar analysis for β with $\alpha = 0$ and a given $\hat{\varphi}$. We can again optimize E pointwise:

$$\hat{\beta}(\boldsymbol{x}) = \arg\min_{\beta(\boldsymbol{x})} \lambda \left(\psi_S(\boldsymbol{x}) - \hat{\varphi}(\boldsymbol{x})\beta(\boldsymbol{x})\right)^2 \forall\, \boldsymbol{x} \in \Omega \ . \tag{3.43}$$

The derivatives are

$$\frac{\partial}{\partial\beta}\mathrm{d}E = -2\lambda\hat{\varphi}(\boldsymbol{x})(\psi_S(\boldsymbol{x}) - \hat{\varphi}(\boldsymbol{x})\beta(\boldsymbol{x})) = 0 \tag{3.44}$$

$$\frac{\partial^2}{\partial\beta^2}\mathrm{d}E = 2\lambda\hat{\varphi}^2(\boldsymbol{x}) \ , \tag{3.45}$$

and the solution is trivial to compute:

$$\hat{\beta}(\boldsymbol{x}) = \frac{\psi_S(\boldsymbol{x})}{\hat{\varphi}(\boldsymbol{x})} \ . \tag{3.46}$$

Gradient regularization

We begin with the analysis on a b-step. In applying the Euler-Lagrange equation, we can break the equation into two parts. One part corresponds to the data fidelity portions of our energy functional, and the other corresponds to the regularization portion (we make $\hat{\varphi}$ implicit in J):

$$E(\beta) = \int_\Omega J(\boldsymbol{x}, \beta, \dot{\beta})\mathrm{d}\boldsymbol{x} = \int_\Omega \lambda J^\beta_{\text{data}}(\boldsymbol{x}, \beta) + \alpha J^\beta_{\text{reg}}(\boldsymbol{x}, \dot{\beta})\mathrm{d}\boldsymbol{x} \ . \tag{3.47}$$

The first variation of J^β_{data} with respect to $\dot{\beta}$ is zero, and the first variation of J^β_{reg} with respect to β is also zero. Therefore we can write the Euler-Lagrange differential equation as

$$\frac{\partial J}{\partial\beta} - \nabla\cdot\left(\frac{\partial J}{\partial\dot{\beta}}\right) = \lambda\frac{\partial J^\beta_{\text{data}}}{\partial\beta} - \alpha\nabla\cdot\left(\frac{\partial J^\beta_{\text{reg}}}{\partial\dot{\beta}}\right) = 0 \ . \tag{3.48}$$

The first term of the equation is simply the derivative that we carried out earlier when there was no regularization term. We see that when $\alpha = 0$, this reduces into the same condition on the first derivative that we observed previously.

For a gradient regularization with a $\mathcal{L}_2$ norm, we get the following equations (we drop all terms that do not directly involve β):

$$J^{\beta}_{\text{data}}(\beta) = (\psi_{\text{S}} - \hat{\varphi}\beta)^2 \tag{3.49a}$$

$$J^{\beta}_{\text{reg}}(\beta) = \|\nabla\beta\|^2 \ . \tag{3.49b}$$

We take the first variations so we can form the Euler-Lagrange equation:

$$\frac{\partial J^{\beta}_{\text{data}}}{\partial \beta} = -2\hat{\varphi}(\psi_{\text{S}} - \hat{\varphi}\beta) \tag{3.50a}$$

$$\frac{\partial J^{\beta}_{\text{reg}}}{\partial \dot{\beta}} = 2\nabla\beta \ . \tag{3.50b}$$

Thus we conclude that a stationary point occurs when

$$\begin{aligned} &-2\lambda\hat{\varphi}(\psi_{\text{S}} - \hat{\varphi}\beta) - 2\alpha\nabla^2\beta = 0 \\ &\Rightarrow \alpha\nabla^2\beta - \lambda\hat{\varphi}^2\beta = -\lambda\hat{\varphi}\psi_{\text{S}} \ . \end{aligned} \tag{3.51}$$

This is a damped Poisson's equation for which a solution does not generally exist in closed form. We can solve this numerically by introducing a time variable t and time dependence into β:

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = -\frac{\partial J}{\partial \beta} + \nabla \cdot \left(\frac{\partial J}{\partial \dot{\beta}}\right) = 2\lambda\hat{\varphi}(\psi_{\text{S}} - \hat{\varphi}\beta) + 2\alpha\nabla^2\beta \ . \tag{3.52}$$

In order to solve this equation, we set $\beta(0,\cdot) = \hat{\beta}^{(0)}(\cdot)$ as our initial conditions. We can then integrate both sides with respect to t to obtain our solution:

$$\hat{\beta}(\boldsymbol{x}) = \int_0^{\infty} \frac{\mathrm{d}\beta(t,\boldsymbol{x})}{\mathrm{d}t}\mathrm{d}t + \hat{\beta}^{(0)}(\boldsymbol{x}) \ . \tag{3.53}$$

To actually implement this integration on a computer, we can approximate it as a discrete sum. If $\hat{\beta}$ converges to a local minimum, $\frac{\mathrm{d}\beta}{\mathrm{d}t} \to 0$, so we only need to sum a finite number of terms. We must choose the step size so that it is small enough to ensure that each gradient step is a descent step but large enough to minimize convergence time.

On a f-step, we can do the same analysis:

$$J^{\varphi}_{\text{data}}(\varphi) = (\psi_{\text{B}} - k\varphi)^2 + \lambda(\psi_{\text{S}} - \hat{\beta}\varphi)^2 \tag{3.54a}$$

$$J^{\varphi}_{\text{reg}}(\varphi) = \|\nabla\varphi\|^2 \ , \tag{3.54b}$$

and we arrive at the following PDE:

$$\begin{aligned} \frac{\mathrm{d}\varphi}{\mathrm{d}t} = -\frac{\partial J}{\partial \varphi} + \nabla \cdot \left(\frac{\partial J}{\partial \dot{\varphi}}\right) &= 2k(\psi_{\mathrm{B}} - k\varphi) + 2\lambda\hat{\beta}(\psi_{\mathrm{S}} - \hat{\beta}\varphi) + 2\gamma\nabla^2\varphi \\ & \varphi(0,\cdot) = \hat{\varphi}^{(0)}(\cdot) \ . \end{aligned} \tag{3.55}$$

When we have a $\mathcal{L}_1$ norm on a f-step (we omit the derivation for a b-step because we do not use $\mathcal{L}_1$ norms on the bias field), this only alters the regularization term:

$$J^{\varphi}_{\mathrm{reg}}(\varphi) = |\nabla\varphi| \ . \tag{3.56}$$

The first variation with respect to $\dot{\varphi}$ is:

$$\frac{\partial J^{\varphi}_{\mathrm{reg}}}{\partial \dot{\varphi}} = \frac{\nabla\varphi}{|\nabla\varphi|} \ . \tag{3.57}$$

This results in the following gradient flow:

$$\begin{aligned} \frac{\mathrm{d}\varphi}{\mathrm{d}t} &= 2k(\psi_{\mathrm{B}} - k\varphi) + 2\lambda\hat{\beta}(\psi_{\mathrm{S}} - \hat{\beta}\varphi) + 2\gamma\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) \\ & \varphi(0,\cdot) = \hat{\varphi}^{(0)}(\cdot) \ . \end{aligned} \tag{3.58}$$

Laplacian regularization

We can follow the same analysis track for a Laplacian regularizer. As noted earlier, the data fidelity term does not change with the regularization, so we can leave that untouched. For a b-step, the only change we need to make is to the regularization term

$$J^{\beta}_{\mathrm{reg}}(\beta) = (\nabla^2\beta)^2 \ . \tag{3.59}$$

We take the first variation with respect to $\dot{\beta}$:

$$\frac{\partial J^{\beta}_{\mathrm{reg}}}{\partial \dot{\beta}} = -2\nabla(\nabla^2\beta) \ . \tag{3.60}$$

This results in the following gradient formulation:

$$\begin{aligned} \frac{\mathrm{d}\beta}{\mathrm{d}t} &= 2\lambda\hat{\varphi}(\psi_{\mathrm{S}} - \hat{\varphi}\beta) - 2\alpha\nabla^4\beta \\ & \beta(0,\cdot) = \hat{\beta}^{(0)}(\cdot) \ . \end{aligned} \tag{3.61}$$

Note that if we do not add the time variable, the PDE that needs to be solved is an inhomogeneous damped biharmonic equation.

We simply state the gradient flow for the f-step:

$$\begin{aligned} \frac{\mathrm{d}\varphi}{\mathrm{d}t} &= 2k(\psi_{\mathrm{B}} - k\varphi) + 2\lambda\hat{\beta}(\psi_{\mathrm{S}} - \hat{\beta}\varphi) - 2\gamma\nabla^4\varphi \\ & \varphi(0,\cdot) = \hat{\varphi}^{(0)}(\cdot) \ . \end{aligned} \tag{3.62}$$

■ 3.3.4 Discrete Solution

Instead of formulating our solution from a continuous problem and then using discrete methods to implement the solution, we can model our problem as discrete and directly proceed from there. There are many advantages for modeling the entire problem as discrete rather than implementing a discrete/continuous hybrid. The discrete method allows us to perform exact line searches at each gradient step. This not only makes gradient descent more attractive, it also lets us apply more advanced gradient search algorithms such as conjugate gradient with preconditioners. We gain a great deal more flexibility in the regularization because the discrete framework easily extends to allow any linear filter rather than the derivative operations we were limited to using in the continuous solution. We will only deal with ℓ_2 norms in this section. More general ℓ_p solutions are covered in Section 3.4.1.

Quadratic Subproblems

We write our discrete energy functional here for convenience, constructing it by substituting (3.24) into (3.16):

$$E(\boldsymbol{f},\boldsymbol{b}) = \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 + \lambda\|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{B}\boldsymbol{f}\|^2 + \alpha\|\boldsymbol{L}_b\boldsymbol{b}\|^2 + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|^2 \ . \tag{3.63}$$

For notational simplicity, we again use the diagonal matrices $\boldsymbol{F}$ and $\boldsymbol{B}$ which have $\boldsymbol{f}$ and $\boldsymbol{b}$ respectively along the diagonals. The second term can be equivalently written in terms of $\boldsymbol{b}$ by noting:

$$\|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{B}\boldsymbol{f}\|^2 = \|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{F}\boldsymbol{b}\|^2 \ . \tag{3.64}$$

When we use ℓ_2 norms, all of the terms in our energy functional are quadratic. Thus we have a quadratic optimization problem in terms of either $\boldsymbol{f}$ or $\boldsymbol{b}$, but the overall problem is complicated by the cross multiplication between $\boldsymbol{f}$ and $\boldsymbol{b}$. Unfortunately, the discrete energy functional does not lend itself easily to analysis. It can be shown

that the discrete problem is non-convex due to the Hadamard product. But the problem is much simpler in terms of just $\boldsymbol{f}$ or just $\boldsymbol{b}$. We observe in (3.63) that if we hold $\boldsymbol{b}$ constant (and thus $\boldsymbol{B}$ as well), E would be quadratic in terms of $\boldsymbol{f}$. Similarly, using (3.64), if $\boldsymbol{f}$ were a constant, then E would be quadratic in terms of $\boldsymbol{b}$.

If we use a coordinate descent approach again, we find that the individual optimization problems are much simpler and have nice properties. When we consider the optimization of just one coordinate, we can put the energy functional into standard quadratic form:

$$E(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + c \tag{3.65}$$

where $\boldsymbol{Q}$ is a MNxMN matrix, $\boldsymbol{a} \in \mathbb{R}^{MN}$, and $c \in \mathbb{R}$.

For obtaining $\hat{\boldsymbol{f}}$ with a given $\hat{\boldsymbol{b}}$, we see that the energy functional is equivalent to the following choices for $\boldsymbol{Q}$ and $\boldsymbol{a}$ (c is inconsequential):

$$\boldsymbol{Q}_f = 2k^2\boldsymbol{I} + 2\lambda\hat{\boldsymbol{B}}^2 + 2\gamma\boldsymbol{L}_f^{\mathrm{T}}\boldsymbol{L}_f \tag{3.66}$$

$$\boldsymbol{a}_f = 2k\boldsymbol{y}_{\mathrm{B}} + 2\lambda\hat{\boldsymbol{B}}\boldsymbol{y}_{\mathrm{S}} \ . \tag{3.67}$$

When $\gamma = 0$, $\boldsymbol{Q}_f$ is a diagonal matrix, and all of the values of $\hat{\boldsymbol{f}}$ are decoupled from each other. In general, $\boldsymbol{Q}_f$ is a sparse and banded matrix when $\boldsymbol{L}_f$ implements a local linear filter. The sparsity of $\boldsymbol{Q}_f$ is determined by the width of the filter kernel.

For obtaining $\hat{\boldsymbol{b}}$ (given $\hat{\boldsymbol{f}}$), we get the following for $\boldsymbol{Q}$ and $\boldsymbol{a}$:

$$\boldsymbol{Q}_b = 2\lambda\hat{\boldsymbol{F}}^2 + 2\alpha\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b \tag{3.68}$$

$$\boldsymbol{a}_b = 2\lambda\hat{\boldsymbol{F}}\boldsymbol{y}_{\mathrm{S}} \ . \tag{3.69}$$

When $\alpha = 0$, $\boldsymbol{Q}_b$ is a diagonal matrix. Otherwise it will also be sparse and banded. An interesting thing to note is that $\boldsymbol{y}_{\mathrm{B}}$ does not appear in the expressions for $\boldsymbol{Q}_b$ and $\boldsymbol{a}_b$. This is not to say that $\boldsymbol{y}_{\mathrm{B}}$ does not contain any useful information in estimating $\boldsymbol{b}$. If this were the case, we could save ourselves a great deal of trouble and not generate the body coil images. All this tells us is that all the information that $\boldsymbol{y}_{\mathrm{B}}$ contains about the bias field is contained in $\hat{\boldsymbol{f}}$.

The $\boldsymbol{Q}$ matrices are positive semi-definite so the subproblems are convex and have nice convergence properties. Note that in certain degenerate cases, $\boldsymbol{Q}_b$ is not invertible. For instance, let $\hat{\boldsymbol{F}} \equiv 0$. Then $\boldsymbol{Q}_b = 2\alpha\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b$. If $\boldsymbol{L}_b$ implements a derivative operator, the nullspace should include constant vectors[3], and $\boldsymbol{L}_b$ is singular. Thus $\boldsymbol{Q}_b$ is not

[3] A derivative is a local differential operator, so the derivative of a constant function should be zero.

invertible. $\boldsymbol{Q}_f$ does not have this problem due to the data fidelity term for the body coil image. We will ignore this degenerate case and stipulate that we cannot initialize our solver with $fvh^{(0)} \equiv 0$. Our iterative solver will not converge to a $\hat{\boldsymbol{f}}$ that is zero except in the degenerate case where both $\boldsymbol{y}_{\mathrm{B}}$ and $\boldsymbol{y}_{\mathrm{S}}$ are zero. In this case, there really is not any information present in our observations. Thus except in rare instances, we do not have existence problems for either of our quadratic subproblems.

Because each subproblem has a positive definite $\boldsymbol{Q}$ matrix, each subproblem is strictly convex. Strict convexity implies that a unique global minimum exists (for a specific choice of $\hat{\boldsymbol{b}}$ or $\hat{\boldsymbol{f}}$ and solving for the other variable), and any local minimum we find is also the global minimum. The convexity can be seen if we compute the first and second derivatives for quadratic functions:

$$\nabla E(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{a} \tag{3.70}$$

$$\mathcal{H}\{E(\boldsymbol{x})\} = \boldsymbol{Q} \ . \tag{3.71}$$

Both of our $\boldsymbol{Q}$ matrices are positive definite so the Hessian is positive definite which is a necessary and sufficient condition for strict convexity.

Necessary and sufficient conditions for a local minimum are that the gradient is zero and the Hessian is positive definite. We see that this occurs only when

$$\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{a} \ . \tag{3.72}$$

We know $\boldsymbol{Q} > 0$, so it is invertible. We can write the solution as

$$\hat{\boldsymbol{x}} = \boldsymbol{Q}^{-1}\boldsymbol{a} \tag{3.73}$$

which is unique.

Unfortunately $\boldsymbol{Q}$ tends to be a large matrix for realistic data sets (*e.g.*, a 256x256 image generates a $\boldsymbol{Q}$ matrix that is 65536x65536 and has over four billion entries). The sparsity of $\boldsymbol{Q}$ makes it feasible to compute and store, but it is still computationally difficult to generate the inverse. There are $2MN$ variables in the linear system and directly solving the linear system is then a $\mathcal{O}(M^2N^2)$ operation. Doubling the resolution (*e.g.*, from 256x256 to 512x512) increases the computational complexity by a factor of 16. In practice the difference is not quite so severe due to the banded and sparse nature of $\boldsymbol{Q}$, but it is still not computationally efficient.

If the function is constant, all local differences should be zero which means that a constant vector is in the nullspace of the matrix.

In addition, in our coordinate descent framework, it is wasteful to compute an exact answer when there is error in the other coordinate (*e.g.*, $\hat{\boldsymbol{f}} \not\equiv \boldsymbol{f}^*$). Going all the way to the exact solution may in fact be taking you further away from the correct answer. Luckily there are a number of iterative methods that can generate sub-optimal approximations. These were detailed in Section 2.1.4 and include techniques such as gradient descent, conjugate gradient, and Newton's method.

Solutions without regularization

When solving for $\hat{\boldsymbol{f}}$ and $\gamma = 0$, $\boldsymbol{Q}_f$ simplifies to $2k^2\boldsymbol{I} + 2\lambda\hat{\boldsymbol{B}}^2$. This matrix is diagonal, so it is trivial to solve $\boldsymbol{Q}_f\boldsymbol{f} = \boldsymbol{a}_f$ as

$$\hat{\boldsymbol{f}}[n] = \frac{k\boldsymbol{y}_{\mathrm{B}}[n] + \lambda\hat{\boldsymbol{b}}[n]\boldsymbol{y}_{\mathrm{S}}[n]}{k^2 + \lambda\hat{\boldsymbol{b}}^2[n]} \quad . \tag{3.74}$$

A similar situation exists when solving for $\hat{\boldsymbol{b}}$ and $\alpha = 0$ (given $\hat{\boldsymbol{f}}$):

$$\hat{\boldsymbol{b}}[n] = \frac{\boldsymbol{y}_{\mathrm{S}}[n]}{\hat{\boldsymbol{f}}[n]} \quad . \tag{3.75}$$

We note that these are the same results that we obtained in (3.38) and (3.46) for the continuous problem (except for a change in notation to reflect our discrete signal model).

Gradient solvers

For finding suboptimal solutions, there are a number of iterative techniques that are guaranteed to find the global minimum for convex optimization problems (as detailed in Section 2.1.4). By using an iterative algorithm to compute the solutions to the subproblems, we now have our coordinate descent iterations and then subiterations within each f- and b-step. We indicate our current estimate for $\boldsymbol{f}^*$ as $\hat{\boldsymbol{f}}^{(i,j)}$ where i refers to the current coordinate descent iteration and j refers to our current iteration in the solver for the subproblem. Similar notation holds for all other parameters. In this section, for simplicity, we will refer to the current subiteration as $\hat{\boldsymbol{f}}^{(j)}$ with the i being implicit. When we are on a b-step, $\hat{\boldsymbol{f}}$ is our current estimate for $\boldsymbol{f}^*$ that we are holding fixed. Similarly, when we are on a f-step, $\hat{\boldsymbol{b}}$ is our current estimate for $\boldsymbol{b}^*$ that we are holding fixed.

Because we are solving a quadratic problem, it is easy to compute gradients. The simplest method that uses this information is gradient descent which only requires the

gradient value at each iteration. When minimizing (3.65), the update equations take on a very simple form using the quadratic gradient as defined in (3.70):

$$\boldsymbol{d}^{(j)} = -\nabla E(\hat{\boldsymbol{x}}^{(j)}) = -\boldsymbol{Q}\hat{\boldsymbol{x}}^{(j)} + \boldsymbol{a} \tag{3.76a}$$

$$\hat{\boldsymbol{x}}^{(j+1)} = \hat{\boldsymbol{x}}^{(j)} + \eta^{(j)}\boldsymbol{d}^{(j)} \ . \tag{3.76b}$$

The conjugate gradient update equations are also relatively simple, and we shall just state them:

$$\boldsymbol{g}^{(j)} = \nabla E(\hat{\boldsymbol{x}}^{(j)}) = \boldsymbol{Q}\hat{\boldsymbol{x}}^{(j)} - \boldsymbol{a} \tag{3.77a}$$

$$\boldsymbol{d}^{(0)} = -\boldsymbol{g}^{(0)} \tag{3.77b}$$

$$\boldsymbol{d}^{(j)} = -\boldsymbol{g}^{(j)} + \zeta^{(j)}\boldsymbol{d}^{(j-1)} (j \in \mathbb{Z}^+, j \geq 1) \tag{3.77c}$$

$$\zeta^{(j)} = \frac{\|\boldsymbol{g}^{(j)}\|^2}{\|\boldsymbol{g}^{(j-1)}\|^2} (j \in \mathbb{Z}^+, j \geq 1) \tag{3.77d}$$

$$\hat{\boldsymbol{x}}^{(j+1)} = \hat{\boldsymbol{x}}^{(j)} + \eta^{(j)}\boldsymbol{d}^{(j)} \ . \tag{3.77e}$$

Once we compute the gradient using $\boldsymbol{Q}$ and $\boldsymbol{a}$, it is a trivial matter to get the other update values for either gradient descent or conjugate gradient except for the stepsize $\eta^{(j)}$. There are a number of ways to choose the stepsize. We can use a constant stepsize as we did for our continuous solver. But it is more efficient to do a line search to find the minimum value in the direction $\boldsymbol{d}^{(j)}$. In fact, for conjugate gradient methods, line searches are required in order to maintain $\boldsymbol{Q}$-conjugacy among descent directions. Note that this line minimization is occurring for the conjugate gradient step, not for the overall coordinate descent step.

It turns out that line searches are easy to obtain for quadratic functions, and a closed-form expression can be found. Given a descent direction $\boldsymbol{d}^{(j)}$, we wish to find $\eta^{(j)}$ to minimize our energy functional:

$$\eta^{(j)} = \arg\min_{\eta} E(\hat{\boldsymbol{x}}^{(j)} + \eta\boldsymbol{d}^{(j)}) \ . \tag{3.78}$$

A necessary and sufficient condition for the minimum is for the derivative in the direction

of $\boldsymbol{d}^{(j)}$ at $(\hat{\boldsymbol{x}}^{(j)} + \eta^{(j)}\boldsymbol{d}^{(j)})$ to be zero (this is true because E is strictly convex):

$$\begin{aligned}
\frac{\partial}{\partial\eta}E(\hat{\boldsymbol{x}}^{(j)} + \eta\boldsymbol{d}^{(j)}) &= E'(\hat{\boldsymbol{x}}^{(j)} + \eta\boldsymbol{d}^{(j)}; \boldsymbol{d}^{(j)}) \\
&= \nabla E(\hat{\boldsymbol{x}}^{(j)} + \eta\boldsymbol{d}^{(j)})^{\mathrm{T}}\boldsymbol{d}^{(j)} \\
&= [\boldsymbol{Q}(\hat{\boldsymbol{x}}^{(j)} + \eta\boldsymbol{d}^{(j)}) - \boldsymbol{a}]^{\mathrm{T}}\boldsymbol{d}^{(j)} \\
&= [\boldsymbol{Q}\hat{\boldsymbol{x}}^{(j)} - \boldsymbol{a}]^{\mathrm{T}}\boldsymbol{d}^{(j)} + \eta(\boldsymbol{d}^{(j)})^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{d}^{(j)} \\
&= (\boldsymbol{g}^{(j)})^{\mathrm{T}}\boldsymbol{d}^{(j)} + \eta(\boldsymbol{d}^{(j)})^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{d}^{(j)} = 0 \ .
\end{aligned}$$

Hence we conclude

$$\eta^{(j)} = -\frac{\langle \boldsymbol{g}^{(j)}, \boldsymbol{d}^{(j)}\rangle}{\|\boldsymbol{d}^{(j)}\|_{\boldsymbol{Q}}^2} \ . \tag{3.79}$$

In the case of gradient descent where $\boldsymbol{d}^{(j)} = -\boldsymbol{g}^{(j)}$, we see that

$$\eta^{(j)} = \frac{\|\boldsymbol{d}^{(j)}\|^2}{\|\boldsymbol{d}^{(j)}\|_{\boldsymbol{Q}}^2} \tag{3.80}$$

We can see that applying conjugate gradient requires minimally more computation than gradient descent. Most of the computation for both methods is involved in generating the gradient and the step size. There is also a minimal amount of extra storage overhead ($\boldsymbol{g}^{(j-1)}$ must be saved).

We can compare the descent direction we obtain in (3.76) to the direction we compute for the continuous case. For a b-step, we get the following equation in the discrete case:

$$\boldsymbol{d}^{(j)} = 2\lambda\hat{\boldsymbol{f}} \circ (\boldsymbol{y}_{\mathrm{S}} - \hat{\boldsymbol{b}}^{(j)} \circ \hat{\boldsymbol{f}}) - 2\alpha\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b\hat{\boldsymbol{b}}^{(j)} \ . \tag{3.81}$$

When $\boldsymbol{L}_b$ represents a gradient operator, we can write it as

$$\boldsymbol{L}_b = \begin{pmatrix} \boldsymbol{D}_x \\ \boldsymbol{D}_y \end{pmatrix} \tag{3.82}$$

where $\boldsymbol{D}_x$ is the x-derivative operator and $\boldsymbol{D}_y$ is the y-derivative operator. Both of the derivative matrices are skew symmetric, so $\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b = \boldsymbol{D}_x^{\mathrm{T}}\boldsymbol{D}_x + \boldsymbol{D}_y^{\mathrm{T}}\boldsymbol{D}_y = -(\boldsymbol{D}_x^2 + \boldsymbol{D}_y^2)$. This is the negative of the Laplacian operator. We compare this equation then to (3.52) which is the descent direction we obtain in the continuous case:

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = 2\lambda\hat{\varphi}(\psi_{\mathrm{S}} - \hat{\varphi}\beta) + 2\alpha\nabla^2\beta \ , \tag{3.83}$$

and we can clearly see that they are identical once the difference in notation is accounted for. Similarly, when $\boldsymbol{L}_b$ represents a Laplacian operator, the matrix is symmetric. Thus

$\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b = \boldsymbol{L}_b^2$. The Laplacian operator applied twice is simply the biharmonic operator. The equivalent gradient flow in the continuous case is (3.61):

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = 2\lambda\hat{\varphi}(\psi_{\mathrm{S}} - \hat{\varphi}\beta) - 2\alpha\nabla^4\beta \ . \tag{3.84}$$

This is also the same as the discrete case. Similar observations can be made for the f-step. Thus we see that there is no real difference between our continuous and discrete formulations.

We can convince ourselves that the gradient descent is doing the right thing from a qualitative analysis. We will do this analysis for an iteration on the b-step. We write our update equation here, combining (3.76) and (3.68):

$$\hat{\boldsymbol{b}}^{(j+1)} = \hat{\boldsymbol{b}}^{(j)} + 2\eta^{(j)}\left(\lambda\hat{\boldsymbol{F}}(\boldsymbol{y}_{\mathrm{S}} - \hat{\boldsymbol{F}}\hat{\boldsymbol{b}}^{(j)}) - \alpha\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b\hat{\boldsymbol{b}}^{(j)}\right) \ . \tag{3.85}$$

We assume for this example that $\boldsymbol{L}_b$ represents a gradient operator, so $\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b$ is the negative Laplacian. If we examine a point n that is in the middle of the image (so boundary effects do not complicate the analysis), we get the following update equation:

$$\begin{aligned}\hat{\boldsymbol{b}}^{(j+1)}[n] = \hat{\boldsymbol{b}}^{(j)}[n] &+ 2\eta^{(j)}\lambda\hat{\boldsymbol{f}}[n]\left(\boldsymbol{y}_{\mathrm{S}}[n] - \hat{\boldsymbol{f}}[n]\hat{\boldsymbol{b}}^{(j)}[n]\right) \\ &+ 2\eta^{(j)}\alpha(-4\hat{\boldsymbol{b}}^{(j)}[n] + \hat{\boldsymbol{b}}^{(j)}[n-1] + \hat{\boldsymbol{b}}^{(j)}[n+1] + \hat{\boldsymbol{b}}^{(j)}[n-M] + \hat{\boldsymbol{b}}^{(j)}[n+M]) \ .\end{aligned}$$

The first portion of the gradient is the estimation error for $\boldsymbol{y}_{\mathrm{S}}[n]$ scaled by the true image estimate. When $\hat{\boldsymbol{b}}^{(j)}[n]$ is too large (meaning $\hat{\boldsymbol{f}}[n]\hat{\boldsymbol{b}}^{(k)}[n] > \boldsymbol{y}_{\mathrm{S}}[n]$), this error is negative, and we will decrease the value of $\hat{\boldsymbol{b}}^{(j+1)}[n]$ relative to $\hat{\boldsymbol{b}}^{(j)}[n]$. The opposite happens when $\hat{\boldsymbol{b}}^{(j)}[n]$ is too small. The second part results from the regularization. We can see that the result is to do a low-pass filtering operation. We decrease the current point by a little and add in a little from all of the neighbors. Both of these effects are what we desire, and a stationary point is reached when they exactly counterbalance for every point in the image.

Using the Hessian

Newton's method does not provide any extra utility for solving quadratic problems. Even though it will converge in one iteration, Newton's method also requires inversion of the Hessian. This defeats the purpose of using iterative methods because the Hessian matrix is the $\boldsymbol{Q}$ matrix. So doing Newton's method simply solves (3.72), the necessary condition for the minimum. Quasi-Newton methods are also inappropriate.

They do not require computation of the inverse, but they do require storage of the inverse approximation which is not feasible for large matrices. Even a 256x256 image will produce a Hessian with 2^{32} elements which requires 32 GB of storage using double precision variables.

Even though Newton's method is not effective, there are ways that we can enhance the regular gradient-based methods by using some information from the Hessian. The average convergence rate for quadratic problems of gradient descent and, to a lesser extent, conjugate gradient is determined by the condition number of the $\boldsymbol{Q}$ matrix. In many instances, we can transform the original optimization problem into a new coordinate system:

$$\tilde{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{x} \tag{3.86}$$

In order to make the following analysis possible, $\boldsymbol{S}$ should be symmetric and invertible. We also define another matrix $\boldsymbol{H} = \boldsymbol{S}^2$.

This changes (3.65) into

$$\begin{aligned} E(\tilde{\boldsymbol{x}}) &= \tfrac{1}{2}\tilde{\boldsymbol{x}}\boldsymbol{S}^{-1}\boldsymbol{Q}\boldsymbol{S}^{-1}\tilde{\boldsymbol{x}} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}^{-1}\tilde{\boldsymbol{x}} + c && (3.87)\\ &= \tfrac{1}{2}\tilde{\boldsymbol{x}}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{a}}^{\mathrm{T}}\tilde{\boldsymbol{x}} + c && (3.88) \end{aligned}$$

Following the treatment by Bertsekas [8], we can generate a steepest descent update in the form of (3.76) for $\tilde{\boldsymbol{x}}$. Through appropriate manipulation, we can then write the updates directly in terms of $\boldsymbol{x}$ (this is often called scaled steepest descent):

$$\begin{aligned} \boldsymbol{d}^{(j)} &= -\boldsymbol{Q}\hat{\boldsymbol{x}}^{(j)} + \boldsymbol{a} && (3.89a)\\ \hat{\boldsymbol{x}}^{(j+1)} &= \hat{\boldsymbol{x}}^{(j)} + \eta^{(j)}\boldsymbol{H}^{-1}\boldsymbol{d}^{(j)} && (3.89b) \end{aligned}$$

We can do a similar analysis to produce conjugate gradient update equations:

$$\begin{aligned} \boldsymbol{g}^{(j)} &= \boldsymbol{Q}\hat{\boldsymbol{x}}^{(j)} - \boldsymbol{a} && (3.90a)\\ \boldsymbol{d}^{(0)} &= -\boldsymbol{H}^{-1}\boldsymbol{g}^{(0)} && (3.90b)\\ \boldsymbol{d}^{(j)} &= -\boldsymbol{H}^{-1}\boldsymbol{g}^{(j)} + \zeta^{(j)}\boldsymbol{d}^{(j-1)} (j \in \mathcal{J}, j > 1) && (3.90c)\\ \zeta^{(j)} &= \frac{\|\boldsymbol{g}^{(j)}\|^2_{\boldsymbol{H}^{-1}}}{\|\boldsymbol{g}^{(j-1)}\|^2_{\boldsymbol{H}^{-1}}} && (3.90d)\\ \hat{\boldsymbol{x}}^{(j+1)} &= \hat{\boldsymbol{x}}^{(j)} + \eta^{(j)}\boldsymbol{d}^{(j)} && (3.90e) \end{aligned}$$

We can still use (3.79) to compute $\eta^{(j)}$ because it finds the minimum along any line in any direction. This method is usually called preconditioned conjugate gradient.

Note that in both methods, we never need $\boldsymbol{S}$ directly. All computations are done in terms of $\boldsymbol{H}^{-1}$. The matrix $\boldsymbol{H}$ is referred to as a preconditioner. The maximal gain is provided when $\boldsymbol{H}$ closely approximates $\boldsymbol{Q}$. One important characteristic of $\boldsymbol{H}$ is that it should be easily invertible. If inverting $\boldsymbol{H}$ is as difficult as inverting $\boldsymbol{Q}$, then using a preconditioner does not benefit us. Due to the size of our problems, another condition we place on $\boldsymbol{H}$ is that either $\boldsymbol{H}$ or $\boldsymbol{H}^{-1}$ must be sparse (for storage purposes). This rules out other methods such as approximate Cholesky factorizations.

Even when $\boldsymbol{H}$ only loosely approximates $\boldsymbol{Q}$, significant speed increases can be achieved. A good example of this is simply letting $\boldsymbol{H}$ be a diagonal matrix with the diagonal entries equal to the diagonal entries of $\boldsymbol{Q}$. This can provide significant speedups in problems where the units in the parameter vector produce wildly different scaling. This scaling will be reflected in the $\boldsymbol{Q}$ matrix with a large condition number, and it is precisely this type of scenario that gradient descent has the most trouble with.

For our problem, the $\boldsymbol{Q}$ matrix is well scaled. For instance, $\boldsymbol{Q}_b$ is composed of the sum of $\lambda \boldsymbol{F}^2$ and $\alpha \boldsymbol{L}_b^{\mathrm{T}} \boldsymbol{L}_b$. In a typical prostate example, the ratio between the largest entry of $\lambda \boldsymbol{F}^2$ and an entry on the diagonal of $\alpha \boldsymbol{L}_b^{\mathrm{T}} \boldsymbol{L}_b$ is approximately 0.025. This means that there is not much fluctuation on the diagonal of $\boldsymbol{Q}$. In other applications, this ratio is actually lower because most coils have smoother sensitivity profiles than the endorectal coil. This results in high α values because the smoothness means that large derivatives should be penalized more heavily.

If we add entries to $\boldsymbol{H}$ from the subdiagonals of $\boldsymbol{Q}$ directly adjacent to the main diagonal, we can generate tridiagonal or quindiagonal matrices. These linear systems can be solved in order $\mathcal{O}(MN)$. Because we form our vectors by stacking the elements of our images columnwise, the subdiagonals directly adjacent to the diagonal represent the interactions of a pixel with the pixels directly above and below it (except at the image boundaries where the vector wraps around). It may be possible that if we alternate between horizontal and vertical stacking (and thus the interactions that get included in our tri- and quindiagonal matrices) when generating our matrices, this could increase the effectiveness of our preconditioning.

We know our matrices are sparse and banded. The majority of the entries of the first two subdiagonals are non-zero, and a very large percentage of the entries are located within the first few subdiagonals. If our linear operator $\boldsymbol{L}$ represents a Laplacian operator, the kernel of $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$ has 13 non-zero elements. Both of the first and second subdiagonals will be mostly filled, so we can say that a tridiagonal matrix will incorporate

approximately 3/13 (23%) of the entries, and a quindiagonal matrix will have about 5/13 (38%) of the entries. If we have $\boldsymbol{L}$ represent a gradient operator and we implement $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$ to avoid the checkerboard effect, then the kernel will have 5 non-zero elements, and a tridiagonal matrix will have 60% of the entries of $\boldsymbol{Q}$. The quindiagonal matrix provides no additional benefits. Thus we expect the preconditioners to work better with gradient regularizers because we can incorporate more of the matrix information into the preconditioner. We will investigate the performance of these preconditioners in a later section.

■ 3.3.5 Convergence

There are no general convergence properties for coordinate descent because it is a very loosely defined concept. For our particular problem, we can show that we possess some convergence qualities similar to the Expectation-Maximization (EM) algorithm [22].

The EM algorithm is a popular technique to solve missing- or hidden-data problems and is similar in flavor to coordinate descent. It does ML estimation over an expected log likelihood of the complete data. One of its nice features is that the likelihood of the data increases monotonically with each iteration. This means that it is guaranteed to converge to a local maximum (with certain technical assumptions). We can make a similar claim about the discrete form of our algorithm. As we have noted, each separate f- and b-step is convex. Assume that we begin with a f-step. Then we can say

$$\hat{\boldsymbol{f}}^{(i+1)} = \arg\min_{\boldsymbol{f}} E(\boldsymbol{f}, \hat{\boldsymbol{b}}^{(i)}) \quad \rightarrow \quad E(\hat{\boldsymbol{f}}^{(i+1)}, \hat{\boldsymbol{b}}^{(i)}) \leq E(\hat{\boldsymbol{f}}^{(i)}, \hat{\boldsymbol{b}}^{(i)}) \tag{3.91}$$

$$\hat{\boldsymbol{b}}^{(i+1)} = \arg\min_{\boldsymbol{b}} E(\hat{\boldsymbol{f}}^{(i+1)}, \boldsymbol{b}) \quad \rightarrow \quad E(\hat{\boldsymbol{f}}^{(i+1)}, \hat{\boldsymbol{b}}^{(i+1)}) \leq E(\hat{\boldsymbol{f}}^{(i+1)}, \hat{\boldsymbol{b}}^{(i)}) \ . \tag{3.92}$$

Thus we can conclude that

$$E(\hat{\boldsymbol{f}}^{(i+1)}, \hat{\boldsymbol{b}}^{(i+1)}) \leq E(\hat{\boldsymbol{f}}^{(i)}, \hat{\boldsymbol{b}}^{(i)}) \ , \tag{3.93}$$

and thus each iteration of coordinate descent monotonically decreases the value of our energy functional. There is a form of EM called generalized EM where the M-step does not require finding the exact maximum of the expected likelihood. Instead the same monotonicity property can be shown to occur for an M-step that simply increases the value of the likelihood. In a similar manner, because we run descent algorithms with exact line searches, each iteration in gradient descent or conjugate gradient is guaranteed to decrease the value of our energy functional (unless we are already at a local minimum).

It makes sense in our algorithm to not let the solvers converge to the exact solution in either the f- or b-step, at least early on. Because we are solving with an incorrect $\hat{\boldsymbol{f}}$ or $\hat{\boldsymbol{b}}$, finding the exact minimum may in fact be driving us further from the correct answer than a result we could have obtained by stopping after a few iterations. It is difficult to say exactly at what point we should stop, but we can exploit this observation by having fairly loose convergence bounds at first and then gradually tightening them.

■ 3.4 Extensions

So far we have presented results for correcting a single surface coil image while using ℓ_2 norms to regularize the solution. There are a number of scenarios for which we can extend our framework. Due to the presence of edges in the underlying intrinsic image, we know that the probability distribution of its derivatives tend to be heavy-tailed and are not well approximated by Gaussians (and hence not well modeled using ℓ_2 norms). Therefore we explore the use of ℓ_p norms which are more forgiving of high gradient values. We can also exploit the spatial structure in the third dimension to produce even better results by doing a full 3D implementation. There are also many imaging techniques where multiple surface coils are used to receive the same MR signal. Traditional techniques use *ad hoc* methods to fuse these multiple observations into one surface coil image. We present a method that optimally joins the data in a way that minimizes distortions from the bias field. There are also situations where we capture different images using identical surface coil configurations. We can produce better bias field estimates by using our multiple observations. One thing that all of these techniques have in common is that they all come with additional computational cost. We thus present a simple coarse-to-fine multigrid implementation of our algorithm that allows us to significantly reduce computation time by exploiting the multiscale structure of our data.

■ 3.4.1 Discrete Half-Quadratic Solution

When $p \neq 2$, our optimization problem becomes non-quadratic, and thus the derivative results in a non-linear condition for a minimum. We will only address this problem for regularization on $\boldsymbol{f}$. A ℓ_2 penalty provides an appropriate fit for $\boldsymbol{b}$. There has been a great deal of work in the literature about solving optimization problems with ℓ_p regularization. Some of this work was discussed in Section 2.1.4. We are primarily

concerned with the case when $p = 1$. This is the largest choice of p that admits step discontinuities while also being the smallest choice that remains convex. The derivative operator of interest to us is the gradient. This regularization scheme is simply total variation (TV) regularization [63]. It is a common model for signals that are piecewise constant. It penalizes non-constant regions and does not overly penalize edges.

We write our complete energy functional for ℓ_1 regularization on $\boldsymbol{f}$ here for convenience:

$$E(\boldsymbol{f}, \boldsymbol{b}) = \|\boldsymbol{y}_{\mathrm{B}} - \boldsymbol{k}\boldsymbol{f}\|^2 + \lambda\|\boldsymbol{y}_{\mathrm{S}} - \boldsymbol{b} \circ \boldsymbol{f}\|^2 + \alpha\|\boldsymbol{L}_b\boldsymbol{b}\|_2^2 + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|_1^1 \ . \tag{3.94}$$

The ℓ_1 norm is non-differentiable at zero. Thus we will use a smoothed version of the ℓ_1 norm:

$$\|\boldsymbol{x}\|_1^1 \approx \sum_i \sqrt{\boldsymbol{x}^2[i] + \xi} \ . \tag{3.95}$$

As $\xi \to 0$, the approximation approaches the unsmoothed norm.

The optimization problem for the b-step remains identical to that described in Section 3.3.4. The f-step becomes more complicated because it can no longer be written in a quadratic form. Once we use the smoothed approximation to the ℓ_1 norm, we can compute gradients and perform gradient descent. Our energy functional remains convex, so gradient descent will converge to a global minimum on each f-step. But gradient descent is not efficient. We can no longer do exact line searches in closed form, so we must apply one of the many inexact approximations available in the literature or simply use fixed step sizes. Vogel and Oman demonstrate in [78] that gradient descent is very slow compared with half-quadratic optimization which we introduced in Section 2.1.4.

As we noted in Section 2.1.4, there has been a great deal of work with non-linear image reconstruction. One method of solving ℓ_p minimization problems that has become quite popular is known as half-quadratic regularization [16,26,54]. Half-quadratic techniques can be used to minimize the following type of energy functional:

$$E(\boldsymbol{f}) = \|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{f}\|^2 + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|_p^p \ . \tag{3.96}$$

In order to fit our observation model into this framework, we observe that the following energy functional attains its minimum for the same argument as (3.96):

$$E(\boldsymbol{f}) = -2(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{z})^{\mathrm{T}}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})\boldsymbol{f} + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|_p^p \ . \tag{3.97}$$

We can see that (3.94) fits this form with the following relations (and $p = 1$):

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = k^2\boldsymbol{I} + \lambda\hat{\boldsymbol{B}}^2 \tag{3.98a}$$

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{z} = k\boldsymbol{y}_{\mathrm{B}} + \lambda\hat{\boldsymbol{B}}\boldsymbol{y}_{\mathrm{S}} \ . \tag{3.98b}$$

In order to implement this minimization, we use the multiplicative form of half-quadratic regularization. We form a weighting matrix $\boldsymbol{W}^{(j)}$ at each half-quadratic iteration which is a diagonal matrix with the following entries along the diagonal:

$$[\boldsymbol{W}^{(j)}]_{l,l} = \frac{1}{\sqrt{((\boldsymbol{L}_f\hat{\boldsymbol{f}}^{(j-1)})[l])^2 + \xi}} \ , \tag{3.99}$$

and we can write the following local quadratic approximation for E:

$$E^{(j)}(\boldsymbol{f}) = -2\boldsymbol{z}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})\boldsymbol{f} + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|^2_{\boldsymbol{W}^{(j)}} \ . \tag{3.100}$$

This weights the ℓ_2 norm less in regions with a high gradient. This is what produces the edge-preserving effect. To see why this is equivalent to minimizing (3.96), we examine the stationary point. Let $\boldsymbol{f}^*$ be the vector that minimizes (3.96), $\boldsymbol{W}^*$ be the weighting matrix generated from $\boldsymbol{f}^*$, and $\xi \to 0$. Then we see the following:

$$\begin{aligned}
\|\boldsymbol{L}_f\boldsymbol{f}^*\|^2_{\boldsymbol{W}^*} &= \sum_l [\boldsymbol{W}^*]_{l,l}((\boldsymbol{L}_f\boldsymbol{f}^*)[l])^2 \\
&= \sum_l \frac{((\boldsymbol{L}_f\boldsymbol{f}^*)[l])^2}{\sqrt{((\boldsymbol{L}_f\boldsymbol{f}^*)[l])^2 + \xi}} \\
&\approx \sum_l \frac{((\boldsymbol{L}_f\boldsymbol{f}^*)[l])^2}{|(\boldsymbol{L}_f\boldsymbol{f}^*)[l]|} \\
&= \sum_l |(\boldsymbol{L}_f\boldsymbol{f}^*)[l]| \ .
\end{aligned}$$

The last line is the ℓ_1 norm of $\boldsymbol{L}_f\boldsymbol{f}^*$, so the weighted-$\ell_2$ approximation holds when $\boldsymbol{f} = \boldsymbol{f}^*$. This simply shows that a fixed point for (3.100) is also a fixed point for the half-quadratic iteration. More generally, the weighted-ℓ_2 approximation holds at the value of $\boldsymbol{f}$ that is used to generate $\boldsymbol{W}$. Thus for each half-quadratic iteration when using an iterative solver, we begin with the exact ℓ_1 energy functional, and then our approximation gets progressively worse. For convergence properties, we refer the reader to [26].

The quadratic approximation achieves a minimum for the following condition:

$$\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} + \gamma\boldsymbol{L}_f^{\mathrm{T}}\boldsymbol{W}^{(j)}\boldsymbol{L}_f\right)\hat{\boldsymbol{f}}^{(j)} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{z} \ . \tag{3.101}$$

This is a positive definite linear system. It can be solved directly, but as was the case for the ℓ_2-norm problem, it is computationally inefficient to do so. We can again apply iterative techniques such as preconditioned conjugate gradient to find approximate solutions.

Note that when using this half-quadratic technique, our algorithm expands from two nested loops to three nested loops. There is the overall coordinate descent loop which alternates between f- and b-steps. Then the half-quadratic method also uses a nested pair of iterations: steps where we generate a new weighting matrix $\boldsymbol{W}^{(j)}$ to form our local quadratic approximation and then iterations to generate approximate solutions. In general we will only run the half-quadratic regularization for one iteration per f-step. There just is not much additional utility to be gained by letting it run longer, especially as we near convergence. So even when using a ℓ_1 norm on $\hat{\boldsymbol{f}}$, we can still implement both the f- and b-steps as conjugate gradient on a linear equation.

This ℓ_1 norm applied in our estimation framework will produce different results from simply minimizing the following energy functional:

$$E(\boldsymbol{f}) = \|\hat{\boldsymbol{f}} - \boldsymbol{f}\|^2 + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|_1^1 \tag{3.102}$$

where $\hat{\boldsymbol{f}}$ is the estimate of $\boldsymbol{f}^*$ we obtain by minimizing (3.94) with $\gamma = 0$. This would be equivalent to simply applying an anisotropic post-processing filter to the output of our algorithm. The reason for this difference is that this formulation does not properly account for the spatially-varying noise levels in $\hat{\boldsymbol{f}}$. If we change the data fidelity term in (3.102) to a weighted-ℓ_2 norm, we end up with this equation:

$$E(\boldsymbol{f}) = \|\hat{\boldsymbol{f}} - \boldsymbol{f}\|_{\boldsymbol{U}}^2 + \gamma\|\boldsymbol{L}_f\boldsymbol{f}\|_1^1 \ . \tag{3.103}$$

The minimum $\boldsymbol{f}$ for (3.103) is the same as the minimum $\boldsymbol{f}$ for (3.96) when the weighting matrix is diagonal with the following entries along the diagonal:

$$[\boldsymbol{U}]_{l,l} = k^2 + \lambda\hat{\boldsymbol{b}}[l]^2 \ , \tag{3.104}$$

and $\hat{\boldsymbol{f}}$ is constructed optimally from $\boldsymbol{y}_{\mathrm{B}}$ and $\boldsymbol{y}_{\mathrm{S}}$ using $\hat{\boldsymbol{b}}$ and (3.74). This post-processing still is not equivalent to using a ℓ_1 prior in the full minimization problem because it does not allow $\hat{\boldsymbol{b}}$ to adjust based on the regularized $\hat{\boldsymbol{f}}$.

We obtain the maximum benefit of using (3.96) or (3.103) instead of (3.102) in regions close to the surface coil. We know that the SNR is very high in this region so we are more inclined to believe the unfiltered results than we would otherwise. In

regions far from the surface coil when we have a coil response that dies to zero, there is no real benefit to the integrated ℓ_1 regularization. In fact, in these regions the $\hat{\boldsymbol{f}}$ we would obtain without regularization is simply $\boldsymbol{y}_{\mathrm{B}}$, so the reconstruction using the full ℓ_1 method is in fact just ℓ_1 reconstruction from $\boldsymbol{y}_{\mathrm{B}}$. When we fully incorporate the regularization on $\hat{\boldsymbol{f}}$ into our algorithm, these transitions between high-noise and low-noise regions are handled seamlessly and automatically.

■ 3.4.2 Higher dimensions

One of the major features of MR imaging is the ability to get data in three dimensions. We get a series of parallel 2D planes, usually separated by a constant slice thickness. We can, of course, estimate the bias field individually on each slice. This will be sub-optimal because the smoothness of the bias field and the piecewise-smoothness of the true image exists in all three spatial dimensions. So we are not using all of the information available to us. In addition, a 2D implementation cannot hope to correct any spatial inhomogeneities in the z-direction. These interslice inhomogeneities will make it difficult to do 3D processing of the corrected volumes.

There are no theoretical complications with extending our algorithm to three dimensions. Let our volume have dimension MxNxO. For the continuous case, the PDEs that we obtain with the Euler-Lagrange equation remain identical except the differential operators act in 3D instead of 2D. Similarly in the discrete case, everything has been rewritten in terms of observation vectors. There is no underlying assumptions about how our sampling lattices were constructed (*e.g.*, we could have indexed the samples in a consistently random order and the same solution would be found). The energy functional remains in the same form as specified in Section 3.3.4, and we can use the exact same linear solver. We need to place the elements of our 3D volume into a vector in a consistent manner. From that point, all that needs to be done is to change our 2D convolutional kernel into a 3D kernel and make the appropriate change in the linear operator matrix. The linear operator matrix remains sparse and banded.

The major issue with a full 3D implementation is computational. These include additional strains on both storage and processing time. MR volumes can be as large as $512 \times 512 \times 400$. In order to do 3D processing, we must have the full volume loaded in memory. With double precision storage, this would require 800 MB just for one vector of length MNO. To execute our algorithm, we need storage many times greater than just one vector. An additional problem has to do with computation time. Our

0	0	0
0	r^2	0
0	0	0

-1

0	1	0
1	-4-2r^2	1
0	1	0

0

0	0	0
0	r^2	0
0	0	0

+1

Table 3.5. Kernel for a 3D Laplacian operator. The three tables contain the entries for $z = -1, 0, +1$ from left to right.

solver does not scale linearly with problem size. As a rough approximation, we observe computation time to be $\mathcal{O}(n^2)$: computation per iteration is $\mathcal{O}(n)$ and the number of iterations is approximately $\mathcal{O}(n)$. Thus if we have a volume with 30 slices, computation time will increase by approximately a factor of 30 for full 3D processing versus 2D processing of each slice.

Another issue that must be contended with is the aspect ratio. We define the aspect ratio r as the ratio of the voxel length in the x- and y-directions to the voxel length in the z-direction. The 3D voxels usually are not cubes. Generally in MR, $r < 1$ because the slice thickness is greater than the dimensions in the imaging plane. This means that in order to make our 3D filter isotropic in continuous space, it must be anisotropic in the discrete space. In Table 3.5, we give an example of a Laplacian kernel for an image with aspect ratio r. Other kernels are similarly generalized. In the context of our energy functional, when $r < 1$, this anisotropic kernel penalizes change in the z-direction less than it penalizes change in the x- or y-directions (change as measured on our discrete samples).

Another case where we can couple data across slices is with 2D heart time-sequence data. These images consist of multiple 2D images of the heart at different times in the cardiac cycle. There are several ways we can handle this case. The most straightforward method would simply be to capture both surface coil and body coil images at each time step and then apply the correction to each slice. It may be helpful to somehow couple the bias field estimates from each slice to achieve better estimates. One way to accomplish this is to enforce a ℓ_p penalty for deviations in the bias field from slice to slice. This would force the bias field to change slowly with time. Another way to accomplish this would be to simply stack the time sequence data into a 3D vector. We could then use a volumetric correction scheme as was detailed above.

A more enticing method would capture surface coil images at each time step but only one body coil image. We can estimate a sensitivity profile from the time step with both body coil and surface coil images. To correct the other images, we could apply the bias field estimate directly, but this may be suboptimal as the field may shift in time as the anatomy moves. We could attempt to minimize this effect by putting the bias field through a transformation for better alignment. If this proves insufficient, we could capture body coil images at a subset of the time steps. We could then interpolate between our "known" bias fields to produce sensitivity profiles in the intermediate slices.

■ 3.4.3 Multiple Bias Fields and Intrinsic Images

So far we have dealt with applications where we have one surface coil image and one body coil image, and we want to recover the true MR image along with the bias field. There are several scenarios where we can extend our bias correction framework to be more flexible. Among these are multiple surface coil imaging; multiple pulse sequence imaging; and time sequence imaging. We will only derive results in this section for the discrete case. Results for the continuous case can be similarly obtained.

We have N_f intrinsic images to estimate and N_b bias fields. We denote each intrinsic image as $\boldsymbol{f}_n^*$ and each bias field as $\boldsymbol{b}_m^*$. We define an ordered set $\mathcal{F}^* = \{\boldsymbol{f}_n^*\}_{n=1}^{N_f}$ which contains all of the intrinsic images and an ordered set $\mathcal{B}^* = \{\boldsymbol{b}_m^*\}_{m=1}^{N_b}$ which contains all of the bias fields. In this section, we find it useful to exploit the concept of graphical models [36]. Graphical models are a way of representing independence between random variables. Nodes in the graph represent random variables and edges in the graph represent possible dependence between the two nodes. The lack of an edge between two nodes does not mean that the two nodes are independent. There can still be indirect dependence due to coupling with other variables. Let there be two nodes $\boldsymbol{x}$ and $\boldsymbol{y}$. We define a cut set as a subset of nodes $\mathcal{G}$ such that the removal of $\mathcal{G}$ completely severs the graph between $\boldsymbol{x}$ and $\boldsymbol{y}$ (this generalizes to more than two nodes). Then we say that $\boldsymbol{x}$ and $\boldsymbol{y}$ are conditionally independent given $\mathcal{G}$. Using graphical models give us insight into how our coordinate descent approach makes sense as well as the consequences of our modeling decisions.

Multiple coils, one intrinsic image

In many surface coil imaging applications (including the prostate, heart, spine, etc.), there are actually multiple surface coils present. The multiple coils are used due to

the typically sharp drop off in sensitivity far away from the coil. By distributing the coils spatially, we can achieve better signal coverage in the FOV. The coils are usually designed so that they are uncoupled (the mutual inductance between the coils is zero). The images received by the surface coils are combined to generate the final surface coil image. For instance, the prostate images that we use have five surface coils: the endorectal coil and four additional coils arranged as a pelvic phased array (PPA). The composite surface coil image that we receive from the MR machine as output is generated from data from the five coils in an *ad hoc* manner. Rather than simply performing bias correction on the composite image, we can estimate each coil profile separately and fuse the resulting images together in a more consistent fashion.

We introduce a new measurement model for the N_b surface coil case. We receive one body coil image and N_b surface coil images:

$$\boldsymbol{y}_{\mathrm{B}} = k\boldsymbol{f}^* + \boldsymbol{n}_{\mathrm{B}} \tag{3.105}$$

$$\boldsymbol{y}_{\mathrm{S},i} = \boldsymbol{b}_i^* \circ \boldsymbol{f}^* + \boldsymbol{n}_{\mathrm{S},i} \tag{3.106}$$

$$i \in \{1, 2, \ldots, N_b\} \ .$$

Roemer *et al.* [62] discuss a number of techniques to combine the images received simultaneously from multiple uncoupled surface coils. The simplest method would be to simply sum them electrically and view the composite signal as the superposition of the induced signal in each individual coil. Due to the linearity of the Fourier transform, this is equivalent to converting each image from k-space to the image domain and then summing the complex images. This method is problematic due to the variable spatially-dependent phase for each coil. Working directly with the complex images provides some benefits. In some applications, the phase of the MR signal is important. Another important use of this technique is due to machine limitations. MR machines can only receive input from a finite number of coils. Thus if we wish to use many coils, we may need to combine signals before they reach the processing system. Roemer *et al.* also discuss a technique that combines the images using the sensitivity profiles of each coil to produce an image that is optimal from a SNR standpoint. This method is basically a weighted sum (using the sensitivity profiles as a weight) with appropriate phase shifting.

We can avoid the phase issues by working with the magnitude images. This does not severely hamper us because we are only interested in the magnitude. It is fairly easy to convert the complex reconstruction into a magnitude reconstruction. This again

requires the sensitivity profiles of all of the coils. Using the magnitude images hurts us from a SNR perspective because the noise that we observe in each coil is actually not completely uncorrelated. Working in the complex domain allows us to exploit these correlations. A common technique when these sensitivity profiles are not known is to combine the individual magnitude surface coil images into one image using ℓ_p norms:

$$\boldsymbol{y}_{\mathrm{S}}[n] = \left(\sum_{i=1}^{N_b} |\boldsymbol{y}_{\mathrm{S},i}[n]|^p \right)^{1/p} . \tag{3.107}$$

When $p = 1$, this is simply summing the magnitude images. When $p = 2$, this is referred to as the sum of squares method. If we set $p = \infty$, this simply chooses the largest intensity value available from any of the surface coils. In all but the last case, the overall noise in the composite image is being increased. In order for the SNR to improve, the signal level in one or more of the coils must be much greater than the noise level.

The sum of squares method is a commonly used technique to join images together. It can be viewed as using the optimal reconstruction for magnitude images and applying the image intensity values as a proxy for the sensitivity profile. If the complex Gaussian noise processes that contaminate all of the surface coil images are identical, then the noise here will remain Rician. If we simply summed the magnitude images, the noise is much more complex because it would be the addition of Rician random variables. This means that the PDF would be the convolution of multiple Rician PDFs which does not have a simple description. So at least in this respect, the sum of squares method is more attractive than the sum of magnitude method. Nonetheless, this is not ideal because we can do better once we have the individual coil profile estimates.

Once all of the surface coil measurements have been combined into one image, we only need to estimate one bias field instead of N_b bias fields. This is a win computationally, but it does hamper us in some respects. For instance, α must be chosen so that our solver admits the strongest derivatives present in $\boldsymbol{b}^*$. So if our coils have vastly different spatial structure, then we must choose α to admit the sharpest transitions present in any of the coil profiles. For example, in the prostate, the endorectal coil has a very sharp intensity drop off while the PPA coils have much smoother properties. So we should choose α based on the characteristics of the endorectal coil. This works well in the regions dominated by the endorectal coil, but far away, α will be set too low. We tend to get a lot of tissue structure in the bias field estimates in those locations.

Additionally, we can not only choose individual regularization parameter strengths, we can also choose linear filters for each coil to pick out desired features. So an additional advantage of processing the coils separately is that we can individually tune the regularization for each coil.

We modify (3.16) to be compatible with our new model:

$$E(\mathcal{F},\mathcal{B}) = E_{\mathrm{B}}(\boldsymbol{f}) + \sum_{i=1}^{N_b} \lambda_i E_{\mathrm{S},i}(\boldsymbol{f},\boldsymbol{b}_i) + \sum_{m=1}^{N_b} \alpha_m \mathcal{R}_{b,m}(\boldsymbol{b}_m) + \gamma \mathcal{R}_f(\boldsymbol{f}) \ . \tag{3.108}$$

We can choose the functions for (3.108) in an analogous manner to the single coil case:

$$E_{\mathrm{B}}(\boldsymbol{f}) = \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}\|^2 \tag{3.109a}$$

$$E_{\mathrm{S},i}(\boldsymbol{f},\boldsymbol{b}_i) = \|\boldsymbol{y}_{\mathrm{S},i} - \boldsymbol{b}_i \circ \boldsymbol{f}\|^2 \tag{3.109b}$$

$$\mathcal{R}_{b,m}(\boldsymbol{b}_m) = \|\boldsymbol{L}_{b,m}\boldsymbol{b}_m\|^2 \tag{3.109c}$$

$$\mathcal{R}_f(\boldsymbol{f}) = \|\boldsymbol{L}_f\boldsymbol{f}\|_p^p \ . \tag{3.109d}$$

This problem is quadratic for each of the individual bias field estimates with the following equations:

$$\boldsymbol{Q}_{b,m} = 2\lambda_m \hat{\boldsymbol{F}}^2 + 2\alpha_m \boldsymbol{L}_{b,m}^{\mathrm{T}} \boldsymbol{L}_{b,m} \tag{3.110a}$$

$$\boldsymbol{a}_{b,m} = 2\lambda_m \hat{\boldsymbol{F}} \boldsymbol{y}_{\mathrm{S},m} \ . \tag{3.110b}$$

Note that these are the exact same equations we would have if we only observed one surface coil image. In fact, given $\hat{\boldsymbol{f}}$ and all of our observation images, all of our bias field estimates $\hat{\boldsymbol{b}}_m$ are conditionally independent. So we can compute $\hat{\boldsymbol{b}}_m$ sequentially or in parallel and receive the same results.

The problem for estimating the intrinsic image is quadratic in terms of $\boldsymbol{f}$ except for the ℓ_p regularization term. We can put the f-step in the same form as (3.97):

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = k^2\boldsymbol{I} + \sum_{i=1}^{N_b} \lambda_i \hat{\boldsymbol{B}}_i^2 \tag{3.111a}$$

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{z} = k\boldsymbol{y}_{\mathrm{B}} + \sum_{i=1}^{N_b} \lambda_i \hat{\boldsymbol{B}}_i \boldsymbol{y}_{\mathrm{S},i} \ . \tag{3.111b}$$

We only have these changes in $\boldsymbol{H}$ and $\boldsymbol{z}$, otherwise we can apply the half-quadratic solver in the exact same manner that we described in Section 3.4.1. In this way we continue to use coordinate descent to iterate between b-steps and f-steps. On the b-step, all of

the $\boldsymbol{b}_m$'s are independent given $\hat{\boldsymbol{f}}$ so we can compute the estimates individually. On the f-step, we use all of our $N_b + 1$ observation images and all N_b bias field estimates to construct an estimate of $\boldsymbol{f}^*$.

For the special case when $\gamma = 0$, there is no spatial coupling in $\hat{\boldsymbol{f}}$ and we get the following solution:

$$\hat{\boldsymbol{f}}[n] = \frac{k\boldsymbol{y}_{\mathrm{B}}[n] + \sum_{i=1}^{N_b} \lambda_i \hat{\boldsymbol{b}}_i[n]\boldsymbol{y}_{\mathrm{S},i}[n]}{k^2 + \sum_{i=1}^{N_b} \lambda_i \hat{\boldsymbol{b}}_i^2[n]} \quad . \tag{3.112}$$

This results in the output being a noise-weighted convex combination of $\boldsymbol{y}_{\mathrm{B}}/k$ and all of the $\boldsymbol{y}_{\mathrm{S},i}/\hat{\boldsymbol{b}}_i$ images which is analogous to what we noted in Section 3.3.3 for the single surface coil case. Thus the SNR will be at least as large as the best SNR from any of the observation images and can be as much as $10\log_{10}(N_b + 1)$ dB better. Except in rare circumstances, processing the multiple surface coil images in this manner will be superior to fusing the surface coil images into one image and then performing bias correction on that.

We note that (3.112) is the same as the results derived by Roemer *et al.* for the case when the sensitivity profile is known and noise correlations among the receivers are ignored. The noise variances are simply the diagonal entries of their noise correlation matrix. We set the bias field of the body coil uniformly to one to arrive at our equation. We have arrived at the same results using similar but different approaches. Roemer *et al.* wish to maximize the SNR while we wish to minimize the estimation error. Due to our linear Gaussian approximation, the estimation error variance can be considered equivalent to the noise variance. Because we are constructing bias corrected images, the signal level is a constant, and maximizing SNR and minimizing estimation error become the same.

Multiple intrinsic images, one surface coil

It is common practice to collect images of the same location in the body and the same coil configuration but using different pulse sequences (*e.g.*, T_1-*weighted* and T_2-*weighted* images). This allows different features to become visible. For instance, in prostate imaging, T_1-*weighted* images are very good at providing a homogeneous intensity within the entire prostate while T_2-*weighted* images allow differentiation between internal structures in the prostate. A similar application is time sequence imaging of a moving object such as the heart. Assuming that the movement of the object does not greatly disturb the location of the coils, we have multiple intrinsic images with one sur-

face coil configuration. Let N_f be the number of intrinsic images we wish to estimate. We must also acquire an identical number of surface coil images. Each image has the same coil configuration and bias field (ignoring tissue dependent effects) but different intrinsic images. We would like to be able to acquire only one body coil image and use that to correct all of our surface coil images. Without loss of generality, we assign $\boldsymbol{f}_1$ to correspond to the intrinsic image in the body coil image. We express our imaging model in this way:

$$\boldsymbol{y}_{\mathrm{B}} = k\boldsymbol{f}_1 + \boldsymbol{n}_{\mathrm{B}} \tag{3.113}$$

$$\boldsymbol{y}_{\mathrm{S},i} = \boldsymbol{b} \circ \boldsymbol{f}_i + \boldsymbol{n}_{\mathrm{S},i} \ . \tag{3.114}$$

We can rewrite (3.16) as

$$E(\mathcal{F}, \mathcal{B}) = E_{\mathrm{B}}(\boldsymbol{f}_1) + \sum_{i=1}^{N_b} \lambda_i E_{\mathrm{S},i}(\boldsymbol{f}_i, \boldsymbol{b}) + \alpha \mathcal{R}_b(\boldsymbol{b}) + \sum_{n=1}^{N_f} \gamma_n \mathcal{R}_{f,n}(\boldsymbol{f}_n) \ . \tag{3.115}$$

This model naturally leads to the following choices for (3.115):

$$E_{\mathrm{B}}(\boldsymbol{f}_1) = \|\boldsymbol{y}_{\mathrm{B}} - k\boldsymbol{f}_1\|^2 \tag{3.116a}$$

$$E_{\mathrm{S},i}(\boldsymbol{f}_i, \boldsymbol{b}) = \|\boldsymbol{y}_{\mathrm{S},i} - \boldsymbol{b} \circ \boldsymbol{f}_i\|^2 \tag{3.116b}$$

$$\mathcal{R}_b(\boldsymbol{b}) = \|\boldsymbol{L}_b \boldsymbol{b}\|^2 \tag{3.116c}$$

$$\mathcal{R}_{f,n}(\boldsymbol{f}_n) = \|\boldsymbol{L}_{f,n} \boldsymbol{f}_n\|_p^p \ . \tag{3.116d}$$

The energy functional is quadratic in terms of $\boldsymbol{b}$ with the following choices:

$$\boldsymbol{Q}_b = 2\sum_{i}^{N_f} \lambda_i \boldsymbol{F}_i^2 + 2\alpha \boldsymbol{L}_b^{\mathrm{T}} \boldsymbol{L}_b \tag{3.117}$$

$$\boldsymbol{a}_b = 2\sum_{i}^{N_f} \lambda_i \boldsymbol{F}_i \boldsymbol{y}_{\mathrm{S},i} \ . \tag{3.118}$$

We can again put the energy functional for $\boldsymbol{f}_1$ into the form of (3.97):

$$\boldsymbol{H}_1^{\mathrm{T}} \boldsymbol{H}_1 = k^2 \boldsymbol{I} + \lambda_1 \hat{\boldsymbol{B}}^2 \tag{3.119}$$

$$\boldsymbol{H}_1^{\mathrm{T}} \boldsymbol{z}_1 = k\boldsymbol{y}_{\mathrm{B}} + \lambda_1 \hat{\boldsymbol{B}} \boldsymbol{y}_{\mathrm{S},1} \tag{3.120}$$

and similarly the energy functionals for $\boldsymbol{f}_n$, $n \neq 1$:

$$\boldsymbol{H}_n^{\mathrm{T}} \boldsymbol{H}_n = \lambda_n \hat{\boldsymbol{B}}^2 \tag{3.121}$$

$$\boldsymbol{H}_n^{\mathrm{T}} \boldsymbol{z}_n = \lambda_n \hat{\boldsymbol{B}} \boldsymbol{y}_{\mathrm{S},n} \ . \tag{3.122}$$

We note that if $\gamma_n = 0 \forall n$, $\hat{\boldsymbol{f}}_1 = (\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{H}_1)^{-1}\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{y}_1$ which can be solved pointwise:

$$\hat{\boldsymbol{f}}_1[n] = \frac{k\boldsymbol{y}_{\mathrm{B}}[n] + \lambda_1\hat{\boldsymbol{b}}[n]\boldsymbol{y}_{\mathrm{S},1}[n]}{k^2 + \lambda_1\hat{\boldsymbol{b}}^2[n]} \ . \tag{3.123}$$

This is the solution for the single surface coil and single body coil case if we ignore the other surface coil images. All of the other $\hat{\boldsymbol{f}}_n = (\boldsymbol{H}_n^{\mathrm{T}}\boldsymbol{H}_n)^{-1}\boldsymbol{H}_n^{\mathrm{T}}\boldsymbol{z}_n = \boldsymbol{B}^{-1}\boldsymbol{y}_{\mathrm{S},n}$. Note that this is simply each surface coil image divided by the bias field estimate. When we get these estimates, $\hat{\boldsymbol{b}}$ becomes completely independent of all of our surface coil images except for $\boldsymbol{y}_{\mathrm{S},1}$. This is because the only term in the energy functional where $\hat{\boldsymbol{f}}_n$, $n \neq 1$ plays a role is in the data fidelity term corresponding to $\boldsymbol{y}_{\mathrm{S},n}$. Thus these terms can always be forced to zero by setting $\hat{\boldsymbol{f}}_n = \boldsymbol{B}^{-1}\boldsymbol{y}_{\mathrm{S},n}$. So when minimizing E with respect to $\boldsymbol{b}$, we need only concern ourselves with the data fidelity terms for $\boldsymbol{y}_{\mathrm{B}}$ and $\boldsymbol{y}_{\mathrm{S},1}$ as well as $\mathcal{R}_b$. Similarly, $\hat{\boldsymbol{f}}_1$ is not affected by any of the additional surface coil images that we capture. This essentially says that if we do not want to enforce priors on any of the $\hat{\boldsymbol{f}}_n$, then we should simply do bias correction on $\boldsymbol{y}_{\mathrm{B}}$ and $\boldsymbol{y}_{\mathrm{S},1}$. Then to debias all of the other $\boldsymbol{y}_{\mathrm{S},n}$, just divide them by the resulting $\hat{\boldsymbol{b}}$.

General formulation

In the previous two subsections, we presented results that are simply special cases of a more general observation model. This could be useful for a combination of the previous two cases (*e.g.*, four surface coil time-sequence heart imaging) or for more esoteric imaging combinations. We observe N_o images. For each $i \in \{1, 2, \ldots, N_o\}$, $\boldsymbol{y}_i$ is the product of an intrinsic image $\boldsymbol{f}_{n_i}^*$ with either 1 or a bias field $\boldsymbol{b}_{m_i}^*$ plus white noise:

$$\boldsymbol{y}_i = \boldsymbol{b}_{m_i}^* \circ \boldsymbol{f}_{n_i}^* + \boldsymbol{n}_i \ . \tag{3.124}$$

Without loss of generality, we drop the notion of the k scaling factor for the body coil images (see Section 3.3.1). This is perhaps best formulated in terms of a graphical model. We form observed nodes for each $\boldsymbol{y}_i$ along with nodes for all of the $\boldsymbol{f}_n$ and the $\boldsymbol{b}_m$. In general there is always one connection from an f node to an observation node and either zero or one from a b node depending on whether it is a body coil or surface coil image. We can either draw a node for the bias field of a body coil image (which we set to 1) and then treat it as observed, or we can omit it from the model entirely. Given our observations and $\{\hat{\boldsymbol{f}}_n\}_{n=1}^{N_f}$, all of our $\boldsymbol{b}_m$ are conditionally independent of each other. We can make similar conditional independence statements for the $\hat{\boldsymbol{f}}_n$. If the graph is not connected, we can split the overall problem into independent subproblems.

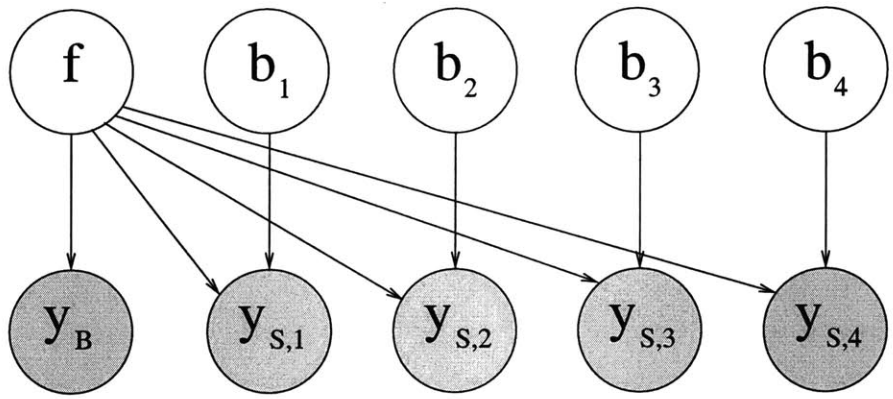


Figure 3.4. Graphical model for four surface coils and one intrinsic image with four surface coil observations and one body coil observation.

We define two sequences $\mathcal{I}_f = \{n_1, n_2, \ldots, n_{N_f}\}$ and $\mathcal{I}_b = \{m_1, m_2, \ldots, n_{N_b}\}$. We also define sets $\mathcal{D}_{f,n}$ and $\mathcal{D}_{b,m}$ which specify the observation images that are dependent on $\boldsymbol{f}_n$ and $\boldsymbol{b}_m$ respectively. If $j \in \mathcal{D}_{f,n}$, this implies that $n_j = n$. Similarly, if $j \in \mathcal{D}_{b,m}$, then $m_j = m$.

To illustrate this, in Figure 3.4 we draw the graph for a set of four surface coil images and one body coil image where we are trying to recover four bias fields and one intrinsic image. This is the case we covered with our first example in this subsection. Then one way we can choose our sets and sequences is through the following:

$$
\begin{aligned}
\mathcal{I}_f &= \{1,1,1,1,1\} \\
\mathcal{I}_b &= \{\emptyset,1,2,3,4\} \\
\mathcal{D}_{f,1} &= \{1,2,3,4,5\} \\
\mathcal{D}_{b,1} &= \{2\} \\
\mathcal{D}_{b,2} &= \{3\} \\
\mathcal{D}_{b,3} &= \{4\} \\
\mathcal{D}_{b,4} &= \{5\} \ .
\end{aligned}
$$

We use $\emptyset$ to indicate the lack of a connection from any $\boldsymbol{b}_m$ for a particular image. There are 5! ways to uniquely specify these sequences and sets because there are 5! ways to order our observation images. Choosing $\mathcal{I}_f$ and $\mathcal{I}_b$ completely specifies $\mathcal{D}_{f,1}$ and all of the $\mathcal{D}_{b,m}$, and the converse is also true. This is true in the general case as well.

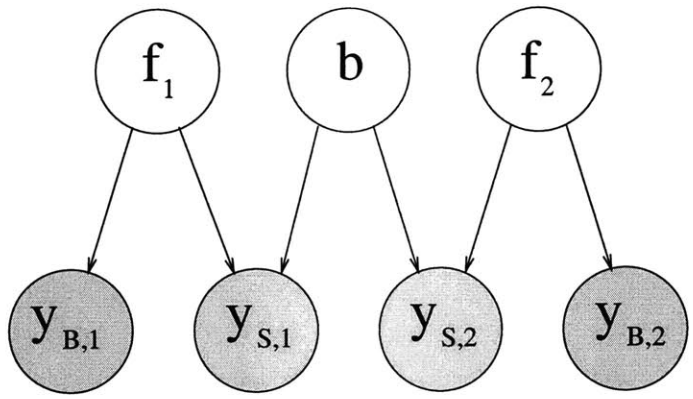


Figure 3.5. Graphical model for one surface coil and two intrinsic images with two surface coil observations and two body coil observations.

Previously we observed that each $\hat{\boldsymbol{b}}_m$ could be computed completely independently of the others. The reason for this can be seen on the graphical model. The only interaction among the different $\hat{\boldsymbol{b}}_m$'s are through $\boldsymbol{f}$.

In Figure 3.5, we draw the graph for a set of two surface coil images and two body coil images. Each surface coil image was captured using the same coil but with different imaging parameters (and likewise for the body coil images). We wish to estimate one bias field and two intrinsic images. One choice for our sets and sequences is this:

$$\begin{aligned}
\mathcal{I}_f &= \{1,1,2,2\} \\
\mathcal{I}_b &= \{\emptyset,1,\emptyset,1\} \\
\mathcal{D}_{f,1} &= \{1,2\} \\
\mathcal{D}_{f,2} &= \{3,4\} \\
\mathcal{D}_{b,1} &= \{2,4\} \ .
\end{aligned}$$

For the next example, we assume that the bias fields for the two surface coil images are similar but different. So now we wish to estimate two bias fields. We draw the graph in Figure 3.6. Note that this results in an unconnected graph. Thus all we do is estimate $\boldsymbol{f}_1$ and $\boldsymbol{b}_1$ only using $\boldsymbol{y}_{\mathrm{B},1}$ and $\boldsymbol{y}_{\mathrm{S},1}$ and similarly for $\boldsymbol{f}_2$ and $\boldsymbol{b}_2$. This method should produce bias field estimates $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ that are similar to $\boldsymbol{b}$ depicted in Figure 3.5. If this is not the case, our assumptions used to estimate a single bias field are extremely flawed.

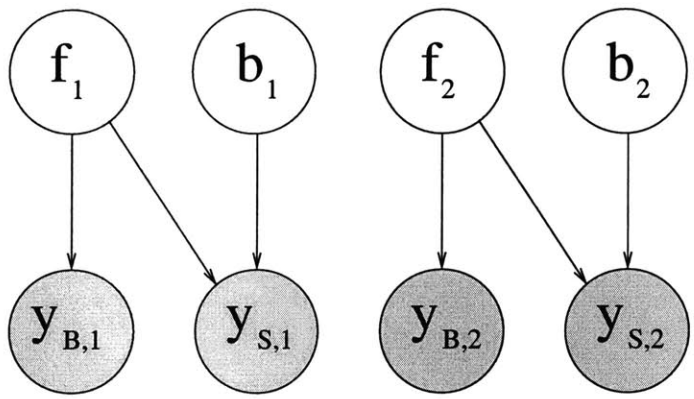


Figure 3.6. Graphical model for one surface coil and two intrinsic images with two surface coil observations and two body coil observations with estimation of a sensitivity profile for each surface coil image.

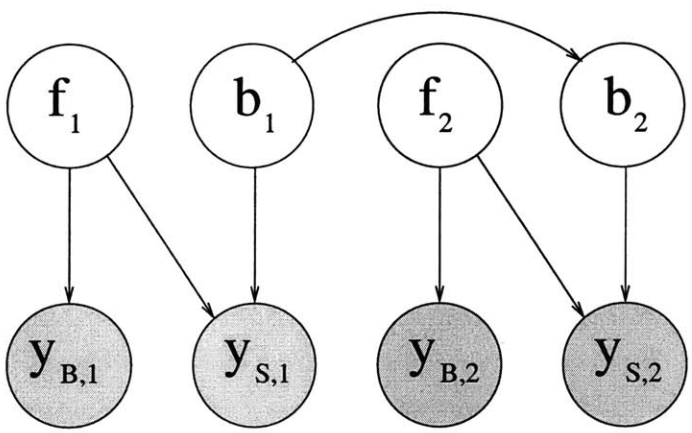


Figure 3.7. Graphical model for one surface coil and two intrinsic images with two surface coil observations and two body coil observations with estimation of a sensitivity profile for each surface coil image with some specified dependence between the two.

The previous example seems quite unsatisfactory. We know that $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the sensitivity profiles produced by the exact same coil. The bias fields will not be exactly the same due to varying magnetic susceptibility of the tissue and other environmental effects, but they certainly should be similar. We can indicate this dependence as in Figure 3.7 by connecting $\boldsymbol{b}_1$ with $\boldsymbol{b}_2$. The direction of the arrow does not matter. We might model this dependence with *e.g.*, a ℓ_2 norm which is equivalent to the assumption that $\boldsymbol{b}_2$ given $\boldsymbol{b}_1$ is Gaussian.

We use our earlier definition of $\mathcal{F}$ and $\mathcal{B}$ to collect all of our intrinsic images and bias fields into sets. We can slightly alter (3.16) to the following:

$$E(\mathcal{F},\mathcal{B}) = \sum_{i=1}^{N_o} \lambda_i E_i(\boldsymbol{f}_{n_i}, \boldsymbol{b}_{m_i}) + \sum_{m=1}^{N_b} \alpha_m \mathcal{R}_{b,m}(\boldsymbol{b}_m) + \sum_{n=1}^{N_f} \gamma_n \mathcal{R}_{f,n}(\boldsymbol{f}_n) \ . \tag{3.125}$$

There is now a data fidelity term for each observation image and a regularization term for each intrinsic image and each bias field. We can write them as follows:

$$E_i = \|\boldsymbol{y}_i - \boldsymbol{b}_{m_i} \circ \boldsymbol{f}_{n_i}\|^2 \tag{3.126a}$$

$$\mathcal{R}_{b,m} = \|\boldsymbol{L}_{b,m}\boldsymbol{b}_m\|^2 \tag{3.126b}$$

$$\mathcal{R}_{f,n} = \|\boldsymbol{L}_{f,n}\boldsymbol{f}_n\|_p^p \ . \tag{3.126c}$$

Of course p can change for each $\boldsymbol{f}_n$ as well, but in general we do not find that necessary. As always, this results in a quadratic optimization for each $\boldsymbol{b}_m$ given all of the $\hat{\boldsymbol{f}}_n$. We can specify the individual problems as follows:

$$\boldsymbol{Q}_{b,m} = 2 \sum_{j \in \mathcal{D}_{b,m}} \lambda_j \hat{\boldsymbol{F}}_{n_j}^2 + 2\boldsymbol{L}_{b,m}^{\mathrm{T}}\boldsymbol{L}_{b,m} \tag{3.127a}$$

$$\boldsymbol{a}_{b,m} = 2 \sum_{j \in \mathcal{D}_{b,m}} \lambda_j \hat{\boldsymbol{F}}_{n_j}\boldsymbol{y}_j \ . \tag{3.127b}$$

This can be solved using an iterative solver such as preconditioned conjugate gradient as before. We can also put our conditions for each $\boldsymbol{f}_n$ into a form compatible with (3.97):

$$\boldsymbol{H}_n^{\mathrm{T}}\boldsymbol{H}_n = \sum_{j \in \mathcal{D}_{f,n}} \lambda_j \hat{\boldsymbol{B}}_{m_j}^2 \tag{3.128a}$$

$$\boldsymbol{H}_n^{\mathrm{T}}\boldsymbol{z}_n = \sum_{j \in \mathcal{D}_{f,n}} \lambda_j \hat{\boldsymbol{B}}_{m_j}\boldsymbol{y}_j \ . \tag{3.128b}$$

We can then use our half-quadratic solver to do the f-step. The conditional independence property that we cited earlier allows us to process the $\hat{\boldsymbol{f}}_n$ in any order on the f-step and the $\hat{\boldsymbol{b}}_m$ in any order on the b-step.

3.4.4 Multigrid

Multiresolution techniques are fast methods to solve problems that are either explicitly multiscale or have multiscale structure. We can take advantage of this structure to improve computation speed for large and/or complex problems. We briefly discussed multigrid methods in Section 2.1.4. Traditional multigrid is built using simple iterative solvers such as Gauss-Seidel using multiple so-called V-cycles. Each V-cycle is a fine-to-coarse sweep followed by a coarse-to-fine sweep. Usually only one iteration of the solver is used at each level. We use a much more basic form of the method where we have a single coarse-to-fine sweep. We downsample our data to the coarsest level we wish to process. We then run our coordinate descent solver at this level and upsample the results to the next finest level. This cycle repeats until we have a solution at the finest level.

As long as our problem has some multiscale structure, the solution at the coarser scale will bear some relation to the solution at the finer scale. We find that this is the case for our problem. Because the bias field is known to be smooth, most of its energy will be concentrated in the lower frequency components. These components will also exist in the downsampled versions of our image, so most of the work at the finest level will be to smooth out the results from the coarser resolution and estimate the higher frequency components. The intrinsic image will have more energy at high frequencies than bias fields due to the presence of edges. But away from edges, we expect low frequency behavior. So we expect the frequency to have a bimodal distribution. At the coarser resolutions, we can obtain reliable estimates of the low frequency components. At the highest scale, we can gain better edge resolution.

This technique is beneficial to us because our solver does not converge in linear time (we estimate the computational cost as approximately $\mathcal{O}(n^2)$). So it is much faster to solve the problem on a coarser grid. This speed increase makes the initial guesses for $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{b}}$ less important. We can converge to a result that is very close to the correct answer at coarser scales and iterations at the coarser scale take very little time, so our algorithm becomes more resistant to poor initializations. Besides the speed advantages, multiresolution methods can also increase robustness by helping to avoid local minima.

We now have an additional parameter for scale: $s = \{0, 1, 2, \ldots S\}$ where S is the coarsest scale we wish to use. The finest level is indicated with $s = 0$. We will specify a new multiscale observation model using the most general model used in Section 3.4.3:

$$\boldsymbol{y}_i^{[s]} = \boldsymbol{b}_{m_i}^{[s]*} \circ \boldsymbol{f}_{n_i}^{[s]*} + \boldsymbol{n}_i^{[s]}. \tag{3.129}$$

At each level, all of the variables are samples of their continuous counterparts (*e.g.*, $\boldsymbol{f}^{[s]}$ is sampled from φ) with new sampling vectors at each scale:

$$\boldsymbol{i}^{[s]}[n] = \left\lfloor \frac{2^s n}{N} \right\rfloor \tag{3.130a}$$

$$\boldsymbol{j}^{[s]}[n] = (2^s)n \bmod N \tag{3.130b}$$

$$n \in \{0, 1, 2, \ldots, 4^{-s}MN - 1\} \ .$$

Each decrease in scale results in a decrease in problem size by a factor of four.

This results in an energy functional at each scale that we seek to minimize:

$$E(\mathcal{F}^{[s]}, \mathcal{B}^{[s]}) = \sum_{i=1}^{N_o} \lambda_i^{[s]} E_i(\boldsymbol{f}_{n_i}^{[s]}, \boldsymbol{b}_{m_i}^{[s]}) + \sum_{m=1}^{N_b} \alpha_m^{[s]} \mathcal{R}_{b,m}(\boldsymbol{b}_m^{[s]}) + \sum_{n=1}^{N_f} \gamma_n^{[s]} \mathcal{R}_{f,n}(\boldsymbol{f}_n^{[s]}) \ . \tag{3.131}$$

This technique works best when the $\mathcal{F}^{[s]}$ and $\mathcal{B}^{[s]}$ that minimize the energy at scale s are similar to the $\mathcal{F}^{[s-1]}$ and $\mathcal{B}^{[s-1]}$ that minimize the energy at the next finer scale.

To implement this technique, we need to specify how we downsample and upsample. To downsample, we can take a nearest neighbor approach (which on our fixed grid is equivalent to block averaging), or we can optimally reconstruct the continuous-space signal, low-pass filter (to avoid aliasing), and resample at a coarser resolution. To upsample, there are a number of techniques we can employ. Among these are nearest neighbor (block replication in our case), bilinear/trilinear interpolation, bicubic/tricubic interpolation, and sampling from the optimal reconstruction of the continuous-space signal.

We also need to decide how the reconstruction parameters $\lambda_i^{(s)}$, $\alpha_i^{[s]}$, and $\gamma_i^{[s]}$ change with s. The $\lambda_i^{[s]}$'s represent the inverse noise variance for $\boldsymbol{y}_i^{[s]}$ and are increased by a factor of 4 at each scale. This is only an approximation because the average of four IID Rician random variables does not yield another Rician random variable with one-fourth of the variance. To the extent that our Gaussian noise assumption holds, this scaling by 4 also holds. There are also effects such as partial-volume effects that become more pronounced at coarser resolutions, but we will ignore them. The regularization

parameters are more difficult to choose. For wavelet-based multiscale reconstruction, others [6,9] have found that a multiplicative scalar is often effective. Hence we need to find scalars $\varepsilon_{\alpha,i}$ and $\varepsilon_{\gamma,i}$ which we can then use in the following manner:

$$\gamma_i^{[s]} = (\varepsilon_{\gamma,i})^s \gamma_i \tag{3.132}$$

$$\alpha_i^{[s]} = (\varepsilon_{\alpha,i})^s \alpha_i \tag{3.133}$$

$$\forall s \in \{0, 1, \ldots, S\} \ .$$

The value of these scalars may vary from application to application. Sometimes the relationship between regularization parameters across scale is not this simple, but we get good results in practice using this method.

■ 3.5 Summary

We began this chapter by introducing our observation model. We assumed that our body coil image and surface coil image share a common intrinsic image that is the true MR signal. The two images are assumed to differ only through additive noise and a multiplicative gain field on the surface coil image that is independent of the underlying tissue. The correct physical noise model is Rician, but we showed that a Gaussian noise model will produce largely similar results while greatly simplifying computation. The main effect of our Gaussian noise assumption is an upward bias that increases as SNR decreases.

We then presented a variational problem whose solution approximates the true bias field and intrinsic image. The main features of our energy functional were ℓ_2 norms for data fidelity terms and ℓ_p norms to regularize the bias field and intrinsic image estimates. We showed the statistical interpretation of our energy functional as a way of justifying its validity.

Our energy functional is difficult to minimize simultaneously for the bias field $\boldsymbol{b}$ and the intrinsic image $\boldsymbol{f}$. The individual subproblems of minimizing for $\boldsymbol{b}$ or $\boldsymbol{f}$ are relatively simple, so we use coordinate descent and alternate between what we term b-steps and f-steps. We showed how to efficiently solve the subproblems with both continuous and discrete methods. In particular, preconditioned conjugate gradient seemed like a promising method.

Finally we generalized our framework to handle a number of extensions. Previously we had only used ℓ_2 norms to regularize our estimates. We use half-quadratic solvers

to efficiently solve more complex ℓ_p regularization problems where $p \leq 1$. We also discussed the special considerations needed to do full 3D processing. We generalized our method to handle an arbitrary number of input images. This allows us to *e.g.*, process the images from each surface coil separately. Finally, we showed how multiresolution solvers can be incorporated into our algorithm as a way to increase the convergence speed.

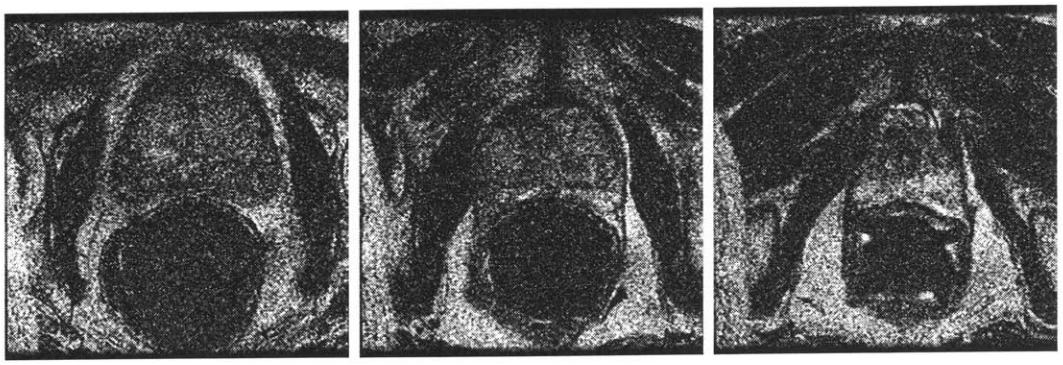
Chapter 4

Results

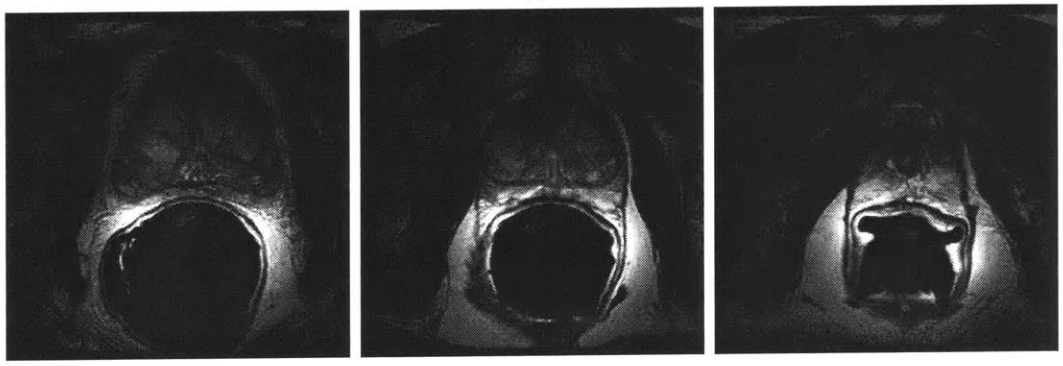
WE present results in this chapter for four different applications: prostate imaging using an endorectal coil and a pelvic phased-array coil; time-gated heart imaging using a phased-array coil; brain imaging using a phased-array coil; and the Montreal Neurological Institute (MNI) simulated brain phantom [19, 40].

Using multiple real data applications allows us to demonstrate the utility of our algorithm on examples with disparate SNR levels, bias field shapes, and tissue structure. The brain phantom allows us to test our algorithm on realistic examples with known ground truth. This allows us to quantitatively assess our performance. Note that our phantom images explicitly conform to our observation model. The positive results on real data that we will show seem to confirm the validity of our observation model.

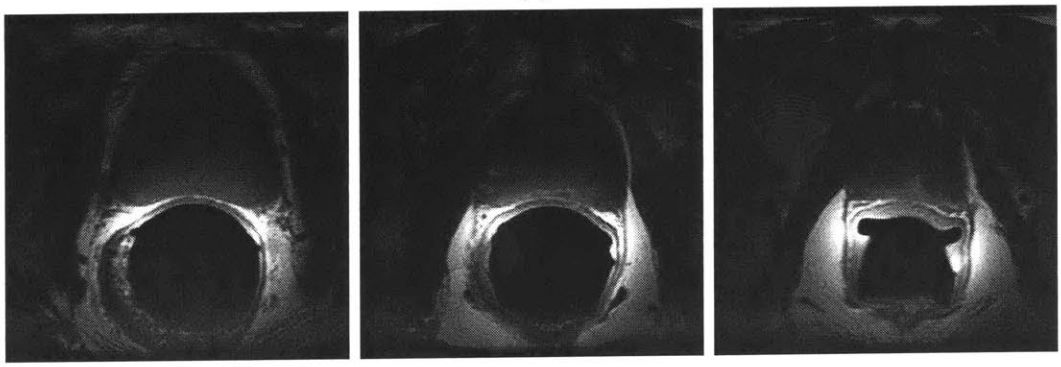
The regularization parameters α and γ used in each example are chosen manually to provide optimal visual results. The α values are generally comparable across examples. The bias field is a multiplicative gain function, so it is invariant to scaling of the intensity values in the observation images. Thus the different α values in each example can be seen as a measurement of the smoothness of the surface coil reception profiles. The different tissue intensities across imaging protocols make the γ values not directly comparable. This is simply a consequence of the fact that different imaging pulse sequences are designed to measure different fundamental tissue properties. However, the γ values are comparable for different scans acquired using the exact same imaging protocol (*e.g.*, the same choices of γ in different T_2-*weighted* prostate samples indicates similar levels of regularization in each image).



(a)



(b)



(c)

Figure 4.1. Three slices from prostate sequence A. (a) The body coil T_2-*weighted* images ($\boldsymbol{y}_{\mathrm{B}}$), (b) the composite T_2-*weighted* surface coil images ($\boldsymbol{y}_{\mathrm{S},T_2}$), and (c) the composite T_1-*weighted* surface coil images ($\boldsymbol{y}_{\mathrm{S},T_1}$).

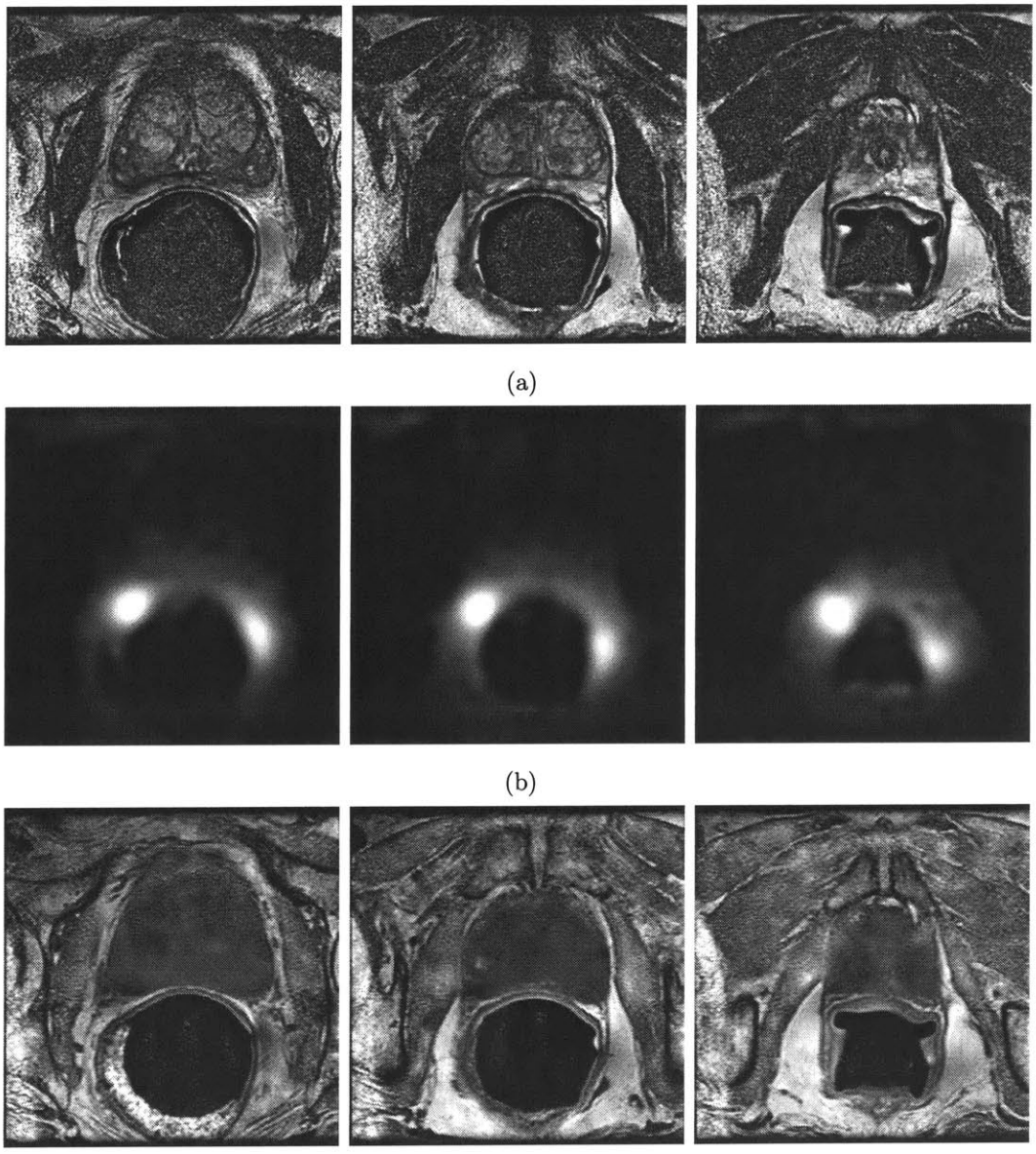
4.1 Prostate

In this section we demonstrate our bias correction techniques on prostate images captured using General Electric Signa 1.5-T MR machines. Both T_1- and T_2-*weighted* fast-spin echo (FSE) images were captured using the surface coil. Only T_2-*weighted* body coil images were captured. The FOV is 12 cm × 12 cm with 3 mm slice thickness. The surface coil images were captured using a transmitting body coil and five receiving surface coils: four coils wrapped around the pelvis in a phased array and an endorectal coil [49,64,65]. The endorectal coil is mounted inside a balloon that is inserted into the rectum. The actual coil is a rectangular wire that is nestled up against the posterior wall of the prostate. The same imaging parameters (*e.g.*, pulse sequence, number of excitations) used in the surface coil imaging were used in the body coil imaging.

Because the prostate is so small, less spatial averaging is possible. The SNR in the prostate body coil images is the lowest out of any of our imaging applications. We can compensate for the resulting high noise levels with higher local signal response. High local signal response in the surface coil generally results in low signal response elsewhere. This leads to a strong inhomogeneity which is quite prominent in the prostate images. The combination of high noise and a strong bias field makes prostate bias correction both more challenging and more essential.

We have acquired data from three patients: one is of very high quality, and the other two have fairly significant visual artifacts. We will refer to the three data sets using letters. The high quality data set is data set A, and the other two are data sets B and C. All results in this section are generated using a Laplacian operator for the linear matrix operator $\boldsymbol{L}_b$ and a gradient operator for $\boldsymbol{L}_f$. The values of α (the regularization parameter on $\hat{\boldsymbol{b}}$) and γ (the regularization parameter for $\hat{\boldsymbol{f}}$) for each example are noted in the figure captions.

In Figure 4.1 we show the observed images $\boldsymbol{y}_{\mathrm{B}}$, $\boldsymbol{y}_{\mathrm{S},T_2}$, and $\boldsymbol{y}_{\mathrm{S},T_1}$ for three slices from data set A. Needless to say, the intensity inhomogeneity is quite prominent in the surface coil images. The SNR for the body coil image is approximately 7 dB in the prostate. The SNR for the surface coil images (both $\boldsymbol{y}_{\mathrm{S},T_2}$ and $\boldsymbol{y}_{\mathrm{S},T_1}$) vary quite a bit but can be as high as 35 dB. This means that the gain from the endorectal coil is up to 28 dB. These SNR and gain measurements are similar for all of our prostate data sets.



(a)

(b)

(c)

Figure 4.2. Intrinsic image and bias field estimates for the prostate data set A with no $\hat{\boldsymbol{f}}$ regularization. (a) The T_2-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_2}$), (b) the bias field estimates ($\hat{\boldsymbol{b}}$), and (c) the T_1-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 125$, $\gamma_{T_2} = 0$, $\gamma_{T_1} = 0$.

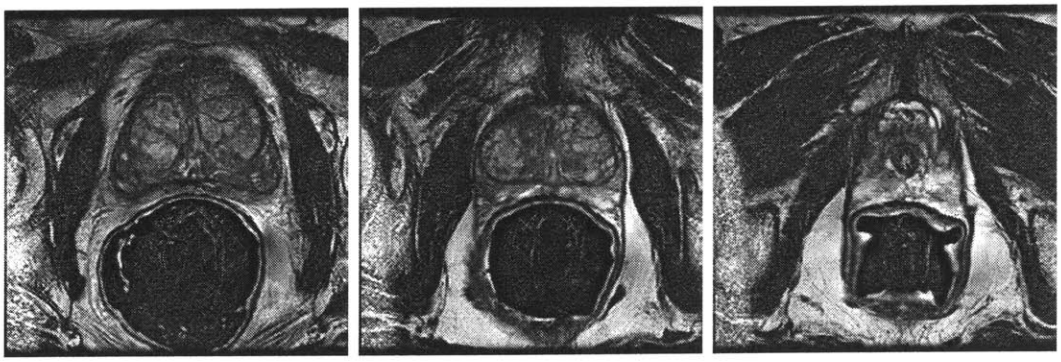
4.1.1 No Regularization on $\hat{f}$

In Figure 4.2 we show the results of our bias correction algorithm on the data from Figure 4.1. Because there is no regularization on $\hat{\boldsymbol{f}}_{T_1}$, the $\boldsymbol{y}_{S,T_1}$ images do not affect the bias field estimates (see Section 3.4.3). Overall the bias field is largely removed in both $\hat{\boldsymbol{f}}_{T_2}$ and $\hat{\boldsymbol{f}}_{T_1}$, though it is difficult to tell whether a slight inhomogeneity remains from either the body coil or imperfect bias field estimates. The noise in the reconstructed images can be seen to be much lower in the regions close to the rectum. The results for the $\hat{\boldsymbol{f}}_{T_1}$ images are fairly effective even without a body coil measurement. However, there is more intensity variation within the prostate capsule than we would wish.

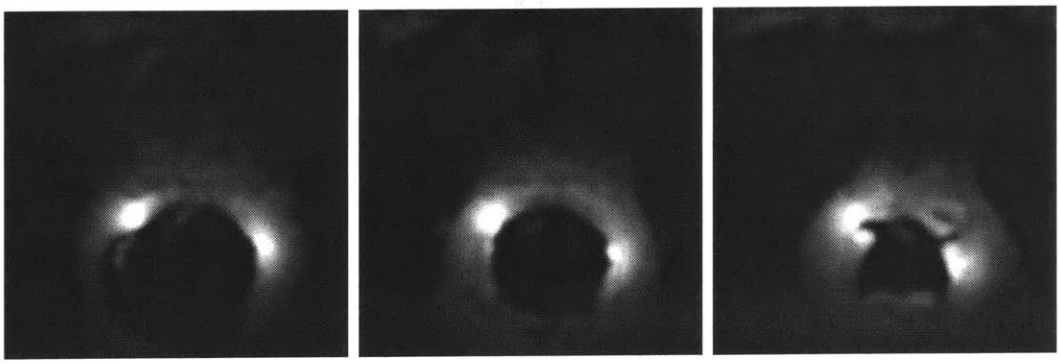
The effects of combining the multiple surface coil measurements into one composite image can be seen in the bias field estimates in Figure 4.2(b). In order to accommodate the large inhomogeneity from the endorectal coil, α must be set relatively low. This ensures that we do not overpenalize the strong curvature associated with the rapidly diminishing coil reception profile. In regions where the endorectal coil has little response and the pelvic coils dominate, the bias field estimate becomes very lumpy. The value of α is too low in these regions, and local variations from the noise and tissue become part of the bias field estimate. This consequently has a deleterious effect on both estimated intrinsic images for the T_2-*weighted* sequence and the T_1-*weighted* sequence. There is subtle oversmoothing in $\hat{\boldsymbol{f}}_{T_2}$ within the prostate, and there are clear inhomogeneities within the prostate in $\hat{\boldsymbol{f}}_{T_1}$ that can be seen to correspond with the lumpiness in $\hat{\boldsymbol{b}}$.

In Figure 4.3 we display the bias correction results obtained using Brey and Narayana's method. In Figure 4.4, we correct the bias using homomorphic unsharp filtering. Overall Brey-Narayana correction provides results very similar to our correction, though the bias field estimates are not as smooth and much more tissue residue remains. Homomorphic filtering can be seen to provide fairly homogeneous results at the expense of contrast. We can, of course, achieve perfect homogeneity by setting the bias field equal to the observed surface coil image. The estimated intrinsic image is then uniformly 1. This is not informative, but homomorphic filtering approaches this if we decrease the size of our filtering kernel. We will not include homomorphic filtering results with any further examples due to the poor quality of the results.

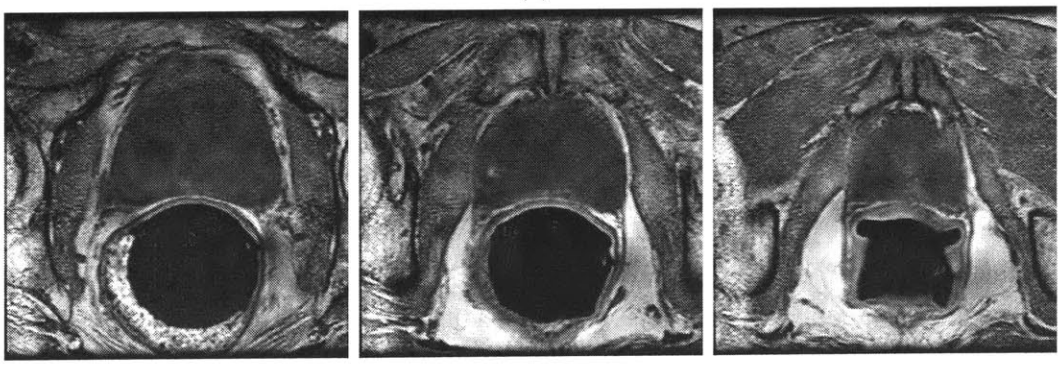
In Figure 4.5, we display the absolute differences between reconstructed images obtained using our algorithm (displayed in Figure 4.2) and Brey-Narayana (displayed in



(a)



(b)



(c)

Figure 4.3. Intrinsic image and bias field estimates from prostate data set A using Brey-Narayana bias correction. (a) T_2-*weighted* true image estimates, (b) bias field estimates, and (c) T_1-*weighted* image estimates.

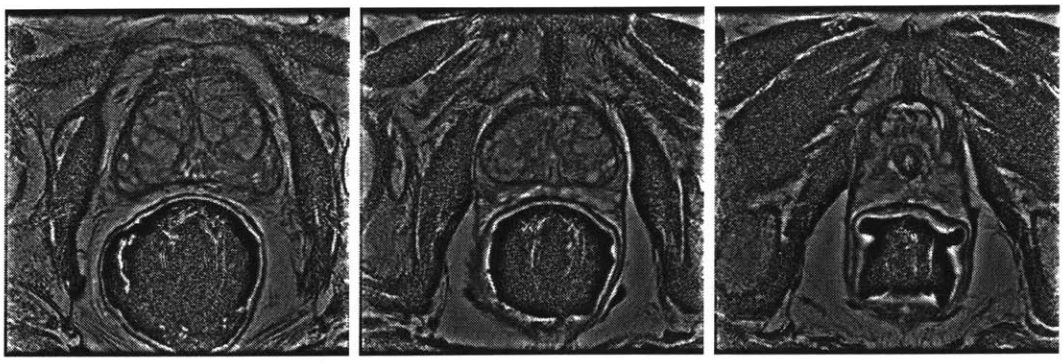
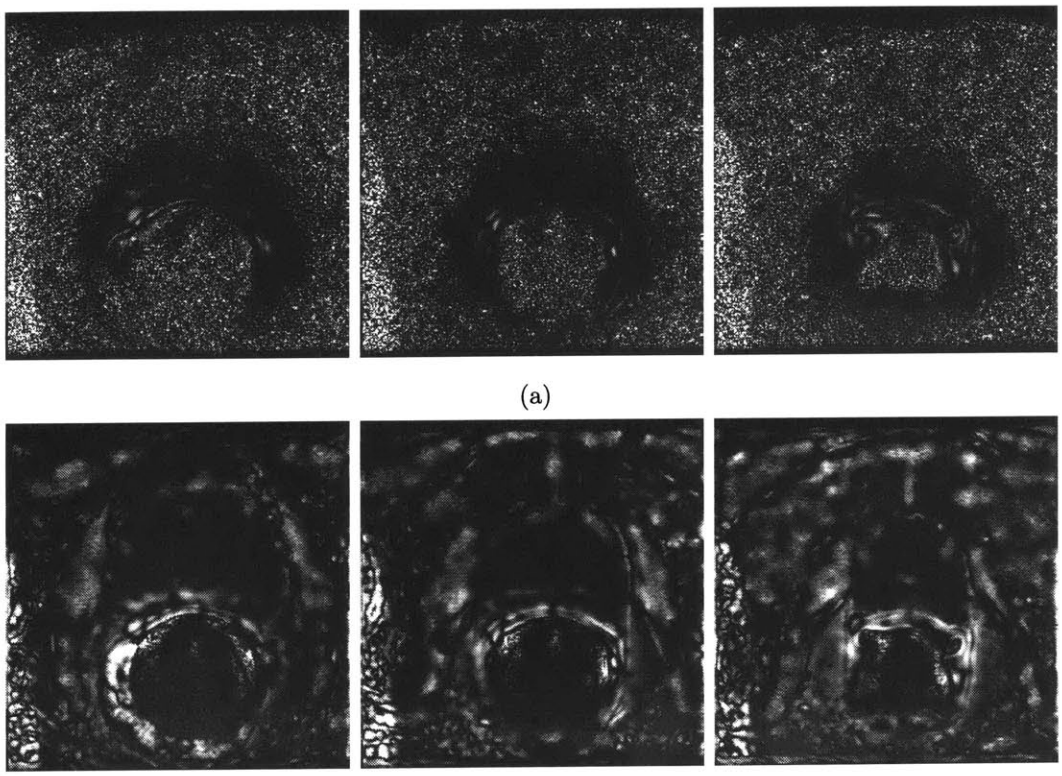


Figure 4.4. Bias correction results for prostate data set A using homomorphic unsharp filtering.

Figure 4.3). The results in Figure 4.5(a) for the T_2-*weighted* images are quite striking. There is a halo around the rectum where both techniques produce nearly identical results. Further away, there is a great deal of variation that appears white in nature. For the T_1-*weighted* images, the results are more varied. The largest differences occur on tissue boundaries. Brey-Narayana tends to underestimate the bias field on one side of edges and overestimate it on the other because it amasses the bias field estimate from a local neighborhood of each pixel.

We show some sample slices from data sets B and C in Figures 4.6 and 4.8 respectively. Both data sets display fairly severe motion artifacts. They can be seen as horizontal disturbances appearing near the middle of the images. We display our correction results in Figures 4.7 and 4.9. The results for data set B are not very good. A great deal of noise remains in the final reconstructed images, especially in the T_2-*weighted* images. The reason for this is that the endorectal coil is mounted so that the largest effects of the endorectal coil are directed at the sides of the rectum rather than on the anterior face adjacent to the prostate. Thus the SNR level within the prostate in the surface coil images is much lower than for the previous example. There is also excessive blurriness in the final reconstructed images. This may be caused by slight misregistration between the two input sequences. There is a brightness artifact on the left side of the rectum that is present in $\boldsymbol{y}_{\mathrm{B}}$ and consequently in $\hat{\boldsymbol{f}}_{T_2}$. This brightness does not appear to also exist in $\boldsymbol{y}_{\mathrm{S},T_2}$ though the intensity inhomogeneity makes it difficult to be sure. This artifact in the body coil images causes the bias field estimate to be too



(a)

(b)

Figure 4.5. Absolute difference comparisons between our algorithm and Brey-Narayana using data set A. Differences for the (a) T_2-*weighted* and (b) T_1-*weighted* intrinsic image estimates.

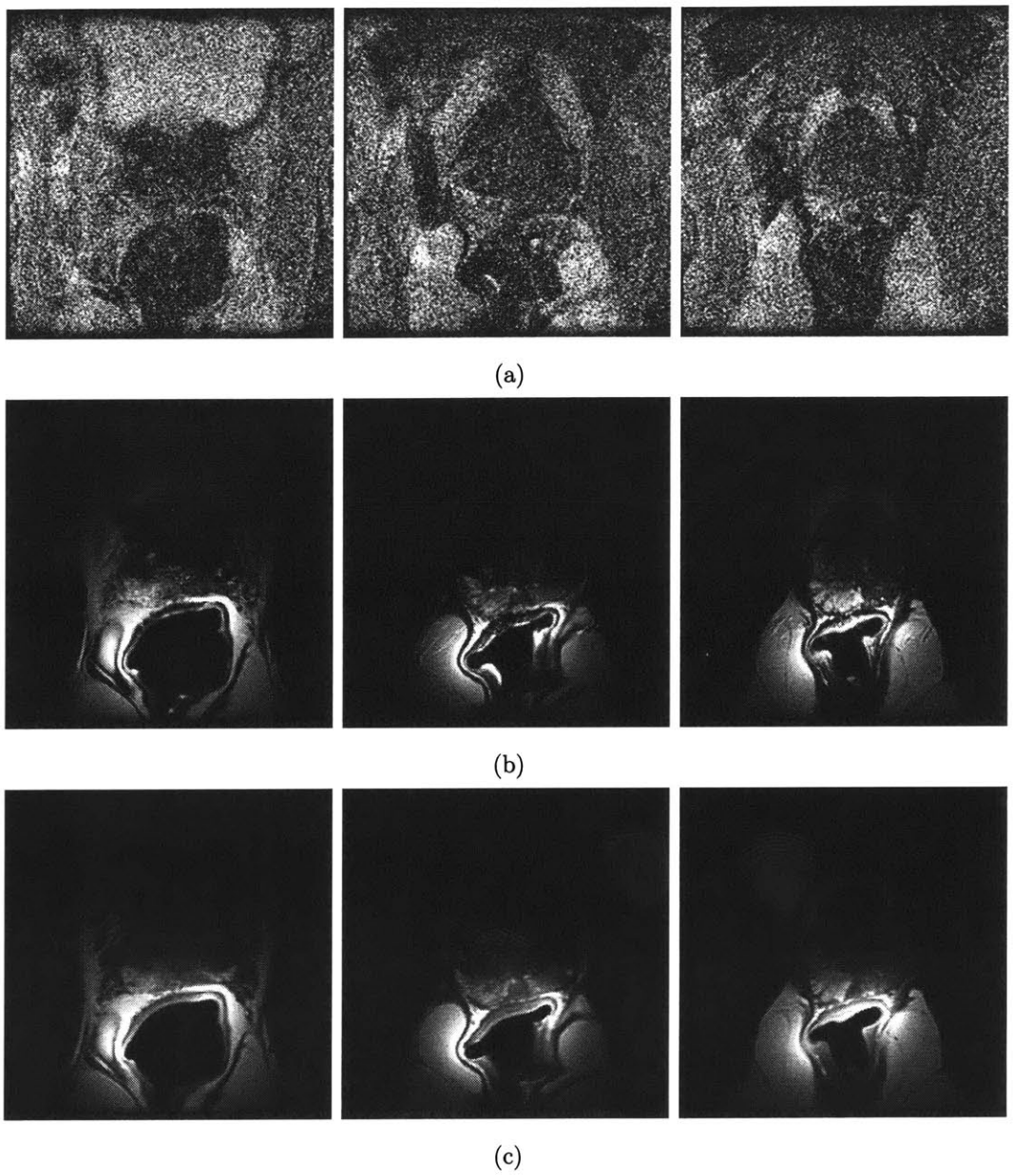


Figure 4.6. Body coil and surface coil images from prostate data set B. T_2-*weighted* (a) body coil ($\boldsymbol{y}_{\mathrm{B}}$) and (b) surface coil slices ($\boldsymbol{y}_{\mathrm{S},T_2}$). (c) T_1-*weighted* surface coil images ($\boldsymbol{y}_{\mathrm{S},T_1}$).

low in those regions and this artifact is then propagated into $\hat{\boldsymbol{f}}_{T_1}$. This artifact could probably be partially alleviated with stronger regularization on $\hat{\boldsymbol{b}}$ to make our estimates less data dependent. Unfortunately, the sharpness of the endorectal coil inhomogeneity makes larger α values impossible.

Data set C does not suffer from the SNR and blurriness problems and has very nice differentiation between the central zone and peripheral zone in the T_2-*weighted* images. Large motion artifacts appear in the final T_1-*weighted* intrinsic image estimates. There is also an interesting artifact that appears just above the prostate in $\hat{\boldsymbol{f}}_{T_1}$. This excessive brightness in two of the slices is caused by the low T_2-*weighted* signal in those locations. Our bias field estimates have a natural tendency to become small where $\boldsymbol{y}_{\mathrm{B}}$ becomes small. Because we do not have a body coil image for the T_1-*weighted* images, errors in the bias field estimate that may not impact $\hat{\boldsymbol{f}}_{T_2}$ may cause problems in $\hat{\boldsymbol{f}}_{T_1}$. Again, more regularization in $\hat{\boldsymbol{b}}$ away from the endorectal coil would help alleviate these issues.

■ 4.1.2 $\hat{f}$ Regularization

In Figure 4.10, we minimize our full energy functional with regularization on both $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{f}}$. The edges in both the T_2-*weighted* and T_1-*weighted* images remain sharply defined due to our usage of a ℓ_1 prior. We can also directly compare these results with Figure 4.2 which contains our results without $\hat{\boldsymbol{f}}$ regularization. The noise levels are visibly lower in regions far from the endorectal coil, while the reconstructed image does not change in the high-SNR regions surrounding the coil (see Section 3.4.1).

There is not an appreciable change in the bias field estimates when we add $\hat{\boldsymbol{f}}$ regularization. The reason for this is that we generally choose γ so that $\hat{\boldsymbol{f}}$ with regularization looks like a denoised version of $\hat{\boldsymbol{f}}$ without regularization. The regularization on $\hat{\boldsymbol{b}}$ makes the bias field estimates relatively resistant to the noise (or lack thereof) in $\hat{\boldsymbol{f}}_{T_2}$ and $\hat{\boldsymbol{f}}_{T_1}$. If γ is chosen too large, the bias field estimates will begin to change. The intensity variation in $\boldsymbol{y}_{\mathrm{S},T_2}$ must be accounted for by a combination of variation in $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{f}}_{T_2}$. Generally an edge will be incorporated into $\hat{\boldsymbol{f}}_{T_2}$. This choice is consistent with the edge that should be present in $\boldsymbol{y}_{\mathrm{B}}$[1], and the ℓ_2 regularization for $\hat{\boldsymbol{b}}$ penalizes large variation more than the ℓ_1 regularization on $\hat{\boldsymbol{f}}_{T_2}$. But when γ becomes large enough, $\hat{\boldsymbol{f}}_{T_2}$ variation is penalized as much as $\hat{\boldsymbol{b}}$ variation, and part of the intensity change is assigned to

[1] If an edge in $\boldsymbol{y}_{\mathrm{S},T_2}$ does not have a corresponding edge in $\boldsymbol{y}_{\mathrm{B}}$, the variation is then properly attributed to our bias field estimate.

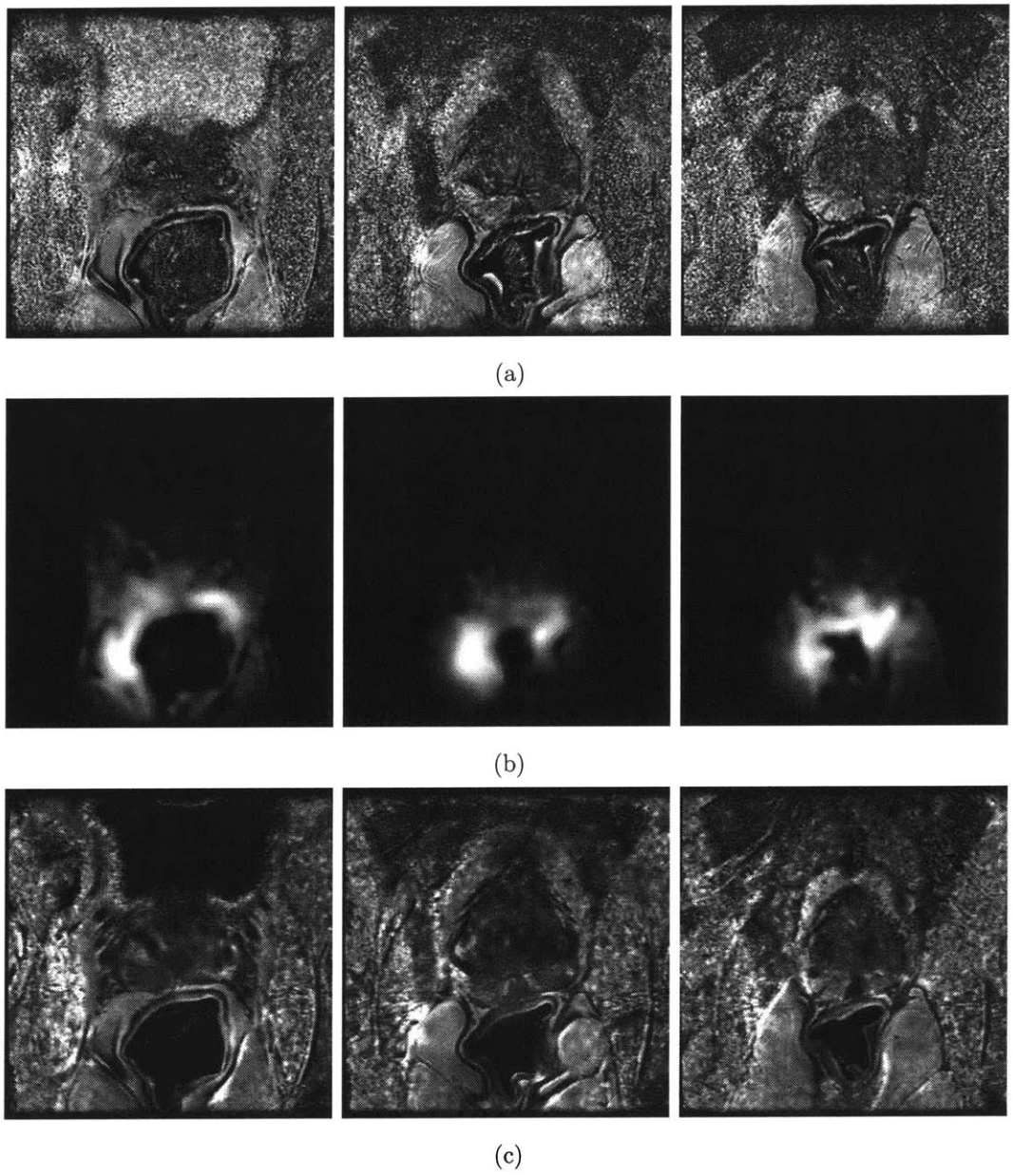


Figure 4.7. Intrinsic image and bias field estimates (no $\hat{\boldsymbol{f}}$ regularization) from data set B. (a) T_2-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_2}$), (b) bias field estimates ($\hat{\boldsymbol{b}}$), and (c) T_1-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 50$, $\gamma_{T_2} = 0$, $\gamma_{T_1} = 0$.

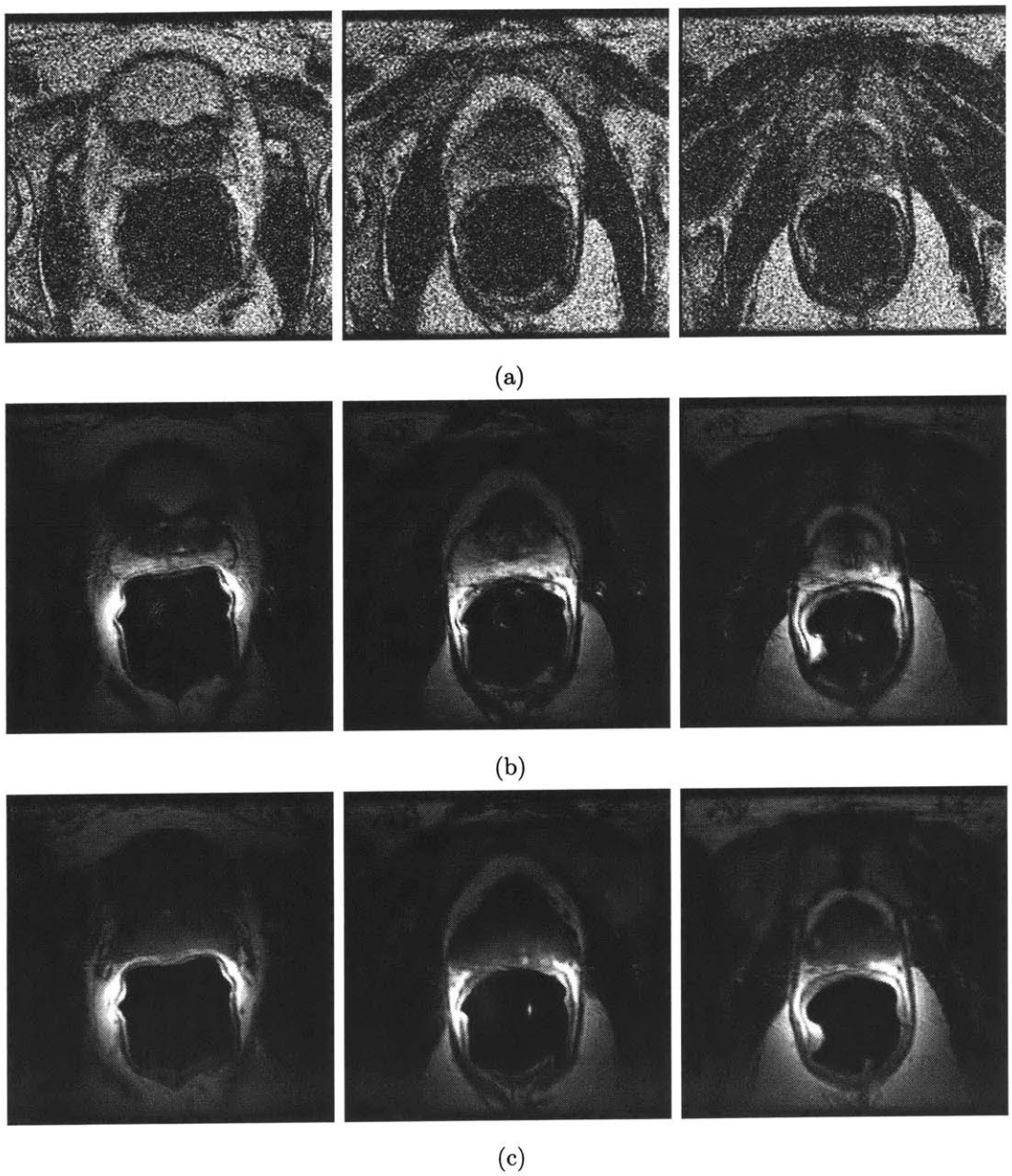


Figure 4.8. Body coil and surface coil images from prostate data set C. (a) Body coil T_2-*weighted* images ($\boldsymbol{y}_{\mathrm{B}}$), (b) surface coil T_2-*weighted* images ($\boldsymbol{y}_{\mathrm{S},T_2}$), and (c) surface coil T_1-*weighted* images ($\boldsymbol{y}_{\mathrm{S},T_1}$).

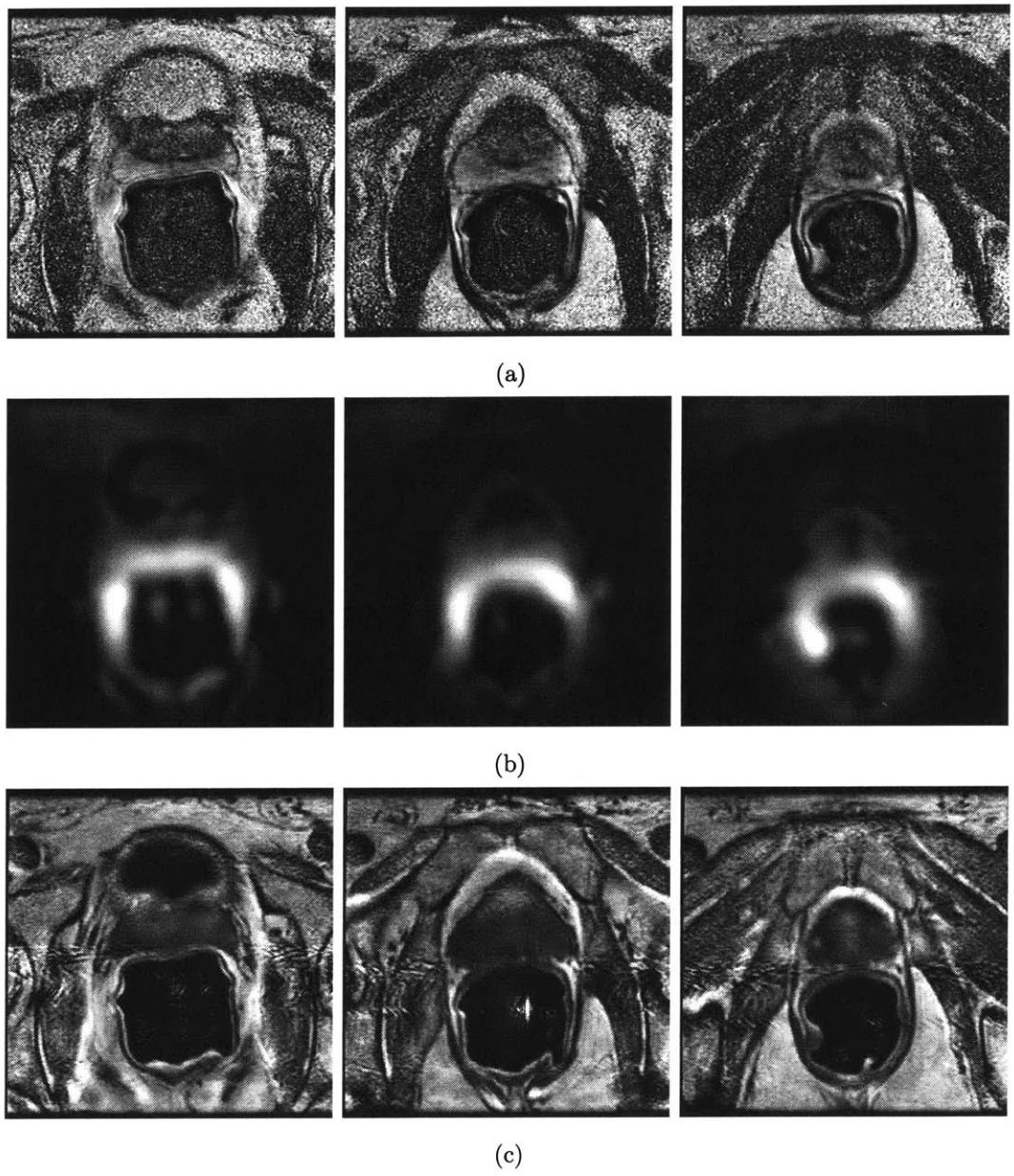


Figure 4.9. Data set C intrinsic image and bias field estimates without $\hat{\boldsymbol{f}}$ regularization. (a) Intrinsic image estimates for the T_2-*weighted* sequence ($\hat{\boldsymbol{f}}_{T_2}$), (b) bias field estimates ($\hat{\boldsymbol{b}}$), and (c) T_1-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 125$, $\gamma_{T_2} = 0$, $\gamma_{T_1} = 0$.

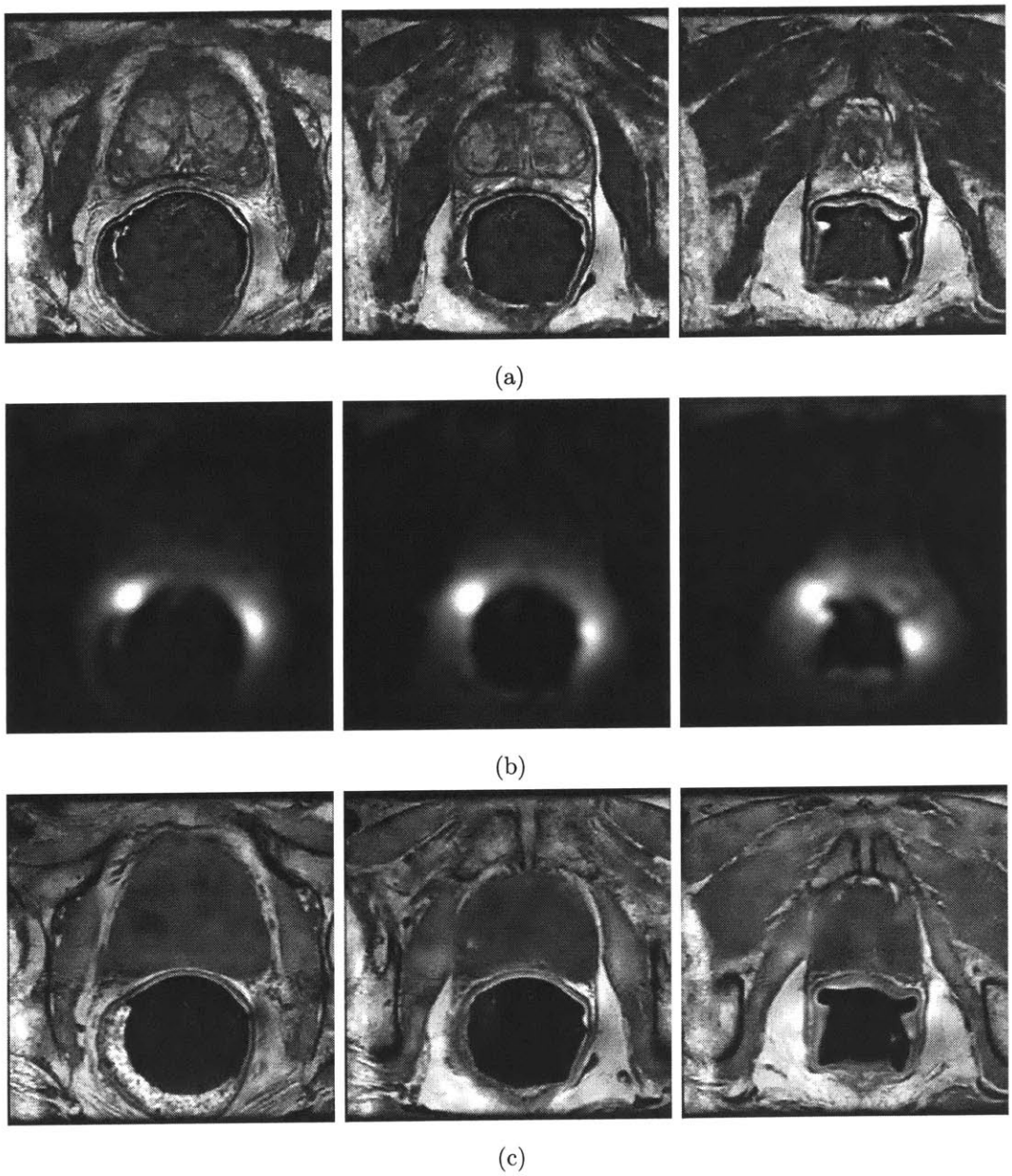


Figure 4.10. Intrinsic image and bias field estimates with $\hat{\boldsymbol{f}}$ regularization from the prostate data set A. (a) T_2-*weighted* intrinsic images ($\hat{\boldsymbol{f}}_{T_2}$), (b) bias fields ($\hat{\boldsymbol{b}}$), and (c) T_1-*weighted* intrinsic images ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 125$, $\gamma_{T_2} = 0.012$, $\gamma_{T_2} = 0.010$.

$\hat{\boldsymbol{b}}$. Thus when γ is very large, $\hat{\boldsymbol{b}}$ changes to incorporate some of the edge information that should be in $\hat{\boldsymbol{f}}_{T_2}$.

We demonstrate in Figure 4.11 the effects of minimizing our energy functional with $\gamma = 0$ and then applying a post-processing filter step. The post-processing filter is a ℓ_1 reconstruction:

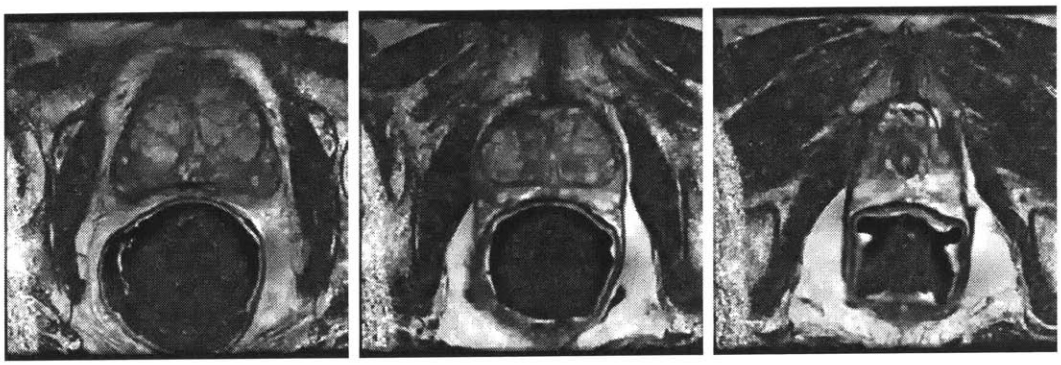
$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \|\hat{\boldsymbol{f}}_0 - \boldsymbol{f}\|^2 + \gamma \|\boldsymbol{L}_f \boldsymbol{f}\|_1^1 \tag{4.1}$$

where $\hat{\boldsymbol{f}}_0$ is $\hat{\boldsymbol{f}}$ obtained with $\gamma = 0$. The top set of examples use the same regularization strength as was used in Figure 4.10. The effects away from the endorectal coil are largely similar, but inside the prostate, much of the detail has been blurred away. The bottom set of examples use a weaker regularization strength that preserves fine details within the prostate at the expense of noise outside. Clearly the latter example produces results superior to those in Figure 4.2, so an edge-preserving filter is beneficial. But the results are inferior to those with the fully integrated ℓ_1 regularization.

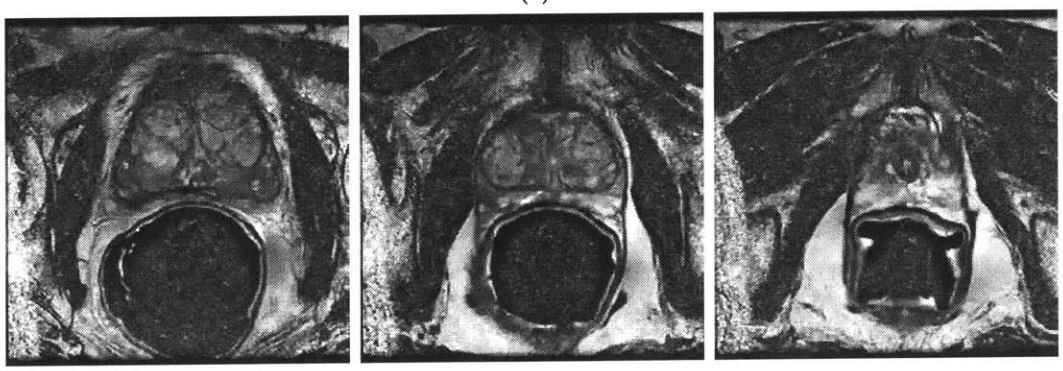
We present results using our full algorithm on data set B in Figure 4.12 and for data set C in Figure 4.13. For data set B, the value of γ used was much lower than for data set A. As mentioned earlier, the endorectal coil does not appear to have been mounted optimally, so the SNR within the prostate is relatively low. Due to the noise-weighted smoothing effect that our algorithm imposes, we must lower γ to avoid oversmoothing within the prostate. The results for data set C are fairly similar to those from data set A. Note that in Figure 4.13(a), the motion artifact actually becomes more pronounced with the $\hat{\boldsymbol{f}}$ regularization. The artifact does not become larger, but it becomes more visible when the noise is reduced. The ℓ_1 regularization does little to reduce the motion artifact. This is because the artifact is more edge-like (in fact, it is simply the actual edges replicated over space) than noise-like, and our regularization is designed to preserve edges. Hence the artifact is preserved while the noise is removed.

■ 4.1.3 Parameter Variation

In this section, we examine the effects of varying the parameters in our total energy functional (3.16). We only use ℓ_2 norms for $\hat{\boldsymbol{b}}$ and ℓ_1 norms for $\hat{\boldsymbol{f}}$. Hence p is fixed at 1 and q is fixed at 2. The data fidelity weighting parameter λ is determined by the noise variances in the images, and we can obtain reliable estimates of these variances (see Section 3.3.1). This leaves α and γ as parameters that can affect the final solution.



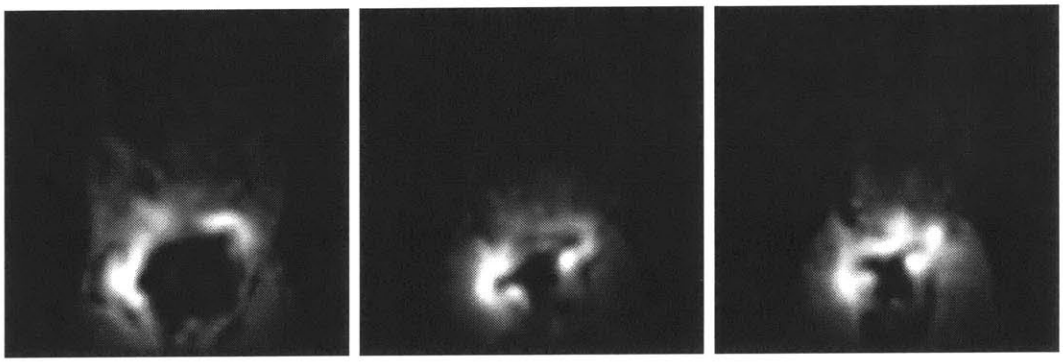
(a)



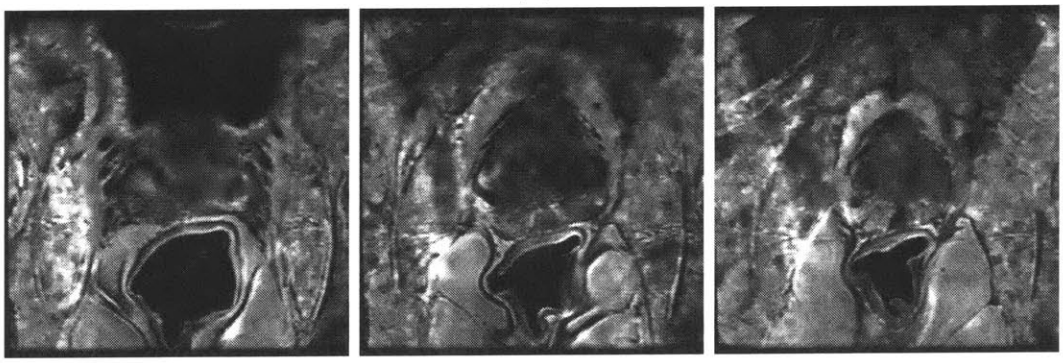
(b)

Figure 4.11. Post-filtered T_2-*weighted* intrinsic image estimates from prostate data set A using ℓ_1 reconstruction. (a) $\gamma = 0.012$, (b) $\gamma = 0.005$.

(a)

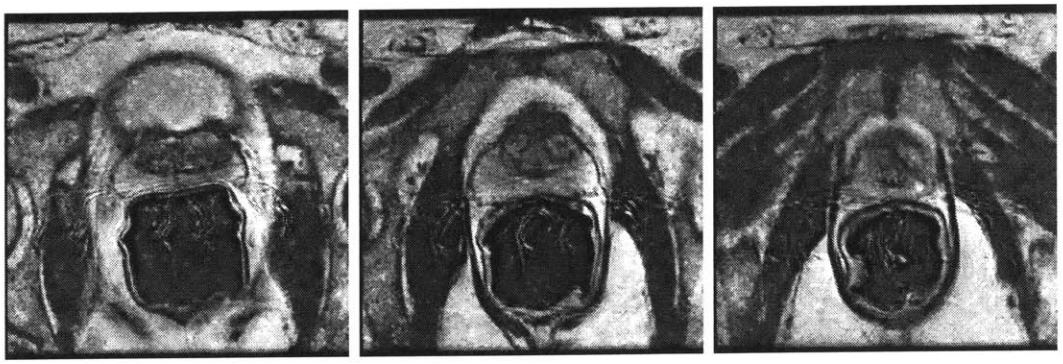


(b)

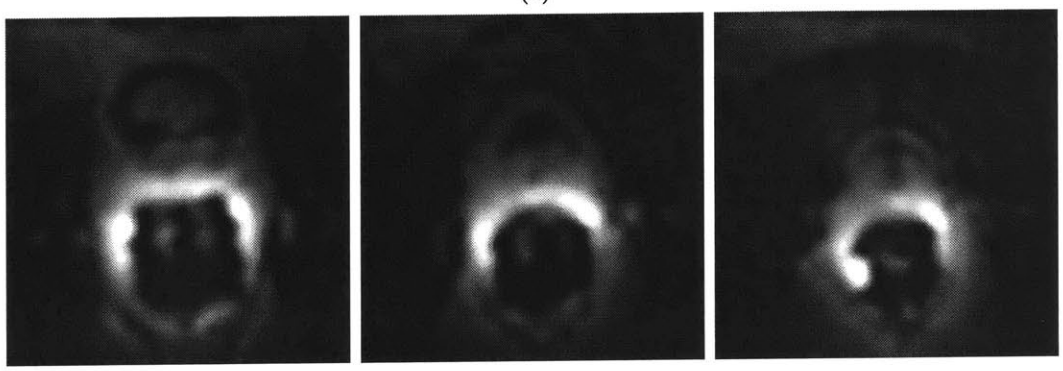


(c)

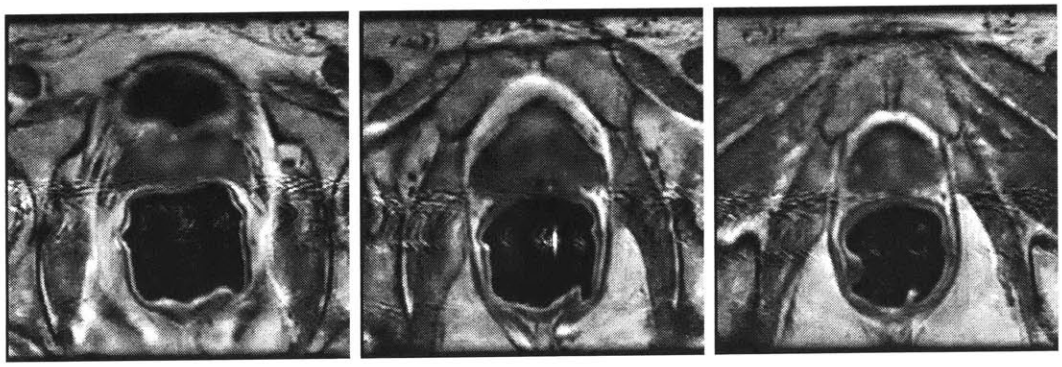
Figure 4.12. Intrinsic image and bias field estimates (with $\hat{\boldsymbol{f}}$ regularization) from prostate data set B. (a) T_2-*weighted* intrinsic images ($\hat{\boldsymbol{f}}_{T_2}$), (b) bias field estimates ($\hat{\boldsymbol{b}}$), and (c) T_1-*weighted* intrinsic images ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 50$, $\gamma_{T_2} = 0.005$, $\gamma_{T_1} = 0.004$.



(a)



(b)



(c)

Figure 4.13. Intrinsic image and bias field estimates (with $\hat{\boldsymbol{f}}$ regularization) from prostate data set C. (a) T_2-*weighted* intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_2}$), (b) bias field estimates ($\hat{\boldsymbol{b}}$), and (c) T_1-*weighted* intrinsic images ($\hat{\boldsymbol{f}}_{T_1}$). $\alpha = 125$, $\gamma_{T_2} = 0.012$, $\gamma_{T_1} = 0.010$.

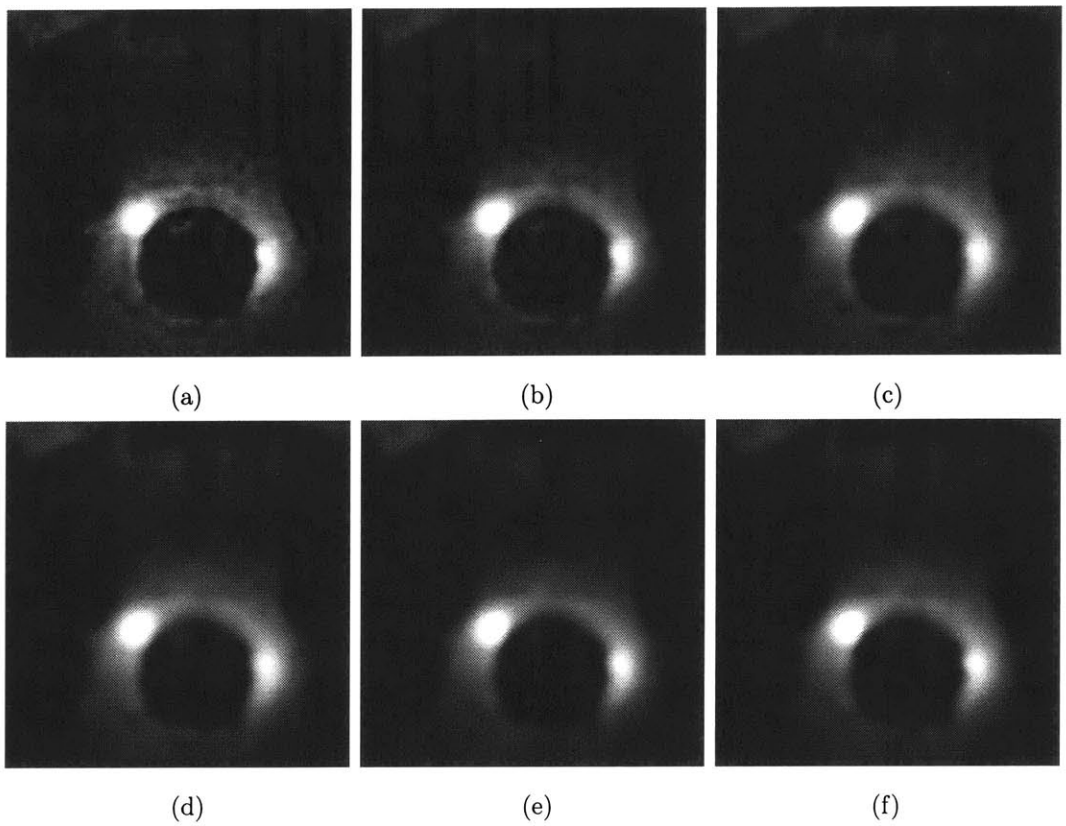
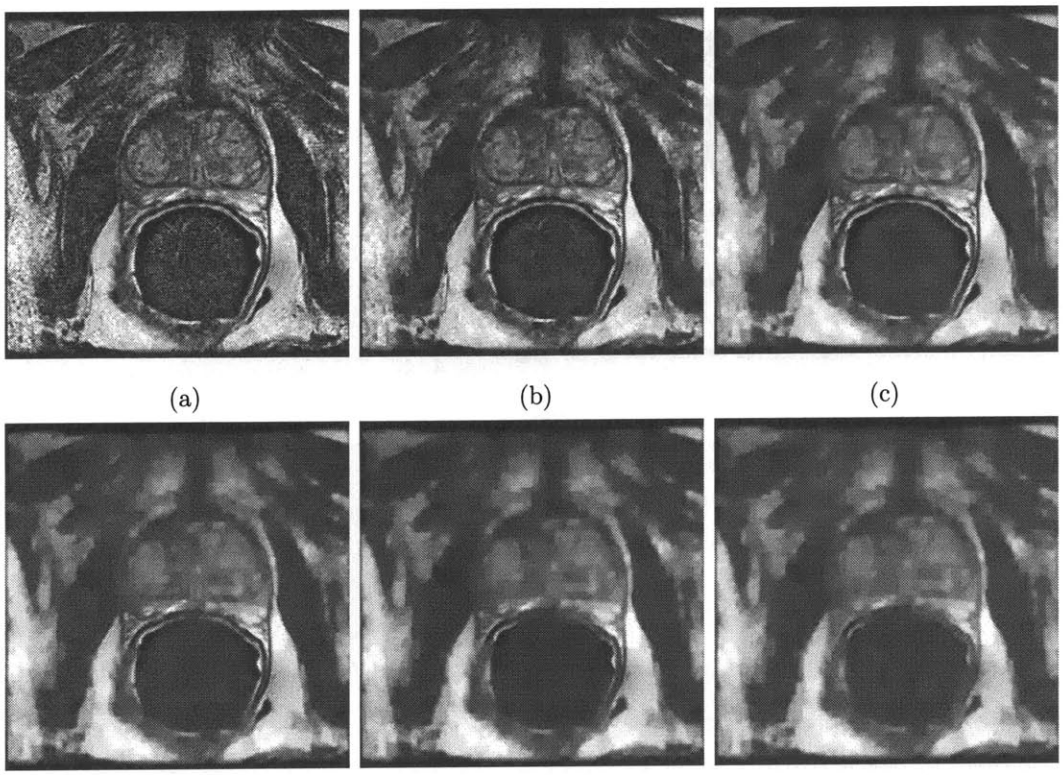


Figure 4.14. Dependence of the bias field estimate ($\hat{\boldsymbol{b}}$) on α (using a slice from data set A). (a) $\alpha = 5$, (b) $\alpha = 15$, (c) $\alpha = 40$, (d) $\alpha = 125$, (e) $\alpha = 375$, and (f) $\alpha = 1200$.

In our experience, $\hat{\boldsymbol{b}}$ is sensitive to changes in α while $\hat{\boldsymbol{f}}$ is not. We show $\hat{\boldsymbol{b}}$ for six different choices of α in Figure 4.14 for one slice from data set A. It is difficult to see the variation because the large bias field near the endorectal coil tends to drown out all other details. The portions of the bias field that are most influenced by the endorectal coil remain largely the same for all of our choices of α. This is because the signal levels are large there so errors in the data fidelity term are punished much more than in other regions. In regions that are farther away, the differences become more apparent. For small α, the bias field is not very smooth. For large α, the regularization term begins to dominate the data fidelity term, and the bias field estimate becomes smoother and conforms less to the noisy observations. The change is more pronounced for small α than large α, something we will quantify in Section 4.4. We do not display



(a) (b) (c)

(d) (e) (f)

Figure 4.15. Variation of prostate intrinsic image estimates ($\hat{\boldsymbol{f}}_{T_2}$) with γ. (a) $\gamma = 0.003$, (b) $\gamma = 0.010$, (c) $\gamma = 0.030$, (d) $\gamma = 0.100$, (e) $\gamma = 0.300$, and (f) $\gamma = 1.00$.

the corresponding $\hat{\boldsymbol{f}}$ because the differences are quite subtle.

In Figures 4.15 and 4.16, we vary γ while holding α fixed and display the resulting corrected images and bias field estimates. The images in Figures 4.15(b) and 4.16(b) are the same as were displayed earlier. As γ increases, the intrinsic image estimates become less noisy but also more blurred. Note that even for the largest choice of γ, the strong edges (*e.g.*, the rectum) remain relatively sharp. Using a ℓ_2 penalty would have destroyed the edges. As mentioned earlier on Page 134, as $\hat{\boldsymbol{f}}$ becomes blurred, some of the edge variation gets assigned to $\hat{\boldsymbol{b}}$. This can be seen in a subtle effect in Figure 4.16. As γ increases (and $\hat{\boldsymbol{f}}$ becomes smoother), the bias field estimate gets lumpier.

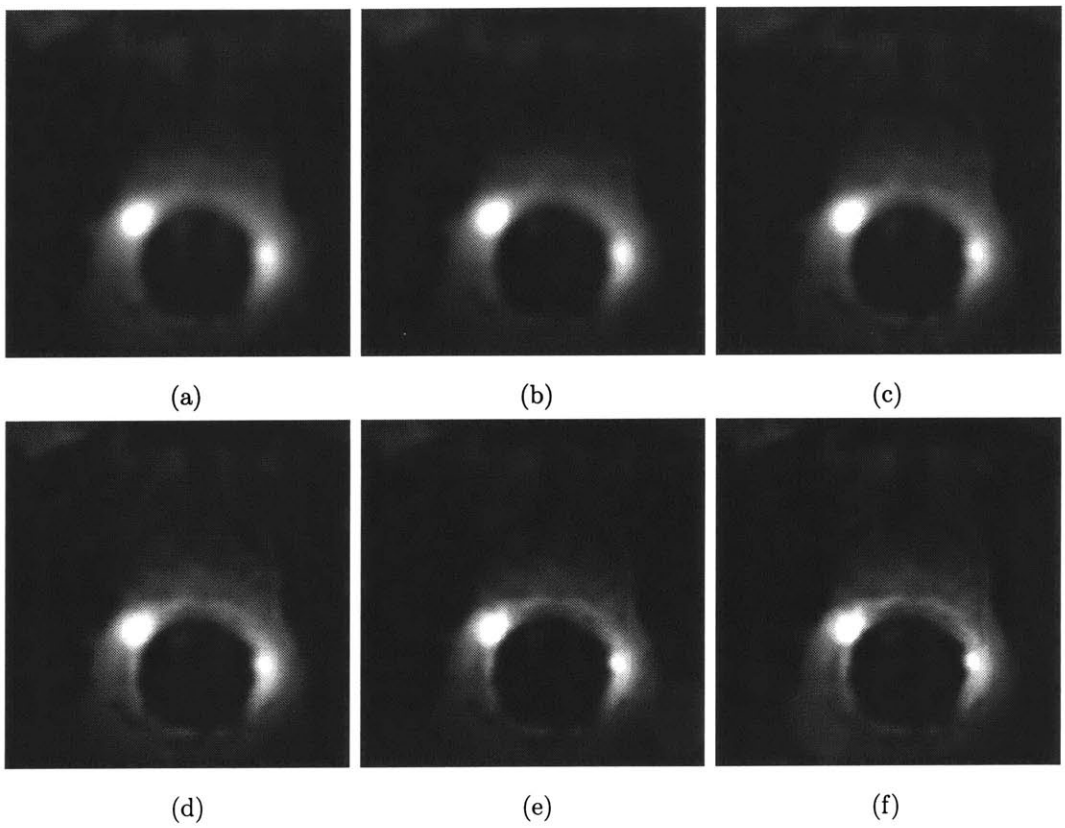


Figure 4.16. Variation of prostate bias field estimates ($\hat{\boldsymbol{b}}$) with γ. (a) $\gamma = 0.003$, (b) $\gamma = 0.010$, (c) $\gamma = 0.030$, (d) $\gamma = 0.100$, (e) $\gamma = 0.300$, and (f) $\gamma = 1.00$.

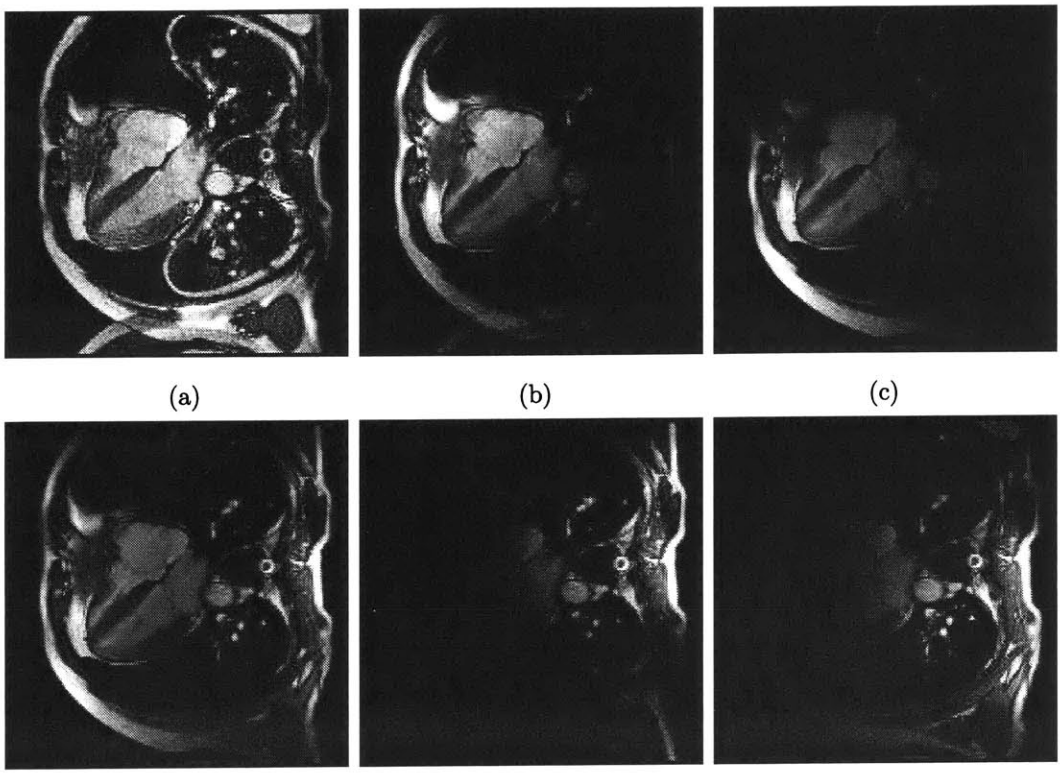


Figure 4.17. Observation images from the cardiac data sequence. (a) Body coil image ($\boldsymbol{y}_{\mathrm{B}}$), (b)–(c), (e)–(f) individual surface coil images ($\boldsymbol{y}_{\mathrm{S},1}$–$\boldsymbol{y}_{\mathrm{S},4}$), and (d) composite surface coil image ($\boldsymbol{y}_{\mathrm{S}}$).

■ 4.2 Cardiac MR

In this section, we explore the application of our algorithm to cardiac image sequences. A time-gated 2D sequence of a heartbeat was captured using a GE Signa 1.5-T machine. FOV was 32 cm × 32 cm and slice thickness was 8 mm. We had available to us body coil images as well as separate images from each of four surface coils arranged in a phased array. Two coils were located on the chest, and two coils were mounted on the back. All results in this section are generated with $\boldsymbol{L}_b$ as a Laplacian operator unless otherwise specified, and $\boldsymbol{L}_f$ as a gradient operator.

In Figure 4.17 we show our five observation images—one body coil image and four surface coil images. We also show the composite surface coil image produced from the

four individual surface coil images using the sum-of-squares method. We only display observations from one time step. All of the images at different times look quite similar with the only variation being movement by the heart. The body coil image is much cleaner compared with the previous prostate examples with a SNR inside the heart of 24 dB. This is mainly due to the larger voxel size. The intensity inhomogeneity in the surface coil image is also much less pronounced than for the prostate. We can see how each surface coil has limited spatial coverage but can be joined together into one image that covers nearly the entire ROI. Note that there are still some regions that have poor response from all of the surface coils (most notably at the top and bottom of the image). The surface coils provide a gain of up to 18 dB though this gain is achieved only in a very small region.

Figure 4.18 shows the reception profiles that we estimate for each surface coil. These estimates are obtained by obtaining the more general energy functional we introduced in Section 3.4.3. The results are largely what we would expect: a strong local response that rapidly diminishes with distance. Note how much larger α is for the heart than the prostate. The bias fields for the heart surface coils are probably similar in smoothness to the bias fields from the pelvic phased array. The difference is that the phased-array bias fields are much smoother than the endorectal coil bias field.

In Figure 4.19 we show the intrinsic image estimates we obtain when applying our algorithm without $\hat{\boldsymbol{f}}$ regularization. We can estimate the intrinsic image from each surface coil image by dividing the surface coil observation by the bias field estimate. This highlights the spatially-varying SNR we obtain with each surface coil image. Note that areas far away from the coil that appear to have no signal in the observation images actually contain fairly accurate (albeit noisy) intensity values. The result of applying the naive (*i.e.*, non-SNR weighted) method of simply averaging the four intrinsic image estimates is shown in Figure 4.19(d). This produces an inferior result that is noisier than the body coil image in most locations. Optimally combining the four intrinsic image estimates with the body coil image results in Figure 4.19(a).

In Figure 4.20, we apply our algorithm to the composite surface coil image. The intrinsic image estimate looks similar to that obtained from the multiple surface coil processing. Figure 4.20(c) shows the areas where the major differences arise. In regions where there is strong surface coil response, the difference is virtually zero. It is in regions far from the surface coils where the largest differences arise. These errors are

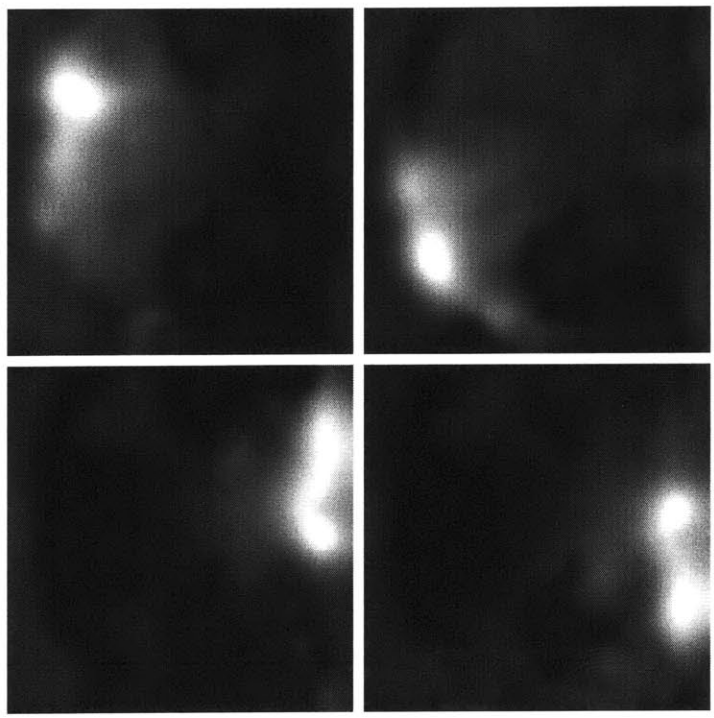


Figure 4.18. Bias field estimates ($\hat{b}_1$–$\hat{b}_4$) with no $\hat{f}$ regularization for the multiple surface coil framework from the cardiac data set. There is one bias field estimate per surface coil. $\alpha = 5000$, $\gamma = 0$.

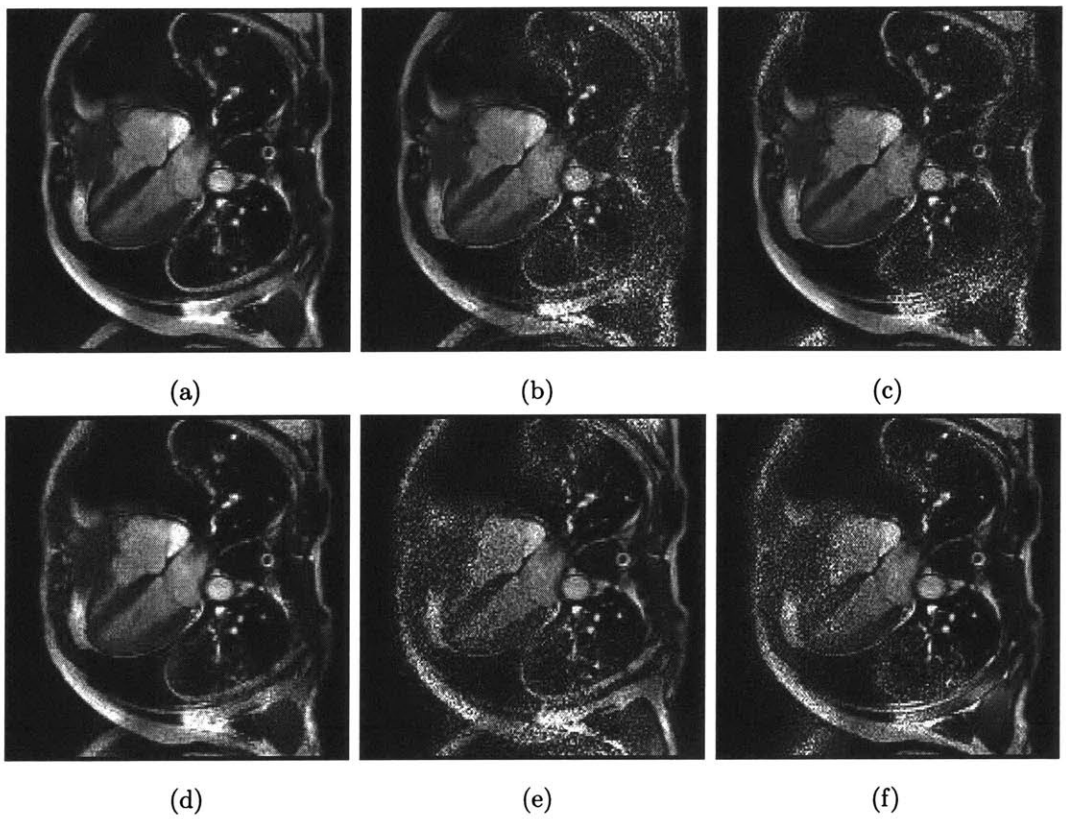
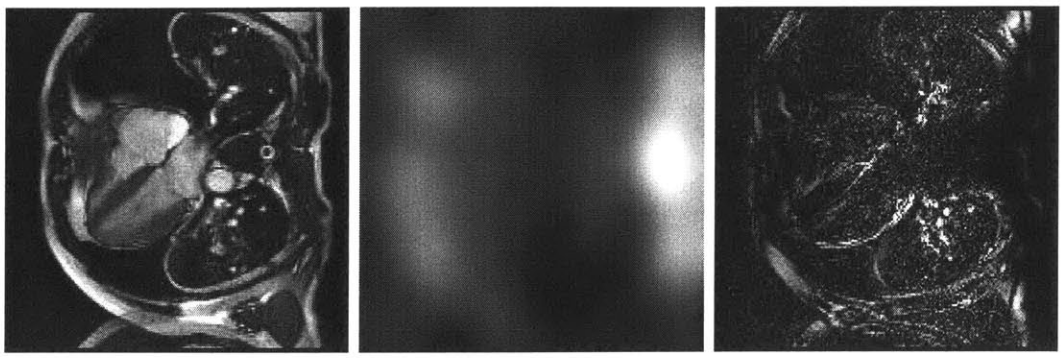


Figure 4.19. Cardiac intrinsic image estimates with no $\hat{\boldsymbol{f}}$ regularization. (a) Final intrinsic image estimate ($\hat{\boldsymbol{f}}$), (b)–(c), (e)–(f) intrinsic image estimates from each surface coil, and (d) average of (b)–(c), (e)–(f). $\alpha_k = 3000$, $\gamma = 0$.



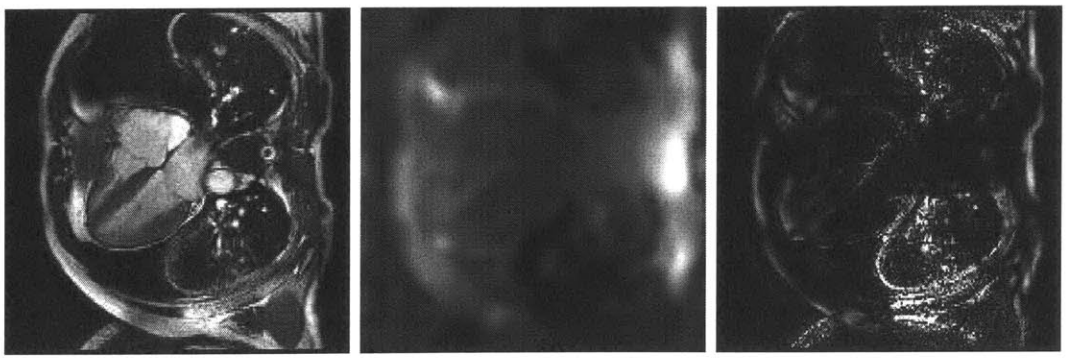
(a) (b) (c)

Figure 4.20. Results from the cardiac data set using the composite surface coil image. (a) Intrinsic image estimate ($\hat{\boldsymbol{f}}$), (b) bias field estimate ($\hat{\boldsymbol{b}}$), and (c) absolute difference between $\hat{\boldsymbol{f}}$ using the multiple surface coil framework and $\hat{\boldsymbol{f}}$ using the composite surface coil image. $\alpha = 30000$, $\gamma = 0$.

quite small on a relative basis. The heart is perhaps not the best example for the independent surface coil processing. The prostate would benefit more due to the large difference in behavior between the endorectal coil and the pelvic phased array coils. The ability to choose $\boldsymbol{L}_b$ and α for each surface coil would be a tremendous boost.

We obtain intrinsic image and composite bias field estimates in Figure 4.21 using Brey-Narayana. Because the body coil image is not used to reconstruct the final image estimate, there are regions where none of the surface coils has a large response and the result is much noisier than we observed in Figure 4.19(a). This is most pronounced at the top and bottom of the image as well as in the middle. This can be seen in Figure 4.21(c) where we show the absolute difference between the Brey-Narayana estimate and our estimate using the separate surface coil images.

We show in Figure 4.22 how our bias field estimates $\hat{\boldsymbol{b}}$ vary as we change α and $\boldsymbol{L}_b$. We only display the results from one of the coils. We can see that for identical values of α, the gradient operator produces bias field estimates that are much smoother than those obtained when using a Laplacian operator. The largest difference occurs outside of tissue regions where the Laplacian operator allows the bias field to quickly drop to zero while the gradient operator extends the bias field pretty far into the air regions. The bias field estimates in air-filled regions are inconsequential and are primarily influenced by the regularization term.



(a) (b) (c)

Figure 4.21. Brey-Narayana applied to the composite surface coil cardiac image. (a) Intrinsic image estimate, (b) bias field estimate, and (c) absolute difference between (a) and $\hat{\boldsymbol{f}}$ using the multiple surface coil framework.

We then display in Figure 4.23 the intrinsic image estimates that correspond to those bias field estimates. The three estimates using Laplacian regularization look roughly the same, though small differences can be seen. For the gradient regularization, $\alpha = 50$ is too small. Too much of the edge information remains in the bias field estimate. On the other hand, $\alpha = 500$ is too large. The most prominent discrepancy occurs in the middle of the patient's back on the right side of the image. The image is too bright there which indicates that the bias field estimates were over-regularized and could not properly incorporate the reception profile peaks.

Figure 4.24 shows the results of our algorithm using $\hat{\boldsymbol{f}}$ regularization. The bias field estimates are largely unchanged from earlier. We can see in Figure 4.24(d) that the largest differences occur, as expected, in the regions with the least surface coil coverage. It also appears that within the heart is moderately less noise than before. We do not display results where we vary γ. The results are largely as expected and analogous to the prostate example in Figure 4.15. When γ is low, the intrinsic image estimate looks like the no $\hat{\boldsymbol{f}}$ regularization result. When γ is large, the images become noise-free but blurred.

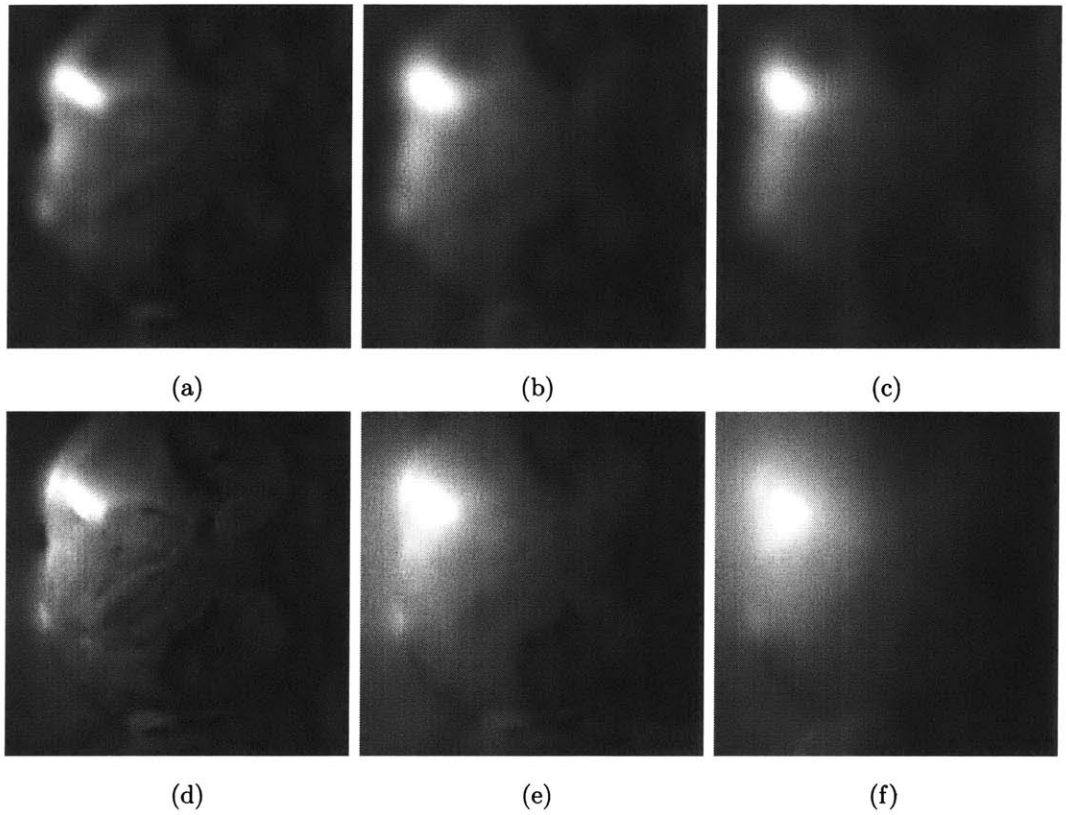


Figure 4.22. Bias field estimates from the cardiac data set with varying regularization on $\hat{\boldsymbol{b}}$. The algorithm was run with all four surface coil images, but results are only presented for one coil. Laplacian regularization with (a) $\alpha_k = 500$, (b) $\alpha_k = 5000$, and (c) $\alpha_k = 50000$. Gradient regularization with (d) $\alpha_k = 50$, (e) $\alpha_k = 500$, and (f) $\alpha_k = 5000$. $\gamma = 0$ for all examples.

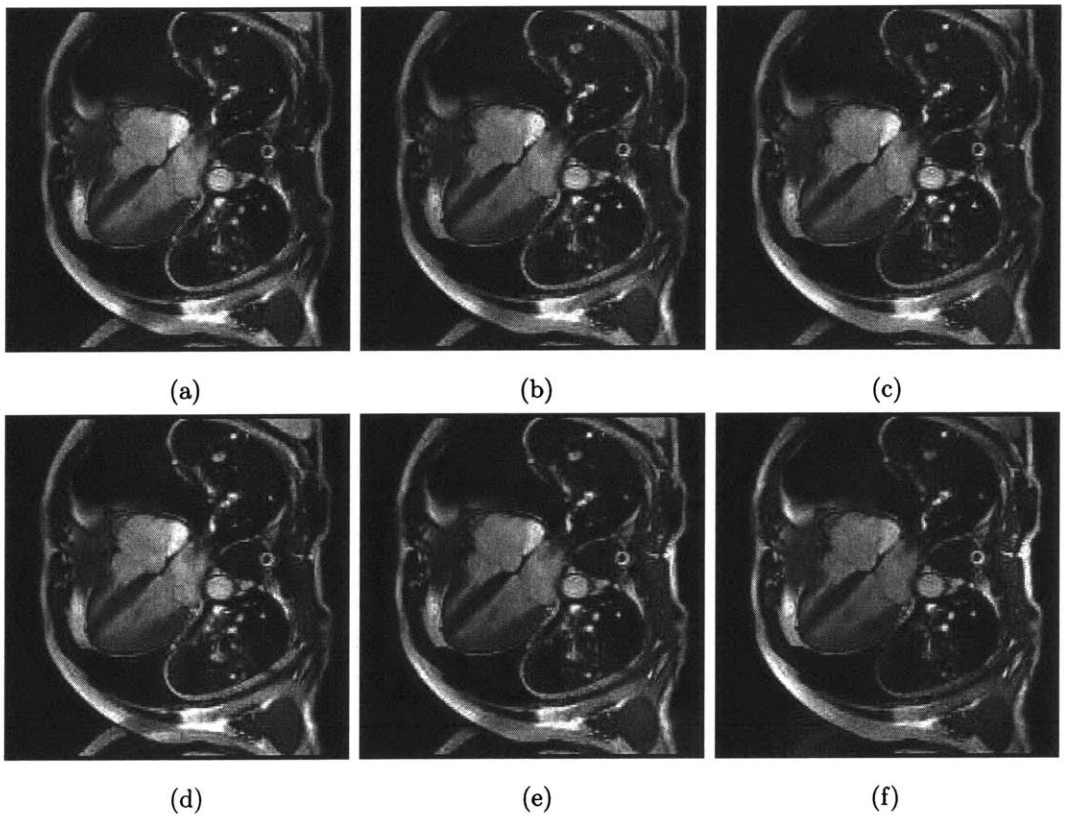


Figure 4.23. True image estimates from the cardiac data set with varying regularization on $\hat{\boldsymbol{b}}$. Laplacian regularization with (a) $\alpha_k = 500$, (b) $\alpha_k = 5000$, and (c) $\alpha_k = 50000$. Gradient regularization with (d) $\alpha_k = 50$, (e) $\alpha_k = 500$, and (f) $\alpha_k = 5000$. All examples have $\gamma = 0$.

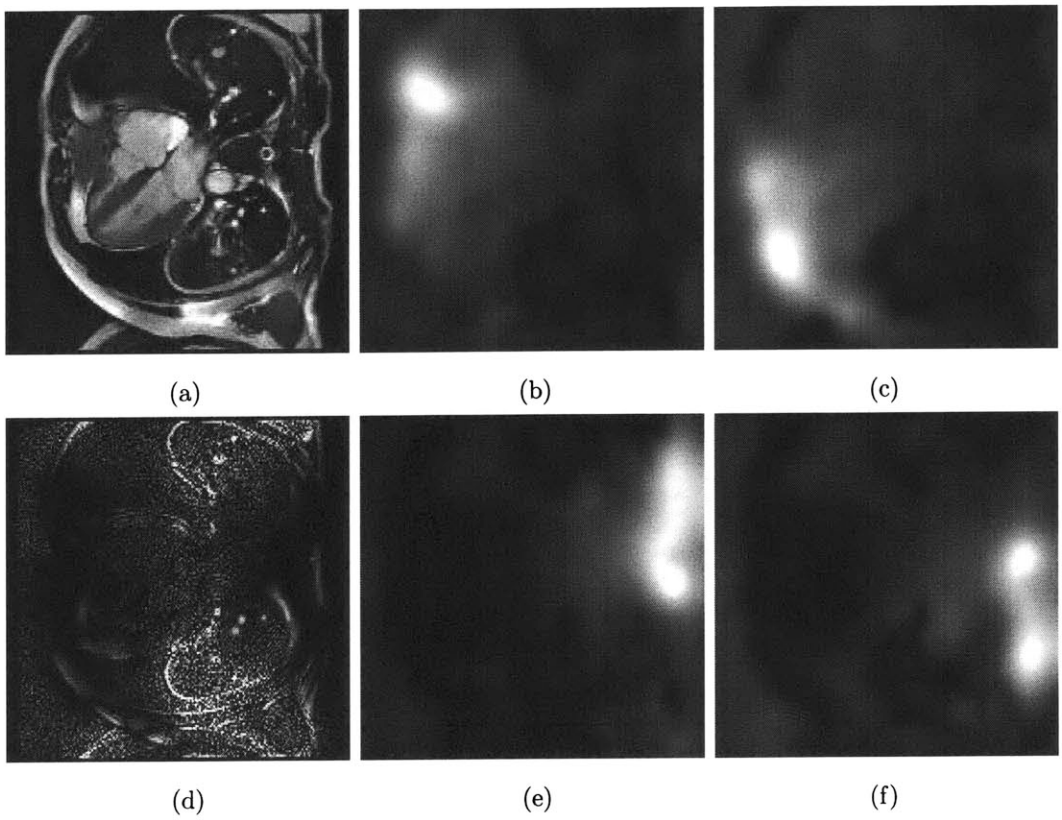


Figure 4.24. Cardiac intrinsic image and bias field estimates with $\hat{\boldsymbol{f}}$ regularization. (a) Final intrinsic image estimate ($\hat{\boldsymbol{f}}$), (b)–(c), (e)–(f) bias field estimates for each surface coil ($\hat{\boldsymbol{b}}_1$–$\hat{\boldsymbol{b}}_4$), and (d) absolute difference of intrinsic image estimate with result from no $\hat{\boldsymbol{f}}$ regularization. $\alpha_k = 3000$, $\gamma = 0.00055$.

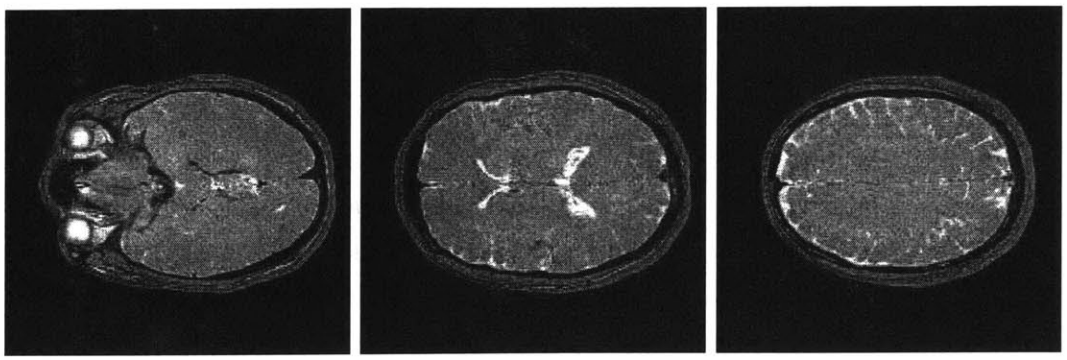


Figure 4.25. Body coil GRE images ($\boldsymbol{y}_{\mathrm{B}}$) from the brain data set.

■ 4.3 Brain Imaging

We apply our bias correction method to real brain images in this section. We captured a full 3D volume using a four-element phased array. We received gradient echo (GRE) images with both the phased array and the body coil, and we received fluid attenuated inversion recovery (FLAIR) fast-spin echo (FSE) using just the phased array. The latter is a modified T_2-*weighted* pulse sequence that suppresses fluid intensities. This is useful in *e.g.*, multiple sclerosis (MS) diagnosis where the cerebrospinal fluid (CSF) brightness can overwhelm the MS lesions. FOV was 24 cm $\times$ 24 cm and slice thickness was 3 mm. All results were generated with $\boldsymbol{L}_b$ implemented as a Laplacian operator and $\boldsymbol{L}_f$ as a gradient operator. We do not present results using the composite surface coil because we feel that the multiple surface coil framework provides superior results. We index the GRE surface coils as $\boldsymbol{y}_{\mathrm{S},1}$ through $\boldsymbol{y}_{\mathrm{S},4}$ and the FLAIR surface coil images as $\boldsymbol{y}_{\mathrm{S},5}$ through $\boldsymbol{y}_{\mathrm{S},8}$.

We display observation images from three slices in the brain. Figure 4.25 contains the GRE body coil images. The image quality is comparable to the cardiac images with a SNR of 21 dB in the brain. The corresponding GRE images for each of the surface coils is located in Figure 4.26. The gain of the coils is as high as 17 dB. Figure 4.27 contains the FLAIR images from each of the surface coils. The SNR is up to 33 dB. Knowing the gain of the surface coils, we can estimate that a hypothetical FLAIR body coil image would have SNR of approximately 16 dB. Note that the FLAIR images reverse the typical intensities observed for gray/white matter so that the gray matter

actually has higher intensity than the white matter.

4.3.1 Bias Correction Results

We present in Figure 4.28 the FLAIR and GRE intrinsic image estimates with $\gamma_{\text{GRE}} = 0$ and $\gamma_{\text{FLAIR}} = 0$. The GRE results have better noise than the body coil images and much better homogeneity than the surface coil images. The FLAIR images also display markedly improved homogeneity, but in the middle of the brain, the images become quite noisy. The reason for this is that all of the surface coils have relatively poor signal response in the middle of the brain, and we do not have a body coil image to help. The noise in the middle of $\hat{\boldsymbol{f}}_{\text{FLAIR}}$ is better than would be available from any of the individual surface coils, but it is still relatively large.

Figure 4.29 shows our bias field estimates that correspond to the intrinsic image estimates that we just covered. There is an artifact in the slices in the left column that emanates from the eyeballs. This artifact can also be seen in the original GRE surface coil and body coil images but not in the FLAIR images. This imperfection does not noticeably alter the final correction results because it appears outside of the head. The reception profiles of the surface coils appear to be quite stable in the vertical direction with little variation from slice to slice.

We show in Figure 4.30 images corrected using regularization on $\hat{\boldsymbol{f}}$. We do not include the bias field estimates because they only change marginally from the earlier results. The noise is reduced from the no regularization case, especially in the GRE images. The FLAIR images are not able to benefit as much due to the preponderance of small-scale structure. Though γ_{FLAIR} is only 20% lower than γ_{GRE}, the regularization in the FLAIR images is actually only half as strong as that in the GRE images due to the differing intensity levels. We conclude that it is more beneficial to capture a body coil image of the pulse sequence that results in the most fine-scale structure because this scan benefits the least from the ℓ_1 reconstruction.

We display in Figure 4.31 the absolute difference between reconstructions obtained with and without $\hat{\boldsymbol{f}}$ regularization. We can see that the amount of correction steadily increases the closer we get to the middle of the brain. This is because the SNR decreases in the middle of the brain. This effect is more pronounced in the FLAIR images than in the GRE images because the latter have a floor on the SNR level imposed by the body coil image. We can see that the ℓ_1 regularization does a good job of preserving

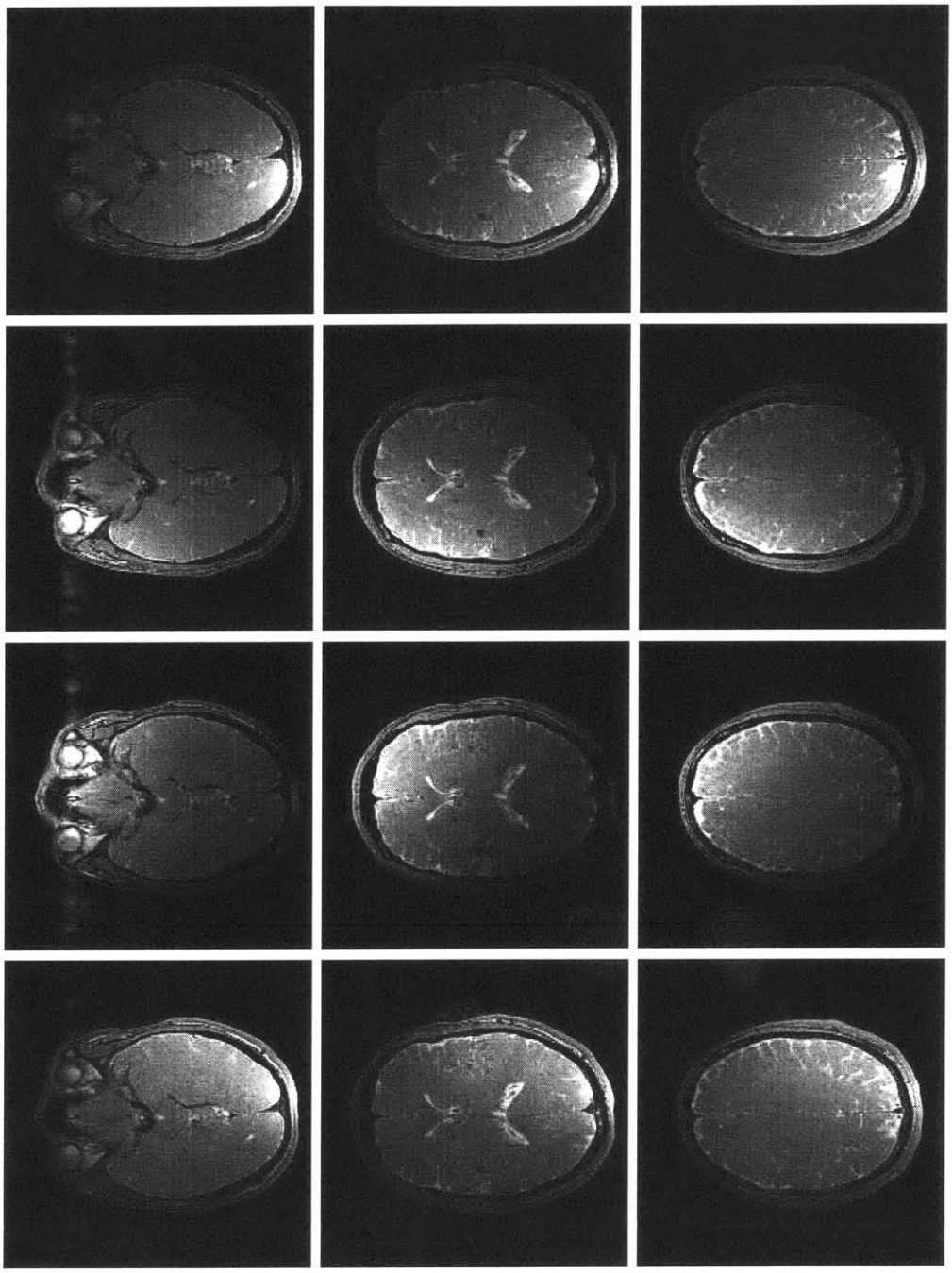


Figure 4.26. Individual axial surface coil GRE images from the brain data set. Each column shows the four surface coil images ($\boldsymbol{y}_{S,1}$–$\boldsymbol{y}_{S,4}$) for each of three slices.

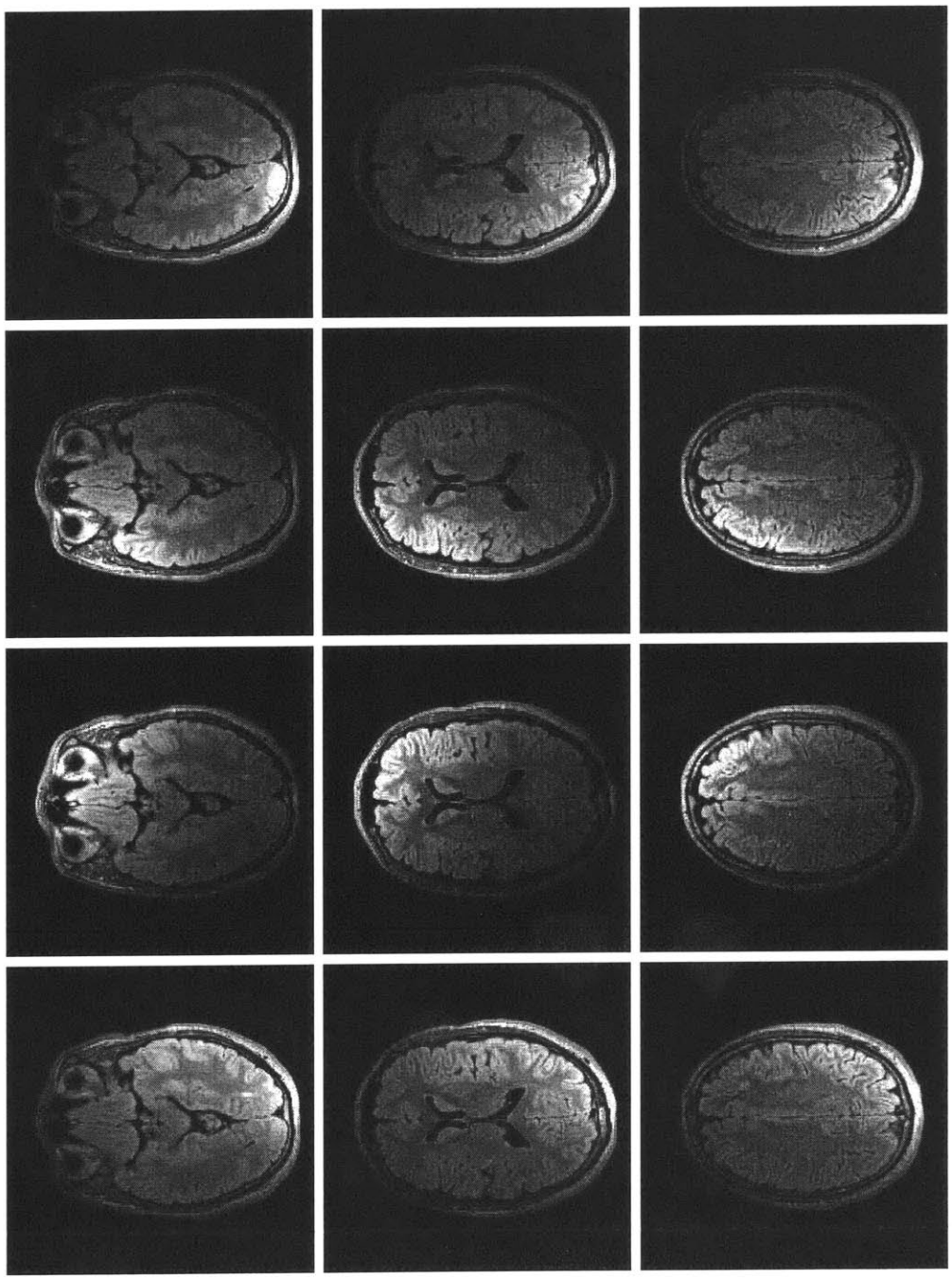
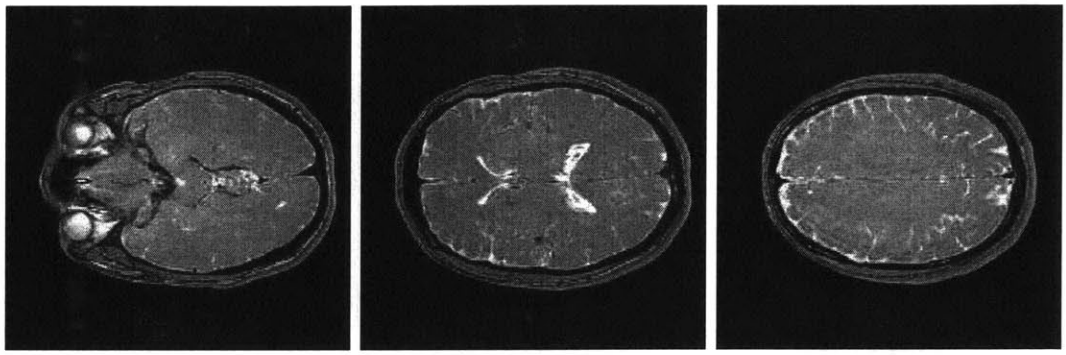
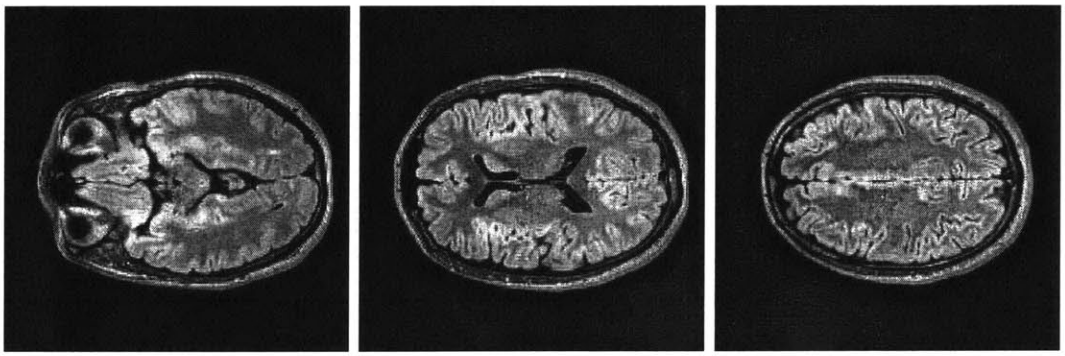


Figure 4.27. Individual FLAIR surface coil images from the brain data set. Each column shows the four surface coil images ($\boldsymbol{y}_{\mathrm{S},5}$–$\boldsymbol{y}_{\mathrm{S,FL},8}$) for each of three slices.



(a)



(b)

Figure 4.28. Intrinsic image brain estimates with no $\hat{f}$ regularization. (a) GRE images ($\hat{f}_{\mathrm{GRE}}$) and (b) FLAIR images ($\hat{f}_{\mathrm{FLAIR}}$). $\alpha_k = 1000$, $\gamma_{\mathrm{GRE}} = 0$, $\gamma_{\mathrm{FLAIR}} = 0$.

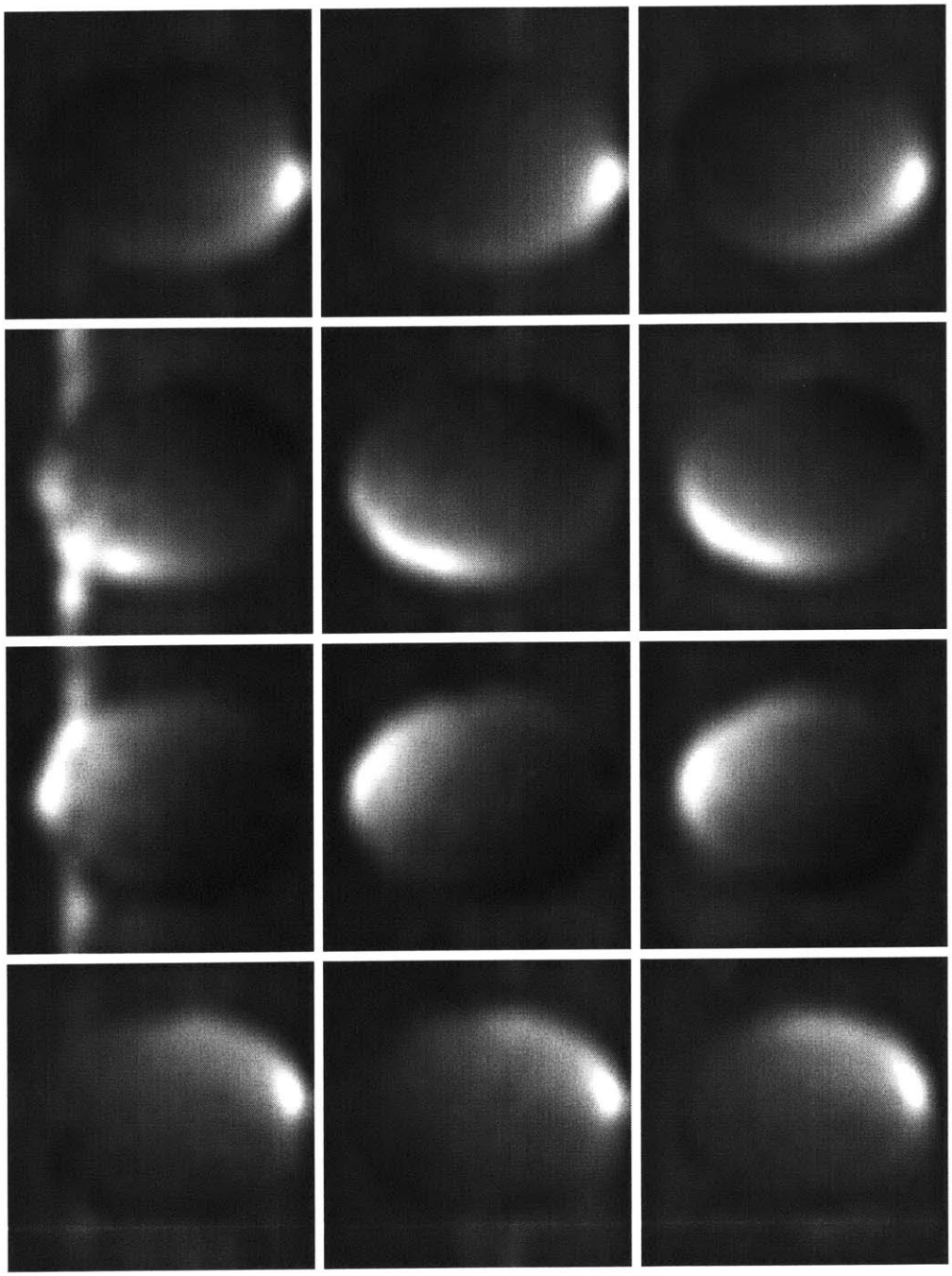
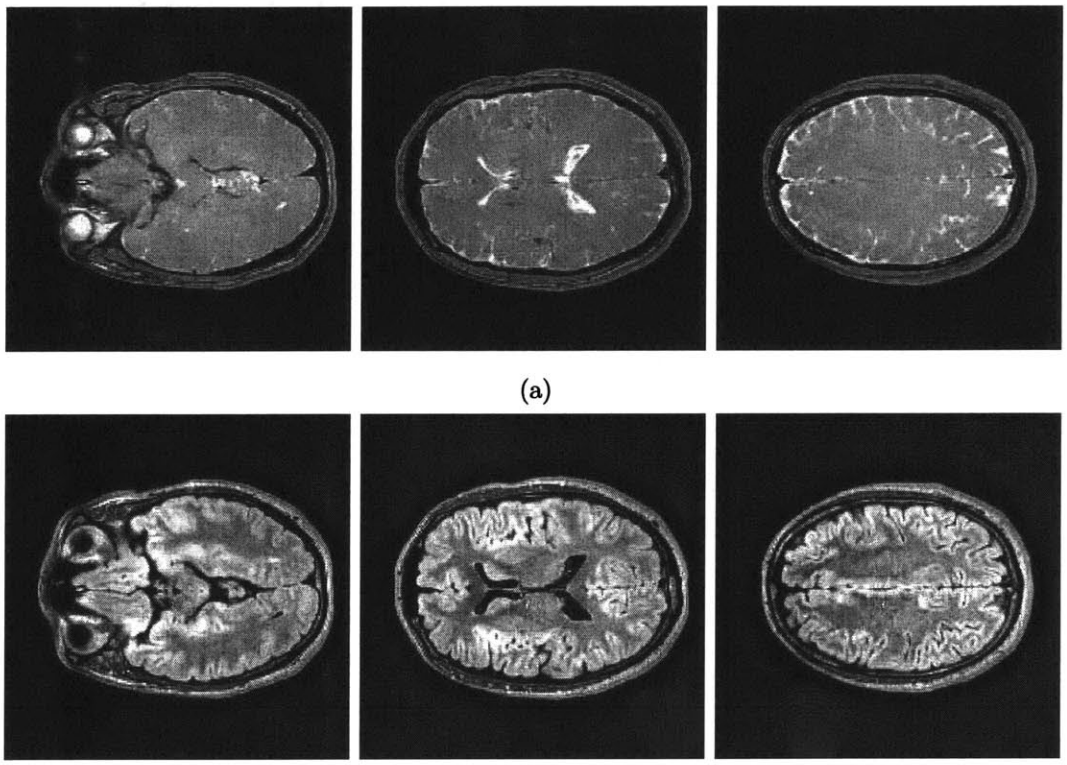


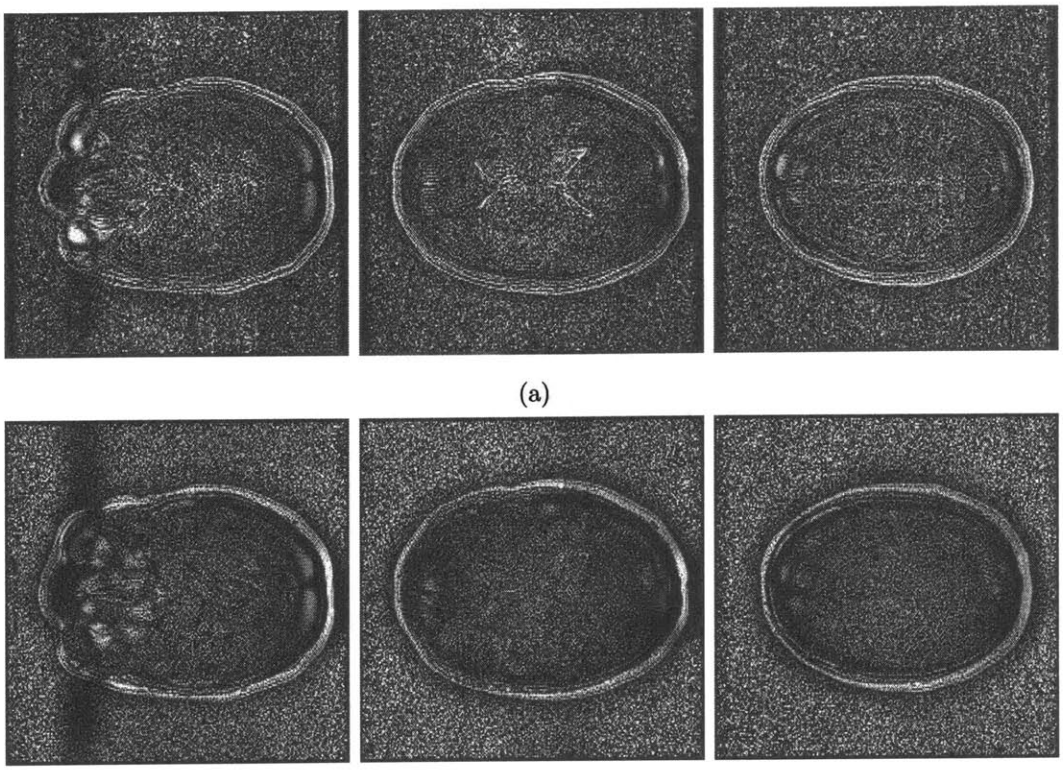
Figure 4.29. Individual bias field estimates ($\hat{\boldsymbol{b}}_1$–$\hat{\boldsymbol{b}}_4$) from the brain data with no $\hat{\boldsymbol{f}}$ regularization. Computed from the GRE body coil and surface coil observations. $\alpha_k = 1000$, $\gamma_{\text{GRE}} = 0$, $\gamma_{\text{FLAIR}} = 0$.



(a)

(b)

Figure 4.30. Corrected images (with $\hat{f}$ regularization) from the brain data set. (a) GRE images ($\hat{f}_{\mathrm{GRE}}$) and (b) FLAIR images ($\hat{f}_{\mathrm{FLAIR}}$). $\alpha_k = 1000$, $\gamma_{\mathrm{GRE}} = 0.025$, $\gamma_{\mathrm{FLAIR}} = 0.020$.



(a)

(b)

Figure 4.31. Absolute difference of brain images for reconstruction with and without $\hat{f}$ regularization. (a) GRE images and (b) FLAIR images. $\alpha_k = 1000$, $\gamma_{\text{GRE}} = 0.025$, $\gamma_{\text{FLAIR}} = 0.020$.

edges as little tissue structure appears in the image differences except in the middle at the ventricles.

■ 4.3.2 Segmentation results

There are a variety of automatic segmentation techniques available in the literature. Perhaps the simplest technique is statistical ML segmentation with an IID Gaussian noise assumption. This results in a segmentation that is only based on thresholding—if an intensity value falls within a proscribed range $[a_k, b_k)$, it is assigned to the segmentation class k. This technique can only be successful when the true tissue intensities are approximately piecewise constant, and the noise and bias are low. Brain segmentation falls into this category, though even when using a head coil, adaptive segmentation

techniques such as Wells *et al.* [85] must be used to account for the bias.

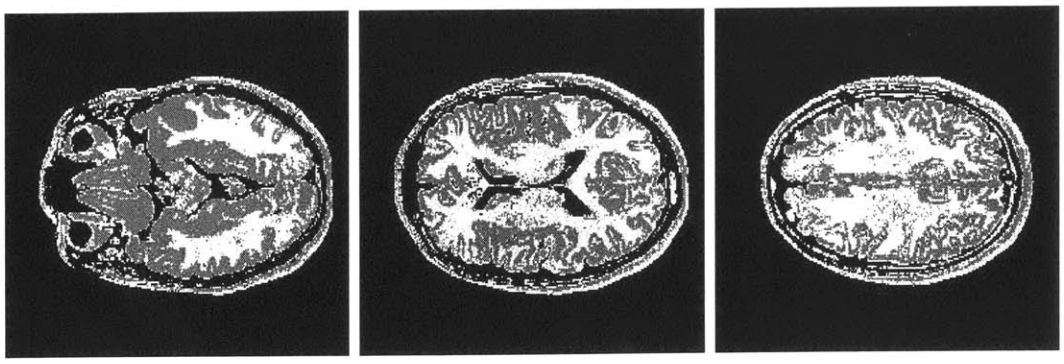
We show in Figure 4.32 thresholded white/gray segmentations of images corrected using three methods: our algorithm without regularization on $\hat{\boldsymbol{f}}$ (see Figure 4.28), our algorithm with regularization on $\hat{\boldsymbol{f}}$ (see Figure 4.30), and Brey-Narayana. We did not include the intrinsic image estimates computed using Brey-Narayana because they did not differ significantly from our results. The thresholds were chosen manually, and the same thresholds were used for all three methods. Note that we use white intensities for white matter and gray intensities for gray matter even though the relative intensities are reversed in the observed FLAIR images.

The segmentations that we present are limited in quality by the homogeneity of the body coil. Even the body coil does not present a perfectly homogeneous response, and we found it difficult to choose one threshold that would provide consistent white matter thickness. Our results are impacted even more because we do not have a body coil image for the GRE images, so all we really have is a "second-hand" bias field estimate. Partial volume blurring effects also affect the quality of the results.

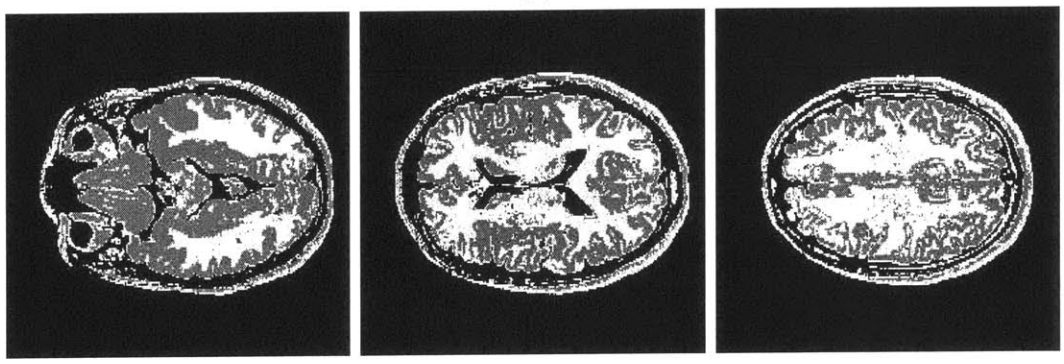
All three segmentations behave similarly except for subtle differences. The segmentations which used regularization on $\hat{\boldsymbol{f}}_{\text{FLAIR}}$ have fewer isolated classified pixels (which are generally attributable to noise). This effect is most pronounced in the middle of the brain due to the higher noise levels there. The ability of the ℓ_1 regularization to preserve edges is highlighted by these segmentations. Even the gray/white boundaries in the folds of the brain are largely preserved. Overall using regularization on $\hat{\boldsymbol{f}}_{\text{FLAIR}}$ produces a modestly cleaner segmentation without sacrificing edge fidelity.

Varying γ_{FLAIR} has the expected results on our intrinsic image estimates: lower values result in more noise while higher values result in blurry images. We can get a clearer picture of these contrasts by presenting in Figure 4.33 the thresholded segmentation results as we alter γ_{FLAIR}. We only include results from one slice. Figure 4.33(b) corresponds to the γ_{FLAIR} value used to generate the previous segmentation maps.

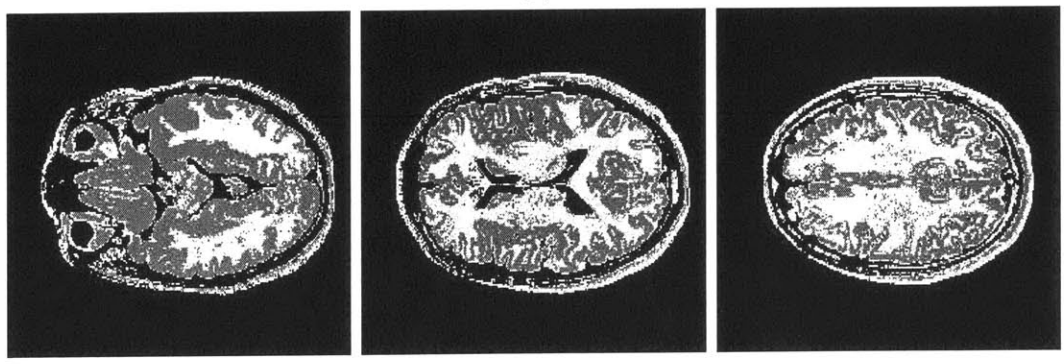
As γ_{FLAIR} becomes large and we over-regularize the solution, the regions become more contiguous. This has the effect of breaking apart some loosely connected regions (that should be connected) and closing up some smaller regions (that should not be closed). Overall, even with the high amount of regularization, many small structures are preserved, and it is arguable that Figure 4.33(d) may be a better segmentation than Figure 4.33(b).



(a)



(b)



(c)

Figure 4.32. Thresholding-based segmentation results from corrected brain FLAIR images. Correction using (a) our algorithm without $\hat{f}$ regularization, (b) our algorithm with $\hat{f}$ regularization ($\gamma_{\text{FLAIR}} = 0.020$), and (c) Brey-Narayana.

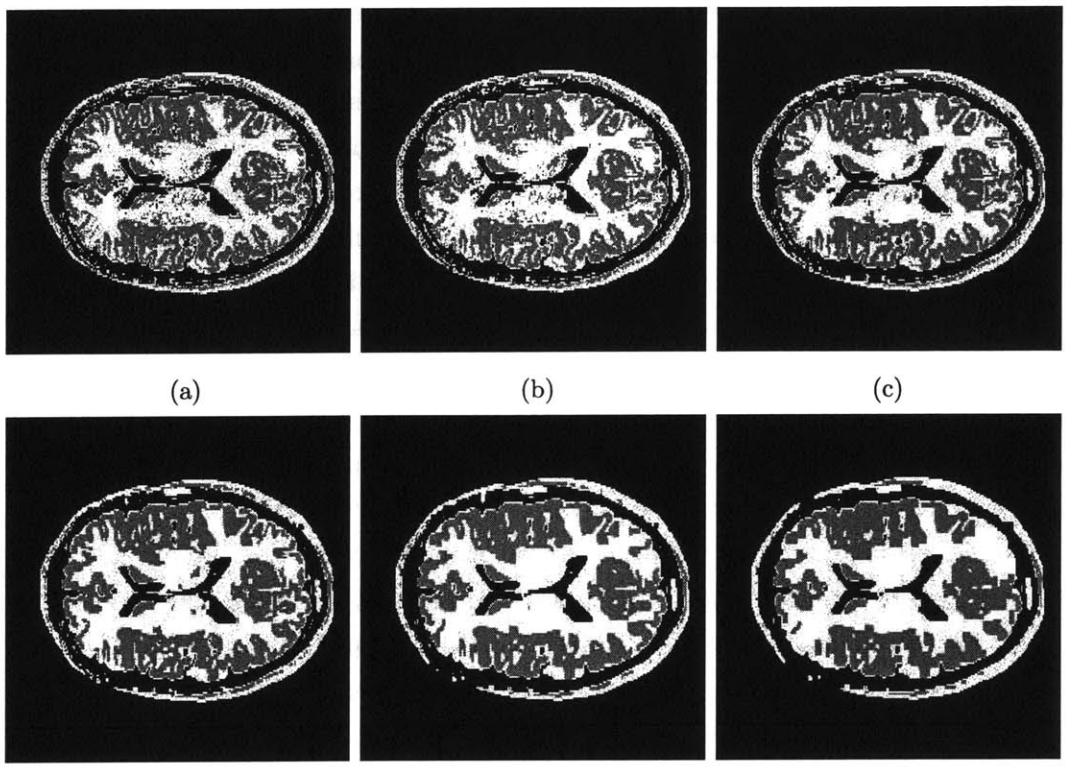


Figure 4.33. Thresholding-based segmentation results for corrected brain FLAIR images with (a) $\gamma_{\text{FLAIR}} = 0.008$, (b) $\gamma_{\text{FLAIR}} = 0.020$, (c) $\gamma_{\text{FLAIR}} = 0.060$, (d) $\gamma_{\text{FLAIR}} = 0.180$, (e) $\gamma_{\text{FLAIR}} = 0.50$, and (f) $\gamma_{\text{FLAIR}} = 1.50$. $\alpha_k = 1000$.

4.4 Phantom Brain Images

In this section, we create artificial body coil and surface coil images using the MNI brain phantom. We use the T_1-*weighted* "normal" volume with 1 mm $\times$ 1 mm pixel size and 3 mm slice thickness. We generate artificial bias fields by convolving a point source with a Gaussian. We construct the bias fields so that they model a four-element phased array that has its coils roughly equally spaced along the outside of the head. Noise-free surface coil images are then generated according to our imaging model, and the observation images are created by adding Rician noise. We chose the noise level to provide SNR of 24 dB in the gray matter regions and 26 dB in the white matter regions. All results were generated with $\boldsymbol{L}_b$ as a Laplacian operator and $\boldsymbol{L}_f$ as a gradient operator. The maximum gain of our bias field is 5.78 which represents a gain of 15 dB.

We display our constructed observation images in Figure 4.34. The images are somewhat artificial in appearance, but are reasonable facsimiles of the real brain data we used in Section 4.3. The composite surface coil image is generated using the sum-of-squares method. Combining all of the surface coil images provides fairly comprehensive coverage of the entire tissue region. The lowest gain within the brain of the composite surface coil image is 0 dB which is equivalent to the body coil gain.

In Figure 4.35 we show the segmentation map we obtain from the ground truth intrinsic image. This segmentation map will later be used to evaluate the effectiveness of our bias correction method. The thresholds were manually chosen to give the best visual appearance, and non-brain structure (such as from the skull) were manually removed. The threshold values are somewhat arbitrary because the boundaries between regions are not step edges. This is because the brain phantom accurately models partial volume effects, so there is significant blurring at tissue boundaries.

4.4.1 Qualitative Results

Figure 4.36 contains the results of our bias correction algorithm with $\gamma = 0$ and α chosen to minimize the mean squared error of the intrinsic image estimate. We show the absolute error (compared with the ground truth) of our true image estimate in Figure 4.36(d). We mask out the non-tissue region because the error there (mainly due to the bias introduced by the Rician noise) is so much larger than the error in tissue regions. We can see that the error diminishes in regions where our surface coils provide

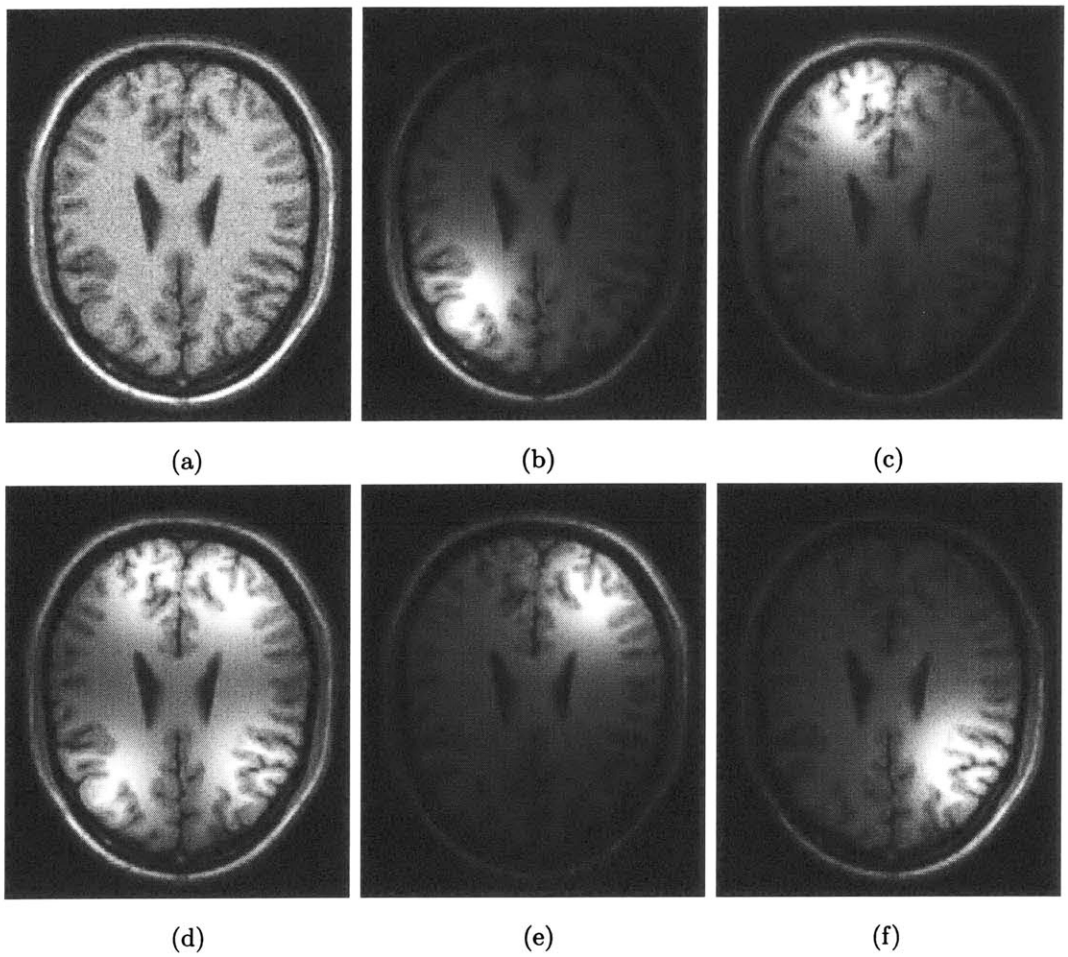


Figure 4.34. (a) Body coil image ($\boldsymbol{y}_{\mathrm{B}}$), (b)–(c), (e)–(f) individual surface coil images ($\boldsymbol{y}_{\mathrm{S}}1$–$\boldsymbol{y}_{\mathrm{S}}4$), and (d) composite surface coil image from the MNI data set.

strong signal response.

Figure 4.37 contains the results of our algorithm when minimizing the full energy functional. The bias field results are largely the same as before, and the intrinsic image estimate looks similar as well (albeit with visibly reduced noise). The effects of the ℓ_1 regularization can be seen in Figure 4.37(d) where we show the absolute error. There are still regions where the noise is clearly lower than other regions, but unlike Figure 4.36(d), these regions occur for large piecewise constant areas. The reason for this is that in these areas, our ℓ_1 prior behaves identically to a ℓ_2 prior and most of the noise

Figure 4.35. Gray matter/white matter ground truth segmentation of a slice from the MNI data set.

is eliminated. Conversely, on an edge, our ℓ_1 prior has little effect and the noise (which is the main source of error) will remain largely unchanged.

■ 4.4.2 Numerical Results

As mentioned earlier, the biggest draw for using the brain phantom is the ability to compute numerical results. We computed errors and biases for various bias correction methods in Table 4.1. The bias correction methods we used were our algorithm with the individual surface coil images with (MSC $\hat{\boldsymbol{f}}$, $\gamma = 0.014$) and without (MSC $\hat{\boldsymbol{f}}$, $\gamma = 0$) regularization on $\hat{\boldsymbol{f}}$; our algorithm on the composite surface coil image without using regularization on $\hat{\boldsymbol{f}}$ (comp. $\hat{\boldsymbol{f}}$, $\gamma = 0$); and Brey-Narayana with (opt. B-N) and without (B-N) the body coil in the final image reconstruction. We did not display the corrected images obtained from the last three methods in this section because there are not large visual deviations from the results already included.

Mean squared error and mean absolute error give two slightly different views of the error. Mean absolute error weights all errors equally while mean squared error weights large errors more heavily. As we would expect, the largest errors occur for Brey-Narayana without the body coil, and the smallest occur for our algorithm with the ℓ_1 prior. We can see that we achieve a 24% reduction in mean absolute error from the worst method to the best method. 12% of the reduction results from using the

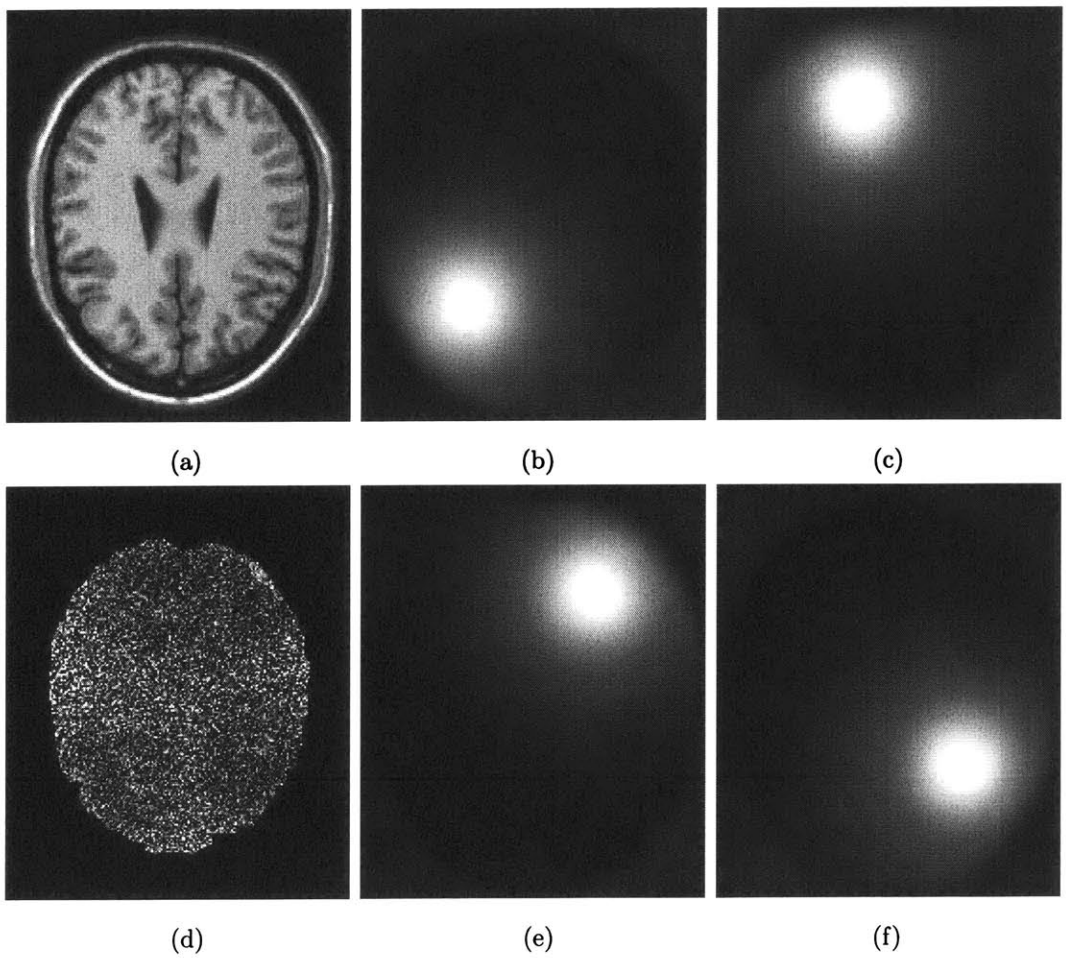


Figure 4.36. (a) Intrinsic image estimate ($\hat{\boldsymbol{f}}$), (b)–(c), (e)–(f) individual coil profile estimates ($\hat{\boldsymbol{b}}_1$–$\hat{\boldsymbol{b}}_4$), and (d) absolute error (within the brain) of the intrinsic image estimate. $\alpha_k = 2000$, $\gamma = 0$.

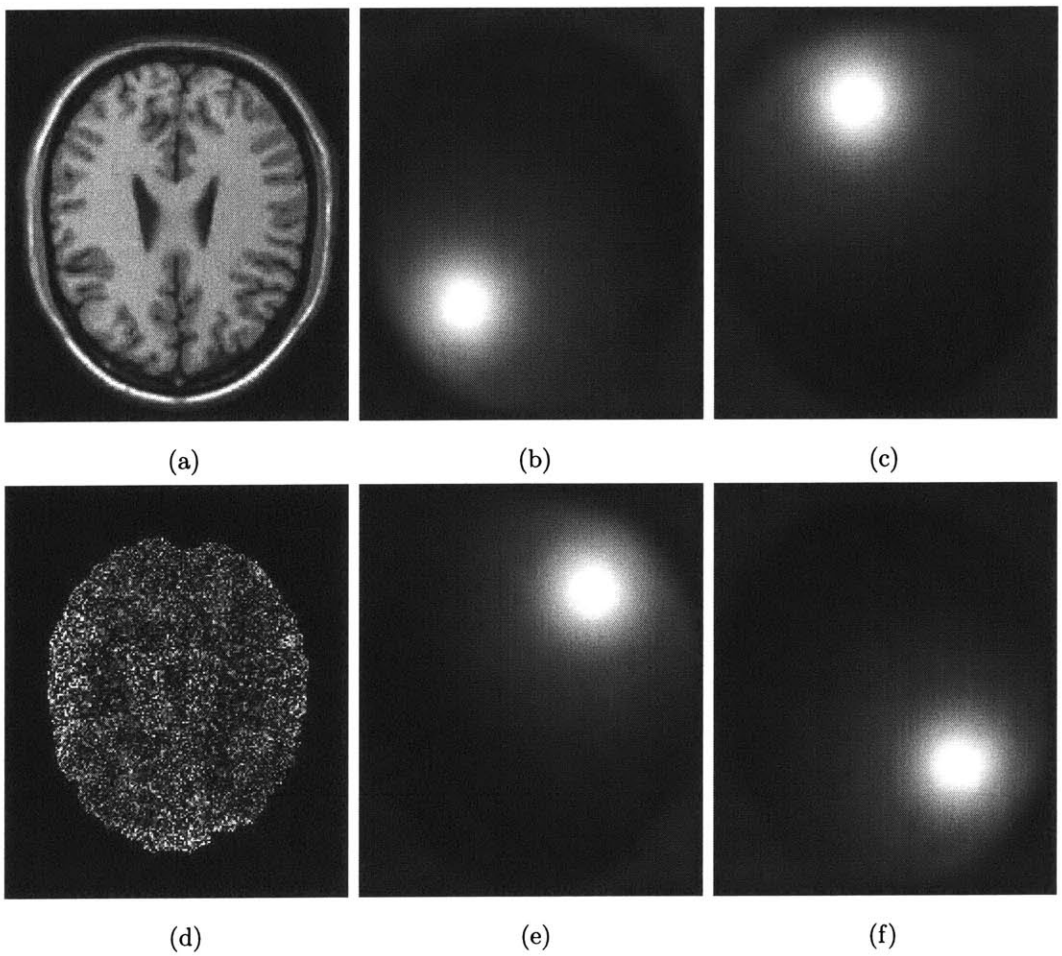


Figure 4.37. (a) Regularized intrinsic image estimate ($\hat{\boldsymbol{f}}$), (b)–(c), (e)–(f) individual coil profile estimates ($\hat{\boldsymbol{b}}_1$–$\hat{\boldsymbol{b}}_4$), and (d) absolute error (within the brain) of intrinsic image estimate. $\alpha_k = 2000$, $\gamma = 0.014$.

	f^*	y_B	comp. $\hat{f}$ $\gamma = 0$	MSC $\hat{f}$ $\gamma = 0$	MSC $\hat{f}$ $\gamma = 0.014$	B-N	opt. B-N
MSE (tissue)	0	14449	1144	1066	802	1318	1233
MAE (tissue)	0	96	26.79	25.53	21.91	28.68	27.87
GM bias	0	5.61	4.54	7.28	3.23	2.18	2.72
WM bias	0	2.31	4.61	6.00	-0.61	6.62	6.62
GM errors	0	1393	346	328	323	377	364
WM errors	0	823	198	189	188	212	208

Table 4.1. Quantitative results comparing several bias correction methods on the MNI brain phantom. The top two lines are the mean squared error and mean absolute error (compared with f^*) of the part of the images in tissue regions. The next two lines are the intensity bias in gray matter and white matter regions. The last two lines are the segmentation errors made in gray matter and white matter regions.

body coil data in the final reconstruction, 16% results from our smoother bias field estimates, 19% occurs from using the individual surface coil images separately, and 53% happens due to the noise reduction from the ℓ_1 prior. Of course this is only a rough approximation as these effects are not linear.

The mean gray matter value is 1781 and the mean white matter value is 2335. Thus we can see that the bias in the body coil is fairly negligible (0.31% in gray matter regions and 0.10% in white matter regions). There are a number of interesting effects to observe. The bias for the multiple surface coil $\hat{f}$ with $\gamma = 0$ is actually larger than the bias in the body coil observation. The reason for this is that we regularize $\hat{b}$ but not $\hat{f}$. Even though we only explicitly penalize curvature in $\hat{b}$, all else being equal, lower values in $\hat{b}$ will reduce the regularization energy. Thus to minimize the energy, $\hat{b}$ has a slight downward bias which forces $\hat{f}$ to have a slight upward bias. In an analogous manner, the bias for the solution with regularization on $\hat{f}$ is lower than the bias for the body coil image. Lower values in $\hat{f}$ decrease the regularization energy for $\hat{f}$, and this forces $\hat{f}$ to become smaller.

In both of the Brey-Narayana corrections, we see the gray matter bias becomes smaller while the white matter bias becomes larger. The reason for this is that on gray/white boundaries, Brey-Narayana underestimates the bias field on the white mat-

ter side (because it includes some of the gray matter intensities in its estimate) and overestimates the bias field on the gray matter side (because white matter intensities are incorporated into the estimate). Hence the final intrinsic image estimate will overestimate white matter intensities and underestimate gray matter intensities.

The ground truth segmentation results were shown in Figure 4.35. Overall 9814 pixels were classified as being white matter and 5466 pixels were classified as being gray matter. This means that classification with $\boldsymbol{y}_{\mathrm{B}}$ results in gray matter error of 25.5% and white matter error of 8.4%. The various bias correction techniques reduce this rate to 5.9–6.9% in the gray matter and 1.9–2.2% in the white matter. We do not display the segmentation maps for the different bias correction techniques because they look largely the same as the ground truth with minor errors interspersed throughout. To obtain the segmentation errors, we thresholded the various images with the same value used to obtain the ground truth. We define an error as a pixel that is either included when it should not have been, or not included when it should have been. This means that a pixel that is labelled as gray matter in the ground truth segmentation but white matter in the corrected image segmentation is counted as an error for both the gray matter and the white matter. Even though there are fewer gray matter pixels than white matter pixels, all of the methods had higher misclassification rates for the gray matter. This is because many of the white matter pixels are far away from other tissue types and are less affected by partial volume effects.

We are only able to reduce the segmentation errors from Brey-Narayana by about 13%. Note that most of this reduction comes from our superior bias field estimates rather than the anisotropic filtering. The reason why the regularization on $\hat{\boldsymbol{f}}$ produces a large reduction in the error while only a minimal reduction in the misclassification rate can be seen in Figure 4.37. As we noted earlier, most of the benefits from the anisotropic filtering come far away from edges. Unfortunately, most of the misclassification errors occur near tissue boundaries due to the partial volume blurring. Hence, the ℓ_1 prior on $\hat{\boldsymbol{f}}$ does not help in the segmentation as much as we would have expected.

Dependence on α

We examine the dependence of the mean squared error on α in Figure 4.38. We can see that there is a definite value that minimizes the error, though the trough is not very steep. We see that choosing α too small is much more detrimental than choosing α

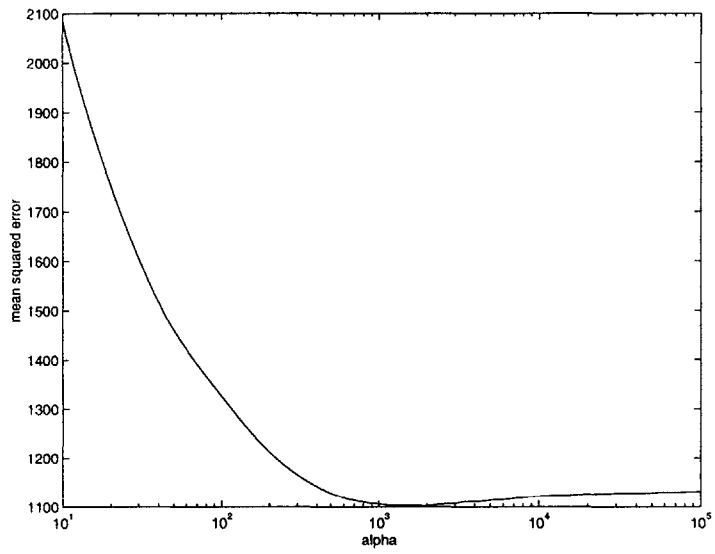


Figure 4.38. Mean squared error as a function of α from the MNI data set.

too large. As $\alpha \to 0$, more noise is admitted into $\hat{b}_k$, and $\hat{f}$ becomes more and more like y_B. So as α gets very small, the mean squared error should approach 14449. The situation for large α is not as dire because the overall curvature in the actual surface coil reception profiles is relatively low because the bias fields actually are smooth and slowly varying. So for the optimal value of α, a small portion of the energy comes from the regularization energy, and most comes from the data fidelity term. Thus the intrinsic image estimates become relatively insensitive to changes in α because they are constrained by the body coil observations.

Dependence on SNR

In Figures 4.39, 4.40, and 4.41, we vary the noise level used to generate the observation images. The lowest noise level corresponds to a SNR of 30 dB in the gray matter and 32 dB in the white matter. The highest noise results in a SNR of -12 dB in the gray matter and -10 dB in the white. The latter noise levels result in completely unusable images. We evaluate three different bias correction techniques: our method with and without regularization on $\hat{f}$, and Brey-Narayana without the body coil in the reconstruction.

In Figure 4.39, we plot the mean squared error as a function of noise level. Brey-

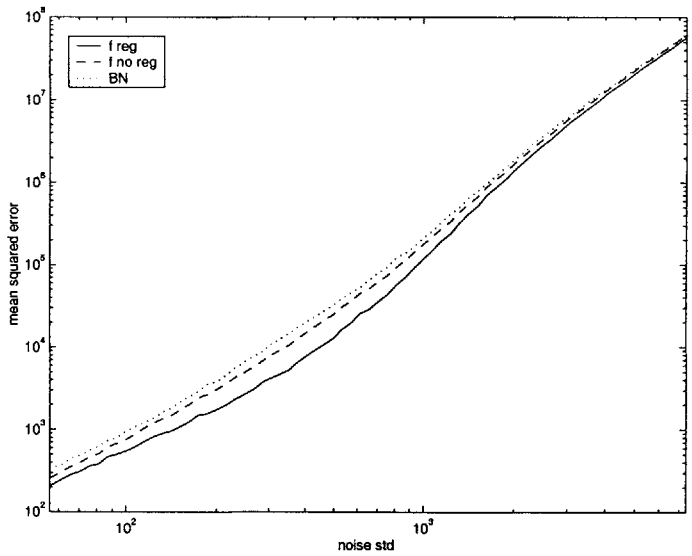


Figure 4.39. Mean squared error as a function of noise level in the MNI data set.

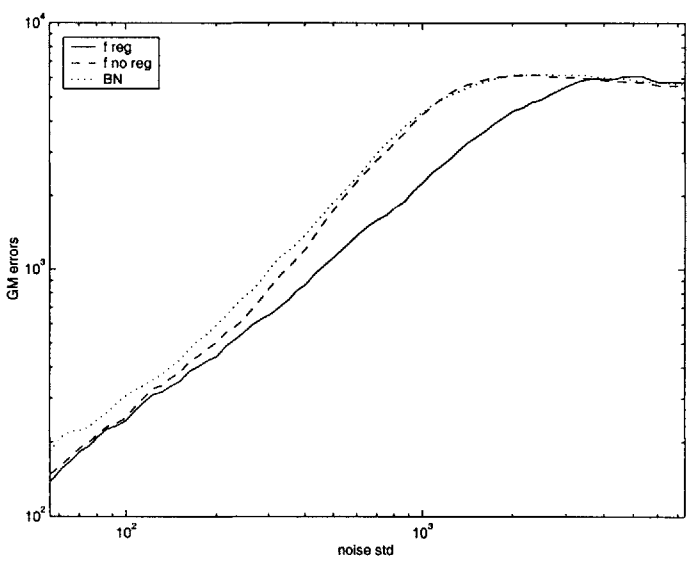


Figure 4.40. Gray matter segmentation errors as a function of noise level in the MNI data set.

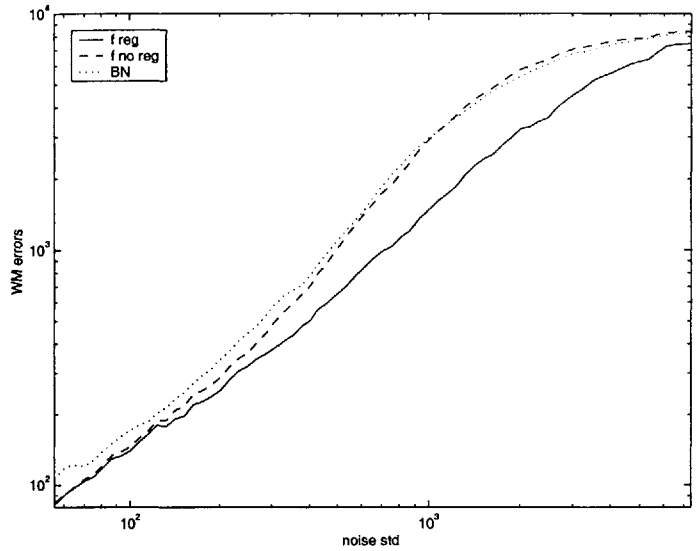


Figure 4.41. White matter segmentation errors as a function of noise level in the MNI data set.

Narayana is always seen to produce worse results than our method with $\gamma = 0$, and the latter is always worse than our method with $\gamma \neq 0$. As the noise becomes very small or very large, all three methods produce similar results. In the middle region (from approximately 100 to 1000 which corresponds to SNRs between 6 dB and 26 dB), regularization on $\hat{\boldsymbol{f}}$ opens a fairly significant performance advantage. The reason for this behavior is that when there is little noise, all methods can produce good results. When there is a lot of noise, there is little information available to perform good estimation, so everyone does poorly. In the middle, the problem is hard, but not too hard, and superior techniques provide superior results.

We observe slightly different behavior in Figures 4.40 and 4.41 where we plot the misclassification rates in gray matter and white matter respectively. While the largest gains in error occur around a standard deviation of 300, the largest gains in classification error occur around a standard deviation of 1000. At the highest noise levels, the misclassification rates level off. In this region, the classifiers basically resort to guessing and they have a 50/50 chance of being right.

4.5 Summary of Results

The prostate was probably the example where our algorithm showed the most dramatic gains. Comparison between our intrinsic image estimates with the original body coil and surface coil images showed a large gain in both SNR and intensity homogeneity. We were able to successfully correct both T_2-*weighted* and T_1-*weighted* images, even though we lacked body coil observations for the latter. This lack of T_1-*weighted* body coil images resulted in a larger number of obvious artifacts in the T_1-*weighted* intrinsic image estimates. We demonstrated our algorithm on three prostate sets of varying quality. Not surprisingly, the best results were obtained from the set with the highest quality observation images. Even with the poorer quality data sets, the regularization ensured that our bias field estimates ignore the motion artifacts. Unfortunately, the quality of the intrinsic image estimates was limited by the quality of the observations. Thus one set resulted in reconstructed images that were noisy and somewhat blurry while another had large motion artifacts that spoiled the visual appearance.

Our bias correction algorithm was also quite successful on the cardiac images. We demonstrated the utility with and without regularization on $\hat{\boldsymbol{f}}$. The regularization provided superior noise behavior, but the result was not as dramatic as in the prostate because the heart SNR is much higher. We investigated the usage of both gradient and Laplacian regularizers on $\hat{\boldsymbol{b}}$ and found that gradient regularization tends to overconstrain the bias field estimates which visibly affects the final intrinsic image estimates. We demonstrated the flexibility of our method by processing the multiple surface coil images in one composite surface coil image and also in four separate images.

The results for the brain were largely what we would expect based on the previous prostate and heart examples. The main difference was the nine observation images available for each slice: one body coil GRE image, four surface coil GRE images, and four surface coil FLAIR images. Note that if we had individual surface coil data for the prostate, we could have up to eleven observations per slice. Our algorithm was able to fuse these observations to generate two intrinsic image estimates and four bias field estimates. An additional feature of the brain examples was the ability to get a more concrete sense of the quality of the bias corrections. In previous examples, we could only look at the images and visually compare them with our body coil and surface coil observations. With the brain images, we could perform threshold-based

segmentations to see how homogeneous the reconstructions really were. Overall we found the corrections to be fairly good (and certainly superior to using the raw surface coil data), though the quality was limited by partial volume effects, noise, and body coil inhomogeneities.

The results with using the MNI brain phantom tracked our experiences with real data. We generated quantitative error figures for different bias correction methods and found that our method with regularization on $\hat{f}$ performed the best and Brey-Narayana performed the worst, but the differences were fairly minor. We also investigated the performance as a function of SNR and found that we generally outperform Brey-Narayana, and the largest outperformance occurs for SNR ranges from 6 dB to 26 dB. This is a positive result because this is also the typical range of SNR we observe in real data.

Chapter 5

Conclusion

We have presented a novel bias correction method that relies on surface coil images to provide superior noise characteristics and body coil images to help constrain the space of possible solutions. The main contribution is a flexible variational framework that unifies bias correction and anisotropic edge-preserving filtering to produce results that are both homogeneous and have high SNR. We used ℓ_2 norms to ensure that our bias field and intrinsic image estimates conformed to our observations, and we used regularization to help mitigate the effects of the noise on our estimates. A key contribution is the ability to fuse together observation images from multiple surface coils and multiple pulse sequences. Compared to similar techniques that use body coil and surface coil images, we provide a more rigorous formalism and superior results to Brey and Narayana [12] and our method is not limited by the parametric representations of Pruessmann *et al.* [59] and Lai and Fang [41].

The energy functional we construct is difficult to minimize in terms of both the bias field and the intrinsic image but is easy to do for just one of those quantities. Hence we use coordinate descent to create subproblems that are easier to solve than the main problem. We then iterate between f-steps and b-steps until our solutions converge. We find that the b-step results in a quadratic optimization problem which can be solved exactly through matrix inversion or iteratively using methods such as conjugate gradient. The f-step results in a ℓ_p regularization problem. We use half-quadratic optimization to convert the nonlinear optimization problem into a series of quadratic optimization problems which can be solved using conjugate gradient. We embed our solvers into a multigrid framework to increase both speed and robustness.

We found the largest gains using our algorithm were achieved for the prostate. This is because the prostate images had the highest noise and the largest intensity inhomogeneities. We also presented results for heart and brain data where we had the

separate data for the individual surface coils. We then used the Montreal Neurological Institute brain phantom to acquire quantitative results on realistic-looking artificial data.

5.1 Future Research Directions

There are a number of additional things that can be accomplished within our basic bias correction framework. First is to apply the method to additional examples. Breast and spine MR also make use of surface coils with fairly significant intensity inhomogeneities. No major modifications to the algorithm should be needed to use these new applications. It would also be interesting to test our algorithm on brain data acquired using a head coil. The bias field in these images is not very severe, but it is enough to prevent ideal results for statistical segmentation. We would also like to obtain prostate data with separate images for each of the surface coils. As we noted earlier, the composite surface coil image limits our ability to effectively regularize the bias field estimates in the regions where the pelvic phased array dominates.

We would like to implement the solver to operate on full 3D volumes. There is no real technical impediment to do so. The main concern is computational speed and memory consumption. The current memory footprint is quite small, but using full $512 \times 512 \times 400$ volumes could be problematic.

An important feature would be automatic or semi-automatic methods to choose optimal settings for our regularization parameters α and γ. Currently we are able to effectively estimate the noise variances, but the regularization strength must be chosen through visual inspection and trial and error.

Better handling of the Rician noise would be desirable. Currently we model the noise as IID Gaussian with zero mean. This ignores the bias (which can be quite large in the prostate data set) and the dependence of the noise variance on the SNR. One crude possibility to help reduce the effects would be to first run our bias correction algorithm with the IID Gaussian assumption. We could then estimate the SNR at each pixel from the corrected image and use this to estimate the true bias and noise variance at each pixel.

For brain segmentations, we found that our segmentation quality may be limited by the homogeneity of the body coil. We could use the main idea of Wells *et al.* [85] where we alternate segmentation and bias correction steps. The segmentation can help

us estimate the reception profiles of both the surface coils and the body coil.

We use multigrid to force our algorithm into a multiresolution framework and we use a frequency-selective penalty to regularize our bias field estimates. This can be accomplished analogously with wavelets. The regularization is then placed directly on the wavelet coefficients rather than derivatives of the bias field.

We can also attempt to improve our probabilistic modeling of both the bias field and the intrinsic image. We can model the bias field as a Markov random field (MRF) to provide better global coupling and more flexible representations without excessively burdensome computation. We can use the MRF ideas of Geman and Geman [28] to provide better explicit modeling of the edges in the intrinsic image.

Appendix A

Linear Algebra

We will only cover the aspects of linear algebra and vector calculus that we use in this work. For a more comprehensive treatment, see Strang [70] or Horn and Johnson [34]. Let there be a function $f(\boldsymbol{x}) : \Re^n \rightarrow \Re$. The gradient is a differential operator $\nabla : \Re \rightarrow \Re^n$ on f:

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix}^{\mathrm{T}} . \tag{A.1}$$

The Hessian is a nxn second derivative operator with

$$[\mathcal{H}]_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} . \tag{A.2}$$

The Laplacian is another second derivative operator:

$$\nabla^2 = \triangle = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \ldots + \frac{\partial^2}{\partial x_n^2} . \tag{A.3}$$

The directional derivative is the derivative of f in the direction of $\boldsymbol{x}_0$. This can be written as:

$$f'(\boldsymbol{x}; \boldsymbol{x}_0) = \lim_{\delta \rightarrow 0} \frac{f(\boldsymbol{x} + \delta \boldsymbol{x}_0) - f(\boldsymbol{x})}{\delta} = \nabla f(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{x}_0 . \tag{A.4}$$

A matrix is said to be positive semi-definite (PSD) if and only if:

1. All eigenvalues are non-negative
2. $\forall \boldsymbol{x} \in \Re^n, \boldsymbol{x}^T M \boldsymbol{x} \geq 0$
3. All determinants of upper-left submatrices are non-negative

All three statements are equivalent. A matrix is positive definite (PD) if all inequalities are strict.

The inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{y} . \tag{A.5}$$

The inner product weighted by $\boldsymbol{W}$ is

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{W}} = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{y} \ . \tag{A.6}$$

We can then define vector norms using inner products:

$$\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle \tag{A.7}$$

$$\|\boldsymbol{x}\|_{\boldsymbol{W}}^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\boldsymbol{W}} \tag{A.8}$$

$$\tag{A.9}$$

Matrix decompositions are methods to write a given matrix in terms of a few matrices to give more insight into the structure of the original matrix. The simplest is the LU decomposition:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} \tag{A.10}$$

where $\boldsymbol{U}$ is the result of Gaussian elimination on $\boldsymbol{A}$ (referred to as row echelon form) and $\boldsymbol{L}$ is the product of the elementary row operation matrices. If we put $\boldsymbol{U}$ into reduced row echelon form (with 1's on the diagonal), we can write this as $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{U}$ where $\boldsymbol{D}$ is a diagonal scaling matrix.

The eigenvector decomposition for an $n \times n$ matrix $\boldsymbol{M}$ highlights the role that the eigenvalues and eigenvectors play:

$$\boldsymbol{M} = \boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}^{-1} \ . \tag{A.11}$$

The columns of $\boldsymbol{S}$ are the eigenvectors of $\boldsymbol{M}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{M}$ along the diagonal. This decomposition is valid for matrices that have a full set of linearly-independent eigenvectors. When this decomposition exists, it tells us that the action of the matrix on a vector $\boldsymbol{x}$ is to: first, transform $\boldsymbol{x}$ into the vector space spanned by the eigenvectors of $\boldsymbol{M}$; second, scale each component by the appropriate eigenvalue; third, transform back into the original vector space.

When $\boldsymbol{M}$ is symmetric, all of its eigenvalues are real and all of its eigenvectors are orthogonal to each other. Hence we can write $\boldsymbol{S}$ as an orthogonal matrix $\boldsymbol{Q}$ and $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{\mathrm{T}}$:

$$\boldsymbol{M} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}} \ . \tag{A.12}$$

We note that positive semi-definite matrices can be written as:

$$\boldsymbol{M} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} \ . \tag{A.13}$$

This can be seen from (A.12). When $\boldsymbol{M}$ is PSD, all entries of $\boldsymbol{\Lambda}$ are non-negative. Hence $\boldsymbol{\Lambda}$ has a real square root and we can rewrite (A.12) as follows:

$$\boldsymbol{M} = (\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2})(\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2})^{\mathrm{T}} \ , \tag{A.14}$$

and we can see $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}$.

This suggests that in an LU decomposition, $\boldsymbol{H} = \boldsymbol{L}$ and $\boldsymbol{U} = \boldsymbol{H}^{\mathrm{T}} = \boldsymbol{L}^{\mathrm{T}}$. This is known as the Cholesky decomposition:

$$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \ . \tag{A.15}$$

Finally, for an arbitrary $m \times n$ matrix $\boldsymbol{M}$, there is the singular value decomposition (SVD). For simplicity, we will only consider matrices with $m \geq n$. For a matrix $\boldsymbol{A}$ with $n \geq m$, the following results may be applied to $\boldsymbol{A}^{\mathrm{T}}$ and $\boldsymbol{A}$ is just that decomposition transposed.

The SVD is based on decompositions of the two PSD matrices we can easily form from $\boldsymbol{M}$: $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}$ and $\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}$. We can decompose both of these matrices using the $\boldsymbol{Q}\boldsymbol{\Lambda}$ decomposition:

$$\begin{aligned} \boldsymbol{M}\boldsymbol{M}^{\mathrm{T}} &= \boldsymbol{Q}_1\boldsymbol{\Lambda}_1\boldsymbol{Q}_1^{\mathrm{T}} \\ \boldsymbol{M}^{\mathrm{T}}\boldsymbol{M} &= \boldsymbol{Q}_2\boldsymbol{\Lambda}_2\boldsymbol{Q}_2^{\mathrm{T}} \ . \end{aligned}$$

So we see that we obtain *two* sets of orthogonal basis vectors: one for $\Re^n$ and one for $\Re^m$. It turns out that the n largest eigenvalues of $\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}$ are the same as the n eigenvalues of $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}$ (the remainder are zero due to the fact that the rank of $\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}$ is at most n).

This leads to the following decomposition:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{A.16}$$

where $\boldsymbol{U} = \boldsymbol{Q}_1$ is $m \times m$, $\boldsymbol{\Sigma}$ is $m \times n$, and $\boldsymbol{V} = \boldsymbol{Q}_2$ is $n \times n$. $\boldsymbol{\Sigma}$ can be written as the following block matrix:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Lambda}_2^{1/2} \\ 0_{m-n\times n} \end{pmatrix} \tag{A.17}$$

The entries along the diagonal of $\boldsymbol{\Sigma}$ are known as the singular values.

We can verify this result by noting that $\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{Q}_1\boldsymbol{\Lambda}_1\boldsymbol{Q}_1^{\mathrm{T}}$ and $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{Q}_2\boldsymbol{\Lambda}_2\boldsymbol{Q}_2^{\mathrm{T}}$.

Bibliography

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions.* Dover, 10th edition, 1974.

[2] American Cancer Society. http://www3.cancer.org/cancerinfo.

[3] L. Armijo. Minimization of functions having continuous partial derivatives. *Pacific J. Math.*, 16:1–3, 1966.

[4] L. Axel. Surface coil magnetic resonance imaging. *J. of Comp. Asst. Tomography*, 8:381–384, 1984.

[5] L. Axel, J. Costantini, and J. Listerud. Intensity correction in surface-coil MR imaging. *Am. J. Roentgenology*, 148:418–420, 1987.

[6] M. Belge, M. E. Kilmer, and E. L. Miller. Wavelet domain image restoration with adaptive edge-preserving regularization. *IEEE Trans. Imag. Proc.*, 9(4):597–608, 2000.

[7] M. Belge, M. E. Kilmer, and E. L. Miller. Efficient selection of multiple regularization parameters in a generalized L-curve framework. *Inv. Prob.*, 18(4):1161–1183, 2002.

[8] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, 1999.

[9] M. Bhatia, W. C. Karl, and A. S. Willsky. A wavelet-based method for multiscale tomographic reconstruction. *IEEE Trans. Med. Imag.*, 15(1):92–101, 1996.

[10] F. Bloch. Nuclear induction. *Phys. Rev.*, 70:460–474, 1946.

[11] C. Brechbühler, G. Gerig, and G. Szekely. Compensation of spatial inhomogeneity in MRI based on a parametric bias estimate. In *Proceedings of VBC '96*, pages 141–146. Springer-Verlag, 1996.

[12] W. W. Brey and P. A. Narayana. Correction for intensity falloff in surface coil magnetic resonance imaging. *Med. Phys.*, 15(2):241–245, 1988.

[13] W. L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[14] B. H. Brinkmann, A. Manduca, and R. A. Robb. Optimized homomorphic unsharp masking for MR grayscale inhomogeneity correction. *IEEE Trans. Med. Imag.*, 17(3):161–171, 1998.

[15] I. Chan. Segmentation of prostate tumors in multi-channel MRI. Master's thesis, MIT, 2002.

[16] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Imag. Proc.*, 6(2):298–310, 1997.

[17] D. K. Cheng. *Field and Wave Electromagnetics.* Addison-Wesley, New York, 2nd edition, 1989.

[18] Z. H. Cho, J. P. Jones, and M. Singh. *Foundations of Medical Imaging.* Wiley, New York, 1993.

[19] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Trans. Med. Imag.*, 17(3):463–468, June 1998.

[20] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

[21] B. M. Dawant, A. P. Zijdenbos, and R. A. Margolin. Correction of intensity variations in MR images for computer-aided tissue classification. *IEEE Trans. Med. Imag.*, 12:770–781, 1993.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, 39:1–38, 1977.

[23] W. A. Edelstein, J. F. Schenck, H. R. Hart, C. J. Hardy, T. H. Foster, and P. A. Bottomley. Surface coil magnetic resonance imaging. *J. Am. Med. Assoc.*, 253:828, 1985.

[24] T. K. Foo, C. E. Hayes, and Y. W. Kang. Analytical model for the design of RF resonators for MR body imaging. *Mag. Res. Med.*, 21:165–177, 1991.

[25] N. D. Gelber, R. L. Ragland, and J. R. Knorr. Surface coil MR imaging: utility of image intensity correction filter. *Am. J. Roentgenology*, 162:695–697, 1994.

[26] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(3):367–383, 1992.

[27] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Imag. Proc.*, 4(7):932–946, July 1995.

[28] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6:721–741, 1984.

[29] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins, Baltimore, 1996.

[30] R. Guillemaud. Uniformity correction with homomorphic filtering on region of interest. In *Proceedings of the 1998 International Conference on Image Processing*, pages 872–875. IEEE, 1998.

[31] R. Guillemaud and M. Brady. Estimating the bias field of MR images. *IEEE Trans. Med. Imag.*, 16(3):238–251, 1997.

[32] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, 34:561–580, 1992.

[33] J. Haselgrove and M. Prammer. An algorithm for compensation of surface-coil images for sensitivity of the surface coil. *Mag. Res. Imag.*, 4:469–472, 1986.

[34] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[35] J. Huang and D. Mumford. Statistics of natural images and models. In *Proc. Conf. Comp. Vision and Patt. Recog.*, pages 541–547, 1999.

[36] F. V. Jensen. *Bayesian Networks and Decision Graphs.* Springer, New York, 2001.

[37] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice-Hall, 2000.

[38] W. C. Karl. *Handbook of Image and Video Processing*, chapter 3.6, pages 141–160. Academic Press, San Diego, 2000. editor Bovik, A.

[39] R. Kruger. Analysis and comparison of the signal difference to noise ratio (SDNR), signal difference (SD), and the signal to noise ratio (SNR): Evaluating the suitability of the SD and SDNR as MRI quality control parameters. In *Proceedings of the 22nd Annual EMBS International Conference*, pages 2157–2160. IEEE, 2000.

[40] R. K.-S. Kwan, A. C. Evans, and G. B. Pike. MRI simulation-based evaluation of image-processing and classification methods. *IEEE Trans. Med. Imag.*, 18(11):1085–1097, Nov. 1999.

[41] S.-H. Lai and M. Fang. Intensity inhomogeneity correction for surface-coil MR images by image fusion. In R. Hamid, A. Zhu, and D. Zhu, editors, *Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, pages 880–887. CSREA Press, 1998.

[42] S.-H. Lai and M. Fang. A new variational shape-from-orientation approach to correcting intensity inhomogeneities in MR images. In *Proceedings of Workshop on Biomedical Image Analysis*, pages 56–63. IEEE, 1998.

[43] S. K. Lee and V. M. W. Post-acquisition correction of MR inhomogeneities. *Mag. Res. Med.*, 36:275–286, 1996.

[44] B. Likar, J. B. A. Maintz, M. A. Viergever, and F. Pernus. Retrospective shading correction based on entropy minimization. *J. of Microscopy*, 197:285–295, 2000.

[45] B. Likar, M. A. Viergever, and F. Pernus. Retrospective correction of MR intensity inhomogeneity by information minimization. In *Med. Img. Comp. and Comp.-Asst. Intervention 2000*, Lecture Notes in Computer Science, pages 375–384, 2000.

[46] G. P. Liney, L. W. Turnbull, and A. J. Knowles. A simple method for the correction of endorectal surface coil inhomogeneity in prostate imaging. *J. Mag. Res. Imag.*, 8(4):994–997, 1998.

[47] S. P. Liou, Chiu, A. H., and R. C. Jain. A parallel technique for signal-level perceptual organization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13:317–325, 1991.

[48] R. B. Lufkin, T. Sharpless, B. Flannigan, and W. Hanafee. Dynamic-range compression in surface-coil MRI. *Am. J. Roentgenology*, 147:379–382, 1986.

[49] J. F. Martin, P. Hajek, L. Baker, V. Gylys-Morin, R. Fitzmorris-Glass, and R. R. Mattrey. Inflatable surface coil for MR imaging of the prostate. *Radiology*, 167:268–270, 1988.

[50] C. R. Meyer, P. H. Bland, and J. Pipe. Retrospective correction of intensity inhomogeneities in MRI. *IEEE Trans. Med. Imag.*, 14(1):36–41, 1995.

[51] P. G. Morris. *Nuclear Magnetic Resonance Imaging in Medicine and Biology.* Clarendon Press, 1986.

[52] S. E. Moyher, D. B. Vigneron, and S. J. Nelson. Surface coil MR imaging of the human brain with an analytic reception profile correction. *J. Mag. Res. Imag.*, 5:139–144, 1995.

[53] S. E. Moyher, L. . Wald, S. J. Nelson, D. Hallam, W. P. Dillon, D. Norman, and D. B. Vigneron. High resolution T2-weighted imaging of the human brain using surface coils and an analytical reception profile correction. *J. Mag. Res. Imag.*, 7(3):512–517, 1997.

[54] M. Nikolova and M. Ng. Comparison of the main forms of half-quadratic regularization. In *Proceedings of the IEEE ICIP 2002*, pages 349–352. IEEE, 2002.

[55] R. D. Nowak. Wavelet-based Rician noise removal for magnetic resonance imaging. *IEEE Trans. Imag. Proc.*, 8(10):1408–1419, 1999.

[56] A. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing.* Prentice-Hall, 1999.

[57] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 4th edition, 2001.

[58] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[59] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: Sensitivity encoding for fast MRI. *Mag. Res. Med.*, 42:952–962, 1999.

[60] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Phys. Rev.*, 69:37–38, 1946.

[61] S. Rice. Mathematical analysis of random noise. *Bell Sys. Tech. J.*, 23:282–332, July 1944.

[62] P. B. Roemer, W. A. Edelstein, C. E. Hayes, S. P. Souza, and O. M. Mueller. The NMR phased array. *Mag. Res. Med.*, 16:192–225, 1990.

[63] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[64] M. D. Schnall, Y. Imai, J. Tomaszewski, H. M. Pollack, R. E. Lenkinski, and H. Y. Kressel. Prostate cancer: Local staging with endorectal surface coil MR imaging. *Radiology*, 178:797–802, 1991.

[65] M. D. Schnall, R. E. Lenkinski, H. M. Pollack, Y. Imai, and H. Y. Kressel. Prostate: MR imaging with an endorectal surface coil. *Radiology*, 172:570–574, 1989.

[66] J. Sijbers, A. J. den Dekker, P. Scheunders, and D. Van Dyck. Maximum-likelihood estimation of Rician distribution parameters. *IEEE Trans. Med. Imag.*, 17(3):357–361, June 1998.

[67] M. Singh and M. NessAiver. Accurate intensity correction for endorectal surface coil MR imaging of the prostate. *IEEE Trans. Nucl. Sci.*, 40:1169–1173, 1993.

[68] J. G. Sled, A. P. Zijdenbos, and A. C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Trans. Med. Imag.*, 17:87–97, 1998.

[69] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[70] G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, Wellesley, MA, 1993.

[71] M. Styner, C. Brechbühler, G. Székely, and G. Gerig. Parametric estimate of intensity inhomogeneities applied to MRI. *IEEE Trans. Med. Imag.*, 19(3):1–14, 2000.

[72] C. M. C. Tempany, editor. *The Male Pelvis.* Magnetic Resonance Imaging Clinics of North America. W.B. Saunders Co., 1996.

[73] A. Tikhonov and V. Arsenin. *Solution of Ill-Posed Problems.* W.H. Winston, Washington, D.C., 1977.

[74] A. Tsai, A. Yezzi, W. M. Wells III, C. Tempany, D. Tucker, A. C. Fan, W. E. L. Grimson, and A. S. Willsky. Model-based curve evolution technique for image segmentation. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 463–468. IEEE, 2001.

[75] A. Tsai, A. Yezzi, W. M. Wells III, C. Tempany, D. Tucker, A. C. Fan, W. E. L. Grimson, and A. S. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Trans. Med. Imag.*, 22(1):1–15, Jan. 2003.

[76] R. P. Velthuizen, J. J. Heine, A. B. Cantor, H. Lin, L. M. Fletcher, and L. P. Clarke. Review and evaluation of MRI nonuniformity corrections for brain tumor response measurements. *Med. Phys.*, 25(9):1655–1666, 1998.

[77] P. Viola. *Alignment by Maximization of Mutual Information.* PhD thesis, MIT, 1995.

[78] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comp.*, 17(1):227–238, 1996.

[79] E. A. Vokurka, N. A. Thacker, and A. Jackson. A fast model independent method for automatic correction of intensity nonuniformity in MRI data. *J. Mag. Res. Imag.*, 10:550–562, 1999.

[80] L. L. Wald, L. Carvajal, S. E. Moyher, S. J. Nelson, P. E. Grant, et al. Phased array detectors and an automated intensity-correction algorithm for high-resolution MR imaging of the human brain. *Mag. Res. Med.*, 34:433–439, 1995.

[81] L. Wang, H.-M. Lai, G. J. Barker, D. H. Miller, and P. S. Tofts. Correction for variations in MRI scanner sensitivity in brain studies with histogram matching. *Mag. Res. Med.*, 39:322–327, 1998.

[82] S. Webb, editor. *The Physics of Medical Imaging.* Adam Hilger, 1988.

[83] R. Weinstock. *Calculus of Variations, With Applications to Physics and Engineering.* Dover, 1974.

[84] Y. Weiss. Deriving intrinsic images from image sequences. In *Proceeedings of the IEEE Internation Conference on Computer Vision*, 2001.

[85] W. M. Wells, W. E. L. Grimson, R. Kikinis, and F. Jolesz. Adaptive segmentation of MRI data. *IEEE Trans. Med. Imag.*, 15(4):429–442, 1996.

[86] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley & Sons, New York, 1992.

[87] A. S. Willsky, G. W. Wornell, and J. H. Shapiro. Stochastic processes, detection and estimation. Course notes for MIT 6.432.

[88] T. Z. Wong, S. G. Silverman, J. R. Fielding, et al. Open-configuration MR imaging, intervention, and surgery of the urinary tract. *Urologic Clinics of N. Amer.*, 25:113–122, 1998.

[89] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the Expectation-Maximization algorithm. *IEEE Trans. Med. Imag.*, 20(1):45–57, 2001.